统计学与R读书笔记(第三版)

徐俊晓

戊子 鼠年 十二月初二
（西元 2008年12月28日)

Contents

版权声明

警告

本文档是一个非正式的阅读笔记. 大部分内容来自其它资料. 虽然尽量注明参考文献与出处, 但是并未严格一一标明来源.

里面的R引用和实现大部分来自个人的尝试和理解, 因此会包含很多错误与不足之处. 敬请批评指正.

请谨慎阅读与引用!!! 由于某些部分的不完整, 强烈建议与其它正式资料一起阅读. 本人不对任何由此文档引发的后果负责.

感谢

感谢所有对R的发展作出贡献的人

第三版序

一切有为法, 如梦幻泡影, 如露亦如电, 应作如是观

若人言, 如来有所说法, 即为谤佛, 不能解我所说故

——摘自《金刚般若波罗密经》

演说：释迦牟尼

记录：阿难等（尊者）

翻译：鸠摩罗什（东晋后秦高僧）

般若波罗密, 即智慧到彼岸

增加

- 增加了"时间序列"部分. 包括
  - AR
  - MA
  - ARMA
  - ARIMA
  - ARCH
  - 谱分析
  - 小波分析
- 重写了"回归与方差分析" 部分
- 流行病学部分增加了筛选检验的一般概念和ROC曲线
- "基本统计分析" 部分"估计" 增加了"矩法" 和"极大似然法"

- ”杂项” 部分增加了 ”马尔可夫链与生物学”
- ”R基础与数学运算” 部分增加 ”运算符号” 与 ”复数基本运算” ”方程式求根” ”优化”

删除

- 删除 ”使用anova()比较多个模型”(可能解释有错误)

修改

- 对 ”R基础与数学运算”, ”基本统计分析”, ”回归与方差分析” 部分的结构顺序, 章节题目等做了较大变动
- 修正了 ”基本统计分析” 中 ”R的统计模型概述” 的公式格式错误, 并转移到 ”回归和方差分析”
- 修正了 ”方差不等的独立样本t检验” d'近似公式(原来有误)
- 修正了若干小错误和格式错误

徐俊晓

2008.12.28

第二版序

本次增加部分主要参考《统计建模与R软件》[15]. 部分参考 ”生物数学”[8]. 其它资源见正文.

增加

- "R 基础" 部分增加了 "数组与矩阵运算", "在 python 中调用 R(rpy2)".
- "回归与方差分析" 部分增加了 "逐步回归", "回归诊断"
- 增加了 "判别,聚类,因子分析等" 部分

改变

- 将 "绘图" 部分转入到 "R 基础".
- "广义线性 (Generalized Linear)模型" 和 "非线性回归与非线性最小平方" 两章补充了一点内容, 放到回归与方差分析(原线性回归与方差分析)
- "数据变换" 放入 "基本统计分析" 部分
- 修正了若干格式问题.

徐俊晓

2008.11.28

序

这是我学习生物统计学和R的笔记. 并不是一个系统介绍概率论与统计学和R应用的东西, 开始只是把用到的R的相关东西记下来，以免忘记。后来看看记的还不少，又想系统学习统计学，就整理了一下。所以如果知道问题的解决方法，直接看命令的用法就可以了。公式什么的是为了参考方便加入的，随意性很强, 对希望系统了解的读者说声抱歉。

全部笔记除了第二部分“R基础”外，统计学部分主要参考 Bernard Rosner 的《生物统计学基础（Fundamentals of Biostatistics）》第五版 [11]。

孙尚拱先生说：我们的医学统计教师及流行病学教师们，如都能认真地阅读此书，我国的医学统计教学及科研水平必定会有大的提高。阅毕此书，深有同感。

由于本人比较懒散，故有的内容记录详细, 有的则简单。绝大部分内容是从参考文献得来, 开始不太在意参考文献的记录, 后来尽量加入了参考文献出处. 若没有注明, 请原文献作者谅解.

图：由于latex水平比较差(且本人很懒)，所有的图都没有放上来。

由于本人水平所限，其中肯定很多错误与不足的地方, 希望大家批评指正。尤其统计学的高级部分，象多元线性回归，广义线性回归（logistic回归，poisson回归，负二项回归），多元数据分析（因子分析，主成分分析，判别分析，聚类分析，典型相关分析）等部分更是似懂非懂，心有余而力不足，希望大家阅读的时候注意鉴别，如果有机会，以后可能会补充上述内容。笔记中TODO标记大多属于这种情况.

正当此笔记基本完成时, 看到一本书: 《统计建模与R软件》，薛毅，陈立萍编著，清华大学出版社。心想, 我想做的已经有人完成了, 而且非常之好, 作者水平也不是我所能比的, 后悔没有早点看到此书. 不过, 如果早看到了, 估计也就没有这篇笔记了.

愈写愈觉得自己所知其实有如沧海一粟, 不禁心生望洋之叹. 希望大家能够从这个笔记有所收获。祝大家学习进步。

徐俊晓

2008.10.01

Part I

R基础与数学运算

此部分是R中的数据结构，语法等语言问题的描述，主要是平时遇到问题的一个汇总，虽然后来经过整理和添加，但是并不是一个系统介绍R的部分。若想系统了解R的语法和其它用法，请读者参考R网站的其它文档，这方面的文档是比较多的。R自带的帮助也是很不错的。主要参考了《simpleR》《R语言简介》《R for beginners》中文版，这几个都不错。

Chapter 1

环境相关

1.1 概述

R 的网站: http://cran.r-project.org/

进入网站, 点击左边的 Task Views, 浏览你需要的功能在哪个包里可以找到

Documentation 下面好多资料供参考.

安装位置在 /usr/share/R/ , 文档也在下面. 不过只是base的.

使用 google 的高级搜索, 站内搜索会更好

1.2 寻求帮助

```
> ?mean
> help(mean)
> help.search("mean")
> apropos(mean) # 或者  apropos("mean")
[1] "kmeans"         "weighted.mean"   "mean"           "mean.data.frame"
[5] "mean.Date"      "mean.default"    "mean.difftime"   "mean.POSIXct"
```

```
[9] "mean.POSIXlt"
```

,
查看，使用 data()

1.2.1 查看所有可用的包

使用 library()

1.2.2 查看某个包的信息

help(package=”xxx”)

1.2.3 查看当前调入内存的包

```
> search()
 [1] ".GlobalEnv"          "package:HSAUR"       "package:scatterplot3d"
 [4] "package:MASS"        "package:lattice"      "package:stats"
 [7] "package:graphics"     "package:grDevices"    "package:utils"
[10] "package:datasets"     "package:methods"      "Autoloads"
[13] "package:base"
```

1.2.4 查看和导入R中预置的数据

查看所有预先提供的数据: data() 查看base包所有预先提供的数据，使用 data(package=“base”)

载入数据，使用data('dataset name')，引号可以不加

1.2.5 查看当前环境下的变量

ls()

1.3 安装,删除非二进制包

install.packages("JGR",dep=TRUE)

有时候, 需要 gcc, g++, gfortran 等, 请单独安装.

coin 包, 需要预先安装 refblas3 refblas3-dev 等

外部手工安装, 例如下载了 rgl包 rgl_0.81.tar.gz, 输入命令 sudo R CMD INSTALL rgl_0.81.tar.gz 即可.

删除: remove.packages(utils)

1.4 运行系统命令

在R环境中运行系统命令使用 system() 函数

```
system("ls x*")
```

如果需要保存输出结果为R对象, 加入参数intern=T

```
files <- system("ls x*",intern=T)
```

1.5 R启动时调用的文件和函数

位置初始化文件的路径可以通过环境变量 R_PROFILE 设

置。若R_PROFILE未设置,则默认为R安装目录下面的子目录etc中的Rprofile.site. 此文件包含执行R时的一些自动命令.

.Rprofile文件允许用户定制它们的工作空间，设置不同的起始命令。此文件可以放在任何目录下。如果R在该目录下面被调用，这个文件就会被载入。如果在起始目录中没有.Rprofile，R会在用户主目录下面搜索.Rprofile文件并且调用它(如果它存在的话)。

另外一个可以配置的文件是.RData

.First()函数：.Rprofile，.RData文件中的函数，此函数可以定义自己的设置。例如：

```
> .First <- function() {
  options(prompt="$ ", continue="+\t") # $是提示符
  options(digits=5, length=999) # 定制数值和输出格式
  x11() # 定制图形环境
  par(pch = "+") # 定制数据点的标示符
  source(file.path(Sys.getenv("HOME"), "R", "mystuff.R")) # 个人编写的函数
  library(MASS) # 导入包
}
```

类似的是，如果定义了函数.Last()，它(常常)会在对话结束时执行。一个例子就是

```
> .Last <- function() {
# 一个小的安全措施。
  graphics.off()
# 该吃午饭了？
  cat(paste(date(),"\nAdios\n"))
}
```

1.6 简单数据编辑器

如果X是一个矩阵，命令data.entry(X)将打开一个图形编辑器并且可以通过点击适当的单元格修改数值或者添加新的行或列.

1.7 字符串合并

paste(... , sep="")

cat(...) 合并参数并打印出来

1.8 数字打印位数

```
> old.digits = options("digits") # 保存默认打印字符长度 7
> options(digits=3)
```

Chapter 2

数据

2.1 读写

2.1.1 导入 Execl 格式

(参考 R-data.pdf) 如果能够避免尽量避免直接导入. 因为xls格式复杂, 还可以包含很多数据表.

把xls保存为csv或其他格式(可以使用openoffice 或 gnumeric), 使用 read.delim2 或 read.csv2 导入.

例如, hormone.xls 使用分隔符 "," 保存为 hormone.csv

```
> d=read.csv2("hormone.csv",sep=",")
```

2.1.2 好用的剪切板

使用表格工具, 例如windows的excel, linux下 的openoffice-spreadsheet, gnumeric等打开数据, 复制要选取的部分, 然后在R控制台下输入

```
data<-read.table(file="clipboard",sep="\t",header=T)
```

分隔使用tab

写入数据到剪切板(windows),

```
write.table(data,file="clipboard",sep="\t",col.names=NA)
```

linux用户需要设置选项'pipe("xclip -i", "w")', 没有成功. 参考help(file)

2.1.3 scan()函数-读取大数据

打开很大的数据建议使用scan函数, 因为可以设定数据类型, 而不是读取完毕才检查数据类型的一致性.

例如文件名称为"filename",默认当前路径(即你运行R的路径), 是",分隔的数据, 想读入364行, 每行有5个数, 因为scan()是按行读取的, 矩阵的顺序是列为先的, 所以应该象下面, 先将矩阵设置为5行, 364列, 然后转置即可得到想要的结果

```
data<-t(matrix(scan("fileName",sep=','), 5, 364))
```

2.1.4 导出/保存

2.1.5 向文件写入数据

write.table 可以在文件中写入一个对象，一般是写一个数据框，也可以是其他数据类型, 向量, 矩阵...

write(x, file="data.txt") 简单的的将对象写入文件. 选项 append 缺省为删除已存在的数据.

2.1.6 保存为R格式

要记录一组任意数据类型的对象，我们可以使用命令save(x, y, z, file= "xyz.RData"). 可以使用选项ASCII=TRUE使得数据在不同的机器之间更简易转移. 数据（用R的术语来说叫做工作空间）.

函数save.image()是save(list =ls(all=TRUE) file=".RData")的一个简捷方式.

可以在使用load("xyz. RData")之后被加载到内存中.

2.1.7 重定向输出

```
> sink("record.lis")
```

将输出重定向到文件"record.lis".

```
> sink()
```

让你的输出流重新定向到控制台。

2.1.8 其它格式(SPSS, SAS, Stata and minitab)

包 foreign 提供读取SPSS, SAS, Stata and minitab格式的数据

```
>  library(foreign)
> search()
 [1] ".GlobalEnv"        "package:foreign"   "package:nlme"
 [4] "package:stats"     "package:graphics"  "package:grDevices"
 [7] "package:utils"     "package:datasets"  "package:methods"
```

```
[10] "Autoloads"        "package:base"
> help(read.spss)
```

2.1.9 latex

library(xtable) 可以把矩阵, data.frame 等变换为 latex 格式.

2.2 数据类型

更多参考《R导论》

R操作的实体在技术上来说就是对象(object).

2.2.1 原子类型

R的对象类型包括数值型（numeric），复数型（complex），逻辑型（logical），字符型（character）和原味型（raw）.

2.2.2 NA

参考《statistics with R》

2.2.3 向量

向量必须保证它的所有元素是一样的模式.

向量必须明确属于逻辑型，数值型，复数型，字符型或者原味型。(这里有一个特定的例外是值为"NA"的元素. 实际上NA有好几种类型).

空向量也有自己的模式.

向量对象的类型的包括: 实数, 复数, 逻辑, 字符串. 它们是原子(atomic), 即元素类型一样.

2.2.4 因子

一个因子不仅包括分类变量本身还包括变量不同的可能水平（即使它们在数据中不出现）。因子函数factor用下面的选项创建一个因子：

factor及ordered函数在统计模型中特别有用. 例如将0,1改变为'y', 'n' 也很方便.

```
factor(x, levels = sort(unique(x), na.last = TRUE),
      labels = levels, exclude = NA, ordered = is.ordered(x))
```

levels用来指定因子可能的水平（缺省值是向量x中互异的值）；labels用来指定水平的名字；exclude表示从向量x中剔除的水平值；ordered是一个逻辑型选项用来指定因子的水平是否有次序。

函数tapply() 将一个功能函数（这里是mean()）用于第二个参数（这里是o）定义于第一个参数（这里是x）上得到的所有组(以factor或ordered决定)

注意，当第二个参数不是因子时，函数tapply() 同样有效，如tapply(x, state)。这对一些其他函数也是有效，因为必要时R会用as.factor() 把参数强制转换成因子。

```
> x=rbinom(n=10,size=2,p=c(0.2,0.3,0.5))
> x
[1] 1 1 2 0 2 0 0 1 1 0

> f=factor(x)
> f
```

```
[1] 1 1 2 0 2 0 0 1 1 0
Levels: 0 1 2

> factor(x,levels=0:3)
 [1] 1 1 2 0 2 0 0 1 1 0
Levels: 0 1 2 3

> factor(x,labels=c('a','b','c'))
 [1] b b c a c a a b b a
Levels: a b c

> t=table(x)
> t
x
0 1 2
4 4 2

> o=ordered(x)
> o
[1] 1 1 2 0 2 0 0 1 1 0
Levels: 0 < 1 < 2

> tapply(x,o,mean)
0 1 2
0 1 2
```

2.2.5 列表(list)

R的列表（list）是一个以对象的有序集合构成的对象。列表中包含的对象又称为它的分量（components）.

列表被认为是一种"递归"结构而不是原子结构, 因为它们的元素可以以它们各自的方式单独列出.

分量可以是不同的模式或类型，如一个列表可以同时包括数值向量，逻辑向量，矩阵，复向量，字符数组，函数等等。下面的例子演示怎么创建一个列表与查看其信息：

```
> Lst <- list(name="Fred", wife="Mary", no.children=3,
            child.ages=c(4,7,9))
# 查看信息
> str(Lst)
List of 4
 $ name      : chr "Fred"
 $ wife      : chr "Mary"
 $ no.children: num 3
 $ child.ages : num [1:3] 4 7 9

> Lst
$name
[1] "Fred"

$wife
[1] "Mary"

$no.children
[1] 3

$child.ages
[1] 4 7 9
```

获取分量

Lst$name 和Lst[[1]] 返回结果都是"Fred",
Lst$wife 和Lst[[2]] 返回的则是"Mary",
而Lst$child.ages[1] 和Lst[[4]][1] 返回一样的数字4。

```
> Lst$name
[1] "Fred"
> Lst[1]
$name
[1] "Fred"
> Lst$child.ages[1]
[1] 4
> Lst[4]
$child.ages
[1] 4 7 9
```

```
> Lst[4][1]
$child.ages
[1] 4 7 9

> Lst[[4]][1]
[1] 4
```

这里特别要注意一下Lst[[1]] 和Lst[1] 的差别。[[. . .]] 是用来选择单个元素的操作符，而[. . .] 是一个更为一般的下标操作符。因此前者得到的是列表Lst 中的第一个对象, 并且含有分量名字的命名列表（named list）中的分量名字会被排除在外的1 。后者得到的则是列表Lst 中仅仅由第一个元素构成的子列表。如果是命名列表，分量名字会传给子列表的。

2.2.6 数据框–data.frame

数据框（data frame）也是列表, 是一个属于"data.frame" 类的列表。不过，对于可能属于数据框的列表对象有一些限制条件.

分量必须是向量(数值, 字符, 逻辑)，因子，数值矩阵，列表或者其他数据框; 每列的行数必须相等.

数据框常常会被看作是一个由不同模式和属性的列构成的矩阵。它能以矩阵形式出现，行列可以通过矩阵的索引习惯访问。

2.2.7 数组(array)及维度命名

数组可以看作是带有多个下标类型相同的元素集合，如数值型, 是矩阵的推广. R 有一些简单的工具创建和处理数组，特别是矩阵。

向量只有在定义了 dim 属性后才能作为数组在R 中使用。假定，z是一个含1500个元素的向量。那么

```
> dim(z) <- c(3,5,100)
```

对dim 属性的赋值使得z向量成一个3维的3 ×5 ×100的数组。

```
> z[1,] # z 的第一行
> z[,1] # z 的第一列
> z[1:3,] # z 的第1:3行
> z[2*(1:3)-1,] # z 的1,3,5行. 括号可以不加. z[2*1:3-1,]

# 命名的顺序总是行,列,第三维,..., 每一维还可以有一个总名字, 也可以没有
Rabbits <-array(
      c(  0, 0, 6, 5,
          3, 0, 3, 6,
          6, 2, 0, 4,
          5, 6, 1, 0,
          2, 5, 0, 0),
   dim = c(2, 2, 5),
   dimnames = list(
      Delay = c("None", "1.5h"),
      Response = c("Cured", "Died"),
      Penicillin.Level = c("1/8", "1/4", "1/2", "1", "4")))
> Rabbits
, , Penicillin.Level = 1/8

     Response
Delay  Cured Died
  None     0    6
  1.5h     0    5

, , Penicillin.Level = 1/4

     Response
Delay  Cured Died
  None     3    3
  1.5h     0    6

, , Penicillin.Level = 1/2
```

```
     Response
Delay  Cured Died
  None     6    0
  1.5h     2    4

, , Penicillin.Level = 1

     Response
Delay  Cured Died
  None     5    1
  1.5h     6    0

, , Penicillin.Level = 4

     Response
Delay  Cured Died
  None     2    0
  1.5h     5    0
```

2.2.8 矩阵

矩阵 (matrix) 是一个双下标(2维)的数组. 但是，它非常的重要，以至于需要单独讨论。

R 包括许多只对矩阵操作的操作符和函数。

命名与数组array()一样.

矩阵的下标顺序是先第一列, 然后第二列, 等等. 例如

```
d<-matrix(c(1,2,3,4,5,6,7,8,9),nc=3)

> d
     [,1] [,2] [,3]
[1,]    1    4    7
```

```
[2,]    2    5    8
[3,]    3    6    9
> d[1:5]
[1] 1 2 3 4 5
```

2.2.9 字符串及相关操作

字符串比较重要, 所以单独讨论.

针对字符串的函数有 print, paste, cat, nchar, strsplit, regexpr, grep, gsub, sub 等.

```
> seq="GGGGCGAAACCGAGACTCTCAAATGACTTTTCTGA"
> seq=strsplit(seq,"")
> seq
[[1]]
 [1] "G" "G" "G" "G" "C" "G" "A" "A" "A" "C" "C" "G" "A" "G" "A" "C" "T" "C" "T"
[20] "C" "A" "A" "A" "T" "G" "A" "C" "T" "T" "T" "T" "C" "T" "G" "A"

> seq[[1]]=="g"|seq[[1]]=="G"
 [1]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
[13] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[25]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
```

2.3 基本操作

2.3.1 产生序列

```
> x <- 1:9 # 初始化
[1] 1 2 3 4 5 6 7 8 9
> x <- seq(1,10,by=0.1)
```

2.3.2 where are they?

```
> x==3 # where are they?
[1] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

2.3.3 what are they?

```
> which(x==3) # what are they?
[1] 3
```

2.3.4 各种计数

```
> length(x) # how many elements?
[1] 9
> sum(x>3)
[1] 7
> x>3
[1] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
> sum(x>7|x<3)
[1] 5
```

2.3.5 反转序列

```
> p <- rev(x) # reverse element
> p
[1] 9 8 7 6 5 4 3 2 1
> p[x==3] # logical extraction. Very useful
[1] 7
> x[x>4]
> x = c(45,43,46,48,51,46,50,47,46,45)
```

2.3.6 取得变量的一部分

```
> x=rnorm(100)
> y=x[x<1] # y 为 x < 1 的值
```

2.3.7 删除变量

```
> rm(x) # x=NULL 不删除， 值为NULL
```

2.3.8 过滤缺失值(missing values)

如果有缺失值, 则使用下述方法过滤

```
> x[!is.na(x)]
```

2.3.9 apply 的用法

第二个参数为 1, 作用于行, 2 则作用于列. c(1,2) 作用于每个值

```
# 两个向量 euclidean 距离
dist.euclidean <- function(x,y){
   res <- sqrt(sum((x-y)^2))
   res
}

# x1,x2 为点的坐标, g 为其所属类别
 X=data.frame(
   x1=c(4,1,3,3,7,4,6,5,3,6,4,4,5,7,5,10,7,4,9,5,8,6,7,8),
   x2=c(3,3,3,7,4,1,5,6,7,2,6,4,8,8,6,5,6,10,7,4,5,6,4,8),
```

```
   g=c(rep(1,10),rep(2,14)) )
> X
   x1 x2 g
1   4  3 1
2   1  3 1
3   3  3 1
4   3  7 1
5   7  4 1
6   4  1 1
7   6  5 1
8   5  6 1
9   3  7 1
10  6  2 1
11  4  6 2
12  4  4 2
13  5  8 2
14  7  8 2
15  5  6 2
16 10  5 2
17  7  6 2
18  4 10 2
19  9  7 2
20  5  4 2
21  8  5 2
22  6  6 2
23  7  4 2
24  8  8 2

# 下面画图看看
# red 为类1, blue 为类2.
> plot(x2~x1,col=c("red","blue")[g],data=X)
# 新样本(6,5)为 "*" 标记
> points(6,5,pch=8,cex=3)


# 计算新样本 (6,5) 与 X 中坐标 (x1, x2) 的距离的平方
> d<-apply(X[,1:2],1,dist.euclidean,y=c(6,5))^2
> d
 [1]  8 29 13 13  2 20  0  2 13  9  5  5 10 10  2 16  2 29 13  2  4  1  2 13

# 联合起来
```

```
> d1<-cbind(d,X$g)
> d1
      d
 [1,]  8 1
 [2,] 29 1
 [3,] 13 1
 [4,] 13 1
 [5,]  2 1
 [6,] 20 1
 [7,]  0 1
 [8,]  2 1
 [9,] 13 1
[10,]  9 1
[11,]  5 2
[12,]  5 2
[13,] 10 2
[14,] 10 2
[15,]  2 2
[16,] 16 2
[17,]  2 2
[18,] 29 2
[19,] 13 2
[20,]  2 2
[21,]  4 2
[22,]  1 2
[23,]  2 2
[24,] 13 2

# 按照距离排序
> o<-order(d1[,1])
> d2<-d1[o,]
> d2
     [,1] [,2]
 [1,]    0    1
 [2,]    1    2
 [3,]    2    1
 [4,]    2    1
 [5,]    2    2
 [6,]    2    2
 [7,]    2    2
 [8,]    2    2
```

```
 [9,]    4    2
[10,]    5    2
[11,]    5    2
[12,]    8    1
[13,]    9    1
[14,]   10    2
[15,]   10    2
[16,]   13    1
[17,]   13    1
[18,]   13    1
[19,]   13    2
[20,]   13    2
[21,]   16    2
[22,]   20    1
[23,]   29    1
[24,]   29    2
```

2.3.10 attach 的用法

```
> attach(x) # x 包含的向量纳入搜索空间，可以直接使用了
> x1
[1] 1 2 3 4 5 6 7 8 9
> detach(x)
> x1
错误: 找不到这个目标对象"x1"
```

2.3.11 总结

how many elements?	length(x)
ith element	x[2] (i = 2)
all but ith element	x[-2] (i = 2)
?rst k elements	x[1:5] (k = 5)
last k elements	x[(length(x)-5):length(x)] (k = 5)
speci?c elements.	x[c(1,3,5)] (First, 3rd and 5th)
all greater than some value	x[x>3] (the value is 3)

```
bigger than or less than some values x[ x< -2 | x > 2]
which indices are largest          which(x == max(x))
```

2.3.12 两个数据操作

```
> x = c(1,3,5,7,9)
> y = c(2,3,5,7,11,13)
> x+1
[1]  2  4  6  8 10
> y*2
[1]  4  6 10 14 22 26
> y[-3] # 去掉第三个
[1]  2  3  7 11 13
> y[x]
[1]  2  5 11 NA NA

> x = 1:10
> y=1:3
> y[4]=NA
> y
[1]  1  2  3 NA
> x[y]
[1]  1  2  3 NA
```

2.4 对象

R操作的实体在技术上来说就是对象(object).

R的对象类型包括数值型（numeric），复数型（complex），逻辑型（logical），字符型（character）和原味型（raw）.

2.4.1 对象的模式

一个对象的模式(mode)是该对象基本要素的类型. 所有对象都有的特征是长度(length).

空对象仍然有其模式.

```
> s=character()
> s
character(0)
> mode(s)
[1] "character"
> typeof(s)
[1] "character"

> e=numeric()
> e
numeric(0)
> mode(e)
[1] "numeric"
> typeof(e)
[1] "double"
```

2.4.2 对象函数

函数mode(object), typeof(object), length(object)可以用于任何数据对象以得到其模式和长度.

typeof是R自己独立的函数, 保留mode是为了和S兼容.

```
> x=1+2i
> x
[1] 1+2i
> mode(x)
[1] "complex"
> typeof(x)
```

```
[1] "complex"
> length(x)
[1] 1
```

2.4.3 获取和改变对象属性-类

attributes(object), str(object)

attr(object, name)

```
> attr(z, "dim") <- c(10,10) # 允许R把z当作一个10×10的矩阵。

> x
 [1] 0 1 0 0 0 1 1 1 0 0

> y=table(x)
> y
x
0 1
6 4
> attributes(y)
$dim
[1] 2

$dimnames
$dimnames$x
[1] "0" "1"


$class
[1] "table"

> str(y)
 int [, 1:2] 6 4
 - attr(*, "dimnames")=List of 1
  ..$ x: chr [1:2] "0" "1"
```

```
 - attr(*, "class")= chr "table"
> y[,1]
错 误 在y[, 1] : 量 度 数 目 不 对
> y[1]
0
6
> y[2]
1
4
> class(y) # 确 定y的class
[1] "table"
> dim(y)
[1] 2
> dimnames(y)
$x
[1] "0" "1"

> dimnames(y)$x
[1] "0" "1"
```

2.4.4 模 式 转 换

参 考 help(as)

```
> as(x,"character")
 [1] "0" "1" "0" "0" "0" "1" "1" "1" "0" "0"
> as.character(x)
 [1] "0" "1" "0" "0" "0" "1" "1" "1" "0" "0"

> s=as.character(x)
> s
 [1] "0" "1" "0" "0" "0" "1" "1" "1" "0" "0"
> as.numeric(s)
 [1] 0 1 0 0 0 1 1 1 0 0
```

2.5 使用data.frame

2.5.1 产生 data.frame

```
> weight = c(150, 135, 210, 140)
> height = c(65, 61, 70, 65)
> gender = c("Fe","Fe","M","Fe")
> study = data.frame(weight,height,gender)
```

2.5.2 行列的变量名称

列的名称可以在赋值的时候指定, 也可以用下面的方法

```
>  study = data.frame(w=weight,h=height,g=gender)
>  row.names(study)<-c("Mary","Alice","Bob","Judy")
> names(study) <- c("wei","hei","gen")
```

2.5.3 取得数据的各种方法

取得行, 列的数据

```
> study[,"wei"]
[1] 150 135 210 140
> study[,1:2]
    wei hei
Mary  150  65
Alice 135  61
Bob   210  70
Judy  140  65
> study["Mary",]
    wei hei gen
Mary 150  65  Fe
```

```
> study["Mary","wei"]
[1] 150
```

使用 $ 符号

```
> study$wei
[1] 150 135 210 140
```

使用名称及缩写

```
> study[["wei"]]
[1] 150 135 210 140
> study["wei"]
   wei
Mary  150
Alice 135
Bob   210
Judy  140
> study["w"]
错误在"[.data.frame"(study, "w") : 选择了未定义的列
> study[["w"]]
[1] 150 135 210 140
```

使用 index(下标)

```
> study[1]
   wei
Mary  150
Alice 135
Bob   210
Judy  140
> study[[1]]
[1] 150 135 210 140
```

2.5.4 条件取得数据

```
> study[study$gen=="Fe",]
   wei hei gen
Mary  150  65  Fe
Alice 135  61  Fe
Judy  140  65  Fe
```

2.5.5 使用 stack 与 unstack

stack 是把一个 data.frame 连接成为两列, 一列为数据, 另外一列为数据原来的列名称.

unstack 相反, 把一列数据按照因子(水平)分离为不同的列. 如果每列数量相等, 则强制为 data.frame, 否则为 list. 默认第一列为数据, 第二列为因子. 如果不是这个顺序的话, 需要 form 参数(模型)

```
> l <- list()
> x <- c("y","n")
> i <- sample(x,10,replace=TRUE)
> i
[1] "y" "n" "y" "y" "y" "n" "n" "y" "y" "n"
> a <- rep("y",5)
> b <- rep("n",5)
> c <- c(a,b)
> l <- list()
> l$ind <- i
> l$val <- rnorm(10)
> unstack(l,form=val~ind) # 注意用法, 第二个参数为 form, val~ind 也可以为 "val~ind"
$n
[1]  0.424591771  0.004047361 -1.208147843 -0.516055218

$y
[1] -0.0708544  0.5732878 -0.6390650 -0.6262143 -0.1372453  0.2929985
```

```
> l$ind <- c
> unstack(l,form=val~ind)
          n         y
1  0.004047361 -0.0708544
2 -1.208147843  0.4245918
3 -0.137245323  0.5732878
4  0.292998458 -0.6390650
5 -0.516055218 -0.6262143
```

如果顺序为默认的话, 不需要form参数

```
> l1 <- list()
> l1$val <- l$val
> l1$ind <- l$ind
> unstack(l1)
错误在inherits(object, "formula") : 缺少变元"form",也没有缺
省值
> unstack(data.frame(l1))
          n         y
1  0.004047361 -0.0708544
2 -1.208147843  0.4245918
3 -0.137245323  0.5732878
4  0.292998458 -0.6390650
5 -0.516055218 -0.6262143
```

绘图(boxplot时候, 需要把list转换为 data.frame, 即使赋值后也是如此)

```
> boxplot(unstack(data.frame(l1)))
> boxplot(l$val~l$ind)
错误在model.frame(formula, rownames, variables, varnames, extras, extranames,  :
      变数种类不对
> boxplot(l1$val~l1$ind)
错误在model.frame(formula, rownames, variables, varnames, extras, extranames,  :
      变数种类不对
```

```
> d <- data.frame(l1)
> boxplot(d$val~d$ind)

> x <-d$val
> y<-d$ind
> boxplot(x~y)
> x <-l$val
> y<-l$ind
> boxplot(x~y)
错误在model.frame(formula, rownames, variables, varnames, extras, extranames,  :
      变数种类不对
```

Chapter 3

类和泛型函数

一个对象的类决定了它会如何被一个泛型函数处理。相反，一个泛型函数由参数自身类的种类来决定完成特定工作或者事务的。如果参数缺乏任何类属性，或者在该问题中有一个不能被任何泛型函数处理的类，泛型函数会有一种默认的处理方式。

3.1 S3和S4类

参考文献[26] 2.3.5.

S3类是R内核带的. 把一个list的属性class赋值, 其它泛型函数在接受此参数的时候首先查看其类, 然后调用合式的方法.

```
h <- list(a=rnorm(3),b="This shouldn't print")
class(h) <- "myclass"
print.myclass<-function(x){cat("A is:",x$a,"\n")}
print(h)
```

```
A is: -0.710968 -1.611896 0.6219214
```

S4类是近期加入R的, 由包methods实现, R已经自带. 一般包含

数据和函数, 类似其它语言的面向对象的语法.

setClass 创建新的类. new()函数创建其对象. 其属性使用"@"引用.

3.2 查看类可用的泛型函数

可以用函数methods() 得到当前对某个类对象可用的泛型函数列表：

```
methods(class="data.frame")
```

3.3 查看泛型函数可处理的类

相反，一个泛型函数可以处理的类同样很多。例如，plot() 有默认的方法和变量处理对象类"data.frame"，"density"，"factor"，等等。一个完整的列表同样可以通过函数methods() 得到：

```
methods(plot)
```

3.4 查看泛型函数代码

许多泛形函数的函数主体部分非常的短，如

```
> coef
function (object, ...)
UseMethod("coef")
```

UseMethod 的出现暗示着这是一个泛形函数。为了查看那些方法可以使用，我们可以使用函数methods()

```
> methods(coef)
[1] coef.aov*        coef.Arima*      coef.default* coef.listof*
[5] coef.nls*        coef.summary.nls*

   Non-visible functions are asterisked
```

这个例子中有六个方法，不过其中任何一个都不能简单地通过键入名字来查看。我们可以通过下面两种方法查看这种方法

```
> getAnywhere("coef.aov")
A single object matching 'coef.aov' was found
It was found in the following places
  registered S3 method for coef from namespace stats
  namespace:stats
with value

function (object, ...)
{
    z <- object$coef
    z[!is.na(z)]
}

> getS3method("coef", "aov")
function (object, ...)
{
    z <- object$coef
    z[!is.na(z)]
}
```

3.5 编写自己的类和泛型函数

下面是一个例子

```
# 编写函数
> xpos
function(x, ...)
   UseMethod("xpos")
> xpos.xypoint <- function(x) x$x
> xpos.rthetapoint <- function(x) x$r * cos(x$theta)

> xpos
function(x, ...)
   UseMethod("xpos")

# 改变数据的类
> x=list(x=c(1,2))
> x
$x
[1] 1 2

> x$x
[1] 1 2
> class(x)="xypoint"
> x
$x
[1] 1 2

attr(,"class")
[1] "xypoint"

# 调用泛型函数
> xpos(x)
[1] 1 2
```

Chapter 4

数值计算

参考 [15] 第二章 与 [31]

4.1 运算符号

- 基本符号

```
+*/-^
< <= > >= == !=
```

- 布尔运算符号

```
!,| &  - (可以 && 代替 &, -- 代替 -)
```

- 模余

```
%% 余数
%/% 商(Euclidian division )
> 12%/%3
[1] 4
> 12%%3
[1] 0
```

```
> 12.1%%3
[1] 0.1
> 12.1%/%3
[1] 4
> 12.1%/%2.2
[1] 5
# 5*-2+1=-9
> -9%%5
[1] 1
> -9%/%5
[1] -2
```

- 集合判断

```
%in%
> 17 %in% 1:100
[1] TRUE
> 17.1 %in% 1:100
[1] FALSE
```

帮助

```
?"+"
?"<"
?"<-"
?"!"
?"["
?Syntax
?kronecker
?match
library(methods)
?slot
```

还可以设计自己的符号

```
> "%w/o%" <- function(x,y) x[!x %in% y]
```

```
> (1:10) %w/o% c(3,7,12)
[1] 1 2 4 5 6 8 9 10
```

4.2 复数基本运算

- 'Re': 取实部
- 'Im': 取虚部
- 'Mod': 求模
- 'Arg': 角度
- 'Conj': 共轭

4.3 四则运算

```
> A <- matrix(1:6, nrow=2, byrow=T); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
>  B <- matrix(1:6, nrow=2); B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
>  C <- matrix(c(1,2,2,3,3,4), nrow=2); C
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    3    4
> D <- 2*C+A/B; D
     [,1]     [,2] [,3]
[1,]    3 4.666667  6.6
[2,]    6 7.250000  9.0
```

形状不一样的数组也可以运算, 一般是把短向量循环使用(谨慎!).

4.4 插值

函数: spline

下面同时展示一个过拟合的例子

```
> n <- 10
> x <- seq(0,1,length=n)
> y <- 1-2*x+.3*rnorm(n)
> plot(spline(x, y, n = 10*n), col = 'red', type='l', lwd=3)
> points(y~x, pch=16, lwd=3, cex=2)
> abline(lm(y~x))
> title(main='Overfit')
```

4.5 排列组合

choose(n,k) 组合数 combn(n,k) 列出所有组合

4.6 积分

integrate() 有时候会出错. 一般用法为

```
integrate(f, lower, upper, ..., subdivisions=100,
            rel.tol = .Machine$double.eps^0.25, abs.tol = rel.tol,
            stop.on.error = TRUE, keep.xy = FALSE, aux = NULL)
```

'adapt' in the 'adapt' package on CRAN, for multivariate integration.

下面是几个例子.

```
> integrate(dnorm, -1.96, 1.96)
```

```
0.9500042 with absolute error < 1.0e-11

> integrate(dnorm, -Inf, Inf)
1 with absolute error < 9.4e-05

> integrand <- function(x) {1/((x+1)*sqrt(x))}
> integrate(integrand, lower = 0, upper = Inf)
3.141593 with absolute error < 2.7e-05

> integrate(integrand, lower = 0, upper = 10)
2.529038 with absolute error < 3e-04

> integrate(integrand, lower = 0, upper = 100000)
3.135268 with absolute error < 4.2e-07

> integrate(integrand, lower = 0, upper = 1000000, stop.on.error = FALSE)
failed with message 'the integral is probably divergent'

## integrate can fail if misused
    integrate(dnorm,0,2)
    integrate(dnorm,0,20)
    integrate(dnorm,0,200)
    integrate(dnorm,0,2000)
    integrate(dnorm,0,20000) ## fails on many systems
    integrate(dnorm,0,Inf)   ## works
```

4.7 求解方程式

参考数值方法的介绍 http://cran.r-project.org/web/views/Optimization.html

参考 [36] chapter 8

线性方程组求解见矩阵运算.

4.7.1 一元(非线性)方程式求根

求一个方程式的多个根使用: rootSolve 包 root.all()

我们想求方程式

$$y = cos(x) - 2x \quad (or \quad cos(x) = 2x)$$

的根, 即使得$y = 0$的x的值.

先画图看看总是不错的.

```
curve(cos(x)-2*x,-10,10)
abline(h=0,lty=2)
```

看到确实有x值使得方程式为零.

下面使用函数 uniroot() 求根. 用法

```
uniroot(f, interval, ...,
        lower = min(interval), upper = max(interval),
        f.lower = f(lower, ...), f.upper = f(upper, ...),
        tol = .Machine$double.eps^0.25, maxiter = 1000)
```

- f: 方程式, 其第一个参数未知. 求使得方程式f值为零的第一个参数的值.
- interval: 根的搜索范围的结束点
- ...: f的其它参数的值
- tol: 需要的精确度
- f.lower, f.upper: the same as 'f(upper)' and 'f(lower)', 为了减少计算量传递的参数.

```
> u=uniroot(f = function(x) cos(x)-2*x, interval=c(-10,10)); u
$root # 根
[1] 0.4501686

$f.root # 在根处的方程式的值
[1] 3.655945e-05

$iter # 迭代次数
[1] 5

$estim.prec # 根的精确度
[1] 6.103516e-05

> r=u$root; cos(r)-2*r # 手工计算在根处的方程式的值
[1] 3.655945e-05

# 下面是函数帮助的例子
> f <- function (x,a) x - a
> str(xmin <- uniroot(f, c(0, 1), tol = 0.0001, a = 1/3))
List of 4
 $ root     : num 0.333
 $ f.root   : num -5.55e-17
 $ iter     : int 2
 $ estim.prec: num 5e-05
```

想计算$1000 = y * (3 + x) * (1 + y)^4$, 未知数是y, x从$1 - 100$变动. 我们绘出根与x的关系

```
eq<-function(y,x){
   return (1000-y*(3+x)*(1+y)^4)
}

r=rep(0,100)
x=1:100
for (i in x){
   r[i]<-uniroot(eq, c(-100,100),x=i)$root
}
plot(r~x)
```

4.7.2 多元(非线性)方程组

非线性微分方程求根: rootSolve 包 multiroot() 求解n个(非线性)方程组的n个根.

下面是 multiroot()帮助的例子. 更具体见帮助文件.

```
> model <- function(x) {
   c(F1=x[1]^2+ x[2]^2 -1,F2=x[1]^2- x[2]^2 +0.5)}
> (ss<-multiroot(f=model,start=c(1,1)))
$root
[1] 0.5000000 0.8660254

$f.root
        F1         F2
2.323138e-08 2.323308e-08

$iter
[1] 5

$estim.precis
[1] 2.323223e-08
# 代入原方程组
> model(ss$root)
        F1         F2
2.323138e-08 2.323308e-08

# 3个方程式2个根
model <- function(x) {
   c(F1= x[1] + x[2] + x[3]^2 - 12,
      F2= x[1]^2 - x[2] + x[3] - 2,
      F3= 2 * x[1] - x[2]^2 + x[3] - 1 )}
# first solution
(ss<-multiroot(model,c(1,1,1),useFortran=FALSE))
(ss<-multiroot(f=model,start=c(1,1,1)))
# second solution; use different start values
(ss<-multiroot(model,c(0,0,0)))
model(ss$root)

# 还可以求解矩阵
```

```
f2<-function(x)
{
 X<-matrix(nr=5,x)
 X %*% X %*% X -matrix(nr=5,data=1:25,byrow=TRUE)
}
x<-multiroot(f2, start= 1:25 )$root
X<-matrix(nr=5,x)
X%*%X%*%X
```

4.8 优化(求极值)

参考数值方法的介绍 http://cran.r-project.org/web/views/Optimization.html

4.8.1 optimize()函数

函数 optimize() 求得一个函数在指定区间的极值. 使用 golden section search(黄金分割搜索) 和 successive parabolic interpolation(连续抛物线插值). 收敛速度不比使用 Fibonacci search 慢多少. 更多详细解释见帮助, 另外参考 "极大似然法"12.2.

用法为

```
optimize(f = , interval = ,  ..., lower = min(interval),
           upper = max(interval), maximum = FALSE,
           tol = .Machine$double.eps^0.25)
```

返回

- minimum(maximum): 函数取得最大(最小)值时自变量x的值
- objective: 函数的极大(极小)值

下面是帮助中的例子

```
f <- function (x,a) (x-a)^2
xmin <- optimize(f, c(0, 1), tol = 0.0001, a = 1/3)
> xmin
$minimum
[1] 0.3333333

$objective
[1] 0

# 不赋值的话, 可以看到函数自变量取值的情况
optimize(function(x) x^2*(print(x)-1), lower=0, upper=10)

# 函数中有部分常数值, 计算区间取的不够大的话, 就会犯错
误
f  <- function(x) ifelse(x > -1, ifelse(x < 4, exp(-1/abs(x - 1)), 10), 10)
fp <- function(x) { print(x); f(x) }

plot(f, -2,5, ylim = 0:1, col = 2)

# 虽然函数极小值在(1,0)附近, 但是区间不够的话收敛到错误
的地方
optimize(fp, c(-4, 20))# doesn't see the minimum
# 这个就正确了
optimize(fp, c(-7, 20))# ok
```

4.8.2 nlm()函数

nlm() 函数使用 Newton-type 算法求最小值. 参考 28 章非线性回归与非线性最小平方. 《R导论》[19](page 73)中统计模型部分中有一个广义线性模型对广义线性模型有很好的描述, 请参考之。

求无约束求化问题(Rosenbrock函数, 或橡胶函数)

$$min \quad f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

的极小点.

```
# 写出目标函数
```

```
obj<-function(x){
  f<-c(10*(x[2]-x[1]^2), 1-x[1])
  sum(f^2)
}

# 初始值设置
x0=c(-1.2,1)
# 求解
nlm(obj,x0)
> nlm(obj,x0)
$minimum # 极小值
[1] 3.973766e-12

$estimate # 极小值对应的x点
[1] 0.999998 0.999996

$gradient # 极小值处的梯度
[1] -6.539277e-07  3.335997e-07

$code # 成功与否
[1] 1

$iterations # 迭代次数
[1] 23
```

4.8.3 其它函数

BB 包求解非线性系统方程组, 并根据约束优化.

R的非线性优化程序是optim()，nlm() 和nlminb().

函数 optim() 是一个广泛意义的优化函数.

函数 nls(): 非线性模型参数的最小平方估计 (Determine the nonlinear (weighted) least-squares estimates of the parameters of a nonlinear model). 使用请参考 28 章非线性回归与非线性最小平方

Chapter 5

矩阵运算

5.1 构造Hilbert矩阵

Matrix包有函数Hilbert()可以产生n阶对称Hilbert矩阵. Hilbert矩阵的阶数n较大的时候是病态的, 故经常用来测试数值方法程序.

```
# 手工计算
n<-4; x<-array(0, dim=c(n,n))
for (i in 1:n){
   for (j in 1:n){
      x[i,j]<-1/(i+j-1)}}
> x
        [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.5000000 0.3333333 0.2500000
[2,] 0.5000000 0.3333333 0.2500000 0.2000000
[3,] 0.3333333 0.2500000 0.2000000 0.1666667
[4,] 0.2500000 0.2000000 0.1666667 0.1428571

# 使用函数 Hilbert()
library(Matrix)
> Hilbert(3)
3 x 3 Matrix of class "dpoMatrix"
        [,1]      [,2]      [,3]
```

```
[1,] 1.0000000 0.5000000 0.3333333
[2,] 0.5000000 0.3333333 0.2500000
[3,] 0.3333333 0.2500000 0.2000000
```

5.2 矩阵转置

使用函数 t()

```
> A <- matrix(1:6, nrow=2, byrow=T); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> t(A)
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

5.3 上下三角矩阵

base 包的函数如下

```
> x=matrix(1:20,c(4,5))
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> upper.tri(x, diag = FALSE)
      [,1]  [,2]  [,3]  [,4] [,5]
[1,] FALSE  TRUE  TRUE  TRUE TRUE
```

```
[2,] FALSE FALSE  TRUE  TRUE TRUE
[3,] FALSE FALSE FALSE  TRUE TRUE
[4,] FALSE FALSE FALSE FALSE TRUE
> x[upper.tri(x)]
 [1]  5  9 10 13 14 15 17 18 19 20

# 下三角矩阵
> lower.tri(x)
```

spam 包的函数功能要多一些

```
> y=matrix(1:20,c(4,5))
> y1=as.spam(y)
> upper.tri(y1,diag=T)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1    1    1    1
[2,]    0    1    1    1    1
[3,]    0    0    1    1    1
[4,]    0    0    0    1    1
Class 'spam'
> y1
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
Class 'spam'
> lower.tri(y1,diag=T)

# 获取
> y1[lower.tri(y1,diag=T)]
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    2    6    0    0
[3,]    3    7   11    0
[4,]    4    8   12   16
Class 'spam'
```

5.4 行列式的值

```
> det(A)
错误于determinant.matrix(x, logarithm = TRUE, ...) :
  'x'必需是正方形矩阵
> det(A[1:2,1:2])
[1] -3
```

5.5 内积与外积

内积(点积)可以使用

```
> x <- 1:5; y <- 2*1:5
> x
[1] 1 2 3 4 5
> y
[1]  2  4  6  8 10

# 向量内积
> x %*% y
     [,1]
[1,]  110

# %*% 符号是通常意义下的矩阵乘
# crossprod() 是内积函数, 执行 t(x) %*% y
> crossprod(x,y)
     [,1]
[1,]  110

# 矩阵内积
> A <- matrix(1:6, nrow=2, byrow=T); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
>  B <- matrix(1:6, nrow=2); B
     [,1] [,2] [,3]
```

```
[1,]    1    3    5
[2,]    2    4    6

> A %*% B
错误于A %*% B : 非整合变元

> t(A) %*% B
     [,1] [,2] [,3]
[1,]    9   19   29
[2,]   12   26   40
[3,]   15   33   51

> crossprod(A,B)
     [,1] [,2] [,3]
[1,]    9   19   29
[2,]   12   26   40
[3,]   15   33   51

# tcrossprod(x,y) 是外积, 执行 x %*% t(y), 或 x %o% y 或 outer(x,y)
> tcrossprod(A,B)
     [,1] [,2]
[1,]   22   28
[2,]   49   64
> tcrossprod(x,y)
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50

> x %o% y
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50

> x %*% t(y)
     [,1] [,2] [,3] [,4] [,5]
```

```
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50
```

```
> outer(x,y)
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50
```

函数 outer() 用法为

```
outer(X, Y, fun = "*", ...)
```

fun 是外积运算的函数, 做三维曲面时非常有用

5.6 对角矩阵与取对角

```
# 当参数为向量时, 产生对角矩阵
> v<-c(1,4,5)
> diag(v)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    4    0
[3,]    0    0    5

# 当参数为矩阵时, 取对角元素
> A <- matrix(1:6, nrow=2, byrow=T); A
     [,1] [,2] [,3]
[1,]    1    2    3
```

```
[2,]    4    5    6
> diag(A)
[1] 1 5
```

5.7 解线性方程组和求矩阵的逆矩阵

求解线性方程组 $Ax = b$, 使用命令 solve(A,b). 求A的逆, 使用命令 solve(A).

```
> A <- t(array(c(1:8, 10),dim=c(3,3))); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10
> b <- c(1,1,1)
# 解方程组
> x <- solve(A,b); x
[1] -1.000000e-00  1.000000e-00  3.806634e-16

# 求逆矩阵
> B <- solve(A); B
          [,1]      [,2] [,3]
[1,] -0.6666667 -1.333333    1
[2,] -0.6666667  3.666667   -2
[3,]  1.0000000 -2.000000    1
```

5.8 求矩阵的特征值与特征向量

```
> A <- t(array(c(1:8, 10),dim=c(3,3))); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10
```

```
# values 为特征值, vectors 的列为对应的特征向量

# 非对称矩阵的特征值与特征向量
> eigen(A)
$values
[1] 16.7074933 -0.9057402  0.1982469

$vectors
          [,1]       [,2]       [,3]
[1,] -0.2235134 -0.8658458  0.2782965
[2,] -0.5039456  0.0856512 -0.8318468
[3,] -0.8343144  0.4929249  0.4801895

# 对称矩阵的特征值与特征向量
> eigen(crossprod(A))
$values
[1] 303.19533618   0.76590739   0.03875643

$vectors
          [,1]        [,2]       [,3]
[1,] -0.4646675  0.833286355  0.2995295
[2,] -0.5537546 -0.009499485 -0.8326258
[3,] -0.6909703 -0.552759994  0.4658502
```

5.9 矩阵分解

5.9.1 三角分解法(LU)

三角分解法是将原正方 (square) 矩阵分解成一个上三角形矩阵 或是排列(permuted) 的上三角形矩阵(L)和一个下三角形矩阵(U)，这样的分解法又称为LU分解法。

$$A = LU$$

它的用途主要在简化一个大矩阵的行列式值的计算过程，求反矩阵，和求解联立方程组。

不过要注意这种分解法所得到的上下三角形矩阵并非唯一，还可找到数个不同的一对上下三角形矩阵，此两三角形矩阵相乘也会得到原矩阵。

```
> library(Matrix)
> x=matrix(rnorm(9),c(3,3)); x
          [,1]       [,2]       [,3]
[1,] -0.6334882 -0.3915563  0.4906192
[2,]  0.4591368  0.5246114  0.6949097
[3,] -0.4435543 -1.5035618 -0.0191876

# 根据例子, 需要将矩阵转换为 CsparseMatrix 类. why???
> lu(x)
错误于function (classes, fdef, mtable) :
  unable to find an inherited method for function "lu", for signature "matrix"

> A = as(x,"CsparseMatrix")
> p=lu(A)
# 结果是 'MatrixFactorization' of Formal class 'sparseLU'
> p
'MatrixFactorization' of Formal class 'sparseLU' [package "Matrix"] with 5 slots
  ..@ L  :Formal class 'dtCMatrix' [package "Matrix"] with 7 slots
  .. .. ..@ i      : int [1:6] 0 1 2 1 2 2
  .. .. ..@ p      : int [1:4] 0 3 5 6
  .. .. ..@ Dim    : int [1:2] 3 3
  .. .. ..@ Dimnames:List of 2
  .. .. .. ..$ : NULL
  .. .. .. ..$ : NULL
  .. .. ..@ x      : num [1:6]  1.000  0.700 -0.725  1.000 -0.196 ...
  .. .. ..@ uplo   : chr "L"
  .. .. ..@ diag   : chr "N"
  ..@ U  :Formal class 'dtCMatrix' [package "Matrix"] with 7 slots
  .. .. ..@ i      : int [1:6] 0 0 1 0 1 2
  .. .. ..@ p      : int [1:4] 0 1 3 6
  .. .. ..@ Dim    : int [1:2] 3 3
  .. .. ..@ Dimnames:List of 2
  .. .. .. ..$ : NULL
  .. .. .. ..$ : NULL
  .. .. ..@ x      : num [1:6] -0.633 -0.392 -1.229  0.491 -0.363 ...
  .. .. ..@ uplo   : chr "U"
```

```
  .. .. ..@ diag    : chr "N"
  ..@ p  : int [1:3] 0 2 1
  ..@ q  : int [1:3] 0 1 2
  ..@ Dim: int(0)

> p@L
3 x 3 sparse Matrix of class "dtCMatrix"

[1,]  1.0000000  .         .
[2,]  0.7001776  1.0000000 .
[3,] -0.7247755 -0.1958845 1

> p@U
3 x 3 sparse Matrix of class "dtCMatrix"

[1,] -0.6334882 -0.3915563  0.4906192
[2,]  .        -1.2294029 -0.3627082
[3,]  .         .         0.9794496

# L*U 既得原来的矩阵, 行顺序可能不同
> p@L %*% p@U
3 x 3 sparse Matrix of class "dgCMatrix"

[1,] -0.6334882 -0.3915563  0.4906192
[2,] -0.4435543 -1.5035618 -0.0191876
[3,]  0.4591368  0.5246114  0.6949097
> A
3 x 3 sparse Matrix of class "dgCMatrix"

[1,] -0.6334882 -0.3915563  0.4906192
[2,]  0.4591368  0.5246114  0.6949097
[3,] -0.4435543 -1.5035618 -0.0191876
```

5.9.2 奇异值分解(svd)

奇异值分解 (sigular value decomposition,SVD) 是另一种正交矩阵分解法； SVD是最可靠的分解法，但是它比QR 分解法要花上近十倍的计算时间。和QR分解法相同， 原矩阵A不必为正

方矩阵。使用SVD分解法的用途是解最小平方误差法和数据压缩。

$$A = UDV^T$$

其中, 其中U和V代表二个相互正交矩阵. D为对角矩阵, 即A的奇异值.

```
> A <- t(array(c(1:8, 10),dim=c(3,3))); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10

> s=svd(A); s
$d
[1] 17.4125052  0.8751614  0.1968665

$u
          [,1]       [,2]      [,3]
[1,] -0.2093373  0.96438514  0.1616762
[2,] -0.5038485  0.03532145 -0.8630696
[3,] -0.8380421 -0.26213299  0.4785099

$v
          [,1]        [,2]      [,3]
[1,] -0.4646675 -0.833286355  0.2995295
[2,] -0.5537546  0.009499485 -0.8326258
[3,] -0.6909703  0.552759994  0.4658502

> s$u %*% diag(s$d) %*% t(s$v)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10
```

5.9.3 QR分解

QR分解法是将矩阵分解成一个正规正交矩阵(Q)与上三角形矩阵(R)。

$$A = QR$$

正规正交矩阵Q满足的条件

$$QQ^T = I$$

所以称为QR分解法与此正规正交矩阵的通用符号Q有关。

类似的，我们可以定义 A 的 QL, RQ 和 LQ 分解。

更一般的說，我们可以因数分解复数 mn 矩阵(有着 $m >= n$) 为 mn 酉矩阵 (在 Q*Q = I 的意义上)和 nn 上三角矩阵的乘积。

如果 A 是非奇异的，则这个因数分解是唯一，当我们要求 R 的对角是正数的时候。

QR分解的实际计算有很多方法，例如Givens旋转、Householder变换，以及Gram-Schmidt正交化等等。每一种方法都有其优点和不足。

设 X 为 $n * p$ 矩阵, 可以求得正交矩阵Q, 使得$Q^T X$在主对角线以下为 0. $n >= p$ 时

$$Q^T X = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

其中 R 为上三角矩阵.

将 Q 分割为(Q_1, Q_2), Q_1 有 P 行, 则 $Q^T = Q^{-1}$(正交矩阵的特性 $QQ^T = I$)

$$X = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R$$

若 X 有 p 秩(rank), 则 由 X 行 向 量 形 成 的 空 间 可 以 找 到 一 个 正 交 投 影 (orthogonal projection) 矩 阵 P

$$P = X(X^TX)^{-1}X^T = Q_1R(R^TQ_1^TRQ_1)^{-1}R^TQ_1^T = Q_1Q_1^T$$
$$(Q^TQ = I -> Q_1^TQ_1 = I)$$

另 外 有 矩 阵$P_x = Q_2Q_2^T$为 对X垂 直 方 向 的 投 影.

```
> x
    [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> q=qr(x)

# 其 中 $qr矩 阵 上 三 角 为QR分 解 的R矩 阵,
# 下 三 角 为 正 交 矩 阵Q的 部 分 信 息, 使 用 压 缩 存 储 方
法(DQRDC and DGEQP3 differs).
# $qraux 为Q的 附 加 信 息.
> q
$qr
          [,1]       [,2]         [,3]         [,4]         [,5]
[1,] -5.4772256 -12.7801930 -2.008316e+01 -2.738613e+01 -3.468910e+01
[2,]  0.3651484  -3.2659863 -6.531973e+00 -9.797959e+00 -1.306395e+01
[3,]  0.5477226  -0.3781696  2.641083e-15  2.056562e-15  5.493622e-15
[4,]  0.7302967  -0.9124744  8.583032e-01 -2.111449e-16  6.562532e-16

$rank
[1] 2

$qraux
[1] 1.182574e+00 1.156135e+00 1.513143e+00 2.111449e-16 6.562532e-16

$pivot
[1] 1 2 3 4 5

attr(,"class")
[1] "qr"
```

```
# $qr的下三角信息结合 $qraux 解压缩为 Q 矩阵
> Q=qr.Q(q); Q
          [,1]         [,2]       [,3]       [,4]
[1,] -0.1825742 -8.164966e-01 -0.4000874 -0.37407225
[2,] -0.3651484 -4.082483e-01  0.2546329  0.79697056
[3,] -0.5477226 -6.163689e-17  0.6909965 -0.47172438
[4,] -0.7302967  4.082483e-01 -0.5455419  0.04882607

# $qr 的上三角矩阵
> R=qr.R(q); R
         [,1]      [,2]         [,3]         [,4]         [,5]
[1,] -5.477226 -12.780193 -2.008316e+01 -2.738613e+01 -3.468910e+01
[2,]  0.000000  -3.265986 -6.531973e+00 -9.797959e+00 -1.306395e+01
[3,]  0.000000   0.000000  2.641083e-15  2.056562e-15  5.493622e-15
[4,]  0.000000   0.000000  0.000000e+00 -2.111449e-16  6.562532e-16
> qr.X(q)
    [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16

# 重构 x
> Q%*%R
    [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
```

5.10 最小二乘法与QR分解

假设求解一个最小二乘法问题. X 为 $n*p$ 矩阵

$$\rho^2 = \| y - Xb \|^2 = min$$

若 X 为行向量间线性独立的矩阵, 则

$$\begin{aligned} X^TXb &= X^Ty \\ b &= (X^TX)^{-1}X^Ty \\ &= (R^TQ_1^TQ_1R)^{-1}R^TQ_1^Ty \\ &= R^{-1}R^{-T}R^TQ_1^Ty \\ &= R^{-1}Q_1^Ty \end{aligned}$$

设

$$z = Q_1^Ty$$

则解$Rb = z$系统即可求得b. 残差向量$r = y - Xb$为y向量投影到X矩阵行向量垂直方向的分量. 由前面

$$r = P_xy = Q_2Q_2^Ty$$

令 $s = Q_2^Ty, r = Q_2s$, 则

$$\rho^2 = \| r \|^2 = \| Q_2s \|^2 = \| s \|^2$$

对于原来的问题可以删减而得到一个部分系统

$$\rho_1^2 = \| y - X_1^{(1)} \|^2$$

求其最小值

$$b^{(1)} = R_{11}^{-1}Q_1^{(1)T}y \equiv R_{11}^{-1}z$$

$$Q_1 = (Q_1^{(1)}, Q_2^{(1)})$$

相同地

$$z^T = (z_1^T, z_2^T)$$

残差平方和

$$\rho_1^2 = \| Q_2^Ty \|^2 + \| Q_2^{(1)T}y \|^2 \equiv \| s \|^2 + \| z_2 \|^2$$

因此, QR因子可以解最小二乘法删去任意组末段行向量的问题.

函数 lsfit() 解最小二乘估计问题中的b向量($coefficients). 下面是一个例子.

```
> x<-c(0.0, 0.2, 0.4, 0.6, 0.8)
> y<-c(0.9, 1.9, 2.8, 3.3, 4.2)
> l <- lsfit(x, y)
> l
$coefficients
Intercept         X
    1.02      4.00

$residuals
[1] -0.12  0.08  0.18 -0.12 -0.02

$intercept
[1] TRUE

$qr
$qt
[1] -5.85849810  2.52982213  0.23749843 -0.02946714  0.10356728

$qr
     Intercept         X
[1,] -2.2360680 -0.8944272
[2,]  0.4472136  0.6324555
[3,]  0.4472136 -0.1954395
[4,]  0.4472136 -0.5116673
[5,]  0.4472136 -0.8278950

$qraux
[1] 1.447214 1.120788

$rank
[1] 2

$pivot
[1] 1 2

$tol
[1] 1e-07

attr(,"class")
[1] "qr"
```

如果使用 QR 分解, 输入矩阵需要加入一列 1 元素. 结果与 lsfit 里的一样.

```
> X<-matrix(c(rep(1,5), x), ncol=2)
> X
     [,1] [,2]
[1,]    1  0.0
[2,]    1  0.2
[3,]    1  0.4
[4,]    1  0.6
[5,]    1  0.8
> qr(X)
$qr
          [,1]       [,2]
[1,] -2.2360680 -0.8944272
[2,]  0.4472136  0.6324555
[3,]  0.4472136 -0.1954395
[4,]  0.4472136 -0.5116673
[5,]  0.4472136 -0.8278950

$rank
[1] 2

$qraux
[1] 1.447214 1.120788

$pivot
[1] 1 2

attr(,"class")
[1] "qr"
```

Chapter 6

绘图

参考文献[?] 中有丰富的图

6.1 图形环境设置-par函数

6.1.1 设置margin大小

```
> op <- par(mar=c(3,4,2,2)+.1)
```

6.1.2 设置显示区域

```
> plot(-4:4, -4:4, type = "n")
```

6.2 lines

```
>  X=c(10.0, 8.0, 13.0, 9.0, 11.0, 14.0, 6.0, 4.0, 12.0, 7.0, 5.0),
```

```
>  Y2=c(9.14,8.14, 8.74,8.77,9.26,8.10,6.13,3.10, 9.13,7.26,4.74),

> plot(Y2~X)

# 下面绘制曲线, 注意把顺序调好.
> o <- order(X)
X.o <- X[o]
> Y2.o<-Y2[o]
> lines(X.o,Y2.o,col="red")
```

6.3 boxplot 水平放置

加入参数 horizontal=TRUE

6.4 添加水平或垂直线

垂直线: abline(v=c(...),...)

水平线: abline(h=c(...),...)

```
> x <- rnorm(100)
> plot(1:100~sort(x))
> abline(v = quantile(x), col = "blue", lwd = 3, lty=2)
```

6.5 xy轴反转

```
> x <- rnorm(100)
> plot(1:100~sort(x))
```

6.6 rug-在一边加入显示密度的小短线

```
> x <- rnorm(100)
> plot(sort(x))
> rug(x,side=2)
```

6.7 绘制到x轴的垂直线

加入参数 type = ”h”

6.8 curve–绘制函数曲线

用法: curve(expr, from, to, n = 101, add = FALSE, type = ”l”, ylab = NULL, log = NULL, xlim = NULL, ...)

绘制函数曲线, expr 为一个函数表达式.

用于添加曲线很方便.

```
> curve(sin(x),-10,10)
> curve(dnorm(x),-3,3)
```

6.9 在一幅图上添加另外一幅图

使用 par(fig=...,new=TRUE)

```
> n <- 1000
> x <- rnorm(n)
> qqnorm(x)
```

```
> qqline(x, col="red")
> op <- par(fig=c(.02,.5,.5,.98), new=TRUE)
> hist(x, probability=T,
+     col="light blue", xlab="", ylab="", main="", axes=F)
> lines(density(x), col="red", lwd=2)
> box()
> op
$fig
[1] 0 1 0 1

$new
[1] FALSE

> par(op)
```

6.10 平滑曲线(density)的绘制

可以选择平滑方法: ""gaussian"', ""rectangular"', ""triangular"', ""epanechnikov"', ""biweight"', ""cosine"' or ""optcosine"', with default ""gaussian"

```
> data(faithful)
> attach(faithful)
> hist(eruptions,15,prob=T)
> lines(density(eruptions))
```

6.11 填充颜色

```
> x=seq(-4,4,by=0.1)
> y=dnorm(x)
> x1=seq(3,4,by=0.1)
> x1=x[(length(x)-20):length(x)]
> y1=y[(length(x)-20):length(x)]
```

```
> x2=c(x1,x1[length(x1):1])
> y2=c(y1,rep(0,length(x1)))
> plot(x,y,type='l')
> polygon(x2,y2,col="red")
```

6.12 cex-绘制按照比例大小的图标

```
> x1=rnorm(100)
> x2=rnorm(100)
> x3=rnorm(100)
> m=cor(cbind(x1,x2,x3))
> m
           x1         x2         x3
x1  1.00000000 -0.01499516 0.24657311
x2 -0.01499516  1.00000000 0.07323174
x3  0.24657311  0.07323174 1.00000000
> class(m)
[1] "matrix"
> plot(col(m), row(m), cex=10*abs(m),xlim=c(0, dim(m)[2]+1),ylim=c(0, dim(m)[1]+1
> col(m)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    1    2    3
[3,]    1    2    3
> row(m)
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    2    2    2
[3,]    3    3    3
```

6.13 同时绘制不同数据不同颜色的图

```
> n <- 100
> x <- runif(n)
```

```
> z <- ifelse(x>.5,1,0)
> y <- 2*z -x + .1*rnorm(n)
> plot( y~x, col=c('red','blue')[1+z] )
```

6.14 等高线图(contour)

将一个矩阵的等高线绘制出来. 不论矩阵的数据是什么. 参数 lev 指明要绘制哪些线

```
> z=matrix(rnorm(10000),100,100)
> dim(z)
[1] 100 100
> contour(z, lev=seq(0.1,0.5)
> contour(1:100,1:100,z) # 一样的, 只是改变了x,y轴的坐标表示
> contour(1:100,1:100,z,lev=c(0.1,0.5)) # 只绘制数据为0.1,0.5的线
> contour(1:100,1:100,z,lev=0.1) # 绘制0.1等高线
> contour(1:100,1:100,z,lev=2,add=T,col='red') # 增添0.5等高线(红色)
```

6.15 一页上绘制多个图

```
n <- 100
v <- .1
x1 <- rlnorm(n)
x2 <- rlnorm(n)
x3 <- rlnorm(n)
x4 <- x1 + x2 + x3 + v*rlnorm(n)
m2 <- cbind(x1,x2,x3,x4)
>  par(mfrow=(c(2,2)))
> for (i in 1:4){
+ plot(m2[,i])
```

```
+ }
```

6.16 数学方程式

```
x=1:10
y=x+rnrom(10)
plot(1:10)
text(4, 9, expression(hat(beta) == (X^t * X)^{-1} * X^t * y))
text(4, 8.4, "expression(hat(beta) == (X^t * X)^{-1} * X^t * y)
text(4, 7, expression(bar(x) == sum(frac(x[i], n), i==1, n)))
```

6.17 3D-绘图

rgl 程序包: 绘制 3D 图形必备. 其它仅做参考.

其它: misc3d

Chapter 7

在 python 中调用 R (rpy2)

安装 rpy2

```
Web: http://rpy.sourceforge.net
```

rpy2 与 rpy1.x 使用方法有点不同. 详细参考见网站 user guide

7.1 introduction

下面是如何从 python 中调用 R 命令的例子. (网站 user guide 之 introduction 的部分)

```
# 导入
import rpy2.robjects as robjects
from  rpy2.robjects import r

# 使用常量
In [16]: r['pi']
Out[16]: 3.14159265358979
```

```
In [17]: r('pi')
Out[17]: 3.14159265358979

# 返回的是 tuple, 获取必须使用下标.
# python 里的 add 函数对应 R 里的 c() 函数
In [18]: r('pi')+2
Out[18]: c(3.14159265358979, 2)

In [19]: r('pi')[0]+2
Out[19]: 5.1415926535897931

# 定义和使用函数
In [21]: r('''f <- function(r) { 2 * pi * r }''')
Out[21]:
function (r)
{
   2 * pi * r
}

In [3]: r('f')
Out[3]:
function (r)
{
   2 * pi * r
}

In [23]: r('f(3)') # r['f(3)'] 是错误的, 应该使用 r['f'](3)
Out[23]: 18.8495559215388

In [4]: r['f']
Out[4]:
function (r)
{
   2 * pi * r
}


# 执行一个字符串
In [9]: letters = r['letters']

In [10]: letters
```

```
Out[10]:
c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l",
"m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y",
"z")

In [11]: type(letters)
Out[11]: <class 'rpy2.robjects.RVector'>

In [12]: rcode = 'paste(%s, collapse="-")' %(repr(letters))

In [13]: rcode
Out[13]: 'paste(c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j",
    "k", "l", \n"m", "n", "o", "p", "q", "r", "s", "t", "u", "v",
    "w", "x", "y", \n"z"), collapse="-")'

In [14]: type(rcode)
Out[14]: <type 'str'>

In [15]: r(rcode)
Out[15]: "a-b-c-d-e-f-g-h-i-j-k-l-m-n-o-p-q-r-s-t-u-v-w-x-y-z"


# 创建 rpy2 对象
In [18]: import rpy2.robjects as robjects

In [19]: robjects.StrVector(['abc', 'def'])
Out[19]: c("abc", "def")

In [20]: robjects.IntVector([1, 2, 3])
Out[20]: 1:3

In [21]: robjects.FloatVector([1.1, 2.2, 3.3])
Out[21]: c(1.1, 2.2, 3.3)

In [22]: type(robjects.FloatVector([1.1, 2.2, 3.3]))
Out[22]: <class 'rpy2.robjects.FloatVector'>

# 直接使用函数
In [35]: r.f(3)
Out[35]: 18.8495559215388
```

```
In [36]: r.sum(r.c(1,2,3))
Out[36]: 6L

# 间接使用函数
In [26]: m = robjects.r['matrix'](v, nrow = 2)

In [27]: m
Out[27]: structure(c(1.1, 2.2, 3.3, 4.4, 5.5, 6.6), .Dim = 2:3)

In [28]: type(m)
Out[28]: <class 'rpy2.robjects.RArray'>
# 函数也可以这样使用
In [29]: m = robjects.r('matrix')(v, nrow = 2)

In [30]: m
Out[30]: structure(c(1.1, 2.2, 3.3, 4.4, 5.5, 6.6), .Dim = 2:3)

# 调用函数对象
In [31]: rsum = robjects.r['sum']

In [32]: rsum(robjects.IntVector([1,2,3]))
Out[32]: 6L

In [33]: rsort = robjects.r['sort']
# 可以使用参数
In [34]: rsort(robjects.IntVector([1,2,3]), decreasing=True)
Out[34]: c(3L, 2L, 1L)

# 下面可以当做例程来使用
import rpy2.robjects as robjects
r = robjects.r

import array

x = array.array('i', range(10))
y = r.rnorm(10)

r.X11()

r.layout(r.matrix(array.array('i', [1,2,3,2]), nrow=2, ncol=2))
r.plot(r.runif(10), y, xlab="runif", ylab="foo/bar", col="red")
```

```
kwargs = {'ylab':"foo/bar", 'type':"b", 'col':"blue", 'log':"x"}
r.plot(x, y, **kwargs)

# s4 类
import rpy2.robjects as robjects
import array

r = robjects.r

r.setClass("Track",
         r.representation(x="numeric", y="numeric"))

a = r.new("Track", x=0, y=1)

a.x

# 下面代码未经测试
# 下面是一个线性回归的例子
# The R 代码:
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)

anova(lm.D9 <- lm(weight ~ group))

summary(lm.D90 <- lm(weight ~ group - 1))# omitting intercept

# 使用 rpy2.robjects
import rpy2.robjects as robjects

r = robjects.r

ctl = robjects.FloatVector([4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14])
trt = robjects.FloatVector([4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69])
group = r.gl(2, 10, 20, labels = ["Ctl","Trt"])
weight = ctl + trt

robjects.globalEnv["weight"] = weight
robjects.globalEnv["group"] = group
```

```
lm_D9 = r.lm("weight ~ group")
print(r.anova(lm_D9))

lm_D90 = r.lm("weight ~ group - 1")
print(r.summary(lm_D90))
```

7.2 把 python 数据转换为 R 可用的数据

robjects 有几个函数执行转换, 下面是常用的几个

```
BoolVector
FloatVector
IntVector
RArray
RDataFrame
RFormula
RFunction
RMatrix
RVector
StrVector
```

下面是几个例子 (RFormula 的用法参考下一节线性回归的例子)

```
# 导入
import rpy2.robjects as robjects
from  rpy2.robjects import r

In [3]: robjects.IntVector(range(10))
Out[3]: 0:9

In [4]: s="ATGCCCGTTAAAGGGTT"

In [5]: robjects.StrVector(s)
```

```
Out[5]:
c("A", "T", "G", "C", "C", "C", "G", "T", "T", "A", "A", "A",
"G", "G", "G", "T", "T")
```

7.3 执行 R 运算

定义和使用函数对象

```
# 导入
import rpy2.robjects as robjects
from  rpy2.robjects import r

fun=r('''f <- function(r) { 2 * pi * r }''')
fun(3)
r['f'](3)
r('f(3))

rnorm = robjects.r.rnorm
rnorm(100)
rnorm(100,mean=1)
```

两种方法运行 R 函数

```
# 直接使用字符串
r("t.test(c(1,2,3))")

# 使用参数
r['t.test'](r.c(1,2,3))

# 参数可以事先准备好
x=robjects.IntVector([1,2,3])
r['t.test'](x,mu=1)
```

下面是一个线性回归的例子

```
z=r('rnorm(100)')
l=len(z)
Z=z.r
N=robjects.IntVector(range(0,l))
res=r.lm("Z~N-1") # r('lm(Z~N-1)') 也可以

# 使用 RFormula
fmla = robjects.RFormula('Z~N-1')
env = fmla.getenvironment()
env['Z']=z # 不需要将 Z = z.r
env['N']=robjects.IntVector(range(0,l))
fit = robjects.r.lm(fmla)
```

7.4 将 R 结果提取到 python

subset 函数为 R 对象中提取函数. getnames() 查看 R 对象里有什么可以提取的 names.

```
# 接上面线性回归的例子
# 查看 R 对象里有什么可以提取的 names
res.getnames()
# Out[229]:
# c("coefficients", "residuals", "effects", "rank", "fitted.values",
   "assign", "qr", "df.residual", "xlevels", "call", "terms", "model"
   )

res.subset("coefficients")[0][0]
# Out[218]: 0.35289430254791571

len(res.subset("coefficients"))
# Out[232]: 1
```

Part II

基本统计分析

“基本统计分析”参考文献除了[11]，R部分主要参考了《simpleR》《Statistics with R》等。

Chapter 8

数据变换

数据变换的目的大概有三种

1. 稳定方差
2. 直线化
3. 使分布正态或接近正态

如果一个变换 $y = f(x)$ 是x的线性函数, 则不影响分析. 但是, 如果是非线性函数, 则y就会表现的和x完全不同, 包括分布方差及数据间的关系.

8.1 delta 方法–随机变量函数的方差

若x,y的非线性函数分别是$f(x).f(x,y)$, 且σ_x^2, σ_y^2已知, 当x,y渐近正态分布且n_x, n_y较大时有

$$var[f(x)] \approx (\frac{df}{dx})^2 var(x)$$

$$var[f(x,y)] \approx (f'(x)^2 var(x) + (f'(y)^2 var(y) + 2(f'(x)(f'(y)cov(x,y)$$

这就是著名的 delta 方法. ([11] Page 556. [12] 第二章)

假定原变量为x, 应用变换 $y = f(x)$, 当x变异系数较小时, 应用第一个式子有

$$var(y) \approx (f'(x)^2 var(x)$$

欲使$var(y)$为常数c, 则应使

$$f'(x) = \frac{c}{\sqrt{var(x)}} = \frac{c}{s}$$

此时可以求得变换

$$y = f(x)$$

其中$var(y) = c$为常数

8.2 Box-Cox变换

8.2.1 茆诗松的定义

[1]Box与Cox(1964)从实际数据出发提出了一个很有效的变换, 把常用变换作为其特例包含其中, 称为Box-Cox变换. 变换如下

$$y = \begin{cases} x^k & if k \neq 0 \\ lnx & if k = 0 \end{cases}$$

Box-Cox变换有如下特点

- 可以改变分布形状, 使之正态分布, 至少是对称的
- 当$x >= 0$, 能够保持数据的大小次序
- 对变换结果可以有很好的解释

[1]茆诗松. 试验设计. Page 60

  - k=2 为平方变换
  - k=1 为恒等变换
  - k=0.5 平方根变换
  - k=0 对数变换
  - k=-0.5 平方根倒数变换
  - k=-1 倒数变换
- 变换是对k连续的
- 注意: 当$x_{max}/x_{min} > 2$时, 特别有效. $x_{max}/x_{min} <= 2$时无效.

关键是寻找k值, 使变换后的数据正态分布. Montgomery在他的书[2]中提出, k的极大似然估计就是使$y_1, y_2, \cdots, y_n$的偏差平方和$Q(k) = \sum(y_i - \bar{y})^2$达到最小的k值. 可以画出Q(k)的曲线, 读出Q(k)最小的k值即可. 也可以选择10-20个k值, 选择Q(k)最小的k值. 若需要进一步精确估计, 则使用精确网络进一步迭代.

8.2.2 R的定义

经过查询R和百度, 发现定义与茆诗松的描述稍微不同. 变换为

$$y = \begin{cases} (x^k - 1)/k & if k \neq 0 \\ lnx & if k = 0 \end{cases}$$

下面是R的一个例子, 使用最大似然法. box.cox.powers 使用最大似然法按照上面的公式计算指数值. 之后使用box.cox函数得到变换结果.

```
library(car)
attach(Prestige)
# 两个变量会计算多元正态分布
box.cox.powers(cbind(income, education))
par=matrix(c(1,2))
```

[2]Montgomery. 实验设计与分析(第三版). 1998.

```
plot(income, education)
plot(box.cox(income, .26), box.cox(education, .42))
# 单变量会直接转换为正态分布
box.cox.powers(income)
qq.plot(income) # car 包的绘图函数
qq.plot(income^.18)
```

还有一个扩展形式, 其a值比较明显, 还是估计k值的问题.

$$y = \begin{cases} ((x+a)^k - 1)/k & if k \neq 0 \\ lnx & if k = 0 \end{cases}$$

在Box和Cox论文中采用了两种方法，其一是最大似然估计，其二是Bayes方法。

8.3 稳定方差的变换

8.3.1 对数变换-方差正比于自变量的平方

当$var(x) \propto x^2$, 即x增大, $var(x)$也增大, 那么有

$$s_x = kx$$

此时x的变异系数$cv = s/\bar{x} = c$为常数.

将$s = kx$带入下式

$$f'(x) = \frac{c}{\sqrt{var(x)}} = \frac{c}{s}$$

合并常数项后有

$$f'(x) = \frac{c}{x} \Longrightarrow f(x) = lg(x)$$

即$y = lg(x)$使方差稳定.

使用的时候, 注意: $x_i <= 0$时不能使用. 将$x_i <= 0$替换为$-x_i$即可. 若有$x = 0$, 常常用$lg(x + 1)$代替$lg(x)$, $x = 0$时, $y = 0$.

```
> rep(1:10,10)
  [1]  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5
 [26]  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10
 [51]  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5
 [76]  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10
> matrix(rep(1:10,10),nc=10) # nr =10也一样
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    1    1    1    1    1    1    1    1    1     1
 [2,]    2    2    2    2    2    2    2    2    2     2
 [3,]    3    3    3    3    3    3    3    3    3     3
 [4,]    4    4    4    4    4    4    4    4    4     4
 [5,]    5    5    5    5    5    5    5    5    5     5
 [6,]    6    6    6    6    6    6    6    6    6     6
 [7,]    7    7    7    7    7    7    7    7    7     7
 [8,]    8    8    8    8    8    8    8    8    8     8
 [9,]    9    9    9    9    9    9    9    9    9     9
[10,]   10   10   10   10   10   10   10   10   10    10
> c(t(matrix(rep(1:10,10),nc=10)))
  [1]  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3
 [26]  3  3  3  3  3  4  4  4  4  4  4  4  4  4  4  5  5  5  5  5  5  5  5  5  5
 [51]  6  6  6  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  7  7  8  8  8  8  8
 [76]  8  8  8  8  8  9  9  9  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10
> s=c(t(matrix(rep(1:10,10),nc=10)))
> x=rnorm(100)
> plot(y<-x*s) # y的方差与x呈正比
> plot(log(y)+10,ylim=c(0,15)) # 查看log变换的y方差基本一致
```

8.3.2 平方根变换-方差正比于自变量

若$var(x) \propto x$, 则有

$$f'(x) = \frac{c}{\sqrt{x}} \Longrightarrow f(x) = \sqrt{x}$$

例如 $var(x) = kx \Longrightarrow var(y) = [(\sqrt{x})']^2(kx) = k/4$, 可见方差稳定.

当x绝对值较小时,常用$y=\sqrt{x+1}$或$y=\sqrt{x}+\sqrt{x+1}$, 效果好于$y=\sqrt{x}$.

```
> s1=c(t(matrix(rep((1:20)^0.5,10),nc=10)))
> x1=rnorm(200)
> plot(s1)
> plot(y1<-s1*x1) # 方差随x变大
> plot(sqrt(y1)) # 看到方差已经基本一致
```

8.3.3 反正弦变换(角变换)-百分率表示的数据

应用于以百分率表示的数据. 即在n次试验中, 成功k次, 则有

$$p=\frac{k}{n}$$

$$var(p)=p(1-p)/n$$

即$var(p)$与p有关. 我们有

$$f'(p)=\frac{c}{\sqrt{p(1-p)}}\Longrightarrow y=sin^{-1}\sqrt{p}$$

可以验证

$$var(y)=(\frac{c}{\sqrt{p(1-p)}})^2\frac{p(1-p)}{n}=\frac{c^2}{4n}$$

注意, 当多组样本数不同时, 使用各自的样本数加权.

8.3.4 倒数变换-方差正比于自变量4次方

若$var(x)\propto x^4$,即$s=kx^2$, x增加时, var(x)增加很快. 我们有

$$f'(x)=\frac{c}{x^2}\Longrightarrow y=\frac{1}{x}$$

这样, x增加到一定程度后y的减小微不足道. 验证

$$var(y) = (\frac{c}{x^2})^2 kx^4 = kc^2$$

常常用于质反应时间为指标的数据

8.4 量反应直线化

某些数据上凸或下凹, 则使用多种变换尺度(metameter)使之直线化. 理想的变换应该使

1. y与x呈直线关系
2. var(y) 稳定
3. 直线斜率较大

8.4.1 对数变换

又分为两种, 单对数变换:

$$y = a + b * lg(x)$$

双对数变换:

$$lg(y) = a + b * lg(x)$$

8.4.2 平方根变换

当对数变换后仍然开始较陡峭, 以后平坦(变换前适合二次拟合), 则使用下面的变换:

$$\sqrt{y} = a + b * lg(x)$$

变换后方差也较变换前稳定

8.4.3 倒数变换

单倒数变换.

下面是一个蛋白质与底物反应的例子. 设x为游离蛋白质浓度, y为游离底物浓度, m为结合物浓度. X=x+m为总蛋白质浓度, Y=y+m为总底物浓度. 已知X,Y, 可以测得y,m,但x不易测定. 根据质量作用定律

$$\frac{xy}{m} = k$$

此处k为常数. k大,说明容易结合, 否则不易结合. 欲得到x或y与k的关系(视测量难度选取容易的). 测得$(m_1, y_1), \cdots, (m_n, y_n)$. 以此做曲线拟合. 由上式

$$\frac{x}{k} = \frac{m}{y} \Longrightarrow \frac{X-m}{k} = \frac{m}{y} \Longrightarrow \frac{m}{y} = \frac{m}{-k} + \frac{X}{k}$$

可以看到$\frac{m}{y}$与m呈直线关系. 使用$\frac{m}{y}$与m作图可以得到一条直线.

双倒数作图法(double reciprocal, Linewaver-Burk plot) 药物动力学常常符合此种情况.

设x为药物浓度, y为效应, k为解离常数, $a = y_{max}$为内在活性. 根据Ariens学说有

$$y = \frac{ax}{x+k} \Longrightarrow \frac{1}{y} = \frac{1}{a} + \frac{k}{a}\frac{1}{x}$$

即$1/y, 1/x$呈线性关系. 此时由数据线性回归即可求得k.

8.5 质反应直线化

质反应是反应特定反应的有无,死活等离散数据的反应. 反应特点一般呈S曲线. 其成功概率

$$p \sim N(\hat{p}, \hat{p}\hat{q}/n)$$

标准正态偏离定义为

$$z_p = \frac{p - \hat{p}}{\sqrt{\hat{p}\hat{q}/n}}$$

8.5.1 probit变换(概率单位变换)

probit变换又叫做概率单位变换(probability unit). probit定义为$y = 5 + z$, 可以使S曲线直线化. 但是$var(y) \neq c$常数. 故不适合最小二乘法. 常常使用最大似然法(maximum likelyhood, ML)

8.5.2 角变换

$y = sin^{-1}_{\sqrt{p}}$可以使S曲线直线化. 当$p \in [0, 1]$时, $y \in [0, \pi/2]$. 即p等距离变化时, y的两端变化大, 中间小, 使S曲线拉直.

$$var(y) = \frac{820.7}{n(90 * 2/\pi)^2}$$

故n不变时, 比probit变换易于分析.

8.5.3 logit变换

p的logit变换定义为

$$y = ln\frac{p}{1-p} \quad or \quad y = \frac{1}{2} + ln\frac{p}{1-p}$$

其效果与probit相似. 当$p = 0, 1$时,$y = -\infty, \infty$. 故修正为

$$y = ln\frac{r + 1/2}{n - r + 1/2}$$

其中$r = np$.

8.6 相关系数的正态化变换–Fisher变换(Z变换)

参考回归部分 chapter 23章 section 23.6节

8.7 正态化变换的方法

很多右偏数据可以正态化.

对数变换后呈正态分布, 又称对数正态分布, 方差稳定.

不太严重的右偏, 使用平方根变换

严重右偏, 倒数变换

8.8 数据挖掘中的变换

数据变换将数据转换或统一成适合于挖掘的形式。数据变换可能涉及如下内容：

- 光滑：去掉数据中的噪声。这种技术包括分箱、回归和聚类。
- 聚集：对数据进行汇总或聚集。例如，可以聚集日销售数据，计算月和年销售量。通常，这一步用来为多粒度数据分析构造数据立方体。
- 数据泛化：使用概念分层，用高层概念替换低层或“原始”数据。例如，分类的属性，如街道，可以泛化为较高层的概念，如城市或国家。类似地，数值属性如年龄，可以映射到较高层概念如青年、中年和老年。
- 规范化：将属性数据按比例缩放，使之落入一个小的特定区间，如－1.0～1.0或0.0～1.0。

Chapter 9

统计函数表与概率分布函数的用法

参考 r cran task view: distribution 里有其它分布的函数与包的介绍, 包括多元正态分布, 多元t分布等.

9.1 统计函数表

在统计学中，产生随机数据是很有用的，R可以产生多种不同分布的随机数序列。这些分布函数的形式为rfunc(n,p1,p2,...)，其中func指概率分布函数，n为生成数据的个数，p1, p2, . . . 是分布的参数数值。大多数这种统计函数都有相似的形式，只需用d、p或者q去替代r，比如密度函数(dfunc(x,...))，累计概率密度函数（也即分布函数）(pfunc(x,...))和分位数函数(qfunc (p, ...)，0 ¡ p ¡ 1)。最后两个函数序列可以用来求统计假设检验中P值或临界值。

概率分布	R 对应的名字	附加参数
β分布	beta	shape1, shape2, ncp
二项式分布	binom	size, prob
Cauchy 分布	cauchy	location, scale
卡方分布	chisq	df, ncp

指数分布	exp	rate
F分布	f	df1, df1, ncp
γ分布	gamma	shape, scale
几何分布	geom	prob
超几何分布	hyper	m, n, k
对数正态分布	lnorm	meanlog, sdlog
logistic 分布	logis	location, scale
负二项式分布	nbinom	size, prob
正态分布	norm	mean, sd
Poisson 分布	pois	lambda
t 分布	t	df, ncp
均匀分布	unif	min, max
Weibull 分布	weibull	shape, scale
Wilcoxon 分布	wilcox	m, n

9.2 简单抽样

更复杂的抽样使用 MCMC.

重复和不重复的采样（放回和非放回的）

```
sample(x, size, replace = FALSE, prob = NULL)
> x <- 1:100
> sample(x,10)
[1] 96 60 86 43 30 81 26 24 94 28
> y <- 1:6  # 掷骰子
> sample(y,4,replace=TRUE)
[1] 5 1 5 1
```

9.2.1 放回式抽样

```
sample(x, size, replace = FALSE, prob = NULL)
> x=c('y','n')
> sample(x,10,replace=TRUE)
```

```
[1] "y" "n" "y" "n" "y" "n" "n" "y" "n" "n"
> y=c(1,2)
> sample(y,10,replace=TRUE)
[1] 1 1 2 1 1 1 1 1 1 2
```

9.2.2 非放回式抽样

```
> x=1:9
> sample(x,3)
[1] 4 3 1
# 只有一个参数时，相当于 shuffle
> sample(x)
[1] 6 8 4 7 2 1 5 3 9
# replace=TRUE 时，采样数目 = length(x)
> sample(x,replace=TRUE)
[1] 9 7 5 8 8 2 1 9 8
```

9.3 贝努里分布（Bernoulli distribution）

贝努里分布（Bernoulli distribution）也称单点分布, 默认为 p=0.5

P(X=1) = p

P(X=0) = 1-p

```
> n <- 200
> x <- sample(c(-1,1), n, replace=T, prob=c(.2,.8))
> plot(cumsum(x),type='l')
```

9.4 均匀分布（Uniform discrete distribution）

使用 sample 模拟

```
> sample(1:10, 20, replace=T)
[1]  7 10 10  4  6  8  6  6  4  8  1  3  9 10  9  8  3  4 10 10
```

9.5 二项分布

帮助 help(rbinom)

9.5.1 产生二项分布随机数

rbinom(n, size, prob)

n 为产生的随机数的个数(可以大于 size), prob 为单点分布(Bonulli 分布)的成功的概率. size 为二项分布的试验次数, 成功 x 的概率为:

```
p(x) = choose(n,x) p^x (1-p)^(n-x)
> rbinom(5,10,0.5)
[1] 7 4 7 5 5
> rbinom(5,10,0.1)
[1] 1 2 3 0 1

> dbinom(5,10,p=0.5)
[1] 0.2460938
```

当size取大于1的值时，结果似乎会产生0,1,2,...,size的正态分布

```
> x=rbinom(10000,9,0.5)
```

```
> table(x)/length(x)
x
   0     1     2     3     4     5     6     7     8     9
0.0028 0.0203 0.0704 0.1605 0.2541 0.2401 0.1623 0.0701 0.0176 0.0018
> table(rbinom(10000,10,0.3))

0    1    2    3    4    5    6    7    8
261 1272 2327 2655 1987 1033  355   91   19
```

9.5.2 期望-方差值

二项分布的期望为$E(X) = np$. 方差为$Var(X) = npq$.

9.5.3 概率密度函数

在n次试验中事件A发生x次的概率$p(x)$，p为单次试验事件A发生的概率公式：
$p(x) = \binom{n}{x} p^x (1-p)^{(n-x)}$

设100次试验，A发生的概率为0.3,A发生20次的概率为：

```
> dbinom(20, 100, 0.3)
[1] 0.007575645
```

A发生20 <=<= 60次的概率为：

```
> sum(dbinom(20:60, 100, 0.3))
[1] 0.9911128
```

其他

```
> dbinom(1,2,0.5)
[1] 0.5
> dbinom(0,2,0.5)
[1] 0.25
> dbinom(2,2,0.5)
[1] 0.25
```

9.5.4 累积概率密度函数及图

```
> pbinom(60,100,0.5)-pbinom(39,100,0.5)
[1] 0.9647998
> pbinom(6,10,0.5)-pbinom(3,10,0.5)
[1] 0.65625
# 也可以使用下面
> sum(dbinom(40:60, 100, 0.5))
[1] 0.9647998
> sum(dbinom(4:6, 10, 0.5))
[1] 0.65625
```

最后画出密度和累积密度的图

```
> plot(dbinom(0:100,100,0.5))
> plot(pbinom(0:100,100,0.5))
```

9.5.5 指定累积概率的q值

求成功概率为0.2, 总次数为10, 指定累积概率为0.5的试验次数为

```
> qbinom(p=0.5,size=10,prob=0.2)
[1] 2
# 检验
```

```
> pbinom(q=2,size=10,prob=0.2)
[1] 0.6777995
> pbinom(q=1,size=10,prob=0.2)
[1] 0.3758096
> pbinom(q=3,size=10,prob=0.2)
[1] 0.8791261
```

9.6 泊松分布

9.6.1 产生泊松分布随机数

rpois(n, lambda) n 为要产生随机数的个数, lambda 为 poisson 的参数.

9.6.2 期望和方差

具有参数 lambda 的泊松分布的均值和期望均为 lambda.

9.6.3 密度-累积概率密度函数

```
ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)
dpois(x, lambda, log = FALSE)
```

```
> x=1:10
> dpois(x,3)
 [1] 0.1493612051 0.2240418077 0.2240418077 0.1680313557 0.1008188134
 [6] 0.0504094067 0.0216040315 0.0081015118 0.0027005039 0.0008101512
> ppois(x,3)
 [1] 0.1991483 0.4231901 0.6472319 0.8152632 0.9160821 0.9664915 0.9880955
 [8] 0.9961970 0.9988975 0.9997077
```

9.6.4 指定累积概率的q值

```
qpois(p, lambda, lower.tail = TRUE, log
.p = FALSE)

> x=seq(0,1,0.1)
> x
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> qpois(x,3)
 [1]   0   1   2   2   2   3   3   4   4   5 Inf
```

9.7 超几何分布（Hypergeometric distribution）

非放回抽样的分布为超几何分布.

rhyper(nn, m, n, k) m: 白球的数目. n: 黑球的数目. k: 抽出球的数目. nn: 观察的次数.

```
> rhyper(10,15,5,5)
[1] 3 3 3 3 5 2 5 4 3 4
```

9.8 正态分布

9.8.1 产生正态分布随机数

```
rnorm(n, mean=0, sd=1)

> rnorm(10,0,1)
 [1]  0.9944192 -0.1384374 -0.8876501  1.0416947 -0.3217919 -0.8546145
 [7] -2.0329649 -0.5276146  0.1380986 -0.8563042
```

9.8.2 期望和方差

期望为均值, 方差为方差

9.8.3 密度-累积概率密度函数

```
dnorm(x, mean=0, sd=1, log = FALSE)
pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
```

```
> dnorm(x)
 [1] 0.2419707 0.2660852 0.2896916 0.3122539 0.3332246 0.3520653 0.3682701
 [8] 0.3813878 0.3910427 0.3969525 0.3989423 0.3969525 0.3910427 0.3813878
[15] 0.3682701 0.3520653 0.3332246 0.3122539 0.2896916 0.2660852 0.2419707
> pnorm(x)
 [1] 0.1586553 0.1840601 0.2118554 0.2419637 0.2742531 0.3085375 0.3445783
 [8] 0.3820886 0.4207403 0.4601722 0.5000000 0.5398278 0.5792597 0.6179114
[15] 0.6554217 0.6914625 0.7257469 0.7580363 0.7881446 0.8159399 0.8413447
```

9.8.4 指定累积概率的q值

```
qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
```

```
> x=seq(0,1,0.1)
> qnorm(x)
 [1]       -Inf -1.2815516 -0.8416212 -0.5244005 -0.2533471  0.0000000
 [7]  0.2533471  0.5244005  0.8416212  1.2815516        Inf
```

9.8.5 转换非标准正态分布到标准正态分布

具有均值为μ标准差为σ的正态分布变量 x, 可以使用下面的

公式变换为标准正态分布

$$Z = \frac{x - \mu}{\sigma}$$

9.9 t分布

9.9.1 产生t分布的随机数

产生10个自由度为5的t分布随机数.

```
> rt(n=10,df=5)
 [1]  0.7965116  0.9019405  0.2392244  0.3129466 -0.2910085 -1.2970800
 [7]  1.4356046  0.1165443  0.9069540  0.3450907
```

9.9.2 密度-累积概率密度函数

```
   dt(x, df, ncp=0, log = FALSE)
   pt(q, df, ncp=0, lower.tail = TRUE, log.p = FALSE)
```

```
> x=-5:5
> x
 [1] -5 -4 -3 -2 -1  0  1  2  3  4  5
> dt(x,df=20)
 [1] 0.0000789891 0.0008224743 0.0079637866 0.0580872152 0.2360456491
 [6] 0.3939885857 0.2360456491 0.0580872152 0.0079637866 0.0008224743
[11] 0.0000789891
> pt(x,df=20)
 [1] 3.436514e-05 3.517616e-04 3.537949e-03 2.963277e-02 1.646283e-01
 [6] 5.000000e-01 8.353717e-01 9.703672e-01 9.964621e-01 9.996482e-01
[11] 9.999656e-01
```

9.9.3 指定累积概率的q值

产生累积概率为0.025, 0.975的自由度为20的t分布的值

```
> qt(p=0.025,df=20)
[1] -2.085963
> qt(p=0.975,df=20)
[1] 2.085963
```

9.10 χ^2分布

9.10.1 产生χ^2分布的随机数

产生10个自由度为20的χ^2分布的随机数

```
> rchisq(n=10,df=20)
 [1] 13.26240 20.74800 17.96519 14.57688 16.04691 28.31448 16.28799 32.64230
 [9] 13.38085 15.97800
```

9.10.2 密度-累积概率密度函数

```
   dchisq(x, df, ncp=0, log = FALSE)
   pchisq(q, df, ncp=0, lower.tail = TRUE
, log.p = FALSE)

> x=0:10
> x
 [1]  0  1  2  3  4  5  6  7  8  9 10
> dchisq(x,df=5)
 [1] 0.00000000 0.08065691 0.13836917 0.15418033 0.14397591 0.12204152
 [7] 0.09730435 0.07437127 0.05511196 0.03988664 0.02833456
> pchisq(x,df=5)
```

```
 [1] 0.00000000 0.03743423 0.15085496 0.30001416 0.45058405 0.58411981
 [7] 0.69378108 0.77935969 0.84376437 0.89093584 0.92476475
```

9.10.3 指定累积概率的q值

```
> qchisq(p=0.025,df=5)
[1] 0.8312116
> qchisq(p=0.975,df=5)
[1] 12.83250
```

Chapter 10

描述性统计

10.1 探索性分析

也可以叫做经验性数据分析. 目的是看一看数据适合哪一种统计模型. 对于单变量数据, 我们可以看看它的分布是否正态, 尾部偏大还是偏小, 对称还是偏态.

主要的工具就是图形工具.

- barplots for categorical data(类型数据)
- histogram, dot plots, stem and leaf plots to see the shape of numerical distributions
- boxplots to see summaries of a numerical distribution, useful in comparing distributions and identifying long and short-tailed distributions.
- normal probability plots To see if data is approximately normal

10.2 样本特征数

若非特别说明， 数据使用这个

```
> x=exp(seq(-1,3,by=0.1))
> x
 [1]  0.3678794  0.4065697  0.4493290  0.4965853  0.5488116  0.6065307
 [7]  0.6703200  0.7408182  0.8187308  0.9048374  1.0000000  1.1051709
[13]  1.2214028  1.3498588  1.4918247  1.6487213  1.8221188  2.0137527
[19]  2.2255409  2.4596031  2.7182818  3.0041660  3.3201169  3.6692967
[25]  4.0552000  4.4816891  4.9530324  5.4739474  6.0496475  6.6858944
[31]  7.3890561  8.1661699  9.0250135  9.9741825 11.0231764 12.1824940
[37] 13.4637380 14.8797317 16.4446468 18.1741454 20.0855369
> plot(x)
```

10.2.1 方差

```
> var(x)
[1] 29.35325
```

10.2.2 标准差

```
> sd(x)
[1] 5.417864
# 可以手工计算验证一下
> sqrt((sum(x^2)-(sum(x))^2/length(x))/(length(x)-1))
[1] 5.417864
```

10.2.3 最大最小值

```
> max(x)
[1] 20.08554
> min(x)
[1] 0.3678794
```

10.2.4 累积最大最小值

```
> cummax(x)
 [1]  0.3678794  0.4065697  0.4493290  0.4965853  0.5488116  0.6065307
 [7]  0.6703200  0.7408182  0.8187308  0.9048374  1.0000000  1.1051709
[13]  1.2214028  1.3498588  1.4918247  1.6487213  1.8221188  2.0137527
[19]  2.2255409  2.4596031  2.7182818  3.0041660  3.3201169  3.6692967
[25]  4.0552000  4.4816891  4.9530324  5.4739474  6.0496475  6.6858944
[31]  7.3890561  8.1661699  9.0250135  9.9741825 11.0231764 12.1824940
[37] 13.4637380 14.8797317 16.4446468 18.1741454 20.0855369
> cummin(x)
 [1] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
 [8] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
[15] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
[22] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
[29] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
[36] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
```

10.2.5 差分

```
> diff(x)
 [1] 0.03869022 0.04275930 0.04725634 0.05222633 0.05771902 0.06378939
 [7] 0.07049817 0.07791253 0.08610666 0.09516258 0.10517092 0.11623184
[13] 0.12845605 0.14196589 0.15689657 0.17339753 0.19163391 0.21178822
[19] 0.23406218 0.25867872 0.28588420 0.31595090 0.34917974 0.38590330
[25] 0.42648910 0.47134335 0.52091497 0.57570007 0.63624698 0.70316166
[31] 0.77711381 0.85884359 0.94916896 1.04899393 1.15931758 1.28124407
[37] 1.41599369 1.56491505 1.72949860 1.91139155
```

10.2.6 平均值

计算平均值的时候，无论 x 是多少维的，都计算所有的 x 的值

```
> y=array(1:20,dim=c(4,5))
> y
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> mean(y)
[1] 10.5
> colMeans(y) # 行均值
[1]  2.5  6.5 10.5 14.5 18.5
> rowMeans(y) # 列均值
[1]  9 10 11 12
```

10.2.7 中位数

```
> median(x)
[1] 2.718282
```

10.2.8 众数

```
> y=c(1,1,2,2,2,3,4)*2
> y
[1] 2 2 4 4 4 6 8
> table(y)
y
2 4 6 8
2 3 1 1
> max(table(y)) # 众数出现的次数
[1] 3
> table(y)==max(table(y))
y
    2     4     6     8
FALSE  TRUE FALSE FALSE
> which(table(y)==max(table(y))) # 众数在table(y)第几个? 第2个
```

```
4
2
```

10.2.9 偏斜度(skewness)

R 没有偏斜度的函数，只好自己编一个，不过很简单，由偏斜度公式

$$m_3 = \frac{\sum(x-\bar{x})^3}{n}$$

编写函数(偏斜度)：

```
skewness<-function(x){
   sum(((x-mean(x))^3))/length(x)
}
# 计算结果
> skewness(x)
[1] 197.8397
```

10.2.10 峭度(kurtosis)

R 没有峭度的函数，只好自己编一个，不过很简单。

4阶中心距

$$m_4 = \frac{\sum(x-\bar{x})^4}{n}$$

2阶中心距

$$m_2 = \frac{\sum(x-\bar{x})^2}{n}$$

峭度

$$g2 = \frac{m^4}{m2^2} - 3$$

编写函数(峭度):

```
kurtosis<-function(x){
   a=mean(x)
   n=length(x)
   m4=sum((x-a)^4)/n
   m2=sum((x-a)^2)/n
   kurt=m4/m2^2 -3
   kurt
}
# 计算结果
> kurtosis(x)
[1] 0.6260693
```

10.2.11 变异系数(coefficient of variability)

公式

$$CV = \frac{sd(x)}{\bar{x}}$$

编写函数(变异系数)

```
CV<-function(x){
   sd(x)/mean(x)
}
> CV(x)
[1] 1.070169
```

10.2.12 异常(极端)值

异常值:

x >上百分位数+1.5×（上百分位数-下百分位数）
x <下百分位数-1.5×（上百分位数-下百分位数）

极端异常值:

x >上百分位数+3×（上百分位数-下百分位数）
x <下百分位数-3×（上百分位数-下百分位数）

```
# 分位数
> q=quantile(x,c(.25,.75)); q
    25%     75%
1.000000 7.389056

# 异常值下侧界限, 故x没有下侧异常值
> out.low=q[1]-1.5*(q[2]-q[1]);out.low
     25%
-8.583584

# 异常值上侧界限, x有上侧异常值
> out.upper=q[1]+1.5*(q[2]-q[1]);out.upper
    25%
10.58358

# 绘图来查看, 可以看到x的上侧异常值
> boxplot(x)
```

10.3 离散数据(Categorical data)

10.3.1 列表:table()

table可以作用于单个因子, 及多个因子. 2因子的会产生2维频数分布表, 相应k因子会产生k维频数分布表.

```
> x
[1] 1 1 2 0 2 0 0 1 1 0
> y=sample(c('y','n'),10,replace=TRUE)
> y
[1] "n" "y" "y" "y" "y" "y" "n" "y" "n" "y"

> table(x)
x
0 1 2
4 4 2
> table(y)
y
n y
3 7
> table(x,y)
y
x  n y
0 1 3
1 2 2
2 0 2

> x=c("Yes","No","No","Yes","Yes")
> table(x)
x
No Yes
2  3
> y=1:9
> table(y)
y
1 2 3 4 5 6 7 8 9
1 1 1 1 1 1 1 1 1
```

10.3.2 factor()函数

```
> factor(x)
[1] Yes No  No  Yes Yes
Levels: No Yes
> factor(y)
[1] 1 2 3 4 5 6 7 8 9
Levels: 1 2 3 4 5 6 7 8 9
> table(x)/length(x)
x
No Yes
0.4 0.6
```

10.3.3 gl()函数

gl() 函数可以方便的产生因子, 一般用法为

```
gl(n, k, length = n*k, labels = 1:n, ordered = FALSE)
```

```
> gl(3,5)
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels: 1 2 3

> gl(3,1,15)
 [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
Levels: 1 2 3
```

10.3.4 条形图，饼图

绘制因子频率(factor)

```
> b = scan()
1:  3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1
```

```
26:
Read 25 items
> barplot(b) # not correct
> barplot(table(b)) # right
> barplot(table(b)/length(b))
> b.count=table(b) # 存于一个变量中
> pie(b.count)
> names(b.count)=c("a","b","c") # 命名
> pie(b.count)
> pie(b.count,col=c("purple","green","cyan","white")) # 改变颜色
```

10.3.5 折线图

```
# 好象需要强制转换一下
> x=as.numeric(t)
> lines(x)
```

10.4 连续数据(numerical data)

```
> s = scan() # 工资
1: 12 .4 5 2 50 8 3 1 4 0.25
11:
Read 10 items
> s
[1] 12.00  0.40  5.00  2.00 50.00  8.00  3.00  1.00  4.00  0.25
```

10.4.1 fivenum

最小, 0.25 , 0.5 0.75, 最大的 5 个数

```
> fivenum(s) # min, lower hinge, Median, upper hinge, max
[1]  0.25  1.00  3.50  8.00 50.00
```

10.4.2 summary

```
> summary(s)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.250   1.250   3.500   8.565   7.250  50.000
```

10.4.3 分位数

最小的值为 0, 最大的为 1, %u 为 min+(max-min)*u

```
> quantile(s) # 分位数
0%   25%   50%   75%  100%
0.25  1.25  3.50  7.25 50.00
> quantile(s,.25) # 分位数
25%
1.25
> quantile(s,c(.25,.75))
25%  75%
1.25 7.25
> sort(s)
[1]  0.25  0.40  1.00  2.00  3.00  4.00  5.00  8.00 12.00 50.00
```

10.4.4 条件性测量

```
> mean(s,trim=1/10)
[1] 4.425
> mean(s,trim=2/10)
[1] 3.833333
```

```
> IQR(s) #  interquartile range is the difference of the 3rd and 1st quartile.
[1] 6
```

10.4.5 茎叶图

```
> stem(s)

The decimal point is 1 digit(s) to the right of the |

0 | 00123458
1 | 2
2 |
3 |
4 |
5 | 0
```

10.4.6 直方图

```
> x=scan()
1: 29.6 28.2 19.6 13.7 13.0 7.8 3.4 2.0 1.9 1.0 0.7 0.4 0.4 0.3
15: 0.3 0.3 0.3 0.3 0.2 0.2 0.2 0.1 0.1 0.1 0.1 0.1
27:
Read 26 items
> a=hist(x) # 频率
> hist(x,probability=TRUE) # 密度

# 获得额外信息--频数、频率、组值、组限、中值等
> str(a)
List of 7
$ breaks    : num [1:13] -3 -2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 ...
$ counts    : int [1:12] 0 1 1 1 3 4 6 2 2 0 ...
$ intensities: num [1:12] 0.0 0.1 0.1 0.1 0.3 ...
$ density   : num [1:12] 0.0 0.1 0.1 0.1 0.3 ...
$ mids      : num [1:12] -2.75 -2.25 -1.75 -1.25 -0.75 -0.25 0.25 0.75 1.25 1.75 ..
$ xname     : chr "x"
```

```
$ equidist   : logi TRUE
- attr(*, "class")= chr "histogram"
```

10.4.7 盒形图

```
> boxplot(x)
```

10.4.8 折线图

```
# 添加折线图
> a=hist(x,breaks=seq(-3,3,by=0.5))
> lines( c(min(a$breaks),a$mids,max(a$breaks)),c(0,a$counts,0),type='l' )
```

10.4.9 区间分割–cut函数

把每个数据归属于某一类, 或某一区间

```
> sals = c(12, .4, 5, 2, 50, 8, 3, 1, 4, .25) # enter data
> cats = cut(sals,breaks=c(0,1,5,max(sals))) # specify the breaks
> cats
[1] (5,50] (0,1]  (1,5]  (1,5]  (5,50] (5,50] (1,5]  (0,1]  (1,5]  (0,1]
Levels: (0,1] (1,5] (5,50]
```

改变水平标签

```
> levels(cats) = c("a","b","c") #
> cats
[1] c a b b c c b a b a
Levels: a b c
```

```
> cats[1]
[1] c
Levels: a b c
> table(cats)
cats
a b c
3 4 3
```

绘图

```
> barplot(table(cats))
# 错误, must be numeric
> hist(cats)
```

10.5 几个例子

10.5.1 类型数据 vs. 类型数据

一个抽烟–学习时间的例子, Problem：验证一项假定，抽烟的学生学习的时间少，抽样了10个人。

```
> x$smokes=c("Y","N","N","Y","N","Y","Y","Y","N","Y") # 抽烟与否
> x$study =  c(1,2,2,3,3,1,2,1,3,2) # 每天学习时间
> table(x)
   study
smokes 1 2 3
   N 0 2 2
   Y 3 2 1
> tmp = table(x)
> tmp
   study
smokes 1 2 3
```

```
   N 0 2 2
   Y 3 2 1
> str(tmp)
int [1:2, 1:3] 0 3 2 2 2 1
- attr(*, "dimnames")=List of 2
..$ smokes: chr [1:2] "N" "Y"
..$ study : chr [1:3] "1" "2" "3"
- attr(*, "class")= chr "table"
> old.digits = options("digits") # 保存默认打印字符长度 7
> options(digits=3)

prop.table 相当于
> tmp[1,1:3]/sum(tmp[1,1:3])
> tmp[2,1:3]/sum(tmp[2,1:3])
> prop.table(tmp,1) # 1 为按行，2 为列
   study
smokes    1    2    3
   N 0.000 0.500 0.500
   Y 0.500 0.333 0.167
> options(digits=7) # 还原打印字符位数

# 下面绘制条形图
> smokes=factor(smokes)
> smokes
[1] Y N N Y N Y Y Y N Y
Levels: N Y
> barplot(table(smokes,amount),
+ beside=TRUE,                # put beside not stacked
+ legend.text=T)   # add legend

# 只加图例
> barplot(table(amount,smokes),legend.text=T)
> barplot(table(smokes,amount), legend.text=T)

# 更改图例文字
> barplot(table(amount,smokes),main="table(amount,smokes)",
+ beside=TRUE,
+ legend.text=c("less than 5","5-10","more than 10"))
```

10.5.2 类型数据 vs. 连续数据

有实验组为x, 对照组为y, 画出盒型图来对照是一个不错的开始.(2 种方法)

```
> x = c(5, 5, 5, 13, 7, 11, 11, 9, 8, 9)
> y = c(11, 8, 4, 5, 9, 5, 10, 5, 4, 10)
> boxplot(x,y)

# 或者也可以这样
> num = scan()
1:  5 5 5 13 7 11 11 9 8 9 11 8 4 5 9 5 10 5 4 10
21:
Read 20 items
> cat = scan()
1: 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
21:
Read 20 items
> boxplot(num~cat)
```

10.5.3 连续数据 vs. 连续数据

常用且比较简单. 最常用的是散点图(plot(x,y)).

Chapter 11

相关与协方差

参考 [15] 3,4 多元数据的数据特征与相关分析

记 $x = x_1, x_2, \cdots, x_n.\quad y = y_1, y_2, \cdots, y_n.$

11.1 协方差

$$s_{xy} = \frac{1}{n-1}\sum (x_i - \bar{x})(y_i - \bar{y})$$

11.2 协方差矩阵

$$S = \begin{bmatrix} s_{xx} & s_{xy} \\ s_{xy} & s_{yy} \end{bmatrix}$$

11.3 相关系数

相关系数实际上是中心化与标准化后的协方差

$$r = \frac{s_{xy}}{\sqrt{s_{xx}}\sqrt{s_{yy}}}$$

```
ore<-data.frame(
    x=c(67, 54, 72, 64, 39, 22, 58, 43, 46, 34),
    y=c(24, 15, 23, 19, 16, 11, 20, 16, 17, 13)
)

# 相关矩阵
> cor(ore)
        x       y
x 1.0000000 0.9202595
y 0.9202595 1.0000000

# 协方差矩阵
> cov(ore)
       x      y
x 252.7667 60.60000
y  60.6000 17.15556
```

11.4 相关系数的区间估计

可以证明, 当样本充分大, 样本相关总体也相关. 但是样本比较少时, 无法得到可靠的结论. 问题是, 样本个数 n 取多少才能保证总体也相关?

Ruben 给出了总体相关系数的区间估计的近似逼近公式. 设

n 为样本个数, r 为样本相关系数, $u = z_{\alpha/2}$, 则计算

$$\begin{aligned} r^* &= \frac{r}{\sqrt{1-r^2}} \\ a &= 2n-3-u^2 \\ b &= r^*\sqrt{(2n-3)(2n-5)} \\ c &= (2n-5-u^2)r^{*2} - 2u^2 \end{aligned}$$

求方程 $ay^2 - 2by + c = 0$ 的根

$$y_1 = \frac{b - \sqrt{b^2-ac}}{a}$$

$$y_2 = \frac{b + \sqrt{b^2-ac}}{a}$$

则 $1-\alpha$ 的双侧置信区间为

$$L = \frac{y_1}{1+y_1^2}$$

$$U = \frac{y_2}{1+y_2^2}$$

下面是一个例子. $n = 6$ 时即使 $r = 0.8$ 也不可靠. $n = 25$ 则总体可以是相关的.

```
ruben.test <- function(n, r, alpha=0.05){
  u <- qnorm(1-alpha/2)
  r_star <- r/sqrt(1-r^2)
  a <- 2*n-3-u^2
  b <- r_star*sqrt((2*n-3)*(2*n-5))
  c <- (2*n-5-u^2)*r_star^2-2*u^2
  y1 <- (b-sqrt(b^2-a*c))/a
  y2 <- (b+sqrt(b^2-a*c))/a
  data.frame(n = n, r = r, conf = 1-alpha,
    L = y1/sqrt(1+y1^2), U = y2/sqrt(1+y2^2))
}
```

```
# n=6, r=0.8
> ruben.test(n=6,r=0.8)
  n   r conf          L         U
1 6 0.8 0.95 -0.09503772 0.9727884

# n=25, r=0.7
> ruben.test(n=25,r=0.7)
   n   r conf         L         U
1 25 0.7 0.95 0.4108176 0.8535657
```

相关系数置信区间的方法还有 David(1954) 提出的图表法, Kendall 与 Stuart (1961) 提出的 Fisher 逼近法等.

最有效的方法是做总体的相关性检验. 可以证明, 当 $(X,Y)^T$ 为二元正态总体, 且 $\rho(X,Y)=0$ 时

$$t=\frac{r_{xy}\sqrt{n-2}}{\sqrt{1-r_{xy}^2}}$$

服从自由度为 $n-2$ 的 t 分布. 由于相关系数 r_{xy} 称为 Pearson 相关系数, 故此检验称为 Pearson 相关检验.

其它还有 Spearman 秩检验和 Kendall 秩检验.

R 函数 cor.test() 可以进行 Pearson, Spearman 秩检验和 Kendall 秩检验三种方法.

```
> attach(ore)
> cor.test(x,y,method='pearson')

        Pearson's product-moment correlation

data:  x and y
t = 6.6518, df = 8, p-value = 0.0001605
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6910290 0.9813009
sample estimates:
```

```
     cor
0.9202595
```

11.5 各种相关的检验

非参数检验很多时候都在讨论各种相关性的度量与检验. 包括二项比例, 列联表, 秩, 多个样本等等的相关性分析.

另外参考回归部分 chapter 23 section 23.6, 讨论回归系数的相关性

R默认的已经有很多函数做相关性的度量及检验.

coin 包含了很多的相关性检验的函数, 可以参考, help(pac="coin") .

Chapter 12

估计

参考文献 [6] 第七章

参考文献 [15] 第四章

参数估计有点估计和区间估计两方面的问题. 点估计有矩法, 极大似然法, 贝叶斯估计, 最小二乘估计等.

非参数估计问题例如随机变量$\xi\eta$之间有一定的相关性, 试问在什么准则下, 由一个对另外一个的预测为最佳.

12.1 矩法

英国统计学家 K. Pearson 引入的矩法是较早的参数点估计的方法.

利用矩法估计均值和方差, 等价于用样本的一阶原点矩估计均值, 二阶中心矩估计方差.

12.1.1 一般描述

设总体X的分布函数$F(x;\theta_1,\cdots,\theta_m$中有m个未知参数. 假设总

体的m阶原点矩存在. n个样本$x_1, \cdots, x_n$令总体的k阶原点矩等于样本的k阶原点矩, 即

$$E(X) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$E(X^2) = \frac{1}{n}\sum_{i=1}^{n} x_i^2$$

$$\cdots$$

$$E(X^m) = \frac{1}{n}\sum_{i=1}^{n} x_i^m$$

解此方程组得到$\hat{\theta}_1, \cdots, \hat{\theta}_m$, 并使用$\hat{\theta}_k$作为参数$\theta_k$的估计, 则称$\hat{\theta}_k$为参数$\theta_k$的矩法估计量.

12.1.2 估计均值与方差

更一般的提法为: 利用样本的数字特征作为总体的数字特征的估计. 例如, 无论总体服从什么分布, 其均值和方差分别为$E(X) = \mu, E[(X - E(X))^2] = \sigma^2$. 使用矩法估计均值和方差. 列出方程组

$$\begin{aligned} E(X) &= \mu = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x} \\ E(X^2) &= Var(X) + [E(X)]^2 = \sigma^2 + \mu^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 \end{aligned}$$

解得均值与方差的矩法点估计

$$\begin{aligned} \hat{\mu} &= \bar{x} \\ \hat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \end{aligned}$$

注意, 方差的矩估计不等于样本方差S^2, 而是

$$\hat{sigma}^2 = \frac{n-1}{n}S^2$$

12.1.3 讨论

矩法是最古老的点估计方法. 不要求知道分布函数. 但是要求随机变量的原点矩存在. 否则就不能估计了.

由于矩与分布函数无关, 那么矩法还没有充分利用分布函数对参数提供的信息.

12.1.4 例1: 贝努里分布

求贝努里分布(两点分布, 硬币实验)参数p的矩法估计量.

设随机变量X服从贝努里分布, 成功$X = 1$, 失败$X = 0$. $E(X) = p$. 设试验n次, 成功m次. 则p的矩法估计为

$$\hat{p} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{m}{n}$$

即使用成功次数出现的频率作为概率p的估计.

12.1.5 例2: 均匀分布

设随机变量X服从$[0, \theta]$的均匀分布, 现有n个样本$x_1, \cdots, x_n$. 试估计参数θ.

均匀分布的一阶矩(均值)为$\theta/2$, 故其估计为

$$E(X) = \frac{\theta}{2} = \bar{x} \Longrightarrow \hat{\theta} = 2\bar{x}$$

12.1.6 例3: 均匀分布

设随机变量X服从$[\theta_1, \theta_2]$的均匀分布, 现有n个样本$x_1, \cdots, x_n$. 试估计参数θ_1, θ_2.

我们使用一阶原点矩估计均值, 二阶耶酥教估计方差, 即

$$\begin{aligned} E(X) &= \frac{theta_1 + \theta_2}{2} = \bar{x} \\ Var(X) &= \frac{(theta_2 - \theta_1)^2}{12} = \frac{1}{n}\sum_{i=1}^{n}(x_1 - \bar{x})^2 = S^2 \end{aligned}$$

解方程组得

$$\begin{aligned} \hat{\theta}_1 &= \bar{x} - \sqrt{3}S \\ \hat{\theta}_2 &= \bar{x} + \sqrt{3}S \end{aligned}$$

我们使用rootSolve 包的函数 multiroot() 解此方程组

```
x=c(4, 5, 2, 9, 5, 1, 6, 4, 6, 2)
m1=mean(x) # 均值
m2=sum((x-mean(x))^2)/10 # 方差
# x=[theta_1, theta_2]
model <- function(x,m1,m2){
   c(F1= x[1]+x[2]-2*m1,
      F2= (x[2] - x[1])^2/12 - m2)}
# 求解
> multiroot(f=model,start=c(0,10),m1=m1,m2=m2)
$root #  theta_1  theta_2
[1] 0.5115551 8.2884449

$f.root
         F1          F2
-1.713101e-10  1.205959e-06

$iter
```

```
[1] 4

$estim.precis
[1] 6.030653e-07

# 按照公式计算的 theta_1 theta_2
> m1-sqrt(3*m2)
[1] 0.5115556
> m1+sqrt(3*m2)
[1] 8.288444
```

12.1.7 例4: 二项分布

设总体服从二项分布$B(N,p)$, N,p为未知参数. 均值(一阶原点矩)为$M1 = N*p$,方差(二阶中心矩)为$M2 = N*p*(1-p)$. 建立方程组

$$\begin{aligned} F1 &= Np - M1 = 0 \\ F2 &= Np(1-p) - M2 = 0 \end{aligned}$$

解析结果为

$$N = \frac{M1^2}{M1 - M2}, \quad p = \frac{M1 - M2}{M1}$$

```
# N=20,p=0.7, 试验次数n=100
x<-rbinom(100, 20, 0.7);
m1=mean(x)
m2=sum((x-mean(x))^2)/100
> m1
[1] 13.84
> m2
[1] 4.8544

# 先给出解析计算的结果
> N=m1^2/(m1-m2); N
[1] 21.31695
```

```
> p=(m1-m2)/m1; p
[1] 0.6492486

# 下面使用 multiroot() 函数计算
# x=[N,p]
model <- function(x,m1,m2){
   c(F1= x[1]*x[2]-m1,
      F2= x[1]*x[2]*(1-x[2])- m2)}
multiroot(f=model,start=c(20,1),m1=m1,m2=m2)
# 下面是结果
$root
[1] 21.3169515  0.6492486

$f.root
         F1          F2
 1.205192e-08 -3.955911e-08

$iter
[1] 5

$estim.precis
[1] 2.580551e-08
```

12.2 极大似然法(MLE)

极大似然估计(Maximum likelyhood estimation, MLE)是Fisher1912年提出的应用非常广泛的参数估计方法, 其思想始于Gauss的误差理论. 它充分利用了分布函数的信息, 克服了矩法的某些不足.

12.2.1 极大似然原理

下面是一个摸球的例子. (参考文献[6] 7.1). 一个布袋里面有黑球和白球. 我们要估计它们的比例是1/4还是3/4. 现在有放回的抽取了3个球, 其中黑球的个数记为x. 我们就要通过黑

球的数目来判断$p=1/4$还是$p=3/4$. 下面是$p=1/4$和$p=3/4$出现黑球个数的概率从表中确定, 当$x=0,1$时, $p=1/4$, 当$x=2,3$时,

Table 12.1: 不同参数下黑球出现个数的概率

x	0	1	2	3
P(x;3/4)	1/64	9/64	27/64	27/64
P(x;1/4)	27/64	27/64	9/64	1/64

$p=3/4$.

一般的说, 我们把参数θ看作未知参数. 观察值是随机变量的一次实现. 不同的θ对应于观察值出现的概率不同. 既然出现了观察值, 我们认为如果某个参数应该是使得此观察值出现的概率比其它参数时观察值出现的概率要大, 那么这个参数应该就是此观察值对应的参数. 这就是极大似然原理.

12.2.2 似然函数

记概率密度函数(离散时为分布律)为$f(x;\theta)$, 观察值$x=x_1,\cdots,x_n$. 称下面的函数

$$L(\theta;x)=L(\theta_1,\cdots,\theta_l;x_1,\cdots,x_n)=\prod_{i=1}^{n}f(x_i;\theta)$$

为参数θ的似然函数(likelyhood function).

显然, 样本固定时, $L(\theta;x)$是θ的函数, 若θ固定, 则$L(\theta;x)$就是样本的联合概率密度函数(离散的时候为联合分布律)

12.2.3 极大似然估计(MLE)

使得$L(\theta;x)$最大的一个(一组)θ值称为参数θ的极大似然估计(MLE), 即

$$L(\hat{\theta};x)=max(L(\theta;x)),$$

称$\hat{\theta}$为参数的极大似然估计量.

12.2.4 似然方程的求解

由极值的一阶必要条件, 似然函数$L(\theta;x)$对参数偏导得似然方程(likelyhood equation)

$$\frac{\partial L(\theta;x)}{\partial \theta_i}=0,\quad i=1,\cdots,l$$

连乘形式计算不方便, 取对数得等价形式, 对数似然方程(loglikelyhood equation)

$$\frac{\partial \ln L(\theta;x)}{\partial \theta_i}=0,\quad i=1,\cdots,l$$

严格讲, 极大似然估计一定是似然方程或对数似然方程的解, 但是似然方程或对数似然方程对参数的二阶Hesse矩阵负定, 则似然方程或对数似然方程的解才是极大似然估计.

12.2.5 例1: 正态分布

设X服从正态分布$N(\mu,\sigma^2)$. $x=x_1,\cdots,x_n$为来自总体的一组样本. 试用极大似然法估计参数μ,σ^2.

似然函数为

$$L(\mu,\sigma^2;x)=\prod_{i=1}^{n}f(x_i;\mu,\sigma^2)=(2\pi\sigma^2)^{-n/2}exp[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2]$$

对数似然函数为

$$\ln L(\mu,\sigma^2;x)=-\frac{n}{2}\ln(2\pi\sigma^2)-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

求偏导得到对数似然方程

$$\frac{\partial \ln L(\mu,\sigma^2;x)}{\partial \mu} = -\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2 = 0$$

$$\frac{\partial \ln L(\mu,\sigma^2;x)}{\partial \sigma^2} = -\frac{n}{2\sigma^2}+\frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i-\mu)^2 = 0$$

解此似然方程组得到[1]

$$\mu = \frac{1}{n}\sum_{i=1}^{n}x_i = \bar{x}, \quad \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$$

验证对数似然函数的二阶Hesse矩阵为负定, 故此估计就是似然方程的极大值点, 与矩法的一阶二阶矩估计是一致的.

```
x=rnorm(10)

# multiroot()函数计算
# e[1]=\mu, e[2]=\sigma, x=样本
model <- function(e,x){
   n=length(x)
   c(F1= sum(x-e[1]),
      F2= -n/e[2] + sum((x-e[1])^2)/e[2]^3)}
> multiroot(f=model,start=c(0,1),x=x)
$root
[1] 0.1273094 1.1256564

$f.root
        F1          F2
```

[1]第二个方程也可以为

$$\frac{\partial \ln L(\mu,\sigma^2;x)}{\partial \sigma} = -\frac{n}{\sigma}+\frac{1}{\sigma^3}\sum_{i=1}^{n}(x_i-\mu)^2 = 0$$

```
5.551115e-17 1.394105e-08

$iter
[1] 5

$estim.precis
[1] 6.970523e-09
# 公式计算
> mean(x)
[1] 0.1273094
> sum((x-mean(x))^2)/10
[1] 1.267102
```

12.2.6 例2: 指数分布

设总体X服从指数分布, 密度函数为

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x >= 0 \\ 0, & x < 0 \end{cases}$$

$x = x_1, \cdots, x_n$为来自总体的一组样本. 估计参数λ.

$$\ln L(\lambda; x) = n \ln \lambda - \lambda \sum_{i=1}^{n} x_i$$

取导数

$$\frac{\partial \ln L(\lambda; x)}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i$$

解得

$$\lambda = \frac{n}{\sum_{i=1}^{n} x_i}$$

二阶导数(对应Hesse矩阵)$-\frac{n}{\lambda^2} < 0$, 故此估计为极大点.

12.2.7 例3: 均匀分布

当参数空间(可能的值)为开区域, 此时似然方程组解的方法不适用.

设总体X服从区间$[a,b]$的均匀分布. $x=x_1,\cdots,x_n$为来自总体的一组样本. 估计参数a,b.

似然函数为

$$L(a,b;x)=\begin{cases}\frac{1}{(b-a)^n}, & a<=x_i<=b, \quad i=1,\cdots,n\\ 0, & others\end{cases}$$

显然, $L(a,b;x)$不是a,b的连续函数, 其似然方程为

$$\begin{aligned}\frac{\partial \ln L(a,b;x)}{\partial a} &= \frac{n}{b-a}=0\\ \frac{\partial \ln L(a,b;x)}{\partial b} &= \frac{-n}{b-a}=0\end{aligned}$$

因此不能求解.

应该从极大似然估计的定义出发来求$L(a,b;x)$的最大值. 要$L(a,b;x)$达到最大, 那么$b-a$应该尽可能的小, 但是a不能大于$min(x)$, b不能小于$max(x)$. 因此a,b的极大似然估计为

$$\hat{a}=min(x), \quad \hat{b}=max(x)$$

12.2.8 例4: 钓鱼问题

在鱼塘钓出r条鱼, 做上记号, 然后再钓出s条, 发现有x条有标记. 试估计鱼塘所有的鱼有多少?

第二次钓出的鱼的条数X服从超几何分布

$$P(X=x)=\frac{C_r^x C_{N-r}^{s-x}}{C_N^s}$$

似然函数为

$$L(N;x) = P(X = x)$$

直接对似然函数求导相当困难, 那么考虑似然函数的比值

$$g(x;N) = \frac{L(N;x)}{L(N-1;x)} = \frac{(N-s)(N-r)}{N(N-r-s+x)} = \frac{N^2-(r+s)N+rs}{N^2-(r+s)N+xN}$$

当$rs > xN$时 有$g(x;N) > 1$, $rs < xN$时 有$g(x;N) < 1$, 即 似 然 函数$L(N;x)$在$N = \frac{rs}{x}$附近达到最大. 即N的极大似然估计为

$$\hat{N} = [\frac{rs}{x}], \quad []\text{表示取整数}$$

12.2.9 例5: Cauchy分布(数值方法)

设总体X服从Cauchy分布, 密度函数为

$$f(x;\theta) = \frac{1}{\pi[1+(x-\theta)^2]}, \quad -\infty < x < \infty$$

$x = x_1, \cdots, x_n$为来自总体的一组样本. 估计参数θ.

Cauchy分布的似然函数为

$$L(\theta;x) = \prod_{i=1}^{n} f(x_i;\theta) = \frac{1}{\pi^n}\prod_{i=1}^{n}\frac{1}{\pi[1+(x_i-\theta)^2]}$$

求导得到对数似然方程为

$$\sum_{i=1}^{n}\frac{x_i-\theta}{1+(x_i-\theta)^2} = 0$$

求对数似然方程的解析解是困难的, 考虑使用数值方法.

使用 uniroot() 函数

```
# 参数为1的cauchy分布
x=rcauchy(100,1)
f<-function(p) sum((x-p)/(1+(x-p)^2))
out<-uniroot(f,c(0,5))
> out
$root
[1] 0.7481134

$f.root
[1] 0.0001692195

$iter
[1] 5

$estim.prec
[1] 6.103516e-05
```

使用 optmize() 函数

```
loglike<-function(p)sum(log(1+(x-p)^2))
> optimize(loglike,c(0,5))
$minimum
[1] 0.7481312

$objective
[1] 129.1854
```

12.3 均值估计

12.3.1 点估计

总体均值μ的最小方差无偏估计为样本的均值 $\bar{x}$.

12.3.2 均值的标准误

均值标准误差的估计量是 $s/\sqrt{n}$ –样本均值集合的标准差. 实际上, 总体方差常常未知. 后面会看到, σ^2 的合理估计是 s^2.

12.3.3 均值的区间估计–总体方差已知

总体方差已知时, 均值为正态分布.

我们常常希望得到均值的严格似乎合理的区间估计. 下面的区间估计仅当未知分布是正态分布才是正确的. 若不是正态分布, 则只能近似成立.

若$\bar{x} \sim N(\mu, \sigma^2/n)$, 那么把 $\bar{x}$ 写为标准形式, 即

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

则 z 应该是标准正态分布. 当重复抽样时, 95% 的 z 值落入 -1.96 到 1.96 之间. 但是 σ 在实际中很少知道.

```
# x 为均值 5, 方差 1 的总体中抽取的 10 个样本
> x=rnorm(10,5)
> x
 [1] 4.927264 4.067237 6.136822 5.722123 6.286754 3.266601 4.443779 3.630787
 [9] 4.874269 3.748306
# z 值为 qnorm(0.025)=-1.959964, qnorm(0.975)=1.959964
> mean(x)+qnorm(0.025)*1/sqrt(10)
[1] 4.090599
> mean(x)+qnorm(0.975)*1/sqrt(10)
[1] 5.330189
```

12.3.4 均值的区间估计–总体方差未知

总体方差未知时, 均值为t分布.

当 σ 未知时, 合理的估计是用样本的标准差 s 估计 σ 而用代替后计算的 z 来构建置信区间. 问题是, 此时的 z 已经不是正态分布了. 此时的 z 的分布是 t 分布.

正态分布中均数的置信区间具未知方差的正态分布的均值 μ 的 $100\% * (1-\alpha)$ 置信区间 (confidence interval, CI) 可以写成

$$(\bar{x} - t_{n-1,1-\alpha/2}s/\sqrt{n}, \bar{x} + t_{n-1,1-\alpha/2}s/\sqrt{n})$$

样本均值为3, 标准差为5, 样本量为20的均值的95%的置信区间为

```
> 3+qt(p=0.025,df=20)*5/sqrt(20)
[1] 0.667822
> 3+qt(p=0.975,df=20)*5/sqrt(20)
[1] 5.332178
```

12.4 方差估计

12.4.1 点估计

按照公式即可

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}^2)$$

```
> x
 [1] -5 -4 -3 -2 -1  0  1  2  3  4  5
> var(x)
[1] 11
> sum((x-mean(x))^2)/(length(x)-1)
[1] 11
```

12.4.2 区间估计

σ^2 的 $100\% * (1-\alpha)$ 置信区间为

$$[\frac{(n-1)s^2}{\chi^2_{n-1,\alpha/2}}, \frac{(n-1)s^2}{\chi^2_{n-1,1-\alpha/2}}]$$

```
> x
 [1] -5 -4 -3 -2 -1  0  1  2  3  4  5
> (10-1)*var(x)/qchisq(0.025,10-1)
[1] 36.66138
> (10-1)*var(x)/qchisq(0.975,10-1)
[1] 5.20429
```

12.5 二项分布的估计

12.5.1 参数p及标准误差的点估计

记 x 是二项随机变量, 其参数为 n 及 p, p 的无偏估计为事件中的样本比例 $\hat{p}$, 标准误差 $\sqrt{pq/n}$ 的精确估计为 $\sqrt{\hat{p}\hat{q}/n}$.

```
> x=rbinom(10,1,0.5)
> x
 [1] 1 1 0 1 1 1 0 0 1 0
> t=table(x)
> t
x
0 1
4 6
> t['1']/length(x) # 此即为p的点估计, 还可以使用binom.test(table(x))得到.
  1
0.6
> sqrt(t['1']*t['0']/length(x)) # 此为标准误差的点估计
```

```
      1
1.549193
```

12.5.2 p的区间估计

```
> binom.test(table(x))

       Exact binomial test

data:  table(x)
number of successes = 4, number of trials = 10, p-value = 0.7539
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.1215523 0.7376219
sample estimates:
probability of success
                   0.4

> b=binom.test(table(x))
> str(b)
List of 9
 $ statistic  : Named int 4
  ..- attr(*, "names")= chr "number of successes"
 $ parameter  : Named int 10
  ..- attr(*, "names")= chr "number of trials"
 $ p.value    : Named num 0.754
  ..- attr(*, "names")= chr "0"
 $ conf.int   : atomic [1:2] 0.122 0.738
  ..- attr(*, "conf.level")= num 0.95
 $ estimate   : Named num 0.4
  ..- attr(*, "names")= chr "probability of success"
 $ null.value : Named num 0.5
  ..- attr(*, "names")= chr "probability of success"
 $ alternative: chr "two.sided"
 $ method     : chr "Exact binomial test"
 $ data.name  : chr "table(x)"
 - attr(*, "class")= chr "htest"
```

Chapter 13

假设检验

13.1 各种情况使用的方法

Aim	Parametric tests	Non-parametric tests
compare two means	Student's T test	Wilcoxon's U test
compare more than two means	Anova (analysis of variance)	Kruskal--Wallis test
Compare two variances	Fisher's F test	Ansari-Bradley or Mood test
Comparing more than	Bartlett test	Fligner test

13.2 如何检验一个分布为指定分布

参考第 18 章

13.3 单样本假设检验

13.3.1 方差未知的正态分布均值的单样本检验

前提条件–数据为正态分布, 使用t.test(). 若数据非正态分布, 应该使用 Wilcoxon's U test（见非参数检验）.

```
> x=rnorm(200)
> t.test(x)

       One Sample t-test

data:  x
t = -1.1695, df = 199, p-value = 0.2436
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.21865305  0.05585082
sample estimates:
  mean of x
-0.08140112
```

可以看一下p值的分布, 若零假设成立, p值在[0,1]之间为均匀分布

```
> p <- c()
> for (i in 1:1000) {
+   x <- rnorm(200)
+   p <- append(p, t.test(x)$p.value)
+ }
> hist(p, col='light blue')
```

13.3.2 数据非正态时的情况

数据非正态时需要做转换使其变为正态分布, 或使用非参数检验.

数据为均匀分布时, 会出现下面的情况.

```
> N <- 1000
> n <- 3
> v <- vector()
> for (i in 1:N) {
+   x <- runif(n, min=-1, max=1)
+   r <- t.test(x)$conf.int
+   v <- append(v, r[1]<0 & r[2]>0)
+ }
> sum(v)/N
[1] 0.919
```

数据正态分布时,

```
> N <- 1000
> n <- 100
> v <- vector()
> for (i in 1:N) {
+   x <- rnorm(n, sd=1/sqrt(3))
+   r <- t.test(x)$conf.int
+   v <- append(v, r[1]<0 & r[2]>0)
+ }
> sum(v)/N
[1] 0.947
```

可以看到, 将均匀分布作为正态分布时其置信区间的概率不是0.95而是0.92. 这增大了2型错误的概率.

但是样本量很大时, 误差就不明显了

```
> N <- 1000
> n <- 100
> v <- vector()
> for (i in 1:N) {
+   x <- runif(n, min=-1, max=1)
+   v <- append(v, t.test(x)$p.value)
+ }
> sum(v>.05)/N
[1] 0.957
```

13.3.3 方差已知的正态分布均值的单样本检验

此时使用z检验.

某些研究中, 根据过去的资料翻查可能方差是知道的. 在这种情况下, 检验统计量 t 可以由 z 代替, 临界值也由相应的标准正态分布的临界值代替. 其中

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

其它的计算完全类似于方差未知时的 t 检验, 不论是单侧还是双侧.

下面例子假设总体方差为1. 检验其零假设为0. 样本量为100

```
> x=rnorm(100)
> z=(mean(x)-0)/(1/sqrt(100))
> z
[1] 2.005832
> pnorm(z)
[1] 0.9775629
```

13.3.4 功效与样本量

参考 power.t.test()

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
             power = NULL,
             type = c("two.sample", "one.sample", "paired"),
             alternative = c("two.sided", "one.sided"),
             strict = FALSE)
```

```
> power.t.test(n = 20, delta = 1) #已知样本量, 求功效

    Two-sample t test power calculation

             n = 20
         delta = 1
            sd = 1
     sig.level = 0.05
         power = 0.8689528
   alternative = two.sided

 NOTE: n is number in *each* group
```

```
> power.t.test(power=0.8, delta = 1)#已知功效, 求样本量

    Two-sample t test power calculation

             n = 16.71477
         delta = 1
            sd = 1
     sig.level = 0.05
         power = 0.8
   alternative = two.sided

 NOTE: n is number in *each* group
```

13.3.5 方差的区间估计及检验–卡方检验

R 中没有chisq.var.test()

在方差的置信区间估计及检验中, 正态条件特别重要. 若样本不满足正态性, 则临界值p-值及置信区间都不是有效的.

欲检验

$$H_0 : \sigma^2 = \sigma_0^2 \quad vs. \quad H_1 : \sigma^2 \neq \sigma_0^2$$

计算检验统计量

$$X^2 = (n-1)s^2/\sigma_0^2 \sim \chi^2_{n-1}$$

如果 $X^2 < \chi^2_{n-1,\alpha/2}$ 或 $X^2 > \chi^2_{n-1,1-\alpha/2}$, 则拒绝 H_0 如果 $\chi^2_{n-1,\alpha/2} \leq X^2 \leq \chi^2_{n-1,1-\alpha/2}$, 则接受 H_0

p-值(双侧备择)

同上计算检验统计量 X^2 如果 $s^2 \leq \sigma_0^2$, 则 p-值=2*(χ^2_{n-1} 分布曲线下从左到 X^2 的面积)

如果 $s^2 > \sigma_0^2$, 则 p-值=2*(χ^2_{n-1} 分布曲线下从右到 X^2 的面积)

下面是一个例子. 由于$var(x) <= 1$, 则$p = 2 * pchisq(q = chi2, df = 99)$. 若$var(x) > 1$, 则$p = 1 - 2 * pchisq(q = chi2, df = 99)$. 单侧检验不用2倍

```
> x=rnorm(100) # 检验x的总体的方差是否为1
> var(x)
[1] 0.9344586

> chi2=(100-1)*var(x)/1 #计算检验统计量
> chi2
[1] 92.5114

> qchisq(df=99,p=0.025) # 区间下侧
```

```
[1] 73.36108
> qchisq(df=99,p=0.975) # 区间上侧
[1] 128.422

> p=2*pchisq(q=chi2,df=99)  # p值
> p
[1] 0.671611
```

13.4 方差齐性检验-F检验

两个样本的均值t检验之前, 需要判断其方差是否相同. 正态样本使用此F检验

13.4.1 F分布的特点

具自由度 d_1, d_2 的F分布的下侧第p个百分位点, 就是具有自由度为 d_12, d_1 的F分布的上侧第p个百分位点的倒数, 即

$$F_{d_1,d_2,p} = 1/F_{d_2,d_1,1-p}$$

13.4.2 F检验

使用t检验之前需要此检验.

参考var.test()

```
> x <- rnorm(50, mean = 0, sd = 2)
> y <- rnorm(30, mean = 1, sd = 1)

> var.test(x, y)  # 第一种用法

        F test to compare two variances
```

```
data:  x and y
F = 6.1786, num df = 49, denom df = 29, p-value = 1.516e-06
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 3.104259 11.624472
sample estimates:
ratio of variances
        6.178575

> var.test(lm(x ~ 1), lm(y ~ 1))  # 第二种用法. The same.

       F test to compare two variances

data:  lm(x ~ 1) and lm(y ~ 1)
F = 6.1786, num df = 49, denom df = 29, p-value = 1.516e-06
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 3.104259 11.624472
sample estimates:
ratio of variances
        6.178575
```

手工计算

```
> f=var(y)/var(x)
> f
[1] 0.1618496
> qf(0.025, 49,29)
[1] 0.5315144
> qf(0.975, 49,29)
[1] 1.990354
> f<qf(0.025,49,29)
[1] TRUE
```

13.4.3 多于2个正态样本的方差检验

参考 bartlett.test

13.4.4 2个非正态样本的方差检验

参考 ansari.test 或 mood.test , 它们是非参数检验

13.4.5 多于2个非正态样本

参考 fligner.test

13.5 两样本均值的t检验

样本需正态分布, 非正态分布的数据需要转换为正态分布或使用非参数检验

对于两个样本方差不一样的情况, p值保持正确, 但是功效下降的很快. 若数据看起来是正态分布但是方差不同, 最好对它们归一化处理(x/var(x), y/var(y))然后使用t检验, 这样比使用非参数检验要好.

13.5.1 t检验

参考t.test()

用法为: 默认为非配对, 方差不相等(paired = FALSE, var.equal = FALSE). 若只有一个样本, mu代表其被检验的均值, 若两个样本(x, y)则mu代表其均值之差.

```
t.test(x, y = NULL,
```

```
          alternative = c("two.sided", "less", "greater"),
          mu = 0, paired = FALSE, var.equal = FALSE,
          conf.level = 0.95, ...)

## S3 method for class 'formula':
t.test(formula, data, subset, na.action, ...)
```

第一种用法, 数据在一个向量里, 由group指明不同的组

```
> d=sleep
> d
   extra group
1    0.7     1
2   -1.6     1
3   -0.2     1
4   -1.2     1
5   -0.1     1
6    3.4     1
7    3.7     1
8    0.8     1
9    0.0     1
10   2.0     1
11   1.9     2
12   0.8     2
13   1.1     2
14   0.1     2
15  -0.1     2
16   4.4     2
17   5.5     2
18   1.6     2
19   4.6     2
20   3.4     2

> t.test(extra ~ group, data = sleep)

        Welch Two Sample t-test

data:  extra by group
t = -1.8608, df = 17.776, p-value = 0.0794
```

```
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean in group 1 mean in group 2
          0.75           2.33
```

第二种用法, 普通的两个数据

```
> attach(d) #将d的数据extra, group纳入名称空间, 可以直接使用
> t.test(extra[group == 1], extra[group == 2])

        Welch Two Sample t-test

data:  extra[group == 1] and extra[group == 2]
t = -1.8608, df = 17.776, p-value = 0.0794
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean of x mean of y
    0.75      2.33
```

13.5.2 功效与样本量

参考 power.t.test type=paired

不配对的样本量估计参考流行病部分两个均值的样本量估计. epicalc 包的函数为 n.for.2means

功效见 epicalc 包的函数 power.for.2means

Part III

非参数统计

“非参数统计”参考文献除了[11]，主要框架及内容参考的是W.J.Conover 著, 崔恒建译《实用非参数统计(第三版)》。R部分主要参考了《simpleR》《Statistics with R》等。

Chapter 14

一些概念

14.1 次序统计量

次序统计量(order statistic): 把观测值$x_1, x_2, \cdots, x_n$按从小到大排列, 取值为第k个值$x^{(k)}$的随机变量称为秩为k的次序统计量(order statistic of rank k). 秩为1的次序统计量总是取最小值.

14.2 无偏检验

无偏检验(unbiasd test)是零假设不成立时拒绝零假设的概率大于等于零假设成立时拒绝零假设的概率.

14.3 相对效率

相对效率(relative efficient): 两个检验用来检验相同的零假设和备择假设, 其对应的α, β相等, 那么两个检验的样本容量之比定义为相对效率.

14.4 渐近相对效率(A.R.E)

渐近相对效率(asymptotic relative efficient, A. R. E): 令n1,n2为相同显著性水平, 相同功效的两个检验T1,T2的样本容量. 若α, β固定, 当n1趋于无穷时, 极限n2/n1存在, 且与α, β独立, 那么, n2/n1的极限称为第一个检验对第二个检验的渐近相对效率.

因为功效依赖于太多的因素, 为了寻找具有最大功效的检验, 通常要找出具有最大渐近相对效率的检验. 故A. R. E是很重要的. 通常两个检验的A. R. E计算比较困难, 其全面研究本身可以写一本书.

14.5 保守性

若真实的显著性水平比规定的低, 称为保守的.

14.6 结(tie)

如果秩次的差的绝对值相同, 称为结. 有结和无结的计算公式不同. why???

14.7 一致对与不一致对

在成对的匹配中, 结局相同的对称为一致对(concordant pair). 结局不同的称为不一致对(discordant pair). 此例中, 有510+90=600个一致对. 有5+16=21个不一致对. 一致对不提供信息, 故分析时抛弃之. 我们集中研究一致对.

不一致对中, 使用A处理后有事件发生而B处理后未发生, 称为A型不一致对. 否则称为B型不一致对.

14.8 二项比例齐性检验与列联表的独立性检验的关系

二项比例齐性检验(test for homogeneity of binomial proportion)检验不同的组的潜在的成功的比例(二项分布的参数 p)是否相同, 或等于某个给定的值. 零假设为$H_0 : p_1 = p_2 = p$对$H_1 : p_1 \neq p_2$. 显著性检验基于两个比例的差值$p_1 - p_2$, 若与零差别显著则拒绝零假设, 否则接受零假设. 可以使用正态逼近法和列联表法.

实际上列联表法是从不同的角度考察问题, 但是与正态逼近法的检验是相同的. 列联表的另外一个用处是检验列联表中两个变量的独立性. 例如两次问卷同一批人的饮食习惯, 在同一个人上做的两次调查是否有某种关联性. 这种检验也称为两个特征的独立性检验(test of independence, 也称为一致性检验, test of concordance)或关联性检验(test of association). 齐性检验与独立性检验的方法是相同的.

我们可以不加区分的使用这些方法, 只是最后对结果的解释不同罢了.

Chapter 15

基于二项分布的检验

15.1 二项分布参数的假设检验

15.1.1 p值与区间

参考binom.test(). 其中第一个参数可以为1个2值向量, 分别为成功和失败的次数; 也可以为2个值, 分别为成功和总试验次数. 结果给出了p值和区间.

例如, 现在的前列腺手术约有一半有副作用(p=0.5). FDA(food and drug administration, 食品与药物管理局)研究了一项新手术, 19例中只有3例有这种不良反应. 那么是否能够说新方法可以有效减轻副作用? 零假设为p=0.5, 备择假设为p¡0.5. 这是一个左单边检验. p-值为0.002213. 所以我们拒绝零假设. 结论是新方法可以有效减轻副作用.

```
> binom.test(c(3,16),p=0.5,alternative="less")

        Exact binomial test

data:  c(3, 16)
number of successes = 3, number of trials = 19, p-value = 0.002213
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
```

```
 0.0000000 0.3594256
sample estimates:
probability of success
           0.1578947
```

在简单孟德尔遗传中, 后代有1/4是矮的, 3/4是高的. 我们做了一个杂交试验验证. 得到243个矮的植株, 682个高的. 零假设为p=1/4, 备择假设为$p \neq 1/4$. p-值为0.3825, 不能拒绝零假设. 所以接受后代矮的概率为1/4.

```
  # 另外一个用法也可以 binom.test(682, 682 + 243, p = 3/4)
> binom.test(c(682, 243), p = 3/4)

        Exact binomial test

data:  c(682, 243)
number of successes = 682, number of trials = 925, p-value = 0.3825
alternative hypothesis: true probability of success is not equal to 0.75
95 percent confidence interval:
 0.7076683 0.7654066
sample estimates:
probability of success
           0.7372973
```

某省随机选20个高中, 其中7个达到优秀. 那么该省所有高中符合优秀的比例p的95%置信区间是什么? 因为此处不要求估计二项比例p, 那么p可以任意选择. p的95%区间为$[0.1539092, 0.5921885]$

```
> binom.test(c(7,13),p=7/20,conf.level = 0.95)
> binom.test(c(7,13),conf.level = 0.95) # 二项比例p默认为0.5

        Exact binomial test

data:  c(7, 13)
number of successes = 7, number of trials = 20, p-value = 0.2632
```

```
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.1539092 0.5921885
sample estimates:
probability of success
                  0.35
```

15.1.2 功效与样本量

参考 power.prop.test() 好像实用非参数统计(第三版) 3.3 节称作容忍限.

下面是样本量为50, 零假设p=0.5, 备择假设p=0.75, 置信水平=0.95, 双边检验的功效为 0.74

```
> power.prop.test(n = 50, p1 = .50, p2 = .75)

     Two-sample comparison of proportions power calculation

              n = 50
             p1 = 0.5
             p2 = 0.75
      sig.level = 0.05
          power = 0.7401659
    alternative = two.sided

 NOTE: n is number in *each* group
```

15.2 二项比例齐性检验: prop.test

二项比例齐性检验(test for homogeneity of binomial proportion)检验不同的组的潜在的成功的比例(二项分布的参数 p)是否相同, 或等于某个给定的值. 零假设为$H_0 : p_1 = p_2 = p$对$H_1 : p_1 \neq p_2$. 显

著性检验基于两个比例的差值$p_1 - p_2$, 若与零差别显著则拒绝零假设, 否则接受零假设. 可以使用正态逼近法和列联表法. 实际上列联表法是从不同的角度考察问题, 但是与正态逼近法的检验是相同的. 列联表的另外一个用处是检验列联表中两个变量的独立性. 例如两次问卷同一批人的饮食习惯, 在同一个人上做的两次调查是否有某种关联性. 这种检验也称为两个特征的独立性检验(test of independence, 也称为一致性检验, test of concordance)或关联性检验(test of association). 齐性检验与独立性检验的方法是相同的. 我们可以不加区分的使用这些方法, 只是最后对结果的解释不同罢了.

例如: x 为成功的次数, y 为试验的总次数, 检验其概率是否相等. p 值必须与 x y 的长度相等.

```
> x=1:5
> y=11:15
> prop.test(x,y)

        5-sample test for equality of proportions without continuity
        correction

data:  x out of y
X-squared = 2.6169, df = 4, p-value = 0.6238
alternative hypothesis: two.sided
sample estimates:
prop 1    prop 2    prop 3    prop 4    prop 5
0.0909091 0.1666667 0.2307692 0.2857143 0.3333333

Warning message:
In prop.test(x, y) : Chi-squared approximation may be incorrect

# 检验单个的二项比例时, 可以给出置信区间(CI)
> prop.test(1,5,p=0.6)

    1-sample proportions test with continuity correction

data:  1 out of 5, null probability 0.6
X-squared = 1.875, df = 1, p-value = 0.1709
alternative hypothesis: true p is not equal to 0.6
95 percent confidence interval:
```

```
0.01052995 0.70120895
sample estimates:
p
0.2

Warning message:
In prop.test(1, 5, p = 0.6) : Chi-squared approximation may be incorrect
```

15.3 二项比例中样本量及功效的估计

15.3.1 独立样本

二项比例在指定的假设$p = p_1$下, 功效的正态近似为

$$power = \Phi[\sqrt{(p_0q_0)/(p_1q_1)}(z_{\alpha/2} + |p_0 - p_1|\sqrt{n}/\sqrt{p_0q_0})]$$

样本量为

$$n = \frac{p_0q_0(z_{1-\alpha/2} + z_{1-\beta}\sqrt{(p_1q_1)/(p_0q_0)})^2}{(p_1 - p_0)^2}$$

例子为: 若一个地区的发病率为0.0015, 期望通过某种方法使发病率降低20%, 双侧检验水平为0.05, 功效为0.80, 则应该多大的样本才能发现差异?(生物统计学基础 10.5)

```
> power.prop.test( p1 = 0.0015, p2 = .0012,power=0.8)

     Two-sample comparison of proportions power calculation

              n = 235147.3
             p1 = 0.0015
             p2 = 0.0012
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

```
NOTE: n is number in *each* group
```

15.3.2 配对样本

TODO:

15.4 分位数检验

分位数检验可以使用二项检验来做. 这种方法可以用于次序(顺序)数据.

例如[1], 某大学新生参加入学考试, 其中15名新生的分数如下: 189 233 195 160 212 176 231 185 199 213 202 193 174 166 248. 认为这15名新生是随机样本. 已知多年来的新生成绩的上四分位数(第75百分位数)为193. 那么某大学的新生与其它大学的比较的假设可以是: 这15个成绩来自一个上四分位数为193的总体. 即H_0: 上四分位数为193. H_1: 上四分位数不是193. 最后结果p-值为0.035, 拒绝零假设. 即上四分位数不是193, 而是高于193.(低于193分的若是8的p-值为0.1399675, 9的p-值为0.4570977, 即可以接受零假设)

```
> z
 [1] 189 233 195 160 212 176 231 185 199 213 202 193 174 166 248
> length(z[z<=193])
[1] 7
> ((binom.test(7,15,0.75))$p.value)*2
[1] 0.03459968
> pbinom(7,15,0.75)*2
[1] 0.03459968
```

[1]实用非参数统计(第三版) Page 99 例 3.2.1

15.5 符号检验

符号检验实际上是二项检验的一个特例.

例子: 要检验两种防晒膏的效果. 随机涂敷于左右手臂, 阳光下一小时.假设我们只能判定手臂红色的程度

- A 防晒膏 ¿ B 防晒膏, 记为+1.
- A 防晒膏 ¡ B 防晒膏, 记为-1.
- 两者一样, 记为 0

45个人被测试, 22人A手臂较好, 18人B手臂较好. 5人两个手臂同样好.

首先去掉 0 值, 因为它对两种防晒膏的好坏不提供任何信息.

如果 +1 远多于 -1, 有理由相信, B 防晒膏的效果要好于 A. 若 -1 远多于 +1, 那么 A 的效果应该好于 B 的. 若 +1 和 -1 差不多, 那么两者效果可以认定没有显著差别.

实际上, 这是二项分布的一个特例. 此处假设

$$H_0 : p = 1/2 \quad vs. \quad H_1 : p \neq 1/2$$

此处 p 为 A 好于 B 的概率.

```
> binom.test(18,40)

        Exact binomial test

data:  18 and 40
number of successes = 18, number of trials = 40, p-value = 0.6358
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.2925884 0.6150932
```

```
sample estimates:
probability of success
               0.45
```

我们自己可以编一个函数. 注意函数参数不需要 alternative = c("two.sided", "less", "greater") (alternative ¡- match.arg(alternative)). prob也一定是与0.5比较, 不需要其它值.

```
# 定义函数
sign.test <- function (x, mu=0) { # does not handle NA
  n <- length(x)
  y <- sum(x<mu) # should warn about ties!
  if(y>n/2) y=n-y
  p.value <- pbinom(y,n,.5)*2
}

# 产生数据
> x <- sample(c(-1,0,1), 100, replace=T, prob=c(.4,.2,.4))
> sum(x<0)
[1] 43
> sum(x>0)
[1] 35
> sum(x==0)
[1] 22

> sign.test(x)
[1] 0.1933479
```

15.6 Cox-Stuart趋势性检验

一列数如果后面数比前面数趋于变大(上升趋势)或变小(下降趋势), 则称为有趋势的. 这个检验将后面的数和前面的数组成对, 并在对上进行符号检验. 若有趋势, 则每一对的一个数比另外一个有变大或变小的趋势. 如果没有趋势, 实际上代表独立同分布的随机变量.

数据组织如下. $X = x_1, \cdots, x_n$以某种顺序排列, 例如观察顺序. 把X从中间分开成为两个序列A与B. 若n为奇数则去掉中间的数. 将A,B按顺序一一对应. 如果$A_i > B_i$就用"+"代替, $A_i < B_i$就用"-"代替. 然后进行符号检验.

这个检验可以用来检验任何给定非随机模式. 我们假定

1. X 互相独立
2. X 至少是有序数据
3. X是同分布或有某种趋势

检验统计量 T="+" 的个数. 零分布为p=1/2.

下面是一个例子[2]. 记录了两年的小溪水流速度. 检验平均水流速度是否降低了. 结果p-值=0.3872. 接受零假设, 即水流速度没有降低.

月份	1	2	3	4	5	6	7	8	9	10	11	
第1年	14.60	12.20	104.00	220.00	110.00	86.00	92.80	74.40	75.40	51.70	29.30	1
第2年	14.20	10.50	123.00	190.00	138.00	98.10	88.10	80.00	75.60	48.80	27.10	1

```
> x=scan()
1: 14.6 14.2 12.2 10.5 104 123 220 190 110 138 86 98.1 92.8 88.1 74.4 80 75.4 75.6 5
25:
Read 24 items
> s=matrix(a,nr=2)
> s
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
[1,] 14.6 12.2  104  220  110 86.0 92.8 74.4 75.4  51.7  29.3  16.0
[2,] 14.2 10.5  123  190  138 98.1 88.1 80.0 75.6  48.8  27.1  15.7
> a=s[1,]
> b=s[2,]
> length(a[a<b])
[1] 5
```

[2]实用非参数统计(第三版) Page 122 例 3.5.3

```
> length(a[a>b])
[1] 7

> pbinom(5,12,p=0.5) # 直接计算p-值
[1] 0.387207
> binom.test(5,12,alt="less") # 使用二项检验

        Exact binomial test

data:  5 and 12
number of successes = 5, number of trials = 12, p-value = 0.3872
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
 0.0000000 0.6847622
sample estimates:
probability of success
             0.4166667
```

Cox-Stuart趋势性检验作为一个简单方法, 还可以检验两个随机变量是否有相关性. 首先将其中一个变量排序(通常是结点较少的变量). 如果有相关性, 那么另一个变量将会呈现出趋势性. 趋势相同就是正相关, 否则就是负相关.

Cochran(1937)比较了一些病人对两种药的反应, 来说明反应是否有正相关. 零假设为没有正相关性. 备择假设为有正相关性. 结果p-value = 0.03125, 且5个对全部是小于. 故拒绝零假设.

病人	1	2	3	4	5	6	7	8	9	10
药物1	0.70	-1.60	-0.20	-1.20	-0.10	3.40	3.70	0.80	0.00	2.00
药物2	1.90	0.80	1.10	0.10	-0.10	4.40	5.50	1.60	4.60	3.40

```
> x=scan()
1: .7 1.9 -1.6 0.8 -.2 1.1 -1.2 .1 -.1 -.1 3.4 4.4 3.7 5.5 .8 1.6 0 4.6 2. 3.4
21:
Read 20 items
> m=matrix(x,nr=2)
```

```
> m
    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  0.7 -1.6 -0.2 -1.2 -0.1  3.4  3.7  0.8  0.0   2.0
[2,]  1.9  0.8  1.1  0.1 -0.1  4.4  5.5  1.6  4.6   3.4
> order(m[1,])
 [1]  2  4  3  5  9  1  8 10  6  7
> m[,order(m[1,])] # 按照第一行排序
    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] -1.6 -1.2 -0.2 -0.1  0.0  0.7  0.8  2.0  3.4   3.7
[2,]  0.8  0.1  1.1 -0.1  4.6  1.9  1.6  3.4  4.4   5.5
> y=m[,order(m[1,])][2,]
> y
 [1]  0.8  0.1  1.1 -0.1  4.6  1.9  1.6  3.4  4.4  5.5
> z=matrix(y,nc=2) # 第二行配对
> z
    [,1] [,2]
[1,]  0.8  1.9
[2,]  0.1  1.6
[3,]  1.1  3.4
[4,] -0.1  4.4
[5,]  4.6  5.5
> z1=z[,1]
> z2=z[,2]
> z1
[1]  0.8  0.1  1.1 -0.1  4.6
> z2
[1] 1.9 1.6 3.4 4.4 5.5
> length(z1[z1<z2])
[1] 5
> length(z1[z1>z2])
[1] 0

> binom.test(5,5,alt="gre") # 检验趋势性

        Exact binomial test

data:  5 and 5
number of successes = 5, number of trials = 5, p-value = 0.03125
alternative hypothesis: true probability of success is greater than 0.5
95 percent confidence interval:
 0.5492803 1.0000000
```

```
sample estimates:
probability of success
```

Chapter 16

列联表

16.1 2×2列联表

Cochran 建议, 格子期望数小于5的不超过总格子数的1/5, 且没有一个格子的期望数小于1, 才可以使用卡方检验.

一般来说, 除了2×2列联表以外不使用Yate连续性修正. 因为经验发现这个修正不能增加对卡方分布的近似性.

16.1.1 Yate修正卡方检验

正态近似法和列联表法(修正的和非修正的卡方检验)都要求正态近似二项分布是有效的. 当不满足时, 特别是小样本时, 请使用基于超几何分布的Fisher精确检验.

Pearson's Chi-squared test: 若x为matrix至少2行或列, 则被看作2维连续table. 否则x y必须长度相等. 边际值被计算. 执行 Pearson's Chi-squared test.

若 correct = TRUE(默认), 则执行 Yate 连续性修正. 否则不执行修正.

若 simulate.p.value = TRUE 则执行 Monte Carlo 模拟来计算 p

值. B 为模拟的次数.

下面是生物统计学基础乳腺癌与初娩年龄关系的一个例子. 初娩大于30岁老年患乳腺癌的为683, 未患的1498, 初娩小于30岁老年患乳腺癌的2537, 未患的8747. 卡方检验p值为很小(2.2e-16), 说明乳腺癌与初娩年龄关系很显著.

```
# 生物统计学基础例 10.7 乳腺癌与初娩年龄的关系
> x <- matrix(c(683,1498,2537, 8747), nr = 2)
> x
     [,1] [,2]
[1,]  683 2537
[2,] 1498 8747

> prop = function(x) x/sum(x)
> apply(x,2,prop)
         [,1]      [,2]
[1,] 0.3131591 0.2248316
[2,] 0.6868409 0.7751684

> chisq.test(x)

        Pearson's Chi-squared test with Yates' continuity correction

data:  x
X-squared = 77.8851, df = 1, p-value < 2.2e-16

> chisq.test(x, simulate.p.value = TRUE, B = 10000)

        Pearson's Chi-squared test with simulated p-value (based on 10000
        replicates)

data:  x
X-squared = 78.3698, df = NA, p-value = 1e-04
```

下面是另外一个例子[1]. 从两辆货车上随机抽样来检查次品率是否一样. 第一辆次品有13件, 非次品73件. 第二辆次品17件,

[1]实用非参数统计(第三版). Page 130. 例 4.1.1

非次品57件. Yate修正卡方检验结果显示p-值=0.286, 未修正的显示p-值=0.204. 所以接受零假设, 即次品率无显著差异.

```
> x=matrix(c(13,17,73,57),nc=2)
> x
     [,1] [,2]
[1,]   13   73
[2,]   17   57
> chisq.test(x) # Yate修正卡方检验

        Pearson's Chi-squared test with Yates' continuity correction

data:  x
X-squared = 1.1372, df = 1, p-value = 0.2863

> chisq.test(x,corr=F) # 未经过Yate修正

        Pearson's Chi-squared test

data:  x
X-squared = 1.6116, df = 1, p-value = 0.2043
```

16.1.2 Fisher精确检验

当列联表的正态近似不满足时, 我们使用超几何分布的Fisher精确检验. 它特别适合于格子内期望数少(小样本)的情况, 即其中一个格子内的期望小于5.

假设行边际固定值为N_1, N_2, 列边际固定值为M_1, M_2. 下面考察4个边际全都固定的2×2表的个数. 我们重新安排行和列使总有$M_1 <= M_2, N_1 <= N_2$. 在边际全都固定时, 4个格子的观察数实际上只有1个可以固定, 例如(1,1)可以随机变动. 其它都可以由(1,1)及边际数给出. 记X为(1,1)格子内的数, 则X的概率分布为

$$P(X) = \frac{N_1!N_2!M_1!M_2!}{N!a!(N_1-a)!(M_1-a)!(M_2-N_1+a)!}, a = 0, 1, \cdots, min(N_1, M_1)$$

此处$N = N_1 + N_2 = M_1 + M_2$. 这样的概率分布为超几何分布. 其

期望为

$$E(X) = \frac{M_1 N_1}{N}$$

方差为

$$Var(X) = \frac{M_1 M_2 N_1 N_2}{N^2(N-1)}$$

fisher.test 检验2×2列联表的优势比是否为1. 详细参考流行病部分优势比和 Mantel-Haenszel检验. epicalc 包的 cc 函数可以精确计算优势比, 有时候与 fisher.test 结果不太一样.

下面是一个例子[2]. 考察饮食中高盐与低盐是否和心血管疾病有关. 收集了两组死亡的男性, 其中一组原因是心血管疾病, 35个中有5个是高盐的. 另外一组是其它疾病, 25人中有2人是高盐的. 结果无论是双侧还是单侧, 都不显著. 即饮食与死亡原因无显著关系.

```
> x=matrix(c(2,5,23,30),nc=2)
> x
     [,1] [,2]
[1,]    2   23
[2,]    5   30
> fisher.test(x) # 双侧检验

        Fisher's Exact Test for Count Data

data:  x
p-value = 0.6882
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.04625243 3.58478157
sample estimates:
odds ratio
  0.527113
```

[2]生物统计学基础. Page 358. 例 10.20

```
> fisher.test(x,alt="less") # 单侧检验

        Fisher's Exact Test for Count Data

data:  x
p-value = 0.3747
alternative hypothesis: true odds ratio is less than 1
95 percent confidence interval:
 0.000000 2.799135
sample estimates:
odds ratio
  0.527113
```

16.1.3 联合多个表: Mantel-Haenszel检验

有时候需要将多个2×2列联表合成一个做整体分析. 当一个整体试验包括几个在不同环境中操作的小试验时, 零假设下共同的概率随环境的不同而不同, 并且每个小试验都有自己的2×2列联表这时常需要这种处理. 因为每个列联表的环境不同, 它们不能合成单一的2×2列联表.

Mantel与Haenszel(1959)提出了一个合并多个2×2列联表的方法. 又称为 Cochran-Mantel-Haenszel 卡方检验. 假设表的数目$k >= 2$. 第i个表的形式为 每个列联表的假设条件与Fisher精确检验相

	列1	列2	
行1	x_i	$r_i - x_i$	r_i
行2	$c_i - x_i$	$N_i - r_i - c_i + x_i$	$N_i - r_i$
	c_i	$N_i - c_i$	N_i

同, 并且几个列联表是由独立的试验得到的.

零假设: 在第i个列联表中, 令p_{1i}是第一行第一列中的观测的概率, p_{2i}是第二行第二列相应的概率. 对于双边检验有

$$H_0 : p_{1i} = p_{2i}, i = 1, 2, \cdots, k$$

$H_1 : p_{1i} > p_{2i}$ $p_{1i} < p_{2i}$ 对某个i成立, 但不同时成立

对于左单边检验有

$$H_0 : p_{1i} >= p_{2i}, i = 1, 2, \cdots, k$$
$$H_1 : p_{1i} <= p_{2i} \text{对所有的i,且对某个i,} \quad p_{1i} < p_{2i}$$

对于右单边检验有

$$H_0 : p_{1i} <= p_{2i}, i = 1, 2, \cdots, k$$
$$H_1 : p_{1i} >= p_{2i} \text{对所有的i,且对某个i,} \quad p_{1i} > p_{2i}$$

若行列非随机, 检验统计量

$$T = \frac{\sum x_i - \sum \frac{r_i c_i}{N_i}}{\sqrt{\sum \frac{r_i c_i (N_i - r_i)(N_i - c_i)}{N_i^2 (N_i - 1)}}}$$

若零假设为真, T的分布近似标准正态分布, 并且可以通过连续修来提高精确性, 即对于左边的概率, 可以将T的分子加0.5, 对于右边的概率, 减去0.5. 这样得到的概率在多数情况下会更精确.

若行列总和是随机的, 那么用下面的统计量更准确

$$T = \frac{\sum x_i - \sum \frac{r_i c_i}{N_i}}{\sqrt{\sum \frac{r_i c_i (N_i - r_i)(N_i - c_i)}{N_i^3}}}$$

参考流行病部分Mantel-Haenszel检验. epicalc包的mhor函数也可以计算.

下面是R自带的一个例子. 比较立即注射和1.5小时后注射盘尼西林(Penicillin)的效果, 分为治愈(Cured)和死亡(Died). 盘尼西林的水平分为5个, "1/8", "1/4", "1/2", "1", "4". 双侧检验表明立即注射比1.5小时后注射的治愈率要高. 精确检验和单边检验的结果也相同.

```
> Rabbits <-
```

```
+     array(c(0, 0, 6, 5,
+             3, 0, 3, 6,
+             6, 2, 0, 4,
+             5, 6, 1, 0,
+             2, 5, 0, 0),
+           dim = c(2, 2, 5),
+           dimnames = list(
+              Delay = c("None", "1.5h"),
+              Response = c("Cured", "Died"),
+              Penicillin.Level = c("1/8", "1/4", "1/2", "1", "4")))
> Rabbits
, , Penicillin.Level = 1/8

     Response
Delay  Cured Died
  None    0   6
  1.5h    0   5

, , Penicillin.Level = 1/4

     Response
Delay  Cured Died
  None    3   3
  1.5h    0   6

, , Penicillin.Level = 1/2

     Response
Delay  Cured Died
  None    6   0
  1.5h    2   4

, , Penicillin.Level = 1

     Response
Delay  Cured Died
  None    5   1
  1.5h    6   0

, , Penicillin.Level = 4
```

```
     Response
Delay  Cured Died
  None     2    0
  1.5h     5    0

> mantelhaen.test(Rabbits) ## Classical Mantel-Haenszel test

       Mantel-Haenszel chi-squared test with continuity correction

data:  Rabbits
Mantel-Haenszel X-squared = 3.9286, df = 1, p-value = 0.04747
alternative hypothesis: true common odds ratio is not equal to 1
95 percent confidence interval:
  1.026713 47.725133
sample estimates:
common odds ratio
                7

> mantelhaen.test(Rabbits, exact = TRUE) # 精确法  p = 0.040
> mantelhaen.test(Rabbits, exact = TRUE, alt = "greater") #单  边
检验  p = 0.020
```

16.1.4 匹配数据二项比例检验–McNemar检验

如果数据不是独立的, 即可以形成匹配数据, 则Yate修正卡方检验是不合适的.

下面是《生物统计学基础》10.4 Page 360 中的一个例子. 按年龄(或其它条件)配对621对病人, 配对的1人随机指定使用A方法治疗, 另外一人使用B方法治疗. 其中A方法生存5年以上, B方法也生存5年以上的有510对；A方法生存5年以上, B方法生存少于5年的有5对；A方法生存少于5年, B方法生存5年以上的有16对；A方法生存少于5年, B方法也少于5年的有90对. 检验A, B两种方法的差异是否显著.

在成对的匹配中, 结局相同的对称为一致对(concordant pair). 结局不同的称为不一致对(discordant pair). 此例中, 有510+90=600 个一致对. 有 5+16=21 个不一致对. 一致对不提供

信息, 故分析时抛弃之. 我们集中研究一致对.

不一致对中, 使用A处理后有事件发生而B处理后未发生, 称为A型不一致对. 否则称为B型不一致对.

记p=A型不一致对的概率. 如果两个处理等效, 那么A型与B型不一致对的数目应该相等. 即p=1/2. 这时, 零假设为: p=1/2. 备择假设: $p \neq 1/2$.

此例的优势比估计请参考流行病部分的匹配数据优势比估计.

我们可以使用精确的二项比例检验, 也可以使用正态近似法. 两种方法都在 McNemar 检验中.

```
# 生物统计学基础 10.4 中的例子.
> Treat<-matrix(c(510,16,5,90),nr=2,
   dimnames=list("A result"=c("more 5 years","less 5 years"),
             "B result"=c("more 5 years","less 5 years")))
> Treat
           B result
A result      more 5 years less 5 years
  more 5 years         510          5
  less 5 years          16         90

> mcnemar.test(Treat)

       McNemar's Chi-squared test with continuity correction

data:  Treat
McNemar's chi-squared = 4.7619, df = 1, p-value = 0.02910
```

R中的一个例子

```
# R的例子
> Performance <-
+     matrix(c(794, 86, 150, 570),
+           nr = 2,
```

```
+           dimnames = list("1st Survey" = c("Approve", "Disapprove"),
+                           "2nd Survey" = c("Approve", "Disapprove")))
>     Performance
          2nd Survey
1st Survey   Approve Disapprove
  Approve       794       150
  Disapprove     86       570

> mcnemar.test(Performance)

       McNemar's Chi-squared test with continuity correction

data:  Performance
McNemar's chi-squared = 16.8178, df = 1, p-value = 4.115e-05
```

16.2 R×C列联表

16.2.1 概率差异(倾向性, 趋势性)的卡方检验

共有r个总体, 从每个总体抽取随机样本, 每个样本的每个观察都可以归入c个不同的类别. 数据排列为下面的形式 每个样

	类1	类2	...	类c	总和
总体1	O11	O12	...	O1c	n1
...	...	...	...	...	...
总体r	Or1	Or2	...	Orc	nr
总和	C1	C2	...	Cc	N

本都是随机样本, 不同样本输出结果是独立的, 每个观测只能归入其中一类.

检验统计量为

$$T = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{ij}-E_{ij})^2}{E_{ij}}, E_{ij} = \frac{n_i C_j}{N}$$

T的零分布为渐近自由度为$(r-1)(c-1)$的卡方分布. 其精确值很难计算, 所以几乎不用.

假设为

$$H_0 : p_{1j} = p_{2j} = \cdots = p_{rj} for all j$$
$$H_1 : 每列至少存在两个概率不相等$$

注意, 零假设只是说概率相等. 没有必要规定概率是多少. 而且检验结果并没有告诉我们关联性的性质, 即是否有倾向性.

Wilcoxon 秩和检验实际上是有倾向性卡方检验的一个特例.

Cochran(1952)发现, 如果存在$E_{ij} < 1$或超过20%的$E_{ij} < 5$, 那么近似可能很差. 但是根据很多其它学者未发表的研究表明, 这似乎太保守了. Conover(实用非参数统计的作者)认为, 即使一些$E_{ij} < 5$, 如果r与c不太小的话, 检验也是有效的.

下面是一个例子.[3]. 列代表初娩的年龄, 分别是小于20岁, 20-24岁, 25-29, 30-34, 大于35岁. 分为疾病和对照. p-值¡0.001, 说明初娩年龄与乳腺癌是有关系的.

```
> x=matrix(c(320,1422,1206,4432,1011,2893,463,1093,220,406),
   nr=2,dimnames=list(c("疾病","对照"),
   c("小于20岁", "20-24岁", "25-29", "30-34", "大于35岁")))
> x
     小于20岁 20-24岁 25-29 30-34 大于35岁
疾病      320    1206  1011   463      220
对照     1422    4432  2893  1093      406

> chisq.test(x)

        Pearson's Chi-squared test

data:  x
X-squared =
130.172, df = 4, p-value < 2.2e-16

# 按照生物统计学基础给出的算法编写的函数
```

[3]生物统计学基础 page 374. 例 10.33 初娩与乳腺癌关系的例子

```
chisq.tendency.test<-function(x)
{
   s=1:dim(x)[2]# 组 的 得 分 变 量
   n_i=apply(x,2,sum) #
   n=sum(n_i) #
   x_all = sum(x[1,])
   p=x_all/n
   q=1-p
   A=sum(x[1,]*s)-x_all*sum(s*n_i)/sum(n)
   B=p*q*(sum(s^2*n_i)-sum(s*n_i)^2/sum(n))
   res.chisq=A^2/B
   res.p.value=1-pchisq(res.chisq,df=1)
   res=list(chisq=res.chisq,p.value=res.p.value,A=A)
   res
}
> chisq.tendency.test(x)
$chisq
[1] 128.8386

$p.value
[1] 0

$A
[1] 566.8084

# coin包 里 面 的 计 算 精 确 值 的 函 数
> library(coin)
> chisq_test(as.table(x))

        Asymptotic Pearson's Chi-Squared Test

data:  Var2 by Var1 (A, B)
chi-squared = 130.172, df = 4, p-value < 2.2e-16
```

下 面 是 另 外 一 个 例 子[4]. 检 验 公 立 中 学 与 私 立 中 学 的 某 次 测 验 成 绩 是 否 一 样. 结 果p-值¡0.001, 因 此 我 们 说 两 种 学 校 测 验 成 绩 不 同.

[4]实 用 非 参 数 统 计(第 三 版). Page 144. 例 4.2.1

```
> x=matrix(c(6,30,14,32,17,17,9,3),nr=2,
   dimnames=list(c("私立","公立"),c("0-275","276-350","351-425","426-500")))
> x
    0-275 276-350 351-425 426-500
私立     6      14      17       9
公立    30      32      17       3
> chisq.test(x)

        Pearson's Chi-squared test

data:  x
X-squared = 17.2858, df = 3, p-value = 0.0006172

Warning message:
In chisq.test(x) : Chi-squared近似算法有可能不准
```

16.2.2 独立性卡方检验

此检验与概率差异的卡方检验计算方法是一样的, 只不过对数据的解释不同.

数据为: 已知容量为N的随机样本. 观察值根据两个准则划分为几类. 按照第一个准则, 每个观察值可以归入r类(行)中的一类. 按照第二个准则, 每个观察值可以归入c类(列)中的一个.

假设为: H_0: 对任意i,j, 事件"一个观测值在行i"与事件"同样的观测在列j"是独立的. 即P(行i,列j)=P(行i)-P(列j), 对所有i,j. H_1: P(行i,列j)≠P(行i)-P(列j), 对所有i,j.

下面是一个例子[5]. 学生根据被录取的院校和是否从州内和州外毕业两个标准来分类. 零假设是每个学生被录取的院系与是否在州内和州外读高中无关. 结果p-值较大, 接受零假设.

```
> x=matrix(c(16,14,14,6,13,10,13,8),nr=2,
   dimnames=list(c("州内","州外"),c("工程学院","艺术学院", "经济学院","其它")))
```

[5]实用非参数统计(第三版). Page 147. 例 4.2.2

```
> x
    工程学院 艺术学院 经济学院 其它
州内       16       14       13   13
州外       14        6       10    8
> chisq.test(x)

        Pearson's Chi-squared test

data:  x
X-squared = 1.5242, df = 3, p-value = 0.6767
```

16.2.3 固定边缘分布的卡方检验

数据纳入r×c列联表, 与前两个不同的是行列总和固定而非随机. 此处的假设检验可以取前两个之一.

固定边际总和的卡方检验也可以检验两个随机变量X和Y是否独立.

下面是一个例子[6]. X与Y的个数的观察值(落入坐标X,Y区域内的点的个数)如下, 构成二元随机变量(X,Y). 可以看到, p-值很小, 所以X与Y不独立.

```
> x=matrix(c(0,2,4,4,1,1,4,2,0,0,3,3),nr=3,dimnames=list("X"=c(),"Y"=c()))
> x
     Y
X     [,1] [,2] [,3] [,4]
  [1,]    0    4    4    0
  [2,]    2    1    2    3
  [3,]    4    1    0    3
> chisq.test(x)

        Pearson's Chi-squared test

data:  x
X-squared = 14, df = 6, p-value = 0.02964
```

[6]实用非参数统计(第三版). Page 150. 例 4.2.3

下面是一个具体的例子[7]. 一位心理学家要求被测人学习25个单词. 给被测人25张蓝色卡片, 其中名词, 动词, 形容词, 副词, 介词各5个. 白色卡片是另外25个词, 词性及个数与蓝色一样. 允许被测人5分钟配对卡片, 5分钟学习卡片. 然后给被测人读蓝色卡片的单词, 被测人尽量提供与所读单词相关的白色卡片上的词. 心理学家关心配对的结构是否显示有某种次序, 例如与词性相关. 零假设为: 没有按照词性配对. 备择假设: 倾向于将蓝色卡片的一种词性与白色的卡片的一种词性配对(不一定相同). 结果显示p-值很小, 拒绝零假设.

```
> a=scan()
1: 0 4 0 0 1 3 1  0 0 1 0 0 0 5 0 0 0 5 0 0 2 0 0 0 3
26:
Read 25 items
> b=matrix(a,nr=5,dimnames=list(c("名词", "动词", "形容词", "副
词", "介词"),
   c("名词", "动词", "形容词", "副词", " 介词")))
> b
      名词 动词 形容词 副词 介词
名词     0    3      0    0    2
动词     4    1      0    0    0
形容词   0    0      0    5    0
副词     0    0      5    0    0
介词     1    1      0    0    3
> chisq.test(b)

        Pearson's Chi-squared test

data:  b
X-squared = 66, df = 16, p-value = 4.953e-08

Warning message:
In chisq.test(b) : Chi-squared近似算法有可能不准
```

[7]实用非参数统计(第三版). Page 151. 例 4.2.4

16.3 三向及多向列联表

以上的列联表分为行列两个方向, 也可以称为双向列联表(two-way contingency table). 若观测按照三个或以上准则分类, 那么数据可以使用三向(或多向列联表). 我们将检验统计量变换为

$$T = \sum_{ij} \frac{[O_{ij} - N\frac{R_i}{N}\frac{C_j}{N}]^2}{N\frac{R_i}{N}\frac{C_j}{N}}$$

T具有$(r-1)(c-1)$的自由度. 在三向列联表中, 有r行, c列, t块. 记块总和为B_k

$$R_i = \sum_{jk} O_{ijk}$$

$$C_j = \sum_{ik} O_{ijk}$$

$$B_k = \sum_{ij} O_{ijk}$$

期望的估计为

$$E_{ijk} = N\frac{R_i C_j B_k}{N * N * N}$$

检验统计量为

$$T = \sum_{ijk} \frac{(O_{ijk} - E_{ijk})^2}{E_{ijk}}$$

T为自由度为$rct - r - c - t + 2$的卡方分布.

对数线性模型可以成功的用来分析多向列联表.

16.4 中位数(分位数)检验

不同总体是否有相同的中位数? 中位数检验实际上是固定

行列总和的卡方检验的具体应用. 因为非常重要, 所以单独讨论.

从c个总体中抽取容量为n_i的随机样本. 首先确定联合总体的中位数, 称为总中位数. 然后每个随机样本计算超过总中位数的个数为O_1i, 小于等于总中位数的个数为O_2i, 将频数排列成2×c列联表中.

零假设为: c个总体有相同的中位数. 备择假设为: 至少两个总体中位数不同.

若零假设被拒绝, 则可以对2×2列联表重复检验, 以发现哪两个总体的中位数不同. 但是要小心显著性水平α上升(下降??).

下面是一个例子[8]. 使用4种不同的方法培植玉米. 在分隔成若干块的地上随机使用1种方法. 计算每亩的产量. 为确定产量是否由种植方法不同引起, 我们采用中位数检验. 零假设为: 所有方法有相同的亩产中位数. 备择假设: 至少两种方法亩产中位数不同. 结果p-值很小, 拒绝零假设.

```
> x1=c(83,91,94,89,89,96,91,92,90)
> x2=c(91,90,81,83,84,83,88,91,89,84)
> x3=c(101,100,91,93,96,95,94)
> x4=c(78,82,81,77,79,81,80,81)
> my_aaa<-function(x,med){
+ a=length(x[x<=med])
+ b=length(x[x>med])
+ res=c(a,b)
+ res}
> y=matrix(c(my_aaa(x1,89),my_aaa(x2,89),my_aaa(x3,89),my_aaa(x4,89)),
   nr=2,dimnames=list(c("<=89",">89"),c("x1","x2","x3","x4")))
> y
    x1 x2 x3 x4
<=89  3  7  0  8
>89   6  3  7  0
> chisq.test(y)

        Pearson's Chi-squared test
```

[8]实用非参数统计(第三版). Page 158. 例 4.3.1

```
data:  y
X-squared = 17.5431, df = 3, p-value = 0.0005464
```

如果检验的是上分位数或下分位数, 那么把列联表的计数更换为相应的分位数即可.

16.5 关联性(相依性)度量

相依性度量很大程度上取决于个人的决定. 一般会依据传统的习惯, 而不是统计学的考虑.

16.5.1 Cramer关联系数

由Cramer(1946)提出, 使用T除以可能达到的最大值(极端不平衡列联表中达到最大, 为N(min(r,c)-1)). 计算公式为

$$R = \sqrt{\frac{T}{N(min(r,c)-1)}}$$

其中T为卡方检验统计量, N为观测总数. 若强行列相关, 则R接近1.

下面是计算公立,私立学校考试成绩的例子.

```
> coef.cramer<-function(x){
+ r=sqrt(chisq.test(x)$statistic/(sum(x)*(min(dim(x))-1)))
+ names(r)<-"Cramer Coefficient"
+ r
+ }
> x=matrix(c(6,30,14,32,17,17, 9,3),nr=2)
> x
     [,1] [,2] [,3] [,4]
[1,]    6   14   17    9
```

```
[2,]   30   32   17    3
> coef.cramer(x)
Cramer Coefficient
       0.3674853
# x10倍后检验统计量增加10倍, 但是cramer系数不变
> chisq.test(x*10)

       Pearson's Chi-squared test

data:  x * 10
X-squared = 172.8581, df = 3, p-value < 2.2e-16

> coef.cramer(x*10)
Cramer Coefficient
       0.3674853
```

16.5.2 Pearson关联系数

均方关联系数 Pearson's coefficient of mean square contingency, 由Yule和Kendall(1950)给出. 文献[11]也称为列联系数(contingency coefficient). 定义为

$$R=\sqrt{\frac{T}{N+T}}$$

记q=min(r,c), 因为T的最大值为N(q-1), 故R的最大值为

$$R_{max}=\sqrt{\frac{N(q-1)}{N+N(q-1)}}=\sqrt{\frac{q-1}{q}}<1.0$$

```
coef.pearson<-function(x){
   s<-chisq.test(x)$statistic
   r<-sqrt(s/(sum(x)+s))
   names(r)<-"contingency coefficient"
```

```
   r}
> coef.pearson(x)
contingency coefficient
           0.3449319
> coef.pearson(x*10)
contingency coefficient
           0.3449319
```

16.5.3 Pearson均方关联系数

此系数也具有Pearson关联系数的特点, 被Yule和Kendall(1950)称为mean square contingency coefficient. 定义为

$$R = T/N$$

我们有$0 <= R <= q-1$

```
> r=chisq.test(x)$statistic/sum(x)
> names(r)<-"mean square contingency coefficient"
> r
mean square contingency coefficient
                     0.1350454
```

16.5.4 TschuProw系数

定义

$$R = \sqrt{\frac{T}{N\sqrt{(r-1)(c-1)}}}$$

16.5.5 正关联和负关联

2×2列联表有时候区分正关联和负关联是有意义的. 例如根据父亲和母亲的头发颜色将40个孩子分类[14](Page 167 例 4.4.5)[9].

```
> x<-matrix(c(28,5,0,7),nr=2,
   dimnames=list("母亲"=c("黑色","金色"),"父亲"=c("黑色","金色")))
> x
     父亲
母亲 黑色 金色
  黑色   28    0
  金色    5    7
```

Phi系数(phi coefficient)就是这样的系数, 定义为

$$R = \frac{ad - bc}{\sqrt{r_1 r_2 c_1 c_2}}$$

其中$r_1 r_2 c_1 c_2$为行列的和. abcd分别为四个格子的观测数. 当$ad - bc > 0$为正关联, $ad - bc < 0$为负关联. 下面计算头发颜色例子的Phi系数.

```
> r<-(x[1,1]*x[2,2]-x[1,2]*x[2,1])/sqrt(prod(colSums(x),rowSums(x))) # prod 为乘法函数
> r
[1] 0.7035265
```

其它2×2列联表关联性系数还有Yule和Kendall(1950)提出的

$$R = \frac{ad - bc}{ad + bc}$$

[9]作者注: 此例似乎应该再根据孩子的头发颜色分为几个2×2列联表

Ives和Gibbons(1967)提出的

$$R = \frac{(a+d)-(b+c)}{a+b+c+d}$$

可定义的关联性度量方法有很多, 选择什么取决于个人喜好.

16.5.6 kappa统计量-重复性度量

在两个类型变量彼此不做预测时, 这个指标很有用处([11], page 386). 它可以表示两个类型数据关联性大小. 特别在可靠性研究(reliability study)中, 人们希望定量表示出对相同变量做多次测量时, 它的重复性有多大.

假设有n个受试者, 都接受了两次关于同一问题的调查, 则kappa统计量常用于测度两次调查的可重复性. 公式为

$$k = \frac{p_o - p_e}{1 - p_e}$$

其中p_o为两次调查中一致性概率. $p_e = \sum_{i=1}^{c} a_i b_i$为零假设下(两次调查彼此独立, 即无重复性)两次调查的期望一致的概率, 此处$a_i b_i$为c×c列联表中两个调查第i个类型的边际概率.

零假设: 两次调查彼此独立, 即无重复性. 备择假设: 两次调查有一定的重复性.

Landis 及 Koch (1977) 提出下面的参考标准

`k>0.75` 表示极好的重复性
`0.4<=k<=0.75` 好的重复性
`0<=k<0.4` 边界(勉强够格)的重复性

Fleiss 提供了kappa统计量的进一步信息, 包括多于两次调查时如何判断重复性.

kappa 值也常常用做相同变量重复估计之间是否有重复性的一种测度.

如果我们对两个变态变量上反应的一致性有兴趣, 而其中一个变量的反应可以作为金标准, 则灵敏度及特异度是比 kappa 统计量更好的指标.

下面是函数及一个例子. 数据 x 为第一次调查及第二次调查牛肉消费的结果, 分为每周消费1次以下和多于1次. 最后看看两次调查的重复性如何. 结果p-值很小, 拒绝零假设, 两次调查有重复性, 重复性大小为 0.378.

```
kappa.test <- function(x)
{
   N=sum(x)
   Po=sum(diag(x)/N) # 观察到的一致数
   mr=apply(x,1,sum) # 行边际
   mr=mr/N
   mc=apply(x,2,sum) # 列边际
   mc=mc/N
   Pe=sum(mc * mr) # 期望一致数
   k=(Po-Pe)/(1-Pe) # kappa统计量
   se_k = sqrt((Pe+Pe^2-sum(mr*mc*(mr+mc)))/(N*(1-Pe)^2)) # kappa统
计量的标准误
   z=k/se_k # 检验统计量
   p.value=1-pnorm(z) # p 值
   res=list(kappa=k,se_k=se_k,p.value=p.value,z=z)
}
```

```
> x=matrix(c(136,69,92,240),nr=2, dimnames=list("1st survey"=c("<= 1 time/week","
> x
             2st survey
1st survey      <= 1 time/week > 1 time/week
  <= 1 time/week          136           92
  > 1 time/week            69          240

> k=kappa.test(x)
> k
$kappa
[1] 0.3781906
```

```
$se_k
[1] 0.04298259

$p.value
[1] 0

$z
[1] 8.798692

> library(epiR)
> epi.kappa(a=136,c=69,b=92,d=240)
$kappa
       est    lower    upper
1 0.3781906 0.2978196 0.4585616

$mcnemar
  test.statistic df    p.value
1      3.285714  1 0.06988521
```

16.5.7 相关性的检验

如何使用相关系数R作为检验统计量检验

H_0: 不存在相关(正相关或负相关), H_1: 存在相关(正相关或负相关)

我们可以看到当卡方检验统计量T大时, R也比较大. 我们可以使用R表示T, 然后用T作为检验统计量. 当T显著时, R也显著. 正负可以看$ad - bc$的值.

16.6 卡方拟合优度检验

拟合优度检验: 若x是matrix只有一行或一列, 或x是vector, 且y没

有给出, 那么会执行"goodness-of-fit test"(拟合优度检验), x被认为是一维列联表. 检验x概率是否与p相等, 若p未给出, 则检验是否概率都一样, 即均匀分布.

Cochran(1952)建议观测期望E_i不小于1, 且不超过20%的不小于5. 最近的研究表明这个限制可以放宽. [14](page 173, 其它修正方法)

(二项比例关联性检验(Pearson's Chi-squared test): 若x为matrix至少2行或列, 则被看作2维连续table. 否则x y必须长度相等. 边际值被计算. 执行 Pearson's Chi-squared test. 此检验为二项比例关联性检验)

下面是一个正态拟合的例子

```
> n=1000
> x=rnorm(n)
> b=seq(-2,2,0.2)

# 计算正态分布的理论概率
> p=c(pnorm(b[1]),diff(pnorm(b)),1-pnorm(b[length(b)]))
> p
 [1] 0.02275013 0.01318019 0.01886897 0.02595737 0.03431301 0.04358558
 [7] 0.05320014 0.06239772 0.07032514 0.07616203 0.07925971 0.07925971
[13] 0.07616203 0.07032514 0.06239772 0.05320014 0.04358558 0.03431301
[19] 0.02595737 0.01886897 0.01318019 0.02275013
> sum(p)
[1] 1

# 实际频数
> bre=c(-1000,b,1000)
> h=hist(x, breaks=bre)
> h$counts # 实际频数
 [1] 18  8 26 23 31 48 53 71 72 84 79 62 61 73 64 49 50 37 24 19 21 27
> sum(h$counts)
[1] 1000

# 卡方检验. 默认执行连续性修正. p值>0.05则两个频率差异不显著
> chisq.test(p*1000,h$counts)
```

```
        Pearson's Chi-squared test

data:  p * 1000 and h$counts
X-squared = 440, df = 420, p-value = 0.2412

Warning message:
Chi-squared近似算法有可能不准 in: chisq.test(p * 1000, h$counts)
```

16.7 相关观测的Cochran检验

普通的处理是所有样本分为c组, 每组使用一个处理方法, 得到一个2×c列联表. 但是, 为了提高功效, 我们有时候需要对每个样本都用c种方法独立的处理. 在这里我们使用r为样本个数或区组数(区别于通常的n). 我们得到了一个r×c列联表, 其中每个观测值为0或1. 行总和为R_i, 列总和为C_i.

假设样本是随机的(即随机选取的). 处理的结果可以按照某种方式分为两种, 记为0和1.

检验统计量为

$$T = c(c-1)\frac{\sum_{j=1}^{c}(C_j - \frac{N}{c})^2}{\sum_{i=1}^{r} R_i(c-R_i)}$$

下式计算更适合

$$T = \frac{c(c-1)\sum_{j=1}^{c} C_j^2 - (c-1)N^2}{cN - \sum_{i=1}^{r} R_i^2}$$

T的精确分布难以求得, 大样本(r比较大)逼近后近似自由度c-1的卡方分布. 零假设为: 所有处理效果相同. 备择假设为: 处理之间效果有差异. 记$p_j = P$为列j中出现1的概率, 则零假设可以描述为: 每个处理中有$p_1 = p_2 = \cdots = p_c$. 备择假设为: 某两个处理i,j有$p_i \neq p_j$.

若拒绝了零假设, 可以使用McNemar对c个处理进行两两比较.

若仅考虑c=2(两种处理), 那么Cochran检验与McNemar检验是一样的.

下面是一个例子([14], page 181, 例 4.6.1). 3个篮球爱好者对12场比赛进行预测. 比赛是从所有比赛中随机选取的. 预测准确记为1, 否则记为0. 零假设为: 3个爱好者的预测是等有效的. 备择假设为: 其中至少2个爱好者的预测不是等有效的. 数据及结果见下面. p-值很小, 拒绝零假设.

```
cochran.test<-function(x){
   c=dim(x)[2]
   C=colSums(x)
   R=rowSums(x)
   N=sum(x)
   T=c*(c-1)*sum((C-N/c)^2)/sum(R*(c-R))
   p=1-pchisq(T,df=c-1)
   res<-list(statistic=T,p.value=p, df=c-1)
   res
}
> x<-matrix(rbinom(36,size=1,p=0.7),nc=3)
> colSums(x)
[1]  3 10  7
> x
      [,1] [,2] [,3]
 [1,]    0    1    0
 [2,]    1    1    1
 [3,]    0    1    0
 [4,]    1    1    1
 [5,]    1    1    1
 [6,]    0    1    1
 [7,]    0    1    1
 [8,]    0    1    0
 [9,]    0    1    1
[10,]    0    0    0
[11,]    0    0    1
[12,]    0    1    0
> cochran.test(x)
$statistic
[1] 9.25
```

```
$p.value
[1] 0.009803655

$df
[1] 2
```

16.8 其它分析方法

16.8.1 似然比统计量

$$T = \frac{\sum(O_i - E)^2}{E}$$

这个统计量是Pearson(1900,1922)引入的, 称为Pearson卡方统计量. 以上使用的分析方法都是这种. 下面是一种不同的方法, 称为似然比检验法, 统计量为

$$T = 2\sum O_i ln(\frac{O_i}{E_i})$$

它来自于统计学里的似然比理论, 与Pearson统计量服从同样自由度的卡方分布, 属于Wilks(1935,1938),收到广泛运用. 但是它的一个弊端是, 如果$N/rc < 5$, 卡方分布的效果就不好. Agresti(1990)说明如果$N/rc < 1$, Pearson方法的近似也不好.

16.8.2 对数线性模型

这种方法可以很好的分析三维以上的列联表. 也可以在线性对数模型中使用 Pearson 统计量或似然比统计量, 不同的是估计E的方法是迭代法.

在双向列联表中, 零假设可以描述为: 对所有ij, $p_{ij} = p_i * p_j$. 两边取对数, 零假设变为: 对所有ij, $logp_{ij} = logp_i + logp_j$. 对零假

设的检验变为检验格子概率的对数是否是边际概率对数的线性函数.

Chapter 17

秩检验

17.1 Wilcoxon符号-秩检验

Wilcoxon符号-秩检验 ([11] Page323 例 9.12. 计算方法及公式见参考文献) 与配对的t检验类似, 是基于二项分布的检验. 在这里, 我们不关心具体打分的大小, 但是打分的秩次(相对大小)是有意义的.

下面是一个例子. 若防晒膏问题中晒红程度打分左手臂打分为x, 右手臂打分为y 则这种情况下适合使用Wilcoxon符号-秩检验. 检验的是变量是否关于 mu 对称. 即为中位数的检验. 设差值d=x-y.

```
# 打分的差值
> d=c(-8,rep(-7,3),-6,-6,-5,-5,-4,rep(-3,5),rep(-2,4),
   rep(-1,4),rep(3,2),rep(2,6),rep(1,10))
> d
 [1] -8 -7 -7 -7 -6 -6 -5 -5 -4 -3 -3 -3 -3 -3 -2 -2 -2 -2 -1 -1 -1 -1  3  3  2
[26]  2  2  2  2  2  1  1  1  1  1  1  1  1  1  1
> wilcox.test(d) # 默认检测 mu=0

        Wilcoxon signed rank test with continuity correction

data:  d
```

```
V = 248, p-value = 0.02869
alternative hypothesis: true location is not equal to 0

Warning message:
In wilcox.test.default(d) : 无法精確計算带连结的p值
```

17.2 Mann-Whitney检验和Hodges-Lehmann估计

wilcox.test 执行单样本(y=NULL)和两样本(x,y)秩和检验. 后者也叫做Mann-Whitney test或Wilcoxon's U test. ([11], Page 328. [14] Page 195. 参考文献有详细描述和其它方法的比较).

Mann-Whitney 检验类似于两独立样本的t检验. (大数据量时可以使用t.test. kruskal.test 用于两或多样本的检验).

Hodges-Lehmann 估计位移的置信区间: Mann-Whitney检验不能检验出差值(位移)的置信区间, 若想知道区间, 一个方法是使用Mann-Whitney检验多次变换不同的差值检验. 然而, Hodges-Lehmann 估计可以实现这个功能. 幸好wilcox.test有这个功能. 只要使参数conf.int=TRUE 即可. 当可以计算精确p-值时, 使用Bauer (1972)的算法, 和 Hodges-Lehmann estimator. 若无精确p-值, 则使用正态逼近.

wilcox.exact in 'exactRankTests' 可以在有结的情况下计算精确的p值. coin 包里的函数 wilcox_test, 可以计算精确的p值.

假设数据来自两个随机样本, $x_1, x_2, \cdots, x_n$来自样本1, $y_1, y_2, \cdots, y_m$来自样本2. 给这N=m+n个观测从小到大排序并赋秩$(1, 2, \cdots, N)$. 若样本的分布一样, 那么其秩和相等. 检验就基于此原理.

令$F(x)$为X的分布函数, $G(x)$为Y的分布函数, 零假设为: $F(x) = G(x)$. 备择假设为: $F(x) \neq G(x)$. 实际上备择假设可以转换为$P(x > y) \neq P(x < y)$, 这样有利于计算.

此函数会自动把数值转换为秩次, 故不需要自己计算秩次,

只需提供原始数据(及分组信息, 见例子).

当只有x或x, y都给出且paired =TRUE(匹配样本), 为符号秩检验. 比较x,y的均值. 零假设为x或x, y对的分布关于mu对称.

当xy都给出且paired = FALSE(独立样本)为秩和检验. 比较x,y的均值. 零假设为xy之差为mu.

默认下小于50样本量, 且exact未指定, 使用精确p值, 否则使用正态近似.

下面是一个例子[1]. 选择12名三年级的同学, 其中4名上过幼儿园的考试成绩排序分别为2,5,6,9. 要检验的零假设是: 三年级学生的学习表现不取决于是否上过幼儿园. 备择假设: 学习表现与是否上过幼儿园不独立.

模型假设12个孩子是三年级学生的一组随机样本, 并根据学习成绩从好到差排序标记. "不独立"是指上过幼儿园的整体表现比没有上过的好, 或不好. 那么假设可以重新描述为, 零假设: 上过幼儿园的4个孩子的秩是秩1-12的一个随机样本. 备择假设: 上过幼儿园的4个孩子的秩整体比12个孩子中随机抽取4个孩子的秩要大或小.

我们选择检验统计量T是上过幼儿园的秩的和. 若T很大或很小, 则拒绝零假设. 故该检验是双边的. 每一个可能的结果是从1-12中抽取4个数, 对应上过幼儿园的4个孩子的秩. 样本空间是$\binom{12}{4} = 495$. 下面是我们的计算. 最后结果是没有显著差异.

```
# x可以是任何的12个不同的排序过的数字. 例如 x=3*(1:12), sort(rnorm(12)) 均可
> x=1:12
> y=rep(0,12)
> y[c(2,5,6,9)]=1 # y是分组
> y
 [1] 0 1 0 0 1 1 0 0 1 0 0 0
> wilcox.test(x~factor(y))

        Wilcoxon rank sum test
```

[1]实用非参数统计(第三版). Page 71. 例 2.3.2

```
data:  x by factor(y)
W = 20, p-value = 0.5697
alternative hypothesis: true location shift is not equal to 0

# 也可以按照下面的方法组织数据
> x=1:12
> y=c(2,5,6,9)
> x=x[-y]
> x
[1]  1  3  4  7  8 10 11 12
> group=c(rep(0,8),rep(1,4))
> wilcox.test(c(x,y)~group)

        Wilcoxon rank sum test

data:  c(x, y) by group
W = 20, p-value = 0.5697
alternative hypothesis: true location shift is not equal to 0

# 手工计算
> choose(12,4) # 495种组合
[1] 495
> c=combn(12,4) # 从1-12选择4个数字的所有组合
> a=colSums(c) # 所有组合的和(秩和)
> length(a[a<=22]) # 所有和小于22的组合个数. 22=sum(2,5,6,9)
[1] 141
> 2*length(a[a<=22])/495 # 所有和小于22的双侧频率
[1] 0.569697

# 考试名次为1,2,3,4时的检验
> y1=rep(0,12)
> y1[1:4]=c(1,1,1,1)
> wilcox.test(x~factor(y1),conf.inf=T)

        Wilcoxon rank sum test

data:  x by factor(y1)
W = 32, p-value = 0.00404
alternative hypothesis: true location shift is not equal to 0

# 手工计算
```

```
> 1/495*2
[1] 0.004040404

# 计 算 差 值(位 移)的 置 信 区 间
> wilcox.test(x~factor(y),conf.int=T)

        Wilcoxon rank sum test

data:  x by factor(y)
W = 20, p-value = 0.5697
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -4  6
sample estimates:
difference in location
                     2
```

另 外 一 个 例 子 ([11], 生 物 统 计 学 基 础. Page 328. 例 9.15). 我 们 要 比 较10-19岁 不 同 遗 传 形 式(RP)的 视 敏 度. 设25人 显 性 病, 30人 有 伴 性 病. 这 些 人 好 的 眼 睛 的 最 好 修 正 视 敏 度 见 下 表.

视 敏 度	显 性	伴 性
20-20	5	1
20-25	9	5
20-30	6	4
20-40	3	4
20-50	2	8
20-60	0	5
20-70	0	2
20-80	0	1

```
> x=c(rep(2,5),rep(2.5,9),rep(3,6),rep(4,3),rep(5,2))
> x
 [1] 2.0 2.0 2.0 2.0 2.0 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 3.0 3.0 3.0 3.0 3.0
[20] 3.0 4.0 4.0 4.0 5.0 5.0
> y=c(rep(2,1),rep(2.5,5),rep(3,4),rep(4,4),rep(5,8),rep(6,5),rep(7,2),1)
> y
```

```
 [1] 2.0 2.5 2.5 2.5 2.5 2.5 3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0 5.0 5.0 5.0 5.0 5.0
[20] 5.0 5.0 5.0 6.0 6.0 6.0 6.0 6.0 7.0 7.0 1.0
> g=c(rep(0,25),rep(1,30))

# 实际上, 只要x与y的相对位置不变, 最后结果就不变.
# 例如 wilcox.test(c(x*10,y*10)~g) 结果是一样的.
> wilcox.test(c(x,y)~g)

        Wilcoxon rank sum test with continuity correction

data:  c(x, y) by g
W = 179, p-value = 0.0007813
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(x = c(2, 2, 2, 2, 2, 2.5, 2.5, 2.5, 2.5,  :
  无法精確計算带连结的p值

# 使用 coin 的 wilcox_test 函数计算精确p值. minitab的结果是0.0002
# 注意: corr = T 和 F 结果是一样的.
> library(coin)
> wilcox_test(c(x,y)~factor(g))

        Asymptotic Wilcoxon Mann-Whitney Rank Sum Test

data:  z by factor(g) (0, 1)
Z = -3.7975, p-value = 0.0001461
alternative hypothesis: true mu is not equal to 0
```

17.3 Kurskal-Wallis 检验

Kurskal-Wallis 检验是Wilcoxon方法在多于两个样本的时候的推广

Kurskal-Wallis 检验是多个独立样本的检验([14] Page 207). 对于有很多结的情况, 应当毫不犹豫的使用 Kurskal-Wallis 检验. 事

实上 Kurskal-Wallis 检验是用于列联表的一个非常好的检验. 对差异很敏感. Kurskal-Wallis 检验统计量合理的运用了卡方逼近. 对于两样本的情况, Kurskal-Wallis 检验和 Mann-Whitney检验是等价的.

普通参数方法称为"单因素方差分析", 或有时候称为单因素F检验. 违反正态假设可能对F有一些影响, 但是某些非正态分布的数据(例如有极值)F检验的功效会比Kurskal-Wallis 检验小很多. 相对于F检验, Kurskal-Wallis 检验的A.R.E.从来不会小于0.864, 若是正态分布, A.R.E.=0.955. 均匀分布=1.0, 双指数分布=1.5.

下面是一个例子([14] Page 209). 检验4个品种的玉米的产量是否不同. p-值很小, 说明不同. 然后可以使用两两比较(Mann-Whitney检验或Kurskal-Wallis 检验都可以)来检验哪两个品种不同.

```
> x1=scan()
1: 83 91 94 89 89 96 91 92 90
10:
Read 9 items
> x2=scan()
1: 91 90 81 83 84 83 88 91 89 84
11:
Read 10 items
> x3=scan()
1: 101 100 91 93 96 95 94
8:
Read 7 items
> x4=scan()
1: 78 82 81 77 79 81 80 81
9:
Read 8 items
> x=c(x1,x2,x3,x4)
> x
 [1]  83  91  94  89  89  96  91  92  90  91  90  81  83  84  83  88  91  89  84
[20] 101 100  91  93  96  95  94  78  82  81  77  79  81  80  81
> g=c(rep(1,9),rep(2,10),rep(3,7),rep(4,8))
> kruskal.test(x,g)
```

```
        Kruskal-Wallis rank sum test

data:  x and g
Kruskal-Wallis chi-squared = 25.6288, df = 3, p-value = 1.141e-05
```

17.4 等方差的检验

TODO: 算法参考[14] Page 217. 方差检验的方法很多. Conover, Johnson, Johnson (1981)对56个方差检验方法作了全面比较.

17.5 秩相关度量

Kruskal(1958)的一篇综述讨论了很多相关度量. 若x,y独立, 则一些相关度量有分布函数, 且不依赖于(x,y)的二维分布函数([14] Page 227).

函数 cor.test 可以使用三种方法, 只要指定参数 method = c("pearson", "kendall", "spearman") 其中的一种即可. 它会自动将数据转换为秩并自动对结校正.

17.5.1 Pearson关联系数

最常用的就是Pearson乘积矩关联系数([14] Page 226).

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{[\sum(x_i - \bar{x})^2(y_i - \bar{y})^2]^{1/2}}$$

17.5.2 Spearmanρ

Spearman(1904)给出了相关性度量. 计算方法如下([14] Page 227): 设数据为二维随机变量$(x_1, y_1), \cdots, (x_n, y_n)$. 分别对x,y排序,

分别取得它们的秩为$R(x_i), R(y_i)$. 即若x_i为最小的x值, $R(x_i) = 1$, x_i为次小的x值, $R(x_i) = 2$. 有结时, 赋予没有结时本应秩的平均值. Spearman 相关系数为:

$$\rho = \frac{\sum_{i=1}^{n} R(x_i)R(y_i) - n(\frac{n+1}{2})^2}{(\sum_{i=1}^{n} R(x_i)^2 - n(\frac{n+1}{2})^2)^{1/2}(\sum_{i=1}^{n} R(y_i)^2 - n(\frac{n+1}{2})^2)^{1/2}}$$

实际上是基于秩与平均秩的简单Pearson乘积矩关联系数. 若数据用秩代替, 则

$$\bar{R(x)} = \frac{n+1}{2}$$

对y也是一样的.

17.5.3 Kendall τ

Kendall([14] Page 230)(1938)提出的. 设数据为二维随机变量$(x_1, y_1), \cdots, (x_n, y_n)$. 若一个观测的两个元素比另外一个观测的两个元素都大, 或都小, 称为是协调的(concordant), 例如(1.3,2.2)和(1.6,2.7)是协调的. 若一个观测的两个元素比另外一个观测的两个元素大小相反, 称为不协调的(discordant), 例如(1.3,2.2)和(1.6,1.1). 记N_c为协调的观测对数, N_d为不协调的观测对数. 由于n个观测可能有$\binom{n}{2} = n(n-1)/2$种不同方式的配对, N_c, N_d与带结的对数之和将等于$\binom{n}{2} = n(n-1)/2$. Kendall提出没有结的相关性度量为

$$\tau = \frac{N_c - N_d}{n(n-1)/2}$$

若所有对是协调的, $\tau = 1$, 若所有对是不协调的, $\tau = -1$. 如果有结, 修正为

$$\tau = \frac{N_c - N_d}{N_c + N_d}$$

17.5.4 Daniels趋势性检验

Daniels(1950)提出用Spearman ρ 作为趋势性检验([14] Page 234).

17.5.5 Jonckheere-Terpstra 检验

Spearman ρ 或 Kendall τ可以用于几个独立样本的零假设: 所有样本来自同一分布, 即

$$H_0 : F_1(x) = \cdots = F_k(x)$$

备择假设: 分布是在有序的某个方向上

$$H_1 : F_1(x) >= F_2(x) >= \cdots >= F_k(x)$$

至少有一个不等式成立.

注意: 此数据集与 Kruskal-Wallis 检验相同. 但是 Kruskal-Wallis 检验对任何差异敏感. 而 Spearman ρ 或 Kendall τ仅对H_1中的特殊有序敏感.

17.5.6 Kendall偏相关系数

TODO:

17.5.7 几个例子

直接使用 cor(), 默认方法为 pearson 方法, 适用于连续数据.

下面是一个强相关的例子

```
> n <- 100
> x <- runif(n)
> b <- rep(NA,n)
> b[1] <- 0
```

```
> for (i in 2:n) {
+   b[i] <- b[i-1] + .1*rnorm(1)
+ }
> y <- 1-2*x+b[1:n]
> plot(x,y) # 绘图查看
> cor(x,y)
[1] -0.8217834
> cor.test(x,y) # p 值很小, 说明相关系数显著不等于 0

        Pearson's product-moment correlation

data:  x and y
t = -14.2774, df = 98, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8766919 -0.7457370
sample estimates:
      cor
-0.8217834
```

spearman相关: 如果配对的数据不是连续的或不满足正态分布, 则可以视数据为秩次值. 由

$$S_x^2 = S_y^2 = n(n+1)/12$$

$$S_{xy} = n(n+1)/12 - 6\sum d^2/12(n-1)$$

其中n为观测样本量. 可以得到

$$r_s = 1 - \frac{6\sum d^2}{n(n^2-1)}$$

对r的显著性检验, 当$n > 10$时, 可以应用t检验. 若p值很小, 则r值显著不为0, 即x, y显著相关. 其中

```
t = r \sqrt{n-2} / \sqrt{1-r^2}  (df = n-2)
```

下面是一个例子. 两个医师对10张片子做评价, 打分结果(x,y)为病情的轻重. 判断两个医师的评价是否一致.

```
> cor(x,y)
[1] 0.6969697
> cor(x,y,method="spearman")
[1] 0.6969697
```

下面是自己编写的函数, 与cor结果一样.

```
> x=c(3,1,5,6,2,4,8,7,9,10)
> y=c(3,2,6,10,1,5,9,7,4,8)
> d=y-x
> d
 [1]  0  1  1  4 -1  1  1  0 -5 -2
> sum(d^2)
[1] 50
> r=1-6*sum(d^2)/(10*(10^2-1))
> r
[1] 0.6969697
> t=r*sqrt(8)/sqrt(1-r^2)
> t
[1] 2.749026
> pt(t,df=8)
[1] 0.9874517
> 1-pt(t,df=8) # p 值, 说明 r值显著不等于0. 即x y评价显著相关
[1] 0.01254834
```

17.6 多个相关样本

Milton Friedman 检验是符号检验的推广([14] Page 268). Quade 检验是符号秩检验的推广. Friedman检验使用更加广泛, 使用假定更少, 但是只有3个处理时, 功效不如符号秩检验, 4,5个处理时, 功效与Quade检验相当, 6个以上时, 功效比较大.

17.6.1 Friedman 检验

试验通常为随机化的完全区组设计. 对应的参数方法叫做双因素方差分析. 秩方法依赖于每组观测的秩, 其方法的发明者是一个著名的经济学家: Milton Friedman.

函数 friedman.test 按照统计量T_1计算([14] Page 270). T_1逼近自由度为k-1的卡方分布. 修正统计量与T_1的关系为

$$T_2 = \frac{(b-1)T_1}{b(k-1) - T_1} \sim F[k-1, (b-1)(k-1)]$$

其中b为区组数, k为处理数. 最近的研究表明, T_2有更好的逼近分布.

TODO: 若拒绝零假设, 多重比较见[14] Page 270.

下面是一个例子. 随机的12名业主在自己的院子的等面积的土地上分别种植4种不同的草, 一段时间后按喜好程度排名, 最喜欢的为4, 最不喜欢的为1. 最后想看看是否哪种草更加受欢迎. 我们最后给出修正的统计量. 两个统计量的p-值差不多.

```
x<-matrix(c(4,3,2,1,
   4,2,3,1,
   3,1.5,1.5,4,
   3,1,2,4,
   4,2,1,3,
   2,2,2,4,
   1,3,2,4,
   2,4,1,3,
   3.5,1,2,3.5,
   4,1,3,2,
   4,2,3,1,
   3.5,1,2,3.5),
   nc=4,byrow=T,dimnames=list(1:12,c("a","b","c","d")))
> x
    a   b   c   d
1  4.0 3.0 2.0 1.0
2  4.0 2.0 3.0 1.0
```

```
3  3.0 1.5 1.5 4.0
4  3.0 1.0 2.0 4.0
5  4.0 2.0 1.0 3.0
6  2.0 2.0 2.0 4.0
7  1.0 3.0 2.0 4.0
8  2.0 4.0 1.0 3.0
9  3.5 1.0 2.0 3.5
10 4.0 1.0 3.0 2.0
11 4.0 2.0 3.0 1.0
12 3.5 1.0 2.0 3.5

> friedman.test(x)

        Friedman rank sum test

data:  x
Friedman chi-squared = 8.0973, df = 3, p-value = 0.04404

# 修正的统计量
T2<-function(T1,b,k){
   T2<- (b-1)*T1/(b*(k-1)-T1)
   names(T2)<-"Correct Friedman F"
   p<-1-pf(T2,b-1,(b-1)*(k-1))
   names(p)<-"p value"
   res<-list(statistic=T2,p.value=p)
   res}

> T2(friedman.test(x)$statistic,dim(x)[1],dim(x)[2])
$statistic
Correct Friedman F
         3.192198

$p.value
   p value
0.004782398
```

17.6.2 Quade检验

Quade检验([14] Page 272).建立在每一区组原始观测值极差的基础上.

下面是一个例子. 5种品牌的洗衣粉在7个商店排开, 一周后, 计算销售数量, 看看是否品牌之间的销售有差异.

```
x<-matrix(c(5,4,7,10,12,
   1,3,1,0,2,
   16,12,22,22,35,
   5,4,3,5,4,
   10,9,7,13,10,
   19,18,28,37,58,
   10,7,6,8,7),
   nc=5,byrow=T,
   dimnames=list(1:7,c("A","B","C","D","E")))
> x
   A  B  C  D  E
1  5  4  7 10 12
2  1  3  1  0  2
3 16 12 22 22 35
4  5  4  3  5  4
5 10  9  7 13 10
6 19 18 28 37 58
7 10  7  6  8  7
> quade.test(x)
       Quade test

data:  x
Quade F = 3.8293, num df = 4, denom df = 24, p-value = 0.01519
```

TODO: 若拒绝零假设, 多重比较见[14] Page 273.

17.6.3 Friedman检验与Kendall系数及Spearman系数的关系

TODO: 参考[14] Page 279.

17.6.4 交互作用

对于交互作用, 没有什么好的非参数方法[14] Page 281.

17.7 平衡的不完全区组设计

完全区组设计中, 每个区组应用所有的处理. 当区组大小有限, 处理又比较多的时候, 很难做到. 每个区组就使用所有处理中的一部分, 叫做不完全区组. 平衡指满足下面条件的设计:(1) 每个区组有k个试验单元, (2) 每个处理出现在r个组中, (3) 每个处理出现的次数相同.

参数方法处理不完全区组设计基本是基于正态假设的. Durbin(1951)提出了一个秩检验可以检验平衡的不完全区组设计的零假设: 不同的处理之间没有显著差异. 若处理数和每个区组的单元数一样, Durbin检验可以转化为 Friedman 检验.

对于不完全区组设计的分析可以首先在每个区组将数据转化为秩, 然后应用软件中相应的程序, 例如SAS中的用于秩的不完全区组设计程序, 或广义线性模型.

下面是 Durbin 检验的算法([14] Page 284). 我们记

- t=处理数
- k=每个区组的单元数 k¡t
- b=区组数
- r=每个处理出现的次数

- λ=同时出现第i处理和第j处理的区组数, 这里要求对每一对处理的λ相同
- x_{ij}表示区组i处理j的结果
- $R(x_{ij})$为每个区组赋秩
- $R_j = \sum_{i=1}^{b} R(x_{ij})$为第j个处理下的r个观测值的秩和. 若某些观测的秩相等, 推荐使用平均秩.

检验统计量为

$$T_1 = \frac{12(t-1)}{rt(k-1)(k+1)} \sum_{j=1}^{t} (R_j - \frac{r(k+1)}{2})^2$$

如果存在结, 则使用平均秩的方法. 记A为秩与平均秩的平方和

$$A = \sum_{i=1}^{b} \sum_{j=1}^{t} [R(x_{ij})]^2$$

同时计算校正因子

$$C = \frac{bk(k+1)^2}{4}$$

调整后的统计量为

$$T_1 = \frac{(t-1)\sum_{j=1}^{t}(R_j - \frac{r(k+1)}{2})^2}{A-C} = \frac{(t-1)[\sum_{j=1}^{t} R_j^2 - rC]}{A-C}$$

另外一个等价的方法是在秩与平均秩上使用通常的方差分析方法, 它仅是T_1的一个函数. 但是近年来的研究表明它更精确一些, 因此人们更愿意使用

$$T_2 = \frac{T_1/(t-1)}{(b(k-1)-T_1)/(bk-b-t+1)}$$

零分布: T_1逼近服从自由度为t-1的卡方分布. T_2逼近自由度为 (t-1,bk-b-t+1)的F分布.

零假设为: 每个区组中, 所有的赋秩都是等可能的, 即处理效应相同. 备择假设: 至少一个处理的效应表现的比某个其它处理不同.

多重比较: 若拒绝零假设, 则使用下面方法进行多重比较. 下式成立则认为处理i,j不同.

$$|R_i - R_j| > t_{1-\alpha/2}[\frac{(A-C)2r}{bk-b-t+1}(1-\frac{T-1}{b(k-1)}]^{1/2}$$

其中t分布的自由度为bk-b-t+1

下面是一个例子([14] Page 286). 7种冰激凌, 请7个人品尝打分. 每个人品尝3种. 并用1,2,3打分(赋秩)表明喜欢程度. 使用 Youden 方阵编排.

```
x<-matrix(c(2,3,0,1,0,0,0,
   0,3,1,0,2,0,0,
   0,0,2,1,0,3,0,
   0,0,0,1,2,0,3,
   3,0,0,0,1,2,0,
   0,3,0,0,0,1,2,
   3,0,1,0,0,0,2),
   nr=7,byrow=T, dimnames=list("人"=LETTERS[1:7],"冰 激 凌"=letters[1:7]))
> x
  冰 激 凌
人 a b c d e f g
  A 2 3 0 1 0 0 0
  B 0 3 1 0 2 0 0
  C 0 0 2 1 0 3 0
  D 0 0 0 1 2 0 3
  E 3 0 0 0 1 2 0
  F 0 3 0 0 0 1 2
  G 3 0 1 0 0 0 2
# 将0转 换 为 缺 失 数 据
> y[y==0]<-NA
> y
```

冰 激 凌
人 a b c d e f g
A 2 3 NA 1 NA NA NA
B NA 3 1 NA 2 NA NA
C NA NA 2 1 NA 3 NA
D NA NA NA 1 2 NA 3
E 3 NA NA NA 1 2 NA
F NA 3 NA NA NA 1 2
G 3 NA 1 NA NA NA 2

其 中t=7,k=3,b=7,r=3,$\lambda = 1$.

```
durbin.test<-function(x){
   Rj=colSums(x,na.rm=T) # 列 秩 和
   d=dim(x) # 维 数
   t=d[2] # 冰 激 凌 种 类
   b=d[1] # 区 组 数
   r=length(x[,1][!is.na(x[1,])]) # 每 个 处 理 被 处 理 的 次
数, 冰 激 凌 被 品 尝 的 次 数
   k=length(x[1,][!is.na(x[1,])]) # 区 组 的 单 元 数
   T1=12*(t-1)*sum((Rj-r*(k+1)/2)^2)/(r*t*(k-1)*(k+1))
   T2=(T1/(t-1))/((b*(k-1)-T1)/(b*k-b-t+1))
   res=list(T1=T1,T2=T2)
   res
}

> durbin.test(y)
$T1
[1] 12

$T2
[1] 8
```

17.8 A.R.E. 不低于1的检验

本节描述的方法的渐近相对效率 A.R.E. 几乎总是大于1. 参数检验如果合适, 则 A.R.E. =1. 否则, A.R.E. 几乎总是大于1. 但是注意, A.R.E. 只是衡量检验的一个方法. 由于很难考虑所有的情况, 故通常使用 A.R.E. ([14] Page 290)

17.8.1 几个独立样本的 van der Waerden (正态得分)检验

van der Waerden (1952,1953)建议了一个简单的方法([14] Page 291). 即在计算中不是使用秩来替换数据, 而是用另外一些数据替换原始数据的秩, 例如近似正态分布的数据.

假设数据为k个随机样本组成. 每一个可能有不同的样本容量. 记第i个样本为 $x_{i1}, \cdots, x_{in_i}. i = 1, \cdots, k$. N 表示样本的总数. 给N个样本排序, 从1到N赋秩. 存在结时使用平均秩. 记 x_{ij} 的秩为 $R(x_{ij})$. 变换 $R(x_{ij})$ 为正态得分, 即标准正态分布的R/(N+1)分位数, 记作 $A_{ij} = z_{R(x_{ij})/(N+1)}$. k个样本每个的平均正态得分为

$$\bar{A}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} A_{ij}$$

方差为

$$S^2 = \frac{1}{N-1} \sum A_{ij}^2$$

检验统计量为

$$T_1 = \frac{1}{S^2} \sum_{i=1}^{k} n_i (\bar{A}_i)^2$$

零分布: 在分析了A的所有置换后, 可以得到T_1的精确分布, 但是很困难. 故经常使用自由度为k-1的卡方分布近似. 近似通常很好.

零假设为: 所有k个总体分布函数相同. 备择假设为: 至少一个总体比另外一个分布产生较大的观测值.

多重比较: 若拒绝了零假设, 那么, 如果下式成立, 则总体i,j不同.

$$|\bar{A}_i - \bar{A}_j| > t_{1-\alpha/2}(S^2\frac{N-1-T_1}{N-k})^{1/2}(\frac{1}{n_i}+\frac{1}{n_j})^{1/2}$$

其中t分布的自由度为N-k

下面是那个玉米的例子([14] Page 209, 293). 我们给出了正态得分. 剩下的计算统计量的任务交给读者完成吧.

```
> x1=scan()
1: 83 91 94 89 89 96 91 92 90
10:
Read 9 items
> x2=scan()
1: 91 90 81 83 84 83 88 91 89 84
11:
Read 10 items
> x3=scan()
1: 101 100 91 93 96 95 94
8:
Read 7 items
> x4=scan()
1: 78 82 81 77 79 81 80 81
9:
Read 8 items
> x=c(x1,x2,x3,x4)
> x
 [1]  83  91  94  89  89  96  91  92  90  91  90  81  83  84  83  88  91  89  84
[20] 101 100  91  93  96  95  94  78  82  81  77  79  81  80  81

# 分组
> g=c(rep(1,9),rep(2,10),rep(3,7),rep(4,8))
> summary(factor(g))
 1  2  3  4
 9 10  7  8
```

```
> rank(x) # 秩次
 [1] 11.0 23.0 28.5 17.0 17.0 31.5 23.0 26.0 19.5 23.0 19.5  6.5 11.0 13.5 11.0
[16] 15.0 23.0 17.0 13.5 34.0 33.0 23.0 27.0 31.5 30.0 28.5  2.0  9.0  6.5  1.0
[31]  3.0  6.5  4.0  6.5
> qnorm(rank(x)/(length(x)+1)) # 正态得分
 [1] -0.48373855  0.40467790  0.89380063 -0.03581663 -0.03581663  1.28155157
 [7]  0.40467790  0.65217899  0.14372923  0.40467790  0.14372923 -0.89380063
[13] -0.48373855 -0.29050677 -0.48373855 -0.18001237  0.40467790 -0.03581663
[19] -0.29050677  1.90221650  1.57921952  0.40467790  0.74355976  1.28155157
[25]  1.06757052  0.89380063 -1.57921952 -0.65217899 -0.89380063 -1.90221650
[31] -1.36762792 -0.89380063 -1.20404696 -0.89380063
```

17.8.2 等方差检验的正态得分法

Klotz(1962)介绍了应用正态得分进行两个样本的等方差检验的方法 ([14] Page 294). 计算检验统计量

$$T_3 = \frac{\sum_{i=1}^{n} A_i^2 - \frac{n}{N}\sum_{i=1}^{N} A_i^2}{(\frac{nm}{N(N-1)}[\sum_{i=1}^{N} A_i^4 - \frac{1}{N}(\sum_{i=1}^{N} A_i^2)^2])^{1/2}}$$

A_i表示正态得分, 样本容量为n,m. N=n+m. 若两个样本均值不同, 则应该先减去均值再赋秩计算正态得分. 还可以编程检验 Klotz 检验的精确p-值.

17.8.3 正态得分用于回归

将正态得分赋秩后, 按照回归算法计算.([14] Page 296)

17.8.4 正态得分与相关系数

换算正态得分后, 按照算法计算 Pearson 相关系数,([14] Page 296) 即为正态得分相关系数(而不是秩相关系数, 如 spearman相关系数)

17.8.5 随机正态离差

使用伪随机正态分布的数按照大小相应的替换秩. 这种方法感觉象蒙特卡洛方法. 每次, 每个人的替换都不同. 在现实的分析中很少使用, 但是它的 A.R.E. 与正态得分相同, 精确分布与参数检验相同, 所以人们从理论的角度很感兴趣.([14] Page 296)

17.8.6 寻找精确分布的方法

我们使用下面的例子介绍([14] Page 298). 例如 Mann-Whitney 检验中, 两个独立样本 $x_1, \cdots, x_n, y_1, \cdots, y_m$. 赋给$x_i$的秩是1到n+m中等可能取到的任何一个. 对于其它的值$x_1, \cdots, x_n, y_1, \cdots, y_m$也可以类推. 给x赋n个秩, 有$\binom{m+n}{n}$种可能, 每一种都是等可能的. 出现概率为$1/\binom{m+n}{n}$. 所以, 使用计数的方法可以得到统计量的零分布.

17.9 Fisher 随机化方法

用数据本身作为得分,是 Fisher(1935) 引出的, 结果检验就是传统的随机化方法([14] Page 300). 好像需要通过计数的方法来自己推导一下零假设的分布.

17.9.1 两个独立样本

两个独立样本 $x_1, \cdots, x_n, y_1, \cdots, y_m$. 统计量是

$$T_1 = \sum x_i$$

假设是

$$H_0 : E(x) = E(y)$$
$$H_1 : E(x) \neq E(y)$$

对于双边检验, 我们将x,y视为一个含有m+n个数据的数组. 取出n个样本, 则方法有$\binom{m+n}{n}$种. 为了找到p分位数w_p, 考虑第$\binom{m+n}{n}(p)$个最小的次序和, 即T_1, 其中最大的T_1就是w_p. 通过计算从m+n个数中选择n个数使和小于或等于(或大于等于, 若T_1在右边)T_1的方式的个数, 再除以$\binom{m+n}{n}$就是p-值. 双边检验乘以2.

下面是一个例子([14] Page 302). 假设随机样本为x=(0,1,1,0,-2) y=(6,7,7,4,-3,9,14). 考察两个期望是否一样. 从12个数里面选择5个的方式为$\binom{12}{5} = 792$. 显著性水平设为0.05, 双边. 那么(792)(0.025)=19.8 即寻找最小的20组统计量. x的和为0, 寻找小于等于0的和的组合数.

```
> a=c(0,1,1,0,-2,6,7,7,4,-3,9,14)
> k=combn(a,5) # 所有组合
> dim(k) # 组合数
[1]   5 792
> k[,1]
[1]  0  1  1  0 -2
> c=colSums(k) # 观测值的和
> length(c[c<=0]) # 0为x的和. 小于等于x和的组合个数有11个.
[1] 11
> 2*11/length(c) # 双边概率
[1] 0.02777778
> sort(c)[1:20] # 前20个最小的和
 [1] -4 -4 -3 -3 -1 -1  0  0  0  0  0  1  1  2  2  2  2  2  2  2
```

17.9.2 配对的随机化检验

([14] Page 303). 与符号检验的原理一样. 还可以参考二项分布中符号检验的例子.

Chapter 18

检验数据是否来自指定分布–Kolmogorov-Smirnov 型统计量

原理是考察样本数据的经验分布函数与假设的分布函数之间之间垂直距离最大的差值作为统计量. 其统计量的分布比较复杂. (参考文献 [14] Page 317. 由于统计量的分布此参考文献中也没有列出详细的统计量分布的推导过程, 但是给出了统计量的计算方法)

18.1 检验数据是否来自某个分布–Kolmogorov-Smirnov Test

注意: 有另外一个函数专门检验正态分布– Shapiro–Wilk test, 但是样本量必须在3-5000之间

若y类型为numeric, 检验为xy是否来自同一个连续分布.

若y为字符串, 且表示一个分布, 则零假设为x来自y定义的分布. 后面是y分布的参数

```
> x <- rnorm(50)
> y <- runif(30)
# x y 是否来自同一个分布?
> ks.test(x, y)

        Two-sample Kolmogorov-Smirnov test

data:  x and y
D = 0.48, p-value = 0.0002033
alternative hypothesis: two.sided

# x 是否来自 a shifted gamma 分布 with shape 3 and rate 2?
> ks.test(x+2, "pgamma", 3, 2) # two-sided, exact

        One-sample Kolmogorov-Smirnov test

data:  x + 2
D = 0.317, p-value = 5.742e-05
alternative hypothesis: two.sided

# x 是否来自正态分布
> x<-rnorm(100)
> ks.test(x, "pnorm", 0, 1)

        One-sample Kolmogorov-Smirnov test

data:  x
D = 0.0634, p-value = 0.816
alternative hypothesis: two.sided
```

18.2 正态性检验: Shapiro–Wilk test

检验样本是否来自正态分布. 样本量必须在3-5000之间

```
> shapiro.test(rnorm(100, mean = 5, sd = 3))

        Shapiro-Wilk normality test
```

```
data:  rnorm(100, mean = 5, sd = 3)
W = 0.9821, p-value = 0.1930

> shapiro.test(rnorm(10000, mean = 5, sd = 3))
错 误 在shapiro.test(rnorm(10000, mean = 5, sd = 3)) :
       样 本 大 小 必 需 在3和5000之 间
```

Chapter 19

TODO:非参数回归

Chapter 20

其它非参数检验

20.1 其它非参数检验

```
?kruskal.test  Kruskal-Wallis rank sum test.
?ansari.test
?mood.test
?fligner.test
library(help=ctest)
help.search('test')
```

20.2 方差齐性检验

两样本的方差齐性检验使用F检验. 多于两样本则使用bartlett.test. 2个非正态样本参考ansari.test 或 mood.test, 它们是非参数检验. 多于2个非正态样本参考 fligner.test

Part IV

回归与方差分析

“回归与方差分析”参考文献除了[11]，R部分主要参考《Statistics with R》，《Practical Regression and Anova using R》，其它《simpleR》《R语言简介》《R for beginners》等也有少量涉及。

Chapter 21

R的统计模型概述

这些统计模型在较复杂的分析, 特别是回归和方差分析中应用广泛, 但是在其它例如因子分析等也会使用这里的模型.

R基本的屏幕输出是简洁的，因此用户需要调用一些辅助函数来提取细节的结果信息。也就是说, 经常会联合使用多个函数来得到更全面详细的结果. 例如, 方差分析中一般会这样使用回归与方差分析函数lm()与anova(), 并进一步使用summary()函数来取得其详细的输出.

```
anova(lm(data~group))
summary(anova(lm(data~group)))
```

21.1 公式

假定y, x, x0, x1, x2, . . . 是数值变量，X是一个矩阵，而A, B, C, . . . 是因子。下面的例子中，左边给出公式，右边给出该公式的统计模型的描述. 下面是一些例子

- `y~x`
  `y~1+x`

二者都反映了y 对x 的简单线性模型。第一个公式包含了一个隐式的截距项，而第二个则是一个显式的截距项。

- `y ~ 0+x`
  `y ~ -1 + x`
  `y ~ x-1`

  y 对x 过原点的简单线性模型(也就是说，没有截距项)。

- `log(y) ~ x1 + x2`

  y 的变换形式log(y) 对x1 和x2 进行的多重回归(有一个隐式的截距项)。

- `y ~ poly(x,2)`
  `y ~ 1 + x + I(x^2)`

  y 对x 的二次多项式回归。第一种是正交多项式（orthogonal polynomial），第二种则显式地注明各项的幂次。

- `y~A`

  y 的单因素方差分析模型，类别由A 决定。

- `y~A+x`

  y 的单因素协方差分析模型，类别由A 决定，协方差项为x。

- `y ~ A*B`
  `y ~ A + B + A:B`
  `y ~ B %in% A`
  `y ~ A|B`

  y 对A 和B 的非可加两因子方差分析模型（two factor non-additive model）。前两个公式表示相同的交叉分类设计（crossed classification），后两个公式表示相同的嵌套分类设计(nested classification)。抽象一点说，这四个公式指明同一个模型子空间。

- `y ~ (A + B + C)^2`
  `y ~ A*B*C - A:B:C`

  三因子实验。该模型包括一个主效应（main efects）和两个因子的交互效应（interactions）。这两个公式等价。

- `y~A*x`
  `y ~ A|x`
  `y ~ A|(1 + x) - 1`

  在A的各个水平独立拟合y 对x 的简单线性回归。三个公式的编码不一样。最后一个公式会对A 各个水平分别估计截距项和斜率项的。

- `y ~ A*B + Error(C)`

  一个实验设计有两个处理因素A 和 B 以及因子 C 决定的误差分层（error strata）。如在裂区实验设计（split plot experiment）中，所有区组（还包括子区组）都由因子 C 决定的。

21.2 符号总结

- `a+b`

  a和b的相加效应

- `a-b`

  包括 a 但排除 b 项。

- `X`

  如果X 是一个矩阵, 这将反映各列的相加效应, 即 $X[,1]+X[,2]+...+X[,ncol(X)]$; 还可以通过索引向量选择特定列进行分析(如, $X[,2:4]$)

- `a:b`

  a 和b 的交互效应. a b 的张量积（tensor product）。如果两项都是因子，那么将产生“子类”因子(subclasses factor, 即因子交互作用)。

- `a*b`
  (等价于`a+b+a:b`)

  相加和交互效应

- `poly(a, n)`

  a的n价多项式

- `^n`
  `(a+b+c)^2` 等价于 `a+b+c+a:b+a:c+b:c`

  a+b 包含所有的直到n阶的交互作用

- `b %in% a`
  `a+a:b`
  `a|b`

  b 和 a 的嵌套分类设计

- `(a+b+c)^2-a:b`
  `a+b+c+a:c+b:c`

  去掉因子b的影响, 如:

- `y~x-1`
  `y~x+0`
  `0+y~x`

  表示通过原点的线性回归(等价于)

- `y~1`

  拟合一个没有因子影响的模型(仅仅是截距)

- `offset(...)`

  向模型中增加一个影响因子但不估计任何参数(如, offset(3*x))

注意，在常常用来封装函数参数的括弧中的操作符按普通的四则运算法则解释。I() 是一个恒等函数（identity function），它使得常规的算术运算符可以用在模型公式中。为了可以在公式中使用常规的运算符,

例如

```
y~x1+x2
```

表示模型

```
y = \beta_1 x1 + \beta_2 x2 + \alpha
```

而不是

```
y = \beta(x1 +  x2) + \alpha
```

公式

```
y~I(x1+x2)
```

就表示

```
y = \beta(x1 +  x2) + \alpha
```

还要特别注意模型公式仅仅指定了模型矩阵的列项，暗含了对参数项的指定。在某些情况下可能不是这样，如非线性模型的参数指定。

尽管细节是复杂的，R里面的模型公式在要求不是太离谱的情况下可以产生统计专家所期望的各种模型。提供模型公式的各种扩展特性是让R更灵活。例如，利用关联项而非主要效应的模型拟合常常会产生令人惊讶的结果，不过这些仅仅为统计专家们设计的。

21.3 AIC(赤池信息量)准则

AIC准则即赤池信息量准则(Akaike' information criterion , AIC) ,是由赤池弘次(H. Akaike) 在研究信息论特别是在解决时间序列定阶问题时提出来的。这是一个在统计分析特别是在统计模型的选择中有着广泛应用的准则。其显著特点之一是“吝啬原理 (principle of parsimony) ”的具体化。

定义为

AIC = - 2ln(模型的极大似然函数) + 2(模型的独立参数个数)

赤池建议,当欲从一组可供选择的模型中选择一个最佳模型时,AIC为最小的模型是最佳的。当两个模型之间存在着相当大的差异时,这个差异在右边第一项得到表现;而当两个模型间的差异几乎没有时,则第二项起作用,从而参数个数小的模型是好的模型.

Chapter 22

开始之前

参考《Practical Regression and Anova using R》

描述问题, 往往比解决更重要. (The formulation of a problem is often more essential than its solution which may be merely a matter of mathematical or experimental skill. —-Albert Einstein)

搜集数据. 理解数据如何搜集的非常重要.

初步分析. 非常重要. 看看数据大概的样子, 是否偏斜, 是否有错误数据, 异常值等等

最后才是回归或方差分析.

22.1 数据转换

开始分析之前, 需要使数据看起来为正态分布. 重要的是对称且没有极端值. 几乎所有情况下, 转换的同时也去除了残差问题, 例如方差不齐. 无一定的转换模式, 通常使用手工转换. 请参考 chapter 8 数据变换.

22.2 决策树

先看看决策树可能对回归有帮助. S-plus 是 tree, 但是 R 中推荐 rpart. 决策树似乎需要 factor(is.na(x1)) 对 x2.

```
library(rpart)
n <- 100
x1 <- rlnorm(n)
x2 <- rlnorm(n)
> r <- rpart(x1~x2)
> r
n=83 (17 observations deleted due to missing)

node), split, n, deviance, yval
   * denotes terminal node

1) root 83 147.093300 1.3673040
   2) x2>=1.334109 20   4.386958 0.8184096 *
   3) x2< 1.334109 63 134.767700 1.5415560
   6) x2< 1.094479 53  94.693460 1.3232420
   12) x2>=0.9443127 8   1.695635 0.6366346 *
   13) x2< 0.9443127 45  88.555910 1.4453060
      26) x2< 0.8501714 38  47.850200 1.3111270
      52) x2< 0.3734032 14   4.773765 1.0510790 *
      53) x2>=0.3734032 24  41.577410 1.4628210
         106) x2>=0.5465227 16  21.178520 1.1821200 *
         107) x2< 0.5465227 8  16.616830 2.0242230 *
      27) x2>=0.8501714 7  36.307570 2.1737080 *
   7) x2>=1.094479 10  24.160210 2.6986210 *
> r <- rpart(factor(is.na(x1))~x2) # 看看 na 与其它值的分类
> r
n=92 (8 observations deleted due to missing)

node), split, n, loss, yval, (yprob)
   * denotes terminal node

1) root 92 9 FALSE (0.90217391 0.09782609) *
```

22.3 缺失数据

简单的去除缺失并不太好. 当一个数据缺失, 需要同时去除其它的数据, 意味着一行数据被去除.

若数据缺失位置是随机的, 可以用平均值或中位数来代替.

但是很多时候, 缺失数据依情况而定. 例如, 收入调查往往缺失高收入的情况.

```
n <- 100
v <- .1
x1 <- rlnorm(n)
x2 <- rlnorm(n)
x3 <- rlnorm(n)
x4 <- x1 + x2 + x3 + v*rlnorm(n)
remove.higher.values <- function (x) {
n <- length(x)
ifelse( rbinom(n,1,(x-min(x))/(max(x)+1))==1 , NA, x)
}
x1 <- remove.higher.values(x1)
x2 <- remove.higher.values(x2)
x3 <- remove.higher.values(x3)
x4 <- remove.higher.values(x4)
m2 <- cbind(x1,x2,x3,x4)
pairs(m2, main="A few missing values")
```

22.4 极端值(outliers)

可以手工去除, 或把它们当做缺失值处理.

异常值:

x >上百分位数+1.5×（上百分位数-下百分位数）

x <下百分位数-1.5×（上百分位数-下百分位数）

极端异常值:

x >上百分位数+3×（上百分位数-下百分位数）
x <下百分位数-3×（上百分位数-下百分位数）

图来查看: boxplot, histogram, density, qqnorm 等

22.5 非正态的残差

如果残差非正态, 最小平方估计就不是最优的. 其它一些鲁棒的方法可能更好, 虽然可能是有偏的. 最坏的情况下, 所有的结果都是错的, 包括检验, 方差, 区间等.

但是, 如果残差分布比正态紧密, 或样本量非常大, 则可以忽略这个问题.

可以使用 shapiro.test() 检验残差的正态性. 参考参考第 18 章: Kolmogorov-Smirnov 型统计量

可以使用 histograms, box-and-whiskers plots (boxplots), qqplot(quantile-quantile plots) 来查看残差.

```
> x <- runif(100)
> y <- 1 - 2*x + .3*exp(rnorm(100)-1) # 产生非均匀的数据
> r <- lm(y~x)
# 绘图
> boxplot(r$residuals, horizontal=T)
> hist(r$residuals, breaks=20, probability=T, col='light blue')
> lines(density(r$residuals), col='red', lwd=3)
> qqnorm(r$residuals) # normal qq plot
> qqline(r$residuals,col='red')
```

22.6 异质性噪声

noise(噪声) 随x不同而不同. 最简单的方法是对数据做变换. 可能的话, 寻找一个转换即可以使残差正态, 又可以使噪声同质.

广义最小平方法允许对异质性噪声的数据做回归, 但是需要知道噪声的变化情况.

```
> x <- runif(100)
> y <- 1 - 2*x + .3*x*rnorm(100)
> r <- lm(y~x)
> n <- 10000
> xp <- sort(runif(n,))
> yp <- predict(r, data.frame(x=xp), interval="prediction")
> yr <- 1 - 2*xp + .3*xp*rnorm(n)
> plot(c(xp,x), c(yp[,1],y), pch='.') # 同时画出点和回归线
> lines(yp[,1]~xp) # 此线与上面的线是一个
> abline(r, col='red') # 此线与上面的线也是一个
> lines(xp, yp[,2], col='blue') # 下侧的区间线
> lines(xp, yp[,3], col='blue') # 上侧的区间线
> points(yr~xp, pch='.') # 散点
> points(y~x, col='orange', pch=16) # 散点
> points(y~x) # 同上的散点
> yp[1:10,] # 对 lm 的预测结果包括预测值, 上侧,下侧值
       fit      lwr     upr
1  0.9775679 0.6200168 1.335119
2  0.9772831 0.6197349 1.334831
3  0.9771233 0.6195768 1.334670
4  0.9765344 0.6189940 1.334075
5  0.9765169 0.6189766 1.334057
6  0.9762817 0.6187438 1.333820
7  0.9761578 0.6186212 1.333694
8  0.9761335 0.6185972 1.333670
9  0.9757138 0.6181818 1.333246
10 0.9755262 0.6179961 1.333056
```

22.7 相关数据

22.7.1 例子

下面是一个相关数据的例子

```
> n <- 100
> x <- runif(n)
> b <- rep(NA,n)
> b[1] <- 0
> for (i in 2:n) {
+   b[i] <- b[i-1] + .1*rnorm(1)
+ } # b 自身相关
> y <- 1-2*x+b[1:n]
> cor(x,y) # xy的相关系数
[1] -0.847911
> plot(x,y) # 绘图查看
> r <- lm(y~x)
> r

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)          x
   1.241      -1.790

> plot(r$res) # 查看残差是相关的
> cor.test(r$res[1:(n-1)], r$res[2:n]) # 检验残差的相关性

        Pearson's product-moment correlation

data:  r$res[1:(n - 1)] and r$res[2:n]
t = 26.2987, df = 97, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.9066943 0.9569760
sample estimates:
   cor
```

```
0.936483
```

下面是一个不相关的例子

```
> n <- 100
> x <- runif(n)
> b <- .1*rnorm(n+1) # b 自身不相关
> y <- 1-2*x+b[1:n] # xy不相关
> r <- lm(y~x)$res
> cor.test(r[1:(n-1)], r[2:n])

        Pearson's product-moment correlation

data:  r[1:(n - 1)] and r[2:n]
t = 0.0748, df = 97, p-value = 0.9405
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.1901025  0.2046993
sample estimates:
       cor
0.007594353
```

在这种情况下, 可以使用 generalized least squares, 即 gls, 包含在 nlme 包里. 其中的 AR1 模型(自回归1模型)假设两个邻近的 error 是相关的.

$$e_{i+1} = r * e_i + f_i$$

其中 r 为 AR1 的系数, f_i 为独立变量.

```
> n <- 100
> x <- rnorm(n)
> e <- vector()
> e <- append(e, rnorm(1))
> for (i in 2:n) {
+   e <- append(e, .6 * e[i-1] + rnorm(1) )
+ } # e 为自相关的
```

```
> y <- 1 - 2*x + e
> i <- 1:n
> plot(y~x)

> library(nlme)
> g <- gls(y~x, correlation = corAR1(form= ~i))
# 绘图查看与 lm 的区别. 此处区别不大
> plot(y~x)
> abline(lm(y~x))
> abline(g, col='red')

> summary(g)
Generalized least squares fit by REML
Model: y ~ x
Data: NULL
   AIC     BIC    logLik
294.0995 304.4394 -143.0498

Correlation Structure: AR(1)
Formula: ~i
Parameter estimate(s):
   Phi
0.5901658

Coefficients:
             Value  Std.Error   t-value p-value
(Intercept)  0.7199012 0.24061996   2.99186  0.0035
x         -2.0964824 0.09715024 -21.57980  0.0000

Correlation:
(Intr)
x -0.003

Standardized residuals:
       Min         Q1        Med         Q3        Max
-2.754014254 -0.626080011 -0.001852298  0.780250206  1.672214722
```

22.7.2 多个线性相关

数据中可能有多个变量相关,例如 X3 = X1+X2. 可以把每个变量(X_k)同其它变量($X_i's$)做回归, 然后检测R^2: 如果比较大(大于0.1), 则 Xk 可以表达为其它变量的线性组合.

为解决此问题, 我们可以去除强相关的变量, 但是小心你的解释会有问题. 很多时候相关是由于变量多而样本量少造成的, 此时可以增加样本量. 另外可以借助其它分析方法, 例如 "ridge regression" or SVM

另外, chapter 38 section 38.5 主成分回归也可以有效解决多个变量相关的问题, chapter 典型相关分析对多变量的相关的分析也很好

```
> n <- 100
> x <- rnorm(n) # x 为随机的
> x1 <- x+rnorm(n) # x1 由 x 而来, 与 x 相关
> x2 <- x+rnorm(n) # x2 由 x 而来, 与 x 相关
> x3 <- rnorm(n) # x3 也为随机的
> y <- x+x3
# 下面是它们表现出的关系. 注意 x1, x2 都与 x 相关, x3 随机
> summary(lm(x1~x2+x3))$r.squared # x1 与 x2,x3 线性相关
[1] 0.1726021
> summary(lm(x2~x1+x3))$r.squared # x2 与 x1,x3 线性相关
[1] 0.1337601
> summary(lm(x3~x1+x2))$r.squared # x3 独立于 x1,x2
[1] 0.07304001
# 可以推测, y 与 x,x1,x2,x3都相关
> summary(lm(y~x1))$r.squared
[1] 0.3182037
> summary(lm(y~x3))$r.squared
[1] 0.6227284
> summary(lm(y~x2))$r.squared
[1] 0.3027603
```

我们还可以查看估计系数的相关矩阵. 下面可以看出, x1 与 x3 是相关的. 注意系数的相关矩阵与变量间的相关系数不同.

```
> n <- 100
> v <- .1
> x <- rnorm(n)
> x1 <- x + v*rnorm(n)
> x2 <- rnorm(n)
> x3 <- x + v*rnorm(n)
> y <- x1+x2-x3 + rnorm(n)
> summary(lm(y~x1+x2+x3), correlation=T)$correlation
            (Intercept)         x1         x2         x3
(Intercept)  1.00000000 -0.08185033 -0.06912212  0.0907740
x1          -0.08185033  1.00000000 -0.20227089 -0.9903370
x2          -0.06912212 -0.20227089  1.00000000  0.2171248
x3           0.09077400 -0.99033697  0.21712483  1.0000000
> cor(cbind(x1,x2,x3)) # 变量间的相关系数
           x1         x2         x3
x1  1.00000000 -0.09422705  0.9900126
x2 -0.09422705  1.00000000 -0.1237598
x3  0.99001258 -0.12375983  1.0000000
# 下面画一个图使用圆圈的大小来表示相关系数的大小. 注意 cex 参数的用法
> m <- summary(lm(y~x1+x2+x3), correlation=T)$correlation
> m
            (Intercept)         x1         x2         x3
(Intercept)  1.00000000 -0.08185033 -0.06912212  0.0907740
x1          -0.08185033  1.00000000 -0.20227089 -0.9903370
x2          -0.06912212 -0.20227089  1.00000000  0.2171248
x3           0.09077400 -0.99033697  0.21712483  1.0000000
> class(m)
[1] "matrix"
> plot(col(m), row(m), cex=10*abs(m),
+     xlim=,
+     ylim=c(0, dim(m)[1]+1),
+ )
> col(m)
   [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    1    2    3    4
[3,]    1    2    3    4
[4,]    1    2    3    4
> row(m)
   [,1] [,2] [,3] [,4]
```

```
[1,]    1    1    1    1
[2,]    2    2    2    2
[3,]    3    3    3    3
[4,]    4    4    4    4
```

22.8 多元数据操作

本部分来自参考文献[28]《Multilevel Modeling in R》的翻译.

主要使用的包为: base, nlme, multilevel

看看数据

```
> library(multilevel)
> data(package="multilevel")
> data(cohesion)
> cohesion
    UNIT PLATOON COH01 COH02 COH03 COH04 COH05
1  1044B     1ST     4     5     5     5     5
2  1044B     1ST     3    NA     5     5     5
3  1044B     1ST     2     3     3     3     3
4  1044B     2ND     3     4     3     4     4
5  1044B     2ND     4     4     3     4     4
6  1044B     2ND     3     3     2     2     1
7  1044C     1ST     3     3     3     3     3
8  1044C     1ST     3     1     4     3     4
9  1044C     2ND     3     3     3     3     3
10 1044C     2ND     2     2     2     3     2
11 1044C     2ND     1     1     1     3     3
```

22.8.1 数据整合(merge)

例如有另外一个变量是platoon的大小, 我们想合并到数据cohesion中, 使用merge()函数

```
> group.size<-data.frame(UNIT=c("1044B","1044B","1044C","1044C"),
   PLATOON=c("1ST","2ND","1ST","2ND"),PSIZE=c(3,3,2,3))
> group.size
  UNIT PLATOON PSIZE
1 1044B    1ST    3
2 1044B    2ND    3
3 1044C    1ST    2
4 1044C    2ND    3
# 合并依据"UNIT","PLATOON"
>  new.cohesion<-merge(cohesion,group.size,by=c("UNIT","PLATOON"))
>  new.cohesion
   UNIT PLATOON COH01 COH02 COH03 COH04 COH05 PSIZE
1  1044B    1ST    4    5    5    5    5    3
2  1044B    1ST    3   NA    5    5    5    3
3  1044B    1ST    2    3    3    3    3    3
4  1044B    2ND    3    4    3    4    4    3
5  1044B    2ND    4    4    3    4    4    3
6  1044B    2ND    3    3    2    2    1    3
7  1044C    1ST    3    3    3    3    3    2
8  1044C    1ST    3    1    4    3    4    2
9  1044C    2ND    3    3    3    3    3    3
10 1044C    2ND    2    2    2    3    2    3
11 1044C    2ND    1    1    1    3    3    3
```

22.8.2 合计(aggregate)

按照分组情况合计, 使用函数aggregate().

```
# 将第3,4列按照UNIT和PLATOON的分组情况统计平均值
TEMP<-aggregate(cohesion[,3:4],
   list(cohesion$UNIT,cohesion$PLATOON),mean)
> TEMP
  Group.1 Group.2   COH01    COH02
1   1044B     1ST 3.000000       NA
2   1044C     1ST 3.000000 2.000000
3   1044B     2ND 3.333333 3.666667
4   1044C     2ND 2.000000 2.000000
```

```
# 按照UNIT合计, 并去除na数据
> TEMP<-aggregate(cohesion[,3:4],list(cohesion$UNIT),mean,na.rm=T)
> TEMP
  Group.1    COH01 COH02
1   1044B 3.166667   3.8
2   1044C 2.400000   2.0
```

22.8.3 按照合计情况再合并

合并统计结果

```
> names(TEMP)<-c("UNIT","PLATOON","G.COH01","G.COH02")
> final.cohesion<-merge(new.cohesion,TEMP,
    by=c("UNIT","PLATOON"))
> final.cohesion
    UNIT PLATOON COH01 COH02 COH03 COH04 COH05 PSIZE  G.COH01  G.COH02
1  1044B     1ST     4     5     5     5     5     3 3.000000       NA
2  1044B     1ST     3    NA     5     5     5     3 3.000000       NA
3  1044B     1ST     2     3     3     3     3     3 3.000000       NA
4  1044B     2ND     3     4     3     4     4     3 3.333333 3.666667
5  1044B     2ND     4     4     3     4     4     3 3.333333 3.666667
6  1044B     2ND     3     3     2     2     1     3 3.333333 3.666667
7  1044C     1ST     3     3     3     3     3     2 3.000000 2.000000
8  1044C     1ST     3     1     4     3     4     2 3.000000 2.000000
9  1044C     2ND     3     3     3     3     3     3 2.000000 2.000000
10 1044C     2ND     2     2     2     3     2     3 2.000000 2.000000
11 1044C     2ND     1     1     1     3     3     3 2.000000 2.000000
```

22.8.4 查看有多少组(unique)

使用数据bhr2000, 详细解释见帮助.

```
> help(bhr2000)
```

```
> data(bhr2000,package="multilevel")#puts data in working environment
# GRP是分组情况
> names(bhr2000)
 [1] "GRP"   "AF06"  "AF07"  "AP12"  "AP17"  "AP33"  "AP34"  "AS14"  "AS15"
[10] "AS16"  "AS17"  "AS28"  "HRS"   "RELIG"
# 共有5400行数据
> nrow(bhr2000)
[1] 5400
# 看看有共多少组
> length(unique(bhr2000$GRP))
[1] 99
```

Chapter 23

一般线性回归(Linear regression)

23.1 数据

下面是某药物浓度(x)与对应的吸光度(y), (实际上不是线性关系, 但是这里我们用做例子)

```
# 某药物浓度
x=c(15.625,  31.250,  62.500, 125.000, 250.000, 500.000, 1000.000)
# 对应的吸光度
y=c(0.103, 0.217, 0.364, 0.678, 0.968, 1.501, 1.927)
```

23.2 模型描述

23.2.1 模型

$$y = b_0 + b_1 * x1$$

检验 b_0, b_1 是否等于 0.

23.2.2 总平方和=残差平方和+回归平方和

$$\sum_{i=1}^{n}(y_i-\bar{y})^2=\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2+\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$$

总平方和(TotalSS) = 残差平方和(ResSS)+回归平方和(RegSS), 即 TotalSS = ResSS + RegSS

实际计算 TotalSS, RegSS 与 ResSS

```
# 其中 res=lm(y~x)
TotalSS = sum( (y-mean(y))^2 ) # 总平方和
ResSS = sum( res$residuals^2 ) # 残差平方和
RegSS = TotalSS - ResSS # 回归平方和
```

```
> TotalSS
[1] 2.816866
> ResSS
[1] 0.2625031
> RegSS
[1] 2.554363
```

23.2.3 回归平均平方(RegMS)与残差平均平方(ResMS)及其自由度

回归平均平方(RegMS) 与 残差平均平方(ResMS)

RegMS = RegSS/模型中预测变量数(k), 在简单线性回归中, 由于 k=1, 所以 RegMS = RegSS.

ResMS = ResSS/(n-k-1), 在简单线性回归中, k=1. 我们称 n-k-1 为残差平方和的自由度, 记为 Res df. 残差平均平方(ResMS)有时在文献中也记为 $s^2_{y\cdot x}$

下面计算

```
RegMS=RegSS/1 # 回归平均平方, k=1, 结果为 2.554363
ResMS=ResSS/(7-1-1) # 残差平均平方, 结果为 0.05250063
```

23.3 使用R计算

23.3.1 回归函数lm()

直接的回归分析的结果比较少. 若需要进一步的结果, 通常需要对结果使用泛型函数, 例如 summary 等.

```
> res=lm(y~x)
> res

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)          x
   0.306489    0.001821

> names(res) # 可以使用的结果
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

23.3.2 进一步分析的泛型函数

下面的显示了一些可以对分析结果对象做一些补充分析的泛型函数, 主要参数一般都是分析结果对象, 但是有些情况下,如泛型函数如predict 或 update 需要一些额外的参数.

```
add1  coef     effects kappa  predict residuals
alias deviance family  labels print   step
anova drop1    formula plot   proj    summary
```

下面是常用的几个介绍

- add1 连续测试所有可以加入模型的元素项
- drop1 连续测试所有可以从模型中移除的元素项
- step 通过AIC (调用add1 和drop1)选择一个模型
- anova 计算一个或多个模型的方差/残差分析表
- predict 通过拟合的模型计算一个新的数据集的预测值
- update 用新的数据或者公式拟合一个模型
- deviance 残差平方和
- logLik 对数似然值
- AIC 赤池信息量
- vcov 返回主要参数的协方差矩阵

23.3.3 summary()函数-对回归结果的统计与检验

summary(lm(x y)) 与 summary(lm(y x)) 的 R^2 是一样的.

summary 里有相关系数 R^2, summary(lm(y x))$r.squared. str(lm()) 里没有.

t 值 表 示 系 数 是 否 显 著 不 等 于0的 概 率. 给 出 了 总 的F值. 若 想 知 道 每 个 系 数 的F值, 参 考anova(res¡-lm(y x))

下 面 是summary结 果 及 其 含 义. 各 个 值 的 计 算 见 后 面.

```
> summary(res)

Call:
lm(formula = y ~ x)

Residuals:
      1       2       3       4       5       6       7
-0.23193 -0.14638 -0.05627  0.14395  0.20638  0.28426 -0.20000

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.306488   0.113905   2.691 0.043260 *
x           0.001821   0.000261   6.975 0.000932 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2291 on 5 degrees of freedom # 残 差 的 标 准 误
Multiple R-squared: 0.9068,     Adjusted R-squared: 0.8882 # 相 关 系 数 的 平 方
F-statistic: 48.65 on 1 and 5 DF,  p-value: 0.0009318 # F值, 整 体 回 归 的 检 验
```

23.3.4 使 用anova检 测 系 数 显 著 性

anova 给 出 斜 率 是 否 显 著 不 等 于0的 概 率(F值 及 其p值). 结 果 与summary()函 数 的 结 果 一 样.

当 p 值 很 小 时, 斜 率 就 显 著 不 为 0. 即 y 依 赖 于 自 变 量x. 下 面 的 结 果 告 诉 我 们, y 依 赖 于 x, 冒 的 风 险 为 0.0009318.

```
> anova(res)
```

```
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
x          1 2.5544  2.5544  48.654 0.0009318 ***
Residuals  5 0.2625  0.0525
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

注意, 在多元回归中, 结果依赖于自变量的顺序. 顺序不同其p值是不同的. 有时候, 结果会相反.

23.3.5 回归系数的置信区间(CI)

help(lm) 并查询 interval. 函数为 confint()

```
> confint(res)
                2.5 %      97.5 %
(Intercept) 0.01368702 0.599289993
x           0.00114960 0.002491426
```

23.3.6 计算回归预测的y值及区间

```
# 参数 interval='confidence' 将得到3列值,后两列为上下区间
> predict(res,interval='confidence')
       fit       lwr      upr
1 0.3349340 0.04883083 0.6210372
2 0.3633795 0.08374200 0.6430171
3 0.4202706 0.15279957 0.6877416
4 0.5340526 0.28734357 0.7807617
5 0.7616168 0.53786603 0.9853675
6 1.2167450 0.95092253 1.4825675
7 2.1270015 1.59723497 2.6567680
```

23.4 检验

23.4.1 手工计算F值

与summary()函数的F值结果一样.

```
F=RegMS/ResMS # 结果 48.65395
```

与 summary 结果一样.

23.4.2 方差齐性的检验

两样本的方差齐性检验使用F检验. 多于两样本则使用bartlett.test. 2个非正态样本参考 ansari.test 或 mood.test, 它们是非参数检验. 多于2个非正态样本参考 fligner.test

23.4.3 回归系数的假设检验

我们要检验回归系数β

$$H_0 : \beta = 0$$
$$H_1 : \beta \neq 0$$

F检验和t检验都可以. F检验的自由度为 1 及 n-2. t检验等价于F检验, 而且还能提供斜率的区间估计, 故此方法广泛使用. F检验是在 H_0 成立下, $F = RegMS/ResMS \sim F$分布.

这里使用car包的linear.hypothesis函数检验回归假设. 可以检验任何指定系数, 或交互效应的系数的显著性, 可以指定使用卡方检验或F检验, 默认使用F检验.

(linear.hypothesis用法很多，也有很多等价形式，方便使用。详细见函数帮助)

```
> library(car)
> mod.davis <- lm(weight ~ repwt, data=Davis)
> linear.hypothesis(mod.davis, diag(2), c(0,1))
Linear hypothesis test

Hypothesis:
(Intercept) = 0
repwt = 1

Model 1: weight ~ repwt
Model 2: restricted model

  Res.Df   RSS  Df Sum of Sq    F Pr(>F)
1    181 12828
2    183 13074  -2     -246 1.74   0.18
```

最下面部分分别给出截距和repwt项的残差自由度, 残差平方和, 自由度, 平方和, F值, p值.

23.4.4 异残差检验(Breusch-Pagan test)–检验残差是否为常量

此检验常常称为Breusch-Pagan test. Cook and Weisberg (1983)也独立提出此检验.

```
> library(car)
> r=lm(interlocks~assets+sector+nation, data=Ornstein)
# 默认使用拟合值 ~ fitted.values检验
> ncv.test(r)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 46.98537    Df = 1     p = 7.151835e-12
# 可以指定残差方程, 此处为 ~ assets+sector+nation
> ncv.test(r,~ assets+sector+nation, data=Ornstein)
Non-constant Variance Score Test
Variance formula: ~ assets + sector + nation
```

```
Chisquare = 74.73535    Df = 13     p = 1.066320e-10
```

23.5 协方差

23.5.1 未修正的协方差

car包里的Var函数计算普通协方差矩阵, 如果参数为lm的结果, 则计算自变量因子(分组后)的协方差.

下面是帮助的例子.

```
> data(Davis)
> attach(Davis)
> Var(cbind(weight, repwt), na.rm=TRUE)
       weight repwt
weight  233.9 176.1
repwt   176.1 189.8
> var(cbind(weight, repwt), na.rm=TRUE)
       weight repwt
weight  233.9 176.1
repwt   176.1 189.8
```

23.5.2 修正的协方差

由于OLS方法的系数估计是无偏的, 但是对系数的协方差的估计不准确. hccm计算按照异方差修正后的协方差矩阵取代vcov函数. hccm函数方法有"'hc0"', "'hc1"', "'hc2"', "'hc3"', or "'hc4"', 其中第一个是白化修正:

$$V(b) = inv(X'X) * X' * diag(e^2) * X * inv(X'X)$$

其中e^2为残差, X为模型矩阵. 其它修正方法见帮助.

```
> options(digits=4)
> mod<-lm(interlocks~assets+nation, data=Ornstein)
# 未修正的协方差,
# 其中nationOTH  nationUK  nationUS为nation的三个水平
> Var(mod)
            (Intercept)    assets  nationOTH  nationUK  nationUS
(Intercept)  1.079e+00 -1.588e-05 -1.037e+00 -1.057e+00 -1.032e+00
assets      -1.588e-05  1.642e-09  1.155e-05  1.362e-05  1.109e-05
nationOTH   -1.037e+00  1.155e-05  7.019e+00  1.021e+00  1.003e+00
nationUK    -1.057e+00  1.362e-05  1.021e+00  7.405e+00  1.017e+00
nationUS    -1.032e+00  1.109e-05  1.003e+00  1.017e+00  2.128e+00
# 白化修正的协方差矩阵
> hccm(mod)
            (Intercept)    assets  nationOTH  nationUK  nationUS
(Intercept)  1.664e+00 -3.957e-05 -1.569e+00 -1.611e+00 -1.572e+00
assets      -3.957e-05  6.752e-09  2.275e-05  3.051e-05  2.231e-05
nationOTH   -1.569e+00  2.275e-05  8.209e+00  1.539e+00  1.520e+00
nationUK    -1.611e+00  3.051e-05  1.539e+00  4.476e+00  1.543e+00
nationUS    -1.572e+00  2.231e-05  1.520e+00  1.543e+00  1.946e+00
```

23.6 相关系数(回归系数)

$R^2 = RegSS/TotalSS$ 与 summary 中 Multiple R-Squared 一致. 校正的(Adjusted R-squared) 值的计算方法未知???

```
> R2=RegSS/TotalSS
> R2
[1] 0.7044103
```

R 即为相关系数. Multiple R-Squared 是多重相关, 即y与所有预测变量的回归函数$b_1x_1 + b_2x_2 + \cdots$之间相关系数.

R^2为拟合精确性的指标的汇总(即使用x预测y的精确程度). 注意到 TotalSS = ResSS + RegSS, 得到 $ResSS = TotalSS(1 - R^2)$.

则R^2可以看作y的方差被x的方差解释的比值. 换句话说, R^2可以看作y的变异被x的变异解释的比值. 也就是x能够预测y的精确程度.

$R^2 = 1$时, 所有点落在回归线上. y的变异全部被x解释. x能够精确预测y.

$R^2 = 0$时, y的方差与x无关. x不能提供y的任何信息.

23.6.1 相关系数(即回归系数)的单样本t检验

在数学上可以证明, 相关系数的单样本t检验等价于斜率的F检验和t检验, 即有相同的p-值.

检验假设

$$\rho = 0 \quad vs \quad \rho \neq 0$$

计算相关系数 r, 然后计算检验统计量

$$t = r\sqrt{n-2}/\sqrt{1-r^2}$$

零假设成立时, t服从n-2自由度的t分布. n为配对数.

下面是cor.test和手工计算的结果

```
> x=rnorm(10)
> y=rnorm(10)
> r<-cor(x,y)
[1] -0.2961736
> cor.test(x,y)

        Pearson's product-moment correlation

data:  x and y
t = -0.8771, df = 8, p-value = 0.406
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.7802920  0.4098881
```

```
sample estimates:
      cor
-0.2961736

> r*sqrt(8)/sqrt(1-r^2) # 手工计算检验统计量t
[1] -0.8770552

> summary(lm(y~x)) # t值相同, p-值也相同

Call:
lm(formula = y ~ x)

Residuals:
   Min     1Q  Median     3Q    Max
-1.3977 -0.5536  0.1954  0.7472  0.9695

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.08585    0.32396   0.265    0.798
x           -0.25379    0.28937  -0.877    0.406

Residual standard error: 0.9316 on 8 degrees of freedom
Multiple R-squared: 0.08772,    Adjusted R-squared: -0.02632
F-statistic: 0.7692 on 1 and 8 DF,  p-value: 0.406
```

23.6.2 相关系数的Fisher变换(Z变换)

有时候, 我们需要检验相关系数是否与一个不为0的值相等. 即检验假设

$$\rho = \rho_0 \quad vs \quad \rho \neq \rho_0$$

如果使用t检验, 那么零假设成立时在非0的ρ下有一个倾斜的分布. 不容易用正态分布近似. Fisher考虑到这个问题, 提出了一个变换使我们可以用正态分布去检验.

相关系数的 Fisher 变换为

$$z = \frac{1}{2} ln(\frac{1+r}{1-r})$$

在零假设成立时近似于正态分布. 均值为

$$z_0 = \frac{1}{2} ln(\frac{1+\rho_0}{1-\rho_0})$$

方差为

$$s^2 = \frac{1}{n-3}$$

r的绝对值很小时, 与z接近, 但是r绝对值大时, z与r相差很大. 故需要z变换.

23.6.3 相关系数差异的单样本z检验

计算相关系数r和其z变换, 然后计算检验统计量

$$\lambda = (z - z_0\sqrt{n-3} \sim N(0,1)$$

例如检验x,y的相关系数是否=-0.5

```
> cor(x,y)
[1] -0.2961736
> r=cor(x,y)
> z=0.5*log((1+r)/(1-r)) # Fisher 变换
> z
[1] -0.30532
> T=(z-(-0.5))*sqrt(7) # 检验统计量
> T
[1] 0.5150748
> 2*(1-pnorm(abs(T))) # p-值接受零假设, 相关系数与-0.5没有显著差异
[1] 0.6065007
> T=(z-0.5)*sqrt(7) # 检验是否等于0.5
> T
[1] -2.130676
> 2*(1-pnorm(abs(T))) # 拒绝等于0.5的假设
[1] 0.03311580
```

23.6.4 相关系数的区间估计

(1) 计算相关系数 r

(2) 然后计算r的Fisher变换$z = \frac{1}{2}ln(\frac{1+r}{1-r})$. ρ的Fisher变换$z_0 = \frac{1}{2}ln(\frac{1+\rho_0}{1-\rho_0})$. 对于$z_0$的双侧$100\% * (1-\alpha)$置信区间为

$$z_1 = z - z_{1-\alpha/2}/\sqrt{n-3}$$

$$z_2 = z + z_{1-\alpha/2}/\sqrt{n-3}$$

(3)ρ的双侧$100\% * (1-\alpha)$置信区间为

$$\rho_1 = \frac{e^{2z_1}-1}{e^{2z_1}+1}$$

$$\rho_2 = \frac{e^{2z_2}-1}{e^{2z_2}+1}$$

下面是一个例子

```
> x=rnorm(100)
> y=rnorm(100)
> cor(x,y)
[1] -0.0320158
> cor.test(x,y)

        Pearson's product-moment correlation

data:  x and y
t = -0.3171, df = 98, p-value = 0.7518
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2270064  0.1654427
sample estimates:
       cor
```

```
-0.0320158

> r=cor(x,y) # 相关系数
> z=0.5*log((1+r)/(1-r)) # r 的 Fisher 变换
> z
[1] -0.03202674
> z1=z-qnorm(0.975)/sqrt(97) # 上侧区间
> z1
[1] -0.2310309
> z2=z+qnorm(0.975)/sqrt(97) # 下侧区间
> z2
[1] 0.1669774
> rho1=(exp(2*z1)-1)/(exp(2*z1)+1) # 变换回r的区间
> rho1
[1] -0.2270064
> rho2=(exp(2*z2)-1)/(exp(2*z2)+1) # 变换回r的区间
> rho2
[1] 0.1654427
```

23.6.5 相关系数的功效及样本量估计

假设对指定的ρ_0检验

$$\rho = 0 \quad vs \quad \rho = \rho_0 > 0$$

单侧及显著性水平为α的检验, 在指定样本量n时

$$power = \Phi(z_0\sqrt{n-3} - z_{1-\alpha})$$

对于指定的$\rho = \rho_0$使用单侧及显著性水平为α, 功效为$1-\beta$的检验所需要的样本量为

$$n = [(z_{1-\alpha} + z_{1-\beta})^2/z_0^2] + 3$$

23.6.6 相关系数的两样本检验

有数据(x1,y1)相关系数为r1, (x2,y2)相关系数为r2. 两样本检验就是检验若 , r2 是否相等. 即

$$\rho_1 = \rho_2 \quad vs \quad \rho_1 \neq \rho_2$$

合理的做法是对相关系数做z变换得到z1, z2, 若变换后的差值$|z1 - z2|$大, 拒绝零假设. 检验统计量

$$T = \frac{z1 - z2}{\sqrt{\frac{1}{n_1-3} + \frac{1}{n_2-3}}} \sim N(0,1)$$

假设数据为正态数据.

23.7 一些图

更多见回归诊断

```
x1=1:100
y=1:100+rnorm(100)*10
res=lm(y~x1) # 回归

# 残差图
plot(res$residual)
hist(res$residual) # 残差是否正态分布
boxplot(res$resdual)
plot(res$res ~ res$fitted.values) # 拟合值与残差作图

# 预测及上下界
pre=predict(res, interval='confidence') # 预测的直线
# 绘图
plot(y~x1) # 绘图在一起
lines(pre[,1]) # abline(res) 也可以
lines(pre[,2], col='red') # 上界
lines(pre[,3], col='red') # 下界
```

Chapter 24

回归诊断

回归完成后应该立即做回归诊断, 看看回归效果如何. summary结果里包含了t检验.

下面是其它检验

回归诊断相关的函数

```
influence.measures rstandard rstudent dffits
cooks.distance    dfbeta   dfbetas  covratio
hatvalues        hat
```

24.1 图的威力

```
Anscombe <-data.frame(
  X=c(10.0, 8.0, 13.0, 9.0, 11.0, 14.0, 6.0, 4.0, 12.0, 7.0, 5.0),
  Y1=c(8.04,6.95, 7.58,8.81,8.33,9.96,7.24,4.26,10.84,4.82,5.68),
  Y2=c(9.14,8.14, 8.74,8.77,9.26,8.10,6.13,3.10, 9.13,7.26,4.74),
  Y3=c(7.46,6.77,12.74,7.11,7.81,8.84,6.08,5.39, 8.15,6.44,5.73),
  X4=c(rep(8,7), 19, rep(8,3)),
  Y4=c(6.58,5.76,7.71,8.84,8.47,7.04,5.25,12.50, 5.56,7.91,6.89)
)
```

```
 lm1=lm(Y1~X, data=Anscombe)
 lm2=lm(Y2~X, data=Anscombe)
 lm3=lm(Y3~X, data=Anscombe)
 lm4=lm(Y4~X4,data=Anscombe)

# 下面是系数部分的结果

# 拟合比较好
> summary(lm(Y1~X, data=Anscombe))

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0001     1.1247   2.667  0.02573 *
X             0.5001     0.1179   4.241  0.00217 **

# 实际上是曲线
> summary(lm(Y2~X, data=Anscombe))

            Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.001      1.125   2.667  0.02576 *
X              0.500      0.118   4.239  0.00218 **

# 有极端值的干扰
> summary(lm(Y3~X, data=Anscombe))

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0075     1.1244   2.675  0.02542 *
X             0.4994     0.1179   4.237  0.00218 **

# 分布很不均匀
> summary(lm(Y4~X4,data=Anscombe))

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0017     1.1239   2.671  0.02559 *
X4            0.4999     0.1178   4.243  0.00216 **

# 绘图查看
par(mfrow=(c(2,2)))
plot(Y1~X, data=Anscombe)
abline(lm(Y1~X, data=Anscombe))
plot(Y2~X, data=Anscombe)
```

```
abline(lm(Y2~X, data=Anscombe))
plot(Y3~X, data=Anscombe)
abline(lm(Y3~X, data=Anscombe))
plot(Y4~X4, data=Anscombe)
abline(lm(Y4~X4, data=Anscombe))
```

从图中看到, 只有第一组拟合是好的. 而单纯的回归结果都是显著的.

第二组可能是二次或更高次数的多项式.

```
> lm2.sol<-lm(Y2~X+I(X^2), data=Anscombe); summary(lm2.sol)

Call:
lm(formula = Y2 ~ X + I(X^2), data = Anscombe)

Residuals:
      Min        1Q    Median        3Q       Max
-0.0013287 -0.0011888 -0.0006294  0.0008741  0.0023776

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.9957343  0.0043299   -1385   <2e-16 ***
X           2.7808392  0.0010401    2674   <2e-16 ***
I(X^2)      -0.1267133  0.0000571   -2219   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘

Residual standard error: 0.001672 on 8 degrees of freedom
Multiple R-squared:    1,     Adjusted R-squared:    1
F-statistic: 7.378e+06 on 2 and 8 DF,  p-value: < 2.2e-16

> attach(Anscombe)
> plot(Y2~X)

# 下面绘制预测值曲线
> o <- order(X)
> Y2.pre <- predict(lm2.sol)
> Y2.pre.o<-Y2.pre[o]
```

```
> lines(X.o,Y2.pre.o,col="red")
```

第三组的异常值需要手工去除.

```
> i<-1:11; Y31<-Anscombe$Y3[i!=3]; X3<-Anscombe$X[i!=3]
>  lm3.sol<-lm(Y31~X3); summary(lm3.sol)

Call:
lm(formula = Y31 ~ X3)

Residuals:
      Min        1Q    Median        3Q       Max
-0.0060173 -0.0012121 -0.0010173 -0.0008225  0.0140693

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.0106277  0.0057115   702.2   <2e-16 ***
X3          0.3450433  0.0006262   551.0   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.006019 on 8 degrees of freedom
Multiple R-squared:     1,      Adjusted R-squared:     1
F-statistic: 3.036e+05 on 1 and 8 DF,  p-value: < 2.2e-16
```

最后数据没有足够的信息来判断. 它对单个的样本非常依赖, 这可能不是一个综合的分析.

24.2 残差及其检验

最小二乘法求回归模型时, 对残差的要求是独立等方差的.

24.2.1 简介 plot.lm()

用法为 which 为 1 是画普通残差与拟合值, 2 为正态 QQ 的残差图, 3 为标准化残差开方与拟合值的残差图, 4 为 Cook 统计量的残差图.

```
plot(x, which = c(1:3,5),
    caption = c("Residuals vs Fitted", "Normal Q-Q",
        "Scale-Location", "Cook's distance",
        "Residuals vs Leverage", "Cook's distance vs Leverage"),
    panel = if(add.smooth) panel.smooth else points,
    sub.caption = NULL, main = "",
    ask = prod(par("mfcol")) < length(which) && dev.interactive(),
    ...,
    id.n = 3, labels.id = names(residuals(x)), cex.id = 0.75,
    qqline = TRUE, cook.levels = c(0.5, 1.0),
    add.smooth = getOption("add.smooth"), label.pos = c(4,2),
    cex.caption = 1)
```

24.2.2 普通残差

设线性回归模型为

$$Y = Xb + \epsilon$$

Y 为 n 维向量, X 为 $n * (p+1)$ 阶设计矩阵, b 为 $p+1$ 向量. ϵ 为 n 维误差向量.

回归系数的估计为

$$\hat{b} = (X^T X)^{-1} X^T Y$$

拟合值

$$\hat{Y} = X\hat{b} = X(X^T X)^{-1} X^T Y = HY$$

其中

$$H = X(X^T X)^{-1} X^T$$

称 H 为帽子矩阵[1]

残差为

$$\hat{\epsilon} = Y - \hat{Y} = (I - H)Y$$

residuals() resid() 计算模型的残差. 然后我们可以对残差做正态检验

```
# 对第一组数据检验残差符和正态分布
> lm.1=lm(Y1~X, data=Anscombe)
> r1=resid(lm.1)
> shapiro.test(r1)

        Shapiro-Wilk normality test

data:  r1
W = 0.9421, p-value = 0.5456

# 第三组数据的残差不符合正态分布
> lm.3=lm(Y3~X, data=Anscombe)
> r3=resid(lm.3)
> shapiro.test(r3)

        Shapiro-Wilk normality test

data:  r3
W = 0.7406, p-value = 0.00157

# 再次次绘图查看
> plot(Y3~X, data=Anscombe)

# 去掉极端值(第 3 个), 残差还是不能满足正态分布
>  i<-1:11; Y31<-Anscombe$Y3[i!=3]; X3<-Anscombe$X[i!=3]
> lm31<-lm(Y31~X3)
> r31<-resid(lm31)
> shapiro.test(r31)
```

[1]因为Y被H左乘后变为$\hat{Y}$, 由此得名

```
        Shapiro-Wilk normality test

data:  r31
W = 0.7615, p-value = 0.004931

# 绘图查看, 发现第9个值异常
> plot(r31)

# 去掉第9个值, 残差符合正态分布
> i<-1:10; Y32<-Y31[i!=9]; X32<-X3[i!=9]
> lm32<-lm(Y32~X32)
> r32<-resid(lm32)
> shapiro.test(r32)

        Shapiro-Wilk normality test

data:  r32
W = 0.9839, p-value = 0.9813
```

24.2.3 标准化(内学生化)残差

由差向量 ϵ 的性质得到

$$E(\hat{\epsilon}) = 0,\ Var(\hat{\epsilon}) = \sigma^2(I - H)$$

故对每个 $\hat{\epsilon}$

$$\frac{\hat{\epsilon}_i}{\sigma\sqrt{1 - h_{ii}}} \sim N(0, 1)$$

其中 h_{ii} 是 H 对角线上的元素. 称

$$r_i = \frac{\hat{\epsilon}_i}{\hat{\sigma}\sqrt{1 - h_{ii}}}$$

为标准化残差(standardized residual), 也叫做内学生化残差(internally studentized residual). 因为 σ^2 的估计用到了包括第i个样本在内的全部数据. $r_i \sim N(0, 1)$

函数 rstandard() 计算标准化残差.

```
> rstandard(lm32)
       1        2        3        4        5        6        7
 0.0576117  0.2864334 -1.5984163  1.7677670 -0.4926925  0.5404789  0.8784586
       8        9
-0.1835970 -1.2598816
```

24.2.4 外学生化残差

记删除第i个样本数据后, 由余下的数据求得的回归系数为 $\hat{b}_{(i)}$, 则 σ^2 的估计为

$$\hat{\sigma}^2_{(i)} = \frac{1}{n-p-2}\sum_{j\neq i}(Y_i - X_j\hat{b}_{(i)})^2$$

其中 X_j 为设计矩阵 X 的第j行. 称

$$\hat{\epsilon}_i(\hat{b}_{(i)}) = \frac{\hat{\epsilon}_i}{\hat{\sigma}_{(i)}\sqrt{1-h_{ii}}}$$

为学生化残差, 或称为外学生化残差(externally studentized residual).

函数 rstudent() 计算标准化残差.

```
> rstudent(lm32)
        1         2         3         4         5         6
 0.05335072  0.26675362 -1.85706229  2.19970673 -0.46426557  0.51116565
        7         8         9
 0.86220749 -0.17038855 -1.32647306
```

24.2.5 残差图

以残差 $\hat{\epsilon}_i$ 为纵坐标, 拟合值 $\hat{y}_i$ 或对应的数据观测序号i或观测时间为横坐标的散点图统称为残差图. 残差图是进行模型诊断的重要工具. (可以直接使用 plot.lm() 函数绘制)

下面我们绘制第一组回归(拟合比较好, 残差服从正态分布) lm1 的残差图和标准化残差图

```
> fit1=fitted(lm1) # predict(lm1) 也可以

# 残差图
> r1=resid(lm1);
> plot(r1~fit1)

# 标准化残差图
> rst1=rstandard(lm1)
> plot(rst1~fit1)

# 两个图画在一起
> par(mfrow=(c(1,2)))
> plot(r1~fit1)
> plot(rst1~fit1)
```

对于标准化残差, 应该有大约 95% 的样本落入$[-2,2]$之间, 则若拟合值 $\hat{Y}$ 为横坐标, 那么标准化残差大概落入 $[-2,2]$ 内, 且不呈现任何趋势. 否则回归模型可能有问题.

下面看第二组回归(曲线) lm2 . 可以看到, 曲线回归后残差图变好.

```
> rst2=rstandard(lm2)
> fit2=fitted(lm2)

# 曲线回归
> lm2curve=lm(Y2~X+I(X^2), data=Anscombe)

> rst2c=rstandard(lm2curve)
```

```
> fit2c=fitted(lm2curve)

# 绘制残差图
> par(mfrow=(c(1,2)))
> plot(rst2~fit2)
> plot(rst2c~fit2c)
```

24.2.6 残差的 Q-Q 图

可以使用 QQ 图检验残差的正态性.

设 $\hat{\epsilon}_{(i)}, i = 1, 2, \cdots, n$ 是残差的次序统计量, 而

$$q_{(i)} = \Phi^{-1}(\frac{i - 0.375}{n + 0.25})$$

为 $\hat{\epsilon}_{(i)}$ 的期望值. 其中 Φ^{-1} 为标准正态分布的反函数.

可以证明, 若 $\hat{\epsilon}_{(i)}$ 来自正态分布, 则 $(q_{(i)}, \hat{\epsilon}_{(i)})$ 应该在一条直线上. 若明显不在直线, 那么怀疑 $\hat{\epsilon}_{(i)}$ 是否为正态分布.

R 中直接使用 plot(lm, 2) 即可

24.3 影响分析

所谓影响分析就是探查对估计有异常大影响的数据. 例如第三组数据.

如果一个样本不遵守某个明显, 但是其余遵守, 称这个样本为强影响点(异常值点).

影响分析的重要功能就是区分这样的点.

24.3.1 帽子矩阵H的对角元素

从前面可以得到, $\hat{Y} = HY$, $\hat{Y}$ 是 Y 在 X 的列向量张成的子空间内的投影, 且有

$$\frac{\partial \hat{Y_i}}{\partial Y_i} = h_{ii}$$

h_{ii} 为 H 的对角元素.

故 h_{ii} 的大小可以表示第i个样本对 $\hat{Y_i}$ 的影响力. 考虑 $\hat{Y_i}$ 的方差

$$var(\hat{Y_i}) = h_{ii}\sigma^2$$

故 h_{ii} 也反映了回归值的波动情况.

由投影矩阵 H 的性质得到

$$0 <= h_{ii} <= 1$$
$$\sum H_{ii} = p + 1$$

所以 Hoaglin 和 Welsch(1978) 给出一致判断异常值的方法, 当

$$h_{i_0 i_0} >= \frac{2(p+1)}{n}$$

可以认为第 i_0 组样本影响较大, 结合其它准则, 可以考虑是否剔除.

由于 H 的对角元素是很重要的统计信息, 故 R 也有计算函数 hatvalues() 和 hat()

```
> hatvalues(lm1)
      1       2       3       4       5       6       7       8
0.1000000 0.1000000 0.2363636 0.0909091 0.1272727 0.3181818 0.1727273 0.3181818
      9      10      11
0.1727273 0.1272727 0.2363636
```

24.3.2 DFFITS 准则

Belsley, Kuh 和 Welsch(1980) 给出另外一致准则, 计算统计量

$$D_i(\sigma) = \sqrt{\frac{h_{ii}}{1-h_{ii}}}\frac{\epsilon_i}{\sigma\sqrt{1-h_{ii}}}$$

对第i个样本, 如果有

$$|D_i(\sigma)| > 2\sqrt{\frac{p+1}{n}}$$

则认为第i个样本影响比较大, 应引起注意.

R 中的函数为 dffits

```
> p=1
> n=nrow(Anscombe)

# 第三组的第三个样本可能异常
> d <- dffits(lm3)
> d> 2*sqrt((p+1)/n)
   1    2    3    4    5    6    7    8    9   10   11
FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
> which(d> 2*sqrt((p+1)/n))
3
3
```

24.3.3 Cook 统计量

Cook 在 1977 年提出了 Cook 统计量, 定义为

$$D_i = \frac{(\hat{b}-\hat{b}_{(i)})^T X^T X(\hat{b}-\hat{b}_{(i)})}{(p+1)\hat{\sigma}^2}$$

其中 $\hat{b}_{(i)}$ 为删除第i个样本数据后由余下的 $n-1$ 个样本数据求得回归系数. 经过推导, Cook 统计量可以改写为

$$D_i = \frac{1}{p+1}(\frac{h_{ii}}{1-h_{ii}})r_i^2$$

R 中 cooks.distance() 计算 Cook 统计量

```
> cooks.distance(lm3)
          1           2           3           4           5           6
0.0118305891 0.0021827101 1.3928277909 0.0055254398 0.0260716064 0.3006335925
          7           8           9          10          11
0.0004804045 0.0331943873 0.0596504117 0.0002176290 0.0067519721
```

直观上, Cook 统计量越大的点月可能是异常点, 但是判定异常值点的临界值的选择是很困难的, 应用中视具体情况而定.

24.3.4 COVARATIO 准则

利用全部样本回归系数的估计值的协方差矩阵为

$$var(\hat{b}) = \sigma^2(X^TX)^{-1}$$

去掉第i个样本点的回归系数的估计值的协方差矩阵为

$$var(\hat{b}_{(i)}) = \sigma^2_{(i)}(X^T_{(i)}X_{(i)})^{-1}$$

其中 $X_{(i)}$ 为 X 剔除第i行得到的矩阵.

考虑其协方差的比

$$covratio = \frac{det(\sigma^2_{(i)}(X^T_{(i)}X_{(i)})^{-1})}{det(\sigma^2(X^TX)^{-1})} = \frac{(\hat{\sigma}^2_{(i)})^{p+1}}{(\hat{\sigma}^2)^{p+1}}\frac{1}{1-h_{ii}}$$

如果一个样本点对应的 covratio 值离开 1 越远, 则认为哪个样本点的影响越大.

```
> covratio(lm3)
         1          2          3          4          5          6
1.340490e+00 1.393999e+00 7.360047e-10 1.358209e+00 1.337257e+00 1.362816e+00
         7          8          9         10         11
1.528312e+00 1.798031e+00 1.341787e+00 1.449234e+00 1.641337e+00

# 绘图查看, 第3个样本点接近0.
> plot(covratio(lm3))
```

24.3.5 总结

influence.measures()可以作为诊断分析的概括.

```
> influence.measures(lm3)
Influence measures of
       lm(formula = Y3 ~ X, data = Anscombe) :

     dfb.1_    dfb.X    dffit    cov.r   cook.d   hat inf
1  -4.64e-03 -4.43e-02  -0.1468 1.34e+00 0.011831 0.100
2  -3.75e-02  1.88e-02  -0.0624 1.39e+00 0.002183 0.100
3  -1.83e+02  2.69e+02 342.7851 7.36e-10 1.392828 0.236   *
4  -3.31e-02 -2.11e-18  -0.0997 1.36e+00 0.005525 0.091
5   4.92e-02 -1.17e-01  -0.2197 1.34e+00 0.026072 0.127
6   4.90e-01 -6.67e-01  -0.7898 1.36e+00 0.300634 0.318
7   2.60e-02 -2.01e-02   0.0292 1.53e+00 0.000480 0.173
8   2.39e-01 -2.07e-01   0.2449 1.80e+00 0.033194 0.318   *
9   1.38e-01 -2.32e-01  -0.3365 1.34e+00 0.059650 0.173
10 -1.54e-02  1.05e-02  -0.0197 1.45e+00 0.000218 0.127
11  1.04e-01 -8.62e-02   0.1098 1.64e+00 0.006752 0.236
```

24.4 共线性,条件数,kappa()函数

当自变量彼此相关时, 某变量可能会因为其它变量的改变而改变其效应, 甚至改变符号. 自变量彼此相关称为共线性或

多重共线性. 若出现共线性, 建议使用主成分回归.

24.4.1 什么是共线性

如果存在某些常数 c0,c1,c2 使得线性等式

$$c1X1 + c2X2 = c0$$

对数据成立, 则称两个变量 X1,X2 共线性.

精确共线性较少发生. 故若上式近似成立, 则两个变量 X1,X2 近似共线性.

常用单不完全合式的共线性度量为 X1,X2 相关系数的平方 r_{12}^2. 精确共线性对应 $r_{12}^2 = 1$, 非共线性对应 $r_{12}^2 = 0$

对 p 个自变量, 若有

$$c_1X_1 + c_2X_2 + \cdots + c_pX_p = c_0$$

近似成立, 称p个变量共线性.

24.4.2 共线性的发现

将 $X_1, X_2, \cdots, X_p$ 中心化和标准化得到 $X = x_{(1)}, x_{(2)}, \cdots, x_{(p)}$. 设 λ 为 X^TX (本质上就是$X_1, X_2, \cdots, X_p$ 的相关矩阵) 的一个特征值, φ 为对应的特征向量, 其长度为 1, 即 $\varphi^T\varphi = 1$. 若 $\lambda \approx 0$ 则

$$X^TX\varphi = \lambda\varphi \approx 0$$

用 φ^T 左乘上式, 得到

$$\varphi^TX^TX\varphi = \lambda\varphi^T\varphi = \lambda \approx 0$$

故有

$$X\varphi \approx 0$$

即

$$\varphi_1x_{(1)} + \varphi_2x_{(2)} + \cdots + \varphi_px_{(p)} \approx 0$$

表明向量 $X = x_{(1)}, x_{(2)}, \cdots, x_{(p)}$ 之间有近似线性关系. 那么 $X_1, X_2, \cdots, X_p$ 之间存在共线性. 其中

$$\varphi = (\varphi_1, \varphi_2, \cdots, \varphi_p)$$

度量共线性严重程度的一个重要指标为方阵 X^TX 的条件数

$$\kappa(X^TX) = \parallel X^TX \parallel * \parallel (X^TX)^{-1} \parallel = \frac{\lambda_{max}(X^TX)}{\lambda_{min}(X^TX)}$$

其中 $\lambda_{max}, \lambda_{min}$ 表示方阵 X^TX 的最大最小特征值.

直观上, 条件数 κ 刻画了方阵 X^TX 的特征值差异的大小. 经验上,

$$\begin{aligned} \kappa < 100, &\text{共线性程度很小} \\ 100 <= \kappa <= 1000, &\text{共线性程度中等} \\ \kappa > 1000, &\text{共线性程度严重} \end{aligned}$$

R 中函数 kappa() 计算矩阵的条件数. 下面的数据中自变量有 6 个, 每个有 12 个样本. 除第一个样本外, 其它 11 个满足

$$X_1 + X_2 + X_3 + X_4 = 10$$

```
d <- data.frame(
  Y=c(10.006, 9.737, 15.087, 8.422, 8.625, 16.289,
      5.958, 9.313, 12.960, 5.541, 8.756, 10.937),
  X1=rep(c(8, 0, 2, 0), c(3, 3, 3, 3)),
  X2=rep(c(1, 0, 7, 0), c(3, 3, 3, 3)),
  X3=rep(c(1, 9, 0), c(3, 3, 6)),
  X4=rep(c(1, 0, 1, 10), c(1, 2, 6, 3)),
  X5=c(0.541, 0.130, 2.116, -2.397, -0.046, 0.365,
      1.996, 0.228, 1.38, -0.798, 0.257, 0.440),
  X6=c(-0.099, 0.070, 0.115, 0.252, 0.017, 1.504,
      -0.865, -0.055, 0.502, -0.399, 0.101, 0.432)
)
```

```
# X^T X 本质上是原始矩阵的相关矩阵
> c <-cor(d[2:7])
> kappa(c,exact=T)
[1] 2195.908
```

计算特征值与特征向量

```
> eigen(c)
$values
[1] 2.428787365 1.546152096 0.922077664 0.793984690 0.307892134 0.001106051

$vectors
          [,1]       [,2]       [,3]       [,4]       [,5]        [,6]
[1,] -0.3907189  0.33968212  0.67980398 -0.07990398  0.25103700 -0.447679719
[2,] -0.4556030  0.05392140 -0.70012501 -0.05768633  0.34446553 -0.421140280
[3,]  0.4826405  0.45332584 -0.16077736 -0.19102517 -0.45363721 -0.541689124
[4,]  0.1876590 -0.73546592  0.13587323  0.27645223 -0.01520870 -0.573371872
[5,] -0.4977330  0.09713874 -0.03185053  0.56356440 -0.65128338 -0.006052127
[6,]  0.3519499  0.35476494 -0.04864335  0.74817535  0.43374630 -0.002166594

# 手工计算特征值
> e <- eigen(c)
> max(e$values)/min(e$values)
[1] 2195.908
```

最小的特征值$\lambda_{min} = 0.001106$, 对应的特征向量为

```
]
# e$vectors 的最后一列
> e$vectors[,which(e$values==min(e$values))]
[1] -0.447679719 -0.421140280 -0.541689124 -0.573371872 -0.006052127
[6] -0.002166594
```

则有

$$0.4476x_{(1)} + 0.4211x_{(2)} + 0.5417x_{(3)} + 0.5734x_{(4)} + 0.006052x_{(5)} + 0.002167x_{(6)} \approx 0$$

由于 $x_{(5)}, x_{(6)}$ 系数近似为 0, 故

$$0.4476x_{(1)} + 0.4211x_{(2)} + 0.5417x_{(3)} + 0.5734x_{(4)} \approx 0$$

所以存在

$$c_1X_1 + c_2X_2 + c_3X_3 + c_4X_4 \approx c_0$$

变量 X_1, X_2, X_3, X_4共线性.

另, kappa() 也可以计算线性模型的共线性, 但是计算的是 $X_1, X_2, X_3, \cdots, X_p, Y$构成的矩阵的条件数, 即

$$kappa(lm.model) = \kappa([X_1, X_2, X_3, \cdots, X_p, Y])$$

```
> kappa(lm3)
[1] 32.14227
```

Chapter 25

多元线性回归

25.1 注意

我们通常希望尽量减少自变量的个数, 于是一个直观的想法是做很多回归检验. 但是, 这会增加II型错误的概率. 最好使用 Tukey 等方法(anova?)

主成分分析及主成分回归是不错的选择.

25.2 模型

```
x = 1:10
y = x+10+rnorm(10)
z = x+y
l = lm(z~x+y)
```

如果确定没有截距(intercept), 则可以加一个 -1 来表示, 即

```
l = lm(z~x+y-1)
```

25.3 系数的置信区间(CI)

confint(l)

25.4 F-值, p-值

可以用 summary 查看

25.5 回归值

predict(res)

25.6 偏相关与多重相关以及ANCOVA

对于自变量为离散或属性数据(性别,年龄等)的回归分析, 请参考单因素协方差分析(ANCOVA)30.5

假设我们研究两个变量x与y的线性相关程度. 但是在控制其它协变量$z_1, \cdots, z_k$之后. 如下两个派生的变量之间的 Pearson 相关称为x,y的偏相关(partial correlation). e_x=x在$z_1, \cdots, z_k$上的线性回归的残差. e_y=y在$z_1, \cdots, z_k$上的线性回归的残差.

y与所有预测变量的回归函数$b_1x_1 + b_2x_2 + \cdots$之间相关系数称为y与$x_1, x_2, \cdots$的多重相关. 即summary(lm(y x)) 结果中的 Multiple R-Squared

下面是计算偏相关的例子.

```
> library(corpcor)
> x=rep(10,10)+rnorm(10)
> y=x+rnorm(10)
> z=y+rnorm(10)
```

```
> m=matrix(c(x,y,z),nc=3,nr=10)
> cor(m)
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.8241164 0.8025136
[2,] 0.8241164 1.0000000 0.9126755
[3,] 0.8025136 0.9126755 1.0000000
> cor2pcor(cov(m)) # 偏相关矩阵
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.3759994 0.2175614
[2,] 0.3759994 1.0000000 0.7436428
[3,] 0.2175614 0.7436428 1.0000000
> cov2cor(cov(m))
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.8241164 0.8025136
[2,] 0.8241164 1.0000000 0.9126755
[3,] 0.8025136 0.9126755 1.0000000

# 手工计算偏相关系数
> e1=lm(z~x)$res #z在x上的线性回归残差
> e2=lm(z~y)$res #z在y上的线性回归残差
> cor(e1,e2)
[1] 0.5192303
> e3=lm(x~z)$res # x在z上的线性回归残差
> e4=lm(y~z)$res # y在z上的线性回归残差
# x y的偏相关系数(控制z后) 与 cor2pcor(cov(m)) 结果一致
> cor(e3,e4)
[1] 0.3759994
```

Chapter 26

多项式回归

26.1 模型函数

```
x = 1:10
y = b+b1*x+b2*x^2+...
l = lm(y~x+I(x^2)+I(x^3)+...)
(I函数可以让我们使用通常的方法来表达指数. 注意: 符
号 ^ 在模型中与在表达式中表示不同的意思)
y~poly(x,5) # 可以使用通用的表达方法来拟合,
```

```
>  summary(lm(y~poly(x,5)))

Call:
lm(formula = y ~ poly(x, 5))

Residuals:
      1       2       3       4       5       6       7       8
 0.08011 -0.30545  0.32234  0.06079 -0.12296 -0.29187  0.27484  0.14366
      9      10
-0.22817  0.06669

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  0.06675    0.10727   0.622  0.56748
poly(x, 5)1 -1.89826    0.33923  -5.596  0.00501 **
poly(x, 5)2 -0.35722    0.33923  -1.053  0.35173
poly(x, 5)3  0.30348    0.33923   0.895  0.42157
poly(x, 5)4 -0.32418    0.33923  -0.956  0.39337
poly(x, 5)5 -0.39118    0.33923  -1.153  0.31307
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

26.2 例子

```
> x = 1:10
> y=10+x+2*x^2+rnorm(10)
> lm(y~x+I(x^2))

Call:
lm(formula = y ~ x + I(x^2))

Coefficients:
(Intercept)            x       I(x^2)
    11.6736       0.4049       2.0484
```

如果确定没有截距(intercept), 则可以加一个 -1 来表示, 即

```
> lm(y~x+I(x^2)-1)

Call:
lm(formula = y ~ x + I(x^2) - 1)

Coefficients:
    x  I(x^2)
4.835   1.697
```

26.3 系数的置信区间(CI)

confint(l)

26.4 F-值, p-值

可以用 summary 查看

26.5 回归值

predict(res)

Chapter 27

广义线性(Generalized Linear)模型

《R导论》[19](page 73)中统计模型部分中有一个广义线性模型对广义线性模型有很好的描述，请参考之。

广义线性建模是线性建模的一种发展，它通过一种简洁而又直接的方式使得线性模型既适合非正态分布的响应值又可以进行线性变换。广义线性模型是基于下面一系列假设前提的：

- 有一个有意义的响应变量y和一系列刺激变量（stimulus variable）x1，x2，. . . 。这些刺激变量决定响应变量的最终分布。
- 刺激变量仅仅通过一个线性函数影响响应值y的分布。该线性函数称为线性预测器（linear predictor），常常写成

  ```
  \eta= \beta_1 x_1 + \beta_2 x_2 + · · · + \beta_p x_p
  ```

  因此x_i当且仅当$\beta_i = 0$时对y的分布没有影响。
- y分布的形式为

$$f_Y(y;\mu,\phi) = exp[\frac{A}{\phi}\{y\lambda(\mu)`\gamma(\lambda(\mu))\} + \tau(y,\phi)]$$

其中ϕ是尺度参数（scale parameter）(可能已知)，对所有观测恒定，A是一个先验的权重，假定知道但可能随观测不同有所不同，μ是y的均值。也就是说假定y的分布是由均值和一个可能的尺度参数决定的。

- 均值μ是线性预测器的平滑可逆函数（smooth invertible function）：

$$\mu = m(\eta), \eta = m^{`1}(\mu) = L(\mu)$$

其中的反函数(inverse function) L() 被称为关联函数（link function）。

这些假定比较宽松，足以包括统计实践中大多数有用的统计模型，同时也足够严谨，使得可以发展参数估计和统计推论(estimation and inference)中一致的方法（至少可以近似一致）。读者如果想了解这方面最新的进展，可以参考McCullagh, Nelder (1989) 或者Dobson (1990)。

27.1 概念

此部分来自 [15] 6.6 广义线性回归模型

广义线性模型对普通线性模型进行了两个方面的推广, 这些推广允许许多线性模型的方法能够应用于一般问题.

- 通过一个连接函数, 将响应变量的期望$E(y)$与线性自变量联系
- 对误差的分布给出一个误差函数

广义线性模型应有以下三个概念

1. 第i个响应变量的期望$E(y_i)$只能通过线性自变量$B^T x_i$依赖于x_i, $B = (p+1)*1$的向量, 可能包含截距.
2. 连接函数, 说明线性自变量和$E(y_i)$的关系, 是线性模型的推广.

3. 误差函数, 说明广义线性模型最后一部分的随机成分

我们保留线性模型中样本相互独立的假设, 去掉可加和正态的假设. 可以从指数分布族中选择一个作为误差函数.

下面的表是常见的连接函数和误差函数

	连接函数	逆连接函数(回归模型)	典型误差函数
恒等(identity)	$x^T\beta = E(y)$	$E(y) = x^T\beta$	正态分布
对数	$x^T\beta = lnE(y)$	$E(y) = exp(x^T\beta$	Poisson 分布
Logit	$x^T\beta = LogitE(y)$	$E(y) = \frac{exp(x^T\beta)}{1+exp(x^T\beta)}$	二项分布
逆	$x^T\beta = \frac{1}{E(y)}$	$E(y) = \frac{1}{x^T\beta}$	Gamma 分布

对分布族提供的连接函数见下面一节.

27.2 族

R 提供了一系列广义线性建模工具，从类型上来说包括高斯(gaussian), 二项式(binomial), 泊松(poisson), 逆高斯(inverse gaussian) 和伽马(gamma) 模型的响应变量分布以及响应变量分布无须明确给定的拟似然（quasi-likelihood）模型。在后者，方差函数（variance function）可以由均值的函数指定，但在其它情况下，该函数可以由响应变量的分布得到。

每一种响应分布允许各种关联函数将均值和线性预测器关联起来。这些自动可用的关联函数如下表所示：

```
族名字            关联函数
binomial        logit, probit, log, cloglog
gaussian        identity, log, inverse
Gamma           identity, inverse, log
inverse.aussian 1/mu^2, identity, inverse, log
poisson         identity, log, sqrt
quasi           logit, probit, cloglog, identity,
                inverse, log, 1/mu^2, sqrt
```

这些用于模型构建过程中的响应分布，关联函数和各种其他必要的信息统称为广义线性模型的族（family）。

27.3 glm()函数

既然响应的分布仅仅通过单一的一个线性函数依赖于刺激变量，那么用于线性模型的机制同样可以用于指定一个广义模型的线性部分。但是族必须以一种不同的方式指定。

R用于广义线性回归的函数是glm()，它的使用形式为

```
>  fitted .model <- glm(formula, family=family.generator , data=data.frame)
```

和lm()相比，唯一的一个新特性就是描述族的参数family.generator。它其实是一个函数的名字，这个函数将产生一个函数和表达式列表用于定义和控制模型的构建与估计过程。尽管这些内容开始看起来有点复杂，但它们非常容易使用。

这些名字是标准的。程序给定的族生成器可以参族部分表格中的"族名"。当选择一个关联函数时，该关联函数名和族名可以同时在括弧里面作为参数设定。在拟（quasi）家族里面，方差函数也是以这种方式设定。

一些例子可能会使这个过程更清楚。

27.3.1 gaussian族

命令

```
> fm <- glm(y ~ x1 + x2, family = gaussian, data = sales)
```

和下面的命令结果一致

```
> fm <- lm(y ~ x1+x2, data=sales)
```

但是效率上，前者差一点。注意，高斯族没有自动提供关联函数设定的选项，因此不允许设置参数。如一个问题需要用非标准关联函数的高斯族，那么只能采用我们后面讨论的拟族。

27.3.2 二项式族

考虑Silvey (1970) 提供的一个人造的小例子。

在Kalythos 的Aegean岛上，男性居民常常患有一种先天的眼科疾病，并且随着年龄的增长而变的愈明显。现在搜集了各种年龄段岛上男性居民的样本，同时记录了盲眼的数目。数据展示如下：

```
Age:         20 35 45 55 70
No.: tested: 50 50 50 50 50
No.: blind:  6  17 26 37 44
```

我们想知道的是这些数据是否吻合logistic 和probit 模型，并且分别估计各个模型的LD50，也就是一个男性居民盲眼的概率为50%时候的年龄。如果y 和n 分别是年龄为x时的盲眼数目和检测样本数目，两种模型的形式都为

$$y \sim B(n, F(\beta_0 + \beta_1 x))$$

其中在probit 模型中，$F(z) = \Phi(z)$ 是标准的正态分布函数，而在logit 模型(默认)中，$F(z) = e^z/(1 + e^z)$。这两种模型中，

$$LD50 = `\beta_0/\beta_1$$

，即分布函数的参数为0时所在的点。

第一步是把数据转换成数据框,

```
kalythos <- data.frame(x = c(20,35,45,55,70),
      y = c(6,17,26,37,44), n = rep(50,5))
```

在glm() 拟合二项式模型时，响应变量有三种可能性：

- 如果响应变量是向量，则假定操作二元（binary）数据，因此要求是0/1向量。
- 如果响应变量是双列矩阵，则假定第一列为试验成功的次数, 第二列为试验失败的次数。
- 如果响应变量是因子，则第一水平作为失败(0) 考虑而其他的作为'成功'(1) 考虑。

这里，我们采用的是第二种惯例。我们在数据框中增加了一个矩阵：

```
>  kalythos$Ymat <- cbind(kalythos$y, kalythos$n - kalythos$y)
> kalythos$Ymat
    [,1] [,2]
[1,]    6   44
[2,]   17   33
[3,]   26   24
[4,]   37   13
[5,]   44    6
```

为了拟合这些模型，我们采用

```
>  fmp <- glm(Ymat ~ x, family = binomial(link=probit), data = kalythos)
>  fml <- glm(Ymat ~ x, family = binomial, data = kalythos) # 默认
为logit关联函数
```

既然logit的关联函数是默认的，因此我们可以在第二条命令中省略该参数。为了查看拟合结果，我们使用

```
> summary(fmp)

Call:
glm(formula = Ymat ~ x, family = binomial(link = probit), data = kalythos)

Deviance Residuals:
      1       2       3       4       5
-0.15582  0.02545 -0.08009  0.51246 -0.40097

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.102270   0.276287  -7.609 2.76e-14 ***
x            0.048147   0.005885   8.181 2.82e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82.14455  on 4  degrees of freedom
Residual deviance:  0.45473  on 3  degrees of freedom
AIC: 24.270

Number of Fisher Scoring iterations: 4

> summary(fml)

Call:
glm(formula = Ymat ~ x, family = binomial, data = kalythos)

Deviance Residuals:
     1       2       3       4       5
-0.1797  0.1157 -0.1182  0.3791 -0.3372

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.53778    0.50232  -7.043 1.88e-12 ***
x            0.08114    0.01082   7.498 6.47e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 82.14455  on 4  degrees of freedom
Residual deviance:  0.31707  on 3  degrees of freedom
AIC: 24.132

Number of Fisher Scoring iterations: 4
```

两种模型都拟合的很好。为了计算LD50，我们可以利用一个简单的函数：

```
> ld50 <- function(b) -b[1]/b[2]
> ldp <- ld50(coef(fmp)); ldl <- ld50(coef(fml)); c(ldp, ldl)
(Intercept) (Intercept)
  43.66335    43.60119
```

从这些数据中得到的年龄分别是43.663年和43.601年。

27.3.3 Poisson模型

Poisson族默认的关联函数是log。在实际操作中，这一族常常用于拟合计数资料的Poisson对数线性模型。这些计数资料的实际分布往往符合二项式分布。这是一个非常重要而又庞大的话题，我们不想在这里深入展开。它甚至是非-高斯广义模型内容的主要部分。

有时候，实践中产生的Poisson数据在对数或者平方根转化后可当作正态数据处理。作为后者的另一种选择是，一个Poisson广义线性模型可以通过下面的方式拟合[1]：

```
> fmod <- glm(y ~ A + B + x, family = poisson(link=sqrt),
            data = worm.counts)
```

[1]找不到数据 worm.counts

27.3.4 拟似然模型

对于所有的族，响应变量的方差依赖于均值并且拥有作为乘数（multiplier）的尺度参数。方差对均值的依赖方式是响应分布的一个特性；例如对于poisson分布$Var[y] = \mu$。

对于拟似然估计和推断，我们不是设定精确的响应分布而是设定关联函数和方差函数的形式，因为关联函数和方差函数都依赖于均值。既然拟似然估计和 gaussian 分布使用的技术非常相似，因此这一族顺带提供了一种用非标准关联函数或者方差函数拟合gaussian 模型的方法。

例如，考虑非线性回归的拟合

$$y = \frac{\theta_1 z_1}{z_2 - \theta_2} + e$$

同样还可以写成

$$y = \frac{1}{\beta_1 x_1 + \beta_2 x_2} + e$$

其中$x_1 = z_2/z_1.x_2 = -1/x_1.\beta_1 = 1/\theta_1, \beta_2 = \theta_2/\theta_1$. 假如有适合的数据框，我们可以如下进行非线性拟合[2]

```
nlfit <- glm(y ~ x1 + x2 - 1,
          family = quasi(link=inverse, variance=constant),
          data = biochem)
```

如果需要的话，读者可以从其他手册或者帮助文档中得到更多的信息。

[2]找不到数据 biochem

27.4 其它资料找到的东东

27.4.1 数据

```
> m1<-rnorm(10) # 理论均值为0, 但是样本量小, 实际可能相差很大.
> a<-rnorm(200,m1,sd=0.3) # a的均值为m1的均值, 标准差为m1的标准差
> m2<-rnorm(10,1) # 理论均值为1, 但是样本量小, 实际可能相差很大.
> b<-rnorm(200,m2,sd=0.3) # a的均值为m2的均值, 标准差为m2的标准差
> x1<-c(a[2*1:100],b[2*1:100]) # x1为a,b的偶数位置的值
> x2<-c(a[2*1:100-1],b[2*1:100-1]) # x2为a,b的奇数位置的值
# y为被预测的值(分类), a偶数,奇数位置确定的分类是0, 用红色表示.
# b偶数,奇数位置确定的分类是1, 蓝色表示.
> y<-c(rep(0,100),rep(1,100))
> plot(x1,x2,col=c('red','blue')[1+y])
```

27.4.2 回归分析

使用线性模型对y分类的预测, 预测效果并不好. 增加交互效应后, 效果没有改善. 因为数据本身不是线性可分的.

```
> plot(x1,x2,col=c('red','blue')[1+y])
> r <- lm(y~x1+x2)
> abline(r)
> r2 <- lm(y~x1+x2+I(x1*x2)) # 增加交互效应
> abline(r2,col='red')
```

TODO: 有一个使用contour的例子, 但是没有看懂

27.4.3 Poisson回归

Poisson回归处理应变量为自然数的情况. 流行病学中, Poisson回归处理前瞻性研究(cohort study)数据, 分析感兴趣的时间-人的事件发生率. 常用于单位时间，单位面积，单位空间内某事件发生数(count)的影响因素分析。

Poisson回归的两个假设: 事件发生率是独立于时间的(与时间无关), 理论均值等于方差(这个是Poisson分布的数学性质).

27.5 logit多元线性回归

此部分请参考流行病学部分的多重logistic回归51有详细的描述。

模型: 因变量为质反应时, 阳性的发生概率为p, 不发生(阴性)的概率为1-p. 阳性与阴性概率的比为p/(1-p), 又称为优势比(odds ratio). 此比例的对数与影响阳性发生率的多个自变量呈线性关系. 可以表示为一个logistic多元线性回归方程.

$$ln(p/(1-p)) = a + b_1 * x_1 + b_2 * x_2 + \cdots + b_k * x_k$$

则质反应为阳性的概率估计为

$$p = \frac{exp(a + b_1 * x_1 + b_2 * x_2 + \cdots + b_k * x_k)}{1 + exp(a + b_1 * x_1 + b_2 * x_2 + \cdots + b_k * x_k)}$$

阴性的概率估计为

$$1 - p = \frac{1}{exp(a + b_1 * x_1 + b_2 * x_2 + \cdots + b_k * x_k)}$$

回归方程左边的p代替以y, 阳性时取y=1, 阴性时取y=0, 则这个虚变量可以使ln(p/(1-p))与k个自变量的关系用最大似然方法估计回归系数$a, b_1, \cdots, b_k$.

参考 family predict.glm函数

glm中的 family 参数. family 是描述模型的 error 分布和连接函数的. family = binomial 为logistic回归(= gaussian 为一般线性模型)

下面是一个连续变量对0,1应变量的例子.

```
> n <- 100
> x <- c(rnorm(n), 1+rnorm(n))
> y <- c(rep(0,n), rep(1,n)) # y 为0,1数据,代表阴性和阳性质反应
> plot(y~x)
> abline(lm(y~x),col='red',lty=2) # 线性模型预测
> r <- glm(y~x, family=binomial) # 加入family表明是logistic回归
> xx <- seq(min(x), max(x), length=100)
> yy <- predict(r, data.frame(x=xx)) # 预测时还是线性, 与lm模型一致
> lines(xx,yy, col='blue', lwd=5, lty=2)
> yy <- predict(r, data.frame(x=xx),type='response') # 加入type='response'预测为logistic的
> lines(xx,yy, col='blue', lwd=5, lty=2)
> r

Call:  glm(formula = y ~ x, family = binomial)

Coefficients:
(Intercept)          x
   -0.4711      0.8418

Degrees of Freedom: 199 Total (i.e. Null);  198 Residual
Null Deviance:     277.3
Residual Deviance: 249  AIC: 253

> y.pre <-exp(r$coe[1]+r$coe[2]*xx)/(1+exp(r$coe[1]+r$coe[2]*xx)) # 按照公式计算预测值
> lines(y.pre~xx,col='red',lwd=3) # 绘出预测线, 与glm模型的预测完全一样
```

Chapter 28

非线性回归与非线性最小平方

主要内容来自 [29].

28.1 非线性回归

一般的回归为

$$y_i = x_i'\beta + \epsilon_i$$

x_i'为行向量, β为待估计参数, $\epsilon_i \sim N(0, \sigma^2)$.

非线性模型中

$$y_i = f(x_i', \beta) + \epsilon_i$$

f为非线性形式.

非线性回归的似然值为

$$L(\beta, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp(-\frac{\sum_{i=1}^{n}[y_i - f(x_i', \beta)]^2}{2\sigma^2})$$

当残差平方和

$$S(\beta) = \sum_{i=1}^{n}[y_i - f(x_i', \beta)]^2$$

取最小时, 似然值取最大. 将$S(\beta)$对β取偏导数, 为0时则得到回归系数的估计.

由于方程非线性, 需要数值优化求解. 就像在线性模型中, 通常由残差平方和除以观察数与参数个数之差来估计误差的方差(线性模型中除以n).

协方差的估计略(见参考文献 [29] page 1-2).

28.2 logistic人口模型及使用nls()函数求解

此logistic不是彼logit

函数 nls(): 非线性模型参数的最小平方估计 (Determine the nonlinear (weighted) least-squares estimates of the parameters of a nonlinear model)

下面是人口增长的例子, 为 logistic 模型

$$y_i = \frac{b_1}{1 + e^{b_2 + b_3 x_i}} + \epsilon_i$$

x_i为时间, y_i为此时的人口. b_1为人口的最大容纳量, b_2为$x = 0$时的人口, b_3为人口增长速率.

下面是美国 1790年到 1990 年每隔 10 年的人口数据.

```
> library(car)
> data(US.pop)
> attach(US.pop) # 将 year population 两个变量纳入名字搜索空
间.
> year
 [1] 1790 1800 1810 1820 1830 1840 1850 1860 1870 1880 1890 1900 1910 1920 1930
[16] 1940 1950 1960 1970 1980 1990
> population
 [1]   3.929   5.308   7.240   9.638  12.866  17.069  23.192  31.443  39.818
[10]  50.156  62.948  75.995  91.972 105.711 122.775 131.669 150.697 179.323
[19] 203.302 226.542 248.710
```

```
> plot(year, population)
```

看到1990年人口为250单位(million), 还没有看出到达上限, 我们不妨设$b_1 = 350$, 有

$$3.929 = \frac{350}{1 + e^{b_2 + b_3 0}}$$

可以解得

$$b_2 = log_e(350/3.929 - 1) = 4.5$$

然后将$x = 1(1800), b_2 = 4.5$带入, 可以得到$b_3 = -0.3$. 我们就获得了迭代的初始值. 下面就使用 nls 函数来拟合. ($trace = T$ 可以显示迭代的残差平方和(最前面的数字).)

```
> time <- 0:20
> pop.mod <- nls(population ~ beta1/(1 + exp(beta2 + beta3*time)),
+     start=list(beta1 = 350, beta2 = 4.5, beta3 = -0.3),
+     trace=T)
13007.48 :  350.0   4.5  -0.3
609.5727 :  351.8074862   3.8405002  -0.2270578
365.4396 :  383.7045367   3.9911148  -0.2276690
356.4056 :  389.1350260   3.9897242  -0.2265769
356.4001 :  389.1462874   3.9903758  -0.2266276
356.4001 :  389.1665272   3.9903412  -0.2266193
356.4001 :  389.1655106   3.9903457  -0.2266199

> pop.mod
Nonlinear regression model
  model:  population ~ beta1/(1 + exp(beta2 + beta3 * time))
   data:  parent.frame()
   beta1    beta2    beta3
389.1655   3.9903  -0.2266
 residual sum-of-squares: 356.4

Number of iterations to convergence: 6
```

```
Achieved convergence tolerance: 1.455e-06

> summary(pop.mod)

Formula: population ~ beta1/(1 + exp(beta2 + beta3 * time))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
beta1 389.16551   30.81196   12.63 2.20e-10 ***
beta2   3.99035    0.07032   56.74  < 2e-16 ***
beta3  -0.22662    0.01086  -20.87 4.60e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.45 on 18 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.455e-06
```

前面的模型的形式为了方便计算而写, 实际上的模型应该将指数部分变化一下形式, 变为

$$y_i = \frac{b_1}{1 + e^{-b_3(x_i - b_2/b_3}} + \epsilon_i$$

那么, $-b_3 = 0.22662$为实际的增长率. b_1, b_2 意义不变, 分别为人口的最大容纳量和和时间为 0 时的人口.

28.3 非线性最小二乘法和最大似然法模型

此节摘自《R导论》[19](page 78).

特定形式的非线性模型可以通过广义线性模型(glm())拟合。但是许多时候，我们必须把非线性拟合的问题作为一个非线性优化的问题解决。R的非线性优化程序是optim()，nlm() 和nlminb()（自R2.2.0开始）。二者分别替

换SPLUS 的ms()和nlminb()但功能更强。我们通过搜寻参数值使得缺乏度（lack-of- fit）指标最低，如nlm() 就是通过循环调试各种参数值得到最优值。和线性回归不同，程序不一定会收敛到一个稳定值。nlm()需要设定参数搜索的初始值，而参数估计是否收敛在很大程度上依赖于初始值设置的质量(可以用一些经验的方法判断初始的参数设定。)

28.3.1 nlm()函数的用法

```
nlm(f, p, ..., hessian = FALSE, typsize = rep(1, length(p)),
   fscale = 1, print.level = 0, ndigit = 12, gradtol = 1e-6,
   stepmax = max(1000 * sqrt(sum((p/typsize)^2)), 1000),
   steptol = 1e-6, iterlim = 100, check.analyticals = TRUE)
```

此函数使用 Newton 型算法求极小值, 返回极小值, 极小点的估计值, 极小点处的梯度, hessen 矩阵, 迭代次数等.

f: 求极小值的目标函数, 若其属性(attr)包含 'gradient' or both 'gradient' and 'hessian', 则在计算过程中会使用它们. 否则使用数值的方法来计算偏导数.

p: 参数初始值

hessian = True 会返回最小化时的 hessian 矩阵

28.3.2 最小二乘法

拟合非线性模型的一种办法就是使误差平方和（SSE）或残差平方和最小。如果观测到的误差极似正态分布，这种方法是非常有效的。

下面是例子来自Bates, Watts (1988)，51页。具体数据是：

```
x <- c(0.02, 0.02, 0.06, 0.06, 0.11, 0.11, 0.22, 0.22, 0.56,
   0.56, 1.10, 1.10)
```

```
y <- c(76, 47, 97, 107, 123, 139, 159, 152, 191, 201, 207, 200)
```

被拟合的模型是(实际上编写一个计算误差平方和的函数, nlm()对此函数进行最小化):

```
fn <- function(p) sum((y - (p[1] * x)/(p[2] + x))^2)
```

为了进行拟合，我们需要估计参数初始值。一种寻找合理初始值的办法把数据图形化，然后估计一些参数值，并且利用这些值初步添加模型曲线。

```
plot(x, y)
xfit <- seq(.02, 1.1, .05)
yfit <- 200 * xfit/(0.1 + xfit)
lines(spline(xfit, yfit))
```

当然，我们可以做的更好，但是初始值200 和0.1 应该足够了。现在做拟合：

```
> out <- nlm(fn, p = c(200, 0.1), hessian = TRUE)
> out
$minimum
[1] 1195.449

$estimate
[1] 212.68384222   0.06412146

$gradient
[1] -0.0001534973  0.0934205639

$hessian
          [,1]       [,2]
[1,]   11.94725   -7661.319
[2,] -7661.31875 8039421.153
```

```
$code
[1] 3

$iterations
[1] 26
```

拟合后，out$minimum 是误差的平方和（SSE），out$estimate 是参数的最小二乘估计值。为了得到参数估计过程中近似的标准误(SE)，我们可以计算：

```
>  sqrt(diag(2*out$minimum/(length(y) - 2) * solve(out$hessian)))
[1] 7.173465192 0.008744815
```

上述命令中的2表示参数的个数。一个95% 的信度区间可以通过± 1.96 SE 计算得到。我们可以把这个最小二乘拟合曲线画在一个新的图上：

```
> plot(x, y)
> xfit <- seq(.02, 1.1, .05)
> yfit <- 212.68384222 * xfit/(0.06412146 + xfit)
> lines(spline(xfit, yfit))
```

标准包stats 提供了许多用最小二乘法拟合非线性模型的扩充工具。我们刚刚拟合过的模型是Michaelis-Menten 模型，因此可以利用下面的命令得到类似的结论。

```
>  df <- data.frame(x=x, y=y)
>  fit <- nls(y ~ SSmicmen(x, Vm, K), df)
> fit
Nonlinear regression model
  model:  y ~ SSmicmen(x, Vm, K)
   data:  df
      Vm        K
```

```
212.68371   0.06412
 residual sum-of-squares: 1195

Number of iterations to convergence: 0
Achieved convergence tolerance: 1.924e-06
> summary(fit)

Formula: y ~ SSmicmen(x, Vm, K)

Parameters:
   Estimate Std. Error t value Pr(>|t|)
Vm 2.127e+02  6.947e+00  30.615 3.24e-11 ***
K  6.412e-02  8.281e-03   7.743 1.57e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.93 on 10 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 1.924e-06
```

28.3.3 最大似然法

最大似然法（Maximum likelihood）也是一种非线性拟合方法。它甚至可以用在误差非正态的数据中。这种方法估计的参数将会使得对数似然值最大或者负的对数似然值最小。下面的例子来自Dobson (1990), pp. : 108–111。这个例子对剂量—响应数据拟合logistic模型（当然也可以用glm() 拟合）。数据是：

```
x <- c(1.6907, 1.7242, 1.7552, 1.7842, 1.8113,
     1.8369, 1.8610, 1.8839)
y <- c( 6, 13, 18, 28, 52, 53, 61, 60)
n <- c(59, 60, 62, 56, 63, 59, 62, 60)
```

要使负对数似然值最小，则：

```
> fn <- function(p)
   sum( - (y*(p[1]+p[2]*x) - n*log(1+exp(p[1]+p[2]*x))
          + log(choose(n, y)) ))
```

我们选择一个适当的初始值，开始拟合：

```
> out <- nlm(fn, p = c(-50,20), hessian = TRUE)
> out
$minimum
[1] 18.71513

$estimate
[1] -60.71727  34.27021

$gradient
[1] 1.345785e-08 2.280689e-08

$hessian
         [,1]     [,2]
[1,]  58.48407 103.9787
[2,] 103.97873 184.9662

$code
[1] 1

$iterations
[1] 21
```

拟合后，out$minimum 就是负对数似然值，out$estimate 就是最大似然拟合的参数值。为了得到拟合过程近似的标准误，我们可以：

```
> sqrt(diag(solve(out$hessian)))
[1] 5.553083 3.122531
```

参数估计的95% 信度期间可由估计值 ± 1.96 SE 计算得到。

Chapter 29

逐步回归

参考文献 [15] 下册 6.4 逐步回归.

29.1 是否拟合的足够好?

基本原理: 若模型恰当, 则$\hat{\sigma}$是σ的无偏估计. 若模型过于复杂, 即过拟合, 则$\hat{\sigma} < \sigma$. 若模型太简单, 则$\hat{\sigma} > \sigma$. 这时候, spline, lowess等可以帮助你查看非线性关系

```
> lines(smooth.spline(x,y), col='red', lwd=2)
> lines(lowess(x,y), col='blue', lwd=2)
```

进一步的问题是什么时候才叫做合适? 这个问题很难回答, 也具有根本性. 拟合过火和不足, 都会导致预测能力下降. 有时候, 我们需要放弃线性和多项式模型, 转而寻找其它的方法(bayes, ann等)

29.1.1 σ^2已知

回忆

$$\frac{\hat{\sigma^2}}{\sigma^2} \sim \frac{\chi^2_{n-p}}{(n-p)}$$

那么若

$$\frac{(n-p)\hat{\sigma^2}}{\sigma^2} > \chi^2_{n-p,1-\alpha}$$

则拟合的不好. 这时我们需要一个更好的模型.

```
> summary(res)$sigma # 即 Residual standard error
[1] 246.0147
> 1-pchisq(summary(res)$sigma^2*44,44) # 此值过小说明拟合不
好
[1] 0
```

29.1.2 过拟合

模型过于复杂或样本过简单, 都可以导致过拟合.

```
> x <- seq(0,1,length=n)
> y <- 1-2*x+.3*rnorm(n)
> summary(lm(y~poly(x,10)))
错误在poly(x, 10) : 'degree'小于数据点的数目
> n
[1] 10
> summary(lm(y~poly(x,3)))

Call:
lm(formula = y ~ poly(x, 3))
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-0.52663 -0.05985  0.01600  0.10827  0.50769

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1193     0.1036   1.151 0.293399
poly(x, 3)1  -2.2960     0.3277  -7.006 0.000422 ***
poly(x, 3)2  -0.1015     0.3277  -0.310 0.767343
poly(x, 3)3  -0.1751     0.3277  -0.534 0.612402
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3277 on 6 degrees of freedom
Multiple R-Squared: 0.8918,     Adjusted R-squared: 0.8377
F-statistic: 16.49 on 3 and 6 DF,  p-value: 0.002654
```

29.1.3 欠拟合

模型太简单. 下面是一个欠拟合的例子.

```
> x <- runif(100, -1, 1)
> y <- 1-x+x^2+.3*rnorm(100)
> plot(y~x)
> abline(lm(y~x), col='red')
```

29.2 外推

我们经常想把数据外推(Extrapolation), 例如我们已知的数据范围是(0,1). 但是我们想知道数据在(0,10)的表现. 下面是几个常见的问题.

首先预测区间会线性变大. 下面是一个例子.数据范围是(-3,3), 我们看看(-20,20)是什么样子.

```
> n=20
> x <- rnorm(n)
> y <- 1 - 2*x - .1*x^2 + rnorm(n)
> plot(y~x, xlim=c(-20,20), ylim=c(-30,30)) # 绘出数据
> r <- lm(y~x)
> abline(r, col='red') # 绘出回归线
> xx <- seq(-20,20,length=100)
> p <- predict(r, data.frame(x=xx), interval='prediction') # 预测值
> lines(xx,p[,2],col='blue') # 绘出上界
> lines(xx,p[,3],col='blue') # 绘出下界
```

有时候, 数据局部可能是接近线性的, 但是整体不是. 这个时候使用局部来预测整体就很危险. 接上个例子的数据

```
> yy <- 1 - 2*xx - .1*xx^2 + rnorm(n)
> points(yy~xx)
```

29.3 最优回归方程的选择

实际问题中影响因变量y的因素很多, 人们可以从中挑选若干建立回归方程. 若忽略了对y有显著影响的自变量, 那么误差就会很大. 若变量选择过多, 使用就不方便, 且当有的自变量对y的影响不大时, 可能因为自由度减小而对误差的估计变大, 从而影响预测精度.

那么如何选择自变量呢? 在不同的最优准则下可以有不同的选择. (即对y有显著影响的被选择, 影响不大的排除掉) 有很多方法可以获得最优回归方程, 例如, "一切子集回归法", "前进法", "后退法", "逐步回归法" 等. 其中 "逐步回归法" 由于计算机程序渐变, 使用也较为普遍.

29.4 逐步回归的计算

R 提供的 step(), add1(), drop1() 等函数可以实现. 其中 step() 以 AIC 信息统计量为准则, 通过下最小的 AIC 来达到删除或增加变量的目的. step() 的格式为

```
step(object, scope, scale = 0,
    direction = c("both", "backward", "forward"),
    trace = 1, keep = NULL, steps = 1000, k = 2, ...)
```

参数 object 主要为 lm 和 glm. (stepAIC in package 'MASS' 有关于 scope 用法的例子). 参数 direction 为 "both" 是 "一切子集回归法", " forward" 是 "前进法", "backward" 是 "后退法".

下面是一个例子. X1,X2,X3,X4 为水泥中的四种成分. Y 为凝固时释放的热量. 我们希望寻找其线性关系. 首先做线性回归.

```
cement<-data.frame(
   X1=c( 7,  1, 11, 11,  7, 11,  3,  1,  2, 21,  1, 11, 10),
   X2=c(26, 29, 56, 31, 52, 55, 71, 31, 54, 47, 40, 66, 68),
   X3=c( 6, 15,  8,  8,  6,  9, 17, 22, 18,  4, 23,  9,  8),
   X4=c(60, 52, 20, 47, 33, 22,  6, 44, 22, 26, 34, 12, 12),
   Y =c(78.5, 74.3, 104.3,  87.6,  95.9, 109.2, 102.7, 72.5,
       93.1,115.9,  83.8, 113.3, 109.4)
)

> lm.sol<-lm(Y ~ X1+X2+X3+X4, data=cement)
> summary(lm.sol)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4, data = cement)

Residuals:
   Min     1Q  Median     3Q    Max
-3.1750 -1.6709  0.2508  1.3783  3.9254

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  62.4054    70.0710   0.891   0.3991
X1            1.5511     0.7448   2.083   0.0708 .
X2            0.5102     0.7238   0.705   0.5009
X3            0.1019     0.7547   0.135   0.8959
X4           -0.1441     0.7091  -0.203   0.8441
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.446 on 8 degrees of freedom
Multiple R-squared: 0.9824,     Adjusted R-squared: 0.9736
F-statistic: 111.5 on 4 and 8 DF,  p-value: 4.756e-07
```

看到系数都不显著, 效果不是很好. 下面使用 step() 来逐步回归.

```
> lm.step<-step(lm.sol)
Start:  AIC=26.94
Y ~ X1 + X2 + X3 + X4

       Df Sum of Sq    RSS    AIC
- X3    1     0.109 47.973 24.974
- X4    1     0.247 48.111 25.011
- X2    1     2.972 50.836 25.728
<none>              47.864 26.944
- X1    1    25.951 73.815 30.576

Step:  AIC=24.97
Y ~ X1 + X2 + X4

       Df Sum of Sq    RSS    AIC
<none>              47.97  24.97
- X4    1      9.93  57.90  25.42
- X2    1     26.79  74.76  28.74
- X1    1    820.91 868.88  60.63
```

start 步中, 全部变量回归时, AIC 为 26.94. 如果去掉 X3, AIC 变为 24.97, 去掉 X4, AIC 为 25.01. 去掉 X2 AIC 为 25.73, 去掉 X1

AIC 为 30.58. 故第一步完成后判断去掉 X3 AIC 最小, 故得到的模型为

```
# 第一步得到的模型. 去掉 X3
Y ~ X1 + X2 + X4
```

然后使用此模型进行下一轮计算. 在下一轮计算中, 看到无论去掉哪个变量 AIC 值都会增加, 故终止计算, 得到最优回归方程.

下面分析一下回归的显著性

```
> summary(lm.step)

Call:
lm(formula = Y ~ X1 + X2 + X4, data = cement)

Residuals:
   Min     1Q  Median     3Q    Max
-3.0919 -1.8016  0.2562  1.2818  3.8982

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  71.6483    14.1424   5.066 0.000675 ***
X1            1.4519     0.1170  12.410 5.78e-07 ***
X2            0.4161     0.1856   2.242 0.051687 .
X4           -0.2365     0.1733  -1.365 0.205395
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.309 on 9 degrees of freedom
Multiple R-squared: 0.9823,     Adjusted R-squared: 0.9764
F-statistic: 166.8 on 3 and 9 DF,  p-value: 3.323e-08
```

可以看到系数显著性水平有提高, 但是 X2, X4 变量的系数仍然不理想. 如何去做?

我们可以使用 drop1() 函数.(add1() 的用法见在线帮助)

```
> drop1(lm.step)
Single term deletions

Model:
Y ~ X1 + X2 + X4
      Df Sum of Sq    RSS    AIC
<none>              47.97  24.97
X1     1    820.91 868.88  60.63
X2     1     26.79  74.76  28.74
X4     1      9.93  57.90  25.42
```

可以看到, 去掉 X4 AIC 增加最小. 另外残差也是逐步回归的重要指标, 残差越小, 拟合就越好. 去掉 X4 也是残差增加最少的. 下面去掉 X4 做回归

```
> lm.opt<-lm(Y ~ X1+X2, data=cement); summary(lm.opt)

Call:
lm(formula = Y ~ X1 + X2, data = cement)

Residuals:
   Min     1Q Median     3Q    Max
-2.893 -1.574 -1.302  1.362  4.048

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 52.57735    2.28617   23.00 5.46e-10 ***
X1           1.46831    0.12130   12.11 2.69e-07 ***
X2           0.66225    0.04585   14.44 5.03e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.406 on 10 degrees of freedom
Multiple R-squared: 0.9787,     Adjusted R-squared: 0.9744
F-statistic: 229.5 on 2 and 10 DF,  p-value: 4.407e-09
```

看到系数都比较显著. 故最终的回归方程为

```
Y ~ X1 + X2
Y = 52.58 + 1.47*X1 + 0.66 * X2
```

29.5 更新拟合模型

参考[19] page 73, 11.5. 使用 update() 函数. 模型中"."表示旧的模型中对应部分. 下面是一个例子, 使用 x1,x2拟合y, 然后额外增加x3再进行拟合. 进一步, 对响应变量y的平方根变换后再拟合.

```
> x1=rnorm(100)
> x2=rnorm(100)
> x3=rnorm(100)
> y=rnorm(100)
> fm2 <- lm(y ~ x1 + x2)
> fm3 <- update(fm2, . ~ . + x3) # 增加x3再进行拟合
> smf3 <- update(fm3, sqrt(.) ~ .) # 对响应变量y的平方根变换后再拟合.
```

Chapter 30

方差分析(ANOVA)

参考文献 [15] chapter 7.

参考文献 [11] chapter 12.

参考文献 [30] chapter 16.

30.1 介绍

两个组均值的比较使用t检验, 多于两个组的时候就使用方差分析(analysis of variance, ANOVA). 实际上是t检验的拓广.

30.2 多组比较的条件及检验

30.2.1 条件

1. 各组方差齐性, 即所有 i, j $\varepsilon_{i,j}$ 有相同的 σ^2.
2. 总体平均数为 0, 使样本平均数为总体平均数的无偏估计.

3. 服从正态分布. 这个要求对假设检验是必需的, 对参数估计不一定需要.

30.2.2 误差的正态性检验

使用 shapiro.test(). 参考第 18 章

30.2.3 方差齐性检验

两个样本可以使用F检验. 多于两个使用Bartlett检验.

两个非正态样本使用 ansari.test 或 mood.test , 它们是非参数检验. 多于两个非正态样本参考 fligner.test.

bartlett.test 有两种用法.

```
> bartlett.test(list(rnorm(100),rnorm(100)+1,rnorm(100)+2))

        Bartlett test of homogeneity of variances

data:  list(rnorm(100), rnorm(100) + 1, rnorm(100) + 2)
Bartlett's K-squared = 2.7132, df = 2, p-value = 0.2575

> bartlett.test(c(rnorm(100),exp(rnorm(100))+1,rnorm(100)+2),
    g=c(rep(1,100),rep(2,100),rep(3,100)))

        Bartlett test of homogeneity of variances

data:  c(rnorm(100), exp(rnorm(100)) + 1, rnorm(100) + 2) and c(rep(1, 100), rep(2,
Bartlett's K-squared = 141.2364, df = 2, p-value < 2.2e-16
```

固定效应模型某个因素的水平是固定的, 例如, 死亡原因中的疾病种类

30.3 单因素方差分析-固定效应模型

单因素方差分析, 也叫做单因素方差分析(one-way nalysis of variance). 目的是比较多个均值是否相等.

30.3.1 数据描述

y_1	$\cdots$	y_k
y_{11}	$\cdots$	y_{k1}
$\cdots$	$\cdots$	$\cdots$
y_{1n_1}	$\cdots$	y_{kn_k}
$\bar{y}_1$	$\cdots$	$\bar{y}_k$

记总平均值为$\bar{y}$

30.3.2 模型

如果 Y 依赖于 X, 例如象下面 $Y = a0+a1*(X == 1)+a2*(X == 2) + a3 * (X == 3) + a4 * (X == 4)$

与 $Y = b0$ 比较. 即

$$y_{ij} = \mu + \alpha_i + e_{ij}$$

R的公式可以这样

```
y ~ x
```

30.3.3 平方和的分解

$$\sum_{i=1}^{k}\sum j=1^{n_i}(y_{ij}-\bar{y})^2 \;=\; \sum_{i=1}^{k}\sum j=1^{n_i}(y_{ij}-\bar{y}_i)^2+\sum_{i=1}^{k}\sum j=1^{n_i}(\bar{y}_i-\bar{y})^2$$

$$SS_T = SS_B + SS_W$$

其中

- SS_T: 总平方和(total)
- SS_B: 组间平方和(between)
- SS_W: 组内平方和(within)
- SS_B的自由度为 $k-1$
- SS_W的自由度为 $n-k$

那么组间平均平方和为

$$MS_B = SS_B/(k-1)$$

那么组内平均平方和为

$$MS_W = SS_W/(n-k)$$

30.3.4 方差分析表

Table 30.1: 单因素方差分析表

方差来源	自由度	平方和	均方	F值	p值
因素	$k-1$	SS_B	$MS_B = SS_B/(k-1)$	$F = MS_B/MS_W$	p
误差	$n-k$	SS_W	$MS_W = SS_W/(n-k)$		
总和	$n-1$	$SS_T = SS_B + SS_W$			

30.3.5 F检验

原理是: 如果组间的差异大于组内的差异, 拒绝零假设, 否则接受零假设.

注意: 不能够得知哪个组的均值有显著差异, 若需进一步知道, 使用多重检验.

检验为

$$H_0: 所有\alpha_i = 0 \quad vs. \quad H_1: 至少一个\alpha_i \neq 0$$

检验统计量为

$$F = MS_B/MS_W \sim F_{k-1,n-k} \quad (H_0下)$$

判断

$$F > F_{k-1,n-k,1-\alpha}, 拒绝零假设$$
$$F <= F_{k-1,n-k,1-\alpha}, 接受零假设$$

p-值为

$$p - value = P(F_{k-1,n-k} > F)$$

30.3.6 例子

使用 anova() 和 aov() 函数

下面是一个例子. y 被 x 分为 3 组, 比较 3 组 y 均值是否相同. 其中第一行为组间变量的信息, Df 为自由度, Sum Sq 为平方和SS, Mean Sq 为平均平方和SS

```
# 数据
n <- 30
x <- sample(LETTERS[1:3],n,replace=T, p=c(3,2,1)/6)
x <- factor(x)
y <- rnorm(n)
```

```
# 绘图
plot(y ~ x,
    col = 'pink',
    xlab = "", ylab = "",
    main = "Simple anova: y ~ x")

# F值小(p值大), 说明均值差异不显著.
# SS_B=0.3478, SS_W=28.368, df_B=2, df_W=27
# MS_B=0.1739, MS_W=1.0507
# F=MS_B/MS_W=0.1655
# p值=P(F_{2,27}>F)=0.8483
> anova(lm(y~x)) # anova 必须与lm联合使用.
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value Pr(>F)
x          2  0.3478  0.1739  0.1655 0.8483
Residuals 27 28.3680  1.0507

# summary(aov(y~x)) 与 anova(lm(y~x)) 的结果是一样的
> summary(aov(y~x)) # p值大, 说明均值差异不显著.
            Df  Sum Sq Mean Sq F value Pr(>F)
x            2  0.3478  0.1739  0.1655 0.8483
Residuals   27 28.3680  1.0507
```

下面是另外一个例子

```
x=rnorm(100)
y=rnorm(100)+1
z=rnorm(100)+2
data=c(x,y,z)
g=c(rep(0,100),rep(1,100),rep(2,100)) # 分组信息
> boxplot(data~g) # 画图看看
>
> bartlett.test(data~g) # 方差齐性检验

        Bartlett test of homogeneity of variances
```

```
data:  data by g
Bartlett's K-squared = 4.4351, df = 2, p-value = 0.1089

> summary(anova(lm(data~g)))
      Df           Sum Sq        Mean Sq          F value
 Min.   :  1.00   Min.   :195.0   Min.   :  1.059   Min.   :184.2
 1st Qu.: 75.25   1st Qu.:225.2   1st Qu.: 49.549   1st Qu.:184.2
 Median :149.50   Median :255.3   Median : 98.040   Median :184.2
 Mean   :149.50   Mean   :255.3   Mean   : 98.040   Mean   :184.2
 3rd Qu.:223.75   3rd Qu.:285.4   3rd Qu.:146.530   3rd Qu.:184.2
 Max.   :298.00   Max.   :315.6   Max.   :195.020   Max.   :184.2
                                                    NA's   :  1.0
    Pr(>F)
 Min.   :5.434e-33
 1st Qu.:5.434e-33
 Median :5.434e-33
 Mean   :5.434e-33
 3rd Qu.:5.434e-33
 Max.   :5.434e-33
 NA's   :1.000e+00
> summary(aov(data~g))
           Df  Sum Sq Mean Sq F value    Pr(>F)
g           1 195.020 195.020  184.15 < 2.2e-16 ***
Residuals   298 315.589   1.059
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

30.3.7 单向ANOVA与多重回归的关系

请参考文献 [11] 12.5.2, 及单因素协方差分析30.5

使用虚变量来表示各组的变量. 单向ANOVA与多重回归最后计算的结果是相同的

下面是例子

```
res<-lm(data~factor(g))
```

```
> res

Call:
lm(formula = data ~ factor(g))

Coefficients:
(Intercept)  factor(g)1  factor(g)2
   -0.2210      1.1945      2.3326

# 产生虚变量
> k= diag(length(coef(res)))[-1,]
> k
     [,1] [,2] [,3]
[1,]    0    1    0
[2,]    0    0    1
# 结果与线性模型一样
> library(multcomp)
> glht(res, linfct = k)

        General Linear Hypotheses

Linear Hypotheses:
      Estimate
1 == 0    1.195
2 == 0    2.333
```

30.4 单因素方差分析中均值的多重比较

当F检验拒绝零假设, 我们需要找到哪两个组的均值不同, 需要使用多重比较.

方法比较多, 原理都是调整临界值或置信水平, 减少假阳性或假阴性.

30.4.1 LSD法(最小显著性差异法)

LSD法(least significant difference), 称为最小显著性差异法.

原理为: 对指定两个组的数据进行t检验, 但是对方差的估计是利用全体数据的均方MS_W, 故检验统计量t的自由度变大.

注意: k个组的方差齐性检验相等时才能利用全体数据的均方MS_W, 否则只能做普通的t检验.

检验假设

$$H_0 : \alpha_i = \alpha_j \quad vs. \quad H_1 : \alpha_i \neq \alpha_j$$

检验统计量

$$t = \frac{\bar{y}_i - \bar{y}_j}{\sqrt{MS_W(\frac{1}{n_i} + \frac{1}{n_j})}} \sim t_{n-k}$$

对于双侧置信水平α有

$$|t| > t_{n-k,1-\alpha/2}, \quad reject\ H_0$$
$$|t| <= t_{n-k,1-\alpha/2}, \quad accept\ H_0$$

p值为

$$p = 2 * P(t_{n-k} > |t|)$$

对于只做一对比较, 显著性水平$\alpha = 0.05$是合适的, 但是如果做多对比较, 那么I型错误的概率会增加. 即假阳性增加. 下面是显著性水平增加的情况([30] 16.1.5)

30.4.2 Bonferroni法-LSD法的修正

由于LSD法的假阳性增加的问题, 需要修正其置信水平α, 或等价的修正其检验统计量的阈值.

Table 30.2: 多重比较I型错误概率(假阳性增加)

组数(k)	2	3	4	5	6
正常I型错误概率	5%	5%	5%	5%	5%
多重比较	5%	12.2%	20.3%	28.6%	36.6%

显著性水平的修正为

$$\alpha^* = \alpha/\binom{k}{2}$$

下面是其理由. 如果有k个组, 两两比较的数目为$c = \binom{k}{2}$. 记E为至少一个两组比较是显著的这一事件, $P(E)$有时候称为"实验性I型误差"(experiment-wise type I error). 需要决定α^*使得$P(E) = \alpha$

如果两两比较是独立的, 有

$$P(\bar{E}) = 1 - \alpha = (1 - \alpha^*)^c$$

当α^*很小的时候有[1]

$$1 - \alpha = (1 - \alpha^*)^c \approx 1 - c\alpha^* \Longrightarrow \alpha^* = \alpha/\binom{k}{2}$$

多重比较比普通的LSD法要严格, 即LSD显著的在多重比较中可能不显著.

应该指出, 通常的两两比较不会都是独立的, 故α^*的合适值一般要大于$\alpha/\binom{k}{2}$, 所以Bonferroni法是保守的.

一般, 在事先没有计划要对某些特定的组比较且k比较大的时候, 使用Bonferroni法, 在组数较小且仅仅对某些特定的组比较的时候(通常称草案分析)建议使用LSD法.

[1]展开略去高阶项

30.4.3 线性约束

参考文献 [11] 12.4.2

比LSD法更一般的是选取a个组和另外的b个组做比较.

下面是一个肺病的例子一般人群中, 轻度, 中度, 重度吸烟

组号	吸烟情况	样本量	肺功能(用力中期呼出量,FEF)
1	非吸烟	200	3.78
2	被动吸烟	200	3.30
3	非吸入吸烟(不把烟吸入)	50	3.32
4	轻度吸烟(1-10支/天)	200	3.23
5	中度吸烟(11-39支/天)	200	2.73
6	重度吸烟(40支以上/天)	200	2.59

的比例大概是$10\%, 70\%, 20\%$.

我们想比较吸烟的(包括轻度, 中度, 重度)和非吸烟的人群的肺功能差异. 对于此问题, 应该使用线性约束的估计及检验.

线性约束(linaer contrast): 指对某些组的均值做线性组合, 其系数之和应该为0. 即

$$L = \sum_{i=1}^{k} c_i \bar{y}_i$$

$$\sum_{i=1}^{k} c_i = 0$$

注意两个组之间的比较可以算做特例.

例如, 比较非吸烟的和被动吸烟的肺功能, 线性约束可以写为

$$L = \bar{y}_1 - \bar{y}_2, \quad \text{其中} \quad c_1 = 1, c_2 = -1$$

吸烟的(包括轻度, 中度, 重度)和非吸烟的人群的肺功能差异, 线性约束可以写为

$$L = \bar{y}_1 - 0.1\bar{y}_4 - 0.7\bar{y}_5 - 0.2\bar{y}_6$$

记μ_L为L的理论值, 即

$$\mu_l = c_1\alpha_1 + \cdots + c_k\alpha_k$$

因为$Var(\bar{y}_i) = MS_W/n_i$, 故

$$Var(L) = MS_W \sum_{i=1}^{k} c_i^2/n_i$$

那么假设检验为

$$H_0 : \mu_L = 0 \quad vs \quad H_1 : \mu_L \neq 0$$

检验统计量为

$$t = \frac{L}{\sqrt{Var(L)}} \sim t_{n-k}$$

对于双侧置信水平α有

$$\begin{aligned} |t| > t_{n-k,1-\alpha/2}, &\quad reject\ H_0 \\ |t| <= t_{n-k,1-\alpha/2}, &\quad accept\ H_0 \end{aligned}$$

p值为

$$p = 2 * P(t_{n-k} > |t|)$$

线性约束的其它用法: 当不同的组与某种特定的数量指标(例如, 药物剂量)对应时, 线性约束的系数可以取能够反映上述数量关系的值. 在不同组中样本量差别很大时, 特别有用. 因为小样本的组统计检验时常常出现不显著的结果, 但是其趋势常在某个方向上.

例如, 考察吸烟的(包括轻度, 中度, 重度)吸烟数量是否影响肺功能. 还要考察吸烟数量与肺功能的方向关系.

轻度吸烟, 我们取平均值$(1+10)/2=5.5$, 中度吸烟平均值$(11+39)/2=25$, 重度吸烟平均值取40代表(这是一个保守的估计), 检验

$$L=5.5\bar{y}_4+25\bar{y}_5+40\bar{y}_6$$

为了使系数和为0, 将每个系数减去系数的平均值$(5.5+25+40)/3=23.5$, 约束变为

$$\begin{aligned} L &= (5.5-23.5)\bar{y}_4+(25-23.5)\bar{y}_5+(40-23.5)\bar{y}_6 \\ &= -18\bar{y}_4+1.5\bar{y}_5+16.5\bar{y}_6 \end{aligned}$$

这个约束表示: 3个组中每天吸烟量的增加数.

下面按照步骤检验即可. 设已知$MS_W=0.636$, 那么

```
L=-18*3.23+1.5*2.73+16.5*2.59
s=sqrt(0.636*((-18)^2/200+1.5^2/200+16.5^2/200))
t=L/s
> t
[1] -8.198171
> pt(t,df=1044) # p值
[1] 3.552091e-16
```

30.4.4 scheffe法-线性约束的多重比较

如果某个线性约束不是事先计划好的, 那么应该使用线性约束的多重比较.

检验假设

$$H_0:\mu_L=0 \quad v.s. \quad H_1:\mu_L\neq 0$$

此处

$$\begin{aligned} L &= sum_{i=1}^{k} c_i\bar{y}_i \\ \sum_{i=1}^{k} c_i &= 0 \\ \mu_L &= sum_{i=1}^{k} c_i\mu_i \end{aligned}$$

显著性水平为α

计算检验统计量

$$t = \frac{L}{\sqrt{Var(L)}} = \frac{L}{\sqrt{MS_W \sum_{i=1}^{k} c_i^2/n_i}}$$

记临界值$a = \sqrt{(k-1)F_{k-1,n-k,1-\alpha}}$, 判断

$$\begin{aligned} |t| > a, &\quad reject\ H_0 \\ |t| <= a, &\quad accept\ H_0 \end{aligned}$$

scheffe法也可以用于两组之间的均值比较, 因为是线性约束的特例. 但是此情况下Bonferroni法要更可取, 因为当差异确实存在的时候, Bonferroni法比Scheffe法在显著性检验上更合适.

如果线性约束个数很少, 而且事先就指定要检验的约束, 则我们可以不使用线性约束的多重比较, 因为如果使用线性约束的多重比较, 在发现差异上就会比直接使用线性约束的功效小很多.

如果约束很多, 且不是在数据前指定检验, 则此scheffe法是合适的.

30.4.5 其它方法

Dunnett方法: 比较k个用药组与一个对照组的均值

Duncan法(Newman-Keuls检验): 多组均值的两两比较, 显著性的差别介于LSD与Tukey法之间.

Tukey法: 多组均值的两两比较, 比较严格.

30.4.6 p.adjust() 函数

p.adjust() 函数计算调整后的p值, 用法为

```
p.adjust(p, method = p.adjust.methods, n = length(p))

p.adjust.methods
# c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY",
#   "fdr", "none")
```

- 默认方法为"Holm"
- 参数p为一个p值向量
- n为比较的次数. 默认为p值的个数.

Bonferroni 法使用p值乘以比较的次数. "holm"法比Bonferroni法保守性稍微小一点. 前四个方法对阳性错误率(family wise error rate)的控制较严. 似乎没有理由使用非修正的Bonferroni法, 应该使用Holm法.

若假设检验是独立的, 或非负相关, 那么 Hochberg's and Hommel's methods 比较合适. Hommel方法比Hochberg方法要强, 但是差别很小, 且Hochberg方法计算速度快. "BH"法和"BY"法控制阴性率(false discovery rate)好一些,

下面是帮助的例子

```
> x <- rnorm(50, mean=c(rep(0,25),rep(3,25)))
>  p <- 2*pnorm( sort(-abs(x)))
>  round(p, 3)
 [1] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.001
[13] 0.002 0.002 0.002 0.002 0.003 0.003 0.005 0.006 0.012 0.020 0.023 0.035
[25] 0.048 0.096 0.103 0.108 0.141 0.191 0.208 0.220 0.261 0.288 0.333 0.399
[37] 0.409 0.452 0.496 0.572 0.577 0.581 0.588 0.598 0.646 0.744 0.776 0.846
[49] 0.868 0.985
> round(p.adjust(p), 3)
 [1] 0.000 0.000 0.001 0.002 0.003 0.005 0.007 0.007 0.013 0.019 0.053 0.054
[13] 0.064 0.067 0.070 0.077 0.087 0.087 0.165 0.195 0.346 0.594 0.632 0.951
[25] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
[37] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
[49] 1.000 1.000
```

```
>  round(p.adjust(p,"BH"), 3)
 [1] 0.000 0.000 0.000 0.000 0.001 0.001 0.001 0.001 0.002 0.002 0.006 0.006
[13] 0.006 0.006 0.006 0.007 0.007 0.007 0.014 0.016 0.027 0.047 0.049 0.073
[25] 0.095 0.184 0.190 0.193 0.242 0.318 0.335 0.343 0.395 0.423 0.475 0.552
[37] 0.552 0.594 0.636 0.679 0.679 0.679 0.679 0.679 0.718 0.809 0.826 0.882
[49] 0.886 0.985
```

30.4.7 pairwise.t.test()函数

计算多重比较, 使用p.adjust()里面的方法. 可以为"none".

```
x=rnorm(100)
y=rnorm(100)+1
z=rnorm(100)+2
data=c(x,y,z)
g=c(rep(0,100),rep(1,100),rep(2,100)) # 分组信息
> pairwise.t.test(data, g, p.adjust.method = "none")

        Pairwise comparisons using t tests with pooled SD

data:  data and g

  0      1
1 3.4e-12 -
2 < 2e-16 1.4e-09

P value adjustment method: none
> pairwise.t.test(data, g, p.adjust.method = "holm")

        Pairwise comparisons using t tests with pooled SD

data:  data and g

  0      1
1 6.7e-12 -
2 < 2e-16 1.4e-09
```

```
P value adjustment method: holm
```

30.4.8 TukyHSD法

计算 Tukey Honest Significant Differences, 即计算置信水平下的均值差值的置信区间与p值.

```
# 使用上面的数据
> TukeyHSD(aov(data~factor(g)))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = data ~ f)

$f
        diff       lwr      upr p adj
1-0 0.9355527 0.5922622 1.278843     0
2-0 1.9749450 1.6316544 2.318236     0
2-1 1.0393923 0.6961017 1.382683     0
```

30.4.9 Kurskal-Wallis-非参数方法多组比较

总体非正态分布, 或根本是有序数据(例如, 得分), 那么应该使用非参数统计的 Kurskal-Wallis 检验17.3

30.5 单因素协方差分析(ANCOVA)

参考文献 [11] 12.5.3

参考 help(rp.ancova,pac="rpanel"), 交互单因素协方差分析

参考 help(sm.ancova,pac="sm"), 非参数单因素协方差分析

在这里, 我们想考察一个因素水平的差异是否对结果变量(正态分布)均值有显著影响, 但是需要控制其它协变量(可以是连续, 也可以是分类变量). ANCOVA(one way analysis-of-covariance model)是控制潜在的混杂变量的基础上去比较2组或多组的连续结果变量均值. 这个模型叫做单向协方差模型, 也称作协方差分析模型(多重回归).

下面是y的单因素协方差分析公式，类别由A决定，协方差项为x。(统计模型一章有其它的模型21)

```
y~A+x
```

下面是一个虚拟的例子. 我们将年龄(age), 性别(sex)作为协变量, 考察用药与否(ctl)与血压(y)的关系. 其模型为

$$y = \alpha + \beta_1 ctl + \beta_2 sex + \beta_3 age + e$$

此模型考察的是控制年龄(age), 性别(sex)后用药与否(ctl)和血压的关系. (注意我们将模型自变量的顺序改变后结果会不同, 有时候甚至相反)

可以看到, 控制age, sex后, ctl的影响是显著的, 对照(ctl=0)比用药(ctl=1)要低6.68个单位. 性别和年龄的影响是不显著的(p=0.72, p=0.97), 女性(sex=0) 比男性(sex=1)平均血压要低0.06个单位, 但是年龄每增加1,平均血压下降0.13个单位.

```
# 年龄
age=sample(c(10:20),100,replace=TRUE)
# 性别
sex=sample(c(1,2),100,replace=TRUE)
# 服药与否, 前50个未服药, 后50个服药
ctl=c(rep(0,50),rep(1,50))
# 血压, 假设服药组血压高
y=round(runif(100)*40+80,1); y[51:100]=y[51:100]+10

# 控制年龄(age), 性别(sex)后用药与否(ctl)和血压的关系.
# lm(y~ctl+(age+sex)) 与写法 lm(y~ctl+age+sex) 结果一样
> lm(y~ctl+(age+sex))
```

```
Call:
lm(formula = y ~ ctl + (age + sex))

Coefficients:
(Intercept)         ctl         age         sex
  105.92229     6.67843    -0.12829     0.06079

> summary(lm(y~ctl+(age+sex)))

Call:
lm(formula = y ~ ctl + (age + sex))

Residuals:
   Min     1Q  Median     3Q    Max
-23.559 -10.051   1.038  10.055  18.448

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
(Intercept) 105.92229    6.32497  16.747  < 2e-16 ***
ctl          6.67843    2.32660   2.870  0.00504 **
age         -0.12829    0.35835  -0.358  0.72113
sex          0.06079    2.31394   0.026  0.97910
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.29 on 96 degrees of freedom
Multiple R-squared: 0.08218,    Adjusted R-squared: 0.0535
F-statistic: 2.865 on 3 and 96 DF,  p-value: 0.04066

# 这里给出了F值及其p值
> summary(aov(lm(y~ctl+(age+sex))))
            Df  Sum Sq Mean Sq F value   Pr(>F)
ctl          1  1079.1  1079.1  8.4673 0.004494 **
age          1    16.3    16.3  0.1276 0.721703
sex          1     0.1     0.1  0.0007 0.979097
Residuals   96 12234.8   127.4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
# anova()函数的结果是一样的
> anova(lm(y~ctl2+age+sex))
Analysis of Variance Table
```

```
Response: y
          Df  Sum Sq Mean Sq F value   Pr(>F)
ctl2       1  1079.1  1079.1  8.4673 0.004494 **
age        1    16.3    16.3  0.1276 0.721703
sex        1     0.1     0.1  0.0007 0.979097
Residuals 96 12234.8   127.4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

可以看到ctl(用药)的标记(0,1 还是 1,2)对分析结果无影响

```
> ctl2=ctl+1
> summary(lm(y~ctl2+age+sex))

Call:
lm(formula = y ~ ctl2 + age + sex)

Residuals:
   Min     1Q  Median     3Q    Max
-23.559 -10.051   1.038  10.055  18.448

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 99.24386    6.64577  14.933  < 2e-16 ***
ctl2         6.67843    2.32660   2.870  0.00504 **
age         -0.12829    0.35835  -0.358  0.72113
sex          0.06079    2.31394   0.026  0.97910
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.29 on 96 degrees of freedom
Multiple R-squared: 0.08218,    Adjusted R-squared: 0.0535
F-statistic: 2.865 on 3 and 96 DF,  p-value: 0.04066
```

改变顺序后结果相同(都是控制其它变量后的结果)

```
Coefficients:
(Intercept)         ctl         age         sex
  105.92229     6.67843    -0.12829     0.06079

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
(Intercept) 105.92229    6.32497  16.747  < 2e-16 ***
ctl          6.67843    2.32660   2.870  0.00504 **
age         -0.12829    0.35835  -0.358  0.72113
sex          0.06079    2.31394   0.026  0.97910
---

> lm(y~age + sex + ctl)

Call:
lm(formula = y ~ age + sex + ctl)

Coefficients:
(Intercept)         age         sex         ctl
  105.92229    -0.12829     0.06079     6.67843

> summary(lm(y~age + sex + ctl))

Call:
lm(formula = y ~ age + sex + ctl)

Residuals:
   Min     1Q  Median     3Q    Max
-23.559 -10.051   1.038  10.055  18.448

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
(Intercept) 105.92229    6.32497  16.747  < 2e-16 ***
age         -0.12829    0.35835  -0.358  0.72113
sex          0.06079    2.31394   0.026  0.97910
ctl          6.67843    2.32660   2.870  0.00504 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.29 on 96 degrees of freedom
Multiple R-squared: 0.08218,    Adjusted R-squared: 0.0535
```

```
F-statistic: 2.865 on 3 and 96 DF,  p-value: 0.04066
```

30.6 两因素方差分析

两因素方差分析又称为 double anova, two-factor anova, two-way anova. 我们需要考察结果变量(正态分布)与两个因素的关系. (需要控制其它协变量的时候使用双向协方差分析)

我们使用ANCOVA(单因素协方差中的例子)30.5, 考察性别(sex)及用药与否(ctl)与血压的关系. 统计模型为

$$y_{ijk} = a + b_i sex + c_j ctl + \gamma_{ij} + e_{ijk}$$

- a: 常数
- b_i: 常数, 代表性别的效应
- c_i: 常数, 代表用药与否的效应
- γ: 交互作用.

R公式可以这样(统计模型一章有其它的模型21)

```
y ~ A*B
y ~ A + B + A:B
y ~ B %in% A
y ~ A|B
```

y 对A 和 B 的非可加两因子方差分析模型（two factor non-additive model）。前两个公式表示相同的交叉分类设计（crossed classification），后两个公式表示相同的嵌套分类设计(nested classification)。抽象一点说，这四个公式指明同一个模型子空间。

下面是计算结果, 首先是方差分析表

```
# 列出平方和分解的值
> aov(y~ctl*sex)
Call:
   aov(formula = y ~ ctl * sex)

Terms:
                     ctl      sex   ctl:sex Residuals
Sum of Squares   1079.122    0.018    44.240 12206.861
Deg. of Freedom        1        1        1       96

Residual standard error: 11.27629
Estimated effects may be unbalanced

# 列出方差分析表, F及p值, 看到控制其它(sex和交互后)ctl的影响是显著的
> summary(aov(y~ctl*sex))
            Df  Sum Sq Mean Sq F value  Pr(>F)
ctl          1  1079.1  1079.1  8.4867 0.00445 **
sex          1 0.01849 0.01849  0.0001 0.99040
ctl:sex      1    44.2    44.2  0.3479 0.55668
Residuals   96 12206.9   127.2
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
# anova()的方差分析表, 与aov() 结果一样的
> anova(lm(y~ctl*sex))
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value  Pr(>F)
ctl        1  1079.1  1079.1  8.4867 0.00445 **
sex        1 0.01849 0.01849  0.0001 0.99040
ctl:sex    1    44.2    44.2  0.3479 0.55668
Residuals 96 12206.9   127.2
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

线性模型的结果: 得到回归系数, 总体F值2.945, p: 0.03681, 说明有一个系数(截距)显著不为0. 虽然不显著, 但是控制了其它后, 男性(sex=1)比女性(sex=0)血压平均值要高1.34单位, 而服药组

比不服药组血压平均高10.77单位

```
> summary(lm(y~sex*ctl))

Call:
lm(formula = y ~ sex * ctl)

Residuals:
   Min     1Q  Median     3Q    Max
-24.277  -9.841   1.245  10.219  17.503

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept)  102.187      4.893  20.883   <2e-16 ***
sex            1.345      3.213   0.419    0.676
ctl           10.772      7.495   1.437    0.154
sex:ctl       -2.726      4.622  -0.590    0.557
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.28 on 96 degrees of freedom
Multiple R-squared: 0.08427,    Adjusted R-squared: 0.05566
F-statistic: 2.945 on 3 and 96 DF,  p-value: 0.03681
```

30.7 两因素协方差分析

当结果变量(正态分布)可能与两个类型变量有关, 而同时需要控制一个或多个协变量(可以是连续或类型变量), 应该使用双向协方差分析.

双向协方差分析也可能表示成多重回归的特例.

这里, 我们仍然使用ANCOVA(单因素协方差中的例子)30.5, 考察性别(sex)及用药与否(ctl)与血压的关系, 但是把age作为协变量来控制. 可以看到age的影响是不显著的, $F = 0.1273, p = 0.722067$, 其系数为-0.12

```
# 方差分解情况
> aov(y~ctl*sex+age)
Call:
   aov(formula = y ~ ctl * sex + age)

Terms:
                    ctl      sex      age   ctl:sex Residuals
Sum of Squares   1079.122    0.018   16.333    43.172 12191.596
Deg. of Freedom        1        1        1        1       95

Residual standard error: 11.32840
Estimated effects may be unbalanced

# 方差分析表
> summary(aov(y~ctl*sex+age))
          Df  Sum Sq Mean Sq F value   Pr(>F)
ctl        1  1079.1  1079.1  8.4088 0.004638 **
sex        1 0.01849 0.01849  0.0001 0.990447
age        1    16.3    16.3  0.1273 0.722067
ctl:sex    1    43.2    43.2  0.3364 0.563284
Residuals  95 12191.6   128.3
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# 多重回归系数
> summary(lm(y~sex*ctl+age))

Call:
lm(formula = y ~ sex * ctl + age)

Residuals:
   Min     1Q  Median     3Q    Max
-23.877  -9.876   1.457  10.238  17.546

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept) 103.9873    7.1703  14.502   <2e-16 ***
sex          1.3612    3.2278   0.422    0.674
ctl         10.8316    7.5316   1.438    0.154
age         -0.1240    0.3597  -0.345    0.731
sex:ctl     -2.6935    4.6439  -0.580    0.563
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.33 on 95 degrees of freedom
Multiple R-squared: 0.08542,    Adjusted R-squared: 0.04691
F-statistic: 2.218 on 4 and 95 DF,  p-value: 0.07278
```

30.8 随机效应模型

包nlme, 线性与非线性混合效应模型, 函数 lme()

包lme4, 线性混合效应模型, 函数 lmer()

30.8.1 问题描述

例如, 一项研究是研究激素与疾病的关系(护士卫生研究,参考文献[11] 12.8), 从5名月经后期的女性获得血样, 然后被分为两等份, 采用双盲的方式把血样送到实验室分析. 研究的目的是判断人与人之间的差异与一个人血样中的波动各有多大. 数据如下

```
# hormone 浓度
horm=c(25.5,30.4,11.1,15.0,8.0,8.1,20.7,16.9,5.8,8.4)
# 每个人重复2次
rep=rep(c(1,2),5)
# 5个人编号
per=rbind(1:5,1:5)[1:10]

blood=data.frame(horm=horm,rep=rep,per=per)
> blood
   horm rep per
1  25.5   1   1
2  30.4   2   1
3  11.1   1   2
4  15.0   2   2
```

```
5   8.0   1   3
6   8.1   2   3
7  20.7   1   4
8  16.9   2   4
9   5.8   1   5
10  8.4   2   5

# 两次重复的均值
m=matrix(horm,nr=2)
mean=colMeans(m)
> mean
[1] 27.95 13.05  8.05 18.80  7.10

# 两次重复的差值
delta=m[2,]-m[1,]
> delta
[1]  4.9  3.9  0.1 -3.8  2.6
```

可以看到, 对于激素平均值较大的人, 其差值也较大. 即重复测量的变异程度与该人的平均值大小有关, 我们将对数据取对数, 再做分析, 这样重复测量的标准差就会独立于取对数后的平均水平[2].

30.8.2 模型与假设检验

估计人与人之间的差异及人内部的差异时, 常使用下面的模型

$$y_{ij} = \mu + \alpha_i + e_{ij}$$

这个模型与固定效应模型是一样的, 只不过对它的解释不同. 其中

- y_{ij}: 第i个人的第j重复
- α_i: 人之间(组间)差异的随机变量, 常被认为服从正态分布$N(0, \sigma_A^2)$

[2]更多数据变换见8

- e_{ij}: 人内部(组内)差异的随机变量, 互相独立, 且独立于α, 服从正态分布$N(0, \sigma^2)$

这个方程常被称为随机效应(random-effect)单向方差分析模型.

第i,j个人的均值分别是$\mu + \alpha_i$, $\mu + \alpha_j$, 故每个人的均值是不同的, 其变异性的指标为σ_A^2. 第i个人多次重复的均值为$\mu + \alpha_i$, σ^2代表其变异性.

随机效应分析的一个重要目的是检验假设σ_A^2是否异于零, 即

$$H_0 : \sigma_A^2 = 0 \quad vs \quad H_1 : \sigma_A^2 > 0$$

零假设成立表明人与人之间没有差异, 所有差异来源于人内部的差异(波动, 也叫做噪声). 如果备择假设为真, 说明人与人之间, 或组之间有真实的差异.

30.8.3 几个公式

组内平均方差的期望为

$$E(MS_W) = \sigma^2$$

组间平均方差的期望为(平衡设计, 即每组重复数相同的时候)

$$\begin{aligned} E(MS_B) &= \sigma^2 + n\sigma_A^2 \\ n &= n_1 = n_2 = \cdots = n_k = \text{每个人的重复数} \end{aligned}$$

组间平均方差的期望为(非平衡设计, 即每组重复数不全相同的时候)

$$\begin{aligned} E(MS_B) &= \sigma^2 + n_0\sigma_A^2 \\ n_0 &= (\sum_{i=1}^{k} n_i - \sum_{i=1}^{k} n_i^2 / \sum_{i=1}^{k} n_i)/(k-1) \end{aligned}$$

显然如果$n = n_1 = n_2 = \cdots = n_k$, 即每个人的重复数相同, 那么

$$n_0 = [kn - kn^2)/(kn)]/(k-1) = (kn-n)/(k-1) = n$$

一般非平衡时, $n_0 < n$, 但是差异常常不大.

σ_A^2的无偏估计为

$$\hat{\sigma}_A^2 = E(\frac{MS_B - MS_W}{n}) = \frac{\sigma^2 + n\sigma_A^2 - \sigma^2}{n} = \sigma_A^2$$

在非平衡设计中, 只要使用n_0代替n即可.

30.8.4 F检验

我们可以使用与固定效应模型相同的检验统计量

$$F = MS_B/MS_W \sim F_{k-1,N-k} \quad (H_0\text{下,即}\sigma_A^2 = 0)$$

组内平均方差

$$MS_W = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_i)^2/(N-k)$$

组间平均方差

$$MS_B = \sum_{i=1}^{k} n_i(\bar{y}_i - \bar{y})^2/(k-1)$$

其中

$$\begin{aligned}
\bar{y}_i &= \sum_{j=1}^{n_i} y_{ij}/n_i \\
\bar{y} &= \sum_{i=1}^{k}\sum_{j=1}^{n_i} y_{ij}/N = \sum_{i=1}^{k} n_i\bar{y}_i/N \\
N &= n_1 + \cdots + n_k
\end{aligned}$$

判断

$if \quad F > F_{k-1,N-k,1-\alpha}, \quad reject \ H_0 if \quad F <= F_{k-1,N-k,1-\alpha}, \quad accept \ H_0$

p值

$$p = P(F_{k-1,N-k} > F)$$

30.8.5 组内,组间平均方差的估计

估计组内方差

$$\hat{\sigma^2} = MS_W$$

估计组间方差(若小于0, 则令其等于0)

$$\hat{\sigma_A^2} = \frac{MS_B - MS_W}{n_0}$$

30.8.6 重复性研究中变异系数的估计

一般说来, 重复测量中变异系数< 20%是理想的, > 30%是不理想的. 重复测量中变异系数的定义为

$$CV = 100\% \frac{\sqrt{MS_W}}{mean \ of \ within \ group}$$

但是, 当方差随均值增加时, 更好的方法是使用下面的方法

- 对每个数取自然对数
- 计算MS_W
- $CV = 100\% \sqrt{MS_W}$

30.8.7 组内相关系数(ICC, 方差估计量分析,可靠性系数)

单向随机效应模型中, 同一个人两个重复之间的相关性称为组内相关系数(intraclass correlation coefficient, ICC), 记为ρ. 有多种方法估计组内相关系数, 最简单也是最普遍使用的方法是基于单向随机效应模型的.

$$\rho = \frac{\sigma_A^2}{\sigma^2 + \sigma_A^2}$$

点估计为

$$\hat{\rho} = max[\frac{\hat{\sigma}_A^2}{\hat{\sigma}^2 + \hat{\sigma}_A^2}, 0]$$

区间估计为

$$c1 = max[\frac{F/F_{k-1,N-k,1-\alpha/2} - 1}{n_0 + F/F_{k-1,N-k,1-\alpha/2} - 1}, 0]$$

$$c2 = max[\frac{F/F_{k-1,N-k,1-\alpha/2}}{n_0 + F/F_{k-1,N-k,1-\alpha/2}}, 0]$$

这个分析也叫做方差估计量分析(analysis-of-variance estimator).

组内相关系数也常常理解为可靠性的一个度量, 有时候也称为可靠性系数(reliability coefficient).

解释

- $\rho < 0.4$: 重复性很差
- $0.4 <= \rho < 0.75$: 重复性中等
- $\rho >= 0.75$: 重复性很好

包multilevel函数ICC1, 计算组内相关系数(重复性), ICC2计算组之间的可靠性(reliability).(详细用法参考31.11)

```
> library(multilevel)
> res=aov(horm~as.factor(per),blood)
> summary(res)
             Df  Sum Sq Mean Sq F value   Pr(>F)
as.factor(per)  4 2.65775 0.66444  22.146 0.002221 **
Residuals       5 0.15001 0.03000
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
# 组内(人内部)相关系数, 即可重复性很大. 即组内部的方差有91.36可以被
> ICC1(res)
[1] 0.9135923
# 组间(人之间)的相关系数也很大, 表示人之间的均值可以很好的区分
> ICC2(res)
[1] 0.9548453
```

30.8.8 例子

参考文献 [27] 10.1

nlme 包的 lme() 函数与 lme4 包的 lmer() 函数计算混合效应模型(固定+随机效应).

注意, 与SAS计算结果中的F值是不同的.

随机效应因素放在竖线后面, 写法见例子.

随机效应的结果主要看Random effects部分, nlme包的lme()函数给出了标准差, 但是可以容易的计算出方差.

- 组间变异 $\sigma_A^2 = 0.5632196^2 = 0.3172163$
- 组间平均方差 $MS_B = n_0\sigma_A^2 = 2 * 0.5632196^2 = 0.6344326$

- 组内变异 $\sigma^2 = 0.1732123^2 = 0.0300025$
- 组内平均变异就是 $MS_W = \sigma^2 = 0.0300025$
- 变异系数$100\%\sigma = 17.32123\%$
- 组内相关系数的点估计 $\rho = \frac{\sigma_A^2}{\sigma^2+\sigma_A^2} = 0.317/(0.317+0.030) = 0.914$
- 组内相关系数的区间估计, 其中$F = 99.27, qf(0.975,4,5) = 7.387886, qf(0.025,4,5) = 1.107$, 那么$c_1 = max[(99.27/7.39-1)/(2+(99.27/7.39-1)),0] = 0.86$, $c_2 = (99.27/1.107-1)/(2+(99.27/1.107-1)) = 0.9779432$ [3]

```
# hormone 浓度
horm=c(25.5,30.4,11.1,15.0,8.0,8.1,20.7,16.9,5.8,8.4)
# 每个人重复2次
rep=rep(c(1,2),5)
# 5个人编号
per=rbind(1:5,1:5)[1:10]
# 数据框内, hormone 水平取对数值
blood=data.frame(horm=log(horm),rep=rep,per=per)

library(nlme)
ll=lme(horm~1,random=~1|per,data=blood)
> summary((lme(horm~1,random=~1|per,data=blood)))
Linear mixed-effects model fit by REML
 Data: blood
      AIC      BIC    logLik
  14.67583 15.26751 -4.337916

Random effects:
 Formula: ~1 | per
        (Intercept)  Residual
StdDev:   0.5632196 0.1732123

Fixed effects: horm ~ 1
             Value Std.Error DF  t-value p-value
(Intercept) 2.568296 0.2577664  5 9.963656   2e-04
```

[3]由于与SASF值计算不同, 区间也不同. SAS的$F = 22.15$

```
Standardized Within-Group Residuals:
      Min        Q1       Med        Q3       Max
-1.2321300 -0.4460533 -0.1258207  0.6986361  0.9061354

Number of Observations: 10
Number of Groups: 5

# 可以使用 VarCorr 得到方差与标准差, 结果与lmer()一样.
> VarCorr(ll)
per = pdLogChol(1)
          Variance   StdDev
(Intercept) 0.31721636 0.5632196 # 组间变异性
Residual    0.03000249 0.1732123 # 组内变异性

# F值, 注意与SAS结果不同.
> anova(ll)
          numDF denDF  F-value p-value
(Intercept)     1     5 99.27445   2e-04

> coef(ll)
$per
  (Intercept)
1    3.292320
2    2.557985
3    2.107447
4    2.912448
5    1.971279
```

lme4包的函数lmer()同时给出了方差和标准差(Random effects部分下面的 Variance Std.Dev.) 如果固定因素多于一个, 还给出固定效应因素的协方差矩阵.

```
library(lme4)
> lmer(horm~1+(1|per),data=blood)
Linear mixed model fit by REML
Formula: horm ~ 1 + (1 | per)
   Data: blood
   AIC   BIC logLik deviance REMLdev
```

```
 14.68 15.58 -4.338    7.749   8.676
Random effects:
 Groups   Name        Variance Std.Dev.
 per      (Intercept) 0.317209 0.56321
 Residual             0.030003 0.17321
Number of obs: 10, groups: per, 5

Fixed effects:
            Estimate Std. Error t value
(Intercept)   2.5683     0.2578   9.964

> coef(lmer(horm~1+(1|per),data=blood))
  (Intercept)
1    3.292321
2    2.557985
3    2.107446
4    2.912449
5    1.971278
```

Chapter 31

一致性(agreement)估计

本部分来自参考文献[28]《Multilevel Modeling in R》的翻译.

主要使用的包为: base, nlme, multilevel

其内容还有: 普通最小二乘法(Ordinary Least Square,简称OLS),OLS方法使用线性模型lm(), 和随机模型lme(), 参考回归与方差分析部分.

增长模型: 是Solow于1956年首次创立的，用来说明储蓄、资本积累和增长之间的关系。自建立以来，这一模型一直是分析以上三个变量关系的主要理论框架。参考文献《Multilevel Modeling in R》[28] chapter 4. 主要使用 lme() 计算, 6个步骤

31.1 Agreement(一致性相关系数, CCC)

包multilevel里有几个函数可以估计推测一致性指标(agreement indices). 函数为rwg, rwg.j, rwg.sim, rwg.j.sim,rwg.j.lindell, ad.m, ad.m.sim rgr.agree. 具体见函数帮助.

另外, 包agreement的函数lin.simulation()使用模拟方法计算两个方法的一致性(参考文献[24]). lin.simulation()函数中两个方法X,Y对测量一致性相关系数(Concordance Correlation Coefficient,

CCC) 是Pearson($\rho_{x,y}$)相关系数与精确度(C_b)的乘积, (详细见函数帮助及其参考文献Lin,1989, Lin2002文献给出了其它几个指标来度量一致性)

$$\begin{aligned} C_b &= 2\sigma_x\sigma_y/[\sigma_x^2+\sigma_y^2+(\mu_x-\mu_y)^2] \\ CCC &= \rho_{xy}C_b \end{aligned}$$

31.2 一致性度量

此处的一致性就是方差分析中随机效应模型中的组内相关系数. 其中r_{wg}度量单独数据组内一致性(within group agreement measure for single item measures, James et al. (1984), 具体文献见函数帮助), 默认期望的随机方差(Expected Random Variance)为2

$$rwg = 1-(ObservedGroupVariance/ExpectedRandomVariance)$$

一般, 一致性系数> 0.7比较好, 否则就比较差.

31.3 估计EV

期望随机方差(Expected Random Variance, EV)的估计是这样的(参考help(rwg)), 默认的$EV=2$是按照分5个等级计算的(例如, Strongly Disagree, Disagree, Neither, Agree, Strongly Agree), 如果不是5个组, 记A为组数, 那么A的方差基于矩形分布(rectangular distribution)

$$EV=(A^2-1)/12$$

31.4 例子

```
data(bhr2000,package="multilevel")
RWG.RELIG<-rwg(bhr2000$RELIG,bhr2000$GRP,ranvar=2)
# 共94组, 查看前10组
> RWG.RELIG[1:10,]
```

```
   grpid       rwg gsize
1      1 0.11046172    59
2      2 0.26363636    45
3      3 0.21818983    83
4      4 0.31923077    26
5      5 0.22064137    82
6      6 0.41875000    16
7      7 0.05882353    18
8      8 0.38333333    21
9      9 0.14838710    31
10    10 0.13865546    35
>  summary(RWG.RELIG)
     grpid        rwg            gsize
 1      : 1   Min.   :0.0000   Min.   :  8.00
 10     : 1   1st Qu.:0.1046   1st Qu.: 29.50
 11     : 1   Median :0.1899   Median : 45.00
 12     : 1   Mean   :0.1864   Mean   : 54.55
 13     : 1   3rd Qu.:0.2630   3rd Qu.: 72.50
 14     : 1   Max.   :0.4328   Max.   :188.00
 (Other):93
```

```
# 对rwg排序, 或查看直方图也比较有用
> sort(RWG.RELIG[,2])
> hist(RWG.RELIG[,2])
```

下面来估计工作时间的r_{wg}, 我们需要改变期望随机方差(expected random variance, EV). 工作时间被要求是11个等级(11-point item, 即按照工作时间分为11等级), 因此EV基于矩形分布(rectangular distribution), 故$\sigma^2_{EV} = (11^2 - 1)/12 = 10.00$.

```
# 工作时间等级数
> length(unique(bhr2000$HRS))
[1] 11
# 计算不同组GRP的工作时间HRS的一致性, 0.73>0.7表明一致性比较好
> RWG.HRS<-rwg(bhr2000$HRS,bhr2000$GRP,ranvar=10.00)
> mean(RWG.HRS[,2])
[1] 0.7353417
```

31.5 rwg.j()

函数rwg.j()与rwg()几乎一样, 但是估计多个item的一致性. 第一个参数为矩阵, 每列是一个item, 每行是一个观察(response), 默认使用5个等级. 下面看到2到12列总的一致性系数很高

```
> RWGJ.LEAD<-rwg.j(bhr2000[,2:12],bhr2000$GRP,ranvar=2)
> summary(RWGJ.LEAD)
    grpid       rwg.j          gsize
 1      : 1   Min.   :0.7859   Min.   :  8.00
 10     : 1   1st Qu.:0.8708   1st Qu.: 29.50
 11     : 1   Median :0.8925   Median : 45.00
 12     : 1   Mean   :0.8876   Mean   : 54.55
 13     : 1   3rd Qu.:0.9088   3rd Qu.: 72.50
 14     : 1   Max.   :0.9440   Max.   :188.00
 (Other):93
```

31.6 rwg.j.lindell()

一般认为(rwg, rwg.j), 随着等级数(item)的增加, 偏差也会增大, 这是基于 Spearman-Brown reliability estimator. 但是t Lindell and colleagues等认为这个估计好像没有理论基础, 即没有理由认为随着等级数(item)的增加, 偏差会增大. 故Lindell and colleagues等发展了一个方法, 使用平均方差代替方差. 函数rwg.j.lindell()计算此定义的一致性. 可以看到结果(均值为0.43)明显低于rwg.j方法(均值0.89).

```
RWGJ.LEAD.LIN<-rwg.j.lindell(bhr2000[,2:12],
   bhr2000$GRP,ranvar=2)
> summary(RWGJ.LEAD.LIN)
    grpid     rwg.lindell        gsize
 1      : 1   Min.   :0.2502   Min.   :  8.00
 10     : 1   1st Qu.:0.3799   1st Qu.: 29.50
 11     : 1   Median :0.4300   Median : 45.00
 12     : 1   Mean   :0.4289   Mean   : 54.55
```

```
13     : 1   3rd Qu.:0.4753   3rd Qu.: 72.50
14     : 1   Max.   :0.6049   Max.   :188.00
(Other):93
```

31.7 置信区间估计

基本的思想是基于一个已知的分布(一般是均匀分布), 随机抽样, 重复估计r_{wg},

Cohen et al., (2001)发现, 组的大小(group size)与item数量对r_{wg}影响很大,

2003, Dunlap and colleagues发现组大小和每个item的等级个数(response number)对r_{wg}影响很大. 在5个等级个数(例如, Strongly Disagree, Disagree, Neither, Agree, Strongly Agree),95%置信区间从3组的1.0变到150组的0.12.

rwg.sim()提供Dunlap and colleagues(2003)方法. 对于组数为10的item, 本身5个等级(response)的变量, 我们可以这样使用函数

```
# run for long time (>30s)
> RWG.OUT<-rwg.sim(gsize=10, nresp=5, nrep=10000)
> summary(RWG.OUT)
$rwg
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.1204  0.2000  0.8667

$gsize
[1] 10

$nresp
[1] 5

$nitems
[1] 1

$rwg.95
```

```
[1] 0.5277778
```

95%置信区间大小为0.53, 其它值可以得到稍微不同的结果.

还提供了一个泛型函数 quantile(agree.sim) 来计算其它置信水平下的置信区间大小.

```
> quantile(RWG.OUT,c(.90,.95,.99))
  quantile.values confint.estimate
1            0.90       0.4166667
2            0.95       0.5277778
3            0.99       0.6666667
```

函数rwg.j.sim()基于Cohen et al. (in press)的工作(扩展了 Dunlap et al., (2003)), 考察多个item和组数(group size), 等级数(response number).

一般模拟采样的次数大于10000.

Cohen et al., (2001)的工作表明相关的item与不相关的item的一致性估计结果差不多, 但是相关的情况更可靠, 推荐使用. 忽略参数 itemcors 表示假设item之间独立.

下面的例子是15个组, 7个item, 5个等级(response, $A = 5$). 95%置信区间上限为$0.54 < 0.7$, 那么0.7可能太严格了. 基于此, 我们可以调整0.55为两个一致性有区别的显著性阈值($p < 0.05$).

```
> RWG.J.OUT<-rwg.j.sim(gsize=15,nitems=7,nresp=5,nrep=1000)
> summary(RWG.J.OUT)
$rwg
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.000000 0.000000 0.009447 0.155400 0.314100 0.753000

$gsize
[1] 15
```

```
$nresp
[1] 5

$nitems
[1] 1

$rwg.95
[1] 0.5425764
```

下面是一个实际的例子, 演示如何使用rwg.j.sim()来计算数据lq2002三个item的平均的一致性, 结果均值为0.58. 我们要考察0.58是否显著的一致. 模拟的阈值为$0.34 < 0.58$, 那么其item可以说是一致的($p = 0.05$)

```
> data(lq2002,package="multilevel")
> RWG.J<-rwg.j(lq2002[,c("TSIG01","TSIG02","TSIG03")],
+ lq2002$COMPID,ranvar=2)
> summary(RWG.J)
    grpid       rwg.j          gsize
 10    : 1   Min.   :0.0000   Min.   :10.00
 13    : 1   1st Qu.:0.5099   1st Qu.:18.00
 14    : 1   Median :0.6066   Median :30.00
 15    : 1   Mean   :0.5847   Mean   :41.67
 16    : 1   3rd Qu.:0.7091   3rd Qu.:68.00
 17    : 1   Max.   :0.8195   Max.   :99.00
 (Other):43

# 模拟, 阈值为0.34. 需要MASS包mvrnorm()函数产生多元正态分布随机数
> library(MASS)
>  RWG.J.OUT<-rwg.j.sim(gsize=42,nitems=3,nresp=5,
+   itemcors=cor(lq2002[,c("TSIG01","TSIG02","TSIG03")]),
+   nrep=1000)
>  summary(RWG.J.OUT)
$rwg
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.09055 0.17340 0.57660

$gsize
```

```
[1] 42

$nresp
[1] 5

$nitems
[1] 1

$rwg.95
[1] 0.3406643
```

31.8 平均偏差(AD)一致性估计

Burke, Finkelstein and Dusig (1999)建议使用平均偏差(Average Deviation, AD)指标度量祖辈一致性. Cohen et al., (in press)也称为Mean, Median Average Deviation(MAD). 每个组的AD为

$$AD = \sum |x_{ij} - X_j|/N$$

N为组内的样本数. 每个item的AD计算后, 取平均作为最后的AD值.

如果AD值小于$A/6$那么表明一致性好, 否则不好. (A为等级数, 例如A=5, Strongly Disagree, Disagree, Neither, Agree, Strongly Agree. $A/6 = 0.83$),

下面的AD值为$0.86 > 0.83$, 说明一致性不好.

```
data(bhr2000)
AD.VAL <- ad.m(bhr2000[, 2:12], bhr2000$GRP)
# 共99个
> AD.VAL
  grpid     AD.M gsize
1     1 0.8481366    59
2     2 0.8261279    45
3     3 0.8809829    83
```

```
4     4 0.8227542   26
5     5 0.8341355   82
>  mean(AD.VAL[,2:3])
     AD.M     gsize
 0.8690723 54.5454545
```

如果使用中位数Median, 结果会不同

```
>  AD.VAL <- ad.m(bhr2000[, 2:12], bhr2000$GRP,type="median")
>  mean(AD.VAL[,2:3])
     AD.M     gsize
 0.8297882 54.5454545
```

只计算一个item的一致性, 例如工时间, 里面有11个等级, 故阈值为$11/6 = 1.83$, 下面计算AD值为$1.25 < 1.83$, 说明一致性还可以.

```
>  AD.VAL.HRS <- ad.m(bhr2000$HRS, bhr2000$GRP)
>  mean(AD.VAL.HRS[,2:3])
    AD.M    gsize
 1.249275 54.545455
```

31.9 AD显著性检验

函数ad.m.sim()基于Cohen et al. (in press) , Dunlap et al., (2003)的工作. 原理也是基于均匀分布采样. 下面的例子表明, 如果有99%的把握说一致性是显著的, 那么AD值要小于1.108

```
library(MASS)
AD.SIM<-ad.m.sim(gsize=55,nresp=5,
   itemcors=cor(bhr2000[,2:12]),type="mean",nrep=1000)
> summary(AD.SIM)
$ad.m
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.088   1.180   1.207   1.207   1.234   1.322

$gsize
[1] 55

$nresp
[1] 5

$nitems
[1] 11

$ad.m.05
[1] 1.141397

$pract.sig
[1] 0.8333333

> quantile(AD.SIM,c(.10,.05,.01))
  quantile.values confint.estimate
1          0.10        1.156424
2          0.05        1.141397
3          0.01        1.108279
```

31.10 随机组采样方法

随机组采样方法(Random Group Resampling)函数为rgr.agree(), 类似rwg.j.sim(), 区别是rgr.agree 使用实际的组数据, 而rwg.j.sim使用期望的均匀分布.下面例子结果得到随机采样的组内方差为3.32, 其标准差为0.79, 真实的组内的方差小于2.65, $z = -8.4 < -1.96 = z_{0.025}$, 显示真实的组内是一致的.

```
> RGR.HRS<-rgr.agree(bhr2000$HRS,bhr2000$GRP,1000)
> summary(RGR.HRS)
$`Summary Statistics for Random and Real Groups`
        随机采样方差                    真实组内方差
N.RanGrps Av.RanGrp.Var SD.Rangrp.Var Av.RealGrp.Var   Z-value
```

```
[1,]      990     3.322136    0.7927865      2.646583 -8.478535

$`Lower Confidence Intervals (one-tailed)`
   0.5%      1%    2.5%      5%     10%
1.314331 1.555556 1.935742 2.157274 2.405185

$`Upper Confidence Intervals (one-Tailed)`
    90%     95%   97.5%     99%   99.5%
4.278615 4.628046 4.978991 5.718599 6.353275
```

而且看到, 5%的随机的值小于2.16, 那么我们有95%的把握说组内的方差小于2.16.

31.11 组内相关系数(ICC)

函数ICC1和ICC2基于Bartko, (1976), James (1982), and Bliese (2000)的描述计算组内相关系数(intraclass correlation coefficient, ICC)(详细解释参考30.8.7)

其中ICC1等价于随机模型中个体水平方差被组内(个体内部)解释的程度. 结果为17%表明工作时间的方差有17%可以被组内(个体内部)解释. 即组内相关性(重复性)不好, 组内差异大.

ICC2组均值的可靠性(区分度,), 为0.92表明组之间的平均工作时间可以很好的区分. reliability of the group means.

```
> data(bhr2000)
> hrs.mod<-aov(HRS~as.factor(GRP),data=bhr2000)
> summary(hrs.mod)
              Df  Sum Sq Mean Sq F value    Pr(>F)
as.factor(GRP)   98  3371.4    34.4  12.498 < 2.2e-16 ***
Residuals     5301 14591.4     2.8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# 0.17表明组内差异大
> ICC1(hrs.mod)
```

```
[1] 0.1741008
# 0.92表明组间可以被工作时间平均值很好的区分
> ICC2(hrs.mod)
[1] 0.9199889



data(bhr2000)
graph.ran.mean(bhr2000$HRS, bhr2000$GRP, nreps=1000,
   limits=c(8,14),bootci=TRUE)
```

Chapter 32

一些非标准模型

在结束本章前，我们简单提一下R里面某些用于某些特殊回归和数据分析问题的工具.

- 混合模型（Mixed models）。用户捐献包nlme里面提供了函数lme() 和nlme()。这些函数可以用于混合效应模型（mixed-effects models）的线性和非线性回归。也就是说在线性和非线性回归中，一些系数受随机因素的影响。这些函数需要充分利用公式来描述模型。

- 局部近似回归(Local approximating regressions)。函数loess() 利用局部加权回归进行一个非参数回归。这种回归对显示一组凌乱数据的趋势和描述大数据集的整体情况非常有用。

  函数loess 和投影跟踪回归（projection pursuit regression）的代码一起放在标准包stats 中。

- 稳健回归(Robust regression)。有多个函数可以用于拟合回归模型，同时尽量不受数据中极端值的影响。在推荐包MASS 中的函数lqs为高稳健性的拟合提供了最新的算法。另外，稳健性较低但统计学上高效的方法同样可以在包MASS 中得到，如函数rlm。

- 累加模型(Additive models)。这种技术期望可以通过决定变量的平滑累加函数（smooth additive function）构建回归函数。一般来说，每个决定变量都有一个平滑累加函数。

用户捐献的包acepack 里面的函数avas 和ace 以及包mda 里面的函数bruto 和mars 为这种技术提供了一些例子。这种技术的一个扩充是用户捐献包gam 和mgcv 里面实现的广义累加模型。

- 树型模型(Tree-based models)。除了利用外在的全局线性模型预测和解释数据，还可以利用树型模型递归地在决定性变量的判断点上将数据的分叉分开。这样做会把数据最终分成几个不同组，使得组内尽可能相似而组间尽可能差异。这样常常会得到一些其他数据分析方法不能产生的的信息。模型可以用一般的线性模型形式指定。该模型拟合函数是tree()，而且许多泛型函数，如plot() 和text() 都可以很好的用于树型模型拟合结果的图形显示。R 里面的树型模型函数可以通过用户捐献的包rpart 和tree 得到。

Part V

判别,聚类,因子分析等

此部分主要参考”统计建模与R软件” [15] 及”生物数学” [8] 相关部分.

Chapter 33

数据的中心化和标准化

33.1 中心化

n 个样本的 p 维向量中心化为

$$x'_{ij} = x_{ij} - \bar{x_j}, \quad i = 1, \cdots, n \quad j = 1, \cdots, p$$

其中

$$\bar{x_j} = \frac{1}{n}\sum_{k=1}^{n} x_{kj}$$

变换后均值为 0, 方差矩阵不变.

```
> x=1:10
> mean(x)
[1] 5.5
> var(x)
[1] 9.166667

# 中心化
> y=scale(x,scale=F);y
      [,1]
 [1,] -4.5
```

```
 [2,] -3.5
 [3,] -2.5
 [4,] -1.5
 [5,] -0.5
 [6,]  0.5
 [7,]  1.5
 [8,]  2.5
 [9,]  3.5
[10,]  4.5
attr(,"scaled:center")
[1] 5.5
> mean(y)
[1] 0
> var(y)
        [,1]
[1,] 9.166667
```

33.2 标准化

标准化也叫做 z-score 规范化（零均值规范化）.

$$x'_{ij} = \frac{x_{ij} - \bar{x_j}}{s_j}, \quad i = 1, \cdots, n \quad j = 1, \cdots, p$$

其中

$$\bar{x_j} = \frac{1}{n} \sum_{k=1}^{n} x_{kj}$$

$$s_j = \frac{1}{n-1} \sum_{k=1}^{n} (x_{kj} - \bar{x_j})^2$$

变换后均值为 0, 方差为 1.

R 函数 scale() 执行此变换.

```
> x=1:10
> mean(x)
[1] 5.5
> var(x)
[1] 9.166667

# 手工计算
> x.zscore=(x-mean(x))/sd(x)
> mean(x.zscore)
[1] 0
> sd(x.zscore)
[1] 1
> x.zscore
 [1] -1.4863011 -1.1560120 -0.8257228 -0.4954337 -0.1651446  0.1651446
 [7]  0.4954337  0.8257228  1.1560120  1.4863011

# 标准化
> y=scale(x);y
           [,1]
 [1,] -1.4863011
 [2,] -1.1560120
 [3,] -0.8257228
 [4,] -0.4954337
 [5,] -0.1651446
 [6,]  0.1651446
 [7,]  0.4954337
 [8,]  0.8257228
 [9,]  1.1560120
[10,]  1.4863011
attr(,"scaled:center")
[1] 5.5
attr(,"scaled:scale")
[1] 3.027650
> mean(y)
[1] 0
> var(y)
     [,1]
[1,]    1

# center 相当于 mean(x)
> y=scale(x,center=0,scale=F);y
```

```
     [,1]
 [1,]    1
 [2,]    2
 [3,]    3
 [4,]    4
 [5,]    5
 [6,]    6
 [7,]    7
 [8,]    8
 [9,]    9
[10,]   10
attr(,"scaled:center")
[1] 0
```

33.3 极差正规化(最小-最大规范化)

最小-最大规范化对原始数据进行线性变换。假定minA和maxA分别为数据A的最小值和最大值。最小-最大规范化通过计算

$$x' = \frac{x - min_A}{max_A - min_A}(newmax_A - newmin_A) + newmin_A$$

将A的值x映射到区间$[newmin_A, newmax_A]$.

映射到区间 $[0,1]$ 称为极差正规化

最小-最大规范化保持原始数据值之间的联系。如果今后的输入落在A的原始数据值域之外，该方法将面临“越界”错误。下面的例子把x映射到$[0,1]$之间

```
> x=1:10
> x1=(x-min(x))/(max(x)-min(x)) *(1-0)+0
> x1
 [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
 [8] 0.7777778 0.8888889 1.0000000
```

33.4 极差标准化

$$x' = \frac{x - mean(x)}{max_X - min_X}$$

变换后均值为 0, 极差为 1

33.5 小数定标规范化

小数定标规范化通过移动属性A的小数点位置进行规范化。小数点的移动位数依赖于A的最大绝对值. 由下式计算：是使得$Max(|v'|) < 1$的最小整数。假定A的取值由-986～917。A的最大绝对值为986。使用小数定标规范化，用1 000（即i = 3）除每个值，这样，－986规范化为－0.986，而917被规范化为0.917。

```
> x=rnorm(10)*1000
> x
 [1]  687.82463 -168.41964  -56.08794 -880.85248 -910.98267 1882.82441
 [7] -978.97664  736.98754 -1723.98835 -384.87254
> i=ceiling(log(max(abs(x)),10)) # 小数定标的指数
> i
[1] 4
> x/10^i
 [1]  0.068782463 -0.016841964 -0.005608794 -0.088085248 -0.091098267
 [6]  0.188282441 -0.097897664  0.073698754 -0.172398835 -0.038487254
```

注意，规范化将原来的数据改变，特别是上面的后两种方法。有必要保留规范化参数（如均值和标准差，如果使用z-score规范化），以便将来的数据可以用一致的方式规范化。

33.6 正则化(normalize)

变量除以它的范数, 使平方和等于 1.

```
> x=1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> x.nor=x/sqrt(sum(x^2))
> x.nor
 [1] 0.05096472 0.10192944 0.15289416 0.20385888 0.25482360 0.30578831
 [7] 0.35675303 0.40771775 0.45868247 0.50964719

# 平方和等于 1.
> sum(x.nor^2)
[1] 1

# 和与方差皆未知
> sum(x.nor)
[1] 2.803060
> sd(x.nor)
[1] 0.1543033
```

Chapter 34

距离系数

参考 [8] 3.1 距离系数

34.1 基本性质

距离系数一般应该满足下面三个基本性质

1. $d_{AB} >= 0$, 当且仅当 $A = B$ 时成立
2. $d_{AB} = d_{BA}$
3. $d_{AB} <= d_{AC} + d_{CB}$ (三角不等式)

有时候第三条修改为

- $d_{AB} <= max(d_{AC}, d_{CB})$

比原来的三角不等式要强, 因为 $max(d_{AC}, d_{CB}) <= d_{AC} + d_{CB}$

R 函数计算各种距离, 包括 ”’euclidean”’, ”’maximum”’, ”’manhattan”’, ”’canberra”’,”’binary”’ or ”’minkowski”’.

注意: 各个系数如果求和之后除以 p 再进行开方运算, 就变成平均 XX 距离系数. 例如平均欧氏距离变为

$$d(x,y)=\sqrt{\frac{1}{n}[(x_1-y_1)^2+\cdots+(x_p-y_p)^2]}=[\frac{1}{n}\sum_{i=1}^{p}(x_i-y_i)^2]^{\frac{1}{2}}$$

但是 R 并没有平均距离的函数.

34.2 绝对距离(曼哈顿距离, absolute distance)

$$d(x,y)=\sum_{i=1}^{p}|x_i-y_i|$$

```
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> x
     [,1] [,2]
[1,]    0    0
[2,]    1    1
[3,]    2    2
[4,]    3    3

> dist(x,diag=T,method="manhattan")
  1 2 3 4
1 0
2 2 0
3 4 2 0
4 6 4 2 0
```

34.3 欧氏距离(Euclidean distance)

p 维空间的两点 $x=(x_1,\cdots,x_p)^T, y=(y_1,\cdots,y_p)^T$, 其欧氏距离系数为

$$d(x,y)=\sqrt{(x_1-y_1)^2+\cdots+(x_p-y_p)^2}=[\sum_{i=1}^{p}(x_i-y_i)^2]^{\frac{1}{2}}$$

```
# 默认计算欧氏距离系数
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> dist(x,method = "euclidean", diag=T, upper = FALSE)
         1        2        3        4
1 0.000000
2 1.414214 0.000000
3 2.828427 1.414214 0.000000
4 4.242641 2.828427 1.414214 0.000000
```

34.4 Minkowski 距离(明氏距离)

$$d(x,y)=\sqrt[r]{|x_1-y_1|^r+\cdots+|x_p-y_p|^r}=[\sum_{i=1}^{p}|x_i-y_i|^r]^{\frac{1}{r}}$$

其中$r>0$. 这个系数常常被化学分类学使用, 比较两个薄层层析的差异. r 充分小时, 对较小的差异敏感, 故适合差异较小的分类单位之间建立相似性比较.

$r=1$ 时转化为 曼哈顿距离, $r=2$ 时转化为欧氏距离.

```
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> dist(x,diag=T,method="minkowski",p=0.5)
  1  2  3  4
```

```
1  0
2  4  0
3  8  4  0
4 12  8  4  0
> dist(x,diag=T,method="minkowski",p=3)
         1        2        3        4
1 0.000000
2 1.259921 0.000000
3 2.519842 1.259921 0.000000
4 3.779763 2.519842 1.259921 0.000000
```

34.5 Chebyshev 距离

$$d_{ij}(\infty) = max_{1<=k<=p}|x_{ik} - x_{jk}|$$

是 Minkowski 距离 $r \to \infty$ 时的情况

dist() 函数 method=”maximum” 是计算 Chebyshev 距离.

```
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> x
     [,1] [,2]
[1,]    0    0
[2,]    1    1
[3,]    2    2
[4,]    3    3
> dist(x,diag=T,method="maximum")
  1 2 3 4
1 0
2 1 0
3 2 1 0
4 3 2 1 0
```

34.6 Canberra 距离

实际上是 Lance 距离的扩展, 不要求 $x_{ij} > 0$

$$d(x,y) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{|x_i + y_i|}$$

```
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> dist(x,diag=T,method="canberra")
        1         2         3         4
1 0.0000000
2 2.0000000 0.0000000
3 2.0000000 0.6666667 0.0000000
4 2.0000000 1.0000000 0.4000000 0.0000000
```

34.7 分离系数

与 Canberra 距离系数类似

$$d(x,y) = [\sum_{i=1}^{p} (\frac{x_i - y_i}{x_i + y_i})^2]^{\frac{1}{2}}$$

34.8 Lance 和 Williams 距离

实际上是 Canberra 距离的特殊形式.

$$d_{ij}(L) = \sum_{k=1}^{p} \frac{|x_{ik} - x_{jk}|}{x_{ik} + x_{jk}}$$

其中

$$\begin{aligned} x_{ij} &> 0 \\ i &= 1, 2, \cdots, n \\ j &= 1, \cdots, p \end{aligned}$$

用法使用 method="canberra" 即可.

34.9 Mahalanobis distance(马氏距离)

参考

- [15] 8.1 判别分析
- http://en.wikipedia.org/wiki/Mahalanobis_distance

设总体 $X = [x_{ij}]_{n*p}$ 为 p 维空间中的 n 个点, 均值为 $\mu = (\mu_1, \mu_2, \cdots, \mu_p)^T$), 协方差矩阵 $\sum$ 为 $p*p$ 的方阵. 则 p 维空间中一个样本点 $x = (x_1, \cdots, x_p)^T$ 与总体 X 的 Mahalanobis 距离为

$$d(x, X) = \sqrt{(x-\mu)^T \sum^{-1} (x-\mu)}$$

其中 $\sum^{-1}$ 为 $\sum$ 的逆矩阵.

实际上是对 x 标准化.

总体内两个点 x,y (即服从均值 μ, 协方差矩阵方差 $\sum$) 之间的 Mahalanobis 距离定义为

$$d(x, y) = \sqrt{(x-y)^T \sum^{-1} (x-y)}$$

实际上是标准化了的 x,y 之间的距离.

R 函数 mahalanobis() 计算 Mahalanobis 距离. 但是未开方用法为

```
mahalanobis(x, center, cov, inverted=FALSE, ...)
```

其中

- x 为 p 维的向量(一个点)或 p 列的矩阵(多个点).
- center 为 p 维向量, 代表总体均值. 如果给出的不是均值, 而是另外一个 p 维向量 y, 则函数计算的就是 x,y之间的 Mahalanobis 距离
- cov 代表 $p*p$ 的协方差矩阵
- inverted=TRUE 代表给出的协方差矩阵已经求逆. 否则函数会计算其逆.

返回 x 的每个点与 X(均值) 的 Mahalanobis 距离或 x,y 之间的 Mahalanobis 距离(未开方的)

考虑一维的例子, 实际上就是 x 的标准化 (函数计算的是未开方的结果)

```
> X=c(1:10)
> X
 [1]  1  2  3  4  5  6  7  8  9 10
> mu=mean(X); mu
[1] 5.5
> cov=var(X); cov
[1] 9.166667

> dist.mahalanobis(0,mu,cov)
    [,1]
[1,]  3.3
> mahalanobis(0,mu,cov)
[1] 3.3
# 实际上是标准化
```

```
> (0-mu)*(1/cov)*(0-mu)
[1] 3.3
> (0-mu)^2/cov
[1] 3.3

#下面为两个总体,查看样本x=15与两个总体的Mahalanobis距离
> X1=c(1:10)
> X2=c(11:20)
> mu1=mean(X1)
> mu2=mean(X2)
> cov1=var(X1)
> cov2=var(X2)

> mahalanobis(15,mu1,cov1)
[1] 9.845455
> mahalanobis(15,mu2,cov2)
[1] 0.02727273
```

下面是二维的例子(函数计算的是未开方的结果)

```
> X=matrix(c(1:10,1:5,10:6),ncol=2)
> X
      [,1] [,2]
 [1,]    1    1
 [2,]    2    2
 [3,]    3    3
 [4,]    4    4
 [5,]    5    5
 [6,]    6   10
 [7,]    7    9
 [8,]    8    8
 [9,]    9    7
[10,]   10    6

> a=colMeans(X)[1]; a
[1] 5.5
> b=colMeans(X)[2]; b
```

```
[1] 5.5

# 绘制X
> plot(X)
# 均值点 (center)
> points(a,b,col='red')

# 编制点 x 与总体 X 的距离函数
# 如果 mu 给出的不是均值, 而是另外一个 p 维向量 y,
# 则函数计算的就是 x,y之间的 Mahalanobis 距离
dist.mahalanobis<-function(x,mu,cov){
   r <- (x-mu)%*%solve(cov)%*%(x-mu)
   r
}

# 以 X 为总体, 计算均值与协方差矩阵
> mu=colMeans(X)
> cov=cov(X)
# 计算点 (0,0) 与 X 的距离
> dist.mahalanobis(c(0,0),mu,cov)
       [,1]
[1,] 3.755172
# 计算点 (1,1) 与 X 的距离
> dist.mahalanobis(c(1,1),mu,cov)
       [,1]
[1,] 2.513793

# 下面使用 R 中的函数计算
> x=matrix(c(0,1,0,1),nrow=2)
> x
    [,1] [,2]
[1,]    0    0
[2,]    1    1

> mahalanobis(x,mu,cov)
[1] 3.755172 2.513793

# 计算 x,y之间的距离
> a=c(0,0)
> b=c(1,1)
> mahalanobis(a,b,cov)
```

```
[1] 0.1241379
> dist.mahalanobis(a,b,cov)
        [,1]
[1,] 0.1241379
```

34.10 二值定性距离

两个 p 维向量 X_i, X_j 元素是二值数据时, 设 0 代表无, 1 代表有. 两个样本都有 p 个值. 第 k 个都是 0, 称在第 k 个值 0-0 配对; 第 k 个都是 1, 称在第 k 个值 1-1 配对; 若第 k 个不一样, 称在第 k 个值不配对.

记 m_0, m_1 分别为 0-0 配对和 1-1 配对的个数, m_2 为不配对的个数. 显然有

$$m_0 + m_1 + m_2 = p$$

两个样本的距离可以定义为

$$d_{ij} = \frac{m_2}{m_1 + m_2}$$

dist() 函数 method="binary" 即计算二值定性距离. 值为非零作为 "on", 值为零的作为 "off" 对待.

下面例子中, 不配对有2个, 1-1 配对有3个故距离为

$$d = 2/(2+3) = 0.4$$

```
> x <- c(0, 0, 1, 1, 1, 1)
> y <- c(1, 0, 1, 1, 0, 1)

> dist(rbind(x,y), method= "binary")
   x
y 0.4
```

Chapter 35

相似系数

参考 [8] 第 3 章

设 r_{ij} 为变量 X_i, X_j 之间的相似系数. 一般要求

- $r_{ij} = \pm 1$ 当且仅当 $X_i = aX_j (a \neq 0)$
- $|r_{ij}| <= 1$ 对一切 i,j 成立
- $r_{ij} = r_{ji}$ 对一切i,j成立

$|r_{ij}|$ 越接近 1, 表示关系越密切, 越接近 0, 关系越疏远.

35.1 角余弦系数

变量 X_i, X_j 的角余弦系数(coefficient of cosine of included angle)定义为

$$\begin{aligned} r_{ij} &= \frac{\sum_{k=1}^{n} x_{ki} x_{kj}}{\sqrt{(\sum_{k=1}^{n} x_{ki}^2)(\sum_{k=1}^{n} x_{kj}^2)}} \\ &= \frac{X_i X_j^T}{\| X_i \| \| X_j \|} \end{aligned}$$

实际上是未标准化的相关系数. 两个变量正交时, $r = 0$. 完全相似时, $r = \pm 1$

设两个变量的夹角为 θ, 则

$$cos\theta = \frac{X_i X_j^T}{\parallel X_i \parallel \parallel X_j \parallel} = r_{ij}$$

35.2 相关系数

最常用的相关系数就是 Pearson 乘积矩关联系数. 实际上是标准化的角余弦系数.

$$r_{ij} = \frac{\sum_{k=1}^{n}(x_{ki} - \bar{x_i})(x_{kj} - \bar{x_j})}{\sqrt{(\sum_{k=1}^{n}(x_{ki} - \bar{x_i})^2)(\sum_{k=1}^{n}(x_{kj} - \bar{x_j})^2)}}$$

```
> x1=1:10
> x2=11:20

===============
# 角余弦系数

> a=sum(x1*x2)
> b=sqrt(sum(x1^2)*sum(x2^2))
> r=a/b; r
[1] 0.9559123

# x3 与 x1 完全相似
> x3=2*x1
> a=sum(x1*x3)
> b=sqrt(sum(x1^2)*sum(x3^2))
> r=a/b; r
[1] 1

# 负相关
> x4=-2*x1
```

```
> a=sum(x1*x4)
> b=sqrt(sum(x1^2)*sum(x4^2))
> r=a/b; r
[1] -1

===============
# 相关系数
> cor(x1,x2)
[1] 1
> cor(x1,x3)
[1] 1
> cor(x1,x4)
[1] -1
```

35.3 联合系数(assosiation coefficient, confusion matrix)

设两个 n 维向量 X_i, X_j 元素是离散数据(二值或多值数据), 联合系数是它们之间一致性度量的函数. 大部分情况以二值数据出现, 这里假定取二值数据 0,1.

第 k 个元素的匹配有四种情况: 0-0 匹配和 1-1 匹配, 0-1 不匹配, 1-0 不匹配. 下表为各种匹配的个数, 明显 $a+b+c+d=n$

	1	0
1	a	b
0	c	d

```
> x <- c(0, 0, 1, 1, 1, 1)
> y <- c(1, 0, 1, 1, 0, 1)
> x==0 & y==0
[1] FALSE  TRUE FALSE FALSE FALSE FALSE

# a,b,c,d 各种匹配的个数
```

```
> a=sum(x==0 & y==0); a
[1] 1
> d=sum(x==1 & y==1); d
[1] 3
> b=sum(x==1 & y==0); b
[1] 1
> c=sum(x==0 & y==1); c
[1] 1
```

最简单的考虑就是计算匹配一致的个数占总个数的百分比(下表第 6 个公式)

$$S = \frac{a+d}{n}$$

35.4 各种系数列表

下面是各种系数的列表注释: $A = \sqrt{(a+b)(a+c)}, D = \sqrt{(d+b)(d+c)}$

联合系数的选择没有同一的标准. 大部分的联合系数对 a 强调, 忽视 d. 徐克学等(1989)[8] (page 98) 设计了联合系数的普遍公式.

公式: 略

编 号	公 式	作 者/系 数 名 称	范 围
1	$\frac{a}{n}$	Russell and Rao, 1940	$[0,1]$
2	$\frac{a}{a+2(b+c)}$	Sokal and Sneath, 1963	$[0,1]$
3	$\frac{a}{a+b+c}$	Jaccard, 1908	$[0,1]$
4	$\frac{a}{2a+b+c}$	Czekanowski, 1913	$[0,1]$
5	$\frac{a+d}{n+b+c}$	Rogers and Tanimoto, 1960	$[0,1]$
6	$\frac{a+d}{n}$	Simple Matching	$[0,1]$
7	$\frac{2(a+d)}{n+a+d}$	Sokal and Sneath, 1963	$[0,1]$
8	$\frac{ad}{ad+bc}$	Unnamed coefficient	$[0,1]$
9	$\frac{2a}{2a+ab+ac+bc}$	Unnamed coefficient	$[0,1]$
10	$\frac{a}{2}(\frac{1}{a+b}+\frac{1}{a+c})$	Kulczynski, 1927	$[0,1]$
11	$\frac{1}{4}(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{d+b}+\frac{d}{d+c})$	Sokal and Sneath, 1963	$[0,1]$
12	$\frac{a}{A}$	Ochiai, 1957	$[0,1]$
13	$\frac{ad}{AD}$	Sokal and Sneath, 1963	$[0,1]$
14	$\frac{AD+ad-bc}{2AD}$	Unnamed coefficient	$[0,1]$
15	$\frac{ad-bc}{AD}$	correlation coefficient, Guifford,1942	$[-1,1]$
16	$\frac{a^2-bc}{A^2}$	McConnaughy, 1964	$[-1,1]$
17	$\frac{a+d-b-c}{n}$	Hamann, 1961	$[-1,1]$
18	$\frac{ad-bc}{ad+bc}$	Yule and Kendall, 1950	$[-1,1]$
19	$\frac{a+d}{b+c}$	Sokal and Sneath, 1963	$[0,\infty)$
20	$\frac{a}{b+c}$	Kulczynski, 1927	$[0,\infty)$
21	$\frac{2a}{ab+ac+bc}$	Sneath and Sokal, 1973	$[0,\infty)$
22	$\frac{b+c}{2a+b+c}$	Watson et. al., 1966	$[0,1]$
23	$\frac{b+c}{n}$	Euclidean Distance	$[0,1]$
22	$\frac{a}{A}-\frac{1}{2\sqrt{a+b}}$	Fager and McGowan, 1963	$(-\infty,1]$

Chapter 36

判别分析(Discriminant Analysis)

参考生物数学[8] 2.3 (生物统计数学模型判别分析数学模型) 和 [15] 8.1 前者的假设与推导过程基于 Fisher 判别, 后者的三者都介绍了, 且自己编写若干函数.

36.1 判别分析与主成分分析的关系

主成分分析(PCA)方法对于代表数据样本非常有效, 但是却不是分类的有效方法. 例如, 区分大写字母"O"与"Q", PCA 可以发现两个字母的相似之处, 却很可能把区分字母的"尾巴"特征抛弃掉了. 也就是说, PCA 寻找有效表示数据的主轴方向, 判别分析 (Discriminant Analysis) 是寻找的是用来有效分类的方向.

36.2 基于 Mahalanobis 距离的数学模型

设两个总体 X_1, X_2 的均值向量分别为 μ_1, μ_2, 协方差矩阵分别为 $\sum_1, \sum_2$, 今有一样本 x, 判断其来自哪个总体.

需要计算 x 与两个总体的 Mahalanobis 距离(的平方)然后比较. x 来自 X_1 若

$$d^2(x, X_1) <= d^2(x, X_2)$$

x 来自 X_2 若

$$d^2(x, X_1) > d^2(x, X_2)$$

即空间划分两个集合(判别准则)

$$R_1 = \{x | d^2(x, X_1) <= d^2(x, X_2)\}$$
$$R_2 = \{x | d^2(x, X_1) >= d^2(x, X_2)\}$$

36.2.1 协方差矩阵相同

当 $\mu_1 \neq \mu_2$, $\sum_1 = \sum_2 = \sum$, 考虑

$$\begin{aligned}
d^2(x, X_2) - d^2(x, X_1) &= (x-\mu_2)^T \sum^{-1}(x-\mu_2) - (x-\mu_1)^T \sum^{-1}(x-\mu_1) \\
&= (x^T \sum^{-1} x - 2x^T \sum^{-1} \mu_2 + \mu_2^T \sum^{-1} \mu_2) - (x^T \sum^{-1} x - 2x^T \sum^{-1} \mu_1 + \mu_1^T \sum^{-1} \\
&= 2x^T \sum^{-1}(\mu_1 - \mu_2) + (\mu_1 + \mu_2)^T \sum^{-1}(\mu_2 - \mu_1) \\
&= 2(x - \frac{\mu_1 + \mu_2}{2})^T \sum^{-1}(\mu_1 - \mu_2) \\
&\equiv 2(x - \bar{\mu})^T \sum^{-1}(\mu_1 - \mu_2) \\
&\equiv 2w(x)
\end{aligned}$$

其中

$$\bar{\mu} = \frac{\mu_1 + \mu_2}{2}$$

称 $w(x)$ 为两个总体距离的判别函数.

$$w(x) = (x - \bar{\mu})^T \sum^{-1}(\mu_1 - \mu_2)$$

判别准则变为

$$R_1 = \{x|w(x) >= 0\}$$
$$R_2 = \{x|w(x) < 0\}$$

实际上总体的均值与协方差矩阵是未知的, 需要用样本的均值与协方差矩阵来代替.

设 $x_1^{(1)}, \cdots, x_{n1}^{(1)}$ 来自总体 X_1 的 $n1$ 个样本, $x_1^{(2)}, \cdots, x_{n2}^{(2)}$ 来自总体 X_2 的 $n2$ 个样本. 则样本的均值为

$$\begin{aligned}
\hat{\mu_i} &= \frac{1}{n_i}\sum_{j=1}^{n_i} x_j^{(i)}, i = 1, 2 \\
\hat{\sum} &= \frac{1}{n_1 + n_2 - 2}\sum_{i=1}^{2}\sum_{j=1}^{n_i}(x_j^{(i)} - \hat{\mu_i})(x_j^{(i)} - \hat{\mu_i})^T \\
&= \frac{1}{n_1 + n_2 - 2}(S_1 + S_2) \\
\hat{\mu} &= \frac{\mu_1 + \mu_2}{2}
\end{aligned}$$

判别函数变为

$$\hat{w}(x) = (x - \hat{\mu})^T \hat{\sum}^{-1} (\hat{\mu_1} - \hat{\mu_2})$$

判别准则变为

$$R_1 = \{x|\hat{w}(x) >= 0\}$$
$$R_2 = \{x|\hat{w}(x) < 0\}$$

36.2.2 协方差矩阵不同

判别准则不变, 与协方差矩阵相同时相同.

判别函数为

$$w(x) = (x-\mu_2)^T \sum_2^{-1}(x-\mu_2) - (x-\mu_1)^T \sum_1^{-1}(x-\mu_1)$$

使用样本代替后判别函数变为

$$\hat{w}(x) = (x-\hat{\mu_2})^T \hat{\sum}_2^{-1}(x-\hat{\mu_2}) - (x-\hat{\mu_1})^T \hat{\sum}_1^{-1}(x-\hat{\mu_1})$$

其中

$$\begin{aligned} \hat{\sum}_i &= \frac{1}{n_i-1}\sum_{j=1}^{n_i}(x_j^{(i)}-\hat{\mu_i})(x_j^{(i)}-\hat{\mu_i})^T \\ &= \frac{1}{n_i-1}S_i, \quad i=1,2 \end{aligned}$$

36.3 Bayes 判别

Bayes 判别是假定分析之前对研究对象已经有一定的认识, 这种认识数学上描述为先验概率. 取得样本后, 再使用样本修正先验概率分布, 就得到后验概率分布. 然后使用后验概率分布进行各种统计推断.

36.3.1 先验概率与损失函数

考虑两个总体的情况. 设 X_1, X_2 分别具有概率密度函数 $f_1(x), f_2(x)$. 其中 x 是 p 维向量. 记 Ω 为样本空间(即 x 所有可能的观察值的全体). R_1 为根据某种规则判为 X_1 的 x 的全体(这些 x 不一定都来自 X_1), R_2 为根据某种规则判为 X_2 的 x 的全体(这些 x 也不一定都来自 X_2), 而 $R_1 + R_2 = \Omega$.

某些样本来自 X_1 但是被误判为 X_2 的概率为

$$P(2|1) = P\{x \in R_2|X_1\} = \int_{R_2} f_1(x)dx$$

来自 X_2 但是被误判为 X_1 的概率为

$$P(1|2) = P\{x \in R_1|X_2\} = \int_{R_1} f_2(x)dx$$

类似, 来自 X_1 被判为 X_1 的概率为

$$P(1|1) = P\{x \in R_1|X_1\} = \int_{R_1} f_1(x)dx$$

来自 X_2 被判为 X_2 的概率为

$$P(2|2) = P\{x \in R_2|X_2\} = \int_{R_2} f_2(x)dx$$

设 p_1, p_2 为总体的先验概率, 且 $p_1 + p_2 = 1$, 于是

$$\begin{aligned} P(\text{正确的判为}X_1) &= P(\text{来自}X_1, \text{被判为}X_1) \\ &= P(x \in R_1|X_1) * P(X_1) \\ &= P(1|1) * p_1 \\ P(\text{误判为}X_1) &= P(\text{来自}X_2, \text{被判为}X_1) \\ &= P(x \in R_1|X_2) * P(X_2) \\ &= P(1|2) * p_2 \end{aligned}$$

类似的

$$\begin{aligned} P(\text{正确的判为}X_2) &= P(2|2) * p_2 \\ P(\text{误判为}X_2) &= P(2|1) * p_1 \end{aligned}$$

设 $L(1|2)$ 表示来自 X_2 被误判为 X_1 引起的损失, $L(2|1)$ 表示来自 X_1 被误判为 X_2 引起的损失, 并规定 $L(1|1) = L(2|2) = 0$.

将误判概率与误判损失结合, 定义平均误判损失(expected cost of misclassification, ECM) 为

$$ECM(R_1, R_2) = L(2|1)P(2|1)p_1 + L(1|2)P(1|2)p_2$$

合理的选择是使 ECM 达到极小.

36.3.2 两个总体的 Bayes 判别

可以证明, 极小化平均误判损失函数的划分为

$$R_1 = \{x|\frac{f_1(x)}{f_2(x)} >= \frac{L(1|2)}{L(2|1)}\frac{p_2}{p_1}\}$$

$$R_2 = \{x|\frac{f_1(x)}{f_2(x)} < \frac{L(1|2)}{L(2|1)}\frac{p_2}{p_1}\}$$

因此可以将此式作为 Bayes 判别的准则. 我们只需要计算

- 样本点 x 的概率密度函数比 $f_1(x)/f_2(x)$
- 损失比 $L(1|2)/L(2|1)$
- 先验概率比 p_2/p_1

首先考虑总体协方差矩阵相同的情况, 此时 X_i 的密度为

$$f_i(x) = (2\pi)^{-\pi/2}|\sum|^{-1/2}exp\{-\frac{1}{2}(x-\mu_i)^T\sum^{-1}(x-\mu_i)\}, \quad i = 1, 2$$

因此, R_1, R_2 的划分区域等价于

$$R_1 = \{x|W(x) >= \beta\}$$
$$R_2 = \{x|W(x) < \beta\}$$

其中

$$\begin{aligned} W(x) &= \frac{1}{2}(x-\mu_2)^T\sum^{-1}(x-\mu_2) - \frac{1}{2}(x-\mu_1)^T\sum^{-1}(x-\mu_1) \\ &= [x - \frac{1}{2}(\mu_1+\mu_2)]^T\sum^{-1}(\mu_1-\mu_2) \\ \beta &= ln\frac{L(1|2)p_2}{L(2|1)p_1} \end{aligned}$$

不难看出, 对于正态分布总体的 Bayes 判别, 其判别规则可以看成 Mahalanobis 距离判别的推广, 当

$$p_1 = p_2, \quad L(1|2) = L(2|1), \quad \beta = 0$$

就是 Mahalanobis 距离判别.

考虑协方差矩阵不同的情况,

$$R_1 = \{x|W(x) >= \beta\}$$
$$R_2 = \{x|W(x) < \beta\}$$

其中

$$\begin{aligned} W(x) &= \frac{1}{2}(x-\mu_2)^T \sum_2^{-1}(x-\mu_2) - \frac{1}{2}(x-\mu_1)^T \sum_1^{-1}(x-\mu_1) \\ \beta &= ln\frac{L(1|2)p_2}{L(2|1)p_1} + \frac{1}{2}ln(\frac{|\sum_1|}{|\sum_2|}) \end{aligned}$$

编写 R 程序: 略

36.3.3 多分类问题的 Bayes 判别

从上面的计算过程可以看到, Bayes 判别本质上就是找到一种判别准则, 使得平均误判损失达到最小, 即相应的概率达到最大.

假设样本有 k 类, 分别为 $X_1, \cdots, X_k$, 相应的先验概率为 $p_1, \cdots, p_k$, 假设所有错判的损失是相同的, 因此相应的判别准则为

$$R_i = \{x|p_i f_i(x) = max_{1<=j<=k} p_j f_j(x)\}, \quad i = 1, \cdots, k$$

当 k 类总体的协方差矩阵相同, 即

$$\sum_1 = \cdots = \sum_k = \sum$$

此时概率密度函数为

$$f_j(x) = (2\pi)^{-\pi/2}|\sum|^{-1/2}exp\{-\frac{1}{2}(x-\mu_j)^T\sum^{-1}(x-\mu_j)\}, \quad j = 1, \cdots, k$$

则计算函数

$$d_j(x) = \frac{1}{2}(x-\mu_j)^T\sum^{-1}(x-\mu_j) - lnp_j$$

计算中, 协方差矩阵使用其估计值代替.

当协方差矩阵不同时, 此时概率密度函数为

$$f_j(x) = (2\pi)^{-\pi/2}|\sum_j|^{-1/2}exp\{-\frac{1}{2}(x-\mu_j)^T\sum_j^{-1}(x-\mu_j)\}, \quad j = 1, \cdots, k$$

则计算函数

$$d_j(x) = \frac{1}{2}(x-\mu_j)^T\sum_j^{-1}(x-\mu_j) - lnp_j - \frac{1}{2}ln(|\sum_j|)$$

计算中, 协方差矩阵也分别使用其估计值代替.

判别准则等价为

$$R_i = \{x|d_i(x) = min_{1<=j<=k}d_j(x)\}, \quad i = 1, \cdots, k$$

编写 R 程序: 略

36.4 Fisher 判别

原理: 如果判别函數表示原觀察變數之線性組合，則依此一判別函數而分類的區隔，將可使不同兩組的平均數之距離的平方相對於兩區隔樣本之pooled變異數，其比值最大。

具体算法: 参考 [8] [15]

36.5 例子

程序包 MASS: 函数 lda(Linear discriminant analysis), qda(Quadratic discriminant analysis.)

mda 程序包提供 mixture and flexible discriminant analysis with mda() and fda()

lda() method 参数有四种方法

- "moment": for standard estimators of the mean and variance, 标准的均值与方差估计
- "mle": for MLEs, 最大似然估计
- "mve": to use 'cov.mve' (minimum volume ellipsoid)
- "t" for robust estimates based on a t distribution. 基于 t 分布的估计

下面是一个一维的例子. lda() 自带的例子是 4 维的

```
> library(MASS)
> d=data.frame(x=(1:20),g=c(rep(1,10),rep(2,10)))
> d
   x g
1   1 1
2   2 1
3   3 1
4   4 1
5   5 1
6   6 1
7   7 1
8   8 1
9   9 1
10 10 1
11 11 2
12 12 2
13 13 2
```

```
14 14 2
15 15 2
16 16 2
17 17 2
18 18 2
19 19 2
20 20 2

> l=lda(g~x,data=d)
> l
Call:
lda(g ~ x, data = d)

Prior probabilities of groups:
  1   2
0.5 0.5

Group means:
    x
1  5.5
2 15.5

Coefficients of linear discriminants:
       LD1
x 0.3302891

# 新数据必须使用data.frame 且 x 标记,
# 即与 lda() 函数使用的变量名称一样
> new = data.frame(x=(5:15))
> predict(l,new)
> predict(l,new)
$class
 [1] 1 1 1 1 1 1 2 2 2 2 2
Levels: 1 2

$posterior
          1         2
1  0.99752738 0.002472623
2  0.99267486 0.007325140
3  0.97850450 0.021495499
4  0.93861689 0.061383107
```

```
5  0.83703953 0.162960471
6  0.63308037 0.366919631
7  0.36691963 0.633080369
8  0.16296047 0.837039529
9  0.06138311 0.938616893
10 0.02149550 0.978504501
11 0.00732514 0.992674860

$x
         LD1
1  -1.8165902
2  -1.4863011
3  -1.1560120
4  -0.8257228
5  -0.4954337
6  -0.1651446
7   0.1651446
8   0.4954337
9   0.8257228
10  1.1560120
11  1.4863011
```

Chapter 37

聚类分析

聚类分析中, 大多数据往往不难直接参加运算, 需要先中心化或标准化.

37.1 系统聚类(hierarchical clustering method)

使用最多. 设开始有 n 个样本,

$$X_1, \cdots, X_n$$

其中每个样本 p 维 (p 个性状), 第 k 个样本为

$$X_k = x_{1k}, \cdots, x_{pk}$$

基本思想是,

1. 开始把每个样本作为一类, 计算 n 个样本之间的 $n * n$ 距离矩阵
2. 取距离最短的两个样本合并成为一个新类.
3. 然后计算此新类与其它样本的距离, 产生 $(n-1) * (n-1)$ 距离矩阵.

4. 取此矩阵中距离最短的两个样本又合并成为一个新类.

5. 然后计算此新类与其它样本的距离, 产生 $(n-2)*(n-2)$ 距离矩阵.

6. $\cdots$

依次进行, 直到最后只有一类, 结束计算.

按照计算新类与其它样本的距离的方法不同可以分为最短距离法, 最长距离法, 中间距离法等.

37.1.1 最短距离法(the shortest distance method)

计算新类与其它样本的距离的方法为: 设第一步 1,2 被合并为 $n+1$, 下面要计算 $n+1$ 与 $3,4,\cdots,n$ 的距离,

$$\begin{aligned} d_{n+1,3} &= min(d_{1,3}, d_{2,3}) \\ d_{n+1,4} &= min(d_{1,4}, d_{2,4}) \\ d_{n+1,5} &= min(d_{1,5}, d_{2,5}) \\ &\cdots \\ d_{n+1,n} &= min(d_{1,n}, d_{2,n}) \end{aligned}$$

37.1.2 最长距离法(the longest distance method)

计算新类与其它样本的距离的方法为: 设第一步 1,2 被合并为 $n+1$, 下面要计算 $n+1$ 与 $3,4,\cdots,n$ 的距离

$$\begin{aligned} d_{n+1,3} &= max(d_{1,3}, d_{2,3}) \\ d_{n+1,4} &= max(d_{1,4}, d_{2,4}) \\ d_{n+1,5} &= max(d_{1,5}, d_{2,5}) \\ &\cdots \\ d_{n+1,n} &= max(d_{1,n}, d_{2,n}) \end{aligned}$$

37.1.3 中间距离法(median method)

计算新类与其它样本的距离的方法为: 设第一步 1,2 被合并为 $n+1$, 下面要计算 $n+1$ 与 $3,4,\cdots,n$ 的距离

$$d_{n+1,3}=\sqrt{\frac{1}{2}(d_{1,3}^2+d_{2,3}^2)-\frac{1}{4}d_{1,2}^2}$$

$$d_{n+1,4}=\sqrt{\frac{1}{2}(d_{1,4}^2+d_{2,4}^2)-\frac{1}{4}d_{1,2}^2}$$

$$d_{n+1,5}=\sqrt{\frac{1}{2}(d_{1,5}^2+d_{2,5}^2)-\frac{1}{4}d_{1,2}^2}$$

$$\cdots$$

$$d_{n+1,n}=\sqrt{\frac{1}{2}(d_{1,n}^2+d_{2,n}^2)-\frac{1}{4}d_{1,2}^2}$$

实际上采用的是 k 与 1,2 连线中线的距离作为新距离

37.1.4 中间距离法的推广

计算新类与其它样本的距离的方法为: 设第一步 1,2 被合并为 $n+1$, 下面要计算 $n+1$ 与 $3,4,\cdots,n$ 的距离

$$d_{n+1,3}=\sqrt{\frac{1-\beta}{2}(d_{1,3}^2+d_{2,3}^2)-\beta d_{1,2}^2}$$

其中 $\beta<1$. 当 $\beta=0$, 称为 Mcquitty 相似分析法.

37.1.5 类平均法(average linkage method)

两种定义. 一种是把类与类之间的距离定义为所有样本之间的平均距离.

设合并后的类 G_K, G_L 分别有 n_K, n_L 个样本, 定义 G_K, G_L 之间的距离为

$$d_{KL} = \frac{1}{n_K n_L} \sum_{i \in G_K, j \in G_L} d_{ij}$$

例如 G_K 为 1,2 合并得到, G_L 为 3,4 合并得到, 那么

$$d_{KL} = \frac{1}{2 * 2}(d_{1,3} + d_{1,4} + d_{2,3} + d_{2,4})$$

另一种定义为样本对之间平方距离的平均作为距离的平方

$$d^2_{KL} = \frac{1}{n_K n_L} \sum_{i \in G_K, j \in G_L} d^2_{ij}$$

递推公式为

$$d^2_{MJ} = \frac{n_K}{n_M} d^2_{KJ} + \frac{n_L}{n_M} d^2_{LJ}$$

类平均法较好利用了所有样本的信息, 很多时候被认为是一种较好的距类方法.

进一步推广为

$$d^2_{MJ} = (1 - \beta)(\frac{n_K}{n_M} d^2_{KJ} + \frac{n_L}{n_M} d^2_{LJ}) + \beta d^2_{KL}$$

其中 $\beta < 1$ 称为可变类平均法.

37.1.6 重心法

类与类(每个类大于 3 个样本, 2 个样本的重心在中点, 1 个就是其本身)之间的距离定义为它们重心之间的 Euclide 距离.

此方法处理异常值比其它方法更稳健, 但是在别的方面一般不如类平均法或离差平方和法效果好.

37.1.7 离差平方和法(Ward 法)

Ward (1936) 提出此方法. 基于方差分析的思想, 如果类分的正确, 则同类内部的离差平方和应较小, 不同类之间的离差平方和应较大.

与重心法比较差别在于一个常数系数(类内一般个数), 那么结果两个大类的距离倾向于比较大, 不易合并, 更符合对聚类的实际要求. 故很多情况下优于重心法.

但是对异常值敏感.

设 G_K, G_L 合并为新类 G_M, 则内部的离差平方和为

$$W_K = \sum_{i \in G_K} (x_{(i)} - \bar{x}_K)^T (x_{(i)} - \bar{x}_K)$$

其中 $\bar{x}_K$ 为 G_K 的重心. 类似的有

$$W_L = \sum_{i \in G_L} (x_{(i)} - \bar{x}_L)^T (x_{(i)} - \bar{x}_L)$$

$$W_M = \sum_{i \in G_M} (x_{(i)} - \bar{x}_M)^T (x_{(i)} - \bar{x}_M)$$

如果 G_K, G_L距离较近, 则合并后增加的离差平方和 $W_M - W_K - W_L$ 应较小. 否则, 应较大. 定义 G_K, G_L 平方距离为

$$d^2_{KL} = W_M - W_K - W_L$$

这种方法称为离差平方和法或 Ward 方法(Ward's minimum variance method)

递推公式为

$$d^2_{MJ} = \frac{n_J + n_K}{n_J + n_M} d^2_{KJ} + \frac{n_J + n_L}{n_J + n_M} d^2_{LJ} - \frac{n_J}{n_J + n_M} d^2_{KL}$$

G_K, G_L 之间的平方距离也可以写成

$$d^2_{KL} = \frac{n_K n_L}{n_M} (\bar{x}_K - \bar{x}_L)^T (\bar{x}_K - \bar{x}_L)$$

37.1.8 其它方法

采用各种平均, 例如使用所有联系两个类群之间的分类单位(样本)距离的平方和再取平均值, 作为距离的平方.

37.2 例子

hclust() 函数执行系统分类.

```
> x<-c(1,2,6,8,11); dim(x)<-c(5,1);x
     [,1]
[1,]    1
[2,]    2
[3,]    6
[4,]    8
[5,]   11
> d<-dist(x); d
   1  2  3  4
2  1
3  5  4
4  7  6  2
5 10  9  5  3
> h1<-hclust(d,"single");
> h2<-hclust(d,"complete")
> h3<-hclust(d,"median")
> h4<-hclust(d,"mcquitty")

> par(mfrow=c(2,2))
> plot(h1,hang=-1)
> plot(h2,hang=-1)
> plot(h3,hang=-1)
> plot(h4,hang=-1)

# hang =-1 表示线画到底. 查看 hang 的用法
> par(mfrow=c(2,1))
> plot(h4,hang=-1)
> plot(h4)
```

```
# as.dendrogram 可以绘制更好的图
> d1<-as.dendrogram(h1)
> str(d1)
--[dendrogram w/ 2 branches and 5 members at h = 4]
  |--[dendrogram w/ 2 branches and 2 members at h = 1]
  |  |--leaf 1
  |  `--leaf 2
  `--[dendrogram w/ 2 branches and 3 members at h = 3]
     |--leaf 5
     `--[dendrogram w/ 2 branches and 2 members at h = 2]
        |--leaf 3
        `--leaf 4

par(mfrow=c(2,2))
plot(d1)
plot(d1, nodePar=list(pch = c(1,NA), cex=0.8, lab.cex=0.8),
   type = "t", center=TRUE)
plot(d1, edgePar=list(col = 1:2, lty = 2:3),
    dLeaf=1, edge.root = TRUE)
plot(d1, nodePar=list(pch = 2:1, cex=.4*2:1, col=2:3),
    horiz=TRUE)
```

下面是一个比较实际的例子 ([15] page 488). 数据为全国 31 个省级地区居民的 8 项基本消费支出, 我们要把 31 个样本分类.

```
X<-data.frame(
x1=c(2959.19, 2459.77, 1495.63, 1046.33, 1303.97, 1730.84,
    1561.86, 1410.11, 3712.31, 2207.58, 2629.16, 1844.78,
    2709.46, 1563.78, 1675.75, 1427.65, 1783.43, 1942.23,
    3055.17, 2033.87, 2057.86, 2303.29, 1974.28, 1673.82,
    2194.25, 2646.61, 1472.95, 1525.57, 1654.69, 1375.46,
    1608.82),
x2=c(730.79, 495.47, 515.90, 477.77, 524.29, 553.90, 492.42,
    510.71, 550.74, 449.37, 557.32, 430.29, 428.11, 303.65,
    613.32, 431.79, 511.88, 512.27, 353.23, 300.82, 186.44,
    589.99, 507.76, 437.75, 537.01, 839.70, 390.89, 472.98,
    437.77, 480.99, 536.05),
```

```
x3=c(749.41, 697.33, 362.37, 290.15, 254.83, 246.91, 200.49,
    211.88, 893.37, 572.40, 689.73, 271.28, 334.12, 233.81,
    550.71, 288.55, 282.84, 401.39, 564.56, 338.65, 202.72,
    516.21, 344.79, 461.61, 369.07, 204.44, 447.95, 328.90,
    258.78, 273.84, 432.46),
x4=c(513.34, 302.87, 285.32, 208.57, 192.17, 279.81, 218.36,
    277.11, 346.93, 211.92, 435.69, 126.33, 160.77, 107.90,
    219.79, 208.14, 201.01, 206.06, 356.27, 157.78, 171.79,
    236.55, 203.21, 153.32, 249.54, 209.11, 259.51, 219.86,
    303.00, 317.32, 235.82),
x5=c(467.87, 284.19, 272.95, 201.50, 249.81, 239.18, 220.69,
    224.65, 527.00, 302.09, 514.66, 250.56, 405.14, 209.70,
    272.59, 217.00, 237.60, 321.29, 811.88, 329.06, 329.65,
    403.92, 240.24, 254.66, 290.84, 379.30, 230.61, 206.65,
    244.93, 251.08, 250.28),
x6=c(1141.82, 735.97, 540.58, 414.72, 463.09, 445.20, 459.62,
    376.82, 1034.98, 585.23, 795.87, 513.18, 461.67, 393.99,
    599.43, 337.76, 617.74, 697.22, 873.06, 621.74, 477.17,
    730.05, 575.10, 445.59, 561.91, 371.04, 490.90, 449.69,
    479.53, 424.75, 541.30),
x7=c(478.42, 570.84, 364.91, 281.84, 287.87, 330.24, 360.48,
    317.61, 720.33, 429.77, 575.76, 314.00, 535.13, 509.39,
    371.62, 421.31, 523.52, 492.60, 1082.82, 587.02, 312.93,
    438.41, 430.36, 346.11, 407.70, 269.59, 469.10, 249.66,
    288.56, 228.73, 344.85),
x8=c(457.64, 305.08, 188.63, 212.10, 192.96, 163.86, 147.76,
    152.85, 462.03, 252.54, 323.36, 151.39, 232.29, 160.12,
    211.84, 165.32, 182.52, 226.45, 420.81, 218.27, 279.19,
    225.80, 223.46, 191.48, 330.95, 389.33, 191.34, 228.19,
    236.51, 195.93, 214.40)
)

# 距离矩阵
> d<-dist(scale(X))
> h<-hclust(d)
# 绘图
> plclust(h)
# 31 个样本分为 5 个大类
> r<-rect.hclust(h,5)
```

37.3 类个数的确定

分成多少个类是合适的? 这个问题没有统一的答案. 一般根据需要确定. 基本方法大概是

1. 给定你认为的距离的最小阈值
2. 观测散点图, 查看类大概的个数
3. 使用某种统计量确定类的个数
4. 根据初步分类的图再次分类确定个数

Bemirman (1972) 提出了根据研究目的来确定的分类方法, 并提出了根据谱系图来分析的准则

- 各类重心距离必须较大
- 各类包含的元素不要太多
- 类的个数需符合使用目的
- 多采用几种方法, 应该结果差不多

rect.hclust() 函数绘出指定类的框.

```
# 需先 plot, plclust() 也可以
> plot(h1)
# 同时在图上绘出框
> re<-rect.hclust(h1, k=3)
> re
[[1]]
[1] 1 2

[[2]]
[1] 5

[[3]]
[1] 3 4
```

37.4 k-均值动态聚类

系统聚类需要计算距离矩阵, 当样本很多时, 需占用很多内存和计算时间. 基于此, 产生了动态聚类方法.

动态聚类的基本思想是, 开始粗略的分一下, 然后按照某种最优原则修改不合理的分类, 直到分类比较合理为止. 此方法计算量较小, 内存较小, 方法简单, 适用于大样本.

算法: 任何多元分析教科书均有

37.4.1 k means 算法

参考 http://en.wikipedia.org/wiki/K-means_algorithm

k-means 算法最早由 MacQueen 1967 年提出, 后来经许多人多次修改.

k 个聚类具有以下特点: 各聚类本身尽可能的紧凑, 而各聚类之间尽可能的分开.

用途:

1. 资料压缩: 以少数的资料点来代表大量的资料, 达到资料压缩的功能
2. 资料分类: 以少数代表点来代表特点类别的资料, 可以降低资料量及计算量

k-means 算法的工作过程说明如下:

1. 从 c 个数据对象任意选择 k 个对象作为初始聚类中心
2. 循环 (3)到 (4) 直到每个聚类不再发生变化为止(收敛)
3. 根据每个聚类对象的均值(中心对象), 计算每个对象与这些中心对象的距离; 并根据最小距离重新对相应对象进行划分

4. 重新计算每个(有变化)聚类的均值(中心对象)

一般都采用均方差作为标准测度函数. 下面是伪代码

```
var m = initialCentroids(x, K);
var N = x.length;
while (!stoppingCriteria) {
  var w = [][];
  // calculate membership in clusters
  for (var n = 1; n <= N; n++) {
    v = arg min (v0) dist(m[v0], x[n]);
    w[v].push(n);
   }
  // recompute the centroids
  for (var k = 1; k <= K; k++) {
    m[k] = avg(x in w[k]);
   }
}
return m;
```

37.4.2 k-means++方法

2006 年提出了一个关于初始值的选择的新的改进, 叫做"k-means++".[1] 基本思想是选择尽量接近大的数量的点的作为初始点, 然后开始聚类. 作者使用 L^2 范数来选择聚类中心. 虽然初始直到计算量比较大, 但是对减少误差很有帮助, 而且后来的聚类过程会很快, 从而整个计算过程会加快 2-10 倍. 大的数据量会减少 1000 倍的误差. 几乎与 vanilla k-means 方法的速度和误差一样.

kmeans() 函数使用 k-均值方法, 采用逐个修改方法, 方法有"Hartigan-Wong", "Lloyd", "Forgy", "MacQueen" 三种, 具体见帮助.

[1] D. Arthur, S. Vassilvitskii: "k-means++ The Advantages of Careful Seeding" 2007 Symposium on Discrete Algorithms (SODA).

下面对 31 个省级地区居民的 8 项基本消费支出使用 k-均值方法分类. 帮助里有绘图的例子

```
> km <- kmeans(scale(X), 5, nstart = 20); km
K-means clustering with 5 clusters of sizes 16, 10, 1, 3, 1

Cluster means:
        x1         x2        x3        x4         x5        x6
1 -0.7008593 -0.33291790 -0.5450901 -0.2500165 -0.54749319 -0.6131804
2  0.2646918  0.04585518  0.2487958 -0.3405821 -0.01812541  0.2587437
3  1.1255255  2.91079330 -1.0645632 -0.4082114  0.53291392 -1.0476079
4  1.8790347  1.02836873  2.1203833  2.1727806  1.49972764  2.2232050
5  1.8042004 -1.12776493  0.9368961  1.2959544  3.90904835  1.6014419
        x7         x8
1 -0.5420723 -0.57966702
2  0.2874133 -0.02413414
3 -0.9562089  1.66126641
4  0.9583064  1.94532737
5  3.8803141  2.01876530

Clustering vector:
 [1] 4 2 1 1 1 1 1 1 4 2 4 1 2 1 2 1 2 2 5 2 1 2 2 1 2 3 1 1 1 1 1

Within cluster sum of squares by cluster:
[1] 30.14432 22.12662  0.00000 10.19134  0.00000

Available components:
[1] "cluster"  "centers"  "withinss" "size"
```

37.5 k 邻近法(K Nearest Neighbors, knn)算法

参考

Teknomo, Kardi. K-Nearest Neighbors Tutorial.
http://people.revoledu.com/kardi/tutorial/KNN/index.html

http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

knn 是有监督的分类方法. 已经用于数据挖掘, 模式识别图像处理等方面. 成功的例子包括手写字体, 卫星图像和 EKG 模式识别.

37.5.1 knn 算法

knn 基于训练样本和新数据的属性分类. knn 不依赖于任何模型, 只依赖于记忆.

给出新的样本, 我们发现在它周围 k 个样本中属于某类最多的样本, 那么这些最多的样本的归类就是新样本的类别. 这实际上是一个它周围样本对新样本的投票过程.

下面是一个例子, 我们知道两个属性 $X_1 = (x_{1,1}, x_{2,1}, \cdots), X_2 = (x_{1,2}, x_{2,2}, \cdots)$ 的24个样本的分类 g, 然后对新样本$s_{25} = (x_{25,1} = 5, x_{25,2} = 6)$分类.

	x1	x2	g
s_1	4.00	3.00	1.00
s_2	1.00	3.00	1.00
s_3	3.00	3.00	1.00
s_4	3.00	7.00	1.00
...	...	...	...
s_{23}	7.00	4.00	2.00
s_{24}	8.00	8.00	2.00
s_{25}	5	6	???

```
 X=data.frame(
   x1=c(4,1,3,3,7,4,6,5,3,6,4,4,5,7,5,10,7,4,9,5,8,6,7,8),
   x2=c(3,3,3,7,4,1,5,6,7,2,6,4,8,8,6,5,6,10,7,4,5,6,4,8),
   g=c(rep(1,10),rep(2,14)) )
> X
  x1 x2 g
1  4  3 1
```

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

# 下面画图看看
# red 为类1, blue 为类2.
> plot(x2~x1,col=c("red","blue")[g],data=X)
# 新样本(5,6)为 "*" 标记
> points(5,6,pch=8,cex=3)
```

已知的样本为训练样本. 下面是步骤

- 确定 k 值
- 计算新样本与所有训练样本的距离
- 排序计算出的距离
- 收集前 k 个最小距离和它们属的类别

- 判别新样本的类别

假设 $k = 8$, 我们使用 8 个最邻近点来确定新样本的分类. 首先计算新样本 s_{25} 与所有 24 个训练样本的距离(此处使用 Euclidean 距离), 然后从小到大排序, 取前 8 个最小值, 查看这 8 个训练样本的分类, 其中类 1 有 3 个, 类 2 有 5 个. 我们判断新样本的分类为类 2.

```
# 两个向量 euclidean 距离
dist.euclidean <- function(x,y){
   res <- sqrt(sum((x-y)^2))
   res
}

# 计算训练样本与新样本的距离
> s=data.frame(x1=X$x1,x2=X$x2)
> apply(s,1,dist.euclidean,y=c(6,5))
 [1] 2.828427 5.385165 3.605551 3.605551 1.414214 4.472136 0.000000 1.414214
 [9] 3.605551 3.000000 2.236068 2.236068 3.162278 3.162278 1.414214 4.000000
[17] 1.414214 5.385165 3.605551 1.414214 2.000000 1.000000 1.414214 3.605551

# 实际上距离的平方更好看一些, 计算也容易(但是我们不准
备编写新的函数了)
> apply(s,1,dist.euclidean,y=c(6,5))^2
 [1]  8 29 13 13  2 20  0  2 13  9  5  5 10 10  2 16  2 29 13  2  4  1  2 13

# [排序并查看类别
> d <- cbind(apply(s,1,dist.euclidean,y=c(6,5))^2, X$g)
> o<-order(d[,1])
> d1<-d[o,]
> d1
     [,1] [,2]
 [1,]    0    1
 [2,]    1    2
 [3,]    2    1
 [4,]    2    1
 [5,]    2    2
 [6,]    2    2
 [7,]    2    2
 [8,]    2    2
```

```
 [9,]    4    2
[10,]    5    2
[11,]    5    2
[12,]    8    1
[13,]    9    1
[14,]   10    2
[15,]   10    2
[16,]   13    1
[17,]   13    1
[18,]   13    1
[19,]   13    2
[20,]   13    2
[21,]   16    2
[22,]   20    1
[23,]   29    1
[24,]   29    2
```

37.5.2 预测

使用 knn 来预测(extrapolation, 外推)

X Y 是按照时间排列的数值型数据. 第六个 $X = 6.5$, 我们要预测 Y 的值.

	X	Y
1	1.00	23.00
2	1.20	17.00
3	3.20	12.00
4	4.00	27.00
5	5.10	8.00
6	6.5	???

- 首先确定 $k = 2$
- 计算新样本 6.5 与其它 X 的距离

	X	Y	6.5 与 X 的距离	标记最近的 2 个 Y
1	1.00	23.00	5.5	
2	1.20	17.00	5.3	
3	3.20	12.00	3.3	
4	4.00	27.00	2.5	y
5	5.10	8.00	1.4	y
6	6.5	???		

- 计算最近的 2 个 Y 值的平均 $\frac{27+8}{2} = 17.5$, 即是 $X = 6.5$ 的预测值

37.5.3 平滑

使用同样的方法可以平滑(interpolation, 内插), 只要取 $X \in [1, 5]$, 然后计算每个 X 值的预测, 即可得到平滑.

```
d<-data.frame(
   X=c(1,1.2,3.2,4,5.1),
   Y=c(23,17,12,27,8) )
> d
   X  Y
1 1.0 23
2 1.2 17
3 3.2 12
4 4.0 27
5 5.1  8

> attach(d)
# 我们想从 0 到 6 平滑 x
> x<-seq(0,6,by=0.5)
> x
 [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0

# 先用 0.1 做例子
> a=0.1
# abs(X-a) 为距离,
# 取 k=2, 即前两个距离最小的对应的 Y 值
```

```
> Y[order(abs(X-a))[1:2]]
[1] 23 17
# 预测值为对应 Y 值的平均
> pre<-mean(Y[order(abs(X-a))[1:2]]); pre
[1] 20

# 编写内插/平滑(预测)函数
# A 相当于 X, B 相当于 Y
> pre<-function(A,B,x,k){
   p<-mean(B[order(abs(A-x))[1:k]]);
   p
}
> x<-seq(0,6,by=0.5)
> x
 [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
# 对 x 中每个值预测(内插)
> apply(X=as.matrix(x),1,pre,A=X,B=Y,k=2)
 [1] 20.0 20.0 20.0 20.0 20.0 14.5 19.5 19.5 19.5 17.5 17.5 17.5 17.5
```

37.5.4 优点与缺点

优点

- 对噪音不敏感(robust), 尤其使用加权距离
- 若训练数据很大, 算法是比较有效的

缺点

- 需要确定 k 值
- 距离的种类难于确定
- 计算量比较大, 因为要计算所有训练数据与新数据的距离. 使用例如 K-D tree 等其它数据结构可能会好些

37.5.5 knn() 函数用法

class 包的 knn() 函数用于通常的分类, 使用 Euclidean 距离, 判别方法为投票法. 如果有个数一样的类, 则随机选择一个.

对于数值型数据及其平均(此处的预测与平均)并不能实现.

```
# 使用内插的例子数据
X=c(1,1.2,3.2,4,5.1),
Y=c(23,17,12,27,8)
> library(class)
> knn(train=X,test=as.matrix(seq(0,6,by=0.5)),cl=Y,k=2)
 [1] 17 17 17 17 17 17 12 27 27 8  8  8  8
Levels: 8 12 17 23 27

# 下面使用算法中的数据
 X=data.frame(
   x1=c(4,1,3,3,7,4,6,5,3,6,4,4,5,7,5,10,7,4,9,5,8,6,7,8),
   x2=c(3,3,3,7,4,1,5,6,7,2,6,4,8,8,6,5,6,10,7,4,5,6,4,8),
   g=c(rep(1,10),rep(2,14)) )

# 新数据
> test<-matrix(rnorm(20)*10,ncol=2)
> test
           [,1]       [,2]
 [1,]  14.497537  26.6676358
 [2,] -19.116549   1.9244647
 [3,]  -3.845555  -6.2873504
 [4,]  -6.286773 -11.1151424
 [5,]  -9.735719  -4.0103464
 [6,]   2.039382   5.3554126
 [7,] -14.485949  14.5503272
 [8,] -19.374097   9.2758593
 [9,]   8.656341  -0.1229066
[10,] -17.240399  -5.9442139

> cl <- X[,3]
> train<-X[,1:2]
> knn(train,test,cl,k=8)
 [1] 2 1 1 1 1 1 1 1 2 1
```

Chapter 38

主成分分析(PCA)

参考 [15] 第九章, [10] 十四章第四节. [8] 第二章第四节主成分分析

主成分分析(principal component analysis, PCA)是 Pearson(1901)提出的. 后来被 Hotelling(1933) 发展.

PCA 是一种降维技术, 把多个变量化为少数几个主成分, 能够反映原始变量大部分信息, 通常表示为原变量的线性组合.

设 X 有 p 个变量, 为 $n * p$ 阶矩阵, 即 n 个样本的 p 维向量. 首先对 X 的 p 个变量寻找正规化线性组合, 使它的方差达到最大(谁的方差?), 这个新的变量称为第一主成分. 抽取第一主成分后, 第二主成分的抽取方法与第一主成分一样, 使抽取第一主成分后的留下的变量的剩余方差达到最大. 依次类推, 直到各主成分累积方差达到总方差的一定比例(一般为 80%)为止.

38.1 协方差矩阵求主成分

38.1.1 记号

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \dots \\ x_{n1} & x_{n2} & \vdots & x_{np} \end{bmatrix} = \begin{bmatrix} X_{(1)}^T \\ X_{(2)}^T \\ \vdots \\ X_{(k)}^T \\ \vdots \\ X_{(n)}^T \end{bmatrix} = [X_1, X_2, \cdots, X_i, \cdots, X_p]$$

其中 X_i 为第 i 列, $i = 1, 2, \cdots, p$

$$X_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ni} \end{bmatrix}$$

其中 $X_{(k)}$ 为第 k 行(第 k 个样本), $k = 1, 2, \cdots, n$

$$X_{(k)} = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{kp} \end{bmatrix}$$

记 $\sum = Var(X)$ 为 X 的协方差矩阵. $\mu = E(X) = (\bar{X_1}, \cdots, \bar{X_p})$ 为 X 的均值向量.

一般, 对于协方差矩阵 $\sum$ 存在正交矩阵 Q, 将它化为对角矩阵, 即

$$Q^T \sum Q = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{bmatrix}$$

且 $\lambda_1 >= \lambda_2 >= \cdots >= \lambda_p$.

则$\lambda_1, \lambda_2, \cdots, \lambda_p$ 就是特征根, 矩阵 Q 的第 i 列就是对应特征根的特征向量.

为方便记 a_i 为 Q 的列向量

$$Q = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \cdots \\ a_{p1} & a_{p2} & \vdots & a_{pp} \end{bmatrix} = [a_1, a_2, \cdots, a_i, \cdots, a_p]$$

38.1.2 求主成分

下面分解 X 的方差. 记

$$\begin{aligned} Z &= \begin{bmatrix} Z_{(1)}^T \\ Z_{(2)}^T \\ \vdots \\ Z_{(k)}^T \\ \vdots \\ Z_{(n)}^T \end{bmatrix} \\ &= XQ = \begin{bmatrix} X_{(1)}^T Q \\ X_{(2)}^T Q \\ \vdots \\ X_{(k)}^T Q \\ \vdots \\ X_{(n)}^T Q \end{bmatrix} \\ &= [X_1, X_2, \cdots, X_i, \cdots, X_p]Q = \\ &= X[a_1, a_2, \cdots, a_i, \cdots, a_p] = [Xa_1, Xa_2, \cdots, Xa_i, \cdots, Xa_p] \\ &= [Z_1, Z_2, \cdots, Z_i, \cdots, Z_p] \\ &= Z \end{aligned}$$

显然

$$\begin{aligned} Var(Z_i) &= Z_i^T Z_i = a_i^T X^T X a_i = a_i^T \sum a_i = \lambda_i, \quad i = 1, \cdots, p \\ Cov(Z_i, Z_j) &= a_i^T X^T X a_j = a_i^T \sum a_j = 0, \quad i, j = 1, \cdots, p \end{aligned}$$

则 Z_1 方差最大, Z_2 次之, $\cdots$.

其中 $Z_1, Z_2, \cdots, Z_i, \cdots, Z_p$ 分别称为 X 的第 1 主成分, 第 2 主成分, $\cdots$.

所有主成分方差的和为 $\lambda_1 + \lambda_2 + \cdots + \lambda_p$.

$$\begin{aligned} E(Z) &= E(XQ) = Q^T \mu \\ Var(Z) &= \Lambda \end{aligned}$$

称 $\lambda_i / \sum_{i=1}^{p} \lambda_i$ 为主成分 Z_i 的贡献率. 贡献率表示的是主成分解释原始变量 X 的能力, 主成分的贡献率越大, 解释原始变量的能力越强. 这样忽略贡献率小的主成分, 通常取前 m 个主成分(对主成分的累积贡献率 80%) 的主成分即可. 此时可以使用 $Z_1, Z_2, \cdots, Z_m$ 代替 $X_1, X_2, \cdots, X_i, \cdots, X_p$. 由于 $m < p$, 我们就达到了简化原始数据的目的. 累积贡献率是前 m 个主成分从原始变量提取了多少信息的度量.

38.1.3 原始变量与主成分的相关系数

由前面知($a_{(i)}$ 为矩阵 Q 的第 i 行)

$$X = ZQ^T$$

$$X_i = Za_{(i)} = Z_1 a_{i1} + Z_2 a_{i1} + \cdots + Z_p a_{ip}$$

对上式两边取方差为

$$\sigma_{ii} = \lambda_1 a_{i1}^2 + \cdots + \lambda_p a_{ip}^2$$

由于 $a_{i1}^2 + \cdots + a_{ip}^2 = 1$, 实际上 σ_{ii} 是 $\lambda_1, \cdots, \lambda_p$ 的加权平均.

故

$$\begin{aligned} Cov(X_i, Z_j) &= Cov(Z_j a_{ij}, Z_j) = a_{ij}\lambda_j, \quad i, j = 1, \cdots, p \\ \rho(X_i, Z_j) &= \frac{Cov(X_i, Z_j)}{\sqrt{X_i}\sqrt{Z_j}} = \frac{\sqrt{\lambda_j}}{\sqrt{\sigma_{ii}}} a_{ij}, i, j = 1, \cdots, p \end{aligned}$$

前面提到累积贡献率是前 m 个主成分 $Z_1, Z_2, \cdots, Z_m$ 从原始变量 $X_1, X_2, \cdots, X_i, \cdots, X_p$ 提取了多少信息的度量. 那么前 m 个主成分 $Z_1, Z_2, \cdots, Z_m$ 包含了 X_i 的多少信息呢? 这个是使用 X_i 与 $Z_1, Z_2, \cdots, Z_m$ 的复相关系数的平方来度量的, 称为前 m 个主成分 $Z_1, Z_2, \cdots, Z_m$ 对原始变量 X_i 的贡献率, 记为 $\rho^2_{i,1\cdots m}$

$$\begin{aligned} \rho^2_{i,1\cdots m} &= \sum_{j=1}^{m} \rho^2(X_i, Z_j) = \sum_{j=1}^{m} \lambda_j a^2_{ij}/\sigma_{ii} \\ &= \sum_{j=1}^{m} \lambda_j a^2_{ij}/(\lambda_1 a^2_{i1} + \cdots + \lambda_p a^2_{ip}) \\ \rho^2_{i,1\cdots p} &= 1 \end{aligned}$$

38.1.4 载荷(loading)

由于

$$Z_j = X a_j = X_1 a_{1j} + \cdots + X_p a_{pj}$$

称 a_{ij} 为第 j 主成分在第 i 个原始变量 X_i 上的载荷(loading), 它度量了 X_i 对 Z_j 的重要程度.

实际上, 在主成分分析中, 载荷就是正交矩阵 Q. 在因子分析中, 就是载荷因子矩阵.

38.2 相关矩阵求主成分

如果原始数据 X 各变量单位不同时, 应该将其标准化后求主成分, 此时协方差矩阵就变为相关矩阵(注: 中心化后协方差矩阵不变). 其它的推导方法内容等基本类似. 得到的主成分的性质更加简单.

设标准化后的 X 为 X^*, 则

$$X_i^* = \frac{X_i - \bar{X}_i}{\sqrt{\sigma_{ii}}}$$

其协方差矩阵, 也就是 X 的相关矩阵记为 R. R 的 p 个特征值记为

$$\lambda_1^* >= \lambda_2^* >= \cdots >= \lambda_p^*$$

相应的单位特征向量记为

$$a_1^*, \cdots, a_p^*$$

p 个主成分记为

$$\begin{aligned} Z^* &= [Z_1^*, \cdots, Z_p^*] \\ Z_i^* &= X_i^* a_i^* \\ Z^* &= X^* R \end{aligned}$$

Z^* 的性质如下

1. $E(Z^*) = 0, Var(Z^*) = \Lambda^*$. $\Lambda^* = diag(\lambda_1^*, \lambda_2^*, \cdots, \lambda_p^*)$
2. $\sum_{i=1}^{p} \lambda_i^* = p$
3. X_i^*, Z_j^* 的相关系数为

$$\rho(X_i^*, Z_j^*) = \sqrt{\lambda_j^*} a_{ij}^*, \quad i, j = 1, \cdots, p$$

4. 前 m 个主成分 $Z_1^*, Z_2^*, \cdots, Z_m^*$ 对 X_i^* 的贡献率为

$$\rho_{i,1\cdots m}^2 = \sum_{j=1}^{m} \rho^2(X_i^*, Z_j^*) = \sum_{j=1}^{m} \lambda_i^* a_{ij}^{*2}$$

5.

$$\rho_{i,1\cdots p}^2 = \sum_{j=1}^{p} \rho^2(X_i^*, Z_j^*) = \sum_{j=1}^{p} \lambda_i^* a_{ij}^{*2} = 1$$

38.3 主成分特征向量的具体问题的相关解释

详细的参考任何主成分分析的书, 有详细的解释. 此处简略一说.

例如特征矩阵如下(见例子)

```
Standard deviations: # 特征值
[1] 1.5748783 0.9948694 0.5971291 0.4164494

# 贡献率
Proportion of Variance 0.6200604 0.2474413 0.0891408 0.04335752

# 累积贡献率
Cumulative Proportion  0.62 0.868 0.9566 1.0000

Rotation: # 特征矩阵(载荷)
             PC1        PC2        PC3         PC4
Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

每列绝对值大的几个代表的向量就是此主成分代表的含义. 第一列绝对值大的是 Murder, Assault. 那么 Murder, Assault 就是第一主成分, 这两个变量可以解释全部方差的 62%, 第二列绝对值大的是 UrbanPop, 这个可以解释全部方差的 24.7%, 这两个加起来可以解释全部方差的 86.8%. 最后一个主成分占方差较小, 就可以忽略了.

剩下的就是使用专业知识(或常识, 经验)解释这些东西了. Murder, Assault 是代表治安方面的问题, 而 UrbanPop 代表人口方面的问题.

38.4 例子

R 中的函数 princomp() 与 prcomp() 用法意义一样, 都是做主成分分析的. 其中一种用法为

```
princomp(x, cor = FALSE, scores = TRUE, covmat = NULL,
         subset = rep(TRUE, nrow(as.matrix(x))), ...)
```

$cor = TRUE$ 是使用相关矩阵求主成分, 否则使用协方差矩阵.

```
prcomp(x, retx = TRUE, center = TRUE, scale. = FALSE,
       tol = NULL, ...)
```

$scale = TRUE$ 即使用相关矩阵求主成分, 否则使用协方差矩阵求主成分.

下面是几个相关的函数

- summary()
- predict(): princomp() 与 prcomp() 预测值(即计算出的主成分)稍微不同. 详细见例子.

- loadings() 只用于 princomp()
- screeplot() 碎石图
- biplot() 主成分的散点图

下面是主成分的计算

```
# 数据
> X=USArrests
> X
            Murder Assault UrbanPop Rape
Alabama       13.2     236       58 21.2
Alaska        10.0     263       48 44.5
Arizona        8.1     294       80 31.0
...
West Virginia   5.7      81       39  9.3
Wisconsin       2.6      53       66 10.8
Wyoming         6.8     161       60 15.6

=======================
# 手工计算
> c=cor(X)
> c
           Murder   Assault   UrbanPop     Rape
Murder   1.00000000 0.8018733 0.06957262 0.5635788
Assault  0.80187331 1.0000000 0.25887170 0.6652412
UrbanPop 0.06957262 0.2588717 1.00000000 0.4113412
Rape     0.56357883 0.6652412 0.41134124 1.0000000
> eigen(c)
$values  # 特征值
[1] 2.4802416 0.9897652 0.3565632 0.1734301

$vectors # 特征向量(载荷矩阵)
          [,1]       [,2]       [,3]       [,4]
[1,] -0.5358995  0.4181809 -0.3412327  0.64922780
[2,] -0.5831836  0.1879856 -0.2681484 -0.74340748
[3,] -0.2781909 -0.8728062 -0.3780158  0.13387773
[4,] -0.5434321 -0.1673186  0.8177779  0.08902432
```

```
> e=eigen(c)
# 矩阵正交
> t(e$vectors)%*%e$vectors
             [,1]         [,2]         [,3]         [,4]
[1,]  1.000000e+00  1.010205e-16 9.343112e-17 -7.732394e-17
[2,]  9.722583e-17  1.000000e+00 2.534323e-17 -1.257742e-16
[3,]  9.343112e-17  5.149960e-19 1.000000e+00  8.527250e-17
[4,] -1.071056e-16 -1.257742e-16 8.952799e-17  1.000000e+00
# 产生 diag(2.4802416 0.9897652 0.3565632 0.1734301)
> t(e$vectors)%*%c%*%e$vectors
             [,1]         [,2]         [,3]         [,4]
[1,]  2.480242e+00  1.338502e-15 -4.781332e-17 -4.455597e-16
[2,]  1.373047e-15  9.897652e-01 -3.856236e-16 -2.785095e-16
[3,] -1.435484e-16 -3.969671e-16  3.565632e-01  2.092510e-16
[4,] -2.162255e-16 -3.227534e-16  1.710769e-16  1.734301e-01

# 计算标准化的主成分(与 prcomp() 函数的预测结果一样, 但是与 princomp() 稍微不同)
> scale( as.matrix(X))%*%e$vectors
                   [,1]       [,2]       [,3]        [,4]
Alabama       -0.97566045  1.12200121 -0.43980366  0.154696581
Alaska        -1.93053788  1.06242692  2.01950027 -0.434175454
Arizona       -1.74544285 -0.73845954  0.05423025 -0.826264240
Arkansas       0.13999894  1.10854226  0.11342217 -0.180973554
California    -2.49861285 -1.52742672  0.59254100 -0.338559240
...
Washington     0.21472339 -0.96037394  0.61859067 -0.218628161
West Virginia  2.08739306  1.41052627  0.10372163  0.130583080
Wisconsin      2.05881199 -0.60512507 -0.13746933  0.182253407
Wyoming        0.62310061  0.31778662 -0.23824049 -0.164976866

=======================
# prcomp() 的用法
> p=prcomp(USArrests, scale=T)
> p
Standard deviations: # 特征值
[1] 1.5748783 0.9948694 0.5971291 0.4164494

Rotation: # 特征矩阵
              PC1       PC2       PC3        PC4
Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
```

```
Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
Rape     -0.5434321 -0.1673186  0.8177779  0.08902432

# 第一行为特征值. 第二行为主分量百分比, 第三行为累加主分量百分比
> summary(p)
Importance of components:
                          PC1   PC2    PC3    PC4
Standard deviation     1.57 0.995 0.5971 0.4164
Proportion of Variance 0.62 0.247 0.0891 0.0434
Cumulative Proportion  0.62 0.868 0.9566 1.0000

# 计算主成分. 注意与手工计算一样
> predict(p)
                    PC1        PC2        PC3         PC4
Alabama       -0.97566045  1.12200121 -0.43980366  0.154696581
Alaska        -1.93053788  1.06242692  2.01950027 -0.434175454
Arizona       -1.74544285 -0.73845954  0.05423025 -0.826264240
Arkansas       0.13999894  1.10854226  0.11342217 -0.180973554
...
Washington     0.21472339 -0.96037394  0.61859067 -0.218628161
West Virginia  2.08739306  1.41052627  0.10372163  0.130583080
Wisconsin      2.05881199 -0.60512507 -0.13746933  0.182253407
Wyoming        0.62310061  0.31778662 -0.23824049 -0.164976866

# 绘图查看
> screeplot(p)
> biplot(p)

=======================
# princomp() 用法. 下面的相当于 prcomp(USArrests, scale=T)
> p1=princomp(USArrests, cor = TRUE)
> p1
Call:
princomp(x = USArrests, cor = TRUE)

Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4
1.5748783 0.9948694 0.5971291 0.4164494
```

```
 4  variables and  50 observations.
> summary(p1)
Importance of components:
                       Comp.1    Comp.2    Comp.3    Comp.4
Standard deviation     1.5748783 0.9948694 0.5971291 0.41644938
Proportion of Variance 0.6200604 0.2474413 0.0891408 0.04335752
Cumulative Proportion  0.6200604 0.8675017 0.9566425 1.00000000

# 载荷矩阵.
> loadings(pr)

Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4
Murder   -0.536  0.418 -0.341  0.649
Assault  -0.583  0.188 -0.268 -0.743
UrbanPop -0.278 -0.873 -0.378  0.134
Rape     -0.543 -0.167  0.818

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00


# 预测. 注意与手工计算稍微不同
> predict(pr)
                  Comp.1     Comp.2     Comp.3      Comp.4
Alabama       -0.98556588  1.13339238 -0.44426879  0.156267145
Alaska        -1.95013775  1.07321326  2.04000333 -0.438583440
Arizona       -1.76316354 -0.74595678  0.05478082 -0.834652924
Arkansas       0.14142029  1.11979678  0.11457369 -0.182810896
...
Washington     0.21690338 -0.97012418  0.62487094 -0.220847793
West Virginia  2.10858541  1.42484670  0.10477467  0.131908831
Wisconsin      2.07971417 -0.61126862 -0.13886500  0.184103743
Wyoming        0.62942666  0.32101297 -0.24065923 -0.166651801
```

38.5 主成分回归

参考 [15] page 516.

当自变量出现多重共线性时, 经典回归方法做回归系数的最小二乘估计效果一般较差. 采用主成分回归能够克夫经典回归的不足.

下面是法国 1949 至 1959 共 11 年的经济分析数据. y 为进口总额. x1 为国内总产值, x2 为存储量, x3 为总消费量.(单位: 10 亿法郎)

```
x1=c(149.3, 161.2, 171.5, 175.5, 180.8, 190.7,
    202.1, 212.4, 226.1, 231.9, 239.0)
x2=c(4.2, 4.1, 3.1, 3.1, 1.1, 2.2, 2.1, 5.6, 5.0, 5.1, 0.7)
x3=c(108.1, 114.8, 123.2, 126.9, 132.1, 137.7,
    146.0, 154.1, 162.3, 164.3, 167.6)
y=c(15.9, 16.4, 19.0, 19.1, 18.8, 20.4, 22.7,
   26.5, 28.1, 27.6, 26.3)
```

38.5.1 线性回归

```
> r1 <- lm(y~x1+x2+x3)
> summary(r1)

Call:
lm(formula = y ~ x1 + x2 + x3)

Residuals:
    Min      1Q  Median      3Q     Max
-0.52367 -0.38953  0.05424  0.22644  0.78313

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.12799    1.21216  -8.355  6.9e-05 ***
x1          -0.05140    0.07028  -0.731 0.488344
```

```
x2          0.58695    0.09462   6.203 0.000444 ***
x3          0.28685    0.10221   2.807 0.026277 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4889 on 7 degrees of freedom
Multiple R-squared: 0.9919,     Adjusted R-squared: 0.9884
F-statistic: 285.6 on 3 and 7 DF,  p-value: 1.112e-07
```

回归方程为

$$y = -10.13 - 0.05 * x1 + 0.59 * x2 + 0.29 * x3$$

发现进口 y 与国内生产总值是负的关系, 这不太合理. 原因是三个变量存在共线性.

38.5.2 主成分分析

下面对三个变量使用主成分分析

```
> p<-princomp(~x1+x2+x3,cor=T)
> summary(p,loadings=TRUE)
Importance of components:
                       Comp.1    Comp.2      Comp.3
Standard deviation    1.413915 0.9990767 0.0518737839
Proportion of Variance 0.666385 0.3327181 0.0008969632
Cumulative Proportion  0.666385 0.9991030 1.0000000000

Loadings:
  Comp.1 Comp.2 Comp.3
x1  0.706        0.707
x2       -0.999
x3  0.707       -0.707
```

第一主成分是国内生产总值和总消费(x1, x3), 因此称第一主成分为产销因子. 第二主成分与存储(x2)相关, 称存储因子.

注 意

$$\lambda_3^2 = 0.05^2 = 0.0025 \approx 0$$

故 变 量 存 在 共 线 性.

38.5.3 主 成 分 回 归

取 前 2 个 主 成 分 做 回 归

```
> pre<-predict(p)
> z1<-pre[,1]
> z2<-pre[,2]
> r2<-lm(y~z1+z2)
> summary(r2)

Call:
lm(formula = y ~ z1 + z2)

Residuals:
    Min      1Q  Median      3Q     Max
-0.89838 -0.26050  0.08435  0.35677  0.66863

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  21.8909     0.1658 132.006 1.21e-14 ***
z1            2.9892     0.1173  25.486 6.02e-09 ***
z2           -0.8288     0.1660  -4.993  0.00106 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.55 on 8 degrees of freedom
Multiple R-squared: 0.9883,     Adjusted R-squared: 0.9853
F-statistic: 337.2 on 2 and 8 DF,  p-value: 1.888e-08
```

回 归 方 程 变 为

$$y = 21.89 + 2.99 * z1 - 0.83 * z2$$

38.5.4 得到与原自变量的关系式

下面我们要得到 y 与 x1, x2, x3 的关系. 由于

$$z_i = Xa_i = a_{1i}X_1 + a_{2i}X_2 + a_{i3}X_3 = a_{1i}x1 + a_{2i}x2 + a_{i3}x3$$

将 z_1, z_2 带入回归方程既得 y 与 x1, x2, x3 的关系式.

Chapter 39

因子分析

数学比较复杂, 具体请参考 [15] 9.2 章因子分析. [10] 14.5 因子分析.

因子分析把数据看作公共因子, 特殊因子和误差构成. 主成分分析把方差划分为不同的正交成分, 因子分析则把方差划分为不同的起因因子. 其特征值计算是从相关矩阵出发, 且将主成分转换为因子, 并计算出因子得分. 目前在心理学, 生物学, 经济学中广泛使用.

39.1 数学模型

下面简单解释一下.

记号与主成分分析中的记号一致. 数学模型为

$$X = \mu + AF + e$$

即

$$\begin{aligned} X_1 - \mu_1 &= a_{11}f_1 + a_{12}f_2 + \cdots + a_{1m}f_m + e_1 \\ &\vdots \\ X_p - \mu_p &= a_{p1}f_1 + a_{p2}f_2 + \cdots + a_{pm}f_m + e_p \end{aligned}$$

其中

$X_{(n*p)}$ 为 p 维原始数据, $Var(X) = \sum = (\sigma_{ij})_{p*p}$

$\mu = \bar{X_1}, \cdots, \bar{X_p}$,

$A = (a_{ij})_{(p*m)}$ 为因子载荷矩阵,

$F = f_1, \cdots, f_m$ 为公共因子向量,

$e = e_1, \cdots, e_p$ 为特殊因子.

$m <= p$ 为公共因子数.

通常假设

$$E(F) = 0, Var(F) = I_m$$
$$E(e) = 0, Var(e) = D = diag(\sigma_1^2, \cdots, \sigma_p^2)$$
$$Cov(F, e) = 0$$

故公共因子 F 彼此不相关且具有单位矩阵.

特殊因子 e 也不相关且与 F 也不相关.

$\sum$ 可以分解为

$$\sum = AA^T + D$$

因子载荷矩阵 A 不是唯一的, 这样可以通过因子旋转使得新因子有更好的实际意义.

A 的统计意义如下

1.

$$Cov(X, F) = A$$
$$Cov(X_i, f_j) = a_{ij}$$

即 a_{ij} 是第 i 个变量与第 j 个公共因子的相关系数. 即度量 X_i 可以由 f_j 表示的强度.

2. 令 $h_i^2 = \sum_{j=1}^{m} a_{ij}^2$, 则 h_i^2 反映了公共因子对 X_i 的方差贡献, 称为 X_i 的共同度(communality)或共性方差(common variance). 而 $\sigma_i^2 = var(e_i)$ 为 X_i 的特殊方差, 是特殊因子 e_i 对 X_i 的贡献.

   当 X 标准化后, 此时

$$h_i^2 + \sigma_i^2 = 1, \quad i = 1, \cdots, p$$

39.2 例子

R 中函数 factanal() 执行因子分析. 用法

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
        subset, na.action, start = NULL,
        scores = c("none", "regression", "Bartlett"),
        rotation = "varimax", control = NULL, ...)
```

- x: 公式, 或数据
- factors: 因子个数
- covmat: 样本协方差矩阵或相关矩阵. 此时不需要 x
- scores: 因子得分方法. scores="regression" 表示用回归方法计算因子得分. scores="Bartlett" 表示用 Bartlett 方法计算因子得分.
- rotation: 表示旋转. 缺省为方差最大旋转

下面是 R 的例子

```
# 数据, 可以假设为某公司对18个新员工的6项个人能力打分
v1 <- c(1,1,1,1,1,1,1,1,1,1,3,3,3,3,3,4,5,6)
v2 <- c(1,2,1,1,1,1,2,1,2,1,3,4,3,3,3,4,6,5)
v3 <- c(3,3,3,3,3,1,1,1,1,1,1,1,1,1,1,5,4,6)
```

```
v4 <- c(3,3,4,3,3,1,1,2,1,1,1,1,2,1,1,5,6,4)
v5 <- c(1,1,1,1,1,3,3,3,3,3,1,1,1,1,1,6,4,5)
v6 <- c(1,1,1,2,1,3,3,3,4,3,1,1,1,2,1,6,5,4)
m1 <- cbind(v1,v2,v3,v4,v5,v6)

> cor(m1)
        v1       v2       v3       v4       v5       v6
v1 1.0000000 0.9393083 0.5128866 0.4320310 0.4664948 0.4086076
v2 0.9393083 1.0000000 0.4124441 0.4084281 0.4363925 0.4326113
v3 0.5128866 0.4124441 1.0000000 0.8770750 0.5128866 0.4320310
v4 0.4320310 0.4084281 0.8770750 1.0000000 0.4320310 0.4323259
v5 0.4664948 0.4363925 0.5128866 0.4320310 1.0000000 0.9473451
v6 0.4086076 0.4326113 0.4320310 0.4323259 0.9473451 1.0000000

# 默认不计算得分
> factanal(m1, factors=3) # varimax is the default
Call:
factanal(x = m1, factors = 3)

Uniquenesses:
   v1    v2    v3    v4    v5    v6
0.005 0.101 0.005 0.224 0.084 0.005

Loadings:
   Factor1 Factor2 Factor3
v1 0.944   0.182   0.267
v2 0.905   0.235   0.159
v3 0.236   0.210   0.946
v4 0.180   0.242   0.828
v5 0.242   0.881   0.286
v6 0.193   0.959   0.196

                 Factor1 Factor2 Factor3
SS loadings        1.893   1.886   1.797
Proportion Var     0.316   0.314   0.300
Cumulative Var     0.316   0.630   0.929

The degrees of freedom for the model is 0 and the fit was 0.4755
```

结果中

- uniquenesses: 特殊方差. 即 $diag(cov(e))$
- loadings: 因子载荷矩阵, 即矩阵 A. 其中, Factor1 的 v1, v2 接近 1, Factor2 的 v5, v6 接近 1, Factor3 的 v3, v4 接近 1. 具体问题中可以根据经验总结其代表的实际意义.
- SS loadings: 公共因子 f_i 对变量 $X_1, \cdots, X_p$ 的总方差贡献.
- Proportion Var: 方差贡献率. 可以看到三个因子的贡献率差不多.
- Cumulative Var: 累积方差贡献率. 总的贡献率达到 0.929.

39.2.1 因子得分

得到公共因子 F 和因子载荷 A 后, 应该反过来考察每个样本的得分情况. 这样可以挑选某个因子得分较高或较低(或某几个因子得分都高/都低, 或指定哪个因子得分较高)的个体进一步研究

```
# 计算得分
> f<-factanal(m1, factors=3, scores="Bartlett")
> names(f)
 [1] "converged"    "loadings"     "uniquenesses" "correlation"  "criteria"
 [6] "factors"      "dof"          "method"       "scores"       "n.obs"
[11] "call"
> f$scores
       Factor1    Factor2    Factor3
 [1,] -0.9039949 -0.9308984  0.9475392
 [2,] -0.8685952 -0.9328721  0.9352330
 [3,] -0.9082818 -0.9320093  0.9616422
 [4,] -1.0021975 -0.2529689  0.8178552
 [5,] -0.9039949 -0.9308984  0.9475392
 [6,] -0.7452711  0.7273960 -0.7884733
 [7,] -0.7098714  0.7254223 -0.8007795
 [8,] -0.7495580  0.7262851 -0.7743704
 [9,] -0.8080740  1.4033517 -0.9304636
```

```
[10,] -0.7452711  0.7273960 -0.7884733
[11,]  0.9272282 -0.9307506 -0.8371538
[12,]  0.9626279 -0.9327243 -0.8494600
[13,]  0.9229413 -0.9318615 -0.8230509
[14,]  0.8290256 -0.2528211 -0.9668378
[15,]  0.9272282 -0.9307506 -0.8371538
[16,]  0.4224366  2.0453079  1.2864761
[17,]  1.4713902  1.2947716  0.5451562
[18,]  1.8822320  0.3086244  1.9547752

# 绘制前两个因子的散点图
> plot(f$scores[, 1:2], type="n")
> text(f$scores[,1], f$scores[,2])
```

39.2.2 与主成分分析对照

下面是主成分分析的结果, 做对照. 可以看到无明显主成分

```
# 主成分分析的结果
> prcomp(m1)
Standard deviations:
[1] 3.0368683 1.6313757 1.5818857 0.6344131 0.3190765 0.2649086

Rotation:
        PC1        PC2       PC3       PC4       PC5        PC6
v1 0.4168038 -0.52292304  0.2354298 -0.2686501  0.5157193 -0.39907358
v2 0.3885610 -0.50887673  0.2985906  0.3060519 -0.5061522  0.38865228
v3 0.4182779  0.01521834 -0.5555132 -0.5686880 -0.4308467 -0.08474731
v4 0.3943646  0.02184360 -0.5986150  0.5922259  0.3558110  0.09124977
v5 0.4254013  0.47017231  0.2923345 -0.2789775  0.3060409  0.58397162
v6 0.4047824  0.49580764  0.3209708  0.2866938 -0.2682391 -0.57719858
```

Chapter 40

典型相关分析

典型相关分析(canonical correlation analysis)是用于分析两组随机变量之间的相关程度的一种统计方法, 可以有效揭示两组随机变量之间的线性关系. 这个方法由 Hotelling (1935) 首先提出的.

如果需要寻找 X (p维) Y(q维) 的相关关系, 普通做法是列出$p*q$ 个相关系数, 然后进行分析. 缺点是不易把握.

典型相关分析原理是分别寻找 X Y 的线性组合

$$U_1 = Xa_1, V_1 = Yb_1$$

使其具有最大相关(注意并不唯一), 称 U_1, V_1 的相关系数为第一典型相关系数.其中 $a_1 = a_{11}, \cdots, a_{1p}$ $b_1 = b_{11}, \cdots, b_{1p}$.

然后如果存在 a_k, b_k 使得

1. $U_k = Xa_k, V_k = Yb_k$ 与前面的 $k-1$ 对典型变量都不相关
2. $Var(U_k) = 1, Var(V_k) = 1$
3. U_k, V_k 相关系数最大

称 U_k, V_k 为第 k 对典型变量, 称它们的相关系数为第 k 典型相关系数.

下面是 [15] 的例子.

X1: 体重. X2: 腰围. X3: 脉搏

Y1: 引体向上. Y2: 仰卧起坐. Y3: 跳跃次数.

```
test<-data.frame(
  X1=c(191, 193, 189, 211, 176, 169, 154, 193, 176, 156,
      189, 162, 182, 167, 154, 166, 247, 202, 157, 138),
  X2=c(36, 38, 35, 38, 31, 34, 34, 36, 37, 33,
      37, 35, 36, 34, 33, 33, 46, 37, 32, 33),
  X3=c(50, 58, 46, 56, 74, 50, 64, 46, 54, 54,
      52, 62, 56, 60, 56, 52, 50, 62, 52, 68),
  Y1=c( 5, 12, 13,  8, 15, 17, 14,  6,  4, 15,
       2, 12,  4,  6, 17, 13,  1, 12, 11,  2),
  Y2=c(162, 101, 155, 101, 200, 120, 215,  70,  60, 225,
      110, 105, 101, 125, 251, 210,  50, 210, 230, 110),
  Y3=c(60, 101, 58, 38, 40, 38, 105, 31, 25, 73,
      60, 37, 42, 40, 250, 115, 50, 120, 80, 43)
)
> test<-scale(test)
> ca<-cancor(test[,1:3],test[,4:6])
> ca
$cor
[1] 0.79560815 0.20055604 0.07257029

$xcoef
        [,1]       [,2]       [,3]
X1 -0.17788841 -0.43230348 -0.04381432
X2  0.36232695  0.27085764  0.11608883
X3 -0.01356309 -0.05301954  0.24106633

$ycoef
       [,1]       [,2]       [,3]
Y1 -0.0801801 -0.08615561 -0.29745900
Y2 -0.2418067  0.02833066  0.28373986
Y3  0.1643596  0.24367781 -0.09608099

$xcenter
         X1          X2          X3
```

```
 2.289835e-16  4.315992e-16 -1.778959e-16

$ycenter
         Y1          Y2          Y3
 1.471046e-16 -1.776357e-16  4.996004e-17
```

其中

- cor: 典型相关系数. 第 1,2,3 典型相关系数分别为: 0.79560815 0.20055604 0.07257029
- xcoef: 对应于 X 的系数.
- ycoef: 对应于 Y 的系数.
- xcenter: X 的中心, 即均值. 因为已经标准化, 故为 0
- ycenter: Y 的中心, 即均值. 因为已经标准化, 故为 0

计算典型变量下的得分

```
> U<-as.matrix(test[, 1:3])%*% ca$xcoef
> V<-as.matrix(test[, 4:6])%*% ca$ycoef
> cor(U[,1],V[,1])
[1] 0.7956082
> cor(U,V)
             [,1]         [,2]         [,3]
[1,]  7.956082e-01  3.069378e-17  1.386142e-16
[2,] -4.049495e-17  2.005560e-01 -4.029166e-17
[3,] -9.089002e-17 -3.131566e-17  7.257029e-02
> diag(cor(U,V))
[1] 0.79560815 0.20055604 0.07257029

# U1 V1 基本在一条直线上. 其它则分散
> plot(U[,1],V[,1])
> plot(U[,2],V[,2])
```

即

$$
\begin{aligned}
U_1 &= -0.178X_1 + 0.362X_2 - 0.136X_3 \\
V_1 &= -0.08Y_1 - 0.242Y_2 + 0.164Y_3 \\
\rho(U_1, V_1) &= 0.7956
\end{aligned}
$$

我们得到结论, 利用 U_1 可以预测 V_1, 即 X 与 Y 存在一定的线性关系.

40.1 TODO: 典型相关系数的检验

Part VI

时间序列

主 要 参 考 [31] chapter 15, Time series

Chapter 41

基本概念

41.1 CRAN Task View: Time Series Analysis

参考: CRAN Task View: Time Series Analysis 有很多的介绍.

41.2 arima.sim()函数-模拟产生各种时间序列

41.2.1 ts()的用法

ts() 用于产生时间序列. 用法如下.(例子来自在线帮助)

```
> ts(data=1:10, frequency = 4, start = c(1959, 2))
    Qtr1 Qtr2 Qtr3 Qtr4
1959        1    2    3
1960    4    5    6    7
1961    8    9   10
> ts(1:10, frequency = 4, start = c(1959, 2),end=c(1970,3))
    Qtr1 Qtr2 Qtr3 Qtr4
1959        1    2    3
```

```
1960    4    5    6    7
1961    8    9   10    1
1962    2    3    4    5
1963    6    7    8    9
1964   10    1    2    3
1965    4    5    6    7
1966    8    9   10    1
1967    2    3    4    5
1968    6    7    8    9
1969   10    1    2    3
1970    4    5    6

# frequency >=5 就需要使用 print() 显示其格式
> ts(1:10, frequency = 4, start = c(1959, 2))
     Qtr1 Qtr2 Qtr3 Qtr4
1959         1    2    3
1960    4    5    6    7
1961    8    9   10
> ts(1:10, frequency = 5, start = c(12, 2))
Time Series:
Start = c(12, 2)
End = c(14, 1)
Frequency = 5
 [1]  1  2  3  4  5  6  7  8  9 10
> ts(1:10, frequency = 7, start = c(12, 2))
Time Series:
Start = c(12, 2)
End = c(13, 4)
Frequency = 7
 [1]  1  2  3  4  5  6  7  8  9 10

# 打印时间序列
> print( ts(1:10, frequency = 7, start = c(12, 2)), calendar = TRUE)
   p1 p2 p3 p4 p5 p6 p7
12     1  2  3  4  5  6
13  7  8  9 10

# 绘图
> gnp <- ts(cumsum(1 + round(rnorm(100), 2)),
             start = c(1954, 7), frequency = 12)
> plot(gnp)
```

41.2.2 产生时间序列

模拟: 时间序列模拟的目的之一是发现序列的结构, 即序列怎么样构成的, 即发现一个算法来近似生成已知序列.

下面是几个模拟的时间序列.

```
n <- 100
k <- 5
N <- k*n
x <- (1:N)/n # x 为时间, 1-5 500个分隔

# 高斯噪声(白噪声)
y1<-rnorm(N)
plot(ts(y1))

# 累积噪声(随机漫步)
 y2<-cumsum(y1)
 plot(y2) # 普通绘图为散点图
 plot(ts(y2))

# 累积噪声(随机漫步)+高斯噪声
 y3<-cumsum(y1)+rnorm(N)
 plot(ts(y3))

# 累积累积噪声(累积随机漫步)
y4 <- cumsum(cumsum(y1))

# 乱七八糟的累积
y5 <- cumsum(cumsum(y1)+rnorm(N))+rnorm(N)

# 趋势+漫步+噪声
y6 <- 1 - x + cumsum(y1) + .2 * rnorm(N)

# x 的二次函数构成趋势, 然后+漫步+噪声
y7 <- 1 - x - .2*x^2 + cumsum(y1) +
```

```
     .2 * rnorm(N)

# 季节趋势+噪声
z <- .3 + .5*cos(2*pi*x) - 1.2*sin(2*pi*x) +
     .6*cos(2*2*pi*x) + .2*sin(2*2*pi*x) +
     -.5*cos(3*2*pi*x) + .8*sin(3*2*pi*x)
y8 <- z + .2 * rnorm(N)
y9 <- z+  cumsum(rnorm(N)) + .2*rnorm(N)

# 画图
op <- par(mfrow = c(3,3))
plot(ts(y1))
plot(ts(y2))
plot(ts(y3))
plot(ts(y4))
plot(ts(y5))
plot(ts(y6))
plot(ts(y7))
plot(ts(y8))
lines(z,type='l',lty=3,lwd=3,col='red')
plot(ts(y9))
par(op)
```

41.2.3 arima.sim()函数产生AR,MA或ARMA过程

我们可以使用 arima.sim() 函数产生一个模拟的 AR, MA 或 ARMA 过程. 下面是 R 的例子.

```
# 产生ARMA 过程, 指定MA的方差为 0.1796
arima.sim(n = 63, list(ar = c(0.8897, -0.4858), ma = c(-0.2279, 0.2488)),
           sd = sqrt(0.1796))
# mildly long-tailed. 可以指定随机数产生函数. 默认为正态
分布 rnorm
    arima.sim(n = 63, list(ar=c(0.8897, -0.4858), ma=c(-0.2279, 0.2488)),
           rand.gen = function(n, ...) sqrt(0.1796) * rt(n, df = 5))

# 产生 ARIMA 序列. 其 d=1. 即1 阶差分是平稳的
```

```
ts.sim <- arima.sim(list(order = c(1,1,0), ar = 0.7), n = 200)
ts.plot(ts.sim)
```

41.3 Hermitian 矩阵与函数

41.3.1 Hermitian 矩阵

若矩阵的值符合 $a_{ij} = \bar{a_{ji}}$, 此矩阵为 Hermitian 矩阵, 即矩阵本身与其共轭转置一样.

对于实矩阵, 实际上就是实对称矩阵. 例如

$$\begin{bmatrix} 3 & 2+i \\ 2-i & 1 \end{bmatrix}$$

求矩阵 A 的 Hermitian 矩阵

```
Conj(t(A))
```

41.3.2 Hermitian 函数

Hermitian 函数是复函数, 如果复共轭等于原始值的相反数. 实部为偶函数, 虚部为奇函数.

$$f(-x) = \bar{f(x)}$$

两个参数的也可以.

$$f(-x1, -x2) = \bar{f(x1, x2)}$$

41.4 自相关(Auto-correlation, ACF)

参考 http://en.wikipedia.org/wiki/Autocorrelation

时间序列数据是不独立的. 我们首先可以看看它的自相关函数: AutoCorrelation Function (ACF). 严格来讲, 自相关分为样本自相关和理论自相关, 分别来自样本数据和理论模型. 延迟 k 的自相关是观测 n 与观测 n-k 之间的相关. 可以假设自相关只和 k 有关, 和 n 无关.

41.4.1 定义

$$R(t,s) = \frac{E[(X_t - \mu_t)(X_s - \mu_s)]}{\sigma_t \sigma_s}$$

若定义 $\tau = t - s$, 则写作熟悉的方式

$$R(\tau) = \frac{E[(X_t - \mu)(X_{t+\tau} - \mu)]}{\sigma^2}$$

这实际上是偶函数, 写作

$$R(\tau) = R(-\tau)$$

对于离散序列 $X_1, \cdots, X_n$, 自回归为

$$R(k) = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} [X_t - \mu][X_{t+k} - \mu]$$

若 μ, σ 已知, 此为无偏估计. 但若使用样本均值和方差代替是有偏估计.

41.4.2 例子

acf() 函数参数 lag.max 默认 $10 * log10(N/m)$. N 为观测个数, m 为序列个数, 此处为 1. [1]

```
# my.acf() 函数计算自回归
my.acf <- function (
   x,
   lag.max = ceiling(5*log(length(x)))
){
  m <- matrix(
   c( NA,
     rep( c(rep(NA, lag.max-1), x),
         lag.max ),
     rep(NA,, lag.max-1)
    ),
   byrow=T,
   nr=lag.max)
  x0 <- m[1,]
  apply(m,1,cor, x0, use="complete")
}

# 计算自回归
> x=1:10
> my.acf(x,lag.max=3)
[1] 1 1 1

# 函数的矩阵 m 是这样
> lag.max=3
> m <- matrix(
+   c( NA,
+     rep( c(rep(NA, lag.max-1), x),
+         lag.max ),
+     rep(NA,, lag.max-1)
+    ),
+   byrow=T,
+   nr=lag.max)
> m
```

[1]貌似[31] 15.1.4 的 my.acf() 函数不正确, 见计算

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

# my.acf 实际使用的函数
# lag = 1
> cor(m[1,],m[2,],use="complete.obs")
[1] 1
# lag = 2
> cor(m[1,],m[3,],use="complete.obs")
[1] 1

=======================
# 按照公式手工计算, 与 R 函数一致
> u=mean(x)
> v=var(x)
> v
[1] 9.166667
# lag = 1,2 3
> sum((x[1:9]-u)*(x[2:10]-u))/(9*v)
[1] 0.7
> sum((x[1:8]-u)*(x[3:10]-u))/(9*v)
[1] 0.4121212
> sum((x[1:7]-u)*(x[4:10]-u))/(9*v)
[1] 0.1484848

=======================
# R 函数计算
> a=acf(x,lag.max=3);a

Autocorrelations of series ‘x’, by lag

    0     1     2     3
1.000 0.700 0.412 0.148

# 真实的例子. lag.max=19
x <- LakeHuron
acf(x, main="ACF of a time series (Lake Huron)")
```

41.5 互相关(Cross-correlation, CCF)

参考 http://en.wikipedia.org/wiki/Cross-correlation

41.5.1 定义

连续函数的互相关为[2]

$$(f \star g)(t) = \int_{-\infty}^{\infty} f^*(\tau) g(t+\tau) d\tau$$

其中 f^* 为复共轭

类似, 离散互相关为

$$(f \star g)[n] = \int_{m=-\infty}^{\infty} f^*[m] g[n+m]$$

自相关是序列对自身的互相关.

标准化的互相关为

$$\frac{1}{(n-1)\sigma_f \sigma_g} \sum (f - \bar{f})(g - \bar{g})$$

实际上是序列 f,g 标准化后的内积除以其 L^2 范数.

41.5.2 性质

- 互相关与卷积的关系

$$(f \star g)(t) = f^*(-t) * g(t)$$

[2]我们使用符号 $\star$ 表示互相关. $*$ 表示卷积

- 若 f,g 都是 Hermitian 的, 那么互相关等于卷积:

$$f \star g = f * g$$

-

$$(f \star g) \star (f \star g) = (f \star f) \star (g \star g)$$

- 与卷积一样,

$$F(f \star g) = F(f)^* \cdot F(g)$$

  其中 F 为傅立叶变换.

- f,h 卷积与 g 的互相关等于 h 与 f,g 互相关的卷积

$$(f * h) \star g = h * (f \star g)$$

41.5.3 例子

```
  x<-1:10
  y=c(3,4,5,1,2,3,6,7,8,9)
> y1=scale(y)
> c=ccf(x,y,plot=F);c

Autocorrelations of series 'X', by lag

   -6    -5    -4    -3    -2    -1     0     1     2     3     4
-0.378 -0.117  0.181  0.503  0.523  0.609  0.745  0.427  0.126 -0.144 -0.367
    5     6
-0.380 -0.291
# =================
#内积(点积)
> x1[,]%*%y1[,]
       [,1]
[1,] 6.709354
# =================
# lag=0
> x1[,]%*%y1[,]/9
        [,1]
```

```
[1,] 0.7454838
# =================
# lag=1
> y1[1:9]%*%x1[2:10]/9
         [,1]
[1,] 0.4265824
# lag=2
> y1[1:8]%*%x1[3:10]/9
         [,1]
[1,] 0.1256278
# lag=6
> y1[1:4]%*%x1[7:10]/9
          [,1]
[1,] -0.2912909
# =================
# lag=-1
> x1[1:9]%*%y1[2:10]/9
         [,1]
[1,] 0.6088118
# lag=-2
> x1[1:8]%*%y1[3:10]/9
         [,1]
[1,] 0.5232192
# lag=-6
> x1[1:4]%*%y1[7:10]/9
         [,1]
[1,] -0.378264
```

41.6 偏自相关(Partial Autocorrelation, PACF)

参考 http://www.qualityamerica.com/knowledgecente/knowctrPartial_Autocorrelation_Func

Partial Autocorrelation Function(PACF): The Partial Autocorrelation at the given lag. The PACF will vary between -1 and +1, with values near 1 indicating stronger correlation. The PACF removes the effect of shorter lag autocorrelation from the correlation estimate at longer lags. This estimate

is only valid to one decimal place.

$$\Phi_{m,m} = \frac{r_m - \sum_{j=1}^{m-1} \Phi_{m-1,j} r_{m-1}}{1 - \sum_{j=1}^{m-1} \Phi_{m-1,j} r_j}$$

其中 r_m 是自相关函数.

pacf() 计算偏自相关.

41.7 卷积(Convolution)

参考 http://en.wikipedia.org/wiki/Convolution

[3]

41.7.1 定义

$$\begin{aligned}(f * g)(t) &= \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau \\ &= \int_{-\infty}^{\infty} f(t-\tau) g(\tau) d\tau\end{aligned}$$

更一般的, 若 f,g 为空间 R^d 的复函数, 其卷积为

$$\begin{aligned}(f * g)(x) &= \int_{R^d} f(y) g(x-y) dy \\ &= \int_{R^d} f(x-y) g(y) dy\end{aligned}$$

[3]我们使用符号 $\star$ 表示互相关. $*$ 表示卷积

循环卷积: 若 g_T 为周期函数, f 不是. 则

$$(f * g_T)(t) = \int_{t_0}^{t_0+T} [\sum_{k=-\infty}^{\infty} f(\tau + kT)] g_T(t-\tau) d\tau$$

离散卷积:

$$\begin{aligned}(f * g)[n] &= \sum_{m=-\infty}^{\infty} f[m] g[n-m] \\ &= \sum_{m=-\infty}^{\infty} f[n-m] g[m]\end{aligned}$$

循环离散卷积: 若 g_N 为周期函数, f 不是. 则

$$(f * g_N)[n] = \sum_{m=0}^{N-1} (\sum_{k=-\infty}^{\infty} f[m+kN]) g_N[n-m]$$

当 f,g 都在 $[0, N-1]$ 有定义, 则循环离散卷积变为

$$\begin{aligned}(f * g_N)[n] &= \sum_{m=0}^{N-1} f[m] g_N[n-m] \\ &= \sum_{m=0}^{n} f[m] g[n-m] + \sum_{m=n+1}^{N-1} f[m] g[N+n-m] \\ &= \sum_{m=0}^{N-1} f[m] g[(n-m)_{modN}] = (f *_N g)[n]\end{aligned}$$

其中 $(f *_N g)[n]$ 表示对整数 N 卷积.

快速计算: 根据卷积定理, 利用快速傅立叶变换(fft)计算卷积.

41.7.2 性质(不全)

- 可交换(Commutativity)

$$f * g = g * f$$

- 结合(Associativity)

$$f * (g * h) = (f * g) * h$$

- 分配(Distributivity)

$$f * (g + h) = (f * g) + (f * h)$$

- 系数

$$a(f * g) = (af) * g = f * (ag)$$

- δ 为冲击函数

$$f * \delta = f$$

- 卷积定理(F为傅立叶变换)

$$F(f * g) = k \cdot F(f)F(g)$$

- 与反函数卷积(记$f^{(-1)}$为f的反函数)

$$f^{(-1)} * f = \delta$$

41.7.3 例子

R 函数 convolve() 使用 fft 计算卷积. 类型为非循环($type = ' open'$), 循环($type = "circular"$). 默认为循环卷积.

非循环时, 设

```
'r <- convolve(x,y, type = "open")'
'n <- length(x)'
'm <- length(y)'
```

那么

```
r[k] = sum(i; x[k-m+i] * y[i])
k = 1,..., n+m-1
```

对所有能够成立的i (即不超出index范围). 里面有一些重复计算的步骤, 如果可以充分利用, 我们可以设计一个精巧的算法(例如象fft的蝴蝶算法?)

```
> x=1:10
> y=11:15
> convolve(x,y,t='o')
 [1]  15  44  86 140 205 270 335 400 465 530 430 326 219 110
```

```
# 手工计算
# r1=15
r1=x[1-5+5]*y[5]

# r2=44
r2=x[2-5+4]*y[4]+
   x[2-5+5]*y[5]

# r3=86
r3=x[3-5+3]*y[3]+
   x[3-5+4]*y[4]+
   x[3-5+5]*y[5]

......

# 总结算式
my_open_conv <-function(x,y,k){
   n<-length(x)
   m<-length(y)
   i=1:m
   a=k-m+i
   a=a[a>0 & a<=n] # 保证下标不越界
   sum(x[a]*y[i[k-m+i>0][1:length(a)]])
```

```
}
> c=c()
> for (i in 1:14) c=append(c,my_open_conv(x,y,i));c
 [1]  15  44  86 140 205 270 335 400 465 530 430 326 219 110
```

如果是循环卷积, 那么需要 x,y 的长度一样. 上面的算法还是有效的,

```
r[k] = sum(i; x[k-m+i] * y[i])
k = 1,..., n

> x=1:5
> y=2:6
> convolve(x,y)
[1] 70 60 55 55 60

my_circular_conv <-function(x,y,k){
   n<-length(x)
   m<-length(y)
   if (n != m) stop('length x,y must be same')
   i=1:m
# 求模. 根据公式应该是 a=(k-m+i)%%5, 但是 R 的结果为下面才对
   a=(k-m+i-1)%%n
   a[a==0] = a[a==0]+n
   sum(x[a]*y)
}
# 确实是循环卷积
> c=c()
> for (i in 1:20) c=append(c,my_circular_conv(x,y,i));c
 [1] 70 60 55 55 60 70 60 55 55 60 70 60 55 55 60 70 60 55 55 60
```

41.8 白噪声(white noise)及其检验

残差的随机性检验在建立模型时非常重要.

正态分布的随机数就是白噪声.

```
rnorm(n)
```

如何判断一个序列是白噪声? 下面是几种方法.

41.8.1 ACF系数

看看 ACF, 如果自相关系数迅速衰减, 就可能是白噪声

```
> z <- rnorm(200)
> op <- par(mfrow=c(2,1), mar=c(5,4,2,2)+.1)
> plot(ts(z))
> acf(z, main = "")
> par(op)
```

41.8.2 Box–Pierce(Ljung–Box) test

此检验考察自相关系数的和服从卡方分布. Ljung–Box 检验对于小样本给出更好的卡方近似. 也叫做 portmanteau test.

零假设为: 给定序列是时间独立的.

```
> x=seq(0,10,by=0.1)
> y=cos(2*pi*x)+0.2*sin(x)+2*cos(x-1)
> plot(ts(y))
# y 不是时间独立的序列, p值很小, 拒绝零假设
> Box.test(y)

        Box-Pierce test

data:  y
X-squared = 91.9547, df = 1, p-value < 2.2e-16
```

```
> Box.test(y,type="Ljung-Box")

        Box-Ljung test

data:  y
X-squared = 94.7134, df = 1, p-value < 2.2e-16

# p值较大, 接受零假设, 此序列为时间独立的
> Box.test(rnorm(100))

        Box-Pierce test

data:  rnorm(100)
X-squared = 1.7053, df = 1, p-value = 0.1916

> Box.test(rnorm(100),type="Ljung-Box")

        Box-Ljung test

data:  rnorm(100)
X-squared = 0.4211, df = 1, p-value = 0.5164
```

41.8.3 其它检验

其它还有 McLeod-Li, Turning-point, difference-sign, rank 检验等.

还可以使用 Durbin–Watson 检验.

```
> library(car)
> ?durbin.watson
> durbin.watson(y)
[1] 0.07259672
```

41.8.4 游程检验(runs.test)

零假设为: 游程是随机的. 备择假设: 游程是增加的(或减少的). 检验基于游程的频率.

```
> library(tseries)
> ?runs.test
> x <- factor(sign(rnorm(100)))  # randomness
# p值较大, 是随机的游程
>     runs.test(x)

       Runs Test

data:  x
Standard Normal = 0.6416, p-value = 0.5212
alternative hypothesis: two.sided

# p值很小, 不是随机游程.
>  x <- factor(rep(c(-1,1),50))  # over-mixing
>     runs.test(x)

       Runs Test

data:  x
Standard Normal = 9.8499, p-value < 2.2e-16
alternative hypothesis: two.sided
```

41.8.5 tsdiag()

用于绘制标准化残差, 自相关的残差, portmanteau test(Box–Pierce(Ljung–Box) test) 的p值. 输入是 arima() 函数拟合的结果(拟合 ARIMA 模型).

```
data(co2)
r <- arima(
  co2,
```

```
  order = c(0, 1, 1),
  seasonal = list(order = c(0, 1, 1), period = 12)
)
tsdiag(r)

> r

Call:
arima(x = co2, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))

Coefficients:
         ma1     sma1
      -0.3501  -0.8506
s.e.   0.0496   0.0257

sigma^2 estimated as 0.0826:  log likelihood = -86.08,  aic = 178.16
```

Chapter 42

线性模型

42.1 包介绍

基本的包就是 tseries, sound, signal

dse: Dynamic Systems Estimation. 可以分析多元时间序列数据. 有一个 dse-guide.pdf. 这个是下载地址为 http://www.bank-banque-canada.ca/pgilbert/dse-guide.pdf.

42.2 时间序列分析的主要问题

统计的时候我们喜欢独立数据, 而时间序列包含非独立数据. 时间序列分析的目的是提取结构并将序列转换为独立的数据(经常叫做 “innovations”), 通常是提供一个模型(model/recipe)来构建接近原始时间序列, 即去除噪声的部分.

我们可以从另一个方面来看: 当研究统计现象时, 一般它有不同的实现(realizations). 对于时间序列, 只有一个实现. 于是, 我们把研究一个时间点的不同实现改变为研究不同时间点的同一个实现. 对于不同的统计现象, 这两种观点可以一样, 也可能不一样.

42.3 经典模型

一般, 我们想找到(分解)一个时间序列的3个部分: 整体趋势, 周期部分, 噪声.

下面看看时间序列的3个典型部分

```
# 自己构建有3个部分的数据
> x=seq(0,10,by=0.1)
> y=x+sin(2*pi*x)+3*cos(2*pi*x)+rnorm(length(x))
> plot(ts(y))

# 或 R 的数据
> data(co2)
> plot(co2)
```

此时使用各种回归都不好使.

42.3.1 一般回归

```
data(co2)
plot(co2)
x <- as.vector(time(co2))
y <- as.vector(co2)

# 对时间序列做线性回归并作预测曲线, 可以看到其拟合不太好
r <- lm( y ~ poly(x,1) + cos(2*pi*x) + sin(2*pi*x) )
plot(y~x, type='l', xlab="time", ylab="co2")
lines(predict(r)~x, lty=3, col='red', lwd=3)
# 拟合的残差图可能更加明显
plot( y-predict(r),
     main = "The residuals are not random yet",
     xlab = "Time",
     ylab = "Residuals" )
```

```
# 多项式拟合
r1 <- lm( y ~ poly(x,2) + cos(2*pi*x) + sin(2*pi*x) )
plot(y~x, type='l', xlab="time", ylab="co2")
lines(predict(r1)~x, lty=3, col='red', lwd=3)
#残差
plot( y-predict(r1),
     main = "Better residuals -- but still not random",
     xlab = "Time",
     ylab = "Residuals" )

# 进一步增加高频成分
r2 <- lm( y ~ poly(x,2) + cos(2*pi*x) + sin(2*pi*x)
                  + cos(4*pi*x) + sin(4*pi*x) )
plot(y~x, type='l', xlab="time", ylab="co2")
lines(predict(r2)~x, lty=3, col='red', lwd=3)
# 残差
plot( y-predict(r2),
     main = "Are those residuals any better?",
     xlab = "Time",
     ylab = "Residuals" )

# 对刚才的两个拟合的残差做自相关, 可以看到衰减比较
慢, 其残差非独立
op <- par(mfrow=c(2,1))
acf(y - predict(r1))
acf(y - predict(r2))
par(op)
```

42.3.2 fft()寻找趋势

寻找趋势就是滤波除去高频成分.

```
> x<-co2
> plot(x) # 周期曲线
> a=fft(x)
> a[20:(l-19)]<-0 # 去除高频成分
```

```
> y<-fft(a,inv=T) # 傅立叶逆变换
> plot(Re(y)) # 几乎成直线

# 去除越来越多的高频成分
n <- 1000
x <- cumsum(rnorm(n))+rnorm(n)
plot(x, type='l', ylab="",
    main="FFT: Removing more and more high frequencies")
for (i in 1:10) {
  y <- fft(x)
  y[(1+i):(length(y)-i)] <- 0
  y <- Re(fft(y, inverse=T)/length(y))
  lines(y, col=rainbow(10)[i])
}
```

42.4 分解时间序列

42.4.1 decompose()

函数 decompose() 用法为:

```
decompose(x, type = c("additive", "multiplicative"), filter = NULL)
```

设 T 为趋势(trend), S 为周期(seasonal), e 为噪声.

- type = "additive": 使用模型

$$Y[t] = T[t] + S[t] + e[t]$$

- type = "multiplicative": 使用模型

$$Y[t] = T[t] * S[t] + e[t]$$

- filter: 滤波系数

```
# decompose 的用法
r <- decompose(co2)
plot(r) # 绘制原始数据, trend, seasonal, 噪声 4个图
```

42.4.2 stl()

函数 stl() 是更加复杂的分解函数. 使用 Loess 方法(但不是 stats 包的 loess() 函数)分解周期性时间序列. 用法见帮助

下面是例子

```
# stl 的用法
s <- stl(co2, s.window="periodic")
r <- stl(co2, s.window="periodic")$time.series

> names(s)
[1] "time.series" "weights"    "call"       "win"        "deg"
[6] "jump"       "inner"      "outer"

# r 包括周期,整体趋势和噪声三部分
> r
          seasonal    trend    remainder
Jan 1959 -0.06100103 315.1954  0.2856440966
Feb 1959  0.59463870 315.3023  0.4130545587
Mar 1959  1.32899651 315.4093 -0.2382530567
Apr 1959  2.46904706 315.5147 -0.4237112836
May 1959  2.95704630 315.6201 -0.4471182024

op <- par(mfrow=c(4,1), mar=c(3,4,0,1), oma=c(0,0,2,0))
plot(co2)
lines(r[,2], col='blue') # 趋势线
lines(r[,2]+r[,1], col='red') # 趋势+周期
plot(r[,1],t='l',col='blue') # 周期部分
plot(r[,3]) # 噪声
acf(r[,3], main="residuals") # 噪声自相关
par(op)
mtext("STL(co2)", line=3, font=2, cex=1.2)
```

```
# 实际上 stl 有 plot
plot(s) # 绘制原始数据, trend, seasonal, 噪声 4个图
```

42.4.3 HoltWinters 分解

相关函数有 predict.HoltWinters, plot.HoltWinters. 用法见帮助.

```
> (m <- HoltWinters(co2))
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
 HoltWinters(x = co2)

Smoothing parameters:
 alpha:  0.4907075
 beta :  0.01197529
 gamma:  0.4536582

Coefficients:
          [,1]
a   364.6866567
b     0.1268701
s1    0.2812220
s2    1.0173743
s3    1.6642371
s4    2.9411121
s5    3.3487805
s6    2.5064789
s7    0.9613233
s8   -1.3122489
s9   -3.3464772
s10  -3.1988220
s11  -1.8558114
s12  -0.5254438
> plot(m)
> lines(co2,col='red')
```

下面是 [31] 15.2.14 的例子

```
data(LakeHuron)
x <- LakeHuron
before <- window(x, end=1935) # 1935年之前的数据
after <- window(x, start=1935) # 1935之后的数据
# 优化的初始值
a <- .2
b <- 0
g <- 0
model <- HoltWinters(
  before,
  alpha=a, beta=b, gamma=g)
# 对1935年后的37年预测
forecast <- predict(
  model,
  n.ahead=37,
  prediction.interval=T)

# 绘图.
plot(model, predicted.values=forecast,
    main="Holt-Winters filtering: constant model")
lines(after)
```

42.5 MA(Moving Average models)-滑 动 平均模型

参考 [31] 15.3.3

filter() 函数使用 fft 计算卷积.

```
filter(x, filter, method = c("convolution", "recursive"),
      sides = 2, circular = FALSE, init)
```

参数

- filter: 滤波系数向量, 顺序是与时间序列逆的
- method: 如果是 convolution, 使用滑动平均(默认值). 公式为[1]

  ```
  'y[i] = f[1]*x[i+o] + ... + f[p]*x[i+o-(p-1)]'
  ```

  如果是 recursive, 使用自回归. 公式为

  ```
  'y[i] = x[i] + f[1]*y[i-1] + ... + f[p]*y[i-p]'
  ```

- side: 只对 convolution 有效. =1, filter系数只对过去的值有效. =2, filter系数在延迟 0 上对称, 此时系数的个数需为奇数. 如果是偶数, 那么前面会多一个.

比较: 自回归相当于加入一个非截断窗, 可以有效的消除滑动平均的截断效应. 即其窗口为直接截断. 如果有异常值, 滑动平均往往有大的变化.

42.5.1 产生滑动平均序列

下面通过白噪声构造 MA 序列. 注意滤波系数的顺序是x的逆顺序, 自相关函数在阶数多的时候衰减慢.

```
# 简单的例子, 纯手工计算
> x=1:10
> y=filter(x, filter=c(0.5,0.3));y
 [1] 1.3 2.1 2.9 3.7 4.5 5.3 6.1 6.9 7.7  NA

# 依次取 1:k, 2:k+1 ... m:k+m-1
```

[1]详细参考帮助

```
# ma[m]=x[m:(k+m-1)]*rev(filter)
1.3=2*0.5+1*0.3
2.1=3*0.5+2*0.3


# 白噪声
n <- 200
x <- rnorm(n)

# 一阶滑动平均
y <- ( x[2:n] + x[2:n-1] ) / 2 # filter=c(1/2,1/2)

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="white noise")
plot(ts(y), xlab="", ylab="MA(1)")
acf(y, main="") # 查看其自相关系数
par(op)

# 三阶滑动平均, 滤波系数 filter=c(1/4,1/4,1/4,1/4)
y <- ( x[1:(n-3)] + x[2:(n-2)] + x[3:(n-1)] + x[4:n] )/4

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="white noise")
plot(ts(y), xlab="", ylab="MA(3)")
acf(y, main="")
par(op)

# 二阶滑动平均, 滤波系数 filter=c(3,-2,1)
y <- 3*x[3:n] - 2 * x[2:(n-1)] + x[1:(n-2)]

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="white noise")
plot(ts(y), xlab="", ylab="Momentum(2)")
acf(y, main="")
par(op)

# 使用R函数 filter 计算滑动平均
y <- filter(x, c(3,-2,1))

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="White noise")
```

```
plot(ts(y), xlab="", ylab="Momentum(2)")
acf(y, na.action=na.pass, main="")
par(op)
```

42.5.2 使用滑动平均查看序列的趋势

MA 相当于低通滤波器. 我们可以使用 fft 去掉高频成分来达到同样的效果.

下面是滑动平均的 [31] 的很好的一个例子

```
# 查看前向平均与后向平均的不同
x <- co2
plot(window(x, 1990, max(time(x))), ylab="co2")
k <- 12
lines( filter(x, rep(1/k,k)),
      col='red', lwd=3)
lines( filter(x, rep(1/k,k), sides=1),
      col='blue', lwd=3)
legend(par('usr')[1], par('usr')[4], xjust=0,
      c('smoother', 'filter'),
      lwd=3, lty=1,
      col=c('red','blue'))
```

42.6 AR(Auto-Regressive models)自回归模型

参考 [31] 15.3.4

1927, 英国统计学家 G. U. Yule 提出 AR 模型. 不久后, 英国数学家, 天文学家 G. T. Walker 爵士提出 MA 模型.

42.6.1 AR(1)

一阶 AR 模型为

$$X(n+1) = aX(n) + noise$$

当系数 $a = 1$, 实际上是随机漫步. 随机漫步的自相关衰减很慢. 下面是一个 AR(1) 的例子, 因为可以从 $y[n-1]$ 预测到 $y[n]$

```
# 简单的例子, 纯手工计算
> x=1:10; x
 [1]  1  2  3  4  5  6  7  8  9 10
> y=filter(x,c(1,2),method='r');y
 [1]    1    3    8   18   39   81  166  336  677 1359

> f=c(1,2)
> y1=x[1]; y1
[1] 1
> y2=x[2]+f[1]*y1; y2
[1] 3
> y3=x[3]+f[1]*y2+f[2]*y1; y3
[1] 8
> y4=x[4]+f[1]*y3+f[2]*y2; y4
[1] 18

# 随机漫步就是 AR(1), a=1
n <- 200
x<-rnorm(n)
y <- rep(0,n)

# 按照定义随机漫步
y[1]=x[1]
for (i in 2:n) {
  y[i] <- x[i]+y[i-1]
}

# 实际上可以使用 cumsum() 函数直接得到
# y1==y
y1 <- cumsum(x)
op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
```

```
plot(ts(x), xlab="", ylab="")
plot(ts(y1), xlab="", ylab="AR(1)")
acf(y, main="") # 随机漫步的自相关衰减很慢
par(op)

# 可以使用 filter() 函数, 参数 method='recursive'
# y2==y1==y
y2<-filter(x,filter=1,method='r')
```

42.6.2 AR(p)

p=3

```
n <- 200
x<-rnorm(n)
f=c(.3,-.7,.5)
y <- rep(0,n)
y[1:3]=x[1:3]
for (i in 4:n) {
  y[i] <- f[1]*y[i-1] +f[2]*y[i-2] + f[3]*y[i-3] + x[i]
}
op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(y), xlab="", ylab="AR(3)")
acf(y, main="", xlab="")
pacf(y, main="", xlab="")
par(op)

# 我们可以使用 arima.sim() 函数产生一个模拟的 AR(p) 过程
n <- 200
x <- arima.sim(list(ar=c(.3,-.7,.5)), n)

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="AR(3)")
acf(x, xlab="", main="")
pacf(x, xlab="", main="")
par(op)
```

42.7 平稳性与各态遍历性

42.7.1 平稳性

弱平稳: 如果时间序列 $X(t)$ 不依赖于时间 t, 并且$X(t), X(s)$的协方差只依赖于$abs(t-s)$, 说这个时间序列是弱平稳的.

平稳: 若$X(t)$同分布, 且$X(t), X(s)$的联合分布对给定$abs(t-s)$也是同分布, 此时间序列叫做平稳. 意味着弱平稳的序列其二阶平稳

例如, 如果时间序列有趋势, 即 $E(X(t))$ 不是常数, 则此时间序列不平稳.

```
n <- 200
x <- seq(0,2,length=n)
trend <- ts(sin(x))
plot(trend,
     ylim=c(-.5,1.5),
    lty=2, lwd=3, col='red',
    ylab='')

# r 是平稳的
r <- arima.sim(
  list(ar = c(0.5,-.3), ma = c(.7,.1)),
  n,
  sd=.1
)

# trend+r 是不平稳的
lines(trend+r)
```

随机漫步也不是平稳的, 期望保持 0, 但是其方差增大.

```
n <- 200
k <- 10
```

```
x <- 1:n
# 产生10个随机漫步序列
r <- matrix(nr=n,nc=k)
for (i in 1:k) {
  r[,i] <- cumsum(rnorm(n))
}
matplot(x, r,
       type = 'l',
       lty = 1,
       col = par('fg'),
       main = "A random walk is not stationnary")
abline(h=0,lty=3)
```

42.7.2 各态遍历(Ergodicity)

给定随机过程 $X(n)$, 如何估计 $X(1)$? 两种方法:

- 将此过程实现 k 次, 计算每次的 $X(1)$ 的平均
- 实现一次, 使用 $mean(X(1), X(2), \cdots$

我们需要的是第一个方法. 但是如果此时间序列是各态遍历(Ergodicity)的, 第二种也可以(好像还有其它条件)[2]

42.7.3 TODO: AR的平稳性

对于 AR(1)

$$Y(t+1) = a * Y(t) + e(t)$$

若 $abs(a) < 1$, 则可以对于 AR(1)

$$Y(t+1) - a * Y(t) = e(t)$$

[2]请参考随机过程教科书

或一般写作

$$\phi(B)Y = e$$

$\phi(B)$叫做一个算子, 而

$$\phi(u) = 1 - a_1 u - a_2 u^2 - \cdots - a_p u^p$$

当 ϕ 的所有根的模大于 1, 此时间序列是平稳的.

42.7.4 TODO: MA与可逆性(invertibility)

$$\begin{aligned} Y &= \psi(B)e \\ where \quad \psi(u) &= 1 + b_1 u + b_2 u^2 + \cdots + bqu^q \end{aligned}$$

依然要求 ψ 的所有根的模大于 1, 此时间序列是可逆的(invertible). 若无此假设, 其自相关函数不能唯一确定其平均的系数. 例如一个 MA(1) 过程

$$Y(t+1) = Z(t+1) + a * Z(t)$$

可以使用 $1/a$ 代替 a, 而其自相关系数不变.

下面是一个例子

```
> x=rnorm(100)
> f1=filter(x,f=2)
> f2=filter(x,f=0.5)
# a1==a2
> a1=acf(f1,plot=F); a1

Autocorrelations of series ‘f1’, by lag

    0     1     2     3     4     5     6     7     8     9    10
1.000 -0.080  0.024  0.035  0.049 -0.008 -0.087  0.211 -0.085  0.050 -0.045
   11    12    13    14    15    16    17    18    19    20
-0.085  0.023  0.009 -0.012 -0.008 -0.021 -0.091  0.037  0.085 -0.006
> a2=acf(f2,plot=F); a2
```

```
Autocorrelations of series 'f2', by lag

    0     1     2     3     4     5     6     7     8     9    10
 1.000 -0.080  0.024  0.035  0.049 -0.008 -0.087  0.211 -0.085  0.050 -0.045
   11    12    13    14    15    16    17    18    19    20
-0.085  0.023  0.009 -0.012 -0.008 -0.021 -0.091  0.037  0.085 -0.006
```

下面是 [31] 的例子, 供参考

```
n <- 200
ma <- 2
mai <- 1/ma
op <- par(mfrow=c(4,1), mar=c(2,4,1,2)+.1)
# 系数为 2
x <- arima.sim(list(ma=ma),n)
plot(x, xlab="", ylab="")
acf(x, xlab="", main="")
lines(0:n,
     ARMAacf(ma=ma, lag.max=n),
     lty=2, lwd=3, col='red')
# 系数为1/2
x <- arima.sim(list(ma=mai),n)
plot(x, xlab="", ylab="")
acf(x, main="", xlab="")
lines(0:n,
     ARMAacf(ma=mai, lag.max=n),
     lty=2, lwd=3, col='red')
par(op)
```

42.8 ARMA

自回归-滑动平均(Auto-Regression–Moving Average，ARMA)模型, 具有AR阶数p和MA阶数q的ARMA过程常记作ARMA(p,q).

ARMA(p,q)可以用线性差分方程进行描述

$X[t] = a[1]X[t-1] + ... + a[p]X[t-p] + e[t] + b[1]e[t-1] + ... + b[q]e[t-q]$

其中 $a[1]X[t-1] + ... + a[p]X[t-p]$为自回归, $e[t] + b[1]e[t-1] + ... + b[q]e[t-q]$是滑动平均.

显然, ARMA模型描述的是一个平稳(时不变)的线性系统.

42.9 差分-得到平稳过程

使用 ARMA 模型拟合一个时间序列, 此时间序列必须平稳.

要想得到平稳序列(简言之, 去除趋势), 可以尝试对其差分.

通常不平稳可以通过图看出, 也可以查看其自相关函数(ACF). 若平稳, ACF通常很快衰减到零(一般指数衰减)

```
data(BJsales)
op <- par(mfrow=c(3,1), mar=c(2,4,3,2)+.1)
plot(BJsales, xlab="",
   main="The trend disappears if we differentiate")

acf(BJsales, xlab="", main="")
# 差分的acf
acf(diff(BJsales), xlab="", main="",
   ylab="ACF(diff(BJsales)")
par(op)
```

42.10 ARIMA过程

参考[31] 15.3.15 和 http://wiki.mbalib.com/wiki/ARIMA模型

42.10.1 起源

1970年, 美国统计学家G.E.P. Box和英国统计学家G.M. Jenkins出版了《Time Series Analysis Forecasting and Control》一书. 系统阐述了ARIMA模型. 为纪念他们的贡献, 常常把ARIMA模型称为Box-Jenkins模型.

自回归移动平均模型(Autoregressive Integrated Moving Average Model,简记ARIMA)

42.10.2 什么是ARIMA模型

ARIMA模型全称为自回归移动平均模型(Autoregressive Integrated Moving Average Model,简记ARIMA)，是由博克思(Box)和詹金斯(Jenkins)于70年代初提出的一著名时间序列预测方法，所以又称为box-jenkins模型、博克思-詹金斯法。其中ARIMA（p，d，q）称为差分自回归移动平均模型，AR是自回归, p为自回归项; MA为移动平均，q为移动平均项数，d为时间序列成为平稳时所做的差分次数。

42.10.3 ARIMA模型的基本思想

ARIMA模型的基本思想是：将预测对象随时间推移而形成的数据序列视为一个随机序列，用一定的数学模型来近似描述这个序列。这个模型一旦被识别后就可以从时间序列的过去值及现在值来预测未来值。现代统计方法、计量经济模型在某种程度上已经能够帮助企业对未来进行预测。

一般写作

$$\phi(B)(1-B)^d X(t) = \theta(B)e(t)$$

ARIMA是不平稳的过程. ARIMA就是ARMA(p,q)过程的积分, 设积分次数为d, 则ARIMA记为ARIMA(p,d,q). 即其d次差分是平稳的. 例如对于ARMA(0,0)的1阶ARIMA过程, 即随机漫步, 其方差是随时间t增大的. 这是对其差分的主要原因.

可以使用差分直到其 ACF 迅速衰减来推断 ARIMA 的阶数 d. 你可能想对一个数据连续差分, 但是如果其 ACF 迅速衰减, 就应该停止. 过多差分是不好的.

下面几个例子, 是否参数设置合适才符合判断标准?

```
n <- 200

# MA 序列
m <- arima.sim(list(ma=c(0.5,0.5)),n )
op <- par(mfrow=c(3,1), mar=c(2,4,3,2)+.1)
plot(m)
acf(m)
pacf(m)
par(op)

# AR 序列
a <- arima.sim(list(ar=0.7),n )
op <- par(mfrow=c(3,1), mar=c(2,4,3,2)+.1)
plot(a)
acf(a)
pacf(a)
par(op)

# ARMA 序列
arma<-arima.sim(list(ar=0.7),n )
```

42.10.4 一些例子与arima()拟合

我们先看一些一般的例子. 最后使用 arima() 函数拟合.

例如可以看到对数据 co2 差分直到 4 次, 其 ACF 迅速衰减. 但是, 其周期为 12, 我们延迟12差分 1, 2 次后其 ACF 分别直线衰减, 指数衰减

```
# co2 差分 4 次
op <- par(mfrow=c(5,2), mar=c(2,4,3,2)+.1)
```

```
plot(co2, xlab="",
   main="The trend disappears if we differentiate")
acf(co2, xlab="", main="")

plot(diff(co2), xlab="",
   main="The trend diff 1")
acf(diff(co2), xlab="", main="",
   ylab="ACF(diff=1)")

plot(diff(co2,diff=2), xlab="",
   main="The trend diff 2")
acf(diff(co2,diff=2), xlab="", main="",
   ylab="ACF(diff=2)")

plot(diff(co2,diff=3), xlab="",
   main="The trend diff 3")
acf(diff(co2,diff=3), xlab="", main="",
   ylab="ACF(diff=3)")

# ACF 迅速衰减
plot(diff(co2,diff=4), xlab="",
   main="The trend diff 4")
acf(diff(co2,diff=4), xlab="", main="",
   ylab="ACF(diff=4)")
par(op)

=================
# 延迟12差分 1, 2 次后其 ACF 分别直线衰减, 指数衰减
op <- par(mfrow=c(3,2), mar=c(2,4,3,2)+.1)
plot(co2, xlab="",
   main="The trend disappears if we differentiate")
acf(co2, xlab="", main="")

# 延迟 12 差分后 ACF 直线衰减
plot(diff(co2,lag=12), xlab="",
   main="The trend diff 1")
acf(diff(co2,lag=12), xlab="", main="",
   ylab="ACF(diff=1)")

# 延迟 12 差分后 ACF 指数衰减
plot(diff(co2,diff=2,lag=12), xlab="",
```

```
   main="The trend diff 2")
acf(diff(co2,diff=2,lag=12), xlab="", main="",
   ylab="ACF(diff=2)")
par(op)
```

对于数据 sunspot.month 其1 阶差分就基本平稳了

```
op <- par(mfrow=c(4,1), mar=c(2,4,3,2)+.1)
plot(sunspot.month, xlab="", ylab="sunspot")
acf(sunspot.month, xlab="", main="")
plot(diff(sunspot.month),
    xlab="", ylab="diff(sunspot)")
acf(diff(sunspot.month), xlab="", main="")
par(op)
```

数据 JohnsonJohnson 其1 阶差分就基本平稳了

```
data(JohnsonJohnson)
x <- log(JohnsonJohnson)
op <- par(mfrow=c(4,1), mar=c(2,4,3,2)+.1)
plot(x, xlab="", ylab="JJ")
acf(x, main="")
plot(diff(x), ylab="diff(JJ)")
acf(diff(x), main="")
par(op)
```

下面看到, 去除了一个数据的趋势后, 其 1 阶差分的 ACF 指数衰减, 更高次数差分的 ACF 衰减更厉害

```
data(austres)
x <- lm(austres ~ time(austres))$res
op <- par(mfrow=c(6,1), mar=c(2,4,0,2)+.1)
plot(x)
acf(x)
```

```
plot(diff(x))
acf(diff(x))
plot(diff(x, difference=2))
acf(diff(x, difference=2))
par(op)
```

模拟一个 2 阶的 ARIMA 过程, 看到其 2 阶差分是指数衰减的.

```
n <- 200
x <- arima.sim(
  list(
    order=c(2,2,2),
    ar=c(.5,-.8),
    ma=c(.9,.6)
    ),
  n
)
op <- par(mfrow=c(3,1), mar=c(2,4,4,2)+.1)
acf(x, main="You will have to differentiate twice")
acf(diff(x), main="First derivative")
acf(diff(x, differences=2), main="Second derivative")
par(op)

# 使用 arima 函数估计其参数
> arima(x,c(2,2,2))

Call:
arima(x = x, order = c(2, 2, 2))

Coefficients:
        ar1     ar2    ma1    ma2
     0.5390  -0.8366  0.7967  0.5815
s.e.  0.0425   0.0398  0.0616  0.0622

sigma^2 estimated as 1.013:  log likelihood = -287.32,  aic = 584.64
```

42.11 如何选择模型: Box-Jenkins 方法

参考 http://en.wikipedia.org/wiki/Box-Jenkins

由统计学家 George Box 和 Gwilym Jenkins 命名. 目的是从过去的时间序列的拟合来预测未来的走势.

42.11.1 模型的步骤

1. 模型识别与选择: 确定序列平稳, 识别周期性(如果必要, 对周期差分), 绘图查看自相关(ACF)与偏自相关(PACF)来决定模型中的 MA 和 AR 成分.
2. 使用例如最大似然法或非线性最小方差法来估计模型参数. [3]
3. 检验模型: 残差应该互相独立, 残差的均值与方差应该平稳(使用 Ljung-Box test(函数 Box.test()), 或绘残差的 ACF 及 PACF 图), 若不符合要求, 回到第一步.

42.11.2 检验平稳性

第一步是识别平稳性. 可以绘出时间序列的图(run sequence plot, 也叫 run chart). 也可以由 ACF 图来查看, 如果衰减很慢, 则不平稳.

42.11.3 检验周期性

可以使用 ACF plot, a seasonal subseries plot(例如先绘制所有第一个月的数据, 然后是第二个月的数据, 然后...), or a spectral plot(谱分析).

下面是 seasonal subseries plot 的例子

[3] Brockwell and Davis, (1987,2002) for the mathematical details.[35]

```
fit <- stl(log(co2), s.window = 20, t.window = 20)
plot(fit)
op <- par(mfrow = c(2,2))
monthplot(co2, ylab = "data", cex.axis = 0.8)
monthplot(fit, choice = "seasonal", cex.axis = 0.8)
monthplot(fit, choice = "trend", cex.axis = 0.8)
monthplot(fit, choice = "remainder", type = "h", cex.axis = 0.8)
par(op)
```

42.11.4 差分得到平稳序列

Box and Jenkins 建议使用差分得到平稳序列.

但是曲线拟合, 然后数据减去拟合值也可以得到平稳序列.

42.11.5 周期差分

识别的目的是检验周期性, 若存在, 识别其MA和AR的阶数(order). 对很多时间序列来说, 周期是知道的, 并且单一的周期足够了. 例如对于月份数据, 周期往往是 AR(12) 或 MA(12). 拟合的时候一般不去除周期, 而使用 ARIMA 来代表它. 但是对其按周期差分可能对拟合会有帮助.

42.11.6 确定参数 p 和 q

一旦平稳性和周期性确定后, 下一步就是确定 AR 的参数 p 和 MA 的参数 q.

基本的方法是绘图 ACF 和 PACF.

42.11.7 AR参数p

对于 AR(1) 过程, 其 ACF 指数衰减.

但是高阶的 AR 过程其 ACF 出现指数衰减和正弦成分的混合. 需要联合使用 ACF 与 PACF. AR(p) 的 PACF 在 $p+1$ 处衰减为 0. 故我们检查 PACF 在何处基本为 0(不明显异于 0). 一般绘出 95% 的置信区间(一般的软件都会给出, 若没有, 则大概是 $\pm 2/\sqrt{N}$, N 为时间序列样本量).

42.11.8 MA参数q

MA(q) 的 ACF 在 $q+1$ 处及其之后衰减为 0. 一般也是绘出 95% 的置信区间(一般的软件都会给出, 若没有, 则大概是 $\pm 2/\sqrt{N}$, N 为时间序列样本量).

PACF 一般对 MA 没有什么帮助.

42.11.9 总结

下面的表是如何使用ACF来选择模型

ACF shape	Indicated Model
指数衰减到 0	AR模型, 使用 PACF 确定其阶数 p
正负交替衰减到 0	AR模型, 使用 PACF 确定其阶数 p
一个或多个尖突起, 其它为 0	MA模型, 阶数由衰减到 0 的点确定
几个延迟后衰减	ARMA 混合模型
全部为 0	数据是随机的
一定区间内值很大	包含周期的 AR 模型
不衰减到 0	序列不平稳

42.11.10 混合模型难以识别

实际中, ACF 和 PACF 多有随机, 使得模型识别困难. 而混合模型识别尤其困难.

虽然经验是有帮助的, 但是使用这些方法发现一个好的拟合要多多试验. 近年来发展了基于信息的 FPE (Final prediction error) and AIC (Akaike Information Criterion) 判别方法, 便于自动选择模型. [4]

42.11.11 Box-Jenkins model diagnostics

Box-Jenkins model 的诊断类似于非线性最小方差拟合的诊断. 即残差应该为白噪声(绘图查看, 或 Box-Ljung statistic).

42.11.12 TODO:例子

42.12 异方差的情况

参考 [18] 5.4

使用 ARIMA 拟合非平稳($d \neq 0$)时间序列有一个重要的假定: 残差为零均值白噪声. 即

- $E(e) = 0$
- 残差为随机序列, 即 $Cov(e_i, e_j) = 0$
- 方差齐性

均值为 0 很容易满足, 直接中心化即可. 但是残差齐性如果不满足, 则需对时间序列进行变换, 如果我们知道方差与均值(时间)之间的函数关系的话.

[4]进一步请参考 Brockwell and Davis (1987, 2002).[35]

方差齐性变换见”数据变换–稳定方差的变换” 部分.

此方法为异方差时间序列的拟合提供了精巧的方法, 但是, 实际中往往不知道异方差的函数形式. 通常只是通过残差图凭经验得到残差方差的函数. 一般的金融序列标准差与均值具有正相关关系, 故异方差函数通常假定为

$$h(\mu_t) = \mu_t^2$$

此种变换被普遍采用. 但是大量实践证明这个假设太简化了. Engle 1982 年提出了条件异方差模型.

42.13 ARCH(条件异方差模型)与GARCH等

参考 [18] 5.6

42.13.1 起源

Engle 1982 年提出了ARCH(条件异方差)模型. [5]

42.13.2 ARCH

条件异方差模型(Autoregressive conditional heteroskedasticity, ARCH)全称为自回归条件异方差模型. 其结构为

$$\begin{aligned} x_t &= f(t, x_{t-1}, x_{t-2}, \cdots) + \epsilon_t \\ \epsilon t_t &= \sqrt{h_t} e_t \\ h_t &= \omega + \sum_{j=1}^{q} \lambda_j \epsilon_{t-j}^2 \end{aligned}$$

[5]1987 年, 计量经济学家 C. Granger 提出了协整(co-integration)理论, 多变量时间序列建模过程中’变量平稳’不再是必须条件, 只要求它们的某种线性组合平稳. Granger 和 Engle 一起获得 2003 年诺贝尔经济学奖

其中 $f(t, x_{t-1}, x_{t-2}$ 为回归函数. λ_j 为系数

原理如下

- 假设数据有异方差性

$$Var(\epsilon_t) = h_t$$

- 在正态分布假定下有

$$\epsilon_t / \sqrt{h_t} \sim N(0, 1)$$

- 异方差等价于残差平方的均值

$$E(\epsilon_t^2) = h_t$$

- 使用残差平方序列的自相关函数(ACF)可以考察异方差的自相关性
- 考察结果无外乎下面两个
  - 自相关系数恒为零. 表示不能有历史数据预测未来的方差.
  - 某个自相关系数不为零, 说明异方差存在自相关性, 我们可以由历史数据预测未来的方差.

实质是使用误差平方序列的 q 阶移动平均MA(q)拟合当前方差. 由于MA(q)具有 q 阶截尾, 故 ARCH 实际上只适合具有短期异方差自相关过程的数据.

42.13.3 GARCH

GARCH 为广义条件异方差模型(Generalized autoregressive conditional heteroskedasticity, GARCH)

有些长期自相关的异方差会产生高的MA阶数, 并影响精度.

为修正此问题, Bollerslov (1985) 提出了 GARCH 模型. 其结构为

$$\begin{aligned} x_t &= f(t, x_{t-1}, x_{t-2}, \cdots) + \epsilon_t \\ \epsilon t_t &= \sqrt{h_t} e_t \\ h_t &= \omega + \sum_{i=1}^{p} \eta_i h_{t-i} \epsilon_{t-j}^2 + \sum_{j=1}^{q} \lambda_j \epsilon_{t-j}^2 \end{aligned}$$

其中 $f(t, x_{t-1}, x_{t-2}$ 为回归函数. η_i, λ_j 为系数.

实际上在 ARCH 的基础上增加了考虑异方差函数的 p 阶自相关性. 那么, ARCH(q) 就是 $p = 0$ 的 GARCH(p,q).

GARCH 要求

- 参数非负, 即 $\omega > 0, \eta_i >= 0, \lambda_j >= 0$
- 参数有界, 即 $\sum_{i=1}^{p} \eta_i + + \sum_{j=1}^{q} \lambda_j < 1$

42.13.4 TODO: 其它变体

EGARCH(指数GARCH), IGARCH(方差无穷GARCH), GARCH-M(依均值GARCH), AR-GARCH, NGARCH(非线性)

42.13.5 例子

下面是 garch() 函数在线例子

```
library(tseries)
n <- 1100
a <- c(0.1, 0.5, 0.2)  # ARCH(2) coefficients
e <- rnorm(n)
x <- double(n)
x[1:2] <- rnorm(2, sd = sqrt(a[1]/(1.0-a[2]-a[3])))
```

```
# 产生 ARCH(2)过程
for(i in 3:n)  # Generate ARCH(2) process
{
x[i] <- e[i]*sqrt(a[1]+a[2]*x[i-1]^2+a[3]*x[i-2]^2)
}
x <- ts(x[101:1100])
# 拟合ARCH(2)
x.arch <- garch(x, order = c(0,2))  # Fit ARCH(2)
# 诊断检验, 检验残差是否随机. 使用方法: Jarque Bera Test, Box-Ljung test
summary(x.arch)                    # Diagnostic tests
plot(x.arch)

data(EuStockMarkets)
dax <- diff(log(EuStockMarkets))[,"DAX"]
dax.garch <- garch(dax)  # Fit a GARCH(1,1) to DAX returns
summary(dax.garch)      # ARCH effects are filtered. However,
plot(dax.garch)         # conditional normality seems to be violated
```

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \cdots + \alpha_q \epsilon_{t-q}^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2$$

42.14 co-integration(协整)

包 urca 执行单位根检验和co-integration分析

42.14.1 起源

1987 年, 计量经济学家 C. Granger 与 Engle 提出了协整(co-integration)理论, 多变量时间序列建模过程中'变量平稳'不再是必须条件, 只要求它们的某种线性组合平稳. [6]

[6]Engle 1982 年提出了ARCH(条件异方差)模型. Granger 和 Engle 一起获得 2003 年诺贝尔经济学奖

42.14.2 概念

有些序列的自身虽然非平稳, 但是某些序列之间具有紧密长期的均衡关系.

例如: 农村家庭人均纯收入对数序列 (lnx_i) 和人均生活消费 (lny_i), 自身都是非平稳的, 但是它们之间具有非常稳定的线性相关关系. 构造回归模型

$$y_i = \beta_0 + \sum_{i=1}^{k} \beta_i x_{it} + \epsilon_t$$

假定回归残差 ϵ_t 平稳, 我们称 x_i, y_i 之间具有协整关系.

意味着, 我们建模不需要所有序列平稳, 只需要有协整关系即可. 这极大拓宽了动态建模的范围.

42.14.3 Phillips-Ouliaris test

Phillips-Ouliaris test 检验多元时间序列是否协整.

若 x 为多元时间序列. Phillips-Ouliaris test 的零假设是: x 非协整.

下面是 R 的例子

```
> library(tseries)
> # no cointegration (非协整)
> x <- ts(diffinv(matrix(rnorm(2000),1000,2)))
> po.test(x)

        Phillips-Ouliaris Cointegration Test

data:  x
Phillips-Ouliaris demeaned = -8.7542, Truncation lag parameter = 10,
p-value = 0.15
```

```
Warning message:
In po.test(x) : p-value greater than printed p-value

# 协整
> x <- diffinv(rnorm(1000))
> y <- 2.0-3.0*x+rnorm(x,sd=5)
> z <- ts(cbind(x,y))  # cointegrated
> po.test(z)

       Phillips-Ouliaris Cointegration Test

data:  z
Phillips-Ouliaris demeaned = -1170.862, Truncation lag parameter = 10,
p-value = 0.01

Warning message:
In po.test(z) : p-value smaller than printed p-value
```

Chapter 43

VAR模型(少例子)

来自 http://en.wikipedia.org/wiki/Vector_autoregression

http://en.wikipedia.org/wiki/General_matrix_notation_of_a_VAR(p)

部分翻译, 部分未翻译. 讲解很好.

VAR(Vector autoregression)模型(向量自回归模型)是一个经济模型, 用来发掘多元时间序列的变化和相互依赖关系. 是AR模型的推广.

43.1 简化模型的定义

其数学描述如下

43.1.1 Var(p)

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + e_t$$

43.1.2 大矩阵形式

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \begin{bmatrix} a_{1,1}^1 & a_{1,2}^1 & \cdots & a_{1,k}^1 \\ a_{2,1}^1 & a_{2,2}^1 & \cdots & a_{2,k}^1 \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1}^1 & a_{k,2}^1 & \cdots & a_{k,k}^1 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ \vdots \\ y_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a_{1,1}^p & a_{1,2}^p & \cdots & a_{1,k}^p \\ a_{2,1}^p & a_{2,2}^p & \cdots & a_{2,k}^p \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1}^p & a_{k,2}^p & \cdots & a_{k,k}^p \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ \vdots \\ y_{k,t-p} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix}$$

43.1.3 方程式形式

Rewriting the y variables one to one gives:

$$\begin{aligned} y_{1,t} &= c_1 + a_{1,1}^1 y_{1,t-1} + a_{1,2}^1 y_{2,t-1} + \cdots + a_{1,k}^1 y_{k,t-1} + \cdots + a_{1,1}^p y_{1,t-p} + a_{1,2}^p y_{2,t-p} + \cdots + a_{1,k}^p y_{k,t-p} \\ y_{2,t} &= c_2 + a_{2,1}^1 y_{1,t-1} + a_{2,2}^1 y_{2,t-1} + \cdots + a_{2,k}^1 y_{k,t-1} + \cdots + a_{2,1}^p y_{1,t-p} + a_{2,2}^p y_{2,t-p} + \cdots + a_{2,k}^p y_{k,t-p} \\ y_{k,t} &= c_k + a_{k,1}^1 y_{1,t-1} + a_{k,2}^1 y_{2,t-1} + \cdots + a_{k,k}^1 y_{k,t-1} + \cdots + a_{k,1}^p y_{1,t-p} + a_{k,2}^p y_{2,t-p} + \cdots + a_{k,k}^p y_{k,t-p} \end{aligned}$$

43.1.4 浓缩矩阵

$$Y = BZ + U$$

其中

$$Y = \begin{bmatrix} y_p & y_{p+1} & \cdots & y_T \end{bmatrix} = \begin{bmatrix} y_{1,p} & y_{1,p+1} & \cdots & y_{1,T} \\ y_{2,p} & y_{2,p+1} & \cdots & y_{2,T} \\ \vdots & \vdots & \vdots & \vdots \\ y_{k,p} & y_{k,p+1} & \cdots & y_{k,T} \end{bmatrix}$$

$$B = \begin{bmatrix} c & A_1 & A_2 & \cdots & A_p \end{bmatrix} = \begin{bmatrix} c_1 & a_{1,1}^1 & a_{1,2}^1 & \cdots & a_{1,k}^1 & \cdots & a_{1,1}^p & a_{1,2}^p & \cdots & a_{1,k}^p \\ c_2 & a_{2,1}^1 & a_{2,2}^1 & \cdots & a_{2,k}^1 & \cdots & a_{2,1}^p & a_{2,2}^p & \cdots & a_{2,k}^p \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ c_k & a_{k,1}^1 & a_{k,2}^1 & \cdots & a_{k,k}^1 & \cdots & a_{k,1}^p & a_{k,2}^p & \cdots & a_{k,k}^p \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_{p-1} & y_p & \cdots & y_{T-1} \\ y_{p-2} & y_{p-1} & \cdots & y_{T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_0 & y_1 & \cdots & y_{T-p} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_{1,p-1} & y_{1,p} & \cdots & y_{1,T-1} \\ y_{2,p-1} & y_{2,p} & \cdots & y_{2,T-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k,p-1} & y_{k,p} & \cdots & y_{k,T-1} \\ y_{1,p-2} & y_{1,p-1} & \cdots & y_{1,T-2} \\ y_{2,p-2} & y_{2,p-1} & \cdots & y_{2,T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k,p-2} & y_{k,p-1} & \cdots & y_{k,T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1,0} & y_{1,1} & \cdots & y_{1,T-p} \\ y_{2,0} & y_{2,1} & \cdots & y_{2,T-p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k,0} & y_{k,1} & \cdots & y_{k,T-p} \end{bmatrix}$$

and

$$U = \begin{bmatrix} e_p & e_{p+1} & \cdots & e_T \end{bmatrix} = \begin{bmatrix} e_{1,p} & e_{1,p+1} & \cdots & e_{1,T} \\ e_{2,p} & e_{2,p+1} & \cdots & e_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ e_{k,p} & e_{k,p+1} & \cdots & e_{k,T} \end{bmatrix}.$$

下面就可以解系数矩阵B了(例如,使用一般最小方差(ordinary least squares)估计$Y \approx BZ$)

43.1.5 解 释

VAR模 型 描 述k维 数 据 从 时 间$t = 1, ..., T$的 变 化, 将 之 看 作 它 自 己 过 去 的 一 个 线 性 函 数. 时 间t的 变 量y_t是 一 个$k * 1$的 向 量, 例 如, 第i个 变 量 是GDP, 那 么$y_{i,t}$是 时 间t的GDP.

其 中e_t满 足

1. $\mathrm{E}(e_t) = 0$ 误 差 均 值 为 零
2. $\mathrm{E}(e_t e_t') = \Omega$ 误 差 协 方 差 矩 阵 是 正 则 的
3. $\mathrm{E}(e_t e_{t-k}') = 0$ 对 于$k \neq 0$, 误 差 互 协 方 差 为 零

43.1.6 Order of integration of the variables

Note that all the variables used have to be of the same order of integration. We have so the following cases:

- All the variables are I(0) (stationary): one is in the standard case, ie. a VAR in level
- All the variables are I(d) (non-stationary) with d¿1:
  - The variables are cointegrated: the error correction term has to be included in the VAR. The model becomes a Vector error correction model (VECM) which can be seen as a restricted VAR.
  - The variables are not cointegrated: the variables have first to be differenced d times and one has a VAR in difference.

43.1.7 简 单 例 子

二 维 向 量 的VAR(1)可 以 写 作

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \end{bmatrix}$$

或等价的

$$\begin{aligned} y_{1,t} &= c_1 + A_{1,1}y_{1,t-1} + A_{1,2}y_{2,t-1} + e_{1,t} \\ y_{2,t} &= c_2 + A_{2,1}y_{1,t-1} + A_{2,2}y_{2,t-1} + e_{2,t} \end{aligned}$$

注意到, 每个向量有一个方程式, 每个向量当前的状态不不仅依赖于自己的过去状态, 还依赖于其它序列的过去状态

43.1.8 将VAR(p)写作VAR(1)

通过变换系数, 我们总可以把延迟为p的形式写作延迟为1的形式. 例如VAR(2)模型

$$yt = c + A1yt`1 + A2yt`2 + et$$

可以写作VAR(1)模型

$$\begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} = \begin{bmatrix} c \\ 0 \end{bmatrix} + \begin{bmatrix} A_1 & A_2 \\ I & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} + \begin{bmatrix} e_t \\ 0 \end{bmatrix}$$

其中I为单位矩阵.

VAR(1)形式分析起来更加方便, 写起来更简便.

43.2 Structural vs. reduced form

43.2.1 Structural VAR

p延迟的structural VAR为

$$B_0 y_t = c_0 + B_1 y_{t-1} + B_2 y_{t-2} + \cdots + B_p y_{t-p} + \epsilon_t$$

其中c_0为$k1$常数向量, B_i是kk矩阵(for every $i = 0, ..., p$), $_t$为$k1$误差向量. B_0矩阵的主对角成分都为1.

误差项$_t$(structural shocks)满足定义中条件(1) - (3), 即误差项$_t$(structural shocks)不相关($E(_t{}'_t) = \ = 0$.

例如二维的VAR(1)为

$$\begin{bmatrix} 1 & B_{0;1,2} \\ B_{0;2,1} & 1 \end{bmatrix} \begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} c_{0;1} \\ c_{0;2} \end{bmatrix} + \begin{bmatrix} B_{1;1,1} & B_{1;1,2} \\ B_{1;2,1} & B_{1;2,2} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{bmatrix}$$

其中

$$\Sigma = \mathrm{E}(\epsilon_t \epsilon_t') = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$$

显式写出第一个方程式, 并带入$y_{2,t}$到右边得到

$$y_{1,t} = c_{0;1} - B_{0;1,2} y_{2,t} + B_{1;1,1} y_{1,t-1} + B_{1;1,2} y_{2,t-2} + \epsilon_{1,t}$$

注意到如果$B_{0;1,2}$不等于零, $y_{2,t}$可以影响同时的$_{1,t}$. 这同B_0为单位矩阵的情况不同, 其$y_{2,t}$可以直接影响$y_{1,t+1}$从而影响将来的值, 但不是$y_{1,t}$.

因为普通最小方差估计(ordinary least squares estimation)确定structural VAR参数的问题, 即产生无解估计(yield inconsistent parameter estimates), 我们可以将其表示为简化方式(reduced form).

下面未翻译

From an economic point of view it is considered that, if the joint dynamics of a se

1. Error terms are not correlated. The structural, economic shocks which dri

2. Variables can have a contemporaneous impact on other variables. This is a

43.2.2 Reduced VAR

左乘B_0的逆

$$y_t = B_0^{-1} c_0 + B_0^{-1} B_1 y_{t-1} + B_0^{-1} B_2 y_{t-2} + \cdots + B_0^{-1} B_p y_{t-p} + B_0^{-1} \epsilon_t$$

并表示为

$$B_0^{-1}c_0 = c, \quad B_0^{-1}B_i = A_i \text{ for } i = 1, \ldots, p \text{ and } B_0^{-1}\epsilon_t = e_t$$

可以得到p阶简化的VAR模型

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + e_t$$

下面未翻译

Note that in the reduced form all right hand side variables are predetermined at ti

However, the error terms in the reduced VAR are composites of the structural shocks

$$\Omega = \mathrm{E}(e_t e_t') = \mathrm{E}(B_0^{-1}\epsilon_t \epsilon_t'(B_0^{-1})') = B_0^{-1}\Sigma(B_0^{-1})'$$

43.3 估计

43.3.1 估计回归系数

由精简形式

$$Y = BZ + U$$

得到B的Multivariate Least Square (MLS):

$$\hat{B} = YZ'(ZZ')^{-1}$$

还可以写作

$$\mathrm{Vec}(\hat{B}) = ((ZZ')^{-1}Z \otimes I_k)\ \mathrm{Vec}(Y)$$

下面未翻译

Where \otimes denotes the Kronecker product and Vec the vectorization of the matrix

This estimator is consistent and asymptotically efficient. It is furthemore equal t

* As the explanatory variables are the same in each equation, the Multivariate M

43.3.2 误差协方差矩阵的估计

As in the standard case, the MLE estimator of the covariance matrix differs from the OLS estimator.

MLE estimator: $\hat{\Sigma} = \frac{1}{T}\sum_{t=1}^{T} \hat{\epsilon}_t \hat{\epsilon}_t'$

OLS estimator: $\hat{\Sigma} = \frac{1}{T-kp-1}\sum_{t=1}^{T} \hat{\epsilon}_t \hat{\epsilon}_t'$ for a model with a constant, k variables and p lags

In a matrix notation, this gives:

$$\hat{\Sigma} = \frac{1}{T-kp-1}(Y - \hat{B}Z)(Y - \hat{B}Z)'$$

43.3.3 参数协方差矩阵的估计

$$\widehat{\mathrm{Cov}}(\mathrm{Vec}(\hat{B})) = (ZZ')^{-1} \otimes \hat{\Sigma}$$

43.4 参考文献

略

43.5 相关函数

ar() 可以实现平稳序列的VAR分析

dse包, dlm包, mAr包等都有相关函数.

Chapter 44

卡尔曼滤波(理论,少例子)

参考 http://zh.wikipedia.org/wiki/卡尔曼滤波

参考"卡尔曼滤波器最好的入门教程" http://bbs.powershock.cn/thread-45-1-1.html

参考 http://www.cs.unc.edu/ welch/kalman/

Andrew D. Straw 非常好的一个介绍,并一个python的例子 http://www.cs.unc.edu/ welch/kalman/kalmanIntro.html

姚旭晨翻译的Andrew D. Straw 的介绍并改编的matlab例子,非常好 http://yaoxuchen.googlepages.com/kalman

为方便查看,下面内容全文来自 http://zh.wikipedia.org/wiki/卡尔曼滤波

44.1 介绍

卡尔曼滤波是一种高效率的递归滤波器(自回归滤波器),它能够从一系列的不完全及包含噪声的测量(英文:measurement)中，估计动态系统的状态。

44.2 应用实例

卡尔曼滤波的一个典型实例是从一组有限的，包含噪声的，对物体位置的观察序列（可能有偏差）预测出物体的位置的坐标及速度。在很多工程应用(如雷达、计算机视觉)中都可以找到它的身影。同时，卡尔曼滤波也是控制理论以及控制系统工程中的一个重要课题。

例如,对于雷达来说，人们感兴趣的是其能够跟踪目标。但目标的位置、速度、加速度的测量值往往在任何时候都有噪声。卡尔曼滤波利用目标的动态信息，设法去掉噪声的影响，得到一个关于目标位置的好的估计。这个估计可以是对当前目标位置的估计(滤波)，也可以是对于将来位置的估计(预测)，也可以是对过去位置的估计(插值或平滑)。

44.3 命名

这种滤波方法以它的发明者鲁道夫.E.卡尔曼（Rudolph E. Kalman）命名，但是根据文献可知实际上Peter Swerling在更早之前就提出了一种类似的算法。

斯坦利.施密特(Stanley Schmidt)首次实现了卡尔曼滤波器。卡尔曼在NASA埃姆斯研究中心访问时，发现他的方法对于解决阿波罗计划的轨道预测很有用，后来阿波罗飞船的导航电脑便使用了这种滤波器。

关于这种滤波器的论文由Swerling (1958)、Kalman (1960)与Kalman and Bucy (1961)发表。

目前,卡尔曼滤波已经有很多不同的实现.卡尔曼最初提出的形式现在一般称为简单卡尔曼滤波器。除此以外，还有施密特扩展滤波器、信息滤波器以及很多Bierman, Thornton 开发的平方根滤波器的变种。也许最常见的卡尔曼滤波器是锁相环，它在收音机、计算机和几乎任何视频或通讯设备中广泛存在。

以下的讨论需要线性代数以及概率论的一般知识。

44.4 基本动态系统模型

卡尔曼滤波建立在线性代数和隐马尔可夫模型(hidden Markov model)上。其基本动态系统可以用一个马尔可夫链表示，该马尔可夫链建立在一个被高斯噪声(即正态分布的噪声)干扰的线性算子上的。系统的状态可以用一个元素为实数的向量表示。随着离散时间的每一个增加,这个线性算子就会作用在当前状态上，产生一个新的状态,并也会带入一些噪声,同时系统的一些已知的控制器的控制信息也会被加入。同时，另一个受噪声干扰的线性算子产生出这些隐含状态的可见输出。

为了从一系列有噪声的观察数据中用卡尔曼滤波器估计出被观察过程的内部状态，我们必须把这个过程在卡尔曼滤波的框架下建立模型。也就是说对于每一步k，定义矩阵F_k, H_k, Q_k, R_k，有时也需要定义B_k，如下图(略, 卡尔曼滤波器的模型。圆圈代表向量，方块代表矩阵，星号代表高斯噪声，其协方差矩阵在右下方标出。)

卡尔曼滤波模型假设k时刻的真实状态是从(k 1)时刻的状态演化而来，符合下式

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k$$

其中

- F_k 是作用在$x_{k`1}$上的状态变换模型（/矩阵/矢量）。
- B_k 是作用在控制器向量u_k上的输入－控制模型。
- w_k 是过程噪声，并假定其符合均值为零，协方差矩阵为Q_k的多元正态分布。

$$\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$$

时刻k，对真实状态 x_k的一个测量z_k满足下式:

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$$

其中H_k是观测模型,它把真实状态空间映射成观测空间，v_k 是观测噪声，其均值为零，协方差为R_k,且服从正态分布。

$$\mathbf{v}_k \sim N(0, \mathbf{R}_k)$$

初始状态以及每一时刻的噪声$x_0, w_1, ..., w_k, v_1...v_k$ 都为认为是互相独立的.

实际上，很多真实世界的动态系统都并不确切的符合这个模型；但是由于卡尔曼滤波器被设计在有噪声的情况下工作,一个近似的符合已经可以使这个滤波器非常有用了。更多其它更复杂的卡尔曼滤波器的变种，在下边讨论中有描述。

44.5 卡尔曼滤波器

卡尔曼滤波是一种递归的估计，即只要获知上一时刻状态的估计值以及当前状态的观测值就可以计算出当前状态的估计值，因此不需要记录观测或者估计的历史信息。卡尔曼滤波器与大多数滤波器不同之处，在于它是一种纯粹的时域滤波器，它不需要像低通滤波器等频域滤波器那样，需要在频域设计再转换到时域实现。

卡尔曼滤波器的状态由以下两个变量表示：

- $\hat{\mathbf{x}}_{k|k}$，在时刻k 的状态的估计；
- $\mathbf{P}_{k|k}$，误差相关矩阵，度量估计值的精确程度。

卡尔曼滤波器的操作包括两个阶段：预测与更新。在预测阶段，滤波器使用上一状态的估计，做出对当前状态的估计。在更新阶段，滤波器利用对当前状态的观测值优化在预测阶段获得的预测值，以获得一个更精确的新估计值。

44.5.1 预测

- $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k$ (预测状态)
- $\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^T + \mathbf{Q}_k$ (预测估计协方差)

44.5.2 更新

- $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}$ (测量余量，measurement residual)
- $\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k$ (测量余量协方差)
- $\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}$ (卡尔曼增益)
- $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k$ (更新的状态估计)
- $\mathbf{P}_{k|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}$ (更新的协方差估计)

使用上述公式计算$\mathbf{P}_{k|k}$仅在最优卡尔曼增益的时候有效。使用其他增益的话，公式要复杂一些，请参见推导。

44.5.3 不变量(Invariant)

如果模型准确，而且$\hat{\mathbf{x}}_{0|0}$与$\mathbf{P}_{0|0}$的值准确的反映了最初状态的分布，那么以下不变量就保持不变：所有估计的误差均值为零

- $\mathrm{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] = \mathrm{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}] = 0$
- $\mathrm{E}[\tilde{\mathbf{y}}_k] = 0$

且协方差矩阵准确的反映了估计的协方差:

- $\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})$
- $\mathbf{P}_{k|k-1} = \mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})$
- $\mathbf{S}_k = \mathrm{cov}(\tilde{\mathbf{y}}_k)$

请注意，其中$\mathrm{E}[\mathbf{a}]$表示a的期望值, $\mathrm{cov}(\mathbf{a}) = \mathrm{E}[\mathbf{a}\mathbf{a}^T]$。

44.6 实例

考虑在无摩擦的、无限长的直轨道上的一辆车。该车最初停在位置0处,但时不时受到随机的冲击。我们每隔Δt秒即测量车的位置，但是这个测量是非精确的；我们想建立一个关于其位置以及速度的模型。我们来看如何推导出这个模型以及如何从这个模型得到卡尔曼滤波器。

因为车上无动力,所以我们可以忽略掉B_k和u_k。由于F、H、R和Q是常数，所以时间下标可以去掉。

车的位置以及速度(或者更加一般的，一个粒子的运动状态)可以被线性状态空间描述如下:

$$\mathbf{x}_k = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

其中$\dot{x}$是速度，也就是位置对于时间的导数。我们假设在(k`1)时刻与k时刻之间，车受到a_k的加速度,其符合均值为0，标准差为σ_a的正态分布。根据牛顿运动定律，我们可以推出

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}a_k$$

其中

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

且

$$\mathbf{G} = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$$

我们可以发现((因为 σ_a 是一个标量)

$$\mathbf{Q} = \mathrm{cov}(\mathbf{G}a) = \mathrm{E}[(\mathbf{G}a)(\mathbf{G}a)^T] = \mathbf{G}\mathrm{E}[a^2]\mathbf{G}^T = \mathbf{G}[\sigma_a^2]\mathbf{G}^T = \sigma_a^2\mathbf{G}\mathbf{G}^T$$

在每一时刻,我们对其位置进行测量,测量受到噪声干扰.我们假设噪声服从正态分布，均值为0，标准差为σ_z。

$$\mathbf{z}_k = \mathbf{H}\ \mathbf{x}_k + \mathbf{v}_k$$

其中

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

且

$$\mathbf{R} = \mathrm{E}[\mathbf{v}_k \mathbf{v}_k^T] = \begin{bmatrix} \sigma_z^2 \end{bmatrix}$$

如果我们知道足够精确的车最初的位置，那么我们可以初始化

$$\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

并且,我们告诉滤波器我们知道确切的初始位置,我们给出一个协方差矩阵：

$$\mathbf{P}_{0|0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

如果我们不确切的知道最初的位置与速度，那么协方差矩阵可以初始化为一个对角线元素是B的矩阵，B取一个合适的比较大的数。

$$\mathbf{P}_{0|0} = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}$$

此时，与使用模型中已有信息相比，滤波器更倾向于使用初次测量值的信息。

44.7 推导

44.7.1 推导后验协方差矩阵

按照上边的定义，我们从误差协方差Pk—k开始推导如下：

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})$$

代入$\hat{\mathbf{x}}_{k|k}$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k))$$

再代入 $\tilde{\mathbf{y}}_k$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})))$$

与 $\mathbf{z}_k$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})))$$

整理误差向量，得

$$\mathbf{P}_{k|k} = \mathrm{cov}((I - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) - \mathbf{K}_k\mathbf{v}_k)$$

因为测量误差vk 与其他项是非相关的，因此有

$$\mathbf{P}_{k|k} = \mathrm{cov}((I - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})) + \mathrm{cov}(\mathbf{K}_k\mathbf{v}_k)$$

利用协方差矩阵的性质，此式可以写作

$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(I - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathrm{cov}(\mathbf{v}_k)\mathbf{K}_k^T$$

使用不变量$P_{k|k-1}$以及R_k的定义这一项可以写作：$\mathbf{P}_{k|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(I - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T$ 这一公式对于任何卡尔曼增益Kk都成立。如果Kk是最优卡尔曼增益，则可以进一步简化，请见下文。

44.7.2 最优卡尔曼增益的推导

卡尔曼滤波器是一个最小均方误差估计器，后验状态误差估计(英文:a posteriori state estimate)是

$$\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}$$

我们最小化这个矢量幅度平方的期望值，$\mathrm{E}[|\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}|^2]$，这等同于最小化后验估计协方差矩阵 $P_{k|k}$的迹(trace)。将上面方程中的项展开、抵消，得到：

$$\begin{aligned}\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)\mathbf{K}_k^T \\ &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T\end{aligned}$$

当矩阵导数是 0 的时候得到$P_{k|k}$的迹(trace)的最小值：

$$\frac{d\,\mathrm{tr}(\mathbf{P}_{k|k})}{d\,\mathbf{K}_k} = -2(\mathbf{H}_k\mathbf{P}_{k|k-1})^T + 2\mathbf{K}_k\mathbf{S}_k = 0$$

此处须用到一个常用的式子, 如下：

$$\frac{d\,\mathrm{tr}(\mathbf{BAC})}{d\,\mathbf{A}} = B^T C^T$$

从这个方程解出卡尔曼增益K_k：

$$\begin{aligned}\mathbf{K}_k\mathbf{S}_k &= (\mathbf{H}_k\mathbf{P}_{k|k-1})^T = \mathbf{P}_{k|k-1}\mathbf{H}_k^T \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}\end{aligned}$$

这个增益称为最优卡尔曼增益，在使用时得到最小均方误差。

44.7.3 后验误差协方差公式的化简

在卡尔曼增益等于上面导出的最优值时，计算后验协方差的公式可以进行简化。在卡尔曼增益公式两侧的右边都乘以 $S_kK_k^T$ 得到

$$\mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T$$

根据上面后验误差协方差展开公式，

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T$$

最后两项可以抵消，得到

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}.$$

这个公式的计算比较简单，所以实际中总是使用这个公式，但是需注意这公式仅在使用最优卡尔曼增益的时候它才成立。如果算术精度总是很低而导致数值稳定性出现问题，或者特意使用非最优卡尔曼增益，那么就不能使用这个简化；必须使用上面导出的后验误差协方差公式。

44.8 与递归Bayesian估计之间的关系

假设真正的状态是无法观察的马尔可夫过程，测量结果是从隐性马尔可夫模型观察到的状态。

Image:HMMKalmanFilterDerivation.png

根据马尔可夫假设，真正的状态仅受最近一个状态影响而与其它以前状态无关。

$$p(\mathbf{x}_k|\mathbf{x}_0, \ldots, \mathbf{x}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$$

与此类似，在时刻 k 测量只与当前状态有关而与其它状态无关。

$$p(\mathbf{z}_k|\mathbf{x}_0, \ldots, \mathbf{x}_k) = p(\mathbf{z}_k|\mathbf{x}_k)$$

根据这些假设，隐性马尔可夫模型所有状态的概率分布可以简化为：

$$p(\mathbf{x}_0, \ldots, \mathbf{x}_k, \mathbf{z}_1, \ldots, \mathbf{z}_k) = p(\mathbf{x}_0)\prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}_i)p(\mathbf{x}_i|\mathbf{x}_{i-1})$$

然而，当卡尔曼滤波器用来估计状态x的时候，我们感兴趣的机率分布，是基于目前为止所有个测量值来得到的当前状态之机率分布

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\, d\mathbf{x}_{k-1}$$

44.9 信息滤波器

44.9.1 非线性滤波器

基本卡尔曼滤波器（The basic Kalman filter）是限制在线性的假设之下。然而，大部份非平凡的（non-trial）的系统都是非线性系统。其中的"非线性性质"（non-linearity ）可能是伴随存在过程模型（process model）中或观测模型（observation model）中，或者两者兼有之。

44.9.2 扩展卡尔曼滤波器

在扩展卡尔曼滤波器（Extended Kalman Filter，简称EKF）中状态转换和观测模型不需要是状态的线性函数，可替换为（可微的）函数。

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \\ \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{v}_k) \end{aligned}$$

函数f可以用来从过去的估计值中计算预测的状态，相似的，函数h可以用来以预测的状态计算预测的测量值。然而f和h不能直接的应用在协方差中，取而代之的是计算偏导矩阵(Jacobian)。

在每一步中使用当前的估计状态计算Jacobian矩阵，这几个矩阵可以用在卡尔曼滤波器的方程中。这个过程，实质上将非线性的函数在当前估计值处线性化了。

这样一来，卡尔曼滤波器的等式为：

预测

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= f(\mathbf{x}_{k-1}, \mathbf{u}_k, 0) \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k\end{aligned}$$

使用Jacobians矩阵更新模型

$$\begin{aligned}\mathbf{F}_k &= \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k} \\ \mathbf{H}_k &= \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k-1}}\end{aligned}$$

更新

$$\begin{aligned}\tilde{\mathbf{y}}_k &= \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}, 0) \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\ \mathbf{P}_{k|k} &= (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}\end{aligned}$$

预测

如同扩展卡尔曼滤波器（EKF）一样, UKF的预测过程可以独立于UKF的更新过程之外，与一个线性的（或者确实是扩展卡尔曼滤波器的）更新过程合并来使用；或者，UKF的预测过程与更新过程在上述中地位互换亦可。

44.10 应用

- 自动驾驶仪
- 动态定位系统
- 经济学, 特别是宏观经济学，时间序列模型, 以及计量经济学

- 惯性引导系统
- 雷达跟踪器
- 卫星导航系统

44.11 参见

快速卡尔曼滤波

比较: 维纳滤波及 the multimodal Particle filter estimator.

44.12 例子

44.12.1 Andrew D. Straw的例子

最初来自 Andrew D. Straw http://www.scipy.org/Cookbook/KalmanFiltering

姚旭晨改编为matlab http://yaoxuchen.googlepages.com/kalman

me 改编为 R

```
# Kalman filter example demo in Matlab

# This M code is modified from Andrew D. Straw's Python
# implementation of Kalman filter algorithm.
# The original code is here:
# http://www.scipy.org/Cookbook/KalmanFiltering
# Below is the Python version's comments:

# Kalman filter example demo in Python

# A Python implementation of the example given in pages 11-15 of "An
# Introduction to the Kalman Filter" by Greg Welch and Gary Bishop,
# University of North Carolina at Chapel Hill, Department of Computer
```

```
# Science, TR 95-041,
# http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html

# by Andrew D. Straw

# matlab by Xuchen Yao
# R by me

# intial parameters
n_iter = 50;
sz = c(n_iter, 1); # size of array
x = -0.37727; # truth value (typo in example at top of p. 13 calls this z)
z = x + sqrt(0.01)*rnorm(n_iter); # observations (normal about x, sigma=0.1)

Q = 1e-5; # process variance

# allocate space for arrays
xhat=rep(0,n_iter);      # a posteri estimate of x
P=rep(0,n_iter);         # a posteri error estimate
xhatminus=rep(0,n_iter); # a priori estimate of x
Pminus=rep(0,n_iter);    # a priori error estimate
K=rep(0,n_iter);         # gain or blending factor

R = 0.01; # estimate of measurement variance, change to see effect

# intial guesses
xhat[1] = 0.0;
P[1] = 1.0;

for (k in 2:n_iter){
# time update(predict)
   xhatminus[k] = xhat[k-1];
   Pminus[k] = P[k-1]+Q;

# measurement update
   K[k] = Pminus[k]/( Pminus[k]+R );
   xhat[k] = xhatminus[k]+K[k]*(z[k]-xhatminus[k]);
   P[k] = (1-K[k])*Pminus[k];
}
# plot predicted value
plot(z, xlab='Iteration',ylab='Voltage')
```

```
lines(xhat,col='red')
lines(x*rep(1,50))

# plot error
valid_iter = 2:n_iter
plot(Pminus[valid_iter]~valid_iter,t='l');
```

44.12.2 kfilter()函 数

它 属 于sspir包

Chapter 45

谱分析

45.1 推荐

《小波与傅里叶分析基础》[17], 入门极佳, 工科足够.

signal 包: 是一个类似Matlab/Octave信号处理命令的工具. 包含滤波, 采样, 差值, 可视化等命令. 命令全, 比较方便. (R自带的命令不太全)

45.2 介绍

谱分析是根据时间序列的频域性质对其统计推断的方法.

一些具有周期的序列其周期如果是复合的, 则很难通过图来看出. 这时候需要使用频域的方法.

45.3 傅立叶变换(FFT)

```
> x=c(15 , -2, 12, 20, -5 , 0 , -8 , -4 , -8, -22)
> ft=fft(x); ft
```

```
 [1]  -2.00000+ 0.00000i   2.39261-55.36555i -12.61397-13.81682i
 [4]  28.10739+ 3.98825i  51.11397-13.79658i  14.00000- 0.00000i
 [7]  51.11397+13.79658i  28.10739- 3.98825i -12.61397+13.81682i
[10]   2.39261+55.36555i
> Mod(ft) # 相当于 abs(ft)
 [1]  2.00000 55.41722 18.70873 28.38893 52.94321 14.00000 52.94321 28.38893
 [9] 18.70873 55.41722
> abs(ft)
 [1]  2.00000 55.41722 18.70873 28.38893 52.94321 14.00000 52.94321 28.38893
 [9] 18.70873 55.41722

> ft=fft(x); ft
 [1]  -2.00000+ 0.00000i   2.39261-55.36555i -12.61397-13.81682i
 [4]  28.10739+ 3.98825i  51.11397-13.79658i  14.00000- 0.00000i
 [7]  51.11397+13.79658i  28.10739- 3.98825i -12.61397+13.81682i
[10]   2.39261+55.36555i

x<-seq(0,1,by=0.001)

# y 在 100, 200, 300 处有峰值
y <- sin(200*pi*x) +3*sin(400*pi*x)+6*sin(600*pi*x)
op <- par(mfrow=c(3,1))
plot(Mod(fft(y)),t='l') # 模
plot(Re(fft(y)),t='l') # 实部
plot(Im(fft(y)),t='l') # 虚部
par(op)
```

45.4 窗函数

在谱分析的时候为了减小截断边界时产生的吉布斯(Gibbs)效应, 往往需要加窗([17] 第一章). fir1() fir2() spectgram() 函数使用 window 参数加窗. 下面是 signal 包提供的窗函数.

```
bartlett(n)
blackman(n)
boxcar(n)
flattopwin(n, sym = c('symmetric', 'periodic'))
```

```
gausswin(n, w = 2.5)
hamming(n)
hanning(n)
triang(n)

# 查看各种窗函数的形状
n = 51
op = par(mfrow=c(3,3))
plot(bartlett(n), type = "l", ylim = c(0,1))
plot(blackman(n), type = "l", ylim = c(0,1))
plot(boxcar(n), type = "l", ylim = c(0,1))
plot(flattopwin(n), type = "l", ylim = c(0,1))
plot(gausswin(n, 5), type = "l", ylim = c(0,1))
plot(hanning(n), type = "l", ylim = c(0,1))
plot(hamming(n), type = "l", ylim = c(0,1))
plot(triang(n), type = "l", ylim = c(0,1))
par(op)

# kaiser 窗
plot(kaiser(101, 2), type = "l", ylim = c(0,1))
lines(kaiser(101, 10), col = "blue")
lines(kaiser(101, 50), col = "green")

# Dolph-Chebyshev window coefficients
plot(chebwin(50, 100))
```

45.5 Periodogram(周期图)

45.5.1 简介

周期图也叫做样本谱(sample spectrum), 实际上就是离散傅立叶变换.

功率谱估计可以分为经典谱估计方法与现代谱估计方法。

经典谱估计中最简单的就是周期图法，又分为直接法与间

接法。都可以编程实现，很简单。

- 直接法先取N点数据的傅里叶变换（即频谱），然后取频谱与其共轭的乘积，就得到功率谱的估计；
- 间接法先计算N点样本数据的自相关函数，然后取自相关函数的傅里叶变换，即得到功率谱的估计.

但是周期图法估计出的功率谱不够精细，分辨率比较低。因此需要对周期图法进行修正，可以将信号序列x(n)分为n个不相重叠的小段，分别用周期图法进行谱估计，然后将这n段数据估计的结果的平均值作为整段数据功率谱估计的结果。还可以将信号序列x(n)重叠分段，分别计算功率谱，再计算平均值作为整段数据的功率谱估计。这2种称为分段平均周期图法，一般后者比前者效果好。加窗平均周期图法是对分段平均周期图法的改进，即在数据分段后，对每段数据加一个非矩形窗进行预处理，然后在按分段平均周期图法估计功率谱。相对于分段平均周期图法，加窗平均周期图法可以减小频率泄漏，增加频峰的宽度。welch法就是利用改进的平均周期图法估计估计随机信号的功率谱，它采用信号分段重叠，加窗，FFT等技术来计算功率谱。与周期图法比较，welch法可以改善估计谱曲线的光滑性，大大提高谱估计的分辨率。

现代谱估计主要针对经典谱估计分辨率低和方差性不好提出的，可以极大的提高估计的分辨率和平滑性。可以分为参数模型谱估计和非参数模型谱估计。参数模型谱估计有AR模型，MA模型，ARMA模型等；非参数模型谱估计有最小方差法和MUSIC法等。由于涉及的问题太多，这里不再详述，可以参考有关资料。[1]

45.5.2 例子

```
> x=c(15 , -2, 12, 20, -5 , 0 , -8 , -4 , -8, -22)
> ft=fft(x)
```

[1]来自网络资料

```
# 直接法. 与共轭的乘积
> a=ft*Conj(ft); a # 虚部全部为零. 相当于 abs(ft)^2
 [1]    4.0000+0i 3071.0684+0i  350.0167+0i  805.9316+0i 2802.9833+0i
 [6]  196.0000+0i 2802.9833+0i  805.9316+0i  350.0167+0i 3071.0684+0i

> Re(a/10)
 [1]   0.40000 307.10684  35.00167  80.59316 280.29833  19.60000 280.29833
 [8]  80.59316  35.00167 307.10684

# 间接法. 自相关的傅立叶变换
> b=fft(acf(x,plot=F)$acf); b
, , 1

                  [,1]
 [1,] 0.5000000+0.0000000i
 [2,] 1.5771143-1.1254158i
 [3,] 0.6227612-0.0241158i
 [4,] 0.7826640-0.3125117i
 [5,] 1.4830890-0.2939962i
 [6,] 0.5687430+0.0000000i
 [7,] 1.4830890+0.2939962i
 [8,] 0.7826640+0.3125117i
 [9,] 0.6227612+0.0241158i
[10,] 1.5771143+1.1254158i

> abs(b)
, , 1

         [,1]
 [1,] 0.5000000
 [2,] 1.9374856
 [3,] 0.6232279
 [4,] 0.8427494
 [5,] 1.5119480
 [6,] 0.5687430
 [7,] 1.5119480
 [8,] 0.8427494
 [9,] 0.6232279
[10,] 1.9374856

> cor(a,b)
```

```
[1] 0.9833278

# 图形基本一样
op=par(mfrow=c(2,1))
plot(a,t='l')
plot(b,t='l')
par(op)
```

R 函数 spectrum() 使用两种方法(当参数 method="pgram" 为 spec.pgram(), method="ar" 为 spec.ar())计算周期图. 一种是 pgram 法, 另外一个是 ar 法(AR模型平滑后的周期图). spec.pgram() 用法如下

```
    spec.pgram(x, spans = NULL, kernel, taper = 0.1,
               pad = 0, fast = TRUE, demean = FALSE, detrend = TRUE,
               plot = TRUE, na.action = na.fail, ...)
```

```
# 手工计算的周期图
f=fft(co2)
p=Re(f*Conj(f))
p=p/length(p) # 此处好像是定义中有的

op<-par(mfrow=c(4,1))
# 如下参数即为无任何处理的周期图. (绘图过程中有处理, 故图看起来有点不同)
spec.pgram(co2,kernel=NULL,taper=0,fast=F,demean=F,det=F)
plot(p[2:(length(p)/2+1)],t='l')
plot(log10(p[2:468/2+1]),t='l') # 取对数后单位变为分贝
plot(log10(p[2:468/2+1]),t='l')
par<-op

# 查看数据
> x=c(15 , -2,  12,  20,  -5 ,  0 , -8 , -4 , -8, -22)
> spectrum(x,taper=0,fast=F,demean=F,det=F,plot=F)$spec[1:10]
 [1] 307.10684  35.00167  80.59316 280.29833  19.60000       NA       NA
 [8]       NA       NA       NA
> Re(fft(x)*Conj(fft(x)))/length(x)
```

```
 [1]   0.40000 307.10684  35.00167  80.59316 280.29833  19.60000 280.29833
 [8]  80.59316  35.00167 307.10684
```

过度平滑不好

```
f=fft(co2)
p=Re(f*Conj(f))
p=p/length(p) # 此处好像是定义中有的

f=filter(co2,f=c(.5,.5))
f=na.exclude(f)
f=fft(f)
p1=Re(f*Conj(f))/length(f)

f=filter(co2,f=rep(1/3,3))
f=na.exclude(f)
f=fft(f)
p2=Re(f*Conj(f))/length(f)

f=filter(co2,f=rep(1/4,4))
f=na.exclude(f)
f=fft(f)
p3=Re(f*Conj(f))/length(f)

op<-par(mfrow=c(4,1))
plot(log10(p),t='l')
plot(log10(p1),t='l')
plot(log10(p2),t='l')
plot(log10(p3),t='l')
par<-op
```

45.6 sound

45.6.1 载入声音文件并查看信息

假设有一个声音文件名为 "frog.wav"

```
> library(sound)
> x <- loadSample("frog.wav")
> typeof(x)
[1] "list"
# .wav 对象属于 Sample 类
> class(x)
[1] "Sample"
> names(x)
[1] "sound" "rate"  "bits"

# 查看信息
> print(x)
type      : mono
rate      : 22050 samples / second
quality   : 16 bits / sample
length    : 73611 samples
R memory  : 294444 bytes
HD memory : 147266 bytes
duration  : 3.338 seconds


# 获取声音数据
> s=sound(x) # 等价于 x$sound
> dim(x$sound)
[1]     1 73611
# 时间长度
> duration(x)
[1] 3.338367
# 采样位数
> bits(x)
[1] 16
# 采样率
> rate(x)
```

```
[1] 22050
```

45.6.2 声 谱,播 放,频 率 图

```
# 绘 出 声 谱
> plot(x)
# 播 放 声 音
> play(x,command='mplayer')

# 绘 制fft图
n <- length(x$sound)
n <- round(n/3)
y <- x$sound[ n:(n+2000) ]
n <- length(y)
op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(y, type='l')
plot(Mod(fft(y)[1: ceiling((length(y)+1)/2) ]), type='l')
```

45.6.3 产 生 调 频 信 号

函 数 用 法

```
chirp( t, f0 = 0, t1 = 1, f1 = 100,
      form = c("linear", "quadratic", "logarithmic"),
      phase = 0
    )
```

- t: 时 间 向 量. (array of times at which to evaluate the chirp signal)
- f0: t=0 的 频 率 (frequency at time t=0.)

- t1: 时间, 单位 秒. (time, s.)
- f1: t=t1 的频率 (frequency at time t=t1.)
- form: 调频(频率变化)的形状. (shape of frequency sweep, one of "linear", "quadratic", or "logarithmic".) 定义为
  - 'linear' is: $f(t) = (f1 - f0) * (t/t1) + f0$
  - 'quadratic' is:$f(t) = (f1 - f0) * t/t1^2 + f0$
  - 'logarithmic' is:$f(t) = (f1 - f0)^{t/t1} + f0$
- phase: t=0 的相位. (phase shift at t=0.)

下面是在线例子

```
ch = chirp(seq(0, .6, len=5000))
plot(ch, type = "l")

# Shows a quadratic chirp of 400 Hz at t=0 and 100 Hz at t=10
# Time goes from -2 to 15 seconds.
# 时 间 为 -2, 15 s, t=0 频 率 为 400, t=10 为 100, 变 化 形 状 是 quadratic
specgram(chirp(seq(-2, 15, by=.001), 400, 10, 100, "quadratic"))

# Shows a logarithmic chirp of 200 Hz at t=0 and 500 Hz at t=2
# Time goes from 0 to 5 seconds at 8000 Hz.
specgram(chirp(seq(0, 5, by=1/8000), 200, 2, 500, "logarithmic"))
```

45.6.4 语图

spectgram() 函数绘制黑白图(灰度图). image() 绘制彩图(library(graphics))

```
library(signal)

# 下 面 使 用 signal 包 内 的 函 数 spectgram() 来 绘 制 语 图(spectrogram)
```

```
wav <- loadSample("frog.wav") # library(sound)
Fs = wav$rate
step = trunc(5*Fs/1000);     # one spectral slice every 5 ms
window = trunc(40*Fs/1000); # 40 ms data window
fftn = 2^nextpow2(window); # next highest power of 2
spg = specgram(wav$sound, fftn, Fs, window, window-step)
S = abs(spg$S[2:(fftn*4000/Fs),]) # magnitude in range 0<f<=4000 Hz.
S = S/max(S)        # normalize magnitude so that max is 0 dB.
S[S < 10^(-40/10)] = 10^(-40/10)   # clip below -40 dB.
S[S > 10^(-3/10)] = 10^(-3/10)    # clip above -3 dB.
image(t(20*log10(S)), axes = FALSE) #, col = gray(0:255 / 255))
```

Chapter 46

小波

46.1 推荐

《小波与傅里叶分析基础》[17], 入门极佳, 工科足够.

R 的包(下面的介绍来自 CRAN Task View: Time Series Analysis):

- wavelets: 包含计算小波滤波, 小波变换, 多尺度分析的内容.
- wmtsa: 基于 Percival and Walden (2000) 的时间序列分析的小波方法
- waveslim: time series (1D), image (2D) and array (3D) analysis. 实现了众多方法(包括 wmtsa).[1]
- brainwaver: 依赖于 waveslim[2]

[1]原包的介绍如下: Basic wavelet routines for time series (1D), image (2D) and array (3D) analysis. The code provided here is based on wavelet methodology developed in Percival and Walden (2000); Gencay, Selcuk and Whitcher (2001); the dual-tree complex wavelet transform (CWT) from Kingsbury (1999, 2001) as implemented by Selesnick; and Hilbert wavelet pairs (Selesnick 2001, 2002). All figures in chapters 4-7 of GSW (2001) are reproducible using this package and R code available at the book website(s) below.

[2]原包的介绍如下: This package computes the correlation matrix for

- wavethresh: 1d, 2d 小波分析[3]
- rwt: 依赖于 matlab

这里使用 waveslim 包

46.2 介绍

小波的出现尽管可以追溯到几十年前, 但是只是在最近的二十多年才成为信号分析流行的工具. 一定程度上, 这应当归功于 Ingrid Daubechies 女士[4]在构造紧支撑正交小波方面的杰出工作. 大部分的小波文章和参考资料需要复杂的数学背景(研究生程度的实分析课程). 傅里叶变换的一个缺点是, 它的构造块是无始无终的周期性正弦和余弦波, 适合压缩(滤除, 分析)那些具有近似周期性的波动信号. 而对于有显著局部特征的信号就无能为力了.

而小波不同于正弦波和余弦波, 它仅仅在有限的一段非零. 小波可以平移和伸缩, 然后将给定的信号展开成小波的伸缩和平移之和, 然后把欲舍弃的系数进行适当处理或直接丢弃. 这就是小波变换.

小波有很多种. 它们(包括傅里叶变换的正弦波和余弦波)应该具有一些基本性质, 其中一个就是正交性, 包括平移和伸缩后. 而正弦波和余弦波具备这样的性质, 导致了求解傅里叶系数的简单公式和高效算法(FFT). ([17] 前言部分)

---

each scale of a wavelet decomposition, namely the one performed by the R package waveslim (Whitcher, 2000). An hypothesis test is applied to each entry of one matrix in order to construct an adjacency matrix of a graph. The graph obtained is finally analysed using the small-world theory (Watts and Strogatz, 1998) and using the computation of efficiency (Latora, 2001), tested using simulated attacks. The brainwaver project is complementary to the camba project for brain-data preprocessing. A collection of scripts (with a makefile) is avalaible to download along with the brainwaver package, see information on the webpage mentioned below.

[3]原包的介绍如下: Software to perform 1-d and 2-d wavelet statistics and transforms

[4]Ingrid Daubechies 女士现在为普林斯顿大学数学系教授

46.3 小波的类型

参考 http://en.wikipedia.org/wiki/Wavelet

46.3.1 Discrete wavelets

- Beylkin (18)
- BNC wavelets
- Coiflet (6, 12, 18, 24, 30)
- Cohen-Daubechies-Feauveau wavelet (Sometimes referred to as CDF N/P or Daubechies biorthogonal wavelets)
- Daubechies wavelet (2, 4, 6, 8, 10, 12, 14, 16, 18, 20)
- Binomial-QMF
- Haar wavelet
- Mathieu wavelet
- Legendre wavelet
- Villasenor wavelet
- Symlet

46.3.2 Continuous wavelets

Real valued

- Beta wavelet
- Hermitian wavelet
- Hermitian hat wavelet
- Mexican hat wavelet

- Shannon wavelet

Complex valued

- Complex mexican hat wavelet
- Morlet wavelet
- Shannon wavelet
- Modified Morlet wavelet

46.3.3 TOBEDEL: wt.filter()支持的小波

wavelets 包返回的值是 S4 对象. wt.filter() 函数产生各种小波.

```
d    Daubechies 2,4,6,8,10,12,14,16,18,20.

la    Least Asymetric 8,10,12,14,16,18,20.

bl    Best Localized 14,18,20.

c    Coiflet 6,12,18,24,30.
```

46.3.4 wave.filter()函数支持的小波

这里我们使用 waveslim 包. waveslim 的文档并没有给出可用的下面小波的全称. 最常用的是 Daubechies wavelet

```
haar
bl14 # Best Localized 小波(or Beylkin 小波??)
bl20
bs3.1
```

```
d16
d4 # Daubechies wavelet
d6
d8
fk14
fk22
fk4
fk6
fk8
la16 # Least Asymetric 小波
la20
la8
mb16
mb24
mb4
mb8
w4
```

46.4 例子

waveslim包dwt() 函数的用法

```
dwt(x, wf="la8", n.levels=4, boundary="periodic")
    dwt.nondyadic(x)

# 返回值
     d?: Wavelet coefficient vectors. 小波系数

     s?: Scaling coefficient vector. 尺度系数

 wavelet: Name of the wavelet filter used.

boundary: How the boundaries were handled.
```

构造一个零之前为高频, 零之后为低频的信号([31] 15.6.1)

```
N <- 1024
k <- 6
x <- ( (1:N) - N/2 ) * 2 * pi * k / N
y <- ifelse( x>0, sin(x), sin(3*x) )
plot(y~x, type='l')

z<-y+rnorm(N)/10

library(waveslim)
# 图的上面是低频尺度系数, 下面四个是高频小波系数. 频率从高到低
d<-dwt(z)
op<-par(mfrow=c(5,1))
plot(d$s4,t='l',ylab='s4')
plot(d$d1,t='l',ylab='d1')
plot(d$d2,t='l',ylab='d2')
plot(d$d3,t='l',ylab='d3')
plot(d$d4,t='l',ylab='d4')
par(op)

# 过滤掉高频成分并重构信号
d$d1<-rep(0,length(d$d1))
d$d2<-rep(0,length(d$d2))
d$d3<-rep(0,length(d$d3))
id<-idwt(d)
#
op<-par(mfrow=c(2,1))
plot(z~x,t='l')
plot(id~x,t='l')
par(op)

# 过滤掉低频成分并重构信号
d$s4<-rep(0,length(d$s4))
id<-idwt(d)

op<-par(mfrow=c(2,1))
plot(z~x,t='l')
plot(id~x,t='l')
par(op)
```

Part VII

流行病学

“流 行 病 学” 参 考 文 献 除 了[11] 第13,14章 的 内 容. 《Analysis of Epidemiological Data using R and Epicalc》 也 是 主 要 的 一 个 。

另 外 一 个 流 行 病 学 的 包 是 epiR, 也 非 常 的 好.

Chapter 47

一些概念

47.1 前瞻性研究

前瞻性研究 (perspective study): 在开始的时间点上没有疾病的一群人, 经过一段时间后, 其中某些人发生了疾病. 发生疾病的人可能与开始时受某个变量(一般称暴露变量)的影响有关. 前瞻性研究的总体常称为队列(cohort), 因此又称为队列研究(cohort study).

47.2 回顾性研究

回顾性研究(retrospective study): 在这个研究中, 共有两组人群: (1)一组在研究中有病(病例) (2)另一组在研究中没有疾病(对照). 研究者要寻找在过去某一时间段内两组人的某种卫生习惯是否有差异. 这种研究也常常称为病例-对照研究(case-control study).

47.3 现状研究

现状研究(cross-sectional study): 在某个时间点上, 询问研究总体的所有成员, 请他们回答现在的疾病状况及他们现在或过去的暴露状况. 有时候也称为患病率研究(prevalence study). 因为它可以在某时刻即时的比较暴露与未暴露个体之间的患病率. 前瞻性研究中感兴趣的是发病率而不是患病率.

47.4 危险率差与比(RR)

令p_1, p_2分别为暴露受试者和非暴露受试者中有病的概率.

危险率差(risk difference)定义为: $p_1 - p_2$. 也称为 attributable risk(AR)

危险比(或相对危险度 risk ratio)(relative risk)定义为 p_1/p_2.

47.5 优势及优势比(OR)

危险率比 RR 受分母的影响太大. 为避免这一限制, 使用另外一个比例的测度称为优势比(odds ratio, OR).

优势: 如果一个事件成功发生的概率为p, 则有利于成功发生的优势为$p/(1-p)$.

优势比: 记p_1, p_2为两组中成功发生的概率. 则优势比定义为

$$OR = \frac{p_1/q_1}{p_2/q2} = \frac{p_1 q_2}{p_2 q_1}$$

47.6 优效性研究与等效性研究

建立在空白对照上的的无效假设为: 两个处理有相同的效应; 对两个处理的效应彼此不同. 临床研究常常是这种形式, 称为优效性研究(superiority study).

有些人认为建立在空白对照上的优效性研究常常是为了考察一种处理的效率. 另外一些人认为如果标准方法已经被证明有效, 则对病人不给治疗的做法是不道德的. 例如对精神分裂症病人使用空白对照去估计某种新方法的疗效).

近年来提出了一种新的研究设计形式, 主要目标是研究两种方法是否等效而不是一种优于另一种. 这种研究称为等效研究(equivalence study).

47.7 筛选检验的一般性概念

设通过筛选检验的结果来假设检验为

$$H_0: \text{此人未患病} \quad vs. \quad H_1: \text{此人患病}$$

筛选检验的结果为两种: 阴性(-), 阳性(+)

47.7.1 预测值阳性/阴性

预测值阳性(predictive value positive. PV^+)是指一个人在该试验中呈阳性条件下患病的概率. 即

$$PV^+ = P(\text{疾病}|\text{检验}^+)$$

实际上是检验的功效, 即此检验判断此人患病而此人确实患病的概率.

预测值阴性(predictive value negative. PV^-)是指一个人在该试验中呈阴性条件下未患病的概率. 即

$$PV^- = P(\text{无疾病}|\text{检验}^-)$$

即此检验判断此人患病而此人确实患病的概率.

例1(预测值阳性/阴性): 设10000名乳房X射线照片检查后为阴性的妇女, 2年内发现有乳腺癌的是20例. 则预测值阴性

$$\begin{aligned} PV^- &= P(\text{无疾病}|\text{检验}^-) \\ &= 1 - P(\text{有疾病}|\text{检验}^-) \\ &= 1 - 0.0002 = 0.9998 \end{aligned}$$

10名X射线检验为阳性的妇女在此2年内发现乳腺癌1例, 则则预测值阳性

$$PV^+ = P(\text{疾病}|\text{检验}^+) = 1/10 = 0.1$$

就是说, 如果X射线阴性, 则几乎可以肯定2年内此妇女不会患乳腺癌($PV^- \approx 1$). 如果X射线阳性, 则2年内此妇女患乳腺癌的概率有10%($PV^+ = 0.1$).

某检验这两个值较高说明此检验有较高的价值. 实际上我们总是寻找$PV^+ = 1, PV^- = 1$的检验, 即只要检验阳性, 就可以判断患病, 只要检验阴性, 就可以判断不患病, 我们就可以准确的对每个病人做出判断.

47.7.2 症状(检验)的灵敏度/特异度

一个症状(或一组症状, 或筛选检验)的灵敏度(sensitivity)是疾病发生后出现症状的概率.

一个症状(或一组症状, 或筛选检验)的特异度(specificity)是疾病不发生时不出现症状的概率.

例2(灵敏度和特异度): 假设肺癌中90%抽烟, 没有肺癌的30%抽烟. 此处疾病为肺癌, 症状为抽烟. 灵敏度为肺癌中抽烟的概率为0.9, 特异度为没有肺癌的不抽烟的概率0.7.

47.7.3 症状有效

预测疾病中某个症状是有效的, 指该症状的两个指标(灵敏度,特异度)都是高的.

47.7.4 假阴性/假阳性

某试验结果是阴性但实际上是阳性(即实际上此人患病)称为假阴性(false negative),

某试验结果是阳性但实际上是阴性(即实际上此人未患病)称为假阳性(false positive),

47.7.5 Bayes法则的应用

我们已经知道某疾病症状的灵敏度和特异度, 还知道此疾病的先验概率(此疾病总的发病率), 那么我们可以由Bayes法则求出某人出现此症状(试验阳性)时的患病概率(预测值阳性/阴性). (Bayes法则的描述见附录61.3)

假设自动血压计把有高血压的84%诊断为高血压, 正常的23%诊断为高血压, 已知成年人中20%为高血压. 那么此血压计的预测值阳性与预测值阴性是多少?

记A=症状, B=疾病. 那么

- 预测值阳性为$PV^+ = P(B|A)$
- 预测值阴性为$PV^- = P(\bar{B}|\bar{A})$
- 灵敏度为$P(A|B) = 0.84$
- 特异度为$P(\bar{A}|\bar{B} = 1 - 0.23 = 0.77$
- 疾病的先验概率$P(B) = 0.2$

那么由贝叶斯法则得

$$PV^+ = P(B|A) = 0.84 * 0.2/(0.84 * 0.2 + 0.23 * 0.8) = 0.48$$
$$PV^- = P(\bar{B}|\bar{A}) = 0.77 * 0.8/(0.77 * 0.8 + 0.16 * 0.2) = 0.95$$

从结果可以看到, 此血压计对于阴性结果的人有很高的预测能力(95%的把握保证此人无高血压), 但是对于阳性的人预测能力不足(48%的把握保证此人有高血压)

47.8 ROC曲线

ROCR包可以计算假阳性, 假阴性, ROC曲线, chi方, 优势比等.

BioConductor项目也有一个ROC包

47.8.1 定义

在某些情况下, 试验结果可能有多个等级而不是简单的阳性/阴性. 另外一些情况下, 试验结果可能是连续的. 在这种情况下, 判断阳性/阴性的切断点(cut-off-point)常常是任意的.

例如: 可能有神经系统疾病的109名受试者(是否有病早已知道)接受某放射学家的CT成像技术检验. 结果用等级(rating)表示. 如果把所有CT结果当做阳性(出现症状), 那么其灵敏度为 判断

Table 47.1: CT成像等级结果

CT结果	实际正常	实际不正常
肯定正常	33	3
可能正常	6	2
有问题	6	2
可能不正常	11	11
肯定不正常	2	33
总数	58	51

为不正常的人数/实际患病人数=51/51=1, 特异度为 判断正常的人数/实际正常人数=0/51=0.

如果把肯定正常当做阴性, 其它结果当做阳性(出现症状)结果, 则灵敏度为 出现症状的人数/实际患病人数=48/51=0.94, 特异度为 未出现症状的人数/实际正常人数=33/58=0.56.

依次类推, 我们可以构造一个ROC(receiver operating characteristic)曲线, x轴为1-特异度, y轴为灵敏度作图, 不同的点对应不同的切断点识别阳性.

曲线下的面积是这个诊断方法的精度的合理指标. 实际上

通过观察可以知道, 无论是灵敏度增大还是特异度增大, 曲线向上凸起的程度都增大. 而灵敏度和特异度都大表明检验方法比较好.

47.8.2 从数据直接计算

函数roc.from.table计算ROC并绘制曲线. 此例子曲线下面积为0.89, 意味着放射学家能够按照CT等级的相对顺序把一个正常人从不正常人中识别出的概率为89%[1].

```
library(epicalc)
t=cbind(c(33,6,6,11,2),c(3,2,2,11,33))
> roc.from.table(t)
$auc 曲线下面积
[1] 0.893171

$original.table
   实际正常   实际患病
 Non-diseased Diseased
         33        3
          6        2
          6        2
         11       11
          2       33

$diagnostic.table
      1-特异度  灵敏度
  1-Specificity Sensitivity
     1.00000000   1.0000000
>    0.43103448   0.9411765
>    0.32758621   0.9019608
>    0.22413793   0.8627451
>    0.03448276   0.6470588
>    0.00000000   0.0000000
# 其它的用法
> roc.from.table(t, title=TRUE, auc.coords=c(.4,.1), cex=1.2)
```

[1]此说法可能证据不足–孙尚拱

47.8.3 logistic回归的ROC曲线

使用函数lroc(), 参考多重logistic回归部分51.9

47.9 生存分析一般概念

47.9.1 (累加)发病率

在类型数据的统计分析中, 很多时候"人"是分析的单位. 前瞻性研究中, 在基线时间把个体分为暴露非暴露组, 比较两组一段时间内的发病的比例, 我们把这些比例称为发病率(incidence rate), 更确切的名称应该称为累加发病率. 累加发病率是一种比例, 以人为分析单位, 值在0,1之间. 计算中隐含所有人被跟踪相同时间. 但是常常不能满足.

47.9.2 发病密度

发病密度(incidence density, ID)定义为该组群中发病的人数除以研究过程中累加的人-时间(年)总数. 分母是人-年数, 值可以是$0, \infty$.

有时候发病密度用更常用的术语发病率(incidence rate)λ表示, 以区别于时间t内的累加发病率CI(t).

下面是一个例子([11] Page 648). 研究口服避孕药(OC)与乳腺癌的关系. 由护士研究课题所收集. 她们在1976年没有乳腺癌, 但OC的使用情况不同. 每两年调查一次, 最后累加使用或不使用OC的时间, 如何判断这些数据在乳腺癌发病率上的差异?

使用OC的情况	病例数	人-年数
现在使用者	9	2935
从不使用者	239	135130

47.9.3 累加发病率与发病密度的关系

为简单起见, 一段时间t内发病密度是不变的, 则由微积分可以证明

$$CI(t) = 1 - e^{-\lambda t}$$

其中, CI(t): 累加发病率. λ:发病密度.

如果累加发病率< 0.1, 则近似有$e^{-\lambda t} = 1 - \lambda t$, 则

$$CI(t) = 1 - (1 - \lambda t) = \lambda t$$

例如, 40-44岁绝经前妇女每100000人-年有200人患乳腺癌, 则40岁没有乳腺癌的妇女今后5年的累加发病率是多少? 此处$\lambda = 200/10^5, t = 5$

```
> lambda=200/10^5; lambda # 发病密度
[1] 0.002
> t=5
> CI5=1-exp(-lambda*t); CI5 # 累加发病率
[1] 0.009950166
> CI=lambda*t; CI # 近似的累加发病率
[1] 0.01
```

47.9.4 率比(RR)

类似于危险率的比(risk ratio, RR), 那里的单位是人, 我们也可以使用于人-时间数据两个发病率的比较. 记λ_1, λ_2分别是暴露和非暴露组的发病率, 称λ_1/λ_2为率比(rate ratio)

47.10 交叉设计

47.10.1 交叉设计(cross over design)

交叉设计(cross over design)是随机临床试验的一种形式. 每个受试者被随机指定为组1或组2, 在第一期处理内组1接受药物A, 组2接受药物B. 第二期相反处理, 即组1接受药物B, 组2接受药物A. 两期之间常常有一段洗脱时间, 以消除药物的残留效应.

若可以控制药物的残留效应, 那么交叉设计是值得考虑的, 例如象高血压这样的研究. 否则需要使用平行组设计. 大多数临床三期试验是长期的研究, 因此使用平行组设计.

47.10.2 洗脱期

洗脱期(washout period)是安排在两个药物处理期之间以消除药物的残留效应的一段时间.

47.10.3 残留效应(剩余效应)

药物的残留效应(carry over effect)指第一个处理期内的一个或多个药物会在第二期内有剩余的生物学效应.

47.11 常用的回归分析

流行病学中常用的回归为线性回归(固定模型和随机模型及其混合模型), logistic回归和 Poisson 回归. 应变量为二分变量的使用logistic回归, 应变量为单位时间(面积)的计数(自然数)的使用Poisson 回归. 详细见 "回归与方差分析".

Chapter 48

函数介绍

48.1 epicalc包

cs: 前瞻性和现状研究. 计算危险率, 危险率差, 危险率比.

cc: 计算回顾性研究的优势比和95%置信区间. 结果基于精确方法.

ci: 可以计算binomial(二项比例), poisson(累加发病率), numeric(均值) 的估计与置信区间.

mhor: 计算回顾研究(case-control study)分层数据的Mantel-Haenszel 优势比. 基于精确方法. 有时候与 mantelhaen.test 结果不太一样. matchTab: 匹配数据的优势比估计.

48.2 rateratio.test包

rateratio.test: 计算率比

48.3 epiR包

此包的函数很强.

epi.2by2: 计算2×2列联表计数数据的各种值。四种方法, method= cohort.count(前瞻性研究数据), cohort.time(人-时间数据), case.control(对照-病例研究数据), cross.sectional(现状研究数据) 分析四种数据. 根据数据类型不同结果不同. 若是多层数据, 返回每一层与联合的的OR, RR等对应结果, 和 Mantel-Haenszel 联合结果, 每一层和联合的卡方值(齐性检验).

epi.kappa: 计算 kappa 统计量, 与 mcnemar 检验(p-值)

epi.dsl: 随机效应的 DerSimonian-Laird meta-analysis

epi.mh: 固定效应的 Mantel-Haenszel meta-analysis

48.4 rmeta

meta.DSL: 随机效应的 DerSimonian-Laird meta-analysis

meta.MH(rmeta): 固定效应的 Mantel-Haenszel meta-analysis

48.5 stats包

prop.test: 二项比例检验

prop.trend.test: 二项比例趋势性检验

fisher.test: 精确二项比例检验(独立性检验), 与 cc 结果一样, 也是精确方法. 但是没有给出卡方值.

Chapter 49

类型(属性)数据的效应测度

我们要在暴露与非暴露的受试者之间比较疾病发生的频率. 这在前瞻性研究中比较的是发病率, 而在现状研究中比较的是患病率.

49.1 危险率差的估计

令p_1, p_2分别为暴露受试者和非暴露受试者中有病的概率.

危险率差(risk difference)定义为: $p_1 - p_2$.

危险比(或相对危险度 risk ratio)(relative risk)定义为 p_1/p_2.

设 $\hat{p_1}, \hat{p_2}$ 分别为暴露和未暴露受试者样本中有病的比例. 样本量分别为 n_1, n_2. 则 $p_1 - p_2$ 的无偏点估计为 $\hat{p_1} - \hat{p_2}$. 假设这个二项分布中正态分布的假定成立, 则可以使用正态分布理论近似求出置信区间的估计. 我们已知

$$\hat{p_1} \sim N(p_1, p_1q_1/n_1)$$
$$\hat{p_2} \sim N(p_2, p_2q_2/n_2)$$

因为是两个独立样本, 我们有

$$\hat{p_1} - \hat{p_2} \sim N(p_1 - p_2, \frac{p_1q_1}{n_1} + \frac{p_2q_2}{n_2}$$

正态分布假设成立时, 使用 $\hat{p_1}, \hat{p_2}$ 代替 p_1, p_2, 我们可以导出危险率差的点及区间估计. 无偏点估计为 $\hat{p_1} - \hat{p_2}$. 区间估计为

$$\hat{p_1} - \hat{p_2} - [1/(2n_1) + 1/(2n_2)] \pm z_{1-\alpha/2}\sqrt{\hat{p_1}\hat{q_1}/n_1 + \hat{p_2}\hat{q_2}/n_2}, \quad if \quad \hat{p_1} >= \hat{p_2}$$

$$\hat{p_1} - \hat{p_2} + [1/(2n_1) + 1/(2n_2)] \pm z_{1-\alpha/2}\sqrt{\hat{p_1}\hat{q_1}/n_1 + \hat{p_2}\hat{q_2}/n_2}, \quad if \quad \hat{p_1} <= \hat{p_2}$$

置信区间适用条件为 $n_1\hat{p_1}\hat{q_1} >= 5, n_2\hat{p_2}\hat{q_2} >= 5$.

(详细的推导见[11] Page 554-555)

下面是一个例子([11] Page 345, 表 10.2 Page 555, 解答). 研究口服避孕药(OC)对心脏病的影响. 5000名在开始服用OC的妇女, 3年后发展有心肌梗塞(MI)的有13人, 10000名未服用OC的妇女3年内有7例MI. 请估计服用及未服用OC的MI发病率的点估计及区间估计.

```
> x=matrix(c(9993,7,4983,13),nc=2,dimnames=list(c("No-MI","MI"),
   c("No-Expose","Expose")))
> x
     No-Expose Expose
No-MI     9993   4983
MI           7     13

> library(epicalc)
# 注: cs, cc等函数只有控制台输出, 无返回值
> cs(outcome=NULL, exposure=NULL, cctable=x,decimal=6)

           Exposure
Outcome       Non-exposed Exposed Total
 Non-diseased 9993        4987    14980
 Diseased     7           13      20
 Total        10000       5000    15000

             Rne         Re      Rt
 Risk         7e-04       0.0026  0.001333

                                    Estimate Lower95ci Upper95ci
Risk difference (attributable risk)    0.0019   0.000661  0.003139
Risk ratio                          3.714286 1.467171  9.403078
Attr. frac. exp. -- (Re-Rne)/Re        0.730769
```

```
 Attr. frac. pop. -- (Rt-Rne)/Rt*100 %  47.5

# 手工计算, 结果与cs不一样. cs算法未知
> n1=5000
> n2=10000
> p1=13/n1
> q1=1-p1
> p2=7/10000
> q2=1-p2
> p1-p2
[1] 0.0019
> p1-p2-(1/(2*n1)+1/(2*n2))-qnorm(0.975)*sqrt(p1*q1/n1+p2*q2/n2)
[1] 0.0002463116
> p1-p2-(1/(2*n1)+1/(2*n2))+qnorm(0.975)*sqrt(p1*q1/n1+p2*q2/n2)
[1] 0.003253688
```

Rne, Re, Rt 分别为: 未暴露患病率, 暴露患病率, 总患病率. Risk difference 为危险率差及上下置信区间. Risk ratio 为危险率比.

49.2 危险率比(RR)的估计

危险率比(risk ratio, RR=p_1/p_2). 其对数$ln(RR)$的样本分布比RR本身更接正态分布. 详细算法参考 [11] Page 556. 参考文献的置信区间的结果为(1.5, 9.3), 与cs的结果稍稍不同. 用法见上面.

49.3 优势比(OR)的估计

危险率比 RR 受分母的影响太大. 为避免这一限制, 使用另外一个比例的测度称为优势比(odds ratio, OR).

优势: 如果一个事件成功发生的概率为p, 则有利于成功发生的优势为$p/(1-p)$.

优势比: 记p_1, p_2为两组中成功发生的概率. 则优势比定义为

$$OR = \frac{p_1/q_1}{p_2/q2} = \frac{p_1 q_2}{p_2 q_1}$$

对优势比点估计及区间估计的具体算法参考 [11] Page 562. epicalc 包中的 cc 函数执行优势比的估计. 其方法应该为 woolf 方法. (下面的例子来自 [11] Page 560 例 13.10. 数据表为 Page 344 表 10.1)对初娩年龄与子宫癌发病的统计. 优势比及其置信区间见下面. 结果与文献结果一致.

```
> x=matrix(c(8747,2537,1498,683),nc=2,
   dimnames=list(c("No-Cancer","Cancer"),c("<=29",">=30")))
> x
          <=29 >=30
No-Cancer 8747 1498
Cancer    2537  683

> fisher.test(x)

       Fisher's Exact Test for Count Data

data:  x
p-value < 2.2e-16
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.419073 1.740189
sample estimates:
odds ratio
  1.571925

> library(epicalc)
> cc(cctable=x,decimal=5)

            Exposure
Outcome       Non-exposed Exposed Total
  Non-diseased       8747    1498 10245
  Diseased           2537     683  3220
  Total             11284    2181 13465
```

```
OR =  1.57192
95% CI = 1.41907 1.74019
Chi-squared = 78.36984 ,  1 d.f. , P value = 0
Fisher's exact test (2-sided) P value = 0
```

49.4 优势比与危险率的比较

当样本中病例(暴露中病例+非暴露病例)比例f_1与非病例(暴露中非病例+非暴露非病例)比例f_2不同时, $\hat{RR}$不是RR的无偏估计, 而$\hat{OR}$是OR的无偏估计. 在病例-对照研究中, f_1, f_2几乎总是不同的, f_1 几乎总是要大于f_2. (证明见 [11] Page 559)

49.5 混杂与分层

混杂变量(confounding variable): 是一个与疾病和暴露变量都有关的变量.

分层(stratification): 在疾病暴露关系分析中, 把数据按照一个或多个潜在的混杂变量的水平分成若干组, 这称为分层. 这些小组称为"层".

正混杂(positive confounder): 如果该变量与疾病和暴露两者关系都是正向的, 或都是负向的. 这样的混杂称为正混杂.

负混杂(negative confounder): 如果该变量与疾病呈正关系而和暴露是负向的关系, 或相反. 这样的混杂称为负混杂.

例如: 我们考察酗酒与肺癌的关系, 可以得到一个2×2列联表. 在混杂变量中, 抽烟是其中一个. 按照抽烟与否, 可以把此列联表分为两个2×2列联表, 其中一个是抽烟的酗酒与肺癌的关系. 另外一个是不抽烟的酗酒与肺癌的关系. 抽烟的列联表中OR=1.0, 非抽烟的列联表中OR=1.0. 故控制抽烟后酗酒与肺癌无关系. 抽烟与肺癌和酗酒都是正向的关系, 是正混杂.

49.6 分层的类型数据统计推断方法-Mantel-Haenszel检验

49.6.1 Mantel-Haenszel检验及优势比估计

Mantel-Haenszel检验也是基于超几何分布, 与 Fisher 检验原理一样. 详细算法请参考 [11] Page 570.

目的是判断二态疾病和二态暴露变量在控制一个或多个混杂变量后的关联性.

哪一行或列被安排在第一是任意的. 即这个检验统计量及判断的显著性不受行列顺序的影响.

零假设: 疾病与暴露之间无联系.

49.6.2 公共优势比与效应修正

一般检验优势比是否齐性是重要的. 如果每一层的关联程度相同, 则可以给出公共优势比. 否则公共优势比没有意义, 应该给出各层单独的优势比.

假设考察疾病变量D和暴露变量E的关联性, 但是有混杂变量C. 于是我们按变量C把总体分成g层且计算每层的优势比. 若各层的真实的优势比不同, 我们认为在E与C之间存在交互作用(interaction)或效应修正(effect modification), 变量C称为效应修正因子(effect modifier). 即若C是效应修正因子, 则C的不同水平会有不同的疾病与暴露的关系.

49.6.3 例子

下面是一个例子 ([11] Page 569). 研究目的是看被动吸烟(passiveSmoke)对癌症(ill)危险率的影响. 此处被动吸烟是暴露变量, 其配偶每天至少1支且吸烟6个月以上. 混杂变量就是被动

吸烟者本人是否吸烟(smoke=yes, no). 因为本人是否吸烟与配偶是否吸烟和癌症都有关系的变量.

各层优势比齐性检验: 首先看 Homogeneity test, 卡方自由度为1, 值 = 3.254582, p-值=0.0712. 接受各层优势比是齐性的, 没有显著不同. 若不同, 则看 Var3 A Var3 B, 分别给出各层的优势比.

关联性检验: M-H Chi2(1)这一行. p-值=0.0001461 很小(也就是卡方值大. M-H Chi2(1)=14.42230, 自由度为1.), 说明控制本人是否吸烟后被动吸烟与癌症还是有高度显著的正相关联系. 公共优势比为 1.63.

```
> x=array(c(120,80,111,155,161,130,117,124),
   dim=c(2,2,2),
   dimnames=list(c("ill","control"),
      "passiveSmoke"=c("yes","no"),
      c("smoke=yes","smoke=no")))
> x
, ,  = smoke=yes

        passiveSmoke
         yes  no
  ill     120 111
  control  80 155

, ,  = smoke=no

        passiveSmoke
         yes  no
  ill     161 117
  control 130 124

> mantelhaen.test(x) # 使用连续修正

      Mantel-Haenszel chi-squared test with continuity correction

data:  x
Mantel-Haenszel X-squared = 13.9423, df = 1, p-value = 0.0001885
alternative hypothesis: true common odds ratio is not equal to 1
95 percent confidence interval:
```

```
 1.263955 2.090024
sample estimates:
common odds ratio
        1.625329

> mantelhaen.test(x,exact=T) # 精确计算

       Exact conditional test of independence in 2 x 2 x k tables

data:  x
S = 281, p-value = 0.0001665
alternative hypothesis: true common odds ratio is not equal to 1
95 percent confidence interval:
 1.254833 2.109589
sample estimates:
common odds ratio
        1.626181

> library(epicalc)
> mhor(mhtable=x,decimal=6)

Stratified analysis by  Var3
             OR lower lim. upper lim.  P value
Var3 A      2.09     1.418      3.10 0.000120
Var3 B      1.31     0.918      1.88 0.138248
M-H combined 1.63     1.264      2.09 0.000146

M-H Chi2(1) = 14.42230 , P value = 0.0001461
Homogeneity test, chi-squared 1 d.f. = 3.254582 , P value = 0.0712241
```

49.7 匹配研究中优势比的估计

匹配数据的McNemar检验与分层数据的Mantel-Haenszel检验密切相关. 匹配是分层的一种特例. 每个配对对应样本量为2的一个层. 可以证明, McNemar检验是层中样本量为2的Mantel-Haenszel检验的一个特例.

在成对的匹配中, 结局相同的对称为一致对(concordant pair). 结局不同的称为不一致对(discordant pair). 不一致对中, 使用A处理后有事件发生而B处理后未发生, 称为A型不一致对. 否则称为B型不一致对.

匹配数据 Mantel-Haenszel 检验中, 疾病与暴露关系的优势比为

$$OR = n_A/n_B$$

n_A为A型不一致对数. n_B为B型不一致对数.

匹配研究中$ln(OR)$近似服从正态分布, 方差为

$$Var[ln(OR)] = 1/(npq)$$

n为不一致对总数. p为A型不一致对的比例. q=1-p.

双侧 $100\% * (1-\alpha)$ 置信区间为 (e^{c_1}, e^{c_2}), 其中

$$c_1 = ln(OR) - z_{1-\alpha}\sqrt{\frac{1}{npq}}$$

$$c_1 = ln(OR) + z_{1-\alpha}\sqrt{\frac{1}{npq}}$$

$$n = n_A + n_B$$

$$p = n_A/n$$

下面是《生物统计学基础》10.4 Page 360 中的一个例子. 按年龄(或其它条件)配对621对病人, 配对的1人随机指定使用A方法治疗, 另外一人使用B方法治疗. 其中A方法生存5年以上, B方法也生存5年以上的有510对 ; A方法生存5年以上, B方法生存少于5年的有5对 ; A方法生存少于5年, B方法生存5年以上的有16对 ; A方法生存少于5年, B方法也少于5年的有90对. 检验A, B两种方法的差异是否显著.

此例中, 有 510+90=600 个一致对. 有 5+16=21 个不一致对. 一致对不提供信息, 故分析时抛弃之. 我们集中研究一致对.

由于 matchTab 函数只是针对原始数据, 没有针对列联表的方法. 当我们有列联表而没有原始数据的时候, 需要把数据仔

细的再还原一下. 设A方法为case, B方法为control. 生存大于5年为暴露, 小于5年为非暴露.

注意变量0,1的取值相反, 结果会不同. 区别在于一个是另一个的倒数.

结果与 [11] Page 577 一致.

```
> x
            B result
A result      more 5 years less 5 years
  more 5 years          90          16
  less 5 years           5         510

# 还原数据并使用 matchTab 函数
> a5=rep(1,106) # A 方法存活大于5年(暴露)
> a4=rep(0,515) # A 方法存活小于5年(非暴露)
> b5=a5
> b4=a4
> b5[91:106]=0 # A>5年的B有16个小于5年
> b4[1:5]=1 # A<5年的B有5个大于5年
> a=c(a5,a4) # A方法(case)的所有结果
> b=c(b5,b4) # B方法(control)的所有结果
> table(a,b)
   b
a    0   1
  0 510   5
  1  16  90

> caseControl=c(rep(0,621),rep(1,621))
> expose=c(a,b)
> pair=c(1:621,1:621) # 配对
> matchTab(caseControl,expose,pair)

Exposure status: expose = 1

Total number of match sets in the tabulation = 621

Number of controls = 1
                No. of controls exposed
```

```
No. of cases exposed   0  1
                0 510 16
                1   5 90

Odds ratio by Mantel-Haenszel method = 0.312

Odds ratio by maximum likelihood estimate (MLE) method = 0.313
 95%CI= 0.114 , 0.853

# 如果expose变量相反, 则结果会不同, 请注意
> ex2=-expose+1 # 0, 1 相反
> matchTab(caseControl,ex2,pair)

Exposure status: ex2 = 1

Total number of match sets in the tabulation = 621

Number of controls = 1
                 No. of controls exposed
No. of cases exposed  0   1
                0 90   5
                1 16 510

Odds ratio by Mantel-Haenszel method = 3.2

Odds ratio by maximum likelihood estimate (MLE) method = 3.2
 95%CI= 1.172 , 8.735

# 手工计算
> i=x[1,2]
> j=x[2,1]
> n=i+j
> p=i/n
> q=1-p
> or=min(i,j)/max(i,j) # odds ratio
> or
[1] 0.3125
> c1=exp(log(or)-qnorm(0.975)*sqrt(1/(n*p*q))) # CI 1
> c1
[1] 0.1144825
> c2=exp(log(or)+qnorm(0.975)*sqrt(1/(n*p*q))) # CI 2
```

```
> c2
[1] 0.8530236

> or1=max(i,j)/min(i,j)
> or1
[1] 3.2
> c1=exp(log(or1)-qnorm(0.975)*sqrt(1/(n*p*q)))
> c1
[1] 1.172301
> c2=exp(log(or1)+qnorm(0.975)*sqrt(1/(n*p*q)))
> c2
[1] 8.734961
```

49.8 存在混杂的趋势性检验

如果有一个二态疾病变量(D), 一个二态暴露变量E, 及一个类型混杂变量C. 则在控制C后, 用Mantel-Haenszel检验去判断D与E的关联性. 若没有混杂, 使用二项比例的两样本检验, 或2×2 列联表法, 如果存在混杂, 使用Mantel-Haenszel检验

如果E是类型变量但多于2个水平, 例如2×k 列联表, 如果没有混杂, 使用趋势性卡方检验. 如果存在混杂, 使用Mantel-Extension 检验.

算法参考的是 [11] Page 578.

假设我们有s层. 每层二态疾病变量和k个有序类型的暴露变量的关系形成 2×k 列联表. 对于第j个类型有分数(打分)x_j, 如下表所示

疾病+	n_{i1}	n_{i2}	...	n_{ik}	n_i
疾病-	m_{i1}	m_{i2}	...	m_{ik}	m_i
	t_{i1}	t_{i2}	...	t_{ik}	N_i
分数(打分)	x_1	x_2	...	x_k	

要检验假设: $H_0 : \beta = 0 \quad H_1 : \beta \neq 0$. 此处$p_{ij}$ =第i层 第j暴露水平上个体中有病的比例=$\alpha_i + \beta x_j$.

计算检验统计量

$$X^2_{TR} = (|O - E| - 0.5)^2/V \sim \chi^2_1(H_0)$$

其中

$$O = \sum_{i=1}^{s} O_i = \sum_{i=1}^{s}\sum_{j=1}^{k} n_{ij}x_j$$

$$E = \sum_{i=1}^{s} E_i = \sum_{i=1}^{s}[(\sum_{j=1}^{k} t_{ij}x_j)\frac{n_i}{N_i}]$$

$$V = \sum_{i=1}^{s} V_i = \sum_{i=1}^{s} \frac{n_i m_i(N_i s_{2i} - s^2_{1i})}{N_i^2(N_i - 1)}$$

$$s_{1i} = \sum_{j=1}^{k} t_{ij}x_j, i = 1, 2, \cdots, s$$

$$s_{2i} = \sum_{j=1}^{k} t_{ij}x_j^2, i = 1, 2, \cdots, s$$

使用条件为 $V >= 5$.

若$X^2_{TR} > \chi^2_{1-\alpha}$我们拒绝$H_0$, 否则接受.

下面是一个例子([11] Page 578). 研究打鼾(ill)与年龄的关系, 混杂变量是性别. R的习惯将有病放在下面第二行, 暴露也放在右边第二列.

```
> x=array(c(603,196,486,223,232,103,348,188,383,313,206,232),
   dim=c(2,3,2),
   dimnames=list("ill"=c("no","yes"),
                 "age"=c("30-39","40-49","50-60"),
                 "sex"=c("M","F")))
> x
```

```
, , sex = M

    age
ill   30-39 40-49 50-60
  no    603   486   232
  yes   196   223   103

, , sex = F

    age
ill   30-39 40-49 50-60
  no    348   383   206
  yes   188   313   232

# R 中没有找到计算 Mantel-Extension 的函数
Mantel.Extension.test<-function(x){
   d=dim(x)
   s=d[3] # s层
   b=d[1] # 疾病二态, b=2
   k=d[2] # 暴露的k个水平
   score=1:k # 打分
   O=sum(x[2,,1:s]*score) # R的习惯将有病放在第二行, 暴露
也放在第二列.
   Ni=array(0,s) # 第s层总和
   ni=array(0,s) # 第s层第二行边际和
   mi=array(0,s) # 第s层第一行边际和
   for(i in 1:s){
      Ni[i]=sum(x[,,i])
      ni[i]=sum(x[2,,i])
      mi[i]=sum(x[1,,i])
}

   s1=colSums(colSums(x[,,1:s])*score)
   s2=colSums(colSums(x[,,1:s])*score^2)
   E=sum(s1*ni/Ni)
   V=sum(ni*mi*(Ni*s2-s1^2)/(Ni^2*(Ni-1)))
   X=(abs(O-E)-0.5)^2/V
   p=1-pchisq(q=X,df=1)
   cat("chi square: ",X,"  df=",1,"  p value=",p,"\n")
   res=list(statistics=X,df=1,p.value=p)
}
```

```
> r=Mantel.Extension.test(x)
chi square:  35.05958   df= 1   p value= 3.197706e-09
> r
$statistics
[1] 35.05958

$df
[1] 1

$p.value
[1] 3.197706e-09
```

Chapter 50

样本量及功效的估计

本节主要参考 [11] Page 580 13.6 和 [25] chapter 24 Sample size calculation.

50.1 计算样本量的函数

epicalc 包有4个计算样本量的函数.

第一个计算现状调查(prevalence survey, 流行度调查)的样本量.

第二个计算两个比例比较(comparison of two proportions)的样本量, 可以是 case-control study, cross-sectional study, cohort study or randomised controlled trial 之一.

第三个计算两个均值比较(comparison of two means)的样本量.

第四个是批质量检验抽样(lot quality assurance sampling.)样本量.

50.2 现场调查(Field survey)

现场调查(Field survey)的目的主要是获得某些人群的某种比例, 例如蠕虫病的发病率, 医疗服务的覆盖率等. 样本量依赖于估计的流行度(prevalence), 即比例, 和可接受的错误水平.

许多情况下采用整群抽样 (cluster sampling), 主要是为了减少采样时间和出行费用. 例如, 一个随机采样需要调查96个村子的96个人. 我们可以把村子减少到例如30个, 通过增加样本量来补偿这种整群抽样带来的影响. 实际上, 整群抽样减少了独立性, 也叫做设计效应(design effect).

函数 n.for.survey 用于计算现场调查的样本量. 首先看看参数

```
> args(n.for.survey)
function (p, delta = "auto", popsize = NULL, deff = 1, alpha = 0.05)
```

p: 估计的发病比例, 0,1 之间.

delta: p与置信区间的差. 例如估计p=0.3, 而最大的比例可能是0.5, 则 $delta = 0.5 - 0.3 = 0.2$. 函数中delta的值根据p值变化. $If 0.3 <= p <= 0.7, delta = 0.1. If 0.1 <= p < .3, or 0.7 < p <= 0.9, then delta = .05. Finally, if p < 0.1, then delta = p/2. If 0.9 < p, then delta = (1-p)/2$. delta 应该设的比较小, 以保证精确性, 但样本量会比较大. 一般从 0.1 每增加 0.1, 样本量会减少一半.

deff: 设计效应(design effect). 默认的是随机抽样deff为1. 对于群(cluster)大小很大, 群内的相似度很高, 那么deff就会很大. 样本量也会升高. 一般deff增加一倍, 样本量增加一倍.

alpha: I 型错误概率.

popsize: 所有总体的总的数量大小. 当比较大时样本量的变化就不大了.

对于整群抽样, 例如要到30个村子抽样, 样本量计算出来是210, 那么每个村子抽样7个就可以了.

```
> n.for.survey( p = .8, delta = .1, popsize = 500000, deff =2)

Sample size for survey.
Assumptions:
  Proportion      = 0.8
  Confidence limit = 95 %
  Delta          = 0.1 from the estimate.
  Population size  = 5e+05
  Design effect    = 2

  Sample size     = 123
# 改变 popsize
> n.for.survey( p = .8, delta = .1, popsize = 500, deff =2)
......
  Sample size     = 109

> n.for.survey( p = .8, delta = .1, popsize = 5000, deff =2)
......
  Sample size     = 121

> n.for.survey( p = .8, delta = .1, popsize = 50000, deff =2)
......
  Sample size     = 123

# 改变deff
> n.for.survey( p = .8, delta = .1, popsize = 50000, deff =4)
......
  Sample size     = 246

> n.for.survey( p = .8, delta = .1, popsize = 50000, deff =8)
......
  Sample size     = 491

# 改变delta
> n.for.survey( p = .8, delta = .2, popsize = 50000, deff =8)
......
  Sample size     = 123

> n.for.survey( p = .8, delta = .3, popsize = 50000, deff =8)
......
  Sample size     = 55
```

```
> n.for.survey( p = .8, delta = .4, popsize = 50000, deff =8)
......
  Sample size     = 31

# 默认popsize为极大值
> n.for.survey( p = .8, delta = .4, deff =8)
......
  Sample size     = 31
```

50.3 两个比例的比较

先看参数

```
> args(n.for.2p)
function (p1, p2, alpha = 0.05, power = 0.8, ratio = 1)
```

在回顾性研究(case-Control study)中, p1为case(diseased group, 患病人群)中暴露于危险因子(药物, 辐射等等)的比例, p2为control(non-diseased group, 对照人群, 非患病人群)中暴露于危险因子的比例.

在前瞻性研究(cohort study)中, p1为暴露人群的发病率, p2为非暴露人群的发病率.

在随机对照试验(randomised controlled trial)中, p1为给予新的治疗方法后有效或治愈的比例, p2为旧的治疗方法有效或治愈的比例.

alpha 为 I 型错误.

power 为功效, 即零假设不正确时拒绝零假设的概率.

ratio 比例p2所在样本量(control)与p1所在样本量(case)的比. 最有效的比例是 1:1.

例 如, 在 疾 病 人 群(case)中 的 危 险 率 为0.5, 而 对 照(control)中 的 危 险 率 为0.2, 那 么 能 够 检 验 出 疾 病-对 照 差 别 的 最 小 的 样 本 量 计 算 为

```
> n.for.2p(p1=0.5, p2=0.2)

Estimation of sample size for testing Ho: p1==p2
Assumptions:

    alpha = 0.05
    power = 0.8
      p1 = 0.5
      p2 = 0.2
    n2/n1 = 1

Estimated required sample size:

       n1 = 45
       n2 = 45
  n1 + n2 = 90
```

若 疾 病 比 较 罕 见, 例 如 一 年 才10个 病 例, 研 究 者 想 早 点 结 束 研 究, 那 么 可 以 提 高case:control的 比 例 到, 例 如1:4. 那 么 样 本 量 为

```
> n.for.2p(p1=0.5, p2=0.2, ratio=4)

Estimation of sample size for testing Ho: p1==p2
Assumptions:

    alpha = 0.05
    power = 0.8
      p1 = 0.5
      p2 = 0.2
    n2/n1 = 4

Estimated required sample size:

       n1 = 27
```

```
      n2 = 108
  n1 + n2 = 135
```

比例再提高可能就不合适了, 例如1:9, n1下降了4个, 而n2几乎增加一倍.

```
> n.for.2p(p1=0.5, p2=0.2, ratio=9)
......
      n1 = 23
      n2 = 207
  n1 + n2 = 230
```

power增加样本量也增加

```
> n.for.2p(p1=0.5, p2=0.2, power=0.9)
......
      n1 = 58
      n2 = 58
  n1 + n2 = 116
```

50.4 病例-对照研究中p1,p2与优势比的关系

优势比的定义为

$$OR = \frac{p_1/q_1}{p_2/q2} = \frac{p_1 q_2}{p_2 q_1}$$

当$p1 = 0.5, p2 = 0.2$时, 优势比为

```
> .5/(1-.5)/(.2/(1-.2))
[1] 4
```

有时候知道p2和优势比, 需要求得p1, 那么根据公式计算就可以

```
> p2=0.2
> or=4 # 优势比
> odds2=p2/(1-p2) # p2的优势
> odds2
[1] 0.25
> odds1=or*odds2 # p1的优势
> odds1
[1] 1
> p1=odds1/(1+odds1) # p1
> p1
[1] 0.5
```

优势比减少, 则样本量会增加. 因为区别减少了, p1与p2的比例更接近1了. 若优势比为1, 样本量趋于无穷.

```
> p2=0.2
> or=2
> odds2=p2/(1-p2)
> odds1=or*odds2
> p1=odds1/(1+odds1)
> p1
[1] 0.3333333
> n.for.2p(p1,p2)
......
       n1 = 187
       n2 = 187
   n1 + n2 = 374
> n.for.2p(p1,p1)
......
Estimated required sample size:

       n1 = NaN
       n2 = NaN
   n1 + n2 = NaN
> n.for.2p(p2,p2)
......
```

```
    n1 = NaN
    n2 = NaN
n1 + n2 = NaN
```

50.5 前瞻性研究和随机对照试验中的样本量估计

使用方法与病例-对照研究一样.

50.6 现状研究中的样本量估计

现状研究(cross-sectional survey)有两个目的: (1) 发现发病率 (2) 检验暴露与结果的关系[1]. 前者是一个描述性的估计, 后者是一个假设检验. 这两者的计算方法是不同的.

对于现状研究中的假设检验, 应该使用 n.for.2p . p1为暴露组的阳性(positive outcomes)率, p2为非暴露组的阳性率. 换句话说, ratio 必须是暴露与非暴露组的比例.

例如, 在一个调查中, 暴露组的发病率(prevalence)可能估计为0.2, 非暴露的患病率可能是0.05. 由于暴露组的发病率(prevalence)可能估计为0.2, ratio(n2/n1)应该为 0.8/0.2=4.[2]

```
> n.for.2p(p1=0.2, p2=0.05, ratio=4)

Estimation of sample size for testing Ho: p1==p2
Assumptions:

    alpha = 0.05
    power = 0.8
```

[1]原文是: to test the association between the exposure and the outcome.

[2]原文描述可能有误, 上一段说 ratio 必须是暴露与非暴露组的比例. 而这里使用的是暴露组的非发病率除以发病率. 我认为上一段描述是正确的. ratio=0.2/0.05=4

```
      p1 = 0.2
      p2 = 0.05
   n2/n1 = 4

Estimated required sample size:

      n1 = 48
      n2 = 192
  n1 + n2 = 240
```

暴露组样本量为48, 非暴露组192.

我们还应该使用其它目的的检验来验证一下, 例如现场调查的方法, 估计的暴露组发病率为0.2.

```
> n.for.survey(p=0.2) # delta = 0.05

Sample size for survey.
Assumptions:
  Proportion      = 0.2
  Confidence limit = 95 %
  Delta          = 0.05 from the estimate.

  Sample size     = 246

> n.for.survey(p=0.2,delta=0.1)

Sample size for survey.
Assumptions:
  Proportion      = 0.2
  Confidence limit = 95 %
  Delta          = 0.1 from the estimate.

  Sample size     = 61
```

50.7 比较两个均值的样本量估计

在流行病学中, 比较两个均值不如比较比例的情况多. 因为治疗的决定和结果常常是二态的. 但是有一些结果是连续变量, 例如智商, 痛苦分数, 生活质量等.

两个均值常常有两个标准差, 因此参数也多一点

```
> args(n.for.2means)
function (mu1, mu2, sd1, sd2, ratio = 1, alpha = 0.05, power = 0.8)
```

对均值和标准差估计后就可以计算大概的样本量了

```
>  n.for.2means(mu1=0.8, mu2=0.6, sd1=0.2, sd2=0.25)

Estimation of sample size for testing Ho: mu1==mu2
Assumptions:

    alpha = 0.05
    power = 0.8
    n2/n1 = 1
      mu1 = 0.8
      mu2 = 0.6
      sd1 = 0.2
      sd2 = 0.25

Estimated required sample size:

       n1 = 21
       n2 = 21
  n1 + n2 = 42
```

50.8 批质量检验的样本量估计

批质量检验抽样(lot quality assurance sampling, LQAS)最初应用于工业领域. 目的是在一批产品中抽样检验, 如果合格率大于某个值(不合格率小于某个值), 这批产品就可以投放市场或交付使用, 否则这批产品被拒绝.

与其它抽样方法的区别是, LQAS不估计精确的次品率. 只是检验不合格率是否被超过. 这样所需的样本量就小于需要估计整体精确次品率(或总体发病率)的样本量. 这样在检验的费用很高时是一个替代的方法.

卫生系统采用LQAS主要是应用于监视问题的比例. 例如, anti-TB 药物的质量监控中, 成分化验和溶液检验非常昂贵. 于是使用LQAS来计算能够保证质量检验合格的最小的样本量.

假设最高可接受的次品率是1%, 如果研究表明比例小于等于此比例, 那么这批药物就被接受. 否则整批药物被拒绝. 而这批药物真实的不合格率并不重要. 如果样本量太小, 例如只有20, 那么即使所有样本合格, 也不能保证1%的不合格率, 而样本量太大, 例如1000, 那么就要浪费1000个药物.

```
> args(n.for.lqas)
function (p0, q = 0, N = 10000, alpha = 0.05, exact = FALSE)
> n.for.lqas(p=0.01)

    Lot quality assurance sampling

                        Method = Normal approximation
                  Population size = 10000
  Maximum defective sample accepted = 0
    Probability of defect accepted = 0.01
                         Alpha = 0.05
             Sample size required = 262
> n.for.lqas(p=0.01,N=1000)

    Lot quality assurance sampling

                        Method = Normal approximation
```

```
                Population size = 1000
 Maximum defective sample accepted = 0
   Probability of defect accepted = 0.01
                        Alpha = 0.05
          Sample size required = 212
```

总体为10000, 样本量为262, 那么检验后我们剩下10000-262=9738件药物可以使用, 而又保证了1%以下的次品率. 总体1000时, 样本量下降到212, 检验后还剩1000-212=788可用.

50.9 两个比例比较的功效

```
> table1 <- matrix(c(35,70,20,30),nr=2)
> table1
     [,1] [,2]
[1,]   35   20
[2,]   70   30
> library(epicalc)
> cc(cctable=table1)

            Exposure
Outcome       Non-exposed Exposed Total
  Non-diseased          35      20    55
  Diseased              70      30   100
  Total                105      50   155

OR =  0.75
95% CI = 0.35 1.61
Chi-squared = 0.66 ,  1 d.f. , P value = 0.417
Fisher's exact test (2-sided) P value = 0.474
```

优势比为0.75, 但是有比较宽的置信区间. 我们想知道当真正的优势比为0.5时其功效为多少?

```
> odds.a=20/30
```

```
> odds.treat=0.5*odds.a
> p.a=20/(20+30)
> p.treat=odds.treat/(1+odds.treat)
> power.for.2p(p1=p.treat,p2=p.a,n1=105,n2=50)

    alpha = 0.05
      p1 = 0.25
      p2 = 0.4
      n1 = 105
      n2 = 50


     power = 0.4082
```

此样本量有40%的概率发现真实的差异是0.5. 故此研究是不太可信的.

50.10 两个均值比较的功效

注意有图绘出.

```
> args(power.for.2means)
function (mu1, mu2, n1, n2, sd1, sd2, alpha = 0.05)
>  power.for.2means(mu1=95, mu2=100, sd1=11.7, sd2=10.1, n1=100, n2=100)

    alpha = 0.05
      mu1 = 95
      mu2 = 100
       n1 = 100
       n2 = 100
      sd1 = 11.7
      sd2 = 10.1

     power = 0.8988
```

50.11 分层类型数据样本量及功效的估计

TODO: 参考[11] 13.6 Page 580

Chapter 51

多重logistic回归

参考广义线性回归部分对此有一些描述27.5

Mantel-Haenszel 检验和 Mantel-Extension 检验是对单个类型协变量C控制后, 检验二态疾病D和一个类型暴露变量E的关联性. 但是, 当下面条件之一成立

- E 是连续变量
- C 是连续变量
- 有多个混杂变量, 每个可能是类型或连续的

我们很难或不可能使用前面的方法去控制混杂.

多重logistic回归技术即可以处理前面的Mantel-Haenszel 检验和 Mantel-Extension 检验的情况, 也可以处理这里提出的三种情况.

多重logistic回归类似于多重线性回归, 但结果变量(或应变量)是二态而不是正态分布的.

51.1 一般模型

考虑一个模型

$$p = a + b_1x_1 + b_2x_2 + \cdots + b_kx_k$$

这里p=疾病发生的概率. 方程的右边可能会出现小于0或大于1的情况, 这是不应该的. 我们使用p的logit变换(即 logistic 变换)作为应变量

$$logit(p) = ln[p/(1-p)]$$

logit(p)可以取任何值. 函数编制很容易

```
logit<-function(p){
   res=log(p/(1-p))
   res
}
> logit(0.1)
[1] -2.197225
> logit(0.95)
[1] 2.944439
```

若把logit(p)作为独立变量$x_1, \cdots, x_k$的函数, 解得p则可得到下面的多重logistic回归模型

$$p = \frac{e^{a+b_1x_1+b_2x_2+\cdots+b_kx_k}}{1+e^{a+b_1x_1+b_2x_2+\cdots+b_kx_k}}$$

下面是一个虚构的例子, 两个物种$species = 0, 1$, 两种处理方法$run = 0, 1$, 一周后看看是否发病ill.

```
> x=0:1
> species=sample(x,200,replace=TRUE)
> n<-100
> run<-sample(x,200,replace=TRUE)
> ill<-c(rep(0,n), rep(1,n))
```

```
# logistic回归
> r<-glm(ill~species+run, family=binomial)
> r

Call:  glm(formula = ill ~ species + run, family = binomial)

Coefficients:
(Intercept)     species         run
    0.1107      0.1513     -0.3736

Degrees of Freedom: 199 Total (i.e. Null);  197 Residual
Null Deviance:     277.3
Residual Deviance: 275.4       AIC: 281.4

# summary 结果更丰富
> summary(r)

Call:
glm(formula = ill ~ species + run, family = binomial)

Deviance Residuals:
      Min         1Q     Median         3Q        Max
-1.2904870  -1.1304017   0.0001820   1.1307753   1.2908787

Coefficients:
          Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.1107     0.2553   0.434    0.665
species       0.1513     0.2876   0.526    0.599
run          -0.3736     0.2855  -1.308    0.191

(Dispersion parameter for binomial family taken to be 1)

   Null deviance: 277.26  on 199  degrees of freedom
Residual deviance: 275.36  on 197  degrees of freedom
AIC: 281.36

Number of Fisher Scoring iterations: 3
```

51.2 回归参数的解释

回归系数Coefficients类似于多重线性回归的偏回归系数. 假设两个个体A和B, 除了第j个暴露变量外其它都相同. 此处A为暴露($x_j = 1$)B为非暴露($x_j = 0$). 个体A,B成功概率的logit变换为$logit(p_A), logit(p_B)$, 则

$$logit(p_A) = a + b_1x_1 + \cdots + b_j(1) + \cdots + b_kx_k$$
$$logit(p_B) = a + b_1x_1 + \cdots + b_j(0) + \cdots + b_kx_k$$

两式相减有

$$logit(p_A) - logit(p_B) = b_j$$

由定义知

$$logit(p_A) = ln[p_A/(1 - p_B)]$$
$$logit(p_B) = ln[p_B/(1 - p_B)]$$

带入得

$$ln[p_A/(1 - p_B)] - ln[p_B/(1 - p_B)] = b_j$$

即

$$ln[\frac{p_A/(1 - p_B)}{p_B/(1 - p_B)}] = b_j$$

取反对数既得

$$\frac{p_A/(1 - p_B)}{p_B/(1 - p_B)}] = e^{b_j}$$

由优势比的定义我们可以重写为

$$\frac{Odd_A}{Odd_B} = b_j$$

我们知道Odd_A/Odd_B就是第j个暴露变量与疾病的优势比. 而这两个个体其它变量都是相同的. 即此优势比是调整模型中其它危险因素后得到的.

我们总结如下.

51.2.1 二态独立变量在多重logistic回归模型中优势比的估计

对于多重logistic回归模型, 假设一个二态变量1表示有暴露, 0表示无暴露. 这个暴露变量对于应变量的优势比(OR)被估计为

$$\hat{OR} = e^{b_j}$$

这个优势比是调整模型中其它变量后的结果. 它的双侧置信区间为

$$[e^{b_j - z_{1-\alpha/2} se(b_j)}, e^{b_j + z_{1-\alpha/2} se(b_j)}]$$

例子中控制了物种 species 后两种处理方法run患病的优势比及区间为

```
# 处理方法的系数
> b2=r$coeff[3]
> b2
      run
-0.3735543

# R函数计算系数置信区间, 使用的是t分布近似
> confint(r)
Waiting for profiling to be done...
              2.5 %    97.5 %
(Intercept) -0.3899164 0.6146411
species     -0.4119937 0.7173819
run         -0.9367140 0.1843467

# 下面是手工计算

# 获取se
> summary(r)$coeff
            Estimate Std. Error    z value  Pr(>|z|)
(Intercept)  0.1107039  0.2552502  0.4337073 0.6645010
species      0.1512521  0.2875845  0.5259395 0.5989302
run         -0.3735543  0.2855024 -1.3084104 0.1907341
```

```
> summary(r)$coeff[3,2]
[1] 0.2855024
> se=summary(r)$coeff[3,2]
> se
[1] 0.2855024

# 系数置信区间

# 正态分布近似
> b2-qnorm(0.975)*se
      run
-0.9331286
> b2+qnorm(0.975)*se
     run
0.1860201

# t分布近似
> b2-qt(0.975,199)*se
[1] -0.9365526
> b2+qt(0.975,199)*se
[1] 0.1894440

# 优势比
> OR=exp(b2) # 优势比
> OR
     run
0.6882836

# 优势比的置信区间. Rosner 给出的是使用正态分布的近似
> exp(b2-qnorm(0.975)*se)
     run
0.3933212
> exp(b2+qnorm(0.975)*se)
    run
1.204446
```

51.2.2 logistic回归分析和列联表分析的关系

设我们有一个二态疾病变量D和一个二态暴露变量E, 数据是由前瞻性研究, 回顾性研究或现状研究的任何一种产生, 列联表如下我们可以用下面两个等价的方法任何一种估

	E(+)	E(-)
D(+)	a	b
D(-)	c	d

计D与E之间的优势比

- 直接从列联表中求出优势比=ad/bc
- 我们建立一个logistic回归模型

$$ln[p/(1-p)] = \alpha + \beta E$$

p=在暴露变量E下有病D的概率. 此处产生的优势比为e^{β}

对于前瞻性研究或现状研究, 我们可以用下面两个等价的方法任何一种估计个体在暴露下的疾病概率(p_E)及未暴露下的疾病概率($p_{\bar{E}}$

- 从列联表中有

$$p_E = a/(a+c), p_{\bar{E}} = b/(b+d)$$

- 由logistic回归模型

$$p_E = e^{\alpha+\beta}/(1+e^{\alpha+\beta}), p_{\bar{E}} = e^{\alpha}/(1+e^{\alpha})$$

p=在暴露变量E下有病D的概率. 此处产生的优势比为e^{b}

对于回顾性研究(病例-对照研究)我们不可能估计疾病发生的概率, 除非病例及对照样本数在总体中的比例已知, 这几乎不可能.

下面是一个例子. 数据来自[11] 例 10.7 数据为表 10.1 Page 344. 关于乳腺癌与初次生育年龄的关系. 数据见下. 列联表分析的Fisher检验给出优势比为 1.57. 使用logistic回归(需要由列联表重构原始数据)的系数为 0.4523, 优势比为$e^{0.4523} = 1.57$, 与列联表法估计相同.

```
> x <- matrix(c(683,1498,2537, 8747), nr = 2,
       dimnames=list(c("D+","D-"),c(">=30","<=29")))
> x
   >=30 <=29
D+  683 2537
D- 1498 8747
> fisher.test(x)

        Fisher's Exact Test for Count Data

data:  x
p-value < 2.2e-16
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.419073 1.740189
sample estimates:
odds ratio
  1.571925

# 重构原始数据
> d=c(rep(1,683+2537),rep(0,1498+8747))
> e=c(rep(1,683),rep(0,2537),rep(1,1498),rep(0,8747))
> table(d,e)
   e
d     0    1
  0 8747 1498
  1 2537  683

# logistic 回归 e~d
> glm(e~d,fam=binomial)

Call:  glm(formula = e ~ d, family = binomial)

Coefficients:
```

```
(Intercept)          d
   -1.7646      0.4523

Degrees of Freedom: 13464 Total (i.e. Null);  13463 Residual
Null Deviance:     11930
Residual Deviance: 11850       AIC: 11860

# logistic 回归 d~e
> glm(d~e,fam=binomial)

Call:  glm(formula = d ~ e, family = binomial)

Coefficients:
(Intercept)          e
   -1.2377      0.4523

Degrees of Freedom: 13464 Total (i.e. Null);  13463 Residual
Null Deviance:     14810
Residual Deviance: 14740       AIC: 14740

# 计算优势比
> exp(0.4523)
[1] 1.571923
```

51.3 协方差,标准差,t值,置信区间等

变量多于1个且不独立时, 有对称的协方差矩阵, 可以是尺度的或非尺度的('scaled' or 'unscaled'). 尺度因子实际上是glm的dispersion.

接乳腺癌与初次生育年龄的关系的例子

```
> r=glm(d~e,fam=binomial)
> names(summary(r))
 [1] "call"          "terms"         "family"        "deviance"
 [5] "aic"           "contrasts"     "df.residual"   "null.deviance"
 [9] "df.null"       "iter"          "deviance.resid" "coefficients"
```

```
[13] "aliased"      "dispersion"    "df"           "cov.unscaled"
[17] "cov.scaled"

# 协方差矩阵
> summary(r)$cov.scaled
            (Intercept)           d
(Intercept)  0.0007818816 -0.0007818816
d           -0.0007818816  0.0026401751

> summary(r)$cov.unscaled
            (Intercept)           d
(Intercept)  0.0007818816 -0.0007818816
d           -0.0007818816  0.0026401751

# 直接计算协方差矩阵
> vcov(r)
            (Intercept)           d
(Intercept)  0.0007818816 -0.0007818816
d           -0.0007818816  0.0026401751

# 尺度因子
> summary(r)$dispersion
[1] 1
```

d的标准差为

```
# 直接计算
> vcov(r)[2,2]^.5->se2
> se2
[1] 0.05138263
> vcov(r)[2,2]^.5
[1] 0.05138263

# summary里的标准差
> summary(r)$coefficients
            Estimate Std. Error    z value    Pr(>|z|)
(Intercept) -1.7645799 0.02796215 -63.106027 0.000000e+00
d            0.4523372 0.05138263   8.803309 1.328402e-18
```

t-值为系数除以标准差(z-值)

```
> t<-summary(r)$coefficients[2,1]/summary(r)$cov.scaled[2,2]^.5
> t
[1] 8.803309

# 方差开平方, 与coefficients里的标准差一致
> summary(r)$cov.scaled[2,2]^.5
[1] 0.05138263
```

p-值

```
> pt(q=t, df=13465, lower.tail=FALSE) * 2
[1] 1.488717e-18
```

置信区间

```
> b=summary(r)$coefficients[2,1]
> confint(r)
Waiting for profiling to be done...
               2.5 %    97.5 %
(Intercept) -1.8197435 -1.7101259
d          0.3512376  0.5526807
> b1=b+qt(c(0.025, 0.975), 13)*se2
> b1
[1] 0.3413318 0.5633426
> b1=b+qnorm(c(0.025, 0.975))*se2
> b1
[1] 0.3516291 0.5530453
```

51.4 logistic.display函数

详见[25] 15章

```
> library(epicalc)
> logistic.display(r)

Logistic regression predicting e

         OR(95%CI)       P(Wald's test) P(LR-test)
d: 1 vs 0  1.57 (1.42,1.74)  < 0.001       < 0.001

Log-likelihood = -5926.7668
No. of observations = 13465
AIC value = 11857.5336
```

51.5 连续独立变量在多重logistic回归模型中优势比的估计

假设有一个连续变量(x_j), 两个个体在x_j上取值分别为$x_j + \delta, x_j$. 则第一个个体相对于第二个个体的优势比估计为

$$OR = e^{\beta_j + \delta}$$

它的双侧置信区间为

$$[e^{[b_j - z_{1-\alpha/2} se(b_j)]\delta}, e^{[b_j + z_{1-\alpha/2} se(b_j)]\delta}]$$

δ常常取一个自己确定的有意义的值.

例子参考 [11] Page 589, 例13.34.

51.6 假设检验

假设检验

$$H_0 : \ b_j = 0, \ all \ other \ b_l \neq 0$$
$$H_1 : \ b_j \neq 0$$

计算检验统计量, 其中se可以使用summary函数得到. 零假设下有

$$z = b_j/se(b_j) \sim N(0,1)$$

双侧检验, 若$z < z_{\alpha/2}$或$z > z_{1-\alpha/2}$, 拒绝零假设. 否则接受.

最后, 仅在x_j与应变量没有显著关系时, OR的置信区间包含1.

乳腺癌与初次生育年龄的关系的例子中, summary函数可以得到z值与p-值.

```
> summary(glm(e~d,fam=binomial))

Call:
glm(formula = e ~ d, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.6905  -0.5623  -0.5623  -0.5623   1.9609

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.76458    0.02796 -63.106   <2e-16 ***
d            0.45234    0.05138   8.803   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

   Null deviance: 11928  on 13464  degrees of freedom
Residual deviance: 11854  on 13463  degrees of freedom
AIC: 11858

Number of Fisher Scoring iterations: 4

# 手工计算z值
> z=0.45234/0.05138
> z
[1] 8.803815
```

```
> (1-pnorm(z))*2 # p-值
[1] 0
```

51.7 多重logistic回归中的预测

我们可以使用多重logistic回归模型去预测有协变量$x_1, \cdots, x_k$的个体的患病概率. 若回归参数已知, 则

$$p = \frac{e^L}{1+e^L}$$

$$L = a + b_1 x_1 + b_2 x_2 + \cdots + b_k x_k$$

L有时称为线性预测量. 线性预测量的置信区间

$$(L_1, L_2) = L \pm z_{1-\alpha/2} se(L)$$

变换回概率尺度得到

$$p_1 = \frac{e^{L_1}}{1+e^{L_1}}, p_2 = \frac{e^{L_2}}{1+e^{L_2}}$$

se(L)的计算比较复杂, 需要矩阵知识, 但是可以由电脑计算出.

实际上R有一个针对glm的函数 predict 可以计算预测值.

51.8 logistic模型回归拟合优良性的估计

TODO: 参考[11] Page 597 13.6.7节

我们可以用预测概率去定义残差及判断logistic回归模型拟合的优良性.

logistic回归中的残差: 若我们的数据以非群组的形式出现, 即每个个体都有一组协变量, 我们可以定义第i个个体的

Pearson 残差

$$r_i = \frac{y_i - \hat{p}_i}{se(\hat{p}_i)}$$

此处 $y_i = 1$若第i个体是成功, 否则$y_i = 0$.

$$\hat{p}_i = \frac{e^{L_i}}{1 + e^{L_i}}$$

$$L_i = \hat{a} + \hat{b_1}x_1 + \cdots + \hat{b_k}x_k = \text{第i个体的预测值}$$

$$se(\hat{p}_i) = \sqrt{\hat{p}_i(1 - \hat{p}_i)}$$

若我们的数据是群组的形式, 即一些个体有相同的协变量形成一个组, 则第i组的 Pearson 残差为

$$r_i = \frac{y_i - \hat{p}_i}{se(\hat{p}_i)}$$

此处

$$y_i = \text{第i组成功的比例}$$

$$\hat{p}_i = \frac{e^{L_i}}{1 + e^{L_i}}(\text{与非群组相同})$$

$$L_i = \hat{a} + \hat{b_1}x_1 + \cdots + \hat{b_k}x_k = \text{第i个体的预测值}$$

$$se(\hat{p}_i) = \sqrt{\hat{p}_i(1 - \hat{p}_i)/n_i}$$

$$n_i = \text{第i组中个体数目}$$

这里的 pearson 残差类似于线性回归中的 Studentized 残差, 残差是各不相同的. 这里的标准误基于二项分布. 在分组数据中, 若$\hat{p}_i$接近0或1, 或n_i增大时, 标准误下降.

pearson 残差比较大时我们可能需要修正模型. 还可以使用 pearson 残差识别异常值. 但是个体残差的使用要比线性回归模型有更多的限制, 特别在非群组的时候.

可以使用其它方法判断拟合优良性. 例如, 可以考察每个值如何影响回归系数. 假设第j个回归系数为b_j, 删除第i个观察值

后回归系数为b_j^i, 那么第i观察值对b_j的影响的测度可以用下式衡量

$$\delta b_j^i = \frac{b_j - b_j^i}{se(b_j)}$$

pearson 残差需要手工计算.

我们可以使用step, add等逐步回归函数来检验. 下面是一个step的例子

```
> x1=rbinom(100,size=1,prob=0.5)
> x2=rbinom(100,size=1,prob=0.5)
> x3=rbinom(100,size=1,prob=0.5)
> y=rbinom(100,size=1,prob=0.5)
> r=glm(y~x1+x2+x3,family=binomial)
> s=step(r)
Start:  AIC=146.35
y ~ x1 + x2 + x3

      Df Deviance    AIC
- x3   1   138.40 144.40
- x2   1   138.42 144.42
- x1   1   138.54 144.54
<none>     138.35 146.35

Step:  AIC=144.4
y ~ x1 + x2

      Df Deviance    AIC
- x2   1   138.46 142.46
- x1   1   138.59 142.59
<none>     138.40 144.40

Step:  AIC=142.46
y ~ x1

      Df Deviance    AIC
- x1   1   138.63 140.63
```

```
<none>     138.46 142.46

Step:  AIC=140.63
y ~ 1

> summary(s)

Call:
glm(formula = y ~ 1, family = binomial)

Deviance Residuals:
   Min     1Q  Median     3Q    Max
-1.177  -1.177   0.000   1.177   1.177

Coefficients:
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) -9.498e-18  2.000e-01 -4.75e-17       1

(Dispersion parameter for binomial family taken to be 1)

   Null deviance: 138.63  on 99  degrees of freedom
Residual deviance: 138.63  on 99  degrees of freedom
AIC: 140.63

Number of Fisher Scoring iterations: 2

> anova(s)
Analysis of Deviance Table

Model: binomial, link: logit

Response: y

Terms added sequentially (first to last)


    Df Deviance Resid. Df Resid. Dev
NULL                 99     138.63
```

51.9 logistic回归的ROC曲线

定义参考概念部分ROC47.8

函数lroc计算及绘制logistic回归的ROC曲线. 下面是帮助的例子.

```
library(epicalc)
> model1 <- glm(case ~ induced + spontaneous, data=infert, family=binomial)
> logistic.display(model1)

Logistic regression predicting case

                     crude OR(95%CI)  adj. OR(95%CI)    P(Wald's test)
induced (cont. var.)    1.05 (0.74,1.5)  1.52 (1.02,2.27)  0.042

spontaneous (cont. var.) 2.9 (1.97,4.26)  3.31 (2.19,5.01)  < 0.001

                     P(LR-test)
induced (cont. var.)    0.042

spontaneous (cont. var.) < 0.001

Log-likelihood = -139.806
No. of observations = 248
AIC value = 285.612

>     # Having two spontaneous abortions is quite close to being infertile!
>     # This is actually not a causal relationship
>
> lroc(model1, title=TRUE, auc.coords=c(.5,.1))
$model.description
[1] "logit (case ~ induced + spontaneous)"

$auc
[1] 0.7285506

$predicted.table
 predicted.prob Non-diseased Diseased
```

```
        0.1534          60       7
        0.2158          33      12
        0.2949          20       9
        0.3750          25      22
        0.4768          11       5
        0.5806           4       4
        0.6651          11      18
        0.7511           1       6

$diagnostic.table
        1-Specificity Sensitivity
          1.000000000  1.00000000
> 0.1534   0.636363636  0.91566265
> 0.2158   0.436363636  0.77108434
> 0.2949   0.315151515  0.66265060
> 0.3750   0.163636364  0.39759036
> 0.4768   0.096969697  0.33734940
> 0.5806   0.072727273  0.28915663
> 0.6651   0.006060606  0.07228916
> 0.7511   0.000000000  0.00000000
```

Chapter 52

meta再分析

本节主要参考

《A Handbook of Statistical Analyses Using R》 Brian S. Everitt and Torsten Hothorn. CHAPTER 12, Meta-Analysis: Nicotine Gum and Smoking Cessation and the Efficacy of BCG Vaccine in the Treatment of Tuberculosis.

《生物统计学基础》13.8 再分析。

软件包为 rmeta。

52.1 概念

前面的分析, 我们都是对某个研究中的数据做分析. 但是在某种研究中, 我们希望把不同组群的研究分析结果综合成一个结果. 某些研究中, 不同的研究结果似乎矛盾, 另外一些研究中, 它们之间似乎没有什么显著差异.

现在的问题是, 用什么方法把这些研究联合起来以便减少抽样误差并增加研究的功效? 如何解决不同研究中的不相容性? 完成这样研究的技术称为再分析(meta analysis).

52.2 DerSimonian-Laird 方 法(随 机 效 应 模 型)

假 设 有k个 研 究, 每 个 的 目 标 都 是 估 计 优 势 比exp(u)–在 每 个 处 理 组 中 的 疾 病 优 势 相 对 于 对 照 组 中 的 疾 病 优 势.

(1) 把k个 研 究 联 合, 平 均 对 数 优 势 比$u = ln(OR)$的 最 好 估 计 为

$$\hat{u} = \sum_{i=1}^{k} w_i' y_i / \sum_{i=1}^{k} w_i'$$

此 处

$$y_i = \text{第i个 研 究 中 的 对 数 优 势 比}$$
$$w_i' = (s_i^2 + \delta^2)^{-1}$$
$$1/w_i = s_i^2 = 1/a_i + 1/b_i + 1/c_i + 1/d_i = \text{第i个 研 究 内 的 方 差}$$
$$a_i, b_i, c_i, d_i\text{是 第i个 研 究 中2*2列 联 表 的 计 数}$$

$$\delta^2 = max\{0, [Q_w - (k-1)]/[\sum_{i=1}^{k} w_i - (\sum_{i=1}^{k} w_i^2 / \sum_{i=1}^{k} w_i)]\}$$

$$Q_w = \sum_{i=1}^{k} w_i (y_i - \bar{y_w})^2 = \sum_{i=1}^{k} w_i y_i^2 - (\sum_{i=1}^{k} w_i y_i)^2 / \sum_{i=1}^{k} w_i$$

$$\bar{y_w} = \sum_{i=1}^{k} w_i y_i / \sum_{i=1}^{k} w_i$$

对 应 的 优 势 比 的 点 估 计$= exp(\hat{u})$

(2) $\hat{u}$的 标 准 误 为

$$se(\hat{u}) = (1/\sum_{i=1}^{k} w_i')^{1/2}$$

(3) u的100%$(1-\alpha)$置 信 区 间 为

$$\hat{u} \pm z_{1-\alpha/2} se(\hat{u}) = (u_1, u_2)$$

OR 的100%$(1-\alpha)$置信区间为$(exp(u_1), exp(u_2))$

(4) 检验假设 $H_0 : u = 0 \quad vs \quad H_1 : u \neq 0$即$H_0 : OR = 1 \quad vs \quad H_1 : OR \neq 1$ 检验统计量为

$$z = \hat{u}/se(\hat{u}) \sim N(0,1)$$

双侧p-值为为$2[1 - \Phi(|z|)]$.

meta.DSL 用于随机效应和异质性的 Woolf's test. 先看看参数

```
> library(rmeta)
> args(meta.DSL)
function (ntrt, nctrl, ptrt, pctrl, conf.level = 0.95, names = NULL,
    data = NULL, subset = NULL, na.action = na.fail, statistic = "OR")
```

ntrt: 暴露组(treated/exposed group)的个体数目.

nctrl: 对照组的个体数目.

ptrt: 暴露组的成功/发病个体数目(Number of events in treated/exposed group)

pctrl: 对照组的成功/发病个体数目(Number of events in control group)

statistic: OR是优势比, RR为相对危险率

下面是一个例子, 取自[23]

```
>  data("smoking", package = "HSAUR")
# 每一行为一个单独的研究.
# qt,tt分别为暴露组的发病个体数目和个体总数. qc, tc分别为对照组的发病数目和个体总数.
> smoking
            qt  tt  qc  tc
Blondal89    37  92  24  90
Campbell91   21 107  21 105
Fagerstrom82 30  50  23  50
```

```
Fee82        23 180  15 172
Garcia89     21  68   5  38
Garvey00     75 405  17 203
Gross95      37 131   6  46
Hall85       18  41  10  36
Hall87       30  71  14  68
Hall96       24  98  28 103
Hjalmarson84  31 106  16 100
Huber88      31  54  11  60
Jarvis82     22  58   9  58
Jensen91     90 211  28  82
Killen84     16  44   6  20
Killen90    129 600 112 617
Malcolm80     6  73   3 121
McGovern92   51 146  40 127
Nakamura90   13  30   5  30
Niaura94      5  84   4  89
Pirie92      75 206  50 211
Puska79      29 116  21 113
Schneider85   9  30   6  30
Tonnesen88   23  60  12  53
Villa99      11  21  10  26
Zelman92     23  58  18  58

>  smokingDSL <- meta.DSL(smoking[["tt"]], smoking[["tc"]],
+                    smoking[["qt"]], smoking[["qc"]],
+                    names = rownames(smoking))
> smokingDSL
Random effects ( DerSimonian-Laird ) meta-analysis
Call: meta.DSL(ntrt = smoking[["tt"]], nctrl = smoking[["tc"]], ptrt = smoking[["q
   pctrl = smoking[["qc"]], names = rownames(smoking))
Summary OR= 1.75    95% CI ( 1.48, 2.07 )
Estimated random effects variance: 0.05

# summary函数除了上面的结果, 还有详细的每个的OR的估计.
> summary(smokingDSL)
Random effects ( DerSimonian-Laird ) meta-analysis
Call: meta.DSL(ntrt = smoking[["tt"]], nctrl = smoking[["tc"]], ptrt = smoking[["q
   pctrl = smoking[["qc"]], names = rownames(smoking))
-------------------------------------
            OR (lower  95% upper)
```

```
Blondal89    1.85    0.99      3.46
Campbell91   0.98    0.50      1.92
Fagerstrom82 1.76    0.80      3.89
Fee82        1.53    0.77      3.05
Garcia89     2.95    1.01      8.62
Garvey00     2.49    1.43      4.34
Gross95      2.62    1.03      6.71
Hall85       2.03    0.78      5.29
Hall87       2.82    1.33      5.99
Hall96       0.87    0.46      1.64
Hjalmarson84 2.17    1.10      4.28
Huber88      6.00    2.57     14.01
Jarvis82     3.33    1.37      8.08
Jensen91     1.43    0.84      2.44
Killen84     1.33    0.43      4.15
Killen90     1.23    0.93      1.64
Malcolm80    3.52    0.85     14.54
McGovern92   1.17    0.70      1.94
Nakamura90   3.82    1.15     12.71
Niaura94     1.34    0.35      5.19
Pirie92      1.84    1.20      2.82
Puska79      1.46    0.78      2.75
Schneider85  1.71    0.52      5.62
Tonnesen88   2.12    0.93      4.86
Villa99      1.76    0.55      5.64
Zelman92     1.46    0.68      3.14
-----------------------------------
SummaryOR= 1.75  95% CI ( 1.48,2.07 )
Test for heterogeneity: X^2( 25 ) = 34.87 ( p-value 0.0905 )
Estimated random effects variance: 0.05
```

52.3 Mantel-Haenszel 方法(固定效应模型)

固定效应模型使用 Mantel-Haenszel 方法, 在个体数目比较少(小于5)时比较精确. 其它的方法有 Peto's method, 计算上简单, 是 Mantel-Haenszel 方法的近似(rmeta没有提供).

固定效应模型函数 meta.MH 的参数与随机效应模型的 meta.DSL 的参数一样.

```
> args(meta.MH)
function (ntrt, nctrl, ptrt, pctrl, conf.level = 0.95, names = NULL,
   data = NULL, subset = NULL, na.action = na.fail, statistic = "OR")
```

下面与随机效应模型是同一个例子.

```
# 固定效应模型
>  smokingOR <- meta.MH(smoking[["tt"]], smoking[["tc"]],
+                    smoking[["qt"]], smoking[["qc"]],
+                    names = rownames(smoking))
> summary(smokingOR)
Fixed effects ( Mantel-Haenszel ) meta-analysis
Call: meta.MH(ntrt = smoking[["tt"]], nctrl = smoking[["tc"]], ptrt = smoking[["qt
   pctrl = smoking[["qc"]], names = rownames(smoking))
-------------------------------------
             OR (lower  95% upper)
Blondal89    1.85    0.99      3.46
Campbell91   0.98    0.50      1.92
Fagerstrom82 1.76    0.80      3.89
Fee82        1.53    0.77      3.05
Garcia89     2.95    1.01      8.62
Garvey00     2.49    1.43      4.34
Gross95      2.62    1.03      6.71
Hall85       2.03    0.78      5.29
Hall87       2.82    1.33      5.99
Hall96       0.87    0.46      1.64
Hjalmarson84 2.17    1.10      4.28
Huber88      6.00    2.57     14.01
Jarvis82     3.33    1.37      8.08
Jensen91     1.43    0.84      2.44
Killen84     1.33    0.43      4.15
Killen90     1.23    0.93      1.64
Malcolm80    3.52    0.85     14.54
McGovern92   1.17    0.70      1.94
Nakamura90   3.82    1.15     12.71
```

```
Niaura94     1.34    0.35      5.19
Pirie92      1.84    1.20      2.82
Puska79      1.46    0.78      2.75
Schneider85  1.71    0.52      5.62
Tonnesen88   2.12    0.93      4.86
Villa99      1.76    0.55      5.64
Zelman92     1.46    0.68      3.14
-------------------------------------
Mantel-Haenszel OR =1.67 95% CI ( 1.47,1.9 )
Test for heterogeneity: X^2( 25 ) = 34.9 ( p-value 0.09 )
```

52.4 优势比的齐性检验

检验假设对数优势比$u = ln(OR)$

$H_0 : u_1 = \cdots = u_k \quad vs \quad H_1$: 至少两个对数优势比不同

使用下面的检验统计量

$$Q_w = \sum_{i=1}^{k} w_i (y_i - \bar{y_w})^2 \sim \chi^2_{k-1}$$

固定效应模型中不同研究之间的方差在近似研究权重时被忽略了, 而仅考察内部方差. 故方程中仅用w_i代替w_i'. 有一个争论是如果优势比有实质性的差异, 则应该研究差异的来源且不应该报告联合的优势比.

一般讲, 固定模型比随机模型会有更小的置信区间, 更易得出显著性的结论. 但是固定模型和随机模型会有不同的权, 故两个模型可以有不同的优势比结果. 更详细的讨论参考 [11] Page 607.

一般, 使用下面的规则决定用什么模型. 若异质性检验的p-值

- >= 0.5 使用固定效应模型.

- $0.05 <= p < 0.5$使用随机效应模型.
- < 0.05 不要报告合并的优势比, 寻找异质性的来源.

52.5 解释

随机效应的结果为:

```
SummaryOR= 1.75  95% CI ( 1.48,2.07 )
Test for heterogeneity: X^2( 25 ) = 34.87 ( p-value 0.0905 )
Estimated random effects variance: 0.05
```

固定效应的结果为:

```
Mantel-Haenszel OR =1.67 95% CI ( 1.47,1.9 )
Test for heterogeneity: X^2( 25 ) = 34.9 ( p-value 0.09 )
```

我们看到随机效应模型比固定效应模型的CI要宽泛. 异质性检验(Test for heterogeneity)的p-值为0.09, 所有我们最后决定使用随机效应模型.

52.6 绘图

可以对所有OR及置信区间绘图.

```
> plot(smokingOR)
```

Chapter 53

等效性研究(equivalence study)

参考 [11] 13.9 等效性研究. Page 608.

近年来提出了一种新的研究设计形式, 主要目标是研究两种方法是否等效而不是一种优于另一种. 这种研究称为等效研究(equivalence study). 具体参考定义部分.

53.1 统计推断

等效性研究实际上是考察危险率差的单侧检验, 即两个二项比例之差的较低的单侧检验.

设p_1, p_2分别是标准方法和试验方法中的生存率. 我们将寻找一个$p_1 - p_2$的较低的单侧$100\%(1-\alpha)$置信区间. 单侧置信区间为

$$p_1 - p_2 < \hat{p_1} - \hat{p_2} + z_{1-\alpha}\sqrt{\hat{p_1}\hat{q_1}/n_1 + \hat{p_2}\hat{q_2}/n_2}$$

若右边不超过事先指定的差值δ, 称这两个处理是等效的.

p-值也是可以计算出来的, 只要将$p_1 - p_2$标准化后(近似服从标准正态分布)计算超过此值的概率.

53.2 样本量的估计

如果试验组样本量(n_2)是标准组(n_1)的k倍(k是事先指定的), 我们有

$$n_1 = \frac{(\hat{p_1}\hat{q_1} + \hat{p_2}\hat{q_2}/k)(z_{1-\alpha} + z_{1-\beta})^2}{[\delta - (p_1 - p_2)]^2}$$

$$n_2 = kn_1$$

无效假设下, 可以认为$\hat{p_1} = \hat{p_2} = p$, 带入上式既得

$$n_1 = \frac{(pq)(1 + 1/k)(z_{1-\alpha} + z_{1-\beta})^2}{\delta^2}$$

$$n_2 = kn_1$$

TODO: 编写函数, 例子

Chapter 54

交叉设计

交叉设计(cross over design)类似于 Wilcoxon 符号-秩检验. 但是考虑到了前后匹配的效应.

54.1 综合的处理效应的估计

记 x_ijk=交叉设计中病人在第k周期, 第i组 第j个病人的得分值, $k = 1, 2; i = 1, 2; j = 1, \cdots, n_i$.

(1) 计算处理有效性的总估计

$$\bar{d} = (\bar{d_1} + \bar{d_2})/2$$

$$\bar{d_1} = \sum_{j=1}^{n_1} d_{1j}/n_1$$

$$\bar{d_2} = \sum_{j=1}^{n_2} d_{2j}/n_2$$

$$d_{1j} = x_{1j1} - x_{1j2}$$
$$d_{2j} = x_{2j2} - x_{2j1}$$

(2) $\bar{d}$的标准误估计为

$$se = \sqrt{\frac{s^2_{d,pooled}}{4}(1/n_1 + 1/n_2)} = \frac{s_{d,pooled}}{2}\sqrt{(1/n_1 + 1/n_2)}$$

$$s^2_{d,pooled} = \frac{(n_1 - 1)s^2_{d_1} + (n_2 - 1)s^2_{d_2}}{n - 1 + n_2 - 2}$$

$$s^2_{d_1} = \sum_{j=1}^{n_1}(d_{1j} - \bar{d_1})/(n_1 - 1)$$

$$s^2_{d_2} = \sum_{j=1}^{n_2}(d_{2j} - \bar{d_2})/(n_2 - 1)$$

(3) 记Δ=真实的平均处理有效性. 检验假设$H_0 : \Delta = 0 \quad vs \quad H_1 : \Delta \neq 0$. 检验统计量

$$t = \frac{\bar{d}}{se} \sim t_{n_1+n_2-2}$$

(4) 判断

$$|t| > t_{n_1+n_2-2,1-\alpha/2}$$

拒绝零假设, 否则接受.

置信区间为

$$\bar{d} \pm t_{n_1+n_2-2,1-\alpha/2}se$$

下面是一个虚拟的例子. 分组为1,2组. 药物为A,B. 第一组先用A, 后用B. 第二组先用B, 后用A. 打分为疼痛减轻的程度. 0为疼痛无减轻, 6为疼痛完全消失. p-值比较大, 说明差异不显著.

```
# 两种药物比较疼痛减轻程度, d1,d2是两组疼痛减轻打分差值.
```

```
> x_1A=round(runif(10,0,6)) # 第一组用A疼痛减轻的程度
> x_1B=round(runif(10,0,6)) # 第一组用B疼痛减轻的程度
> x_2A=round(runif(10,0,6)) # 第二组用A疼痛减轻的程度
> x_2B=round(runif(10,0,6)) # 第二组用B疼痛减轻的程度
> d1=x_1A-x_1B
> d2=x_2B-x_2A
> d1
 [1] -3 -3 -1 -1  3  2 -3 -1  0 -5
> d2
 [1]  1  3  2 -3  0  1  2 -5  0  2
> d=(mean(d1)+mean(d2))/2
> d
[1] -0.45
> se=(9*var(d1)+9*var(d2))/18
> se
[1] 6.094444
> t=d/se
> t
[1] -0.07383774
> qt(0.025,df=18)
[1] -2.100922
> p=pt(t,df=18)*2
> p
[1] 0.9419539
```

54.2 剩余效应的估计

如果有剩余效应, 先给药后给安慰剂的组的平均效应大于先给安慰剂后给药的组. 定义

$\bar{x_{ij}} = (x_{ij1} + x_{ij2})/2 =$ 两个处理合并后第j受试者在第i组的平均得分

$$\bar{x_i} = \sum_{j=1}^{n_i} \bar{x_{ij}}/n_i == \text{两个处理合并后第i组的平均得分}$$

我 们 假 定$\bar{x_{ij}} \sim N(u_i, \sigma^2), i = 1, 2, j = 1, \cdots, n_i$, 检 验 假 设

$$H_0 : u_1 = u_2 \quad vs \quad H_1 : u_1 \neq u_2$$

计 算 检 验 统 计 量

$$t = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{s^2(1/n_1 + 1/n_2)}}$$

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

$$s_i^2 = \sum_{j=1}^{n_i} (x_{ij} - \bar{x_i})^2/(n_i - 1), i = 1, 2$$

如 果

$$|t| > t_{n_1+n_2-2,1-\alpha/2}$$

拒 绝 零 假 设, 否 则 接 受.

TODO: 例 子

54.3 样 本 量 的 估 计

没 有 剩 余 效 应 时, 每 组 近 似 样 本 量 为

$$n = \frac{\sigma_d^2(z_{1-\alpha/2} + z_{1-\beta})^2}{2\Delta^2}$$

σ_d^2为 获 益 得 分 的 方 差, Δ为 两 个 处 理 内 在 的 获 益 的 差, 即$\bar{d}$.

TODO: 例 子

Chapter 55

聚集性的二态数据

请参考 [11] Page 618. 13.11 聚集性的二态数据.

二项比例的两样本检验是最常用的统计方法, 它要求样本中的观察值是统计独立的. 下面的一个例子却不能认为是独立的. Rowe 等报告了一个经典的临床试验, 使用 3% 的阿糖苷(vidarbine)对比安慰剂处理多发性嘴唇疱疹. 在有效的药物期内, 对31个病人的53个损伤性特征使用阿糖苷, 对39个病人的69个损伤性特征使用安慰剂. 都治疗7天, 我们要比较损伤性的比例是否相同. 1个病人身上的多个特征是有关联的.

我们把这种数据称为聚集性数据, 也称为相关性二态数据(correlated binary data).

这时, 随机化单元可以不同于分析使用的单元, 例如, 临床中的随机化是人为单位的, 但是分析单元是以疱疹或牙齿等特征. 例如, 5个学校随机化的取做有效的食物干预组(目的是减少脂肪摄入量), 另外5个学校取做对照. 假设计算结果是1年后干预组脂肪摄入量比对照少30%, 此结果当然是学校的学生计算出的. 同一学校的学生可能有类似的饮食. 那么学生的反应应该是相关性的二态数据.

可以把聚集性二态数据用于基于 Mantel-Haenszel 检验的控制混杂变量上, 也可以扩展到连续变量并做回归分析. 回归分析中, 同一个单元内的观察值之间的相关性收到重视, 这种回归有时称为相关反应模型, 也称为谱系模型, 混合效应模型或

多水平模型.

55.0.1 聚集性数据二项比例的两样本检验

假设我们有两个受试者组, 样本量分别为n_1, n_2, m_{ij}是第i组 第j个个体的提供的观察数, 其中成功了a_{ij}个. 要检验$H_0 : p_1 = p_2 \quad vs \quad H_1 : p_1 \neq p_2$. 计算检验统计量

$$z = [|p_1 - p_2| - (\frac{c_1}{2M_1} + \frac{c_2}{2M_2})]/\sqrt{pq(c_1/M_1 + c_2/M_2)}$$

其 中

$$p_{ij} = a_{ij}/m_{ij}$$

$$p_i = \sum_{j=1}^{n_i} a_{ij} / \sum_{j=1}^{n_i} m_{ij} = \sum_{j=1}^{n_i} m_{ij}p_{ij} / \sum_{j=1}^{n_i} m_{ij} = \text{i组 总 成 功 比 例}$$

$$M_i = \sum_{j=1}^{n_i} m_{ij}$$

$$p = \sum_{i=1}^{2}\sum_{j=1}^{n_i} a_{ij} / \sum_{i=1}^{2}\sum_{j=1}^{n_i} m_{ij} = \sum_{i=1}^{2} M_i p_i / \sum_{i=1}^{2} M_i$$

$$q = 1 - p$$

$$c_i = \sum_{j=1}^{n_i} m_{ij}c_{ij}/M_i = \text{第i组 聚 集 性 修 正 因 子}$$

$$c_{ij} = 1 + (m_{ij} - 1)\rho$$

$$\rho = (MSB - MSW)/[MSB + (m_A - 1)MSW] = \text{类 内 的 相 关 系 数}$$

$$MSB = \sum_{i=1}^{2}\sum_{j=1}^{n_i} m_{ij}(p_{ij} - p_i)^2/(N-2) = \text{个 体 之 间 均 方 误 差}$$

$$MSW = \sum_{i=1}^{2}\sum_{j=1}^{n_i} a_{ij}(1 - p_{ij})/(M - N) = \text{内 部 均 方 误 差}$$

$$m_A = [M - \sum_{i=1}^{2}(\sum_{j=1}^{n_i} m_{ij}^2/M_i)]/(N-2)$$

$$N = n_1 + n_2$$

$$M = \sum_{i=1}^{2} M_i$$

聚 集 性 修 正 因 子 有 时 候 也 称 为 设 计 效 应(design effect).

显 著 性 检 验: 若$|z| > z_{1-\alpha/2}$拒 绝 零 假 设, 否 则 接 受.

对$p_1 - p_2$的近似100%$(1-\alpha)$置信区间为

$$if\ p_1 > p_2, p_1 - p_2 - [c_1/(2M_1) + c_2/(2M_2)] \pm z_{1-\alpha/2}\sqrt{p_1q_1c_1/M_1 + p_2q_2c_2/M_2)}$$

$$if\ p_1 <= p_2, p_1 - p_2 + [c_1/(2M_1) + c_2/(2M_2)] \pm z_{1-\alpha/2}\sqrt{p_1q_1c_1/M_1 + p_2q_2c_2/M_2)}$$

适用条件为$M_1pq >= 5, M_2pq >= 5$.

下面是[11] Page 621, 例 13.52. expose为每个病人暴露牙龈的的数目, damage为龋齿损伤的数目. 11个男性的27个暴露牙龈6个损伤(22.6%), 29个妇女的99个暴露牙龈6个损伤(6.1%). 判断男性和女性的牙面是否有相同的龋齿发病率.

聚集性二项比例的检验p-值=0.186, 差异不显著[1]. 而卡方检验, 精确Fisher检验, 正态方法得到的结果都是p-值=0.03, 差异显著

```
# cat 是为了调试.
aggregation.test<-function(expose,damage,group,alpha=0.05){
   group=factor(group)
   l=levels(group)
   nl=nlevels(group)
   m1j=expose[group==l[1]]
   m2j=expose[group==l[2]]
   a1j=damage[group==l[1]]
   a2j=damage[group==l[2]]
   M1=sum(m1j)
   M2=sum(m2j)
   p1j=a1j/m1j
   p2j=a2j/m2j
   p1=sum(a1j)/M1
   p2=sum(a2j)/M2
   p=sum(damage)/sum(expose)

#cat(p1,p2,p)

   M1=sum(m1j)
```

[1]文献的结果与本结果有差异, 调试发现MSW结果不同, 可能原始数据输入有误

```
    M2=sum(m2j)
    M=M1+M2
    N=length(expose)

#cat("==",M1,M2,M,N,"==\n")

    MSB=(sum(m1j*(p1j-p1)^2)+sum(m2j*(p2j-p2)^2))/(N-2)

    MSW=sum(damage*(1-damage/expose)) / (M-N)
# the same
#MSW=(sum(a1j*(1-p1j))+sum(a2j*(1-p2j))) / (M-N)
    mA=(M-(sum(m1j^2)/M1+sum(m2j^2)/M2))/(N-2)
    rho=(MSB-MSW)/(MSB+(mA-1)*MSW)

#cat("==",MSB,MSW,mA,rho,"==\n")

    C1j=1+(m1j-1)*rho
    C2j=1+(m2j-1)*rho
    C1=sum(m1j*C1j)/M1
    C2=sum(m2j*C2j)/M2
    se=sqrt(p*(1-p)*(C1/M1+C2/M2))
    tmp1=C1/(2*M1)+C2/(2*M2)
    z=(abs(p1-p2)-tmp1)/se

#cat("==",C1,C2,M1,M2,se,tmp1,"==\n")

    z_=qnorm(1-alpha/2)
    se1=sqrt(p1*(1-p1)*C1/M1+p2*(1-p2)*C2/M2)
    CI1=0.0
    CI2=0.0
    delta=p1-p2
    if(delta>0){
       CI1=delta-tmp1-z_*se1
       CI2=delta-tmp1+z_*se1
}
    if(delta<=0){
       CI1=delta+tmp1-z_*se1
       CI2=delta+tmp1+z_*se1
}
    if(M1*p*(1-p)<5 || M2*p*(1-p)<5){
       cat("\ndamage may not enough\n")
```

```
}
   res=list(delta=delta, z=z,p.value=(1-pnorm(abs(z)))*2,conf.int.delta=c(CI1,CI
   res
}

> expose=c(4,1,2,2,4,3,3,3,1,2,2,
   2,6,8,5,4,4,2,4,4,4,6,2,4,4,2,3,2,2,4,2,2,2,2,4,4,4,3,2,2)
> damage=c(0,1,2,0,2,0,0,0,0,1,0,
   1,1,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0)
> sex=c(rep("M",11),rep("F",29))

# 聚集性二项比例的检验
> aggregation.test(expose,damage,sex)
damage may not enough
$delta
[1] -0.1616162

$z
[1] 1.322749

$p.value
[1] 0.1859188

$conf.int.delta
[1] -0.3387652  0.1056469

$alpha
[1] 0.05

# 卡方检验
> chisq.test(x)

       Pearson's Chi-squared test with Yates' continuity correction

data:  x
X-squared = 4.6918, df = 1, p-value = 0.03031

Warning message:
In chisq.test(x) : Chi-squared近似算法有可能不准

# 二项比例齐性检验(与卡方检验一样)
```

```
> prop.test(x)

        2-sample test for equality of proportions with continuity correction

data:  x
X-squared = 4.6918, df = 1, p-value = 0.03031
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.65355134  0.02197239
sample estimates:
   prop 1    prop 2
0.1842105 0.5000000

Warning message:
In prop.test(x) : Chi-squared近似算法有可能不准

# 参考文献的结果. 正态分布计算出的p-值(即prop.test)
> 2*(1-pnorm(2.166))
[1] 0.03031119

# 精确Fisher检验
> fisher.test(x)

        Fisher's Exact Test for Count Data

data:  x
p-value = 0.02077
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.05506787 0.94962880
sample estimates:
odds ratio
 0.2293778
```

55.0.2 样本量及功效估计

假设我们要检验$H_0: p_1 = p_2 \quad vs \quad H_1: p_1 \neq p_2$. 如果用双侧检验, 显著性水平为$\alpha$, 功效为$1-\beta$, 则合适的样本量(指每组观察总

数)为

$$M_s = M[1 + (\bar{m} - 1)\rho] = \text{每 组 观 察 总 数}$$

此 处

$$M = (z_{1-\alpha/2}\sqrt{2\bar{p}\bar{q}} + z_{1-\beta}\sqrt{p_1q_1 + p_2q_2})^2/(p_1 - p_2)^2$$
$$\bar{p} = (p_1 + p_2)/2$$
$$\bar{q} = 1 - \bar{p}$$

每 组 个 体 数 为

$$n = Ms/\bar{m}$$

这 里$\bar{m}$为 每 个 受 试 者 平 均 观 察 数. ρ为 类 内 相 关 系 数.

如 果 每 组 观 察 总 数 固 定, 对 特 定 备 择 假 设 的 功 效 为

$$power = \Phi(z_{1-\beta})$$

此 处

$$z_{1-\beta} = \frac{\sqrt{M_s/[1 + (\bar{m} - 1)\rho]}|p_1 - p_2| - z_{1-\alpha/2}\sqrt{2\bar{p}\bar{q}}}{\sqrt{p_1q_1 + p_2q_2}}$$

TODO: 例 子

Chapter 56

TODO:测量误差方法

请参考 [11] Page 627. 13.12 测量误差方法, 讲误差对结果的影响.

Chapter 57

人-时间数据及生存分析

57.1 单样本发病率数据的统计推断

57.1.1 大样本方法

假设随访研究过程的t人-年中有a个事件, 且ID=未知的发病密度(率). 检验 $H_0 : ID = ID_0 \quad vs \quad H_1 : ID \neq ID_0$. 计算检验统计量

$$X2 = \frac{(a - u_0)^2}{u_0} \sim \chi_1^2$$

$$u_0 = t * ID_0$$

57.1.2 精确方法

如果事件a太少, 应该使用建立在Poisson分布基础上的精确检验方法. 发病密度(率)即Poisson分布的参数λ此处为$u_0 = t * (ID)$.

注意H_0下, 事件数a服从Poisson分布, 且有参数 $u_0 = t * ID_0$, 则

精确p-值为

$$p = min(2 * \sum_{k=0}^{a} \frac{e^{-u_0} u_0^k}{k!}, 1), if \quad a < u_0$$

$$p = min[2 * (1 - \sum_{k=0}^{a-1} \frac{e^{-u_0} u_0^k}{k!}), 1], if \quad a >= u_0$$

例子([11] Page 650-651 例 14.4 14.6). 1990-1994年建立了一套记录系统, 对还没有乳腺癌但是怀疑有遗传性乳腺癌的妇女作了标记. 500名60-64岁的妇女被识别并随访至2000年末. 整个随访长度为4000人-年. 此期间28例乳腺癌发生. 已知全国60-64岁乳腺癌平均发病率为$400/(10^5)$人-年. (1) 判断这些人乳腺癌发病率与全国是否有差异?

此处$a = 28, u_0 = 4000 * (400/10^5) = 16$, 则检验统计量为

```
> a=28
> u0=4000*(400/10^5)
> X2=(a-u0)^2/u0; X2
[1] 9
> p=1-pchisq(9,df=1); p
[1] 0.002699796
```

(2) 假设500名有遗传学标记的妇女中125人有乳腺癌家族史. 此125人的1000人年共发生8例乳腺癌. 判断这个人群的乳腺癌发病率是否与全国水平一样?

此处$a = 8, u_0 = 1000 * (400/10^5) = 4$, 使用精确方法. p-值为

```
> p=2*(1-ppois(7,4)); p
[1] 0.1022672
```

故没有显著差异. 要想检出差异, 必须加大样本量.

57.1.3 发病率的置信区间

Poisson 分布下, 我们有$\hat{u} = a, var(\hat{u}) = a$. 在t个人年中正态分布近似得发病密度ID的点估计为$\hat{ID} = a/t$, u的双侧估计为$a \pm z_{1-\alpha/2}\sqrt{a} = (c1, c2)$, 若$a < 10$, 使用精确的置信区间. ID的双侧置信区间为$(c1/t, c2/t)$.

ci函数(epicalc包)可以计算 binomial(二项比例), poisson(累加发病率), numeric(均值) 的估计与置信区间(confidence interval).

例如, 上面例子中500人的发病率(发病密度(率)即 Poisson 分布的参数λ此处为$u_0 = t * (ID)$).的点估计为$ID = 28/4000 = 0.007 = 700/10^5$人年. 置信区间为$(28 \pm 1.96\sqrt{28})$. 精确置信区间可以使用ci函数.

```
> c1=28-pnorm(0.975)*sqrt(28); c1
[1] 23.58043
> c2=28+pnorm(0.975)*sqrt(28); c2
[1] 32.41957
> ID1=c1/4000; ID1
[1] 0.005895108
> ID2=c2/4000; ID2
[1] 0.008104892

> ci.poisson(28,4000, alpha=.05) # 500人中的发病率估计
 events person.time incidence         se exact.lower95ci exact.upper95ci
     28       4000     0.007 0.001322876       0.004648       0.010122
#
> ci.poisson(4,1000, alpha=.05)
 events person.time incidence    se exact.lower95ci exact.upper95ci
      4       1000     0.004 0.002       0.001088       0.010244

# ID 的置信区间
> ID.conf.int=ci.poisson(4,1000, alpha=.05)[5:6]*1000; ID.conf.int
 exact.lower95ci exact.upper95ci
         1.088         10.244
```

57.2 两样本发病率数据的统计推断

暴露组	事件数	人-时间数
1	a1	t1
2	a2	t2
总数	a1+a2	t1+t2

我们要比较ID1=组1的真实发病密度(组1单位人-时间的事件发生数) 与ID2=组2的真实发病密度(组2单位人-时间的事件发生数) 是否一样.

零假设下, 两个组可以合并. 一个事件属于组1的个数被看作二项随机变量. 参数$n = a1 + a2, p_0 = t1/(t1 + t2)$. 零假设可以描述为$H_0 : p = p_0(ID1 = ID2)$. 近似正态分布的平均数为$n * p_0 = (a1 + a2)t1/(t1 + t2) = E$, 方差为$np_0q_0 = (a1 + a2)t1t2/(t1 + t2)^2 = V$. 正态分布近似检验统计量为

$$z = \frac{a1 - E - 0.5}{\sqrt{V}}, if \quad a1 > E$$

$$z = \frac{a1 - E + 0.5}{\sqrt{V}}, if \quad a1 <= E$$

$z \sim N(0, 1)$

如果事件数比较小(5), 那么我们使用精确二项分布. p-值为

$$p = 2\sum_{k=0}^{a1} \binom{a1 + a2}{k} p_0^k q_0^{a1+a2-k}, if \quad a1 < (a1 + a2)p_0$$

$$p = 2\sum_{k=a1}^{a1+a2} \binom{a1 + a2}{k} p_0^k q_0^{a1+a2-k}, if \quad a1 >= (a1 + a2)p_0$$

30-34岁妇女乳腺癌与OC使用的关系. 判断使用者和不使用者的发病率的差异显著性. a1=3, a2=9, t1=8250, t2=17430. 计算$V = 2.62 < 5$, 使用精确方法. $n = 3 + 9 = 12, p = 8250/25680 = 0.321, a1 = 3 < 12 * 0.321 = 3.9$, p-值为

使用OC的情况	病例数	人-年数
现在使用者	3	8250
从不使用者	9	17430

```
> 2*pbinom(3,12,prob=8250/25680)
[1] 0.8564199
```

57.3 率比

类似于危险率的比(risk ratio, RR), 那里的单位是人, 我们也可以使用于人-时间数据两个发病率的比较. 记λ_1, λ_2分别是暴露和非暴露组的发病率, 称λ_1/λ_2为率比(rate ratio). 属于 Poisson 分布. 精确的置信区间值来自 binom.test(此处的推导)

率比的点估计为

$$RR = (a1/t1)/(a2/t2)$$

ln(RR)近似于正态分布, 则

$$Var(ln(RR)) = 1/a1 + 1/a2$$

ln(RR)的置信区间为

$$(d_1, d_2) = ln(RR) \pm z_{1-\alpha/2}\sqrt{1/a1 + 1/a2}$$

(d_1, d_2)取反对数既得RR的置信区间.

ln(RR)的精确分布为二项分布.(推导略, 见[34] 公式 1)

对于下面的数据估计率比的点估计和区间估计.

使用OC的情况	病例数	人-年数
现在使用者	9	2935
从不使用者	239	135130

```
> library(rateratio.test)
> t1=2935
> t2=135130
> a1=9
> a2=239
# 精确的置信区间值来自 binom.test
> rateratio.test(c(a1, a2), c(t1, t2))

       Exact Rate Ratio Test, assuming Poisson counts

data:  c(a1, a2) with time of c(t1, t2), null rate ratio 1
p-value = 0.1702
alternative hypothesis: true rate ratio is not equal to 1
95 percent confidence interval:
 0.7831867 3.3470290
sample estimates:
 Rate Ratio      Rate 1      Rate 2
1.733757208 0.003066440 0.001768667

# 与二项分布比例的比较
> binom.test(a1,a1+a2, p = t1/(t1 + t2))

       Exact binomial test

data:  a1 and a1 + a2
number of successes = 9, number of trials = 248, p-value = 0.1160
alternative hypothesis: true probability of success is not equal to 0.02125810
95 percent confidence interval:
 0.01672615 0.06777020
sample estimates:
probability of success
          0.03629032
# 从二项分布计算置信区间
> b.ci=binom.test(a1,a1+a2, p = t1/(t1 + t2))$conf.int
> lambda.ci=t2 * b.ci/(t1 * (1 - b.ci))
> lambda.ci
[1] 0.7831867 3.3470290


>  fisher.test(matrix(c(a1, a2, t1-a1, t2-a2), 2, 2))
```

```
        Fisher's Exact Test for Count Data

data:  matrix(c(a1, a2, t1 - a1, t2 - a2), 2, 2)
p-value = 0.1158
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.7834183 3.3558917
sample estimates:
odds ratio
  1.736001
```

57.4 人-时间数据的功效及样本量估计

实际上是单样本二项检验的特例.

二项比例在指定的假设$p = p_1$下, 双侧检验的功效的正态近似为

$$power = \Phi[\sqrt{(p_0q_0)/(p_1q_1)}(z_{\alpha/2} + |p_0 - p_1|\sqrt{m}/\sqrt{p_0q_0})]$$

$$p_0 = t1/(t1 + t2)$$

$$p_1 = t1RR/(t1RR + t2)$$

$$m = m1 + m2 = \text{两组联合后事件的期望数}$$

$$m1 = n1[1 - exp(-ID_1t1^*)]$$

$$m2 = n2[1 - exp(-ID_2t2^*)]$$

$$n1, n2 = \text{组1,组2的个体数}$$

$$t1, t2 = \text{组1,组2的人-年数}$$

$$t1^*, t2^* = \text{组1,组2的个体平均人-年数}$$

$$ID_1, ID_2 = H_1\text{成立时组1,组2的发病密度}$$

对应于样本量, 两组联合后事件的期望数

$$m = \frac{p_0 q_0 (z_{1-\alpha/2} + z_{1-\beta}\sqrt{(p_1 q_1)})^2}{(p_1 - p_0)^2}$$

$$p_0 = t1/(t1 + t2)$$
$$p_1 = t1RR/(t1RR + t2)$$
$$t1, t2 = \text{组1,组2的人-年数}$$
$$ID_1, ID_2 = H_1\text{成立时组1,组2的发病密度}$$

对应于上述m, 每组个体数分别为

$$n1 = \frac{m}{(k+1) - exp(-ID_1 t1^*) - kexp(-ID_2 t2^*)}$$

$$n2 = kn1$$

假定10000名绝经后妇女, 没有癌症. 5000人随机指定接受雌激素补充疗法(ERT), 另外5000人指定安慰剂. 每个人平均随访5年, 对照组中期望发病率$300/10^5$, 假定ERT可以增加乳腺癌25%的发病率. 求此研究的功效.

下面是按照公式的解

```
power.persontime<-function(p0,p1,m,alpha=0.05){
   power=pnorm( sqrt(p0*(1-p0)/(p1*(1-p1))) * ( qnorm(alpha/2)+abs(p0-p1) * sqrt(
   power
}

> n1=n2=5000
> t_1=t_2=5 # t_1,t_2 为平均人年数
> ID2=300/10^5 # 发病密度/发病率
> ID1=1.25*ID2
> RR=ID1/ID2 # 率比
> m1=n1*(1-exp(-ID1*5)); m1
[1] 92.87656
> m2=n2*(1-exp(-ID2*5)); m2
[1] 74.4403
```

```
> m=m1+m2; m # 总的事件数
[1] 167.3169
> t1=t2=5000*5 # 人年数
> p0=t1/(t1+t2)
> p1=t1*RR/(t1*RR+t2)
> power.persontime(p0,p1,m)
[1] 0.2994232
```

需要多少样本量才会到80%的功效?

```
# t_1,t_2 为平均人年数
n.persontime<-function(p0,p1,ID1,ID2,t_1,t_2,alpha=0.05,power=0.8,k=1){
   m=( sqrt(p0*(1-p0))*qnorm(1-alpha/2) + sqrt(p1*(1-p1))*qnorm(power) )^2 /(p0-p
   n1=m/((k+1)-exp(-ID1*t_1)-k*exp(-ID2*t_2))
   n2=k*n1
   res=list(n1=n1,n2=n2)
   res
}

> n.persontime(p0,p1,ID1,ID2,t_1,t_2,alpha=0.05,power=0.8,k=1)
$n1
[1] 18928.05

$n2
[1] 18928.05
```

57.5 分层的人-时间数据的统计推断

一个研究是绝经后期妇女使用绝经后期激素是否引起心血管疾病和癌症的发生? 从1976年到1986年采用邮寄问卷在每2年随访得到下面的数据. 1976年有23607个绝经后妇女没有癌症, 其它妇女在随访期间都变成绝经后期. 随访在下列条件之一结束: 乳腺癌, 死亡, 到达随访最后. 乳腺癌及绝经后激素的使用与年龄有关, 因此控制年龄很重要.

年龄	(现在使用激素)病例数	人-年	(从不使用激素)病例数	人-年
39-44	12.00	10199.00	5.00	4722.00
45-49	22.00	14044.00	26.00	20812.00
50-54	51.00	24948.00	129.00	71746.00
55-59	72.00	21576.00	159.00	73413.00
60-64	23.00	4876.00	35.00	15773.00

我们可以象对累加发病率数据或计数数据(poisson分布)使用Mantel-Haenszel检验一样, 分析此处数据.

假设疾病与暴露的率比为RR(rate ratio), 我们假定所有层中的$RR = p_{1i}/p_{2i}$是相同的. 要检验假设 $H_0 : RR = 1 \quad vs \quad H_1 : RR = 1$

另外参考cox回归分析. 多个混杂变量时, 此方法也是合适的, 但是比较麻烦, 可以使用 Poisson 回归代替.

```
> a=c(12,22,51,72,23)
> b=c(10199,14044,24948,21576,4876)
> c=c(5,26,129,159,35)
> d=c(4722,20812,71746,73413,15773)
> epi.2by2(a, b, c, d, method = "cohort.time", conf.level = 0.95,verbose=T)

# incidence rate ratio
$IRR
      est      se    lower    upper
1 1.111168 1.702828 0.3914651 3.154033
2 1.253927 1.336004 0.7107125 2.212334
3 1.136953 1.179874 0.8221422 1.572309
4 1.540769 1.152634 1.1663375 2.035404
5 2.125741 1.307897 1.2561175 3.597415

# 直接计算RR
$IRR.crude
      est      se    lower    upper
1 1.253430 1.095866 1.047556 1.499764

# Mantel-Haenszel adjusted RR
$IRR.summary
      est      se lower upper
```

```
1 1.396736 47354.32    0   Inf

# 危险率差 Risk difference (attributable risk)
$AR
          est          se        lower       upper
1 0.0001177126 0.0005827568 -0.0010244697 0.0012598948
2 0.0003172260 0.0004142095 -0.0004946098 0.0011290618
3 0.0002462424 0.0003271105 -0.0003948824 0.0008873672
4 0.0011712122 0.0004291462  0.0003301011 0.0020123233
5 0.0024979993 0.0010526489  0.0004348454 0.0045611533

$AR.crude
          est          se       lower       upper
1 0.0004811295 0.0002040578 8.118347e-05 0.0008810755

$AR.summary
         est se    lower    upper
1 0.000536203  0 -0.792985 0.7940575

# population attributable risk
$PAR
          est          se        lower       upper
1 8.046047e-05 0.0005482703 -0.0009941295 0.0011550505
2 1.278151e-04 0.0003154916 -0.0004905372 0.0007461673
3 6.353295e-05 0.0002105057 -0.0003490507 0.0004761166
4 2.660316e-04 0.0002347413 -0.0001940529 0.0007261161
5 5.898709e-04 0.0005260331 -0.0004411350 0.0016208768

# population attributable fraction
$PAF
        est      lower    upper
1 0.07062062 -1.09729861 0.4820793
2 0.09281504 -0.18655943 0.2511615
3 0.03412920 -0.06129480 0.1031313
4 0.10939426  0.04445151 0.1585549
5 0.21000422  0.08085536 0.2863193

# 暴露与非暴露比例的差异
$chisq
  test.statistic df    p.value
1     0.0392107  1 0.843031819
```

```
2     0.6118946  1 0.434075353
3     0.6017654  1 0.437905228
4     9.3771646  1 0.002197051
5     8.2403647  1 0.004096890

# 暴露与非暴露比例的联合差异
$chisq.summary
  test.statistic df    p.value
1      6.100691  1 0.01351290
```

下面是使用前瞻性方法的结果

```
> r=epi.2by2(a, b, c, d, method = "cohort.count", conf.level = 0.95,verbose=T)
> r
# risk ratio
$RR
      est      se    lower    upper
1 1.111037 1.702333 0.3916422 3.151865
2 1.253530 1.335729 0.7107738 2.210742
3 1.136673 1.179682 0.8222029 1.571420
4 1.538970 1.152392 1.1654570 2.032189
5 2.120456 1.307245 1.2542193 3.584966

$RR.crude
      est      se   lower    upper
1 1.252829 1.077345 1.082623 1.449793

$RR.summary
      est        se   lower    upper
1 1.396736 0.09258788 1.16494 1.674655

# ODDS ratio
$OR
      est      se    lower    upper
1 1.111168 1.703324 0.3912419 3.155832
2 1.253927 1.336278 0.7104259 2.213226
3 1.136953 1.180067 0.8218792 1.572812
4 1.540769 1.152877 1.1658554 2.036245
```

```
5 2.125741 1.308551 1.2548879 3.600940

$OR.crude
      est      se    lower    upper
1 1.253430 1.095977 1.047348 1.500063

$OR.summary
      est       se    lower    upper
1 1.397811 0.1361457 1.070437 1.825306

$AR
         est          se        lower       upper
1 0.0001174499 0.0005817975 -0.0010228523 0.0012577521
2 0.0003163347 0.0004133069 -0.0004937319 0.0011264013
3 0.0002452990 0.0003261382 -0.0003939201 0.0008845181
4 0.0011647941 0.0004271271  0.0003276404 0.0020019477
5 0.0024807669 0.0010457419  0.0004311505 0.0045303832

$AR.crude
         est          se      lower       upper
1 0.0004790778 0.0002033675 8.04849e-05 0.0008776707

$AR.summary
         est          se       lower      upper
1 0.0007177012 0.0003394705 5.235121e-05 0.001383051

$AF
       est     lower    upper
1 0.09994006 -1.5533511 0.6827276
2 0.20225288 -0.4069174 0.5476632
3 0.12023980 -0.2162448 0.3636330
4 0.35021476  0.1419675 0.5079197
5 0.52840334  0.2026913 0.7210573

$PAR
         est          se        lower       upper
1 8.028389e-05 0.0005473838 -0.0009925687 0.0011531365
2 1.274800e-04 0.0003148774 -0.0004896684 0.0007446284
3 6.330109e-05 0.0002099306 -0.0003481553 0.0004747575
4 2.648127e-04 0.0002339375 -0.0001936964 0.0007233217
5 5.869163e-04 0.0005240597 -0.0004402218 0.0016140545
```

```
$PAF
       est      lower    upper
1 0.07054593 -1.09648311 0.4819253
2 0.09269924 -0.18650383 0.2510123
3 0.03406794 -0.06126936 0.1030294
4 0.10915785  0.04424960 0.1583126
5 0.20953926  0.08037758 0.2859365

$chisq
  test.statistic df    p.value
1     0.0392107  1 0.843031819
2     0.6118946  1 0.434075353
3     0.6017654  1 0.437905228
4     9.3771646  1 0.002197051
5     8.2403647  1 0.004096890

$chisq.summary
  test.statistic df   p.value
1      6.100691  1 0.01351290

# RR的齐性检验
$RR.homog
  test.statistic df   p.value
1      4.774888  4 0.3111848

# OR的齐性检验
$OR.homog
  test.statistic df   p.value
1      4.002160  4 0.4057135
```

57.6 分层的人-时间数据的功效及样本量

TODO: [11] Page 671, 14.6

57.7 发病率数据中趋势性的检验

TODO: [11] Page 676, 14.7

Chapter 58

生存分析

前面的发病率比较中, 一个假设是发病率不随时间变化. 但是在许多情况下这个假设是不能保证的. 这样就产生了生存分析.

R-cran 网站的介绍生存分析的页面很好 http://cran.r-project.org/web/views/Survival.html

(参考 prodlim , survival 包)

58.1 概念

58.1.1 危险率(hazard rate)

可以随时间变化的发病率称为危险率(hazard rate)

58.1.2 死亡危险率(mortality risk)

生物统计中危险率函数常被看作是死亡危险率的(mortality risk)指标

58.1.3 生存概率(survival probability)

不发生疾病的概率通常称为生存概率(survival probability)

58.1.4 生存函数(survival function)

将生存概率记为时间的函数, 即对每个$t >= 0$的点, 可以存活到时间t以上的概率称为生存函数(survival function).

58.1.5 危险函数(hazard function)

h(t)是单位时间内时刻t上一个事件瞬时发生的概率, 即一个到t时刻存活的个体(即还没有发生事件)在t时的瞬时发病率. 特例

$$h(t) = \lim_{\Delta t \to 0} [\frac{S(t) - S(t + \Delta t)}{\Delta t}]/S(t)$$

例如, 0岁(出生时)的100000名男性, 80908名活到60岁, 79539名活到61岁, 则60岁的危险率近似为

$$h(60) = \frac{80908 - 79539}{80908} = \frac{1369}{80908} = 0.017$$

即60岁存活的男性, 下一年有1.7%的人会死亡.

58.1.6 失访或截尾观察(censored observation)

随访周期内未达到疾病终点的病人称为失访或截尾观察(censored observation). 一个病人在随访到时刻t时失访, 称为t时失访. 即缺失状态. problem 函数可以计算失访.

58.2 时间序列的 Kaplan-Meier 估计

Kaplan-Meier 估计又叫做乘积限(product limit)估计

Kaplan与Meier(1985)提供了一种从缺失(loss)数据中获取信息的方法, 即在缺失前死亡(death)还没有发生.([14] Page 62) 一个事实是: 若死亡发生在时刻x后, 那么很显然也发生在x前的任意时刻之后. 由条件概率的定义, 对于$x_0 < x_1$, 我们有

$$P(X > x_1) = P(X > x_1, X > x_0) = P(X > x_1 | X > x_0) P(X > x_0)$$

假设第一年初有100个研究对象, 年底剩下30个存活. 我们用下式估计$P(1)$

$$P(1) = P(X > 1) = 30/100 = 0.3$$

这里X表示研究对象个体的寿命.

第二年初又有另外1000个个体参加试验. 第二年底, 1000个中有250个存活, 而最初的100个中存活的只有10个了. 我们可以使用最初的100个个体来估计$P(2)$

$$P(2) = P(X > 2) = 10/100 = 0.1$$

但是我们可以用第二年新参加的个体信息来更新估计$P(1)$. 因为到第二年底参加了1年的个体共有1100个, 其中共有$250 + 30 = 280$个存活, 改进后的$P(1)$的估计为

$$P(1) = P(X > 1) = 280/1100 = 0.255$$

由条件概率, 我们使用改进后的$P(1)$来改进$P(2)$

$$P(2) = P(X > 2) = P(X > 2 | X > 1) P(X > 1)$$

不幸的是, 我们无法改进$P(X > 2|X > 1)$, 因为第三年的试验还没有做, 我们不知道在接下来的1年1000个个体有多少存活.故我们使用下面的估计量(它仅用到了已知信息. 即第一年底有30个存活, 第二年底有10个存活)

$$P(X > 2|X > 1) = 10/30$$

那么$P(2)$的改进为

$$P(2) = P(X > 2|X > 1) P(X > 1) = \frac{10}{30} \frac{280}{1100} = 0.085$$

Kaplan与Meier推广了上面的方法. 设$u_1 < u_2 < ... < u_k$表示k个个体的寿命(从开始到死亡, 或缺失的持续时间). 令

$$p_i = P(X > u_i | X > u_{i-1}$$

用下式估计

$$p_i = \frac{\text{到时刻}u_i\text{存活的个体数}}{\text{到时刻}u_{i-1}\text{仍然观测到的存活的个体数}}$$

在时刻u_i缺失的个体, 可以认为在时刻u_i以后仍然存活. 第一次死亡或缺失的计算中, P_1的分母是个体的总数.

$P(x)$的Kaplan-Meier估计为

$$P(x) = \begin{cases} 1 & if x < u_1 \\ \prod_{u_i <= x} p_i & if x >= u_i \end{cases}$$

有时候需要求出删失数据的方差.

下面是另外一个例子. 要测试10个汽车风扇皮带的质量. 我们记录每个皮带所能承受的里程数. 测试结束后, 5个带都断了, 寿命分别为77,47,81,56,80(千英里). 另外5个没有断, 分别是62,60,43,71,37. 那么生存函数Kaplan-Meier估计如下

	u	r	p_i	$P(u_i)$
1	37.00	loss	10/10	1.00
2	43.00	loss	9/9	1.00
3	47.00	death	7/8	0.88
4	56.00	death	6/7	0.75
5	60.00	loss	6/6	0.75
6	62.00	loss	5/5	0.75
7	71.00	loss	4/4	0.75
8	77.00	death	2/3	0.50
9	80.00	death	1/2	0.25
10	81.00	death	0/1	0.00

下面是survival包的结果, 与手工计算一致

```
> f= survfit(Surv(time, status) )
> summary(f)
Call: survfit(formula = Surv(time, 1 - status))

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   47      8       1    0.875   0.117       0.673           1
   56      7       1    0.750   0.153       0.503           1
   77      3       1    0.500   0.228       0.204           1
   80      2       1    0.250   0.210       0.048           1
   81      1       1    0.000      NA          NA          NA
> plot(f) # 同时绘制上界和下界
```

下面是R prodlim 包的结果, 和手工计算一致.

```
> time=c(37,43,47,56,60,62,71,77,80,81)
# 缺失的状态设为0, 死亡的设为1
> status=c(0,0,1,1,0,0,0,1,1,1)
> fit=prodlim(Hist(time,status)~1)
> summary(fit) # surv 为概率, 后面依次为surv的标准误, 下界, 上界
   n.risk n.lost n.event  surv   se.surv     lower    upper
37     10      1       0 1.000 0.0000000 1.00000000 1.0000000
43      9      1       0 1.000 0.0000000 1.00000000 1.0000000
47      8      0       1 0.875 0.1169268 0.64582770 1.0000000
56      7      0       1 0.750 0.1530931 0.44994302 1.0000000
60      6      1       0 0.750 0.1530931 0.44994302 1.0000000
62      5      1       0 0.750 0.1530931 0.44994302 1.0000000
71      4      1       0 0.750 0.1530931 0.44994302 1.0000000
77      3      0       1 0.500 0.2282177 0.05270146 0.9472985
80      2      0       1 0.250 0.2104064 0.00000000 0.6623889
81      1      0       1 0.000       NaN       NaN      NaN

> plot(fit) # 绘出生存函数的图像
> fit

Call: prodlim(formula = Hist(time, status) ~ 1)

Kaplan-Meier estimator for the event time survival function
```

```
No covariates

RightCensored response of a survival model

No.Observations: 10

Pattern:
             Freq
 event         5
 right.censored 5
```

58.3 对数秩(log rank)检验

累加发病率如果随时间不同, 则前面介绍的两个发病率之间的比较的方法不是很有效. 我们将使用对数秩检验方法(此方法与"对数"完全没有关系)检验两个生存曲线的发病率是否相同. [1]

下面表的行为年龄, 列为戒烟天数的人数, 例如大于40岁的戒烟天数小于90天的人数为92, 即92人在小于90天内又开始吸烟.

年龄/戒烟天数	<= 90	91-180	181-270	271-364	365
> 40	92	4	4	1	19
<= 40	88	7	3	2	14

[1]关于logrank名称的解释(下面资料来自网络): SAS的"LOG窗口"的中文意思是"对数窗",因为生存分析的 Log rank 在网络上就被译为"对数秩"。不信？在Google里用"对数秩"检索，至少可见四五个页面都是"对数秩（log rank）"，其中也有很出名的院校的统计教学计划，还有教材、辅导、教学大纲，更有著名杂志和期刊。学学生存分析的 log rank 检验，就知道log rank 检验和"对数"毫无关系，log rank 检验的LOG是SAS"LOG窗口"LOG，非"对数"LOG。如果 Log rank 译为"对数秩"，SAS的"LOG窗口"当然就是"对数窗"了。log 还有登录, 日志的意思.

最近一本翻译的美国生物统计教材，也把 Log rank 译为"对数秩"，正式出版物，或许也不算错。

log rank 可以翻译成"时序秩"，更切合生存分析的用途，也比较合本意！

我们把时间分段, 数据归为4个列联表, 然后使用Mantel-Haenszel检验, 就得到对数秩检验. 如果检验统计量chisq比较小, 则接受零假设, 此处 X-squared = 0.2932, df = 1, p-value = 0.5882, 即两个年龄的恢复吸烟的发病率上没有显著不同. [2]

```
> x=array(c(92,88,28,26,4,7,24,19,4,3,20,16,1,2,19,14),
      dim=c(2,2,4),
      dimnames=list(c(">40","<=40"),
         c("恢复抽烟","继续戒烟"),
         c("0-90天","91-180天","181-270天","271-365天")))
> x
, , 0-90天

    恢复抽烟 继续戒烟
>40       92       28
<=40      88       26

, , 91-180天

    恢复抽烟 继续戒烟
>40        4       24
<=40       7       19

, , 181-270天

    恢复抽烟 继续戒烟
>40        4       20
<=40       3       16

, , 271-365天

    恢复抽烟 继续戒烟
>40        1       19
<=40       2       14

# 对数秩检验
> mantelhaen.test(x)
```

[2]TODO: 此处结果与survdiff及surv_test函数的结果不同, 差距较大, 不知为何? 可能是我的数据重构有问题

```
        Mantel-Haenszel chi-squared test with continuity correction

data:  x
Mantel-Haenszel X-squared = 0.2932, df = 1, p-value = 0.5882
alternative hypothesis: true common odds ratio is not equal to 1
95 percent confidence interval:
 0.5036551 1.3988182
sample estimates:
common odds ratio
        0.839358
```

下面给出另外几个例子(取自[22]). 注意数据需要是data.frame, time 为存活时间, event=TRUE 为死亡, =FALSE 为中途缺失, group 为分组.

```
> data("glioma", package = "coin")
> g3 <- subset(glioma, histology == "Grade3")
> g3
   no. age    sex histology   group event time
1    1  41 Female    Grade3    RIT  TRUE   53
2    2  45 Female    Grade3    RIT FALSE   28
3    3  48   Male    Grade3    RIT FALSE   69
4    4  54   Male    Grade3    RIT FALSE   58
5    5  40 Female    Grade3    RIT FALSE   54
6    6  31   Male    Grade3    RIT  TRUE   25
7    7  53   Male    Grade3    RIT FALSE   51
8    8  49   Male    Grade3    RIT FALSE   61
9    9  36   Male    Grade3    RIT FALSE   57
10  10  52   Male    Grade3    RIT FALSE   57
11  11  57   Male    Grade3    RIT FALSE   50
20   1  27   Male    Grade3 Control  TRUE   34
21   2  32   Male    Grade3 Control  TRUE   32
22   3  53 Female    Grade3 Control  TRUE    9
23   4  46   Male    Grade3 Control  TRUE   19
24   5  33 Female    Grade3 Control FALSE   50
25   6  19 Female    Grade3 Control FALSE   48
> survdiff(Surv(time, event) ~ group, data = g3)
Call:
survdiff(formula = Surv(time, event) ~ group, data = g3)
```

```
            N Observed Expected (O-E)^2/E (O-E)^2/V
group=Control  6        4     1.49      4.23      6.06
group=RIT     11        2     4.51      1.40      6.06

 Chisq= 6.1  on 1 degrees of freedom, p= 0.0138
>  library("coin")
> surv_test(Surv(time, event) ~ group, data = g3,distribution = "exact")

       Exact Logrank Test

data:  Surv(time, event) by group (Control, RIT)
Z = 2.1711, p-value = 0.02877
alternative hypothesis: two.sided
```

一些计算对数秩相关的函数: survival: survdiff(surv.marr sex)

survival: summary(clogit(case alcohol + strata(matset)))

coin: surv_test(Surv(time, event) stadium, data = ocarcinoma)

Hmisc: cpower(2, 1000, .2, 25, accrual=2, tmin=1, noncomp.c=10, noncomp.i=17.5)

58.4 Cox比例风险回归模型

当有多个危险因素, 又有多层时, 方便的方法是对生存数据使用回归模型, 常用的是Cox比例风险回归模型(Cox proportional hazards regression model).

58.4.1 模型及检验

在模型中, 危险率可以表示为

$$h(t) = h_0(t)exp(b_1x_1 + \cdots + b_kx_k)$$

此处$x_1, \cdots, x_k$是一组独立变量, $h_0(t)$是基准状态下t时刻的基准危险率, 它代表所有变量全部取0时的危险率. 假设$H_0 : b_i = 0 \quad vs \quad H_1 : b_i \neq 0$. 对此的检验方法为:

(1) 计算检验统计量$z = \hat{b_i}/se(\hat{b_i}) \sim N(0, 1)$

(2) 判断显著性, $|z| > z_{1-\alpha/2}$, 拒绝零假设, 否则接受

我们把方程变形, 可以写作

$$ln[\frac{h(t)}{h_0(t)}] = b_1 x_1 + \cdots + b_k x_k$$

我们可以按照多重logistic回归模型的方式去解系数b_i, 特别在x为二态独立变量时.

58.4.2 对二态独立变量危险比的估计

设有一个二态危险因子x_i, 当危险存在时$x_i = 1$, 不存在时$x_i = 0$, 量$exp(b_i)$代表了如下两个个体的危险率之比: 在其它协变量全部相同的情况下, 一个个体有x_i出现$(x_i = 1)$而另外一个没有$(x_i = 0)$, 这个危险率之比可以称为相对危险率, 可以看作其它协变量全部相同时, 在t时刻有危险因子$(x_i = 1)$相对于没有危险因子$(x_i = 0)$的个体在单位时间内发生事件的相对危险率. b_i的双侧$100\%(1 - \alpha)$CI为(e^{c1}, e^{c2})

$$c1 = \hat{b_i} - z_{1-\alpha/2} se(\hat{b_i})$$

$$c2 = \hat{b_i} + z_{1-\alpha/2} se(\hat{b_i})$$

58.4.3 对连续独立变量危险比的估计

设有一个连续的危险因子x_i, 两个个体在其它协变量全部相同, 仅在第i个独立变量(危险因子)x_i上相差Δ, 则量$exp(b_i \Delta)$两个个体的危险率比. 可以看作其它协变量全部相同时, 在t时刻一个危险因子取$x_i + \Delta$另外一个危险因子取x_i在单位时间内发生事

件的瞬时相对危险率. $b_i\Delta$的双侧$100\%(1-\alpha)$CI为(e^{c1}, e^{c2})

$$c1 = \Delta[\hat{b_i} - z_{1-\alpha/2}se(\hat{b_i})]$$

$$c2 = \Delta[\hat{b_i} + z_{1-\alpha/2}se(\hat{b_i})]$$

它可以看作是多重 logistic 回归的拓广: 即事件发生与时间有关, 而不是简单考察事件是否发生.

由于没有数据, 我使用[22] 的例子

```
> library(ipred)
> data(GBSG2)
> GBSG2
   horTh age menostat tsize tgrade pnodes progrec estrec time cens
1     no  70     Post    21     II      3      48     66 1814    1
2    yes  56     Post    12     II      7      61     77 2018    1
3    yes  58     Post    35     II      9      52    271  712    1
4    yes  59     Post    17     II      4      60     29 1807    1
5     no  73     Post    35     II      1      26     65  772    1
6     no  32      Pre    57    III     24       0     13  448    1
......
683   yes  53     Post    25    III     17       0      0  186    0
684    no  51      Pre    25    III      5      43      0  769    1
685    no  52     Post    23     II      3      15     34  727    1
686    no  55     Post    23     II      9     116     15 1701    1

# 对去除了 time, cens 的所有其它变量进行回归
# exp(coef) 即其它协变量全部相同时, 在t时刻的瞬时相对危险比
> coxph(Surv(time, cens) ~ ., data = GBSG2)
Call:
coxph(formula = Surv(time, cens) ~ ., data = GBSG2)


                coef exp(coef) se(coef)     z      p
horThyes    -0.346278     0.707 0.129075 -2.683 7.3e-03
age         -0.009459     0.991 0.009301 -1.017 3.1e-01
menostatPost  0.258445     1.295 0.183476  1.409 1.6e-01
tsize         0.007796     1.008 0.003939  1.979 4.8e-02
```

```
tgrade.L     0.551299     1.736 0.189844  2.904 3.7e-03
tgrade.Q    -0.201091     0.818 0.121965 -1.649 9.9e-02
pnodes       0.048789     1.050 0.007447  6.551 5.7e-11
progrec     -0.002217     0.998 0.000574 -3.866 1.1e-04
estrec       0.000197     1.000 0.000450  0.438 6.6e-01

Likelihood ratio test=105  on 9 df, p=0  n= 686
```

58.4.4 功效及样本量估计

TODO: [11] Page 699, 14.12

Part VIII

杂项

一 件 东 西 如 果 完 整, 必 须 包 含 乱 七 八 糟.

Chapter 59

马尔可夫链与生物学

参考 [32] chapter 10, 11, 12

参考 [8] 第五章 马尔可夫链数学模型

随机过程包括各种不同的过程. 两个主要的类别为到达时间过程和马尔可夫过程. 前者包括Bernoulli过程和Poisson过程.

59.1 马尔可夫过程

定义: 请参考相关教科书

下面是基因($5' - 3'$)随时间变化的情况([32] figure 10-2), 这就是一个马尔可夫链, 因为每一个的状态只与前一个有关, 而与前一个之前的所有状态无关. 虽然碱基之间可能互相影响, 但是我们的模型一般会忽略, 并假设碱基之间的变化互不影响(独立性).

```
ATCGCCATCGAATACTCTAGCATG t=0
ATCcCCATCGAATACTCTAGCATG t=1
ATCcCCAaCGAATACTCTAGCATG t=2
ATCcCCAaCGAATACcCTAGCATG t=3
ATCcCCATaGAATACgCTAcCATG t=4
```

59.2 转移图

略. 很有用, 尤其分析马尔可夫链的吸收性!!!

59.3 几个例子

59.3.1 动物健康

下面是一个马尔可夫链的例子([8] Page 163). 某动物群体, 患病是唯一死亡原因. 经验表明一日内健康个体患病的概率是0.01, 患病个体恢复健康的概率是0.9, 死亡的概率是0.01. 下面是单位时间(一天)转移概率矩阵

开始状态/到达状态	健康	患病	死亡
健康	0.99	0.01	0
患病	0.9	0.09	0.01
死亡	0	0	1

59.3.2 豌豆杂交(Aa基因型)

下面是杂交的例子([8] Page 165). 使用三种基因型AA,Aa,aa同杂交型Aa杂交, 其马尔可夫转移矩阵(即得到后代基因型的概率矩阵)为 表示为

状态/后代基因型	AA	Aa	aa
AA与Aa杂交	0.5	0.5	0
AA与Aa杂交	0.25	0.5	0.25
AA与Aa杂交	0	0.5	0.5

```
> Aa=matrix(c(0.5,0.25,0,0.5,0.5,0.5,0,0.25,0.5),nc=3,
      dimnames=list(c("AA*Aa","Aa*Aa","aa*Aa"),c("AA","Aa","aa")))
> Aa
       AA   Aa   aa
AA*Aa 0.50 0.25 0.00
Aa*Aa 0.25 0.25 0.25
aa*Aa 0.00 0.25 0.50
```

59.3.3 豌豆杂交(AA基因型)

如果三种基因型与AA杂交, 其转移矩阵具体值会改变为

```
# 三种基因型与AA杂交的转移矩阵
> AA=matrix(c(1,0.5,0,0,0.5,1,0,0,0),nc=3,
      dimnames=list(c("AA*AA","Aa*AA","aa*AA"),c("AA","Aa","aa")))
> AA
      AA  Aa aa
AA*AA 1.0 0.0  0
Aa*AA 0.5 0.5  0
aa*AA 0.0 1.0  0
```

假设开始群体基因型的比例为0.2,0.3,0.5

```
> x=matrix(c(0.2,0.3,0.5),nr=1,
      dimnames=list(c("ratio"),c("AA","Aa","aa")))
> x
      AA  Aa  aa
ratio 0.2 0.3 0.5
```

与基因型Aa杂交, 下一代的基因型分配比例为

```
# 注意 * 是不对的. 矩阵相乘应该使用 %*% 符号
```

```
# 下一代的基因型分配比例
> x2<-x%*%Aa; x2
       AA  Aa   aa
ratio 0.175 0.5 0.325
```

若继续与Aa杂交,则

```
> x3<-x2%*%Aa; x3 # 第三代基因型分配比例
        AA  Aa    aa
ratio 0.2125 0.5 0.2875
> x4<-x3%*%Aa; x4 # 第4代基因型分配比例
         AA  Aa     aa
ratio 0.23125 0.5 0.26875

......

> x9<-x8%*%Aa; x9 # 第9代基因型分配比例
           AA  Aa       aa
ratio 0.2494141 0.5 0.2505859
```

实际上, 可以看到, Aa 是转移一代的概率, $Aa\%*\%Aa$ 是转移2代的概率矩阵, 自乘 n 次的结果就是转移 n 代的概率矩阵. 有

```
# 效率非常低的矩阵连乘函数!!!
mulprod<-function(X,n){
   tmp<-X
   if(n>=2){
   for (i in 2:n){
   tmp<-tmp%*%X
}
}
   tmp
}
# 转移8代, x到第九代 x9
> x%*%mulprod(Aa,8)
```

```
          AA  Aa       aa
ratio 0.2494141 0.5 0.2505859
```

看看转移很多代的结果, 分配比例稳定下来了. 实际上其n代转移矩阵随着n的增大而平稳.

```
# 第21代基因型分配比例
> x%*%mulprod(Aa,20)
          AA  Aa       aa
ratio 0.2499999 0.5 0.2500001
# 第51代基因型分配比例
> x%*%mulprod(Aa,50)
      AA  Aa   aa
ratio 0.25 0.5 0.25

# 看看n代转移矩阵
# x %*% 此矩阵就是转移20代后的基因型分配比例
# x %*% 两次就是转移40代的基因型分配比例(顺便猜想一个提高mulprod()函数的方法?)
> mulprod(Aa,20)
           AA  Aa       aa
AA*Aa 0.2500005 0.5 0.2499995
Aa*Aa 0.2500000 0.5 0.2500000
aa*Aa 0.2499995 0.5 0.2500005
> mulprod(Aa,50)
       AA  Aa   aa
AA*Aa 0.25 0.5 0.25
Aa*Aa 0.25 0.5 0.25
aa*Aa 0.25 0.5 0.25
> mulprod(Aa,100)
       AA  Aa   aa
AA*Aa 0.25 0.5 0.25
Aa*Aa 0.25 0.5 0.25
aa*Aa 0.25 0.5 0.25
```

59.4 正则马尔可夫链

59.4.1 定理

对于马尔可夫链, 若存在正整数k使得其转移概率矩阵乘幂P^k的所有元素值都大于0, 则称该马尔可夫链是正则的(regular).

乘幂计算非常不方便, 一般利用转移图来查看. 如果任何状态经有限步可以到达任何其它状态, 则该马尔可夫链是正则的.

定理([8] Page 171, 有证明): 对于正则马尔可夫链的转移矩阵P, 有以下结论

1. 当 $t \to \infty$, $P^t \to W$(随机矩阵)
2. W的行向量均相同, 即

$$W = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \\ v_1 & v_2 & \cdots & v_n \\ \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

3. W的所有分量都大于零

其中定理2说明, 随着时间延续, 无论初始状态分布如何, 最终将以一定的概率分布到达某一状态. 定理2的推论告诉我们, 分配比例将趋近稳定.

定理2的推论([8] Page 173, 有证明): 如果 $V = [v_1, \cdots, v_n]$ 是矩阵W的行向量, 那么

1. 对于任何随机向量 $X = [x_1, \cdots, x_n]$, 当 $t \to \infty$, 有 $XP^t \to V$
2. 存在唯一的随机向量V使得 $VP = V$, V亦称作随机矩阵P的不动点向量(stationary vector)

59.4.2 不动点向量的计算

推论给出了计算不动点向量的公式.

$$VP = V$$

方程组形式为

$$\begin{aligned} v_1P_{11} + v_2P_{21} + \cdots + v_nP_{n1} &= v_1 \\ v_1P_{12} + v_2P_{22} + \cdots + v_nP_{n2} &= v_2 \\ &\cdots \\ v_1P_{1n} + v_2P_{2n} + \cdots + v_nP_{nn} &= v_n \\ v_1 + \cdots + v_n &= 1 \end{aligned}$$

可以变换为

$$\begin{aligned} v_1(P_{11} - 1) + v_2P_{21} + \cdots + v_nP_{n1} &= 0 \\ v_1P_{12} + v_2(P_{22} - 1) + \cdots + v_nP_{n2} &= 0 \\ &\cdots \\ v_1P_{1n} + v_2P_{2n} + \cdots + v_n(P_{nn} - 1) &= 0 \\ v_1 + \cdots + v_n &= 1 \end{aligned}$$

前n个方程式是奇异的, 从其中去掉一个, 例如, 选择去掉第一个方程式, 然后与最后一个约束构成方程组, 解此方程组即得到不动点向量.

根据上面的公式及思路, 我们编写函数求解不动点向量

```
# 求解不动点向量的函数, P为转移矩阵(不需要转置)
SV<-function(P){
   d<-dim(P)[1]
   for (i in 1:d){
      P[i,i]<-P[i,i]-1
}
   P1<-t(cbind(P[,2:d],rep(1,d)))
   b<-c(rep(0,d-1),1)
   v<-solve(P1,b)
   v
}
```

现在看与Aa杂交的例子. 其不动点为

```
# 再次写出转移矩阵
Aa=matrix(c(0.5,0.25,0,0.5,0.5,0.5,0,0.25,0.5),nc=3,
       dimnames=list(c("AA*Aa","Aa*Aa","aa*Aa"),c("AA","Aa","aa")))

> SV(Aa)
[1] 0.25 0.50 0.25
```

说明多次与Aa杂交, 最终基因型分配比例将稳定在$\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$

59.5 Hardy-Weiberg定理

[16] Page 25 [8] Page 176-178

英国数学家Hardy与德国医生Weiberg于1908年分别发现. 并由Punett于1950年与Stem于1943年的论文中分别介绍. 现在已经被公认为群体遗传学的创始理论

59.5.1 定理

设某群体有n个个体. 其中三种基因型AA,Aa,aa的个体数量分别为x,y,z, $x+y+z=n$. 基因型频率的估计可以是

$$d = \frac{x}{n}$$

$$h = \frac{y}{n}$$

$$r = \frac{z}{n}$$

$$d + h + r = 1$$

群体中等位基因A与a的频率分别用p, q表示. (其中p=A的个数/(A的个数+a的个数), q=a的个数/(A的个数+a的个数)), 那么

$$p = d + \frac{h}{2}$$

$$q = r + \frac{h}{2}$$

$$p + q = 1$$

那么随机交配第二代产生基因型AA的频率为p^2, Aa为$2pq$, aa为q^2

如果我们描述为: 群体(继代)随机交配, 在不产生选择, 突变和迁移的情况下, 基因频率与基因型频率每代保持不变, 合子系列频率等于配子系列频率的二项式平方

$$AA\ p^2 + Aa\ 2pq + aa\ q^2 = (Ap + aq)^2$$

这就是Hardy-Weiberg定理.

实际上可以这样列表多个基因亦可用平方式展开.

交配/后代基因型	AA	Aa	aa
$AA * AA$	d^2		
$AA * Aa$	$dh/2$	$dh/2$	
$AA * aa$		dr	
$Aa * AA$	$dh/2$	$dh/2$	
$Aa * Aa$	$h^2/4$	$h^2/2$	$h^2/2$
$Aa * aa$		$hr/2$	$hr/2$
$aa * AA$		dr	
$aa * Aa$		$hd/2$	$hd/2$
$aa * aa$			r^2
	$(d + h/2)^2$ p^2	$2(d + h/2)(h/2 + r)$ $2pq$	$(h/2 + r)^2$ q^2

转移矩阵可以写作

$$P = \begin{bmatrix} d + h/2 & r + h/2 & 0 \\ d/2 + h/4 & d/2 + h/2 + r/2 & r/2 + h/4 \\ 0 & d + h/2 & r + h/2 \end{bmatrix} = \begin{bmatrix} p & q & 0 \\ p/2 & 1/2 & q/2 \\ 0 & p & q \end{bmatrix}$$

解得不动点

$$V = [p^2, 2pq, q^2]$$

59.5.2 例子

设基因型频率$d = 0.3, h = 0.2, r = 0.5$, A的基因频率为$p = d + h/2 = 0.4$, a的基因频率为$q = r + h/2 = 0.6$, 则转移矩阵

```
# 转移矩阵
P=matrix(c(0.4,0.2,0,.6,.5,.4,0,.3,.6),nc=3)

> P
     [,1] [,2] [,3]
[1,]  0.4  0.6  0.0
[2,]  0.2  0.5  0.3
[3,]  0.0  0.4  0.6
# 不动点向量为
> SV(P)
[1] 0.16 0.48 0.36

# 初始分配
x=c(0.3,0.2,0.5)


> x%*%P  # 一次转移(第一代)后已经达到不动点
     [,1] [,2] [,3]
[1,] 0.16 0.48 0.36
> x%*%P%*%P # 二次转移
     [,1] [,2] [,3]
[1,] 0.16 0.48 0.36
```

59.6 吸收马尔可夫链

(证明及其它证明参考[8] 第五章相关部分)

59.6.1 吸收状态

马尔可夫链中若状态i(转移矩阵第i行)满足

1. $p_{ii} = 1$
2. $p_{ik} = 0, \quad k \neq i$

称该状态为吸收状态(absorbing state). 即不能离开的状态, 例如死亡, 称为吸收状态. 在转移图中表现就是到自身的值为1的弧.

59.6.2 吸收马尔可夫链

满足下面条件的马尔可夫链称吸收马尔可夫链.

1. 至少存在一个状态为吸收状态
2. 从任何状态经有限步可以到达吸收状态

实际上转移图中表示就是从任何其它状态可以到达某个吸收状态.

非吸收状态称转移状态(transient state)

59.6.3 规范的转移矩阵写法

一般将有r个吸收状态, k个转移状态的吸收马尔可夫链转移矩阵写作

$$P = \begin{bmatrix} E & O \\ R & Q \end{bmatrix}$$

其中

- E: $r*r$单位矩阵
- O: $r*k$零矩阵
- R: $k*r$矩阵, 表示从转移状态一步就达到吸收状态的概率
- O: $k*k$矩阵, 从一个转移状态到另一个转移状态的概率

59.6.4 定理: 最终进入吸收状态的概率

对于吸收马尔可夫链, 从任何状态出发最终进入吸收状态的概率为1

59.6.5 转移矩阵的幂

转移矩阵的t次幂写作

$$P = \begin{bmatrix} E & O \\ R_t & Q^t \end{bmatrix}$$

其中 $R_t = (E + Q + Q^2 + \cdots + Q^{t-1})R$

对于转移矩阵的幂有以下结论

1. $t \to \infty$时, $Q^t \to O$, O为零矩阵
2. 矩阵$E - Q$可逆. 此处 E 为k阶单位矩阵, 与Q阶数相同
3. $N = (E - Q)^{-1} = E + Q + Q^2 + \cdots$, 为k阶方阵, 称为该吸收马尔可夫链的基本矩阵(fundamental matrix)

59.6.6 定理: 进入次数的数学期望

具有r个吸收状态的吸收马尔可夫链, 从转移状态$i_1 = r + i$开始, 到达吸收状态前, 进入指定转移状态$j_1 = r + j$的次数的数学期望是基本矩阵$N = (E - Q)^{-1}$的第i行 第j列.

推论: 具有r个吸收状态的吸收马尔可夫链, 从转移状态$i_1 = r + i$开始, 到达吸收状态前, 在所有转移状态之间传递的步数的数学期望是基本矩阵$N = (E - Q)^{-1}$的第i行之和.

59.6.7 例子: 豌豆杂交

我们使用与AA基因型杂交的例子[1], 那么它就是一个吸收马尔可夫链. 再次写出转移矩阵

```
# 三种基因型与AA杂交的转移矩阵, Q就是右下4个值
AA=matrix(c(1,0.5,0,0,0.5,1,0,0,0),nc=3,
      dimnames=list(c("AA*AA","Aa*AA","aa*AA"),c("AA","Aa","aa")))
Q=AA[2:3,2:3]
```

```
> AA
      AA  Aa aa
AA*AA 1.0 0.0  0
Aa*AA 0.5 0.5  0
aa*AA 0.0 1.0  0
> Q
      Aa aa
Aa*AA 0.5  0
aa*AA 1.0  0
```

```
# 再次引用乘幂函数. 效率非常低的矩阵连乘函数!!!
mulprod<-function(X,n){
   tmp<-X
   if(n>=2){
   for (i in 2:n){tmp<-tmp%*%X }}
   tmp}
```

```
# 转移矩阵乘幂
> mulprod(AA,5)
         AA      Aa aa
AA*AA 1.00000 0.00000  0
Aa*AA 0.96875 0.03125  0
aa*AA 0.93750 0.06250  0
```

[1]例子描述见前面. 更多例子参考[8]

```
> mulprod(AA,10)
           AA          Aa aa
AA*AA 1.0000000 0.0000000000  0
Aa*AA 0.9990234 0.0009765625  0
aa*AA 0.9980469 0.0019531250  0
> mulprod(AA,100)
     AA          Aa aa
AA*AA  1 0.000000e+00  0
Aa*AA  1 7.888609e-31  0
aa*AA  1 1.577722e-30  0
> mulprod(AA,1000)
     AA           Aa aa
AA*AA  1  0.000000e+00  0
Aa*AA  1 9.332636e-302  0
aa*AA  1 1.866527e-301  0
> mulprod(AA,10000)
     AA Aa aa
AA*AA  1  0  0
Aa*AA  1  0  0
aa*AA  1  0  0

# Q乗 冪
> mulprod(Q,10)
              Aa aa
Aa*AA 0.0009765625  0
aa*AA 0.0019531250  0
> mulprod(Q,100)
              Aa aa
Aa*AA 7.888609e-31  0
aa*AA 1.577722e-30  0
> mulprod(Q,1000)
               Aa aa
Aa*AA 9.332636e-302  0
aa*AA 1.866527e-301  0
> mulprod(Q,10000)
     Aa aa
Aa*AA  0  0
aa*AA  0  0
```

下面计算基本矩阵N并分析之. 可以看到:

- 从状态Aa开始进入吸收状态AA在非吸收状态停留的总次数为2+0=2次, 即从Aa开始, 经过2步大多就纯化了.
- 从状态aa开始进入吸收状态AA在非吸收状态停留的总次数为2+1=3次, 即从aa开始, 经过3步大多就获得显性性状.

```
# 计算基本矩阵N
> E=diag(c(1,1)); E
     [,1] [,2]
[1,]    1    0
[2,]    0    1
# 基本矩阵.

> N=solve(E-Q); N
   Aa*AA aa*AA
Aa     2     0
aa     2     1
```

59.6.8 例子: 动物健康

我们写出转移矩阵[2]. 从基本矩阵可以看到

- 由good状态出发的寿命(即到dead)平均为$9100+100=9200$(天)
- 由ill状态出发的寿命平均为$9000+100=9100$(天)

```
P=matrix(c(1,0.,0.01,0,0.99,0.9,0,0.01,0.09),nc=3,
       dimnames=list(c("dead","good","ill"),c("dead","good","ill")))

# 基本矩阵
> N=solve(diag(c(1,1))-P[2:3,2:3]);N
     good ill
```

[2]例子描述见前面

```
good 9100 100
ill  9000 100
```

59.6.9 多个吸收状态

具有$r(r>1)$个吸收状态的吸收马尔可夫链, 从转移状态$i_1=r+i$开始, 最终进入第j个吸收状态的概率是矩阵$B=NR$的第i行第j列元素值.

下面是一个多吸收状态的例子. 其中

- "dead1" 其它死亡
- "dead2" 呼吸疾病死亡
- "dead3" 循环疾病死亡
- "good" 健康
- "ill1" 呼吸疾病
- "ill2" 循环疾病

```
P=matrix(c(1,0.,0, 0.001,0,0,
          0,1,0,0,0.2,0,
          0,0,1,0,0,0.1,
          0,0,0,0.889,0.7,0.8,
          0,0,0,0.01,0.1,0,
          0,0,0,0.1,0,0.1),nc=6,
       dimnames=list(
          c("dead1","dead2","dead3","good","ill1","ill2"),
          c("dead1","dead2","dead3","good","ill1","ill2")))
```

```
> R=P[4:6,1:3];R
    dead1 dead2 dead3
good 0.001   0.0   0.0
ill1 0.000   0.2   0.0
ill2 0.000   0.0   0.1
```

```
> Q=P[4:6,4:6];Q
     good ill1 ill2
good 0.889 0.01  0.1
ill1 0.700 0.10  0.0
ill2 0.800 0.00  0.1
> E=diag(c(1,1,1))
> N=solve(E-Q); N
        good     ill1     ill2
good 69.76744 0.7751938 7.751938
ill1 54.26357 1.7140396 6.029285
ill2 62.01550 0.6890612 8.001723
> B=N%*%R; B
         dead1     dead2     dead3
good 0.06976744 0.1550388 0.7751938
ill1 0.05426357 0.3428079 0.6029285
ill2 0.06201550 0.1378122 0.8001723
```

对于B的分析表明, 从健康开始, 疾病3的死亡概率最大. (应该引起谁的注意?)

对疾病矩阵N的分析表明, 从健康开始, 其寿命期望为$69.77 + 0.78 + 7.75 = 78.29$岁

若考虑生育增长, 则需要带输入的马尔可夫链

59.7 带输入的马尔可夫链

59.7.1 水塘氮循环的例子

考虑一个水塘, 鱼吃水藻, 水藻从水中吸收氮, 鱼排泄与水藻生长中排除部分氮. 鱼可能捕捞卖掉, 水藻可能溢出池塘(或打捞). 那么此系统的氮的转移矩阵为

```
 \begin{verbatim}
P=matrix(c(1,0,0, 0,0.75,
```

```
          0,1,0,0.2,0,
          0,0,0.5,0.1,0.125,
          0,0,0.5,0.2,0,
          0,0,0,0.5,0.125),nc=5,
       dimnames=list(
          c("catch","out","water","plant","fish"),
          c("catch","out","water","plant","fish")))
```

```
> P
     catch out water plant  fish
catch  1.00 0.0 0.000   0.0 0.000
out    0.00 1.0 0.000   0.0 0.000
water  0.00 0.0 0.500   0.5 0.000
plant  0.00 0.2 0.100   0.2 0.500
fish   0.75 0.0 0.125   0.0 0.125
```

每单位时间在转移状态上补充氮肥, 设每年向水中投入有效氮肥80kg, 则输入向量为 $F = [80, 0, 0]$

我们有如下定理

59.7.2 定理: 转移向量的极限

若带有输入的马尔可夫链, 输入向量为F, 则其状态向量序列的转移部分存在极限FN, 其中N为转移矩阵的基本矩阵

```
> Q=P[3:5,3:5]; Q
     water plant  fish
water 0.500   0.5 0.000
plant 0.100   0.2 0.500
fish  0.125   0.0 0.125
> E=diag(c(1,1,1))
> N=solve(E-Q); N # 基本矩阵, 转入吸收前停留的时间
        water    plant     fish
water 2.5454545 1.5909091 0.9090909
plant 0.5454545 1.5909091 0.9090909
```

```
fish  0.3636364 0.2272727 1.2727273
> R=P[3:5,1:2]
> B=N%*%R; B # 最终三个转移状态转入吸收状态的概率
        catch       out
water 0.6818182 0.31818182
plant 0.6818182 0.31818182
fish  0.9545455 0.04545455
> F=c(80,0,0)
> F%*%N # 最终三个状态稳定的氮含量
      water   plant    fish
[1,] 203.6364 127.2727 72.72727

# 若每条鱼的含氮量为0.02kg, 则每年投入80kg的有效氮最多能够养的鱼为3636条
> (F%*%N)[3]/0.02
[1] 3636.364
```

Chapter 60

z-curve

60.1 解释

DNA 碱基由 ATGC 四种构成, 若某 DNA 序列有 N 个碱基 A_n, C_n, G_n, T_n 分别为从序列开始计算到第n个碱基时的 ATGC 的个数. 实际上, 此序列由 A_n, C_n, G_n, T_n 唯一确定. 下面将之映射到三维空间

$$\begin{aligned} x_n &= (A_n + G_n) - (C_n + T_n) \equiv R_n - Y_n \\ y_n &= (A_n + C_n) - (G_n + T_n) \equiv M_n - K_n \\ z_n &= (A_n + T_n) - (C_n + G_n) \equiv W_n - S_n \end{aligned}$$

其中, $A_0 = C_0 = G_0 = T_0 = 0$, 因此 $x_0 = y_0 = z_0 = 0$.

可以证明, A_n, C_n, G_n, T_n 与 x_n, y_n, z_n 互相唯一确定.

下面将 z_n 对 $n = 0, 1, 2, \cdots, N$ 做线性回归

$$z \sim kn$$

其中 k 为回归的斜率系数. 令

$$z'_n = z_n - k * n$$

将 $z'_n \sim n$ 作曲线图, 就得到所谓的 z-curve (z 曲线). z 曲线在基因分析中起关键作用.

下面是计算 z 曲线的函数. 输入为 DNA 序列, 输出 x,y,z 为坐标, a,t,g,c 为 A_n, C_n, G_n, T_n 序列. z1 为 z' 值.

```
zcurve<-function(s){
   a=cumsum(s=="a"|s=="A")
   t=cumsum(s=="t"|s=="T")
   g=cumsum(s=="g"|s=="G")
   c=cumsum(s=="c"|s=="C")
   a=append(0,a)
   t=append(0,t)
   g=append(0,g)
   c=append(0,c)
   x=a+g-(c+t)
   y=a+c-(g+t)
   z=a+t-(c+g)
   n=0:(length(z)-1)
   lm.z = lm(z~n-1)
   k=lm.z$coefficients[1]
   z1=z-k*n
   r <-list(x=x,y=y,z=z,z1=z1,a=a,t=t,g=g,c=c)
   r
}

> s='GCTTCTAGCCTGACATATTAACTCCTG'
> s<-strsplit(s,"")
> s=s[[1]]
> s
 [1] "G" "C" "T" "T" "C" "T" "A" "G" "C" "C" "T" "G" "A" "C" "A" "T" "A" "T" "T"
[20] "A" "A" "C" "T" "C" "C" "T" "G"
> r<-zcurve(s); r
$x
 [1]  0  1  0 -1 -2 -3 -4 -3 -2 -3 -4 -5 -4 -3 -4 -3 -4 -3 -4 -5 -4 -3 -4 -5 -6
[26] -7 -8 -7

$y
 [1]  0 -1  0 -1 -2 -1 -2 -1 -2 -1  0 -1 -2 -1  0  1  0  1  0 -1  0  1  2  1  2
[26]  3  2  1
```

```
$z
 [1]  0 -1 -2 -1  0 -1  0  1  0 -1 -2 -1 -2 -1 -2 -1  0  1  2  3  4  5  4  5  4
[26]  3  4  3

$z1
 [1]  0.0000000 -1.1050505 -2.2101010 -1.3151515 -0.4202020 -1.5252525
 [7] -0.6303030  0.2646465 -0.8404040 -1.9454545 -3.0505051 -2.1555556
[13] -3.2606061 -2.3656566 -3.4707071 -2.5757576 -1.6808081 -0.7858586
[19]  0.1090909  1.0040404  1.8989899  2.7939394  1.6888889  2.5838384
[25]  1.4787879  0.3737374  1.2686869  0.1636364

$a
 [1] 0 0 0 0 0 0 0 1 1 1 1 1 1 2 2 3 3 4 4 4 5 6 6 6 6 6 6 6

$t
 [1] 0 0 0 1 2 2 3 3 3 3 3 4 4 4 4 4 5 5 6 7 7 7 7 8 8 8 9 9

$g
 [1] 0 1 1 1 1 1 1 1 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4

$c
 [1] 0 0 1 1 1 2 2 2 2 3 4 4 4 4 5 5 5 5 5 5 5 6 6 7 8 8 8

# 查 看 z 曲 线
> n=0:(length(r$z1)-1)
> plot(r$z1~n,col='red',type='l')
```

Part IX

附录A-概率统计基础理论

“概率统计基础”是理论部分，不涉及R的使用。实际上是我的生物统计学的部分讲课大纲，仅供参考。参考文献除了[11]，还参考了《初等概率论附随机过程》[4]和复旦大学的《概率论第一册概率论基础》等。

这部分只写到估计和假设检验，其它部分的理论随各个命题一起讨论。

Chapter 61

条件概率与统计独立性

61.1 条件概率

61.1.1 定义

我们有时会碰到下面的情况, 第二种情况就是条件概率

- 求A事件发生的概率
- 知道B已经发生, 求A事件发生的概率

条件概率设(Ω, F, P)是一个概率空间, $B \in F$, 且$P(B) > 0$, 则对任意$A \in F$, 记

$$P(A|B) = \frac{P(AB)}{P(B)}$$

并称$P(A|B)$为在事件B发生的条件下事件A发生的条件概率

考虑两个孩子的家庭. 事件A: 随机选取的家庭有一个男孩和一个女孩这一事件, 则$P(A) = 1/2$, 若我们事先知道这个家庭至少有一个女孩, 那么上述事件的概率是多少?

甲乙两市都位于长江下游. 根据100多年的记录, 知道一年中雨天的比例甲市占20%, 乙市占18%, 两地同时下雨占12%。求甲市下雨时乙市下雨的概率和乙市下雨时甲市下雨的概率.

61.1.2 性质

三个基本性质

条件概率具有概率的三个基本性质

- $P(A|B) \geq 0$ 非负性
- $P(\Omega|B) = 1$ 规范性
- $P(\sum_{i=1}^{\infty} A_i|B) = \sum_{i=1}^{\infty} P(A_i|B)$ 完全可加性

导出性质

- $P(\Phi|B) = 0$
- $P(A|B) = 1 - P(\bar{A}|B)$
- $P(A_1 + A_2|B) = P(A_1|B) + P(A_2|B) - P(A_1A_2|B)$

当$B = \Omega$时,条件概率化为无条件概率, 故可以把一般的概率看作条件概率.

试证明导出性质的第3条

乘法定律的推广

$$P(A_1A_2 \cdots A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1A_2) \cdots P(A_n|A_1A_2 \cdots A_{n-1})$$

Polya 的 坛 子 [1] 在 一 只 坛 子 里 有b只 黑 球 和r只 红 球. 随 机 取 出 一 只, 把 原 球 放 回, 并 加 入 与 抽 出 的 球 相 同 眼 色 的 球c只. 再 摸 第 二 次. 这 样 下 去 共 摸 了n次. 问 前 面 的m次 出 现 黑 球, 后 面 的n-m次 出 现 红 球 的 概 率 是 多 少?(这 个 模 型 曾 经 被 用 来 描 述 传 染 病 模 型)

在Polya的 坛 子 模 型 中, 抽 得 前3个 球 依 次 为(黑,黑,红),(黑,红,黑),(红,黑,黑)的 概 率 是 多 少?进 一 步, 3个 球 中 有2个 是 黑 球 的 概 率 是 多 少?n个 球 中 有k个 黑 球 的 概 率?

俄 罗 斯 轮 盘 赌 的 时 候, 很 多 老 练 的 赌 徒 相 信"若 红 已 经 连 续 多 次 出 现, 则 在 下 一 次 旋 转 中 把 赌 本 压 在 黑 上 是 明 智 的", 你 怎 么 看?

61.2 全概率公式

设$A_1, ..., A_n$为 一 般 空 间 的 一 个 分 割, 则

$$B = \sum_{i=1}^{\infty} A_i B$$

由 完 全 可 加 性 和 乘 法 定 理 得

$$P(B) = \sum_{i=1}^{\infty} P(A_i) P(B|A_i)$$

播 种 用 的 一 等 小 麦 种 子 混 合 有2%的 二 等 种 子, 1.5%的 三 等 种 子, 1%的 四 等 种 子. 一 二 三 四 等 种 子 长 出 的 穗 含50个 以 上 麦 粒 的 概 率 分 别 为0.5,0.15,0.1, 0.05. 求 这 批 种 子 长 出 的 穗 含50个 以 上 麦 粒 的 概 率.

[1] Polya, 斯 坦 福 名 誉 教 授, 20世 纪 最 著 名 的 分 析 家 之 一. 写 有 多 部 科 普 著 作. 推 荐 《 数 学 与 猜 想 》

61.3 Bayes公式

若事件B能且只能与两两互不相容的事件$A_1, ..., A_n$同时发生, 即

$$B = \sum_{i=1}^{\infty} BA_i$$

由于

$$P(A_i|B) = P(B)P(A_i|B) = P(A_i)P(B|A_i)$$

故

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{\sum_{i=1}^{\infty} P(A_i)P(B|A_i)}$$

再利用全概率公式即得

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{P(B)}$$

此公式称Bayes公式, 其中

- $P(A_i)$称先验概率
- $P(A_i|B)$称后验概率
- $P(B|A_i)$称条件概率/关于B的似然函数

此式表明了获取信息B后对A_i的新认识. 具有完整的理论依据. 对Bayes公式的应用可以参考《模式分类》

在数字通信中, 由于存在随机干扰, 因此接收到的信号与发出的信号可能不同。为确定发出的信号通常需要计算各种概率。下面是一个简单的模型–二进信道。若发报机以0.7和0.3 的概率发出信号0和1（譬如分别用低电平和高电平表

示）。由于随机干扰, 当发出信号0时, 接收机不一定收到0, 而以概率0.8和0.2收到信号0和1; 同样, 当发出信号1时, 接收机以概率0.9和0.1收到信号1和0。求当接收到信号0时发报机发出信号0的概率。

罐头厂想通过颜色自动识别鱼的种类, 例如鲤鱼和草鱼.设事件A_1为出现鲤鱼, A_2为出现草鱼. B_1为出现黑色, A_2为出现白色. 通过往年的经验(先验概率)知道$P(A_1) = 0.6$,$P(A_2) = 0.4$, 通过分析知道(条件概率), $P(B_1|A_1) = 0.2$,$P(B_2|A_1) = 0.8$, $P(B_1|A_2) = 0.1$,$P(B_2|A_2) = 0.9$. 那么怎样通过颜色来识别鱼的种类呢?

61.4 事件独立性

61.4.1 让我们来"创造"概率测度

仅对任意可数空间$\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$, 下面我们将要看到建立概率测度是多么容易. 分两步

1. 对每个样本点ω_i, 我们指定一个任意的"权值"P_i, 使之满足

$$P_i \geq 0; \sum P_i = 1, i = 1, 2, ...n$$

2. 现在,对Ω中任意子集$A \subset \Omega$, 定义 A的概率=A中全体点上的权值的和, 即

$$P(\omega_i) = P_i$$

那么

$$P(A) = \sum_{\omega_i \in A} P_i$$

只要我们验证满足公理即可(这很容易)

可见, 我们可以得到大批的概率测度

实际上, 通过这种方法, 我们可以得到概率测度的全体(但是仅仅对可数情况而言, 对于不可数情况,需要用测度来定义)

以上构造中有一种非常特殊的情况, 即Ω只含有有限个点, 每个点给以相同的权值, 即

$$P_i = \frac{1}{n}$$

这样,我们回到了等可能的情况.(想一下, Ω为可数无穷时, 权值能不能相等? 为什么?)

61.4.2 重复独立试验

我们用一个例子来导出事件独立的概念...

对事件A与B, 若

$$P(AB) = P(A)P(B)$$

则称它们是统计独立的, 简称独立

三个事件统计独立, 若下面四个式子同时成立

$$P(AB) = P(A)P(B) \tag{61.1}$$
$$P(AC) = P(A)P(C) \tag{61.2}$$
$$P(BC) = P(B)P(C) \tag{61.3}$$
$$P(ABC) = P(A)P(B)P(C) \tag{61.4}$$

一个问题. 若$P(AB) = P(A)P(B), P(AC) = P(A)P(C), P(BC) = P(B)P(C)$, 是否有$P(ABC) = P(A)P(B)P(C)$?

n个事件独立你能写出满足的条件吗?写之前估计一下, 共需要多少式子?

几个推论(试着证明一下)

1. 若A, B独立,且$P(B) > 0$, 则

$$P(A|B) = P(A)$$

2. 若A, B独立, 则下列各对事件也相互独立: $(\bar{A}, B)$, $(A, \bar{B})$, $(\bar{A}, \bar{B})$

61.4.3 独立性与概率计算

先介绍一个常用的公式.

如果A_1, A_2相互独立, 则由于$\overline{A_1 \cup A_2} = \bar{A}_1\bar{A}_2$, 所以有

$$P(A_1 \cup A_2) = 1 - P(\bar{A}_1)P(\bar{A}_2)$$

(想一下如果直接计算3个并集的概率会怎么样? n个呢?)

若每个人血清携带某病毒的概率为0.4%, 则混合100个人的血清, 求含病毒的概率?

可靠性理论...

Chapter 62

随机变量的分布和数字特征

62.1 随机变量

62.1.1 定义

在定义域Ω上ω的一个数值函数X

$$\omega \longrightarrow X(\omega)$$

称为Ω上的一个随机变量

62.1.2 随机在哪里

随机是指对ω选择的随机性.

一旦ω确定, X的值也确定了.

62.1.3 让我们来构造随机变量

抛一枚硬币, 试着构造出现结果的随机变量

抛两枚硬币, 试着构造出现结果的随机变量

设Ω是一个包含n个人的母体, 可以记为$\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$. 每个人有很多特征. 试着构造他们年龄, 体重, 收入的随机变量

设Ω是容器内分子的全体. 试着构造向左运动的分子的随机变量.

若X, Y是随机变量, 则X+Y, X-Y, X*Y, X/Y 也是随机变量.

若 φ 是两个变量的函数, 且X, Y是随机变量, 那么

$$\omega \longrightarrow \varphi(X(\omega), Y(\omega))$$

也是随机变量. 当不引起误会的时候,也可以简写为$\varphi(X, Y)$.

[注: 上一个定理包含在这个定理内]

62.2 分布

62.2.1 分布列

样本空间为可数有穷时, 我们可以把每个基本事件的概率一一列出, 称为分布列.

连续形式时, 设A为实数的某个子集, 例如 $A = [a, b]$. 那么, $P(\{\omega | X(\omega \in A)\}) = P(\{X \in A\}) = P(\{a <= X <= b\})$

当A缩为一点x时, 我们得到一个重要情况 $A = \{x\}$ 为单点集. 此时, $P(X = x) = P(X \in \{x\})$ 为基本事件的概率.

62.2.2 分布函数

当$A = (-\infty, x]$时, 引入一个记号

$$F_X(x) = P(X <= x)$$

称X的分布函数(就是把所有小于等于x的X值的概率捡出来相加), 又称"累积分布函数"

离散形式:

$$F_X(x) = \sum_{X<=x} P(X) = P(X <= x)$$

连续形式:

$$F_X(x) = \int_{-\infty}^{x} P(u)du$$

例如$F_X(18)$就是小于等于18岁的人的集合的概率

62.2.3 累积分布图

F_X对X作图就是累积分布图

62.3 期望

期望又称数学期望, 均值

62.3.1 离散情况

对于可数样本空间上的一个随机变量X, 定义其数学期望为

$$E(X) = \sum_{\omega \in \Omega} X(\omega)P(\{\omega\})$$

若X取值$x_1, x_2...$, 其概率分别为$p_1, p_2...$

$$E(X) = \sum_{i=1}^{\infty} x_i p_i$$

- 当$E(X)$绝对收敛, 称为数学期望(直观上是合理的, 因为顺序对期望并不是本质的)
- 当$E(X)$发散, 称数学期望不存在

期望可以解释为对X的加权平均

甲乙二射手成绩如下,问平均起来二人谁枪法好?

	甲			乙		
成绩	8	9	10	8	9	10
概率	0.3	0.1	0.6	0.2	0.5	0.3

62.3.2 连续情况

数学期望定义为

$$E(X) = \int_{-\infty}^{\infty} xp(x)dx$$

一块土地分为4块, 面积与价格如下, 问平均价格是多少?

面积	30%	15%	35%	20%
价格	5	10	10	30

62.3.3 一些定理

若X和Y是可和的, 则$X+Y$也是可和的, 并且有

$$E(X+Y)=E(X)+E(Y)$$

某彩票有100张, 其中1张奖金10000元, 其余皆为零. 如果我买一张彩票, 我的期望收益是多少? 两张呢? 再进一步, 如果有两种这样的彩票, 那么与其从同一种彩票买两张, 我不如从两种中各买一张, 那么我可能有机会赢得20000元, 这样做是不是对我更加有利?

一个袋子中有N张不同的票券, 我们以有放回的方式一张一张的抽取. 假设我们想收集r张不同的票券, 要期望抽取多少次才能得到它们? (这个问题同随机投弹打击目标类似)

(庞加来公式) 对任意事件$A_1, A_2, \cdots, A_n$, 我们有

$$P(\cup_{i=1}^{n} A_j)=\sum_j P(A_j)-\sum_{j,k} P(A_j,A_k)+\sum_{j,k,l} P(A_j,A_k,A_l)-\cdots+(-1)^{n-1}P(A_1,\cdots,A_n)$$

其中各下标是不同的且从1变到n.

(匹配问题) 两套各标记上从1到n的卡片被随机的匹配, 问至少出现一个匹配成对的概率是多少? 匹配成对的期望数是多少?

62.4 方差和协方差

如果X和Y是相互独立的可和随机变量, 则

$$E(XY)=E(X)E(Y)$$

对离散情况有

$$E(XY)=\sum_j \sum_k x_j y_k P(A_{jk})$$

对连续情况有

$$E(XY) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} uvf(u,v)dudv$$

62.4.1 方差

若$E(X-E(X))^2$存在, 则称之为X的方差, 并记为

$$D(X) = E(X-E(X))^2$$

根据期望的定义, $D(X) = E(X^2) - (E(X))^2$. [你能推导出来吗?试一下!]

标准差称$\sqrt{D(X)}$为根方差或标准差

62.4.2 方差的性质

1. $D(X) = 0 \Longleftrightarrow p(X = C) = 1$ 即X为常数
2. $D(aX) = a^2D(X)$
3. 若$a \neq E(X)$则$E(X-a)^2 > D(X) = E(X-E(X))^2$
4. 若X和Y相互独立并且都有有限方差,则

$$D(X+Y) = D(X) + D(Y)$$

试着证明一下性质3, 不难

(柯西-施瓦茨不等式)

$$E(XY)^2 <= E(X^2)E(Y^2)$$

(这个不等式的证明方法有很多, 你能想出一个吗?)

62.4.3 把随机变量标准化

记

$$X' = \frac{X - E(X)}{\sqrt{D(X)}}$$

为标准化的随机变量,显然

$$E(X') = 0, D(X') = 1$$

这就是标准化的理由

62.4.4 协方差与相关系数

若X和Y都有有限方差,则

$$E[(X - E(X))(Y - E(Y))] = E(XY) - E(X)E(Y)$$

称为X和Y的协方差, 记为$cov(X, Y)$. 量

$$\rho(XY) = \frac{cov(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

称为相关系数.

Chapter 63

怎样描述数据

63.1 原始数据

63.1.1 收集

- 调查
  - 普查
  - 抽样调查
- 试验

63.1.2 分类

- 数量性状数据对排序有自然的意义
  - 离散变量小数部分往往无意义
  - 连续变量往往是实际情况的近似

  注: 实际测得的数据都是有限值,但离散和连续在本质上是不同的
- 质量性状(属性)数据例如: 颜色深浅, 甜味的浓淡, 叶子的种类等. 分析的时候把它们映射为数值. 排序一般无意义, 除非我们能够指定意义

63.2 位置测度

63.2.1 算术平均数(arithmetic mean)

所有观察值的和除以观察的个数.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

缺点: 对极端值敏感

63.2.2 样本中位数(sample median)

简称中位数(median). 先排序才行.

- 如果n为奇数, 则第$(n+1)/2$个最大观察值就是样本中位数
- 如果n为偶数, 则第$n/2$个最大观察值与$n/2+1$个最大观察值的平均数就是样本中位数

缺点: 对中位值以外的值不敏感

对称性: 利用平均数和中位数可以判断样本分布的对称性(你能看出来吗?)

63.2.3 众数

频数(frequency)与频数表

参考《生物统计学》–董时富编 28-29页

众数(mode)

在一个样本的所有观察值中, 发生频率最大的那个值称为样本的众数

按照众数个数分类,

- 只有一个众数的分布称单峰分布;
- 有两个众数的分布称为双峰分布;
- 有三个众数的分布称为三峰分布
- 没有众数

63.2.4 几何平均(geometric mean)

国外的定义:

$\overline{logx}$的反对数称为几何平均, 这里

$$\overline{logx} = \frac{1}{n}\sum_{i=1}^{n}\log x_i$$

国内的定义:

$$(x_1 x_2 \cdots x_n)^{1/n}$$

63.3 算术平均数的某些性质

63.3.1 改变数据的起点

如果$y_i = x_i + c, i = 1, 2, \cdots, n$,则$\bar{y} = \bar{x} + c$

63.3.2 数据伸缩

如果$y_i = cx_i, i = 1, 2, \cdots, n$,则$\bar{y} = c\bar{x}$

63.3.3 伸缩+改变起点

如果$y_i = c_1 x_i + c_2, i = 1, 2, \cdots, n$,则$\bar{y} = c_1\bar{x} + c_2$

63.4 离散性测度

63.4.1 极差(range)

一个样本中最大与最小观察值之间的差异称为极差

63.4.2 分位数(quantiles)或百分位数

第p个百分位数定义如下：

- 如果$np/100$不是一个整数, 而k是小于$np/100$的最大整数, 则第$k+1$个最大样本点即是第p个百分位数
- 如果$np/100$是一个整数, 则第$np/100$与$np/100+1$个大的观察值的平均值定义为第p个百分位数

63.4.3 偏差

- 偏差
- 绝对偏差

63.4.4 方差与标准差

63.4.4.1 偏差

一个样本中每个观察值与样本平均值偏差的总和永远为零

$$d = \frac{\sum_{i=1}^{n}(x_i - \bar{x})}{n}$$

63.4.4.2 平均偏差

$$\sum_{i=1}^{n} |x_i - \bar{x}|/n$$

63.4.4.3 样本方差(variance)

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

63.4.4.4 样本标准差(standard deviation)

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{s^2}$$

63.5 方差与标准差的某些性质

- 假设有两样本$x_1, \cdots, x_n$及$y_1, \cdots, y_n$, 此处$y_i = x_i + c, i = 1, \cdots, n$. 若样本方差分别记为$s_x^2$及$s_y^2$, 则

$$s_x^2 = s_y^2$$

- 假设有两样本$x_1, \cdots, x_n$及$y_1, \cdots, y_n$, 此处$y_i = cx_i, i = 1, \cdots, n, c > 0$. 若样本方差分别记为$s_x^2$及$s_y^2$, 则

$$s_y^2 = c^2 s_x^2, \qquad s_y = cs_x$$

63.6 变异系数(coefficient variation, CV)

$$100\% * \frac{s}{\bar{x}}$$

63.7 数据的分组

频数分布是按数值大小有序地显示数据中的每个值及出现的频数. 所谓频数即是指数值在数据中出现的次数

63.8 图示法

63.8.1 条形图(bar graph)

63.8.2 直方图(histogram)

63.8.3 茎叶图(stem-and-leaf plot)

63.8.4 盒型图(box plot)

- 异常值(outlying value) 一个观察值x如果属于下面之一, 则为异常值
  - $x >$上百分位数+1.5×(上百分位数-下百分位数)
  - $x <$下百分位数-1.5×(上百分位数-下百分位数)
- 极端异常值(extreme outlying value) 一个观察值x如果属于下面之一, 则为极端异常值
  - $x >$上百分位数+3×(上百分位数-下百分位数)
  - $x <$下百分位数-3×(上百分位数-下百分位数)

63.9 偏斜度与峭度

63.9.1 偏斜度(skewness)

度量数据围绕众数呈不对称的程度, 为标准化了的三阶中心矩, 记为

$$g_1 = \frac{m_3}{m_2^{3/2}}$$

其中m_3为三阶中心矩(third central moment)

$$m_3 = \frac{\sum(x-\bar{x})^3}{n}$$

其中m_2为二阶中心矩(third central moment)

$$m_2 = \frac{\sum(x-\bar{x})^2}{n}$$

类似,三阶原点矩记为

$$m_3' = \frac{\sum x^3}{n}$$

二阶原点矩记为

$$m_2' = \frac{\sum x^2}{n}$$

一阶原点矩记为

$$m_1' = \bar{x}$$

一阶中心矩记为

$$m_1 = \frac{\sum(x-\bar{x})}{n} = 0$$

63.9.2 峭度(kurtosis)

表数据陡峭或平坦的程度, 记为

$$g_2 = \frac{m_4}{m_2^2} - 3$$

Chapter 64

离散分布

64.1 退化分布(单点分布)

当随机变量只取常数值时, 即

$$P(X(\omega) = c) \equiv 1$$

为退化分布, 其分布函数为

$$F(X) = \begin{cases} 0, & x < c \\ 1, & x >= c \end{cases} \tag{64.1}$$

或

$$F(X - c) = \begin{cases} 0, & x <= c \\ 1, & x > c \end{cases} \tag{64.2}$$

期望

$$E(X) = \sum_{\infty}^{\infty} xp(x) = c$$

方差

$$D(X) = E(X^2) - (E(X)^2 = c^2 - c^2 = 0$$

64.2 贝努里分布(两点分布)

一次试验中只有两个结果$\Omega = \{A, \bar{A}\}$, 这种试验称为贝努里试验. 其中

$$P(A) = p, \qquad P(B) = 1 - p = q$$

记X为事件A出现的次数, 则

$$X = \begin{cases} 0, & X\text{不出现} \\ 1, & X\text{出现} \end{cases} \tag{64.3}$$

概率取值为

$$\begin{cases} P(X = 1) = q \\ P(X = 0) = p \end{cases} \tag{64.4}$$

那么我们有

$$P(X) = \begin{cases} p, & X = 1 \\ q, & X = 0 \\ 0. & X = \text{其它} \end{cases} \tag{64.5}$$

期望

$$E(X) = 0 * q + 1 * p = p$$

方差

$$D(X) = E(X^2) - (E(X)^2 = p - p^2 = pq$$

(考虑一下X的取值变为A出现为2, 否则为0, 期望和方差会是什么? [1])

[1]答案为$E(X) = 2p, \quad E(X^2) = 4pq$

下面考虑贝努里分布的母函数

$$g(z) = qz^0 + pz^1 = q + pz$$

那么期望和方差可以由下面得到

$$\begin{aligned} E(X) &= g'(1) = p \\ E(X^2) &= g''(1) + g'(1) = p \\ D(X) &= p - p^2 = pq \end{aligned}$$

64.3 二项分布

在n重贝努里试验中, 记k为A出现的次数, 则k的取值为$0, 1, 2, \cdots, n$.

记A_i为第i次试验中出现事件A, $\overline{A_i}$为第i次试验中A不出现. 若记B_k为n重贝努里试验中, A出现k次这一事件, 则

$$B_k = (A_1 \cdots A_k \overline{A_{k+1}} \cdots \overline{A_n}) + (\cdots) + (\overline{A_1} \cdots \overline{A_{n-k}} A_{n-k+1} \cdots \overline{A_n})$$

右边一共有$\binom{n}{k}$项, 且两两互不相容. 由独立性得出

$$P(A_1 \cdots A_k \overline{A_{k+1}} \cdots \overline{A_n}) = P(A_1) \cdots P(A_k) P(\overline{A_{k+1}}) \cdots P(\overline{A_n})$$

利用概率的加法定理得

$$P(B_k) = \binom{n}{k} p^k q^{n-k}$$

我们常常把此概率记为

$$B(k; n, p) = \binom{n}{k} p^k q^{n-k}$$

期望 [2]

$$E(X) = \sum_{k=0}^{n} P(B_k) = np$$

方差 [3]

$$D(X) = E(X^2) - (E(X)^2 = p - p^2 = npq$$

下面考虑二项分布的母函数

$$g(z) = \sum_{k=0}^{n} P(X = k)z^k = (pz + q)^n$$

也可以这样考虑, 记$X = X_1 + X_2 + \cdots + X_n$, 其中$X_i$为第i次贝努里试验. 由于$X_i$相互独立, 则二项分布的母函数可以由$g(z) = q + pz$的n次方给出

$$g(z) = (pz + q)^n = \sum_{k=0}^{n} \binom{n}{k} q^{n-k} p^k z^k$$

由母函数的定义知, z^k的系数$\binom{n}{k} q^{n-k} p^k = P(X = k)$

那么期望和方差可以由下面得到

$$\begin{aligned} E(X) &= g'(1) = np \\ E(X^2) &= g''(1) + g'(1) = n^2p^2 - np^2 + np \\ D(X) &= npq \end{aligned}$$

[2]小提示: 可以直接计算, 也可以使用独立随机变量的和的期望等于期望的和的性质来计算. 后者更简单一点. 还有一种有点技巧但容易公式化的方法. 母函数的方法最简单

[3]小提示: 同期望一样, 也可以使用几种不同的方法

64.4 几何分布

在n重贝努里试验中, 设A的第一次出现是在第k次试验, 记此事件为W_k, 则

$$W_k = \overline{A_1 A_2} \cdots \overline{A_{k-1}} A_k$$

$$P(W_k) = P(\overline{A_1})P(\overline{A_2}) \cdots P(\overline{A_{k-1}} A_k) = q^{k-1}p$$

记

$$g(k;p) = q^{k-1}p, \quad k = 0, 1, 2, \cdots$$

$g(k;p)$是几何级数的一般项, 因此上式称为几何分布.

验证

$$\sum_{k=1}^{\infty} g(k;p) = \frac{1}{1-q}p = 1$$

期望[4]

$$E(X) = \sum_{k=1}^{\infty} kg(k;p) = \frac{1}{p}$$

而

$$E(X^2) = \sum_{k=1}^{\infty} k^2 g(k;p) = \frac{1+q}{p^2}$$

则方差

$$D(X) = \frac{q}{p^2}$$

[4]虽然有一点点复杂, 但是鼓励你尝试一下

母函数

$$g(z)=\sum_{k=0}^{n}P(X=k)z^k=\frac{pz}{1-qz}$$

期望

$$E(X)=g'(1)=\frac{1}{p}$$

$$g''(1)=\frac{2q}{p^2}$$

方差

$$D(X)=\frac{q}{p^2}$$

64.5 负二项分布(巴斯卡分布)

接着几何分布考虑, 若$T_1,T_2,\cdots,T_n$每个以几何分布的母函数为母函数(回忆一下母函数与分布函数互相唯一确定), 也就是每个都是几何分布(等待第一次成功的次数的随机变量).

记$S_n=T_1+T_2+\cdots+T_n$, 则S_n为第n次成功的等待时间(1次算一个单位时间的话).

我们先来推导两个式子.

第一个式子

$$\frac{1}{1-x}=1+x+x^2+\cdots$$

$$(\frac{1}{1-x})' = \frac{1}{(1-x)^2} = 1 + 2x + 3x^2 + \cdots$$

$$(\frac{1}{1-x})'' = \frac{2!}{(1-x)^3} = 2! + 32x + 43x^2 + \cdots$$

$$(\frac{1}{1-x})^{(n)} = \frac{(n-1)!}{(1-x)^n} = (n-1)! + \frac{n!}{1!}x + \frac{(n+1)!}{2!}x^2 + \cdots + \frac{(n+j-1)!}{j!}x^j + \cdots$$

两边同除以$(n-1)!$, 由归纳法得

$$\frac{1}{(1-x)^n} = \sum_{j=0}^{\infty} \binom{n-1+j}{j} x^j$$

第二个式子–负二项分布(牛顿二项分布的推广)

$$\begin{aligned}\binom{-n}{j} &= \tfrac{n(n+1)\cdots(n+j-1)}{j!}(-1)^j \\ &= \tfrac{(n-1+j)!}{j!(n-1)!}(-1)^j \\ &= \tbinom{n-1+j}{j}(-1)^j \\ &= \tbinom{n-1+j}{n-1}(-1)^j\end{aligned}$$

由这两个式子得

$$\frac{1}{(1-x)^n} = \sum_{j=0}^{\infty} \binom{-n}{j} (-1)^j x^j$$

下面我们来看S_n, 由于T_i相互独立, 则S_n的母函数由下式给出(把上式代入)

$$g(z)^n = (\frac{pz}{1-qz})^n = (pz)^n \sum_{j=0}^{\infty} \binom{-n}{j} (-1)^j (qz)^j = \sum_{j=0}^{\infty} \binom{n+j-1}{n-1} p^n q^j z^{n+j}$$

设$k = n + j$, 则

$$g(z)^n = \sum_{k=n}^{\infty} \binom{k-1}{n-1} p^n q^{k-n} z^k$$

根据母函数的定义, 第n次成功出现在第$n + j$次试验的概率为

$$P(S_n = n + j) = \binom{n+j-1}{n-1} p^n q^j$$

下面的等式也是成立的

$$P(S_n = n + j) = \binom{n+j-1}{j} p^n q^j = \binom{-n}{j} p^n (-q)^j$$

由上式给出的分布叫做负二项分布.

我们再来看

$$\frac{g(z)}{z} = \sum_{j=1}^{\infty} \frac{q^{j-1} p z^j}{z} = \sum_{k=0}^{\infty} q^k p z^k$$

观察z^k的系数为$T_i - 1$即第一次成功前失败的次数. 那么

$$(\frac{g(z)}{z})^n = (\frac{p}{1-qz})^n = \sum_{k=0}^{\infty} \binom{n+k-1}{k} p^n (qz)^k$$

就是$S_n - n$的母函数, 即第n次成功前失败的次数.

另外可以这样考虑, 若第n次成功发生在第$n+j$次试验, 当且仅当$n+j-1$次试验中成功$n-1$次, 失败j次, 且第$n+j$次成功, 故有

$$P(S_n = n+j) = \binom{n+j-1}{n-1} p^{n-1} q^j p = \binom{n+j-1}{j} p^n q^j = \binom{-n}{j} p^n (-q)^j$$

64.6 泊松分布

64.6.1 定义等

若随机变量X可以取一切非负整数值, 且

$$P(X=k) = \frac{a^k e^{-a}}{k!}, \qquad a > 0$$

则称X服从泊松分布

验证[5]

$$\sum_{k=0}^{\infty} P(X) = 1$$

期望[6]

$$E(X) = a$$

而

$$E(X^2) = a^2 + a$$

[5]您可以在大部分的教科书中找到
[6]同上

方差[7]

$$D(X) = a$$

母函数[8]

$$g(z) = \sum_{k=0}^{\infty} \frac{a^k e^{-a} z^k}{k!} = e^{a(z-1)}$$

$$g'(z) = ae^{a(z-1)}$$

$$g''(z) = a^2 e^{a(z-1)}$$

由此我们又一次可以方便的得到期望与方差

64.6.2 从二项分布到泊松分布

==有时间的话推导一下==

[7]同上
[8]同上

Chapter 65

连续分布

65.1 定义

随机变量取某个区间[a,b]或$(-\infty,\infty)$的一切值. 其分布函数F(X)绝对连续, 即存在可积的函数p(x)使

$$F(X)=\int_{-\infty}^{x}p(y)dy$$

称p(x)为X的密度函数

65.2 性质

由公理知

$$p(x)>=0$$

$$\int_{-\infty}^{\infty}p(x)dx=1$$

$$p(x) = F'(x)$$

由于$p(x)$与$F(X)$相互确定, 则若$p(x)$满足上式, 推出$F(X)$也是一个分布函数.

由$F(X)$的定义知

$$P(a < x <= b) = F(b) - F(a) = \int_a^b p(x)dx$$

下面看X等于定值的概率

$$P(x = c) <= \lim_{h \to 0} \int_c^{c+h} p(x)dx = 0$$

由于$P(x = a) >= 0$, 故

$$P(X = c) = 0$$

即连续型随机变量取个别值的概率为0.(概率为0的事件不一定不可能; 概率为1的事件不一定必然发生)

由于

$$p(x)\Delta x \approx \int_x^{x+\Delta x} p(y)dy = F(x + \Delta x) - F(x)$$

故$p(x)$反映了取x邻近值的概率的大小.

65.3 均匀分布

密度函数

$$p(x) = \begin{cases} 1/(b-a) & a <= x <= b \\ 0 & x < a \quad or \quad x > b \end{cases}$$

分布函数

$$F(x)=\int_{-\infty}^{x}p(y)dy=\begin{cases}0 & x<=a\\ (x-a)/(b-a) & a<x<=b\\ 1 & x>b\end{cases}$$

其它[1]

$$E(X)=\int_{-\infty}^{\infty}xp(x)dx=\frac{a+b}{2}$$

$$E(X^2)=\int_{-\infty}^{\infty}x^2p(x)dx=\frac{a^2+ab+b^2}{3}$$

$$D(X)=\frac{(b-a)^2}{12}$$

65.4 正态分布

推导泊松分布的时候总是感觉有点不太正常(还记得泊松分布的条件吗?), 而且还有计算二项分布值的广泛需要.例如: n=100, p=0.5, k=50时$B(k;n,p)$的值到底是多少?

下面我们将一步一步推导出正态分布的表达式(如果时间允许的话)

[1]在几乎任何概率论教科书上都可以找到推导, 并且它们很简单

65.4.1 Stirling 公式

Stirling 公式[2] 为阶乘的近似计算公式

$\chi(n) = (e/n)^n \sqrt{2\pi n} e^{\omega(n)} = n! \quad (\ 1/(12(n+1/2)) < \omega < 1/12n\ )$

65.4.2 从二项分布到正态分布

- 首先推导当$n \to \infty$时二项系数的值趋于0
- 其次证明当$n \to \infty$时, 对于固定的区间, 二项分布的概率值之和为0
- 再次设$0 < p < 1$, $q = 1 - p$, 且

$$x = \frac{k - np}{\sqrt{npq}} \qquad 0 <= k <= n$$

设A是一个任意而固定的正常数. 于是在满足$|x| <= A$的k的范围内, 我们有

$$\binom{n}{k} p^k q^{n-k} \approx \frac{1}{\sqrt{2\pi npq} e^{-x^2/2}}$$

且收敛是一致的.

- 再次 (棣莫佛-拉普拉斯定理)对任意两个常数a和b, 我们有

$$\lim_{n\to\infty} P(a < \frac{S_n - np}{\sqrt{npq}} <= b) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2} dx$$

65.4.3 定义

以下面的函数

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2} dx$$

[2]推导见《数学分析原理》第二卷第一分册 52页. 作者: 格.马.菲赫金哥尔茨. 译者: 丁寿田

作为分布函数的概率分布称做正态分布. 概率密度函数显然(?)就为

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

(好了, 我们该松一口气了, 我们要推导的最困难的公式终于出来了. 但是要知道, 对于分析它还只是很基础的. 有用的是技巧!)

下面来验证一下[3]

$$\int_{-\infty}^{\infty} p(x)dx = 1$$

设随机变量

$$X_j \sim N(\mu_j, \sigma_j^2) \quad j = 1, 2, \cdots, n$$

其中μ_i为均值, σ_j^2为方差. 则

$$X_1 + X_2 + \cdots + X_n \sim N(\sum_{j=1}^{n} \mu_j, \sum_{j=1}^{n} \sigma_j^2)$$

65.5 指数分布

65.5.1 定义

符合下述密度函数

$$p(x) = \begin{cases} ae^{-ax} & x >= \\ 0 & x < 0 \end{cases}$$

[3]其中的技巧很好. 在菲赫金哥尔茨的《数学分析原理》中至少提供了4中方法来得到这个非正常积分的结果

和分布函数

$$F(x)=\begin{cases}1-e^{-ax} & x>=\\ 0 & x<0\end{cases}$$

的分布称为指数分布.

65.5.2 性质

指数分布有类似几何分布的"无记忆性", 即

$$p(x>s+t|x>s)=\frac{p(x>s+t)}{p(x>s)}=\frac{e^{-a(s+t)}}{e^{-as}}=e^{-at}=p(x>t)$$

指数分布是唯一具有次性质的连续分布.

(可以这样理解, 已知寿命长于s年, 则再活t年的概率与年龄s无关.)

65.5.3 与泊松分布的关系

记X(t)为参数at的泊松分布(过程), 则

$$p(X(t)=k)=\frac{e^{-at}(at)^k}{k!}$$

当k=0时

$$p(X(t)=0)=e^{-at}\sim \text{指数分布}$$

65.6 Γ分布

若X(t)是服从参数为at的泊松分布(过程). 记τ_r为第r个跳跃发生的时刻(第r个例子到来的时刻). 则

$$\{\tau_r < t\} \Longleftrightarrow \{X(t) >= r\}$$

即第r个跳跃发生在时刻t之前, 也就是t时刻之前发生至少r次跳跃. 我们以F(x)记τ_r的分布函数, 则有

$$F(t) = p(\tau_r < t) = p(X(t) >= r) = 1 - \sum_{k=0}^{r-1} \frac{(at)^k e^{-at}}{k!}$$

那么[4]

$$p(t) = F'(t) = \frac{a^r t^{r-1} e^{-at}}{(r-1)!} = \frac{a^r t^{r-1} e^{-at}}{\Gamma(r)}$$

称

$$p(x) = \begin{cases} \frac{a^r x^{r-1} e^{-ax}}{\Gamma(r)} & x > 0 \\ 0 & x <= 0 \end{cases}$$

的分布为Γ-分布. 其中a¿0, r¿0为参数.

泊松过程的第r个跳跃发生的时刻服从Γ-分布.

r=1时, Γ-分布变为指数分布

r=正整数时, Γ-分布为r个服从指数分布的随机变量之和的分布, 与负二项分布类似.

[4]中间的推导只有一点点的烦琐. 鼓励大家推导出来以增加信心

Chapter 66

从总体中抽取样本的方法

66.1 总体与样本的关系

- 已知总体, 研究样本. 一般是演绎性的.
- 已知样本, 推断总体潜在或未知的分布性质, 一般是归纳性推论. 也就是说, 拟合数据可以有很多种概率模型, 原则上我们选取一个最好的.

  (我们的内容基本集中在此)

66.2 推断的方法

- 估计用样本数据去估计指定的总体参数
- 假设检验用样本数据去检验总体参数是否等于某个指定的值

66.3 抽样

66.3.1 随机数的产生方法

均匀分布随机数的产生方法

- 真正的随机数–物理方法 抛硬币, 摸球, 粒子发生器, 白噪声...
- 伪随机数 具有类似随机表现的函数–计算机产生
  - 平方取中法
  - 线性同余法
  - 乘同余法
  - 素数模乘同余

  (如何检验伪随机数发生器的效果?)

有人已经编制成随机数表(两种方法都可以), 方便使用.

其它连续分布随机数的产生方法

由均匀分布的随机数取反函数得到.[1]

66.3.2 抽样的方法

随机选择

随机分配

随机化临床试验

- 区组随机化 比较不同组(¿1)的处理效果而随机分组. 通常

[1]参考《概率论》第一册 148页(复旦大学编)

为了更有可比性, 每组相等, 但不是必须.

- 分层在某些临床研究中, 病人可以再分为子群(层), 他们是按照某种特征来划分的. 典型的分层特征有: 年龄, 性别, 临床表现等.

66.4 临床研究中的盲法

双盲是医学临床研究中的金标准

- 双盲医生和病人都不知道每个病人的处理
- 单盲病人不知道, 但医生知道
- 非盲医生和病人都知道

(盲法的问题: 实际操作比较困难, 有时候病人或医生很容易通过某种表现猜测得到处理的类型)

Chapter 67

估计

67.1 均值的估计

问题: 如何使用一组指定的随机样本去估计潜在的总体的均值?

67.1.1 点估计

抽样分布一个 $\bar{x}$ 是从参考总体中所有可能大小为 n 的样本中的一个样本计算出来的样本均值. $\bar{x}$ 的抽样分布是指大量 $\bar{x}$ 值的分布.

关键是: 我们得到的总是一个出现的样本, 而在抽样分布中, 需要我们考虑所有可能的含量为 n 的样本. 即在不同的时候(不同的人抽样)出现的样本是不同的.

设 $x_1, x_2, \cdots, x_n$ 为从具有均值为 μ 的同一个总体出去的一个随机样本, 定义 $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. 则所有 $\bar{x}$ 的均值为 $E(\bar{x}) = \mu$, 对所有分布成立.

无偏估计量一个参数 θ 的估计量是 $\hat{\theta}$, 如果$E(\hat{\theta}) = \theta$, 则称 $\hat{\theta}$ 是 θ 的无偏估计. 这意味着, 在大量重复的抽样中(样本量为 n), $\hat{\theta}$ 的平均值将会是 θ.

最小方差无偏估计 当x的潜在总体分布是正态时, 可以证明, $\bar{x}$ 的方差是最小的. 即 $\bar{x}$ 是 μ 的最小方差无偏估计.

67.1.2 均值的标准误

均值的标准误差 设 $x_1, x_2, \cdots, x_n$ 是从严格总体中抽得的一组随机样本, 总体的方差为 σ^2/n. 则在大小为n的样本的重复抽样中, 样本均数的集合总体中, 这个 $\bar{x}$ 集合的方差为 σ^2/n, 标准差为 $\sigma/\sqrt{n}$, 后者也常常称为均值的标准误差(standard error of mean(sem)), 或者称为标准误差(standard error).

均值标准误的计算(此处和以后用var(x)表示变量x的方差):

$$var(\bar{x}) = \frac{1}{n^2} var(\sum_{i=1}^{n} x_i) = \frac{1}{n^2} \sum_{i=1}^{n} var(x_i)$$

由定义知, $var(x_i) = \sigma^2$, 因此

$$var(\bar{x}) = \frac{1}{n^2}(\sigma^2 + \sigma^2 + \cdots + \sigma^2) = \frac{1}{n^2}(n\sigma^2) = \sigma^2/n$$

实际上, 总体方差常常未知. 后面会看到, σ^2 的合理估计是 s^2, 那么均值标准误差的估计量是 $s/\sqrt{n}$ –样本均值集合的标准差

(注意: 不是样本的标准差)

67.1.3 均值的区间估计

我们常常希望得到均值的严格似乎合理的区间估计. 下面的区间估计仅当未知分布是正态分布才是正确的. 若不是正态分布, 则只能近似成立.

若$\bar{x} \sim N(\mu, \sigma^2/n)$, 那么把 $\bar{x}$ 写为标准形式, 即

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

则 z 应该是标准正态分布. 当重复抽样时, 95% 的 z 值落入 -1.96 到 1.96 之间. 但是 σ 在实际中很少知道.

67.1.4 t 分布

当 σ 未知时, 合理的估计是用样本的标准差 s 估计 σ 而用代替后计算的 z 来构建置信区间. 问题是, 此时的 z 已经不是正态分布了. 此时的 z 的分布是 t 分布.[1]

这个分布的形状和 n 的关系很大. 即 t 分布并不是一个分布, 而是一组分布, 依赖于称为 "自由度(degree of freedom–简写为df)" 的一个量.

如果$x_1, \cdots, x_n \sim N(\mu, \sigma^2)$ 且彼此独立, 则 $\frac{\bar{x}-\mu}{s/\sqrt{n}}$ 的分布称为具有 (n-1) 自由度的 t 分布.

具有自由度 d 的 t 分布上的第 100*u 的百分位点记为 $t_{d,u}$, 即

$$P(t_d < t_{d,u}) = u$$

正态分布中均数的置信区间具未知方差的正态分布的均值 μ 的 $100\% * (1-\alpha)$ 置信区间 (confidence interval, CI) 可以写成

$$(\bar{x} - t_{n-1,1-\alpha/2} s/\sqrt{n}, \bar{x} + t_{n-1,1-\alpha/2} s/\sqrt{n})$$

正态分布中均数的置信区间(大样本) 具未知方差的正态分布的均值 μ 的 $100\% * (1-\alpha)$ 置信区间 (confidence interval, CI) 当 n ¿ 200 时, 可以写成

$$(\bar{x} - z_{1-\alpha/2} s/\sqrt{n}, \bar{x} + z_{1-\alpha/2} s/\sqrt{n})$$

[1]这个问题首先在1908年由统计学家 Willianm Gossett 解决. 在其职业生涯中, 他在爱尔兰的应该叫做 Guinness Brewery 的酿酒厂工作. 他为自己选了一个笔名 "Student", 于是 $\frac{\bar{x}-\mu}{s/\sqrt{n}}$ 的分布就常常称为学生氏 t 分布(Student's t distribution)

置信区间的含义: μ 是一个未知但固定的值, 但是不同的样本会有不同的均值及方差, 故有不同的边界. 所以对总体可以重复抽样然后构建出上述置信区间. 即在所有的构建出的置信区间中有 95% 含有未知的参数 μ.

影响置信区间长度的因素有 n, s, α, 因为置信区间由这三个量决定.

- n—-增加时区间长度减小
- s—-反映了分散性, 它增加时区间长度增加
- α—-希望增加置信度, 即减小 α, 则置信区间长度会增加.

67.2 方差的估计

67.2.1 点估计

设 $x_1, \cdots, x_n$ 是均值为 μ 方差为 σ^2 的某总体的一组样本. 在样本量为 n 的所有随机样本中, 样本方差 s^2 是 σ^2 的无偏估计. 即 $E(s^2) = \sigma^2$.

如果我们在总体中重复样本量为 n 的抽样, 计算每一组样本的方差 s^2, 则大量的样本方差 s^2 取平均, 它就是总体方差 σ^2. 此公式对任何分布有效.

这里再次写出样本方差的公式

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}^2)$$

如果用 n 代替 s^2 中的 n-1, 那么它将是对 σ^2 的一个偏低的估计. (大样本时(¿200)可以忽略).

67.2.2 卡方分布

要寻找方差的区间估计, 我们需要介绍一类新的分布–卡方分布(Chi-square distribution, χ^2)

设 $x_1, \cdots, x_n \sim N(0,1)$, 且彼此独立. 如果 $G = \sum_{i=1}^{n}$, 那么 G 称为自由度(df)为 n 的卡方分布. 这个分布常常被记为 χ_n^2. 同 t 分布一样, 卡方分布也是一组分布, 依赖于自由度 df. 但是它不是对称的分布, 只有正值且向右倾斜.

可以证明, χ_n^2 的期望是 n, 方差是 2n.

具有 n df 的χ_n^2 的第 u 个百分位点记为 $\chi_{n,u}^2$, 即

$$P(\chi_n^2 < \chi_{n,u}^2) = u$$

注意: 卡方分布是倾斜分布, 没有对称性. 即没有上下百分位点的对称关系.

67.2.3 区间估计

为求 σ^2 的估计, 我们需要知道 s^2 的分布. 设 $x_1, \cdots, x_n \sim N(\mu, \sigma^2)$, 则可以证明

$$s^2 \sim \frac{\sigma^2 \chi_{n-1}^2}{n-1}$$

回忆把 x 标准化后就服从标准正态分布, 从卡方分布的定义我们有

$$\sum z_i^2 = \sum \frac{(x_i - \mu)^2}{\sigma^2} \sim \chi_n^2$$

因为通常不知道 μ, 我们使用 $\bar{x}$ 代替, 损失一个自由度, 结果就有下面的关系式

$$\sum \frac{(x_i - \bar{x})^2}{\sigma^2} \sim \chi_{n-1}^2$$

再回忆 s^2 的定义, 简单处理后我们有

$$(n-1)s^2 = \sum (x_i - \bar{x})^2$$

代入上式有

$$\frac{(n-1)s^2}{\sigma^2} \sim \chi^2_{n-1}$$

由这个方程我们就有求得 σ^2 的 $100\% * (1-\alpha)$ 的置信区间. 实际上, 可以看出

$$P(\frac{\sigma^2 \chi^2_{n-1,1-\alpha/2}}{n-1} < s^2 < \frac{\sigma^2 \chi^2_{n-1,\alpha/2}}{n-1}) = 1-\alpha$$

把不等式分成两个, 然后移项分别求得 σ^2 的表达式, 联合起来, 我们得到

$$P(\frac{(n-1)s^2}{\chi^2_{n-1,1-\alpha/2}} < \sigma^2 < \frac{(n-1)s^2}{\chi^2_{n-1,\alpha/^2}}) = 1-\alpha$$

σ^2 的 $100\% * (1-\alpha)$ 置信区间为

$$[\frac{(n-1)s^2}{\chi^2_{n-1,\alpha/2}}, \frac{(n-1)s^2}{\chi^2_{n-1,1-\alpha/2}}]$$

67.3 二项分布的估计

67.3.1 参数 p 的点估计

记 x 是二项随机变量, 其参数为 n 及 p, p 的无偏估计为事件中的样本比例 $\hat{p}$, 标准误差 $\sqrt{pq/n}$ 的精确估计为 $\sqrt{\hat{p}\hat{q}/n}$.

67.3.2 区间估计

67.3.2.1 正态近似法

回忆参数为 n 及 p, p 的二项分布近似服从正态分布 $N(np, npq)$, 若样本中发生的事件数为X, 则对应的比例 $\hat{p} = X/n$ 也是正态分布, 且参数分别为 p 和 pq/n. 即

$$\hat{p} \sim N(p, pq/n)$$

若在两边乘以 n, 则 n 次贝努里使用中成功的次数 $X = n\hat{p}$, 则有下式, 实际上与二项分布的正态近似相同

$$X \sim N(np, npq)$$

类似于样本均值的区间估计, 二项分布中参数 p 的 $100 * (1 - \alpha)$ 正态近似估计区间为(同样要求 $n\hat{p}\hat{q} \geq 5$)

$$\hat{p} \pm z_{1-\alpha/2}\sqrt{\hat{p}\hat{q}/n}$$

67.3.2.2 精确法

当需要知道参数 p 精确的置信区间时, 使用此方法. 困难之处在于如何根据下式求解, 幸好一般的统计软件和 excel 已经有了相应的函数.

二项分布中参数 p 的 $100 * (1 - \alpha)$ 的精确的置信区间是求得区间 (p_1, p_2) 满足下式

$$P(X \geq x | p = p_1) = \alpha/2 = \sum_{k=x}^{n} \binom{n}{k} p_1^k (1 - p_1)^{n-k}$$

$$P(X \leq x | p = p_2) = \alpha/2 = \sum_{k=0}^{x} \binom{n}{k} p_2^k (1 - p_2)^{n-k}$$

67.4 泊松分布的估计

下面我们考察泊松分布的参数 λ 的估计.

67.4.1 点估计

若单位时间(面积, 长度等) 的泊松分布的参数为 λ, 则时间(面积,长度等)为 T 的泊松分布的参数为 $\mu = \lambda T$. 若一事件在时间(面积,长度等) T 内观察到 X 次, 则 μ 的无偏估计 $\hat{\mu} = X$, λ 的无偏估计 $hat\lambda = X/T$. 即

$$E(\mu) = E(X)$$

$$E(\lambda) = E(X)/T$$

67.4.2 区间估计

方法类似于二项分布中求精确置信区间. 若 x 为事件的观察数. T 为观察的时间(面积,长度). 则对 λ 的精确的 $100 * (1 - \alpha)$ 的置信区间是 $(\mu_1/T, \mu_2/T)$ 满足

$$P(X \geq x|\mu = \mu_1) = \alpha/2 = \sum_{k=x}^{\infty} e^{-\mu_1}\mu_1^k/k!$$

$$P(X \leq x|\mu = \mu_2) = \alpha/2 = \sum_{k=0}^{x} e^{-\mu_2}\mu_2^k/k!$$

67.5 单侧置信区间

如果我们只关心置信区间的一个边界, 那么就可以构建单侧置信区间.

正态分布的 $100*(1-\alpha)$ 下单侧置信区间为

$$x_{下} = \bar{x} + z_{\alpha}\sigma/\sqrt{n}$$

正态分布的 $100*(1-\alpha)$ 上单侧置信区间为

$$x_{上} = \bar{x} + z_{1-\alpha}\sigma/\sqrt{n}$$

注意: $z_{1-\alpha}$ 用于构建单侧置信区间, 而 $z_{1-\alpha/2}$ 用于构建双测置信区间.

其它分布的单侧置信区间构建方法类似.

Chapter 68

假设检验: 单样本推断

68.1 一般概念

- 零假设(null hypothesis, 也叫无效假设) 常记为 H_0, 指需要检验的假设.
- 备择假设(alternative hypothesis) 常记为 H_1, 是在某种意义上与零假设相反的假设.

如果要估计某分布的均值 μ 是否等于某个值 μ_0, 我们常常写成下面的形式

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu \neq \mu_0$$

这样有 4 种可能的结果

1. 我们接受 H_0, 实际上 H_0 也是正确的
2. 我们接受 H_0, 实际上 H_1 是正确的
3. 我们拒绝 H_0, 实际上 H_0 是正确的
4. 我们拒绝 H_0, 实际上 H_1 是正确的

实际使用时, 我们不可能证明零假设是否正确, 这样就可能出现两类不同类型的错误

- I 型错误(概率) 当 H_0 为真时我们拒绝 H_0 (的概率). 常常用 α 来表示, 也常称为一个检验的显著性水平.
- II 型错误(概率) 当 H_1 为真时我们接受 H_0 (的概率). 常常使用 β 来表示.

功效(power)

$power = 1 - \beta = 1 - II$型错误的概率 $= P$(拒绝$H_0|H_1$是真)

假设检验中, 我们的目的是使 α β 尽可能的小. 但是两者是矛盾的. 因为 α 变小时很难拒绝接受 H_0 从而使 β 增大, 反之 β 变小则 α 会增大. 我们一般先固定 $\alpha(0.10, 0.05, 0.01, ...)$, 然后再找某个某个检验使 β 尽可能的小, 或等价的使功效尽可能的大.

68.2 正态分布均值的单样本检验: 单侧备择

对正态分布均值的最好的检验是建立在样本均值上.

接受域(acceptance region) H_0 被接受时 $\bar{x}$ 的取值范围称为接受域.

拒绝域(rejection region) H_0 被拒绝时 $\bar{x}$ 的取值范围称为拒绝域.

单尾检验(one-tailed test, 单侧检验) 如果可以确定拒绝域由较小或较大的值构成, 但是不能同时成立, 即备择假设的未知均值小于或大于零假设下的未知均值, 这种情况下的检验称为单尾检验.

其假设可以写作

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu > \mu_0 (\mu < \mu_0)$$

68.2.1 方差未知的正态分布均值的单样本 t 检验

68.2.1.1 备择均值¡无效均值的假设检验

检验假设

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu < \mu_0$$

指定的显著性水平为 α, 计算

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

如果 $t < t_{n-1,\alpha}$ 我们拒绝 H_0, 如果 $t \geq t_{n-1,\alpha}$ 我们接受 H_0.

检验统计量(test statistic) 上式中的 t 值称为检验统计量, 因为我们的检验过程是建立在这个统计量上的.

临界值(critical value) 上式中的值 $t_{n-1,\alpha}$ 称为临界值. 因为检验结果依赖于 t 值与它的比较.

临界值方法在预先指定的 I 型错误概率后, 通过比较检验统计量与临界值从而判断检验结果的方法称为假设检验的临界值方法.

- α 的选择 α 水平的选择应该依赖于 I 型及 II 型错误的相对重要性. 大多数人不喜欢 α 水平远超过 0.05. 因此传统上使用 0.05 是最普遍的.

p-值法由检验统计量(例如t)计算出来的其末端或更末端的概率值. 它可以告诉我们检验结果是如何的显著. 其显著性的常用判断标准如下

- 0.01 ≤ p ¡ 0.05, 则结果是显著的
- 0.001 ≤ p ¡ 0.01. 则结果是高度(极其)显著的

- p ¡ 0.001, 在结果是很高的显著.
- p¿0.05, 则结果被认为没有统计显著性.(有的情况下(0.05 ≤ p ¡ 0.1)被认为有弱的显著性)

(科学上的显著性与统计学上的显著性是有区别的, 二者不必一致. 一个结果在统计上有显著性, 并不表明此结果在科学上有多么重要. 这种情况特别容易发生在大样本时, 因为大样本中一个很小的差异也可以被统计学家发现. 相反, 某些统计杀光你不显著的差异可能在科学上是重要的, 因为它可以促使科学家进一步用大样本去判断结果)

68.2.1.2 备择均值¿无效均值的假设检验

检验假设

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu > \mu_0$$

指定的显著性水平为 α, 计算

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

如果 $t > t_{n-1,\alpha}$ 我们拒绝 H_0, 如果 $t \leq t_{n-1,\alpha}$ 我们接受 H_0.

这个检验的 p-值 为

$$p = P(t_{n-1} > t)$$

68.3 正态分布均值的单样本检验: 双侧备择

大部分情况下, 先验知识是不足以判断在无效假设被否定后备择假设的均值应该取什么方向. 此时, 应该使用双侧检验.

双侧检验(two-tailed test, two-sided test)在备择假设下做研究的参数(此处为 μ)允许大于或小于无效假设下的参数(μ_0).

检验假设

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu \neq \mu_0$$

指定的显著性水平为 α, 最好的检验统计量

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

判断, 如果 $|t| > t_{n-1,1-\alpha/2}$ 则拒绝 H_0, 如果 $|t| \leq t_{n-1,1-\alpha/2}$ 则接受 H_0.

p-值的计算如同单侧检验一样.

$$p = \begin{cases} 2 * P(t_{n-1} \leq t) & \text{如果} t \leq 0 \\ 2 * [1 - P(t_{n-1} \leq t)], & \text{如果} t > 0 \end{cases}$$

(一般情况下, 双侧检验总是合适的, 因为它得出的显著性结论在任何应该单侧检验中也是可以满足的. 但是如果我们能够从专业知识判断应该是单侧, 则采用单侧检验会比双侧检验有更大的功效. 另外, 决定单侧还是双侧应该在数据分析之前. 如果计算 t 值后再考虑单侧还是双侧, 会产生人为的主观偏差)

68.4 方差已知时的正态分布均值的单样本 z 检验

某些研究中, 根据过去的资料翻查可能方差是知道的. 在这种情况下, 检验统计量 t 可以由 z 代替, 临界值也由相应的标准正态分布的临界值代替. 其中

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

其它的计算完全类似于方差未知时的t检验, 不论是单侧还是双侧.

68.5 检验的功效

功效可以告诉我们, 在备择假设是真时(应该否定 H_0 时), 我们可以否定 H_0 的可信程度. 若功效太低, 即使真实的 μ 与 μ_0 之间有差异, 也很难被所用的检验方法发现. 而不充分的样本量总是造成检验的低功效.

68.5.1 已知方差时正态分布均值的单样本z检验的功效

这个检验的假设是

$$H_0 : \mu = \mu_0 \quad vs. H_1 : \mu = \mu_1$$

此处已知潜在的分布是正态分布而总体方差为 σ^2, 则该检验的功效是

$$\Phi(z_\alpha + |\mu_0 - \mu_1|\sqrt{n}/\sigma) = \Phi(-z_{1-\alpha} + |\mu_0 - \mu_1|\sqrt{n}/\sigma)$$

影响功效的因素

- α 变小, 则 z_α 减小, 所以功效也减小.
- 若备择均值远离无效均值(即$|\mu_0 - \mu_1|$增加), 则功效增加.
- σ 增加, 功效减小
- 样本量 n 增加, 功效增加

68.5.2 双侧备择

双侧检验为

$$H_0 : \mu = \mu_0 \quad vs. H_1 : \mu \neq \mu_0$$

在 $\mu = \mu_1$ 的指定下,若分布是正态, 总体方差已知, 则z检验的功效的精确公式为

$$\Phi[-z_{1-\alpha/2} + \frac{(\mu_0 - \mu_1)\sqrt{n}}{\sigma}] + \Phi[-z_{1-\alpha/2} + \frac{(\mu_1 - \mu_0)\sqrt{n}}{\sigma}]$$

近似公式为

$$\Phi[-z_{1-\alpha/2} + \frac{|\mu_0 - \mu_1|\sqrt{n}}{\sigma}]$$

实际上, 若 $\mu_1 > \mu_0$, 则精确公式的第一项相对于第二项,第一项常常被忽略, 反之第二项常常被忽略.

68.6 样本量的决定

问题的描述是这样: 给出了将要进行的研究的显著性水平 α, 备择均值的期望 μ_1, 我们应该取多么大的样本才能达到希望的功效?

其实根据功效的公式,把功效当作已知而样本量 n 未知, 则可以很容易的求得需要的样本量. 从而我们有下面的公式.

68.6.1 单侧备择下的样本量

在单侧检验, 对于正态分布的均值, 显著性水平为 α, 检出有显著性差异的概率为 $power = 1 - \beta$ 时, 所求的样本量为

$$n = \frac{\sigma^2(z_{1-\beta} + z_{1-\alpha})^2}{(\mu_0 - \mu_1)^2}$$

明显, 影响样本量大小的因素有

- σ 增加时, 样本量增加

- α 变小, 样本量也增加
- power 增加时, 样本量也增加
- 无效均值与备择均值的距离($|\mu_0 - \mu_1|$)增加时, 样本量会减少. (距离增加1倍, 样本量会缩小到 1/4)

(一个问题是, 在估计样本量的时候, 如何估计这些参数? 通常无效假设 μ_0 是容易指定的, α 水平也容易指定. 而 power 却不太容易确定. 大多数研究者认为, $power < 80\%$ 是不太合适的. 备选的 μ_1 和总体方差通常是未知的. 它们可以从先前的工作, 经验, 或先验知识中得到. 在缺乏上述知识时, 有时由专业知识判断, 有时则做一些小的试验来估计. 最后要指出, 样本量的估计由于 μ_1 及 α 的不精确而通常只是提示性的, 它们通常都不精确)

68.6.2 双侧备择下的样本量

公式如下

$$n = \frac{\sigma^2(z_{1-\beta} + z_{1-\alpha/2})^2}{(\mu_0 - \mu_1)^2}$$

通常大于对应的单侧检验中的样本量, 因为 $z_{1-\alpha/2}$ 要大于 $z_{1-\alpha}$.

68.6.3 基于置信区间宽度的样本量估计

假设我们要估计正态分布中的均值, 且具有样本方差 s^2, 及要求有双侧 $100 * (1 - \alpha)$ 的置信区间, 使得 μ 的CI 的宽度不超过 L, 则样本量的近似估计为

$$n = 4z^2_{1-\alpha/2}s^2/L^2$$

68.7 假设检验与置信区间的关系

若双侧检验

$$H_0 : \mu = \mu_0 \quad vs. \quad H_1 : \mu \neq \mu_0$$

显著性水平为 α, 则对 μ 的双侧 $100 * (1 - \alpha)$ 置信区间包含了所有被 H_0 接受的值, 而置信区间之外就是拒绝 H_0 而接受 H_1 的值. 故置信区间与假设检验的结果是相同的.

区别

p-中法告诉我们此结果的统计显著性如何精细, 但是统计学的显著性在实际中往往并不是很重要. 因为大样本时, 实际差异可能是不大的,但是样本越大, 统计上就越显著, 这种差异自然不是那么重要.

置信区间往往给出均值可能存在的范围, 但不包括这个结果如何的显著.

所以, 一个好的作法是, 同时给出置信区间和p-值.

68.8 正态分布方差的估计–单样本卡方检验

在方差的置信区间估计及检验中, 正态条件特别重要. 若样本不满足正态性, 则临界值p-值及置信区间都不是有效的.

68.8.1 卡方检验

欲检验

$$H_0 : \sigma^2 = \sigma_0^2 \quad vs. \quad H_1 : \sigma^2 \neq \sigma_0^2$$

计算检验统计量

$$X^2 = (n-1)s^2/\sigma_0^2 \sim \chi_{n-1}^2$$

如果 $X^2 < \chi^2_{n-1,\alpha/2}$ 或 $X^2 > \chi^2_{n-1,1-\alpha/2}$, 则拒绝 H_0 如果 $\chi^2_{n-1,\alpha/2} \leq X^2 \leq \chi^2_{n-1,1-\alpha/2}$, 则接受 H_0

68.8.2 p-值(双侧备择)

同上计算检验统计量 X^2

如果 $s^2 \leq \sigma_0^2$, 则 p-值=2*(χ^2_{n-1} 分布曲线下从左到 X^2 的面积)

如果 $s^2 > \sigma_0^2$, 则 p-值=2*(χ^2_{n-1} 分布曲线下从右到 X^2 的面积)

68.9 二项分布的单样本检验

68.9.1 正态近似法

68.9.1.1 单样本检验

双侧备择的假设检验为

$$H_0 : p = p_0 \quad vs. \quad p \neq p_0$$

记检验统计量为

$$z = (p - p_0)/\sqrt{p_0 q_0/n}$$

如果$|z| \leq z_{1-\alpha/2}$, 则接受 H_0. 否则接受 H_1.

68.9.1.2 p-值计算

若 $p < p_0$, 则 p-值=$2 * \Phi(z)$

若 $p \geq p_0$, 则 p-值=$2 * [1 - \Phi(z)]$

68.9.2 精确的p-值计算

双侧备择下, 若 x 为 n 次试验成功的次数, 则精确p-值为

如果 $p \le p_0$, 则 p-值$=2 * P(X \le x)$

如果 $p > p_0$, 则 p-值$=2 * P(X \ge x)$

(注意: 任何时候, p-值都对应于出现在样本点末端或更末端的事件的概率)

68.10 功效及样本量的计算

双侧备择下的假设下,

$$H_0 : p = p_0 \qquad vs. \qquad p \ne p_0$$

在备择假设具体的指定值 $p = p_1$ 下, 正态近似法检验的功效为

$$\Phi[\sqrt{\frac{p_0 q_0}{p_1 q_1}}(z_{\alpha/2} + \frac{|p_0 - p_1|\sqrt{n}}{\sqrt{p_0 q_0}})]$$

(注意: 这个公式只在 $np_0q_0 \ge 5$ 时使用)

指定功效为 $1 - \beta$, 双侧备择下样本量的估计为

$$n = \frac{p_0 q_0 (z_{1-\alpha/2} + z_{1-\beta}\sqrt{\frac{p_1 q_1}{p_0 q_0}})^2}{(p_1 - p_0)^2}$$

68.11 泊松分布的单样本推断–小样本检验

(对于大样本的检验使用正态近似或二项近似)

如果在研究中事件很罕见(例如某些稀有疾病), 则事件的观察数可以考虑为泊松分布, 其未知期望为 μ, 我们要做的检验是 $H_0 : \mu = \mu_0 \quad vs. \quad H_1 : \mu \neq \mu_0$.

对于临界值法, 我们先要根据观察数x计算出 μ 的双侧置信区间 $(c_1, c_2$, 然后判断: 若 μ_0 在此区间内, 则接受 H_0, 否则接受 H_1.[1]

p-值法中, 在 H_0 为真时, 随机变量 X 是具有参数 μ_0 的泊松分布. 因此 μ 的精确 p-值为

$$min(2 * \sum_{k=0}^{x} \frac{e^{-\mu_0}\mu_0^k}{k!}, 1) \qquad \text{如果} x < \mu_0$$

$$min[2 * (1 - \sum_{k=0}^{x} \frac{e^{-\mu_0}\mu_0^k}{k!}), 1] \qquad \text{如果} x \geq \mu_0$$

[1]现在一般使用软件计算置信区间, 其值往往不是整数. 一些科学用表也可以查到

Chapter 69

假设检验: 两样本推断

单样本的检验都是一个样本上的检验, 建立在一个一般性的大总体上, 这个总体的参数被认为是已知的, 而吧一般所在总体的参数与一般性的总体的已知参数作比较.

两一般的假设检验问题是指两个不同总体的潜在参数都是未知的, 是需要做比较的.

69.1 匹配样本 t 检验

当第一组样本中每一个数据点都与第二组样本中的唯一数据点相联系, 这样的两个样本称为匹配(或配对)样本. (paired-sample)这样的研究称为配对研究设计.

69.1.1 匹配t检验

当两个样本是匹配样本, 服从正态分布, 均值分别为 μ 和 $\mu+\Delta$, 方差都是 σ^2. 我们想知道两个样本的均值是否相等, 即 Δ 是否为 0. 检验假设为

$$H_0 : \Delta = 0 \qquad vs. \qquad H_1 : \Delta \neq 0$$

由于是匹配样本, 每一对样本的差记为 d_i, 则 d_i 也是正态分布, 其均值为 Δ 且方差记为 σ_d^2. 于是样本均值之差 $\bar{d}$ 也具有正态分布, 其均值为 Δ 且方差为σ_d^2/n. 因此假设检验可以当作单样本t检验. 故有下面的检验方法, 称为匹配 t 检验. 记

$$t = \bar{d}/(s_d/\sqrt{n})$$

此处 $\bar{d}$ 为平均差异

$$\bar{d} = \Delta = (d_1 + \cdots + d_n)/n$$

s_d 是观察值差异的样本标准差

$$s_d = \sqrt{[\sum_{i=1}^{n} d_i^2 - (\sum_{i=1}^{n} d_i)^2/n]/(n-1)}$$

n = 匹配组数.

如果 $|t| > t_{n-1,1-\alpha/2}$, 则拒绝H_0.

如果 $|t| \leq t_{n-1,1-\alpha/2}$, 则接受H_0.

69.1.2 匹配检验的p-值计算

如果 $t \leq 0$, 则 p-值=2*[t_{n-1}分布曲线下从左到$t = \bar{d}/(s_d/\sqrt{n})$点的面积]

如果 $t > 0$, 则 p-值=2*[t_{n-1}分布曲线下从右到$t = \bar{d}/(s_d/\sqrt{n})$点的面积]

69.1.3 匹配样本均值比较的区间的估计

两匹配样本潜在均值差(Δ)的 $100\% * (1-\alpha)$ 置信区间是

$$\bar{d} \pm t_{n-1,1-\alpha/2} s_d/\sqrt{n}$$

69.2 等方差的两独立样本均值比较的t检验

当两个样本中的数据不发生关系时, 称为独立的两样本. (independent-sample)

设第一组样本量为 n_1, 每一个值都服从正态分布 $N(\mu_1, \sigma^2)$, 其均值为 x_1, 样本方差为 s_1^2

设第二组样本量为 n_2, 每一个值都服从正态分布 $N(\mu_2, \sigma^2)$, 其均值为 x_2, 样本方差为 s_2^2

(此处假定两组潜在方差相等)

69.2.1 t检验

我们要检验

$$H_0 : \mu_1 = \mu_2 \qquad vs. \qquad H_1 : \mu_1 \neq \mu_2$$

我们知道

$$\bar{X_1} - \bar{X_2} \sim N[\mu_1 - \mu_2, \sigma^2(\frac{1}{n_1} + \frac{1}{n_2})]$$

如果 H_0 为真, 则 $\mu_1 = \mu_2$, 于是上式成为

$$\bar{X_1} - \bar{X_2} \sim N[0, \sigma^2(\frac{1}{n_1} + \frac{1}{n_2})]$$

标准化后成为

$$\frac{\bar{X_1} - \bar{X_2}}{\sigma\sqrt{1/n_1 + 1/n_2}} \sim N(0, 1)$$

下一个问题是方差的合并估计, 合理的估计应该是对两个方差加权平均, 权值就是样本方差中的自由度, 于是有

$$s^2 = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}$$

此处 s_2 的自由度为 n_1+n_2-2. 代入后均值差就变成了自由度为 n_1+n_2-2 的 t 分布, 而不再是 $N(0,1)$. 从而检验统计量为

$$t = \frac{\bar{x_1} - \bar{x_2}}{s\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}, \qquad df = n_1+n_2-2$$

如果 $|t| > t_{n_1+n_2-2,1-\alpha/2}$, 则拒绝 H_0

如果 $|t| \leq t_{n_1+n_2-2,1-\alpha/2}$, 则接受 H_0

69.2.2 p-值

类似, 我们也可以得到 p-值

如果 $t \leq 0$, 则 p-值=$2*$(t_{n-1+n_2-2}分布曲线下t值左边的面积)

如果 $t > 0$, 则 p-值=$2*$(t_{n-1+n_2-2}分布曲线下t值右边的面积)

69.2.3 区间估计

我们也可以计算两样本均值真实差异的置信区间.

双侧及等方差下, 两独立样本真实均值差异的双侧 $100\%*(1-\alpha)$ 的置信区间为

$$\bar{x_1} - \bar{x_2} \quad \pm \quad t_{n_1+n_2-2,1-\alpha/2} s\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}$$

69.3 两方差相等性检验-F检验

检验假设

$$H_0 : \sigma_1^2 = \sigma_2^2 \quad vs. \quad H_1 : \sigma_1^2 \neq \sigma_2^2$$

合理的方法应建立在两方差的相对度量上, 即比值. 如果比值很大或很小, 则拒绝 H_0, 若接近 1, 则接受.

首先来看σ_1^2/σ_2^2 的分布

69.3.1 F 分布

方差比的分布由统计学家 R.A.Fisher 和 G.Snedecor 研究完成. 他们在两方差相等的假设下得到上述分布, 称为 F 分布.

F 分布由两个参数, 分子的自由度和分母的自由度决定.

若我们记分子样本的样本量为 n_1, 其自由度为 $n_1 - 1$, 分母样本的样本量为 n_2, 其自由度为 $n_2 - 1$. 则F 分布由 $n_1 - 1$ 及 $n_2 - 1$ 共同决定, 记此时的分布为 F_{n_1-1,n_2-1}

自由度为 d_1, d_2 的F 分布的第 $100 * p$ 百分位点记为 $F_{d_1,d_2,p}$, 即

$$P(F_{d_1,d_2} \leq F_{d_1,d_2,p}) = p$$

具自由度 d_1, d_2 的F分布的下侧第p个百分位点, 就是具有自由度为 d_12, d_1 的F 分布的上侧第p个百分位点的倒数, 即

$$F_{d_1,d_2,p} = 1/F_{d_2,d_1,1-p}$$

69.3.2 F 检验

若要检验 $H_0 : \sigma_1^2 = \sigma_2^2$ *vs.* $H_1 : \sigma_1^2 \neq \sigma_2^2$, 而显著性水平为 α, 则

我 们 计 算 统 计 量

$$F = s_1^2/s_2^2$$

如 果$F > F_{n_1-1,n_2-1,1-\alpha/2}$ 或$F < F_{n_1-1,n_2-1,\alpha/2}$, 则 拒 绝 H_0.

如 果$F_{n_1-1,n_2-1,\alpha/2} \leq F \leq F_{n_1-1,n_2-1,1-\alpha/2}$, 则 接 受 H_0.

精 确 的p-值 由 下 面 得 到

如 果$F \geq 1$, 则 p-值$=2 * P(F_{n_1-1,n_2-1} > F)$

如 果$F < 1$, 则 p-值$=2 * P(F_{n_1-1,n_2-1} < F)$

69.4 方 差 不 等 的 两 个 独 立 样 本 的 t 检 验

现 假 设 有 两 个 正 态 分 布 的 样 本, 第 一 个 样 本 量 为 n_1, 服 从 $N(\mu_1, \sigma_1^2$. 第 二 个 样 本 量 为 n_2, 服 从 $N(\mu_2, \sigma_2^2$.

我 们 要 检 验$H_0 : \mu - 1 = \mu_2$ $vs.$ $H_1 : \mu_1 \neq \mu_2$. 统 计 学 家 常 称 这 个 问 题 为 Behrens-Fisher 问 题.

我 们 有

$$\bar{X_1} - \bar{X_2} \sim N(\mu_1 - \mu_2, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})$$

在H_0成 立 时

$$\bar{X_1} - \bar{X_2} \sim N(0, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})$$

标 准 化 后 的 检 验 统 计 量 为

$$z = (\bar{x_1} - \bar{x_2})/\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

由于方差通常未知, 使用样本方差分别估计时, 因为潜在方差不同, 所以加权合并方差的方法不可用. 若使用样本方差代替后, 检验统计量变成

$$t = (\bar{x_1} - \bar{x_2}) / \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

在H_0成立时, 上述 t 的精确分布难于找出, 但是在合适的 I 型错误下, 已经找到了几个近似的分布.

69.4.1 不等方差下两个独立样本的t检验

此方法为 Satterthwaite 近似方法.

先计算检验统计量 t 如上.

再计算近似自由度

$$d' = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(\frac{s_1^2}{n_1})^2/(n_1 - 1) + (\frac{s_2^2}{n_2})^2/(n_2 - 1)}$$

把 d' 四舍五入到最近的整数 d'', 则

如果$|t| > t_{d'',1-\alpha/2}$, 则拒绝 H_0.

如果$|t| \leq d'', 1 - \alpha/2$, 则接受 H_0.

69.4.2 p-值

类似地,

如果$t \geq 0$, 则 p-值=$2 * (t_{d''}$分布在t值左边的面积)

如果$t < 0$, 则 p-值=$2 * (t_{d''}$分布在t值右边的面积)

69.4.3 置信区间

类似的可以证明, 不等方差下也有均值差的 $100\% * (1-\alpha)$ 置信区间

$$\bar{x_1} - \bar{x_2} \quad \pm \quad t_{d'',1-\alpha/2}\sqrt{s_1^2/n_1 + s_2^2/n_2}$$

69.5 独立样本均值比较中样本量及功效的估计

估计等样本数, 正态分布的两独立样本均值比较, 双侧检验且显著性水平为 α 功效为 $1-\beta$ 下, 样本量的估计为

$$n = \frac{(\sigma_1^2 + \sigma_2^2)(z_{1-\alpha/2} + z_{1-\beta})^2}{\Delta^2} = \text{每一组的样本量}$$

其中 $\Delta = |\mu_1 - \mu_2|$.

换言之, 每一组样本量为 n 时, 将有 $1-\beta$ 的机会发现两组中真实存在 Δ 的差异.

不等样本数所需要的样本量为

$$n_1 = \frac{(\sigma_1^2 + \sigma_2^2/k)(z_{1-\alpha/2} + z_{1-\beta})^2}{\Delta^2} = \text{第一组的样本量}$$

$$n_2 = \frac{(k\sigma_1^2 + \sigma_2^2)(z_{1-\alpha/2} + z_{1-\beta})^2}{\Delta^2} = \text{第二组的样本量}$$

其中 $k = n_2/n_1$ 为两样本事先指定的比值.

功效的估计为

$$power = \Phi[-z_{1-\alpha/2} + \frac{\sqrt{n_1}\Delta}{\sqrt{\sigma_1^2 + \sigma_2^2/k}}]$$

其中 $k = n_2/n_1$ 为两样本事先指定的比值.

Chapter 70

非参数检验

之前的数据被假设来自某个潜在的分布, 这个分布的一般形式是已知的, 只是参数的具体值未知. 估计和检验方法都是基于这个分布, 来得到具体值的点或区间等. 这种方法通常被称为参数统计方法.

而如果分布的形状未知, 中心基线定理似乎又不太合适, 例如样本数太少, 这时就必须使用非参数统计方法(nonparametric statistical method). 该方法对肺部形状很少有要求.

下面介绍几个概念.

基数数据(cardinal data) 是一种有尺度的输送就, 可以用某种尺度测出任何两个数据之间的距离.

进一步, 如果该数据的零点是固定的, 称为比例尺度(ratio scale)数据. 若零点是任意的, 则称为区间尺度(interval scale)数据.

例如, 体温是一种区间尺度数据, 因为它的零点是不确定的, 例如在华氏和摄氏温度中, 零点有不同的意义.

体重及身高是比例尺度数据, 因为零点对它们有明确的意义.

比例尺度中, 任何两个数据的比值是有意义的.

区间尺度数据, 比值可能没有意义, 如温度的比值是没有意义的.

不论哪种形式的基数数据, 均值和标准差都是有意义的.

有序数据(ordinal data) 指它们之间可以排成次序但是却没有指定的数值, 因此通常的算数运算是没有意义的.

例如,颜色的深浅程度和视力的等级, 病情的恶化程度, 对每一水平, 可以用数值去代表, 但是没有一个唯一的标准. 而且它们之间的算数运算是没有意义的.

由于无法使用一组有意义的数值代表此类数据, 故计算它们的均值和标准差是不合适的. 但是我们仍然对此类数据之间的比较感兴趣. 非参数检验即适用于此类数据.

名义尺度数据(nominal sccale data) 不同的数据被分为类型或属性, 而类型是没有次序的.

例如疾病的种类, 零件的型号等, 它们都是某类事物的属性, 是没有次序的.

70.1 匹配数据的符号检验(sign test)

对于有序数据, 我们可以度量它们的相对大小, 但是不能相减, 即不能用差值的大小来衡量其相对关系. 此时使用符号检验.

例如要检验两种防晒膏的效果. 随机涂敷于左右手臂, 阳光下一小时. 假设我们只能判定手臂红色的程度

- A 防晒膏 ¿ B 防晒膏, 记为+1.
- A 防晒膏 ¡ B 防晒膏, 记为-1.
- 两者一样, 记为 0

首先去掉 0 值, 因为它对两种防晒膏的好坏不提供任何信息.

如果 +1 远多于 -1, 有理由相信, B 防晒膏的效果要好于 A. 若 -1 远多于 +1, 那么 A 的效果应该好于 B 的. 若 +1 和 -1 差不多, 那么两者效果可以认定没有显著差别.

实际上, 这是二项分布的一个特例. 此处假设

$$H_0 : p = 1/2 \quad vs. \quad H_1 : p \neq 1/2$$

此处 p 为 A 好于 B 的概率.

设 +1 和 -1 的个数有 n 个, 记 +1 的个数为 c, 那么 $E(c) = np$, $var(c) = npq$. 在零假设成立时, $c = n/2, var(c) = n/4$. 所以我们有

70.1.1 正态近似法

根据上面的描述, $c \sim N(n/2, n/4)$. 在显著性水平为 α 时, 使用双侧检验, 我们有

如果$|c| > \frac{n}{2} + \frac{1}{2} + z_{1-\alpha/2}\sqrt{n/4}$, 则拒绝 H_0

如果$|c| \leq \frac{n}{2} + \frac{1}{2} + z_{1-\alpha/2}\sqrt{n/4}$, 则接受 H_0

精确的p-值为

$$p = 2 * [1 - \Phi[\frac{c - \frac{n}{2} - 0.5}{\sqrt{n/4}}]], \text{如果} c \geq \frac{n}{2}$$

$$p = 2 * [\Phi[\frac{c - \frac{n}{2} + 0.5}{\sqrt{n/4}}]], \text{如果} c < \frac{n}{2}$$

(正态近似适用于 $npq \geq 5, \ or \ n(1/2)(1/2) \geq 5, \ or \ n \geq 20$)

70.1.2 精 确 方 法

如 果 n ¡ 20, 则 需 要 使 用 精 确 的 二 项 分 布 公 式. 其 p-值 计 算 为

$$p = 2 * \sum_{k=c} n \binom{n}{k} \frac{1}{2}^n, \quad say \quad c > n/2$$

$$p = 2 * \sum_{k=0} c \binom{n}{k} \frac{1}{2}^n, \quad say \quad c < n/2$$

$c = n/2, \quad p = 1.0$

Chapter 71

试验设计

[1]

71.1 基本原理

71.1.1 意义

- 广义指整个研究课题的设计, 包括从申请到结题评估的全部过程
- 狭义仅指试验单位的选择, 分组与排列等

71.1.2 基本要求

- 试验目的要明确
- 试验条件有代表性–可推广
- 试验结果可靠–严格试验要求与操作, 减少试验误差
- 试验结果能够重演–在不同的时间, 地域等(主要指农业试验)

[1]主要根据《生物统计学》(第二版) 第八章. 著者: 李春喜 等

71.1.3 试验设计的基本要素

- 处理因素–对对象给予的某种外部干扰(或措施), 简称处理. 例如: 温度, 压力是两个不同的因素
- 受试对象–就是被处理的对象
- 处理效应–处理完后的结果, 体现在数据上, 其中包含误差.

71.1.3.1 试验误差及控制途径

试验误差的分类

- 系统误差(可以避免)–设计不当造成的误差. 例如其它条件本应一样而不一样.
- 随机误差(不可避免)–不可控制的偶然因素造成

误差来源

- 试验材料固有的差异
- 试验条件不一致
- 操作技术不一致
- 偶然性因素的影响

控制途径

- 选择纯合一致的试验材料
- 改进操作管理制度
- 精心选择试验单位
- 采用合理的试验设计

71.1.3.2 试验设计的基本原理

- 重复
- 随机化
- 局部控制

71.2 对比设计及其统计分析

71.2.1 对比设计

- 在农作物试验上讲究地块平行
- 动植物试验就是配对试验

71.2.2 统计分析

一般使用 t 检验

71.3 随机区组设计及统计分析

71.3.1 设计

因为试验单位性质不同, 因而分为不同的组.

例如, 有 2 个组(长势好的A, 不好的B), 每个组试验 3 种温度(1个因素的 3 个水平), 设 A 组有90个个体, B组也有90个个体, 那么每个组都可以分为3组处理.

71.3.2 统计

一般使用方差分析, 可以比较A B两组的不同(两个组产量是否有显著差异), 也可以比较3种温度的不同影响(例如3种温度下产量是否有显著差异).

71.4 拉丁方设计

优点: 其误差是随机区组设计的73%.

缺点: 需要保持行, 列, 处理数三者相等. 故处理数不能太多(一般5-10, ¡=4 自由度不够, ¿10就太庞大了). 自由度至少12, 最好20以上.

步骤

1. 选择标准拉丁方
2. 行随机化
3. 列随机化
4. 处理随机化

统计分析

行之间, 列之间, 处理之间都可以当作区组, 均可比较, 故比随机化区组多了一项.

71.5 裂区设计(主要针对农业试验)

如果是多因素试验处理的组合数太多. 应用条件:

1. 若已知某因素内部差异大, 则选择作为主因素

2. 若某因素需要更多的区域(资源), 则应作为主因素

3. 若某因素要求的精度高宜作为主因素

4. 若需临时再加入一个试验因素, 可以在原设计的小区里(随机区组)再加一个因素, 这样就成了裂区设计(但是应尽量在试验前设计好, 避免中途更改试验方案).

这里只介绍二因素设计.

71.6 正交设计

对付多因素多水平的试验. 例如, 3因素3水平的完全试验组合要$3^3 = 27$个, 4因素4水平的完全试验组合要$4^4 = 256$个. D. J. Finney 倡议部分试验, 后来成为正交设计. 实质上是把具有代表性的试验做了, 其余忽略掉了.

Part X

参考文献

Bibliography

[1] 张奠宙. *20*世纪数学经纬. 华东师范大学出版社. 2001

[2] 朱慧明 韩玉启 著. 贝叶斯多元统计推断理论. 科学出版社. 2006

[3] 张尧庭 陈汉峰 编著. 贝叶斯统计推断. 科学出版社. 1991

[4] 钟开莱. 初等概率论附随机过程. 人民教育出版社. 1979

[5] 复旦大学. 概率论第一册概率论基础, 人民教育出版社, 1979.

[6] 中山大学数学系梁之舜 邓集贤 杨维权 司徒荣 邓永录 概率论及数理统计*(*第二版*)*, 高等教育出版社, 1988.

[7] (美) Richard O. Duda, Peter E. Hart, David G. Stork. 李 宏 东 姚天翔 等译. 模式分类*(Pattern Classification)*. 机械工业出版社, 中信出版社. 2003

[8] 徐克学. 生物数学. 科学出版社. 2002

[9] 杜荣骞. 生物统计学, 高等教育出版社

[10] 李春喜 王志和 王文林 生物统计学 科学出版社. 2000.

[11] Bernard Rosner. 孙尚拱 译. 生物统计学基础（*Fundamentals of Biostatistics*）第五版. 科学出版社, 2004.

[12] 徐端正.生物统计学-在实验和临床药理学中的应用. 科学出版社. 2004

[13] 茆诗松, 周纪芗, 陈颖. 试验设计, 中国统计出版社. 2004.

[14] [美] W.J.Conover 著, 崔恒建译. 实用非参数统计*(*第三版*)*. 人民邮电出版社. 2006

[15] 薛毅陈立萍编著. 统计建模与*R*软件. 清华大学出版社. 2006.

[16] 顾万春著. 统计遗传学. 科学出版社. 2004

[17] [美] Albert Boggess, Francis J. Narcowich 著, 芮国胜康健等译. 小波与傅里叶分析基础*(A First Course in Wavelets with Fourier Analysis)*. 人民邮电出版社. 2006

[18] 王燕编著. 应用时间序列分析. 中国人民大学出版社. 2005

[19] W. N. Venables, D. M. Smith, R 核心开发小组（the R Development Core Team）, 丁国徽译 *R* 导论–关于 *R* 语言的注解：一个数据分析和图形显示的程序设计环境英文版本2.3.0 (2006-04-24) 中文版本0.1 (2006-06-15). 2006.

[20] Emmanuel Paradis. 翻译: （Chap1–2: 王学枫；Chap3: 谢益辉；Chap4: 李军焘；Chap5–7: 丁国徽 ）*R for Beginners* Chinese Edition 2.0. 2006

[21] *R*语言简介–*R*语言笔记:数据分析与绘图的编程环境 (版本1.7) R Development Core Team. June 10, 2006

[22] Brian S. Everitt and Torsten Hothorn. *A Handbook of Statistical Analyses Using R* CHAPTER 9. Survival Analysis: Glioma Treatment and Breast Cancer Survival

[23] Brian S. Everitt and Torsten Hothorn. *A Handbook of Statistical Analyses Using R* CHAPTER 12, Meta-Analysis: Nicotine Gum and Smoking Cessation and the Efficacy of BCG Vaccine in the Treatment of Tuberculosis.

[24] Fabio Frascati, Elia Biganzoli and Bruno Mario Cesana *'agreement': Analyse the agreement between two measurement methods*

[25] Virasakdi Chongsuvivatwong. *Analysis of Epidemiological Data using R and Epicalc*. Prince of Songkla University.

[26] Grant V. Farnsworth *Econometrics in R*. June 26, 2006

[27] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S(Fourth edition)*. Springer (mid 2002) Final 15 March 2002

[28] Paul Bliese (paul.bliese@us.army.mil) . *Multilevel Modeling in R (2.2)–A Brief Introduction to R, the multilevel package and the nlme package*. October 28, 2006

[29] John Fox. *Nonlinear Regression and Nonlinear Least Squares*. January 2002

[30] Julian J. Faraway *Practical Regression and Anova using R* July 2002.

[31] Vincent Zoonekynd (zoonek@math.jussieu.fr) *Statistics with R* 28th August 2005

[32] Kim Seefeld, Ernst Linder *Statistics Using R with Biological Examples*. 2007

[33] John Verzani. *simpleR – Using R for Introductory Statistics* 2001.

[34] Michael P. Fay. *Testing the Ratio of Two Poisson Rates*. June 5, 2007

[35] Brockwell, Peter J. and Davis, Richard A. *Time Series: Theory and Methods* Springer-Verlag. 1987

[36] Karline Soetaert *Using R for scientific computing*. September 2008