> getwd()#显示当前工作目录

>installed.packages(“”)[,c('Package','Version','LibPath')]#查找R包版本和安装位置（不要装错地方）

下载不了以管理员身份运行打开rstudio

>remove.packages(“”)#移除安装包

> setwd("C:/Users/YueYiFan/Documents")#修改当前工作目录至。。。要加双引号

> ls()列出当前工作空间中的对象

> rm(df)#移除某个对象

> help(options)#显示可用选项的说明，右下help中

>savehistory("C:/Users/YueYiFan/Documents")#保存命令到一个文件夹中

> loadhistory("C:/Users/YueYiFan/Documents")#载入一个历史命令文件

>q()#退出R

>dir.create("C:/Users/YueYiFan/Rlanguage")#创建一个目录

> install.packages("vcd")#安装vcd包

>installed.packages()[,c('Package','Version','LibPath')]#查找R包版本和安装位置（不要装错地方）。如果还是下载不成功

https://blog.csdn.net/binma2542/article/details/79947153

> library(vcd)#载入要用的包

> help(Arthritis)#查看数据集Arthritis

> example(Arthritis)#查看例子

* 创建向量

> a<-c(1,2,3)#使用c()创建向量，a是数值型向量

a<-c(1:3)#等价

> b<-c("one","two")#b是字符型向量

> c<-c(TRUE ,FALSE,TRUE)#c是逻辑型向量

> c[c(1,2)]#访问c中第一第二个元素

* 创建矩阵

①> y<-matrix(1:20,nrow = 5,ncol = 4)#创建五行四列的矩阵

②> cells<-c(1,26,24,68)#定义向量

> rnames<-c("R1","R2")

> cnames<-c("C1","C2")

> mymatrix<-matrix(cells,nrow=2,ncol = 2,byrow = TRUE,dimnames = list(rnames,cnames))#按行排列定义行列名

> mymatrix

C1 C2

R1 1 26

R2 24 68

③> x<-matrix(1:10,nrow = 2)

> x

> x[2,]

[1] 2 4 6 8 10

> x[,2]#查看矩阵第二列

> x[1,4]#查看矩阵第一行第四列

> x[1,c(4,5)]#查看矩阵第一行，第四列第五列

* 创建数组（数组是矩阵的一个自然推广）

> dim1<-c("A1","A2")

> dim2<-c("B1","B2","B3")

> dim3<-c("C1","C2","C3","C4")

> z<-array(1:24,c(2,3,4),dimnames=list(dim1,dim2,dim3))

> z

输出：, , C1

B1 B2 B3

A1 1 3 5

A2 2 4 6

, , C2

B1 B2 B3

A1 7 9 11

A2 8 10 12

, , C3

B1 B2 B3

A1 13 15 17

A2 14 16 18

, , C4

B1 B2 B3

A1 19 21 23

A2 20 22 24

* 创建数据框

①> patientID<-c(1,2,3,4)#建立一个有关病人信息的数据框

> age<-c(25,34,28,52)

> diabetes<-c('Type1',"Type2","Type1","Type1")

> status<-c("Poor","Improved","Excellent","Poor")

> patientdata<-data.frame(patientID,age,diabetes,status)

> patientdata

输出patientID age diabetes status

1 1 25 Type1 Poor

2 2 34 Type2 Improved

3 3 28 Type1 Excellent

4 4 52 Type1 Poor

②读取数据框中信息的三种方法

> patientdata[1:2]

> patientdata[c("diabetes","status")]

> patientdata$age#$被用于选取一个给定数据框中的某个特定变量，可用于生成特定列表，如下

> table(patientdata$diabetes,patientdata$status)

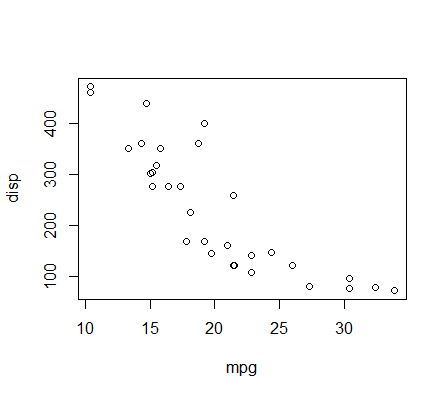
输出Excellent Improved Poor

Type1 1 0 2

Type2 0 1 0

③attach()将数据框添加到R的搜索路径中detach()将数据框从路径中移除

> attach(mtcars)

> plot(mpg,disp)#绘制点图，mpg为横坐标

> detach(mtcars)#将数据框mtcars从路径中移除,当已经对一个数据赋值可以去除其路径重新赋值

> with(mtcars,{print(summary(mpg))+ plot(mpg,disp)}) #with函数解决重复赋值，但赋值仅在括号后有用

④区分数据不同集体

patientID<-c(1,2,3,4)

age<-c(25,34,28,52)

diabetes<-c('Type1',"Type2","Type1","Type1")

status<-c("Poor","Improved","Excellent","Poor")

patientdata<-data.frame(patientID,age,diabetes,status,row.names = patientID)# row.names指定区分对象

patientdata

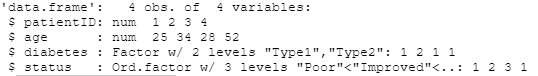
* 因子

diabetes<-c('Type1',"Type2","Type1","Type1")

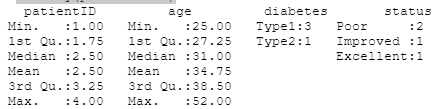
diabetes<-factor(diabetes)#将内部存储转化为（1，2，1，1），在内部将其关联为1=Type1，2=Type2（按照字母顺序订）

status<-factor(status,ordered=TRUE,levels=c("Poor","Improved","Excellent"))#赋值为poor=1，improved=2，excellent=3，如果不加levels=，则默认excellent=1（字母顺序）ordered=TRUE用于表示有序型变量

str(patientdata)#显示病人数据的结构，diabete是因子，status是有序型因子



summary(patientdata)#显示对象的统计概要



* 创建列表

g<-"My First List"

h<-c(25,26,18,39)

j<-matrix(1:10,nrow = 5)

k<-c("one","two","three")

mylist<-list(title=g,ages=h,j,k)#船舰一个列表，第一部分是title，第二部分是年龄，第三部分是j，第四部分是k

mylist

mylist[[2]]#输出表中第二部分

mylist[["ages"]]#输出ages部分

* 数据的输入

a.手动输入

> edit()#需要赋值给一个目，否则容易丢失

>mydata<-data.frame(age=numeric(0),gender=character(0),weight=numeric(0))#age是数值型，gender是字符型，体重是数值型；创建了一个指定模式但不含实际数据的变量

> mydata<-edit(mydata)#等价写法> fix(mydata)

> mydata

b.带分隔符的文本文件导入数据

> mydata<-read.table("C:/Users/YueYiFan/Desktop/数据处理/6-18数据处理/6-18 root/abcg29--root-full sacn---POS.csv",header = TRUE,row.names = "Mass",sep = ",",colClasses = "numeric")#header表示文件是否在第一行包含了变量名的逻辑型变量；sep是分开数据值的分隔符，默认sep=””,sep=”,”读取逗号分隔行内数据的文件，sep=”\t”读取制表符来分割行内数据的文件; row.names一列中不能有重复数值，用于指定一个或多个行标记符的可选参数，一般=1即第一列为行名。表格内不出现，‘（逗号单引号）;colClasses表示每一列的类numerics ，character等#csv文件保存的时候不能存成UTF-8

* #如果出现乱码应该在excl中按照如下操作完成：

第一步：点击“file”，即“文件”；

第二步：点击“export”，即“导出”；

第三步：点击“Change File Type”，即“改变文件格式”，选择"CSV"，进行导出即可；

* 如果出现line 1 did not have 833 elements，加上fill = TRUE
* 如果出现incomplete final line found by readTableHeader on，不要用read.table

使用total<-read.csv("D:/结肠样品/中山大学-结肠样品/chuli/全表.csv",header=T, row.names = 1 ,stringsAsFactors = F) #保留行名列名

* 导入excel数据

> install.packages("xlsxjars")#前三步缺一不可不行再下个CDK64

> install.packages("rJava")#前三步缺一不可

> install.packages("xlsx")#前三步缺一不可

> library(xlsx)

> workbook<-"C:/Users/YueYiFan/Desktop/数据处理/6-18数据处理/6-18 root/处理数据root.xlsx"

> mdafr<-read.xlsx(workbook,1)#1指第一个sheet

> mdafr

* 导入SPSS数据

> install.packages("Hmisc")

> library(Hmisc)

> mdf<-spss.get("C:/Users/YueYiFan/Desktop/农药生测/方差分析1.sav",use.value.labels = TRUE)# use.value.labels让函数将带有标签的变量导入R中水平对应相同的因子，mdf是导入R后的数据框

* 导出文件

csv格式：

> write.table(data,"mrbin1.csv", mes=FALSE,col.names=TRUE,sep=",")#失去第一列，getwd（）查看文件在哪里。

* 数据集的标注

①变量标签

> names(patientdata)[2]<-"Age at hospitalization"#将patientdata中的age改名为Age at hospitalization

②值标签

> patientdata$patientID<-factor(patientdata$patientID,levels = c(1,2,3,4),labels = c("Ann","Kitty","Lucy","Amy"))#levels代表变量实际值，labels是实际显示的向量（给四个病人取名字）

* 处理数据对象的实用函数

> length(patientID)#显示对象中元素的数量

> dim(patientdata)#显示对象的维度（4\*4）

> str(patientdata)#显示对象结构

> class(patientdata)#显示对象的类型（数据框）

> mode(patientdata)#显示对象模式（list）

> names(patientdata)#显示对象中各个成分名（pID age status）

> c(1,2)#将两个对象合并如一个向量

> cbind(1,2)#按列合并向量（为行向量）

> rbind(1,2)#按行合并向量（为列向量）

> patientdata#输出对象patientdata

> head(patientdata)#列出patientdata的开始部分

> tail(patientdata)#列出patientdata的结束部分

> ls()#显示当前工作环境对象列表

> rm(zxc,zxc2,zxc3)#删除zxc zxc2 zxc3

>rm(list = ls())#删除当前工作环境所有对象

>no<-edit(patientdata)#编辑patientdata并将其赋值给no

> fix(patientdata)#直接编辑patientdata

* 第三章初阶图形

①> pdf("C:/Users/YueYiFan/Desktop/test.pdf")#将图像存入这个pdf中

> attach(mtcars)#绑定数据框mtcars

> plot(wt,mpg)#打开了一个图形窗口并生成一副散点图（横轴wt车身重量，纵轴mpg每加仑汽油行驶的英里数）

> abline(lm(mpg~wt))#向图形添加了一条最优拟合曲线

> title("Regression of MPG on weight")#添加标题

> detach(mtcars)#为数据框解除绑定

> dev.off()#关闭指定的设备dev.new()：打开一个新的设备。dev.set()： 指定一个特定的设备为活跃设备。dev.next()：查看紧随当前活跃进程的下一个进程。dev.prev()：查看当前活跃进程的前一个进程。dev.cur()：查看当前的活跃设备。graphics.off()：关闭所有打开的设备。dev.list()：返回打开的设备的名单列表，但不包含空设备。

#也可直接鼠标右下框保存

> boxplot(wt,mpg)#绘制箱型图hist绘制直方图

②> dose<-c(20,30,40,45,60)

> drugA<-c(16,20,27,40,60)

> drugB<-c(15,18,25,31,40)

> plot(dose,drugA,type = "b")#b同时在图上绘制点和线（点线不相交）

> plot(dose,drugA,type = "p")#p只有点

> plot(dose,drugA,type = "l")#l只有线连续

> plot(dose,drugA,type = "c")#c只有线不连续

> plot(dose,drugA,type = "o")#o有点有线，点线相交

> plot(dose,drugA,type = "h")#h线直方图> hist(dose,drugA)绘制的是方块直方图

> plot(dose,drugA,type = "n")#n表no无任何点线

③①线条和点

> opar<-par(no.readonly = TRUE)#no.randomly可以生成一个可以修改的当前图形的参数列表

> par(lty=2,pch=7)#lty=2线条类型为虚线，pch=17点为实心三角可以写为> par(lty=2) > par(pch=2)

> plot(dose,drugB,type = "b")

> par(opar)

> plot(dose,drugB,type = "b",lty=3,pch=5)#对于plot可以直接指定线和点的形式

> plot(dose,drugB,type = "b",lty=3,pch=5,lwd=3,cex=2)

#cex表示符号（点）为几倍大小，lwd指定线条宽度

③②颜色

> colors()#查看所有可用颜色

> install.packages("RColorBrewer")#首次使用下载此包

> n<-7

> library(RColorBrewer)

> mcolors<-brewer.pal(n,"Set1")#使用n个颜色，引号内set是色条对应名称

> barplot(rep(1,n),col = mcolors)#rep将1循环n遍（这里为七条柱状）；barplot绘制柱状图；col默认绘图的颜色

> brewer.pal.info#得到所有可选调色表列表

> display.brewer.all()#输出所有调色板颜色

> n<-10

> mcolors<-rainbow(n)#生成十种彩虹型颜色heat.colors、terrain.colors、topo.colors、cm.colors也可用

> pie(rep(1,n),labels = mcolors,col = mcolors)#饼状图

> mygreys<-grey(0:n/n)#调节灰色为十个等级

> pie(rep(1,n),labels = mygreys,col = mygreys)

③④文本属性

> par(font.lab=3,cex.lab=1.5,font.main=4,cex.main=1)

# font.lab：绘图使用的字体样式（整数）；cex.lab坐标轴标签的缩放倍数；font.main=4标题的字体；cex.main=1标题的缩放倍数

#cex.axis、font.axis坐标轴刻度文字的缩放倍数、字体样式；cex.sub、font.sub副标题

③④图形尺寸与边界尺寸

> par(pin=c(4,3),mai=c(1,.2,1,.4))#pin以英寸表示图形的宽和高；mai（英寸，mar英分）以数值向量表示边界的大小，顺序为“下、左、上、右”

> par(pin=c(4,3),mai=c(1,.2,1,.4))

> plot(dose,drugB,type = "b",lty=3,pch=5,lwd=3,cex=2)

> opar<-par(no.readonly = TRUE)

> par(pin=c(2,3))

> par(lwd=2,cex=1.5)

> par(cex.axis=.75,font.axis=3)

> plot(dose,drugA,type = "b",pch=19,lty=2,col="red")

> plot(dose,drugB,type = "b",pch=16,lty=7,col="blue",bg="green")#bg图形的背景色；fg前景色（边框）

> par(opar)

> plot(dose,drugA,xaxt="s",yaxt="s")#横坐标数据显示，纵坐标数据显示，显示不出来右下清楚所有plot重来

④添加文本、自定义坐标轴和图例

> plot(dose,drugA,type = "b",col="pink",lty=2,pch=2,lwd=2,main="Clinical Trials for Drug A",sub = "This is hypothetical data",xlab = "dosage",ylab = "Drug Response",xlim = c(0,60),ylim = c(10,70))#标题main；副标题sub；坐标轴标签xlab、ylab；坐标轴范围xlim、ylim

④①标题

> plot(dose,drugA,xaxs="i",yaxs="i")#绘图边框不留白

> plot(dose,drugA,bty="n")#去掉边框

> plot(dose,drugA,ann = F)#去掉横纵坐标标签

> title(main="main title",sub = "subtitle",xlab = "x-axis label",ylab = "y-axis label")#使用title为图形添加标题和坐标轴标签

> > title(main="main title",col.main="red",sub = "subtitle",col.sub="blue",xlab="x",ylab="y",col.lab="green",cex.lab=0.75)#指定主标题副标题颜色和坐标轴标签颜色

④②坐标轴

出现报错figure margins too large需要把plot窗口拖大一点，做一点保存一点 Rstuio容易死机

> y<-c(1:10)

> y<-x

> z<-10/x

> opar<-par(no.readonly = TRUE)

> par(mar=c(5,4,4,8)+0.1)#增加边界大小

> plot(x,y,type="b",pch=21,col="red",yaxt="n",lty=3,ann = FALSE)# ann=FALSE来移除标题和标签

> lines(x,z,type = "b",pch=22,col="blue",lty=2)#lines添加x对1/x的直线

> axis(2,at=x,labels = x,col.axis="red",las=2)#第一个2表示在左侧、1下3上4右；at表示需要绘制刻度线的位置

> axis(4,at=z,labels = round(z,digits = 2),col.axis="blue",las=2,cex.axis=0.7,tck=-0.01)#tck表示刻度线长度，正值在图形内侧，负值在图形外侧，默认-0.01

> mtext("y=1/x",side=4,line = 3,cex.lab=1,las=2,col="blue")#mtext添加文本

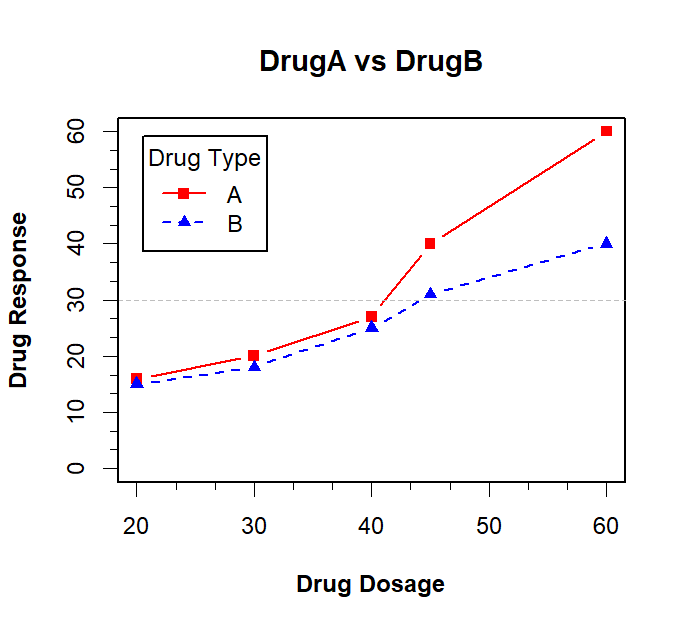
> title("An example of Creative Axes",xlab="X values",ylab="Y=X")

> par(opar)

④③参考线

> abline(h=c(1,5,8))##abline(h=yvalues,v=xvalues)

> abline(v=seq(1,10,2),lty=2,col="blue")#seq函数从1开始到10结束间距为2

④④图例

> dose<-c(20,30,40,45,60)

> drugA<-c(16,20,27,40,60)

> drugB<-c(15,18,25,31,40)

> opar<-par(no.readonly = TRUE)

> par(lwd=2,cex=1.5,font.lab=2)

> plot(dose,drugA,type="b",pch=15,lty=1,col="red",ylim = c(0,60),main="DrugA vs DrugB",xlab = "Drug Dosage",ylab="Drug Response")

> lines(dose,drugB,type = "b",pch=17,lty=2,col="blue")#添加线

> abline(h=c(30),lwd=1.5,lty=2,col="grey")#y=30处参考线

> library(Hmisc)

> minor.tick(nx=3,ny=3,tick.ratio = 0.5)#添加次要刻度线，每个x、y主要刻度分成三个次要刻度，次要刻度线大小是主要刻度线的0.5

> legend("topleft",inset=0.05,title="Drug Type",c("A","B"),lty = c(1,2),pch = c(15,17),col = c("red","blue"))#legend添加图例，格式legend(location,title,legend…);location有bottom,bottomleft,left,topleft,top,topright,right,bottomright,center等，inset=指定图例向图形内测移动大小；title=图例标题；legend是图例字符串；…可自定义颜色等、指定线条形状

> par(opar)

④⑤文本标注

Ⅰ> attach(mtcars)

> plot(wt,mpg,main = "Mileage vs. Car weight",xlab = "Weight",ylab = "Mileage",pch=18,col="blue")

> text(wt,mpg,row.names(mtcars),cex=0.6,pos = 4,col = "red")#文本标注，row.names()定义行名；pos表示文本相对于参数的位置1下2左3上4右

> detach(mtcars)

Ⅱ> opar<-par(no.readonly = TRUE)

> par(cex=1.5)

> plot(1:7,1:7,type = "n")

> text(3,3,"Example of default text")

> text(4,4,family="mono","Example of mono-spaced text")#family=绘制文本时使用的字体族。标准的取值为serif（衬线）、sans（无衬线）和mono（等宽）

> text(5,5,family="serif","Example of serif text")

> par(opar)

④⑥数学标注

> help("plotmath")#具体查看如何使用加减乘除等

> x%/%y#x除以y

> x%\*%y#x叉乘y

⑤图形组合

Ⅰ> attach(mtcars)

> opar<-par(no.readonly = TRUE)

> par(mfrow=c(2,2))#建立row=2，col=2的组合图形

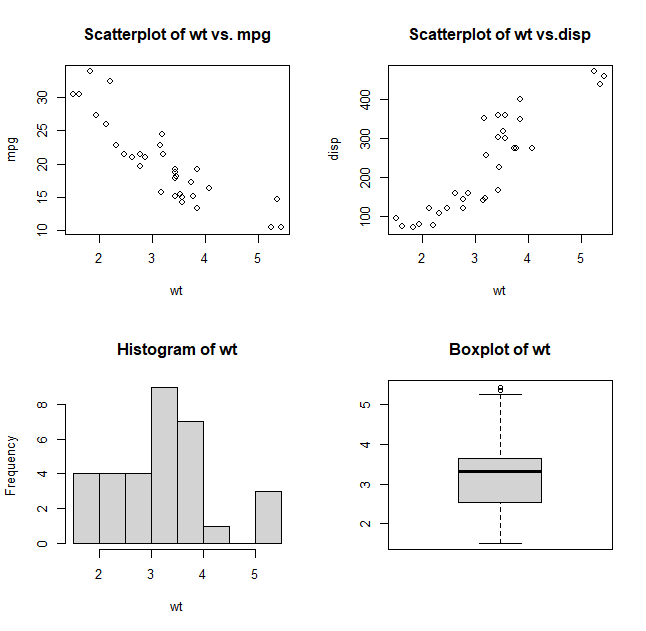
> plot(wt,mpg,main = "Scatterplot of wt vs. mpg")

> plot(wt,disp,main="Scatterplot of wt vs.disp")

> hist(wt,main = "Histogram of wt")

> boxplot(wt,main="Boxplot of wt")

> par(opar)

> detach(mtcars)

Ⅱ> attach(mtcars)

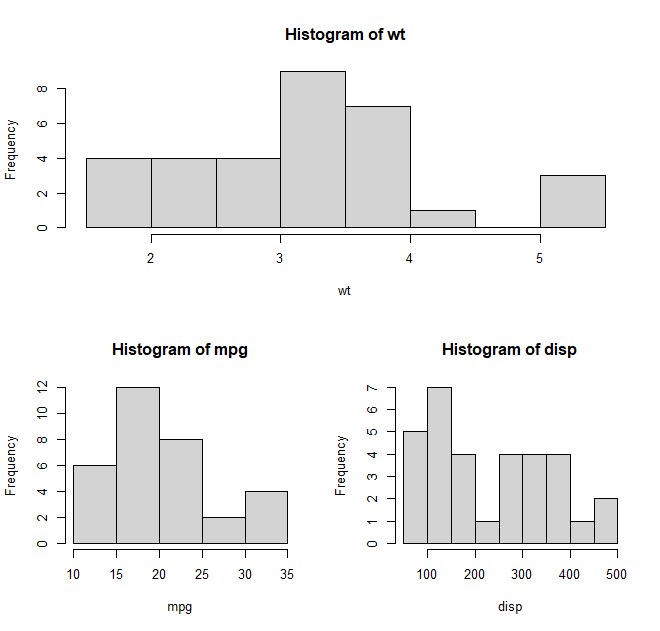
> layout(matrix(c(1,1,2,3),2,2,byrow = TRUE))#第一幅图位于第一行，第二三幅图位于第二行

> hist(wt)

> hist(mpg)

> hist(disp)

> detach(mtcars)

Ⅲ> attach(mtcars)

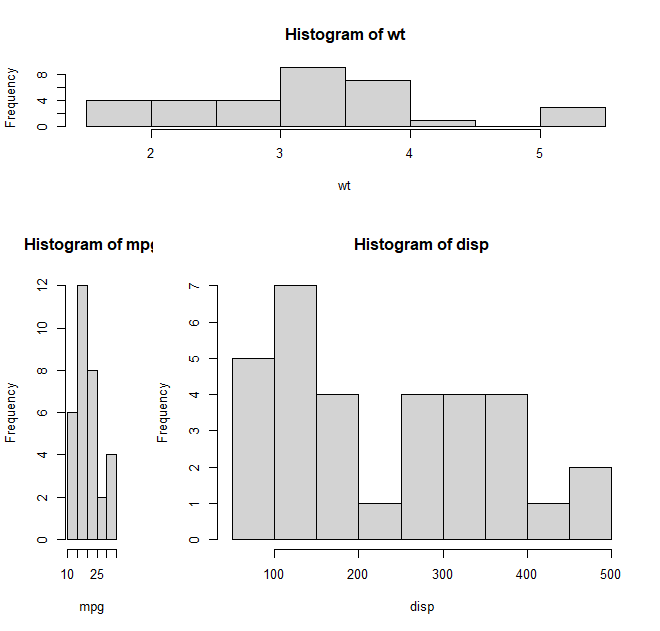
> layout(matrix(c(1,1,2,3),2,2,byrow = TRUE),widths = c(3,10),heights = c(1,2))#widths表示各列宽度，heights表示各行高度

> hist(wt)

> hist(mpg)

> hist(disp)

> detach(mtcars)

> opar<-par(no.readonly = TRUE)

> par(fig=c(0,0.8,0,0.8))#fig=c(x1,x2,y1,y2)左下角坐标为0，0

> plot(mtcars$wt,mtcars$mpg,xlab="Miles per Gallon",ylab = "Car Weight")

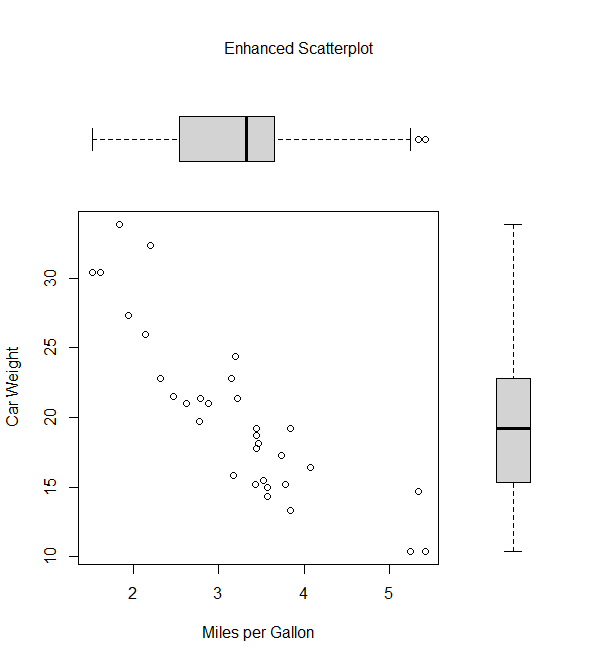
> par(fig=c(0,0.8,0.55,1),new=TRUE)

> boxplot(mtcars$wt,horizontal = TRUE,axes=FALSE)# horizontal = TRUE水平放置

> par(fig=c(0.65,1,0,0.8),new=TRUE)# 公用纵坐标轴,可以把两个变量画在同一个坐标上

> boxplot(mtcars$mpg,axes=FALSE)# axes=FALSE表示图形没有坐标轴

> mtext("Enhanced Scatterplot",side=3,outer = TRUE,line=-3)# outer = TRUE表示在边框外，false在边框内

> par(opar)

第四章 基本数据管理

4.1.示例：建立领导行为的性别差异数据

> manager<-c(1,2,3,4,5)

> date<-c("10/24/08","10/28/08","10/1/08","10/12/08","5/1/09")

> country<-c("US","US","UK","UK","UK")

> gender<-c("M","F","F","M","F")

> age<-c(32,45,25,39,99)

> q1<-c(5,3,3,3,2)

> q2<-c(4,5,5,3,2)

> q3<-c(5,2,5,4,1)

> q4<-c(5,5,5,NA,2)

> q5<-c(5,5,2,NA,1)

> leadership<-data.frame(manager,date,country,gender,age,q1,q2,q3,q4,q5,stringsAsFactors = FALSE)# stringsAsFactors = FALSE不变成属性数据，按字符串读入

4.2创建新变量的三种方法（可以将创建变量整合到数据框中

①> mydata<-data.frame(x1=c(2,2,6,4),x2=c(3,4,2,8))

> mydata$sumx<-mydata$x1+mydata$x2

②> attach(mydata)

> mydata$sumx<-x1+x2

> mydata$meanx<-(x1+x2)/2

> detach(mydata)

③> mydata<-transform(mydata,sumx=x1+x2,meanx=(x1+x2)/2)

4.3变量的重编码

定义年龄老、年轻

①> leadership$age[leadership$age==99]<-NA

> leadership<-within(leadership,{

+ agecat<-NA

+ agecat[age>75]<-"Elder"

+ agecat[age>=55&age<=75]<-"Middle Aged"

+ agecat[age<55]<-"Young"})

②> leadership$agecat[leadership$age>75]<-"Elder"

> leadership$agecat[leadership$age>=55&leadership$age<=75]<-"Middle Aged"

> leadership$agecat[leadership$age<55]<-"Young"

4.4 变量的重命名

①> fix(leadership)#然后点击要修改的变量名

②> names(leadership)[2]<-"testDate"#把第二个变量名更改为testdate

>names(leadership)[6:10]<-c("item1","item2","item3","item4","item5")#将q1-q5重命名为item1-item5

③> install.packages("plyr")#安装此包使用rename函数

> library(plyr)

>leadership<-rename(leadership,c(manager="managerID",date="testDate"))

4.5 缺失值

> y<-c(1,2,3,NA)

> is.na(y)#返回[1] FALSE FALSE FALSE TRUE（是否缺失值）

> is.na(leadership[,6:10])#输出

item1 item2 item3 item4 item5

1 FALSE FALSE FALSE FALSE FALSE

2 FALSE FALSE FALSE FALSE FALSE

3 FALSE FALSE FALSE FALSE FALSE

4 FALSE FALSE FALSE TRUE TRUE

5 FALSE FALSE FALSE FALSE FALSE

正无穷Inf 负无穷-Inf 不是一个数NaN

4.5.1重编码某些值为缺失值

> leadership$age[leadership$age==99]<-NA

4.5.2 在分析中排除缺失值

错误：

> x<-c(1,2,NA,3)

> y<-x[1]+x[2]+x[3]+x[4]

> z<-sum(x)

> z#输出[1] NA

修改为：

> x<-c(1,2,NA,3)

> y<-sum(x,na.rm = TRUE)#使用na.rm = TRUE排除缺失值

> y#输出6

#使用help（）查看具体如何处理缺失值

> newdata<-na.omit(leadership)#使用na.omit移除所有缺失值的行

> newdata#移除了所有含缺失值的行

4.6 日期值

> mydates<-as.Date(c("2007-06-22","2004-02-13"))#输入日期格式，注意默认格式 年-月-日

> strDates<-c("01/05/1965","08/16/1975")

> dates<-as.Date(strDates,"%m/%d/%Y")#%m月份、%d数字表示的日期、%Y四位数的年份

> dates#输出[1] "1965-01-05" "1975-08-16"

> myformat<-"%m%d%y"#指定格式读取字符型变量

>leadership$testDate<-as.Date(leadership$testDate,myformat)

> Sys.Date()#返回当天日期

> date()#返回当前日期和时间

> today<-Sys.Date()

> format(today,format="%B %d %Y")#输出[1] "七月 27 2021"（B大写月份 d 天数数字 Y四位年份）

> format(today,format="%A")#输出星期二 （A 星期几 汉字）

> startdate<-as.Date("2004-02-13")

> endDate<-as.Date("2011-01-22")

> days<-endDate-startdate

> days#输出Time difference of 2535 days，可用R做1970-1-1后的天数减法

> today<-Sys.Date()

> dob<-as.Date("1956-10-12")

> difftime(today,dob,units="weeks")#输出Time difference of 3380.571 weeks，可以检测两天之间相差了几周

4.6.1将日期转化为字符型变量

> strDates<-as.character(dates)

更多> help("as.Date")

> help("strftime")#format函数

> help("ISOdatetime")#日期和时间格式知识

4.7类型转换

> a<-c(1,2,3)

> is.numeric(a)#判定是否数值型，输出TRUE

> is.vector(a)#判定是否向量，输出TRUE

> a<-as.character(a)#将a转化为字符型

> is.numeric(a)#输出FALSE

> is.vector(a)#输出TRUE

> is.character(a)#输出TRUE

4.8 数据排序

实现分类升序oder

> attach(leadership)

> newdata<-leadership[order(gender,age),]#先男女分开再按年龄升序，降序只需要由age改为-age

> detach(leadership)

4.9数据集的合并

4.9.1 向数据框添加列 merge

>total<-merge(dataframeA,dataframeB,by=c("ID","Country"))#通过ID以及Country使两个数据框合并

4.9.2 向数据框添加行 rbind

> total<-rbind(dataframeA,dtaframeB)#两个数据框必须拥有相同的变量，不过他们的顺序不一定相同；如没有，可以删除多的，或添加少的并赋值为NA

4.10数据集取子集

4.10.1选入（保留）变量

选取leadership数据集中6-10列的两种方法，输出结果相同

1. > newdata<-leadership[,c(6:10)]
2. > myvars<-c("item1","item2","item3","item4","item5")

#等价于> myvars<-paste("item",1:5,sep = "")

> newdata<-leadership[myvars]

4.10.2剔除（丢弃）变量

①> myvars<-names(leadership)%in%c("item3","item4")#使item3，4值为TRUE

> newdata<-leadership[!myvars]#剔除了item3 4两列；！为非运算符，逻辑为非TRUE的值

②> newdata<-leadership[c(-8,-9)]#与①等价方式

③>leadership$item3<-leadership$item4<-NULL

4.10.3选入观测

①>newdata<-leadership[leadership$gender=="M"&leadership$age>30,]#选取30岁以上的男性为新数据集

②> attach(leadership)

> newdata<-leadership[gender=="M"&age>30,]#达成结果同上

> detach(leadership)

③选取选定日期范围内的人

>leadership$testDate<-as.Date(leadership$testDate,"%m/%d/%y")

> startdate<-as.Date("2009-01-01")

> endDate<-as.Date("2009-10-31")

>newdata<-leadership[which(leadership$testDate>=startdate&leadership$testDate<=endDate),]

4.10.4 subset()函数

①> newdata<-subset(leadership,age>=35|age<24,select = c(q1,q2,q3,q4))#挑选leadership中24到35岁的人q1q2q3q4组成newdata表

②> newdata<-subset(leadership,gender=="M"&age>25,select = gender:q4)#选择年龄大于25岁的男性并保留变量gender到q4组成newdata

4.10.5随机抽样

> mysample<-leadership[sample(1:nrow(leadership),3,replace = FALSE)]#一个要从中抽样的元素组成的向量，行不变；要抽取的元素数量；无放回抽样

4.11使用SQL语句操作数据框

> install.packages("sqldf")

> library(sqldf)

①> newdf<-sqldf("select\*from mtcars where carb=1 order by mpg",row.names = TRUE)#newdf保留mtcars中所有使用carb的车型，按照mpg对车型进行了升序排序；row.names = TRUE表示将原始数据框的行名延续到了新数据框中

②> sqldf("select avg(mpg) as avg\_mpg,avg(disp) as avg\_disp,gear from mtcars where cyl in(4,6) group by gear")#输出四缸和六缸车型每一gear水平的mpg和disp的平均值

第五章 高级数据管理

5.1一个数据处理难题

5.2 数值和字符处理函数

5.2.1数学函数

> abs(x)#x的绝对值

> sqrt(x)#x的平方根

> ceiling(x)#不小于x的最小整数

> floor(x)#不大于x的最大整数

> trunc(x)#向0方向截取x的整数部分；5.99->5

> round(x,digits = 2)#将x设为指定2位的小数3.48

> signif(x,digits = 2)#将x设为有效数字为两位的小数 3.5

> cos(x)#还可以求sin tan（无cot） 以及cosh(x)等双曲余弦

>acos(x)#还有asin atan 反函数 以及acosh(x)等反双曲余弦

> log(x,base=2)#对x取以2为底的对数，log(x)为自然对数，log10（x）为常用对数

> exp(x)#指数函数

#注> sqrt(c(4,16,25))#输出2 4 5（可以直接输出一连串数字）

5.2.2 统计函数

①> x<-c(2,4,6,8)

> z<-mean(x,trim = 0.05,na.rm = TRUE)#丢弃了最大和最小的0.05的数，和所有缺失值后算算数平均数

> z#输出5

②median(x)#中位数

③sd(x)#标准差

④var(x)#方差

⑤mad(x)#绝对中位差：用原数据减去中位数后得到的新数据的绝对值的中位数乘上比例因子1.4826

⑥quantile(x,c(y,z))#求x在y和z处的分位数，y、z均在0-1之间

⑦range()#求值域，x<-c(1,2,3,4)则返回c(1,4); diff(range(x))返回3

⑧sum(x)#求和

⑨diff(x,lag=n)#滞后差分，默认lag=1滞后一项，输入x<-c(1,5,23,29)返回c(4,18,6)

⑩min(x)#求最小值；max()最大值

11 scale(x,center=TRUE,scale=TRUE)#为x按列进行中心化或标准化

5.2.3 概率函数

①正态分布中：dnorm 密度函数、pnorm分布函数、qnorm分位数函数、rnorm随机数生成函数

> x<-pretty(c(-3,3),30)# 创建美观的分割点。选取30+1等间距的取整数，将连续变量-3-3分割为30个区间

> y<-dnorm(x)#x的密度函数

> plot(x,y,type = "l",xlab = "Normal Deviate",ylab = "Density",yaxs="i")#正态离差

> pnorm(1.96)#求得位于1.96左侧曲线下面积为0.975

> qnorm(.9,mean=500,sd=100)#均值为500，标准差为100的正态分布的0.9分位点值为多少

> rnorm(50,mean = 50,sd=10)#随机生成50个均值为50，标准差为10的正态随机数

生成服从正态分布的0-1间伪随机数：

> runif(6)#可以通过set.seed(1234)使得生成结果可以重现

②生成多元正态多元数据

> library(MASS)

> options(digits = 3)

> set.seed(1234)

> mean<-c(230.7,146.7,3.6)

> sigma<-matrix(c(15360.8,6721.2,-47.1,6721.2,4700.9,-16.5,-47.1,-16.5,0.3),nrow = 3,ncol = 3)#方差-协方差矩阵

> mydata<-mvrnorm(500,mean,sigma)

> mydata<-as.data.frame(mydata)

> names(mydata)<-c("y","x1","x2")#显示对象中各个成分名为

> dim(mydata)# dim 有一个 data.frames 方法，它返回 x 和 x 的 row.names 属性的长度（分别作为行数和列数）返回500，3

> head(mydata,n=10)#我猜是从mydata里随机抽10个样

5.2.4 字符处理函数

①nchar(x[3])#返回x中第三个元素的字符数量

②substr(x,2,4)#返回x中第二个字符到第四个字符之间的字符

substr(x,2,4)<-“22222”#将x中第二个到第四个字符变成2

③grep() sub()#看不懂，需要使用时help

④> y<-strsplit("abc","")#分割元素，返回y为[1] "a" "b" "c"；fixed=FALSE则搜索一个正则表达式（正则表达式regular expression是对[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6" \t "_blank)串操作的一种逻辑公式，就是用事先定义好的一些特定字符、及这些特定字符的组合，组成一个“规则字符串”，这个“规则字符串”用来表达对字符串的一种过滤逻辑。）；TRUE为文本字符串

⑤> paste("x",1:3,sep = " ")#输出[1] "x 1" "x 2" "x 3"，sep分隔符，paste函数连接字符串

> paste("x",1:3,sep = "M")#输出[1] "xM1" "xM2" "xM3"

> paste("Today is",date())#输出[1] "Today is Fri Jul 30 16:42:44 2021"

⑥> toupper("abc")#大写转换，返回值“ABC”

⑦> tolower("ABC")#小写转换，返回值“abc”

5.2.5其他实用函数

①> x<-c(2,5,6,9)

> length(x)#输出4，计量x长度

②> seq(1,10,2)#输出1 3 5 7 9，生成一个序列从1开始，10为终点，两两间相差2

③rep(x,n)#将x重复n次

> y<-rep(1:3,2)#输出1 2 3 1 2 3

④cut(x,n)#使得连续型变量x分割为有着n个水平的因子

> cut(c(2,5,6,9),2,ordered\_result = TRUE)# 输出[1] (1.99,5.5] (1.99,5.5] (5.5,9.01] (5.5,9.01] Levels: (1.99,5.5] < (5.5,9.01]；创建一个有序型因子

⑤pretty(x,n)#创建n+1个等间距取整值，将连续型变量x分割为n个区间

> pretty(c(2,5,6,9),2)#输出[1] 0 5 10

⑥> cat("Hello","Jane","\n")#输出Hello Jane 连接两个对象

>at(“文本1“，”文本2”, “\n”,file=”文件位置”,append=FALSE)#\n 新行 \t制表符\’单引号\b退格

5.2.6将函数应用于矩阵和数据框

①> a<-5

> sqrt(a)#输出2.24

> b<-c(1.243,5.654,2.99)

> round(b)#round后不接digits=即看作取整，输出1 6 3

> c<-matrix(runif(12),nrow=3)

> c#输出一个三行四列的矩阵

> log(c)#对矩阵中每一个元素取自然对数

> mean(c)#对矩阵中所有的值取平均值

②mydata<-matrix(rnorm(30),nrow = 6)#生成 30个元素六行的矩阵

> mydata

> apply(mydata,1,mean)#计算每行的平均值

> apply(mydata,2,mean)#计算每列的平均值

> apply(mydata,2,mean,trim=0.2)#计算每列的截尾平均值（去掉最大和最小的0.2）

5.3数据处理难题的一套解决方法

> options(digits=2)#限定输出小数点后数字位数为2

> Students<-c("John Davis","Angela Williams","Bullwinkle Moose","David Jones","Janice Markhammer","Cheryl Cushing","Reuven Ytzrhak","Greg Knox","Joel England","Mary Rayburn")

> Math<-c(502,600,412,358,495,512,410,625,573,522)

> Science<-c(95,99,80,82,75,85,80,95,89,86)

> English<-c(25,22,18,15,20,28,15,30,27,18)

> roster<-data.frame(Students,Math,Science,English,stringsAsFactors = FALSE)

> z<-scale(roster[,2:4])#标准化2-4列，每科考试都用单位标准差来表示

> score<-apply(z,1,mean)#计算每行标准差的平均值（每个人三科考试标准差均值）

> roster<-cbind(roster,score)#按列合并矩阵score放在最后一列

> y<-quantile(score,c(.8,.6,.4,.2))#得出学生综合得分的百分位数

> roster$grade[score>=y[1]]<-"A"

> roster$grade[score<y[1]&score>=y[2]]<-"B"

> roster$grade[score<y[2]&score>=y[3]]<-"c"

> roster$grade[score<y[3]&score>=y[4]]<-"D"

> roster$grade[score<y[4]]<-"F"#建立新的grade

> name<-strsplit((roster$Students)," ")#以空格为界把学生姓氏名字拆开

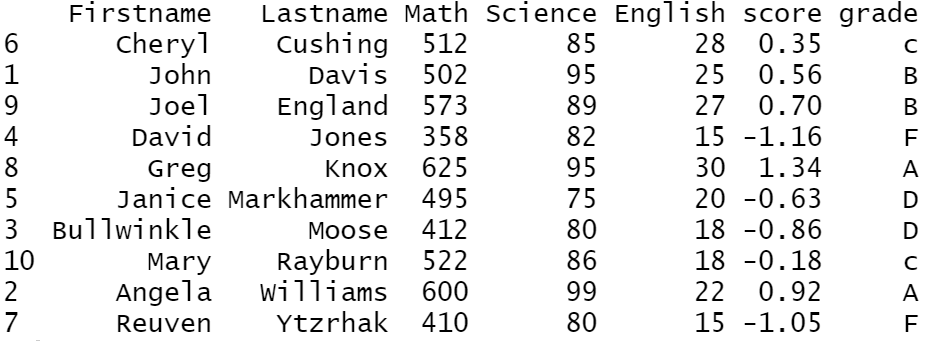
> Lastname<-sapply(name,"[",2)#提取每个成分第二个元素，可以提取某个对象一部分的函数

> Firstname<-sapply(name,"[",1)#提取第一部分

> roster<-cbind(Firstname,Lastname,roster[,-1])#按列将姓名合并入roster前一列

> roster<-roster[order(Lastname,Firstname),]#按姓氏和名字对数据集排序 不要漏逗号

> roster

#输出

5.4 控制流

* 语句statement:一条单独的R语句或一组复合语句
* 条件cond是一条最终被解析为真TRUE或假FALSE的表达式
* 表达式expr是一条数值或字符串的求值语句
* 序列seq是一个数值或字符串序列

5.4.1 重复和循环

①for

> for (i in 1:10)

+ print("Hello")#输出十个“Hello”

②while（尽量少用）

> i<-10

> while(i>0)

+ {print("Hello");i<-i-1}#输出十个“Hello”

5.4.2条件执行

①if-else

A.> grade<-c(199,100,84)

> if(is.character(grade)) grade<-as.factor(grade)#如果grade是一个字符向量，将其转化为因子

B.> if(!is.factor(grade)) grade<-as.factor(grade) else print("Grade already is a factor")#如果grade不是一个字符向量，将其转化为因子，否则输出一段信息

②if else

> ifelse(x>0.5,print("Passed"),print("Failed"))#x>0.5输出PASSED，不然输出failed

③switch

> feelings<-c("sad","afraid")

> for (i in feelings) print(switch(i,happy="I am glad you are happy",afraid="There is nothing to fear",sad="Cheer up",angry="Calm down now"))#输出[1] "Cheer up"

[1] "There is nothing to fear"

5.5用户自编函数

①第一步：> mystats<-function(x,parametric=TRUE,print=FALSE){

+ if(parametric){

+ center<-mean(x);spread<-sd(x)

+ } else{

+ center<-median(x);spread<-mad(x)

+ }

+ if(print&parametric){

+ cat("Mean=",center,"\n","SD=",spread,"\n")

+ }else if(print&!parametric){

+ cat("Median=",center,"\n","MAD",spread,"\n")

+ }

+ result<-list(center=center,spread=spread)

+ return(result)

+ }

第二部：> set.seed(1234)

> x<-rnorm(500)

> y<-mystats(x)

> y#输出$center[1] 0.001838821平均值$spread[1] 1.034814标准差

> y<-mystats(x,parametric = FALSE,print = TRUE)#得到结果Median= -0.02070734 MAD 1.000984

②第一步> mydate<-function(type="long"){

+ switch(type,

+ long=format(Sys.time(),"%A%B%d%Y"),

+ short=format(Sys.time(),"%m-%d-%y"),

+ cat(type,"is not a recognized type\n")#输入格式不匹配时是好方法

+ )

+ }

> rm(mydata)

第二部：> mydate("long")#输出[1] "星期六七月312021"

> mydate("short")#输出[1] "07-31-21"

> mydate()#输出[1] "星期六七月312021"

> mydate("medium")#输出medium is not a recognized type

5.6整合aggregate与重构reshape

5.6.1转置

> cars<-mtcars[1:5,1:4]#生成五行四列的矩阵（出去行列标题）

> t(cars)#行列转置

5.6.2整合数据

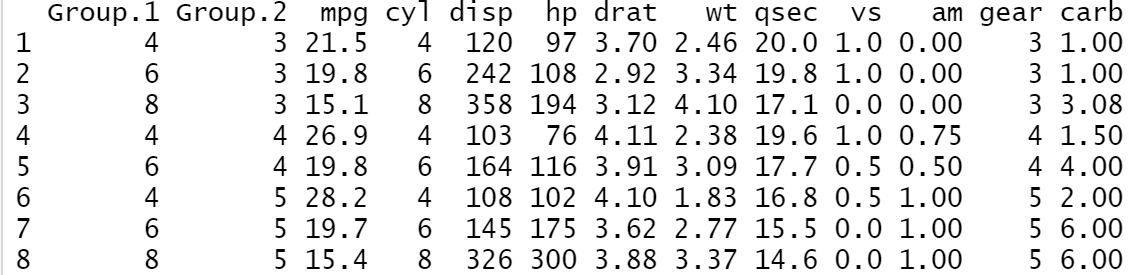
> options(digits = 3)

> attach(mtcars)

> aggdata<-aggregate(mtcars,by=list(cyl,gear),FUN=mean,na.rm=TRUE)

#可以自定义名称比如by=list(Group.cyl=cyl,Group.gear=gear)

> aggdata



#group1表示气缸数（4，6，8）；group2表示档位数（3、4、5）;拥有4气缸3挡位的车型每加仑汽油行驶英里数（mpg）均值为21.5

5.6.3 reshape2包

> install.packages("reshape2")

> library(reshape2)

①融合melt

> mydata<-data.frame(ID=c(1,1,2,2),

+ Time=c(1,2,1,2),

+ X1=c(5,3,6,2),

+ X2=c(6,5,1,4))#创建mydata表，不要用fix（）创建表

> md<-melt(mydata,id=c("ID","Time"))#按照ID和Time把X1 X2并入一列为变量

②重铸decast

第一步：建立数据集

> x<-data.frame(id=1:6,

+ name=c("wang","zhang","li","chen","zhao","song"),

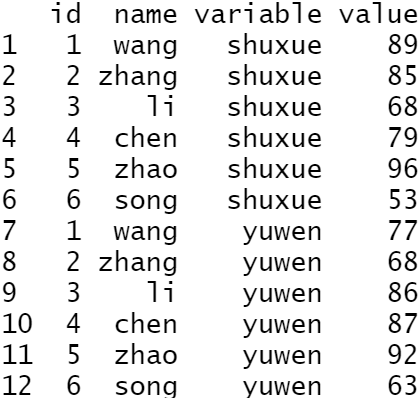
+ shuxue=c(89,85,68,79,96,53),

+ yuwen=c(77,68,86,87,92,63))



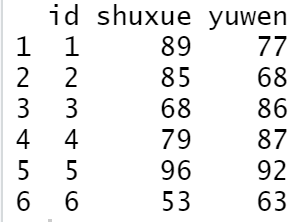
第二部：合并数学语文为一列

> x1<-melt(x,id=c("id","name"))



第三步：重铸（删除变量，保留id）

> x2<- dcast(x1,id~variable)



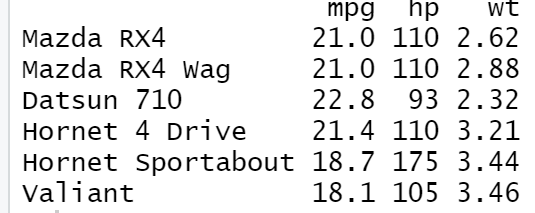
第七章 基本统计分析

参考文献McCall2000 Kirk2007

7.1 描述性统计分析

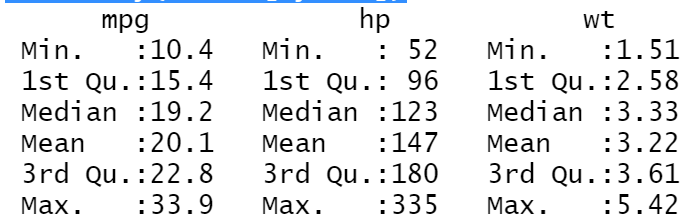
> myvars<-c("mpg","hp","wt")

> head(mtcars[myvars])#列出表格前几列



7.1.1方法云集

> summary(mtcars[myvars])#提供最小、最大值、四分位数（1st Qu/3rd Qu）、数值型变量的均值以及中位数median

  
> mystats<-function(x,na.omit=FALSE)#不返回补齐缺失值

{

+ if(na.omit)# 返回删除NA后的向量a

+ x<-x[!is.na(x)]#x不是缺失值时

+ m<-mean(x)

+ n<-length(x)

+ s<-sd(x)

+ skew<-sum((x-m)^3/s^3)/n#偏斜度

+ kurt<-sum((x-m)^4/s^4)/n-3#峭度

+ return(c(n=n,mean=m,stdev=s,skew=skew,kurtosis=kurt))

+ }

> sapply(mtcars[myvars],mystats)#计算任意描述性统计量，函数

7.1.2 更多方法

①> library(Hmisc)

> myvars<-c("mpg","hp","wt")

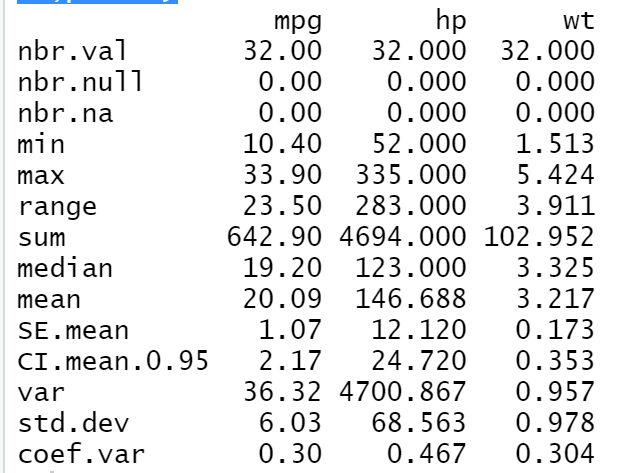
> describe(mtcars[myvars])#返回变量和观测的数量、缺失值missing和唯一值的数目distinct、平均值mean、分位数、五个最小值和最大值。（info Gmd 不懂什么意思）

②> install.packages("pastecs")

> library(pastecs)

> myvars<-c("mpg","hp","wt")

> stat.desc(mtcars[myvars],basic = TRUE,desc = TRUE,norm = FALSE,p=0.95) #（默认值）计算所有值、空值、缺失值的数量、最小值最大值、值域、总和；（默认值）中位数、平均数、平均数的标准误(SE.mean)、平均数置信度为95%的置信区间(CI.mean.0.95)、方差var、标准差(std.dev)以及变异系数(coef.var)；（不是默认的），返回正态分布统计量：偏斜度skewness和峭度kurtosis（及其统计显著程度.2SE）和Shapiro-Wilk正态检验结果normtest.W/normtest.p



③> install.packages("psych")

> library(psych)

> myvars<-c("mpg","hp","wt")

> describe(mtcars[myvars])#分析各种常用统计量

7.1.3分组计算描述性统计量

①无法返回多个统计值

>

aggregate(mtcars[myvars],by=list(am=mtcars$am),mean)#命名新列，合并am列与其他三列，并且不同am分别计算其他几列平均值

> aggregate(mtcars[myvars],by=list(am=mtcars$am),sd)#标准差

②> dstats<-function(x)sapply(x,mystats)#拼接x和mystats（已在前面写了）

> myvars<-c("mpg","hp","wt")

> by(mtcars[myvars],mtcars$am,dstats)#得到每一水平am的统计值

7.1.4 分组计算的扩展

①> install.packages("doBy")

> library(doBy)

> summaryBy(mpg+hp+wt~am,data=mtcars,FUN=mystats)#计算各组的统计变量

②> library(psych)

> myvars<-c("mpg","hp","wt")

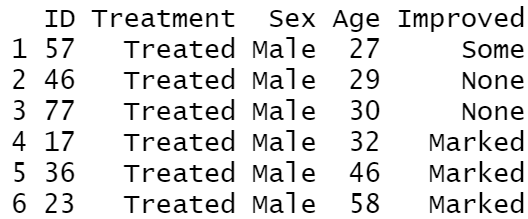
> describeBy(mtcars[myvars],list(am=mtcars$am))#不能指定任意函数

7.2频数表和列联表

> install.packages("vcd")

> library(vcd)

> head(Arthritis)#载入来自vcd包的Arthritis前几行



7.2.1 生成频数表

①一维列联表

> mytable<-with(Arthritis,table(Improved))#使用N个类别型变量创建一个N维列联表

> mytable#输出Improved

None Some Marked

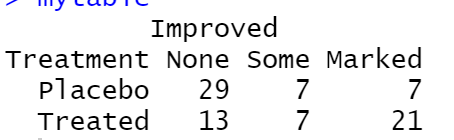
42 14 28

> prop.table(mytable)#使得以上输出频数转化为频率

> prop.table(mytable)\*100#将频率以百分比形式显示

②二维列联表

> mytable<-xtabs(~Treatment+Improved,data = Arthritis)#根据一个公式和一个矩阵或数据框创建一个N维列联表要交差分类的变量在~右侧；Treated、Placebo为treatment的等级。improved是列，treatment是行；none some marked是improved等级



> margin.table(mytable,1)#1表示mytable中第一个变量；依照要求计算某条目的和

> prop.table(mytable,1)#将表中条目表示为分数形式

> addmargins(mytable)#添加最后一列计算每行的和，最后一行计算每列的和

> addmargins(prop.table(mytable))#使频数变为频率

> addmargins(prop.table(mytable,1),2)#加入各行的和

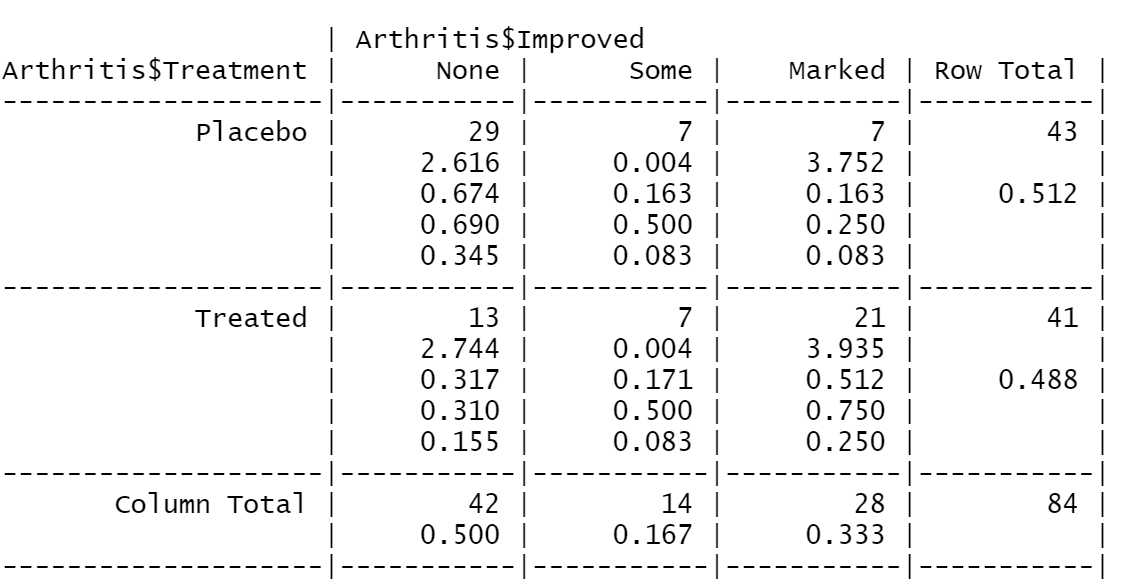
> addmargins(prop.table(mytable,2),1)#加入每列的和

* 另一种创建二维表的方式#详情> help("CrossTable")

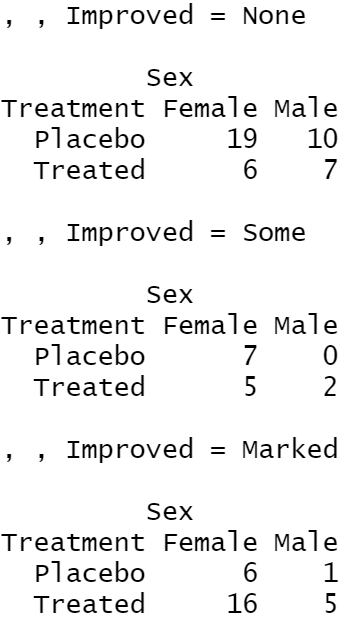
> install.packages("gmodels")

> library(gmodels)

> CrossTable(Arthritis$Treatment,Arthritis$Improved)

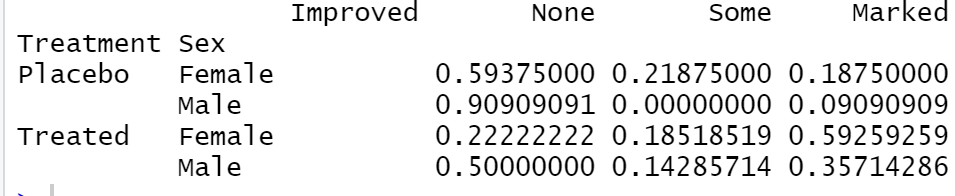


③多维列表

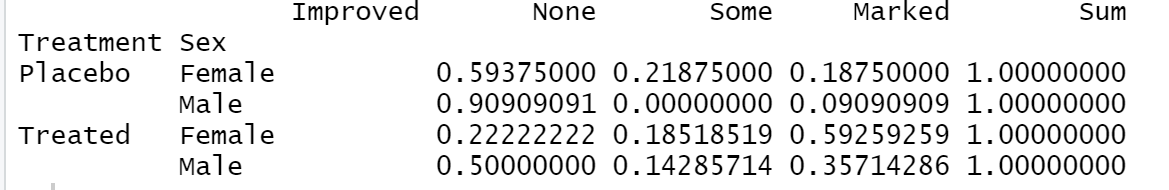
> mytable<-xtabs(~Treatment+Sex+Improved,data = Arthritis)#三维表格

> margin.table(mytable,1)#数字表示mytable中第一个变量treatment，2则为Sex，3为Improved；c(1,3)生成第一个和第三个的二维联表

>ftable(prop.table(mytable,c(1,2)))#创建一个紧凑的平铺式列表；第一第二个变量组成一个联表treatment和sex分类，每一小类占不同improved状态的比例



> ftable(addmargins(prop.table(mytable,c(1,2)),3))



> ftable(addmargins(prop.table(mytable,c(1,2)),3))\*100#可得到百分比表

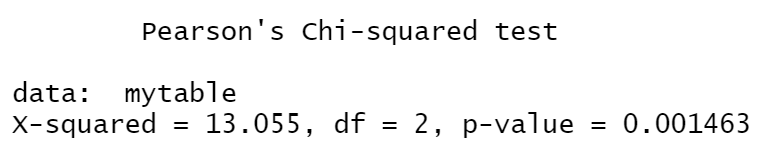
7.2.2[独立性检验](https://baike.baidu.com/item/%E7%8B%AC%E7%AB%8B%E6%80%A7%E6%A3%80%E9%AA%8C/4031921?fr=aladdin)书P125

1. 卡方独立性检验chisq.test()对二维表的行变量和列变量进行卡方独立性检验

> library(vcd)

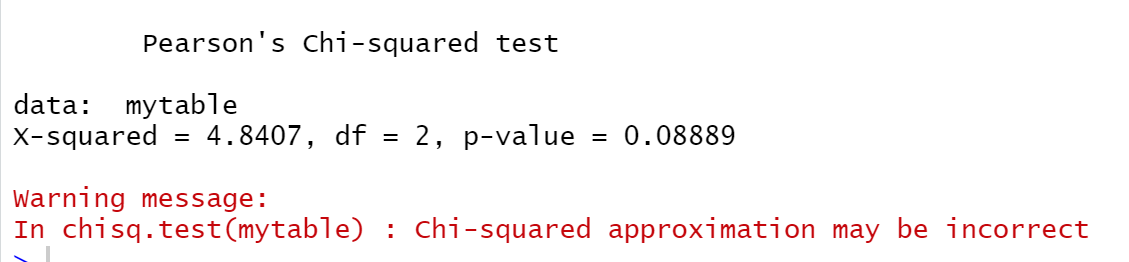
> mytable<-xtabs(~Treatment+Improved,data = Arthritis)

> chisq.test(mytable)#df是自由度；p值小于0.05或0.01不独立



> mytable<-xtabs(~Improved+Sex,data = Arthritis)

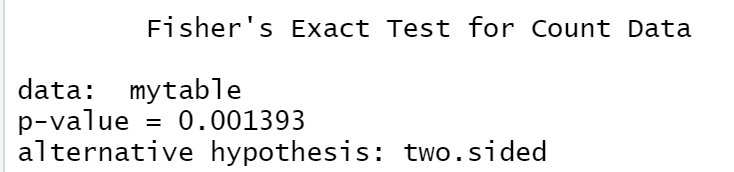
> chisq.test(mytable)#p值大于0.05表示两个变量相互独立但由于概率过小没有足够理由说明治疗结果和性别之间不是独立的。



1. Fisher精确检验（能在2\*2以上的二维联表上用，不包括2\*2）

> mytable<-xtabs(~Treatment+Improved,data=Arthritis)

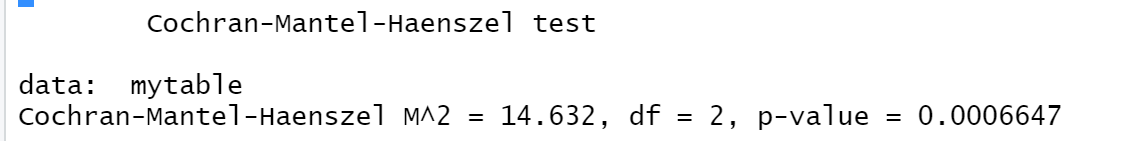
> fisher.test(mytable)#输出结果不独立p<0.05



3.Cochran-Mantel-Haenszel检验

> mytable<-xtabs(~Treatment+Improved+Sex,data = Arthritis)

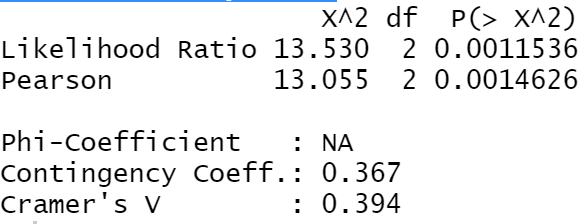
> mantelhaen.test(mytable)#检验治疗情况和改善情况在性别的每一水平下是否独立（结果看来不独立p<0.05）



7.2.3相关性的度量

> library(vcd)

> mytable<-xtabs(~Treatment+Improved,data=Arthritis)

>assocstats(mytable) 

7.3相关

7.3.1相关的类型：Pearson Spearman Kendall相关

> states<-state.x77[,1:6]

> cov(states)#计算协方差

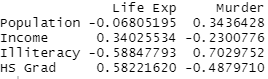
> cor(states)#没指定method计算Pearson积差相关系数

> cor(states,method = "spearman")#计算了Spearson等级相关系数

> x<-states[,c("Population","Income","Illiteracy","HS Grad")]

> y<-states[,c("Life Exp","Murder")]

> cor(x,y)#计算相关矩阵，输出结果如下（显著不为0即为相关）



2.偏相关

> install.packages("ggm")

> colnames(states)#输出[1] "Population" "Income" "Illiteracy" "Life Exp"

[5] "Murder" "HS Grad"

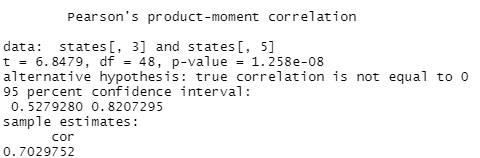
> pcor(c(1,5,2,3,6),cov(states))#偏相关函数；要计算相关系数的变量下标，其余数值表示要排除影响的变量的下标；协方差矩阵；输出0.346,表示人口和谋杀率的相关系数，排除收入、文盲率和高中毕业率的影响时

3.其他类型的相关：polycor包中的hetcor()函数

7.3.2 相关的显著性检验：计算好相关系数以后对它们进行统计显著性检验cor.test(x,y,alternative=“two.side”双侧检验 “less”<0或”greater”>0,method=”pearson” “kendall”或“spearman”)

①> cor.test(states[,3],states[,5])#检验了文化和谋杀率的person相关系数为0的原假设。

假设总体相关度为0，则预计在一千万次中只有少于一次的机会见到0.703这样大的样本相关度，几乎不可能发生，拒绝原假设。



②> library(psych)

> corr.test(states,use = "complete")#对缺失值执行成对删除或成行删除。参数method=取值可为“pearson”、“spearson”或“kendall”。人口数量和高中毕业率的相关系数（-0.10）并不显著地不为0（p=0.5）

③其他显著性检验 help(r.test)

7.3.3相关关系的可视化

7.4 t检验

7.4.1 独立样本的t检验

> library(MASS)

> t.test(Prob~So,data = UScrime)#数值型变量，二分变量 比较南方和非南方各州的监禁概率p<0.001拒绝南方和非南方各州有相同监禁概率的假设

7.4.2非独立样本的t检验：较年轻男性的失业率 较年长男性的失业率 两组数据不独立

> library(MASS)

> sapply(UScrime[c("U1","U2")],function(x)(c(mean=mean(x),sd=sd(x))))

> with(UScrime,t.test(U1,U2,paired=TRUE))#得出均值差异61.5，拒绝年轻男性和年长男性失业率相同的假设；假定组间差异呈正态分布

7.5 组件差异的非参数检验

7.5.1 两组的比较

①若两组数据独立，可以用Wilcoxon秩和检验（Mann-Whitney U检验）来评估观测是否从相同的概率分布中抽得

> wilcox.test(Prob~So,data = UScrime)#得出p<0.001再次拒绝南方各州和非南方各州监禁率相同的假设（p<0.001）

②> sapply(UScrime[c("U1","U2")],median)

> with(UScrime,wilcox.test(U1,U2,paired=TRUE))

7.5.2 多于两组的比较

如果各组间独立：Krusal-Wallis检验

各组不独立：Friedman检验

①> states<-data.frame(state.region,state.x77)

> kruskal.test(Illiteracy~state.region,data=states)#检验结果p<0.001美国四个地区文盲率各不相同

②> source("http://www.statmethods.net/RiA/wmc.txt")#下载wmc函数的R脚本

> states<-data.frame(state.region,state.x77)

> wmc(Illteracy~state.region,data=states,method = "holm")#给出样本量、样本中位数、每组的绝对中位差，西部地区文盲率最低，南部最高；生成六组统计比较，由结果可知南部与其他三个区域有明显差别\*

第八章 回归

8.1回归的多面性

重点：OLS 普通最小二乘回归法：包括简单线性回归、多项式回归和多元线性回归。

8.1.1OLS回归的适用情境

8.2 OLS回归

8.2.1用lm（）拟合回归模型

①回归模型包含一个应变量和一个自变量——简单线性回归

②只有一个预测变量，但同时包含变量的幂（X、X2 、X3 ）——多项式回归

③不止一个预测变量——多元线性回归

例：

> fit<-lm(weight~height,data = women)

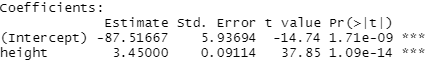
> summary(fit)

> fitted(fit)

> residuals(fit)

> plot(women$height,women$weight,xlab = "Height(in inches)",ylab = "Weight(in pounds)")

> abline(fit)



所以，预测等式为Weight=-87.52+3.45\*Height

8.2.3 多项式回归

可以通过添加一个二次项（X2 ）来提高回归精度。

> fit2<-lm(weight~height+I(height^2),data=women)#表示向预测等式添加一个身高的平方项；I函数将括号的内容看作R的一个常规表达式，^有特殊含义会调用不需要的东西

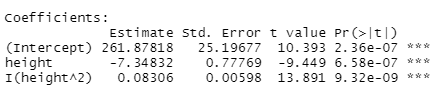
> summary(fit2)

> plot(women$height,women$weight,xlab = "Height(in inches)",ylab = "Weight(in lbs)")

> lines(women$height,fitted(fit2))

> plot(women$height,women$weight,xlab = "Height(in inches)",ylab = "Weight(in lbs)")

> lines(women$height,fitted(fit2))



新的预测等式：Weight=261.88-7.35\*Height+0.083\*Height2

N次多项式生成一个n-1个弯曲的曲线：例如拟合三次多项式可用：

> fit3<-lm(weight~height+I(height^2)+I(height^3),data=women)#高于三次的多项式没必要

非线性模型用nls（）拟合

\*car包中的scatterplot()函数，绘制二元关系图

> install.packages("car")

> library(car)

> scatterplot(weight~height,data=women,spread=FALSE,smoother.args=list(lty=2),pch=19,main="Women Age 30-39",xlab = "Height(inches)",ylab = "Weight(lbs.)")#删除了残差正负均方根在平滑曲线上的展开和非对称信息；设置平滑拟合loess拟合曲线为虚线；设置点为实心圆

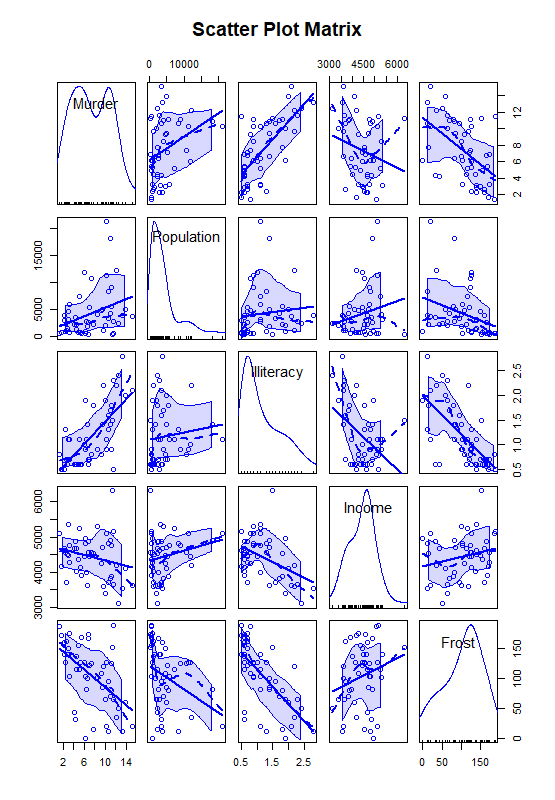
8.2.4 多元线性回归（预测变量不止一个）

例：探究一个州的犯罪率和其他因素的关系，包括人口、文盲率、平均收入和结霜天数（温度在冰点以下的平均天数）

> states<-as.data.frame(state.x77[,c("Murder","Population","Illiteracy","Income","Frost")])

> cor(states)#多元回归分析中第一步最好检查一下变量间的相关性

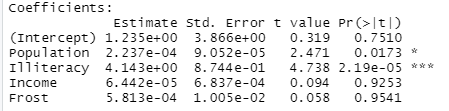
> library(car)

> scatterplotMatrix(states,spread=FALSE,smoother.arge=list(lty=2),main="Scatter Plot Matrix")#默认在非对角线区域绘制变量间的散点图，并添加平滑和线性拟合曲线；对角线绘制每个变量的密度图和轴须图谋杀绿是双峰曲线随人口、文盲率增加而增加，随收入和结霜天数增加而下降。

> states<-as.data.frame(state.x77[,c("Murder","Population","Illiteracy","Income","Frost")])

> fit<-lm(Murder~Population+Illiteracy+Income+Frost,data = states)

> summary(fit)



#文盲率的回归系数是4.14\*10^00=4.14,文盲率上升1%，谋杀率上升4.14%

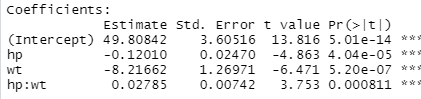
人口的回归系数是2.237\*10^-4

Frost p值为0.954 系数没有显著不为0，控制其他变量不变时，Frost与Murder不呈线性相关。

8.2.5有交互项的的多元线性回归（以mtcars中汽车重量和马力为例）

> fit<-lm(mpg~hp+wt+hp:wt,data=mtcars)# 每加仑燃料所行英里数; 马力；车身重量

> summary(fit)



#Pr马力与车重的交互项是显著的，说明每加仑汽油行驶的里程数与汽车马力的关系依车重不同而不同

#预测mpg的模型：mpg=49.81-0.12\*hp-8.22\*wt+0.03\*hp\*wt

> install.packages("effects")#用图形展示交互项结果

> library(effects)

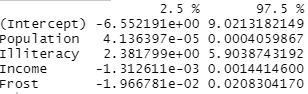
> plot(effect("hp:wt",fit,,list(wt=c(2.2,3.2,4.2))),multiline = TRUE)#表示作图wt分别为三个数值时、hp与mpg的关系。

8.3回归诊断

> plot(effect("hp:wt",fit,,list(wt=c(2.2,3.2,4.2))),multiline = TRUE)

> states<-as.data.frame(state.x77[,c("Murder","Population","Illiteracy","Income","Frost")])

> fit<-lm(Murder~Population+Illiteracy+Income+Frost,data = states)

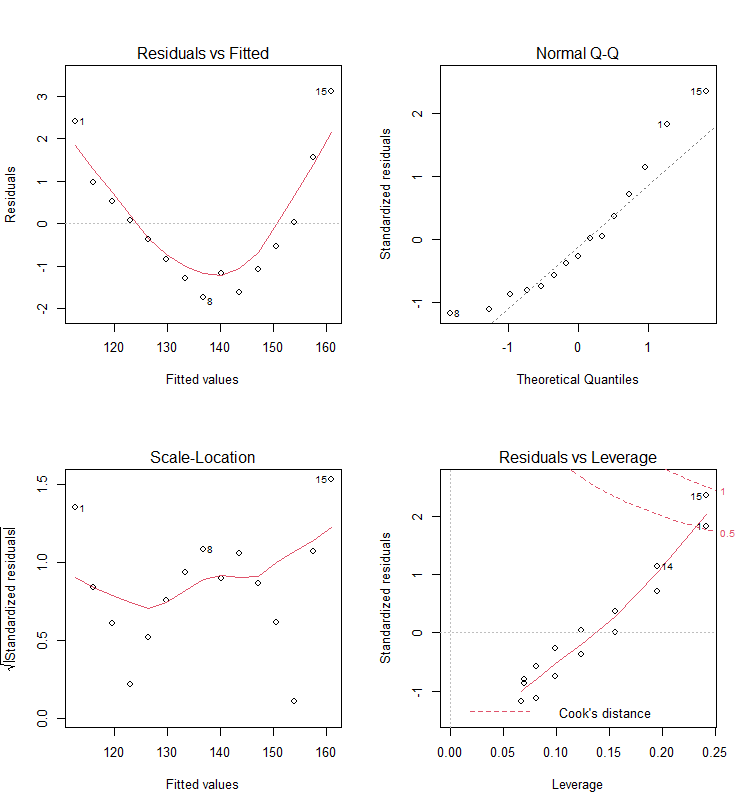
> confint(fit) #结果表示文盲率改变1%，谋杀率就在95%的置信区间【2.38，5.90】中变化。

因为Frost置信区间包含0，故得出结论其他变量不变时，温度的改变与谋杀率无关。

8.3.1标准方法

> fit<-lm(weight~height,data=women)

> par(mfrow=c(2,2))#绘制2\*2的四幅图

> plot(fit)

#右上图：正态性：是在正态分布对应的值下，标准化残差的概率图。如果满足正态假设，图上的点应该落在呈45°角的直线上

#左上：线性：残差图residuals与拟合图fitted：可以清楚得看到一个曲线关系，暗示着可能需要对回归模型加上一个二次项；若因变量与自变量线性相关，则残差值与拟合值没有任何系统关联（除了白噪声，模型应该包含数据中所有的系统方差）

#左下：同方差性：若满足不变方差的假设，那么在位置尺度图中，水平线周围的点应该随机分布，此图满足假设。

#右下：残差与杠杆图：鉴别出离群点、高杠杆值点和强影响点。

离群点：表明拟合回归模型对其预测效果不佳（产生了巨大的正或负残差）

杠杆值：很高表明他是一个异常的预测变量值的组合。

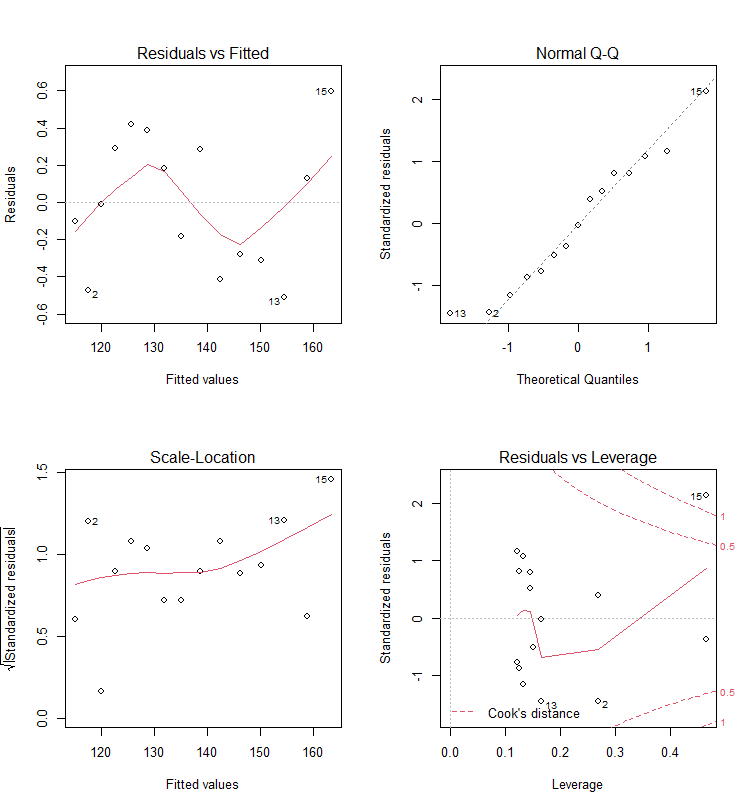
强影响点：表明他对模型参数的估计产生的影响过大，非常不成比例，通过Cook距离统计量来鉴别。

二次拟合的诊断图：

> fit2<-lm(weight~height+I(height^2),data = women)

> par(mfrow=c(2,2))

> plot(fit2)

#二次拟合较为理想，符合了线性假设（左上）、残差正态性（右上除了点13）和同方差性（左下）观测点15看起来是强影响点（Cook距离大）；删除点13和15使模型拟合更好用（但是要小心，因为本该是模型去匹配数据而不是反过来）。

> newfit<-lm(weight~height+I(height^2),data=women[-c(13,15),])#剔除两个点，然后再作图

8.3.2改进的方法（car包）

〇实用函数：

qqPlot() 分位数比较图

durbinWatsonTest()对误差自相关性做durbinWatson检验

crPlots()成分与残差图

ncvTest()对非恒定的误差方差做得分检验

spreadLevelPlot()分散水平检验

outlierTest() Bonferroni离群点检验

avPlots()添加的变量图形

inluencePlot()回归影响图

scatterplot()增强的散点图

scatterplotMatrix()增强的散点图矩阵

vif()方差膨胀因子

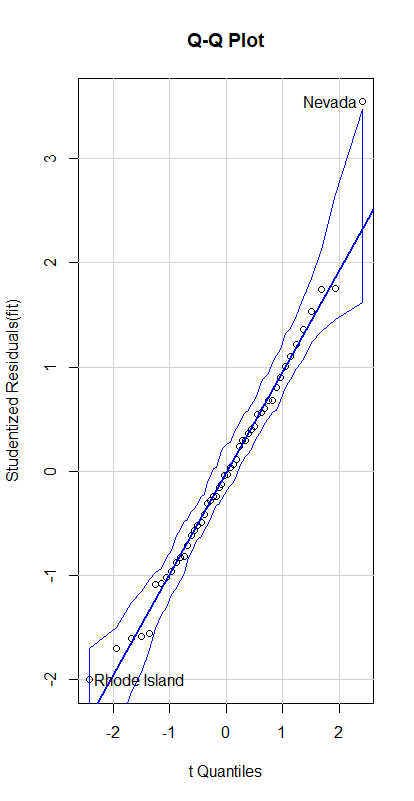
①正态性

> library(car)

> states<-as.data.frame(state.x77[,c("Murder","Population","Illiteracy","Income","Frost")])

> fit<-lm(Murder~Population+Illiteracy+Income+Frost,data=states)

> qqPlot(fit,labels=row.names(states),id.method="identify",simulate = TRUE,main = "Q-Q Plot")#画出了在n-p-1个自由度t分布下的学生化残差(studentized residual学生化残差是残差xi-x均除以它的标准差后得到的数值)n是样本大小，p是回归参数的数目（包括截距项）



除了点Nevada，所有点都离直线很近，且都落在置信区间内，表明正态性好；必须关注Nevada，其有很大正残差值（真实值-预测值），表明该模型低估了该州谋杀率。

> states["Nevada",]



> fitted(fit)["Nevada"]#预测输出3.878958，实际谋杀率是百分之11.5

> residuals(fit)["Nevada"]#残差输出7.621042

> rstudent(fit)["Nevada"]#输出3.542929

〇可视化误差的其他方法

> residplot<-function(fit,nbreak=10){

+ z<-rstudent(fit)

+ hist(z,breaks = nbreak,freq = FALSE,

+ xlab = "Studentized Residual",

+ main = 'Distribution of Errors')

+ rug(jitter(z),col = "brown")

+ curve(dnorm(x,mean = mean(z),sd=sd(z)),

+ add = TRUE,col="blue",lwd=2)

+ lines(density(z)$x,density(z)$y,

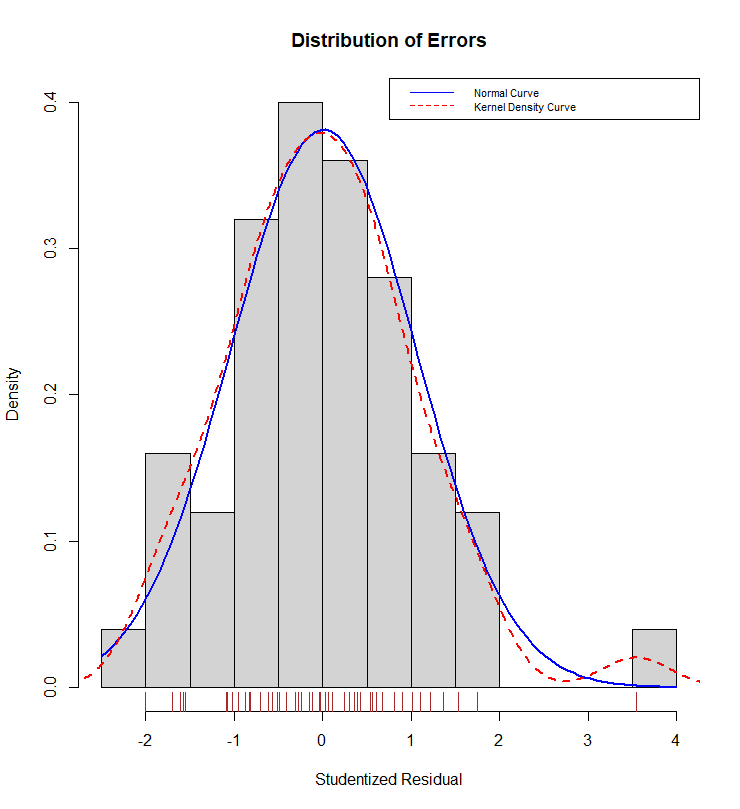
+ col="red",lwd=2,lty=2)

+ legend("topright",

+ legend = c("Normal Curve","Kernel Density Curve"),

+ lty=1:2,col = c("blue","red"),cex=.7)

+ }

> residplot(fit) 

除了一个很明显的离群点，误差很好地服从正态分布

②误差的独立性（Durbin-Watson检验-检测误差的序列相关性）

检验变量值（残差）是否相互独立，最好的方法是依据收集数据方式的先验知识。

时间序列数据通常呈现自相关性——相隔时间越近的观测相关性大于相隔越远的观测

> durbinWatsonTest(fit)



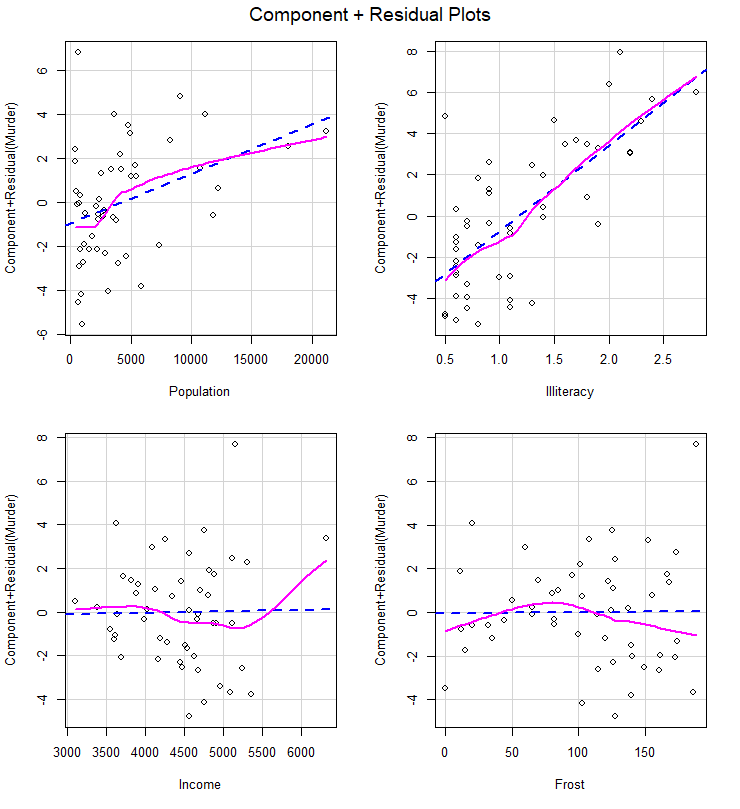
#P值不显著（0.282），说明无自相关性，误差项之间独立。滞后项lag=1表明数据集中，每个数据都是与其后一个数据进行比较的。该检验适用于时间独立的数据，对于非聚集型的数据并不适用。

③线性

通过成分残差图（component plus residual plot/partial residual plot）可以看因变量与自变量之间是否呈非线性关系，也可以看看是否有不同于已设定线性模型的系统偏差。

> library(car)

> crPlots(fit)

若图形存在非线性，则说明对预测变量的函数形式建模不够充分，需要添加一些曲线成分（如多项式），或对某些变量进行变换（如用log（x）代替x），或者不用线性回归。此图合理

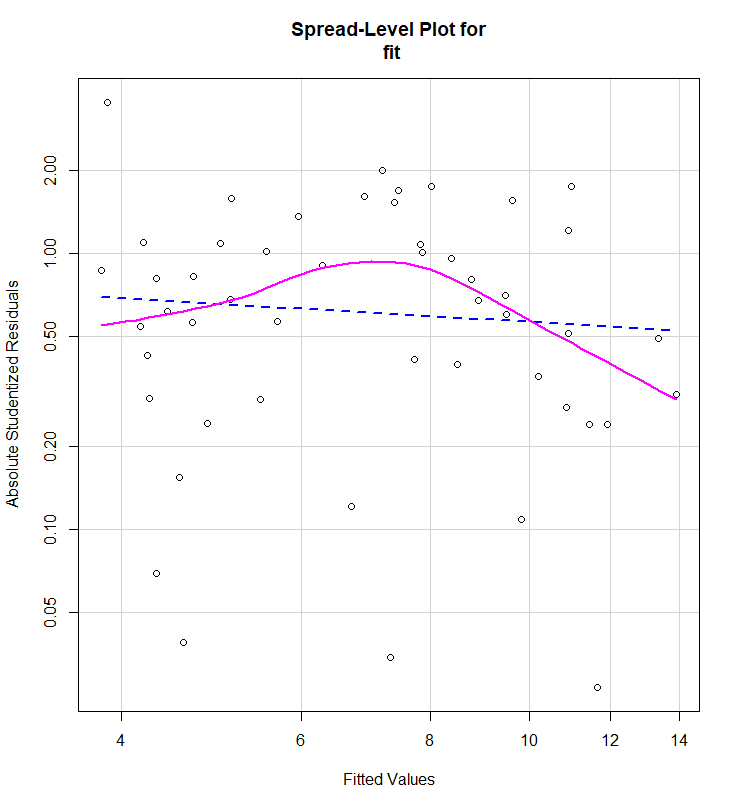
④同方差性（ncvTest()生成计分检验，零假设为误差方差不变，备选假设为误差方差随着拟合值水平变化而变化；若检验显著，则说明存在异方差性）

> library(car)

> ncvTest(fit)#计分检验不显著（p=0.19）说明满足方差不变的假设。



> spreadLevelPlot(fit)#通过水平分布图看到其中的点在水平的最佳拟合曲线周围呈水平随机分布。若违反该假设，得到非水平的曲线；代码结果建议1.2次幂变换





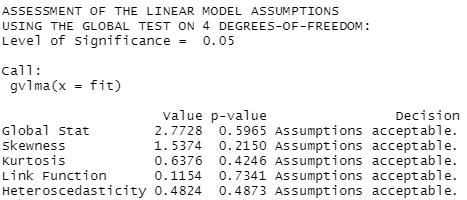
8.3.3 线性模型假设的综合验证（gvlma）

> install.packages("gvlma")

> library(gvlma)

> gvmodel<-gvlma(fit)

> summary(gvmodel)



#输出项Global Stat可以看到数据满足OLS回归模型所有统计假设p=0.597；p<0.05表明违反了假设条件

8.3.4 多重共线性multicollinearity：假设正在进行一项握力研究，自变量包括出生日期和年龄，用握力对出生日期和年龄进行回归，F检验显著，p<0.001，但当你观察出生日期和年龄的回归系数发现他们不显著（无法证明他们与握力相关）

回归系数测量相当于假定年龄不变然后测量握力与年龄的关系，就是多重共线性问题，它会导致模型参数的置信区间过大，使单个系数解释起来很困难。

多重共线性可用统计量VIF（方差膨胀因子）检测，VIF的平方根表示变量回归参数的置信区间膨胀为与模型无关的预测变量的程度。**>2**表明存在多重共线性问题

> library(car)

> vif(fit)



> sqrt(vif(fit))>2#probelm

#结果表明预测变量不存在多重共线性问题。

8.4异常观测值：包括离群点、高杠杆值点和强影响点

8.4.1离群点：是指那些模型预测效果不佳的观测点。通常有很大的或正或负的残差;正残差说明模型低估了响应值，负残差说明高估了响应值。

> library(car)

> outlierTest(fit)#该函数求得最大标准化残差绝对值Bonferroni调整后的p值：Nevada被判定为离群点Bonferroni p是0.048。若不显著，说明数据集中没有离群点；若显著，则必须删除该离群点，然后再检验是否还有其他离群点存在。



8.4.2高杠杆值点：与其他预测变量有关的离群点；它们由许多异常的预测变量值组合起来，与响应变量值没有关系。

#通过*帽子统计量（hat statistic）*判断，对于一个给定的数据集，帽子均值为p/n，p是模型估计的参数数目（包含截距项），n是样本量。若观测点的帽子值大于帽子均值的2倍或3倍，就可以认定为高杠杆值点；

高杠杆值点是否强影响点要看其是否离群点。

> hat.plot<-function(fit){

+ p<-length(coefficients(fit))

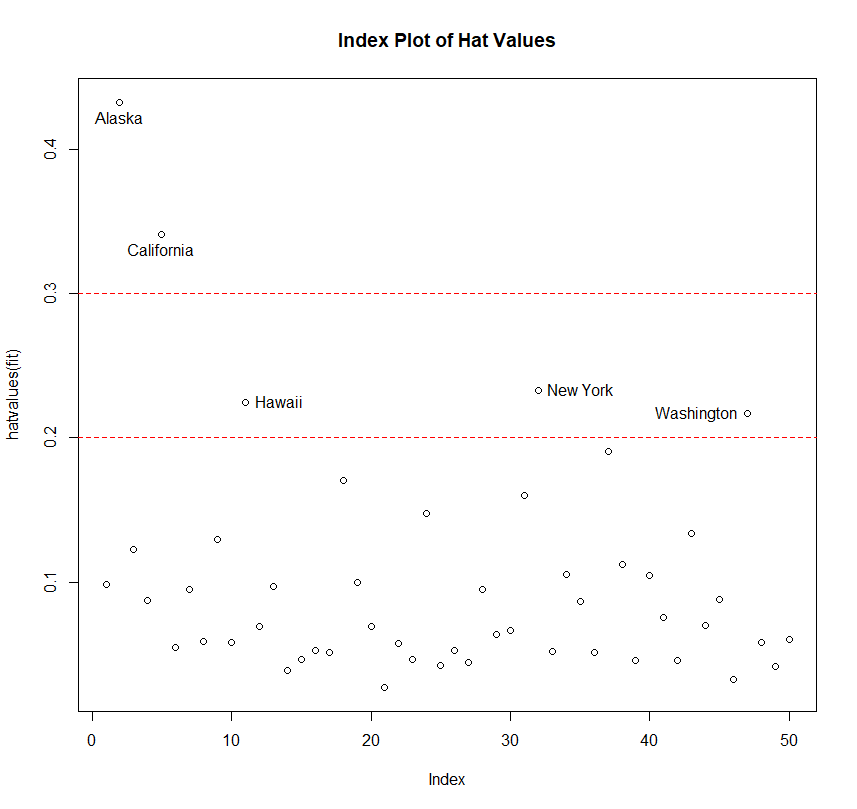
+ n<-length(fitted(fit))

+ plot(hatvalues(fit),main = "Index Plot of Hat Values")

+ abline(h=c(2,3)\*p/n,col="red",lty=2)

+ identify(1:n,hatvalues(fit),names(hatvalues(fit)))

+ }

> hat.plot(fit)#点图中要看名称的点再按esc；水平线标注的即帽子均值两倍和三倍的位置。Alaska和California非常异常；Alaska收入比其他州高很多，人口和温度很低、California人口比其他州府多得多，但收入和温度也很高。

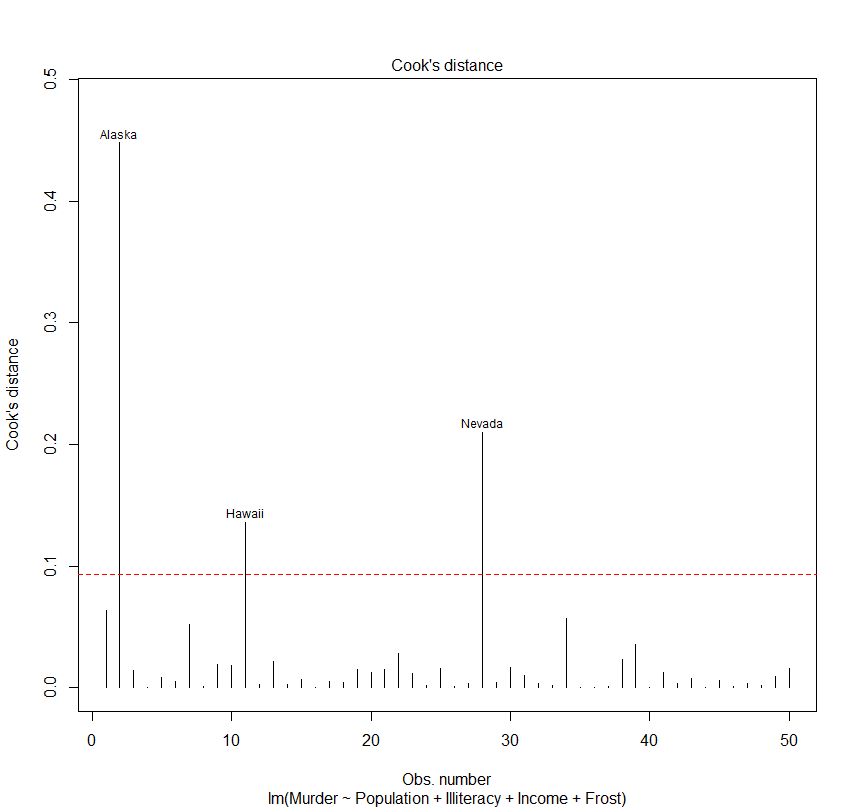
8.4.3强影响点（对模型参数估计值影响有些比例失衡的点）：*若移除模型的一个观测点时模型会发生巨大的改变，就需要检测一下数据中是否存在强影响点。*

#两种检测方法：①Cook距离(D统计量)：大于4/(n-k-1)是强影响点，n样本量大小、k是预测变量数目②变量添加图

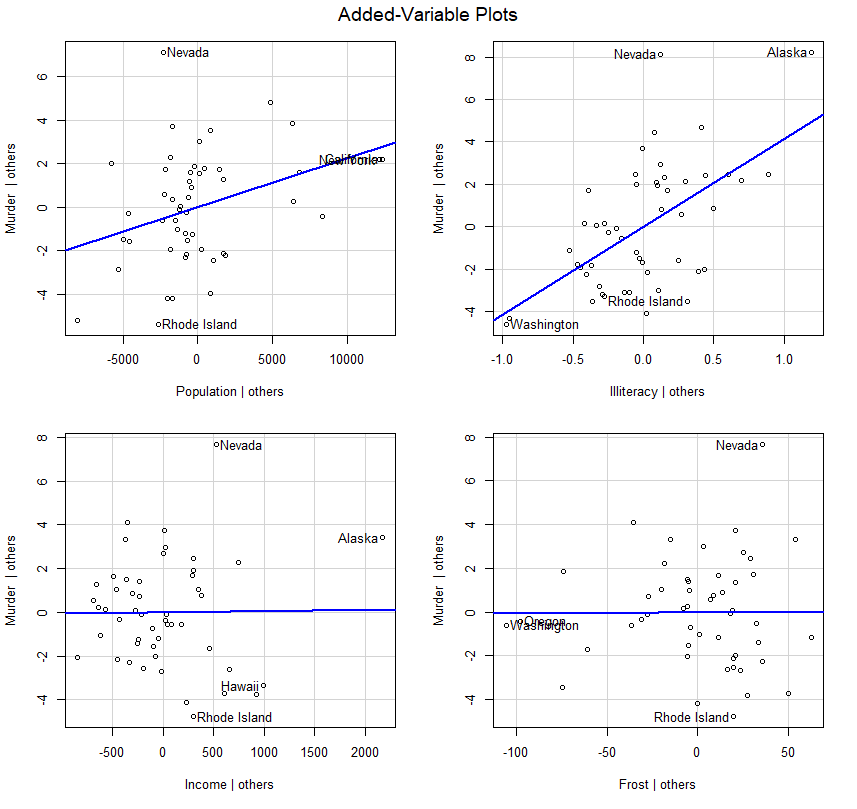
①> cutoff<-4/(nrow(states)-length(fit$coefficients)-2)

> plot(fit,which = 4,cook.levels = cutoff)

> abline(h=cutoff,lty=2,col="red")

#看出Alaska、Hawaii和Nevada是强影响点，若删除这些点会导致回归模型截距项和斜率发生显著变化；Cook’sD有助于鉴别强影响点，但并不提供关于这些点如何影响模型的信息，变量添加图可以弥补此缺陷

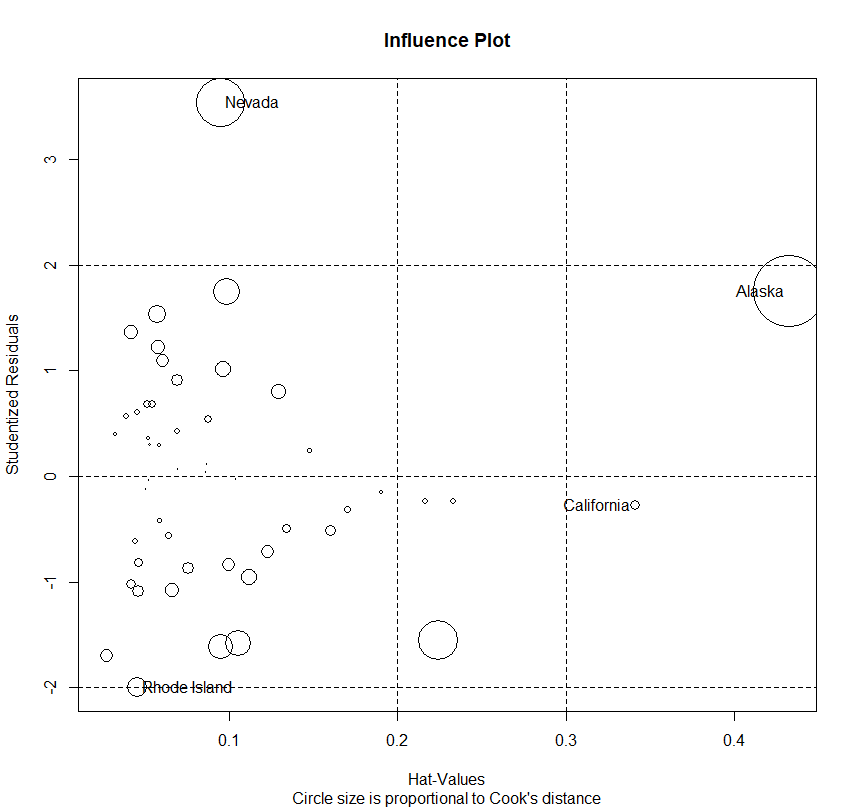
②> library(car)

> avPlots(fit,ask = FALSE,id.method="identify")

图中直线表示相应预测变量的实际回归系数。可以想象删除某些强影响点后直线的改变以此来估计它的影响效果（如左下图删除Alaska，Income的回归系数会向负移）。

③将离群点、杠杆值和强影响点信息整合到一幅图形中

> library(car)

> influencePlot(fit,id.method="identify",main="Influence Plot",sub="Circle size is proportional to Cook's distance")

有问题：显示不出所有需要的点的名字

Nevada和Rhode Island是离群点；纵坐标超过+-2

NewYork、California、Hawaii和Washington有高杠杆值；水平轴超过0.2或0.3

Nevada、Alaska和Hawaii为强影响点；圆圈大小与影响成比例，圆圈很大可能是对模型参数估计造成的不成比例影响的强影响点。

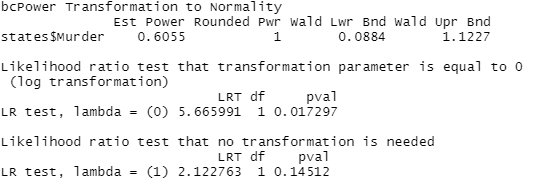
8.5改进措施

#四种方法处理违背回归假设的问题：

a.删除观测点；

b.变量变换；如使用代替Y

> library(car)

> summary(powerTransform(states$Murder))

#此结果表示可以用来正态化变量Murder。（可以尝试平方根）X=1时p=0.145假设无法拒绝

> library(car)

> boxTidwell(Murder~Population+Illiteracy,data = states)#获得预测变量幂数的最大似然估计来改善线性关系。



#结果显示和能改善线性关系，但是p值分别为0.75和0.54的计分检验又表明变量不需要变换。

c.添加或删除变量；常见删除>2的值

d.使用其他回归方法：①存在离群点、强影响点——使用稳健回归模型替代OLS回归②违背正态性假设——使用非参数回归模型③存在显著非线性——非线性回归模型④违背误差独立性假设——时间序列模型或多层次回归模型

8.6选择“最佳”回归模型

8.6.1模型比较（anova比较两个嵌套模型的拟合优度）

①

> states<-as.data.frame(state.x77[,c("Murder","Population",

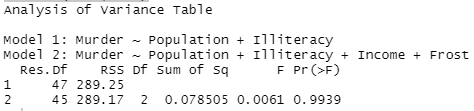
+ "Illiteracy","Income","Frost")])

> fit1<-lm(Murder~Population+Illiteracy+Income+Frost,data = states

+ )

> fit2<-lm(Murder~Population+Illiteracy,data = states)

> anova(fit2,fit1)#此处模型1嵌套在模型2中；p值0.994不显著故得出结论不需要将后两个变量添加到线性模型中。



②AIC赤池信息准则可以用来比较模型

> fit1<-lm(Murder~Population+Illiteracy+Income+Frost,data = states)

> fit2<-lm(Murder~Population+Illiteracy,data = states)

> AIC(fit1,fit2)



#优先选择AIC值小的模型fit2，说明模型用较少的参数获得了足够的拟合度。

8.6.2变量选择

①逐步回归：模型会一次添加或删除一个变量，直到达到某个判停准则为止。

a.向前逐步回归：每次添加一个预测变量到模型中，直到添加变量不会使模型有所改进为止。

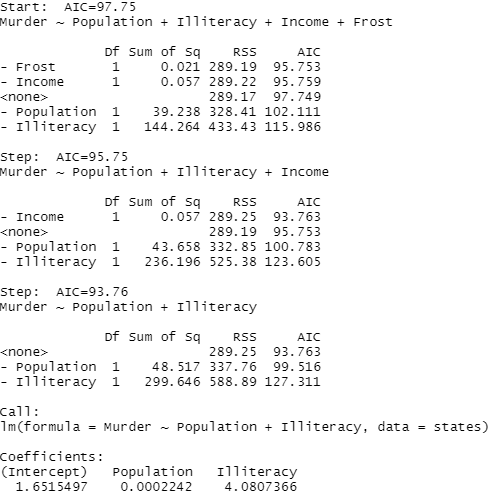
b.向后逐步回归：从模型包含的所有预测变量开始，一次删除一个变量直到会降低模型质量为止。

> library(MASS)

> states<-as.data.frame(state.x77[,c("Murder","Popultion","Illiteracy","Income","Frost")])

> fit<-lm(Murder~Population+Illiteracy+Income+Frost,data = states)

> stepAIC(fit,direction = "backward")#开始时模型包含四个变量，每一步删除一个变量后模型的AIC值。None表示没有变量被删除时模型的AIC。第一步删除Frost，AIC降低；第二步删除Income，AIC继续下降，再删AIC上升，终止删除。

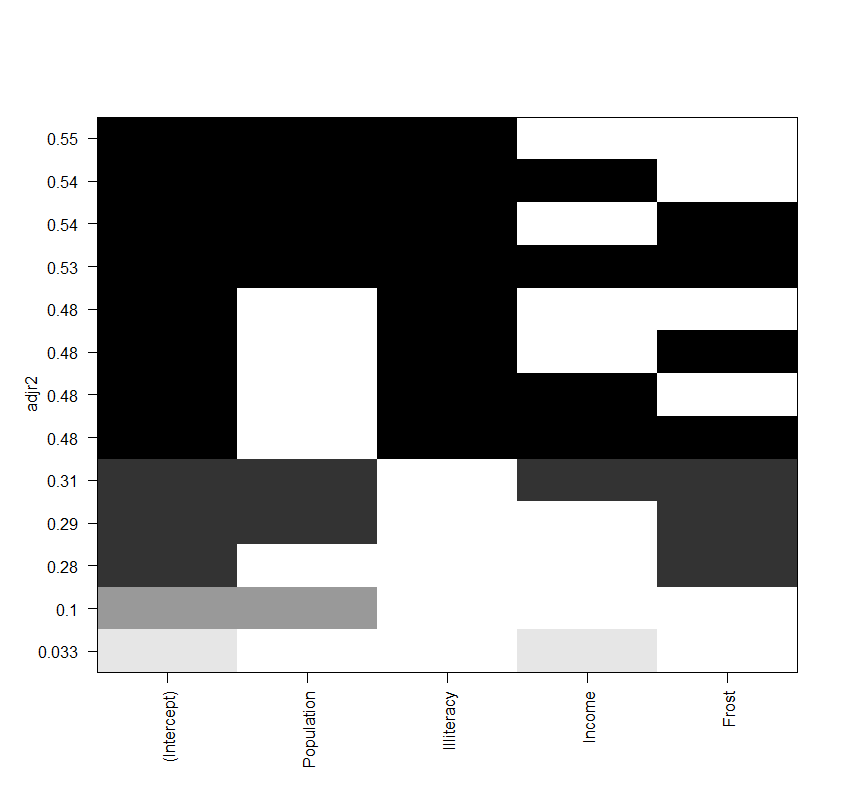
c.向前向后逐步回归：变量每进入一个都会被重新评价，对模型无贡献的会被删除，与测量可能被删减好几次，直到获得最优模型为止。

②全子集回归（优于逐步回归但是有大量预测变量时全子集回归很慢）：所有可能的模型都会被检验，分析员可以选则所有可能结果，也可以展示n个不同子集大小（一个、两个或多个预测变量）的最佳模型，若nbest=2，先展示两个最佳的单预测变量模型，然后展示两个最佳的双变量模型，直到包含所有的预测变量。

> library(leaps)

> states<-as.data.frame(state.x77[,c("Murder","Population","Illiteracy","Income","Frost")])

> leaps<-regsubsets(Murder~Population+Illiteracy+Income+Frost,data = states,nbest = 4)

> plot(leaps,scale = "adjr2")#含预测变量越少的模型调整R平方越大、模型越好（含intercept截距项、Population、illiteracy的双预测模型是最佳模型）

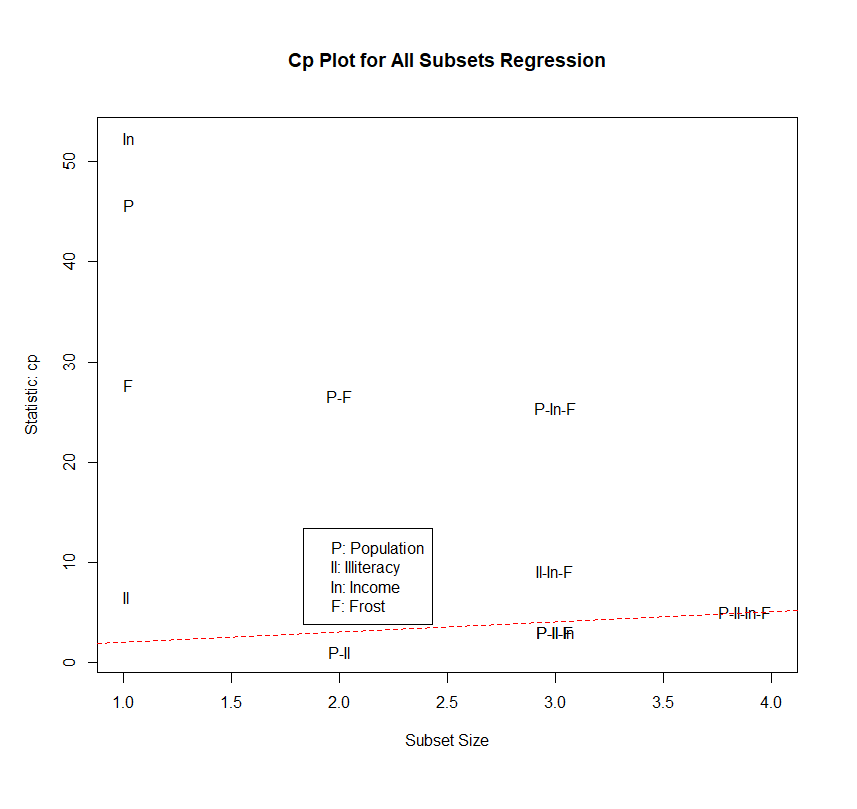
#R平方含义是预测变量解释响应变量的程度；调整R平方与之类似，防止丢失数据的偶然变异信息。R平方总会随着变量数目增加而增加。

>library(car)

>subsets(leaps,statistic = "cp",

main="Cp Plot for All Subsets Regression")

>abline(1,1,lty=2,col="red")

#要点一下图像；越好的模型离截距项和斜率为1的直线越近；此图表明可以考虑的模型有①P-Il双变量模型②P-Il-F三变量模型③P-Il-In模型④含P-Il-In-F四变量模型。

8.7深层次分析

8.7.1交叉验证：将一定比例的数据挑选出来作为训练样本，另外的样本作保留样本，先在训练样本上获取回归方程，然后在保留样本上预测。由于保留样本不涉及模型参数的选择，该样本可获得比新数据更为精确的估计。

①k重交叉验证：样本被分为k个子样本，轮流将k-1个子样本组合为训练集，另外一个子样本作为保留集；获得k个预测方程，记录k个保留样本的预测表现结果，然后求其平均值。

> install.packages("bootstrap")

> shrinkage<-function(fit,k=10){

+ require(bootstrap)

+ theta.fit<-function(x,y){lsfit(x,y)}

+ theta.predict<-function(fit,x){cbind(1,x)%\*%fit$coef}

+ x<-fit$model[,2:ncol(fit$model)]

+ y<-fit$model[,1]

+ results<-crossval(x,y,theta.fit,theta.predict,ngroup=k)

+ r2<-cor(y,fit$fitted.values)^2

+ r2cv<-cor(y,results$cv.fit)^2

+ cat("Original R-square=",r2,"\n")

+ cat(k,"Fold Cross-Validated R-square=",r2cv,"\n")

+ cat("Change=",r2-r2cv,"\n")

+ }#对模型的R平方统计量做了k重交叉验证；创建了一个包含预测变量和预测值的矩阵，可获得初始R平方以及交叉验证的R平方。二者之差越小，预测越精确。

> states<-as.data.frame(state.x77[,c("Murder","Population","Illiteracy","Income","Frost")])

> fit<-lm(Murder~Population+Income+Illiteracy+Frost,data = states)

> shrinkage(fit)



#R平方0.567

交叉验证后的R平方0.448

> fit2<-lm(Murder~Population+Illiteracy,data = states)#两个预测变量的模型

> shrinkage(fit2)

# R平方0.567

交叉验证后的R平方0.500比全模型减少的更少，表示结果更精确。

8.7.2相对重要性（各个变量的重要性排序）

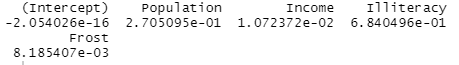
①

> states<-as.data.frame(state.x77[,c("Murder","Population","Illiteracy","Income","Frost")])

> zstates<-as.data.frame(scale(states))

> zfit<-lm(Murder~Population+Income+Illiteracy+Frost,data=zstates)

> coef(zfit)



#由此看出其他因素不变时文盲率一个标准差的变化将增加0.68个标准差的谋杀率。根据标准化的回归系数大小，认为Illiteracy是最重要的，Frost是最不重要的（最小）。

②相对权重（比回归系数更直观）

relweights<-function(fit,...){

R<-cor(fit$model)

nvar<-ncol(R)

rxx<-R[2:nvar,2:nvar]

rxy<-R[2:nvar,1]

svd<-eigen(rxx)

evec<-svd$vectors

ev<-svd$values

delta<-diag(sqrt(ev))

lambda<-evec%\*%delta%\*%t(evec)

lambdasq<-lambda^2

beta<-solve(lambda)%\*%rxy

rsquare<-colSums(beta^2)

rawwgt<-lambdasq%\*%beta^2

import<-(rawwgt/rsquare)\*100

import<-as.data.frame(import)

row.names(import)<-names(fit$model[2:nvar])

names(import)<-"Weights"

import<-import[order(import),1,drop=FALSE]

dotchart(import$Weights,labels = row.names(import),

xlab = "% of R-square",pch = 19,

main = "Relative Importance of Predictor Variables",

sub=paste("Total R-square=",round(rsquare,digits = 3)),

...)

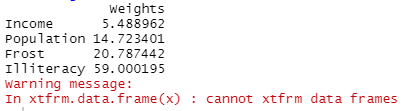
return(import)

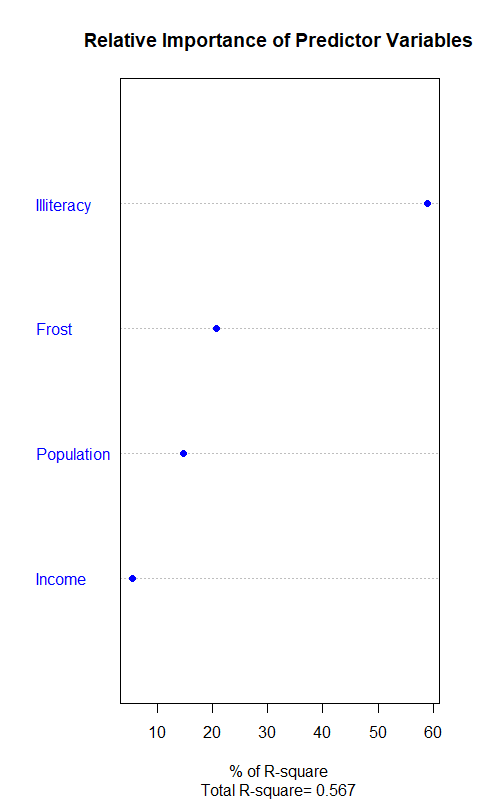
}#太容易打错了

states<-as.data.frame(state.x77[,c("Murder","Population","Illiteracy","Income","Frost")])

fit<-lm(Murder~Population+Income+Illiteracy+Frost,data = states)

relweights(fit,col="blue")





#illiteracy解释了59%的R方具有最大的相对重要性，Frost解释了21%次重要。

**第九章 方差分析ANOVA**

> install.packages("gplots")

> install.packages("HH")

> install.packages("rrcov")

> install.packages("multicomp")

> install.packages("effects")

> install.packages("MASS")

> install.packages("mvoutlier")

9.1术语速成

例：治疗焦虑症的两种方案：①认知行为疗法CBT②眼动脱敏再加工法EMDR；招募10个焦虑症患者为志愿者，随机分配一半人接受为期5周的CBT，另一半人5周EMDR；治疗结束时每位患者都填写状态特质焦虑问卷STAI。

治疗方案是CBT、EMDR的两水平组间因子。S代表患者。STAI是因变量，治疗方案是自变量。由于在每种治疗方案下的观测数相等，因此称为均衡设计；否则为非均衡设计。

〇单因素方差分析one-way ANOVA通过F检验来进行效果评测，若治疗方案的F检验显著，则说明两种疗法的STAI得分均值不同。

〇假设仅对CBT疗法的不同时间的疗效感兴趣，将时间分为两个水平（五周、六个月）。单因素组内方差分析；由于每个受试者不止一次被测量，也称作重复测量方差分析。

〇假设对治疗方案和它随时间的改变都感兴趣，则结合两个设计，随机分配五位患者到CBT，另外五位到EMDR，五周和六个月后分别评价他们的STAI结果。

其中，分析疗法的影响和分析时间的影响称作主效应。分析疗法和时间的交互影响称作交互效应。（双因素混合模型方差分析）

两因子——双因素方差分析。

若因子设计包括组内和组间因子——混合模型方差分析。

本例中做三次F检验①疗法因素②时间因素③两者交互因素。

①若疗法结果显著，说明CBT和EMDR对焦虑症的治疗效果不同；

②若时间结果显著，说明焦虑度从五周到六个月发生了变化；

③若两者交互效应显著，说明两种疗法随着时间变化对焦虑症治疗影响不同。。

〇扩展受试者患有抑郁症是混淆因素，可以解释因变量的组间差异。由于对抑郁症不感兴趣，他也被称作干扰变数。

〇假设招募患者使用抑郁症自我评价报告，如白氏抑郁症量表（BDI），记录了他们的抑郁水平。BDI为协变量，该设计为协方差分析（ANCOVA）

〇以上设计只记录了单个因变量情况（STAI），为增强研究的有效性，可以对焦虑症进行其他的测量（家庭评分、医师评分、以及焦虑症对日常行为的影响评价）。因变量不止一个——多元方差分析（MANOVA）。协变量也存在——多元协方差分析（MANCOVA）。

三种方法：ANOVA/ANCOVA/MANOVA

9.2ANOVA模型拟合

9.2.1aov()函数

〇符号：

①~分隔符号，左边为响应变量，右边为解释变量：y~A+B+C用ABC预测y

②：变量的交互项，y~A+B+A:B用A、B和AB的交互项预测y

③\*所有可能的交互项y~A\*B\*C等价于y~A+B+C+A:B+A:C+B:C+A:B:C

④^表示交互项到达某个次数y~(A+B+C)^2可展开为y~A+B+C+A:B+A:C+B:C

⑤.表示包含除因变量外的所有变量，若一个数据框包含y、A、B、C;y~.可展开为y~A+B+C

〇常见研究设计表达式

①单因素ANOVA：y~A

②含单个协变量的单因素ANCOVA：y~x+A

③双因素ANOVA：y~A\*B

④含两个协变量的双因素ANCOVA：y~x1+x2+A\*B

⑤随机化区组：y~B+A

⑥单因素组内ANOVA：y~A+Error(Subject/A)

⑦含单个组内因子（w）和单个组间因子（B）的重复测量ANOVA：y~B\*W+Error（Subject/W）

9.2.2表达式中各项的顺序

两种情况顺序对表达式效应造成影响：a.因子不止一个，且是非平衡设计。b.存在协变量。

例如：对于双因素方差分析，若不同处理方式观测数不同，那么模型y~A\*B与y~B\*A的结果不同。

默认类型Ⅰ方法计算ANOVA效应。第一个模型可以写：y~A+B+A:B。结果将评价：①A对y的影响②控制A时，B对y的影响③控制A和B的主效应时，A与B的交互效应。

三种类型方法：类型Ⅰ（序贯型）R默认：A不做调整，B根据A调整，A:B交互项根据A和B调整。类型Ⅱ（分层型）：效应根据同水平或低水平的效应做调整，A根据B调整，B根据A调整，A：B交互项同时根据A和B调整。类型Ⅲ（边界型）SAS SPSS：每个效应根据模型其他各效应做相应调整。A根据B和A:B做调整，A:B交互项根据A和B调整。

样本大小越不平衡，效应项的顺序对结果的影响越大：越基础的效应项越需要放在表达式前面：协变量——主效应——双因素的交互项——三因素的交互项。

Car包中的Anova函数提供Ⅱ Ⅲ型方法的选项，使结果与SAS SPSS保持一致，而aov()函数使用Ⅰ型方法。

9.3 单因素方差分析

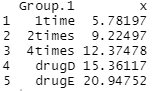
* 例

> library(multcomp)

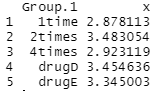
> attach(cholesterol)#50个患者接受降低胆固醇药物治疗（trt）五种疗法中的一种疗法。20mg一天一次、10mg一天两次、5mg一天四次表示的是相同的药物，剩下两种方式drugD和drugE代表候选药物。

> table(trt)#输出分组样本大小



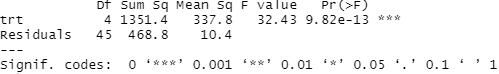
> aggregate(response,by=list(trt),FUN=mean)#求各组药物降低胆固醇响应变量的均值

> aggregate(response,by=list(trt),FUN=sd)#求各组求药物降低胆固醇响应变量的标准差

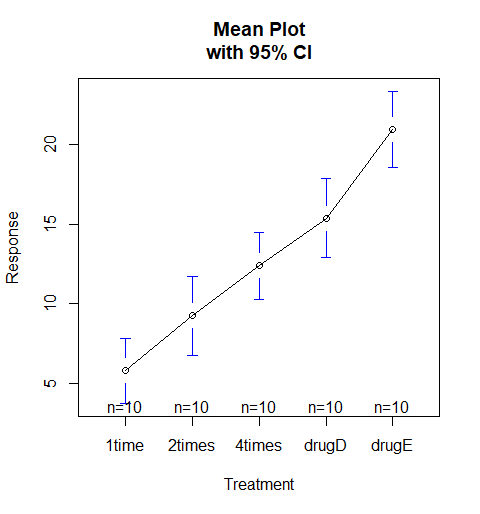


> fit<-aov(response~trt)#检验组间差异（ANOVA）

> summary(fit) #trt的F检验非常显著p<0.0001说明五种疗法效果不同



> library(gplots)

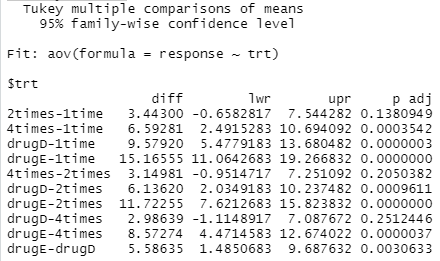
> plotmeans(response~trt,xlab = "Treatment",ylab = "Response",main="Mean Plot\nwith 95% CI") #绘制各组均值及置信区间的图形，上下线是标准差。

> detach(cholesterol)

9.3.1 多重比较

* ANOVA对各疗法的检验表明五种药物疗法效果不同，但并没有告知哪种疗法与其他疗法不同

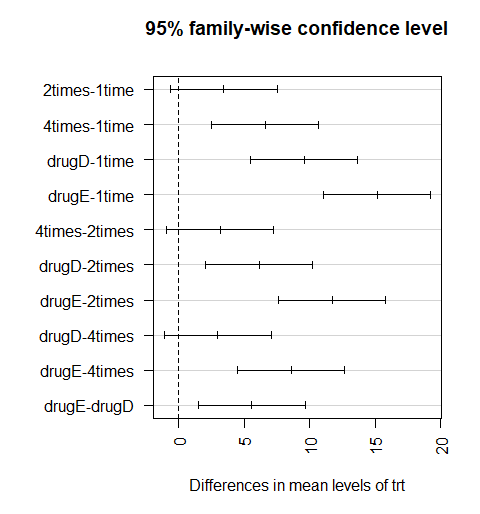
> TukeyHSD(fit)#提供了对各组均值差异的成对检验，1times和2times均值差异不显著（p=0.138）1times和4times显著（p<0.001）



> par(las=2)

> par(mar=c(5,8,4,2))

> plot(TukeyHSD(fit))#图形中置信区间包含0的疗法说明差异不显著（p>0.5）



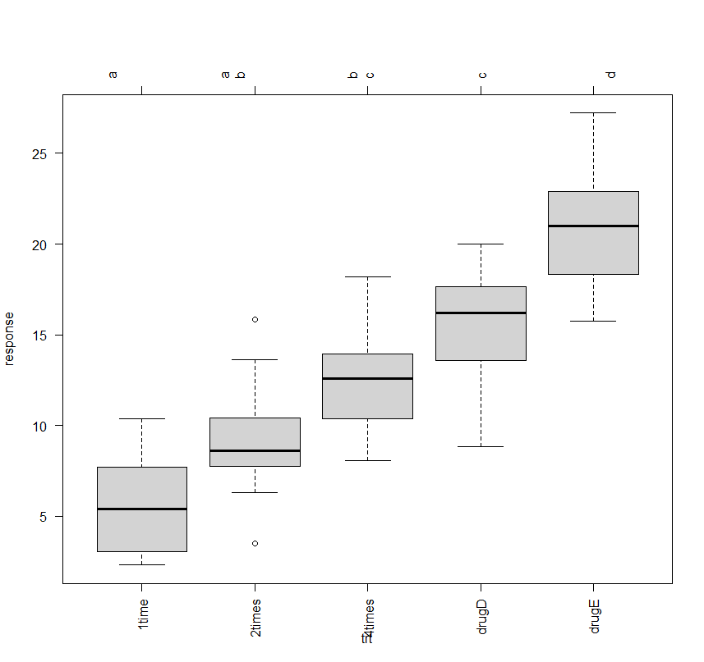
* Multcomp包中的glht函数提供了多重均值比较更全面的方法，既适用于线性模型，也适用于广义线性模型。

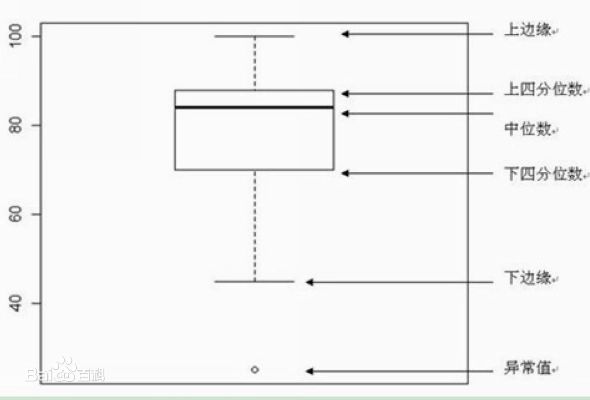
> library(multcomp)

> par(mar=c(5,4,6,2))#mar表示边界尺寸大小单位英分，“ 下、左、上、右”

> tuk<-glht(fit,linfct=mcp(trt="Tukey"))

> plot(cld(tuk,level=.05),col="lightgrey")#函数中的level设置了显著性水平



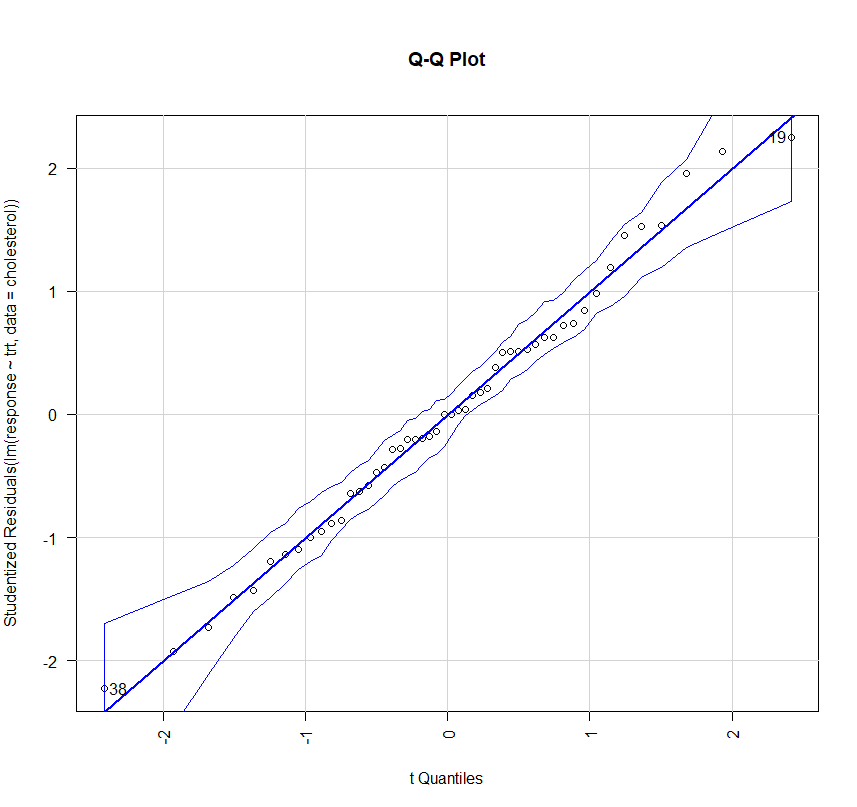


#有相同字母的组说明均值差异不显著1time2times不显著a，2times和4times也不显著b

9.3.2评估检验的假设条件

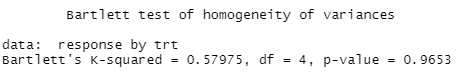
单因素方差分析中，我们假设因变量服从正态分布，各组方差相等。可以使用Q-Q图来检验正态性假设

> library(car)

> qqPlot(lm(response~trt,data=cholesterol),simulate = TRUE,main="Q-Q Plot",labels=FALSE)#数据都落在95%的置信区间范围内，说明满足正态性假设

* 一些做方差齐性检验的函数：Bartlett检验

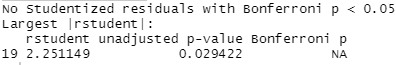
> bartlett.test(response~trt,data = cholesterol)#p=0.97表明五组的方差没有显著不同。



* 检测离群点

> library(car)

> outlierTest(fit)



从输出结果来看，没有证据说明胆固醇数据中含有离群点（Bonferroni p>1,产生NA）

9.4 单因素协方差分析

Multcomp包中的litter数据集，怀孕小鼠被分为四个小组，每个小组接受不同剂量0，5，50，500的药物处理，产下幼崽的体重均值为因变量，怀孕时间为协变量。此包在我电脑上有错误。解决方案rm(dose)

> data(litter,package = "multcomp")

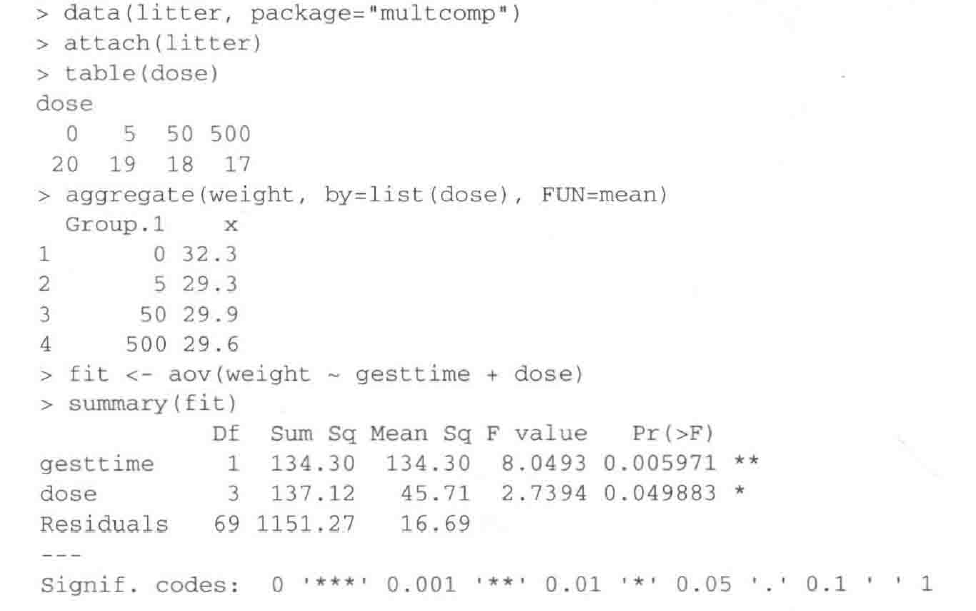
> attach(litter)

> table(dose)

> aggregate(weight,by=list(dose),FUN=mean)

> fit<-aov(weight~gesttime+dose)#协变量在前

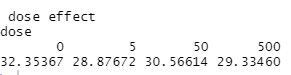
> summary(fit)

#先利用table函数，看到每种剂量下所产的幼崽数并不相同，再用aggregate函数获取各组均值，发现未用药组的幼崽体重均值最高。再使用ANCOVA的F检验表明：a.怀孕时间与幼崽出生体重相关；b.控制怀孕时间，药物剂量与出生体重相关。控制怀孕时间，确实发现每种药物剂量下幼崽出生体重均值不同。

* 使用effects来计算调整的均值（去除协变量效应gesttime后的均值）

> library(effects)

> effect("dose",fit)



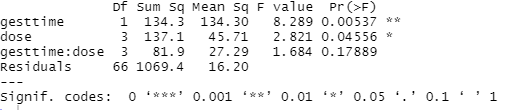
9.4.1评估检验的假设条件

* ANCOVA和ANOVA相同，都需要正态性和同方差性假设。ANCOVA假定回归斜率相同。本例中，假定四个处理组通过怀孕时间来预测出生体重的斜率回归率都相同。ANCOVA模型包含怀孕时间\*剂量的交互项时可对回归斜率的同质性进行检验。交互效应若显著，则意味着时间和幼崽出生体重间的关系依赖于药物剂量的水平。

> library(multcomp)

> fit2<-aov(weight~gesttime\*dose,data = litter)

> summary(fit2)#可以看到交互项的p值表示不显著。

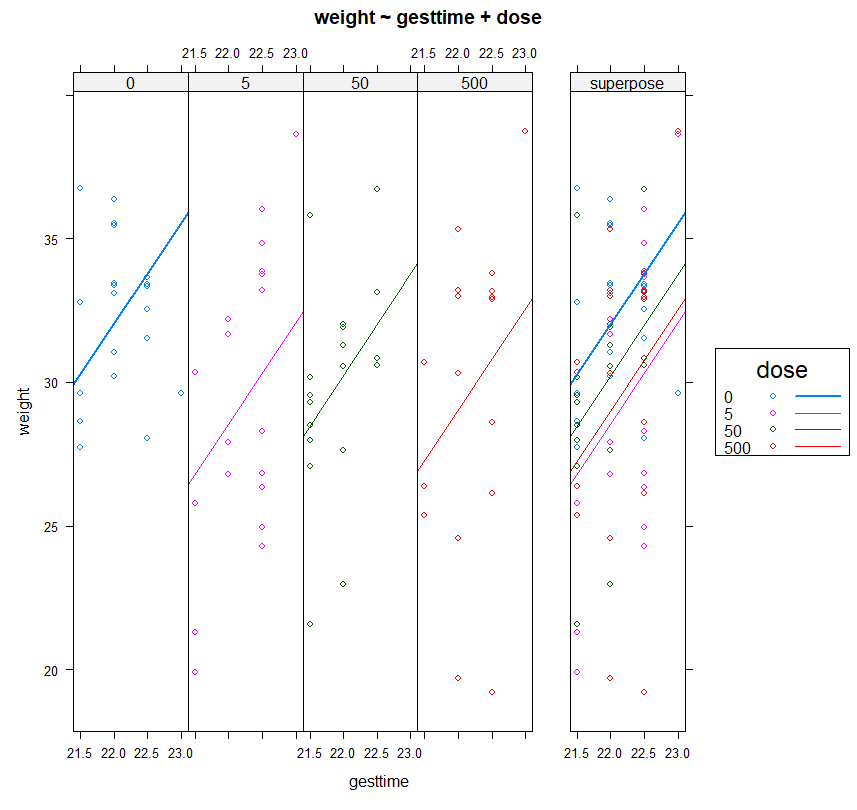


9.4.2结果可视化

* HH包中的ancova()函数可以绘制因变量、协变量和因子之间的关系图。ANCOVA假定回归斜率相同，但此图允许斜率依据组别而变化。

> library(HH)

> ancova(weight~gesttime+dose,data=litter)

#用怀孕时间来预测出生体重的回归线相互平行，只是截距项不同；0剂量截距最大，5剂量截距最小。

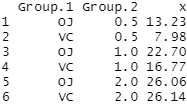
9.5双因素方差分析

* 例：随机60只豚鼠，两种supp喂食方法（橙汁或维生素C），各喂食方法中抗坏血酸含量dose有三种水平（0.5、1、2mg），每种组合处理方式分配10只豚鼠。牙长len为因变量。

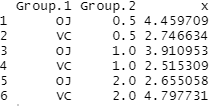
> attach(ToothGrowth)

> table(supp,dose)

> aggregate(len,by=list(supp,dose),FUN=mean)

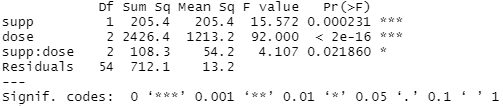


> aggregate(len,by=list(supp,dose),FUN=sd)



> dose<-factor(dose)#转化为因子变量

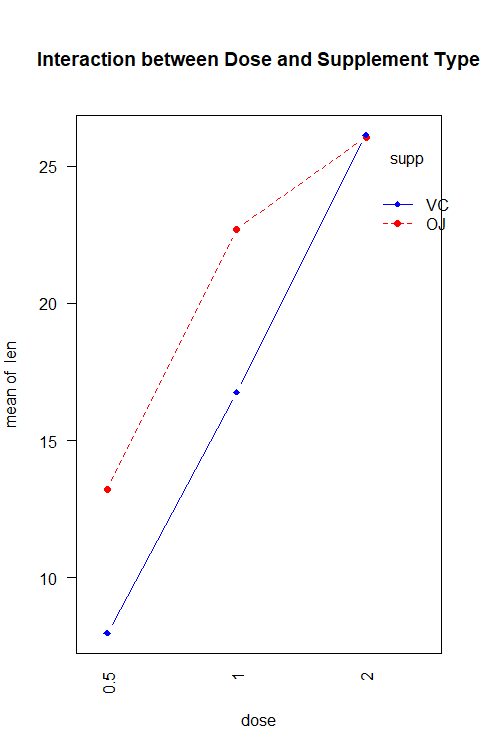
> fit<-aov(len~supp\*dose)

> summary(fit) 

> detach(ToothGrowth)

作图①

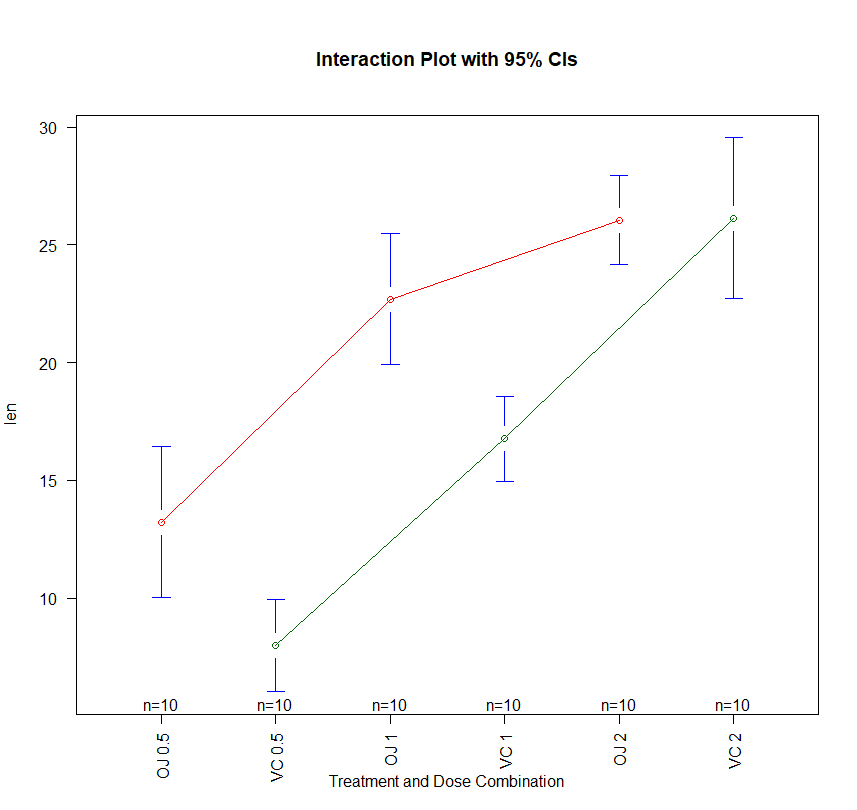
> interaction.plot(dose,supp,len,type="b",col = c("red","blue"),pch=c(16,18),main="Interaction between Dose and Supplement Type")#展示双因素方差的交互效应



作图②

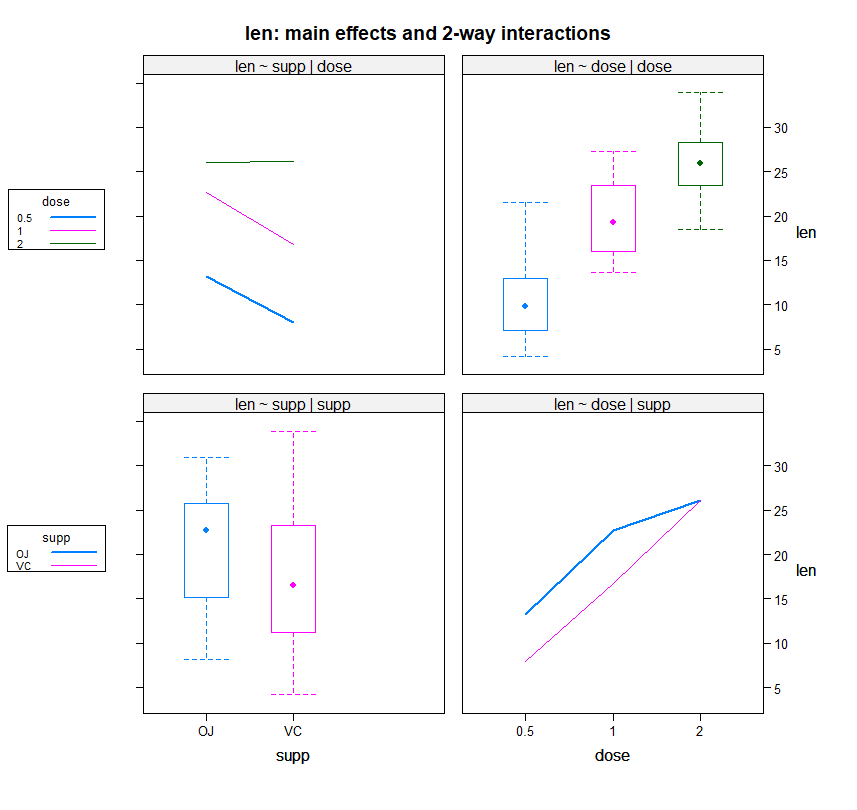
> library(gplots)

> plotmeans(len~interaction(supp,dose,sep=" "),connect = list(c(1,3,5),c(2,4,6)),col = c("red","darkgreen"),main="Interaction Plot with 95% CIs",xlab = "Treatment and Dose Combination")#展示了均值 、95%置信区间的误差棒（竖着的）和样本大小。

作图③

> library(HH)

> interaction2wt(len~supp\*dose)#图形对于任意顺序的因子设计的主效应和交互效应都会进行展示。

#以上三幅图表示2mg时两种喂食方法下牙齿长度增长相同，0.5和1mg剂量橙汁比维生素C更能促进牙齿生长。

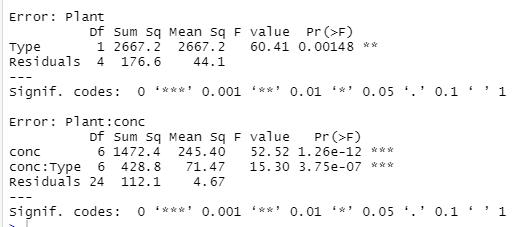
**9.6重复测量方差分析**

* 受试者被测量不止一次。例：CO2数据集包含某牧草的寒冷度研究结果，某浓度CO2下对寒带植物与非寒带植物的光合作用进行了比较，一半植物来自魁北克，一半来自密西西比。因变量是CO2吸收量uptake，自变量是植物类型type和CO2浓度的七种水平conc，Type是组间因子，conc是组内因子。

> CO2$conc<-factor(CO2$conc)#需要将conc转化为因子变量

> wlb1<-subset(CO2,Treatment=='chilled')#选择

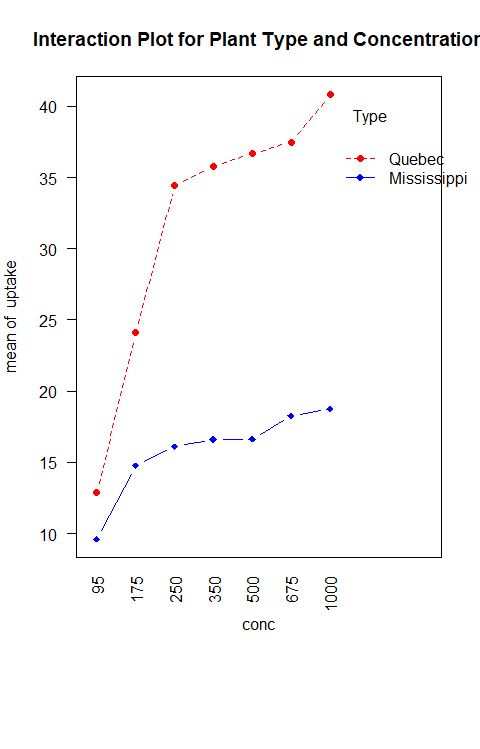
> fit<-aov(uptake~conc\*Type+Error(Plant/(conc)),wlb1)

> summary(fit) #表明交互效应非常显著

> par(las=2)

> par(mar=c(10,4,4,2))

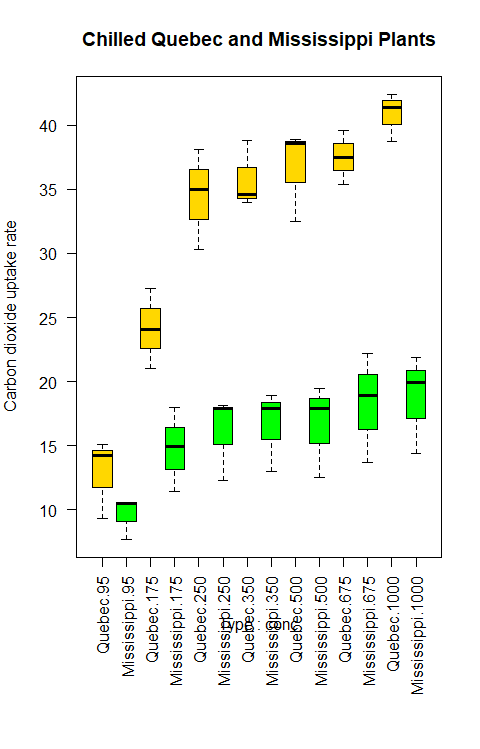
> with(wlb1,interaction.plot(conc,Type,uptake,type = "b",col = c("red","blue"),pch = c(16,18),main="Interaction Plot for Plant Type and Concentration"))



> boxplot(uptake~Type\*conc,data=wlb1,col=(c("gold","green")),main="Chilled Quebec and Mississippi Plants",ylab="Carbon dioxide uptake rate")

#可以看出魁北克的植物比密西西比的植物CO2吸收率高，而且随着CO2浓度升高，差异越来越明显。

9.7 多元方差分析



9.7多元方差分析

当因变量（结果变量）不止一个时，可用多元方差分析（MANOVA）对他们同时进行分析。研究谷物中卡路里、脂肪、糖含量（因变量）是否会因为储存架位置（自变量）不同而发生变化：1是底层货架，2是中层货架、3是顶层货架。

> library(MASS)

> attach(UScereal)

> shelf<-factor(shelf)#转化shelf为因子变量，代表一个分组变量

> y<-cbind(calories,fat,sugars)#按列合并向量，将三个因变量（卡路里、脂肪和糖）合并成一个矩阵。

> aggregate(y,by=list(shelf),FUN=mean)#获取货架的各个均值

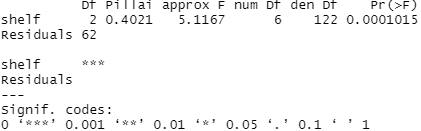


> cov(y)#输出谷物间的方差和协方差

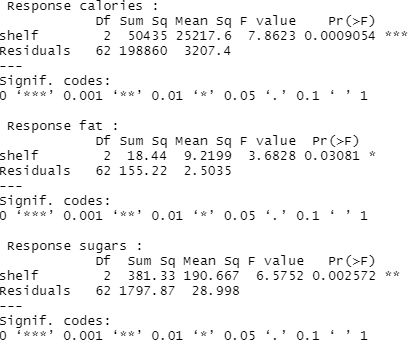


> fit<-manova(y~shelf)#对组间差异进行多元检验

> summary(fit)



> summary.aov(fit)#对每一个变量做单因素方差分析



9.7.1评估假设检验

①单元素方差分析有两个前提假设，一个是多元正态性，一个是方差-协方差矩阵同质性；第一个假设即指因变量组合成的向量服从一个多元的正态分布。可用Q-Q图来检验该假设条件。

以下图若数据服从多元正态分布，点将落在直线上。

> center<-colMeans(y)

> n<-nrow(y)#以y为行

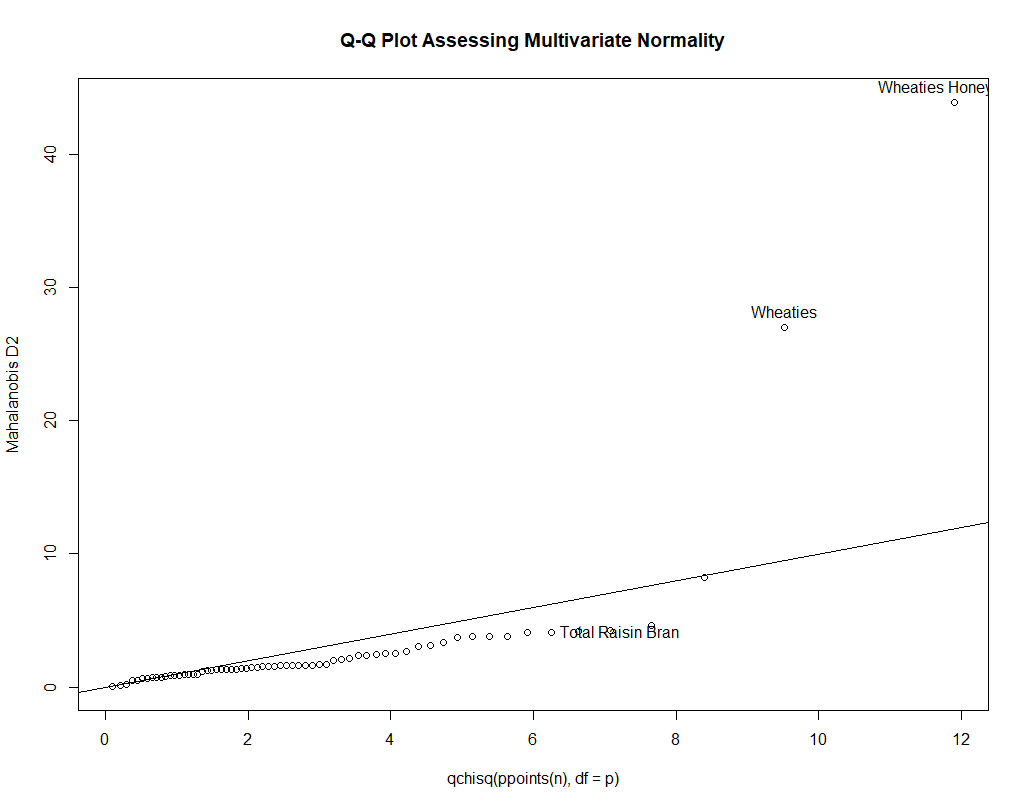
> p<-ncol(y)#以y为列

> cov<-cov(y)

> d<-mahalanobis(y,center,cov)

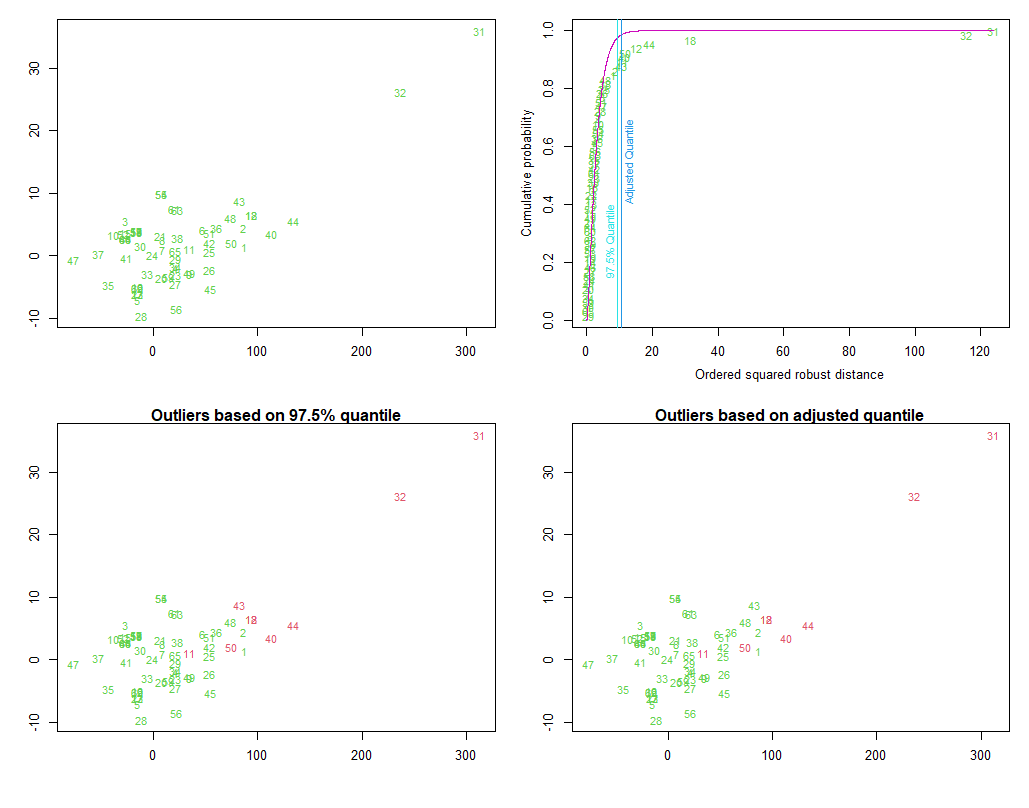
> coord<-qqplot(qchisq(ppoints(n),df=p),d,main="Q-Q Plot Assessing Multivariate Normality",ylab = "Mahalanobis D2")

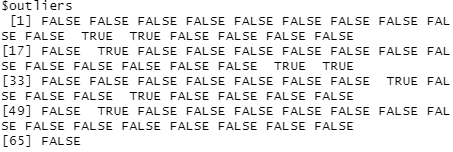
> abline(a=0,b=1)

> identify(coord$x,coord$y,labels = row.names(UScereal))#对图中点鉴别，可删除Wheaties和Wheaties Honey Gold点再观察

②> library(mvoutlier)

> outliers<-aq.plot(y)

> outliers#输出



9.7.2稳健多元方差分析

* 如果都不满足多元正态性或方差-协方差均值假设，或者担心离群点，则使用稳健或非参数版本的MANOVA检验。

> library(rrcov)

> Wilks.test(y,shelf,method="mcd")#死机了，结果应该是p值显著，营养成分随货架高低不同

9.8用回归来做ANOVA

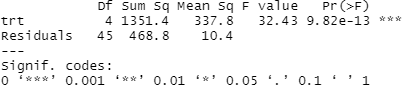
* ANOVA和回归都是广义线性模型的特例。因此本章所有的设计都可用lm函数来分析

> library(multcomp)

> levels(cholesterol$trt)#输出[1] "1time" "2times" "4times" "drugD" "drugE"

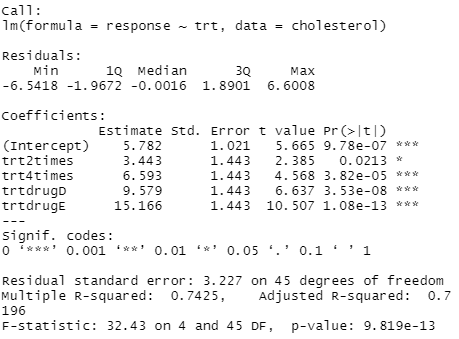
> fit.aov<-aov(response~trt,data = cholesterol)#先用aov（）函数拟合模型

> summary(fit.aov)

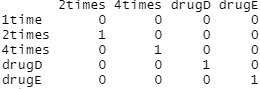


> fit.lm<-lm(response~trt,data = cholesterol)#再用lm函数拟合

> summary(fit.lm) #当lm()函数碰到因子时，他会用一系列与因子水平对应的数值型对照变量来代替因子。



> contrasts(cholesterol$trt)#查看编码水平，因子的第一个水平为参考组，随后的变量都以他为标准。



* 修改对照方法：

①> fit.lm<-lm(response~trt,data = cholesterol,contrasts = "contr.helmert")

②> options(contrasts = c("contr.SAS","contr.helmert"))#修改R中默认的对照方法

第十章 功效分析

* 功效分析可以帮助在给定置信度的情况下，判断检测到给定效应值时所需的样本量；也可以计算在给定置信度水平下，计算某样本量内能检测到给定效应值的概率。

10.1假设检验速览

* 在统计假设检验中，首先对总体分布参数设定一个假设（零假设，H0），然后从总体分布中抽样，通过样本计算所得的统计量来对总体参数进行推断。假定零假设为真，如果计算获得观测样本的统计量的概率非常小，便可以拒绝原假设，接受它的对立面（H1备择假设、研究假设）。
* 假设想评价使用手机对驾驶员反应时间的影响，则零假设H0：μ1-μ2=0；μ1是驾驶员使用手机时的反应时间均值，μ2是驾驶员不使用手机时的反应时间均值。假如拒绝该零假设，备选假设或研究假设就是H1：μ1-μ20。

现选一个不同个体构成的样本，两种情况：①一边打手机一边驾驶②不打手机驾驶。然后评估每个个体的反应时间，X1X2表示两种情况下反应时间的均值，S是标准差，n是每个条件下参与者数目；如果零假设为真，则可以假定反应时间呈正态分布，样本统计量服从2n-2自由度的t分布。但如果概率p小于阈值（如0.05显著性水平），则拒绝原假设接受H1。

事实上零假设永远不可能为真，足够大的样本总会被拒绝。

* 得出四种情况：

①零假设错误，统计检验拒绝它，我做了正确判断，使用手机影响反应时间。

②零假设正确，我没有拒绝他，我做了正确判断，使用手机不影响反应时间。

③零假设真实而我拒绝，犯了Ⅰ型错误。得到使用手机会影响反应时间的结论，而实际上不会。显著性水平α由Ⅰ型错误的概率定义，发现效应不发生的概率。

④零假设错误而我没有拒绝，犯了Ⅱ型错误，使用手机影响反应时间而我没判断出来。功效为1-Ⅱ型错误的概率，真实效应发生的概率。

效应值：备择假设下效应的量。

* 研究者可以控制样本大小和显著性水平，但对功效和效应值的影响是间接的。放宽显著性水平（阈值变大）或样本量增加，功效增加。
* 样本大小、显著性水平、功效和效应值给定三个量可推算第四个量。

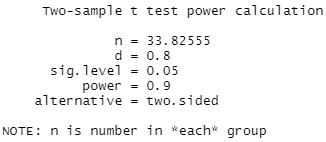
10.2 pwr包做功效分析

10.2.1t检验

①> install.packages("pwr")

> library(pwr)

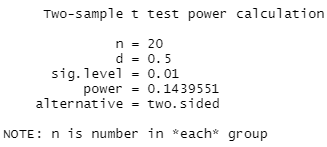
> pwr.t.test(d=.8,sig.level=.05,power=.9,type="two.sample",alternative="two.sided")#n为样本大小，d为效应值即标准化的均值之差，sig.level表示显著性水平，power为功效水平，type指检验类型：双样本t检验two.sample（默认 主要确定两个独立组样本的总体均值是否相等）、单样本t检验one.sample、相依样本t检验paired，alternative是指(默认)双侧检验（two.sided）还是单侧检验（less或greater）



#结果表示需要34个受试者（共68人），才能保证有90%的把握检测到0.8的效应值，并且最多5%的可能性会误报差异存在。

②> library(pwr)

> pwr.t.test(n=20,d=.5,sig.level = .01,type = "two.sample",alternative = "two.sided")#假定只能有40个受试者，检测到总体均值0.5个标准差的差异，并且将误报差异的几率限制在1%以内

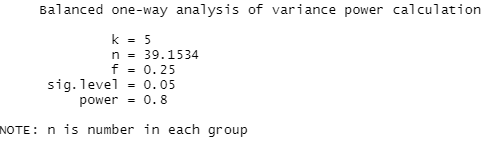
#结果表示因变量的标准差为1.25s，有低于14%的可能性断言差值为0.625s或者不显著。

③假定两组样本大小不等：

pwr.t2n.test()

10.2.2方差分析

> pwr.anova.test(k=5,f=.25,sig.level = .05,power = .8)#对五个组做单因素方差分析，要达到0.8的功效，效应值为0.25，0.05的显著性水平，计算各组需要样本大小

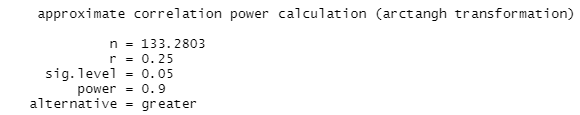


总样本大小为5\*39

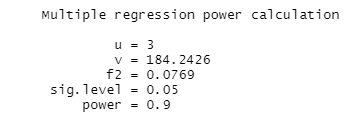
10.2.3相关性

* 假定在研究抑郁与孤独的关系，零假设和研究假设为：H0：ρ0.25和H1:ρ0.25

> pwr.r.test(r=.25,sig.level = .05,power = .90,alternative = "greater")#n是观测值，r是效应值，sig.level是显著性水平，power是功效水平，alternative是显著性检验是单边（greater或less）双边(two.sided)，因此要满足以上关系，需要134个受试者评价抑郁与孤独的关系，以便在零假设为假的情况下有90%的信心拒绝他

10.2.4 线性模型

* 想要研究老板的领导风格对员工的满意程度影响是否超过薪水和消费。经验表明：薪水和小费能解释30%的员工满意度的方差，从现实出发领导风格能解决35%的满意度方差，假定显著性水平0.05，那么在90%置信度的情况下，需要多少受试者才能得到这样的方差贡献率呢

> pwr.f2.test(u=3,f2=0.0769,sig.level = 0.05,power = 0.90)#f2=（0.35-0.30）/(1-0.35)

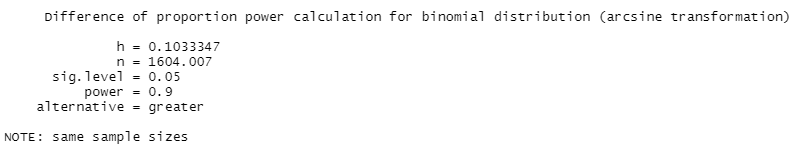
#u分子自由度，v分母自由度，f2效应值；分母自由度是N-k-1，N总观测数，k预测变量数，本例为v=185+7+1

评价一组变量对结果的影响程度;R2多重相关性的总体平方值；

评价一组变量对结果的影响程度超过第二组（协变量）多少；RA2集合A中变量对总体方差的解释率；RAB2集合A和B中变量对总体方差的 解释率。

10.2.5比例检验

* 假定某流行药物能缓解60%使用者的症状，对此感到怀疑，而一种更贵的新药能缓解65%使用者的症状就会被投放到新市场，在研究中需要多少受试者才能检测到这两种药物存在这一特定差异？

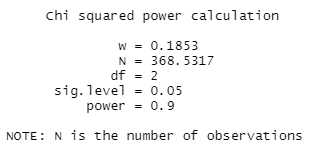
> pwr.2p.test(h=ES.h(.65,.6),sig.level=.05,power=.9,alternative="greater")#h效应值；得出结果需要各1605人适用新药和已有药物

10.2.6卡方检验

* 卡方检验常常用来评价两个变量之间的关系。典型的零假设是变量之间独立，备选假设是不独立。

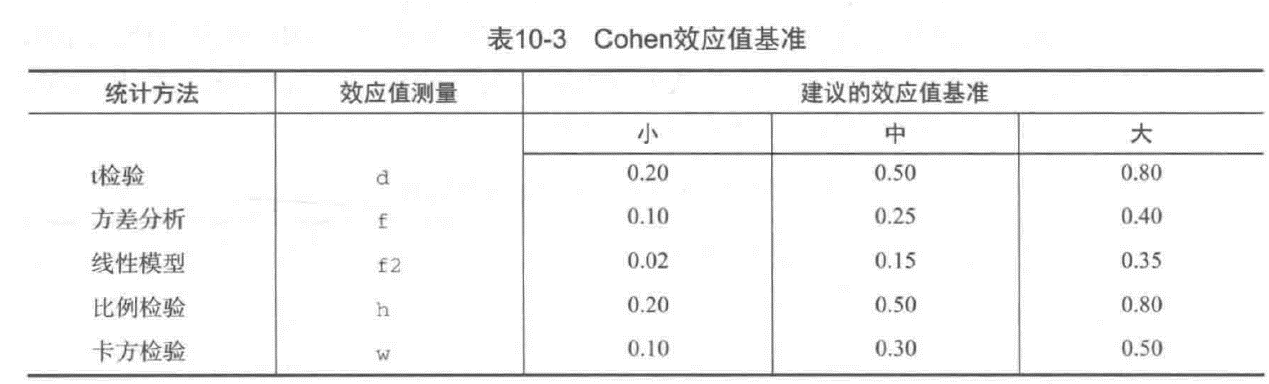
> prob<-matrix(c(.42,.28,.03,.07,.10,.10),byrow = TRUE,nrow = 3)#创建了黑人白人西班牙人晋升未晋升比例的矩阵（按顺序）

> ES.w2(prob)#输出0.1853198，这步是计算w效应值

> pwr.chisq.test(w=.1853,df=2,sig.level = .05,power = .9)

#结果表明要369个受试者才能检验人种与工作晋升的关系。

10.2.7在新情况中选择合适的效应值

* 假如想在0.05的显著性水平下，对5个组、每组25个受试者的设计进行单因素方差分析，那么拒绝错误零假设的概率（发现真实效应值）的概率多大呢？

> library(pwr)

> es<-seq(.1,.5,.01)

> nes<-length(es)#显示对象中元素的数量

> samsize<-NULL#代表连位置都没有，变量为空，其长度为0，表明“空无一物”

> for (i in 1:nes){

+ result<-pwr.anova.test(k=5,f=es[i],sig.level = .05,power = .9)

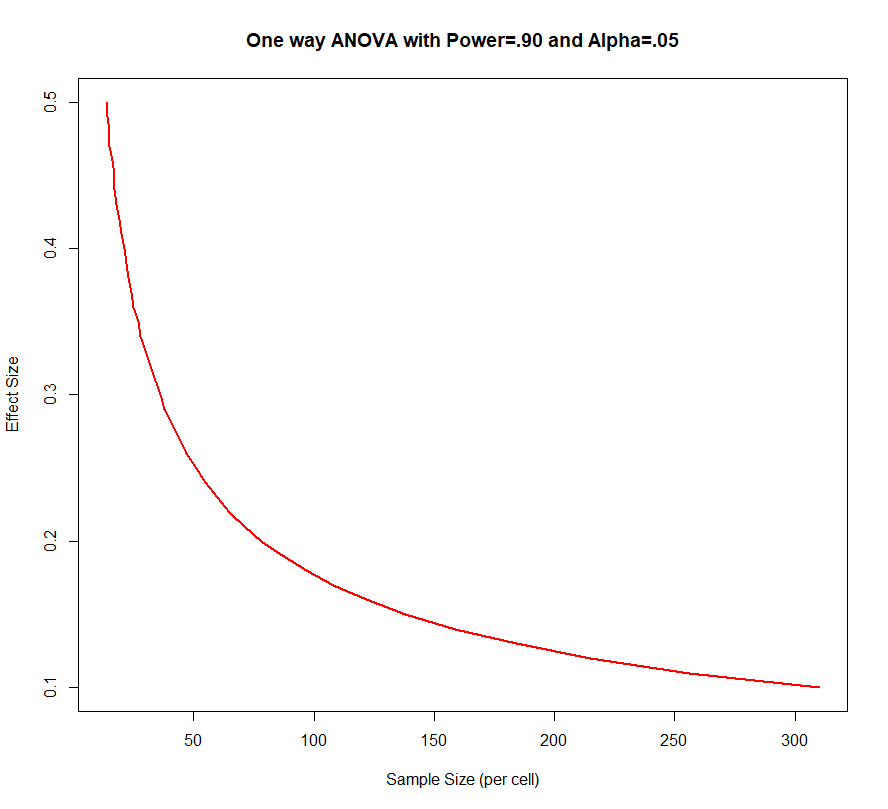
+ samsize[i]<-ceiling(result$n)

+ }

> plot(samsize,es,type="l",lwd=2,col="red",

+ ylab="Effect Size",

+ xlab="Sample Size (per cell)",

+ main="One way ANOVA with Power=.90 and Alpha=.05")

做PCA图，主成分分析#

> library(ggplot2)

> ggplot(data=df,aes(x=group,y=Lipid))+geom\_boxplot(aes(fill=group))+theme\_bw()

> library(readxl)

> df<-read("C:/Users/YueYiFan/Desktop/Document 1.xlsx",sheet=1)

> df<-read\_excel("C:/Users/YueYiFan/Desktop/Document 1.xlsx",sheet=1)

> df

> df.pca<-prcomp(df[,3:901])#只有数字的列

> summary(df.pca)#选两个主成分就够了，第二行达到百分之七十就可，第三行到百分之九十

> pca.result<-df.pca$x

> class(pca.result)#转为数据框

> pca.result<-data.frame(pca.result)

> pca.result$group<-as.character(df$Class)#加入组别

> library(ggplot2)

> ggplot(pca.result,aes(x=PC1,y=PC2))+geom\_point(aes(color=group))+theme\_bw()+labs(x="PC1",y="PC2")+stat\_ellipse(geom = "polygon",aes(group=group,color=group,fill=group),alpha=0.5)#画图，polygon椭圆

做热图

> library(pheatmap) #没下载的话先用install.packages(pheatmap)下载

> filepath<-file.choose() #选择文件，pheatmap需要用csv方便处理

> tes<-read.csv(filepath,header = T,row.names = 1) #建立一个数据框df(名称自己随便取）#表的行列名不要数字开头会bug

>test<-as.matrix(tes) #将数据框转化为矩阵（必不可少）

> pheatmap(test)

> pheatmap(test,scale="row") #以行为参数标准化(组别为列，代谢物为行）/如果以列为参数标准化，则scale="column",test是矩阵名

> pheatmap(test,scale="row",cluster\_cols=F)#使列顺序不变（列为分组，行为代谢物）、使行顺序不变为下一条

> pheatmap(test,scale="row",cluster\_rows=F)

>pheatmap(test, color = colorRampPalette(c("navy", "white", "firebrick3"))(10))#设置三个颜色，后面括号里的数字表示梯度，10就是将这三种颜色设置为10个梯度

>pheatmap(test, cluster\_rows = F, gaps\_row = c(10)) #gaps\_row有效的前提是cluster\_rows = F；cutree\_col有效的前提是cluster\_cols = T；利用gaps\_row, gaps\_col自己设定要分隔开的位置 ，在第十行分开，c(10, 14)是在第十行第十四行分开

>pheatmap((test,scale="column"),cutree\_rows=4)#分为四大行；cutree\_rows=3,cutree\_cols=2是分为三大行两大列

> pheatmap(test, cluster\_rows = F, gaps\_row = c(10), display\_numbers=TRUE, cellwidth=22, cellheight=15,fontsize =10) #显示每格数据，方块宽度，方块高度，字体大小

>pheatmap(test, cellwidth = 15, cellheight = 12, main = "Example heatmap", fontsize = 8, filename = "test.pdf") #filename将文件存出

>pheatmap(test, border=FALSE)#去除热图边框

>pheatmap(test, border\_color = "red")#设定边框颜色为红色

annotation\_col = data.frame( CellType = factor(rep(c("CT1", "CT2"), 5)), Time = 1:5 )

rownames(annotation\_col) = paste("Test", 1:10, sep = "")

annotation\_row = data.frame( GeneClass = factor(rep(c("Path1", "Path2", "Path3"), c(10, 4, 6))))

rownames(annotation\_row) = paste("Gene", 1:20, sep = "")

>pheatmap(test, annotation\_col = annotation\_col, annotation\_row = annotation\_row) #添加行列注释

>pheatmap(test, display\_numbers = TRUE, number\_format = "%.1e",number\_color="purple")#显示数值，保留一位小数，结果用科学计数法显示；e改为f则为用小数显示;数值颜色为紫色

>pheatmap(test, display\_numbers=ifelse (-0.05≤test ≤0.0 5, "\*",""))#以下几条是在满足条件时标记星号

>pheatmap(test,display\_numbers=ifelse(test＜0.05,"\*", "))

>pheatmap(test, display\_numbers = matrix(ifelse(test ＜＜0.0 5, "\*", ""), nrow(test)))

>pheatmap(test,display\_numbers=matrix(ifelse(-0.05≤test≤0.05, "\*", ""), 19,62))

>pheatmap(test, display\_numbers=matrix(ifelse(test>- 0.05 "\*", ""), 19,63))

> sample(x=x,size=5,replace=T)#在x中抽样，size指定抽样次数， “replace”可以重复对元素进行抽样

tapply(x,y,z)按y对x分组，并且对分组后的x进行z操作

unique返回向量、数据框或数组，但删除重复元素/行。

#install.packages("circlize")

library(circlize)

vignette("circlize")

#2做弦图的顺序

#circos.initialize()

#circos.track()

#circo.point() circos.lines() circos.text()

#circos.track()

#circo.point() circos.lines() circos.text()

#circos.clear()

sectors = c("d", "f", "e", "c", "g", "b", "a")#圈分为7部分

s2 = factor(sectors, levels = sectors)

circos.initialize(s2, xlim = c(0, 1))

circos.track(sectors, ylim = c(0, 1))

set\_track\_gap(cm\_h(1.5))#设置圆圈之间的间隔宽度cm

circos.track(ylim = c(0, 1))

set\_track\_gap(mm\_h(1))#设置圆圈之间的间隔宽度mm

circos.track(ylim = c(0, 1))

set\_track\_gap(mm\_h(0.5))#设置圆圈之间的间隔宽度mm

#circos.par查看弦图的默认参数

circos.clear()#下个弦图重新开始画

circos.info()#获取当前图像信息

#3

#3.1设置弦图颜色

#3.1.1

col\_fun = colorRamp2(c(-2, 0, 2), c("blue", "white", "red"))

col\_fun(seq(-5, 1, by = 1)) # all the values smaller than -2 are all mapped to blue

#3.1.2

col\_fun = colorRamp2(c(-2, 0, 2), c("blue", "white", "red"))

value = seq(-2, 2, by = 0.2)

value

col = col\_fun(value)

col2value(col, col\_fun = col\_fun)

#3.2点circos.points

#3.2线circos.lines()

#3.4片段circos.segments()

#3.5文本circos.text()

#3.6正方形circos.rect()和多边形circos.polygon()

#3.7每个框内的坐标circos.axis() circos.xaxis() circos.yaxis()

#3.8柱状图circos.barplot() 箱线图circos.boxplot() 小提琴图circos.violin()

par(mfrow = c(1, 2))

circos.initialize(letters[1:4], xlim = c(0, 10))

circos.track(ylim = c(0, 1), panel.fun = function(x, y) {

value = runif(10)#生成十个随机数

circos.barplot(value, 1:10 - 0.5, col = 1:10)

})

circos.track(ylim = c(-1, 1), panel.fun = function(x, y) {

value = runif(10, min = -1, max = 1)

circos.barplot(value, 1:10 - 0.5, col = ifelse(value > 0, 2, 3))

})

circos.clear()

circos.initialize(letters[1:4], xlim = c(0, 10))

circos.track(ylim = c(0, 4), panel.fun = function(x, y) {

value = matrix(runif(10\*4), ncol = 4)

circos.barplot(value, 1:10 - 0.5, col = 2:5)

})

#3.9圆圈箭头circos.arrow()

#3.10标签circos.labels()

#3.11图片circos.raster()

#3.12连线circos.link()

#3.13突出扇形区和轨道区draw.sector() highlight.sector()

#3.14与基本图像结合#text(0, 0, "This is\nthe center", cex = 1.5)#legend("bottomleft", pch = 1, legend = "This is the legend")#title("This is the title")

#4.添加标签

#5.高级弦图

#5.1条形柱状图

category = paste0("category", "\_", 1:9)#paste0拼接字符串

percent = sort(sample(40:80, 9))

color = rev(rainbow(length(percent)))

library(circlize)

circos.par("start.degree" = 90, cell.padding = c(0, 0, 0, 0))

circos.initialize("a", xlim = c(0, 100)) # 'a` just means there is one sector（仅有一个扇形区域）

circos.track(ylim = c(0.5, length(percent)+0.5), track.height = 0.8,

bg.border = NA, panel.fun = function(x, y) {

xlim = CELL\_META$xlim

circos.segments(rep(xlim[1], 9), 1:9,

rep(xlim[2], 9), 1:9,

col = "#CCCCCC")

circos.rect(rep(0, 9), 1:9 - 0.45, percent, 1:9 + 0.45,

col = color, border = "white")

circos.text(rep(xlim[1], 9), 1:9,

paste(category, " - ", percent, "%"),

facing = "downward", adj = c(1.05, 0.5), cex = 0.8)

breaks = seq(0, 85, by = 5)

circos.axis(h = "top", major.at = breaks, labels = paste0(breaks, "%"),

labels.cex = 0.6)

})

#5.2直方图circos.trackHist()

x = rnorm(1600)

sectors = sample(letters[1:16], 1600, replace = TRUE)

circos.initialize(sectors, x = x)

circos.trackHist(sectors, x = x, col = "#999999",

border = "#999999")

circos.trackHist(sectors, x = x, bin.size = 0.1,

col = "#999999", border = "#999999")

circos.trackHist(sectors, x = x, draw.density = TRUE,

col = "#999999", border = "#999999")

circos.clear()

#5.3系统发育树

#install.packages("ape")

library(ape)

data(bird.orders)

hc = as.hclust(bird.orders)

labels = hc$labels # name of birds

ct = cutree(hc, 6) # cut tree into 6 pieces

n = length(labels) # number of bird species

dend = as.dendrogram(hc)

circos.par(cell.padding = c(0, 0, 0, 0))

circos.initialize("a", xlim = c(0, n)) # only one sector

circos.track(ylim = c(0, 1), bg.border = NA, track.height = 0.3,

panel.fun = function(x, y) {

for(i in seq\_len(n)) {

circos.text(i-0.5, 0, labels[i], adj = c(0, 0.5),

facing = "clockwise", niceFacing = TRUE,

col = ct[labels[i]], cex = 0.5)

}

})

suppressPackageStartupMessages(library(dendextend))

dend = color\_branches(dend, k = 6, col = 1:6)

dend\_height = attr(dend, "height")

circos.track(ylim = c(0, dend\_height), bg.border = NA,

track.height = 0.4, panel.fun = function(x, y) {

circos.dendrogram(dend,facing="outside")#circos.dendrogram(dend, facing = "inside")反向发育树

})

circos.clear()

#5.4手动创建热图circos.heatmap()

#6.circos.heatmap()热图功能

#普通热图

set.seed(123)

mat1 = rbind(cbind(matrix(rnorm(50\*5, mean = 1), nr = 50), #cbind按列合并，rbind按行合并

matrix(rnorm(50\*5, mean = -1), nr = 50)),

cbind(matrix(rnorm(50\*5, mean = -1), nr = 50), #五十行5列的矩阵

matrix(rnorm(50\*5, mean = 1), nr = 50))

)

rownames(mat1) = paste0("R", 1:100)

colnames(mat1) = paste0("C", 1:10)

mat1 = mat1[sample(100, 100), ] # randomly permute rows 随机换行

split = sample(letters[1:5], 100, replace = TRUE)

split = factor(split, levels = letters[1:5])

library(ComplexHeatmap)#此包使用bioconductor下载

Heatmap(mat1, row\_split = split)

#6.1输入数据

library(circlize) # >= 0.4.10

col\_fun1 = colorRamp2(c(-2, 0, 2), c("blue", "white", "red"))#colorRamp2设置图的颜色

circos.heatmap(mat1, split = split, col = col\_fun1)#split用于分组，col设定颜色

circos.clear()#一定不能不写

#6.2圆形布局

circos.par(start.degree = 90, gap.degree = 10)#起始角度；间隔角度

circos.heatmap(mat1, split = split, col = col\_fun1, track.height = 0.4, #track.height表示扇形宽度

bg.border = "green",

bg.lwd = 2, #线条宽度

bg.lty = 2, #线条类型

show.sector.labels = TRUE)#是否展示扇形的标签

circos.clear()#一定不能不写

#6.3树状图Dendrograms和行名

circos.heatmap(mat1, split = split, col = col\_fun1,

dend.side = "inside",#dend.side控制树的位置（内外）

rownames.side = "outside",#行名的位置（与树不能在同一侧）

rownames.col = 1:nrow(mat1) %% 10 + 1,#文字的颜色

rownames.cex = runif(nrow(mat1), min = 0.3, max = 2),#文字的缩放倍数

rownames.font = 1:nrow(mat1) %% 4 + 1)#文字的字体样式

circos.clear()

#6.4聚类：默认情况下，数值矩阵按行聚集。cluster参数可以设置为FALSE关闭集群。当然，当cluster设置为时FALSE，即使设置了也不会绘制树状图dend.side。

#不支持矩阵列上的聚类

#6.5树状图的回调

library(dendextend)

dend\_col = structure(1:5, names = letters[1:5])

circos.heatmap(mat1, split = split, col = col\_fun1, dend.side = "inside",

dend.track.height = 0.2,#树状图的轨道高度

dend.callback = function(dend, m, si) {#dend：当前扇区的树状图。m：当前扇区对应的子矩阵。si：当前扇区的扇区索引（或扇区名称）。

# when k = 1, it renders one same color for the whole dendrogram

color\_branches(dend, k = 1, col = dend\_col[si])#为五个扇区中的树状图分配不同的颜色

}

)

circos.clear()

#6.6多个热图轨迹

mat2 = mat1[sample(100, 100), ] # 随机打乱数据

col\_fun2 = colorRamp2(c(-2, 0, 2), c("green", "white", "red"))#设置第二个数据的参数

circos.heatmap.initialize(mat1, split = split)#根据mat1生成树

circos.heatmap(mat2, col = col\_fun2, dend.side = "outside")

circos.heatmap(mat1, col = col\_fun1)#虽然mat1在第二个轨道，树依旧根据第mat1生成

#6.7与其他轨道叠加

#CELL\_META$dend：当前扇区中的树状图。如果没有进行聚类，则值为NULL。

#CELL\_META$order：聚类后当前扇区中子矩阵的行排序。如果没有进行聚类，则值为c(1, 2, ..., )

#CELL\_META$subset：原始完整矩阵中的索引子集。值按递增排序。

#与散点图结合

circos.heatmap(mat1, split = split, col = col\_fun1)

row\_mean = rowMeans(mat1[, 1:5])

circos.track(ylim = range(row\_mean), panel.fun = function(x, y) {

y = row\_mean[CELL\_META$subset]

y = y[CELL\_META$row\_order]

circos.lines(CELL\_META$cell.xlim, c(0, 0), lty = 2, col = "grey")

circos.points(seq\_along(y) - 0.5, y, col = ifelse(y > 0, "red", "blue"))

}, cell.padding = c(0.02, 0, 0.02, 0))#添加cell.padding = c(0.02, 0, 0.02, 0)这样最大和最小点不会与单元格的顶部和底部边界重叠。

circos.clear()

#与箱线图结合

circos.heatmap(mat1, split = split, col = col\_fun1)

circos.track(ylim = range(mat1), panel.fun = function(x, y) {

m = mat1[CELL\_META$subset, 1:5, drop = FALSE]

m = m[CELL\_META$row\_order, , drop = FALSE]

n = nrow(m)

# circos.boxplot 用于矩阵列, 因此我们需要转置.

circos.boxplot(t(m), pos = 1:n - 0.5, pch = 16, cex = 0.3)#pos位置；pch设置点的形状

circos.lines(CELL\_META$cell.xlim, c(0, 0), lty = 2, col = "grey")

}, cell.padding = c(0.02, 0, 0.02, 0))

circos.clear()

#6.8添加注释

#设置自己的扇形标签

circos.heatmap(mat1, split = split, col = col\_fun1)

circos.track(track.index = get.current.track.index(), #设置track.index = get.current.track.index()以确保始终将标签添加到正确的轨道中

panel.fun = function(x, y) {

circos.text(CELL\_META$xcenter,#x轴的数据点

CELL\_META$cell.ylim[2] + convert\_y(2, "mm"), #y轴的数据点

paste0("this is group ", CELL\_META$sector.index),#每个点的标签

facing = "bending.inside",#文字的朝向

cex = 0.8,#字体大小

adj = c(0.5, 0), #文本的偏移度

niceFacing = TRUE)#文字的朝向是否应该被调整到适合人眼

}, bg.border = NA)#不要边框

circos.clear()

#添加矩阵列名

circos.par(gap.after = c(2, 2, 2, 2, 10))#设置扇形之间的间隔，10的间隔用于存放列名

circos.heatmap(mat1, split = split, col = col\_fun1, track.height = 0.4)

circos.track(track.index = get.current.track.index(), panel.fun = function(x, y) {

if(CELL\_META$sector.numeric.index == 5) { # 如果在最后一个扇形后

cn = colnames(mat1)

n = length(cn)

circos.text(rep(CELL\_META$cell.xlim[2], n) + convert\_x(1, "mm"),

1:n - 0.5, cn,

cex = 0.5, adj = c(0, 0.5), facing = "inside")

}

}, bg.border = NA)

#生成图例，使用complexheatmap包

library(ComplexHeatmap)

lgd = Legend(title = "mat1", col\_fun = col\_fun1)

grid.draw(lgd)

circos.clear()

#7.高级布局

#7.1扇区缩放

#首先生成一个包含六个类别的数据框。

set.seed(123)

df = data.frame(

sectors = sample(letters[1:6], 400, replace = TRUE),

x = rnorm(400),

y = rnorm(400),

stringsAsFactors = FALSE

)

#缩放扇区 a 和扇区 b 的前 10 个点。首先我们提取这些数据并格式化为一个新的数据框。

zoom\_df\_a = df[df$sectors == "a", ]

zoom\_df\_b = df[df$sectors == "b", ]

zoom\_df\_b = zoom\_df\_b[order(zoom\_df\_b[, 2])[1:10], ]#order默认升序 a[order(a)]返回排序好的数据

zoom\_df = rbind(zoom\_df\_a, zoom\_df\_b)#rbind按行连接

#更改缩放数据框中的扇区名称。在这里，我们只是简单地将“zoom\_”前缀添加到原始名称中，以表明它们是“缩放”扇区。之后，将其附加到原始数据框。

zoom\_df$sectors = paste0("zoom\_", zoom\_df$sectors)

df2 = rbind(df, zoom\_df)

#对正常扇区的宽度进行归一化，并分别对缩放扇区的宽度进行归一化。注意现在原始扇区的扇区宽度之和为 1，缩放扇区的扇区宽度之和为 1，这意味着这两种类型的扇区有自己的半圆。

xrange = tapply(df2$x, df2$sectors, function(x) max(x) - min(x))#tapply(x,y,z)按y对x分组，并且对分组后的x进行z操作

normal\_sector\_index = unique(df$sectors)#unique返回向量、数据框或数组，但删除重复元素/行。

zoomed\_sector\_index = unique(zoom\_df$sectors)

sector.width = c(xrange[normal\_sector\_index] / sum(xrange[normal\_sector\_index]),

xrange[zoomed\_sector\_index] / sum(xrange[zoomed\_sector\_index]))#计算每个类别归一化后的长度

sector.width

circos.par(start.degree = 90,

points.overflow.warning = FALSE)#由于扇区名称被添加到第一个轨道之外， points.overflow.warning因此设置为FALSE关闭警告消息。

circos.initialize(df2$sectors, x = df2$x, sector.width = sector.width)

circos.track(df2$sectors, x = df2$x, y = df2$y,

panel.fun = function(x, y) {

circos.points(x, y, col = "red", pch = 16, cex = 0.5)#pch点的形状

circos.text(CELL\_META$xcenter, CELL\_META$cell.ylim[2] + mm\_y(2), #xlim[2] ylim[2]表示横纵坐标的终点

CELL\_META$sector.index, niceFacing = TRUE)#niceFacing是否调整文本使其适合人眼

})

#添加圆形图中的连接

circos.link("a", #sector.index1一个链路端所在的第一个扇区的索引

get.cell.meta.data("cell.xlim", sector.index = "a"),#point1单个值或长度为2的数值向量。如果是2元素向量，则链接将是皮带/色带。

"zoom\_a", #sector.index2另一个链路端所在的第一个扇区的索引

get.cell.meta.data("cell.xlim", sector.index = "zoom\_a"),#point2

border = NA, #如果连接是一个条带，它就是条带边界的颜色

col = "#00000020")#连接的颜色

circos.link("b",

c(zoom\_df\_b[1, 2], zoom\_df\_b[10, 2]),#连接起始处的位置

"zoom\_b",

get.cell.meta.data("cell.xlim", sector.index = "zoom\_b"),#连接终止处的位置

rou1 = get.cell.meta.data("cell.top.radius", sector.index = "b"),#连接的终止位置

border = NA,

col = "#999999")

circos.clear()

#7.2可视化圆的一部分

#圆形图始终绘制在画布中，其中 x 值范围从 -1 到 1，y 值范围从 -1 到 1。

#因此，如果canvas.xlim和canvas.ylim 全部设置为c(0, 1)，这意味着画布受到限制到右上方，则只能看到 0 到 90 度之间的扇区。

#1直接指定数据子集,然后首先创建空轨迹，然后在单元格中填充图形

sectors = letters[1:4]

circos.initialize(sectors, xlim = c(0, 1))

df = data.frame(sectors = rep("a", 100),

x = runif(100),#runif(100) 随机生成0-1之间的100个数

y = runif(100))

circos.track(df$sectors, x = df$x, y = df$y,

panel.fun = function(x, y) {

circos.points(x, y, pch = 16, cex = 0.5)

})

circos.track(ylim = range(df$y), bg.border = NA)

circos.update(sector.index = "a", #update 更新一个已经存在的画图地区

bg.border = "black")#更新区域的边界颜色

circos.points(df$x, df$y, pch = 16, cex = 0.5)

circos.track(sectors = sectors, ylim = c(0, 1))

circos.track(sectors = sectors, ylim = c(0, 1))

circos.clear()

#7.3组合多个圆图

#par(new = TRUE)它允许直接在前一个画布区域上绘制一个新图形作为新层。

#设置不同的canvas.xlim canvas.ylim，它允许制作包含多个圆形图的更复杂的图。

#例一

sectors = letters[1:4]

circos.initialize(sectors, xlim = c(0, 1))

circos.track(ylim = c(0, 1), panel.fun = function(x, y) {

circos.text(0.5, 0.5, "outer circos", niceFacing = TRUE)

})

circos.clear()

#以上第一个圈

par(new = TRUE) # 直接在前一个画布区域上绘制一个新图形作为新层。

circos.par("canvas.xlim" = c(-2, 2), "canvas.ylim" = c(-2, 2))#canvas.xlim canvas.ylim，它允许制作包含多个圆形图的更复杂的图

sectors = letters[1:3]

circos.initialize(sectors, xlim = c(0, 1))

circos.track(ylim = c(0, 1), panel.fun = function(x, y) {

circos.text(0.5, 0.5, "inner circos", niceFacing = TRUE)

})

circos.clear()

#例二 制作了四个圆形图，其中每个图只绘制一个扇区。

sectors = letters[1:4]

lim = c(1, 1.1, 1.2, 1.3)

for(i in 1:4) {

circos.par("canvas.xlim" = c(-lim[i], lim[i]), #画布的x坐标范围

"canvas.ylim" = c(-lim[i], lim[i]), #画布的y坐标范围

"track.height" = 0.4)#轨道宽度

circos.initialize(sectors, xlim = c(0, 1))

circos.track(ylim = c(0, 1), bg.border = NA)

circos.update(sector.index = sectors[i], bg.border = "black")

circos.points(runif(10), runif(10), pch = 16)

circos.clear()

par(new = TRUE)#允许在原来的圆上继续添加图层

}

par(new = FALSE)

#7.4在一页上排列多个圆形图：layout()或par("mfrow")/par("mfcol")在一页中排列多个圆形图

layout(matrix(1:9, 3, 3))#一页上排列9个图3\*3

for(i in 1:9) {

sectors = 1:8

par(mar = c(0.5, 0.5, 0.5, 0.5))

circos.par(cell.padding = c(0, 0, 0, 0))# 单元格填充：该参数有四个值，分别控制底部、左侧、顶部和右侧填充。第一和第三填充值是根据单位圆半径的百分比，第二和第四值是度。

circos.initialize(sectors, xlim = c(0, 1))

circos.track(ylim = c(0, 1), track.height = 0.05,

bg.col = rand\_color(8), bg.border = NA)

for(i in 1:20) {

se = sample(1:8, 2)#从1-8抽样，每次抽两个

circos.link(se[1],

runif(2),

se[2],

runif(2),

col = rand\_color(1, transparency = 0.4), #rand\_color生成随机颜色 1表示生成

border = NA)

}

circos.clear()

}

Dplyr包

library(dplyr)

#vignette("dplyr")

starwars

#1行

#1.1filter 根据列值选择行

starwars %>% filter(skin\_color == "light", eye\_color == "brown")#筛选出skincolor为light,eyecolor为brown的行

#1.2slice 根据位置选择行

starwars %>% slice(5:10)#获取第五到第十行的数据

starwars %>% slice\_head(n = 3)#获取头三行

starwars %>% slice\_tail(n = 3)#获取尾三行

starwars %>% slice\_sample(n = 5)#随机获取5行

starwars %>% slice\_sample(prop = 0.1)#随机获取1/10行

starwars %>% filter(!is.na(height)) %>% slice\_max(height, n = 3)#先确定height不为NA，随后挑出3个height最大的行

starwars %>% filter(!is.na(height)) %>% slice\_min(height, n = 3)#挑出3个height最小的行

#1.3arrange 更改行的顺序（默认升序）

starwars %>% arrange(height, mass)#先按height排，height相同按mass排

starwars %>% arrange(desc(height))#降序排列

#2 列

#2.1 select 选择展示的列

starwars %>% select(hair\_color, skin\_color, eye\_color)#列名选择

starwars %>% select(hair\_color:eye\_color)#选择hair\_color到eye\_color之间的列

starwars %>% select(!(hair\_color:eye\_color))#选择除去hair\_color到eye\_color之间的列

starwars %>% select(ends\_with("color"))#选择以color结尾的列名的列；starts\_with\ends\_with\matches\contains

#2.2 rename 更改列的名称

starwars %>% rename(home\_world = homeworld)#将列名homeworld改为home\_world

#2.3 mutate 创建新列

starwars %>% mutate(height\_m = height / 100)#创建新列并且为其赋值

starwars %>%

mutate(height\_m = height / 100) %>%

select(starts\_with("height"))

starwars %>%

transmute(

height\_m = height / 100)#只要新增的列用transmute

#2.4 relocate 更改列的顺序

starwars %>% relocate(sex:homeworld, .before = height)#将sex到homeworld之间的列放到height前

#3行组

#summarise 将一个组折叠成一行

starwars %>% summarise(height = mean(height, na.rm = TRUE))#以均值身高替代原来的身高

#%>%是连接符号，用于连接操作

starwars %>%

group\_by(species, sex) %>%#根据species和sex分组

select(height, mass) %>%

summarise(

height = mean(height, na.rm = TRUE),

mass = mean(mass, na.rm = TRUE)

)

#将dataframe转化为tibble

data <- data.frame(a = 1:3, b = letters[1:3], c = Sys.Date() - 1:3)

as\_tibble(data)

ti<-tibble(x = 1:5, y = 1, z = x ^ 2 + y)#创建tibble，tibble和dataframe都是按列创建

as.data.frame(ti)#tibble转化为dataframe

#14和弦图

#和弦图从多个层次显示关系的信息。1. 链接直接显示对象之间的关系；2.链接的宽度与关系的强度成正比，比其他图形映射更具说明性；3. 链接的颜色可以是另一种关系的图形映射；4. 扇形宽度表示一个物体与其他物体相连或与其他物体相连的总强度。

#创建一个矩阵

set.seed(999)

mat = matrix(sample(18, 18), 3, 6) #从1-18中抽样，组成3行六列的数据

rownames(mat) = paste0("S", 1:3)

colnames(mat) = paste0("E", 1:6)

mat

df = data.frame(from = rep(rownames(mat), times = ncol(mat)),#rep(x, time = , length.out = , each = ,) times是复制的次数,复制是指的是对整个向量进行复制

to = rep(colnames(mat), each = nrow(mat)),#each：代表的是对向量中的每个元素进行复制的次数。

value = as.vector(mat),#将矩阵元素向量化, 注意是逐列转换

stringsAsFactors = FALSE)

df

chordDiagram(mat)

#或者用df

chordDiagram(df)

circos.clear()

#两个和弦图

par(mfrow = c(1, 2))

chordDiagram(mat, order = c("S2", "S1", "S3", "E4", "E1", "E5", "E2", "E6", "E3"))

circos.clear()

chordDiagram(mat, order = c("S2", "S1", "E4", "E1", "S3", "E5", "E2", "E6", "E3"))

circos.clear()

#14.2调整依据circos.par()

#调整组间间距

#1

circos.par(gap.after = c(rep(5, nrow(mat)-1), 15, rep(5, ncol(mat)-1), 15))#间隙为5，5，15，5，5，5，5，15

chordDiagram(mat)

circos.clear()

#2

circos.par(gap.after = c(rep(5, length(unique(df[[1]]))-1), 15,

rep(5, length(unique(df[[2]]))-1), 15))

chordDiagram(df)

circos.clear()

#3

circos.par(gap.after = c("S1" = 5, "S2" = 5, "S3" = 15, "E1" = 5, "E2" = 5,

"E3" = 5, "E4" = 5, "E5" = 5, "E6" = 15))

chordDiagram(mat)

circos.clear()

#4

chordDiagram(mat, big.gap = 30)#仅设置行与列之间间距为30

circos.clear()

#多图

par(mfrow = c(1, 2))

circos.par(start.degree = 85, clock.wise = FALSE)#从85°开始画，且按照逆时针顺序

chordDiagram(mat)

circos.clear()

circos.par(start.degree = 85)#默认是顺时针

chordDiagram(mat, order = c(rev(colnames(mat)), rev(rownames(mat))))#rev对向量经行逆序处理

circos.clear()

#14.3 颜色

#设置网格颜色grid.col

grid.col = c(S1 = "red", S2 = "green", S3 = "blue",

E1 = "grey", E2 = "grey", E3 = "grey", E4 = "grey", E5 = "grey", E6 = "grey")#grid.col假定的颜色与扇形区相同

chordDiagram(mat, grid.col = grid.col)

#设置连接颜色

col\_mat = rand\_color(length(mat), transparency = 0.5)#随机颜色

dim(col\_mat) = dim(mat) # to make sure it is a matrix

chordDiagram(mat, grid.col = grid.col, col = col\_mat)#col连接颜色和透明度

#关系的强度（例如相关性）表示为连续值时， col也可以指定为自定义颜色映射函数

col\_fun = colorRamp2(range(mat), c("#CCFFFF", "#CCCCFF"), transparency = 0.5)

chordDiagram(mat, grid.col = grid.col, col = col\_fun)

#按照行、列设置连接颜色

chordDiagram(mat, grid.col = grid.col, row.col = 1:3)

chordDiagram(mat, grid.col = grid.col, column.col = 1:6)

#14.4连接边框link.lwd，link.lty并link.border控制链接边框的线宽、线型和颜色。

chordDiagram(mat, grid.col = grid.col,

link.lwd = 1, #线宽

link.lty = 1, #线型

link.border = "#CCFFFF")#颜色

#按照数据设置边框

lwd\_mat = matrix(1, nrow = nrow(mat), ncol = ncol(mat))#1是填充的数据，nrow、ncol是行、列数

lwd\_mat[mat > 12] = 2#mat>12返回TRUE FALSE 只填充返回TRUE的内容

border\_mat = matrix(NA, nrow = nrow(mat), ncol = ncol(mat))#NA表示不填充内容

border\_mat[mat > 12] = "red"

#注意matrix必须要有行、列名才能正确映射到图中

chordDiagram(mat, grid.col = grid.col, link.lwd = lwd\_mat, link.border = border\_mat)

#将图形参数设置为三列数据框，其中前两列对应矩阵中的行名和列名，第三列对应图形参数

lty\_df = data.frame(c("S1", "S2", "S3"), c("E5", "E6", "E4"), c(1, 2, 3))#dataframe是按列创建的

lwd\_df = data.frame(c("S1", "S2", "S3"), c("E5", "E6", "E4"), c(2, 2, 2))

border\_df = data.frame(c("S1", "S2", "S3"), c("E5", "E6", "E4"), c(1, 1, 1))

chordDiagram(mat, grid.col = grid.col, link.lty = lty\_df, link.lwd = lwd\_df,

link.border = border\_df)

#14.5 高亮链接

col\_fun = function(x) ifelse(x < 12, "#00000000", "#FF000080")#x<12就为透明，否则有颜色

chordDiagram(mat, grid.col = grid.col, col = col\_fun)

#或者使用矩阵映射

col\_df = data.frame(c("S1", "S2", "S3"), c("E5", "E6", "E4"),

c("#FF000080", "#00FF0080", "#0000FF80"))

chordDiagram(mat, grid.col = grid.col, col = col\_df)

#14.6链接顺序：link.sort和link.decreasing设置控制扇区上定位链路的顺序

chordDiagram(mat, grid.col = grid.col, link.sort = TRUE, link.decreasing = TRUE)

title("link.sort = TRUE, link.decreasing = TRUE", cex = 0.6)

chordDiagram(mat, grid.col = grid.col, link.sort = TRUE, link.decreasing = FALSE)

title("link.sort = TRUE, link.decreasing = FALSE", cex = 0.6)

#14.7链接的z-index link.zindex定义在另一个之上添加链接的顺序的参数来设置。较大的值意味着稍后添加相应的链接

chordDiagram(mat, grid.col = grid.col, transparency = 0)

chordDiagram(mat, grid.col = grid.col, transparency = 0, link.zindex = rank(mat))#较大的值较后链接，用于数据框

#from to框则数据也类似

chordDiagram(df, grid.col = grid.col, transparency = 0, link.zindex = rank(df[[3]]))

#14.8自链接 self.link=1或者2

df2 = data.frame(start = c("a", "b", "c", "a"), end = c("a", "a", "b", "c"))

chordDiagram(df2, grid.col = 1:3, self.link = 1)

title("self.link = 1")

chordDiagram(df2, grid.col = 1:3, self.link = 2)

title("self.link = 2")

#14.9对称矩阵

mat3 = matrix(rnorm(25), 5)

colnames(mat3) = letters[1:5]

cor\_mat = cor(mat3)#相关系数矩阵

col\_fun = colorRamp2(c(-1, 0, 1), c("green", "white", "red"))

chordDiagram(cor\_mat, grid.col = 1:5, symmetric = TRUE, col = col\_fun)#设置symmetric = TRUE，只使用没有对角线的下三角矩阵

title("symmetric = TRUE")

chordDiagram(cor\_mat, grid.col = 1:5, col = col\_fun)

title("symmetric = FALSE")

#14.10方向关系 directional=1方向从行到列 2双向 -1从列到行，起始处较短

par(mfrow = c(1, 3))

chordDiagram(mat, grid.col = grid.col, directional = 1)

chordDiagram(mat, grid.col = grid.col, directional = 1, diffHeight = mm\_h(5))

chordDiagram(mat, grid.col = grid.col, directional = -1)

#用箭头表示链接方向

arr.col = data.frame(c("S1", "S2", "S3"), c("E5", "E6", "E4"),

c("black", "black", "black"))#这里颜色不能改

chordDiagram(mat, grid.col = grid.col,

directional = 1, #方向从行到列

direction.type = "arrows",#arrows表示绘制箭头

link.arr.col = arr.col, #根据arr.col矩阵画箭头

link.arr.length = 0.2)

#结合箭头和不等高

arr.col = data.frame(c("S1", "S2", "S3"), c("E5", "E6", "E4"),

c("black", "black", "black"))

chordDiagram(mat, grid.col = grid.col, directional = 1,

direction.type = c("diffHeight", #出发点短，设为负值则出发点长

"arrows"),#箭头

link.arr.col = arr.col, #箭头矩阵

link.arr.type = "big.arrow",#当箭头很多时使用大箭头

link.arr.length = 0.2)#箭头宽度

#14.11缩放

#scale=TRUE在每个扇区上进入另一个扇区的交互的比例（被进入的扇区大小将相等）

set.seed(999)

mat = matrix(sample(18, 18), 3, 6)

rownames(mat) = paste0("S", 1:3)

colnames(mat) = paste0("E", 1:6)

grid.col = c(S1 = "red", S2 = "green", S3 = "blue",

E1 = "grey", E2 = "grey", E3 = "grey", E4 = "grey", E5 = "grey", E6 = "grey")

par(mfrow = c(1, 2))

chordDiagram(mat, grid.col = grid.col)#E的大小不相等，列未缩放

title("Default")

chordDiagram(mat, grid.col = grid.col, scale = TRUE)#缩放列

title("scale = TRUE")

#14.12将某行、列设置为0以删除扇区

#冲积图

install.packages("ggalluvial")

vignette("ggalluvial")

library(ggalluvial)

#Alluvial冲击

#1冲击数据

head(as.data.frame(UCBAdmissions), n = 12)#三维数据

#加州大学伯克利分校录取和拒绝，按性别和部门

ggplot(as.data.frame(UCBAdmissions),

aes(y = Freq, axis1 = Gender, axis2 = Dept)) +#坐标轴y是聘书，第一根轴是性别，第二根轴是部门

geom\_alluvium(aes(fill = Admit), width = 1/12) +#冲击层信息

geom\_stratum(width = 1/12, fill = "black", color = "grey") +#小方块信息

geom\_label(stat = "stratum", aes(label = after\_stat(stratum))) +#geom\_label（）在文本后面绘制一个矩形，便于阅读。

scale\_x\_discrete(limits = c("Gender", "Dept"), expand = c(.05, .05)) +#用于设置离散x尺度美学的值。expand对于位置刻度，范围扩展常数的向量，用于在数据周围添加一些填充，以确保它们与轴保持一定距离。

scale\_fill\_brewer(type = "qual", palette = "Set1") +#scale\_fill\_brewer：配色方案；typetype: seq (sequential), div (diverging) 或 qual (qualitative)

ggtitle("UC Berkeley admissions and rejections, by sex and department")

#Titanic的生存情况冲积图

ggplot(as.data.frame(Titanic),

aes(y = Freq,

axis1 = Survived, axis2 = Sex, axis3 = Class)) +

geom\_alluvium(aes(fill = Class),#根据class来填色

width = 1/5, knot.pos = 0, reverse = FALSE) +#冲积层信息

guides(fill = "none") +#可以使用引导参数按比例设置每个比例的引导线

geom\_stratum(width = 1/5, reverse = FALSE) +#小方块 width：方块宽度

geom\_text(stat = "stratum", aes(label = after\_stat(stratum)),

reverse = FALSE) +

scale\_x\_continuous(breaks = 1:3, labels = c("Survived", "Sex", "Class")) +# scale\_x\_continuous连续x美学的默认比例。

#coord\_flip() +#翻转坐标

ggtitle("Titanic survival by class and sex")

#长格式

head(as.data.frame(UCBAdmissions), n = 12)#三维数据

UCB\_lodes <- to\_lodes\_form(as.data.frame(UCBAdmissions),#to\_lodes\_form将具有若干指定变量的数据帧用作冲积地块中的轴，并对数据帧进行整形，以使轴变量名称构成一个新的因子变量，其值构成另一个因子变量。其他变量的值将重复，并且可以引入行分组变量。

axes = 1:3,

id = "Cohort")

head(UCB\_lodes, n = 12)

is\_lodes\_form(UCB\_lodes, key = x, value = stratum, id = Cohort, silent = TRUE)

#难民图:四张图

#install.packages("alluvial")

data(Refugees, package = "alluvial")

country\_regions <- c(

Afghanistan = "Middle East",

Burundi = "Central Africa",

`Congo DRC` = "Central Africa",

Iraq = "Middle East",

Myanmar = "Southeast Asia",

Palestine = "Middle East",

Somalia = "Horn of Africa",

Sudan = "Central Africa",

Syria = "Middle East",

Vietnam = "Southeast Asia"

)

Refugees$region <- country\_regions[Refugees$country]

ggplot(data = Refugees,

aes(x = year, y = refugees, alluvium = country)) +

geom\_alluvium(aes(fill = country, colour = country),

alpha = .75, decreasing = FALSE) +

scale\_x\_continuous(breaks = seq(2003, 2013, 2)) +

theme\_bw() +

theme(axis.text.x = element\_text(angle = -30, hjust = 0)) +

scale\_fill\_brewer(type = "qual", palette = "Set3") +

scale\_color\_brewer(type = "qual", palette = "Set3") +

facet\_wrap(~ region, scales = "fixed") +

ggtitle("refugee volume by country and region of origin")

#学生课程图

data(majors)

majors$curriculum <- as.factor(majors$curriculum)

ggplot(majors,

aes(x = semester, #横轴为semester

stratum = curriculum, #小方块（层）的名称curriculum

alluvium = student,#冲积层是学生

fill = curriculum,

label = curriculum)) +

scale\_fill\_brewer(type = "qual", palette = "Set2") +#颜色排布

geom\_flow(stat = "alluvium", #geom\_flow接收冲积地块矿脉的水平（x）和垂直（y，ymin，ymax）位置的数据集，冲积物与地层的交叉点。

lode.guidance = "frontback",#去了不影响

color = "darkgray") +

geom\_stratum() +#geom\_stratum接收冲积地块地层的水平（x）和垂直（y，ymin，ymax）位置的数据集。

theme(legend.position = "bottom") +

ggtitle("student curricula across several semesters")

#三个时间段的疫苗反映

data(vaccinations)

vaccinations <- transform(vaccinations,

response = factor(response, rev(levels(response))))#rev对向量做逆序处理

ggplot(vaccinations,

aes(x = survey, #横轴是哪个调查

stratum = response, #小方块（层）是response

alluvium = subject,#冲积层是人员编号

y = freq,#纵轴是频率freq

fill = response, #填充颜色

label = response)) +#小方块上的标签

scale\_x\_discrete(expand = c(.1, .1)) +#离散x，离轴0.1

geom\_flow() +

geom\_stratum(alpha = .5) +#alpha是透明度

geom\_text(stat = "stratum", size = 3) +#小方块上的字大小和内容

theme(legend.position = "bottom") +#小方块标签的位置，不要标签"none"

ggtitle("vaccination survey responses at three points in time")