第二章

类型分类

数值向量

逻辑向量

字符向量

向量运算

与常数的四则运算

数值向量之间的运算

字符向量的运算函数

nchar()、paste()、strsplit()、substr()、sub()

matrix()

matrix(data, nrow=?, nol=?, byrow=FALSE, dimnames=NULL)

byrow为FALSE时，表示按列填充；byrow为TRUE时，表示按行填充

dimnames设置行名和列名

dimnames = list(c(colnames), c(rownames))

矩阵的建立

现有一向量(1,2,3,4,5,6)，希望得到一个2×3的矩阵。

theData <-c(1,2,3,4,5,6)

mat1<-matrix(theData, 2, 3)

mat2<-matrix(theData, 2, 3, byrow=FALSE)

mat3<-matrix(theData, 2, 3, byrow=TRUE)

mat4<-matrix(theData, 2, 3, byrow=TRUE, dimnames = list(c('1st','2nd'), c('1st','2nd','3rd')))

取得或修改矩阵对象的行名和列名

rownames()：取得和修改矩阵对象的行名

colnames() ：取得和修改矩阵对象的列名

rownames(mat3) <- c(‘1st’, ‘2nd’)

cownames(mat3) <- c(‘1st’, ‘2nd’, ‘3rd’)

mat3[‘1st’, ‘2nd’]

删除行名或列名

rownames(mat3) <-NULL

使用索引取得矩阵元素的值

将多个向量合并成矩阵

rbind()：将两个或多个向量组成矩阵，每个向量各占用一行。

cbind()：将两个或多个向量组成矩阵，每个向量各占用一列。

注意：向量长度相等。如向量对象类型不同，则会转换为字符或者数字。

str()：查看矩阵对象的结构

nrow()：得到矩阵的行数

ncol()：得到矩阵的列数

dim()：得到矩阵的行数和列数

length()：取得矩阵或高维数组对象的元素个数

is.matrix()：检查对象是否是矩阵

is.array()：检查对象是否是Array

rowSums()：计算行中元素的总和

colSums()：计算列中元素的总和

rowMeans()：计算行中元素的平均值

colMeans()：计算列中元素的平均值

矩阵的运算

转置矩阵t()

实现矩阵的行列元素互相对调

%\*%矩阵相乘

矩阵的运算

diag()：当第一个参数是矩阵时，可传回矩阵对角线的向量值。

solve()：可传回反矩阵

det()：计算矩阵的行列式值。

三维或高维数组

array()可用于建立三维数组

identical()比较两个对象是否完全相同。

第五章

factor(x)或as.factor(x)

其中，x向量是欲转换为因子的向量。levels是原x向量内元素的可能值。

因子的转换

as.character()：将因子转换成字符串向量

as.numeric()：将因子转换成数值向量

当要转换的因子为数值型因子时，可以使用as.numeric(as.character())

str()函数查看此因子的结构

levels()函数查看此因子的levels

nlevels()函数可返回Levels的数量

table()函数自动统计在因子的所有元素中，levels中各值出现的次数，其结果按照levels的顺序显示。

有序因子主要是处理有序的数据

可用两种方法建立：

factor()函数，增加参数“ordered=TRUE”

ordered()函数

建立列表list()

使用names()添加元素名称

length()函数获取列表的元素个数

获得列表内对象的元素内容

$ stu.inf2$Year

[[]] stu.inf2[[1]]

[] stu.inf2[1:2]

编辑列表内对象的元素值

修改和添加 $或[[]]

建立第一个数据框

ZS<-c(95.5,85,78)

LS<-c(80,85,75)

WW<-c(82,56,70)

stu.inf2<-data.frame(ZS, LS, WW)

如不想字符串向量变为因子变量，则建立数据框时，增加参数“stringsAsFactors = FALSE”

特殊符号$ 读取数据框列名内的数据 stu.newinfo$stu.name

添加一条列记录 weight<-c(65,71,58,55)

添加行数据和列数据

rbind()函数：添加单行数据

添加多行数据，现将这些数据组合成数据框，在使用rbind()函数将来给你个数据框组合即可。

cbind()：添加单列数据

添加多列数据，同rbind()，建立数据框，再添加。

第七章 后20

软件包graphics 散点图plot 条形图barplot

箱线图boxplot 直方图hist 正态Q-Q图qqnorm

函数curve 散点图

如果在x和y之间有任何关系，plot是看清数据间关系的快捷方法plot(x,y)

在图形窗口绘制(x,y)数据对的图形

plot(CJD$NO,CJD$C)

plot(CJD[,4:6])

plot(CJD$C, type=“l”)

添加网格

grid() #绘制网格

points(x, y) #绘制点

注意：在绘制图形之前，网格必须先绘制。如果先画图形，网格将覆盖图形或者部分隐藏它们。

改变线的类型

参数lty

lty=“solid”或者lty=1(默认)

lty=“dashed”或者lty=2

lty=“dotted”或者lty=3

lty=“dotdash”或者lty=4

lty=“longdash”或者lty=5

lty=“twodash”或者lty=6

lty=“blank”或者lty=0(抑制绘图)

改变线的宽度

参数lwd，默认值为1

改变图的颜色

col

colors<-ifelse(x>0, “black”, “gray”)

plot(x, type=“h”, lwd=3,col=colors)

绘制回归线

绘制数据点对，同时需要添加一条说明它们线性回归关系的线。

m<-lm(y~x) plot(y~x)

abline(m) #绘制最佳拟合回归线

添加垂直线和水平线

abline(v=x) abline(h=y)

m<-mean(CJD$C) abline(h=m)

多组散点图

plot(x, y, pch=as.integer(f))

with(CJD,plot(CJD$NO,CJD$C,pch=as.integer(as.factor(CJD$grade))))

绘制多个数据集

使用plot或者curve来初始化图形

lines和points来添加额外的数据集

xlim<-range(c(x1,x2))

ylim<-range(c(y1,y2))

plot(x1,y1,type=“l”,xlim=xlim,ylim=ylim)

lines(x2,y2,lty=“dashed”)

条形图barplot()

barplot(c(height , height ,…, height))

直方图hist()

在函数hist中，必须决定为了容纳数据需要创建多少直方块

samp<-rgamma(500,2,2)

hist(samp, 20,prob=T)

lines(density(samp)) #近似估计样本密度

箱线图boxplot()

data(UScereal, package="MASS")

boxplot(sugars ~ shelf, data=UScereal)

提供可视化总结，用于因子数据

summary()

boxplot(CJD$JAVA)

boxplot(JAVA~SEX,data=CJD)

正态Q-Q图qqnorm()

data(Cars93,package = "MASS")

qqnorm(Cars93$Price)

qqline(Cars93$Price)#添加对角线

判断数据是否服从正态分布

绘制函数curve()

curve(sin, -3, +3)

-3到+3区间内sin函数的值

curve可以绘制任何接受一个参数并且返回一个值的函数

小技巧

par(ask=TRUE)#图形暂停覆盖

par(mfrow=(c(N,M)))#一页显示多个图

par(mfcol=(c(N,M)))#一页显示多个图

win.graph()#打开第二个图形窗口

savePlot(filename=“.ext”,type=“type”)#将图保存到当前目录

help(Devices)查看可用类型列表

main：标题

xlab：x轴标签

ylab：y轴标签

legend()：添加图例

lty：线的类型

lwd：线的粗细

col：线的颜色

abline(v=x)：添加垂直线

abline(h=y)：添加水平线