# Ch4探索性数据分析

本章概要

**♢ 探索性数据分析的思想**

**♢ 分布的图形概括**

**♢单组数据的描述性统计分析**

**♢多组数据的描述性统计分析**

**♢分组数据的描述性统计分析**

**♢分类数据的描述性统计分析**

**引言：数据的统计分析分为描述性统计分析和统计推断两部分，前者又称为探索性统计分析，它是通过制作统计图形、编制统计表格、计算统计量等方法来探索数据的主要分布特征。本章节针对不同类型的数据，介绍R中探索性数据分析的技巧，分别从图形和统计量两个角度。**

**4.1常用概率函数图（密度曲线图）**

**有两种方法：plot()和curve( )**

* **函数plot(x,y):对x和y作散点图,离散和连续型分布都适用**
* **函数Curve(expr, from = NULL, to = NULL, n = 101, add = FALSE,**

**type = "l", xname = "x", xlab = xname, ylab = NULL,。。):对某个函数在给定区间范围内作曲线图, 只适用于连续型分布，不适用于离散型。**

**如：curve(sin(x),xlim=c(0,2\*pi),col="blue")**

**#等价于curve(sin(x),from=0,to=2\*pi,col="blue")**

**二项分布**

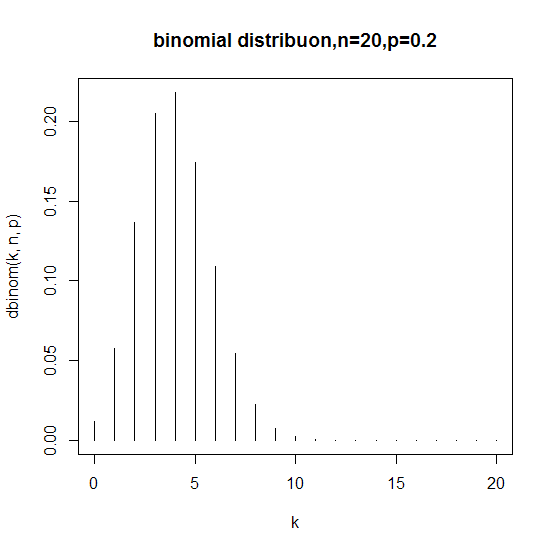
n<-20

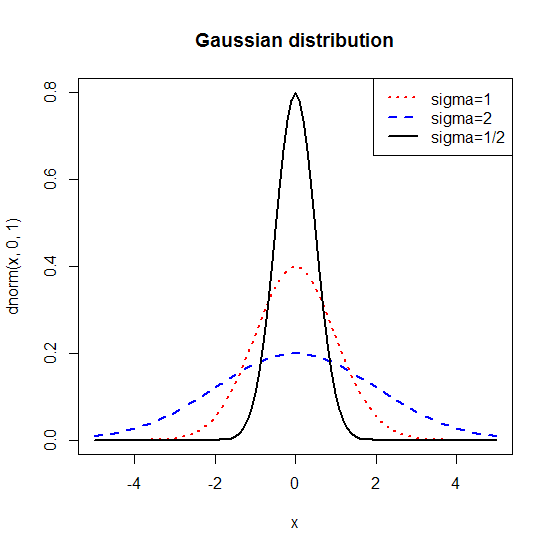
p<-0.2

k<-seq(0,n)

plot(k,dbinom(k,n,p),type="h",main="binomial distribuon,n=20,p=0.2",xlab="k")

curve(dbinom(x,n,p),xlim=c(0,20)) #离散型的不适合



**正态分布** 

x<-seq(-5,5,0.1)

y<-dnorm(x,0,1)

plot(x,y,type="h")

curve(dnorm(x,0,1),xlim=c(-5,5),ylim=c(0,0.8),col="red",lwd=2,lty=3)

curve(dnorm(x,0,2),xlim=c(-5,5),ylim=c(0,0.8),col="blue",lwd=2,lty=2,add=T)

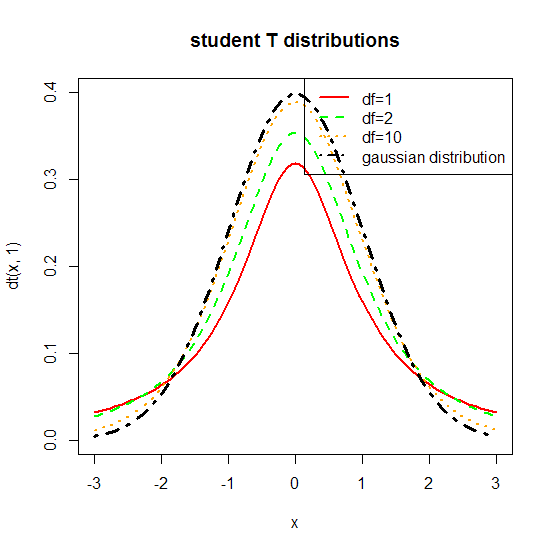
curve(dnorm(x,0,1/2),xlim=c(-5,5),ylim=c(0,0.8),lwd=2,lty=1,add=T)

title(main="Gaussian distribution")

legend(par('usr')[2],par('usr')[4],xjust=1,

c("sigma=1","sigma=2","sigma=1/2"),

lwd=c(2,2,2),lty=c(3,2,1),col=c("red","blue",par("fg")))

**t分布** 

curve(dt(x,1),xlim=c(-3,3),ylim=c(0,0.4),col="red",lwd=2,lty=1)

curve(dt(x,2),col="green",lwd=2,lty=2,add=T)

curve(dt(x,10),col="orange",lwd=2,lty=3,add=T)

curve(dnorm(x),add=T,lwd=3,lty=4)

title(main="student T distributions" )

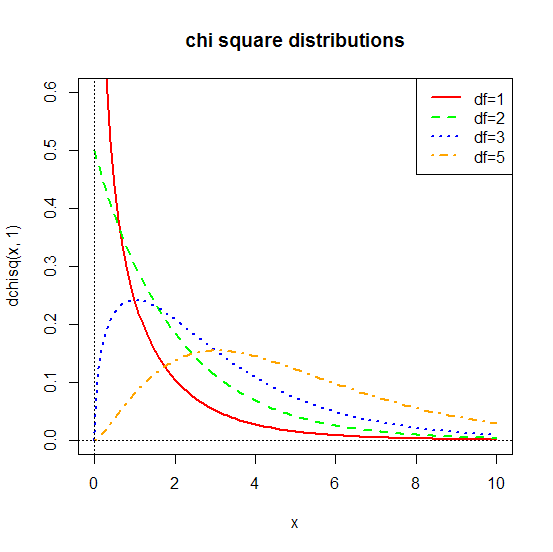
legend(par("usr")[2],par("usr")[4],xjust=1,

c("df=1","df=2","df=10","gaussian distribution"),

lwd=c(2,2,2,2),

lty=c(1,2,3,4),

col=c("red","green","orange",par("fg")))

**x2卡方分布** 

curve(dchisq(x,1),xlim=c(0,10),ylim=c(0,0.6),col="red",lwd=2,lty=1)

curve(dchisq(x,2),col="green",lwd=2,lty=2,add=T)

curve(dchisq(x,3),col="blue",lwd=2,lty=3,add=T)

curve(dchisq(x,5),col="orange",lwd=2,lty=4,add=T)

abline(h=0,lty=3)

abline(v=0,lty=3)

title(main="chi square distributions" )

legend(par("usr")[2],par("usr")[4],xjust=1,

c("df=1","df=2","df=3","df=5"),

lwd=c(2,2,2,2),

lty=c(1,2,3,4),

col=c("red","green","blue","orange"))"

**4.2直方图与核密度估计**

* **函数hist()：作直方图**

**hist(x, breaks = "Sturges", freq = NULL, probability = !freq,**

**col = NULL,main = paste("Histogram of" , xname),**

**xlim = range(breaks), ylim = NULL,**

**xlab = xname, ylab,**

**axes = TRUE, nclass = NULL)**

说明**:** 若选项breaks取向量，则用于指明直方图区间的分割位置;若取正整数, 则用于指定直方图的小区间数. freq取T表示使用频数画直方图, 取F则使用频率画直方图. probability与freq恰好相反.col用于指明小矩形的颜色. 其它选项可参考hist( )的帮助说明.

* **函数density():作样本的核密度估计**

**density(x, bw = "nrd0"，**

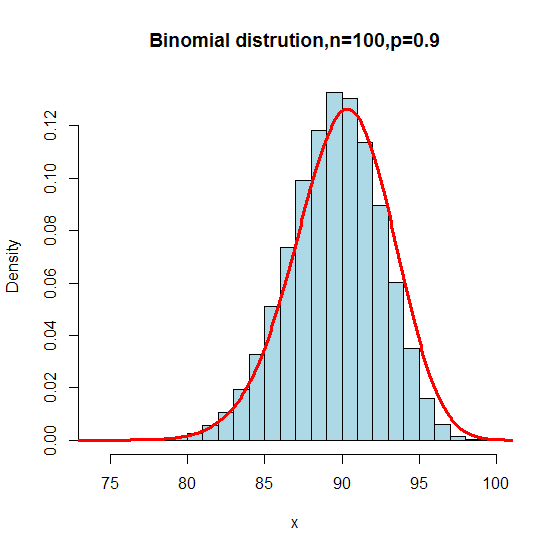
**kernel = c("gaussian", "epanechnikov", "rectangular",**

**"triangular", "biweight", "cosine", "optcosine"),**

**n = 512, from, to)**

说明**:** 选项bw指定核密度估计的窗宽, 也用字符串表示窗宽选择规则,具体可参考函数bw.nrd( ). kernel为核密度估计所使用的光滑化函数, 缺省为正态核函数. n给出等间隔的核密度估计点. from与to分别给出需要计算核密度估计的左右端点。

例1：二项分布binom(100,0.9)中抽取容量为N=100000的样本，试作出它的直方图及核密度估计曲线



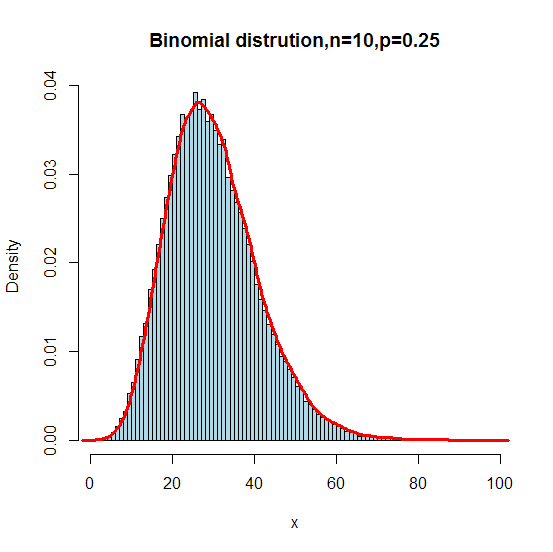
N<-100000;n<-100;p<-0.9

x<-rbinom(N,n,p)

hist(x,xlim=c(min(x),max(x)),prob=T,col="lightblue",

nclass=max(x)-min(x)+1,main="Binomial distrution,n=100,p=0.9")

lines(density(x,bw=1),col="red",lwd=3)

#例2：负二项分布nbinom(10,0.25)中抽取容量为N=100000的样本，试作出它的直方图及核密度估计曲线

N<-100000;n<-10;p<-0.25

x<-rnbinom(N,n,p)

hist(x,xlim=c(min(x),max(x)),prob=T,col="lightblue",

nclass=max(x)-min(x)+1,main="Binomial distrution,n=10,p=0.25")

lines(density(x,bw=1),col="red",lwd=3)

**4.3单组数据的描述性统计分析**

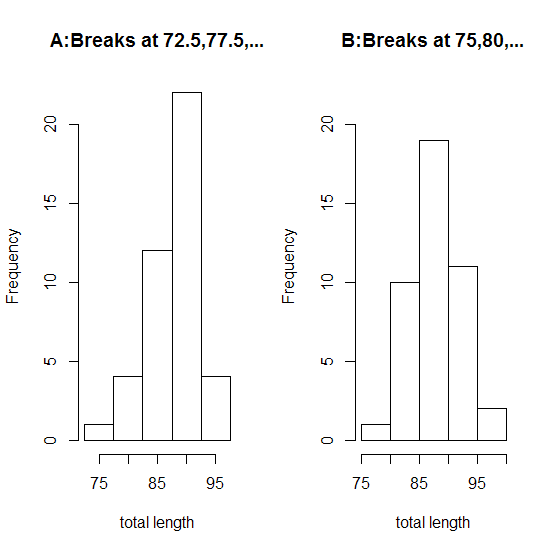
**4.3．1单组数据的图形描述**

主要使用：

* 函数hist():作直方图
* 函数stem()：作茎叶图
* 函数boxplot()：作盒子图

例4.3.1程序包DAAG 中有内嵌数据集“possum”，它包括了从维多利亚南部到皇后区的七个地区的104 只负鼠(possum) 的年龄、尾巴的长度、总长度等9 个特征值，我们仅考虑43 只雌性负鼠的特征值，我们建立子集fpossum, 考查雌性负鼠(fpossum) 的总长度的频率分布.

**直方图**



install.packages("DAAG")

library(DAAG)

data(possum)

fpossum<-possum[possum$sex=="f",]

attach(fpossum)

par(mfrow=c(1,2))

hist(totlngth,breaks=72.5+(0:5)\*5,ylim=c(0,22),

xlab="total length",main="A:Breaks at 72.5,77.5,...")

hist(totlngth,breaks=75+(0:5)\*5,ylim=c(0,22),

xlab="total length",main="B:Breaks at 75,80,...") #直方图的形状与区间端点有关

**茎叶图**

stem(totlngth)

The decimal point is at the |

74 | 0

76 |

78 |

80 | 05

82 | 0500

84 | 05005

86 | 05505

88 | 0005500005555

90 | 5550055

92 | 000

94 | 05

96 | 5

sort(totlngth) #可查看茎叶图的区间是左闭右开

[1] 75.0 81.0 81.5 82.0 82.5 83.0 83.0 84.0 84.5

[10] 85.0 85.0 85.5 86.0 86.5 86.5 87.0 87.5 88.0

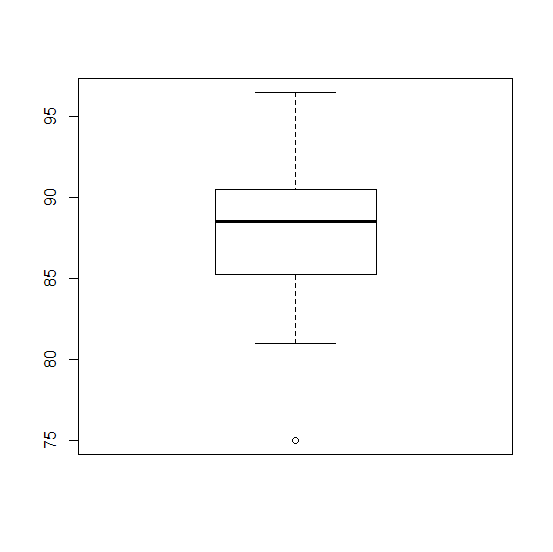
[19] 88.0 88.0 88.5 88.5 89.0 89.0 89.0 89.0 89.5

[28] 89.5 89.5 89.5 90.5 90.5 90.5 91.0 91.0 91.5

[37] 91.5 92.0 92.0 93.0 94.0 95.5 96.5

**盒子图（框须图）**

boxplot(totlngth)



>quantile(totlngth,probs=c(0.25,0.5,0.75)) #计算上下四分位数和中位数

25% 50% 75%

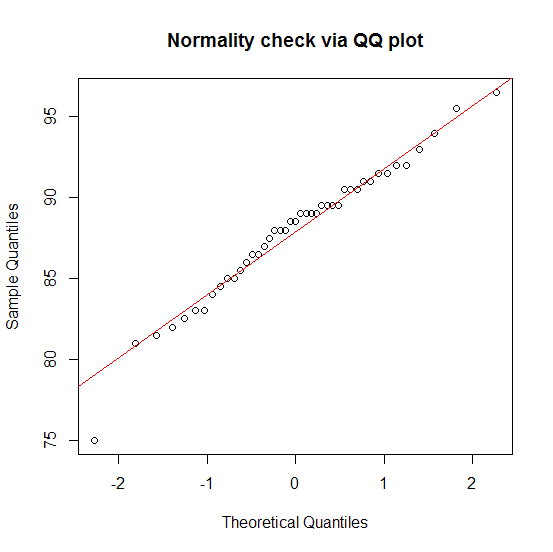
85.25 88.50 90.50

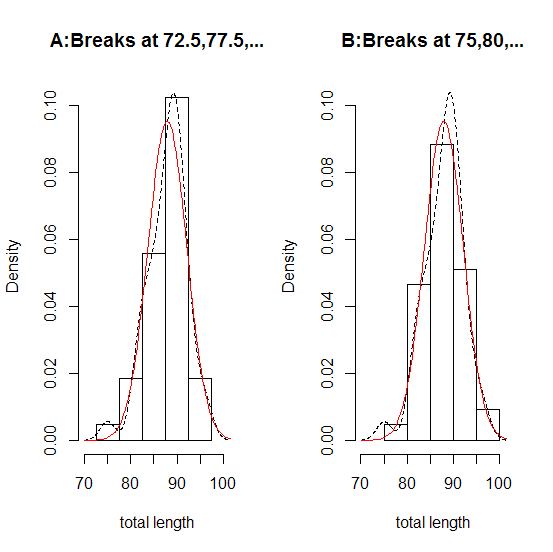
**#正态性检验**

**#1)QQ图**

qqnorm(totlngth,main="Normality check via QQ plot")

qqline(totlngth,col='red')



**#2）与正态密度函数比较**

dens<-density(totlngth) #核密度估计

xlim<-range(dens$x);ylim<-range(dens$y)

par(mfrow=c(1,2))

hist(totlngth,breaks=72.5+(0:5)\*5,xlim=xlim,ylim=ylim,

prob=TRUE,xlab="total length",main="A:Breaks at 72.5,77.5,...")

lines(dens,col=par('fg'),lty=2)

m<-mean(totlngth)

s<-sd(totlngth)

curve(dnorm(x,m,s),col="red",add=T)

hist(totlngth,breaks=75+(0:5)\*5,xlim=xlim,ylim=ylim,

prob=TRUE,xlab="total length",main="B:Breaks at 75,80,...")

lines(dens,col=par('fg'),lty=2)

curve(dnorm(x,m,s),col="red",add=T)

**#3）使用经验分布函数**

x<-sort(totlngth)

n<-length(x)

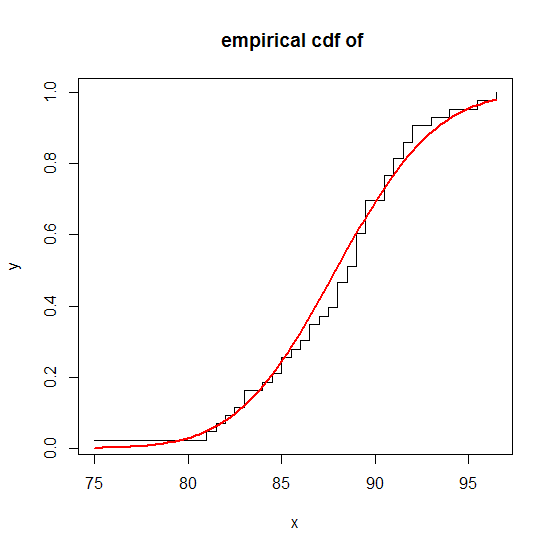
y<-(1:n)/n

m<-mean(totlngth)

s<-sd(totlngth)

plot(x,y,type="s",main="empirical cdf of ")

curve(pnorm(x,m,s),col="red",lwd=2,add=T)

****

**4.3.2单组数据的描述性统计量**

**1.总体描述**

**summary():计算单组数据的均值和五数（最小值、第一4分位数、中位数、第三4分位数、最大值）**

summary(totlngth)

Min. 1st Qu. Median Mean 3rd Qu. Max.

75.00 85.25 88.50 87.91 90.50 96.50

**2.五数及样本分位数概括**

> fivenum(totlngth) #五数

[1] 75.00 85.25 88.50 90.50 96.50

>

> quantile(totlngth) #常见几个分位数，即五数

0% 25% 50% 75% 100%

75.00 85.25 88.50 90.50 96.50

> quantile(totlngth,probs=c(0.25,0.5,0.75))

25% 50% 75%

85.25 88.50 90.50

>

> median(totlngth) #中位数

[1] 88.5

> max(totlngth)

[1] 96.5

> min(totlngth)

[1] 75

> range(totlngth)

[1] 75.0 96.5

**3.离差的概括**

> max(totlngth)-min(totlngth) #最大值-最小值，即全距

[1] 21.5

> range(totlngth)[2]-range(totlngth)[1]

[1] 21.5

>

> IQR(totlngth) #四分位极差=第三4分位数-第一4分位数

[1] 5.25

> quantile(totlngth,probs=0.75)-quantile(totlngth,probs=0.25)

75%

5.25

>

> sd(totlngth) #标准差

[1] 4.182241

> sd(totlngth)^2

[1] 17.49114

> var(totlngth) #方差

[1] 17.49114

>

> mad(totlngth) #绝对离差 ，R中的定义为1.4826\*median(abs(x-median(x)))

[1] 3.7065

**4.偏度与峰度**

**.**

install.packages("fBasics")

Warning: package ‘fBasics’ is in use and will not be installed

> library(fBasics)

> skewness(totlngth)

[1] -0.54838

attr(,"method")

[1] "moment"

> kurtosis(totlngth) #method="excess",为减去3后的值； method="moment",没减3

[1] 0.6170082

attr(,"method")

[1] "excess"

>

**4.4多组数据的描述性统计分析**

**4.4.1两组数据的图形概括**

**1）先考查散点图**

**例1：在R 的程序包DAAG 中有数据集cars, 希望估计其中速度(speed) 与终止距离(dist) 之间的关系。**

library(DAAG)

data(cars)

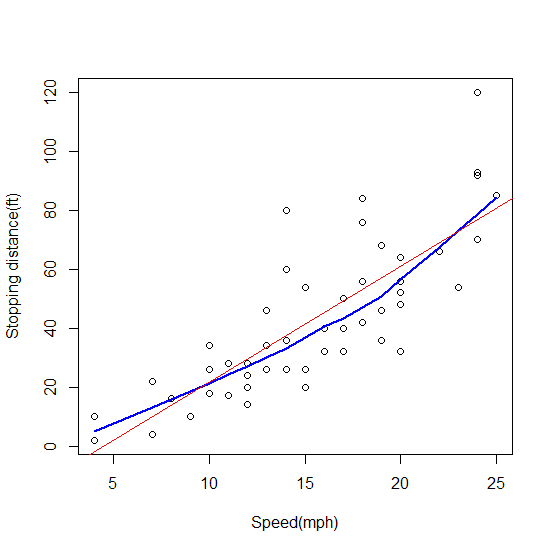
cars

str(cars)

plot(cars$dist~cars$speed,xlab="Speed(mph)",ylab="Stopping distance(ft)")

lines(lowess(cars$speed,cars$dist),lwd=2,col="blue") #注：speed为自变量，dist为因变量，lowess

abline(lm(cars$dist~cars$speed),col="red") #添加回归直线



#2)等高线图

#例2：在程序包chplot包中有数据集hdr，希望作出age与income的关系图

install.packages("chplot") #程序包安装不了

library(chplot)

data(hdr)

x<-hdr$age

y<-log(hdr$income)

plot(x,y)

library(MASS)

z<-kde2d(x,y)

contour(z,col="red",drawlabels=FALSE,main="density estimation:contour plot")

persp(z)

####换成以下数据(程序包DAAG中的cars数据集的speed和dist变量)

install.packages("DAAG")

library(DAAG)

data(cars)

x<-cars$speed

y<-cars$dist

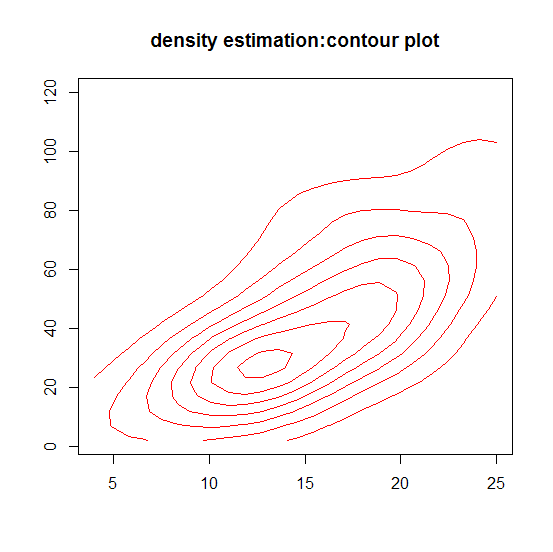
plot(x,y)

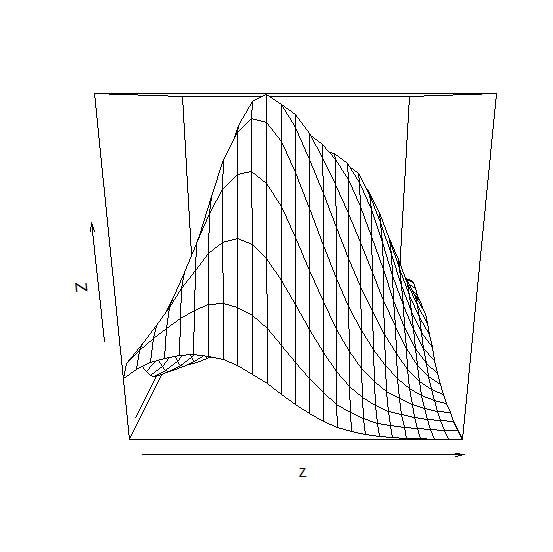
library(MASS)

z<-kde2d(cars$speed,cars$dist) #二维数据的核密度估计

contour(z,col="red",drawlabels=FALSE,main="density estimation:contour plot") #等高线

persp(z) #透视图





**4.4.2多组数据的图形描述**

* **函数pairs(x)\plot(x):产生矩阵x的第i列对第j列的散点图形成的图形矩阵**
* **函数matplot(x,y):产生矩阵x的第一列对矩阵y的第一列作散点图，x的第二列对y的第二列**作图，。。。如果y矩阵缺失，则y取向量1：n作图

例：产生五个样本容量为10的标准正态分布的样本，取绝对值后生成数据框d,并对这5个变量作散点图

n<-10

d<-data.frame(y1=abs(rnorm(n)),

y2=abs(rnorm(n)),

y3=abs(rnorm(n)),

y4=abs(rnorm(n)),

y5=abs(rnorm(n)))

d

plot(d) #或 pairs(d) #产生散点图矩阵

> d

y1 y2 y3 y4 y5

1 1.1843061 0.31180036 1.70797772 0.7683419 0.7355604

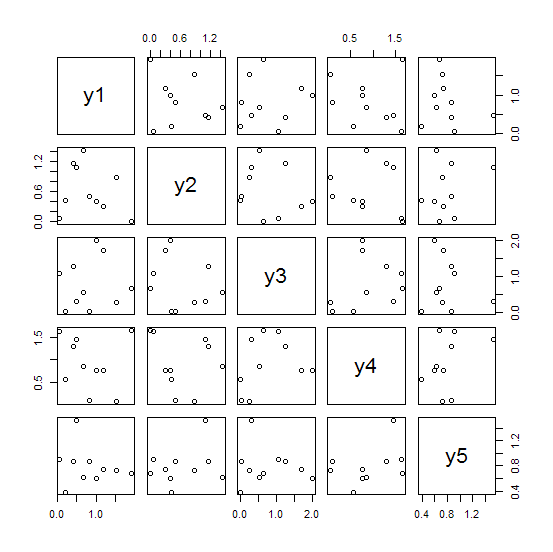
2 0.0614572 0.06046887 1.07726090 1.6314058 0.9059981

3 0.1977752 0.41683772 0.01055951 0.5746956 0.3833550

4 0.9989972 0.40163900 2.01035388 0.7615965 0.5917116

5 1.5267430 0.87876436 0.26065776 0.0663792 0.7198095

6 0.6716882 1.42289764 0.54922012 0.8594739 0.6189202

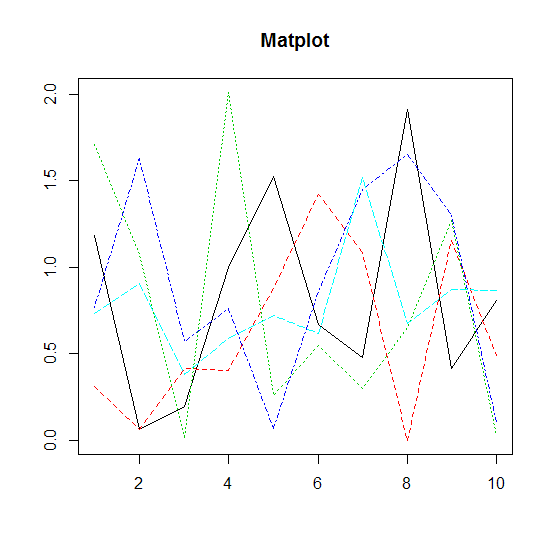
7 0.4782820 1.08512186 0.30280483 1.4507207 1.5226178

8 1.9104904 0.00154064 0.65447202 1.6550163 0.6734010

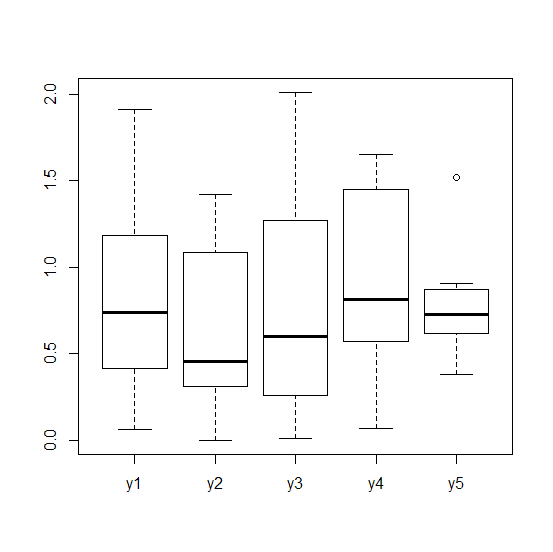
9 0.4133912 1.15669560 1.27167787 1.3023653 0.8696345

10 0.8110886 0.49246162 0.02797226 0.1042640 0.8670649

matplot(d,type="l",ylab="",main="Matplot") #各个散点图放在同一个作图区域



boxplot(d) #对数据框的各列作盒子图，并放在同一作图区域



**4.4.3多组数据的描述性统计量**

**1。多组数据的概括**

* **summary():可以得到多组数据中各个变量的均值和五数**

**例：程序包datasets中数据框state.x77 描述了美国50个州的人口数、人均收入、人均寿命、一年中有雾的天数等情况. 数据如下:**

state.x77 #自带数据集

state.region #自带数据集

summary(state.x77) #各个变量汇总

Population Income Illiteracy

Min. : 365 Min. :3098 Min. :0.500

1st Qu.: 1080 1st Qu.:3993 1st Qu.:0.625

Median : 2838 Median :4519 Median :0.950

Mean : 4246 Mean :4436 Mean :1.170

3rd Qu.: 4968 3rd Qu.:4814 3rd Qu.:1.575

Max. :21198 Max. :6315 Max. :2.800

Life Exp Murder HS Grad

Min. :67.96 Min. : 1.400 Min. :37.80

1st Qu.:70.12 1st Qu.: 4.350 1st Qu.:48.05

Median :70.67 Median : 6.850 Median :53.25

Mean :70.88 Mean : 7.378 Mean :53.11

3rd Qu.:71.89 3rd Qu.:10.675 3rd Qu.:59.15

Max. :73.60 Max. :15.100 Max. :67.30

Frost Area

Min. : 0.00 Min. : 1049

1st Qu.: 66.25 1st Qu.: 36985

Median :114.50 Median : 54277

Mean :104.46 Mean : 70736

3rd Qu.:139.75 3rd Qu.: 81163

Max. :188.00 Max. :566432

* aggregate **():可以得到多组数据按指定的分组变量，计算各个变量的统计函数**

x77,list(Region=state.region),mean) # 各个变量按地区（州）分组计算均值

Region Population Income Illiteracy

1 Northeast 5495.111 4570.222 1.000000

2 South 4208.125 4011.938 1.737500

3 North Central 4803.000 4611.083 0.700000

4 West 2915.308 4702.615 1.023077

Life Exp Murder HS Grad Frost Area

1 71.26444 4.722222 53.96667 132.7778 18141.00

2 69.70625 10.581250 44.34375 64.6250 54605.12

3 71.76667 5.275000 54.51667 138.8333 62652.00

4 71.23462 7.215385 62.00000 102.1538 134463.00

> aggregate(state.x77,list(Region=state.region,Cold=state.x77[,"Frost"]>130),mean) #各个变量按州和有雾天数是否大于130交叉分组计算均值

Region Cold Population Income Illiteracy

1 Northeast FALSE 8802.8000 4780.400 1.1800000

2 South FALSE 4208.1250 4011.938 1.7375000

3 North Central FALSE 7233.8333 4633.333 0.7833333

4 West FALSE 4582.5714 4550.143 1.2571429

5 Northeast TRUE 1360.5000 4307.500 0.7750000

6 North Central TRUE 2372.1667 4588.833 0.6166667

7 West TRUE 970.1667 4880.500 0.7500000

Life Exp Murder HS Grad Frost Area

1 71.12800 5.580000 52.06000 110.6000 21838.60

2 69.70625 10.581250 44.34375 64.6250 54605.12

3 70.95667 8.283333 53.36667 120.0000 56736.50

4 71.70000 6.828571 60.11429 51.0000 91863.71

5 71.43500 3.650000 56.35000 160.5000 13519.00

6 72.57667 2.266667 55.66667 157.6667 68567.50

7 70.69167 7.666667 64.20000 161.8333 184162.17

**2。标准差与协方差矩阵的计算**

> sd(state.x77) #有点问题，尝试： apply(state.x77,2,sd)

[1] 37801.78

> var(state.x77)

Population Income Illiteracy

Population 19931683.7588 571229.7796 292.8679592

Income 571229.7796 377573.3061 -163.7020408

Illiteracy 292.8680 -163.7020 0.3715306

Life Exp -407.8425 280.6632 -0.4815122

Murder 5663.5237 -521.8943 1.5817755

HS Grad -3551.5096 3076.7690 -3.2354694

Frost -77081.9727 7227.6041 -21.2900000

Area 8587916.9494 19049013.7510 4018.3371429

Life Exp Murder HS Grad

Population -4.078425e+02 5663.523714 -3551.509551

Income 2.806632e+02 -521.894286 3076.768980

Illiteracy -4.815122e-01 1.581776 -3.235469

Life Exp 1.802020e+00 -3.869480 6.312685

Murder -3.869480e+00 13.627465 -14.549616

HS Grad 6.312685e+00 -14.549616 65.237894

Frost 1.828678e+01 -103.406000 153.992163

Area -1.229410e+04 71940.429959 229873.192816

Frost Area

Population -77081.97265 8.587917e+06

Income 7227.60408 1.904901e+07

Illiteracy -21.29000 4.018337e+03

Life Exp 18.28678 -1.229410e+04

Murder -103.40600 7.194043e+04

HS Grad 153.99216 2.298732e+05

Frost 2702.00857 2.627039e+05

Area 262703.89306 7.280748e+09

> aggregate(state.x77,list(Region=state.region),sd)

Region Population Income Illiteracy

1 Northeast 6079.565 559.0771 0.2783882

2 South 2779.508 605.4505 0.5524189

3 North Central 3702.828 283.0825 0.1414214

4 West 5578.607 663.9004 0.6084870

Life Exp Murder HS Grad Frost Area

1 0.7438769 2.670570 3.928104 30.89408 18075.69

2 1.0221994 2.627602 5.741773 31.30682 57965.27

3 1.0367285 3.569345 3.622865 23.89307 14967.02

4 1.3519715 2.675770 3.500000 68.87652 134981.73

#3.相关系数的计算

>

> x<-c(44.4,45.9,46.0,46.5,46.7,47,48.7,49.2,60.1)

> y<-c(2.6,10.1,11.5,30.0,32.6,50,55.2,85.8,86.8)

> cor(x,y) #默认为pearson，线性相关系数

[1] 0.768587

> cor(x,y,method="spearman")

[1] 1

> cor(x,y,method="kendall")

[1] 1

**4.4.4分组数据的图形概括**

分组数据可视为特殊的多组数据, 他们的区别是: 在多组数据中各数值型变量的观测值指向不同的对象，而分组数据是指同一个数值型变量的观测值按另一个分类变量分成若干个子集, 因此, 这些子集指向同一个变量。

**例4.4.5** 杜鹃把蛋下在其它种类鸟的鸟巢中, 这些鸟会帮它们孵化, 我们希望了解在不同类的鸟巢中杜鹃蛋的长度

install.packages("DAAG")

library(DAAG)

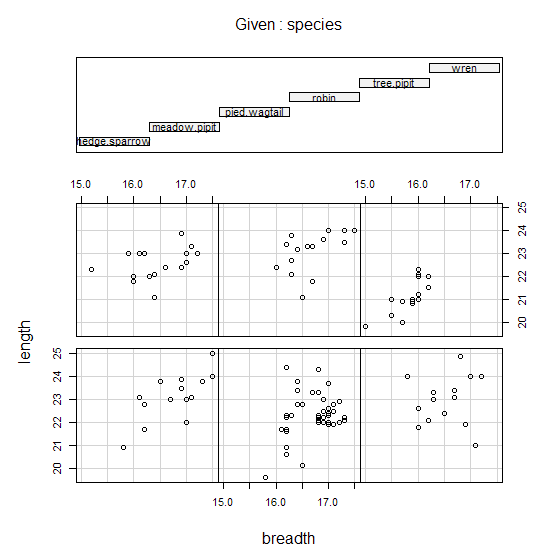
data(cuckoos)

cuckoos

**1.使用条件散点图**

* **函数coplot(y~x|a):对于一个因子变量，绘制y对x的散点图**
* **函数coplot(y~x|a\*b):对于两个因子变量，绘制y对x的散点图**

coplot(length~breadth | species,data=cuckoos ) #书上有误



注：排列顺序为先下层1，2，3，

再上层为4，5，6

**2.使用直方图**

data(cuckoos)

attach(cuckoos)

length.mp <- length[species=="meadow.pipit"]

length.tp <- length[species=="tree.pipit"]

length.hs <- length[species=="hedge.sparrow"]

length.r <- length[species=="robin"]

length.pw <- length[species=="pied.wagtail"]

length.w <- length[species=="wren"]

par(mfrow=c(3,2))

hist(length.mp,breaks=6,probability=T,

xlim=c(19,25),ylim=c(0,1),main="",col=6)

hist(length.tp,breaks=6,probability=T,

xlim=c(19,25),ylim=c(0,1),main="",col=6)

hist(length.hs,breaks=6,probability=T,

xlim=c(19,25),ylim=c(0,1),main="",col=6)

hist(length.r,breaks=6,probability=T,

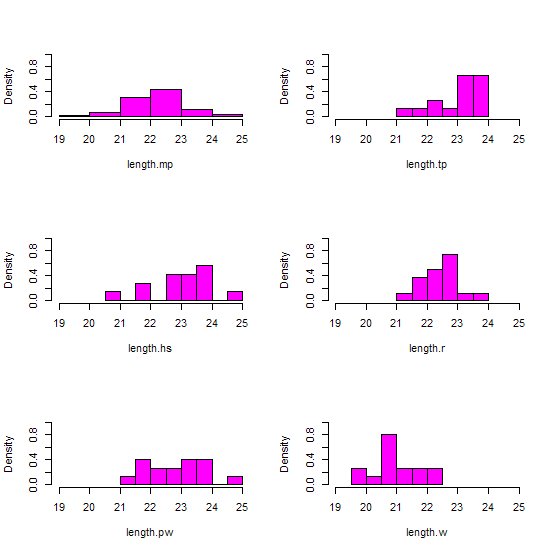
xlim=c(19,25),ylim=c(0,1),main="",col=6)

hist(length.pw,breaks=6,probability=T,

xlim=c(19,25),ylim=c(0,1),main="",col=6)

hist(length.w,breaks=6,probability=T,

xlim=c(19,25),ylim=c(0,1),main="",col=6)



使用自定义函数hists()作直方组图：

hists<-function(x,y){

y<-factor(y)

n<-length(levels(y))

op<-par(mfcol=c(n,1),mar=c(2,4,1,1))

b<-hist(x,plot=F)$breaks #共同区间端点

print(b)

for (i in levels(y)){

hist(x[y==i],breaks=b,prob=T,ylim=c(0,1.0),

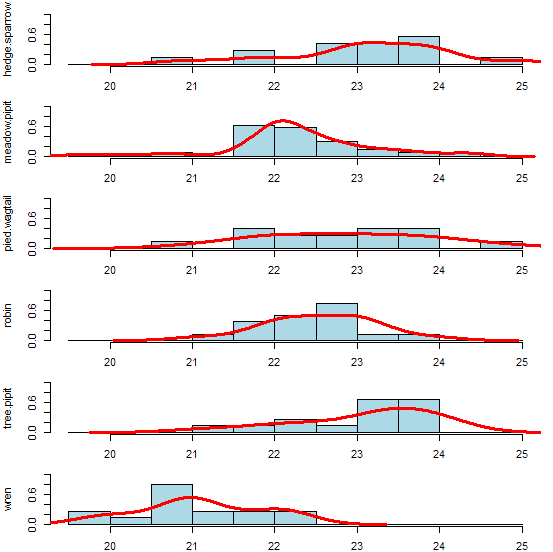
main="",ylab=i,col="lightblue",xlab="")

points(density(x[y==i]),type='l',lwd=3,col="red")

}

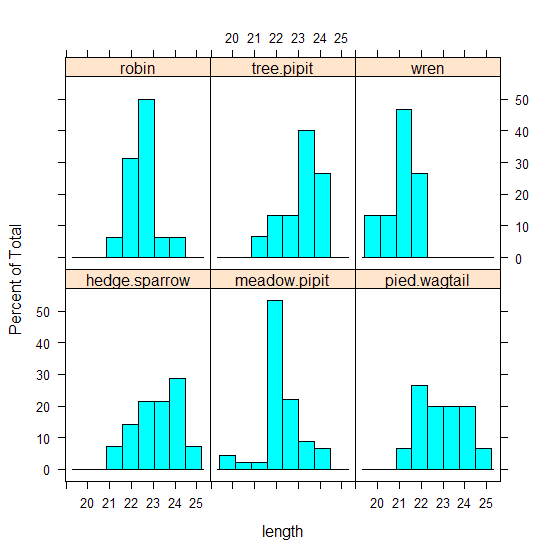
}

hists(cuckoos$length,cuckoos$species)



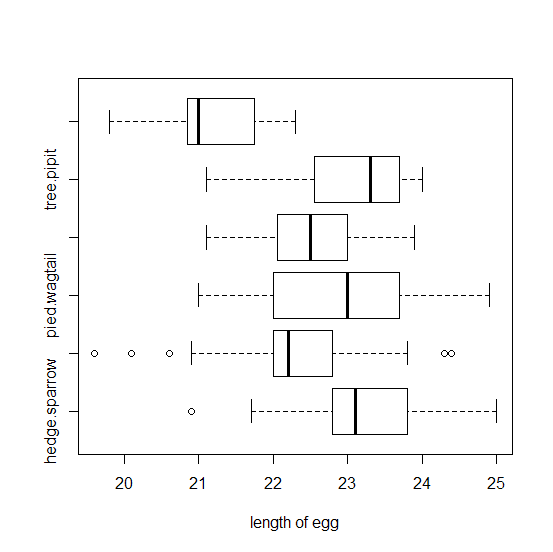
lattice包中的直方图函数histogram（）###不作要求

histogram(~length|species,data=cuckoos)



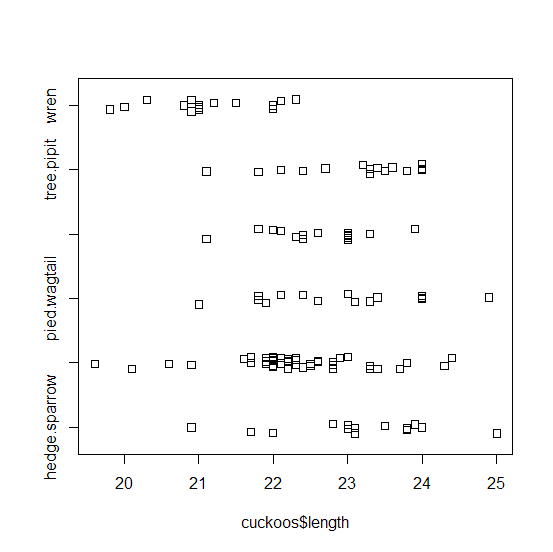
**3.使用框须图**

boxplot(length~species,data=cuckoos,xlab="length of egg",horizontal=TRUE)



**4.使用条形图**

stripchart(cuckoos$length~cuckoos$species,method="jitter") #尝试method="overplot","stack"



**4.5分类数据的描述性统计分析**

如果数据集中对应的变量都是定性变量, 这样的数据称为分类数据. 这种数据常使用表格来描述, 并为进一步的统计分析服务. 我们主要考虑由二元定性数据所构成的二维列联表数据. 这一节主要描述如何制作列联表和图形描述,

**4.5.1列联表的制作**

1. **由分类(频数)数据构造列联表**

* 函数matrix():矩阵构造残联表

例4.5.1 为考查眼睛的颜色(Eye) 与头发的颜色(Hair) 之间的关系

x<-c(68,20,15,5,119,84,54,29,26,17,14,14,7,94,10,16)

Eye.Hair<-matrix(x,nrow=4,byrow=T)

colnames(Eye.Hair)<-c("Brown","Blue","Hazel","Green")

rownames(Eye.Hair)<-c("Black","Brown","Red","Blond")

Eye.Hair

Brown Blue Hazel Green

Black 68 20 15 5

Brown 119 84 54 29

Red 26 17 14 14

Blond 7 94 10 16

1. 由原始数据构造列联表

* 函数table（factor1,factor2）：制作列联表
* 函数xtabs(~factor1+factor2)：制作列联表

例4.5.2 数据包ISwR中的数据集juul中含有三个分类变量:sex, tanner, menarche. 则我们可以得到下面的一些列表。

install.packages("ISwR")

library(ISwR)

data(juul)

juul

attach(juul)

table(sex) #等价于 xtabs(~sex)

table(sex,menarche) #等价于 xtabs(~menarche+tanner)

table(menarche,tanner,sex) #等价于 xtabs(~menarche+tanner+sex)

输出结果：

（1）sex

1 2

621 713

（2）

menarche

sex 1 2

1 0 0

2 369 335

（3）

, , sex = 1

tanner

menarche 1 2 3 4 5

1 0 0 0 0 0

2 0 0 0 0 0

, , sex = 2

tanner

menarche 1 2 3 4 5

1 221 43 32 14 2

2 1 1 5 26 202

注意到原始数据有1339行，而交叉残联表中总和只有704，为什么呢？不统计缺失值的原因。

#补充例子

x<-factor(c(1,1,2,2,2,2,NA))

y<-factor(c("a","b","b","b","b","b","b"))

z=data.frame(xx=x,yy=y);z

attach(z)

table(xx,yy) #尝试： table(z$xx,z$yy)

table(xx,yy,useNA="always") #尝试：useNA="ifany"

1. 获得边际列表

* margin.table():按列联表中某个属性(因子) 求和, 称之为边际列表.

> margin.table(Eye.Hair,1) #对行求和

Black Brown Red Blond

108 286 71 127

> margin.table(Eye.Hair,2) #对列求和

Brown Blue Hazel Green

220 215 93 64

> apply(Eye.Hair,1,sum) #对行求和

Black Brown Red Blond

108 286 71 127

> apply(Eye.Hair,2,sum) #对列求和

Brown Blue Hazel Green

220 215 93 64

1. 频率残联表

* #函数prop.table（）：获得频率残联表

> prop.table(Eye.Hair,margin=1) #注：1为行百分比加起来为100%，尝试margin=2

Brown Blue Hazel Green

Black 0.62962963 0.1851852 0.13888889 0.0462963

Brown 0.41608392 0.2937063 0.18881119 0.1013986

Red 0.36619718 0.2394366 0.19718310 0.1971831

Blond 0.05511811 0.7401575 0.07874016 0.1259843

> round(prop.table(Eye.Hair,margin=1),digits=2) #round(x, n)，求 x的精确到小数点后的n位的约数 ,默认n=0

Brown Blue Hazel Green

Black 0.63 0.19 0.14 0.05

Brown 0.42 0.29 0.19 0.10

Red 0.37 0.24 0.20 0.20

Blond 0.06 0.74 0.08 0.13

>

> round(Eye.Hair/sum(Eye.Hair),digits=2) #全局相对频率残联表

Brown Blue Hazel Green

Black 0.11 0.03 0.03 0.01

Brown 0.20 0.14 0.09 0.05

Red 0.04 0.03 0.02 0.02

Blond 0.01 0.16 0.02 0.03

4.5.2残联表的图形描述

* 条图barplot(height, ...):以向量height的值或矩阵height的列求和作为条图的矩形高度作条图

data(HairEyeColor)

?HairEyeColor #A 3-dimensional array

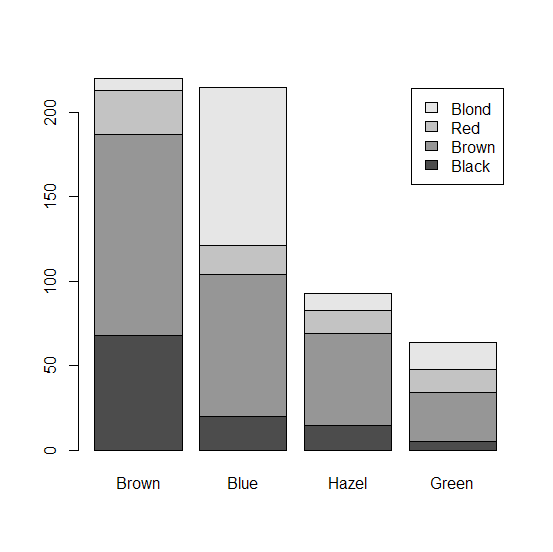
b<-apply(HairEyeColor,c(1,2),sum) #注margin=c(1,2)表示对同行同列元素运算,此处返回的是矩阵

a<-as.table(b)

a

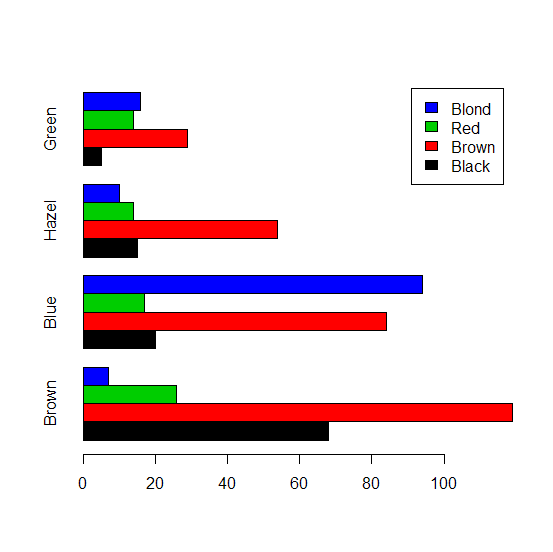
barplot(b,legend.text=attr(a,"dimnames")$Hair)

class(b);class(a)



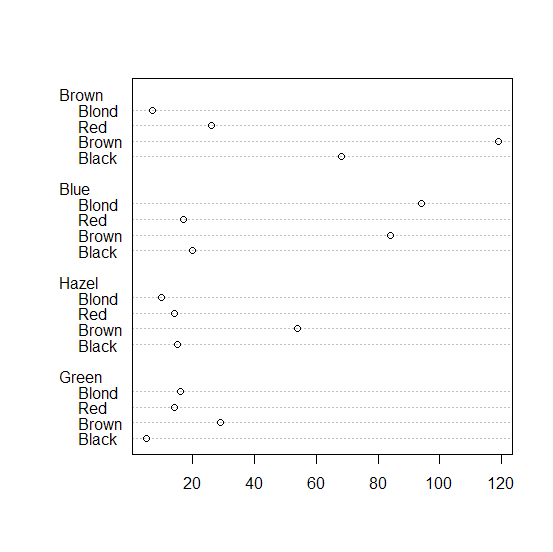
尝试：彩色的水平并列条图

barplot(b,legend.text=attr(a,"dimnames")$Hair,beside=TRUE,col=1:4,horiz=T)



使用点图

dotchart(Eye.Hair)



本节学习的R函数小结

|  |  |  |
| --- | --- | --- |
| 第4章 探索性数据分析的主要函数 | | |
| 函数 | 功能 | 示例 |
|  |  |  |
|  |  |  |

**作业：练习本节所有命令**

* **4.3 4.5 4.7 4.8**