# Ch2.6R的图形功能

本章概要

*♢* **高级绘图函数**

*♢* **低级绘图函数**

*♢* **绘图参数**

*♢* **一个实例**

**两组演示例子：**

* demo(graphics) # 二维图形示例
* demo(persp) # 三维图形示例

2.6.1**高级绘图函数（教材P51页）**

**重点掌握：**

**plot(x):** 以*x* 的元素值为纵坐标、以序号为横坐标绘图

**plot(x,y):**绘制以x为横轴，y为纵轴的二元图

**pie(x):** 饼图

**boxplot(x):** 盒形图(\box-and-whiskers")

**hist(x): x**的频率直方图

**barplot(x) :x** 的值的条形图

**绘图函数的部分共同选项及缺省值：**

***add=FALSE ：****如果是TRUE，叠加图形到前一个图上(如果有的话)——错误：没有此选项*

**axes=TRUE：** 如果是FALSE，不绘制轴与边框

**type="p"：** 指定图形的类型，"p": 点，"l": 线，"b": 点连线，"o":同上，但是线在点上，"h": 垂直线，"s": 阶梯式，垂直线顶端显示数据，"S": 同上，但是在垂直线底端显示数据

**xlim=, ylim=：** 指定轴的上下限， 例如xlim=c(1, 10)或者xlim=range(x)

**xlab=, ylab= ：**坐标轴的标签，必须是字符型值

**main=：** 主标题，必须是字符型值

**sub= ：**副标题(用小字体)

2.6.2**低级绘图函数（教材P52页）**

低级作图函数作用于现存的图形上，重点掌握：

**points(x, y)：** 添加点(可以使用选项type=)

**lines(x, y)：** 同上，但是添加线

**abline(a,b)：** 绘制斜率为b和截距为a的直线

**abline(h=y) ：**在纵坐标*y* 处画水平线

**abline(v=x)：** 在横坐标*x* 处画垂直线

**abline(lm.obj)：** 画由lm.obj确定的回归线

**legend(x, y, legend)：** 在点(*x*,*y*) 处添加图例，说明内容由legend给定

**title( )：** 添加标题，也可添加一个副标题

2.6.3**绘图参数（教材P54页）**

除了低级作图命令之外，图形的显示也可以用绘图参数来改良.绘图参数可以作为图形函数的选项(但不是所有参数都可以这样用。

**函数par():可以显示或设定当前绘图参数，如**

**> par( ) #显示绘图参数缺省值**

**> par(bg="yellow") #将导致后来的图形都以黄色的背景来绘制.**

有73 个绘图参数，其中一些有非常相似的功能. 这些参数详细的列表可以通过help(par)获得.

**主要掌握：**

**bg：** 指定背景色(例如bg="red", bg="blue"; 用colors( )可以显示657 种可用的颜色名)

**cex ：**控制缺省状态下符号和文字大小的值；另外，cex.axis控制坐标轴刻度数字大小, cex.lab控制坐标轴标签文字大小, cex.main控制标题文字大小, cex.sub控制副标题文字大小

**col** ：控制符号的颜色；和cex类似，还可用：col.axis, col.lab, col.main, col.sub

**lty：** 控制连线的线型，可以是整数(1: 实线，2: 虚线，3: 点线，4: 点虚线，5: 长虚线，6: 双虚线），或者是不超过8 个字符的字符串（字符为从"0" 到"9"之间的数字) 交替地指定线和空白的长度，单位为磅(points) 或象素，例如lty="44"和lty=2效果相同

**lwd** ：控制连线宽度的数字

**mar** ：控制图形边空的有4 个值的向量c(bottom, left, top, right), 缺省值为c(5.1, 4.1,

4.1, 2.1)

**mfcol：** c(nr,nc)的向量，分割绘图窗口为nr行nc列的矩阵布局，按列次序使用各子窗口

**mfrow** 同上，但是按行次序使用各子窗口

**pch:** 控制符号的类型，可以是1 到25 的整数，也可以是""里的单个字符(见图2.2)

2.6.4一个实例

以R自带的数据集Puromycin为例，说明基本绘图函数的使用.

> dim(Puromycin)

[1] 23 3

> names(Puromycin) #返回数据集的变量名

[1] "conc" "rate" "state"

> str(Puromycin) #返回数据集的变量取值信息

'data.frame': 23 obs. of 3 variables:

$ conc : num 0.02 0.02 0.06 0.06 0.11 0.11 0.22 0.22 0.56 0.56 ...

$ rate : num 76 47 97 107 123 139 159 152 191 201 ...

$ state: Factor w/ 2 levels "treated","untreated": 1 1 1 1 1 1 1 1 1 1 ...

- attr(\*, "reference")= chr "A1.3, p. 269"

> head(Puromycin)

conc rate state

1 0.02 76 treated

2 0.02 47 treated

3 0.06 97 treated

4 0.06 107 treated

5 0.11 123 treated

6 0.11 139 treated

>

1. **简单的散点图**

* **函数plot（x,y）**:绘制以x为横轴，y为纵轴的二元散点图

例：取数据集中状态为treated的观测为子集，作出rate与conc的散点图

> PuroA<-subset(Puromycin,state=="treated")

> PuroA

conc rate state

1 0.02 76 treated

2 0.02 47 treated

3 0.06 97 treated

4 0.06 107 treated

5 0.11 123 treated

6 0.11 139 treated

7 0.22 159 treated

8 0.22 152 treated

9 0.56 191 treated

10 0.56 201 treated

11 1.10 207 treated

12 1.10 200 treated

#方式一：使用$ 直接指向数据集与变量

plot(x=PuroA$conc, y=PuroA$rate)

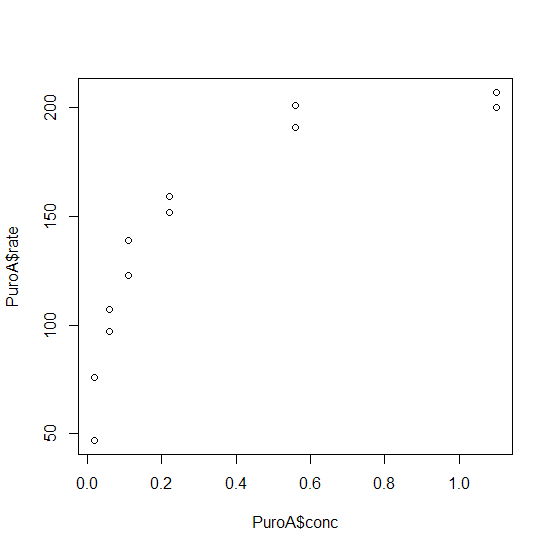
plot(PuroA$conc, PuroA$rate) #尝试：plot(PuroA$rate),选项type="l","b","o"

#方式二： plot( )函数中使用data选项;

plot(rate~conc,data=PuroA) #注：横、纵坐标轴的变化，“~”实际是告诉R系统变量间有因果关系 #尝试：“~”改为“，”行吗？

#方式三： 在with( )中使用plot( );

with(PuroA, plot(conc,rate))



1. **美化图形**
2. 修改绘图的符号、颜色、尺寸

* 选项“pch=”指定绘图的符号， 可取1~25的数值或单个字符。默认的是空心圆

u<-1:25

plot(u)

plot(u,pch=16) #尝试pch=1,2,3,"a"或“\*”

plot(u,pch=u) #注：pch=如果取向量，则向量的长度必须与变量相同，且向量元素的值要在1~25范围内

plot(u,pch=c(1,16)) #尝试：前面10个点是空心圆，后面15个点是实心圆,使用rep(1,10)

* 选项“col=”指定绘图符号的颜色，可用1~8的整数或英文颜色名（如“red”),默认黑色。

plot(u,pch=16,col="red") #或者col=2 ,注：函数colors()列出所有颜色

plot(u,pch=16,col=u)

plot(u,pch=16,col=c("blue","green")) #循环法则，col=如果取向量,用法要求同pch=

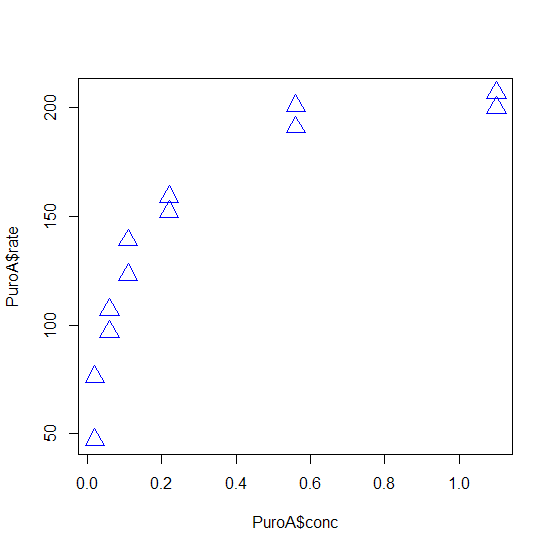
plot(u,pch=16,col=rainbow(4)) #彩虹调色盘中的4 种颜色

* 选项“cex=”指定绘图符号的尺寸，默认是1，如果cex=2,表示尺寸取缺省尺寸的2倍

plot(u,cex=2)

#选用蓝色(选项为col=4或col="blue") 小三角形(选项为pch=2或pch="T") 得到的, 大小为cex=2.5倍缺省值,

plot(x=PuroA$conc, y=PuroA$rate ,pch=2,col=4,cex=2)

****

1. 设定坐标轴与标题

plot(x=PuroA$conc, y=PuroA$rate,

pch=2,col=4,cex=1,

xlim=c(0,1.2), ylim=c(40,210),

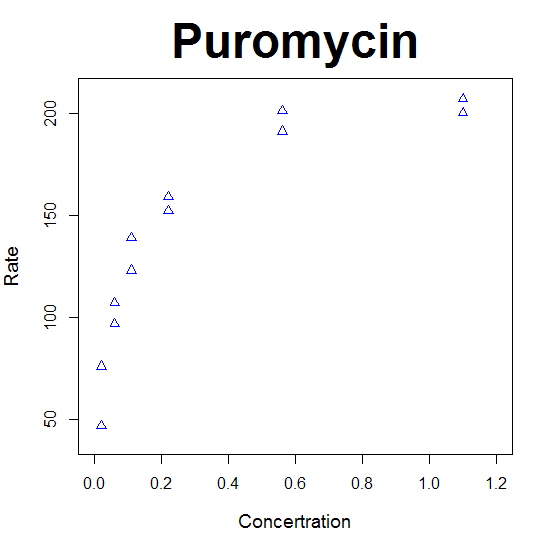
ylab="Rate",xlab="Concertration",

cex.lab=1.2)

title(main="Puromycin",cex.main=3)

内容如下：

* 限定*X* 轴范围为0 到1.2, *Y* 轴范围为40 到210
* *X* 轴标为“Concentration”, Y 轴标为“Rate”
* 规定坐标轴标签大小(cex.lab=1.2)
* 增加主标题

****

1. **主图添线line()**

* 1）在数据集PuroA的rate与conc的散点图上，每一浓度处的平均反应速度作点，并连接这些点

#使用doby包的summaryBy()计算每一浓度处的平均值

install.packages("doBy")

library(doBy)

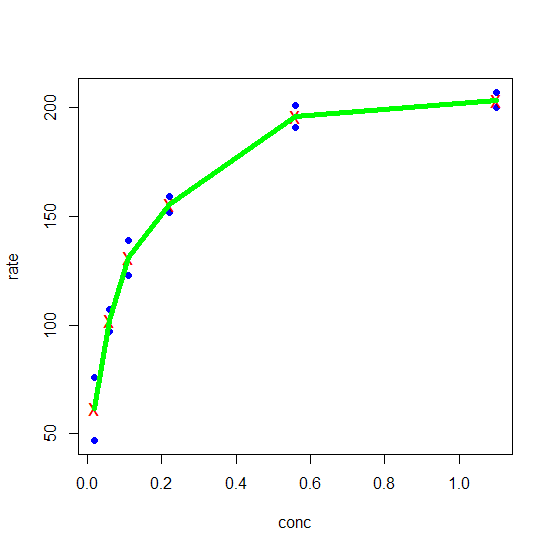
PuroA.mean<-summaryBy(rate~conc,data=PuroA,FUN=mean)

PuroA.mean

plot(rate~conc,data=PuroA,pch=16,col=4,cex=1)

points(rate.mean~conc,data=PuroA.mean,col="red",pch="x",cex=1.5)

lines(rate.mean~conc,data=PuroA.mean,col="green",lwd=5) #lwd=，设置线宽



* 2)添加局部多项式拟合线（略）

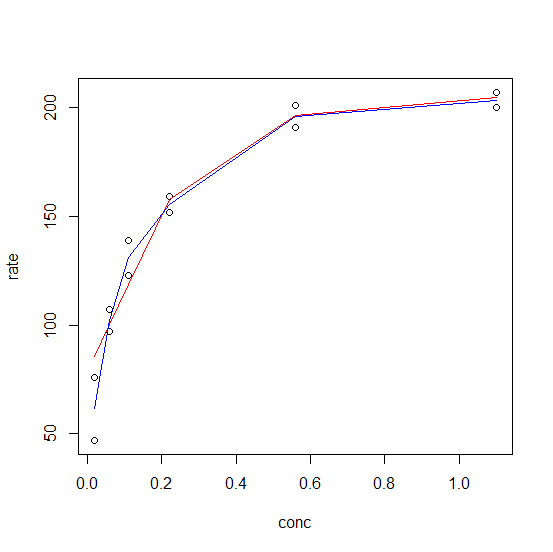
plot(rate~conc,data=PuroA)

smooth1<-with(PuroA,lowess(rate~conc,f=0.9))

smooth2<-with(PuroA,lowess(rate~conc,f=0.1))

lines(smooth1,col="red")

lines(smooth2,col="blue")"



**3）**添加一次、二次、三次多项式拟合线line()

plot(rate~conc,data=PuroA)

m1<-lm(rate~conc,data=PuroA)

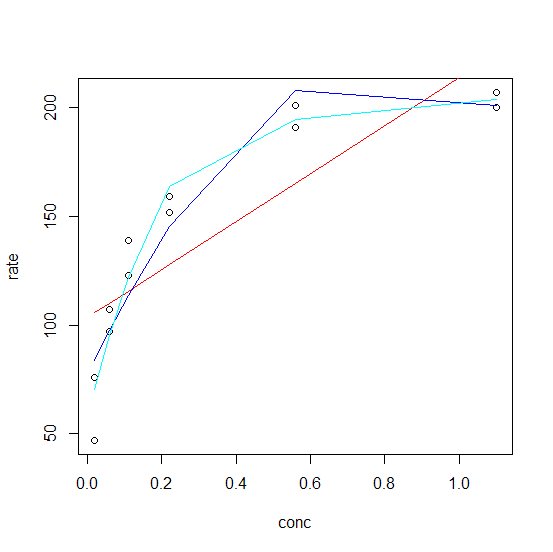
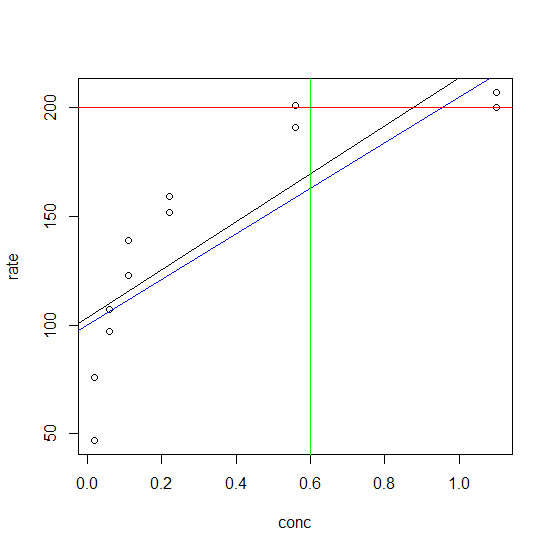
m2<-lm(rate~conc+I(conc^2),data=PuroA)

m3<-lm(rate~conc+I(conc^2)+I(conc^3),data=PuroA)

lines(fitted(m1)~conc,data=PuroA,col="red")

lines(fitted(m2)~conc,data=PuroA,col="blue")

lines(fitted(m3)~conc,data=PuroA,col="cyan")

****

4）添加参考线:abline()

plot(rate~conc,data=PuroA)

abline(lm(rate~conc,data=PuroA))

abline(a=100,b=105,col="blue")

abline(h=200,col="red")

abline(v=0.6,col="green")

1. 两个散点图的叠加.

* 例：将 Puromycin中变量 rate 与 conc之间的关系按 state 的两个值分别画出散点（在一幅图中）. 对于“treated”使用符号 1 和颜色 1, 对于“untreated”使用符号 2 和颜色 2

**#自己的想法:使用plot()两次，并用**add=TRUE,失败！

> PuroA<-subset(Puromycin,state=="treated")

> PuroB<-subset(Puromycin,state=="untreated")

> plot(rate~conc,data=PuroA,col=1,pch=1)

> plot(rate~conc,data=PuroB,col=2,pch=1,add=TRUE) #plot()没有add=选项

Warning messages:

1: In plot.window(...) : "add" is not a graphical parameter

2: In plot.xy(xy, type, ...) : "add" is not a graphical parameter

3: In axis(side = side, at = at, labels = labels, ...) :

"add" is not a graphical parameter

4: In axis(side = side, at = at, labels = labels, ...) :

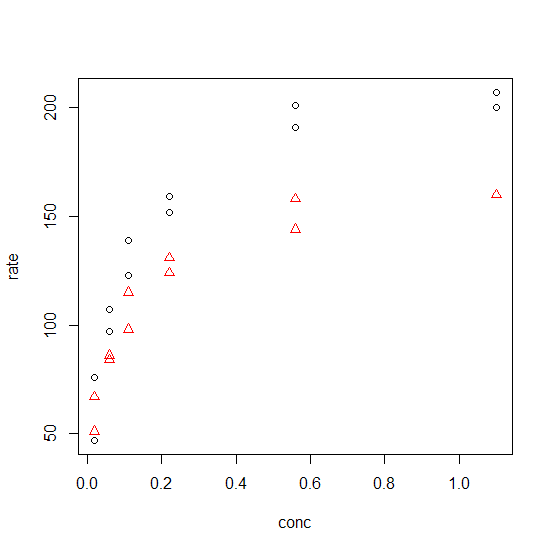
"add" is not a graphical parameter

5: In box(...) : "add" is not a graphical parameter

6: In title(...) : "add" is not a graphical parameter

>

**书上的程序思路：两幅图的数据放在一个数据集中，设置绘图符号和颜色为等长度的数值型向量**



> #尝试因子的新用法：

> F=factor(c("a","b","b","b"));F

[1] a b b b

Levels: a b

> c(1,2)[F] # #思考：结果与因子的区别？

[1] 1 2 2 2

>

> mysymb<-c(1,2)[Puromycin$state]

> plot(rate~conc,data=Puromycin,col=mysymb,pch=mysymb)

思考：还有其它方式可以将state的状态由treated和untreated转化为1，2表示吗？

Puromycin$bj[Puromycin$state=="treated"]<-1 #新增变量bj

Puromycin

Puromycin$bj[Puromycin$state=="untreated"]<-2

plot(rate~conc,data=Puromycin,col=bj,pch=bj)

* 再对每一 state 在散点图上添加局部多项式光滑线（略）

# install.packages("locfit")

library(locfit)

PuroA<-subset(Puromycin,state=="treated")

PuroB<-subset(Puromycin,state=="untreated")

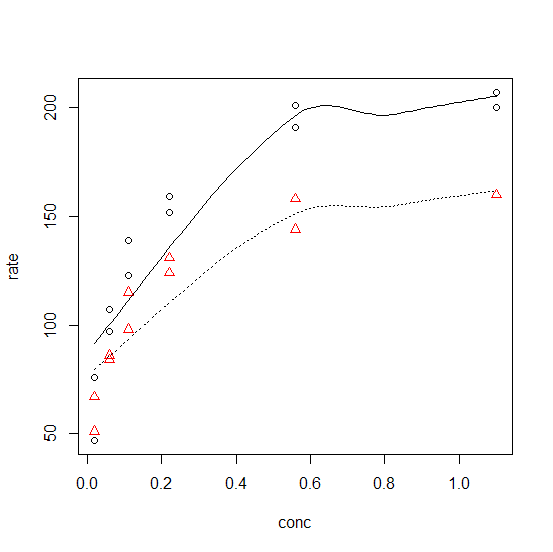
smoothA<-locfit(rate~lp(conc,nn=1,deg=1),data=PuroA)

smoothB<-locfit(rate~lp(conc,nn=1,deg=1),data=PuroB)

plot(rate~conc,data=Puromycin,col=mysymb,pch=mysymb)

lines(smoothA,lty=1) #选项lty=，设定线型

lines(smoothB,lty=3)



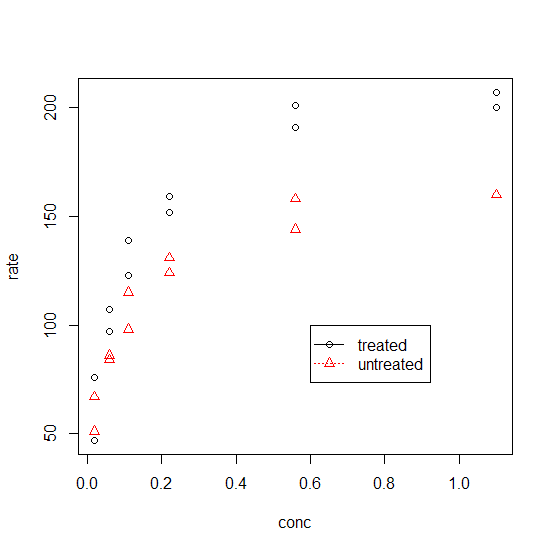
* 添 加 图 例 ：函数legend（）

plot(rate~conc,data=Puromycin,col=c(1,2)[state],pch=c(1,2)[state])

legend(x=0.6,y=100,legend=c("treated","untreated"),col=c(1,2),pch=c(1,2),lty=c(1,3))

LL=locator(1) #获取合适的位置放置图例，鼠标在图上点一下

legend(LL,legend=c("treated","untreated"),col=c(1,2),pch=c(1,2),lty=c(1,3))



5.作并列图

1) par()函数:可以用来设置或者获取图形参数

* 若要设置图形参数，则可用par(tag = value)的形式，其中tag的详细说明参见下面的列，value就是参数值,如

par()

?par

par(pch=16,col="red",cex=1.5) #设置绘图符号为16号实心圆，颜色为红色，大小为默认的1.5倍

par(mar = c(4, 4, 1, 0.5), bg = "yellow") # 设置边距参数和背景色

par(lty=2,lwd=2,cex=1.5) #线条为虚线，线宽为默认的2倍宽

par(cex.axis=0.75,font.axis=3) #坐标轴文字缩放为原来的75%，斜体

x=1:10;y=x^2

plot(x,y) #采用当前参数值

plot(x,y,lty=1,col="blue") #函数中设定参数选项

* 注：par()中这些参数只能设置一个单值，而绘图函数中可以对它们设置一个向量，这个向量的值将依次运用到各个元素上，若向量长度短于元素个数，那么向量会被循环使用，直到所有的元素都被画出来，事实上，向量的循环使用也是R图形参数的一大特点

points(x,y,col=1:8,pch=c(1,2))

* **par(mfrow = c(m, n)), 它表示将当前的窗口平均分割为 m 行n列 个窗口**

例如, 要在同一个窗口中作出 state的两个值对应的两个散点图 (一个窗口中作两幅并列图)

windowsr(width=7,height=3.5) #设定窗口大小

par(mfrow=c(1,2))

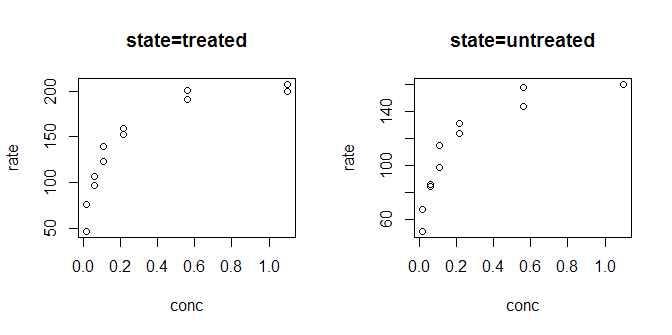
plot(rate~conc,data=PuroA)

title("state=treated")

plot(rate~conc,data=PuroB)

title("state=untreated")

dev.off() #关闭图形窗口



2）函数layout():用矩阵设置窗口的划分

* layout(mat,widths,heights)函数的一般形式为layout(mat)，mat为一矩阵，mat元素的数量决定了一个output device被等分成几份相同元素为一块。矩阵的0元素表示该位置不画图，非0元素必须包括从1开始的连续的整数值，比如：1……N，按非0元素的大小设置图形的顺序。widths用来设置窗口不同列的宽度，heights设置不同行的高度。
* 补充例子

attach(mtcars)

M<-matrix(c(1,1,2,3),2,2,byrow = TRUE)

layout(M)

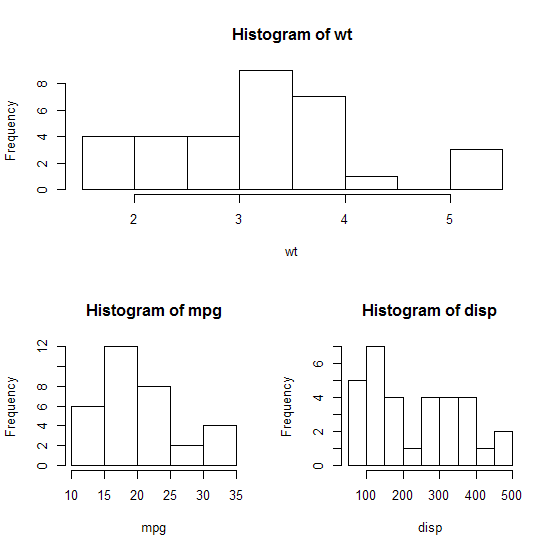
layout.show(3)

hist(wt)

hist(mpg)

hist(disp)

detach(mtcars)



* 再举例layout

m<-matrix(0:3,2,2)

layout(m) #矩阵中有0，0是不绘图的

layout.show(3)

layout(matrix(1:4,2,2)) #建立矩阵，将装置分割为4部分

layout.show(4)

layout(matrix(c(1:3,3),2,2)) #建立矩阵，将装置分割为3部分

layout.show(3) #显示布局编号

m<-matrix(1:4,2,2);m #建立矩阵m,2列2行

layout(m,widths=c(1,3),heights=c(3,1)) #将当时装置按照m进行划分，宽度之比为1:3，高度之比为3:1

layout.show(4)

layout(matrix(c(1,2,3,0,2,3,0,0,3),nr=3))

layout.show(3)

* 再学习书上的例子

#函数layout(matrix(1:m\*n,m,n)):分为不等大小的窗口

x<-pmin(3,pmax(-3,stats::rnorm(50))) #stats::rnorm(50)等价于rnorm(50)，产生50个标准正态分布的随机数

y<-pmin(3,pmax(-3,stats::rnorm(50))) #pmax(),pmin()是返回平行最大值（最小值），即结果中的第一个元素是各参数的第一个元素的最大值（最小值）

xhist<-hist(x,breaks=seq(-3,3,0.5),plot=TRUE)

yhist<-hist(y,breaks=seq(-3,3,0.5),plot=FALSE)

top<-max(c(xhist$counts,yhist$counts))

xrange<-c(-3,3);yrange<-c(-3,3)

layout(matrix(c(2,0,1,3),2,2,byrow=TRUE),c(3,1),c(1,3),TRUE) #矩阵中的1，2，3表示绘图窗口被划分成的区域序号（即后面作图在窗口中出现的次序），0为不作图，c（3，1）宽度按3：1分隔，c(1,3)表示高度按1：3

layout.show(3) #显示图形区域1到3

par(mar=c(3,3,1,1)) #设置图形到窗口的边界，按（下，左，上，右）顺序设置

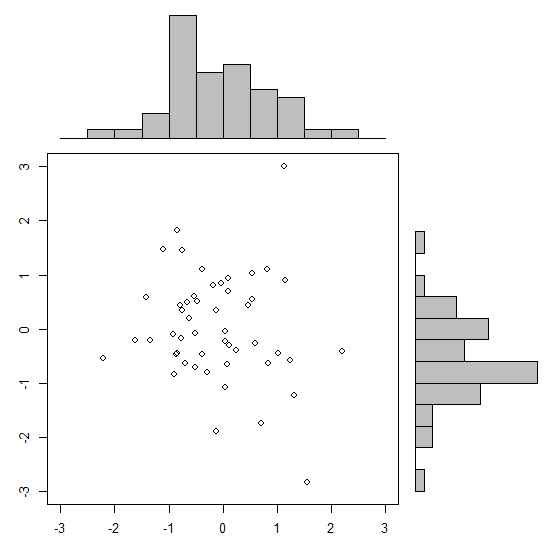
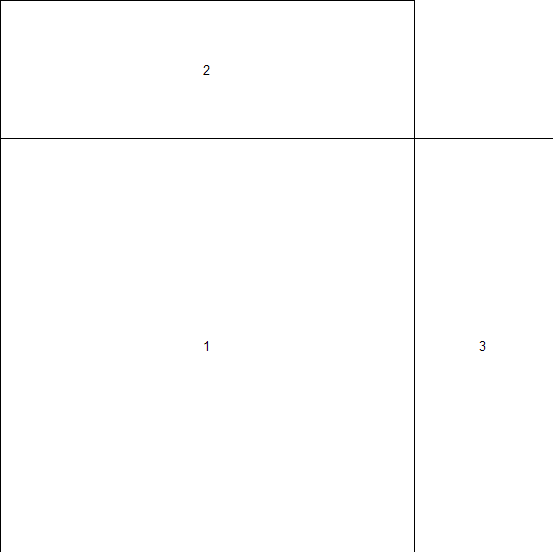
plot(x,y,xlim=xrange,ylim=yrange,xlab="",ylab="") #x,y间散点图

par(mar=c(0,3,1,1))

barplot(xhist$counts,axes=FALSE,ylim=c(0,top),space=0) #x的纵向条图

par(mar=c(3,0,1,1))

barplot(yhist$counts,axes=FALSE,ylim=c(0,top),space=0,horiz=TRUE) #y的水平条图)

****

本节学习的R函数小结

|  |  |  |
| --- | --- | --- |
| 第2章 2.6 R的绘图所介绍的主要函数 | | |
| 函数 | 功能 | 示例 |
|  |  |  |
|  |  |  |

**作业：练习本节所有命令**