**如果熟悉第一节基本绘图函数，可直奔 P4 第二、三节，提交练习 1-5。**

# 1 基本绘图函数

## 1.1 散点图 (函数 plot)

散点图（plot 函数）是研究变量关系的基本工具。我们以 R 自带的斐济地震数据集 quakes 为例，该数据是 1000× 5 的矩阵，各列 (变量) 为

lat (纬度), long (经度), depth (深度), mag (震级), stations (检测到地震的监测点个数)。

查看在线帮助文档了解数据背景: help(quakes) 。

```
colnames(quakes)
[1] "lat" "long" "depth" "mag" "stations"

## plot(x,y)
plot(x=quakes[,2], y=quakes[,1]) # x轴是经度, y轴是纬度
plot(quakes[,2], quakes[,1]) #前一个向量是x，后一个是y
plot(quakes[,2:1]) #quakes[,2:1]矩阵，第一列作为x-轴，第二列为y-轴。

## 选项type （"p", 散点; "l" : 连线; "b" (both) 既画点也连线; "n": nothing
plot(quakes[,2:1], type="b")

## 选项 pch: 点的类型（可为整数值， 也可以是字符）:
plot(quakes[,2:1], pch=1) #pch= 1,2,... 分别代表圆圈，三角，...
plot(quakes[,2:1], pch="*")
plot(quakes[,2:1], pch=c(rep("+",500),rep("-",500)))#前500个点为"+",其它为"-"
d = quakes[,3] <50 # 判断depth是否小于50
plot(quakes[,2:1], pch=d+1) # depth <50的点形状为2（三角）, 否则为点形状1
plot(quakes[,2:1], pch=d*16+1) # depth <50的点类型为17（实心三角）, 否则为1

## 选项 xlab, ylab: 指定坐标轴的标号(label)
plot(quakes[,2:1], xlab="long", ylab="lat",main="主标题",sub="副标题")

## 选项 col : 指定点的颜色（整数值或字符"red", "black"等）
plot(quakes[,2:1], pch=".", col=2) #红色标记, col颜色 (1:黑, 2:红, 3:绿, ...)
plot(quakes[,2:1], pch=".", col="blue") #蓝色
plot(quakes[,2:1], col=d+3) # depth<50的点为蓝色(4)，否则绿色(3)
m=quakes[,4] > 5.5 # mag > 5.5?
plot(quakes[,2:1], pch=d +1,col=m+1) #深度<50的为圆圈,否则三角

## 选项 cex: 定点或字符的大小（相对于标准大小的倍数，cex=2表示增大1倍）
plot(quakes[,2:1], cex=1.5)
plot(quakes[,2:1], pch=d+1, cex=m +1,col=m+1)
```

```
##多元散点图
plot(quakes)
pairs(quakes)
```

### 1.2 在已有图上添加元素 (points, text, symbols,legend,title,abline...)

points: 点 (pch 可取数值 1-20，对应于各种常用符号；也可取字符型值：pch=”ABC”)：

abline: 直线、

text: 文字、

symbols: 符号

title: 标题

legend：图例

```
## points, text, abline, title,....
plot(quakes[,2:1], type="n") # nothing
points(quakes[,2:1],pch=".") #
points(quakes[d,2:1],pch=16) # 深度d<50的点用点类型16标记
points(quakes[m,2:1],pch=17,col=2) #震级>5.5的点用点型17标记
text(176,-12, labels="Fiji earthquakes",cex=1.5,font=3,col=4)
  #cex: 字体大小, 缺省=1; font: 字体, 1,2,...)
abline(a=-25,b=0, col=3) #a:intercept, b:slope, # 或 abline(h=-25)
abline(v=175,col=3,lwd=3) #v: vertical line位置, lwd: 线的宽度
title(main="主标题 ", sub="子标题") # 标题

##legend: 图例说明前述图中符号含义(pch=16,17的含义):
legend(x=165.5,y=-33, legend=c("depth<50","mag>5.5"), pch=c(16,17),col=c(1,2))
#x,y: 图例的位置, legend: 说明, legend, pch, col的分量相互对应, 比如
pch=17, col=2 对应 "mag>5.5"

## symbols: 在图上添加圆圈, 方形, 星形图, boxplot
plot(quakes[,2:1], type="n") # nothing
symbols(quakes[,2:1], circles=quakes[,4], inches=0.1 ) #泡泡图(bubble plot)
 # circles大小由震级即第4列mag决定 (单位0.1 inches)
```

### 1.3 一元数据的分布（boxplot, hist, stem, qqnorm, qqplot）

(1) **boxplot**: 盒型图，若中位数线（盒子中间的分隔线）居中，上下关于中位数对称，则数据分布是对称的，或正态的。

(2) **hist, stem**: 直方图

(3) **qqnorm**: 检查数据是否正态分布，两个轴分别是数据的和正态分布的分位点 (quantile)，故称为 qq。若 qq 点在一条直线上则认为数据服从正态分布。

(4) **qqplot**: 检查两个数据的分布是否相同，两个轴分别是两个数据的分位点。若 qq 点在一条直线上，则认为两个数据分布相同。

```
boxplot(quakes[,3:5]) # 3-5列的盒形图
qqnorm(quakes[,4])
qqplot(quakes[,3],quakes[,4])
```

## 1.4 曲线和曲面

> **一元函数**: plot,
> **二元函数**: persp, image, contour, persp3d (in:rgl)

### 绘制 (已知) 一、二元概率密度函数

已知概率密度函数的数学表达式情况下，用 plot，persp 绘制密度图。

```
##一元正态密度函数
x=seq(-3, 3, length=100)
y=exp(-x^2/2)/sqrt(2*pi) # 一元标准正态 N(0,1) 的概率密度分布函数
plot(x,y, type="l", xlab="x", ylab="y", main="N(0,1)概率密度函数", col="blue")

##二元正态密度函数 (persp, image controur)
x = y = seq(-3, 3, length= 100)
f = function(x, y) { r <- exp(-(x^2+y^2)/2)/(2*pi) }
z = outer(x, y, f) # 二元正态密度在 (x[i],y[j]) 处的函数值
persp(x,y,z,col = "lightblue",main="二元标准正态分布的概率密度函数")
persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue",
       ltheta = 120, shade = 0.75, xlab = "", ylab = "", zlab = "",
       main="二元标准正态分布的概率密度函数")
persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "green3",
       ltheta = 120, shade = 0.8,xlab = "", ylab = "", zlab = "",
       main="二元标准正态分布的概率密度函数",
       box=F, axes=F, border=NA)
       #不画坐标轴(axes=F), 不画方框(box=F), 不画格子线(border=NA)

image(x,y,z) #x,y为坐标，z为坐标点(x,y)处的值
image(x,y,z, col=gray((0:32)/32) ) #
contour(x,y,z)
## persp3d(参见下一节)
```

### 估计并绘制密度函数: kde2d

实际二维数据的概率分布需要估计，可用 MASS 程序包中的 kde2d （kernel density estimate）估计，例子如下：

```
##数据:
paper = read.table("http://staff.ustc.edu.cn/~ynyang/vector/databook/T1-2.DAT")
colnames(paper)=c("Density", "Strength_MD", "Strength_CD")

## kde2d估计概率密度
library(MASS)
k <- kde2d(paper[,2],paper[,3], n=25) #n: x,y轴划分区间的个数

#二维变量的密度函数（左）和概率密度的热图、等高线图:
image(k, xlab="Strength_MD", ylab="Strength_CD" )
contour(k, add = TRUE, drawlabels = FALSE,nlevels=6)
persp(k, xlab="x", ylab="y",zlab="Prob. density",theta=30 )
```

## 2 3D 绘图软件包 rgl

R 本身没有 3D 作图功能。需要安装有 3D 功能的 R 软件包 (package)，比如 *rgl*.

```
> install.packages("rgl") #安装rgl，会弹出一个选择镜像站点的框
> library(rgl) #安装完毕后， 加载 rgl 。
plot3d(quakes[,1:3]) #按住鼠标左键,移动光标旋转图像。
plot3d(quakes[ ,c(1,2,4)], type="h") #z方向竖线
plot3d(quakes[ ,c(1,2,4)], type="s",radius=0.2) # 球形

## 动画演示上一个图
play3d(spin3d(axis = c(1, 0, 1), rpm = 5), duration = 30)
       # axis=c(0,0,1): 沿z轴旋转, rpm:速度, duration:演示时间


## rgl 包中的persp3d 可画三维动态图
bg3d("white")

#persp3d(x,y,z, col="green3") #拖动可旋转图像

persp3d(x, y, z, aspect = c(1, 1, 0.5), col = "lightblue", axes=F,
box=F,xlab="",ylab="",zlab="")
       #aspect:x,y,z方向的压缩比例，不画坐标轴(axes=F), 不画方框(box=F)

############## 制作动画########################
 #针对前面最后一个图(这里时persp3d(...)), 设置动画参数:
 M = par3d("userMatrix")
 f=par3dinterp(time = (0:2)*12, zoom=c(1,1.1,0.9 ),
 userMatrix = list(M, rotate3d(M, pi ,0, -1, 1), rotate3d(M, pi , -1,0, -1)))
 dur=5 #播放时间
 play3d(f , duration =dur)
#将动画存成moive.gif (dir指定存放位置)
 movie3d(f, duration = dur, dir=getwd(), clean=T)
```

软件包 rgl 中函数 persp3d 的帮助文档给出了旋转地球的例子

执行以下代码，观看地球动画。

```
       # install.packages("rgl")
       # library(rgl)
       lat <- matrix(seq(90, -90, len = 50)*pi/180, 50, 50, byrow = TRUE)
       long <- matrix(seq(-180, 180, len = 50)*pi/180, 50, 50)
       ##q 球坐标/极坐标:
       r <- 6378.1 # radius of Earth in km
       x <- r*cos(lat)*cos(long)
       y <- r*cos(lat)*sin(long)
```

```
z <- r*sin(lat)

persp3d(x, y, z, col = "white", texture = system.file("textures/world.png",
package = "rgl"),specular = "black", axes = FALSE, box = FALSE,
xlab = "", ylab = "",zlab = "", normal_x = x, normal_y = y, normal_z = z)
# 其中texture将世界地图放到球面上。

#连续播出persp3d:

play3d(spin3d(axis = c(0, 0, 1), rpm = 10), duration = 10)
```

# 3 球面均匀分布和球对称分布

## 3.1 球面均匀分布

$R^n$ 中单位球面记作 $S^{n-1}$，其上的均匀分布记作 $\sim U(S^{n-1})$。我们已知，

$$\mathbf{x} \sim N_n(\mathbf{0}, \sigma^2 I_n) \Longrightarrow \mathbf{u} = \mathbf{x}/\|\mathbf{x}\| \sim U(S^{n-1}),$$

我们将以此生成若干球面均匀分布随机向量，通过直方图散点图等形式考察 $\mathbf{u} = (u_1, ..., u_n)^\top \sim U(S^{n-1})$ 的一元边际 $u_1$ 或二元边际 $(u_1, u_2)^\top$ 的分布（因为球对称性，我们只需考察前一或两个坐标)。这种通过产生大量随机数/样本考察统计性质的方法称为蒙塔卡洛方法。

```
n=5 # 4维欧氏空间
N=5000 #样本量
X=matrix(rnorm(n*N), N, n) #N行，每一行都是n=4元标准正态随机向量
r2=apply(X^2, 1, sum) #每行的平方和
r= sqrt(r2) #共N个数，每个数是n=4元标准正态随机向量的模长
u=X/r # n维空间中球面上N个均匀分布的点
hist(u[,1] )  #一元边际分分布 （半圆律）
plot(u[,1:2],pch=".") #二元边际的散点图 （圆内均匀）

library(rgl)
plot3d(u[,1:3],xlab="",ylab="",zlab="", size=1,axes=F)
#三元边际散点图,
```

## 3.2 球内均匀分布

我们以单位球内均匀分布为例考虑从球对称分布中抽样（产生随机数）。我们称 $\mathbf{x} \sim U(B^n)$, 若其概率密度在单位球内是常数：

$$f(\mathbf{x}) = 1/|B^n|, \|\mathbf{x}\| \le 1.$$

如何产生随机数 $\mathbf{x} \sim U(B^n)$? 至少有三个途径

- 拒绝抽样：

  (a) 产生 $\mathbf{x} \sim U([-1,1]^n)$ 。注：产生 $x_1, ..., x_n$ iid $\sim U(-1,1)$, 则 $\mathbf{x} = (x_1, ..., x_n)^\top \sim U([-1,1]^n)$.

  (b) 若 $\mathbf{x}$ 满足“接受”条件 $\|\mathbf{x}\| \le 1$, 则 $\mathbf{x}$ 即是所求，即 $\mathbf{x} \sim U(B^n)$; 否则，弃用 $\mathbf{x}$, 重复 (a) 中的操作，直到满足录用条件 $\|\mathbf{x}\| \le 1$ 为止。

- 我们已知

$$\mathbf{u}\in R^{n+2},\mathbf{u}\sim U(S^{n+1})\Longrightarrow \mathbf{x}\overset{\Delta}{=}\mathbf{u}_{[1:n]}\sim U(B^n).$$

  即 $\mathbf{u}\sim U(S^{n+1})$ 的 $n$ 元边际 $\mathbf{u}[1:n]$ 服从 $B^n$ 球内均匀分布。

  因此，为了产生 $U(B^n)$ 随机数，我们只需产生 $U(S^{n+1})$ 随机数并取其 $n$ 个分量即可。

- 第三种生成方式：利用随机表示

$$\mathbf{x}\sim U(B^n)\Longleftrightarrow \mathbf{x}\overset{d}{=}\mathbf{u}r,\ \mathbf{u}\perp\!\!\!\perp r,\mathbf{u}\sim U(S^{n-1}),r^n\sim U(0,1).$$

  因此，为了产生 $U(B^n)$ 随机数 $\mathbf{x}$，我们只需产生 $\mathbf{u}\sim U(S^{n-1})$ 随机数 (参见 3.1), 以及 $U\sim U(0,1)$, 令 $r=U^{1/n}$, $\mathbf{x}=\mathbf{u}r$ 即可。

我们在练习 3 实现这三种方法。

## 3.3 其它球对称分布

利用一般球对称分布的随机表示

$$\mathbf{x}\text{球对称}\Longleftrightarrow \mathbf{x}\overset{d}{=}\mathbf{u}r,\ \mathbf{u}\perp\!\!\!\perp r,\mathbf{u}\sim U(S^{n-1}),r\text{分布任意}.$$

比如，假设 $r\sim U(0,1)$ 就可以得到单位球内的一个不同于均匀分布的某种分布（练习 5）。

---

**练习**

1. 用蒙特卡洛方法实验验证 $U(S^4)$ 的三元边际分布是单位球内的均匀分布
2. 对 $n=6,11,50,100,N=5000$, 产生 $\mathbf{u}_1,\ldots,\mathbf{u}_N$ iid $\sim U(S^{n-1})$, 考察一元边际和二元边际分布随 $n$ 增大而变化的规律。
3. 试用 3.2 所列三种方法生成 $N=5000$ 个 $U(B^3)$ 随机向量（样本）。
4. 若 $\mathbf{x}=(x_1,\ldots,x_n)^\top\sim U(B^n)$, 其一元边际 $x_1$ 服从什么分布？利用上一题中任一方法生成的 $U(B^3)$ 随机样本，用直方图考察 $U(B^3)$ 的一元边际分布。你能否猜测出该密度的数学表达形式？
5. 假设二元随机向量 $(x,y)^\top=(u_1,u_2)^\top\times r$, 其中 $(u_1,u_2)\sim U(S^1)$(单位圆周上的均匀分布), $r\sim U(0,1)$, 用蒙特卡洛方式产生大量二元随机数 $(x_i,y_i),i=1,\ldots,N$ 考察该分布，该分布与圆盘均匀分布有何差异？

---