R 数据图示创意扩展

张思艺 16307110398

#数据集

```
help(cars)
cars

plot(cars$dist~cars$speed) #散点图
```

#plot()函数

```
plot(cars$dist~cars$speed), # y~x
main="Relationship between car distance & speed", # 画标题
xlab="Speed (miles per hour)", #X 坐标轴标题
ylab="Distance travelled (miles)", #Y 坐标轴标题
xlim=c(0,30), #设置 X 轴范围为从 0 到 30
ylim=c(0,140), #设置 Y 轴范围为从 0 到 140
xaxs="i", #设置 X 轴风格 internal
yaxs="i", #设置 Y 轴风格 internal
col="red", #设置“散点”的颜色为红色
pch=19, #设置散点的形状为实心圆点—1 圆圈 2 三角形 3 加号等等；也可直接 pch="*"
cex=2) #设置散点的大小
#还可以设置标题、坐标轴的颜色
col.axis="blue",
col.lab="red",
col.main="darkblue"
```

#导入数据集

```
getwd()
setwd("E:/… ")
sales<-read.csv("dailysales.csv", header=TRUE)
```

#file 参数：必须的，相对路径或绝对路径（注：Windows 下路径要用斜杠'/'或者双反斜杠'\\'）。header 参数：默认为 FALSE 即数据框的列名为 V1,V2...,设置为 TRUE 时第一行作为列名。

```
plot(sales$units~as.Date(sales$date,"%d/%m/%y"))
```

改变参数：type="l" #注意是 l 不是| **画线图**

```
          main="Unit Sales in the month of January 2010"
          xlab="Date",
          ylab="Number of units sold",col="blue")
```

```
lines(sales$units~as.Date(sales$date,"%d/%m/%y"),col="red")  #lines()函数

sales<-read.csv("citysales.csv",header=TRUE)
barplot(sales$ProductA,names.arg= sales$City,col="black")  #柱形图 barplot()函数
```
改变参数：horiz=TRUE #水平柱形图
```
barplot(as.matrix(sales[,2:4]),beside=TRUE,legend=sales$City,col=heat.colors(5), border="white")  #彩色柱形图；beside=T 表示循环颜色；分别利用 heat.colors()、利用 rainbow()；5 换成 length(sales)试试


hist(rnorm(1000)) #直方图
islands{datasets} #岛屿数据集
hist(islands)

plot(density(rnorm(1000))) #密度图

metals<-read.csv("metals.csv",header=TRUE)
boxplot(metals, xlab="Metals", ylab="Atmospheric Concentration in ng per cubic metre", main="Atmospheric Metal Concentrations in London")  #箱型图
参数：outline=FALSE #排除离群值；horizonal=TRUE #水平放置；range=0 #延长虚线


iris #鸢尾花数据集
pairs(iris[,1:4]) #散点图阵
attach(iris) #attach 函数可以不用再给数据集打$，但记得最后要 detach()
summary(iris) #基本叙述统计资料
plot(Sepal.Length,Sepal.Width) #两个连续变量
plot(Species) #分类变量
plot(Species,Sepal.Length) #分类变量与连续变量
plot(iris) #散点图阵
改变参数：
    main="Relationships between characteristics of iris flowers",
    pch=19,
    col="blue"
    cex=0.9
detach(iris)

# 用定义函数的方式来画一组直方图
panel.hist <- function(x, ...)
{
par(usr = c(par("usr")[1:2], 0, 1.5) )
hist(x, prob=TRUE,add=TRUE,col="black",border="white")
}
```

```
plot(iris[,1:4],main="Relationships between characteristics of iris
flowers",pch=19,col="blue",cex=0.9,diag.panel=panel.hist)
# 对比 pairs(iris[,1:4])



par(mfrow=c(2,3)) #在一张画板上画多个散点图
plot(rnorm(100),col="blue",main="Plot No.1")
plot(rnorm(100),col="blue",main="Plot No.2")
plot(rnorm(100),col="green",main="Plot No.3")
plot(rnorm(100),col="black",main="Plot No.4")
plot(rnorm(100),col="green",main="Plot No.5")
plot(rnorm(100),col="orange",main="Plot No.6")

# par()的作用直到画板被关闭为止（或直至下一条 par()设置命令，或重新开一个图形设备）
par(bg="gray") #设置背景颜色
plot(rnorm(100))



market<-read.csv("dailymarket.csv",header=TRUE) #市场数据集
par(mfrow=c(3,1)) #在一张画板上画多个散点图
plot(market$revenue~as.Date(market$date,"%d/%m/%y"), type="l",  main="Revenue",
xlab="Date", ylab="US Dollars", col="blue")
plot(market$profits~as.Date(market$date,"%d/%m/%y"), type="l", main="Profits",
xlab="Date", ylab="US Dollars", col="red")
plot(market$customers~as.Date(market$date,"%d/%m/%y"), type="l", main="Customer
visits", xlab="Date", ylab="Number of people", col="black")



rain<-read.csv("cityrain.csv",header=TRUE)

plot(rain$Tokyo,type="l",col="red",ylim=c(0,300),main="Monthly Rainfall in
major cities",xlab="Month of Year",ylab="Rainfall (mm)",lwd=2)
lines(rain$NewYork,type="l",col="blue",lwd=2)
lines(rain$London,type="l",col="green",lwd=2)
lines(rain$Berlin,type="l",col="orange",lwd=2)

legend("topright",legend=c("Tokyo","NewYork","London","Berlin"),col=c("red","bl
ue","green","orange"),lty=1,lwd=2) #增加图例
legend("top",legend=c("Tokyo","NewYork","London","Berlin"),ncol=4,cex=0.8,bty="
n",col=c("red","blue","green","orange"),lty=1,lwd=2) #另一种增加图例（颜色数和
样本数要相等）
```

```
#再扩展一个填充颜色的
install.packages("maps")
library("maps")
map()
map('world', fill = TRUE,col=heat.colors(10))
```

扩展：一步步教你用 R 画“社交数据可视化：Facebook 好友联系图”
（https://flowingdata.com/2011/05/11/how-to-map-connections-with-great-circles/）

```
# attach(mtcars)
plot(mpg~disp, data=mtcars)
text(258,22,"Hornet") #用 text()标识点
#detach(mtcars)

health<-read.csv("HealthExpenditure.csv",header=TRUE)
plot(health$Expenditure,health$Life_Expectancy,type="n") # type="n"是在图形中数据不显示
text(health$Expenditure,health$Life_Expectancy,health$Country) #用 text()标识点
```

 # 误差条

```
plot(mpg~disp,data=mtcars)
arrows(x0=mtcars$disp,y0=mtcars$mpg*0.95,x1=mtcars$disp,y1=mtcars$mpg*1.05,angle=90,code=3,length=0.04,lwd=0.4)
arrows(x0=mtcars$disp*0.95,y0=mtcars$mpg,x1=mtcars$disp*1.05,y1=mtcars$mpg,angle=90,code=3,length=0.04,lwd=0.4)
```

 # jitter()函数：给向量加上少许噪音

```
x <- rbinom(1000, 10, 0.25)
y <- rbinom(1000, 10, 0.25)
plot(x,y)
plot(jitter(x), jitter(y))
```

 # 线性模型：画回归直线

```
plot(mtcars$mpg~mtcars$disp)
lmfit<-lm(mtcars$mpg~mtcars$disp)
abline(lmfit)
```

 # lowess：局部加权回归散点平滑法

```
plot(cars, main = "lowess(cars)")
lines(lowess(cars), col = "blue")
lines(lowess(cars, f=0.3), col = "orange") #f 是 lowess 算法的一个参数。
```

（扩展：一个关于量化投资的现实应用例子
http://blog.sina.com.cn/s/blog_7dd6586501l0tpm2.html）

 # QQ 图（研究正态分布的）

```
qqnorm(mtcars$mpg)
qqline(mtcars$mpg)
```

 # 线性回归模型的检验

```
lmfit<-lm(mtcars$mpg~mtcars$disp)
par(mfrow=c(2,2))
plot(lmfit)
```

 # rug()地毯函数 在坐标轴上显示数据密度

```
x<-rnorm(1000)
plot(density(x))
rug(x)

metals<-read.csv("metals.csv")
plot(Ba~Cu,data=metals,xlim=c(0,100))
rug(metals$Cu)
rug(metals$Ba,side=2,col="red",ticksize=0.02)
```

 # 扩展：smoothScatter()函数

对于高密度的散点图（重叠很严重，很难观察变量关系）来说，smoothScatter()函数可利用核密度估计生成用颜色密度来表示点分布的散点图，利用**光平滑密度**估计绘制的散点图，使密度易读性更强

```
n <- 10000
x <- matrix(rnorm(n), ncol=2)
y <- matrix(rnorm(n, mean=3, sd=1.5), ncol=2)
plot(x,y)  #对比
smoothScatter(x,y)
```

 # 扩展：三维散点图

```
install.packages("scatterplot3d")
library(scatterplot3d)
scatterplot3d(x=mtcars$wt,y=mtcars$disp,z=mtcars$mpg)
scatterplot3d(mtcars$wt,mtcars$disp,mtcars$mpg,pch=16,
highlight.3d=TRUE,angle=20,xlab="Weight",ylab="Displacement",zlab="Fuel Economy
(mpg)",type="h",main="Relationships between car specifications")
```

（再扩展：高维数据展现专业扩展包 rggobi——install.packages("rggobi")
http://www.ggobi.org/rggobi/ 动画 demo http://www.ggobi.org/demos/）

 # 饼图

```
browsers<-read.table("browsers.txt",header=TRUE)
browsers<-browsers[order(browsers[,2]),]
pie(browsers[,2],labels=browsers[,1],clockwise=TRUE,radius=1,col=brewer.pal(7,"
Set1"),border="white",main="Percentage Share of Internet Browser usage")

pielabels <- sprintf("%s = %3.1f%s", browsers[,1],
100*browsers[,2]/sum(browsers[,2]), "%") # 在饼图上标注百分比
pie(browsers[,2],labels=pielabels,clockwise=TRUE,radius=1,col=brewer.pal(7,"Se
t1"),border="white",cex=0.8,main="Percentage Share of Internet Browser usage")

pie(browsers[,2],labels=NA,clockwise=TRUE,col=brewer.pal(7,"Set1"),border="whit
e",radius=0.7,cex=0.8,main="Percentage Share of Internet Browser usage")
legend("bottomright",legend=pielabels,bty="n",fill=brewer.pal(7,"Set1")) # 增加
图释
```

 # 输出为图形（或其他）文件：

```
png("scatterplot.png")
plot(rnorm(1000))
dev.off()
```

分享：R 语言自学视频 https://www.bilibili.com/video/av28145975?t=145&p=7