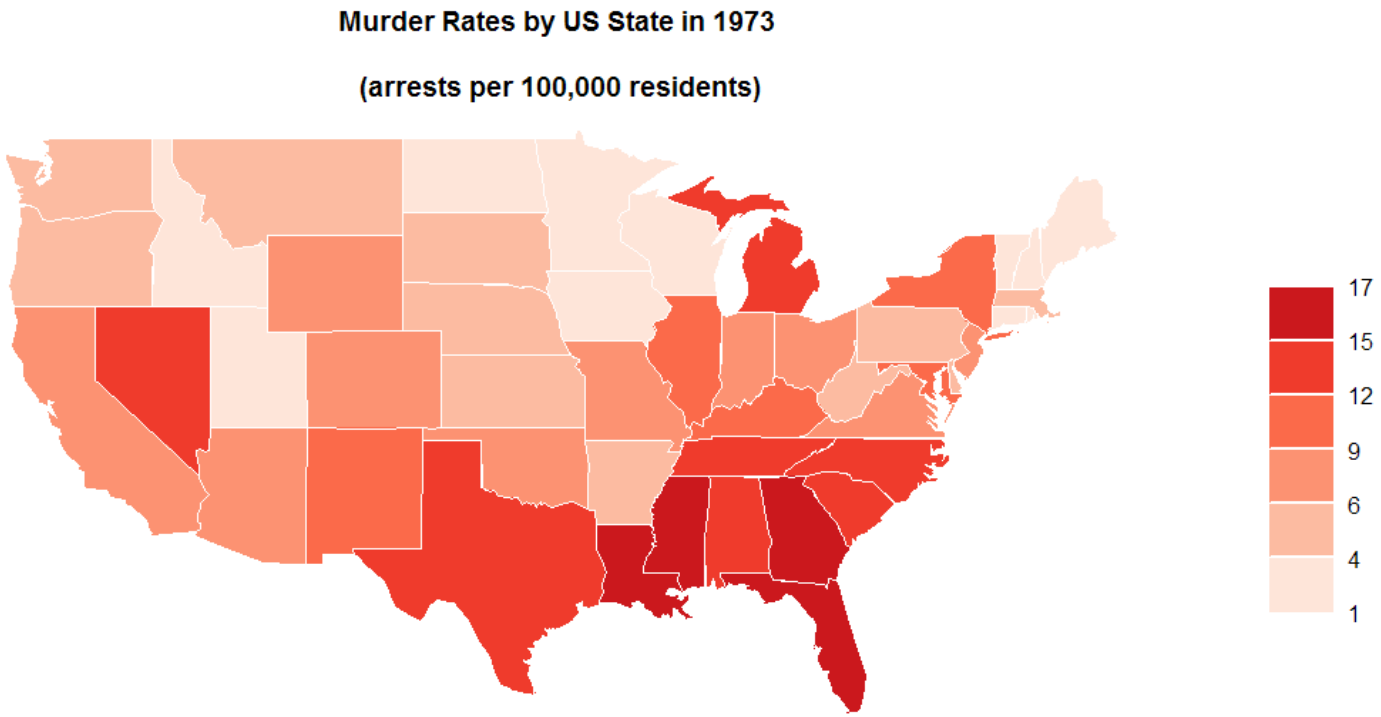
DATAGURU 炼数成金



数据分析、展现与R语言 第14周

2013.05.11

法律声明

【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

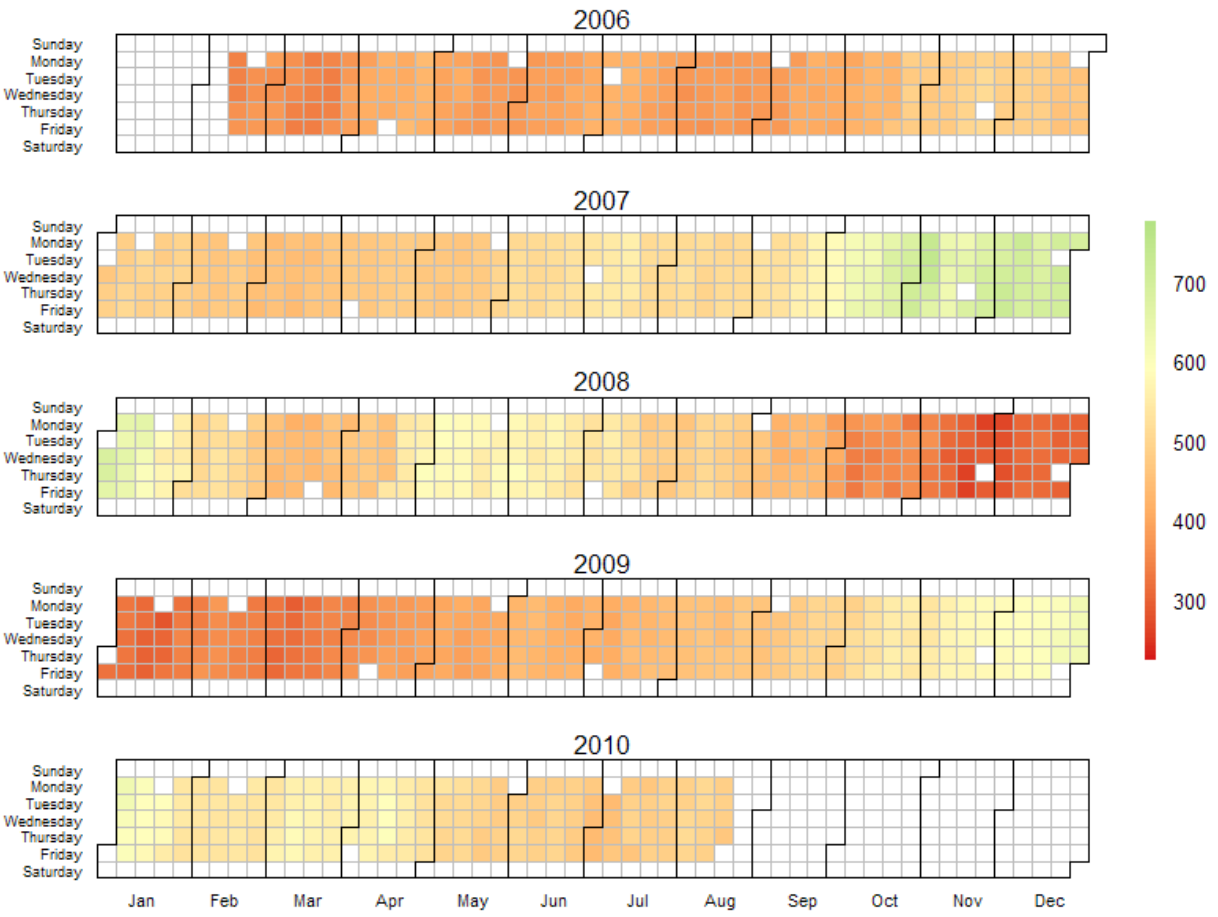
http://edu.dataguru.cn

使用chron包进行日历形式展示

使用第八章数据

```
install.packages("chron")
library("chron")

source("calendarHeat.R")
stock.data <- read.csv("google.csv")

calendarHeat(dates=stock.data$Date,
values=stock.data$Adj.Close,
varname="Google Adjusted Close")
```

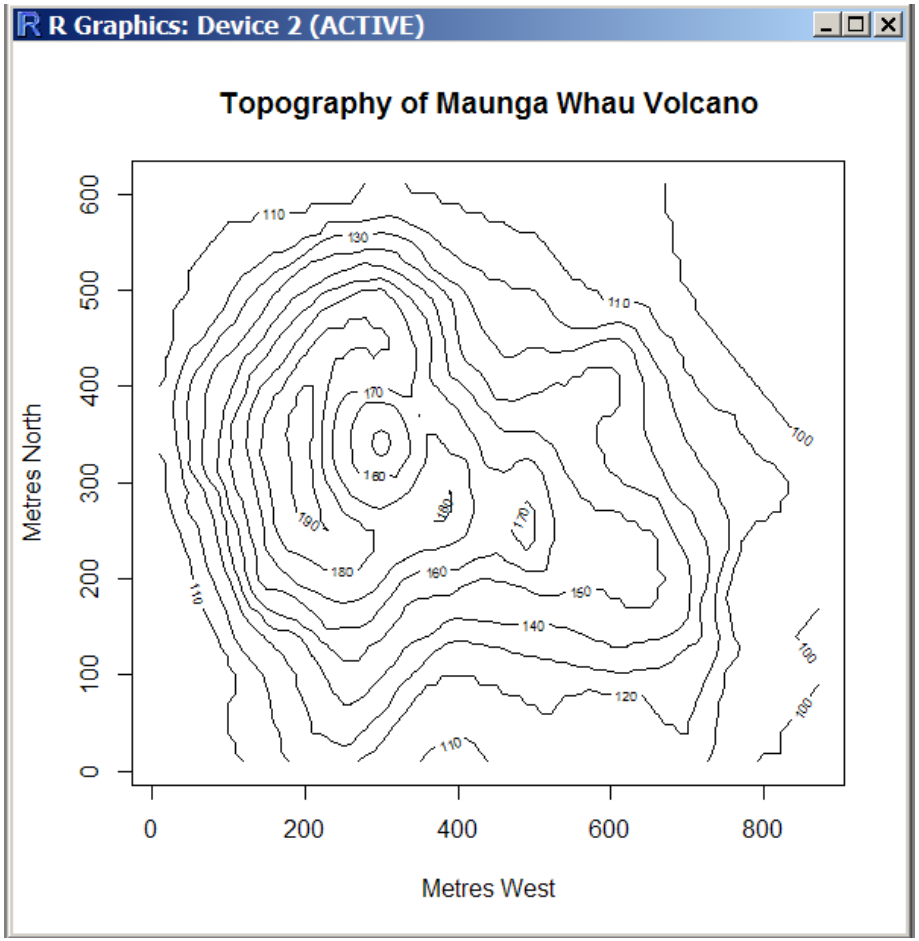
效果图



2013.05.11

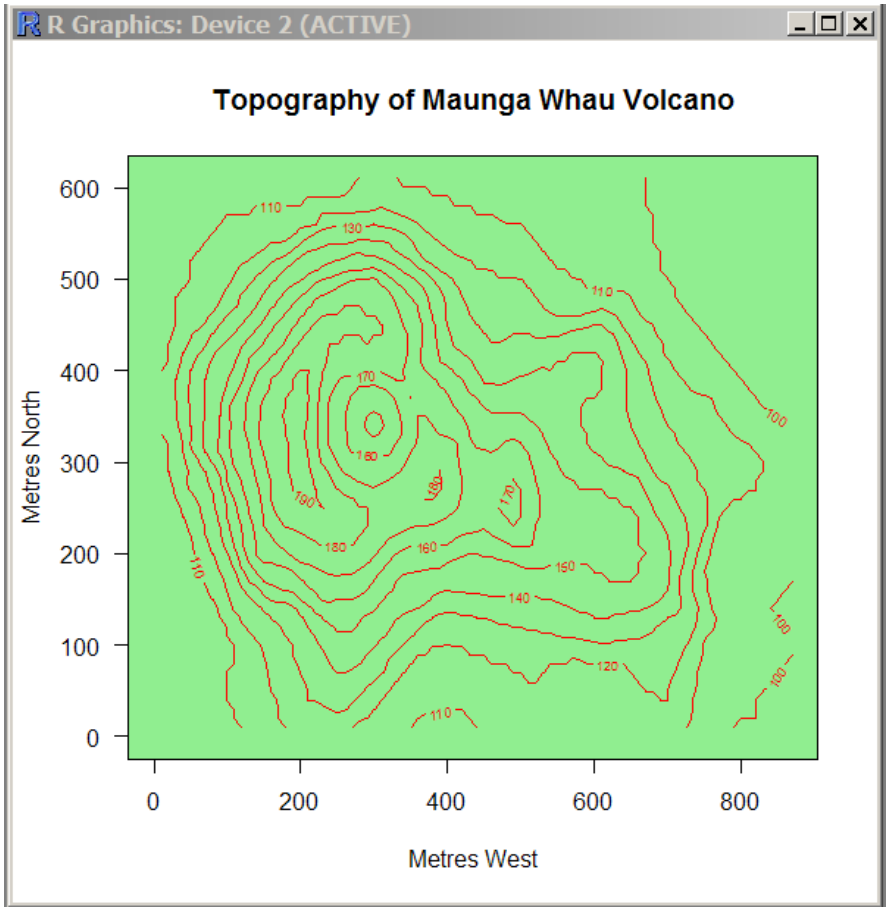
等高线图

```
contour(x=10*1:nrow(volcano),
    y=10*1:ncol(volcano), z=volcano,
xlab="Metres West",ylab="Metres
    North",
main="Topography of Maunga
    Whau Volcano")
```



润色等高线图

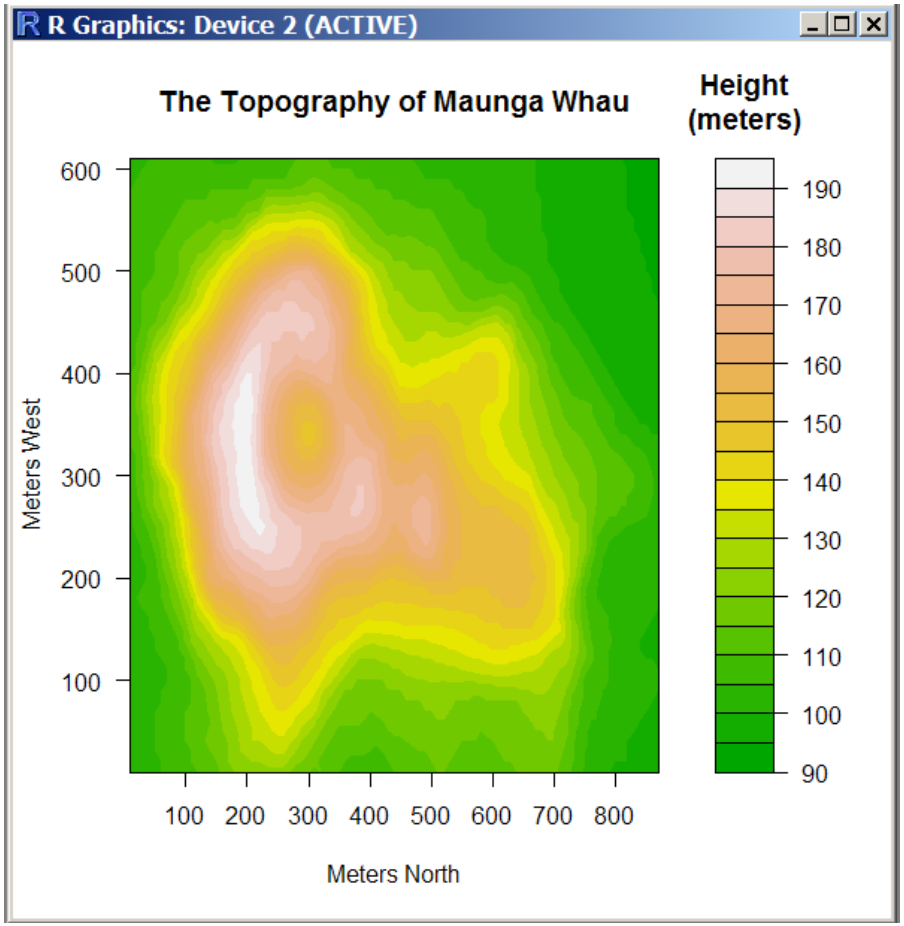
```
par(las=1)
plot(0,0,xlim=c(0,10*nrow(volcano)),ylim=
     c(0,10*ncol(volcano)),
type="n",xlab="Metres West",
ylab="Metres North",main="Topography
     of Maunga Whau Volcano")
u<-par("usr")
rect(u[1],u[3],u[2],u[4],col="lightgreen")
contour(x=10*1:nrow(volcano),y=10*1:nco
     l(volcano),
volcano,col="red",add=TRUE)
```



2013.05.11

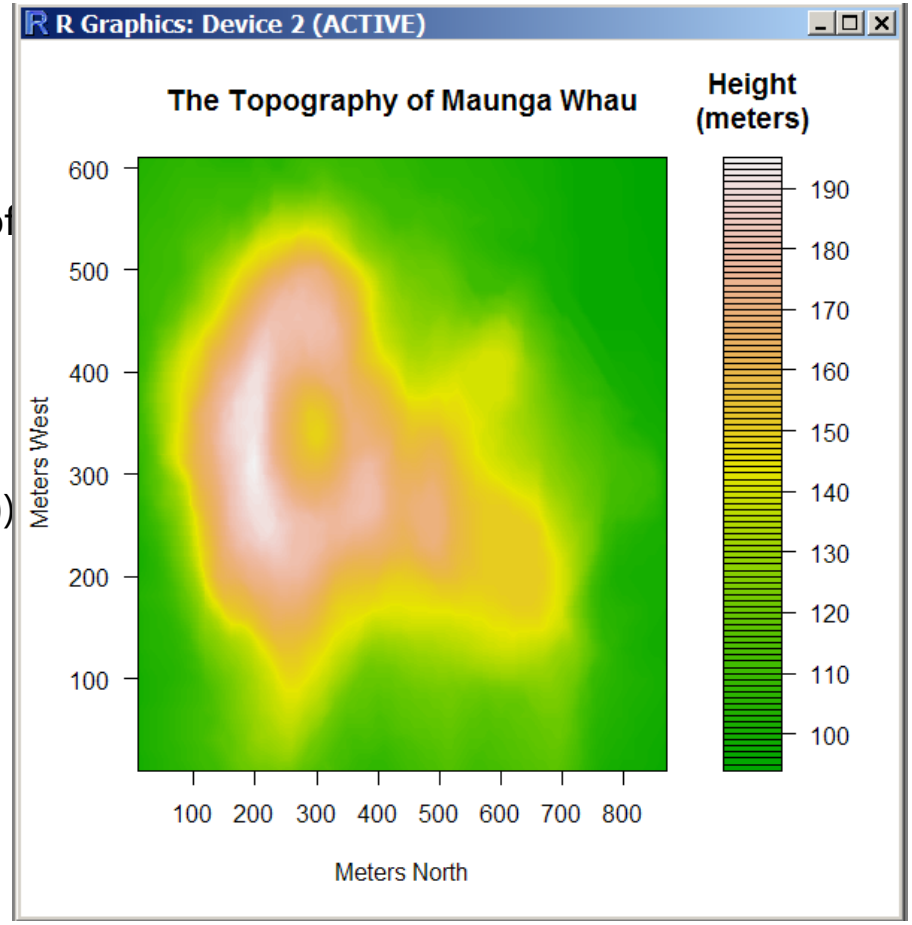
填充颜色

```
filled.contour(x = 10*1:nrow(volcano),y =
    10*1:ncol(volcano),
z = volcano, color.palette = terrain.colors,
plot.title = title(main = "The Topography of
    Maunga Whau",
xlab = "Meters North",ylab = "Meters
    West"),
plot.axes = {axis(1, seq(100, 800, by = 100))
axis(2, seq(100, 600, by = 100))},
key.title = title(main="Height\n(meters)"),
key.axes = axis(4, seq(90, 190, by = 10)))
```



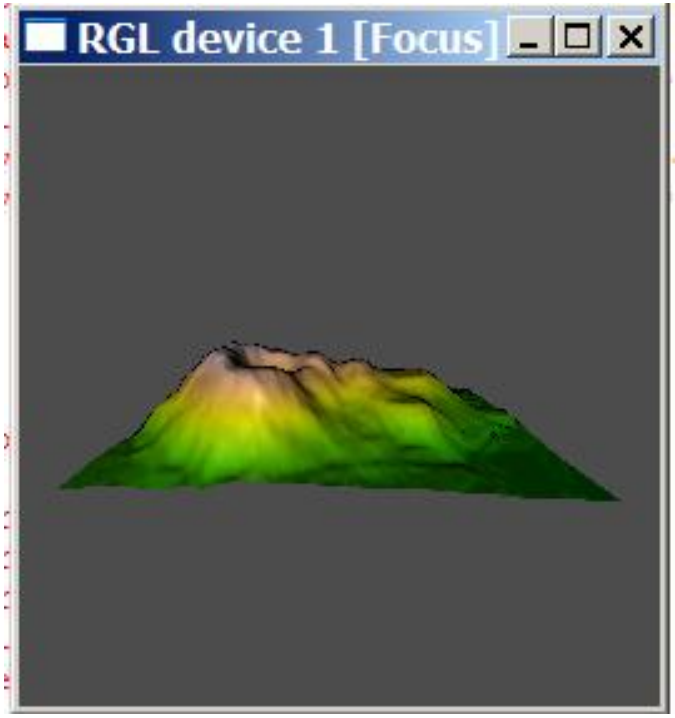
改善光滑度

```
filled.contour(x = 10*1:nrow(volcano),
y = 10*1:ncol(volcano), z = volcano,
color.palette = terrain.colors,
plot.title = title(main = "The Topography of
    Maunga Whau",
xlab = "Meters North",ylab = "Meters
    West"),nlevels=100,
plot.axes = {axis(1, seq(100, 800, by = 100))
axis(2, seq(100, 600, by = 100))},
key.title = title(main="Height\n(meters)"),
key.axes = axis(4, seq(90, 190, by = 10)))
```



使用rgl包制作三维曲面图

```
library(rgl)


z <- 2 * volcano
x <- 10 * (1:nrow(z))
y <- 10 * (1:ncol(z))
zlim <- range(z)
zlen <- zlim[2] - zlim[1] + 1
colorlut <- terrain.colors(zlen)
col <- colorlut[ z-zlim[1]+1 ]
rgl.open()
rgl.surface(x, y, z, color=col, back="lines")
```



利用maps包画出地图

install.packages("maps")

library(maps)

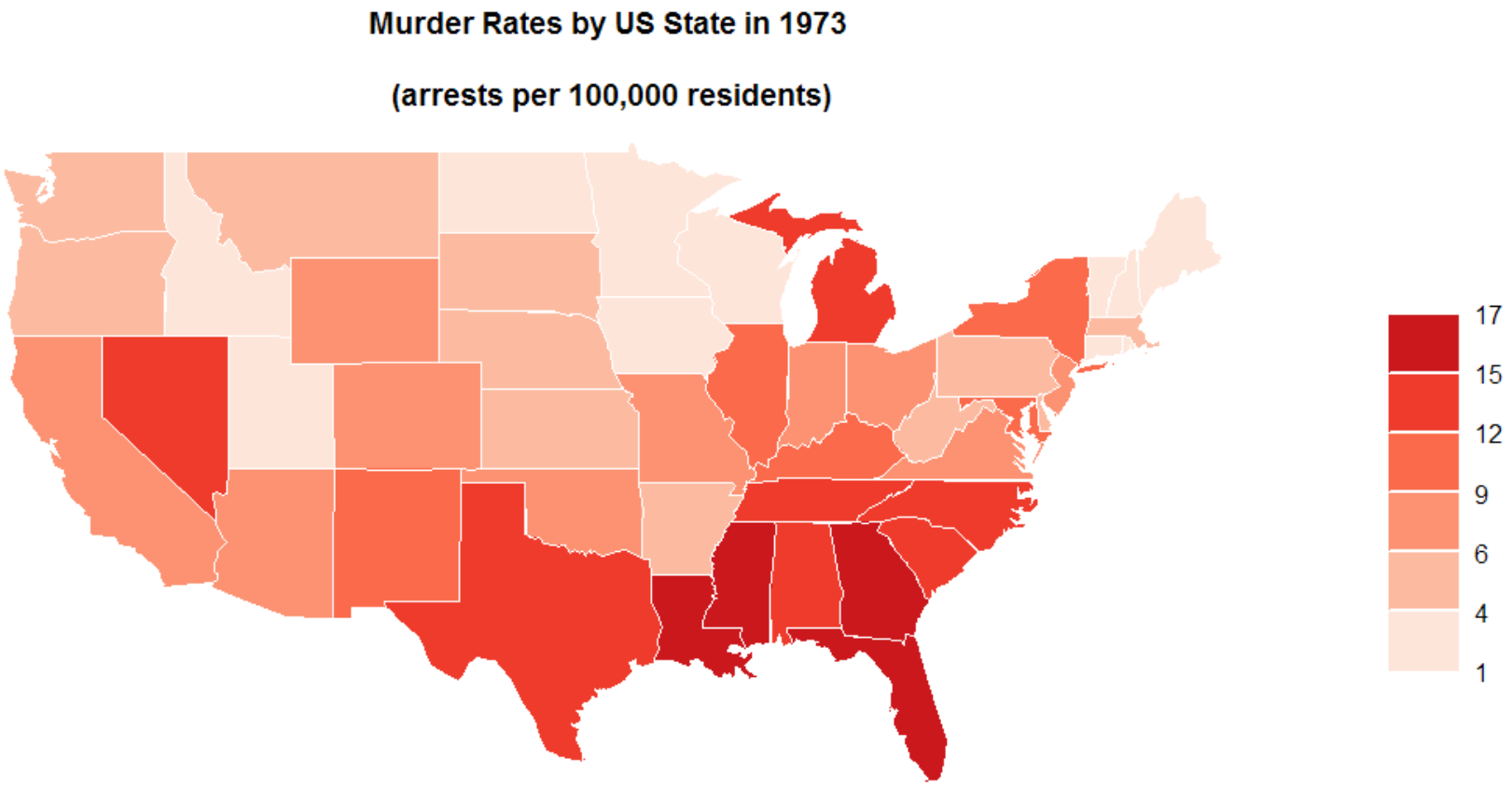
代码

```
x<-map("state",plot=FALSE)
for(i in 1:length(rownames(USArrests))) {
for(j in 1:length(x$names)) {
if(grepl(rownames(USArrests)[i],x$names[j],ignore.case=T))
x$measure[j]<-as.double(USArrests$Murder[i])
}
}
colors <- brewer.pal(7,"Reds")
sd <- data.frame(col=colors,
values=seq(min(x$measure[!is.na(x$measure)]),
max(x$measure[!is.na(x$measure)])*1.0001,
length.out=7))
breaks<-sd$values
matchcol<-function(y) {
as.character(sd$col[findInterval(y,sd$values)])
}
layout(matrix(data=c(2,1), nrow=1, ncol=2),
widths=c(8,1), heights=c(8,1))
```

代码

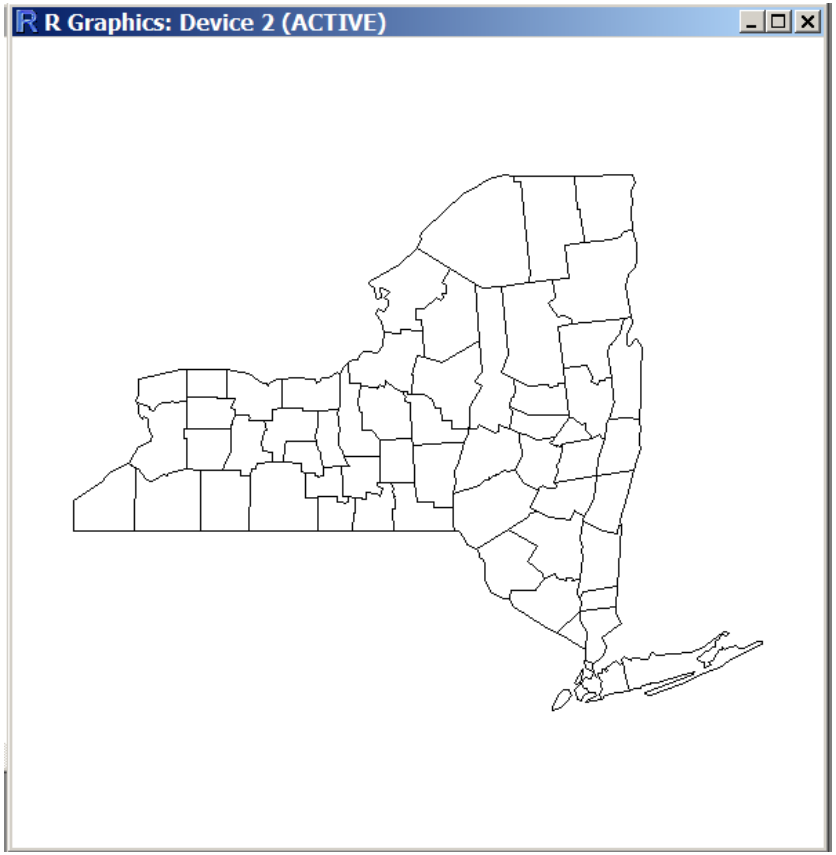
```
# Color Scale first
par(mar = c(20,1,20,7),oma=c(0.2,0.2,0.2,0.2),mex=0.5)
image(x=1, y=0:length(breaks),z=t(matrix(breaks))*1.001,
col=colors[1:length(breaks)-1],axes=FALSE,breaks=breaks,
xlab="", ylab="", xaxt="n")
axis(4,at=0:(length(breaks)-1),
labels=round(breaks),col="white",las=1)
abline(h=c(1:length(breaks)),col="white",lwd=2,xpd=F)
#Map
map("state", boundary = FALSE,col=matchcol(x$measure),
fill=TRUE,lty="blank")
map("state", col="white",add = TRUE)
title("Murder Rates by US State in 1973 \n
(arrests per 100,000 residents)", line=2)
```

效果图


Murder Rates by US State in 1973
(arrests per 100,000 residents)
17
15
12
9
6
4
1

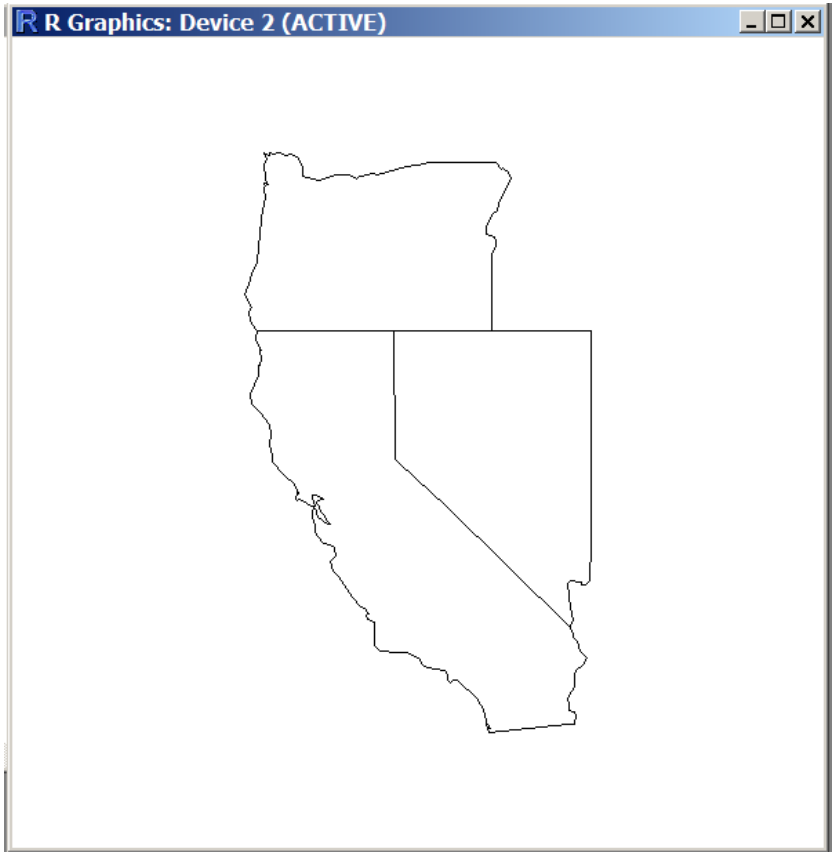
纽约

map("county", "new york")



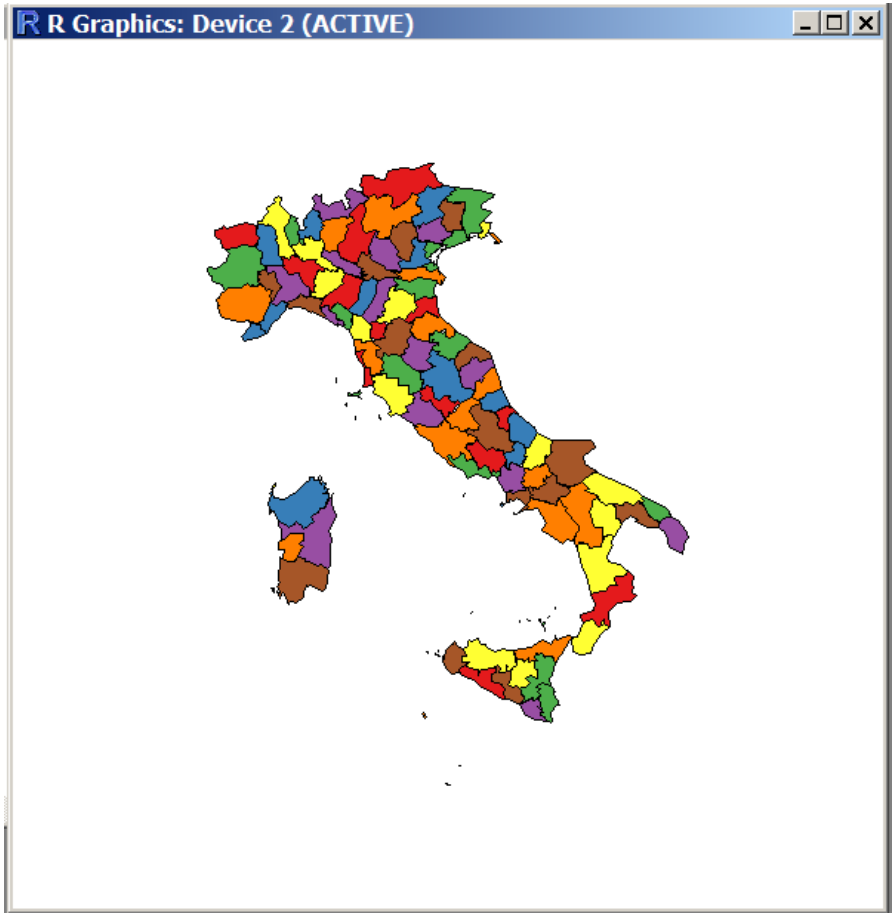
加州，俄勒冈，内华达

```
map("state", region = c("california", "oregon", "nevada"))
```



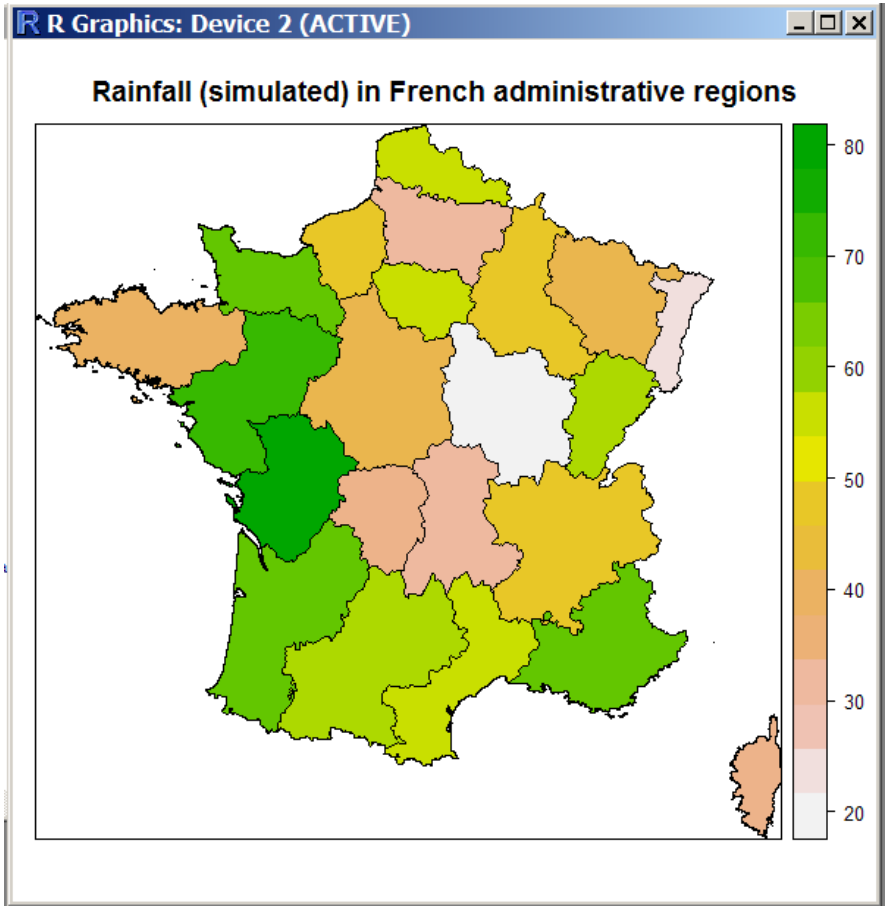
意大利

map('italy', fill = TRUE, col = brewer.pal(7,"Set1"))



利用sp包画降雨量图

```
install.packages("sp")
library(sp)


load(url("http://gadm.org/data/rda/FR
    A_adm1.RData"))
gadm$rainfall<-
    rnorm(length(gadm$NAME_1),mea
    n=50,sd=15)
spplot(gadm,"rainfall",
col.regions =
    rev(terrain.colors(gadm$rainfall)),
main="Rainfall (simulated) in French
    administrative regions")
```



使用google地图包RgoogleMaps

install.packages("rgdal")

library(rgdal)

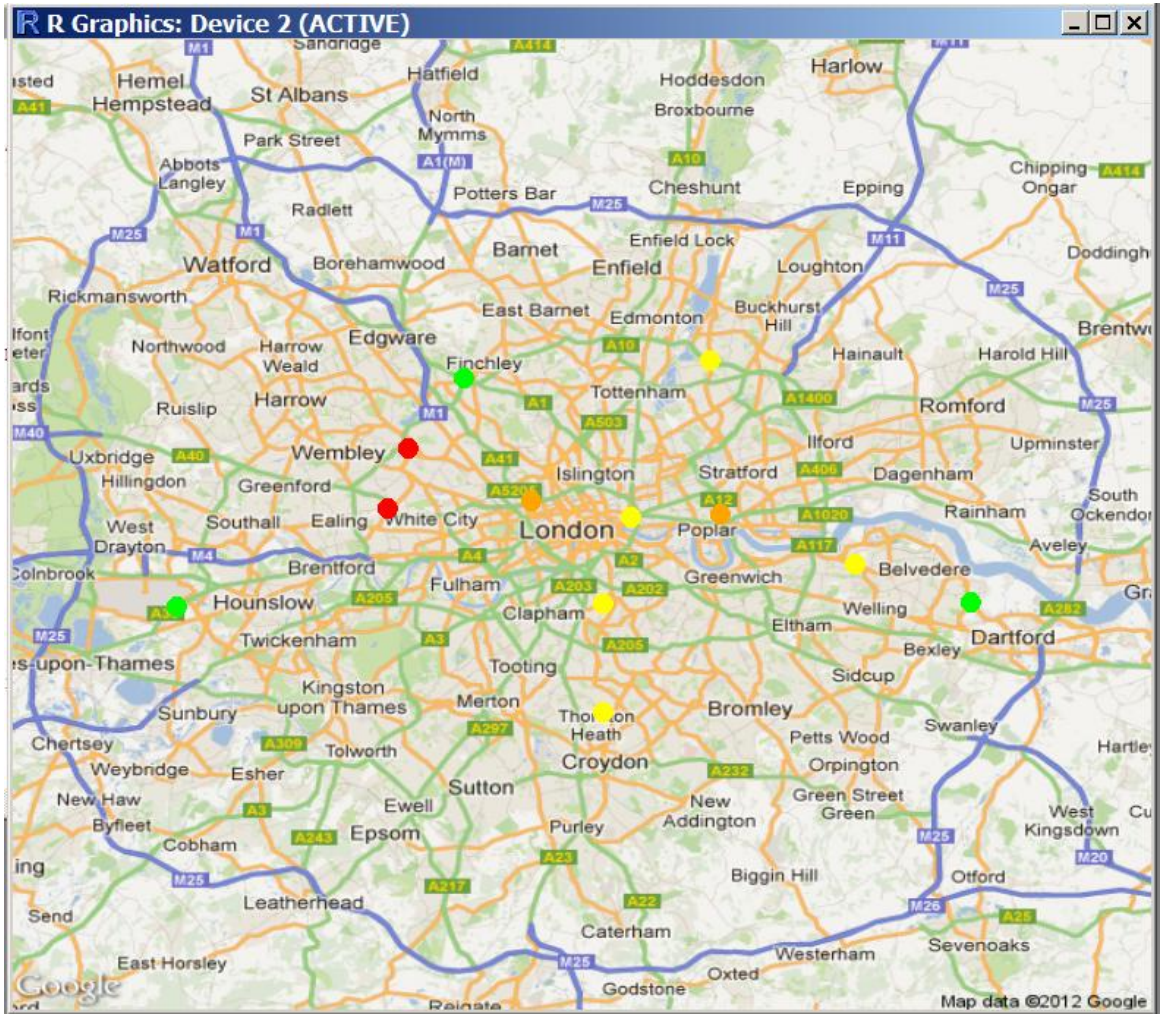
install.packages("RgoogleMaps")

library(RgoogleMaps)

伦敦

```
air<-read.csv("londonair.csv")


london<-GetMap(center=c(51.51,-0.116),
zoom =10, destfile = "London.png",maptype = "mobile")
PlotOnStaticMap(london,lat = air$lat, lon = air$lon,
cex=2,pch=19,col=as.character(air$color))
```

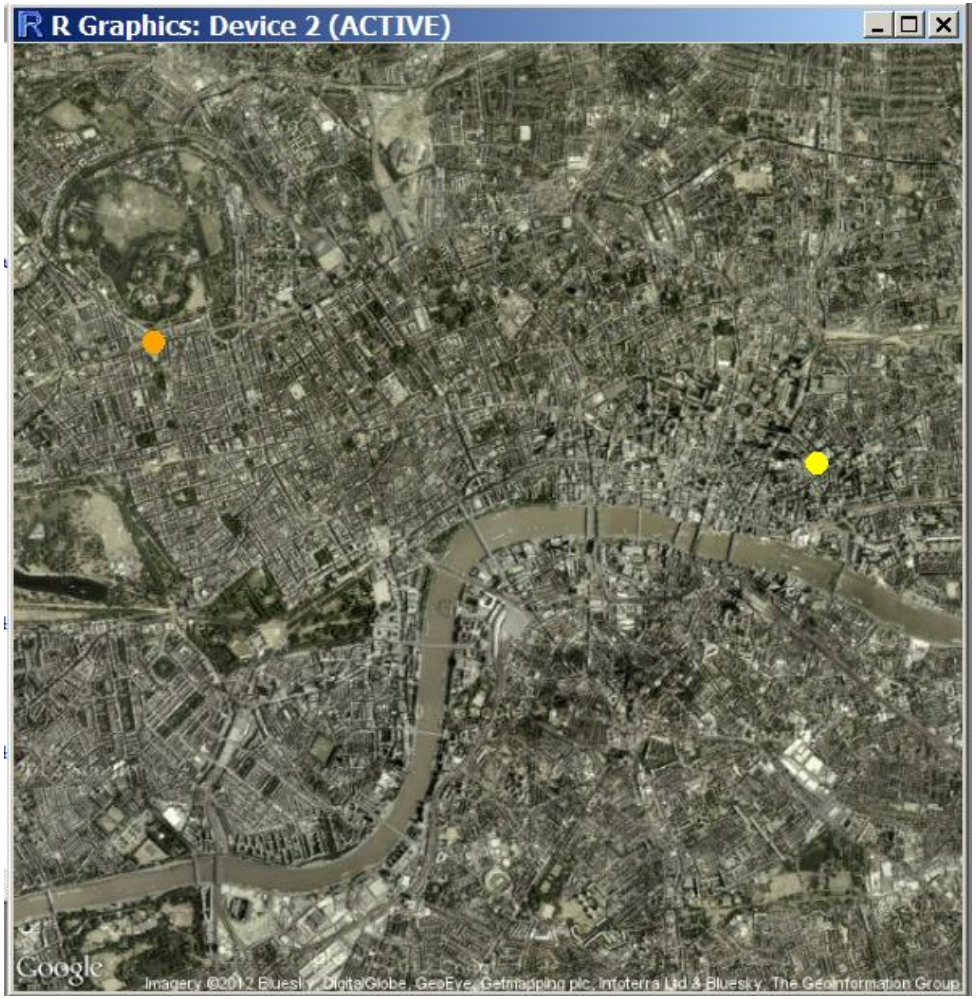
效果图



2013.05.11

卫星地图

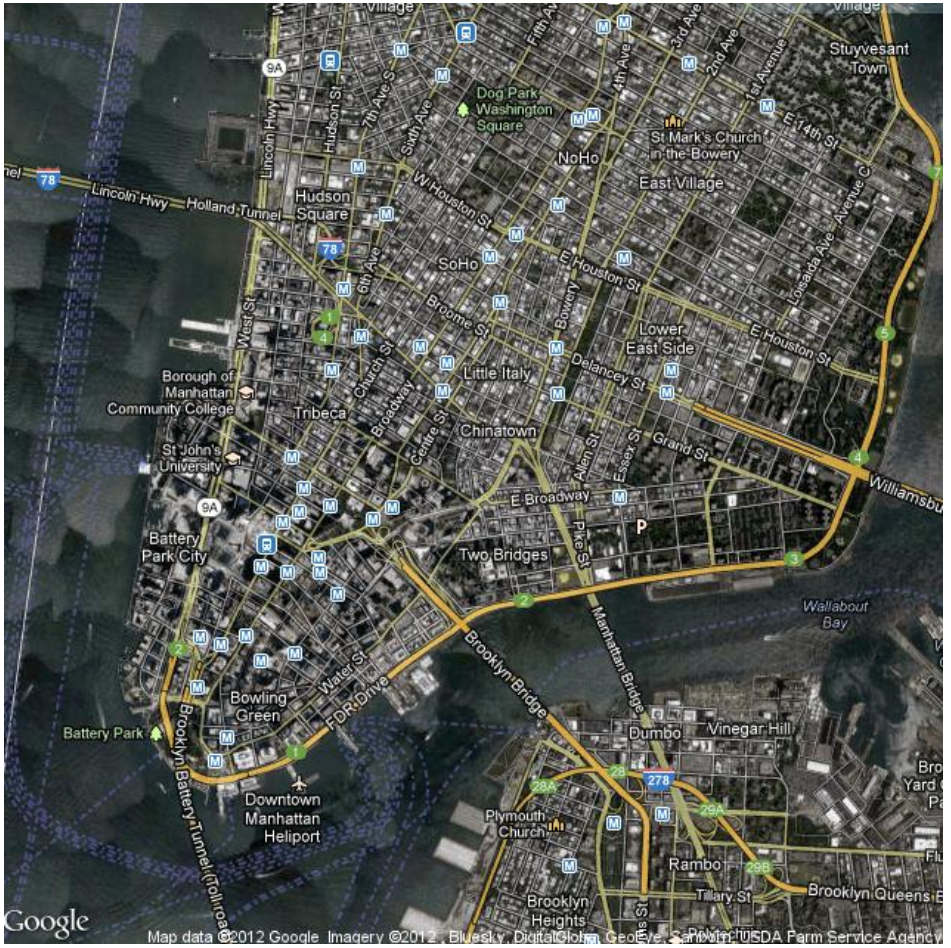
```
london<-
    GetMap(center=c(51.51,-
    0.116),zoom =13,
destfile =
    "London_satellite.png",mapty
    pe = "satellite")
PlotOnStaticMap(london,lat =
    air$lat, lon = air$lon,
cex=2,pch=19,col=as.character(ai
    r$color))
```



2013.05.11

把地图直接输出到图像文件

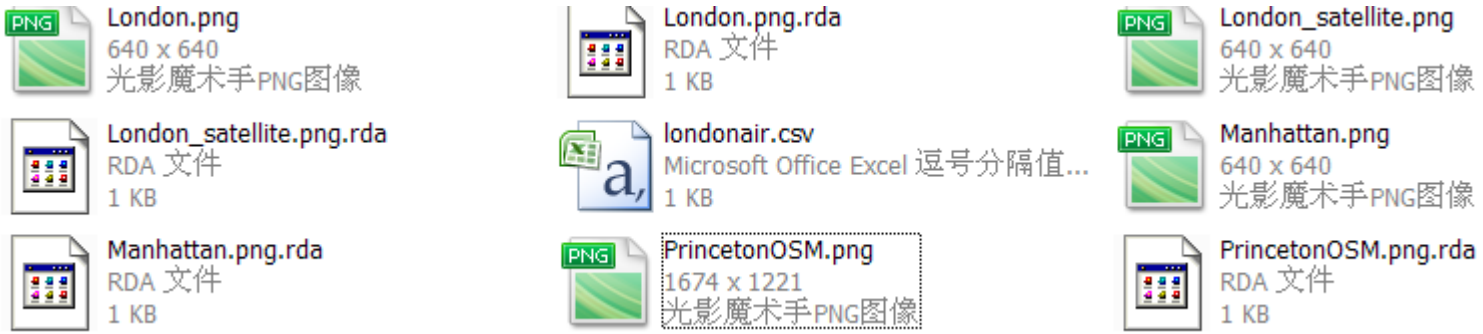
```
GetMap(center=c(40.714728,-
    73.99867), zoom =14,
destfile = "Manhattan.png",
    maptype = "hybrid");
```



2013.05.11

高精度输出图像文件

```
GetMap.OSM(lonR= c(-74.67102, -74.63943),
latR = c(40.33804,40.3556),scale = 7500,
destfile = "PrincetonOSM.png")
```



效果图

2013.05.11

KML数据

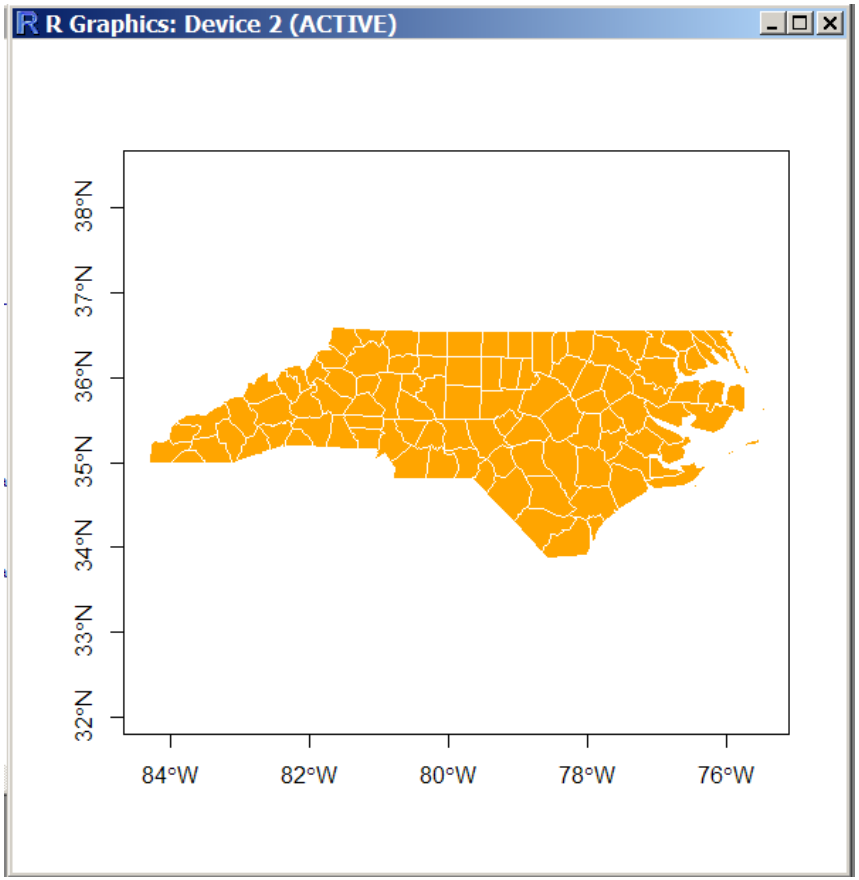
- **Google's Keyhole Markup Language (KML) format**

```
install.packages("rgdal")
library(rgdal)

cities <-
    readOGR(system.file("vectors",pack
    age = "rgdal")[1],"cities")
writeOGR(cities, "cities.kml", "cities",
    driver="KML")
df <- readOGR("cities.kml", "cities")
```

```
> cities <- readOGR(system.file("vectors",package = "rgdal")[1],"cities")
OGR data source with driver: ESRI Shapefile
Source: "C:/Program Files/R/R-2.14.2/library/rgdal/vectors", layer: "cities"
with 606 features and 4 fields
Feature type: wkbPoint with 2 dimensions
> writeOGR(cities, "cities.kml", "cities", driver="KML")
> df <- readOGR("cities.kml", "cities")
OGR data source with driver: KML
Source: "cities.kml", layer: "cities"
with 606 features and 2 fields
Feature type: wkbPoint with 2 dimensions
> df
               coordinates                     Name Description
1       (33.086, 68.9635)                  Murmansk
2      (40.6462, 64.5207)                Arkhangelsk
3      (30.4533, 59.9519)           Saint Petersburg
4        (150.78, 59.571)                   Magadan
5      (56.2325, 58.0002)                     Perm'
6      (60.6101, 56.8465)              Yekaterinburg
7      (43.9407, 56.2897)           Nizhniy Novgorod
8     (-4.26995, 55.8628)                   Glasgow
9       (49.1455, 55.733)                     Kazan'
10      (61.3926, 55.145)                Chelyabinsk
```

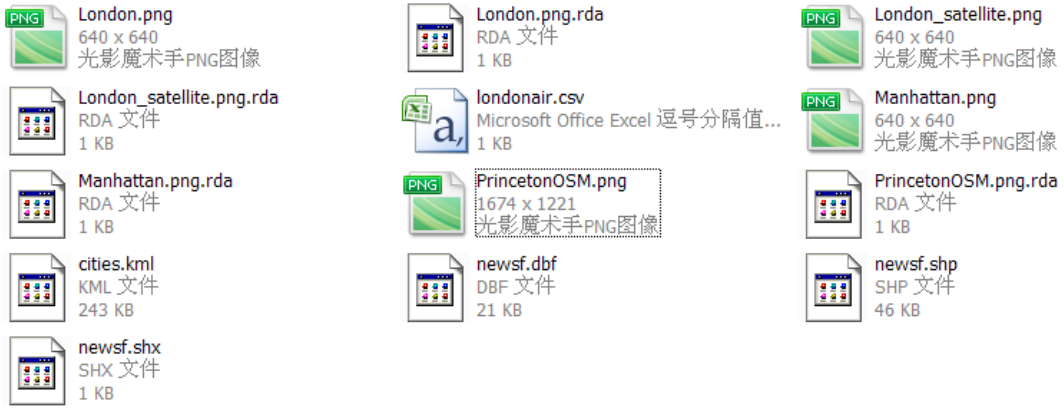
ESRI形状文件

```
install.packages("maptools")
library(maptools)

sfdata <-
    readShapeSpatial(system.file("sha
    pes/sids.shp",
package="maptools")[1],
    proj4string=CRS("+proj=longlat")
    )
plot(sfdata, col="orange",
    border="white", axes=TRUE)
```



shapefiles 包

```
install.packages("shapefiles")
library(shapefiles)

sf<-system.file("shapes/sids.shp", package="maptools")[1]
sf<-substr(sf,1,nchar(sf)-4)
sfdata <- read.shapefile(sf)

write.shapefile(sfdata, "newsf")
```

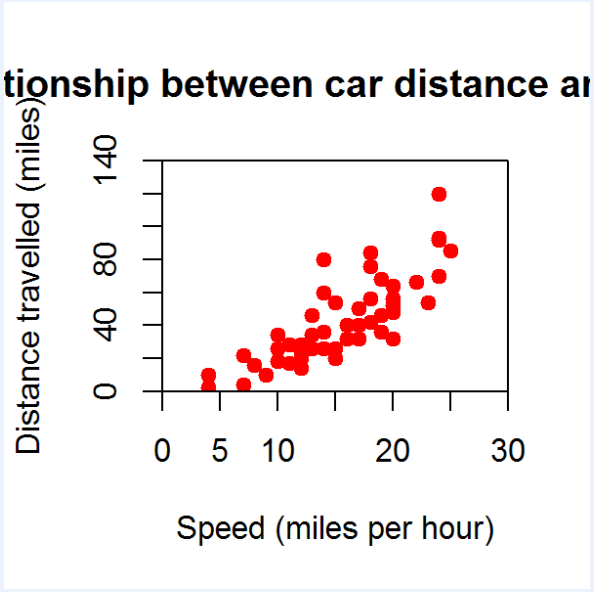


输出图像文件的一般用法

```
png("cars.png",res=200,height=600,width=600)
plot(cars$dist~cars$speed,
main="Relationship between car distance and speed",
xlab="Speed (miles per hour)",ylab="Distance travelled (miles)",
xlim=c(0,30),ylim=c(0,140),
xaxs="i",yaxs="i",col="red",pch=19)
dev.off()
```

结果

名称	大小	类型	修改日期
cars.png	10 KB	光影魔术手PNG图像	2012-12-16 12:44
dailysales.csv	1 KB	Microsoft Office Exc...	2010-4-9 13:58
themes.csv	1 KB	Microsoft Office Exc...	2010-11-29 11:07

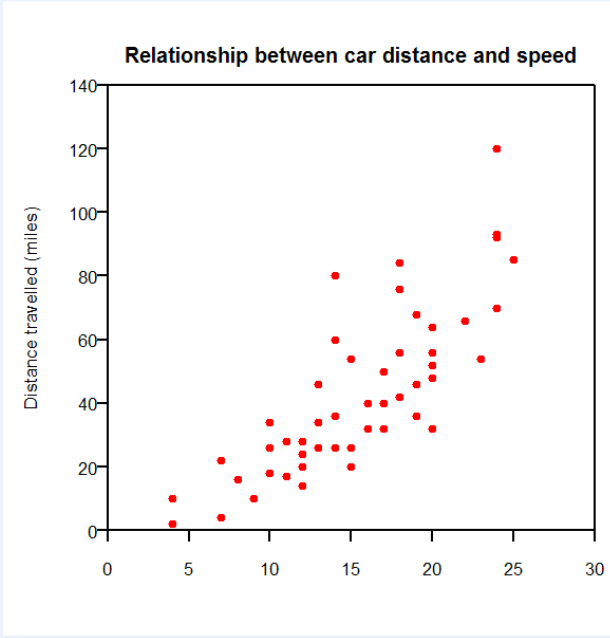


提高精度

```
png("cars.png",res=200,height=600,width=600)
par(mar=c(4,4,3,1),omi=c(0.1,0.1,0.1,0.1),mgp=c(3,0.5,0),
las=1,mex=0.5,cex.main=0.6,cex.lab=0.5,cex.axis=0.5)
plot(cars$dist~cars$speed,
main="Relationship between car distance and speed",
xlab="Speed (miles per hour)",ylab="Distance travelled (miles)",
xlim=c(0,30),ylim=c(0,140),
xaxs="i",yaxs="i",
col="red",pch=19,cex=0.5)
dev.off()
```

结果

名称	大小	类型	修改日期
cars.png	6 KB	光影魔术手PNG图像	2012-12-16 12:46
dailysales.csv	1 KB	Microsoft Office Exc...	2010-4-9 13:58
themes.csv	1 KB	Microsoft Office Exc...	2010-11-29 11:07

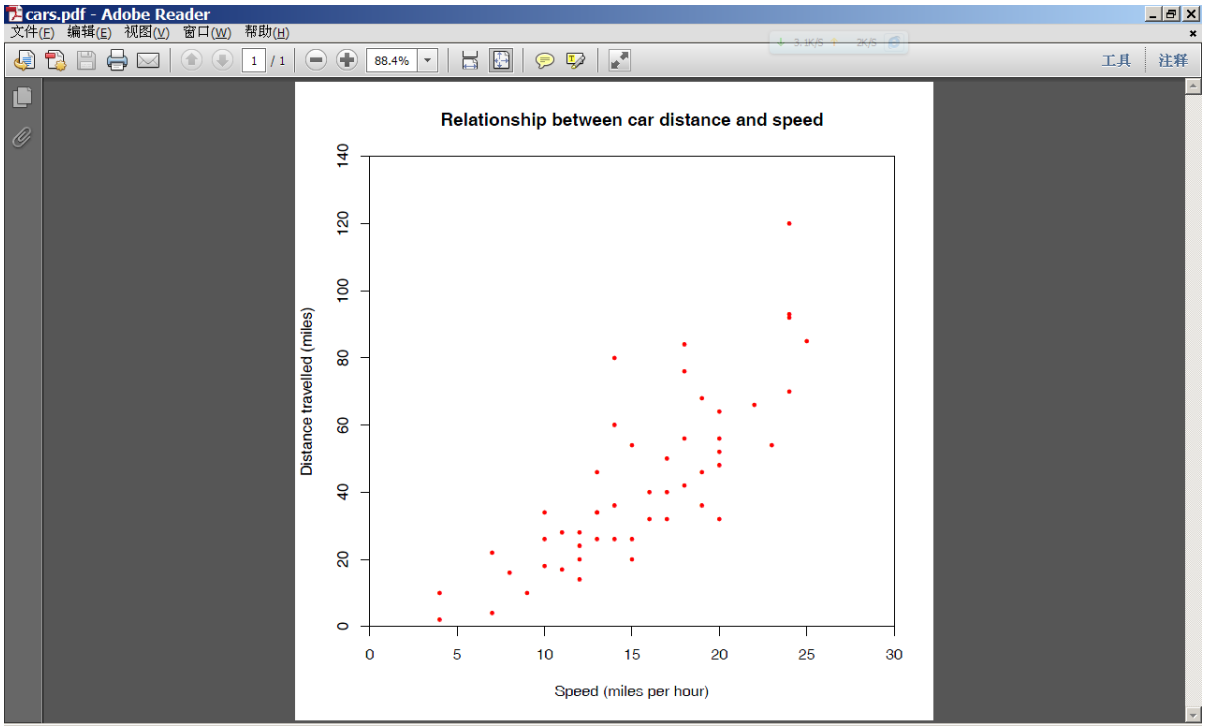


保存矢量格式到pdf文件

```
pdf("cars.pdf")
plot(cars$dist~cars$speed,
main="Relationship between car distance and speed",
xlab="Speed (miles per hour)",ylab="Distance travelled (miles)",
xlim=c(0,30),ylim=c(0,140),
xaxs="i",yaxs="i",
col="red",pch=19,cex=0.5)
dev.off()
```

结果

名称	大小	类型	修改日期
cars.png	6 KB	光影魔术手PNG图像	2012-12-16 12:46
dailysales.csv	1 KB	Microsoft Office Exc...	2010-4-9 13:58
themes.csv	1 KB	Microsoft Office Exc...	2010-11-29 11:07
cars.pdf	5 KB	Adobe Acrobat Docu...	2012-12-16 12:49



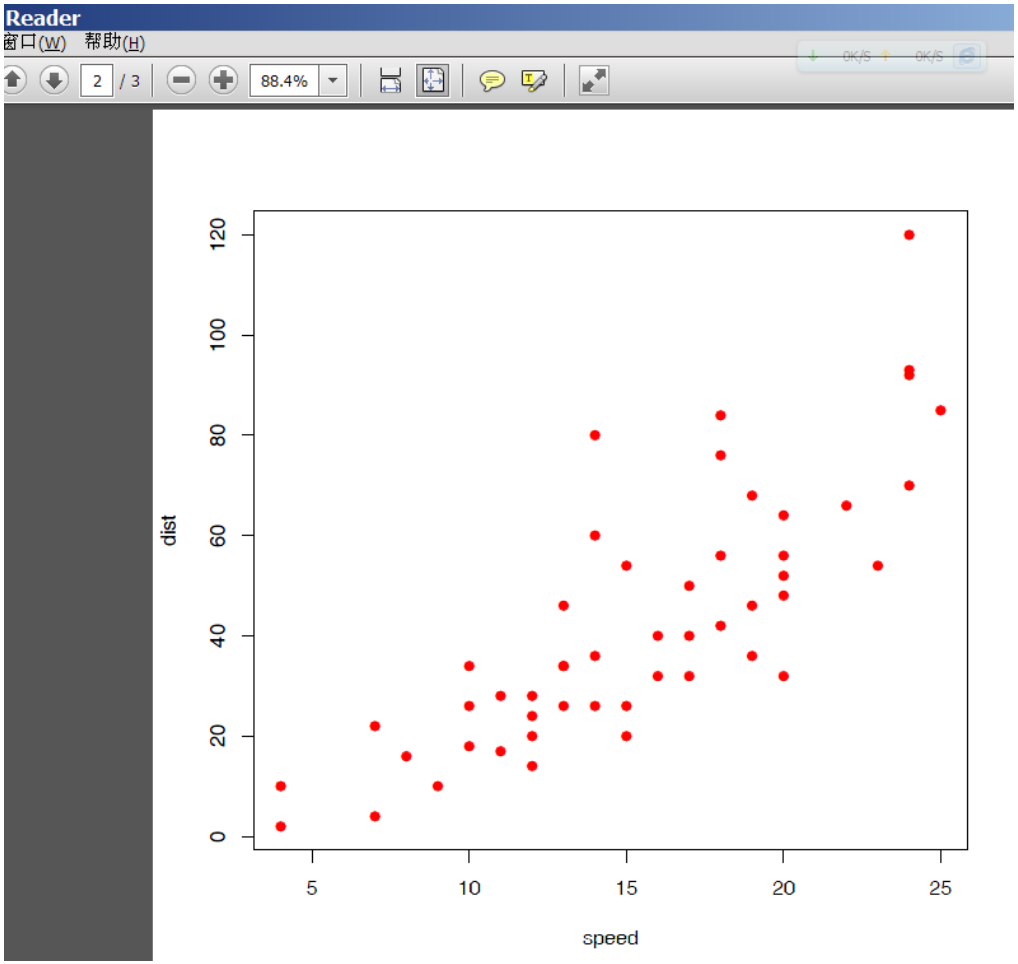
2013.05.11

svg文件，ps文件

```
svg("3067_10_03.svg")
#plot command here
dev.off()
postscript("3067_10_03.ps")
#plot command here
dev.off()
```

在一个pdf文件中输出多张图

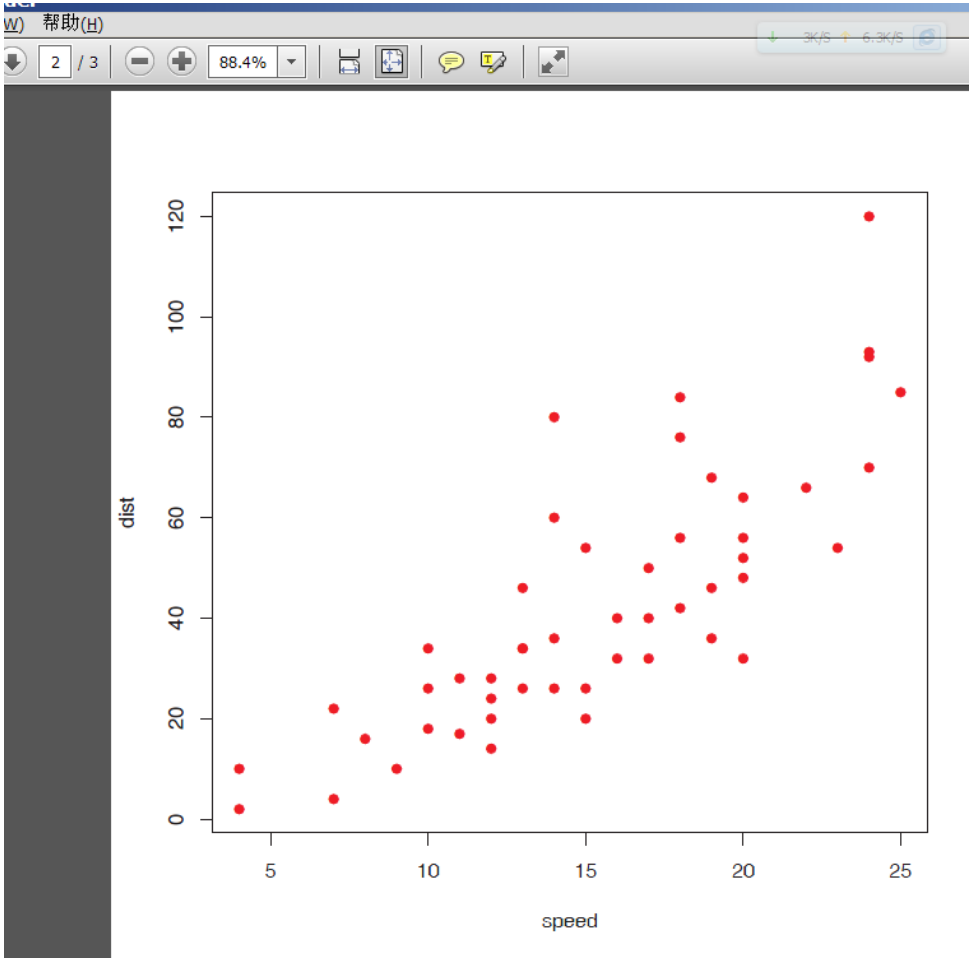
```
pdf("multiple.pdf")
for(i in 1:3)
plot(cars,pch=19,col=i)
dev.off()
```



2013.05.11

改变色彩模式

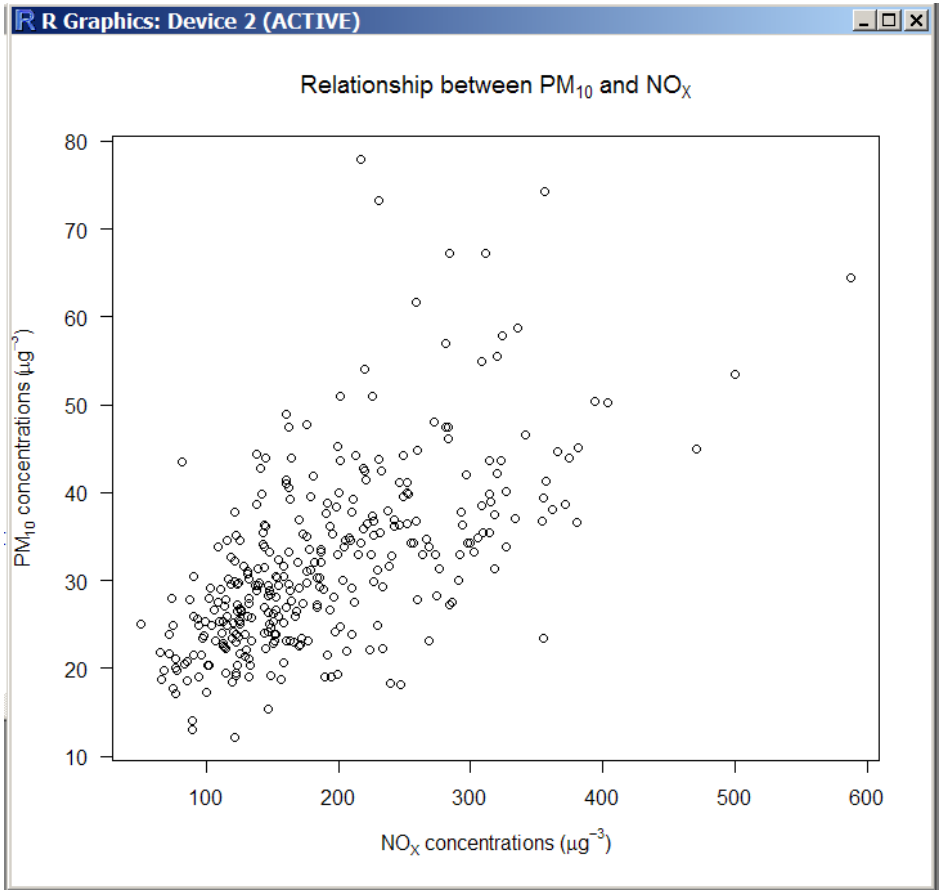
```
pdf("multiple.pdf",colormod
    el="cmyk")
for(i in 1:3)
plot(cars,pch=19,col=i)
dev.off()
```



在输出中表现数学公式

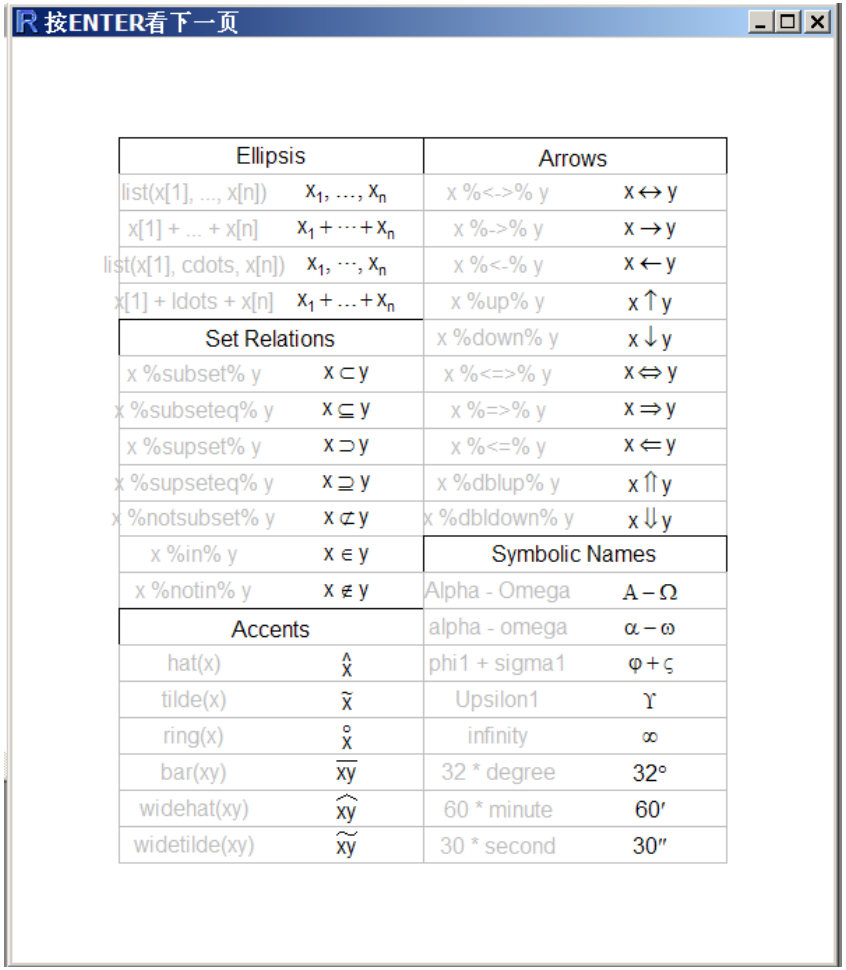
使用第七章数据

```
air<-read.csv("airpollution.csv")
plot(air,las=1,
main=expression(paste("Relations
    hip between ",PM[10]," and
    ",NO[X])),
xlab=expression(paste(NO[X],"
    concentrations (",mu*g^-
    3,")")),
ylab=expression(paste(PM[10],"
    concentrations (",mu*g^-
    3,")")))
```



公式表达

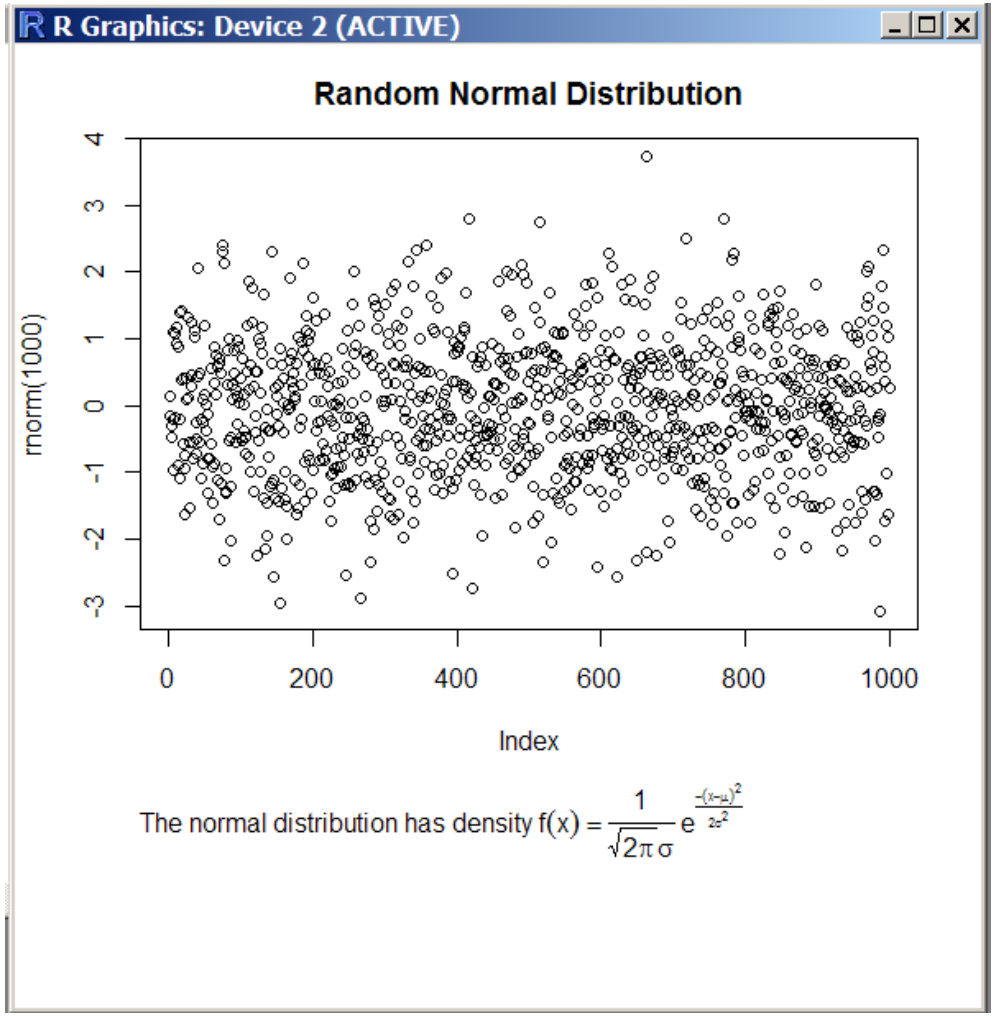
demo(plotmath)



例子

```
par(mar=c(12,4,3,2))
plot(rnorm(1000),main="Random Normal Distribution")
desc<-expression(paste("The normal distribution has density ",
f(x) == frac(1,sqrt(2*pi)*sigma)~ plain(e)^frac(-(x-mu)^2,2*sigma^2)))
mtext(desc,side=1,line=4,padj=1,adj=0)
mtext(expression(paste("where ", mu, " is the mean of the distribution
and ",sigma," the standard deviation.")),
side=1,line=7,padj=1,adj=0)
```

结果


R Graphics: Device 2 (ACTIVE)
Random Normal Distribution
rnorm(1000)
4
3
2
1
0
-1
-2
-3
0
200
400
600
800
1000
Index
The normal distribution has density $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$

2013.05.11

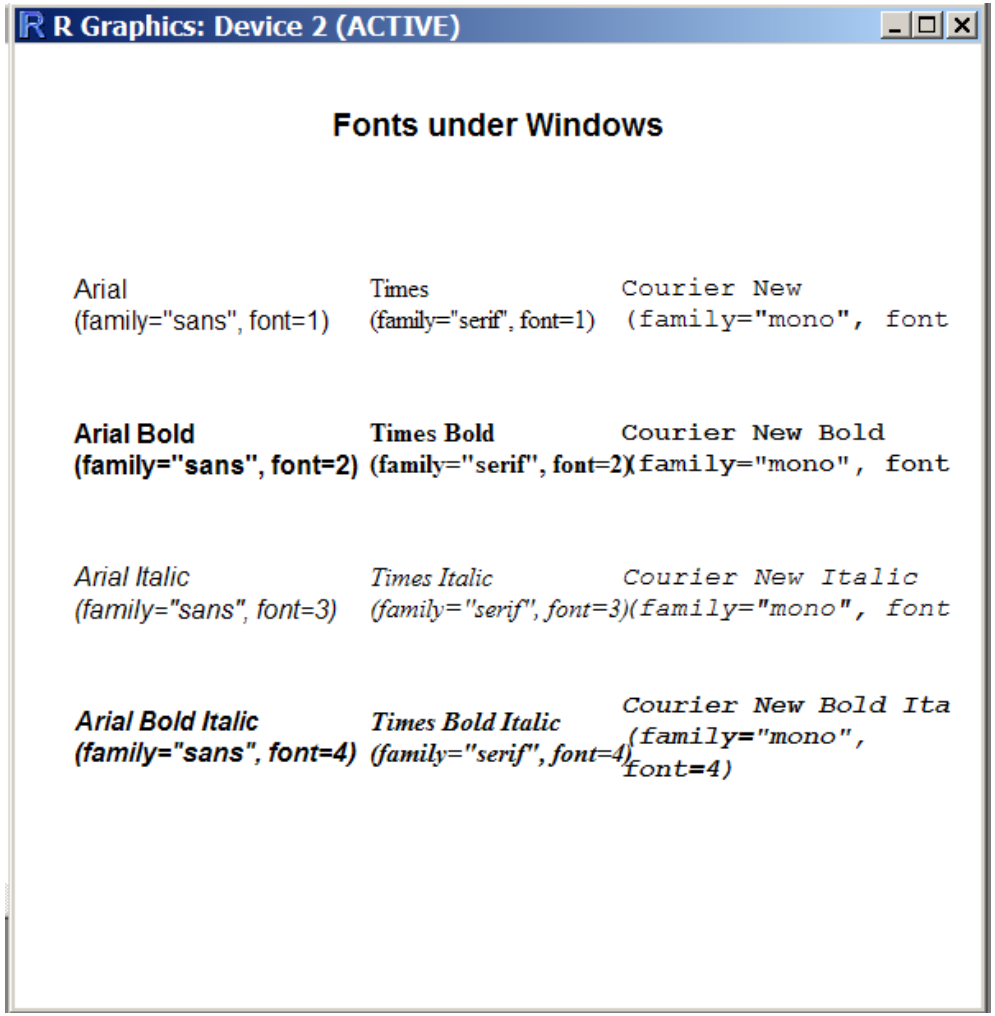
使用不同的字体

```
par(mar=c(1,1,5,1))
plot(1:200,type="n",main="Fonts under Windows",axes=FALSE,xlab="",
ylab="")
text(0,180,"Arial \n(family=\"sans\", font=1)",
family="sans",font=1,adj=0)
text(0,140,"Arial Bold \n(family=\"sans\", font=2)",
family="sans",font=2,adj=0)
text(0,100,"Arial Italic \n(family=\"sans\", font=3)",
family="sans",font=3,adj=0)
text(0,60,"Arial Bold Italic \n(family=\"sans\", font=4)",
family="sans",font=4,adj=0)
text(70,180,"Times \n(family=\"serif\", font=1)",
family="serif",font=1,adj=0)
text(70,140,"Times Bold \n(family=\"serif\", font=2)",
```

代码（续）

```
family="serif",font=2,adj=0)
text(70,100,"Times Italic \n(family=\"serif\", font=3)",
family="serif",font=3,adj=0)
text(70,60,"Times Bold Italic \n(family=\"serif\", font=4)",
family="serif",font=4,adj=0)
text(130,180,"Courier New\n(family=\"mono\", font=1)",
family="mono",font=1,adj=0)
text(130,140,"Courier New Bold \n(family=\"mono\", font=2)",
family="mono",font=2,adj=0)
text(130,100,"Courier New Italic \n(family=\"mono\", font=3)",
family="mono",font=3,adj=0)
text(130,60,"Courier New Bold Italic \n(family=\"mono\",
font=4)",
family="mono",font=4,adj=0)
```

效果


R Graphics: Device 2 (ACTIVE)
Fonts under Windows
Arial
(family="sans", font=1)
Times
(family="serif", font=1)
Courier New
(family="mono", font
Arial Bold
(family="sans", font=2)
Times Bold
(family="serif", font=2)
Courier New Bold
(family="mono", font
Arial Italic
(family="sans", font=3)
Times Italic
(family="serif", font=3)
Courier New Italic
(family="mono", font
Arial Bold Italic
(family="sans", font=4)
Times Bold Italic
(family="serif", font=4)
Courier New Bold Ita
(family="mono",
font=4)

画联系图的例子

2013.05.11

代码

```
library(maps)
library(geosphere)

xlim <- c(-171.738281, -56.601563)
ylim <- c(12.039321, 71.856229)
map("world", col="#f2f2f2", fill=TRUE, bg="white", lwd=0.05, xlim=xlim, ylim=ylim)

airports <- read.csv("http://datasets.flowingdata.com/tuts/maparcs/airports.csv", header=TRUE)
flights <- read.csv("http://datasets.flowingdata.com/tuts/maparcs/flights.csv", header=TRUE, as.is=TRUE)

map("world", col="#f2f2f2", fill=TRUE, bg="white", lwd=0.05, xlim=xlim, ylim=ylim)
fsub <- flights[flights$airline == "AA",]
for (j in 1:length(fsub$airline)) {
    air1 <- airports[airports$iata == fsub[j,]$airport1,]
    air2 <- airports[airports$iata == fsub[j,]$airport2,]
    inter <- gcIntermediate(c(air1[1,]$long, air1[1,]$lat), c(air2[1,]$long, air2[1,]$lat), n=100, addStartEnd=TRUE)
    lines(inter, col="black", lwd=0.8)
}
```

代码

```
pal <- colorRampPalette(c("#f2f2f2", "black"))
colors <- pal(100)

map("world", col="#f2f2f2", fill=TRUE, bg="white", lwd=0.05, xlim=xlim, ylim=ylim)

fsub <- flights[flights$airline == "AA",]
maxcnt <- max(fsub$cnt)
for (j in 1:length(fsub$airline)) {
    air1 <- airports[airports$iata == fsub[j,]$airport1,]
    air2 <- airports[airports$iata == fsub[j,]$airport2,]

    inter <- gcIntermediate(c(air1[1,]$long, air1[1,]$lat), c(air2[1,]$long, air2[1,]$lat), n=100, addStartEnd=TRUE)
    colindex <- round( (fsub[j,]$cnt / maxcnt) * length(colors) )

    lines(inter, col=colors[colindex], lwd=0.8)
}
```

代码

```
pal <- colorRampPalette(c("#f2f2f2", "black"))
pal <- colorRampPalette(c("#f2f2f2", "red"))
colors <- pal(100)

map("world", col="#f2f2f2", fill=TRUE, bg="white", lwd=0.05, xlim=xlim, ylim=ylim)

fsub <- flights[flights$airline == "AA",]
fsub <- fsub[order(fsub$cnt),]
maxcnt <- max(fsub$cnt)
for (j in 1:length(fsub$airline)) {
    air1 <- airports[airports$iata == fsub[j,]$airport1,]
    air2 <- airports[airports$iata == fsub[j,]$airport2,]

    inter <- gcIntermediate(c(air1[1,]$long, air1[1,]$lat), c(air2[1,]$long, air2[1,]$lat), n=100, addStartEnd=TRUE)
    colindex <- round( (fsub[j,]$cnt / maxcnt) * length(colors) )

    lines(inter, col=colors[colindex], lwd=0.8)
}
```

代码

```
pal <- colorRampPalette(c("#f2f2f2", "red"))



# Unique carriers
carriers <- unique(flights$airline)

# Color
pal <- colorRampPalette(c("#333333", "white", "#1292db"))
colors <- pal(100)
for (i in 1:length(carriers)) {

    pdf(paste("carrier", carriers[i], ".pdf", sep=""), width=11, height=7)
    map("world", col="#191919", fill=TRUE, bg="#000000", lwd=0.05, xlim=xlim, ylim=ylim)
```

代码

```
fsub <- flights[flights$airline == carriers[i],]
        fsub <- fsub[order(fsub$cnt),]
        maxcnt <- max(fsub$cnt)
        for (j in 1:length(fsub$airline)) {
                air1 <- airports[airports$iata == fsub[j,]$airport1,]
                air2 <- airports[airports$iata == fsub[j,]$airport2,]

                inter <- gcIntermediate(c(air1[1,]$long, air1[1,]$lat), c(air2[1,]$long, air2[1,]$lat), n=100,
        addStartEnd=TRUE)
                colindex <- round( (fsub[j,]$cnt / maxcnt) * length(colors) )

                lines(inter, col=colors[colindex], lwd=0.6)
        }

        dev.off()
}
```

社交数据可视化：Facebook好友联系图

http://flowingdata.com/2011/05/11/how-to-map-connections-with-great-circles/

炼数成金逆向收费式网络课程

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 http://edu.dataguru.cn

Thanks

FAQ时间