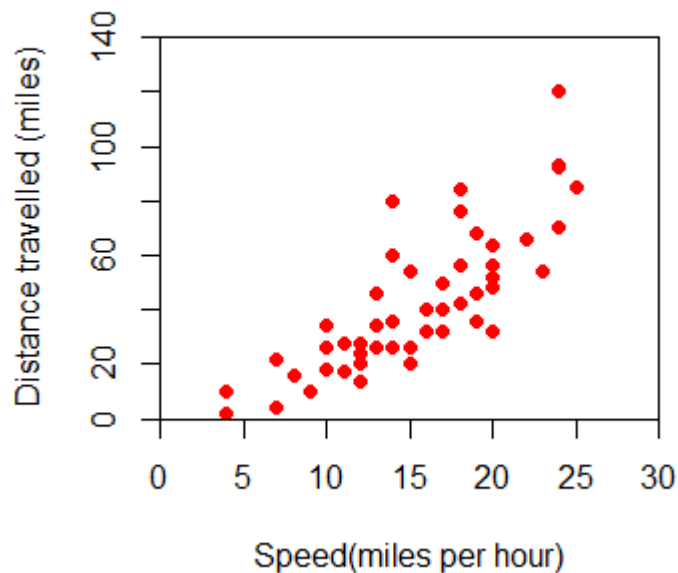


```

plot(cars$dist~cars$speed,#y~x, cars 是 R 自带的数据
main="Relationship between car distance & speed",#标题
xlab = "Speed(miles per hour)",#x 轴标题
ylab = "Distance travelled (miles)",#Y 轴标题
xlim = c(0,30),#设置 x 轴的取值区间为 0 到 30
ylim = c(0,140),#设置 y 轴的取值区间为 0 到 140
xaxs = "i",#这里是设置 x 轴的风格，暂时没看明白有多大区别
yaxs = "i",
col = "red",#设置颜色
pch = 19)#pch 指代点的形状，用数字表示，可查看帮助文档

```

Relationship between car distance & sp



#如果要保存图片怎么办呢？我觉得最简单的方法就是使用 RStudio 这个 IDE，极其得好，可惜很多人都不知道。#如果你不会，可以用如下代码实现：#（图形的参数还有很多个，我这里只使用了其中的几个）

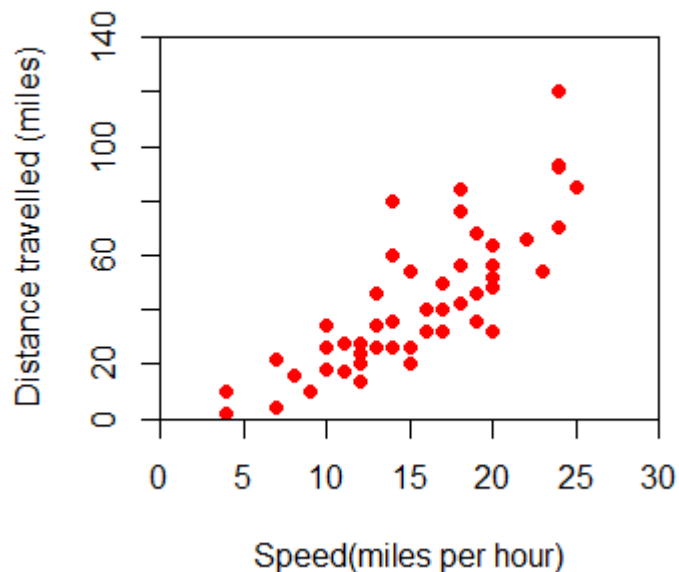
```

png(filename="散点图.png",width=480,height=480)
plot(cars$dist~cars$speed,#y~x
main="Relationship between car distance & speed",#标题
xlab = "Speed(miles per hour)",#x 轴标题
ylab = "Distance travelled (miles)",#Y 轴标题
xlim = c(0,30),#设置 x 轴的取值区间为 0 到 30
ylim = c(0,140),#设置 y 轴的取值区间为 0 到 140

```

```
xaxs = "i",#这里是设置 x 轴的风格，暂时没看明白有多大区别
yaxs = "i",
col = "red",
pch = 19)#pch 指代点的形状，用数字表示，可查看帮助文档
dev.off()
```

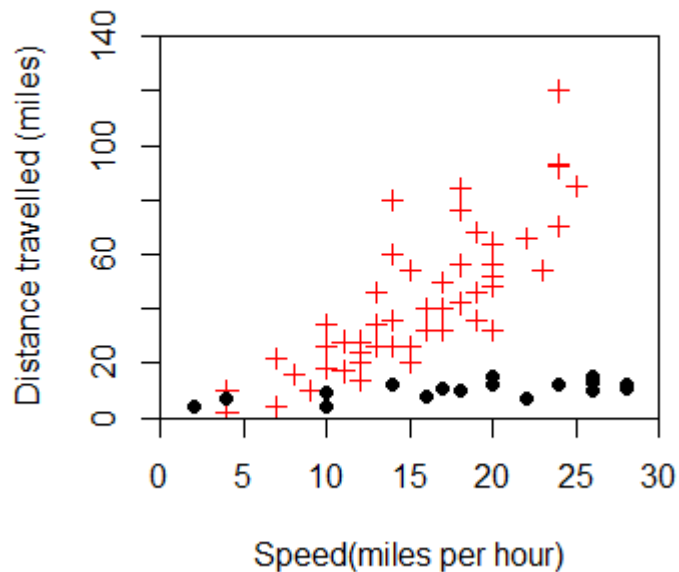
Relationship between car distance & sp



如果你还有另一组数据，需要在上面这幅图里绘制，也就是在同一幅图里面绘制两组数据的散点图，可以像下面这样用 `point` 函数：

```
plot(cars$dist~cars$speed,#y~x
main="Relationship between car distance & speed",#标题
xlab = "Speed(miles per hour)",#x 轴标题
ylab = "Distance travelled (miles)",#Y 轴标题
xlim = c(0,30),#设置 x 轴的取值区间为 0 到 30
ylim = c(0,140),#设置 y 轴的取值区间为 0 到 140
xaxs = "i",#这里是设置 x 轴的风格，暂时没看明白有多大区别
yaxs = "i",
col = "red",
pch = 3)#pch 指代点的形状，用数字表示，可查看帮助文档
points(cars$speed~cars$dist,pch=19)#因为比较难弄数据，就把原先的数据因果关系颠倒一下，
pch 设置与前面不同以区分
```

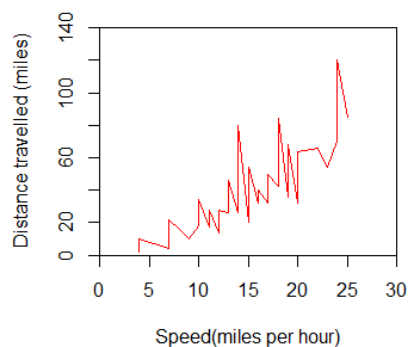
Relationship between car distance & sp



如果自己没数据的话，可以用 `data()` 查看 R 语言内置的数据的哦。还是有挺多数据的。
上面的这些是散点图，但是你只要在参数里面 `type="l"` 是字母 l，那么就会将这些点串联起来画成线了。

```
plot(cars$dist~cars$speed,#y~x
main="Relationship between car distance & speed",#标题
xlab = "Speed(miles per hour)",#x 轴标题
ylab = "Distance travelled (miles)",#Y 轴标题
xlim = c(0,30),#设置 x 轴的取值区间为 0 到 30
ylim = c(0,140),#设置 y 轴的取值区间为 0 到 140
xaxs = "i",#这里是设置 x 轴的风格，暂时没看明白有多大区别
yaxs = "i",
col = "red", type="l",#是字母 l，那么就会将这些点串联起来画成线了
pch = 19)#pch 指代点的形状，用数字表示，可查看帮助文档
```

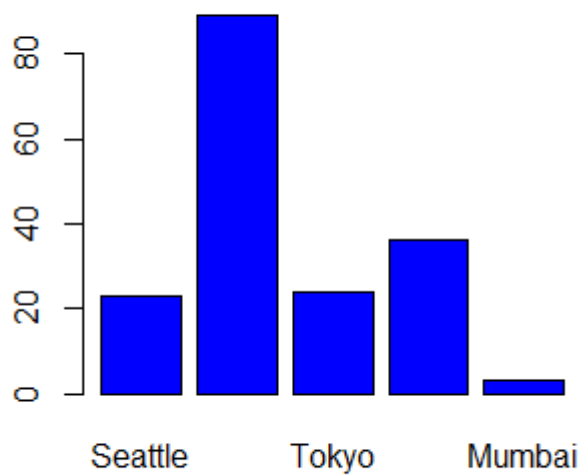
Relationship between car distance & sp



条形图的绘制方法（bar plot），数据是该书自带的

`Sales <- read.csv("citysales.csv",header=TRUE)` #header 设置为 TRUE 表示把数据行和列的名称也读取进来

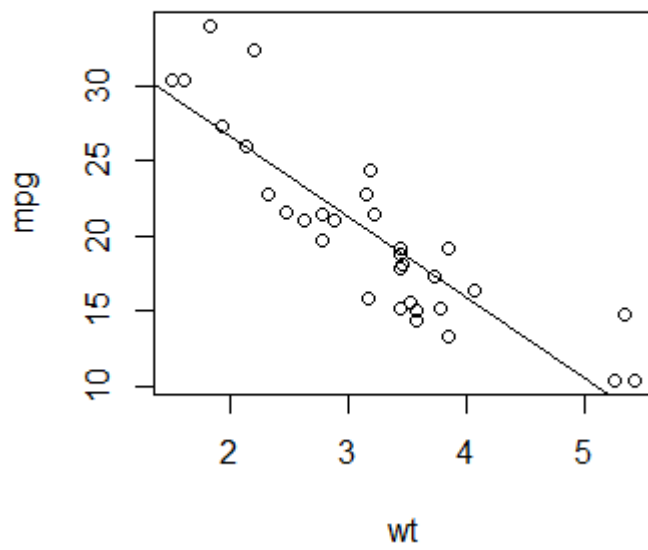
```
barplot(Sales$ProductA,  
names.arg=Sales$City,  
col="blue")
```



用代码保存图形，将绘图语句写在 开启目标图形设备的语句 和 关闭目标图形设备的语句 之间即可。

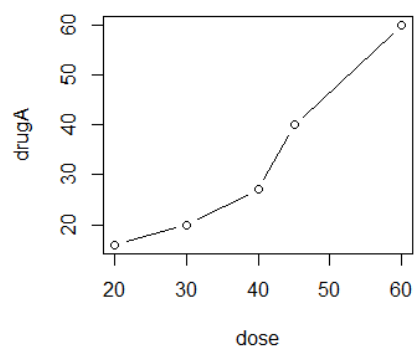
```
#pdf("mygraph.pdf")#pdf 文件  
win.metafile("filename.wmf")#windows 图形文件  
#png("filename.png")#PBG 文件  
#jpeg("filename.jpg")#JPEG 文件  
#bmp("filename.bmp")#BMP 文件  
#postscript("filename.ps")#PostScript 文件  
attach(mtcars)  
plot(wt,mpg)          #散点图  
abline(lm(mpg~wt))   #加直线  
title("Regression of MPG on Weight")  
detach(mtcars)  
dev.off()
```

Regression of MPG on Weight



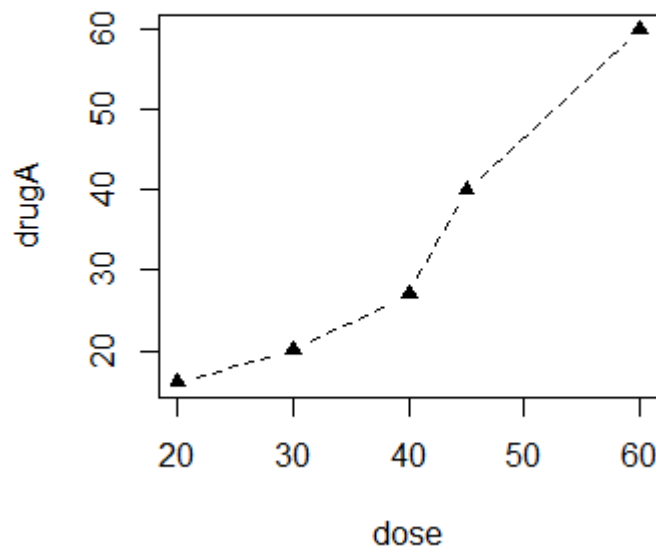
```
dose=c(20,30,40,45,60)
drugA=c(16,20,27,40,60)
drugB=c(15,18,25,31,40)
plot(dose,drugA,type="b")
```

#type="b" 同时画出点和线
#type= "c" 没有点只有线



在上例中，我们想使用实心三角而不是空心圆圈作为点符号的话，并想用虚线代替实现连接这些点：

```
dose=c(20,30,40,45,60)
drugA=c(16,20,27,40,60)
drugB=c(15,18,25,31,40)
plot(dose,drugA,type="b")
opar = par(no.readonly=TRUE) #复制一份单签的图形参数
par(lty=2,pch=17) #将默认的线性类型修改为虚线（lty=2）并且将默认的点符号改为了实心三角（pch=17）
#也可以使用 par(lty=2);par(pch=17)两句
plot(dose,drugA,type="b")#绘制了图形
par(opar)#还原了原始设置
#或者这样写 plot(dose,drugA,type="b",lty=2,pch=17)来画图，但只是针对于这张图
```



pch :指定绘制点时使用的符号

cex: 指定符号的大小。**cex** 是一个数值，表示绘图符号相对于默认值大小的缩放倍数。默认大小为 1,1.5 表示放大为默认值的 1.5 倍.

lty: 指定线条类型

lwd: 指定线条宽度。（默认值的几倍）

col: 默认的绘图颜色。如这是 `col=c("red","blue")`并需要绘制三条线，第一条为红色，第二条为蓝色，第三条为红色

col.axis 坐标轴颜色

col.main 标题颜色

col.sub 副标题的颜色

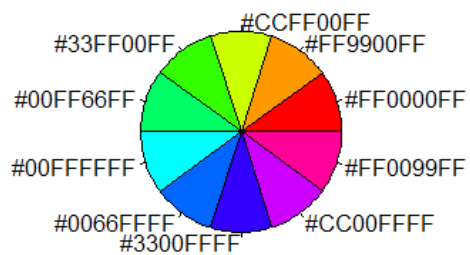
fg 前景色

bg 背景色

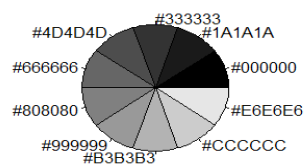
example: `col=1,col="white" col="#FFFFFF" col=rgb(1,1,1) col=hsv(0,1,1)` 都可以表示白色
R 中也用多种用于创建连续型颜色向量的函数:

`rainbow()`
`heat.colors()`
`terrain.colors()`
`top.colors()`
`cm.colors()`
`gray()`可以生成多节灰度

`n=10`
`mycolors=rainbow(n)`
`pie(rep(1,n),labels=mycolors,col=mycolors)`



`mygrays=gray(0:n/n)`
`pie(rep(1,n),labels=mygrays,col=mygrays)`



6.文本属性

cex: 表示相对默认大小缩放倍数的数值。(倍数)

cex.axis: 坐标轴刻度文字的缩放倍数。

cex.lab: 坐标轴标签(名称)的缩放倍数。

cex.main: 标题的缩放倍数

cex.sub: 副标题的缩放倍数

font: 整数类型。1=常规, 2=粗体, 3=斜体, 4=粗斜体, 5=符号字体(adobe 编码)

font.axis font.lab font.main font.sub

ps 磅值文本最终的大小为 $ps * cex$

family 绘制文本是使用的字体族。标准取值为 **serif** (衬线)、**sans** (无衬线) 和 **mono** (等宽)

Windows 中, 可以通过函数 **windowsFont ()** 创造新的映射。(Mac, 采用 **quartzFont ()**)

PDF 或者 PostScript 输出格式图形, 修改相对简单。

PDF 使用 **names(pdfFonts())** 找出系统有哪些字体可用, 然后用 **pdf(file= "myplot.pdf",family="fontname")** 生成图形。

PostScript 输出格式的图形, 可以使用 **names (postscriptFonts())** 和 **postscript(file="myplot.ps",family="fontname")**

7.图形尺寸和边界尺寸

pin: 以英寸表示的图形尺寸(宽和高)

mai: 以数值向量表示的边界大小, 顺序为“下、左、上、右”单位为英寸

mar: 以数值向量表示的边界大小, 顺序为“下、左、上、右”单位为英分。默认=c(5,4,4,2)+0.1

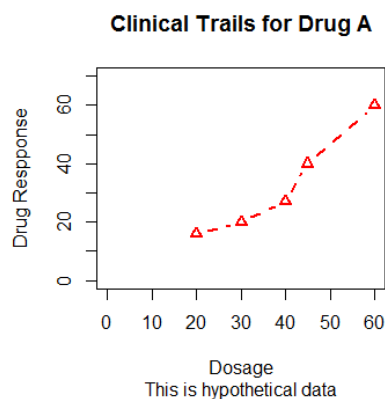
8.添加文本、自定义坐标和图例

dose=c(20,30,40,45,60)

drugA=c(16,20,27,40,60)

drugB=c(15,18,25,31,40)

plot(dose,drugA,type="b",col="red",lty=2,pch=2,lwd=2,main="Clinical Trails for Drug A",sub="This is hypothetical data",xlab="Dosage",ylab="Drug Response",xlim=c(0,60),ylim=c(0,70))



某些高级绘图函数已经包含了默认的标题和标签。可以通过 `plot()` 语句或单独的 `par()` 语句中添加 `ann=FALSE` 来移除他们

标题

可以使用 `title()` 函数为图形添加标题和坐标轴标签。

坐标轴

side: 一个整数，表示图形的那边会画坐标（1,2,3,4 对应下、左、上、右）

at: 一个数值型向量，表示需要绘制刻度线的位置

labels: 一个字符型向量，表示至于刻度线旁边的文字表全(如果是 `NULL`，直接使用 `at` 中的值)

pos: 坐标轴线位置的坐标

lty: 线条类型

col: 线条的刻度线颜色

las: 标签是否平行于（=0）或垂直于（=2）坐标轴

tck: 刻度线的长度，以向对于绘图区域大小的分数表示（负数表示在图形外侧，整数表示在图形内侧）

`Hmisc` 包中的 `minor.tick()` 函数 用来创建次要刻度线。

tick.ratio 表示次要刻度线相对于主刻度线的大小比例。当前主刻度线长度可以用 `par("tck")` 获取。

参考线

函数 `abline()` 可以用来为图形添加参考线。`abline(h=yvalues,v=xvalues)`

example:`abline(v=seq(1,10,2),lty=2,col="blue")`

图例

用 `legend(location,title,legend,...)` 添加图例

location: 可以直接给定 `xy` 值；**location**（1）通过鼠标单击给出图例的位置；关键字:`bottom`、`bottomleft`、`left`、`topleft`、`topright`、`right`、`bottomright`、`center`，同时使用参数 `inset`=指定图形想图形内侧移动的大小（以绘图大小的分数表示）

title: 图例标题的字符串（选）

legend: 图例标签组成的字符型向量

文本标注

`text(location,"",pos...)`

`mtext("",side,line=n...)`

location: 可以直接给定 `xy` 值；**location**（1）通过鼠标单击给出图例的位置

pose: 整数，文本相对位置的方向参数。如果指定参数 `offset=,,` 作为偏移量，以相对于单个字符宽度的比例表示

side: 整数，指定用来放置文本的边。

`par()` 增大字号

`plotmath()` 数学标注

Mileage vs Car Weight

This scatter plot illustrates the relationship between car weight and mileage. The x-axis represents 'Weight' (ranging from 2 to 5) and the y-axis represents 'Mileage' (ranging from 10 to 35). The data points are labeled with car models. A blue regression line indicates a negative correlation, suggesting that as car weight increases, the mileage tends to decrease.

Car Model	Weight (approx.)	Mileage (approx.)
Toyota Corolla	1.8	34
Fiat 128	2.3	33
Lotus Elise	1.5	30
Honda Civic	1.7	30
Fiat X1-9	1.9	27
Porsche 914-2	2.2	26
Datsun 710	2.5	23
Toyota Corolla	2.6	21
Volvo 142G	2.7	21
1988 Ford Mustang	2.8	21
1988 Ford Taurus	2.9	21
Ferrari Dino	2.8	19
1988 Ford Taurus	3.1	21
1988 Ford Taurus	3.2	21
1988 Ford Taurus	3.3	21
1988 Ford Taurus	3.4	21
1988 Ford Taurus	3.5	21
1988 Ford Taurus	3.6	21
1988 Ford Taurus	3.7	21
1988 Ford Taurus	3.8	21
1988 Ford Taurus	3.9	21
1988 Ford Taurus	4.0	21
1988 Ford Taurus	4.1	21
1988 Ford Taurus	4.2	21
1988 Ford Taurus	4.3	21
1988 Ford Taurus	4.4	21
1988 Ford Taurus	4.5	21
1988 Ford Taurus	4.6	21
1988 Ford Taurus	4.7	21
1988 Ford Taurus	4.8	21
1988 Ford Taurus	4.9	21
1988 Ford Taurus	5.0	21
1988 Ford Taurus	5.1	21
1988 Ford Taurus	5.2	21
1988 Ford Taurus	5.3	21
1988 Ford Taurus	5.4	21
1988 Ford Taurus	5.5	21
1988 Ford Taurus	5.6	21
1988 Ford Taurus	5.7	21
1988 Ford Taurus	5.8	21
1988 Ford Taurus	5.9	21
1988 Ford Taurus	6.0	21
1988 Ford Taurus	6.1	21
1988 Ford Taurus	6.2	21
1988 Ford Taurus	6.3	21
1988 Ford Taurus	6.4	21
1988 Ford Taurus	6.5	21
1988 Ford Taurus	6.6	21
1988 Ford Taurus	6.7	21
1988 Ford Taurus	6.8	21
1988 Ford Taurus	6.9	21
1988 Ford Taurus	7.0	21
1988 Ford Taurus	7.1	21
1988 Ford Taurus	7.2	21
1988 Ford Taurus	7.3	21
1988 Ford Taurus	7.4	21
1988 Ford Taurus	7.5	21
1988 Ford Taurus	7.6	21
1988 Ford Taurus	7.7	21
1988 Ford Taurus	7.8	21
1988 Ford Taurus	7.9	21
1988 Ford Taurus	8.0	21
1988 Ford Taurus	8.1	21
1988 Ford Taurus	8.2	21
1988 Ford Taurus	8.3	21
1988 Ford Taurus	8.4	21
1988 Ford Taurus	8.5	21
1988 Ford Taurus	8.6	21
1988 Ford Taurus	8.7	21
1988 Ford Taurus	8.8	21
1988 Ford Taurus	8.9	21
1988 Ford Taurus	9.0	21
1988 Ford Taurus	9.1	21
1988 Ford Taurus	9.2	21
1988 Ford Taurus	9.3	21
1988 Ford Taurus	9.4	21
1988 Ford Taurus	9.5	21
1988 Ford Taurus	9.6	21
1988 Ford Taurus	9.7	21
1988 Ford Taurus	9.8	21
1988 Ford Taurus	9.9	21
1988 Ford Taurus	10.0	21

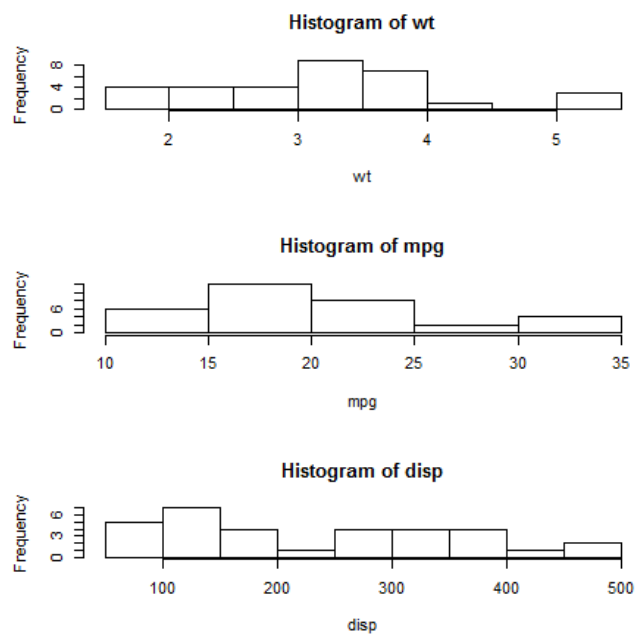
A 7x7 grid with text samples. The vertical axis is labeled 1:7 and the horizontal axis is labeled 1:7. The text samples are:

- Sample of serif text
- Sample of mono-spaced text
- Sample of default text

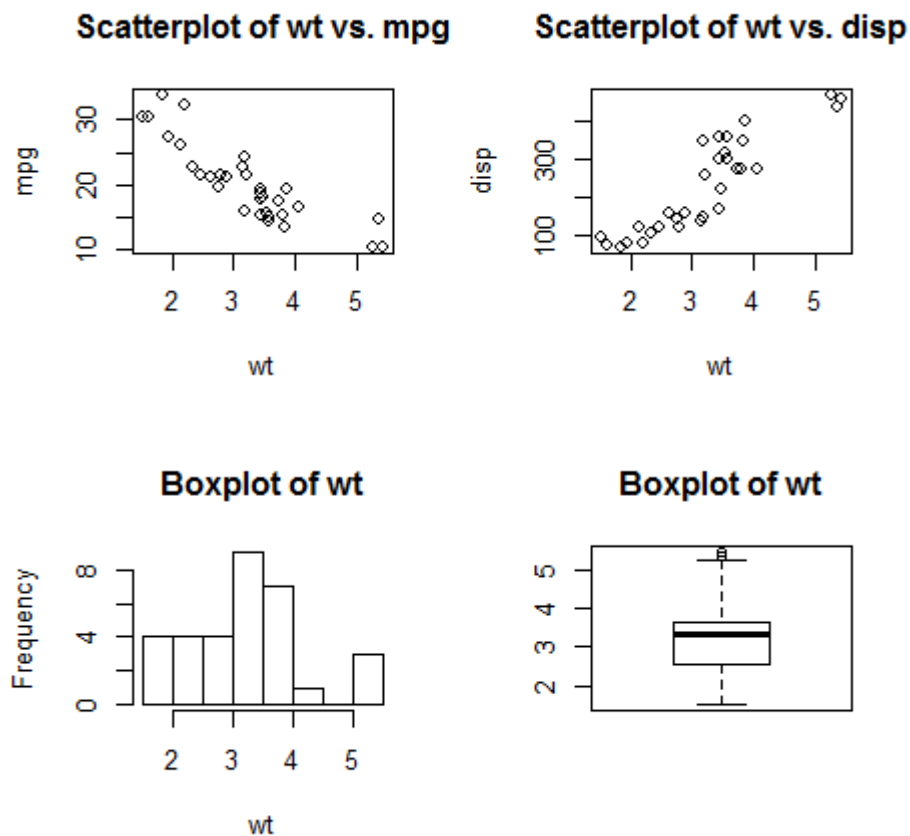
图形组合

在 R 中使用函数 `par()` 或 `layout()` 可以容易地组合多幅图形为一幅图形。`par()` 函数中使用图形参数 `mfrow=c(nrows,ncols)` 来穿件按行填充的、行数为 `nrows`、列数为 `ncols` 的图形矩阵。另外，可以使用 `nfclos=c(nrows,ncols)` 按列填充矩阵。

```
attach(mtcars)
opar=par(no.readonly=TRUE)
par(mfrow=c(3,1))
hist(wt)
hist(mpg)
hist(dis)
par(opar)
detach(mtcars)
```



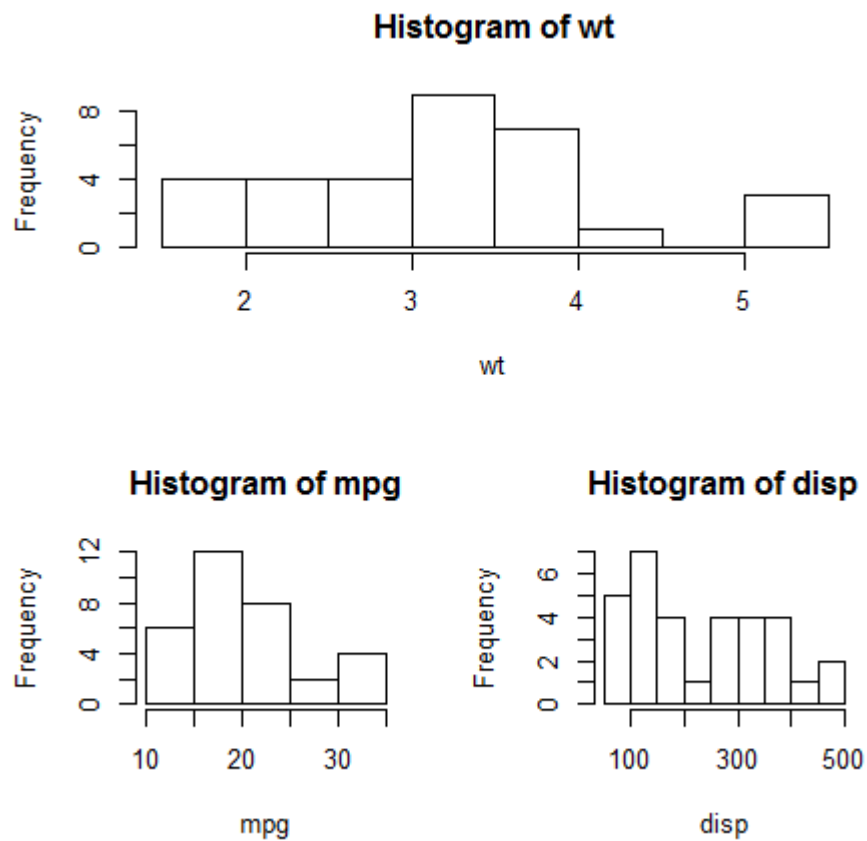
```
attach(mtcars)
opar= par(no.readonly=TRUE)
par(mfrow=c(2,2))
plot(wt,mpg,main="Scatterplot of wt vs. mpg")
plot(wt,disp,main="Scatterplot of wt vs. disp")
hist(wt,main="Boxplot of wt")
boxplot(wt,main="Boxplot of wt")
par(opar)
detach(mtcars)
```



函数 `layout()` 的调用形势为 `layout(mat)`，其中的 `mat` 是一个矩阵，它指定了所要组合的多

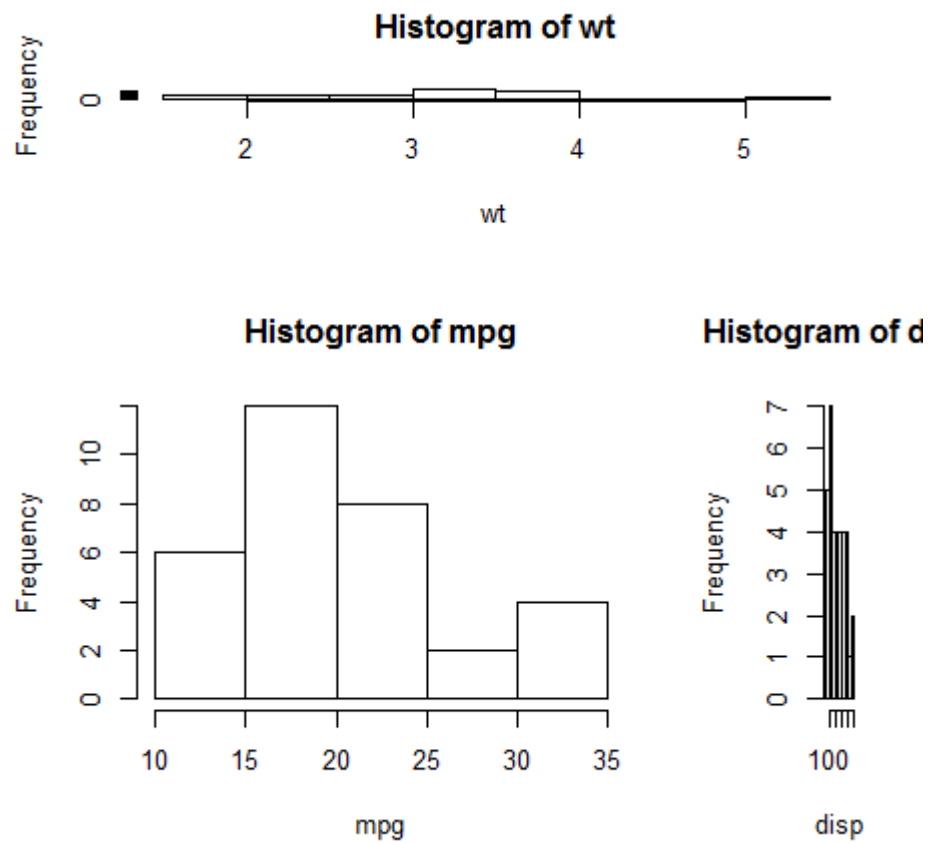
个图形所在位置。

```
attach(mtcars)
layout(matrix(c(1,1,2,3),2,2,byrow=TRUE))
hist(wt)
hist(mpg)
hist(dis)
detach(mtcars)
```



下面代码，将一幅图形放在第一行，两幅图放在第二行，但第一行高度是第二行中图形高度的三分之一，并且，右下角图形宽度是左下角图形宽度的四分之一。

```
attach(mtcars)
layout(matrix(c(1,1,2,3),2,2,byrow=TRUE),widths=c(3,1),heights=c(1,2))
hist(wt)
hist(mpg)
hist(dis)
detach(mtcars)
```



图形布局的精细控制

fig=完成这个任务

基本图形

条形图

简单的条形图

vcd 包

如果要绘制的类别类型是一个 **Factor** 或者是有序性 **Factor**，就可以使用 **plot()**函数快速创建一幅垂直条形图。这时候不用 **table()**函数

（这是关节炎研究，变量 **Improved** 记录了对每位接受了安慰剂或药物治疗的病人的治疗效果）

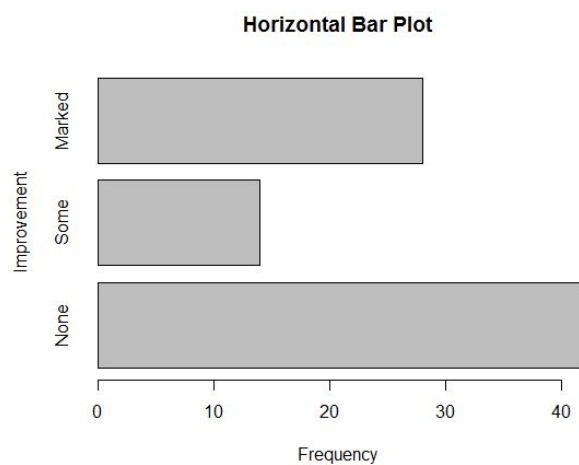
```
library(vcd)
```

```
counts=table(Arthritis$Improved)
```

```
counts
```

```
barplot(counts,main="Simple Bar Plot",xlab="Improvement",ylab="Frequency")
```

```
barplot(counts,main="Horizontal Bar Plot",xlab="Frequency",ylab="Improvement",horiz=TRUE)
```



堆砌条形图和分组条形图

如果 `height` 是一个矩阵而不是一个向量，则绘制结果将是一副堆砌条形图或分组挑衅图。

`besides=false`(默认)->堆砌图 or not 分组条形图

```
counts=table(Arthritis$Improved,Arthritis$Treatment)
```

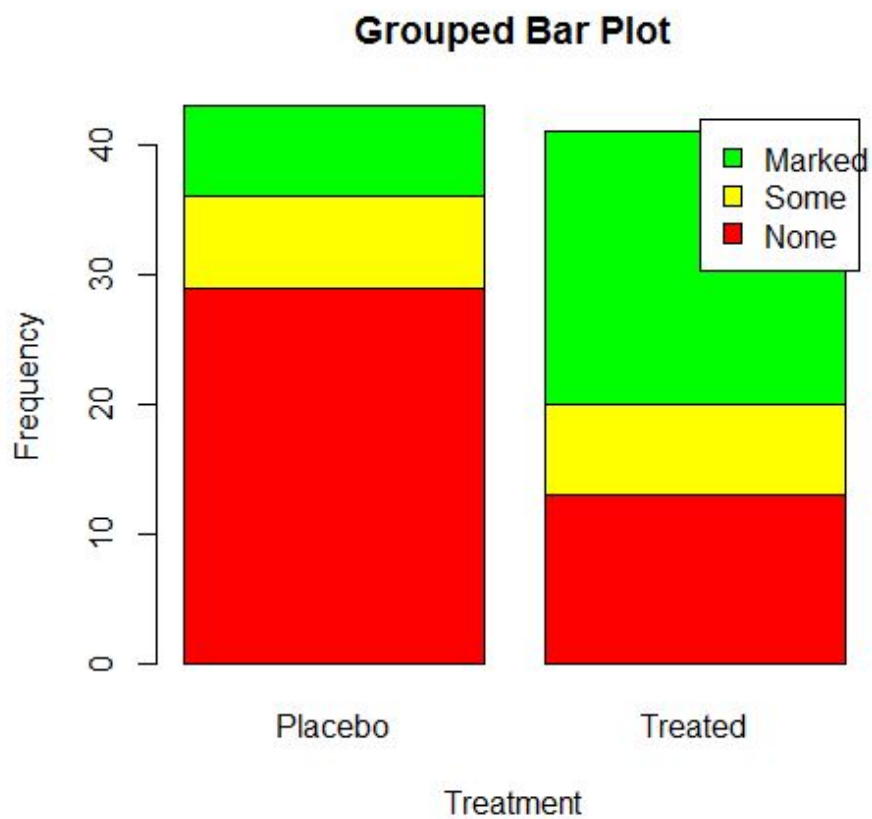
```
counts
```

```
#      Placebo Treated
# None      29      13
# Some       7       7
# Marked     7      21
```

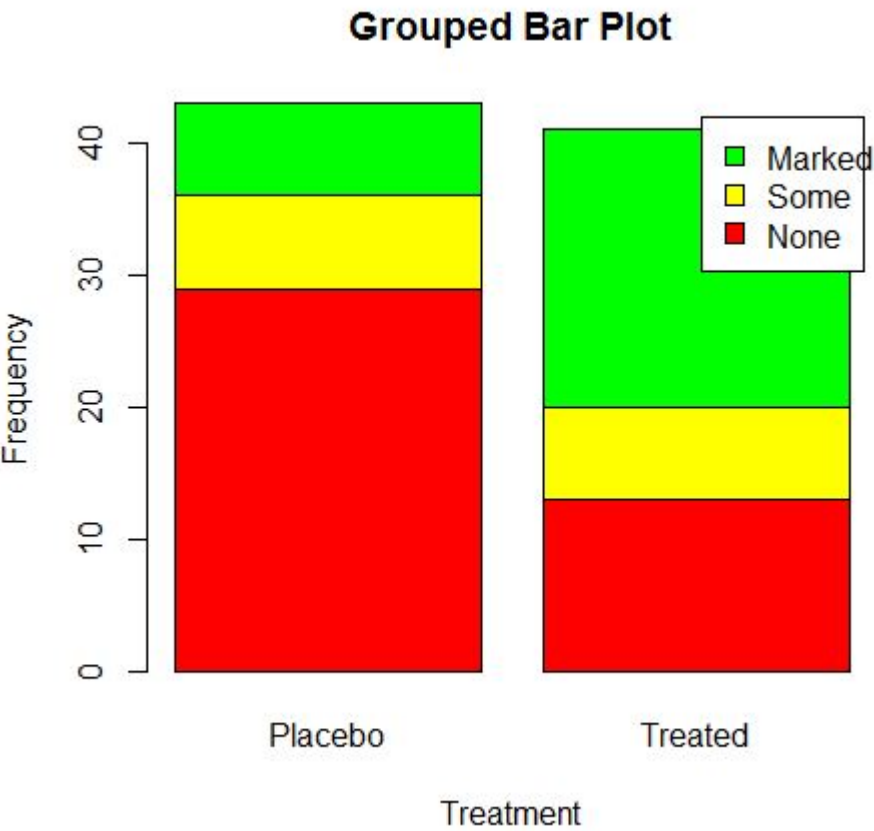
```
barplot(counts,main="Grouped
```

```
Bar
```

```
Plot",xlab="Treatment",ylab="Frequency",col=c("red","yellow","green"),legend=rownames(counts))
```



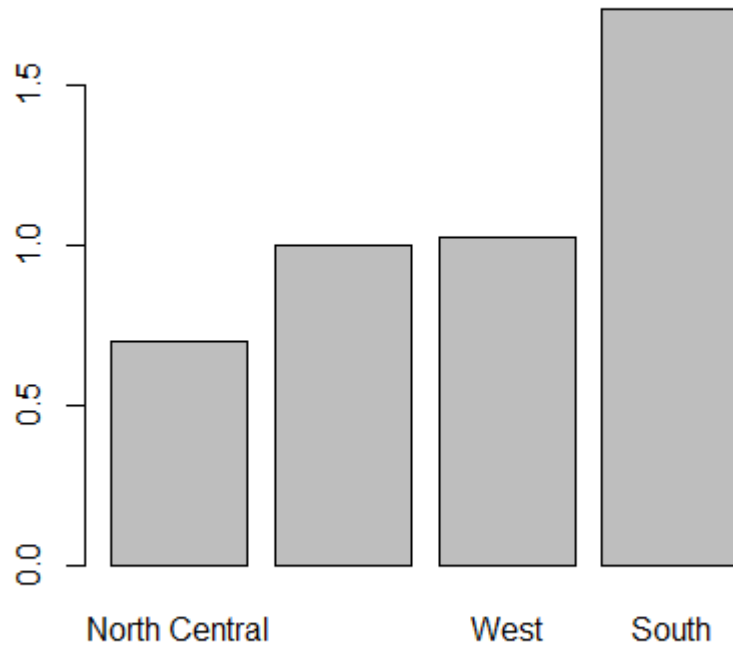

```
barplot(counts,main="Grouped Bar Plot",xlab="Treatment",ylab="Frequency",col=c("red","yellow","green"),legend=rownames(counts),beside=TRUE)
```



均值条形图
条形图并不一定要基于计数数据或者频率数据。（均值，中位数，标准差）

```
states=data.frame(state.region,state.x77)
means=aggregate(states$Illiteracy,by=list(state.region),FUN=mean)
means
#      Group.1      x
#1 Northeast 1.000000
#2      South 1.737500
#3 North Central 0.700000
#4      West 1.023077
means=means[order(means$x),]
means
#      Group.1      x
#3 North Central 0.700000
#1 Northeast 1.000000
#4      West 1.023077
```

```
#2          South 1.737500
barplot(means$x,names.arg=means$Group.1)
title("Mean Illiteracy Rate")
```



中级绘图

主要关注用于展示双变量间关系（二元关系）和多变量间关系（多远关系）的绘图方法。

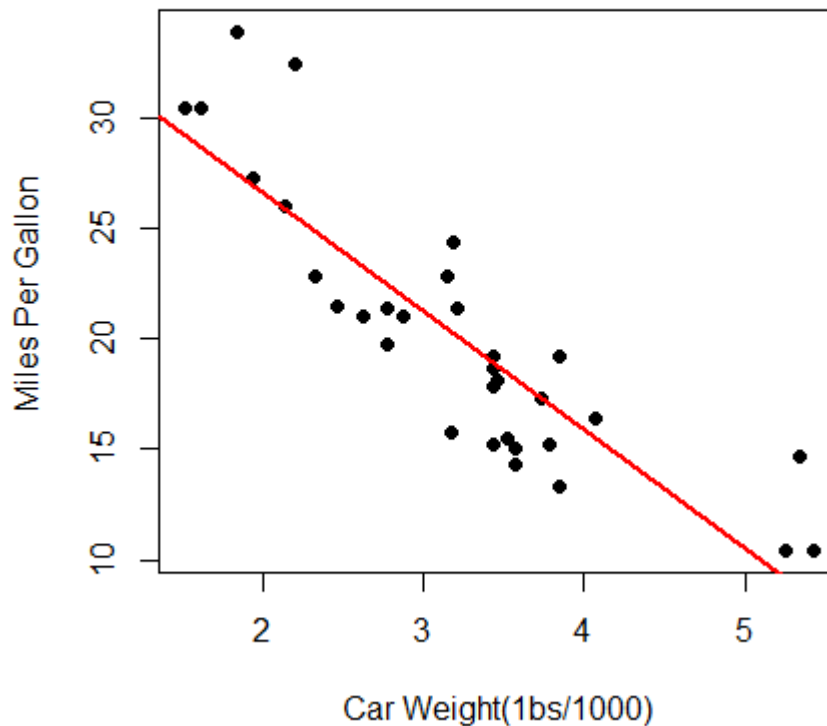
散点图

散点图可以用来描述两个连续变量间的关系。

首先，描述一个二元变量关系，然后探究各种通过添加而外信息来增强图形表达功能的方法。最后，通过添加第三个连续型变量，我们将把二维图形扩展到三位，包括三维散点图和气泡图。

```
attach(mtcars)
plot(wt,mpg,main="Basic Scatter plot of MPG vs Weight",xlab="Car
Weight(1bs/1000)",ylab="Miles Per Gallon ",pch=19)
abline(lm(mpg~wt),col="red",lwd=2,lty=1)
lines(lowess(wt,mpg),col="blue",lwd=2,lty=2)
```

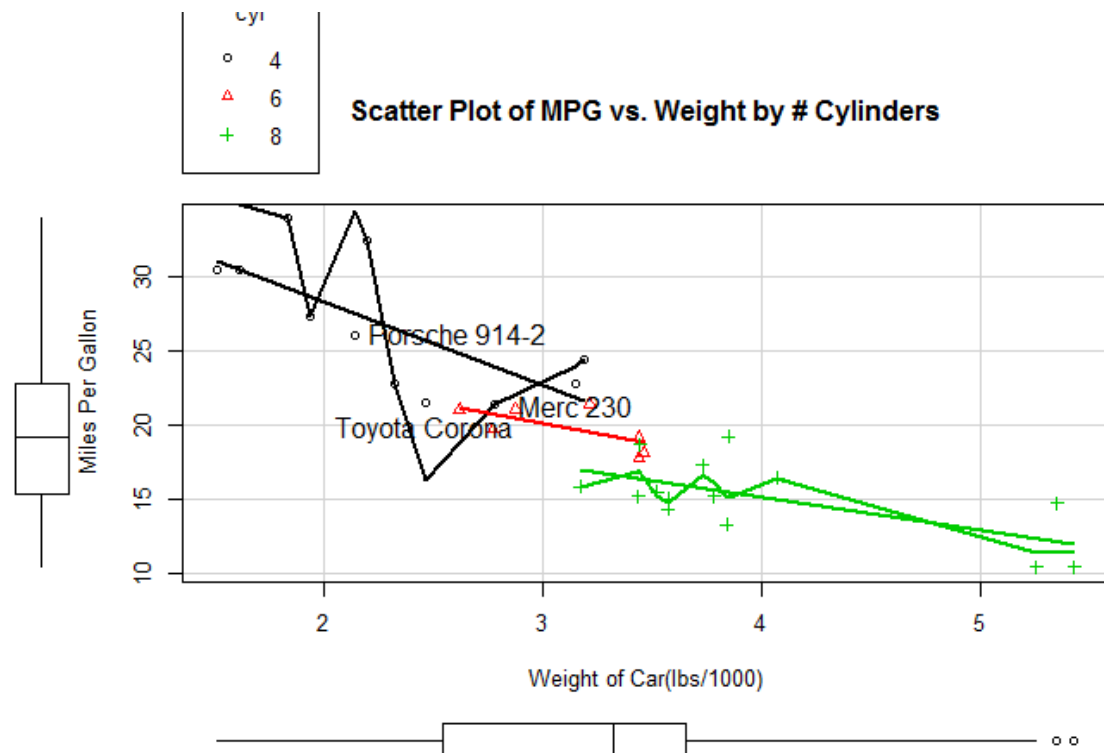
Basic Scatter plot of MPG vs Weight



`abline()`函数用来添加最佳拟合的线性直线。`lowess()`函数用来添加一条平滑曲线。该平滑曲线拟合是一种基于局部加权多项式回归的非参数方法。算法来源于 Cleveland（1981）

`car`包中的 `scatterplot()`函数增强了散点图的需求功能，主要作用在添加拟合曲线，边界箱线图和之心椭圆，还可以按子集绘图和交互式地识别点。

```
library(car)
scatterplot(mpg~wt|cyl,data=mtcars,lwd=2,main="Scatter Plot of MPG vs. Weight by #
Cylinders",xlab="Weight of Car(lbs/1000)",ylab="Miles Per
Gallon",legend.plot=TRUE,id.method="identify",labels=row.names(mtcars),boxplots="xy")
```



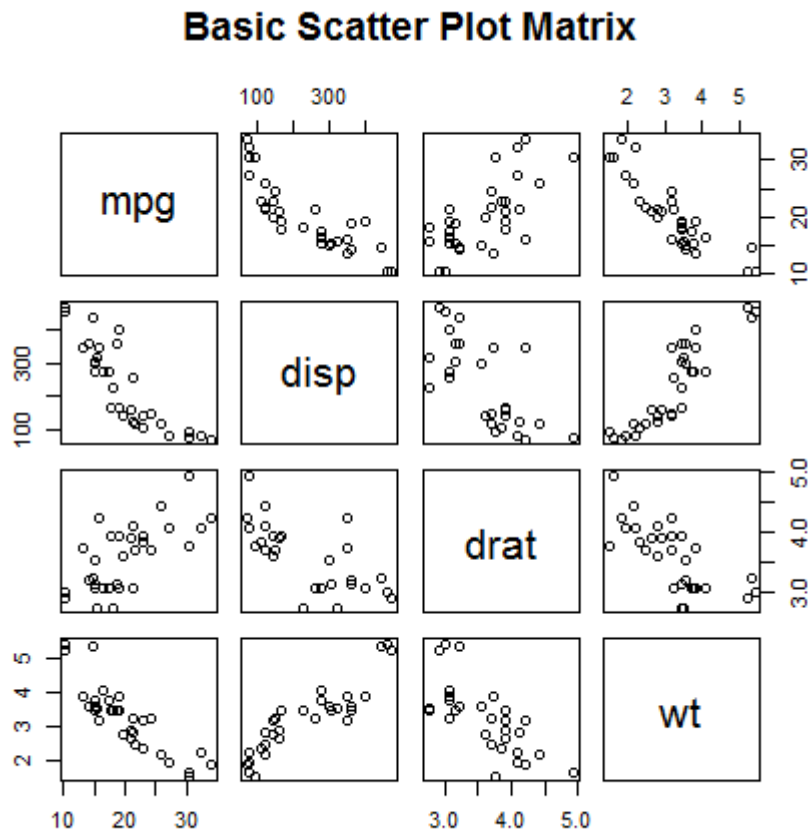
1. `scatterplot()` 函数用来描绘有 4 个 6 个 8 个气缸的汽车每加仑英里数对车重的图形。
2. 表达式 `mpg~wt|cyl` 表示按条件绘图，按照 `cyl` 的水平分别绘制 `mpg` 和 `wt` 关系图。
3. 默认条件下，各个子集通过颜色和图形符号加以区分，并同时绘制线性拟合和平滑拟合曲线。平滑拟合默认需要 5 个单独的数据点。
4. `idmethod` 选项的设点表明可以通过鼠标单击方式来交互式地识别数据点。知道 **Stop**。（在 `studio` 里面是 **ESC**）
5. `legend.plot` 表明在左上边界添加图例。

散点图矩阵

R 中至少有四种创建散点图矩阵的实用函数。

`pairs()` 函数可以创建基础的散点图矩阵。

```
pairs(~mpg+disp+drat+wt,data=mtcars,main="Basic Scatter Plot Matrix")
```

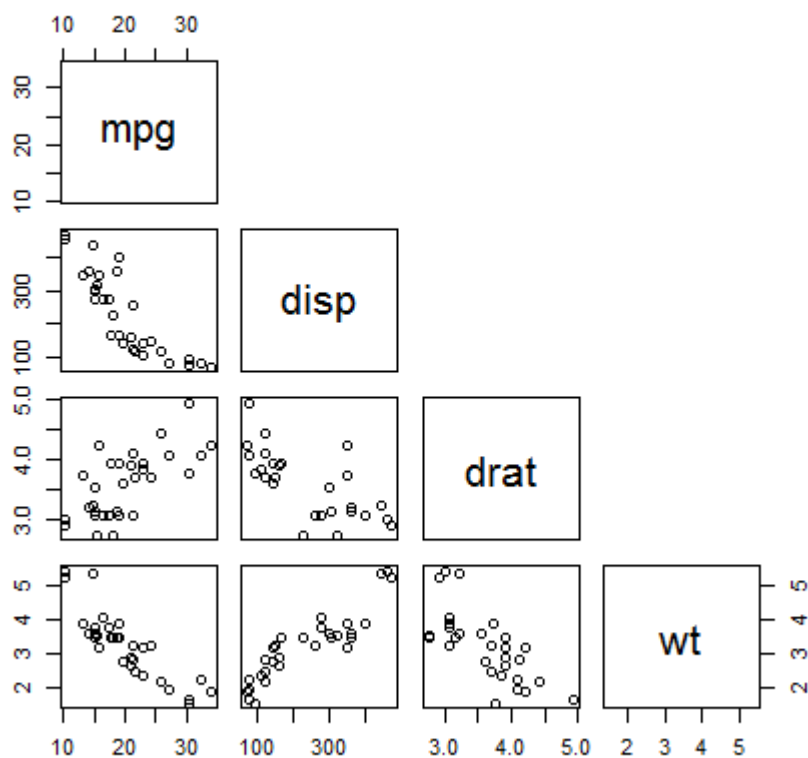


通过调整参数，可以只显示下三角或上三角的图形。

`upper.panel=NULL` 将众生成下三角图形。

```
pairs(~mpg+disp+drat+wt,data=mtcars,main="Basic Scatter Plot Matrix",upper.panel=NULL)
```

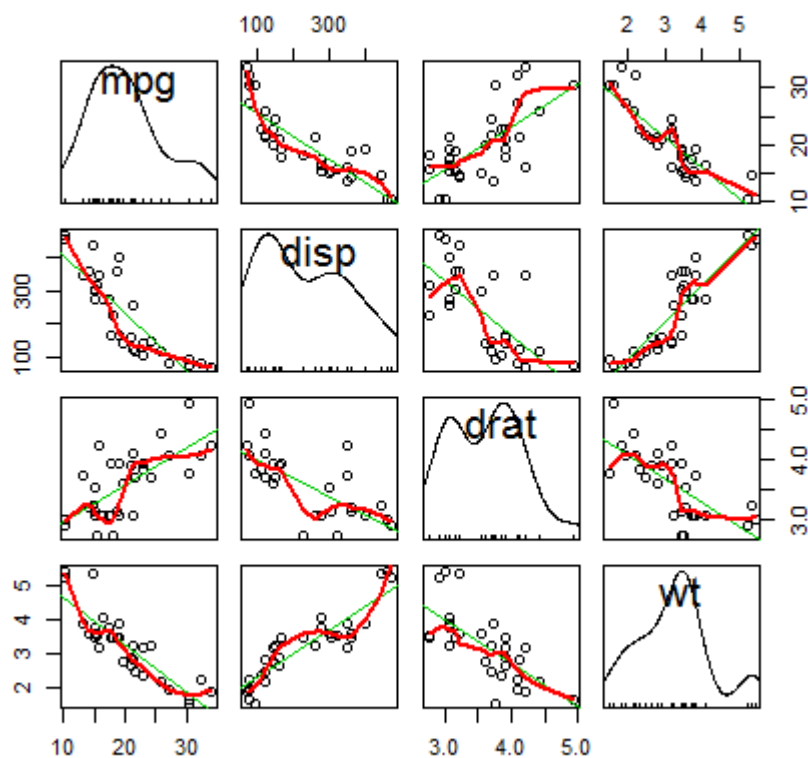
Basic Scatter Plot Matrix



```
scatterplotMatrix(~mpg+disp+drat+wt,data=mtcars,spread=FALSE,lty.smooth=2,main="Scatter  
Plot Matrix via car Package")
```

这里 `lty.smooth=2` 本想设定平滑拟合曲线为虚线，但是报错 `lty.smooth` 不是图形参数，还没有解决。但是如果 `lty=2`，就把拟合的直线设置为虚线。

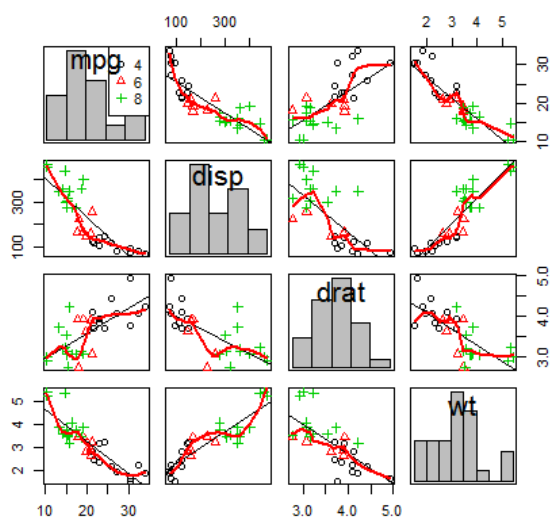
Scatter Plot Matrix via car Package



```
library(car)
scatterplotMatrix(~mpg+disp+drat+wt | cyl,data=mtcars,spread=FALSE,diagonal="histogram",main="Scatter Plot Matrix via car Package")
```

这里，我们想主对角线的和密度曲线改成了直方图（各车汽缸数为条件）默认地，回归直线拟合整个样本，包好 `by.groups=TRUE`。

Scatter Plot Matrix via car Package



`gclus` 包中的 `cpairs()` 函数提供了其变种。它可以重排矩阵中变量位置的选项，可以让相关性更好的变量更靠近对角线。该函数还能对各个单元格进行颜色编码来展示变量间的相关性大小。

```
cor(mtcars[c("mpg","wt","disp","drat")])
```

	mpg	wt	disp	drat
mpg	1.0000000	-0.8676594	-0.8475514	0.6811719
wt	-0.8676594	1.0000000	0.8879799	-0.7124406
disp	-0.8475514	0.8879799	1.0000000	-0.7102139
drat	0.6811719	-0.7124406	-0.7102139	1.0000000

`dmat.color()` `order.single()` `cpairs()` 函数都来自于 `gclus` 包。

第一步，选出需要的变量，并计算他们的相关系数的绝对值。

第二步，使用 `dmat.color()` 获取绘图颜色。给定一个对称矩阵，`dmat.color()` 将返回一个颜色矩阵。

第三步，排序。通过 `order.single()` 散点图矩阵将根据新的变量顺序(`myorder`)和颜色列表(`mycolor`)绘图上色，`gap` 使得矩阵各单元格间的间距稍微增大一些。

```
library(gclus)
```

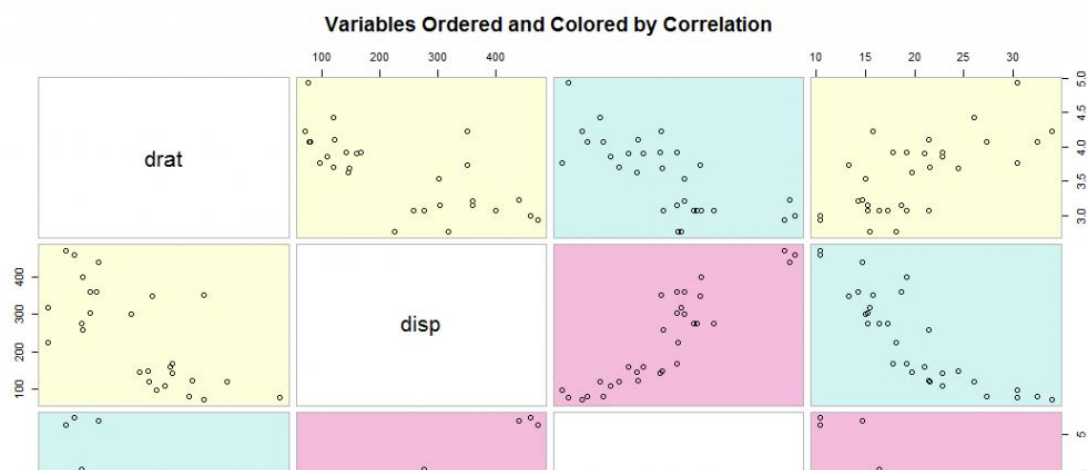
```
mydata=mtcars[c(1,3,5,6)]
```

```
mydata.corr=abs(cor(mydata))
```

```
mycolors=dmat.color(mydata.corr)
```

```
myorder=order.single(mydata.corr)
```

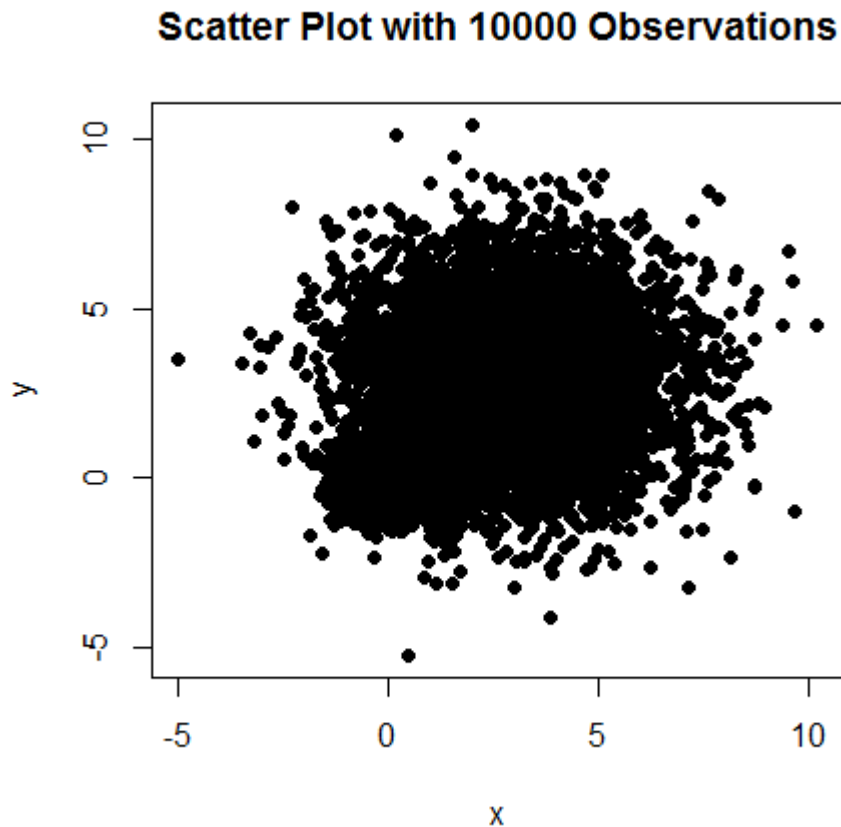
```
cpairs(mydata,myorder,panel.colors=mycolors,gap=.5,main="Variables Ordered and Colored by Correlation")
```




```

set.seed(1234)
n=10000
c1=matrix(rnorm(n,mean=0,sd=.5),ncol=2)
c2=matrix(rnorm(n,mean=3,sd=2),ncol=2)
mydata=rbind(c1,c2)
mydata=as.data.frame(mydata)
names(mydata)=c("x","y")
with(mydata,plot(x,y,pch=19,main="Scatter Plot with 10000 Observations"))

```



针对上述数据点的重叠导致识别 xy 间关系变得异常困难。可以使用封箱、颜色和透明度来知名途中人一点上重叠点的数目。

`smoothScatter()`函数可利用核密度估计声称用颜色密度来表示店分布的散点图。

```

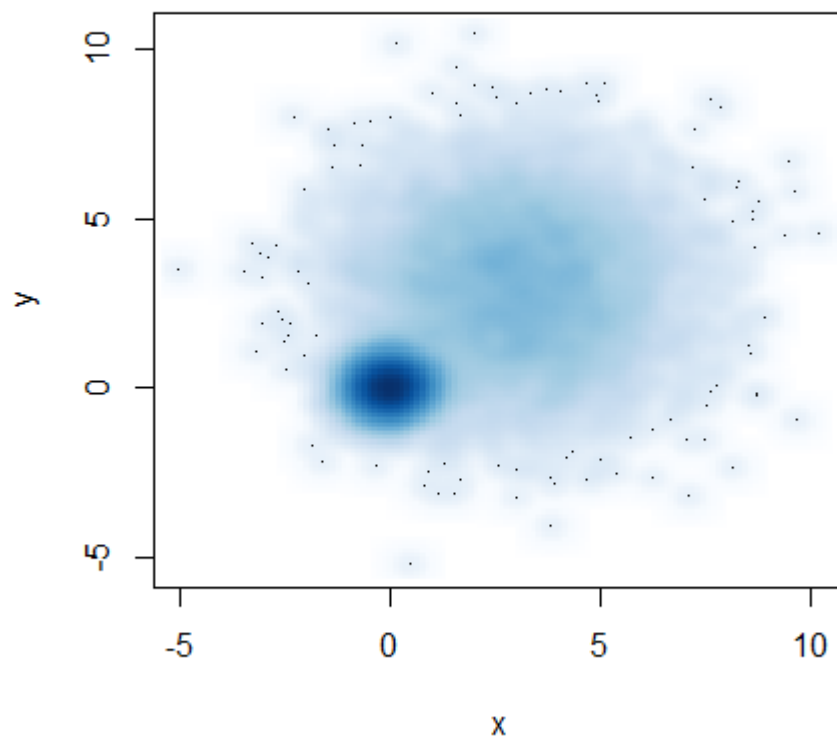
with(mydata,smoothScatter(x,y,main="Scatterplot Colored by Smoothed Densities"))

```

KernSmooth 2.23 loaded

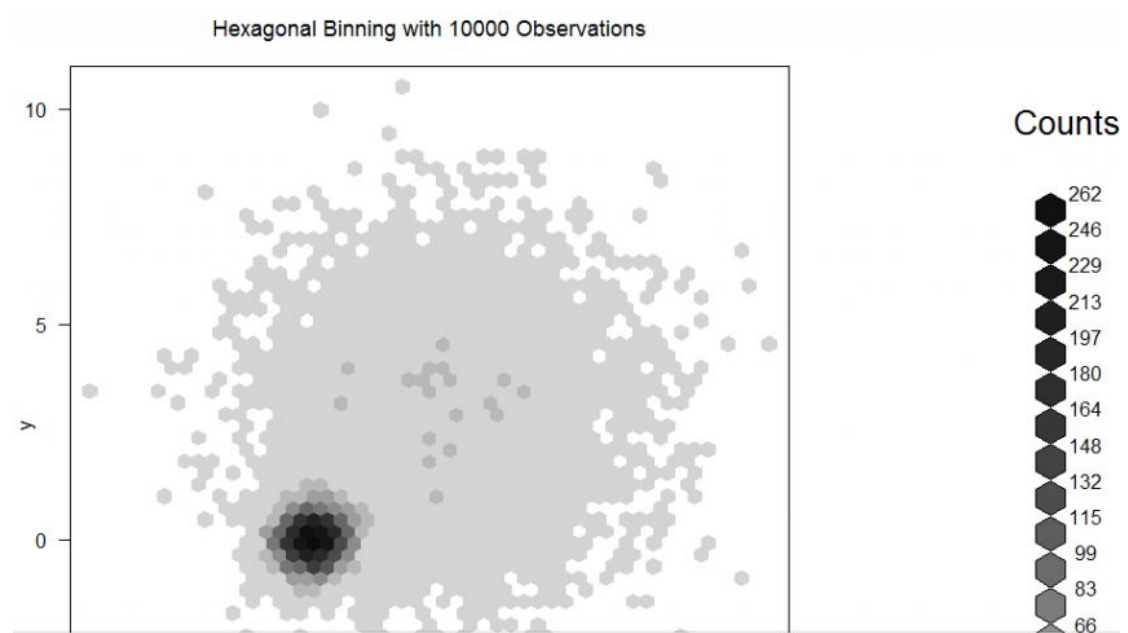
Copyright M. P. Wand 1997-2009

Scatterplot Colored by Smoothed Densities



hexbin 包中的 `hexbin()` 函数将二元变量的封箱放到六边形单元格中(图形比名称更直观)

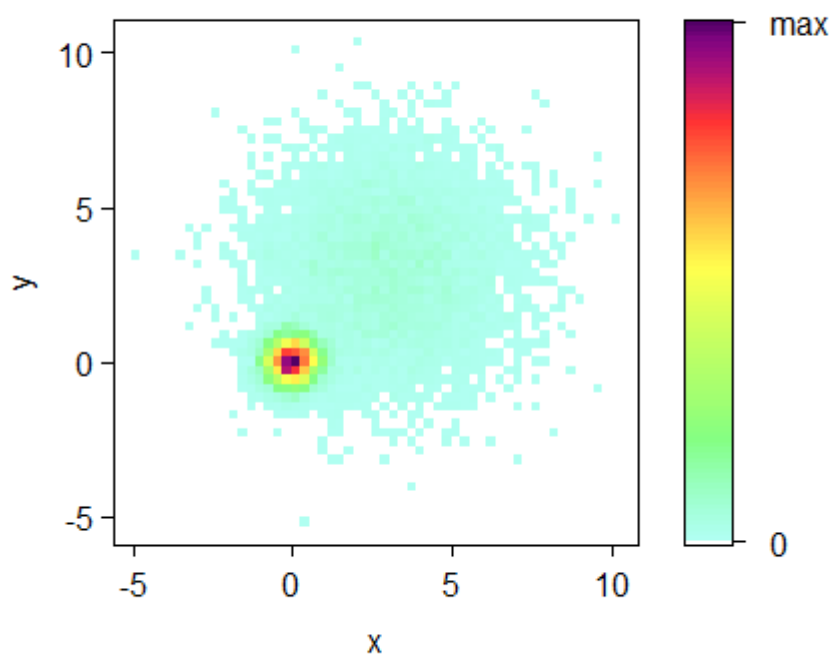
```
library(hexbin)
with(mydata,{
  bin=hexbin(x,y,xbins=50)
  plot(bin,main="Hexagonal Binning with 10000 Observations")
})
```



IDPmisc 包中的 `iplot()` 函数也可以通过颜色来展示店的密度（特定点上的数据数目）。

```
with(mydata,iplot(x,y,main="Image Scatter Plot with Color Indicating Density"))
```

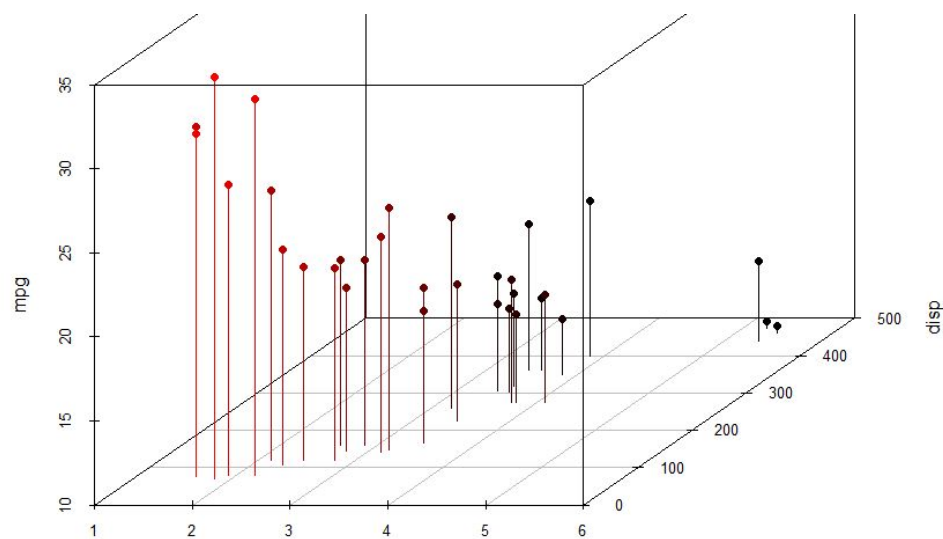
Image Scatter Plot with Color Indicating Density



三维散点图

可以用 `scatterplot3d` 中的 `scatterplot3d()` 函数来绘制他们的关系。

```
library(scatterplot3d)
attach(mtcars)
scatterplot3d(wt, disp, mpg, pch=16, highlight.3d=TRUE, type="h", main="Basic 3D Scatter Plot")
```



```
fit=lm(mpg~wt+disp)
s3d$plane3d(fit)
```

气泡图

先创建一个二维散点图，然后点的大小代表第三个变量的值。

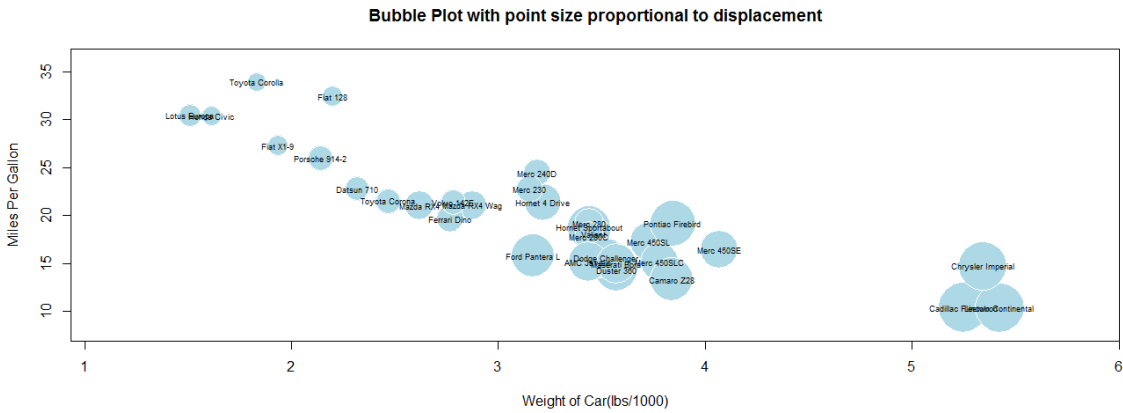
```
sysbols(x,y,circle=radius)
```

`inches` 是比例因子，控制着圆圈大小（就默认最大圆圈为 `1 i n c h`）`t e x t ()` 函数是可选函数，添加各个汽车名称。

一般来说，统计热源倾向于避免使用气泡图，相对于长度来说，人们对于体积或面积判断更吃力。但是在商业应用中非常受欢迎。

```
attach(mtcars)
r=sqrt(disp/pi)
symbols(wt,mpg,circle=r,inches=0.30,fg="white",bg="lightblue",main="Bubble Plot with point
```

```
size proportional to displacement",ylab="Miles Per Gallon",xlab="Weight of Car(lbs/1000)")
text(wt,mpg,rownames(mtcars),cex=0.6)
detach(mtcars)
```



折线图

相关图

相关系数矩阵是多元统计分析的一个基本方法。

哪些被考察的变量与其他变量相关性很强，哪些不强？

相关变量是否以某种特定的方式聚集在一起？

随着变量数的增加，这类问题将变得更难回答。

```
options(digits=2)
cor(mtcars)

      mpg   cyl  disp    hp  drat    wt   qsec    vs    am
mpg   1.00 -0.85 -0.85 -0.78  0.681 -0.87  0.419  0.66  0.600
cyl  -0.85  1.00  0.90  0.83 -0.700  0.78 -0.591 -0.81 -0.523
disp -0.85  0.90  1.00  0.79 -0.710  0.89 -0.434 -0.71 -0.591
hp   -0.78  0.83  0.79  1.00 -0.449  0.66 -0.708 -0.72 -0.243
drat  0.68 -0.70 -0.71 -0.45  1.000 -0.71  0.091  0.44  0.713
wt   -0.87  0.78  0.89  0.66 -0.712  1.00 -0.175 -0.55 -0.692
qsec  0.42 -0.59 -0.43 -0.71  0.091 -0.17  1.000  0.74 -0.230
vs    0.66 -0.81 -0.71 -0.72  0.440 -0.55  0.745  1.00  0.168
am    0.60 -0.52 -0.59 -0.24  0.713 -0.69 -0.230  0.17  1.000
gear  0.48 -0.49 -0.56 -0.13  0.700 -0.58 -0.213  0.21  0.794
carb -0.55  0.53  0.39  0.75 -0.091  0.43 -0.656 -0.57  0.058

      gear   carb
mpg   0.48 -0.551
cyl  -0.49  0.527
disp -0.56  0.395
hp   -0.13  0.750
drat  0.70 -0.091
wt   -0.58  0.428
```

```

qsec -0.21 -0.656
vs    0.21 -0.570
am    0.79  0.058
gear  1.00  0.274
carb  0.27  1.000

```

这里使用的是 `corrgram` 包中的 `corrgram()` 函数。

下三角单元格中：

默认蓝色和从左下指向右上的斜杠表示单元格中的两个变量呈正相关。反过来，红色且从左上指向右下的斜杠表示负相关。

色彩越深，饱和度越高，说明变化相关性越大。

相关性接近于 0 的单元格基本白色。

上三角单元格中：

颜色功能同上，但相关性大小由被填充的饼图块大小展示出来。争先惯性将从 12 点钟处开始顺时针填充饼图，而负相关将逆时针填充饼图。

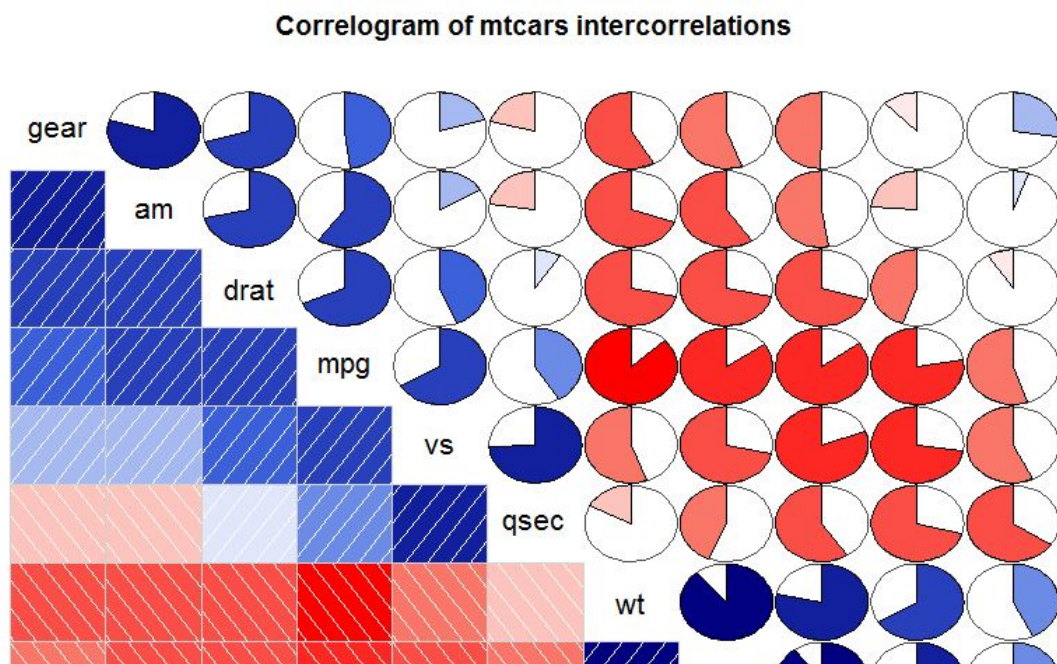
`order=TRUE` 时候，相关矩阵将使用 PCA 分析对变量重新排序。

`panel` 设定非对角线使用的元素的类型。`lower.panel upper.panel ; panel.pie(饼图)/panel.shade`（阴影深度）/`panel.ellipse`（置信椭圆）/`panel.pts`（平滑拟合曲线）

`text.panel` and `diag.panel` 控制着主对角线元素类型。`panel.minmax`（输出最大最小值）/`panel.txt`

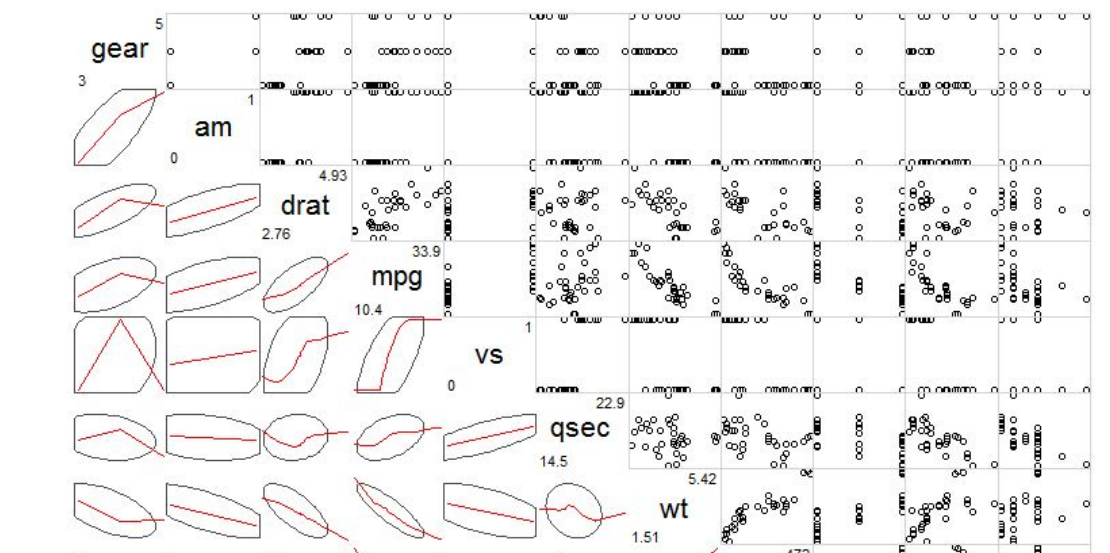
```
library(corrgram)
```

```
corrgram(mtcars,order=TRUE,lower.panel=panel.shade,upper.panel=panel.pie,text.panel=panel.txt,main="Correlogram of mtcars intercorrelations")
```

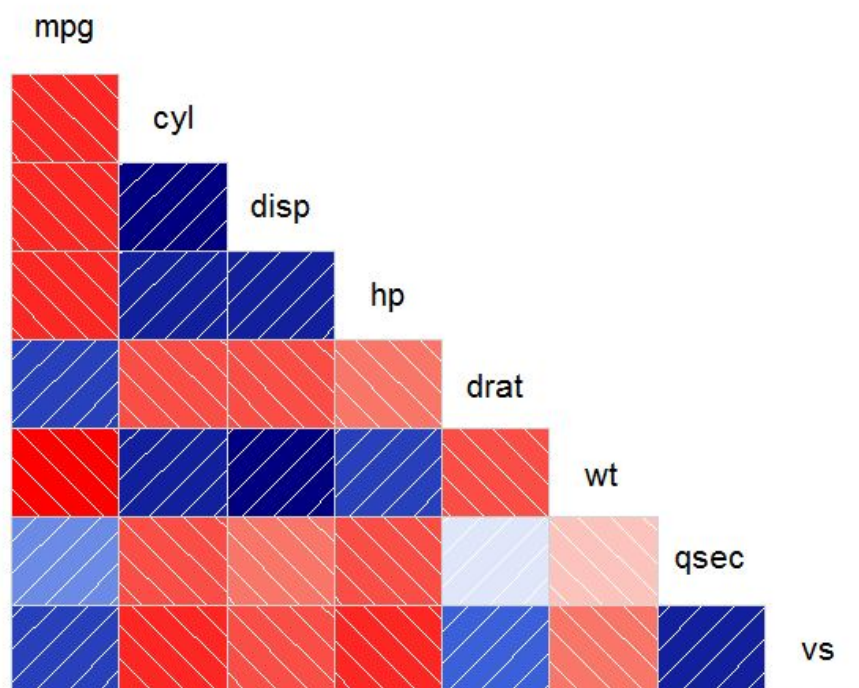


```
corrgram(mtcars,order=TRUE,lower.panel=panel.ellipse,upper.panel=panel.pts,text.panel=panel.txt,diag.panel=panel.minmax,main="Correlogram of mtcars data using scatter plots and ellipses")
```

Correlogram of mtcars data using scatter plots and ellipses



```
corrgram(mtcars,lower.panel=panel.shade,upper.panel=NULL,text.panel=panel.txt,main="Car Mileage Data(unsorted)")
```



马赛克图