

《数据统计分析与R语言》

基本图形



内容回顾

- 🦌 数学和统计函数
- 🦌 字符和处理函数
- 🦌 循环和条件执行
- 🦌 自编函数
- 🦌 数据整合和重塑





内容（一）

- ❧ 图形的创建和保存
- ❧ 自定义符号、线条、颜色和坐标轴
- ❧ 标注文本和标题
- ❧ 控制图形维度
- ❧ 组合多个图

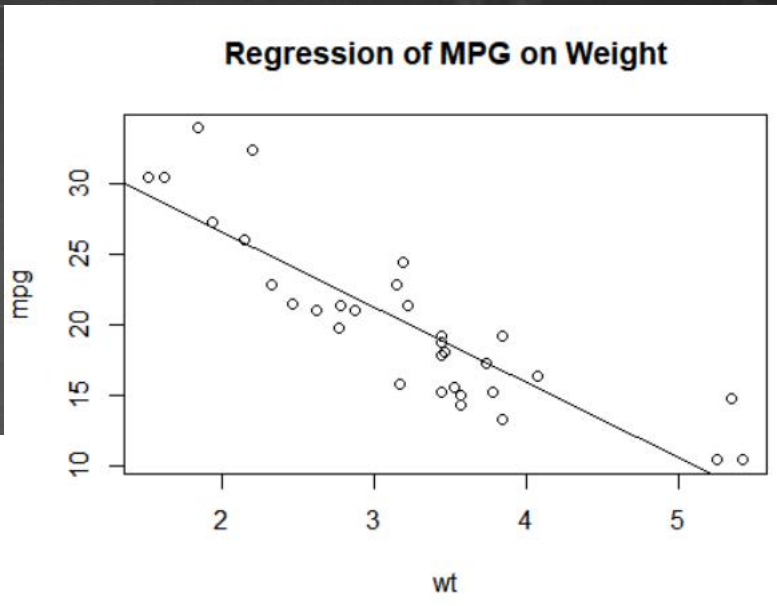




使用图形

🦌 R一个惊艳的图形构建平台

```
attach(mtcars)
plot(wt, mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
detach(mtcars)
```





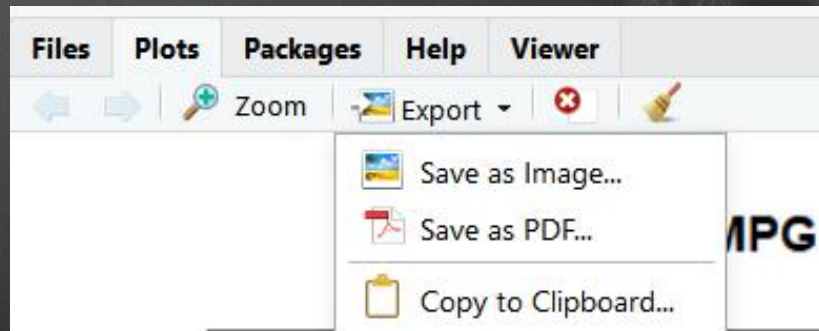
使用图形

保存图形

函数pdf(),png(),jpeg()...

图形用户界面

```
pdf("mygraph.pdf")
attach(mtcars)
plot(wt, mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
detach(mtcars)
dev.off()
```

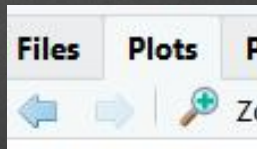




使用图形

🌀 创建多个图形并随时查看每一个

- 1、打开新窗口，每一幅新图形将出现在近一次打开的窗口中
- 2、可以通过图形用户界面来查看多个图形
- 3、可以使用函数`dev.new()`、`dev.next()`、`dev.prev()`、`dev.set()`和`dev.off()`同时打开多个图形窗口





一个简单的例子

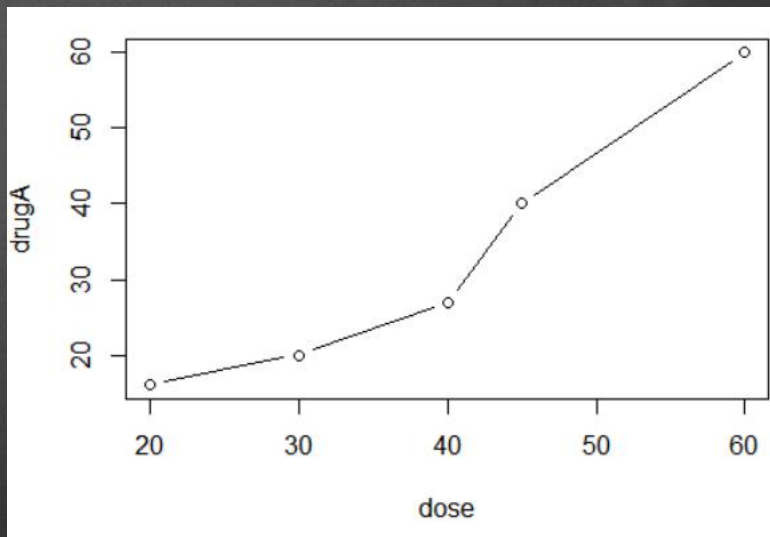
表3-1 病人对两种药物五个剂量水平上的响应情况

剂 量	对药物A的响应	对药物B的响应
20	16	15
30	20	18
40	27	25
45	40	31
60	60	40



一个简单的例子

🔗 `plot(dose, drugA, type="b")`

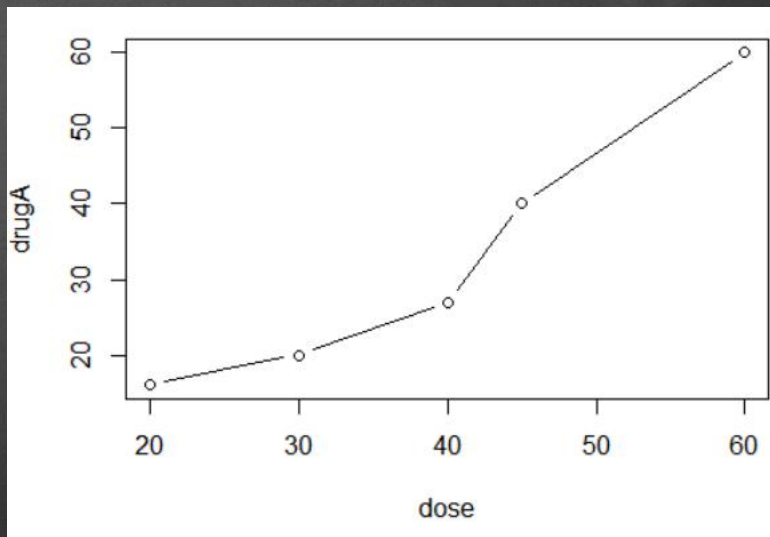




图形参数

函数par()

par(optionname=value, optionname=name,...)

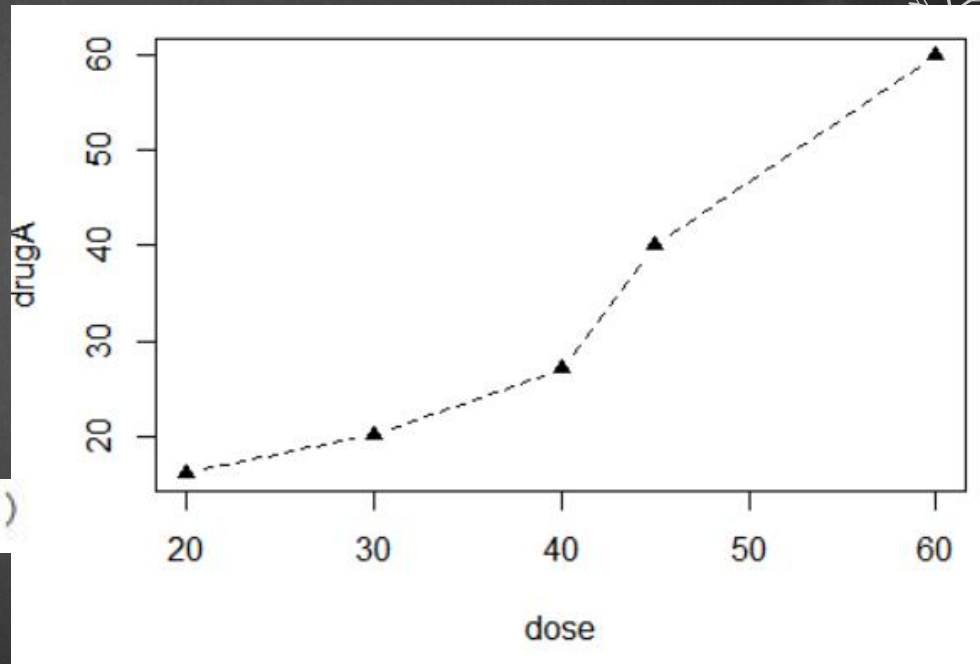




一个简单的例子

```
opar <- par(no.readonly=TRUE)
par(lty=2, pch=17)
plot(dose, drugA, type="b")
par(opar)
```

```
plot(dose, drugA, type="b", lty=2, pch=17)
```





图形参数

符号和线条

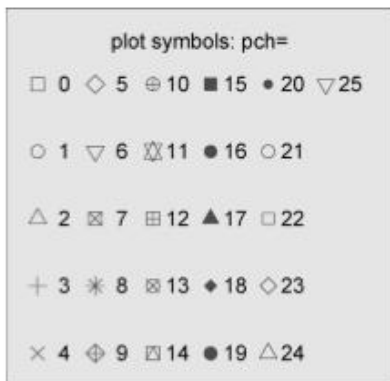


图3-4 参数pch可指定的绘图符号

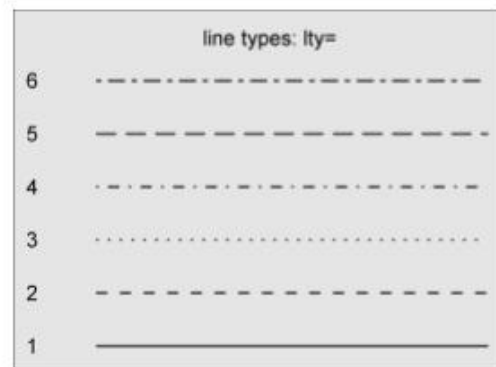


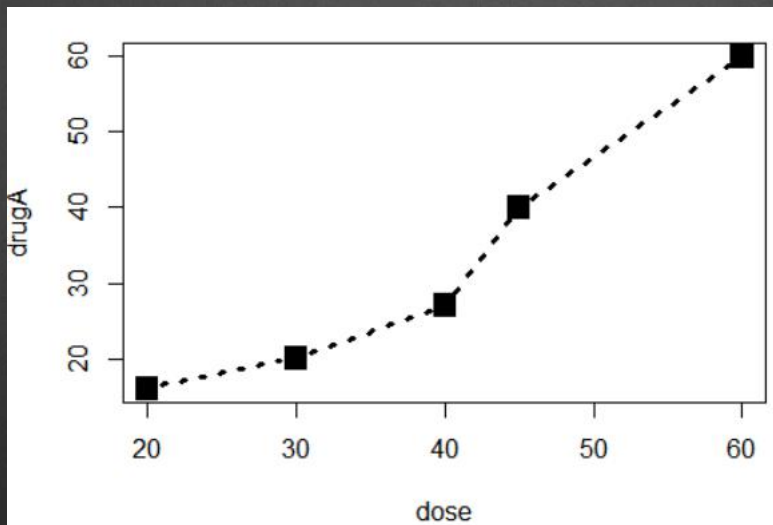
图3-5 参数lty可指定的线条类型

参 数	描 述
pch	指定绘制点时使用的符号（见图3-4）
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，0.5表示缩小为默认值的50%，等等
lty	指定线条类型（参见图3-5）
lwd	指定线条宽度。lwd是以默认值的相对大小来表示的（默认值为1）。例如，lwd=2将生成一条两倍于默认宽度的线条



图形参数

❧ `plot(dose, drugA, type="b", lty=3, lwd=3, pch=15, cex=2)`





图形参数

颜色

参 数	描 述
<code>col</code>	默认的绘图颜色。某些函数（如 <code>lines</code> 和 <code>pie</code> ）可以接受一个含有颜色值的向量并自动循环使用。例如，如果设定 <code>col=c("red", "blue")</code> 并需要绘制三条线，则第一条线将为红色，第二条线为蓝色，第三条线又将为红色
<code>col.axis</code>	坐标轴刻度文字的颜色
<code>col.lab</code>	坐标轴标签（名称）的颜色
<code>col.main</code>	标题颜色
<code>col.sub</code>	副标题颜色
<code>fg</code>	图形的前景色
<code>bg</code>	图形的背景色

`col=1`、`col="white"`、`col="#FFFFFF"`、`col=rgb(1,1,1)`和`col=hsb(0,0,1`



图形参数

🦌 R语言由新西兰

```
n <- 10  
mycolors <- rainbow(n)  
pie(rep(1, n), labels=mycolors, col=mycolors)  
mygrays <- gray(0:n/n)
```





图形参数

文本属性

表3-4 用于指定文本大小的参数

参 数	描 述
<code>cex</code>	表示相对于默认大小缩放倍数的数值。默认大小为1, 1.5表示放大为默认值的1.5倍, 0.5表示缩小为默认值的50%, 等等
<code>cex.axis</code>	坐标轴刻度文字的缩放倍数。类似于 <code>cex</code>

参 数	描 述
<code>cex.lab</code>	坐标轴标签（名称）的缩放倍数。类似于 <code>cex</code>
<code>cex.main</code>	标题的缩放倍数。类似于 <code>cex</code>
<code>cex.sub</code>	副标题的缩放倍数。类似于 <code>cex</code>



图形参数



文本属性

`par(font.lab=3, cex.lab=1.5, font.main=4, cex.main=2)`

参 数	描 述
<code>font</code>	整数。用于指定绘图使用的字体样式。1=常规，2=粗体，3=斜体，4=粗斜体，5=符号字体（以Adobe符号编码表示）
<code>font.axis</code>	坐标轴刻度文字的字体样式
<code>font.lab</code>	坐标轴标签（名称）的字体样式
<code>font.main</code>	标题的字体样式
<code>font.sub</code>	副标题的字体样式
<code>ps</code>	字体磅值（1磅约为1/72英寸）。文本的最终大小为 $ps * cex$
<code>family</code>	绘制文本时使用的字体族。标准的取值为 <code>serif</code> （衬线）、 <code>sans</code> （无衬线）和 <code>mono</code> （等宽）



图形参数

图形尺寸与边界尺寸

```
par(pin=c(4,3), mai=c(1,.5, 1, .2))
```

表3-6 用于控制图形尺寸和边界大小的参数

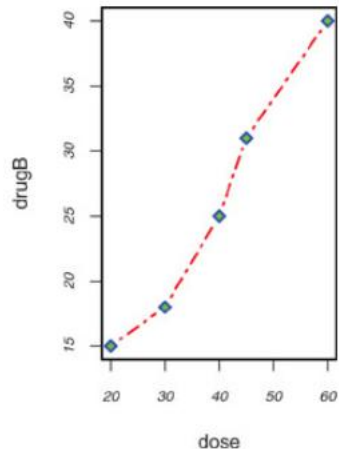
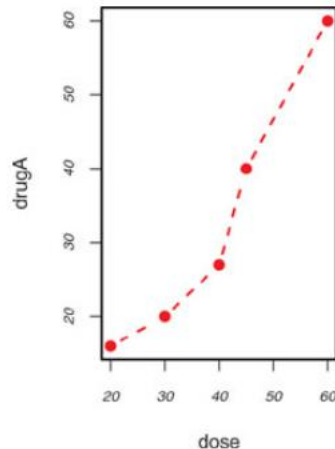
参 数	描 述
pin	以英寸表示的图形尺寸（宽和高）
mai	以数值向量表示的边界大小，顺序为“下、左、上、右”，单位为英寸
mar	以数值向量表示的边界大小，顺序为“下、左、上、右”，单位为英寸。默认值为c(5, 4, 4, 2) + 0.1



图形参数

🌀 示例：使用图形参数控制图形外观

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
opar <- par(no.readonly=TRUE)
par(pin=c(2, 3))
par(lwd=2, cex=1.5)
par(cex.axis=.75, font.axis=3)
plot(dose, drugA, type="b", pch=19, lty=2, col="red")
plot(dose, drugB, type="b", pch=23, lty=6, col="blue", bg="green")
par(opar)
```

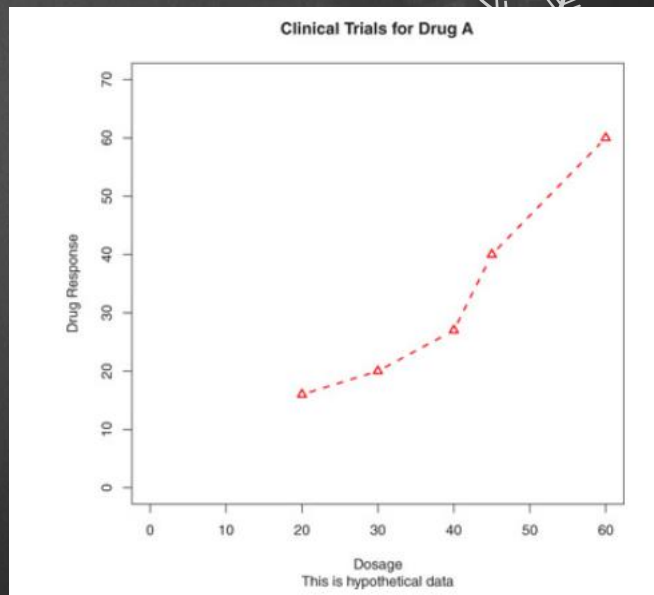




添加文本、自定义坐标轴和图例

许多高级绘图函数允许自行设定坐标轴和文本标注选项

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





添加文本、自定义坐标轴和图例

❧ 标题

```
title(main="main title", sub="sub-title",  
      xlab="x-axis label", ylab="y-axis label")
```

```
title(main="My Title", col.main="red",  
      sub="My Sub-title", col.sub="blue",  
      xlab="My X label", ylab="My Y label",  
      col.lab="green", cex.lab=0.75)
```



添加文本、自定义坐标轴和图例

坐标轴

`axis(side, at=, labels=, pos=, lty=, col=, las=, tck=, ...)`

表3-7 坐标轴选项

选 项	描 述
<code>side</code>	一个整数，表示在图形的哪边绘制坐标轴（1=下，2=左，3=上，4=右）
<code>at</code>	一个数值型向量，表示需要绘制刻度线的位置
<code>labels</code>	一个字符型向量，表示置于刻度线旁边的文字标签（如果为NULL，则将直接使用 <code>at</code> 中的值）
<code>pos</code>	坐标轴线绘制位置的坐标（即与另一条坐标轴相交位置的值）
<code>lty</code>	线条类型
<code>col</code>	线条和刻度线颜色
<code>las</code>	标签是否平行于（=0）或垂直于（=2）坐标轴
<code>tck</code>	刻度线的长度，以相对于绘图区域大小的分数表示（负值表示在图形外侧，正值表示在图形内侧，0表示禁用刻度，1表示绘制网格线）；默认值为-0.01
<code>(...)</code>	其他图形参数



添加文本、自定义坐标轴和图例

```
#4.2主坐标轴
x <- c(1:10)
y <- x
z <- 10/x

opar <- par(no.readonly=TRUE)
par(mar=c(5, 4, 4, 8) + 0.1)

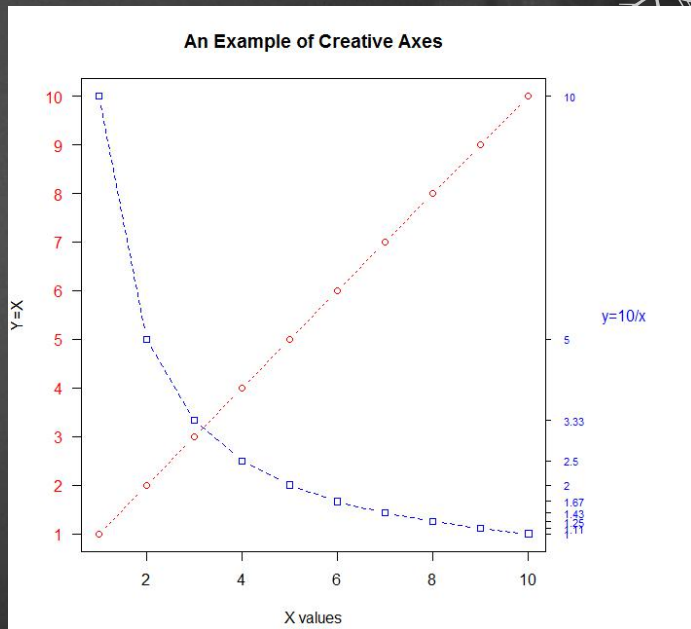
plot(x, y, type="b",
      pch=21, col="red",
      yaxt="n", lty=3, ann=FALSE)#ann=FALSE移除默认标签 #yaxt="n"禁用y轴
lines(x, z, type="b", pch=22, col="blue", lty=2)

axis(2, at=x, labels=x, col.axis="red", las=2)#las=2垂直于坐标轴

axis(4, at=z, labels=round(z, digits=2),
      col.axis="blue", las=2, cex.axis=0.7, tck=-.01)#tck=-.01刻度线长度

#mtext在图形边缘加文字, side=4右侧, line=3文字与图形边界距离 cex.lab坐标轴名称
mtext("y=10/x", side=4, line=3, cex.lab=1, las=2, col="blue")

title("An Example of Creative Axes",
      xlab="X values",
      ylab="Y=X")
par(opar)
```





添加文本、自定义坐标轴和图例

自定义坐标轴 次刻度线

```
library(Hmisc)  
minor.tick(nx=n, ny=n, tick.ratio=n)
```

```
minor.tick(nx=2, ny=3, tick.ratio=0.5)
```




添加文本、自定义坐标轴和图例

参考线

`abline(h=yvalues, v=xvalues)`

函数`abline()`中指定其他图形参数

```
abline(h=c(1,5,7))
```

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





添加文本、自定义坐标轴和图例

图例

`legend(location, title, legend, ...)`

选 项	描 述
<code>location</code>	有许多方式可以指定图例的位置。你可以直接给定图例左上角的x、y坐标，也可以执行 <code>locator(1)</code> ，然后通过鼠标单击给出图例的位置，还可以使用关键字 <code>bottom</code> 、 <code>bottomleft</code> 、 <code>left</code> 、 <code>topleft</code> 、 <code>top</code> 、 <code>topright</code> 、 <code>right</code> 、 <code>bottomright</code> 或 <code>center</code> 放置图例。如果你使用了以上某个关键字，那么可以同时使用参数 <code>inset</code> =指定图例向图形内侧移动的大小（以绘图区域大小的分数表示）
<code>title</code>	图例标题的字符串（可选）
<code>legend</code>	图例标签组成的字符型向量
<code>...</code>	其他选项。如果图例标示的是颜色不同的线条，需要指定 <code>col</code> =加上颜色值组成的向量。如果图例标示的是符号不同的点，则需指定 <code>pch</code> =加上符号的代码组成的向量。如果图例标示的是不同的线条宽度或线条类型，请使用 <code>lwd</code> =或 <code>lty</code> =加上宽度值或类型值组成的向量。要为图例创建颜色填充的盒形（常见于条形图、箱线图或饼图），需要使用参数 <code>fill</code> =加上颜色值组成的向量



添加文本、

图例

示例：依剂

量对比药物A

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
```

```
opar <- par(no.readonly=TRUE)
```

```
par(lwd=2, cex=1.5, font.lab=2)
```

```
plot(dose, drugA, type="b",
     pch=15, lty=1, col="red", ylim=c(0, 60),
     main="Drug A vs. Drug B",
     xlab="Drug Dosage", ylab="Drug Response")
```

```
lines(dose, drugB, type="b",
      pch=17, lty=2, col="blue")
```

```
abline(h=c(30), lwd=1.5, lty=2, col="gray")
```

```
library(Hmisc)
minor.tick(nx=3, ny=3, tick.ratio=0.5)
```

```
legend("topleft", inset=.05, title="Drug Type", c("A","B"),
      lty=c(1, 2), pch=c(15, 17), col=c("red", "blue"))
```

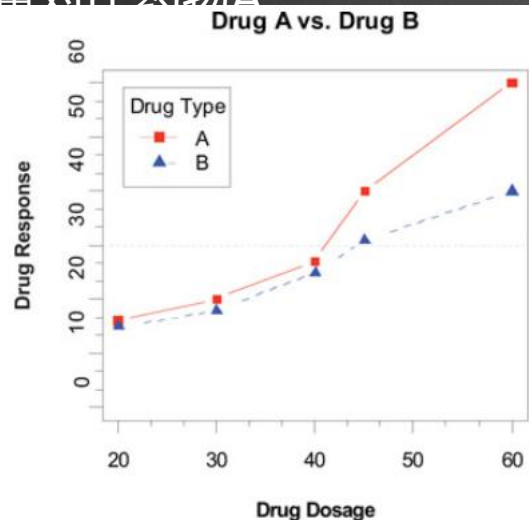
```
par(opar)
```

增加线条、文本、符号、
标签的宽度或大小

绘制图形

添加次要刻度线

添加图例





添加文本、自定义坐标轴和图例

❧ 文本标注

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

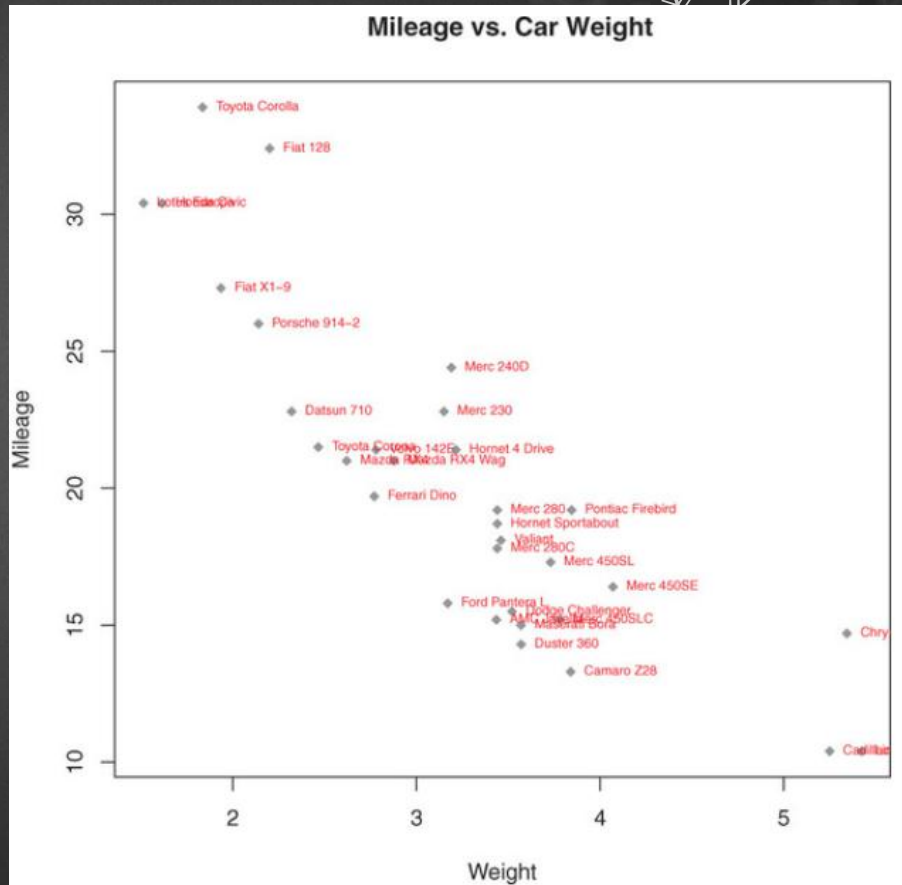
`mtext("text to place", side, line=n, ...)`

表3-9 函数`text()`和`mtext()`的选项

选 项	描 述
location	文本的位置参数。可为一对x,y坐标,也可通过指定location为locator(1)使用鼠标交互式地确定摆放位置
pos	文本相对于位置参数的方位。1=下,2=左,3=上,4=右。如果指定了pos,就可以同时指定参数offset=作为偏移量,以相对于单个字符宽度的比例表示
side	指定用来放置文本的边。1=下,2=左,3=上,4=右。你可以指定参数line=来内移或外移文本,随着值的增加,文本将外移。也可使用adj=0将文本向左下对齐,或使用adj=1右上对齐



```
attach(mtcars)
plot(wt, mpg,
     main="Mileage vs. Car Weight",
     xlab="Weight", ylab="Mileage",
     pch=18, col="blue")
text(wt, mpg,
     row.names(mtcars),
     cex=0.6, pos=4, col="red")
detach(mtcars)
```





图形的组合

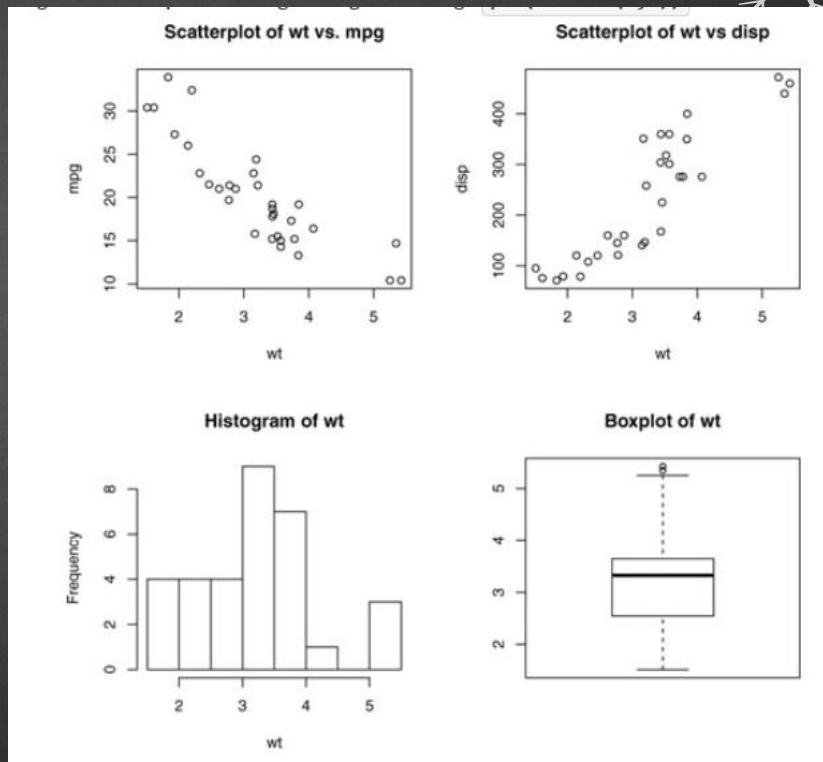
- 在par()函数中使用图形参数mfrow=c(nrows, ncols)来创建按行填充的、行数为nrows、列数为ncols的图形矩阵
- 可以使用mfcol=c(nrows, ncols)按列填充矩阵

```
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="Histogram of wt")
boxplot(wt, main="Boxplot of wt")
par(opar)
detach(mtcars)
```



图形的组合

🌀 `par(mfrow=c(2,2))`

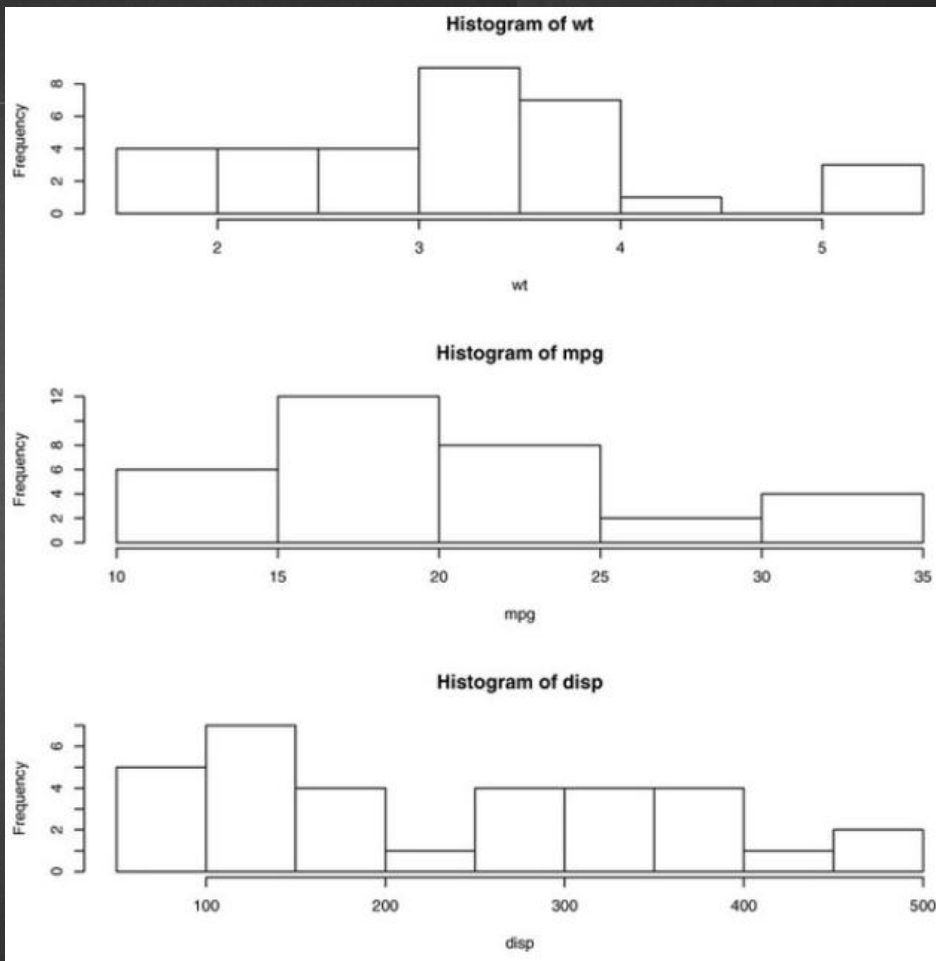




图形的组合

🌀 `par(mfrow=c(3,1))`

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

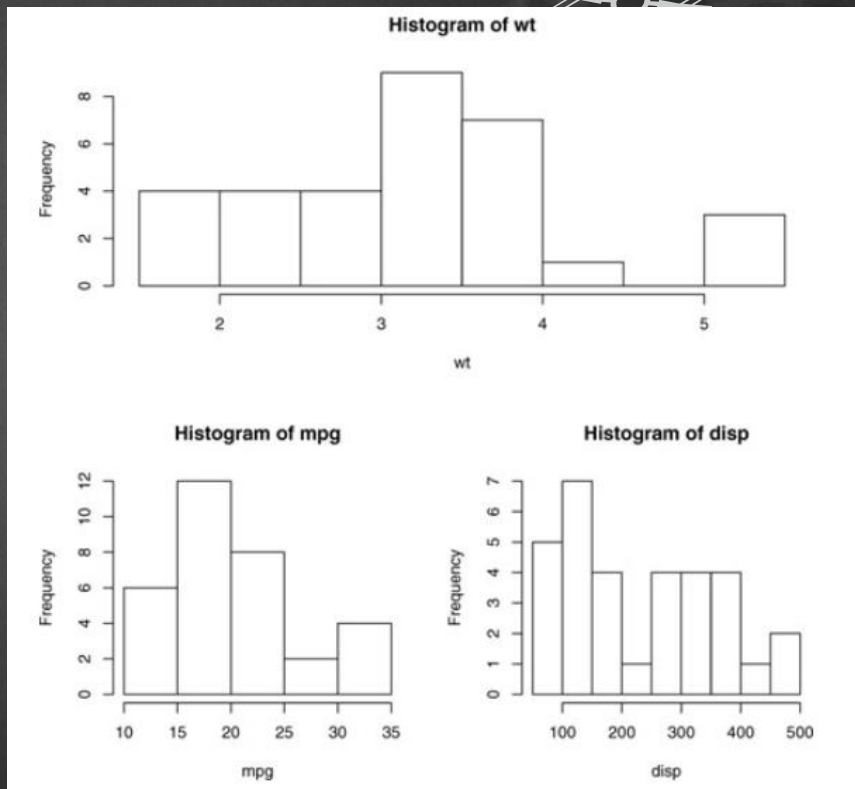




图形的组合

❧ 函数`layout()`的调用形式为
`layout(mat)`, 其中的`mat`是一个矩阵,
它指定了所要组合的多个图形的所在位
置

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

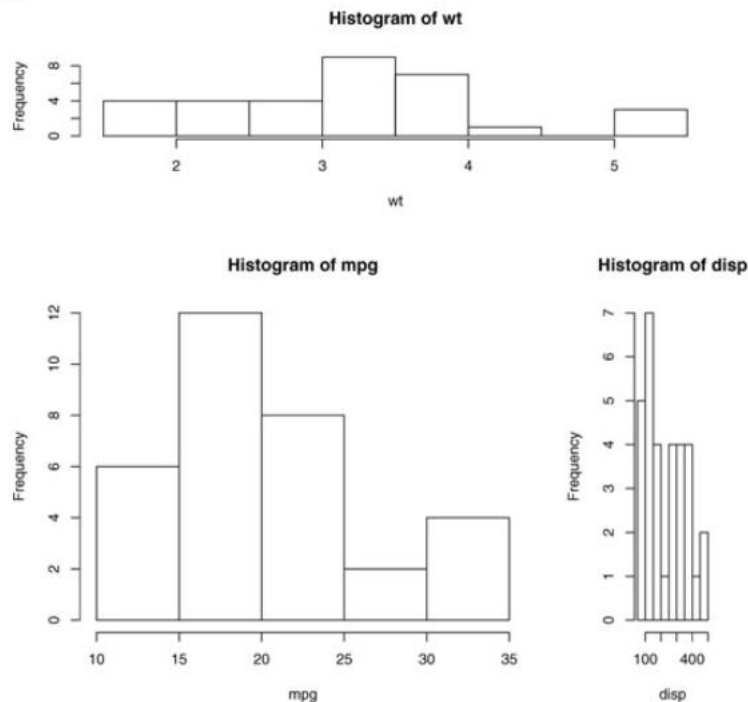




图形的组合

❧ `layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE), widths=c(3, 1), heights=c(1, 2))`

```
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)
```





图形的组合

图形布局的精细控制
使用图形参数fig=完成

```
opar <- par(no.readonly=TRUE)
par(fig=c(0, 0.8, 0, 0.8))
plot(mtcars$wt, mtcars$mpg,
      xlab="Miles Per Gallon",
      ylab="Car Weight")
```

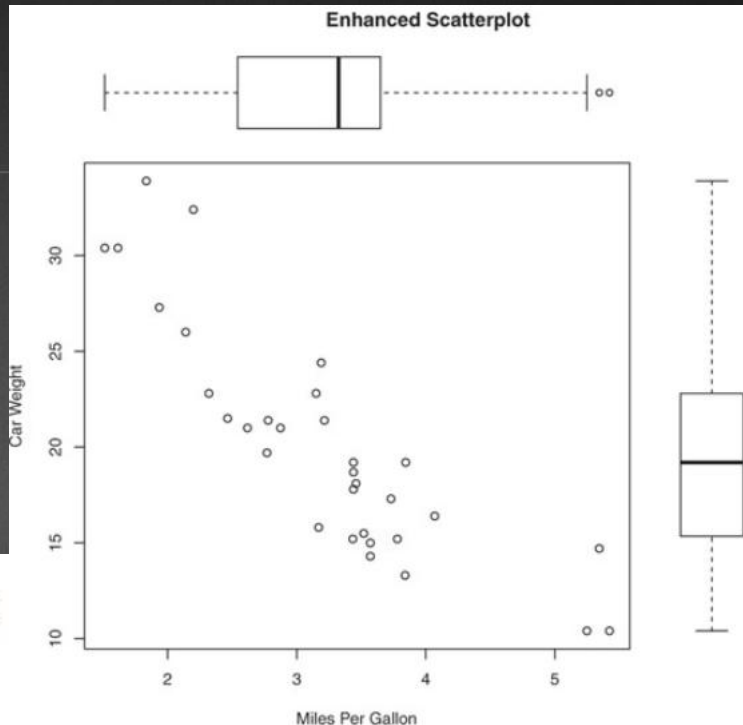
```
par(fig=c(0, 0.8, 0.55, 1), new=TRUE)
boxplot(mtcars$wt, horizontal=TRUE, axes=FALSE)
par(fig=c(0.65, 1, 0, 0.8), new=TRUE)
boxplot(mtcars$mpg, axes=FALSE)
```

```
mtext("Enhanced Scatterplot", side=3, outer=TRUE, line=-3)
par(opar)
```

← 设置散点图

← 在上方添加箱线图

← 在右侧添加箱线图





内容（二）

- ❧ 条形图、箱线图和点图
- ❧ 饼图和扇形图
- ❧ 直方图与和密度图





条形图

通过垂直的或水平的条形展示了类别型变量的分布（频数）

`barplot(height)` `height`是一个向量或矩阵



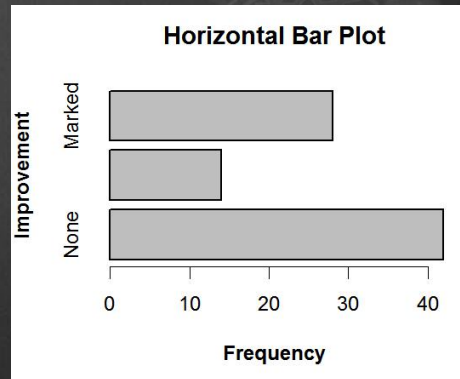
条形图

简单条形图

barplot(height)

```
> library(vcd)
> counts <- table(Arthritis$Improved)
> counts
None    Some Marked
  42     14     28
```

```
# vertical barplot
barplot(counts,
        main="Simple Bar Plot",
        xlab="Improvement", ylab="Frequency")
# horizontal bar plot
barplot(counts,
        main="Horizontal Bar Plot",
        xlab="Frequency", ylab="Improvement",
        horiz=TRUE)
```

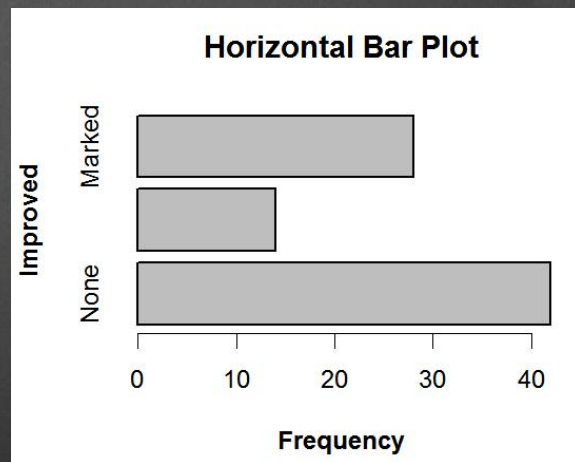
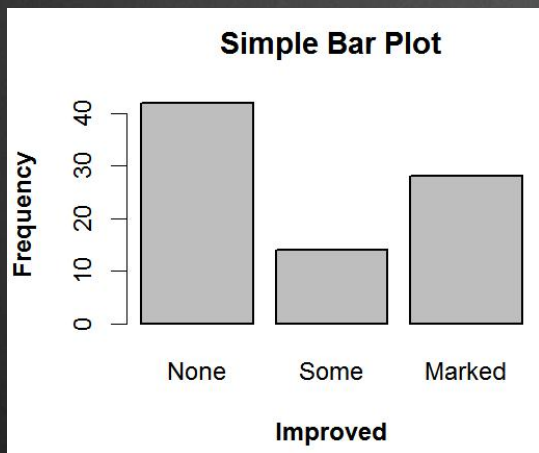




条形图

```
plot(Arthritis$Improved, main="Simple Bar Plot",  
      xlab="Improved", ylab="Frequency")  
plot(Arthritis$Improved, horiz=TRUE, main="Horizontal Bar Plot",  
      xlab="Frequency", ylab="Improved")
```

简单条





条形图

堆砌条形图和分组条形图

barplot(height)

```
# stacked barplot
barplot(counts,
        main="Stacked Bar Plot",
        xlab="Treatment", ylab="Frequency",
        col=c("red", "yellow", "green"),
        legend=rownames(counts))

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

```
> library(vcd)

> counts <- table(Arthritis$Improved, Arthritis$Treatment)

> counts
```

	Placebo	Treated
None	29	13
Some	7	7
Marked	7	21



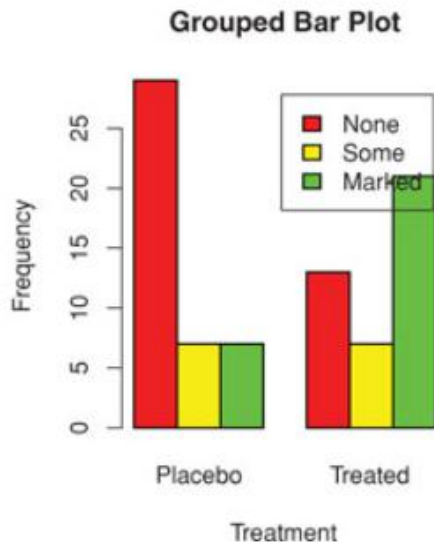
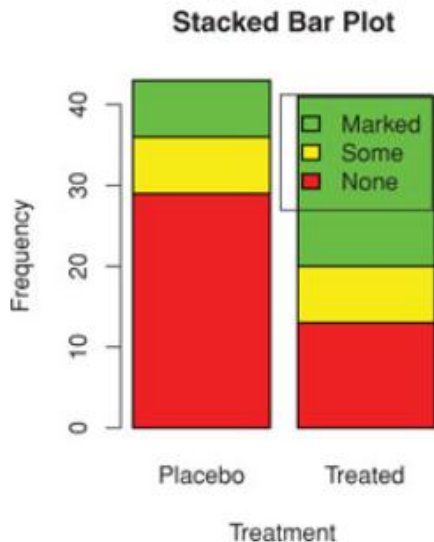
条形图

堆砌条形图和分组条形图

barplot(height)

```
# stacked barplot
barplot(counts,
        main="Stacked Bar Plot",
        xlab="Treatment", ylab="Frequency",
        col=c("red", "yellow", "green"),
        legend=rownames(counts))

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





条形图

均值条形图

```
> states <- data.frame(state.region, state.x77)
> means <- aggregate(states$Illiteracy, by=list(state.region), FUN=mean)
> means
      Group.1      x
1 Northeast 1.00
2 South 1.74
3 North Central 0.70
4 West 1.02
> means <- means[order(means$x),]
> means
      Group.1      x
3 North Central 0.70
1 Northeast 1.00
4 West 1.02
2 South 1.74
> barplot(means$x, names.arg=means$Group.1)
> title("Mean Illiteracy Rate")
```

① 将均值从小到大排序

② 添加标题

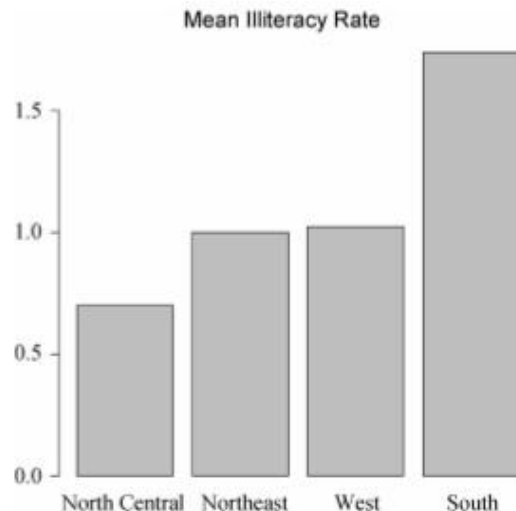


图6-3 美国各地区平均文盲率排序的条形图



条形图

均值条形图

```
> states <- data.frame(state.region, state.x77)
> means <- aggregate(states$Illiteracy, by=list(state.region), FUN=mean)
> means
      Group.1      x
1 Northeast 1.00
2 South 1.74
3 North Central 0.70
4 West 1.02
> means <- means[order(means$x),]
> means
      Group.1      x
3 North Central 0.70
1 Northeast 1.00
4 West 1.02
2 South 1.74
> barplot(means$x, names.arg=means$Group.1)
> title("Mean Illiteracy Rate")
```

① 将均值从小到大排序

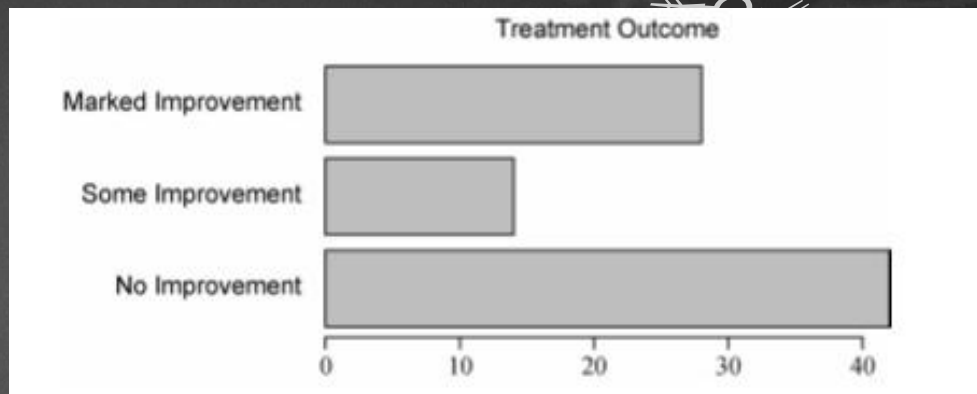
② 添加标题



条形图

❧ 条形图的微调

```
1 par(mar=c(5,8,4,2))
2 par(las=2)
3 counts <- table(Arthritis$Improved)
4
5 barplot(counts,
6         main="Treatment Outcome",
7         horiz=TRUE, cex.names=0.8,
8         names.arg=c("No Improvement", "Some Improvement",
9                     "Marked Improvement"))
```

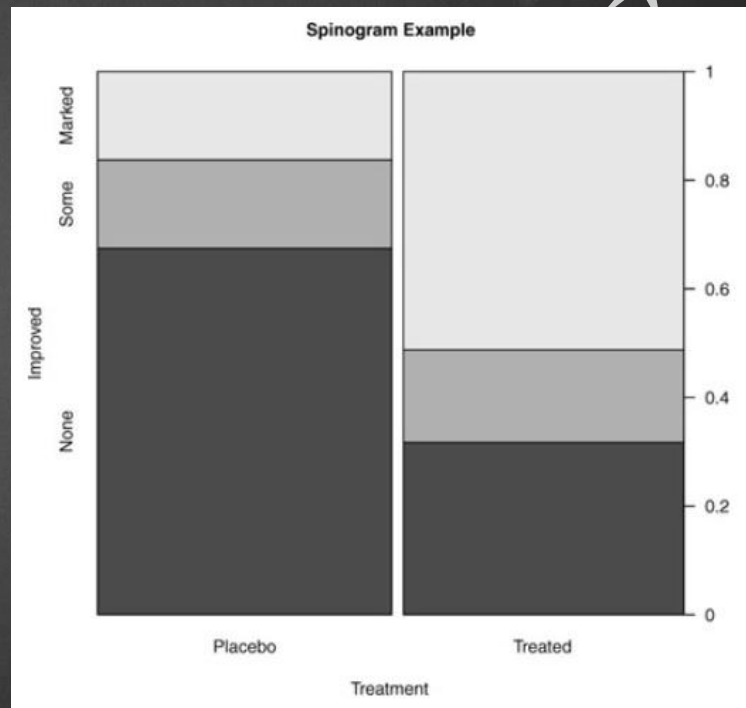




条形图

棘状图

```
1 library(vcd)
2 attach(Arthritis)
3 counts <- table(Treatment, Improved)
4 spine(counts, main="Spinogram Example")
5 detach(Arthritis)
```





饼图

🌀 饼图可由以下函数创建: `pie(x, labels)`

```
par(mfrow=c(2, 2))
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
```

```
pie(slices, labels = lbls,
    main="Simple Pie Chart")
```

```
pct <- round(slices/sum(slices)*100)
lbls2 <- paste(lbls, " ", pct, "%", sep="")
pie(slices, labels=lbls2, col=rainbow(length(lbls2)),
    main="Pie Chart with Percentages")
```

```
library(plotrix)
pie3D(slices, labels=lbls,explode=0.1,
    main="3D Pie Chart ")
```

```
mytable <- table(state.region)
lbls3 <- paste(names(mytable), "\n", mytable, sep="")
pie(mytable, labels = lbls3,
    main="Pie Chart from a Table\n (with sample sizes)")
```

1 将四幅图形组合为一幅

2 为饼图添加比例数值

3 从表格创建饼图

```
par(mfrow=c(2, 2))
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
```

```
pie(slices, labels = lbls,
    main="Simple Pie Chart")
```

```
pct <- round(slices/sum(slices)*100)
lbls2 <- paste(lbls, " ", pct, "%", sep=" ")
pie(slices, labels=lbls2, col=rainbow(length(lbls2))),
    main="Pie Chart with Percentages")
```

```
library(plotrix)
pie3D(slices, labels=lbls,explode=0.1,
      main="3D Pie Chart ")
```

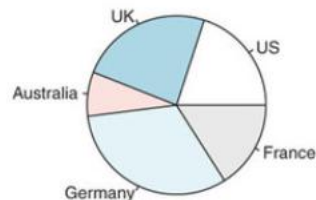
```
mytable <- table(state.region)
lbls3 <- paste(names(mytable), "\n", mytable, sep=" ")
pie(mytable, labels = lbls3,
    main="Pie Chart from a Table\n (with sample sizes)")
```

① 将四幅图形组合为一幅

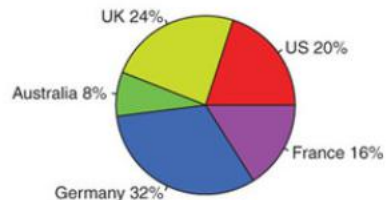
② 为饼图添加比例数值

③ 从表

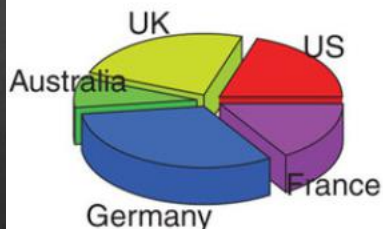
Simple Pie Chart



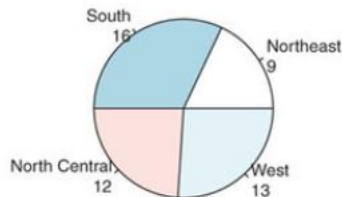
Pie Chart with Percentages



3D Pie Chart



Pie Chart from a Table
(with sample sizes)

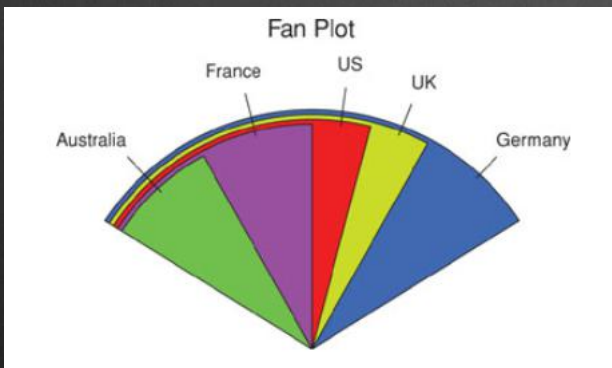




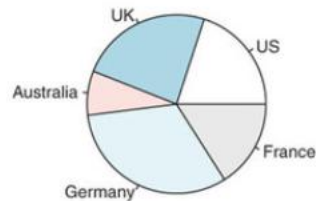
饼图

扇形图

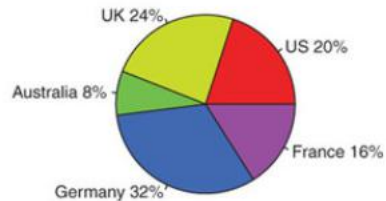
```
1 library(plotrix)
2 slices <- c(10, 12, 4, 16, 8)
3 lbls <- c("US", "UK", "Australia", "Germany", "France")
4 fan.plot(slices, labels = lbls, main="Fan Plot")
```



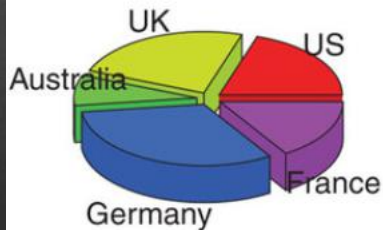
Simple Pie Chart



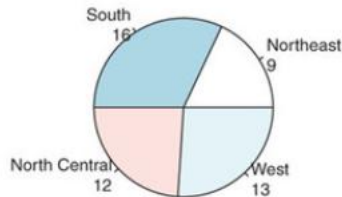
Pie Chart with Percentages



3D Pie Chart



Pie Chart from a Table
(with sample sizes)





直方图

使用如下函数创建直方图：hist(x)

```
par(mfrow=c(2,2))
```

```
hist(mtcars$mpg)
```

1 简单直方图

```
hist(mtcars$mpg,  
     breaks=12,  
     col="red",  
     xlab="Miles Per Gallon",  
     main="Colored histogram with 12 bins")
```

2 指定组数和颜色

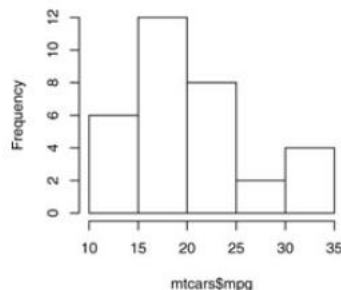
```
hist(mtcars$mpg,  
     freq=FALSE,  
     breaks=12,  
     col="red",  
     xlab="Miles Per Gallon",  
     main="Histogram, rug plot, density curve")  
rug(jitter(mtcars$mpg))  
lines(density(mtcars$mpg), col="blue", lwd=2)
```

3 添加轴须图

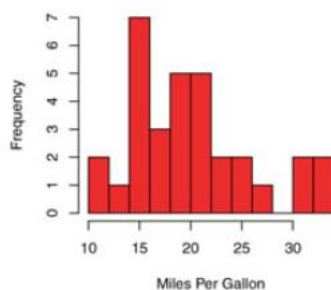
```
x <- mtcars$mpg  
h<-hist(x,  
        breaks=12,  
        col="red",  
        xlab="Miles Per Gallon",  
        main="Histogram with normal curve and box")  
xfit<-seq(min(x), max(x), length=40)  
yfit<-dnorm(xfit, mean=mean(x), sd=sd(x))  
yfit <- yfit*diff(h$mids[1:2])*length(x)  
lines(xfit, yfit, col="blue", lwd=2)  
box()
```

4 添加正态密度曲线和外框

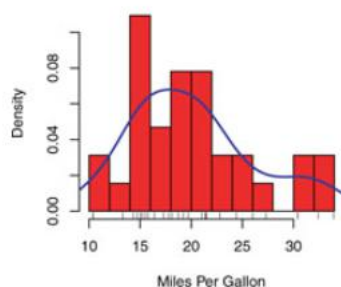
Histogram of mtcars\$mpg



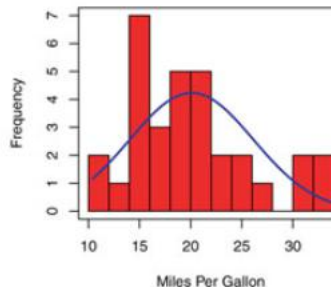
Colored histogram with 12 bins



Histogram, rug plot, density curve

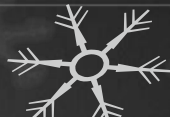


Histogram with normal curve and box





直方图



```
par(mfrow=c(2,2))
```

```
hist(mtcars$mpg)
```

← ❶ 简单直方图

```
hist(mtcars$mpg,
     breaks=12,
     col="red",
     xlab="Miles Per Gallon",
     main="Colored histogram with 12 bins")
```

← ❷ 指定组数和颜色

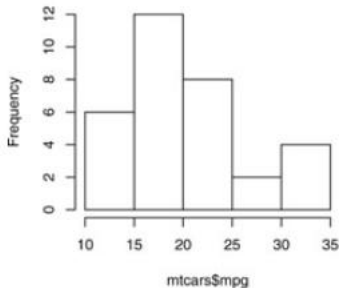
```
hist(mtcars$mpg,
     freq=FALSE,
     breaks=12,
     col="red",
     xlab="Miles Per Gallon",
     main="Histogram, rug plot, density curve")
rug(jitter(mtcars$mpg))
lines(density(mtcars$mpg), col="blue", lwd=2)
```

← ❸ 添加轴须图

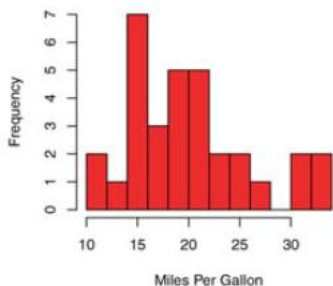
```
x <- mtcars$mpg
h<-hist(x,
        breaks=12,
        col="red",
        xlab="Miles Per Gallon",
        main="Histogram with normal curve and box")
xfit<-seq(min(x), max(x), length=40)
yfit<-dnorm(xfit, mean=mean(x), sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
box()
```

← ❹ 添加正态密度曲线和外框

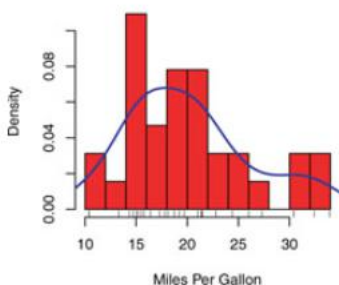
Histogram of mtcars\$mpg



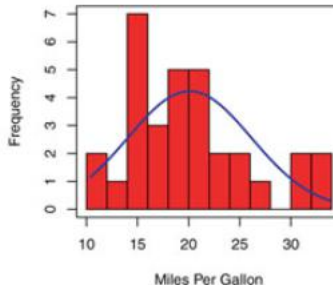
Colored histogram with 12 bins



Histogram, rug plot, density curve



Histogram with normal curve and box



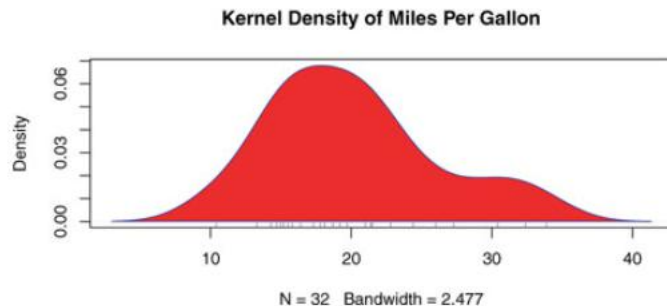
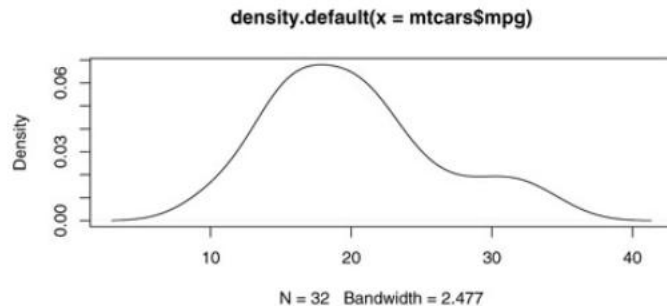


核密度图

🌀 `Plot(density(x))`

其中的x是一个数值型向量。由于`plot()`函数会创建一幅新的图形，所以要向一幅已经存在的图形上叠加一个核密度图，可以使用`lines()`函数。

```
1 par(mfrow=c(2,1))
2 d <- density(mtcars$mpg)
3
4 plot(d)
5
6 d <- density(mtcars$mpg)
7 plot(d, main="Kernel Density of Miles Per Gallon")
8 polygon(d, col="red", border="blue")
9 rug(mtcars$mpg, col="brown")
```





核密度图

🔗 【了解】用核密度图比较两组差异

`sm.density.compare(x, factor)`

```
par(lwd=2)
library(sm)
attach(mtcars)
```

← ① 双倍线条宽度

```
cyl.f <- factor(cyl, levels= c(4,6,8),
               labels = c("4 cylinder", "6 cylinder",
                           "8 cylinder"))
```

← ② 创建分组因子

```
sm.density.compare(mpg, cyl, xlab="Miles Per Gallon")
title(main="MPG Distribution by Car Cylinders")
```

← ③ 绘制密度图

```
colfill<-c(2:(1+length(levels(cyl.f))))
legend(locator(1), levels(cyl.f), fill=colfill)
```

← ④ 通过鼠标单击添加图例

```
detach(mtcars)
```



核密度图

```
labels = c("4 cylinder", "6 cylinder",  
           "8 cylinder"))
```

2 创建分组因子

```
sm.density.compare(mpg, cyl, xlab="Miles Per Gallon")  
title(main="MPG Distribution by Car Cylinders")
```

3 绘制密度图

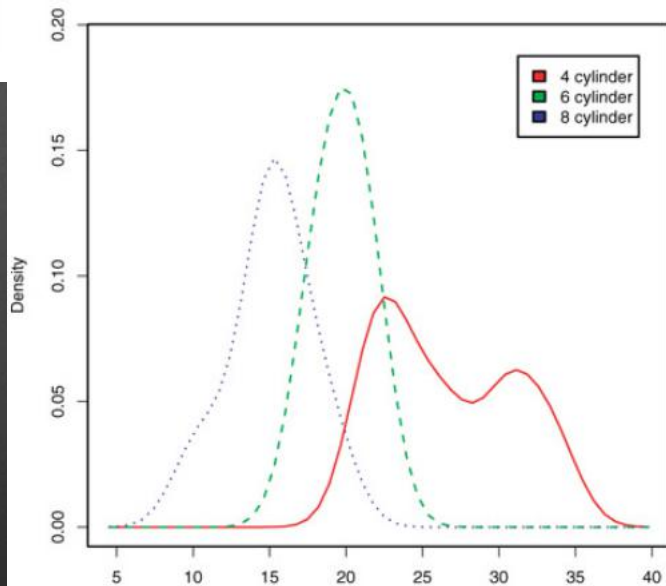
```
colfill<-c(2:(1+length(levels(cyl.f))))  
legend(locator(1), levels(cyl.f), fill=colfill)
```



4 通过鼠标单击添加图例

```
detach(mtcars)
```

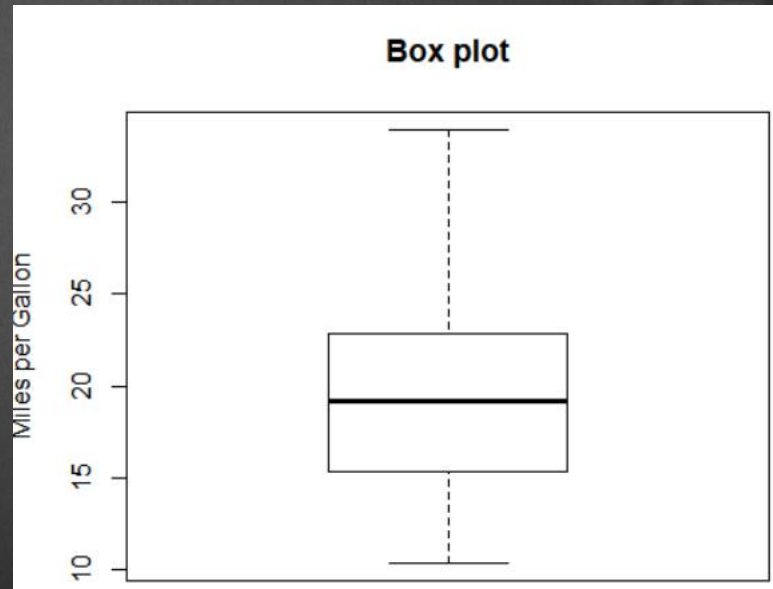
MPG Distribution by Car Cylinders





箱线图

❧ `boxplot(mtcars$mpg, main="Box plot",
ylab="Miles per Gallon")`



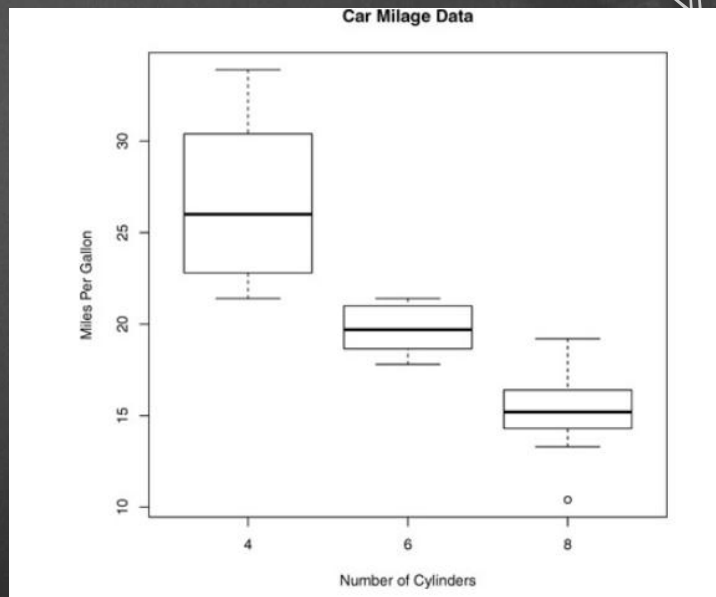


箱线图

使用并列箱线图进行跨组比较

`boxplot(formula, data=dataframe)`

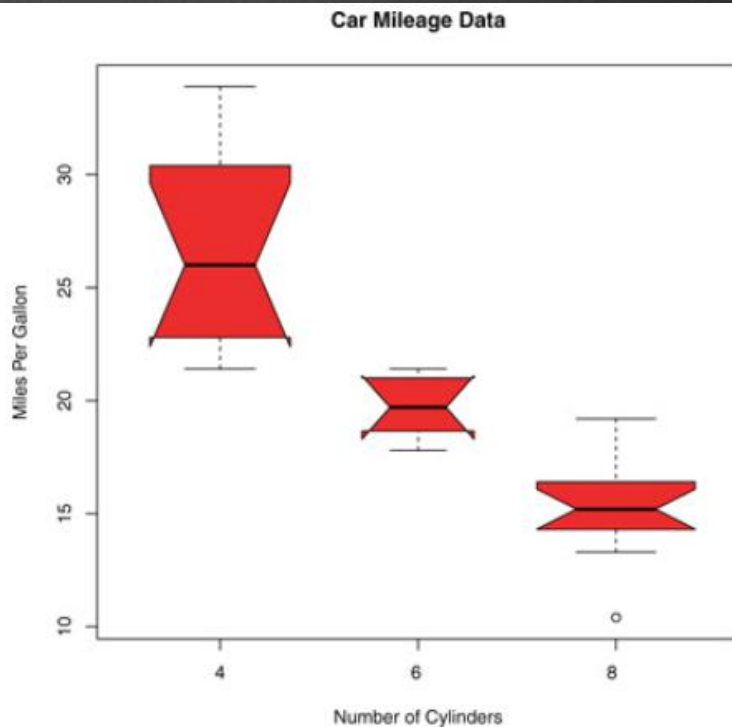
```
1 boxplot(mpg ~ cyl, data=mtcars,  
2         main="Car Mileage Data",  
3         xlab="Number of Cylinders",  
4         ylab="Miles Per Gallon")
```





箱线图

```
1 boxplot(mpg ~ cyl, data=mtcars,  
2         notch=TRUE,  
3         varwidth=TRUE,  
4         col="red",  
5         main="Car Mileage Data",  
6         xlab="Number of Cylinders",  
7         ylab="Miles Per Gallon")
```





箱线图

【了解】两个交叉因子的箱线图

```
mtcars$cyl.f <- factor(mtcars$cyl,  
  levels=c(4,6,8),  
  labels=c("4", "6", "8"))
```

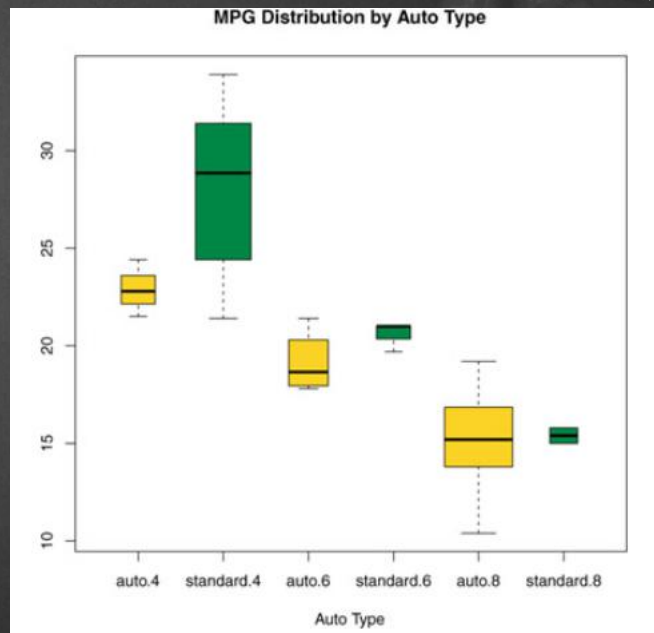
← 创建汽缸数量的因子

```
mtcars$am.f <- factor(mtcars$am,  
  levels=c(0,1),  
  labels=c("auto", "standard"))
```

← 创建变速箱类型的因子

```
boxplot(mpg ~ am.f * cyl.f,  
  data=mtcars,  
  varwidth=TRUE,  
  col=c("gold", "darkgreen"),  
  main="MPG Distribution by Auto Type",  
  xlab="Auto Type")
```

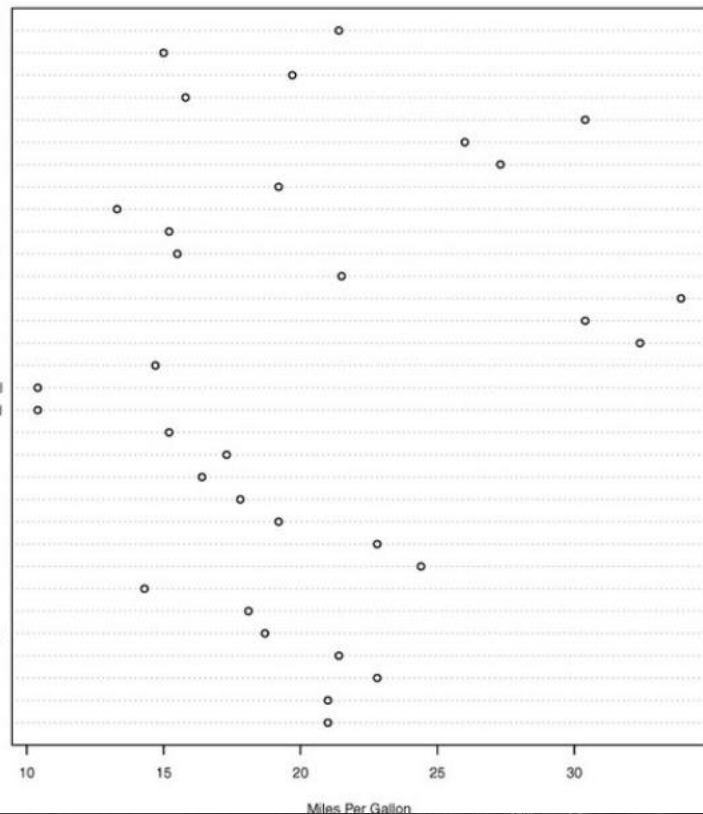
← 生成箱线图





dotchart(x, labels=)

Volvo 142E
Maserati Bora
Ferrari Dino
Ford Pantera L
Lotus Europa
Porsche 914-2
Fiat X1-9
Pontiac Firebird
Camaro Z28
AMC Javelin
Dodge Challenger
Toyota Corona
Toyota Corolla
Honda Civic
Fiat 128
Chrysler Imperial
Lincoln Continental
Cadillac Fleetwood
Merc 450SLC
Merc 450SL
Merc 450SE
Merc 280C
Merc 280
Merc 230
Merc 240D
Duster 360
Valiant
Hornet Sportabout
Hornet 4 Drive
Datsun 710
Mazda RX4 Wag
Mazda RX4

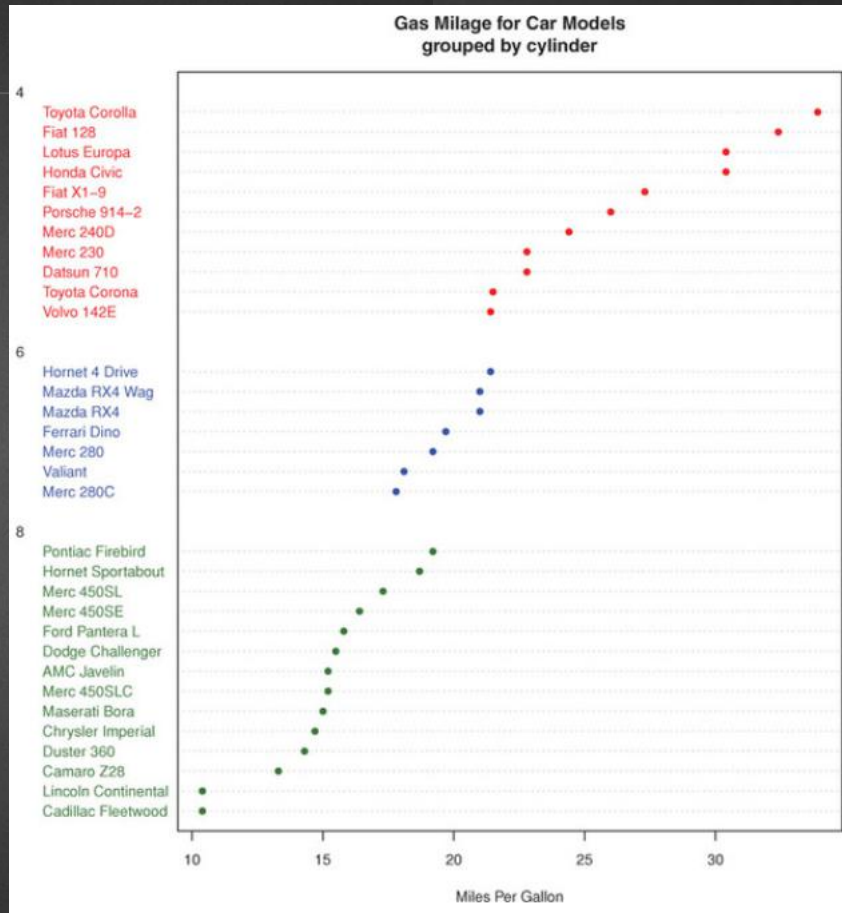


```
1 dotchart(mtcars$mpg, labels=row.names(mtcars), cex=.7,  
2           main="Gas Mileage for Car Models",  
3           xlab="Miles Per Gallon")
```



❧ 示例

```
1 x <- mtcars[order(mtcars$mpg),]  
2 x$cyl <- factor(x$cyl)  
3 x$color[x$cyl==4] <- "red"  
4 x$color[x$cyl==6] <- "blue"  
5 x$color[x$cyl==8] <- "darkgreen"  
6 dotchart(x$mpg,  
7         labels = row.names(x),  
8         cex=.7,  
9         groups = x$cyl,  
10        gcolor = "black",  
11        color = x$color,  
12        pch=19,  
13        main = "Gas Mileage for Car Models\ngrouped by cylinder",  
14        xlab = "Miles Per Gallon")
```





小结

- ❧ 图形的创建和保存
- ❧ 自定义符号、线条、颜色和坐标轴
- ❧ 标注文本和标题
- ❧ 控制图形维度
- ❧ 组合多个图





小结

- ❧ 条形图、箱线图和点图
- ❧ 饼图和扇形图
- ❧ 直方图与和密度图





Thankyou !