

R基本绘图

温灿红

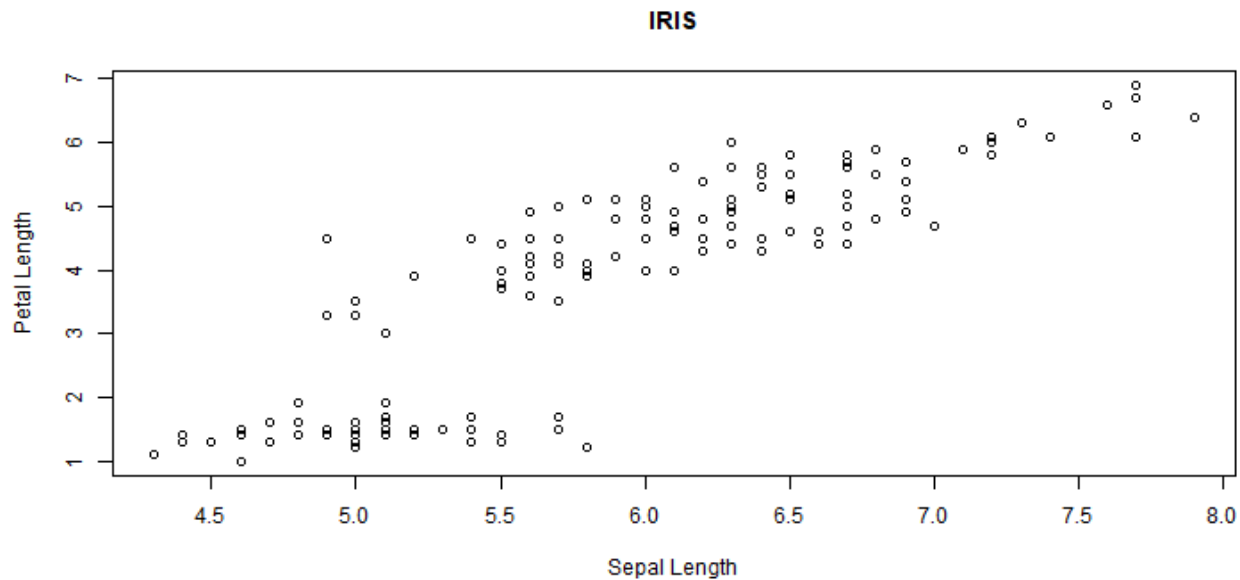
中国科学技术大学管理学院

大纲

- 我的第一张图
 - 绘图
 - 保存图片
- 常用高级图形
 - 条形图
 - 饼图
 - 箱线图
 - 直方图和核密度图
 - 散点图
 - 折线图
 - 其它
- 图形定制
 - 低级图形函数
 - 图形参数
 - 一页多图

我的第一张图

```
plot(iris$Sepal.Length, iris$Petal.Length, xlab = "Sepal Length", ylab = "Petal Length", main =
```



保存图片

- 鼠标操作
- 代码实现
 - 在画图前用 `postscript()`, `pdf()`, `tiff()`, `jpeg()`, `png()` 等函数制定输出的文件名, 文件大小等。
 - 在画图后用 `dev.off()` 函数关闭当前画图设备并生成输出文件。
 - 通过 `width` 和 `height` 等控制图的大小。

```
pdf("myfig.pdf", width = 10, height = 8)
plot(iris$Sepal.Length, iris$Petal.Length, xlab = "Sepal Length", ylab = "Petal Length", main =
dev.off()
```

```
## png
## 2
```

R 的图形支持

- R 支持多种图形系统，其中传统的有base部分的图形（graphics包），比较新的有ggplot2等。
- 这一节介绍base图形，包括两类图形函数：
 - 高级绘图函数：直接针对某一绘图任务作出完整图形
 - 低级绘图函数：在已有的图形上添加元素、控制图形外观、配色和布局等等

高级绘图函数

plot函数

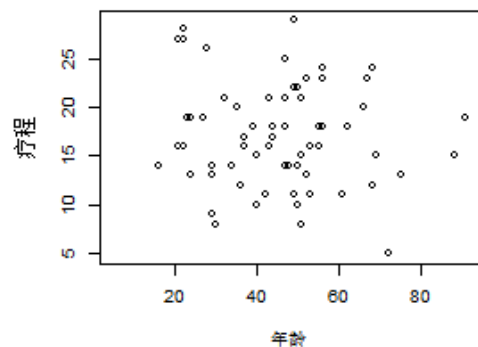
- plot函数是graphics包中最重要的高级绘图函数，可以绘制多种图形
- plot函数是一个泛函数，给函数传递不同类型的数据，会绘制不同的图形。

plot 绘制不同的图形

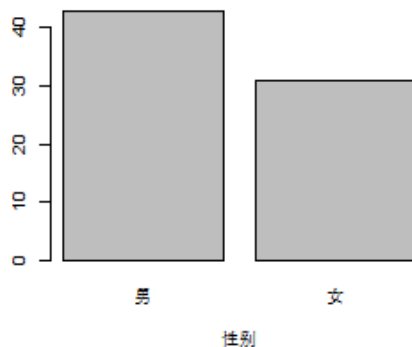
```
d <- read.csv("covid19.csv", header = TRUE, na.strings = "", row.names = "序号")
d$性别 <- as.factor(d$性别)
d$分型 <- as.factor(d$分型)
par(mfrow=c(2,2))
plot(d$年龄, d$疗程, ylab = "疗程", xlab = "年龄", main = "(a) 散点图")
plot(d$性别, xlab = "性别", main = "(b) 条形图")
plot(年龄~性别, data = d, main = "(c) 箱线图")
plot(性别~分型, data = d, main = "(d) 马赛克图")
```


plot 绘制不同的图形

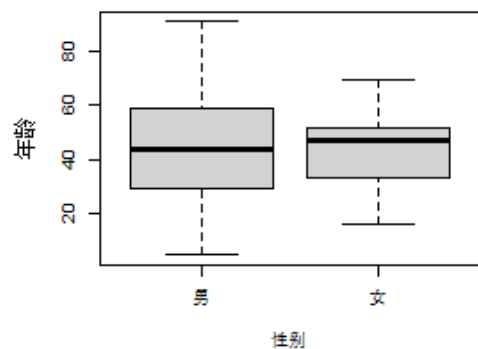
(a) 散点图



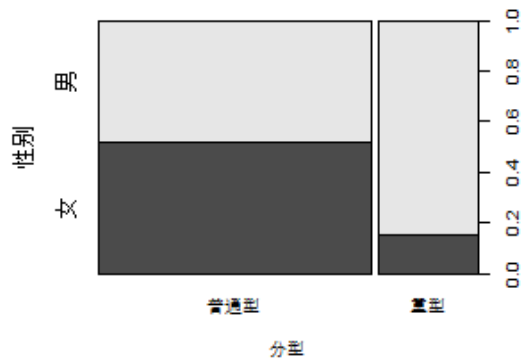
(b) 条形图



(c) 箱线图



(d) 马赛克图



常用的图形

- 单变量：
 - 因子型：饼图、条形图
 - 数值型：箱线图、直方图、核密度估计图
- 两变量：
 - 均为因子型：堆叠条形图、并排条形图
 - 一个数值一个因子型：箱线图
 - 均为数值型：散点图、QQ图
- 多变量：
 - pairwise plots
 - 三维图(三维曲面图、等高线图、色块图)

条形图

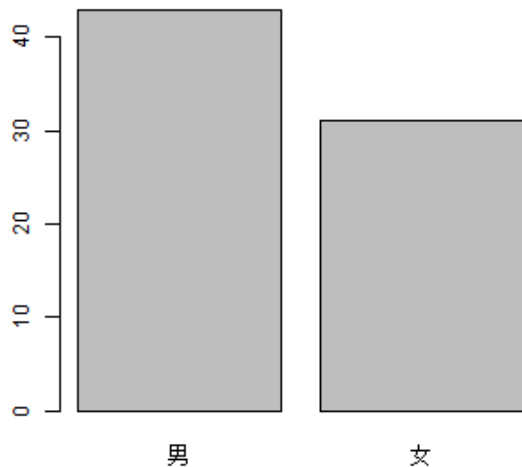
条形图

- 条形图是常用的描述分类变量的可视化方法，用来展示各分类的频数。
- 可展示单个分类变量的频数，也可展示两个变量的频数分布，即列联表。
- `barplot()` 函数可以对这样的频数结果绘制条形图。
 - `main =`: 标题
 - `col =`: 不同条形的颜色。函数 `colors()` 可以返回R中定义的用字符串表示的六百多种颜色名字
 - `horiz =` 是否需要横排，默认为FALSE
 - `beside =` 是否为并排条形图，默认为FALSE，即堆叠条形图。

条形图的例子（一）

- 回顾之前讲过的数据集 “covid19” ， 里面有“分型”， “性别” 等分类变量。
- 用 `table(d[, “性别”])` 可以统计数据框中变量 “性别” 的不同取值的频数。

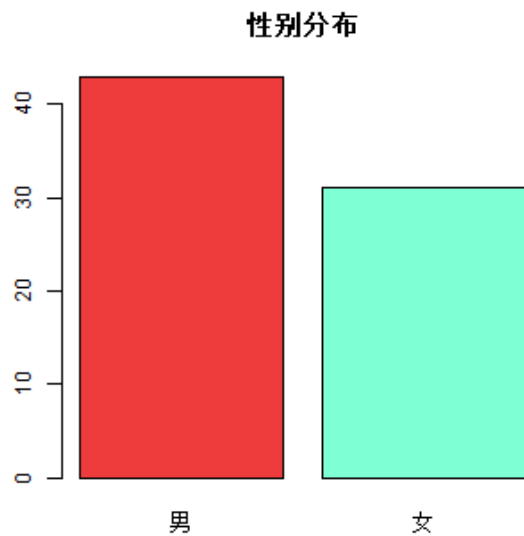
```
d <- read.csv("covid19.csv", header = TRUE, na.strings = "", row.names = "序号")  
res <- table(d[, "性别"])  
barplot(res)
```



条形图的例子（二）

- 增加颜色

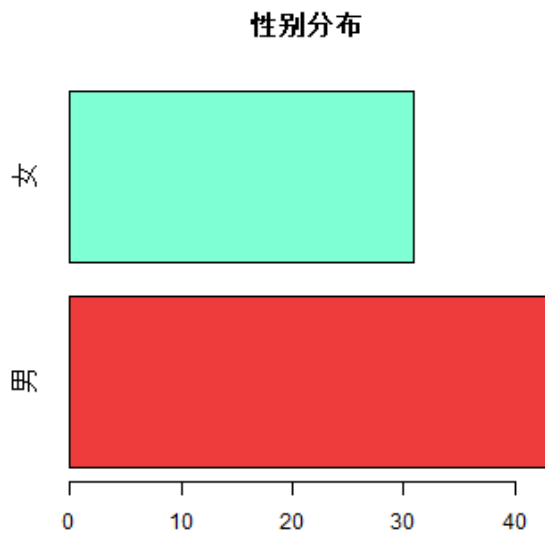
```
barplot(res, main="性别分布", col=c("brown2", "aquamarine1"))
```



条形图的例子（三）

- 横排的条形图

```
barplot(res, main="性别分布", col=c("brown2", "aquamarine1"), horiz = TRUE)
```

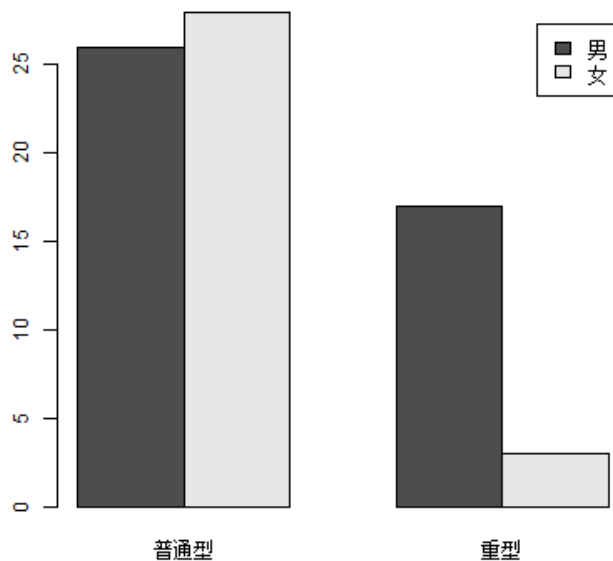


列联表的条形图

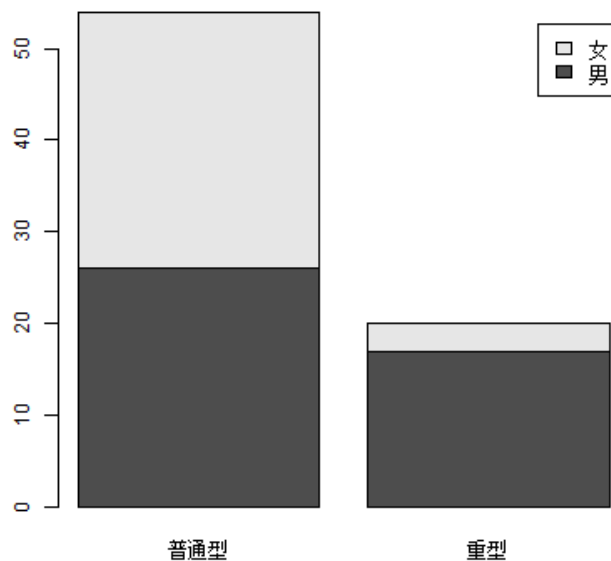
- 用 `table` 函数按性别和年龄长幼交叉分组

```
res2 <- with(d, table(性别, 分型))
```

```
barplot(res2, beside=TRUE, legend=TRUE)
```



```
barplot(res2, legend = TRUE)
```

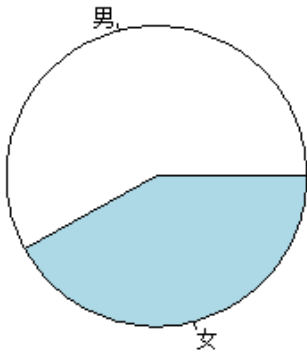


饼图

饼图

- 饼图可以看成一种特殊的条形图，它是将条形图的y值变成了角度。
- pie

```
pie(res)
```

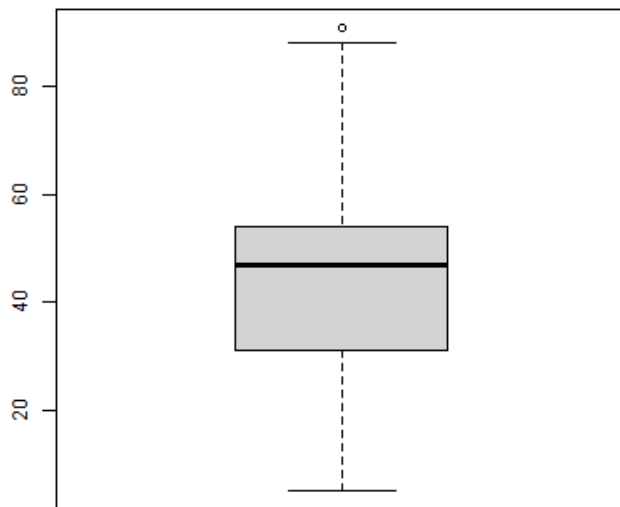


箱线图

箱线图 (boxplot)

- 箱线图可以简洁地表现变量分布，主要利用5个点来绘制。
- 中间粗线是中位数;
- 盒子上下边缘是 $\frac{3}{4}$ 和 $\frac{1}{4}$ 分位数;
- 两条触须线延伸到取值区域的边缘。

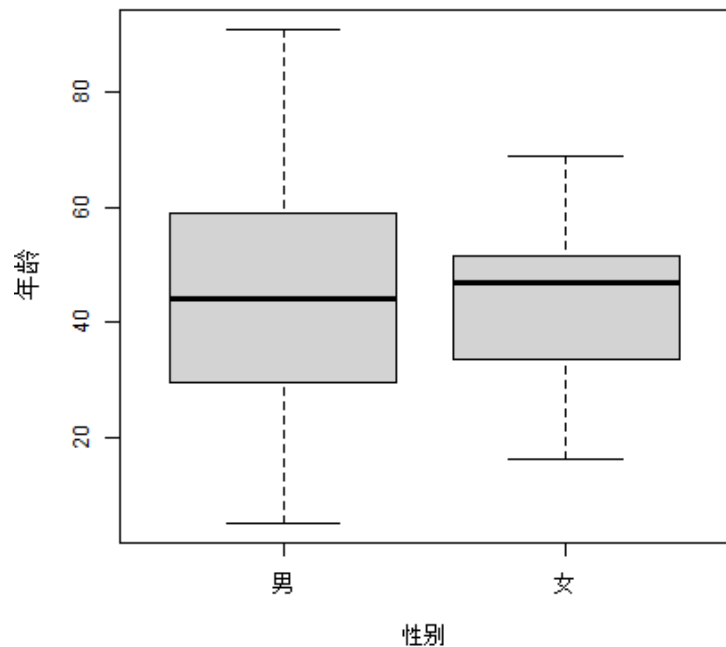
```
boxplot(d[, "年龄"])
```



多组箱线图

- 盒形图可以很容易地比较两组或多组数据的分布情况。

```
boxplot(年龄 ~ 性别, data = d)
```



```
# with(d, boxplot(年龄 ~ 性别))
```

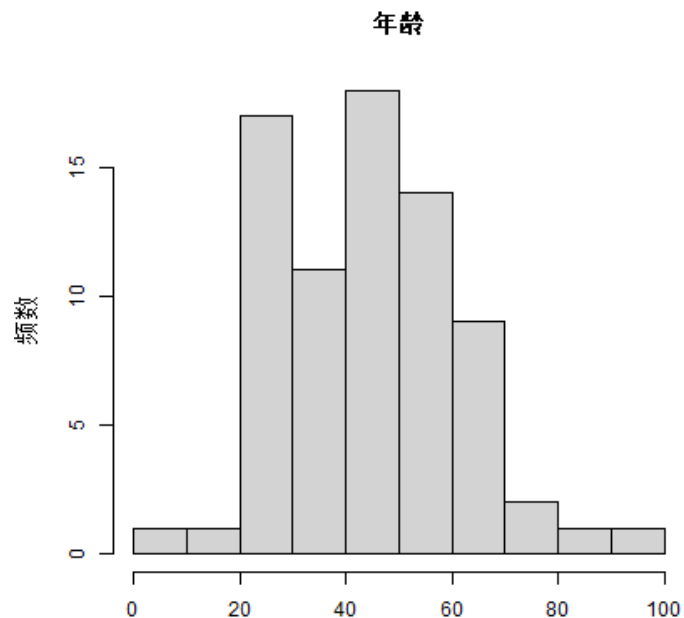
直方图和密度估计图

直方图和核密度图

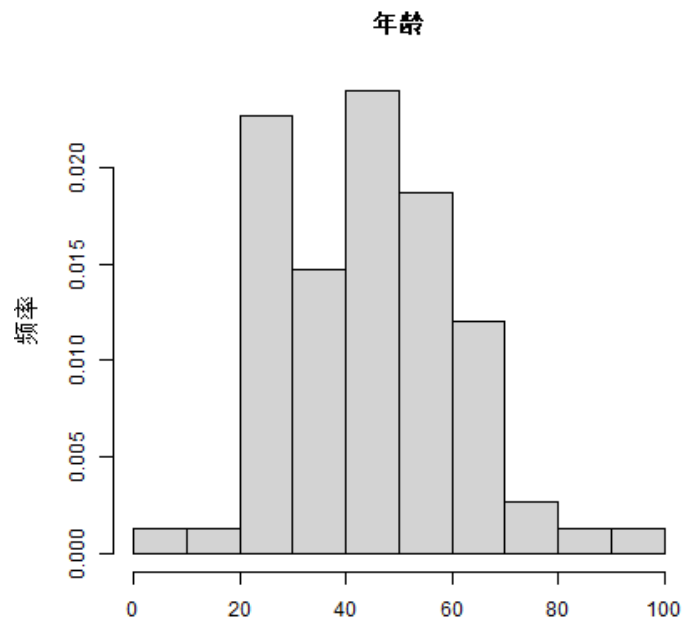
- 直方图和核密度图是反应数值变量分布的图形。
- `hist(x)` 对 `x` 作频数直方图，直方图自动等距分段，纵坐标为每段的样本点个数。
 - `freq = FALSE` 改变y轴的计数为比例
 - `main =` : 标题
 - `xlab =` , `ylab =` : x/y轴标签
 - `xlim =` , `ylim =` : x或y轴的绘图范围
- `density(x)` 对数据x进行核密度估计，需配合`plot()`来进行画图。

直方图的例子

```
hist(d[, "年龄"], main = '年龄', xlab = '',
```

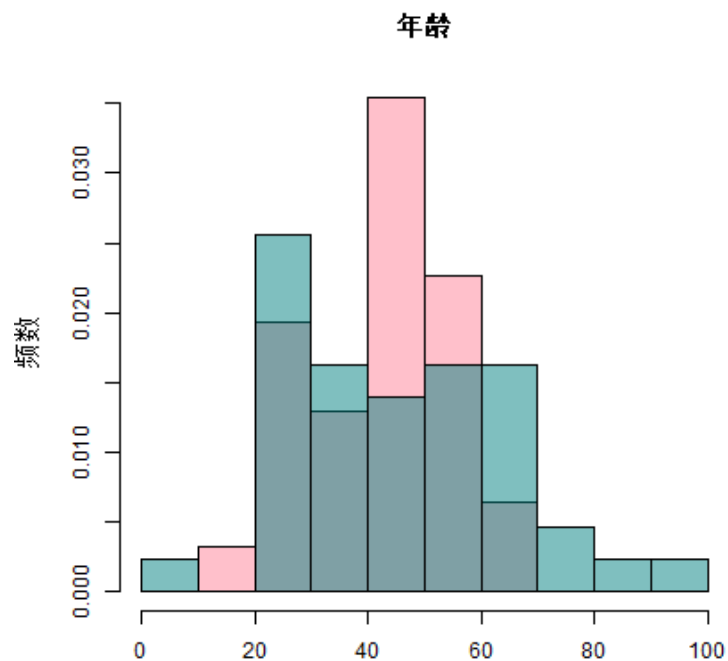


```
hist(d[, "年龄"], freq = FALSE, main = '
```



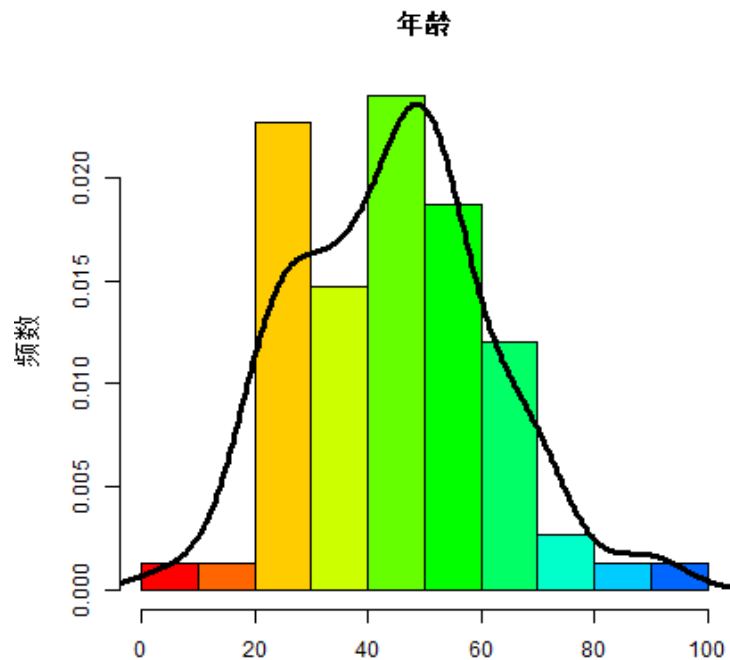
直方图的叠加

```
idx <- d$性别=="女"  
hist(d[idx, "年龄"], freq = FALSE, col = "pink", main = '年龄',  
      xlab = '', ylab = '频数', xlim = c(0, 100))  
hist(d[!idx, "年龄"], freq = FALSE, col = rgb(0, 0.5, 0.5), add = TRUE)
```



密度估计图

```
hist(d[, "年龄"], freq = FALSE, col = rainbow(15), main = '年龄',  
      xlab = '', ylab = '频数')  
lines(density(d[, "年龄"]), lwd = 3, col = "black")
```



散点图

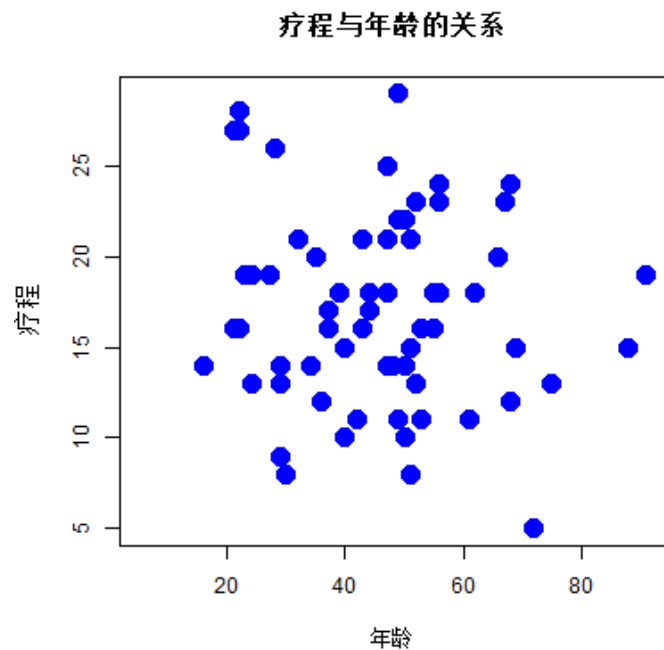
散点图 (scatterplot)

- 散点图是将所有的数据以点^点的形式展现在坐标系上，用来显示变量之间的相互影响程度，点的位置有变量的数值决定。
- `plot(x, y)`或者`plot(y ~ x)`
 - `pch`：点的形状
 - `col`：点的颜色，输入可以是数字，也可以是颜色的名字
 - `cex`：点的大小
 - 以上参数可以按组进行设置，即先以上述参数作为分组的依据对数据分组，然后绘制图形。通过属于一个与点相同大小的向量来进行控制。

散点图的例子（一）

- 下图展示的是数据框 d 中病人的疗程长短和年龄的关系。

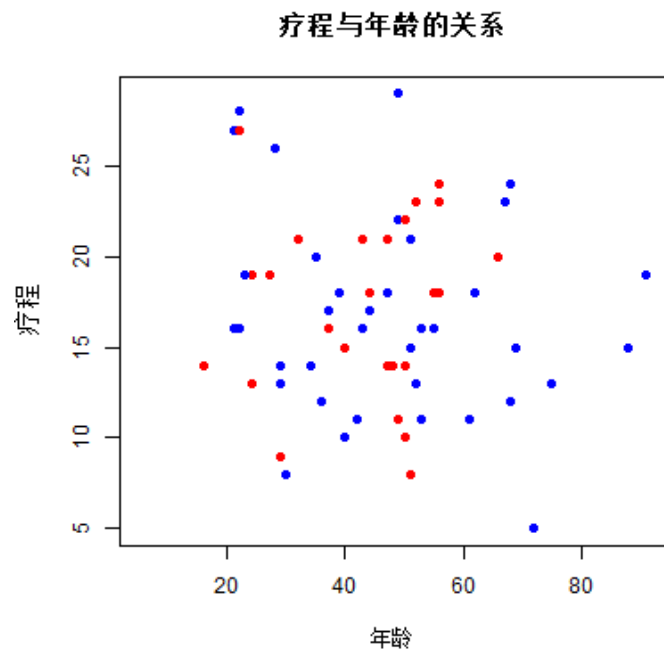
```
plot(疗程 ~ 年龄, data = d, main = "疗程与年龄的关系",  
     pch= 16, col = "blue", cex = 2)
```



散点图的例子（二）

- 用颜色区分性别

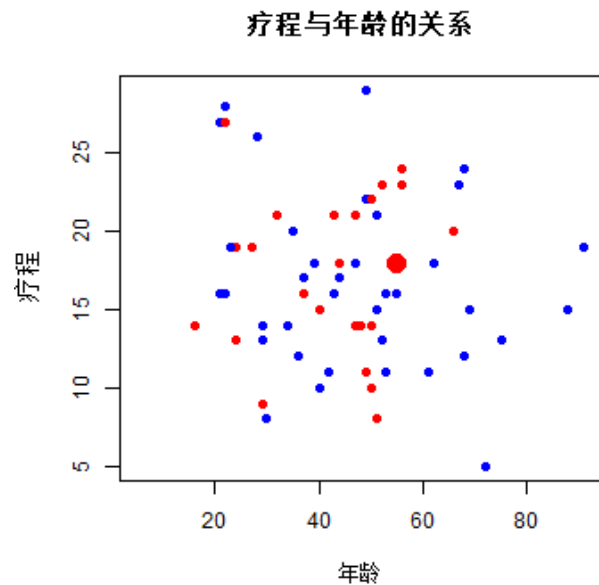
```
plot(疗程 ~ 年龄, main = "疗程与年龄的关系", data = d,  
     col=ifelse(性别=='男', 'blue', 'red'), pch=16)
```



散点图的例子（三）

- 用气泡大小表现第三维（入院时间的早晚）

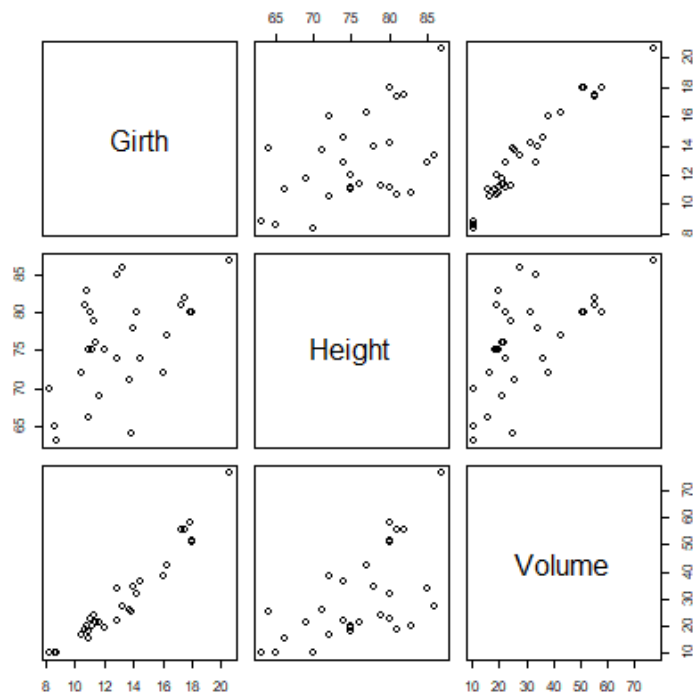
```
plot(疗程 ~ 年龄, main = "疗程与年龄的关系", data = d,  
     col=ifelse(性别=='男', 'blue', 'red'), pch=16,  
     cex = 1 + (入院时间- min(入院时间))/(max(入院时间)- min(入院时间)))
```



散点图矩阵

- `pairs` 函数对矩阵或数据框的各列两两做散点图，构成一个散点图矩阵

```
pairs(trees)
```



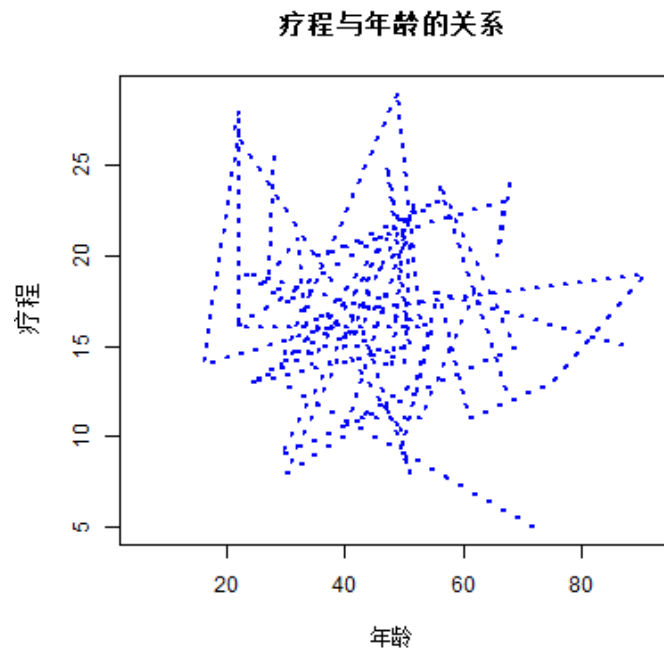
折线图

折线图

- 折线图用于显示数据的变化趋势，给定曲线上一组坐标点时，把相邻点用直线连接起来。
- 在 `plot` 函数中使用 `type = 'l'` 参数可以作折线图。
- `lwd` 指定线宽度，`lty` 指定虚线

折线图的例子

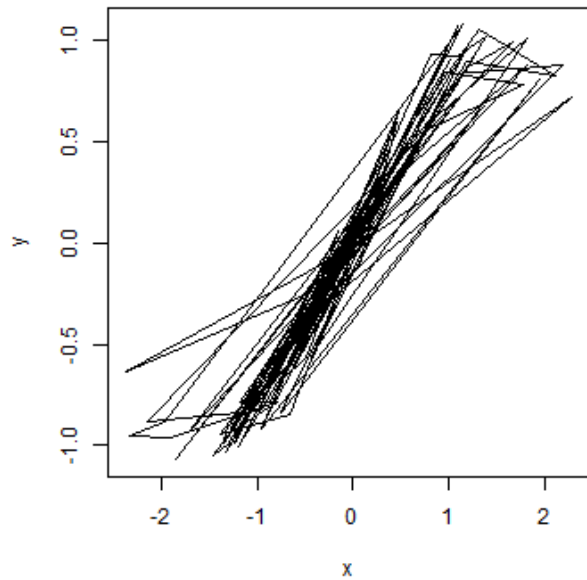
```
plot(疗程 ~ 年龄, type = "l", data = d, main = "疗程与年龄的关系",  
     pch= 16, col = "blue", cex = 2, lwd=2, lty=3)
```



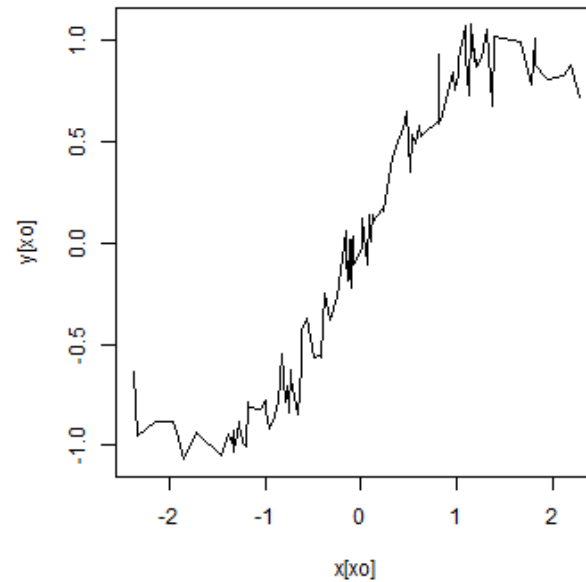
折线图的例子

```
x <- rnorm(100); y <- sin(x)+rnorm(100, 0, 0.1)
xo <- order(x)
```

```
plot(x, y, type = "l")
```



```
plot(x[xo], y[xo], type = "l")
```



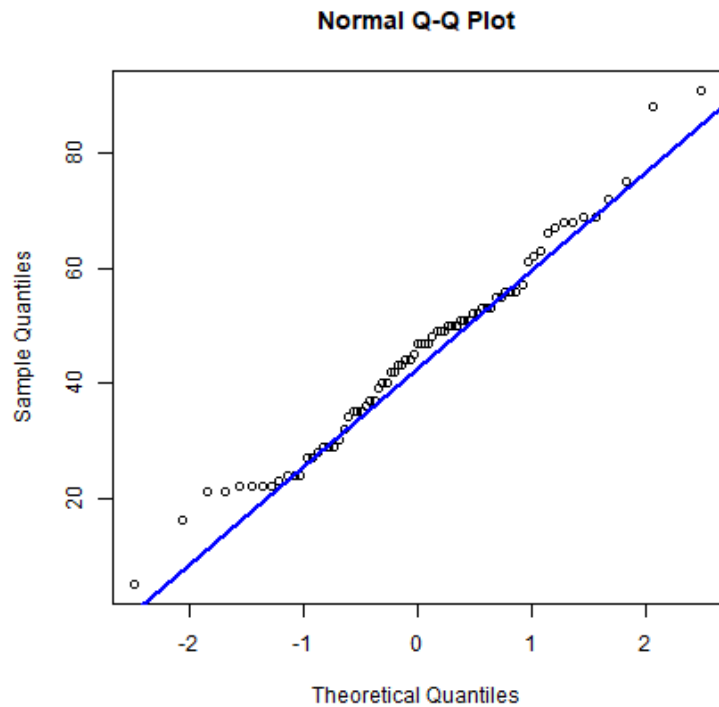
其它

正态QQ图

- 从直方图与箱线图可以模糊地判断可能的分布类型。
- 为了确定数值型变量是否来自某个理论分布 F ，如正态分布，可以作QQ图 (quantile-quantile plot)。
- 大概原理：
 - 设有 n 个来自于正态分布总体 $N(\mu, \sigma^2)$ 的观察值 x_1, \dots, x_n ，记 t_i 为标准正态分布的 i/n 分位数，则 $x_{(i)}$ 为总体的 i/n 分位数的估计。
 - 理论上应有 $x_{(i)} \rightarrow \mu + \sigma t_i$ 。
 - 以 $(t_i, x_{(i)})$ 作为横坐标画散点图应该接近呈现为截距为 μ ，斜率为 σ 的一条直线。

正态QQ 图的例子

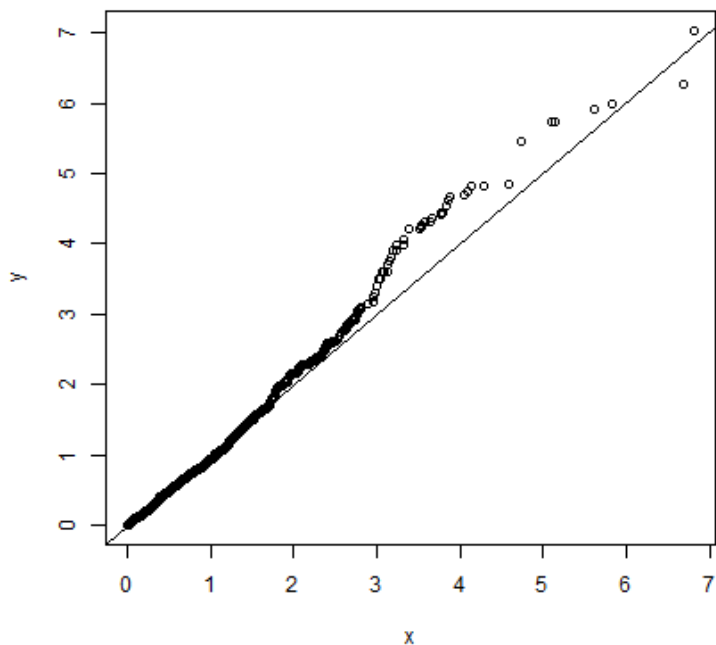
```
qqnorm(d[, "年龄"])  
qqline(d[, "年龄"], lwd=2, col='blue')
```



QQ 图

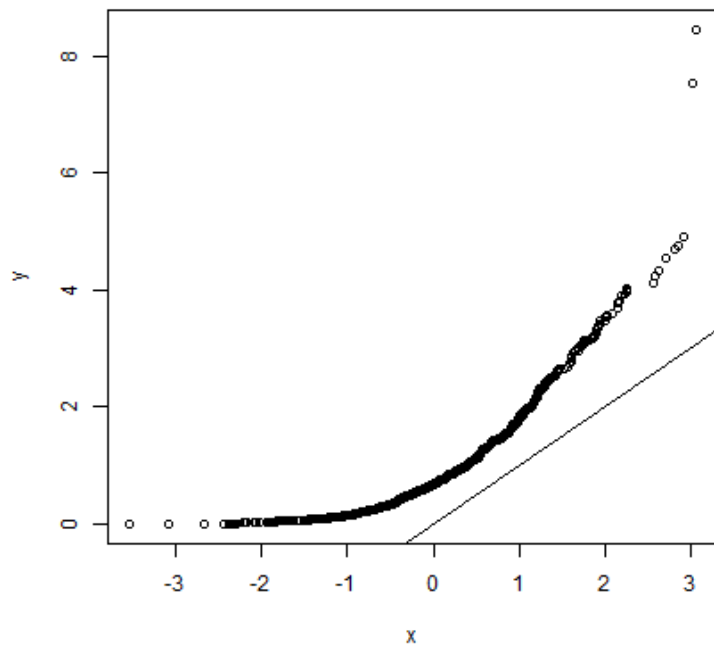
- QQ图也可以用来判别两个数据是否来自于同一个分布，用到`qqplot()`

```
x <- rexp(1000); y <- rexp(1000)
qqplot(x, y)
abline(a=0, b=1)
```



QQ图的反例

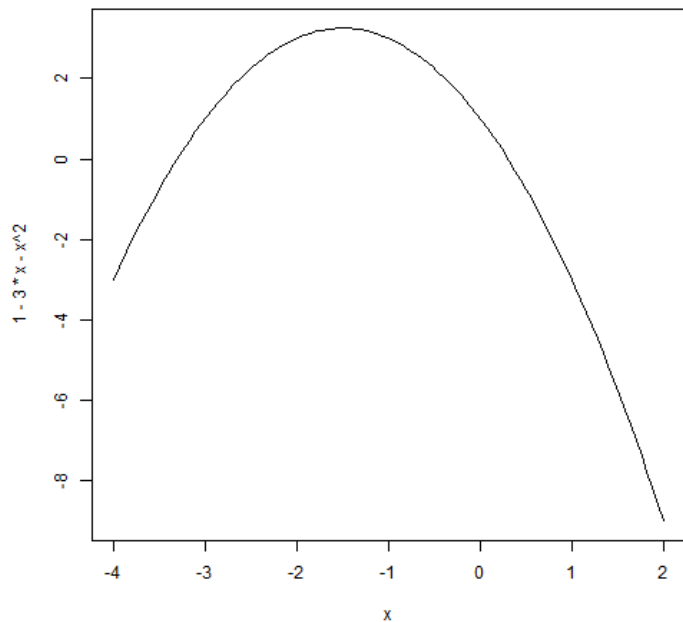
```
x <- rnorm(1000); y <- rexp(1000)  
qqplot(x, y)  
abline(a=0, b=1)
```



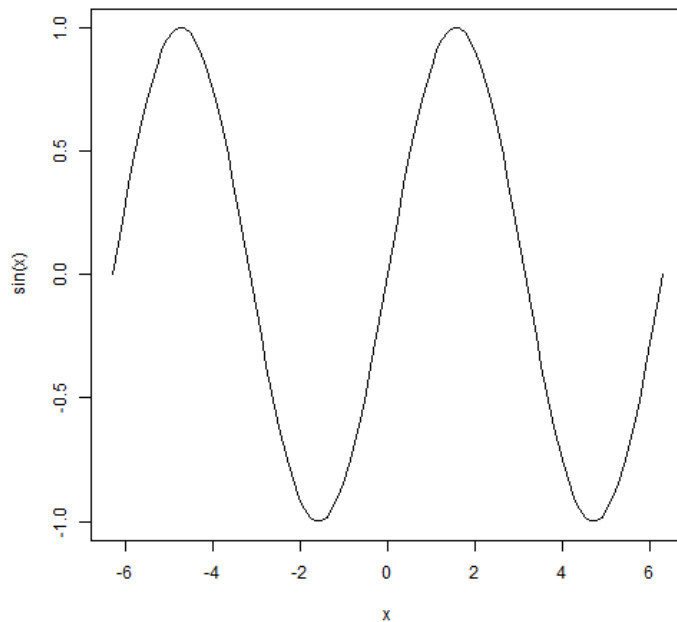
曲线图

- `curve(expr, from, to)` 可以对以 x 为自变量的表达式做函数曲线，或者对某个函数作函数曲线。如

```
curve(1 - 3*x - x^2, -4, 2)
```



```
curve(sin, -2*pi, 2*pi)
```



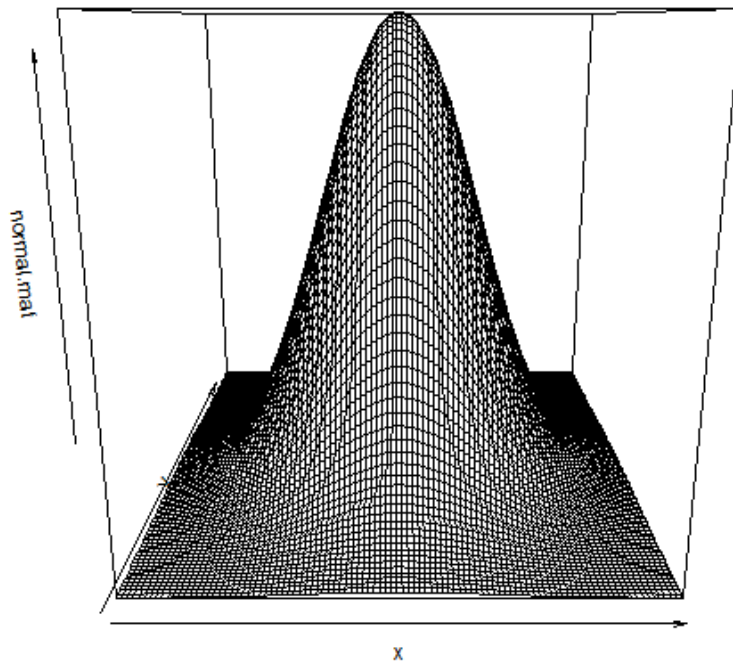
三维图（一）

- 用 `persp` 函数作三维曲面图, `contour` 作等值线图, `image` 作色块图。
- 坐标 `x` 和 `y` 构成一张平面网格, 数据 `z` 是包含 `z` 坐标的矩阵, 每行对应一个横坐标, 每列对应一个纵坐标。

```
x <- seq(-3, 3, length = 100)
y <- x
phi <- dnorm(x)
normal.mat <- phi %o% phi
```

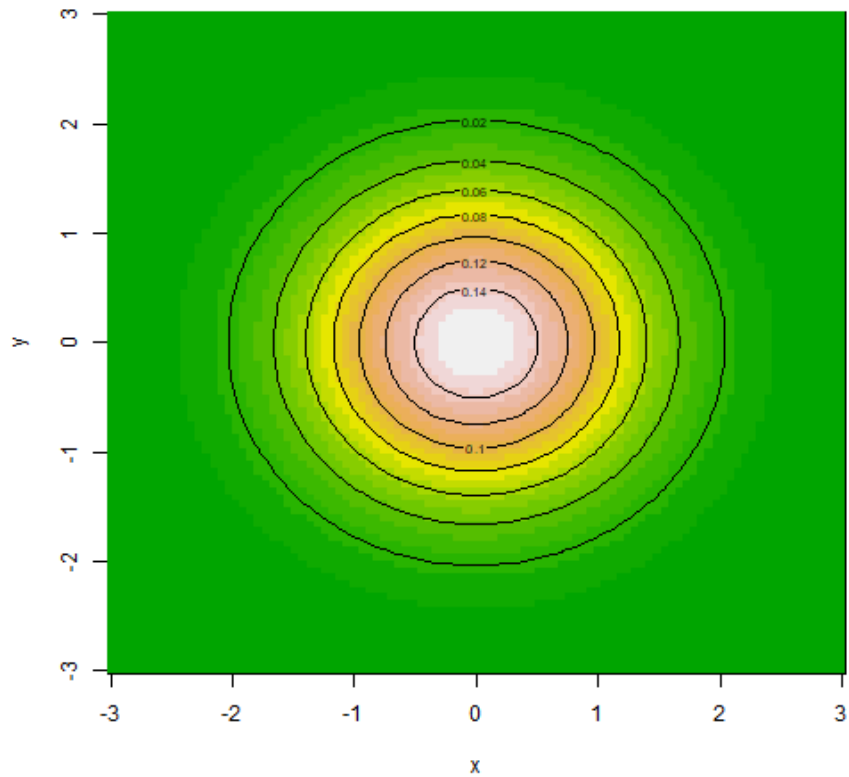
三维图 (二)

```
persp(x, y, normal.mat)
```



三维图 (三)

```
image(x, y, normal.mat, col=terrain.colors(20))  
contour(x, y, normal.mat, add = TRUE)
```



三维散点图

- `rgl`包里的`plot3d`函数可绘制三维散点图，可以通过点击和拖动旋转图形，也可以通过鼠标滚轮来缩放图形。

```
library(rgl)
x <- sort(rnorm(1000))
y <- rnorm(1000)
z <- rnorm(1000) + atan2(x, y)
plot3d(x, y, z, col = rainbow(1000))
```

图形定制

低级图形函数

低级图形函数

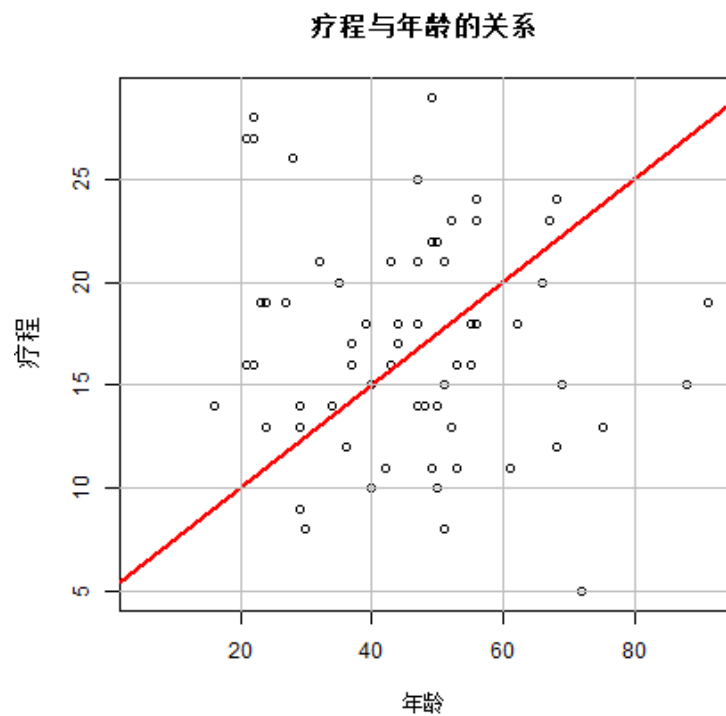
- `barplot`, `plot`, `hist`, `boxplot` 等函数可以直接绘制一幅新图形，称为高级图形函数。
- `abline`, `points`, `lines` 等函数在已有图形中添加新内容，添加的元素包括坐标轴、图例、文本注释、标题等，称为低级图形函数。
- 低级图形也包含 `legend()`, `axis()`, `text()`, `mtext()`, `par()` 等给图形添加图标、坐标系、文字和改变图形参数等函数。

添加线和点

- 用 `abline()` 函数在图中增加直线 `lines()` 函数在图中增加折线。
 - `a` 指定截距
 - `b` 指定斜率
 - `v` 指定横坐标画竖线
 - `h` 指定纵坐标画水平线
- `points()` : 在图中增加散点
- `lines()` : 在图中增加折线

abline() 的例子

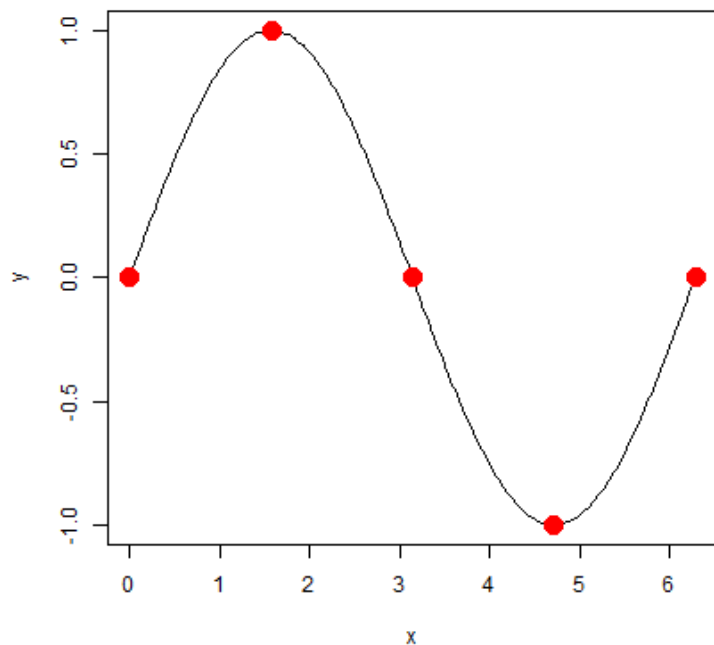
```
plot(疗程 ~ 年龄, data = d, main = "疗程与年龄的关系")  
abline(5, 1/4, col = "red", lwd = 2)  
abline(v = seq(20, 80, 20), col = "gray")  
abline(h = seq(5, 30, 5), col = "gray")
```



points()

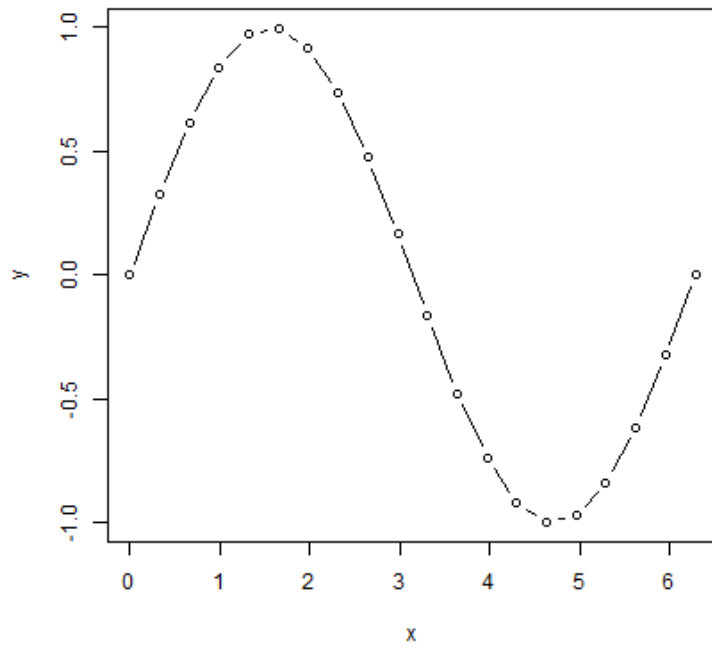
- 用 `points()` 函数在图中增加散点图

```
x <- seq(0, 2*pi, length=200); y <- sin(x)
plot(x, y, type='l')
points((0:4)*pi/2, sin((0:4)*pi/2), col = "red", pch = 16, cex = 2)
```



与 `plot(type= "b")` 的区别

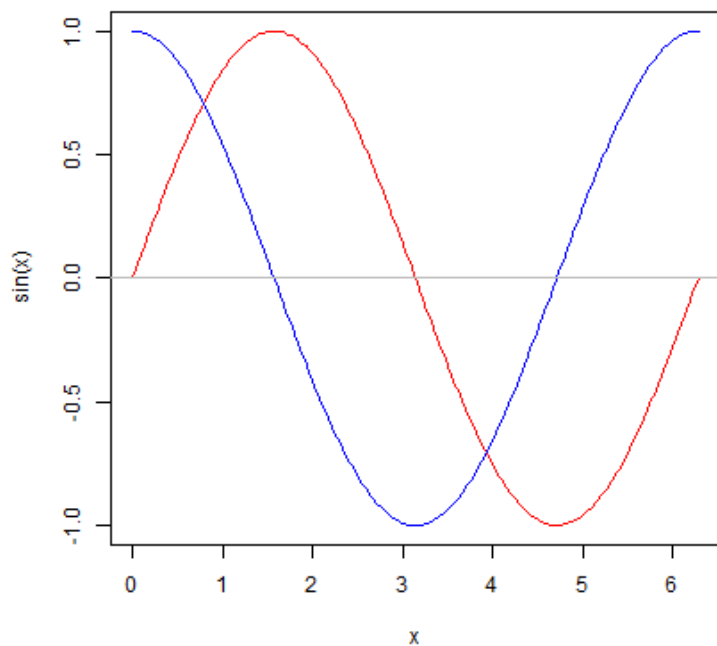
```
x <- seq(0, 2*pi, length=20); y <- sin(x)
plot(x, y, type='b')
```



lines()

- 用 `lines()` 函数在图中增加折线

```
x <- seq(0, 2*pi, length=200);  
plot(x, sin(x), type='l', col = "red")  
lines(x, cos(x), col= "blue")  
abline(h = 0, col = "gray")
```



添加图例

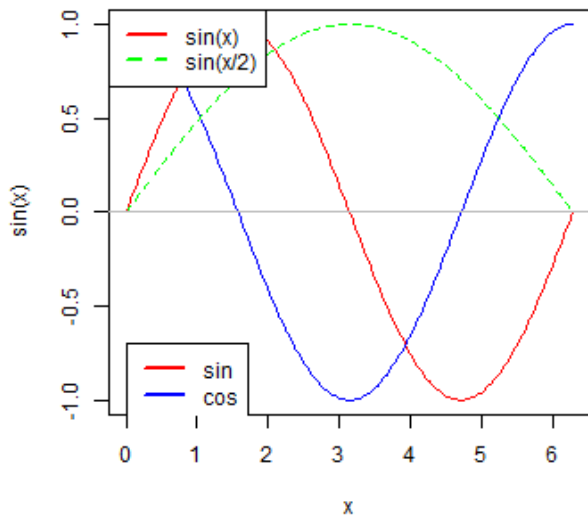
- 用 `legend()` 函数增加图例，比如上个正弦、余弦曲线图，为了区分两条曲线，用如下语句增加图例：

```
legend(0, -0.7, col=c("red", "blue"), lty=c(1,1), lwd=c(2,2), legend=c("sin", "cos"))
```

- `legend()` 前两个参数是图例的横纵坐标位置，也可以不指定位置，只是说明其在图中的位置，如 `top`, `bottom`, `topleft`, `center` 等

legend() 的例子

```
x <- seq(0, 2*pi, length=200);  
plot(x, sin(x), type='l', col = "red")  
lines(x, cos(x), col= "blue")  
lines(x, sin(x/2), col="green", lty=2)  
abline(h = 0, col = "gray")  
legend(0, -0.7, col=c("red", "blue"), lty=c(1,1), lwd=c(2,2), legend=c("sin", "cos"))  
legend("topleft", col=c("red", "green"), lty=c(1,2), lwd=c(2,2), legend=c("sin(x)", "sin(x/2)"))
```

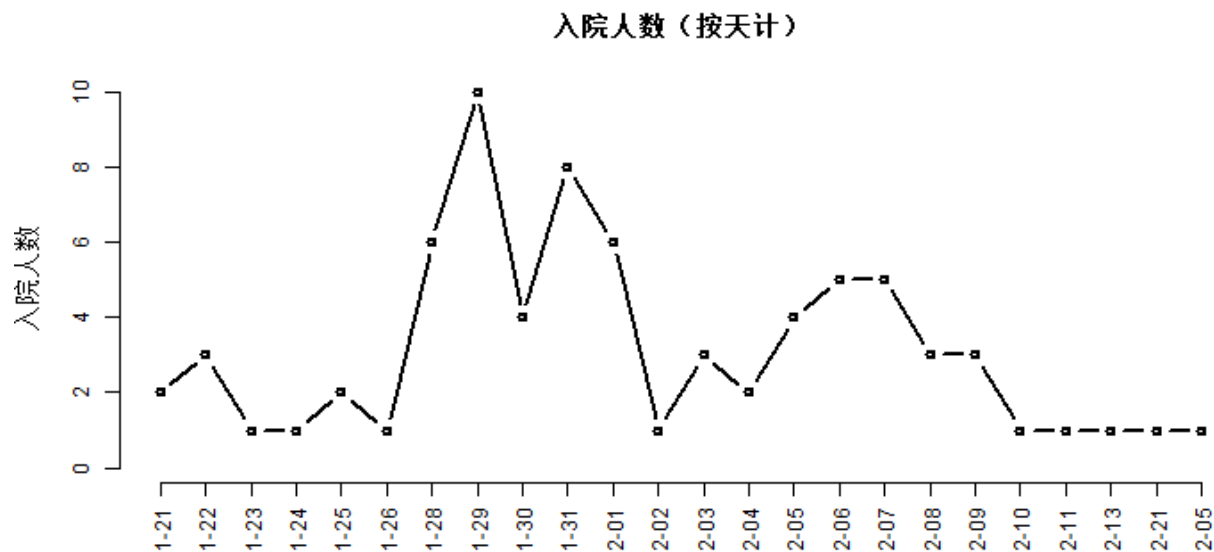


添加坐标轴

- 用 `axis()` 函数控制坐标轴。
 - `axis()` 的第一个参数取1, 2, 3, 4, 分别表示横轴、纵轴、上方和右方;
 - `axis()` 的参数 `at` 为刻度线位置;
 - `labels` 为标签。

axis() 的例子

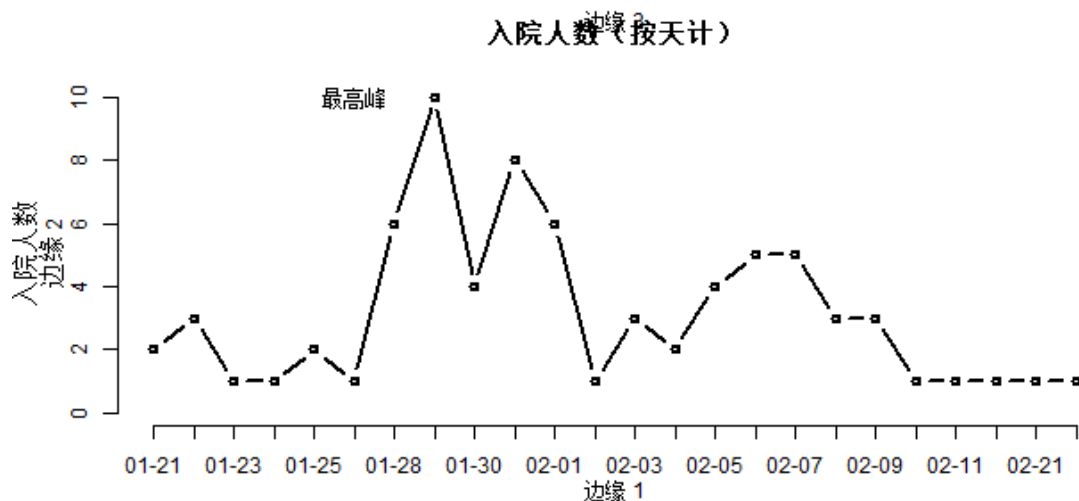
```
x <- table(as.Date(as.character(d[, "入院时间"]), format = "%Y%m%d"))
plot(x, type="b", main = "入院人数（按天计）", ylab = "入院人数", axes = FALSE) # axes = FALSE 表示不显示坐标轴
axis(2); axis(1, at = seq(along = x), labels = substr(names(x), 7, 10), las=2) # las = 2 表示总长度
```



添加文本注释

- 用 `text()` 函数在坐标区域内标注文本。 用 `mtext()` 函数在坐标区域外标注文本。

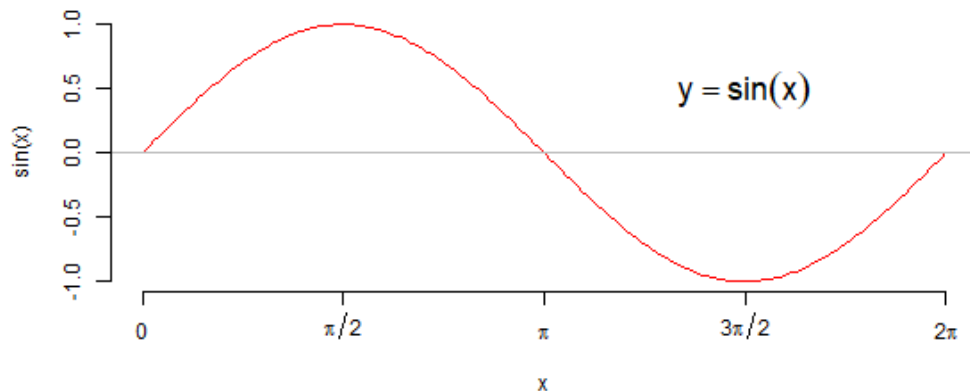
```
plot(x, type="b", main = "入院人数（按天计）", ylab = "入院人数", axes = FALSE) # axes = FALSE 表示不显示坐标轴
axis(2); axis(1, at = seq(along = x), labels = substr(names(x), 6, 10))
text(which.max(x)-2, max(x), "最高峰") # 前两个参数是坐标位置，第三个参数 `labels` 是要标注的文本
mtext(paste("边缘", 1:4), side = 1:4, line = 2)
```



添加数学公式

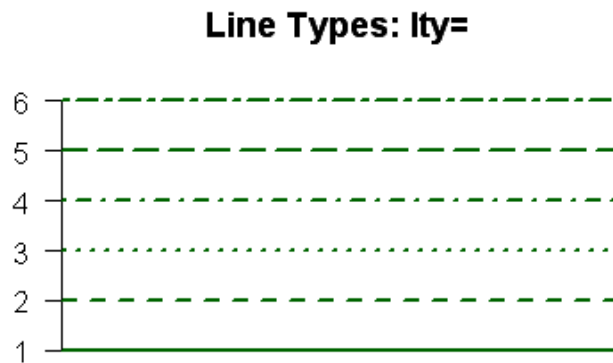
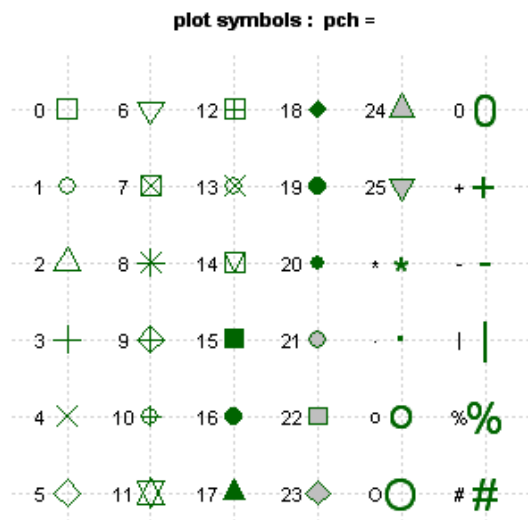
- R中的 `expression` 支持把数学公式显示到坐标轴，文本等。

```
x <- seq(0, 2*pi, length=200);  
plot(x, sin(x), type='l', col = "red", axes = FALSE)  
abline(h = 0, col = "gray")  
axis(2)  
axis(1, at = (0:4)/2*pi, labels=c(0, expression(pi/2), expression(pi),  
                                expression(3*pi/2), expression(2*pi)))  
text(1.5*pi, 0.5, expression(y==sin(x)), cex = 1.5)
```



控制图形参数

- `par()` 可以用于定制任意的图形参数，包括字体、颜色、坐标轴、标题等。均可用到前面的绘图函数中。
 - `col, col.axis, col.lab, ...`: 指定图形颜色、坐标轴颜色、标签颜色
 - `lty`: 线的类型, 1是实线, 2是虚线, 3以后是点虚线的各种组合
 - `lwd`: 线的宽度, 默认为1
 - `pch`: 点的类型
 - `cex, cex.axis, cex.lab, cex.main`: 符号大小倍数, 基本值为1。
 - `font`: 字体, 一般 `font=1` 是正体, 2 是粗体, 3 是斜体, 4 是粗斜体。



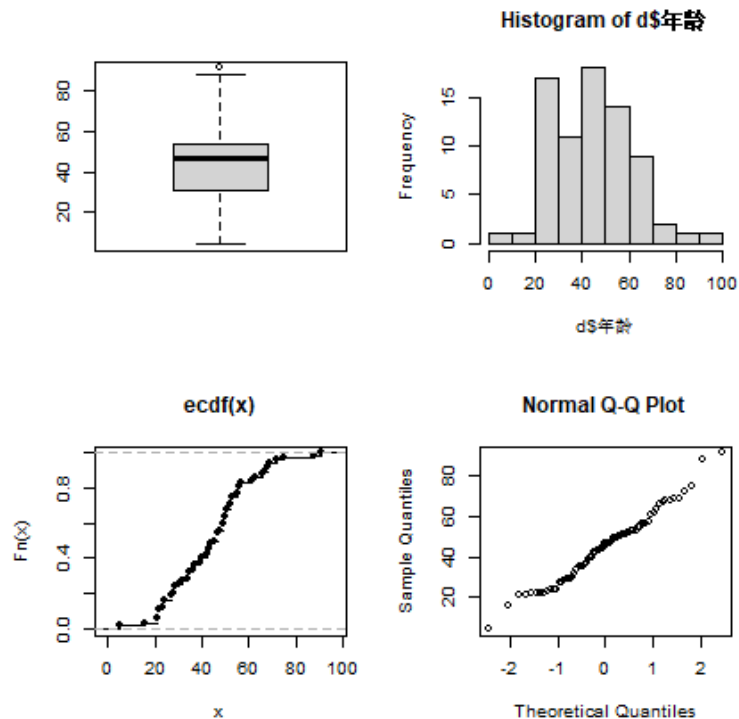
一页多图

一页多图

- 常见的控制页面布局函数有`par()`和`layout()`。
 - `par()`函数通过设置`mfrow`（按行）或者`mfcol`（按列）将页面分成几个区域，每个区域对应一个图形。
 - `layout()`函数提供一种更灵活的图形组合方式，允许有着不同尺寸的区域。

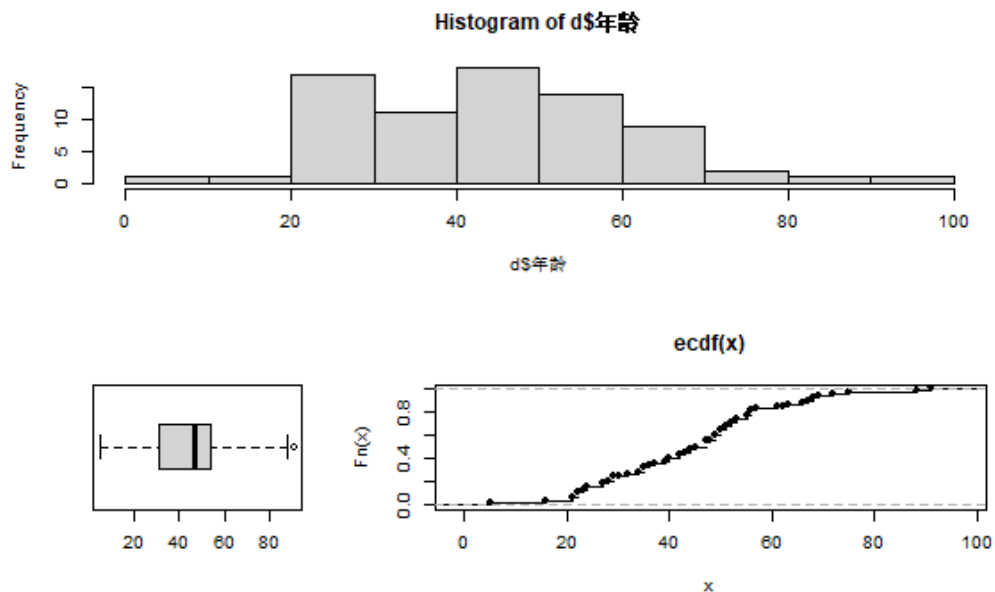
一页多图: par()

```
opar <- par(mfrow = c(2, 2))  
boxplot(d$年龄)  
hist(d$年龄)  
plot.ecdf(d$年龄)  
qqnorm(d$年龄)
```



一页多图: layout()

```
layout(matrix(c(1,1,2,3), 2,2, byrow = TRUE), widths = c(1, 2), heights = c(1,1))  
hist(d$年龄)  
boxplot(d$年龄, horizontal =TRUE)  
plot.ecdf(d$年龄)
```



谢 谢