

R 语言入门 (C)

R for Beginners - III

胡 高 博士

植保学院昆虫信息生态研究室

<http://eco.njau.edu.cn>

May 14, 2013

R 语言入门 (C)

1 绘图函数

- 绘图基础
- `plot()`
- 高阶绘图函数
- 低阶绘图函数
- `par()`
- 一页多图
- 图形设备

2 元素

- 颜色
- 点
- 线

- 矩形、多边形
- 网格线
- 文字
- 图例
- 坐标轴

3 图库

- 直方图
- 箱线图
- 条形图
- 散点图
- 饼图

4 绘图原则

5 总结与练习

§ 1

绘图函数

WHAT: 矢量图 & 位图 & 图片格式

WHAT: 绘图软件 & Excel & Photoshop & Matlab & SAS

1.1 绘图基础

■ demo(graphics): 了解 R 绘图功能

■ 绘图函数

1 高阶绘图函数 High-level Plotting Function

- 产生一个新的图区，可能包括坐标轴、标签、标题等

2 低阶绘图函数 Low-level Plotting Function

- 在已有的图上加上更多的图形元素

■ 绘图参数

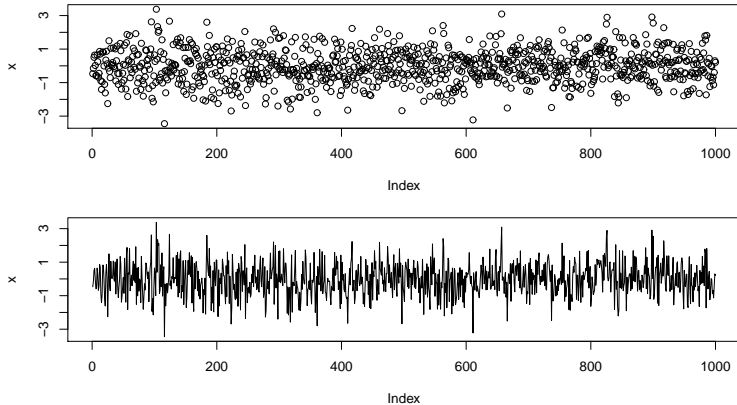
- ▶ 缺省值
- ▶ par()

```
1 > demo( graphics )
```

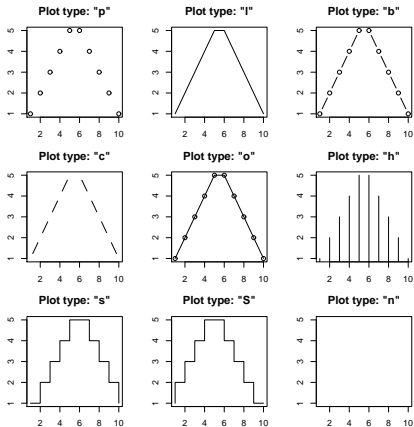
1.2 plot()

■ plot(x, y, ...)

- ▶ 泛型函数：根据第一个参数的类别决定调用何种函数作图
- ▶ type: 图形样式类别
- ▶ main / sub / xlab / ylab: 主标题 / 副标题 / X 轴标题 / Y 轴标题
- ▶ asp: 图形纵横比
- ▶ xlim / ylim: 设置坐标系界限
- ▶ log: 设置坐标取对数, 取 ("x", "y", "xy") 之一
- ▶ ann: 一些默认标记是否显示 (坐标轴标题、标题)
- ▶ axes: 是否显示坐标轴
- ▶ xaxt / yaxt: 坐标轴样式, 默认标准样式"s", "n" 不画坐标轴
- ▶ frame.plot: 是否显示边框
- ▶ col / pch / lty / lwd / bg: 颜色 / 符号 / 线型 / 线宽 / 符号背景
- ▶ cex: 文本缩放倍数
- ▶ panel.first / panel.last: 作图前 (后) 要完成的工作
 - panel.first=grid(): 作图之前先添加网格线



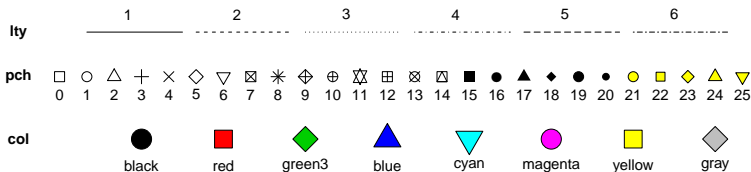
```
1 > x <- rnorm(1000)
2 > par(mfrow=c(2:1), mar=c(4,4,0.5,0.5)) # 设置一页多图和图形边界空白
3 > plot(x)
4 > plot(x, type='l')
```



```

1 > par(mfrow = c(3, 3), mar = c(2, 2.5, 3, 2))
2 > for(i in c("p", "l", "b", "c", "o", "h", "s", "S", "n")) {
3 +   plot(c(1:5, 5:1), type = i, main = paste("Plot_type:", i),
4 +   i, "\\ ", sep = " "), xlab = "")
5 + }

```

```
1 > par(mar = c(2, 2.5, 3, 2)) # 设置图边距
2 > x <- c(0:25); y <- rep(1,26); z<-c(1,5,9,13,17,21)
3 > plot(x,y,pch=x,ylim=c(0,1.5),bg=c("yellow"),ann=F,axes=F,cex=1.5)
4 > text(x,y-0.2,x) # 添加文字
5 > segments(z,1.5,z+3.5,1.5,lty=c(1:6)) # 添加线段
6 > text(z+1.5,rep(1.7,6),c(1:6), xpd=T)
7 > p <- c(3,6,9,12,15,18,21,24)
8 > points(p, rep(0.3,8), pch=c(21:25),bg=c(1:8), cex=3) # 添加点
9 > text(p,rep(0,8), palette(), xpd=T)
10 > text(rep(-1.5,3),c(1,1.5,0.3),c("pch","lty","col"),font=2,xpd=T)
```

1.3 高阶绘图函数²

<code>plot(x, y)</code>	二元图	<code>plot(x)</code>	序号为横坐标
<code>pie(x)</code>	饼图	<code>sunflowerplot(x, y)</code>	向日葵散点图
<code>boxplot(x)</code>	箱线图	<code>stripchart(x)</code>	带状图
<code>coplot(x ~ y z)</code>	条件分割图	<code>interaction.plot(f1, f2, y)</code>	交互效应图
<code>matplot(x, y)</code>	矩阵图	<code>dotchart(x)</code>	Cleveland 点图
<code>fourfoldplot(x)</code>	四瓣图	<code>assocplot(x)</code>	关联图
<code>mosaicplot(x)</code>	马赛克图	<code>plot.ts(x)</code>	时间序列曲线
<code>pairs(x)</code>	散点图矩阵	<code>ts.plot(x)</code>	同 <code>plot.ts()</code>
<code>hist(x)</code>	直方图	<code>barplot(x)</code>	条形图
<code>qqplot(x, y)</code>	QQ 图	<code>qqnorm(x)</code>	QQ 图
<code>contour(x, y, z)</code>	等高线图	<code>filled.contour(x, y, z)</code>	颜色等高线图
<code>persp(x, y, z)</code>	三维透视图	<code>image(x, y, z)</code>	颜色图
<code>stars(x)</code>	星状图	<code>symbols(x, y, ...)</code>	符号图
		<code>termplot(mod.obj)</code>	回归模型影响图

²详细内容请参考 < 现代统计图形 >

■ 一些共同选项及其缺省值

- ▶ `add=FALSE`: 是否叠加图形到前一个图上 (仅部分函数适用)
- ▶ `axes=TRUE`: 是否绘制坐标轴
- ▶ `type="p"`: 图形类型
- ▶ `xlim / ylim`: 坐标系统界限
- ▶ `xlab / ylab`: X 轴标题 / Y 轴标题
- ▶ `main / sub`: 主标题 / 副标题

1.4 低阶绘图函数

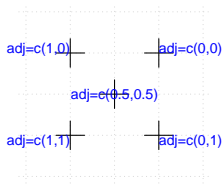
<code>points(x, y)</code>	添加点
<code>lines(x, y)</code>	添加线
<code>text(x, y, labels)</code>	添加文字
<code>mtext(text, side=3, line=0)</code>	在边空添加文字
<code>segments(x0, y0, x1, y1)</code>	添加线段
<code>arrows(x0, y0, x1, y1, angle= 30, code=2)</code>	画箭头
<code>abline(a,b)</code>	画直线 (斜率 b , 截距 a)
<code>abline(h=y)</code>	画水平线
<code>abline(v=x)</code>	画垂直线
<code>rect(x1, y1, x2, y2)</code>	绘制长方形
<code>polygon(x, y)</code>	绘制多边形
<code>legend(x, y, legend)</code>	添加图例
<code>title()</code>	添加标题
<code>axis(side, vect)</code>	画坐标轴
<code>box()</code>	画边框
<code>rug(x)</code>	在 x 轴上用短线画出数据位置

1.5 *par()*

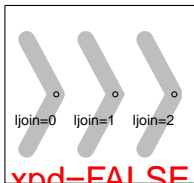
- *par()*: 返回当前的图形设置
- *par(tag = value)*: 设置图形参数

```
1 > par(mar=c(4, 4, 1, 0.5), bg="yellow")
2 > par()
3 $xlog
4 [1] FALSE
5
6 $ylog
7 [1] FALSE
8
9 $adj
10 [1] 0.5
11
12 ...
13 ...
```

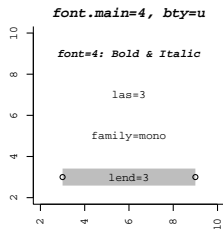
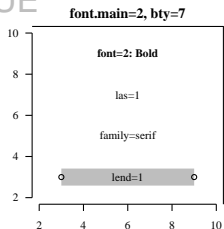
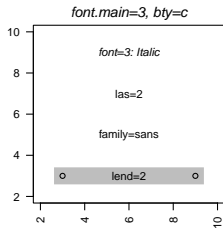
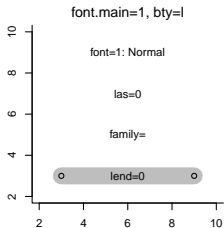
参数	含义	缺省	参数	含义	缺省
adj	字符相对位置	0.5	bty	边框样式	"o"
bg	图形背景色	"transparent"	cex	文本缩放倍数	1
col	图中符号颜色	"black"	family	字体族	""
fg	前景色	"black"	font	文本字体样式	1
lab	坐标轴刻度数目	5,5,7	las	坐标轴标签样式	0
lend	线条末端样式	"round"	lheight	图中文本行高	1
ljoin	线条相交样式	"round"	lty	线条样式	1
lwd	线条宽度	1	mex	坐标轴边界宽度	1
mar	图形边界空白	c(5,4,4,2)+0.1		缩放倍数	
mfrow	一页多图	c(1,1)	pch	点符号	1
mfcoll	一页多图	c(1,1)	pty	作图区域形状	"m"
mgp	坐标轴边界宽度	c(3,1,0)	srt	字符串旋转角度	0
oma	外边界	c(0, 0, 0, 0)	xaxt	X 坐标轴样式	"s"
tck	坐标刻度线高度	NA	yaxt	Y 坐标轴样式	"s"
tcl	坐标刻度线高度	-0.5	xlog	X 坐标取对数	F
xpd	处理超出边界	F	ylog	Y 坐标取对数	F



xpd=NA xpd=TRUE



xpd=FALSE



adj、xpd、ljoin、family、font、las、bty、lend

lend: 0→round, 1→mitre, 2→bevel)

■ font / col / cex

- ▶ font.axis: 坐标轴刻度
- ▶ font.lab: 坐标轴标题
- ▶ font.main: 主标题
- ▶ font.sub: 副标题

■ mar / oma

- ▶ 下、左、上、右

1.6 一页多图

- 1 `mfrow(nrow, ncol) / mfcrow(nrow, ncol)`
 - ▶ 将图形区域拆分为等长等宽的网格
 - ▶ `mfrow`: 横向排列
 - ▶ `mfcrow`: 纵向排列
- 2 `layout(mat, widths=rep(1, ncol(mat)), heights=rep(1, nrow(mat)))`
 - ▶ `mat`: 矩阵, 作图顺序以及图形版面安排
 - ▶ `widths / heights`: 各矩形图的长宽比
 - ▶ `layout.show(n)`: 显示第 n 区域
- 3 `split.screen()`

```

1 > par(mar=c(1,1,1,1), xaxt="n", yaxt="n")
2 > par(mfrow=c(2,2))      # 横向排列
3 > for(i in c(1:4)) {
4 +   plot(1,1,xlim=c(0,2),ylim=c(0,2),type="n")
5 +   text(1,1,i,cex=4,font=4,col=i)
6 + }
7 > par(mfcol=c(2,2))      # 纵向排列
8 > for(i in c(1:4)) {
9 +   plot(1,1,xlim=c(0,2),ylim=c(0,2),type="n")
10 +   text(1,1,i,cex=4,font=4,col=i)
11 + }
12 > ### layout() 更为灵活, 各单元不必等长等宽
13 > mat <- matrix(c(1,4,2,4,2,5,3,5),2); mat
14      [,1] [,2] [,3] [,4]
15 [1,]    1    2    2    3
16 [2,]    4    4    5    5
17 > layout(mat, widths=c(2,1,1,2))
18 > for(i in c(1:5)) {
19 +   plot(1,1,xlim=c(0,2),ylim=c(0,2),type="n")
20 +   text(1,1,i,cex=4,font=4,col=i)
21 + }

```

1	2
3	4

1	3
2	4

1	2	3
4	5	

1.7 图形设备

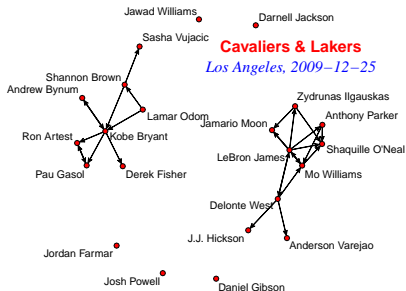
- 通过图形设备将 R 的图形输出为各种格式的文件³
- 位图设备
 - ▶ `bmp(file, width=480, heights=480, ...)`
 - ▶ `jpeg(file, width=480, heights=480, ...)`
 - ▶ `png(file, width=480, heights=480, ...)`
 - ▶ `tiff(file, width=480, heights=480, ...)`
- 矢量设备
 - ▶ `postscript(file, onefile, family, width=w, height=h, paper="a4", ...)`
 - ▶ `pdf(file, width, height, onefile, family, paper="a4", ...)`
- `dev.off()`: 关闭图形设备

³ 打开图形窗口也是打开图形设备, `X11()`、`windows()`、`quartz()` 分别对应于 UNIX 类 X11、Windows、MacOS X 的桌面系统的图形设备

§ 2

元素

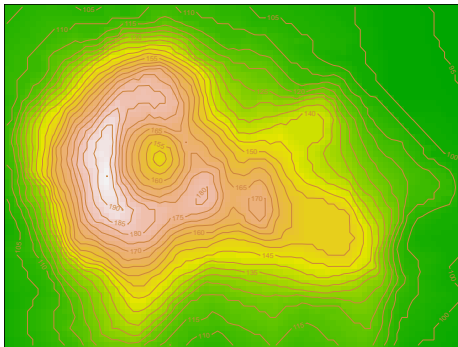
WHAT: 图形构成元素



2.1 颜色

- `colors()` / `colours()`: 生成 657 种颜色名称
- `palette(value)`: 调色板
 - ▶ `palette('default')`: 设置为默认调色板
- 特定颜色主题调色板
 - ▶ `rainbow()` / `heat.colors()` / `terrain.colors()` / `topo.colors()` / `cm.colors`

```
1 > colors()
2 > palette()
3 [1] "black"      "red"        "green3"     "blue"       "cyan"       "magenta"
4 [7] "yellow"     "gray"
5 > palette(colors()[1:8])
6 > palette()
7 [1] "white"          "aliceblue"      "antiquewhite"  "antiquewhite1"
8 [5] "antiquewhite2"  "antiquewhite3"  "antiquewhite4" "aquamarine"
```



新西兰 Maunga Whau 火山高度颜色图

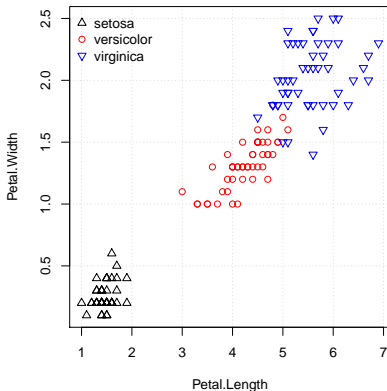
```
1 > par(mar = rep(0, 4), ann = FALSE)
2 > x = 10 * (1:nrow(volcano)); y = 10 * (1:ncol(volcano))
3 > image(x, y, volcano, col = terrain.colors(100), axes = FALSE)
4 > contour(x, y, volcano, levels = seq(90, 200, by = 5),
5 +       add = TRUE, col = "peru");
6 > box()
```

2.2 点

■ `points(x, y, pch=1)`

- ▶ `pch`: 字符样式, 21 ~ 25 可以填充背景颜色
- ▶ `lwd`: 设置点边缘线宽

```
1 > install.packages("MSG")           # 安装 MSG 包
2 > demo("pointTypes", package = "MSG") # 查看演示: 点的类型
3 > demo("pointArts", package = "MSG")  # 查看演示: 点的艺术
```

```

1 > ### 先将鸢尾花的类型转化为整数 1、2、3, 便于使用向量
2 > idx = as.integer(iris[["Species"]])
3 > plot(iris[, 3:4], pch = c(24, 21, 25)[idx], col=c("black",
4 +       "red", "blue")[idx], panel.first = grid())
5 > legend("topleft", legend = levels(iris[["Species"]]), col=
6 +       c("black", "red", "blue"), pch = c(24, 21, 25), bty = "n")

```

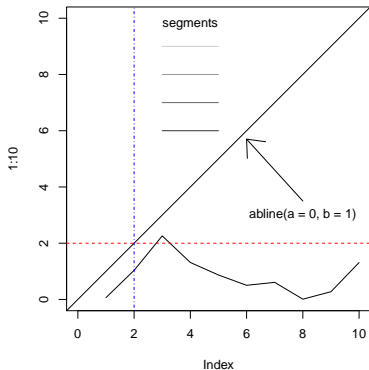
2.3 线

■ 主要函数

- ▶ `lines(x, y)`
- ▶ `abline()`
 - `abline(a=a, b=b)`: 斜率 a 、截距 b
 - `abline(h=h)`: 水平线
 - `abline(v=v)`: 垂直线
- ▶ `segments(x0, y0, x1=x0, y1=y0)`
- ▶ `arrows(x0, y0, x1=x0, y1=y0, angle=30, code=2)`
- ▶ `xspline(x, y)`: 画曲线

■ 共同选项及其缺省值

- ▶ `lty=1`: 线条样式
- ▶ `lwd=1`: 线宽



```

1 > par(mar = c(4,4,0.2,0.2))
2 > plot(1:10, type = "n", xlim = c(0, 10), ylim = c(0, 10))
3 > lines(1:10, abs(rnorm(10))) # 10个正态随机数绝对值的波动线
4 > abline(a=0, b=1, v=2, h=2, col=c(1,2,4), lty=c(1,2,4)) # 3条直线
5 > arrows(8, 3.5, 6, 5.7, angle = 40) # 添加箭头
6 > segments(rep(3, 4), 6:9, rep(5, 4), 6:9,
7 + col = gray(seq(0.2, 0.8, length = 4))) # 不同灰度的线段
8 > text(8, 3, "abline(a=0, b=1)"); text(4, 9.8, "segments")

```

2.4 矩形、多边形

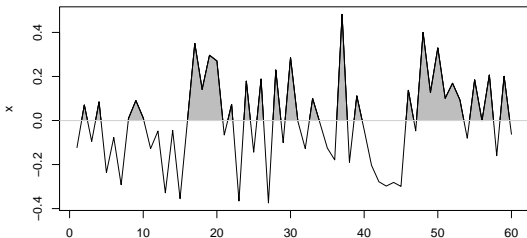
■ 基本函数

- ▶ `rect(xleft, ybottom, xright, ytop)`: 画矩形
- ▶ `polygon(x, y)`: 画多边形

■ 一些共同选项及其缺省值

- ▶ `density=NULL`: 设置阴影线的填充密度
- ▶ `angle=45`: 阴影线条的角度
- ▶ `col=NA`: 填充颜色或阴影线的颜色
- ▶ `border=NULL`: 边框颜色

```
1 > demo("kaleidoscope", package = "MSG") # 查看演示: 千纸鹤
```



```
1 > x = rnorm(60)      # 产生 60 个正态随机数
2 > plot(x, xlab = "", type = "l")
3 > polygon(c(1, 1:60, 60), c(0, x, 0), col = "gray")
4 > xy = par("usr")    # 获取当前图形区域坐标范围
5 > # 用白色矩形挡住了 0 以下的部分
6 > rect(xy[1], xy[3], xy[2], 0, col = "white")
7 > lines(x)           # 重画一遍 x 的线条
8 > abline(h = 0, col = "lightgray")
```

2.5 网格线

■ grid()

- ▶ nx / ny : 横纵轴上网格线的条数
- ▶ `equilogs=TRUE`: 当坐标取对数时, 是否仍采用等距网格线
- ▶ `col / lty / lwd`

■ `abline(h= h , v= v)`

2.6 文字

■ 主要函数

- ▶ `title(main=NULL, sub=NULL, xlab=NULL, ylab=NULL, line=NA)`
- ▶ `text(x, y, labels, adj=NULL, cex=1, font=NULL, xpd=FALSE)`
- ▶ `mtext(text, side=3, line=0, adj=NULL, cex=NA)`

■ 常用选项及其缺省值

- ▶ `adj / cex / font / xpd`
- ▶ `srt=0`: 字符串旋转角度
- ▶ `line=0`: 文本距离图形边缘的距离
- ▶ `side=3`: `mtext` 文本的位置 (1:4 → 下左上右)

■ 数学公式: `expression()`

```
1 > demo(plotmath)           # 查看演示: 数学公式输入
```

```
1 > plot(0:10, type = "n", xlab = "", ylab = "", xlim = c(0,12))
2 > grid(col = "gray") # 添加网格线
3 > title(main = "Demonstration of text in R Graphics",
4 + xlab = "X-axis title", ylab = "Y-axis title")
5 > mtext("Here is \"side = 4\"", side = 4, line = 1)
6 > x = c(6, 4, 6, 8); y = c(8, 5, 2, 5); s = c(0, 90, 180, 270)
7 > for(i in 1:4) text(x[i], y[i], sprintf('srt = %d', s[i]), srt=s[i])
8 > segments(c(6, 0, 6, 12), c(10, 5, 0, 5), c(0, 6, 12, 6),
9 + c(5, 0, 5, 10), lty = c(2, 1, 1, 2))
10 > legend(-0.2, 9.8, c("Upper", "Lower"), lty=2:1, cex =0.8, bty="n")
```


2.7 图例

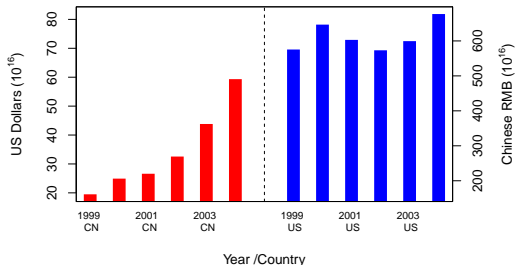
■ `legend(x, y, legend, fill=NULL, col="black", lty, lwd, pch, bty)`

- ▶ `fill`: 是否绘制颜色块
- ▶ `title`: 图例标题
- ▶ `cex`: 文本缩放倍数
- ▶ `col` / `lty` / `lwd` / `pch`: 点线颜色和样式, 同 `par()`
- ▶ `bty`: 图例框样式: "n" → 不画边框; "o" → 画边框
- ▶ `pt.bg` / `pt.cex` / `pt.lwd`: 符号的背景、大小、边缘线宽

```
1 > ### 示例详见上一页   
2 > legend(-0.2, 9.8, c("Upper", "Lower"), lty=2:1, cex =0.8, bty="n")
```

2.8 坐标轴

- `axis(side, at, labels, tick=TRUE)`
 - ▶ `side=1`: `mtext` 文本的位置 (1:4 → 下左上右)
 - ▶ `at`: 画坐标轴刻度标记线的位置
 - ▶ `labels`: 坐标轴刻度标记的位置
 - ▶ `tick`: 是否画坐标轴刻度线
 - ▶ `font / lty / lwd / col`: 同 `par()`
 - ▶ `tick.lty / tick.lwd / tick.col`: 刻度线线条样式、宽度、颜色



```

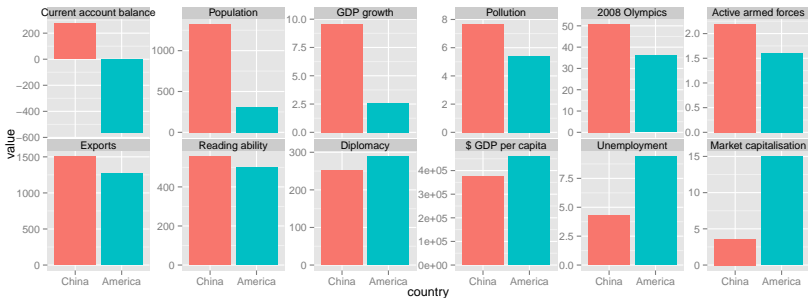
1 > library("MSG"); data(Export.USCN); par(mar=c(4,4.5,0.1,4.5))
2 > ylabel=expression(paste("US␣Dollars␣(",10^{16},")"))
3 > plot(1:13, Export.USCN$Export, xlab="Year␣/␣Country", ylab=ylabel,
4 + xaxt='n',type="h", lwd=15, col=c(rep(2,6),NA,rep(4,6)), lend=1)
5 > xlabel=paste(Export.USCN$Year, "\n", Export.USCN$Country)
6 > xlabel[7]=""
7 > abline(v=7, lty=2) # 添加一条分隔线
8 > axis(1, 1:13, labels = xlabel, tick = FALSE, cex.axis = 0.75)
9 > ylabel = pretty(Export.USCN$Export * 8.27)
10 > axis(4, at = ylabel/8.27, labels = ylabel)
11 > ylabel = expression(paste("Chinese␣RMB␣(",10^{16},")"))
12 > mtext(ylabel, side=4, line=3); box()

```

§ 3

图库

WHAT: 直方图 & 条形图 & 散点图 & 饼图 & 箱线图



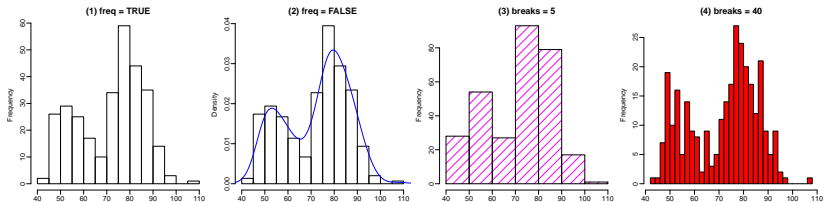
3.1 直方图

■ 直方图 Histogram

- ▶ 展示连续数据分布最常用的工具
- ▶ 对密度函数的一种估计

■ `hist(x, breaks="Sturges", freq=NULL, probability=!freq)`

- ▶ `breaks`: 分段区间方法
 - 1 向量: 区间端点
 - 2 数字: 拆分多少段
 - 3 字符串: 计算算法名称
- ▶ `freq / probability`: 设置以频数还是以概率密度作图
- ▶ `density / angle / border / col`: 同 `rect()`
- ▶ `main / xlab / ylab / xlim / ylim`



```

1 > par(mfrow = c(1, 4), mar = c(2, 3, 2, 0.5), mgp = c(2, 0.5, 0))
2 > data(geyser, package = "MASS")
3 > hist(geyser$waiting, main="(1) freq=TRUE")
4 > hist(geyser$waiting, freq=FALSE, main="(2) freq=FALSE")
5 > lines(density(geyser$waiting), col="blue")
6 > hist(geyser$waiting, breaks=5, density=10, col="magenta",
7 + border="black", main="(3) breaks=5")
8 > hist(geyser$waiting, breaks=40, col="red", main="(4) breaks=40")

```

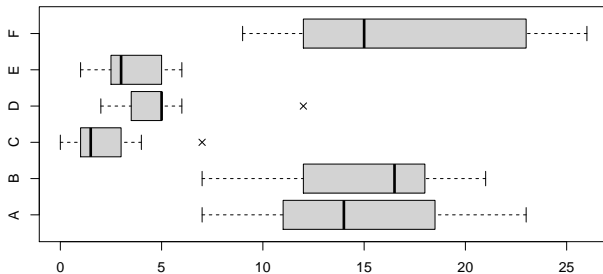
3.2 箱线图

■ 箱线图 Bos Plt / Box-and-Whisker Plot

- ▶ 四分位数
- ▶ 四分位间距 IQR
- ▶ 离群点 / 异常值

■ `boxplot(x) / boxplot(formula, data=NULL)`

- ▶ *formula*: 以 `~` 连接应变量和自变量
 - 纵坐标变量 `~` 横坐标变量 | 条件变量
- ▶ `horizontal=FALSE`:
- ▶ `varwidth=FALSE`: 箱子宽度是否与样本量的平方根成正比
- ▶ `notch=FALSE`: 是否画凹槽表示中位数的区间估计



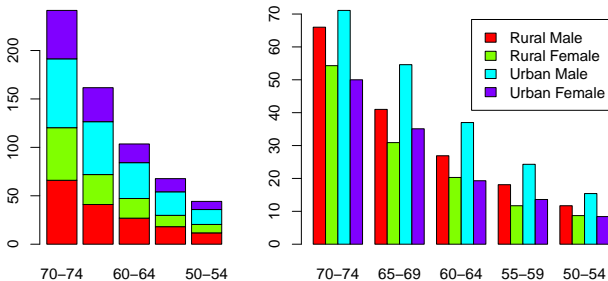
```

1 > InsectSprays[sample(nrow(InsectSprays), 3), ]
2   count spray
3  55      1    E
4   3     20    A
5  45      5    D
6 > par(mar = c(2, 2.5, 0.2, 0.1))
7 > boxplot(count ~ spray, data = InsectSprays, col = "lightgray",
8 + horizontal = TRUE, pch = 4)

```


3.3 条形图

- `barplot(height, width=1, space=NULL, names.arg=NULL)`
 - ▶ `width`: 矩形条的宽度
 - ▶ `space`: 矩形条的间距
 - ▶ `names.arg`: 条形图的标签
 - ▶ `beside`: `FALSE`→ 堆砌方式; `TRUE`→ 并列方式
 - ▶ `legend`: 是否添加图例
 - ▶ `horiz / density / angle`: 同 `rect()`



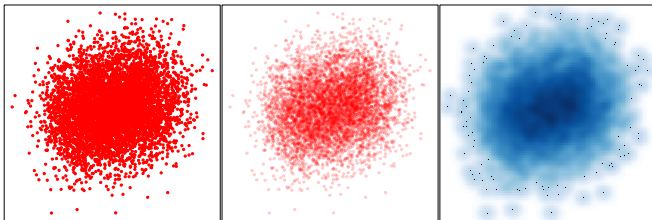
```

1 > par(mar = c(2.5, 3, 0.5, 1))
2 > layout(matrix(c(1:2),1),widths=c(1,1.5))
3 > death = t(VADeaths)[, 5:1]      # t() 数据框转置
4 > barplot(death, col=rainbow(4))
5 > barplot(death, col=rainbow(4), beside=TRUE, legend = TRUE)

```

3.4 散点图

- `plot(x,y)`
- `smoothScatter(x,y)`: 平滑散点图
 - ▶ 基于二维核密度估计
 - ▶ 用特定颜色深浅表示某个位置的密度值大小

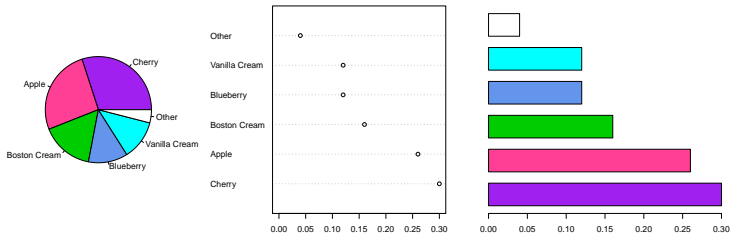


```
1 > par(mfrow = c(1, 3), pch=20, ann = FALSE, mar = rep(.1,4))
2 > x<-c(rnorm(3000,1),rnorm(3000,-1))
3 > y<-c(rnorm(3000,1),rnorm(3000,0.5))
4 > plot(x,y,col=rgb(1,0,0),axes=FALSE); box()
5 > plot(x,y,col=rgb(1,0,0,alpha=0.2),axes=FALSE); box()
6 > smoothScatter(x,y,axes=FALSE)
```

3.5 饼图⁴

- `pie(x, labels=names(x))`
 - ▶ `edges=200`: 圆弧光滑度
 - ▶ `density / angle / col / border`: 同 `rect()`

⁴ 这种以比例展示数据的统计图形实际上是很糟糕的可视化方式，R 关于饼图的帮助文件中清楚地说明了并不推荐使用饼图，而使用条形图或点图作为替代。



```

1 > par(mar=c(2,2,0.2,2))
2 > layout(matrix(c(1:3),1),widths=c(2,3,3))
3 > pie.sales = c(0.12, 0.3, 0.26, 0.16, 0.04, 0.12)
4 > names(pie.sales) = c("Blueberry", "Cherry", "Apple",
5 + "Boston_Cream", "Other", "Vanilla_Cream")
6 > pie.col = c("purple", "violetred1", "green3",
7 + "cornflowerblue", "cyan", "white")
8 > pie.sales=sort(pie.sales, decreasing=TRUE) # 排序有助于可读性
9 > pie(pie.sales, col = pie.col, xpd=NA)
10 > dotchart(pie.sales, xlim = c(0, 0.3))
11 > barplot(pie.sales, col = pie.col, horiz = TRUE,
12 + names.arg="", space=0.5)

```

§ 4

绘图原则

WHAT: 好图片

摄影: 主题 & 构图 & 色彩 & 光线 & 黄金分割点 & 镜头



4. 绘图原则

1 数据至上

- ▶ 分清元素主次
- ▶ 符号明确可分
- ▶ 谨慎处理数据

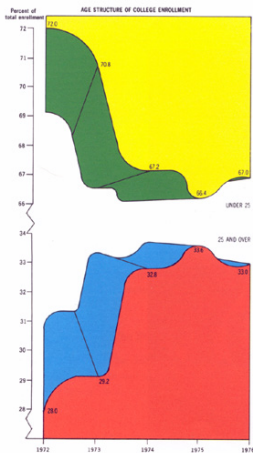
2 节约墨水

3 设计布局

4 附带解释

5 考虑心理

6 统计原则



史上最糟糕的垃圾图形：大学新生年龄结构⁵

1972-1976 年 25 岁以上的美国大学新生比例

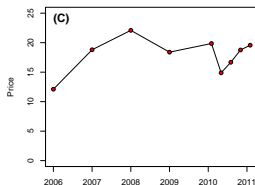
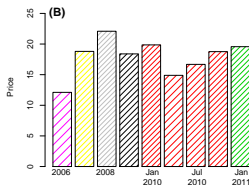
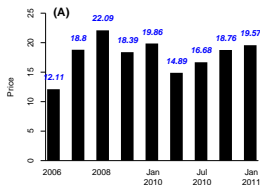
眼花缭乱的颜色、3D 效果、伪装有丰富的内容

⁵Tufte ER, 2001.The Visual Display of Quantitative Information . 2nd edition. Cheshire, CT, USA: Graphics Press. ISBN 0-9613921-4-2.

5.1. 关键词

- 1 高阶绘图函数、低阶绘图函数
- 2 `plot()`、`type`、`main/sub/xlab/ylab`、`xlim/ylim`、`log`、`col/pch/lty`
- 3 `par()`、`xpd`、`cex`、`mar`
- 4 `mfrow()`、`mfcoll()`
- 5 `points()`、`lines()`、`abline()`、`text()`、`legend()`
- 6 `hist()`、`boxplot()`、`barplot()`

5.2. 练习题



- 1 2006 年至 2011 年 1 月的猪肉价格如图 (A) 所示，请用 R 软件完成以上三个图（包括文字标注）。
- 2 什么是位图和矢量图？各有什么优势和不足？
- 3 比较 EXCEL 与 R 作图，各自有什么优势和不足？
 - ▶ 为什么还要学 R？

Thanks!