

45分钟搞定R语言之 数据库交互及统计绘图

王显峰 2012年4月

目录

- R 数据库交互
 - ✓ R 语言与数据库
 - ✓ 通过RODBC进行数据存取
- R 统计绘图
 - ✓ R 语言与统计绘图
 - ✓ R 的高水平绘图函数
 - ✓ R 的常用绘图参数
 - ✓ R 的低水平绘图函数
 - ✓ 组合多个图形
 - ✓ 高级格函数(Lattice Function)
- 小结

R与数据库：各有所长、取长补短

数据的组织

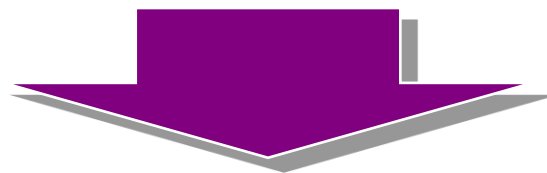
统计分析

数据的存储



数据的管理

统计图形



RODM、teradataR、Rhive.....

R与数据库：两种连接方式

ODBC(Open DataBase Connectivity),开放式数据库连接

- ✓ 大部分计算机都有（Windows/Linux/Mac OS X）
- ✓ 支持多种数据源（Oracle、MySQL、DB2、SQL Sever.....）
- ✓ 需要建立一个DSN（Data Source Name）或数据源的名称（控制面板-〉系统和安全-〉管理工具-〉数据源(ODBC)）



DBI(DataBase Interface)方式: ROracle、RMySQL.....

- ✓ 需要针对特定的数据库下载专门的软件包
- ✓ 比ODBC方式表现更好
- ✓ 目前对Windows平台支持不好

目录

- R 数据库交互
 - ✓ R 语言与数据库
 - ✓ 通过RODBC进行数据存取
- R 统计绘图
 - ✓ R 语言与统计绘图
 - ✓ R 的高水平绘图函数
 - ✓ R 的常用绘图参数
 - ✓ R 的低水平绘图函数
 - ✓ 组合多个图形
 - ✓ 高级格函数(Lattice Function)
- 小结

通过RODBC进行数据存取：主要函数

R通过RODBC方式进行数据存取的主要函数如下：

函数名称	函数描述
<code>odbcConnect(dsn, uid="", pwd="")</code>	建立并打开连接
<code>sqlFetch(channel, sqtable)</code>	从数据库读取数据表，并返回一个数据框对象
<code>sqlQuery(channel, query)</code>	向数据库提交一个查询，并返回结果
<code>sqlSave(channel, mydf, tablename = sqtable, append = FALSE)</code>	将一个数据框写入或更新(<code>append=True</code>)到数据库
<code>sqlDrop(channel, sqtable)</code>	从数据库删除一个表
<code>sqlClear(channel, sqtable)</code>	删除表中的内容
<code>sqlTables(channel)</code>	返回数据库中表的信息
<code>sqlColumns(channel, sqtable)</code>	返回数据库表 <code>sqtable</code> 列的信息
<code>close(channel)</code>	关闭连接

操作步骤：建立DSN——> 打开连接——> 数据操作——> 关闭连接

通过RODBC进行数据存取：读取数据

示例：通过RODBC从Oracle读取数据

加载RODBC包

```
library(RODBC)
```

创建连接

```
channel <- odbcConnect("BTEST",uid="dw",pwd="dw")
```

执行查询

```
org1 <- sqlFetch(channel,'DIM.DIM_ORGANIZATION')
```

```
org2 <- sqlQuery(channel,'SELECT * FROM DIM.DIM_ORGANIZATION')
```

关闭连接

```
odbcClose(channel)
```

使用数据

```
head(org1)
```

```
head(org2)
```

通过RODBC进行数据存取：写入数据库

示例：通过RODBC将数据写入Oracle

```
library(RODBC)
channel <- odbcConnect("BITEST",uid="dw",pwd="dw")

## 创建数据框（待写入对象）
mydata <- sqlQuery(channel,'SELECT * FROM DIM.DIM_ORGANIZATION')

## 写入数据库表，并读取结果表
sqlSave(channel,mydata,'NEW_TABLE_NAME',append = FALSE)
mydata2 <- sqlQuery(channel,'SELECT * FROM NEW_TABLE_NAME')

## 更新数据库表，并读取数据
sqlSave(channel,mydata,'NEW_TABLE_NAME',append = TRUE)
mydata3 <- sqlQuery(channel,'SELECT * FROM NEW_TABLE_NAME')

## 删除刚刚建立的表NEW_TABLE_NAME
sqlDrop(channel, 'NEW_TABLE_NAME')
odbcClose(channel)

## 查看执行结果
head(mydata,n = 30); head(mydata2,n = 30); head(mydata3,n = 30)
```


目录

- R 数据库交互
 - ✓ R 语言与数据库
 - ✓ 通过RODBC进行数据存取
- R 统计绘图
 - ✓ R 语言与统计绘图
 - ✓ R 的高水平绘图函数
 - ✓ R 的常用绘图参数
 - ✓ R 的低水平绘图函数
 - ✓ 组合多个图形
 - ✓ 高级格函数(Lattice Function)
- 小结

R与统计绘图：工具选择

图形的首要作用是“直观”的展示信息，但是“直观”不等于“简单”，统计量才是统计图形的最关键构成元素。

Microsoft Excel

- ✓ 价格高！
- ✓ 展示的几乎都是原始数据！
- ✓ 不能灵活的控制图形的组成元素！

其他商业统计软件

- ✓ 价格昂贵！
- ✓ 不能灵活的控制图形的组成元素！

R: is a free software environment for statistical computing and graphics

- ✓ 完全免费！
- ✓ 任何理论上可以计算出来的统计量都能在R中很方便地以图形的方式表达出来！
- ✓ 可以灵活的控制图形的组成元素！

R 与统计绘图：图形的最基础构成元素

任何一幅统计图形都是由以下这些最基础的图形元素构成

- 点 (points)
- 线 (lines, abline, segments, arrows)
- 多边形 (rect, polygon, box)
- 颜色 (colors)
- 文本 (text)
- 图例 (legend)

R与统计绘图：绘图函数的分类

高水平（high-level）绘图函数

- 在图形设备上产生一个新的绘图区域，并生成一个新的图形
- 通过其参数可以设置坐标轴，标签，标题等等

低水平（low-level）绘图函数

- 在已存在的图形上加上更多的图形元素，是绘制图形的基础函数
- 如额外的点、线、多边形和标签

交互式绘图函数

- 允许交互式地用定点设备（如鼠标）在已经存在的图上添加/提取图形信息

目录

- R 数据库交互
 - ✓ R 语言与数据库
 - ✓ 通过RODBC进行数据存取
- R 统计绘图
 - ✓ R 语言与统计绘图
 - ✓ R 的高水平绘图函数
 - ✓ R 的常用绘图参数
 - ✓ R 的低水平绘图函数
 - ✓ 组合多个图形
 - ✓ 高级格函数(Lattice Function)
- 小结

R的高水平绘图函数：plot()

plot()是最常用R绘图函数，是一个泛型函数，它产生的图形依赖于第一个参数的类型或者类

测试数据

```
set.seed(1); x <- sample(c(1:50),10)
set.seed(2); y <- sample(c(1:50),10)
xt <- ts(x)
xy <- cbind(x, y)
f <- as.factor(c(rep('a',3),rep('b',5),rep('c',2)))
```

调用示例	调用描述
plot(x)	x是一个向量，绘制向量x的值对其索引的图
plot(xt)	xt是一个时间序列对象，绘制xt的时间序列图
plot(f)	f是一个因子，绘制f的条形图
plot(x,y)	x,y是一个向量，绘制 y 对x的散点图
plot(xy)	xy是一个矩阵，绘制 y 对x的散点图
plot(f,y)	f是一个因子，y是一个向量，绘制y在f的各水平下的盒图

更多参见：*methods(plot)*

高水平绘图函数的更多示例参加：*example(plot)* *example(barplot)* 等等

R的高水平绘图函数：点图dochart()

点图（Cleveland dot plot）/克里夫兰点图：

基本实现，可用于检测离群点

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

按mpg进行排序，按cylinder进行分组并设定不同的颜色，可用于确定聪明变量

```
x <- mtcars[order(mtcars$mpg),] # sort by mpg
```

```
x$cyl <- factor(x$cyl) # it must be a factor
```

```
x$color[x$cyl==4] <- "red"
```

```
x$color[x$cyl==6] <- "blue"
```

```
x$color[x$cyl==8] <- "darkgreen"
```

```
dotchart(x$mpg, labels=row.names(x), cex=.7, groups= x$cyl, main="Gas Milage for Car  
Models\ngrouped by cylinder", xlab="Miles Per Gallon", gcolor="black", color=x$color)
```

R的高水平绘图函数：线图plot()+lines()

线图（Line Chart）：

基本实现，仅仅使用plot()

```
plot(mtcars$mpg, type="l")
```

plot()和lines()共同完成，plot()负责绘制一个没有点的图，lines()完成线条绘制，观察选项type=的不同选择值的差异

```
x <- c(1:5); y <- x; par(pch=22, col="red"); par(mfrow=c(2,4))
```

```
opts <- c("p","l","o","b","c","s","S","h")
```

```
for(i in 1:length(opts)){
```

```
  plot(x, y, type="n", main=paste("type=",opts[i]))
```

```
  lines(x, y, type=opts[i])
```

```
}
```

添加多条线

```
sal_num <- c(45,21,50,43,90,32); click_num1 <- sal_num+5; click_num2 <- sal_num+10;
```

```
plot(sal_num, col="black", type="o", ylim=c(0,105), axes=TRUE)
```

```
lines(click_num1, col="red", type="o", pch=22, lty=2)
```

```
lines(click_num2, col="blue", type="o", pch=22, lty=2)
```


R的高水平绘图函数:直方图hist()、核密度图density()

直方图（Histogram）：

基本实现，使用hist观察变量的分布

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

设置桶的数量和填充颜色

```
hist(mtcars$mpg, breaks=5, col="red")
```

```
hist(mtcars$mpg, breaks=c(10,25,30,40), col="red") #精确设置断点
```

核密度图（Kernel Density Plots）：

基本实现，使用density观察变量分布

```
d <- density(mtcars$mpg)
```

```
plot(d, main="Kernel Density of Miles Per Gallon")
```

```
polygon(d, col="gray71", border="blue") #设置图形格式
```

R的高水平绘图函数：盒图boxplot()

盒图（Boxplot）：

基本实现

```
boxplot(x=c(cars$speed,35))
```

```
boxplot(x=c(cars$speed,35),range=3) # range参数的值改变胡须的延伸位置
```

按cyl进行分组，绘制盒图

```
boxplot(mpg~cyl,data=mtcars, main="Car Milage Data", xlab="Number of Cylinders",  
ylab="Miles Per Gallon")
```

更多盒图参见：*sfsmisc*包*boxplot.matrix()*、*gplots*包*boxplot.n()*、*Rlab*包*bplot()*

R的高水平绘图函数：条图barplot()

条图（Barplot）：

基本实现

```
counts <- table(mtcars$gear)
```

```
barplot(counts, main="Car Distribution", xlab="Number of Gears")
```

```
barplot(counts, main="Car Distribution", horiz=T, names.arg=c("3 Gears", "4 Gears", "5 Gears")) # 水平放置，设置标签
```

堆积条形图，设置填充颜色和图例

```
counts <- table(mtcars$vs, mtcars$gear)
```

```
barplot(counts, main="Car Distribution by Gears and VS", xlab="Number of Gears",  
col=c("darkblue","red"), legend=rownames(counts))
```

分组条形图

```
counts <- table(mtcars$vs, mtcars$gear)
```

```
barplot(counts, main="Car Distribution by Gears and VS", xlab="Number of Gears",  
col=c("darkblue","red"), legend = rownames(counts), beside=TRUE)
```

R的高水平绘图函数：饼图pie()/pie3D()

饼图（Pie Chart）：

基本实现，绘制饼图并设置标签

```
slices <- c(10, 12, 4, 16, 8)
```

```
lbls <- c("US", "UK", "Australia", "Germany", "France")
```

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

设置百分比标签、颜色

```
slices <- c(10, 12, 4, 16, 8)
```

```
lbls <- c("US", "UK", "Australia", "Germany", "France")
```

```
pct <- round(slices/sum(slices)*100)
```

```
lbls <- paste(paste(lbls, pct) ,"%",sep="") # ad % to labels
```

```
pie(slices, labels = lbls, col=rainbow(length(lbls)), main="Pie Chart of Countries")
```

3D饼图

```
library(plotrix)
```

```
slices <- c(10, 12, 4, 16, 8)
```

```
lbls <- c("US", "UK", "Australia", "Germany", "France")
```

```
pie3D(slices, labels=lbls, explode=0.1, main="Pie Chart of Countries ")
```

R的高水平绘图函数：散点图plot()/scatterplot()/pairs()

散点图（Scatter Plots）：

基本实现-使用plot()

```
plot(mtcars$wt, mtcars$mpg, main="Scatterplot Example", xlab="Car Weight ", ylab="Miles  
Per Gallon ", pch=19)  
abline(lm(mtcars$mpg~mtcars$wt), col="red") # 添加回归线
```

散点图矩阵

```
pairs(~mpg+disp+drat+wt,data=mtcars, main="Simple Scatterplot Matrix")
```

3D散点图

```
library(scatterplot3d)  
scatterplot3d(mtcars$wt,mtcars$disp,mtcars$mpg, main="3D Scatterplot")
```

R的高水平绘图函数：条件图coplot()

条件图（Conditioning Plots）：

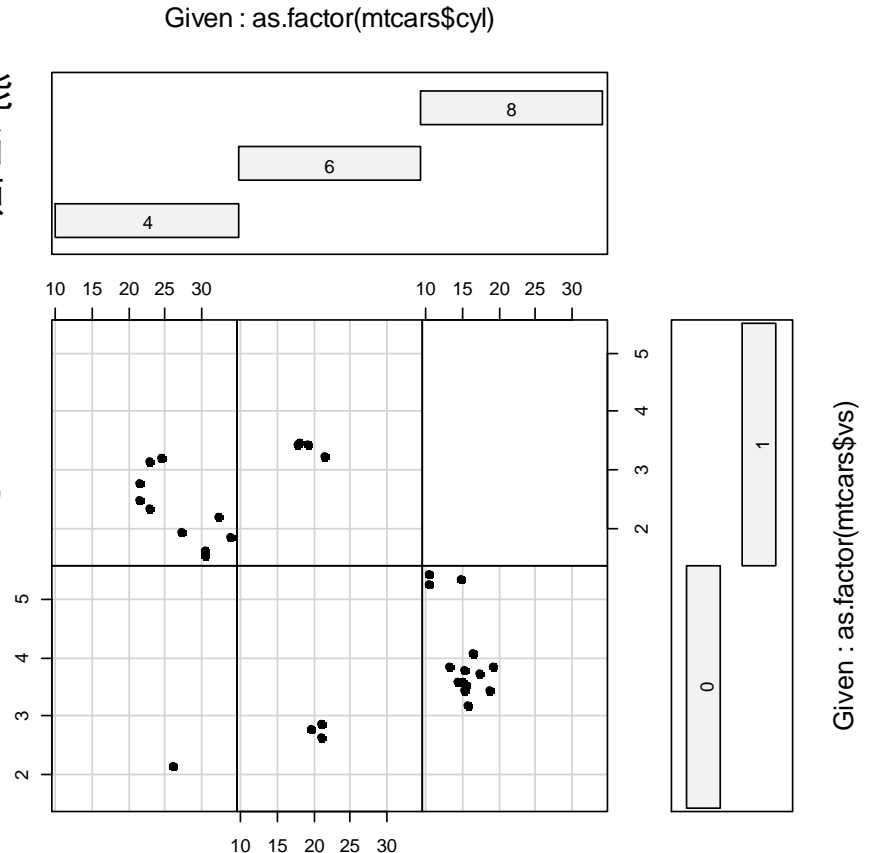
pairs()函数只能显示双向关系，而coplot()函数能够说明三向甚至四向关系，它特别适合于观察当给定其他预测变量时，反应变量如何根据一个预测变量变化。

单个条件变量的条件图

```
coplot(mtcars$wt ~ mtcars$mpg |  
as.factor(mtcars$cyl), main="", xlab="", ylab="",  
pch=19)
```

两个条件变量的条件图

```
coplot(mtcars$wt ~ mtcars$mpg |  
as.factor(mtcars$cyl) * as.factor(mtcars$vs),  
main="", xlab="", ylab="", pch=19)
```



R的高水平绘图函数：Q-Q图qqnorm()

Q-Q图（Quantile-Quantile Plots）：

设置绘图参数、准备数据

```
par(mfrow = c(1, 2))
```

```
x = rnorm(100)
```

绘制Q-Q图

```
qqnorm(x, cex = 0.7, asp = 1, main = "", xlim = c(-3,3), ylim=c(-3,3))
```

```
abline(0, 1,col='red')
```

绘制数据密度曲线

```
plot(density(x), main = "", xlim = c(-3,3), ylim=c(0,0.4))
```

绘制实际正态分布密度曲线

```
curve(dnorm, from = -3, to = 3, lty = 2, add = TRUE, col='red')
```

R的高水平绘图函数：其他

地图绘制：

地图绘制相关包

```
library(maps)
```

```
library(mapdata)
```

```
library(maptools)
```

读取地图数据

```
x <- readShapeSpatial('E:\\map\\bou2_4m\\bou2_4p')
```

用不同的颜色绘制地图

```
par(mar=c(0,0,0,0))
```

```
plot(x,col=rainbow(n=33));
```



地图数据文件下载：国家基础地理信息中心 <http://nfgis.nsdi.gov.cn/default.asp>

目录

- R 数据库交互
 - ✓ R 语言与数据库
 - ✓ 通过RODBC进行数据存取
- R 统计绘图
 - ✓ R 语言与统计绘图
 - ✓ R 的高水平绘图函数
 - ✓ R 的常用绘图参数
 - ✓ R 的低水平绘图函数
 - ✓ 组合多个图形
 - ✓ 高级格函数(Lattice Function)
- 小结

R 的常用绘图参数：指定绘图参数的两种方式

方式一：在高级绘图函数中直接指定

```
hist(mtcars$mpg, col.lab="red")
```

临时性参数设置

更多高水平绘图函数的参数参加具体的函数（如hist/boxplot/plot等等）

方式二：通过par()函数

```
par() # 查看当前绘图参数设置
```

```
opar <- par() # 保存当前设置
```

```
par(col.lab="red") # 设置坐标轴标签为红色
```

```
hist(mtcars$mpg) # 利用新的参数绘图
```

```
par(opar) # 恢复绘图参数的原始设置
```

全局性参数设置

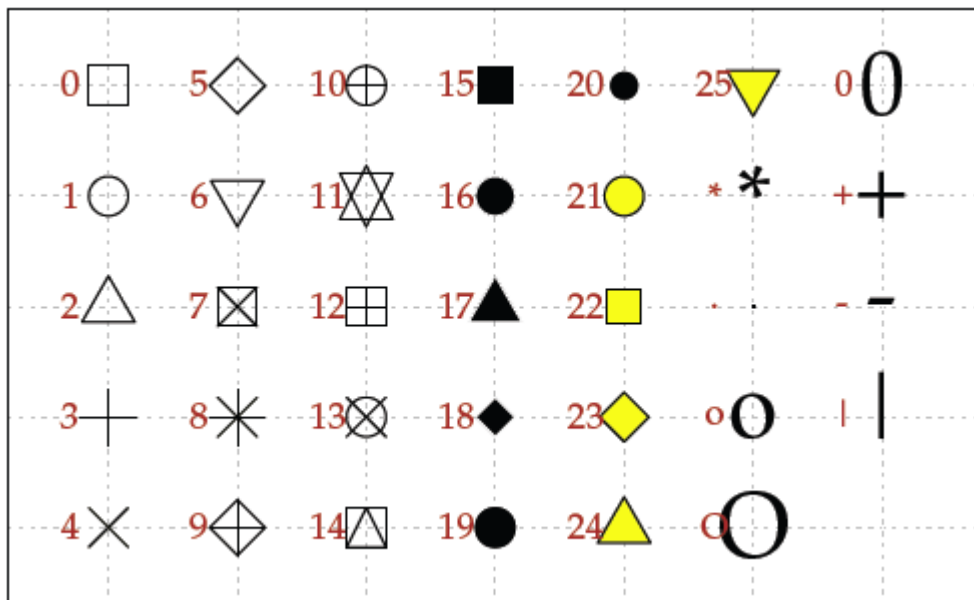
R 的常用绘图参数：设置文本和符号的大小

设置文本和符号大小的常用参数如下：

参数名称	参数描述
cex	指定文本和符号的缩放比例，默认值为1，若设置为1.5，则表示比默认大小大50%，若设置为0.5，则表示比默认值小50%
cex.axis	坐标轴刻度标记的缩放比例
cex.lab	坐标轴标题的缩放比例
cex.main	指定主标题的缩放比例
cex.sub	子标题的缩放比例
.....

R 的常用绘图参数：设置点的类型

使用pch=选项设置点的类型



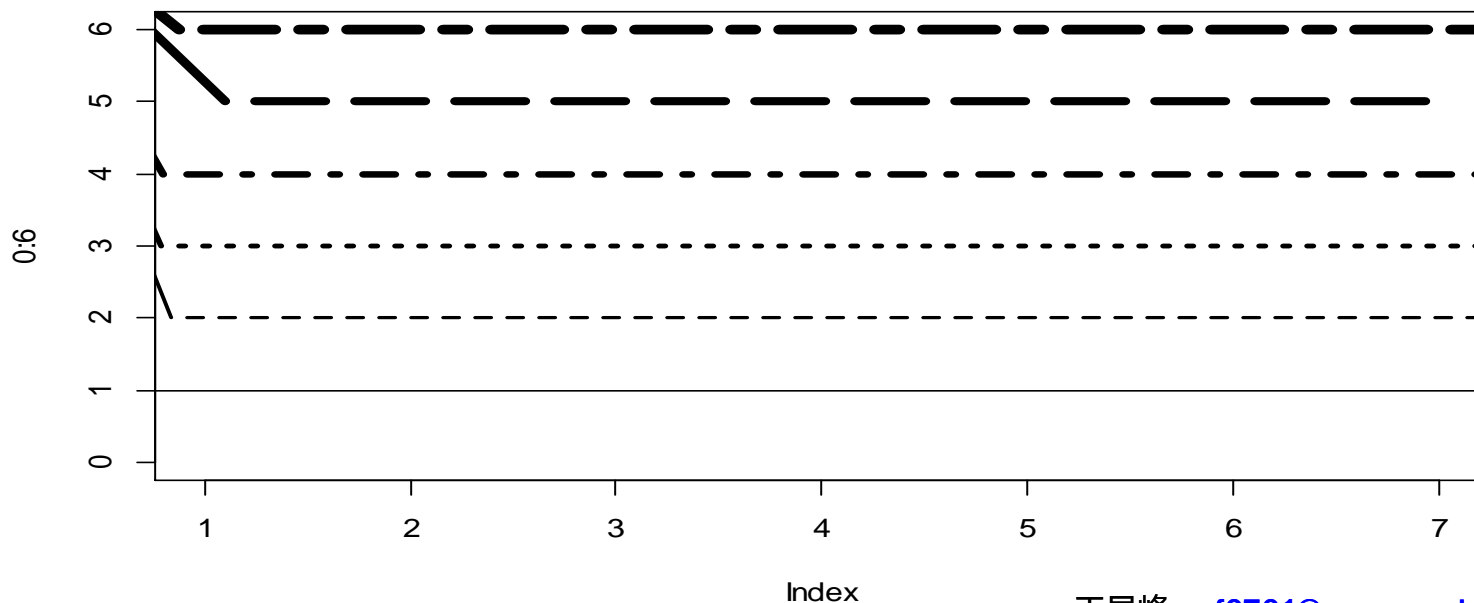
- ✓ 0-18为S语言兼容的点类型
- ✓ 21-25的点可以设置轮廓颜色和填充颜色
- ✓ 25之后的采用文本作为pch的参数值
- ✓ 更多参数值的设置参加?par

R 的常用绘图参数：设置线的类型

设置线条类型的相关参数如下

参数名称	参数描述
lty(Line TYPe)	参数值可以为整数或字符串：0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash
lwd(Line WiDth)	设置线条的相对宽度，默认值为1

```
plot(0:6,type='n')  
for(i in c(0:6)){abline(h=i,lty=i,lwd=i)}
```



R 的常用绘图参数：设置颜色

设置颜色的相关参数如下，可以设置的参数值包括：索引、名称、十六进制、RGB

例如：`col=1`；`col=white`；和`col="#FFFFFF"`是等价的

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425
426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475
476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525
526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550
551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575
576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625
626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650
651	652	653	654	655	656	657																		

参数名称	参数描述
col	默认的绘图颜色
col.axis	坐标轴刻度标记的颜色
col.lab	坐标轴标题的颜色
col.main	主标题的颜色
col.sub	子标题的颜色
fg	设置前景色
bg	设置背景色

固定颜色选择

`colors()`; `palette()`;

颜色的生成和转化

`rgb()`; `hsv()`; `hcl()`; `gray()`; `rgb2hsv()`;

返回一组持续的颜色

`rainbow(n)`; `heat.colors(n)`;

R 的常用绘图参数：设置字体

参数名称	参数描述
font	1=普通, 2=粗体, 3=斜体, 4=粗斜体, 5=符号字体
font.axis	坐标轴刻度标签的字体样式
font.lab	坐标轴标题的字体
font.main/font.sub	主标题/子标题的字体
family	字体族, 标准值包含“serif”, “sans”, “mono”, “symbol”, 其映射依赖于具体的设备, 在Windows设置中, 分别被映射为” TT Times New Roman”、” TT Arial”、” TT Courier New”、” TT Symbol” (TT=True Type)

创建新的字体映射

```
plot(1:10,1:10,type="n")
```

```
windowsFonts(A=windowsFont("Arial Black"),B=windowsFont("Bookman Old Style"),C=windowsFont("Comic Sans MS"),D=windowsFont("Symbol"))
```

```
text(3,3,"Hello World Default")
```

```
text(4,4,family="A","Hello World from Arial Black")
```

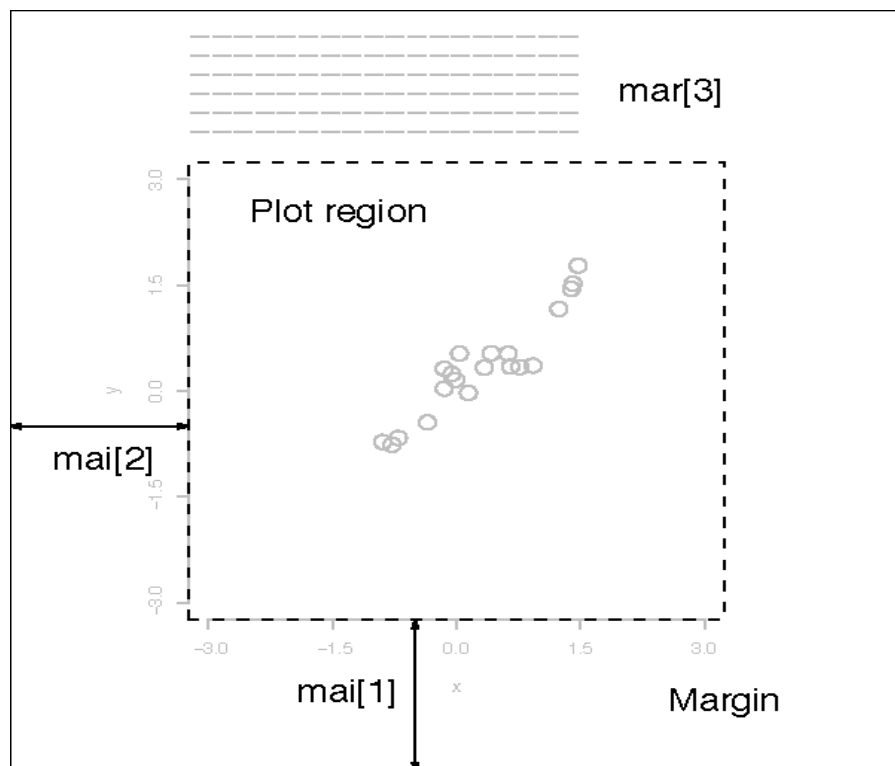
```
text(5,5,family="B","Hello World from Bookman Old Style")
```

```
text(6,6,family="C","Hello World from Comic Sans MS")
```

```
text(7,7,family="D","Hello World from Symbol")
```

R 的常用绘图参数：设置图形边缘大小

R中的图形除了包含一个由坐标轴包含的绘图区域(Plot Region)外，还包含底部、左侧、上部、右侧四个边缘区域



选项	描述
	形如 <code>c(bottom, left, top, right)</code> 的数值向量，指定底部、左边、上部和右边的空间，单位为行数，默认值为 <code>c(5, 4, 4, 2) + 0.1</code>
<code>mar</code>	
<code>mai</code>	与 <code>mar</code> 参数类似，只是单位为英寸

- ✓ 只能通过`par()`设置
- ✓ `mar`、`mai`可以相互转化
- ✓ 参数的默认值往往偏大
- ✓ 顶部和右侧常常不需要

`par(mar = c(2, 0, 0, 1))`

目录

- R 数据库交互
 - ✓ R 语言与数据库
 - ✓ 通过RODBC进行数据存取
- R 统计绘图
 - ✓ R 语言与统计绘图
 - ✓ R 的高水平绘图函数
 - ✓ R 的常用绘图参数
 - ✓ R 的低水平绘图函数
 - ✓ 组合多个图形
 - ✓ 高级格函数(Lattice Function)
- 小结

R 的低水平绘图函数：点、直线、线段、箭头、网格线

低水平绘图函数：点、直线、线段、箭头、网格线

函数名称	函数描述
points	在当前绘图区增加点
lines	在当前绘图区增加连接线
abline(a,b)	在当前绘图区增加一个斜率为b，截距为a的直线。
abline(h=y)	h=y可用于指定贯穿整个图的水平线高度的y-坐标。
abline(v=x)	v=x类似地用于指定垂直线的x-坐标。
abline(lm:obj)	lm:obj 可能是一个有长度为2的coefficients 分量(如模型拟合的结果)的列表。该分量中依次含有截距和斜率。
segments	绘制点对之间的线段
arrows	绘制点对之间的箭头
grid	在当前绘图区增加网格线

R 的低水平绘图函数：点、直线、线段、箭头

基本实现

```
plot(-4:4, -4:4, type = "p", col="blue")
```

绘制点、连接点

```
points(x=c(3,-2,-1,3,2), y=c(1,2,-2,2,3), col = "red")
```

```
lines(x=c(3,-2,-1,3,2), y=c(1,2,-2,2,3),col="black")
```

绘制直线

```
abline(h=0)
```

```
abline(v=0)
```

```
abline(a=1,b=1)
```

```
abline(lm(mtcars$mpg ~ mtcars$qsec),col="red")
```

绘制线段

```
segments(x0=2, y0=-4.5, x1=4, y1=-2, col="red", lty="dotted")
```

绘制箭头,并设置箭头的长度、角度、样式

```
arrows(x0=-4, y0=4, x1=-2, y1=0, length=0.15, angle=30, code=3)
```

##绘制网格线

```
grid(nx=3, ny=5, col = "lightgray", lty = "dotted")
```

R 的低水平绘图函数：多边形和矩形

绘制由(x,y)作为顶点的多边形的低水平函数如下：

```
polygon(x, y = NULL, density = NULL, angle = 45, border = NULL, col = NA, lty = par("lty"),  
..., fillOddEven = FALSE)
```

对于多边形的特例矩形，R还提供了专门的函数rect()来绘制：

```
rect(xleft, ybottom, xright, ytop, density = NULL, angle = 45,  
col = NA, border = NULL, lty = par("lty"), lwd = par("lwd"),...)
```

基本实现

```
plot(-4:4, -4:4, type = "p", col="blue")
```

```
polygon(x=c(3,-2,-1,3,2), y=c(1,2,-2,2,3), col = "red") # 绘制多边形
```

```
rect(xleft=c(-4,0), ybottom=c(2,-4), xright=c(-2,2), ytop=c(4,-2), col = c("blue", "yellow")) # 绘  
制两个矩形，并填充颜色
```

R 的低水平绘图函数：标题、任意文本、周边文本

使用title()函数添加红色标题和蓝色子标题，设置坐标轴标签为绿色，字体相对大小为0.75

```
plot(mtcars$wt, mtcars$mpg, main="", sub="", xlab="", ylab="")
```

```
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)
```

使用text()/mtext()函数为绘图区域/边缘区域添加文本注释

```
plot(x=mtcars$wt, y=mtcars$mpg, main="Milage vs. Car Weight", xlab="Weight",  
ylab="Mileage", pch=18, col="blue")
```

```
text(x=mtcars$wt, y=mtcars$mpg, labels=row.names(mtcars), cex=0.6, pos=4, col="red")
```

```
mtext("Added by mtext()", side=2, line=2, col='blue')
```

R 的低水平绘图函数：坐标轴和图例

##添加坐标轴

```
x <- c(1:10); y <- x; z <- 10/x
par(mar=c(5, 4, 4, 8) + 0.1)
plot(x, y,type="b", pch=21, col="red", yaxt="n", lty=3, xlab="", ylab="")
lines(x, z, type="b", pch=22, col="blue", lty=2)
axis(side=2, at=x,labels=x, col.axis="red", las=2) # 左侧添加坐标轴，设置坐标轴刻度标签样式
axis(side=4, at=z,labels=round(z,digits=2), col.axis="blue", las=2, cex.axis=0.7, tck=-.01) #
右侧添加坐标轴，设置坐标轴标签及刻度线的长度
mtext("y=1/x", side=4, line=3, cex.lab=1,las=2, col="blue")
title("An Example of Creative Axes", xlab="X values", ylab="Y=X")
```

添加图例，并设置格式

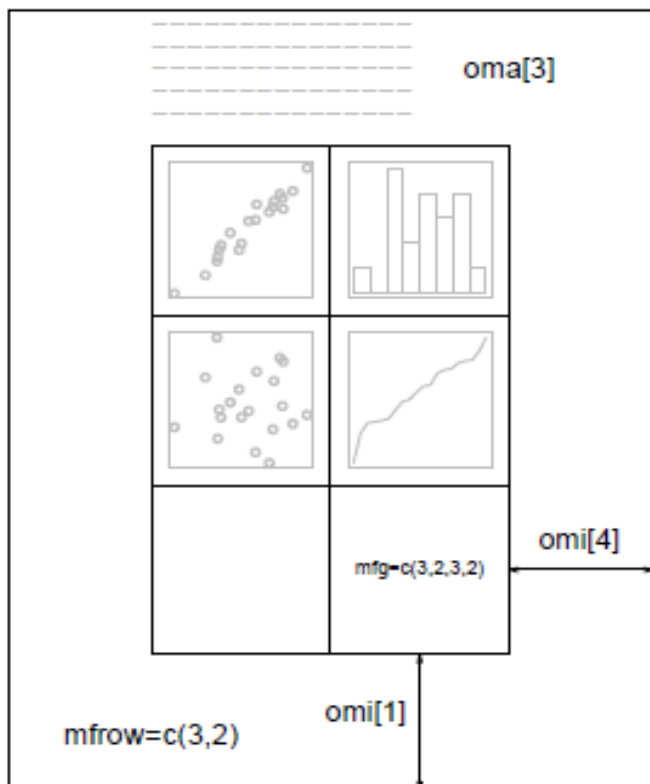
```
counts <- table(mtcars$vs, mtcars$gear)
barplot(counts, main="Car Distribution by Gears and VS",xlab="Number of Gears",
col=c("darkblue","red"), beside=TRUE)
legend(x=7.5, y=12, legend=c("L-A","L-B"), pch=15, col=c("blue","red"), cex=0.8, pt.cex=1,
box.lty="dashed")
```

目录

- R 数据库交互
 - ✓ R 语言与数据库
 - ✓ 通过RODBC进行数据存取
- R 统计绘图
 - ✓ R 语言与统计绘图
 - ✓ R 的高水平绘图函数
 - ✓ R 的常用绘图参数
 - ✓ R 的低水平绘图函数
 - ✓ 组合多个图形
 - ✓ 高级格函数(Lattice Function)
- 小结

组合多个图形：使用par()函数的mfrow/mfcol参数

R允许在同一个页面中创建多组图形，以便对这些图形进行对比



方式一：使用par()函数的mfrow/mfcol参数

选项	描述
mfrow	形如c(nrow,ncol)的数值向量，分别表示图形矩阵的行数和列数,按行绘制
mfcol	与mfrow参数类似，但是按列绘制

使用frow/fcol

```
mypar <- par(mfrow=c(2,2))
```

```
plot(mtcars$wt,mtcars$mpg, main="Scatterplot of wt  
vs. mpg")
```

```
plot(mtcars$wt,mtcars$disp, main="Scatterplot of wt  
vs disp")
```

```
hist(mtcars$wt, main="Histogram of wt")
```

```
boxplot(mtcars$wt, main="Boxplot of wt")
```

```
par(mypar)
```


组合多个图形：使用layout函数

方式二：使用layout(mat, widths = rep(1, ncol(mat)), heights = rep(1, nrow(mat)), respect = FALSE)函数

参数	描述
mat	一个数值型矩阵，指定多个图形的位置，矩阵中的值必须为大于等于0的整数，并且如果出现N，则整数1，2.....N-1必须在矩阵中出现至少一次
widths	一个数值向量，指定窗口被分割的相对宽度或绝对宽度
heights	一个数值向量，指定窗口被分割的相对高度或绝对高度

基本实现

```
attach(mtcars)
```

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

```
layout.show(3)
```

```
hist(wt)
```

```
hist(mpg)
```

```
hist(displ)
```

组合多个图形：使用par()函数的fig参数

方式三：使用par()函数的fig参数

- ✓ fig参数的值是一个形如c(x1,x2,y1,y2)的数值型向量，指定各个图形的绘制位置
- ✓ new参数是一个逻辑指，指定是否将图形绘制到已有图形

在散点图上添加盒图

```
par(fig=c(0,0.8,0,0.8), new=FALSE)
```

```
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)
```

更为灵活的方法，参见：*split.screen()*

目录

- R 数据库交互
 - ✓ R 语言与数据库
 - ✓ 通过RODBC进行数据存取
- R 统计绘图
 - ✓ R 语言与统计绘图
 - ✓ R 的高水平绘图函数
 - ✓ R 的常用绘图参数
 - ✓ R 的低水平绘图函数
 - ✓ 组合多个图形
 - ✓ 高级格函数(Lattice Function)
- 小结

高级格函数(Lattice Function): 简介

lattice是由Deepayan Sarkar基于grid包的一套统计图形系统，它的图形设计理念来自于Cleveland的Trellis图形，其主要特征是**根据特定变量（往往是分类变量）将数据分解为若干子集，并对每个子集画图**。就像数理统计中的条件期望、条件概率一样，lattice的图形也是一种“**条件作图**”。

格包中的高级函数如下：

函数	默认显示	函数	默认显示
histogram	直方图	xyplot	散点图
densityplot	核密度图	splom	散点图阵列
qqmath	理论分位数图	contourplot	表面等高线图
qq	双样本分位数图	levelplot	表面伪色彩图
stripplot	带形图	wireframe	三维表面透视图
bwplot	盒图	cloud	三维散点图
dotplot	克里夫兰点图	parallel	平行坐标图
barchart	条形图		

高级格函数(Lattice Function): 举例

加载包

```
library(lattice)
```

创建gear对应的因子类型变量

```
attach(mtcars)
```

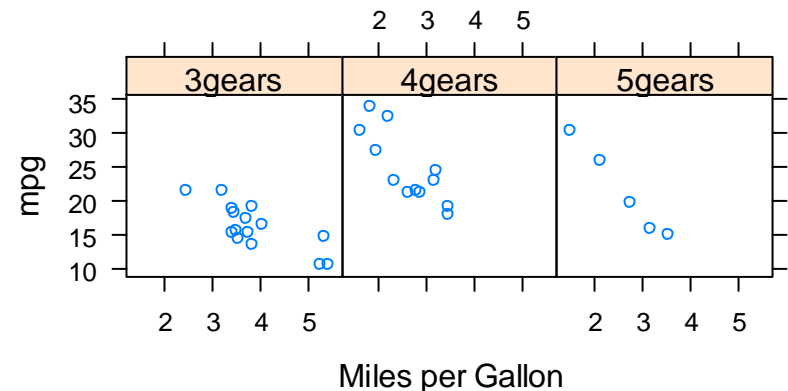
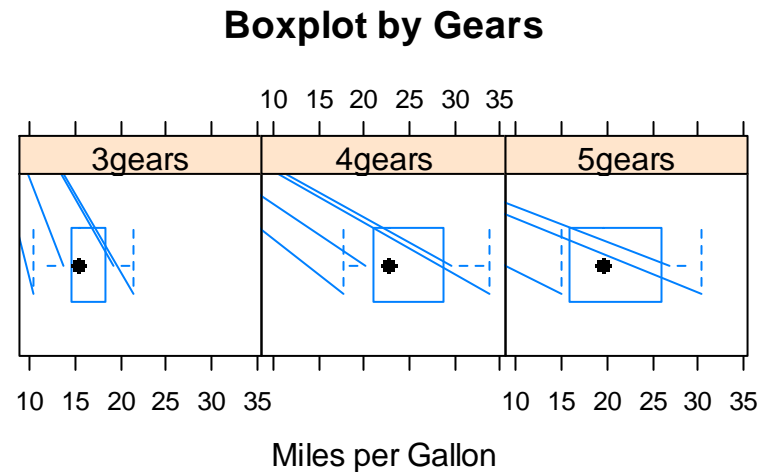
```
gear.f <- factor(gear,levels=c(3,4,5),  
labels=c("3gears","4gears","5gears"))
```

按gear.f绘制mpg的多面板盒图

```
bwplot(x=~ mpg|gear.f, main="Boxplot by  
Gears", xlab="Miles per Gallon", layout=c(3,1))
```

按gear.f绘制mpg~wt的多面板散点图

```
xyplot(x=mpg~wt|gear.f, main="Scatterplots by  
Gears", xlab="Miles per Gallon", layout=c(3,1))
```



目录

- R 数据库交互

- ✓ R 语言与数据库
- ✓ 通过RODBC进行数据存取

- R 统计绘图

- ✓ R 语言与统计绘图
- ✓ R 的高水平绘图函数
- ✓ R 的常用绘图参数
- ✓ R 的低水平绘图函数
- ✓ 组合多个图形
- ✓ 高级格函数(Lattice Function)

- 小结

小结

R 数据库交互

- ✓ 连接数据库的主要方式：RODBC
- ✓ 与数据库交互的主要函数：odbcConnect/SqlFetch/SqlQuery/SqlSave/SqlDrop/close
- ✓ 与数据库交互的步骤：建立DSN——> 建立连接——> 执行操作——> 关闭连接

R 统计绘图

- ✓ 图形的构成元素：点/线/多边形/颜色/文本/图例
- ✓ 绘图函数分类：高水平绘图函数/低水平绘图函数/交互式绘图函数
- ✓ 高水平绘图：plot/dohart/hist/density/boxplot/barplot/pie/pie3D/plot/scatterplot/pairs/qnorm
- ✓ 绘图参数：文本、符号的大小cex*/点的类型pch/线的类型lty、lwd/颜色col*、fg、bg/字体font*、ps、family/图形边缘mar、mai
- ✓ 低水平绘图参数：点points/直线lines、abline/线段segments/箭头arrows/网格线grid/多边形polygon/矩形rect/标题title/任意文本text/周边文本mtext/坐标轴axis/图例legend
- ✓ 组合多个图形：par()函数的mfrow/mfcol参数、layout函数、par()函数的fig参数
- ✓ 高级格函数：根据特定变量（往往是分类变量）将数据分解为若干子集，并对每个子集画图