R\_attack

Minajia ZHU

Tuesday, January 20, 2015

## 预处理

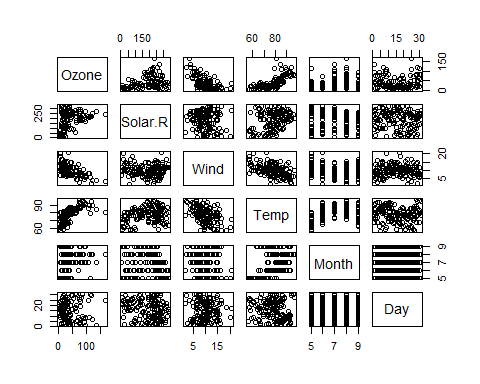
1.读取任何形式的数据(包括mp3数据):

G:\R\datascience\Getting and Cleaning Data\quiz2

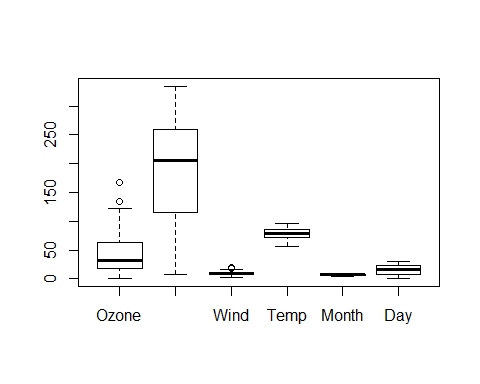
2.展示数据结构和概况:

list.dir()  
list.files()   
ls.str()

plot(airquality) #二维散点图



boxplot(airquality) #框须图



summary(airquality)

Ozone Solar.R Wind Temp   
 Min. : 1.00 Min. : 7.0 Min. : 1.700 Min. :56.00   
 1st Qu.: 18.00 1st Qu.:115.8 1st Qu.: 7.400 1st Qu.:72.00   
 Median : 31.50 Median :205.0 Median : 9.700 Median :79.00   
 Mean : 42.13 Mean :185.9 Mean : 9.958 Mean :77.88   
 3rd Qu.: 63.25 3rd Qu.:258.8 3rd Qu.:11.500 3rd Qu.:85.00   
 Max. :168.00 Max. :334.0 Max. :20.700 Max. :97.00   
 NA's :37 NA's :7   
 Month Day   
 Min. :5.000 Min. : 1.0   
 1st Qu.:6.000 1st Qu.: 8.0   
 Median :7.000 Median :16.0   
 Mean :6.993 Mean :15.8   
 3rd Qu.:8.000 3rd Qu.:23.0   
 Max. :9.000 Max. :31.0

str(airquality)

'data.frame': 153 obs. of 6 variables:  
 $ Ozone : int 41 36 12 18 NA 28 23 19 8 NA ...  
 $ Solar.R: int 190 118 149 313 NA NA 299 99 19 194 ...  
 $ Wind : num 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...  
 $ Temp : int 67 72 74 62 56 66 65 59 61 69 ...  
 $ Month : int 5 5 5 5 5 5 5 5 5 5 ...  
 $ Day : int 1 2 3 4 5 6 7 8 9 10 ...

attributes(airquality)

$names  
[1] "Ozone" "Solar.R" "Wind" "Temp" "Month" "Day"   
  
$class  
[1] "data.frame"  
  
$row.names  
 [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17  
 [18] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34  
 [35] 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51  
 [52] 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68  
 [69] 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85  
 [86] 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102  
[103] 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119  
[120] 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136  
[137] 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153

attr(airquality, "row.names")

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17  
 [18] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34  
 [35] 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51  
 [52] 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68  
 [69] 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85  
 [86] 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102  
[103] 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119  
[120] 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136  
[137] 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153

3.

x = c(1, 2, "cat", 3)  
x

[1] "1" "2" "cat" "3"

mode(x)

[1] "character"

y = c(5, T, 3, 7)  
y

[1] 5 1 3 7

mode(y)

[1] "numeric"

4.全局设置

names(options())

[1] "add.smooth" "browserNLdisabled"   
 [3] "CBoundsCheck" "check.bounds"   
 [5] "citation.bibtex.max" "continue"   
 [7] "contrasts" "defaultPackages"   
 [9] "demo.ask" "deparse.cutoff"   
[11] "device" "device.ask.default"   
[13] "digits" "echo"   
[15] "editor" "encoding"   
[17] "example.ask" "expressions"   
[19] "help.search.types" "help.try.all.packages"  
[21] "help\_type" "HTTPUserAgent"   
[23] "internet.info" "keep.source"   
[25] "keep.source.pkgs" "knitr.in.progress"   
[27] "locatorBell" "mailer"   
[29] "max.print" "menu.graphics"   
[31] "na.action" "nwarnings"   
[33] "OutDec" "pager"   
[35] "papersize" "pdfviewer"   
[37] "pkgType" "prompt"   
[39] "repos" "scipen"   
[41] "show.coef.Pvalues" "show.error.messages"   
[43] "show.signif.stars" "showErrorCalls"   
[45] "str" "str.dendrogram.last"   
[47] "stringsAsFactors" "tikzMetricsDictionary"  
[49] "timeout" "ts.eps"   
[51] "ts.S.compat" "unzip"   
[53] "useFancyQuotes" "verbose"   
[55] "warn" "warning.length"   
[57] "width" "windowsTimeouts"   
[59] "xtable.comment"

5.保存变量和读取变量

#前者保存全部变量，后者保存指定变量  
save.image()  
save(变量x,变量y,变量z, file = "mydata.rda")  
load("mydata.rda")

6.输入与输出

#文本输出  
sink("test.txt",append = T, SPLIT = T)  
#图片输出  
pdf("test.pdf"")  
   
head(CO2)  
dev.off()  
  
write.table(CO2,file = "co2.txt", row.names = F,col.names = F,sep = ",")  
  
#输出 LATEX 格式的表格还可以 quantreg 包中的 latex.table()函数  
#从其他程序读取数据,采用foreign包

# 调用R的输出信息  
glmout = capture.output(example(glm))  
glmout[1:5]

[1] ""   
[2] "glm> ## Dobson (1990) Page 93: Randomized Controlled Trial :"  
[3] "glm> counts <- c(18,17,15,20,10,20,25,13,12)"   
[4] ""   
[5] "glm> outcome <- gl(3,1,9)"

7.调用R的输出信息

glmout = capture.output(example(glm))  
glmout[1:5]

[1] ""   
[2] "glm> ## Dobson (1990) Page 93: Randomized Controlled Trial :"  
[3] "glm> counts <- c(18,17,15,20,10,20,25,13,12)"   
[4] ""   
[5] "glm> outcome <- gl(3,1,9)"

8.字符串转变为命令行

x = 1:10  
a = "print(x)"  
class(a)

[1] "character"

eval(parse(text = a))

[1] 1 2 3 4 5 6 7 8 9 10

## 数据操作

### 基础

**读取数据**

options(stringsAsFactors = F)  
read.table(file, col.names = c(), row.names = NULL, as.is = 索引向量(哪些变量不被转换成因子) / 与变量长度相符的逻辑向量 / TRUE(全部不转换),  
 na.strings = , comment.char = '', encoding = 其他字符, skip = , nrows = 想要读取的行数 ,fill = TRUE(用count.fields来确定问题所在),   
 colClasses = 字符向量 / "NULL"指跳过某列 / NA指默认设置)  
  
#若要读取相同宽度的不规则数据,使用read.fwf函数

#读取xls数据  
library(RODBC)  
z <− odbcConnectExcel ( ”.xls” )  
dd <− sqlFetch ( z , ” Sheet1 ” )  
close(z)

**检验两组向量或者数据框是否都为完整的，或者不同**

# 数据框是否完整  
x <- c(1, 2, NA, 4, NA, 5)  
y <- c("a", "b", NA, "d", "e", NA)  
complete.cases(x, y)

[1] TRUE TRUE FALSE TRUE FALSE FALSE

airquality[1:6, ]

Ozone Solar.R Wind Temp Month Day  
1 41 190 7.4 67 5 1  
2 36 118 8.0 72 5 2  
3 12 149 12.6 74 5 3  
4 18 313 11.5 62 5 4  
5 NA NA 14.3 56 5 5  
6 28 NA 14.9 66 5 6

good <- complete.cases(airquality)  
airquality[good, ][1:6, ]

Ozone Solar.R Wind Temp Month Day  
1 41 190 7.4 67 5 1  
2 36 118 8.0 72 5 2  
3 12 149 12.6 74 5 3  
4 18 313 11.5 62 5 4  
7 23 299 8.6 65 5 7  
8 19 99 13.8 59 5 8

#数据框的元素不同  
identical( a1 , a2 )  
which ( a1 !=a2 , arr.ind = TRUE) #arr.ind为array.indices

**去掉多余元素**

x = c (9 : 20 , 1 : 5 , 3 : 7 , 0 : 8 )  
( xu = x [ !duplicated(x) ] )  
unique (x)   
#is more efficient  
  
#删除list中的元素  
lst = list(a = list(b = 1,c = 2), b = list(d = 3,e = 4))  
lst$a$b = NULL

**缺失值的处理**

x[!is.na(x)]  
na.omit()  
对于统计慨括函数,mean,var等,可使用参数na.rm = T

**关于挑选数据**

subset ( iris , select = c ( Sepal.Length , Petal.Length ) ,  
Species == ”setosa” )

set.seed(13435)  
X <- data.frame(var1 = sample(1:5), var2 = sample(6:10), var3 = sample(11:15))  
X <- X[sample(1:5), ]  
X$var2[c(1, 3)] = NA  
X[(X$var1 <= 3 | X$var3 > 15), ]

var1 var2 var3  
1 2 NA 15  
4 1 10 11  
2 3 NA 12

X[1:2, "var2"]

[1] NA 10

X[which(X$var2 > 8), ]

var1 var2 var3  
4 1 10 11  
5 4 9 13

sort(X$var2, na.last = TRUE)

[1] 6 9 10 NA NA

X[order(X$var1, X$var3), ]

var1 var2 var3  
4 1 10 11  
1 2 NA 15  
2 3 NA 12  
5 4 9 13  
3 5 6 14

library(plyr)  
arrange(X, var1)

var1 var2 var3  
1 1 10 11  
2 2 NA 15  
3 3 NA 12  
4 4 9 13  
5 5 6 14

arrange(X, desc(var1))

var1 var2 var3  
1 5 6 14  
2 4 9 13  
3 3 NA 12  
4 2 NA 15  
5 1 10 11

X$var4 <- rnorm(5)  
Y <- cbind(X, rnorm(5))

**合并数据框**

merge(x,y,by.x = ,by.y = ,all = )

**简单的数据生成**

rep(1:5, each = 2)

[1] 1 1 2 2 3 3 4 4 5 5

rep(1:5, time = 2)

[1] 1 2 3 4 5 1 2 3 4 5

seq(1, 10, by = 2)

[1] 1 3 5 7 9

seq(1, 10, length = 3)

[1] 1.0 5.5 10.0

# 追加元素  
x = 1:5  
(foo = c(x[1], 0, x[2:5])) # expected r e s u l t

[1] 1 0 2 3 4 5

append(x, 0, after = 1)

[1] 1 0 2 3 4 5

**数据标准化**

x = c(rnorm(100), 2 \* rnorm(30))  
m = scale(x, scale = F) #only centering  
n = scale(x, center = F) #only scaling

**抽样**

sample(x)  
sample(x,replace = T)  
sample(x,n)  
sample(x,n,replace = T)  
sample(x,n,replace = T,prob = p)

**生成因子,转化为数字**

# 因子的基础知识  
  
data = c(1, 2, 2, 3, 1, 2, 3, 3, 1, 2, 3, 3, 1)  
fdata = factor(data)  
levels(fdata) = c("I", "II", "III")  
  
# 有序因子  
  
mons = factor("march", "april", "august", levels = c("april", "march", "august"),   
 order = T)  
  
# 对因子重新排序  
  
with(InsectSprays, reorder(spray, count, mean))

[1] A A A A A A A A A A A A B B B B B B B B B B B B C C C C C C C C C C C  
[36] C D D D D D D D D D D D D E E E E E E E E E E E E F F F F F F F F F F  
[71] F F  
attr(,"scores")  
 A B C D E F   
14.500000 15.333333 2.083333 4.916667 3.500000 16.666667   
Levels: C E D A B F

# gl(x,y) x为所需生成不同因子的数目，y为重复的次数  
  
thelevels = data.frame(group = gl(3, 10, length = 30), subgroup = gl(5, 2, length = 30),   
 obs = gl(2, 1, length = 30))  
  
# 第一个因子乘以第二个因子  
  
expand.grid(seq(1, 5, by = 2), seq(2, 5, by = 2))

Var1 Var2  
1 1 2  
2 3 2  
3 5 2  
4 1 4  
5 3 4  
6 5 4

as.numeric(as.charactor(x))

**两组平行向量的极值**

x = 1:10  
y = rev(x)  
pmax(x, y)

[1] 10 9 8 7 6 6 7 8 9 10

pmin(x, y)

[1] 1 2 3 4 5 5 4 3 2 1

**维度变换**

x = array(1:24, 2:4)  
xt = aperm(x,c(2,1,3))  
dim(x) ; dim(xt)

**序列转矩阵**

x = 1:10  
t(x)

[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]  
[1,] 1 2 3 4 5 6 7 8 9 10

class(t(x))

[1] "matrix"

t(t(x))

[,1]  
 [1,] 1  
 [2,] 2  
 [3,] 3  
 [4,] 4  
 [5,] 5  
 [6,] 6  
 [7,] 7  
 [8,] 8  
 [9,] 9  
[10,] 10

class(t(t(x)))

[1] "matrix"

**序列的处理**

# rle()为游程处理  
seq1 = c(1, 3, 5, 2, 4, 2, 2, 2, 7, 6)  
rle.seq1 = rle(seq1)  
rle.seq1

Run Length Encoding  
 lengths: int [1:8] 1 1 1 1 1 3 1 1  
 values : num [1:8] 1 3 5 2 4 2 7 6

any(rle.seq1$values == 2 & rle.seq1$length >= 3)

[1] TRUE

**时间序列**

ts(1:47, frequency = 4, start = c(1959, 2))

Qtr1 Qtr2 Qtr3 Qtr4  
1959 1 2 3  
1960 4 5 6 7  
1961 8 9 10 11  
1962 12 13 14 15  
1963 16 17 18 19  
1964 20 21 22 23  
1965 24 25 26 27  
1966 28 29 30 31  
1967 32 33 34 35  
1968 36 37 38 39  
1969 40 41 42 43  
1970 44 45 46 47

**矩阵**

rmat = matrix(rnorm(15), 5, 3, dimnames = list(NULL, c("A", "B", "C")))  
rmat

A B C  
[1,] 0.41567374 -1.6439381 -0.04233969  
[2,] 1.21261485 0.4926826 -0.03954086  
[3,] 0.73924856 -1.2092492 0.92157616  
[4,] -0.80156682 0.9182104 0.90542409  
[5,] 0.06324795 -0.2336559 2.10962374

# 或者  
  
rmat = matrix(rnorm(15), 5, 3)  
dimnames(rmat) = list(NULL, c("A", "B", "C"))  
rmat

A B C  
[1,] 0.3519587 0.8108234 -0.0453648  
[2,] -0.3999672 -0.1163880 1.7543884  
[3,] -4.1277771 -0.2766007 1.6906914  
[4,] -0.3116420 -0.9648843 -0.1237950  
[5,] 0.5143262 1.0315970 -0.7584366

# 矩阵乘法,crossprod(X,y)  
outer(month.abb, 1999:2003, FUN = "paste") #outer的使用

[,1] [,2] [,3] [,4] [,5]   
 [1,] "Jan 1999" "Jan 2000" "Jan 2001" "Jan 2002" "Jan 2003"  
 [2,] "Feb 1999" "Feb 2000" "Feb 2001" "Feb 2002" "Feb 2003"  
 [3,] "Mar 1999" "Mar 2000" "Mar 2001" "Mar 2002" "Mar 2003"  
 [4,] "Apr 1999" "Apr 2000" "Apr 2001" "Apr 2002" "Apr 2003"  
 [5,] "May 1999" "May 2000" "May 2001" "May 2002" "May 2003"  
 [6,] "Jun 1999" "Jun 2000" "Jun 2001" "Jun 2002" "Jun 2003"  
 [7,] "Jul 1999" "Jul 2000" "Jul 2001" "Jul 2002" "Jul 2003"  
 [8,] "Aug 1999" "Aug 2000" "Aug 2001" "Aug 2002" "Aug 2003"  
 [9,] "Sep 1999" "Sep 2000" "Sep 2001" "Sep 2002" "Sep 2003"  
[10,] "Oct 1999" "Oct 2000" "Oct 2001" "Oct 2002" "Oct 2003"  
[11,] "Nov 1999" "Nov 2000" "Nov 2001" "Nov 2002" "Nov 2003"  
[12,] "Dec 1999" "Dec 2000" "Dec 2001" "Dec 2002" "Dec 2003"

### 探索性数据分析

barplot(as.table(x))  
  
hist(x,breaks = 100)

Writing for, while loops is useful when programming but not particularly easy when working interactively on the command line. There are some functions which implement looping to make life easier.

* lapply: Loop over a list and evaluate a function on each element
* sapply: Same as lapply but try to simplify the result
* apply: Apply a function over the margins of an array
* tapply: Apply a function over subsets of a vector
* mapply: Multivariate version of lapply

An auxiliary function split is also useful, particularly in conjunction with lapply.

### example:

### sapply

# x可以为dataframe,list或者向量  
x <- 1:4  
lapply(x, runif, min = 0, max = 10)

[[1]]  
[1] 4.615508  
  
[[2]]  
[1] 6.016591 2.432595  
  
[[3]]  
[1] 8.37990865 0.08214777 6.52794376  
  
[[4]]  
[1] 1.4697061 5.6923372 2.0052372 0.5119882

x <- list(a = matrix(1:4, 2, 2), b = matrix(1:6, 3, 2))  
lapply(x, function(elt) elt[, 1])

$a  
[1] 1 2  
  
$b  
[1] 1 2 3

### mapply

noise <- function(n, mean, sd) {  
 rnorm(n, mean, sd)  
}  
noise(5, 1, 2)

[1] -1.8580674 -0.6222998 5.1758410 2.8752683 2.8485747

mapply(noise, 1:5, 1:5, 2)

[[1]]  
[1] 1.409456  
  
[[2]]  
[1] 1.1710330 0.2393748  
  
[[3]]  
[1] 5.829352 3.002883 3.245268  
  
[[4]]  
[1] 2.8360804 3.3797103 0.9605644 4.8474006  
  
[[5]]  
[1] 4.306285 8.640375 5.537962 3.395720 8.845117

### tapply(用于向量,而aggregate用于数据框)

x <- c(rnorm(10), runif(10), rnorm(10, 1))  
f <- gl(3, 10)  
f

[1] 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3  
Levels: 1 2 3

tapply(x, f, mean)

1 2 3   
-0.2594292 0.5337055 1.3156601

attach(warpbreaks)  
tapply(breaks, list(wool, tension), mean)

L M H  
A 44.55556 24.00000 24.55556  
B 28.22222 28.77778 18.77778

aggregate(breaks, list(wool, tension), mean)

Group.1 Group.2 x  
1 A L 44.55556  
2 B L 28.22222  
3 A M 24.00000  
4 B M 28.77778  
5 A H 24.55556  
6 B H 18.77778

### split

s <- split(airquality, airquality$Month)  
lapply(s, function(x) colMeans(x[, c("Ozone", "Solar.R", "Wind")]))

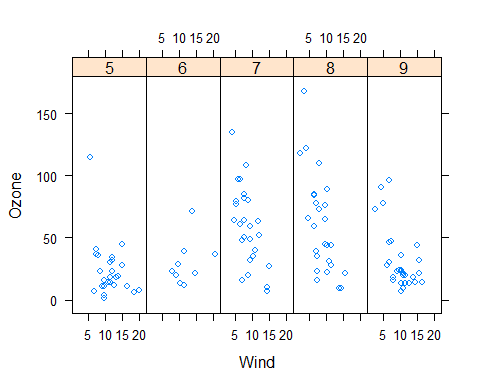
$`5`  
 Ozone Solar.R Wind   
 NA NA 11.62258   
  
$`6`  
 Ozone Solar.R Wind   
 NA 190.16667 10.26667   
  
$`7`  
 Ozone Solar.R Wind   
 NA 216.483871 8.941935   
  
$`8`  
 Ozone Solar.R Wind   
 NA NA 8.793548   
  
$`9`  
 Ozone Solar.R Wind   
 NA 167.4333 10.1800

sapply(s, function(x) colMeans(x[, c("Ozone", "Solar.R", "Wind")], na.rm = TRUE))

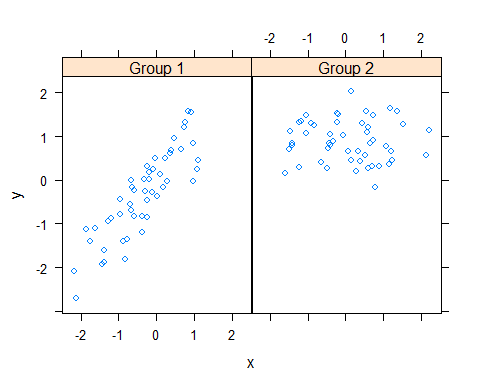
5 6 7 8 9  
Ozone 23.61538 29.44444 59.115385 59.961538 31.44828  
Solar.R 181.29630 190.16667 216.483871 171.857143 167.43333  
Wind 11.62258 10.26667 8.941935 8.793548 10.18000

## lattice

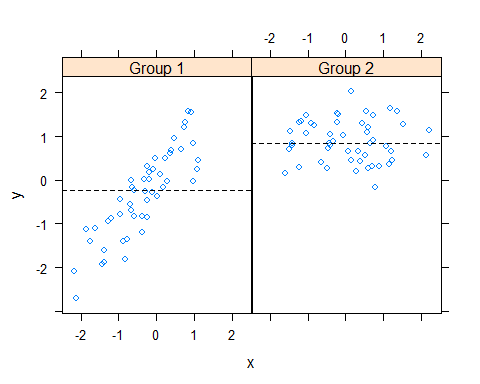
library(lattice)  
## Convert 'Month' to a factor variable  
airquality <- transform(airquality, Month = factor(Month))  
xyplot(Ozone ~ Wind | Month, data = airquality, layout = c(5, 1))



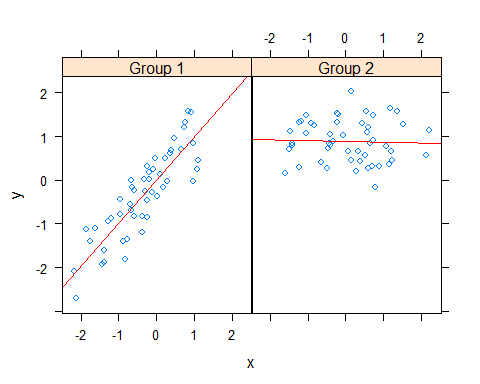
set.seed(10)  
x <- rnorm(100)  
f <- rep(0:1, each = 50)  
y <- x + f - f \* x + rnorm(100, sd = 0.5)  
f <- factor(f, labels = c("Group 1", "Group 2"))  
xyplot(y ~ x | f, layout = c(2, 1)) ## Plot with 2 panels



## Custom panel function  
xyplot(y ~ x | f, panel = function(x, y, ...) {  
 panel.xyplot(x, y, ...) ## First call the default panel function for 'xyplot'  
 panel.abline(h = median(y), lty = 2) ## Add a horizontal line at the median  
})



## Custom panel function  
xyplot(y ~ x | f, panel = function(x, y, ...) {  
 panel.xyplot(x, y, ...) ## First call default panel function  
 panel.lmline(x, y, col = 2) ## Overlay a simple linear regression line  
})



## 数学运算

**积分**

integrate(dnorm, -1.96, 1.96)

0.9500042 with absolute error < 1e-11

integrate(dnorm, -Inf, Inf)

1 with absolute error < 9.4e-05

fun = function(x) {  
 x^2  
}  
integrate(fun, lower = 0, upper = 2)

2.666667 with absolute error < 3e-14