# Octave：矩阵计算的新宠

实话实说，MatLab 是迄今为止矩阵计算最强大的工具（没有之一）。可惜 MatLab 是商用的，一般个体还真买不起。MatLab 的 Windows 版本 比 Linux 版本要好些，这让我不敢轻易断言 Windows 一无是处，毕竟其下有 MatLab 这样强悍的软件。以前在 Windows 下工作，MatLab 一直是我的首选矩阵计算工具，在统计计算工具 S-PLUS 出现之前，人们快乐地用着 MatLab 简陋的统计工具箱。后来有了 R，它彻底地坐稳了统计计算的 头把交椅，MatLab 似乎也无意去争夺全料冠军，但事实上它在很多方面都做得无可挑剔。这让我们这些买不起却很需要 MatLab 的穷人感慨不 已，MatLab 如果是免费的该多好……



## *为何选择使用 **octave**?*

SciLab 和 octave 是开源的且免费的矩阵计算工具，二者都有希望成为矩阵计算的新宠。相比之下，

- octave 与 MatLab 的兼容性更高。
- octave 遵循 GPL 协议（GNU General Public License），用户可以单独发行 octave 或者包含在其产品中发行。而 scilab 则不允许，你只能免费地使用它。
- octave 没有图形界面，是命令交互的。在某些人眼里这是不可饶恕的缺点，而在另外一些人眼里则是大大的优点。

它们都具备以下特点：以矩阵为基本数据类型，内置支持复数，有内置函数和外部函数库，用户自定义函数的可扩展性等特点。UNIX 的很多用户选择使用 octave，看中的就是它与 MatLab 兼容性好这一事实。随着开源运动的深入人心，octave 不断地发展壮大，它会吸引一大批 MatLab 的使用 者。

GNU octave 网站：http://www.octave.org/

好习惯从头开始：

- 首先学会使用 help，搞不定再到网上查，最后才求人。
- 学习 octave 的捷径：读 octave 的函数源码 。
- 每个命令都以“;”结束，否则矩阵的具体内容会显示出来。
- 学会适当地使用内置命令 clear，从内存中清除一些无用数据或变元。
- 如果没有必要，不要轻易改变矩阵大小。
- 重要的中间结果要保存。

---

### *导入文件*

octave 和 MatLab 一样用 load 导入数据文件，譬如

```
octave>  A = load data.txt ;
```

将把 data.txt 里的数据导入 octave 并赋给矩阵 A。对于图像文件，octave 用 imread 将图像导入并存为矩阵 img，

```
img = imread("jam.jpg") ;
```

在 octave 里显示图像很简单，用命令:

```
imshow(img) ;
```

除了 jpeg 和 png 格式的图像可以直接导入，其他格式的图像必须经过 ImageMagick 的 convert 函数转换后才可读入。ImageMagick 是命令行的强大的图像处理工具，convert 几乎涵盖了所有格式图像的转换。

如果你关心 imread 函数的源码，可以去读 /usr/local/share/octave/packages/image-1.0.8/imread.m，该函数把灰度图像导入为 MxN 矩阵，把彩色图像导入为 MxNx3 矩阵。具体的帮助文件，可以

```
help imread ;
```

或者来个更详细点儿的

```
help -i imread ;
```

---

### ***Octave** 与 **MatLab** 的一些小区别*

MatLab 用户转而使用 octave 几乎不需要什么培训，只是要一些小细节上注意一下。下面我们罗列一些 octave 和 MatLab 的区别。

布尔值的乘积

```
X = ones(2,2) ;
prod(size(X)==1)
```

MatLab 和 octave 的输出是不同的：

```
Matlab: ??? Function 'prod' is not defined for values of class 'logical'.
Octave: ans = 0
```

octave 输出为 0 的原因是 size(X) 为

```
ans =

   2   2
```

逻辑运算符、算术运算符

Octave 与 MatLab 兼容，甚至更为宽松。如，

| 运算 | Matlab | octave |
| --- | --- | --- |
| 或 | \| | “\|” 或者“\|\|” |
| 且 | & | & 或者 && |
| 否 | ~= | ~= 或者 != |

MatLab 用 x^2，octave 用 x^2 或者 x**2 表示 “x 的平方”。Octave 用 x**2 是为了照顾 GnuPlot 的用户。总而言之，octave 在运算符方面彻底兼容 MatLab，MatLab 用户放心大胆地用 octave 吧，但 octave 用户用 MatLab 的时候就要小心了。

C-风格的自动增量、赋值、屏幕打印

Octave 允许 C-风格的

```
i++ ; ++i ; i+=1 ;
printf('My result is: %d/n', 4)
```

而 MatLab 不认它们。MatLab 打印至屏幕和文件都用 fprintf 函数。

注意空格

octave 对空格是作为一个符号识别的，在列合并中短的列自然扩充，例如

```
A = ['123 ';'123'] ;
size(A)
```

的结果是 2 4，而 MatLab 则返回列合并有问题：

```
?? Error using ==> vertcat
```

另外，转置符号与矩阵之间如果有空格

```
[0 1] '
```

在 MatLab 里不允许，octave 则允许，且与 [0 1]' 的结果是一样的。

## 直方图内置函数 **hist**

octave 的 hist 为

```
hist (Y, X, NORM)
```

其中 NORM 为所有柱高之和。

## 导入空文件

MatLab 允许导入空文件，老版本的 octave 不允许，新版本的 octave-3.0.3 则允许。

## 行续符

MatLab 中用 `...' 做行续符，如用

```
A = rand (1, ...
          2) ;
```

表达

```
A = rand (1,2) ;
```

Octave 与 MatLab 兼容，除此之外，octave 还允许如下两种表示方法。

```
A = rand (1,
          2) ;
```

和

```
A = rand (1, /
          2) ;
```

## **if**、**for** 等环境的结束符

Octave 用

```
end{if,for, ...}
```

而 MatLab 则统一用 end。

## ***R* 和 *octave* 命令的对照表**

octave 和 R 联合起来用的时候，我们需要下面的命令对照表帮助我们理清楚它们的区别。“无”仅仅是说没有一个命令行的简单表示，并不代表不能表示。这种对比不是比

较谁更强大，而是为了记忆，无论是对 R 用户学习 octave 或者 octave 用户学习 R，都是有所裨益的。

| octave | R |
| --- | --- |
| | |
| 帮助 | |
| help -i | help.start() |
| help | help(help) |
| help sort | help(sort) |
| | demo() |
| lookfor plot | apropros('plot') |
| | help.search('plot') |
| 复数 | |
| 3+4i | 3+4i |
| i | 1i % R 把"i"视为变量名 |
| abs(3+4i) | Mod(3+4i) |
| arg(3+4i) | Arg(3+4i) |
| conj(3+4i) | Conj(3+4i) |
| real(3+4i) | Re(3+4i) |
| imag(3+4i) | Im(3+4i) |
| | |
| 向量、序列 | |
| 1:10 | 1:10 或 seq(10) |
| 1:3:10 | seq(1,10,by=3) |
| 10:-1:1 | 10:1 |
| 10:-3:1 | seq(from=10,to=1,by= -3) |
| linspace(1,10,7) | seq(1,10,length=7) |
| (1:10)+i | 1:10+1i |
| a=[2 3 4 5]; # 不显示结果 | a <- c(2,3,4,5) % 不用加分号 |

| | |
|---|---|
| a=[2 3 4 5] #显示结果 | (a <- c(2,3,4,5)) % 显示结果 |
| adash=[2 3 4 5]' ; | adash <- t(c(2,3,4,5)) |
| [a a] | c(a,a) |
| [a a*3] | c(a,a*3) |
| a.*a | a*a |
| a.^3 | a^3 |
| | |
| 向量的合并与重复 | |
| [1:4 a] | c(1:4,a) |
| [1:4 1:4] | rep(1:4,2) |
| 无 | rep(1:4,1:4) % 结果是：1 2 2 3 3 3 4 4 4 4 |
| 无 | rep(1:4,each=3) % 结果是：1 1 1 2 2 2 3 3 3 4 4 4 |
| a=1:100; | a <- 1:100 |
| a(2:100) | a[-1] % a 去掉第 1 个元素 |
| a([1:9 11:100]) | a[-10] % a 去掉第 10 个元素 |
| 无 | a[-seq(1,50,3)] % a 去掉第 1,4,7,...个元素 |
| | |
| 向量的赋值 | |
| a(a>90)= -44; | a[a>90] <- -44 |
| | |
| 向量的最大、最小 | |
| a=randn(1,4); | a <- rnorm(4) |
| b=randn(1,4); | b <- rnorm(4) |
| max(a,b) | pmax(a,b) |
| max([a' b']) | cbind(max(a),max(b)) |
| max([a b]) | max(a,b) |
| [m i] = max(a) | m <- max(a) ; i <- which.max(a) |

| | |
|---|---|
| "min" 类似 | |
| | |
| 向量的秩 | |
| ranks(rnorm(8,1)) | rank(rnorm(8)) |
| ranks(rnorm(randn(5,6))) | apply(matrix(rnorm(30),6),2,rank) |
| | |
| 矩阵的行合并与列合并 | |
| [1:4 ; 1:4] | rbind(1:4,1:4) |
| [1:4 ; 1:4]' | cbind(1:4,1:4) 或 t(rbind(1:4,1:4)) |
| [2 3 4 5] | c(2,3,4,5) |
| [2 3;4 5] | rbind(c(2,3),c(4,5)) % rbind() 合并行; cbind() 合并列 |
| [2 3;4 5]' | cbind(c(2,3),c(4,5)) 或 matrix(2:5,2,2) |
| a=[5 6]; | a <- c(5,6) |
| b=[a a;a a]; | b <- rbind(c(a,a),c(a,a)) |
| [1:3 1:3 1:3 ; 1:9] | rbind(1:3, 1:9) |
| [1:3 1:3 1:3 ; 1:9]' | cbind(1:3, 1:9) |
| 无 | rbind(1:3, 1:8) |
| | |
| 产生矩阵 | |
| ones(4,7) | matrix(1,4,7) 或 array(1,c(4,7)) |
| ones(4,7)*9 | matrix(9,4,7) 或 array(9,c(4,7)) |
| eye(3) | diag(1,3) % 对角线都为 1 的对角阵 |
| diag([4 5 6]) | diag(c(4,5,6)) % 对角线为 4,5,6 的对角阵 |
| diag(1:10,3) | 无 |
| reshape(1:6,2,3) | matrix(1:6,nrow=2) 或 array(1:6,c(2,3)) |
| reshape(1:6,3,2) | matrix(1:6,ncol=2) 或 array(1:6,c(3,2)) |
| reshape(1:6,3,2)' | matrix(1:6,nrow=2,byrow=T) |

| | |
|---|---|
| a=reshape(1:36,6,6); | a <- matrix(1:36,c(6,6)) |
| rem(a,5) | a %% 5 |
| a(rem(a,5)==1)= -999 | a[a%%5==1] <- -999 |
| a(:) | as.vector(a) |
| | |
| 矩阵中抽取元素 | |
| a=reshape(1:12,3,4); | a <- matrix(1:12,nrow=3) |
| a(2,3) | a[2,3] |
| a(2,:) | a[2, ] |
| a(2:3,:) | a[-1,] |
| a(:,[1 3 4]) | a[,-2] |
| a(:,1) | a[ ,1] |
| a(:,2:4) | a[ ,-1] |
| a([1 3],[1 2 4]) | a[-2,-3] |
| | |
| 矩阵赋值 | |
| a(:,1) = 99 | a[ ,1] <- 99 |
| a(:,1) = [99 98 97]' | a[ ,1] <- c(99,98,97) |
| | |
| 矩阵：转置、共轭 | |
| a' | Conj(t(a)) |
| a.' | t(a) |
| | |
| 矩阵：求和 | |
| a=ones(6,7) | a <- matrix(1,6,7) |
| sum(a) | apply(a,2,sum) |
| sum(a') | apply(a,1,sum) |
| sum(sum(a)) | sum(a) |

| | |
|---|---|
| cumsum(a) | apply(a,2,cumsum) |
| cumsum(a') | apply(a,1,cumsum) |
| | |
| 矩阵排序 | |
| a=rand(3,4); | a <- matrix(runif(12),c(3,4)) |
| sort(a(:)) | sort(a) |
| sort(a) | apply(a,2,sort) |
| sort(a') | apply(a,1,sort) |
| cummax(a) | apply(a,2,cummax) |
| | |
| 矩阵：最大、最小 | |
| a=randn(100,4) | a <- matrix(rnorm(400),4) |
| max(a) | apply(a,1,max) |
| [v i] = max(a) | v <- apply(a,1,max) ; i <- apply(a,1,which.max) |
| b=randn(4,4); | b <-matrix(rnorm(16),4) |
| c=randn(4,4); | c <-matrix(rnorm(16),4) |
| max(b,c) | pmax(b,c) |
| | |
| 矩阵的乘法 | |
| a=reshape(1:6,2,3); | a <- matrix(1:6,2,3) |
| b=reshape(1:6,3,2); | b <- matrix(1:6,3,2) |
| c=reshape(1:4,2,2); | c <- matrix(1:4,2,2) |
| v=[10 11]; | v <- c(10,11) |
| w=[100 101 102]; | w <- c(100,101,102) |
| x=[4 5]' ; | x <- t(c(4,5)) |
| a*b | a %*% b |
| v*a | v %*% a |
| a*w' | a %*% w |

| | |
|---|---|
| b*v' | b %*% v |
| v*x | x %*% v 或 v %*% t(x) |
| x*v | t(x) %*% v |
| v*a*w' | v %*% a %*% w |
| v .* x' | v*x 或_ x*v |
| a .* [w ;w] | w * a |
| a .* [x x x] | a * t(rbind(x,x,x)) 或 a*as.vector(x) |
| v*c | v %*% c |
| c*v' | c %*% v |
| | |
| 其他矩阵操作 | |
| a=rand(3,4); | a <- matrix(runif(12),c(3,4)) |
| fliplr(a) | a[,4:1] |
| flipud(a) | a[3:1,] |
| a=reshape(1:9,3,3) | a <- matrix(1:9,3) |
| vec(a) | as.vector(a) |
| vech(a) | a[row(a) <= col(a)] |
| size(a) | dim(a) |
| | |
| 网格 | |
| [x y]=meshgrid(1:5,10:12); | 无 |
| | |
| 查找 | |
| find(1:10 > 5.5) | which(1:10 > 5.5) |
| a=diag([4 5 6]) | a <- diag(c(4,5,6)) |
| find(a) | which(a != 0) % which() 的变元是布尔变元 |
| [i j]= find(a) | which(a != 0,arr.ind=T) |
| [i j k]=find(a) | ij <- which(a != 0,arr.ind=T); k <- a[ij] |

| | |
|---|---|
| | |
| 读文件 | |
| load foo.txt | f <- read.table("~/foo.txt") |
| | f <- as.matrix(f) |
| 写文件 | |
| save -ascii bar.txt f | write(f,file="bar.txt") |
| | |
| 图形输出 | |
| gset output "foo.eps" | postscript(file="foo.eps") |
| gset terminal postscript eps | plot(1:10) |
| plot(1:10) | dev.off () |
| | |
| 赋值 | |
| string="a=234"; | string <- "a <- 234" |
| eval(string) | eval(parse(text=string)) |
| | |
| 产生随机数 | |
| 均匀分布 | |
| rand(10,1) | runif(10) |
| 2+5*rand(10,1) | runif(10,min=2,max=7) 或 runif(10,2,7) |
| rand(10) | matrix(runif(100),10) |
| 正态分布 | |
| randn(10,1) | rnorm(10) |
| 2+5*randn(10,1) | rnorm(10,2,5) |
| rand(10) | matrix(rnorm(100),10) |
| beta 分布 | |
| hist(beta_rnd(4,2,1000,1) | hist(rbeta(1000,shape1=4,shape2=10)) 或 hist(rbeta(1000,4,10)) |

| | |
|---|---|
| | |
| FOR 循环 | |
| for i=1:5; disp(i); endfor | for(i in 1:5) {print(i)} |
| | |
| 多项式的根 | |
| roots([1 2 1]) | polyroot(c(1,2,1)) |
| polyval([1 2 1 2],1:10) | 无 |
| | |
| 集合论 | |
| a = create_set([1 2 2 99 2 ]) | a <- sort(unique(c(1,2,2,99,2))) |
| b = create_set([2 3 4 ]) | b <- sort(unique(c(2,3,4))) |
| intersect(a,b) | intersect(a,b) |
| union(a,b) | union(a,b) |
| complement(a,b) | setdiff(b,a) |
| any(a == 2) | is.element(2,a) |
| | |
| 绘图 | |
| a=rand(10); | a <- array(runif(100),c(10,10)) |
| help plot | help (plot) and methods(plot) |
| plot(a) | matplot(a,type="l",lty=1) |
| plot(a,'r') | matplot(a,type="l",lty=1,col="red") |
| plot(a,'x') | matplot(a,pch=4) |
| plot(a,'—') | matplot(a,type="l",lty=2) |
| plot(a,'x-') | matplot(a,pch=4,type="b",lty=1) |
| plot(a,'x—') | matplot(a,pch=4,type="b",lty=2) |
| semilogy(a) | matplot(a,type="l",lty=1,log="y") |
| semilogx(a) | matplot(a,type="l",lty=1,log="x") |
| loglog(a) | matplot(a,type="l",lty=1,log="xy") |

| | |
|---|---|
| plot(1:10,'r') | plot(1:10,col="red",type="l") |
| hold on | matplot(10:1,col="blue",type="l",add=T) |
| plot(10:-1:1,'b') | |
| grid | grid() |
| a=randn(10); | a <- matrix(rnorm(100),nr=10) |
| contour(a) | contour(a) |
| contour(a,77) | contour(a,nlevels=77) ; filled.contour(a) |
| mesh(rand(10)) | persp(matrix(runif(100),10),theta=30,phi=30,d=1e9) |
| | |
| 文件与操作系统 | |
| system("ls") | system("ls") |
| pwd | getwd() |
| cd | setwd() |

### *R 读入 octave 导出的数据*

统计计算软件 R 的 foreign 包提供了函数 read.octave，可以读入 octave 用命令 save -ascii 创建的文本数据文件，且支持变量的大多数通用类型，包括标准的原子型（复矩阵， N 维数组，字符串，布尔矩阵等）和 递归式（结构体，单元和列表）。

在 octave 中用 save -ascii 保存的矩阵数据，也可以在 R 中用命令 read.table 导入，然后用 as.matrix() 强制为 R 中的矩阵使用。我比较倾向于这种方法。这样我们就能充分利用 octave 擅长矩阵计算和 R 擅长统计计算的优势，将二者联合起来使用。我们将详细介绍 octave 读入图像文件，输出能被 R 处理的矩阵数据 。

下面举个例子：我们投掷一枚硬币，已知正面出现的概率为 p，恰好掷出 R 正面所用的次数 N 是我们要考察的，我们做 E 次随机试验，看看 N 的经验分布情况。

```
## File        : toss.m
## Purpose     : The numbers of tossing to get R heads
## Author      : Jiangsheng Yu (yujs@pku.edu.cn)
## Data        : 11-26-2008
## Available : http://icl.pku.edu.cn/member/yujs/Computing.htm
## Usage       : run toss.m

more off ;    ## turn the pagination off
E = 10000;    ## the number of experiments
result = zeros(E,1);   ## the sequence of E results
R = 6   ;     ## required number of heads
p = 0.3 ;     ## the probability of head
H = 0 ;       ## no heads at the beginning
```

```
N = 0 ;       ## no tosses at the beginning
for i = 1:E
    do
          ## if head, outcome=1; otherwise, outcome=0
          outcome = (rand(1,1) < p) ;
          H += outcome ; ## the total number of heads
          N += 1 ;       ## the total number of tosses
        until ( H >= R ) ## until R heads
        result(i,1) = N ;
        N = 0 ;
        H = 0 ;
endfor
hist (result,40,1) ;
```

对于 p=0.3 ， R=2 做 E=10000 次随机试验得到 N 的直方图如下：

掷出 R 个正面所用次数的直方图

对于 p=0.3 ， R=6 做 E=10000 次随机试验得到 N 的直方图如下：

掷出 R 个正面所用次数的直方图

我们把结果保存为 result.data，再读到 R 中处理这些数据。

```
> x = result' ;
> save result.dat x ;
```

在 R 中我们读入数据，然后画出直方图。

```
> library(foreign)
> a <- read.octave("result.dat")
> hist(a$x, freq= FALSE, col="blue", border="pink")
```

得到 p=0.3 ， R=6 的直方图：

掷出 R 个正面所用次数的直方图