第 9 讲 数据分析与数值微积分

（第 6 章　MATLAB 数据分析与多项式计算）

（第 7 章　MATLAB 数值微分与积分）

目的:

一、掌握概率论与数理统计相关函数。

二、掌握导数值的数值计算方法。

三、掌握积分的数值计算方法。

一、掌握概率论与数理统计相关函数。

1、排列组合

（1）求阶乘 **factorial(n)**，例如 factorial(3)=3!

（2）求组合数 **nchoosek(n,k)**，例如 $C_5^3=\mathrm{nchoosek}(5,3)$

（3）求排列数，无专用函数，需用 $A_n^k=n!\cdot C_n^k$ 计算；

例如 $A_6^4=\mathrm{nchoosek}(6,4)*\mathrm{factorial}(4)$

2、伪随机数的产生

在做研究时常常需要生成实验模拟数据，此时需要用到大量伪随机数，matlab 提供了很多生成伪随机数的函数，例如 normrnd(1,4,1,1000)可以生成一个服从 $N(1,16)$ 的正态分布的 1*1000 的随机数向量。

生成函数的调用格式通常为 **'name'rnd(para,m,n)**，其中"name"是分布名称，para 是对应参数，m，n 是生成矩阵的行列列数。下面是一些常见的随机数生成函数，**在后面的练习中，大家可以用下面函数产生对应分布的随机数矩阵，然后再对该矩阵进行其它运算。**

分布名称，对应的随机数函数及其调用方式：

bino ， **binornd(N,P,m,n)**：参数为 N 和 P 的二项分布随机数；

unid， **unidrnd(N,m,n)**：整数$1,2,\cdots,N$ 上的均匀分布随机数（离散）；

unif， **unifrnd(A,B,m,n)**：区间(A,B)上的均匀分布随机数（连续）；

poiss， poissrnd(lambda,m,n)：参数为 lambda 的泊松分布随机数;

norm， normrnd(mu, sigma, m, n)：参数为 mu ,sigma 的正态分布随机数;

chi2， chi2rnd(N,m,n)：自由度为 N 的 χ^2 分布随机数;

t， trnd(N,m,n)：自由度为 N 的 t 分布随机数;

f， frnd(N1,N2,m,n)：第一自由度为 N1，第二自由度为 N2 的 F 分布随机数;

exp， exprnd(lambda,m,n)：参数为 lambda 的指数分布随机数;

randperm(n)：生成 1 到 n 的随机排列;

randperm(n,k)：从 1 到 n 中取随机取 k 个数出来随机排列。

rand(m,n)：区间 $(0,1)$ 上的均匀分布随机数;

randn(m,n)：标准正态分布随机数;

练习：用以上随机数生成函数生成一些常见分布的数据，并用 histogram 函数绘制数据直方图，验证数据分布。

3、计算概率密度函数和概率分布函数的函数值（由于大家还没学习概统，这个暂时不做）

3.1 计算概率函数密度函数值的方法有两个，

方法 1：namepdf(k，para)

方法 2：pdf('name',k,para)

其中 name 是上面第 2 部分中提到的分布的名称，k 是为随机变量取值（如果 k 是矩阵，那么对矩阵中的每个元素求值），para 是参数值。例如，正态分布的名称是 norm，对应的概率密度函数是 normpdf；二项分布的名称是 bino，对应的概率函数是 binopdf。

pdf 是 probability distribution function 的缩写。

例 1：设 $X \sim B(10,0.1)$ 求 $P(X=1)$（此处 k=1，参数 para 是 10，0.1）

```
>> p1=binopdf(1,10,0.1)    %也可以使用 pdf('bino',1,10,0.1)计算，结果相同
>>p1=0.3874
```

上面结果计算的是概率值：$P(X=1)=C_{10}^{1}(0.1)^1\cdot(0.9)^9=10\cdot0.1\cdot(0.9)^9$

练习 1：设泊松分布参数 $\lambda=5$ 求概率 $P(X=3)$ 并用公式 $P(X=k)=\dfrac{\lambda^k}{k!}e^{-\lambda}$ 验证结果。

注：对离散型，概率函数值就是随机变量取某值的概率。

例 2：设 $Y \sim e(2)$，求 $f(3)$

>> a=exppdf(3,2)   %也可以使用 pdf('exp',3,2)计算，结果相同.

>>a=0.1116

指数分布的密度函数是 $f(x)=\begin{cases}\frac{1}{\lambda}e^{-\frac{1}{\lambda}x} & x>0 \\ 0 & x\le 0\end{cases}$，所以 $f(3)=\frac{1}{2}\cdot e^{-\frac{1}{2}\cdot 3}=0.1116$。

注：对连续型，概率函数值是密度函数的函数值不是概率值。

练习 2：设 k=-3:0.05:3，**计算**标准正态分布密度函数值，并利用函数值**绘制**概率密度函数图。

3.2 计算分布函数值的方法（即累积概率值）：（由于大家还没学习概统，这个暂时不做）

方法 1：namecdf(k,para)

方法 2：cdf('name',k,para)

例 1：设 $X \sim N(1,4)$ 求 $P(X \le 1)$

>>a=normcdf(1,1,2)   %注意这里 $N(1,4)$ 中 $4=2^2$，参数是 2。或者 cdf('norm',1,1,2)

>>a=0.5000

例 2：设 $X \sim B(1000,0.015)$，求 $P(X \le 15)$

>>a=binocdf(15,1000,0.015)   %或者使用 cdf('bino',15,1000,0.015)

>>a=0.5681

练习 3：设 k=-5:0.05:5，**计算**标准正态分布的分布函数值，并**绘制**分布函数图。

4、常规运算和数字特征

（1）max 函数

max(A):如果 A 是向量，返回 A 的最大值；如果 A 是矩阵，返回 A 的每列的最大值。如果想得到整个矩阵元素的最大值可以使用 max(max(A))或者 max(A(:))或者 max(A,[ ],'all')。

[m,i]=max(A)：m 是最大值，i 是 m 在 A 中的位置。

例如：$A=\begin{bmatrix}6 & 9 & 7\\0 & 6 & 3\\8 & 7 & 6\end{bmatrix}$

>> [m,i]=max(A)

m=8　9　7

i = 3　1　1

>>max(max(A))

ans=9

C=max(A,B)： 结果 C 是 A 与 B 对应位置的元素取最大值所构成的矩阵。

max(A,[ ],dim): 按维度 dim 求最大值，dim 默认值为 1 代表列，2 代表行；此处中括号[ ]不能省略，不能省略的原因是：比如 max(A,2)按照 max(A,B)的运算机制，是拿 A 中每个元素与 2 作比较，取最大值构成新矩阵。所以为了求 A 的行最大值，需要使用 max[A,[ ],2]，意思是先拿 A 与空矩阵[ ]取最大值（结果仍是 A）然后再按行取最大值。

练习 4：生成矩阵，完成上面几种计算

（2）min 函数。使用方法同 max 函数

练习 5：用练习 4 中生成的矩阵，完成对应计算

（3）mean 均值/期望函数

格式：mean(A)：如果 A 是向量，则求 A 的均值；如果 A 是矩阵，则默认按列求每列均值。

mean(A,'all')：求 A 所有元素均值。

mean(A,dim)：按指定维度 dim 求均值，dim 设为 1 表示列（默认），设为 2 表示行。

mean(A,vecdim)：如果 A 是三维矩阵，vecdim 设为[1,2]表示按 x,y 页求均值；设为[1,3]表示按 x，z 页求均值，设为[2,3]按 y，z 页求均值。

例：>>A(: , : , 1)=fix(10*rand(3,4));

>> A(: , : , 2)=fix(10*rand(3,4));

>> A(: , : , 3)=fix(10*rand(3,4));

则 A 是一个 3*4*3 的立方体三维矩阵，其中按高度分成三页，每页是一个 3*4 的矩阵。mean(A,[1,2])将计算不同高度的每页元素的均值。可以设 B=A(:, :, 1)后 mean(B,'all')验证。

练习 6：生成一个三维矩阵，求 mean 值，熟悉维度设置。

(4) sum 求和

格式：sum(A)、sum(A,'all')、sum(A, dim)、sum(A, vecdim)用法与 mean 类似。

练习 7：生成一个随机数构成的三维矩阵 A，求 sum 值，熟悉维度设置。

(5) 排序[s,q]=sort(A,dim,mode)

dim 默认是 1 即按列，2 是按行；mode 默认是'ascend'即升序，'descend'是降序.

s 是排序后的矩阵，q 与 s 矩阵同型，记录 s 矩阵中的元素在原矩阵中列或者行中的位置

练习 8：生成一个矩阵，完成排序

(6) median(x) 求 x 的中位数，即左右两边各为 50%的中间值

median(x)、median(x,'all')、median(x,dim)、median(x,vecdim)

练习 9：生成某各分布的随机数字矩阵 A，求所有数的中位数 m。

注：用 c=(A<m)，sum(c,'all')与 A 中数据个数做比较，验证结果。

(7) range 求极差

range(x)、range(x,'all')、range(x,dim)、range(x,vecdim)。

(8) var 求样本方差（由于大家还没学习概统，这个暂时不做）

var(A)、var(A,w)、var(A,w,'all')、var(A,w,dim)、var(A,w,vecdim)

w=0(默认值)表示分母是（n-1)，w=1 表示分母是 n。

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2 \ , \quad s^2 = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2$$

练习 10：…

(9) std 求样本标准差，即方差的根值；

std(A)、std(A,w)、std(A,w,'all')、std(A,w,dim)、std(A,w,vecdim)

w=0(默认值)表示分母是（n-1)，w=1 表示分母是 n。

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2} \ , \quad s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2}$$

练习 11：…

(10) cov 样本协方差矩阵（由于大家还没学习概统，这个暂时不做）

cov(A),   求 A 的各列之间的协方差矩阵，即 A 的列表示随机变量，行表示观测值。

cov(A,w)，w=0(默认值)表示分母是（n-1)， w=1 表示分母是 n

cov(A,B)，求 A 向量和 B 向量间的协方差矩阵（是一个 2*2 矩阵）

$$\operatorname{cov}(X,Y)=\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i-\overline{x}\right)\left(y_j-\overline{y}\right),\quad \operatorname{cov}(X,Y)=\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\overline{x}\right)\left(y_j-\overline{y}\right)$$

练习 12：…

（11）corrcoef 样本相关系数（由于大家还没学习概统，这个暂时不做）

corrcoef(A),求 A 的各列之间的样本相关系数矩阵，即 A 的列表示随机变量，行表示观测值

corrcoef(A,B)，求 A 向量和 B 向量间的样本相关系数矩阵（是一个 2*2 矩阵）

练习 13：

（12）cumsum 累计求和

练习 14：

（13）cumprod 累计求积

练习 15：

基本运算的综合练习：

（1）学习从 excel 文件中导入班级考试成绩数据；

（2）对成绩进行汇总，包含：

单科成绩排序、单科成绩平均分、单科成绩中位数、单科成绩方差、单科成绩极差；

个人成绩总分、个人成绩平均分、个人成绩排序，个人总分后总分全班排序、个人成绩极差。

（3）对单科成绩做成绩直方图。

5、最大似然估计（mle 函数）（由于大家还没学习概统，这个暂时不做）

使用 **mle** 函数对指定分布中的参数进行最大似然估计，其格式为：

[ps,ps_ci]=mle(X,'Distribution',name,'alpha',alphavalue,Pn); 或者

[ps,ps_ci]=mle(X,'pdf',pdfname_handler,'start',startvalue)

其中：ps 和 ps_ci 是待估计参数似然估计值和置信区间，X 为样本数据**向量**，name 是指定分布的名称（如正态分布的名称为 norm，泊松分布的名称为 poiss，其它部分分布名称见前面伪随机数产生），alpha 是做区间估计时的置信水平，默认 0.05；pdfname_handler 是概率函数

句柄，例如泊松分布函数句柄格式是：@poisspdf；startvalue 对参数的猜测值。

例如：指数分布参数的最大似然估计(用格式 1)

```
x=exprnd(0.01,1,1000)   %产生服从 e(0.01)的伪随机数向量 1*1000，用于检验 mle 函数
[ps,ps_ci]=mle(x,'distribution','exp','alpha',0.05);   %alpha 可以缺省，默认 0.05
```

```
ps=
    0.0098
ps_ci=
    0.0092
    0.0104
```

例如：指数分布参数的最大似然估计(用格式 2)

```
x=exprnd(0.01,1,1500)   %产生服从 e(0.01)的伪随机数向量 1*1500，用于检验 mle 函数
[ps,ps_ci]=mle(x,'pdf',@exppdf,'start',0.02,'alpha',0.05);   %alpha 可以缺省，默认 0.05
```

```
ps=
    0.0098
ps_ci=
    0.0093
    0.0103
```

注：可以看到，由于样本数据增多，在同样置信水平 0.05 下，置信区间宽度变窄了。

注：（1）直接使用 **a=mle(data)**，返回的 a 是数据 data 的均值和根方差，可以用这个命令替代 **mean** 和 **std**。

（2）这里的函数句柄**@exppdf** 可以使用自己定义的函数，比如自己定义密度函数 **myexp**

```
function y=myexp(x,lambda)
    if x<=0
          y=0;
    else
          y=(1/lambda)*exp(-x/lambda);
    end
end
```

然后运行 [ps,ps_ci]=mle(x,'pdf',@myexp,'start',0.02,'alpha',0.05)。结果和使用系统函数

差不多。这样我们可以用这个命令求解任意密度函数参数的似然估计。

练习 16：生成 1000 个服从 $\lambda=2$ 的泊松分布的随机数，当置信水平为 0.05 时，对该分布中的参数进行估计。

```
>>x=…
>>[a,b]=mle(x,'distribution','poiss')
```

```
>>x=…
>>[a,b]=mle(x,'pdf',@poisspdf,'start',1.5)   %必须指定一个 start 值。
```

注：除了 mle 这个普遍适用的用于参数的最大似然估计函数外，matlab 还提供了一些常见分布密度函数参数的专有估计函数，比如 normfit（专用于正态分布参数估计），expfit（专用于指数分布的参数）等等，通常这些函数的名称都是 namefit，它们的使用方法和 mle 类似，感兴趣的同学可以查看帮助文档。

二、掌握导数值的数值计算方法。

1、计算差分和导数值

由导数定义 $f'(x)=\lim\frac{\Delta y}{\Delta x}$，$f'(x)\approx\frac{\Delta y}{\Delta x}$，所以当 Δx 不大时，$f'(x)$ 近似等于 y 与 x 的差分比。

Matalb 中计算差分的函数是 **diff(y)**，运算结果是 y 中元素后项减去前项产生的向量，差分后产生的向量比原向量少一个维度。使用 **diff(y,2)**计算二阶差分，等同于 **diff（diff(y)）**即对一阶差分向量再求差分。

练习 1：产生具有 10 个元素的向量 *x*，其元素是两位随机整数，求 *x* 的 1~3 阶差分。

给出程序及运行结果：

```
x=(99-10)*rand(1,10)+10; %10<x<99，实数
x=round(x) %10≤x≤99，四舍五入取整
Dx=diff(x)
D2x=diff(x,2)
D3x=diff(x,3)
```

```
命令行窗口
>> Untitled2
x =
    24    96    95    53    81    23    48    92    81    95
Dx =
    72    -1   -42    28   -58    25    44   -11    14
D2x =
   -73   -41    70   -86    83    19   -55    25
D3x =
    32   111  -156   169   -64   -74    80
>>
```

练习 2：求 $y=\sin(x)$ 在 $x=\pi/4$ 处的数值导数。（由公式知导数值为 $\frac{\sqrt{2}}{2}$）

（1）用差分近似

（2）用 while 循环语句利用导数定义 $f'\left(\frac{\pi}{4}\right)=\lim\limits_{h\to 0}\frac{f\left(\frac{\pi}{4}+h\right)-f\left(\frac{\pi}{4}\right)}{h}$ 求值；

（3）对 $y=\sin(x)$ 给 x 赋值，对产生的数据点用 polyfit 进行多项式拟合，对生成的多项式使用 polyder 求导函数，对导函数使用 polyval 求在 $\frac{\pi}{4}$ 处的取值。

```
%（1）用差分近似
x=pi/4:0.01:11*pi/40;
y=sin(x);
dy=diff(y); dx=diff(x);
f1=dy(1)/dx(1);
s1=['导数近似值为：',num2str(f1)];
disp(s1);
```

```
%(2)利用导数定义求导（复习循环语句）
  f=@(x)sin(x);
  x0=pi/4;
  h=1/2;
  f0=(f(x0+h)-f(x0))/h;   %求 h=1/2 时的Δy / Δx
  h=h/2;
  f1=(f(x0+h)-f(x0))/h;   %求 h=h/2=1/4 时的Δy / Δx
  while abs(f0-f1)>1e-6   %如 f0 和 f1 的差距过大说明极限还不收敛，需要继续
       f0=f1;
       h=h/2;
       f1=(f(x0+h)-f(x0))/h;
end
s2=['用定义求出的导数值为',num2str(f1)];
disp(s2);
```

```
%(3)利用多项式拟合求近似导数值
x=pi/5:0.01:pi/3;
y=sin(x);
P=polyfit(x,y,5);
Q=polyder(P);
df=polyval(Q,pi/4);
s3=['用多项式拟合求出的导数值为',num2str(df)];
disp(s3);
```

三、掌握积分的数值计算方法。

定积分的数值计算是根据定积分定义求积分值，根据定积分几何意义，可以用不同的分割方法求定积分值，常用的有矩形法、梯形法、变步长辛普森法、自适应辛普森法等等。

Matlab 中对应的数值积分函数：梯形法 s=trapz(x,y)、变步长辛普森法 quad(fun,a,b)、自适应法 integral(fun,a,b)、二重积分的 integral2(fun, a,b,c,d)、三重积分的 integral3(fun,a,b,c,d,e,f)等等。

比较常用的是：

integral(fun,a,b)定积分, integral2(fun, a,b,c,d)二重积分、integral3(fun,a,b,c,d,e,f)

注：（1）函数调用中**“fun”**是被积函数的函数句柄，在 matlab 中，凡是参数显示是**“fun”**的都是指函数句柄。

如果设置被积函数时采用的是匿名函数方式，比如：f1=@(x)(x.^2.*sin(x))，那么 f1 的类型本身就是函数句柄，积分时在 fun 的位置填 f1 就可以了——**s=integral(f1, 0, 1)**。

如果设置被积函数时使用的是 m 文件方式，比如：

```
function y=lab1(x)
    y=x.^2.*sin(x);
end
```

那么函数名 lab1 不是函数句柄，需要在它前面加@符号才表示函数句柄，所以在积分时需要在 fun 的位置填@lab1—— **s=integral(@lab1, 0, 1)**。

（2）由于是采用定义进行数值积分，所以函数表达式里常常使用的是**点运算**，做练习时可以观察使用点运算和不使用点运算时系统的提示。

（3）计算二重或三重积分时参数中的 c,d,e,f 可以是函数句柄，用于处理非矩形区域积分。

（4）integral 系列函数需要知道被积函数表达式才能求积分，所以如果只知道曲线上某些点处的 x 与 y 的值，那么可以选用类似梯形法 s=trapz(x,y)参数显示是 x 和 y 的积分函数求数值积分。

练习 1：$I_1 = \int_0^{2\pi} \sqrt{\cos t^2 + 4\sin^2(2t) + 1}\mathrm{d}t$ 的近似值。

给出程序及运行结果：

```
%用m文件设计被积函数
function y=f1(t)
y=sqrt(cos(t.^2)+4*sin(2*t).^2+1);
end
```

```
%在脚本或命令窗口中运行
s1=integral(@f1,0,2*pi)   %使用 m 文件定义被积函数，调用时函数名前需要加“@”。
f2=@(t)sqrt(cos(t.^2)+4*sin(2*t).^2+1);
s2=integral(f2,0,2*pi)    %使用匿名函数定义被积函数，调用时函数名前不用加”@”。
s3=integral(@(t)sqrt(cos(t.^2)+4*sin(2*t).^2+1),0,2*pi) %可以直接在 fun 位置填写匿名函数。
```

```
%在脚本或命令窗口中运行
t=linspace(0,2*pi,1000);
y=sqrt(cos(t.^2)+4*sin(2*t).^2+1);
s4=trapz(t,y);  %这里假设只知道离散点(t,y)，使用 trapz 求积分
```

练习 2：使用多种定义被积函数方式求积分 $I_2 = \int_0^{2\pi} \frac{\ln(1+x)}{1+x^2}\mathrm{d}x$。

练习 3：求二重定积分 $\iint\limits_D xe^y d\sigma$，其中 D 是矩形区域 $\begin{cases} 0 \le x \le 1 \\ 2 \le y \le 4 \end{cases}$。

```
%用m文件设计被积函数
function z=fxy(x,y)
z=x.*exp(y);
end
```

```
>>s1=integral2(@fxy,0,1,2,4) %使用m文件定义被积函数，调用时函数名前需要加“@”。
```

```
>>fxy=@(x,y)x.*exp(y); %使用匿名函数定义被积函数，调用时函数名前不用加”@”。
>>s2=integral2(fxy,0,1,2,4)
```

```
>>s3=integral2(@(x,y)x.*exp(y),0,1,2,4) %可以直接在fun位置填写匿名函数。
```

练习 4：求二重积分 $\iint\limits_{D}\left(x^{2}+y\sin x\right)d\sigma$，其中 D 是矩形区域 $\begin{cases}1\le x\le 3\\ 0\le y\le 2\end{cases}$

练习 5：求 $I_1=\int_0^1\int_0^{x+1}\frac{1}{\sqrt{x^2+y^2}}\mathrm{dydx}$ **（非矩形区域积分）**

```
%用m文件设计被积函数
function z=f1(x,y)
z=1./sqrt(x.^2+y.^2);
end
```

```
function y=yup1(x)   % y 的上限是函数，需要用 m 文件或者匿名函数方式定义。
  y=x+1;
end
```

```
>>s1=integral2(@f1,0,1,0,@yup1)   %被积函数和上限都使用 m 文件定义
```

```
>>f2=@(x,y) 1./sqrt(x.^2+y.^2);
>>yup2=@(x)x+1;
>>s2=integral2(f2,0,1,0,yup2)   %被积函数和上限都使用匿名函数定义
```

```
>>s3=integral2(@(x,y) 1./sqrt(x.^2+y.^2),0,1,0,@(x)x+1) %可以在积分中直接使用匿名函数
```

```
>>s4=integral2(@f1,0,1,0,yup2)   %可以混合使用m文件和匿名函数
>>s5=integral2(f2,0,1,0,@yup1)
```

练习 6：用多种方式定义被积函数和上下限求 $I_2=\int_{-1}^{1}dx\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}}e^{x^2+y^2}\mathrm{d}y$。

练习 7：求 $I=\int_{-\infty}^{0}dx\int_{-100}^{0}dy\int_{-100}^{0}\frac{10}{x^2+y^2+z^2+2}dz$（正无穷：inf，负无穷:-inf）

练习 8：$\iiint\limits_{\Omega}xyzdv$，其中 Ω 是曲面 $x+y+z=1$ 与三个坐标平面所围空间。