**Matlab图形可视化**

**一、实验目的**

掌握运用Matlab进行二维和三维图像输出的基本方法，能够根据需要完成不同形式的二维图形输出以及三维图形输出。

**二、实验内容**

完成Matlab多窗口绘图和同窗口多条函数曲线的绘制；绘制给定函数的二维线图，双对数坐标图和极坐标图，并添加必要的标注；绘制给定函数的三维曲线，在一定范围内绘制三维网格图及表面图。

**三、实验步骤**

（1）练习Matlab二维绘图函数

熟悉基本函数格式：plot(x,y,‘s’) 参数： x-横坐标 y-纵坐标 s-颜色线型描述 。表1列出了plot函数参数情况。

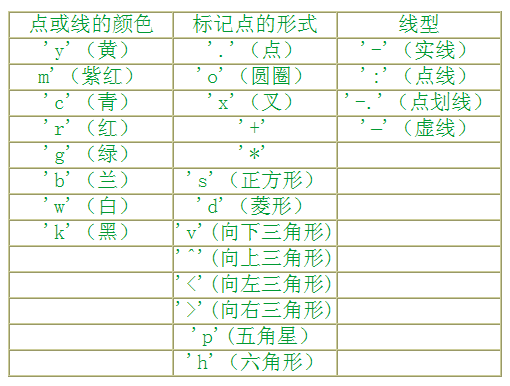


表1 plot函数常规参数

子图绘制函数格式：subplot(row,col,num) 参数： row、col：将整个窗口切分为row\*col个子窗口； num：当前选中第num个窗口。

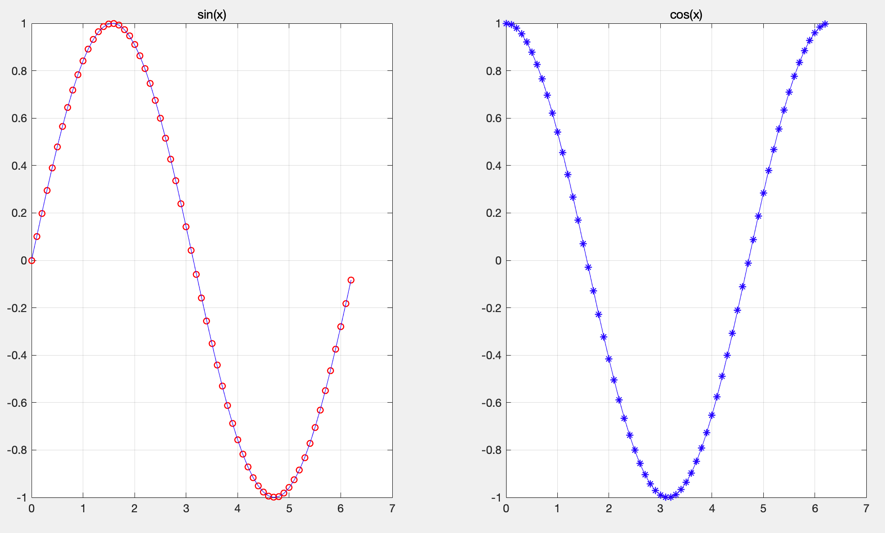
其他绘图常用标注：标题-title；坐标轴变量-xlabel、ylabel 图标-legend。

函数polar(t,r)用来绘制极坐标图，t表示极角，r表示极径，同时也可以添加线型描述参数。

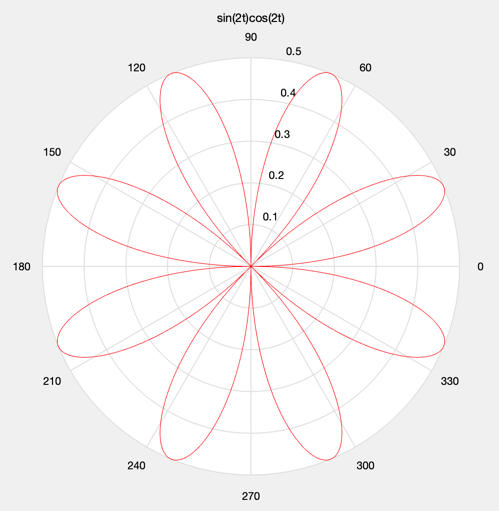
函数loglog(x,y)用来绘制双对数坐标图，同时可以添加线型描述参数。

**任务一**：

1. 编程实现在[0,2]上，左边窗口用红色圆圈散点和蓝色实线绘制sin函数；右边窗口用蓝色星形散点和蓝色实线绘制cos函数，标题分别为sin(x)和cos(x)。(sincos.m)
2. clc;close all;clear;
3. t=0:0.1:2\*pi;
4. subplot(1,2,1);
5. plot(t,sin(t),'ro');
6. hold on
7. plot(t,sin(t),'b');
8. title('sin(x)');
9. subplot(1,2,2);
10. plot(t,cos(t),'b\*');
11. hold on
12. plot(t,cos(t),'b');
13. title('cos(x)');



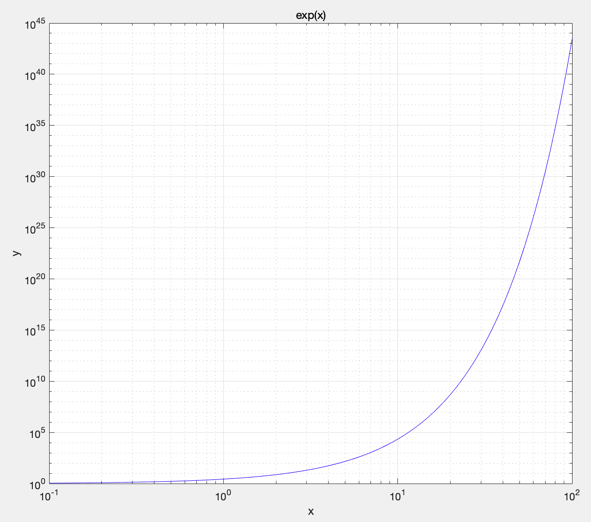
1. 在[0,2]上绘制的极坐标图,线型为红色实线。
2. clc;close all;clear;
3. theta=linspace(0,2\*pi,1000);
4. r=sin(2\*theta).\*cos(2\*theta); %点乘，每个元素进行相乘
5. polar(theta,r,'r');
6. title('sin(2t)cos(2t)');



c)在[]上绘制e指数函数的双对数坐标图，加网格显示，并用蓝色实线绘制,坐标轴标签分别为X和Y。

\*\*用到logspace（a,b,n）函数生成从10^a到10^b对数等分行向量，n为份数，省略时默认等于50。

1. clc;close all;clear;
2. t=logspace(-1,2,100);
3. loglog(t,exp(t),'b');
4. grid on;
5. xlabel('x');
6. ylabel('y');
7. title('exp(x)');



（2）练习Matlab三维绘图函数

熟悉三维曲线绘制函数基本格式： plot3(x,y,z,’s’)，包含一个自变量。

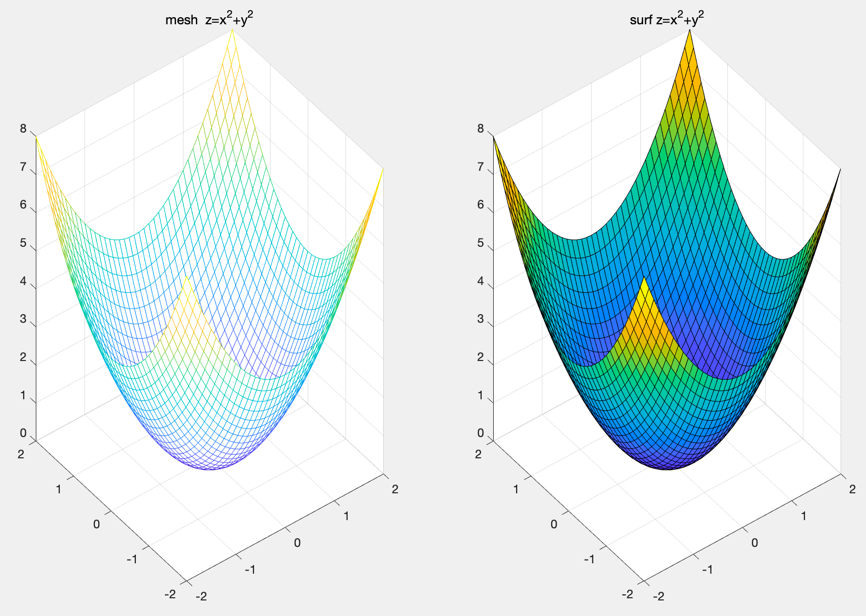
三维曲面绘制函数基本格式：mesh(x,y,z)、surf(x,y,z) ，包含两个自变量。注意：surf与mesh的区别在于网格的填充方式不同。x,y,z的数据要求，应该是二维的网格数据，通常与meshgrid指令联合使用。

**任务二：**在x([-2 2])×y([-2 2])的范围内绘制 z=x2+y2函数的曲面。（qumian.m）

* 获取二维网格数据:[x,y]=meshgrid(-2:0.1:2, -2:0.1:2)
* 函数表达式
* 绘制三维曲面：mesh()(与surf()绘图效果进行比较)

**四、实验结果**

1. [x,y]=meshgrid(-2:0.1:2,-2:0.1:2);%使用meshgrid得到带范围的二维网格
2. z=x.^2+y.^2; %按元素求幂
3. subplot(1,2,1);
4. mesh(x,y,z);
5. title('mesh  z=x^2+y^2');
6. subplot(1,2,2)
7. surf(x,y,z);
8. title('surf z=x^2+y^2');

****

**Matlab数据统计与分析**

**一、实验目的**

运用Matlab对样本观测数据进行数据统计计算以及已知分布参数进行估计，多项式计算以及曲线拟合。

**二、实验内容**

（1）给定一组观测样本数据某几个班级体育测试成绩，按照**班级**计算身高的均值，标准差，最小值和最大值统计量；提取身高，体重，肺活量，以及各类体育项目成绩，计算他们之间的相关系数矩阵。

（2）给定某厂生产的滚珠直径样本数据，假定服从正态分布，用normfit函数估计分布参数。

（3）根据已知系数向量构造多项式，求多项式的根以及多项式在X 处的值。

（4）给定一组实验数据{}, 求一个n次多项式，使其曲线能够逼近由向量x和y的对应分量构成的数对.

**三、实验步骤**

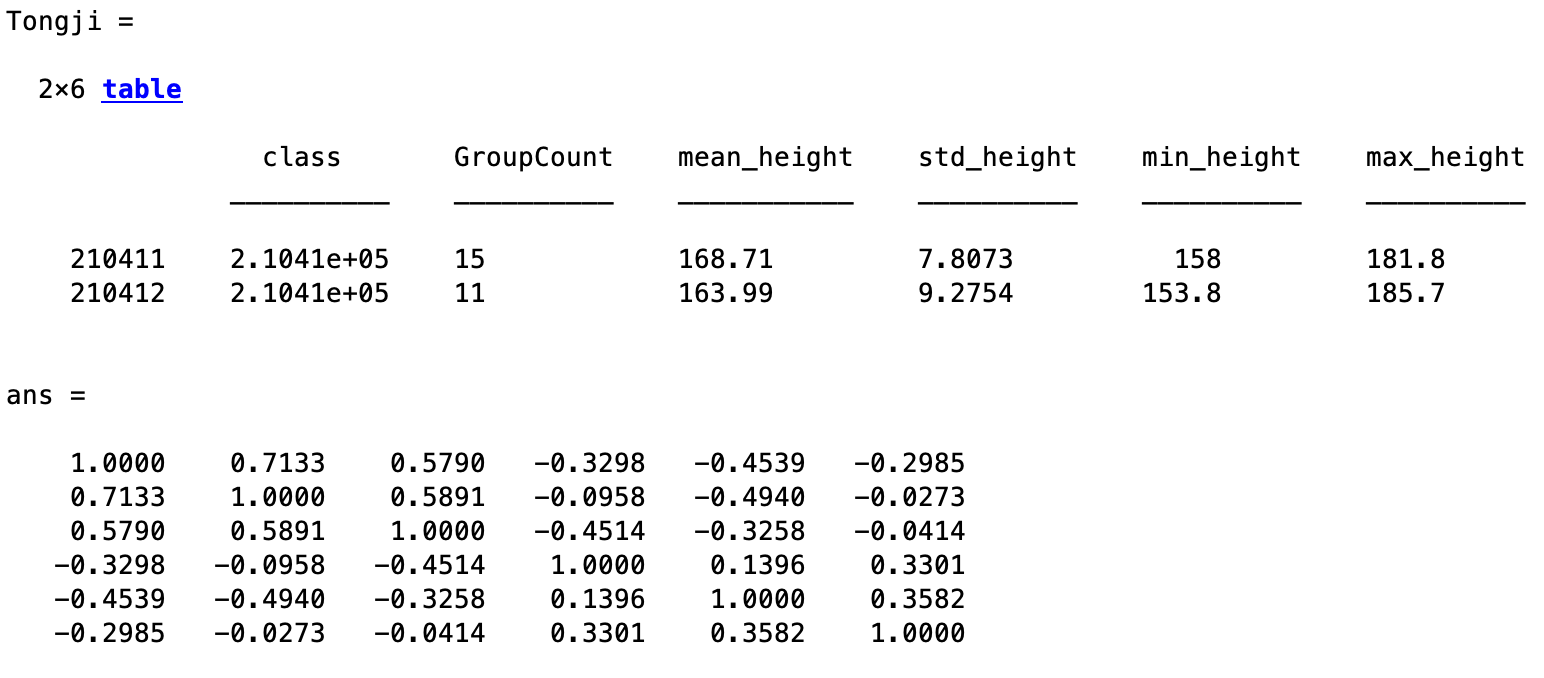
（1）练习Matlab计算样本数字特征统计量及分组统计(fenzutj.m)

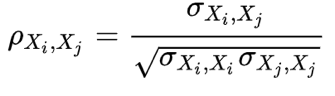
* 基本统计量函数：mean(A), std(A),max(A),min(A),其中A为向量或矩阵，是矩阵时对每一列求值 。分组统计函数grpstats(X, ‘group’,whichstats),其中X为样本数据，group为分组统计的组名，whichstats为需要计算的统计量。
* 运用readtable读取已有excel文件（体测成绩.xls）中的数据，创建表格型数据组。 readtable(‘filename’)。
* whichstats函数指定需要计算的统计量名称。whichstats={‘’, ‘’…}
* 提取身高数据，分组统计均值，标准差，最小值和最大值。T1=T(:,{‘class’,‘height’})

Tongji=grpstats(T1,’class’,whichstats);

* 提取身高，体重，肺活量数据，以及耐力项目，力量项目，速度项目分数等变量数据，运用corrcoef()计算他们之间的相关系数矩阵(\*table2array数据转换)。

1. clc;close all;clear;
2. A=readtable('体测成绩.xls');
3. T=A(:,{'class','height'});
4. whichstats={'mean','std','min','max'};  %统计量
5. Tongji=grpstats(T,'class',whichstats)
6. T1=A(:,{'height','weight','VC','score1','score2','score3'});
7. T1=table2array(T1); %将excel表格转化为数组
8. corrcoef(T1)    %相关系数矩阵，统计两个随机变量之间线性相关程度

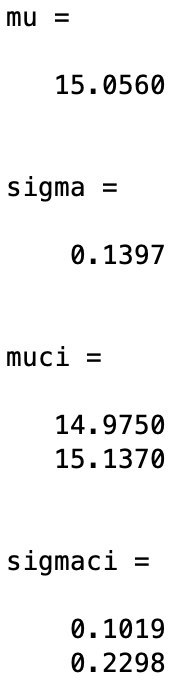


相关系数：

（2）常见分布参数估计（normcanshu.m）

* 从某厂生产的滚珠中随机抽取10个，定义样本观测值向量：[15.14,14.81,15.11,15.26,15.08,15.17,15.12,14.95,15.05,14.87],假定滚珠直径付出正态分布。
* 用normfit（x,0.1）函数进行均值和标准差的估计。

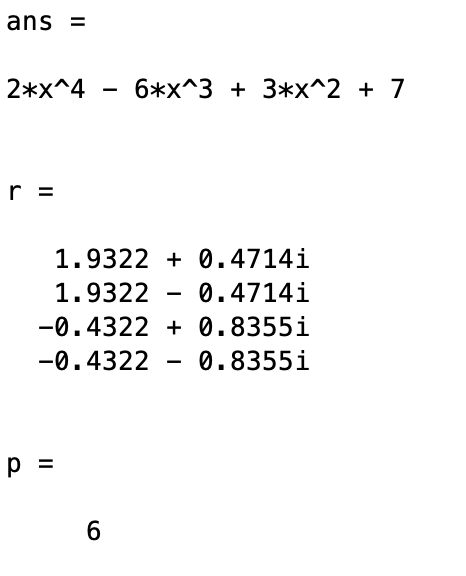
1. clc;close all;clear;
2. x=[15.14,14.81,15.11,15.26,15.08,15.17,15.12,14.95,15.05,14.87];
3. [mu,sigma,muci,sigmaci]=normfit(x,0.1)



（3）多项式求根(polyroots.m)

* 根据系数向量a=[2 -6 3 0 7]用poly2sym(a)构造多项式，用roots(a)计算多项式的根，用polyval(a,x)计算多项式在x=1处的值。

1. clc;close all;clear;
2. a=[2 -6 3 0 7];
3. poly2sym(a)
4. r=roots(a)
5. p=polyval(a,1)

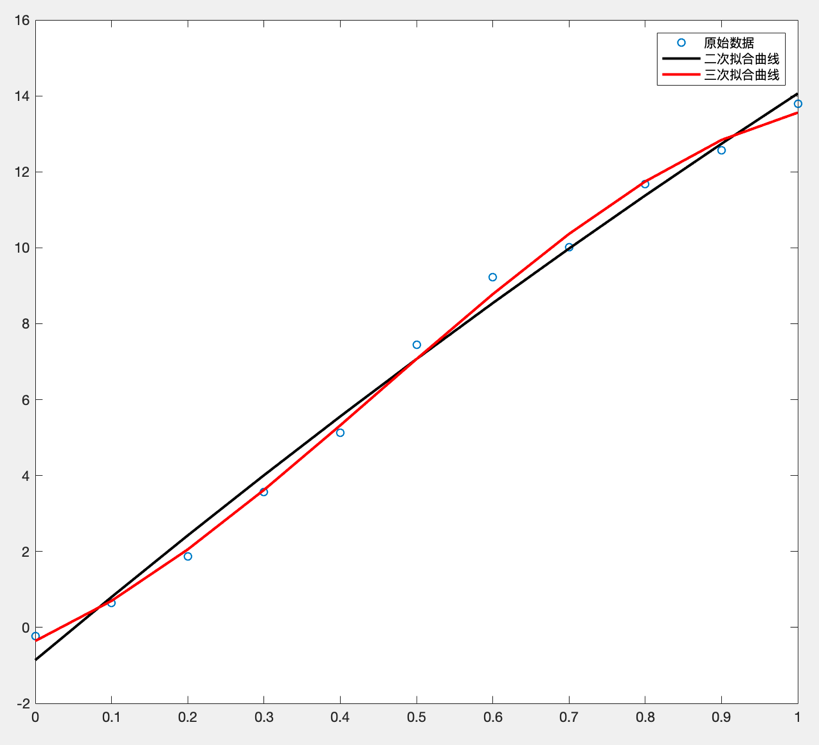


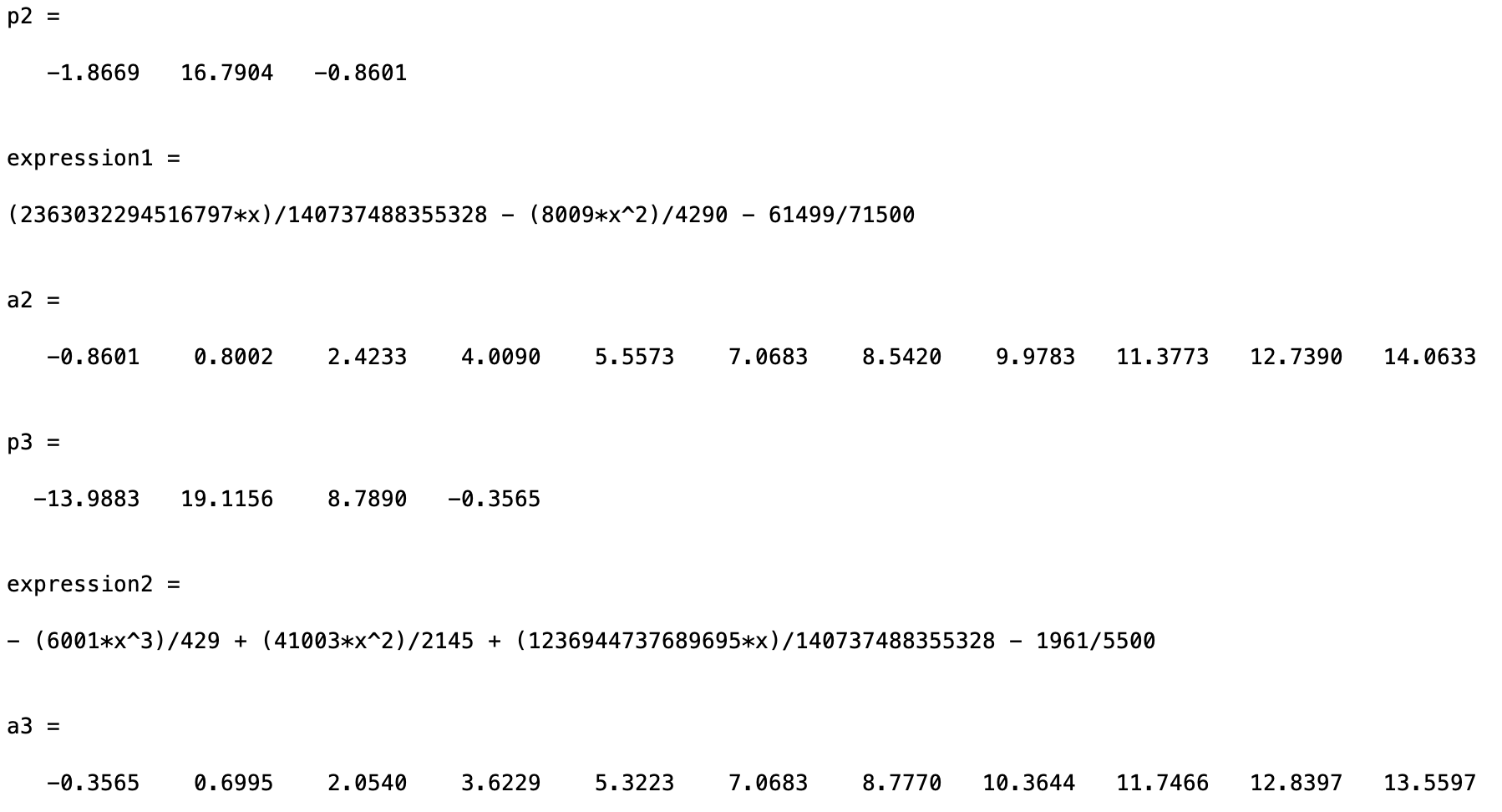
（4）多项式拟合(dxsnihe.m)

* 定义实验样本数组{}，其中=**=**[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];==[-0.232 0.647 1.877 3.565 5.134 7.443 9.221 10.011 11.678 12.566 13.788];
* 用polyfit(x,y,n)函数完成二次，三次多项式拟合。
* 绘制原始数对，和拟合函数曲线，观察拟合效果。

注：函数ployfit算法原理：求一个n次多项式使其按照最小二乘法逼近由向量x和y的对应分量构成的数对，即求使最小的函数f(x)。

1. clc;close all;clear;
2. x=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
3. y=[-0.232  0.647  1.877  3.565  5.134  7.443 9.221 10.011  11.678  12.566   13.788];
4. p2=polyfit(x,y,2)
5. expression1=poly2sym(p2)
6. a2=polyval(p2,x)
7. p3=polyfit(x,y,3)
8. expression2=poly2sym(p3)
9. a3=polyval(p3,x)
10. plot(x,y,'o')
11. hold on;
12. plot(x,a2,'k','linewidth',2)
13. hold on;
14. plot(x,a3,'r','linewidth',2)
15. legend('原始数据', '二次拟合曲线', '三次拟合曲线')





在涉及很多点的问题中，使用 polyfit 增加多项式拟合的次数并不总能得到较好的拟合。高次多项式可以在数据点之间振动，导致与数据之间的拟合较差。在这些情况下，可使用低次多项式拟合（点之间倾向于更平滑）或不同的方法，具体取决于该问题。

**四、实验结果**