数据回归分析和拟合的 Matlab 实现

本次将教程的主要内容包含：

一、多元线性回归 2#
多元线性回归：regress

二、多项式回归 3#
一元多项式：polyfit 或者 polytool
多元二项式：rstool 或者 rsmdemo

三、非线性回归 4#
非线性回归：nlinfit

四、逐步回归 5#
逐步回归：stepwise

一、多元线性回归

多元线性回归： $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$

1、b=regress(Y, X)　　确定回归系数的点估计值

①b的表达式　　$b = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \cdots \\ \hat{\beta}_p \end{bmatrix}$

②Y的表达式　　$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \cdots \\ Y_n \end{bmatrix}$

③X的表达式　　$X=\begin{bmatrix}1 & x_{11} & x_{12} & \cdots & x_{1p}\\ 1 & x_{21} & x_{22} & \cdots & x_{2p}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 1 & x_{n1} & x_{n2} & \cdots & x_{np}\end{bmatrix}$

2、[b, bint,r,rint,stats]=regress(Y,X,alpha)　　求回归系数的点估计和区间估计、并检验回归模型

①bint 表示回归系数的区间估计.

②r 表示残差

③rint 表示置信区间

④stats 表示用于检验回归模型的统计量,有三个数值：相关系数 r^2、F 值、与 F 对应的概率 p

说明：相关系数 r^2 越接近 1，说明回归方程越显著；$F > F_{1-\alpha}(k, n-k-1)$ 时拒绝 H0，F 越大，说明回归方程越显著；与 F 对应的概率 p<α 时拒绝 H0

⑤alpha 表示显著性水平(缺省时为 0.05)

3、rcoplot(r,rint)　画出残差及其置信区间具体参见下面的实例演示

4、实例演示，函数使用说明

(1)输入数据

复制内容到剪贴板

代码:

```
>>x=[143 145 146 147 149 150 153 154 155 156 157 158 159 160 162 164]';
>>X=[ones(16,1) x];
>>Y=[88 85 88 91 92 93 93 95 96 98 97 96 98 99 100 102]';
```

(2)回归分析及检验

复制内容到剪贴板

代码:

```
   >> [b,bint,r,rint,stats]=regress(Y,X)

b =

  -16.0730
```

```
    0.7194

bint =

  -33.7071    1.5612
    0.6047    0.8340

r =

    1.2056
   -3.2331
   -0.9524
    1.3282
    0.8895
    1.1702
   -0.9879
    0.2927
    0.5734
    1.8540
    0.1347
   -1.5847
   -0.3040
   -0.0234
   -0.4621
    0.0992

rint =

   -1.2407    3.6520
   -5.0622   -1.4040
   -3.5894    1.6845
   -1.2895    3.9459
   -1.8519    3.6309
   -1.5552    3.8955
   -3.7713    1.7955
   -2.5473    3.1328
```

```
   -2.2471    3.3939
   -0.7540    4.4621
   -2.6814    2.9508
   -4.2188    1.0494
   -3.0710    2.4630
   -2.7661    2.7193
   -3.1133    2.1892
   -2.4640    2.6624

stats =

    0.9282  180.9531    0.0000    1.7437
```

运行结果解读如下

参数回归结果为 $\hat{\beta}_0 = -16.073, \hat{\beta}_1 = 0.7194$

，对应的置信区间分别为[-33.7017,1.5612]和[0.6047,0.834]

r^2=0.9282(越接近于 1，回归效果越显著)，F=180.9531， p=0.0000，由 p<0.05，可知回归模型 y=-16.073+0.7194x 成立

(3)残差分析 作残差图

复制内容到剪贴板

代码:

```
rcoplot(r,rint)
```

二、多项式回归

一元多项式回归

1、一元多项式回归函数

$$y = a_1 x_m + a_2 x_{m-1} + \ldots + a_m x + a_{m+1}$$

(1)[p,S]=polyfit(x,y,m)　　确定多项式系数的 MATLAB 命令

说明：x=(x_1,x_2,…,x_n),y=(y_1,y_2,…,y_n)；p=(a_1,a_2,…,a_{m+1})是多项式 $y=a_1x^m+a_2x^{m-1}+…+a_mx+a_{m+1}$ 的系数；S 是一个矩阵,用来估计预测误差

(2)polytool(x,y,m)　　调用多项式回归 GUI 界面，参数意义同 polyfit

2、预测和预测误差估计

(1)Y=polyval(p,x)　　求 polyfit 所得的回归多项式在 x 处的预测值 Y

(2)[Y,DELTA]=polyconf(p,x,S,alpha)　　求 polyfit 所得的回归多项式在 x 处的预测值 Y 及预测值的显著性为 1-alpha 的置信区间 Y±DELTA，alpha 缺省时为 0.5

3、实例演示说明

观测物体降落的距离 s 与时间 t 的关系，得到数据如下表，求 s 的表达式(即回归方程 $s=a+bt+ct^2$)

t (s) 1/30 2/30 3/30 4/30 5/30 6/30 7/30

s (cm) 11.86 15.67 20.60 26.69 33.71 41.93 51.13

t (s) 8/30 9/30 10/30 11/30 12/30 13/30 14/30

s (cm) 61.49 72.90 85.44 99.08 113.77 129.54 146.48

解法一：直接作二次多项式回归

复制内容到剪贴板

代码:

```
>>t=1/30:1/30:14/30;
>>s=[11.86 15.67 20.60 26.69 33.71 41.93 51.13 61.49 72.90 85.44 99.08
113.77 129.54 146.48];
>>[p,S]=polyfit(t,s,2)

p =

  489.2946   65.8896    9.1329
```

```
S =

        R: [3x3 double]
       df: 11
    normr: 0.1157
```

故回归模型为 $\hat{s}=489.2946t^2+65.8896t+9.1329$

解法二：化为多元线性回归

复制内容到剪贴板

代码:

```
>>t=1/30:1/30:14/30;
>>s=[11.86 15.67 20.60 26.69 33.71 41.93 51.13 61.49 72.90 85.44 99.08
113.77 129.54 146.48];
>>T=[ones(14,1) t' (t.^2)'];
>>[b,bint,r,rint,stats]=regress(s',T)

b =

    9.1329
   65.8896
  489.2946


bint =

    9.0614    9.2044
   65.2316   66.5476
  488.0146  490.5747


r =
```

```
   -0.0129
   -0.0302
   -0.0148
    0.0732
    0.0040
    0.0474
   -0.0165
   -0.0078
   -0.0363
   -0.0222
    0.0046
   -0.0059
   -0.0237
    0.0411


rint =

   -0.0697    0.0439
   -0.0956    0.0352
   -0.0876    0.0580
    0.0182    0.1283
   -0.0709    0.0789
   -0.0192    0.1139
   -0.0894    0.0563
   -0.0813    0.0658
   -0.1062    0.0335
   -0.0955    0.0511
   -0.0704    0.0796
   -0.0793    0.0675
   -0.0904    0.0429
   -0.0088    0.0910

stats =

  1.0e+007 *
```

```
    0.0000    1.0378         0    0.0000
```

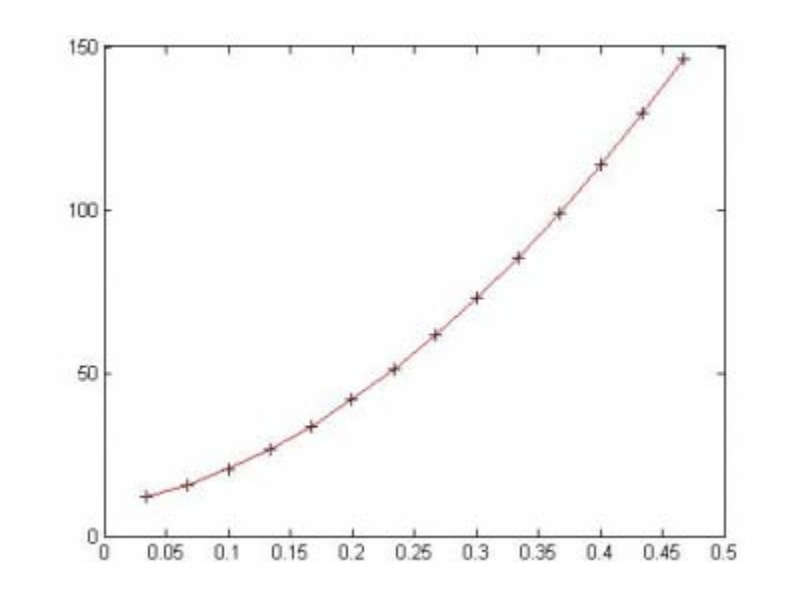
故回归模型为： $\hat{s} = 9.1329 + 65.8896t + 489.2946t^2$

预测及作图

复制内容到剪贴板

代码：

```
Y=polyconf(p,t,S);
plot(t,s,'k+',t,Y,'r')
```



多元二项式回归

1、多元二项式回归 Matlab 命令

rstool(x,y,'model',alpha)

输入参数说明：

x：n*m 矩阵；

Y：n 维列向量；

alpha：显著性水平(缺省时为 0.05)；

mode：由下列 4 个模型中选择 1 个(用字符串输入,缺省时为线性模型)

Linear(线性)：$y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m$

PureQuadratic(纯二次)：$y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m+\sum_{j=1}^{m}\beta_{jj}x_j^2$

Interaction(交叉)：$y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m+\sum_{1\le j<k\le m}\beta_{jk}x_jx_k$

Quadratic(完全二次)：$y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m+\sum_{1\le j,k\le m}\beta_{jk}x_jx_k$

2、实例演示说明

设某商品的需求量与消费者的平均收入、商品价格的统计数据如下,建立回归模型,预测平均收入为 1000、价格为 6 时的商品需求量

需求量 100 75 80 70 50 65 90 100 110 60

收入 1000 600 1200 500 300 400 1300 1100 1300 300

价格 5 7 6 6 8 7 5 4 3 9

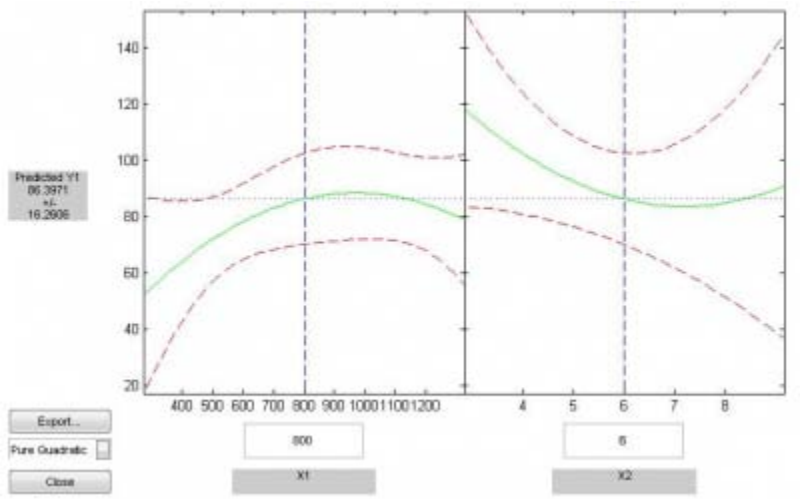
解法一：选择纯二次模型

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_{11}x_1^2+\beta_{22}x_2^2$$

复制内容到剪贴板

代码:

```
%直接用多元二项式回归如下
x1=[1000 600 1200 500 300 400 1300 1100 1300 300];
x2=[5 7 6 6 8 7 5 4 3 9];
y=[100 75 80 70 50 65 90 100 110 60]';
x=[x1' x2'];
rstool(x,y,'purequadratic')
```



在 x1 对应的文本框中输入 1000，X2 中输入 6，敲回车键，此时图形和相关数据会自动更新此时在 GUI 左边的“Predicted Y1”下方的数据变为 88.47981，表示平均收入为 1000、价格为 6 时商品需求量为 88.4791

点击左下角的 Export 按钮，将会导出回归的相关参数 beta、rmse 和 residuals 到工作空间 (workspace)

在 Export 按钮下面可以选择回归类型

在 Matlab 命令窗口中输入

复制内容到剪贴板

代码:

```
>>beta, rmse
```

将得到如下结果

复制内容到剪贴板

代码:

```
beta =
            110.5313
            0.1464
            -26.5709
            -0.0001
            1.8475
        rmse =
```

4.5362

故回归模型为 $y=110.5313+0.1464x_1-26.5709x_2-0.0001x_1^2+1.8475x_2^2$

解法二：将上面饿模型转换为多元线性回归

$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_{11}x_1^2+\beta_{22}x_2^2$

复制内容到剪贴板

代码:

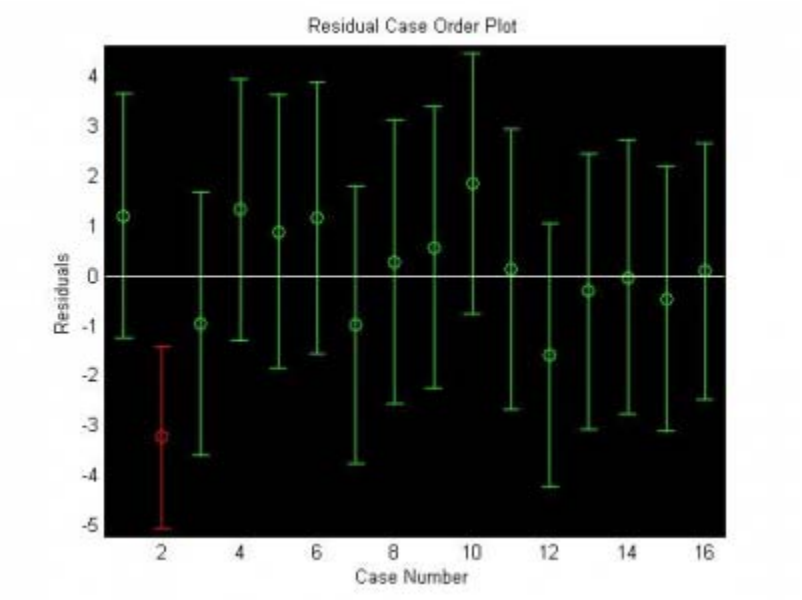
```
>>X=[ones(10,1) x1' x2' (x1.^2)' (x2.^2)'];
>>[b,bint,r,rint,stats]=regress(y,X);
>>b,stats

b =

  110.5313
    0.1464
  -26.5709
   -0.0001
    1.8475


stats =

    0.9702   40.6656    0.0005   20.5771
```



从残差图可以看出，除第二个数据外,其余数据的残差离零点均较近，且残差的置信区间均包含零点，这说明回归模型 y=-16.073+0.7194x 能较好的符合原始数据，而第二个数据可视为异常点。

(4)预测及作图

复制内容到剪贴板

代码:

```
z=b(1)+b(2)*x
plot(x,Y,'k+',x,z,'r')
```

三、非线性回归

1、非线性回归

[beta,r,J]=nlinfit(x,y,'modelfun', beta0)　　非线性回归系数的命令

nlintool(x,y,'modelfun', beta0,alpha)　　非线性回归 GUI 界面

参数说明

beta：估计出的回归系数；

r：残差；

J：Jacobian 矩阵；

x,y：输入数据 x、y 分别为矩阵和 n 维列向量，对一元非线性回归,x 为 n 维列向量；

modelfun：M 函数、匿名函数或 inline 函数，定义的非线性回归函数；

beta0：回归系数的初值；

2、预测和预测误差估计

[Y,DELTA]=nlpredci('modelfun', x,beta,r,J)

获取 x 处的预测值 Y 及预测值的显著性为 1-alpha 的置信区间 Y±DELTA

3、实例演示说明

解：(1)对将要拟合的非线性模型，建立 M 函数如下

复制内容到剪贴板

代码:

```
function yhat=modelfun(beta,x)
%beta 是需要回归的参数
%x 是提供的数据
yhat=beta(1)*exp(beta(2)./x);
```

(2)输入数据

复制内容到剪贴板

代码:

```
x=2:16;
y=[6.42 8.20 9.58 9.5 9.7 10 9.93 9.99 10.49 10.59 10.60 10.80 10.60
10.90 10.76];
beta0=[8 2]';
```

(3)求回归系数

复制内容到剪贴板

代码:

```
[beta,r ,J]=nlinfit(x',y',@modelfun,beta0);
beta
```

```
beta =
              11.6036
              -1.0641
```

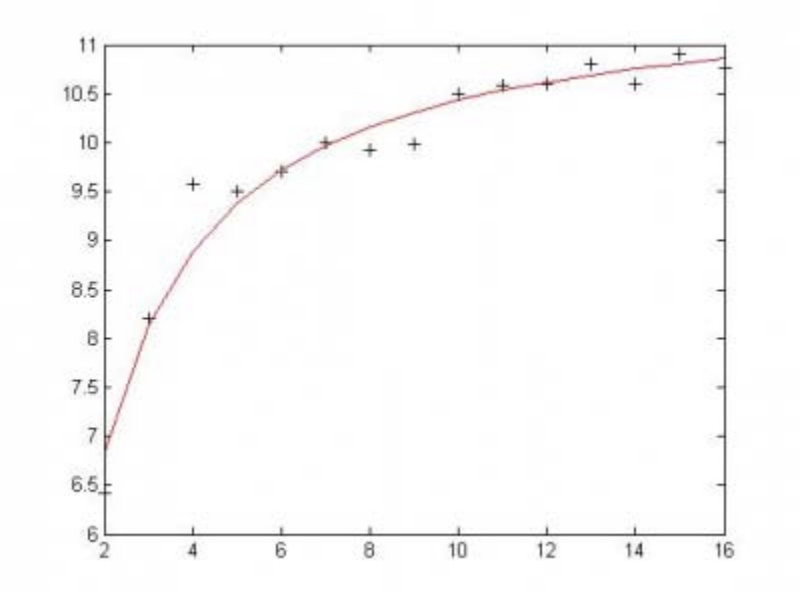
即得回归模型为 $y = 11.6036e^{\frac{-1.0641}{x}}$

(4)预测及作图

复制内容到剪贴板

代码:

```
[YY,delta]=nlpredci('modelfun',x',beta,r ,J);
plot(x,y,'k+',x,YY,'r')
```



四、逐步回归

1、逐步回归的命令

stepwise(x,y,inmodel,alpha) 根据数据进行分步回归

stepwise 直接调出分步回归 GUI 界面

输入参数说明

x：自变量数据， 阶矩阵；

y：因变量数据，阶矩阵；

inmodel：矩阵的列数的指标,给出初始模型中包括的子集(缺省时设定为全部自变量)；

alpha：显著性水平(缺省时为 0.5)；

2、实例演示分析

水泥凝固时放出的热量 y 与水泥中 4 种化学成分 x1、x2、x3、 x4 有关，今测得一组数据如下，试用逐步回归法确定一个线性模型

序号	1	2	3	4	5	6	7	8	9	10	11	12	13
x1	7	1	11	11	7	11	3	1	2	21	1	11	10
x2	26	29	56	31	52	55	71	31	54	47	40	66	68
x3	6	15	8	8	6	9	17	22	18	4	23	9	8
x4	60	52	20	47	33	22	6	44	22	26	34	12	12
y	78.5	74.3	104.3	87.6	95.9	109.2	102.7	72.5	93.1	115.9	83.8	113.3	109.4

(1)数据输入

复制内容到剪贴板

代码:

```
x1=[7 1 11 11 7 11 3 1 2 21 1 11 10]';
x2=[26 29 56 31 52 55 71 31 54 47 40 66 68]';
```

```
x3=[6 15 8 8 6 9 17 22 18 4 23 9 8]';
x4=[60 52 20 47 33 22 6 44 22 26 34 12 12]';
y=[78.5 74.3 104.3 87.6 95.9 109.2 102.7 72.5 93.1 115.9 83.8 113.3
109.4]';
x=[x1 x2 x3 x4];
```

(2)逐步回归

①先在初始模型中取全部自变量

复制内容到剪贴板

代码:

```
stepwise(x,y)
```