

MATLAB统计分析与应用

——回归分析

主讲人：谢中华

ssstudy.com

科学软件学习网

主要内容

- 一元线性回归
- 一元非线性回归
- 多元线性和广义线性回归
- 多元非线性回归
- 曲线（面）拟合工具

第一节 一元线性回归

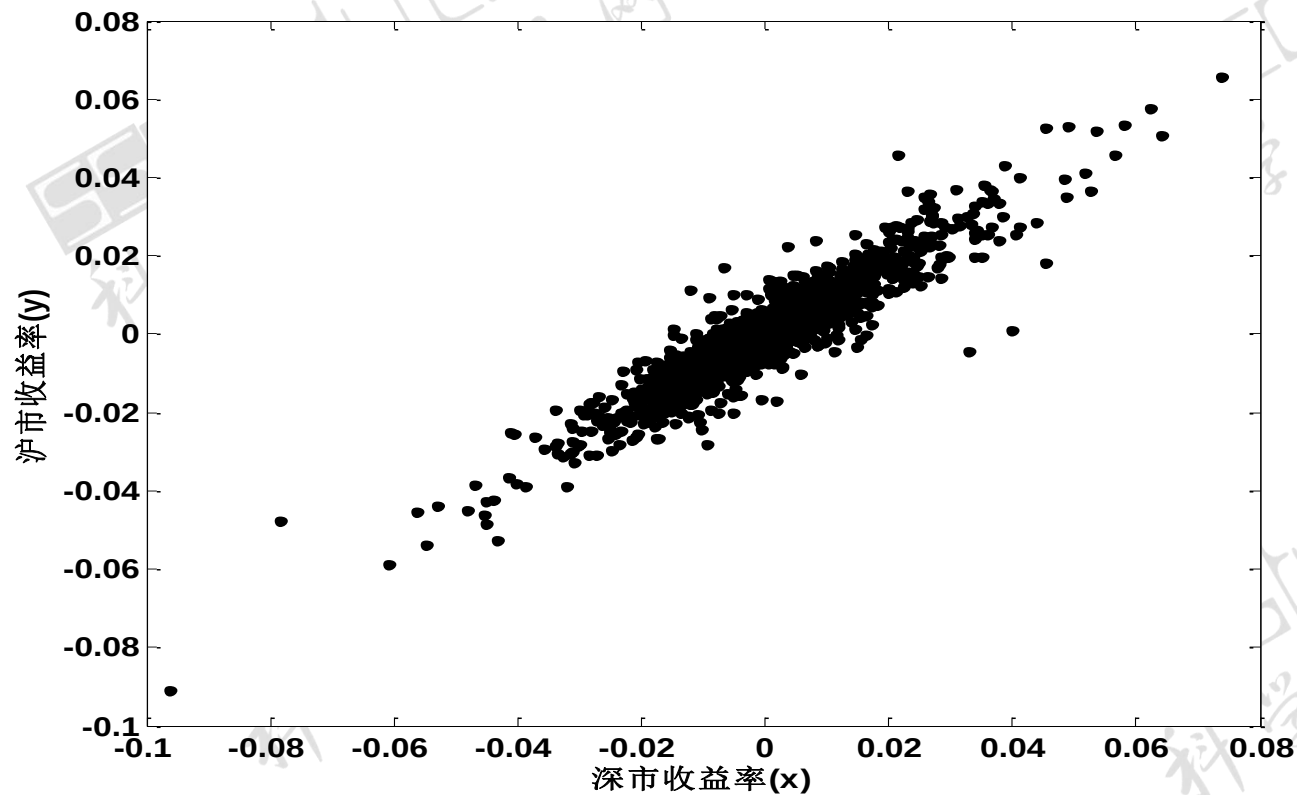
【例12.1-1】现有上海和深圳股市同时期日开盘价、最高价、最低价、收盘价、收益率等数据，跨度为2000年1月至2007年4月，各1696组数据。据此研究沪市收益率 y 和深市收益率 x 的关系。

一、数据的散点图

先从 x 和 y 的散点图上直观的观察它们之间的关系，然后再作进一步的分析。

```
>> x = xlsread('shenshi.xls');  
>> x = x(:, 5);  
>> y = xlsread('hushi.xls');  
>> y = y(:, 5);  
>> plot(x, y, 'k.', 'Markersize', 15);  
>> xlabel('深市收益率(x)');  
>> ylabel('沪市收益率(y)');  
%计算x和y的线性相关系数矩阵R  
>> R = corrcoef(x, y)
```

$$\rho_{xy} = 0.9314$$



二、模型建立

一元线性回归模型：

$$\begin{cases} y = a + bx + \varepsilon \\ \varepsilon \sim N(0, \sigma^2) \end{cases}$$

三、MATLAB线性回归模型类

MATLAB R2012a (即MATLAB7.14)

中对回归分析的实现方法作了重大调整，给出了三种回归模型类：

- LinearModel class：线性回归模型类
- NonLinearModel class：非线性回归模型类
- GeneralizedLinearModel class：广义线性回归模型类

1. 线性回归模型类的类构造函数

➤ 调用 LinearModel 函数

```
>> mdl = LinearModel
```

➤ 调用 fit 方法

```
>> mdl = LinearModel.fit(X,y,modelspec,Name,Value)
```

```
>> mdl = LinearModel.fit(ds,modelspec,Name,Value)
```

2. 线性回归模型类的类方法

方法名	功能说明
addTerms	在线性回归模型中增加项
anova	对线性模型做方差分析
coefCI	系数估计值的置信区间
coefTest	对回归系数进行检验
disp	显示线性回归分析的结果
dwtest	对线性模型进行Durbin-Watson检验
feval	利用线性回归模型进行预测
fit	利用观测数据对线性回归模型进行拟合
plot	绘制模型拟合效果图
plotAdded	绘制指定项的拟合效果图

2. 线性回归模型类的类方法（续）

方法名	功能说明
<code>plotAdjustedResponse</code>	绘制调整后的响应曲线
<code>plotDiagnostics</code>	绘制模型诊断图
<code>plotEffects</code>	绘制线性回归模型中每个自变量的主效应图
<code>plotInteraction</code>	绘制线性回归模型中两个自变量的交互效应图
<code>plotResiduals</code>	绘制线性回归模型的残差图
<code>plotSlice</code>	绘制通过回归面的切片图
<code>predict</code>	利用线性回归模型进行预测
<code>random</code>	利用线性回归模型模拟响应值（因变量值）
<code>removeTerms</code>	从线性回归模型中移除项
<code>step</code>	通过增加或移除项来改进线性回归模型
<code>stepwise</code>	利用逐步回归方法建立线性回归模型

四、程序实现

1. 创建数据集

```
>> ds = dataset([x,y], 'x', 'y');
```

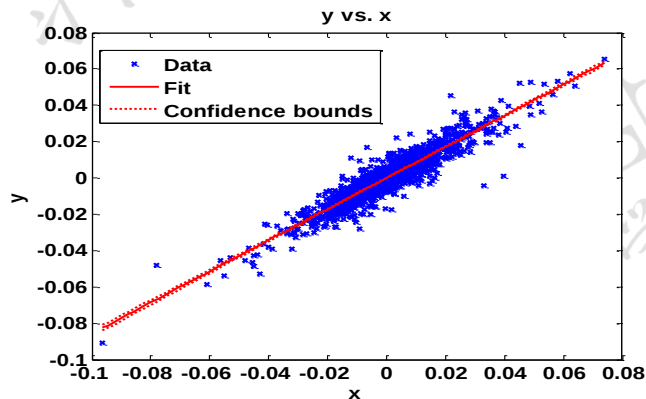
2. 数据拟合

```
>> mdl = LinearModel.fit(ds)
```

%或 `mdl = LinearModel.fit(x,y)`

3. 拟合效果图

```
>> h = mdl.plot;
```



4. 拟合结果

Linear regression model:

$$y \sim 1 + x$$

$$\hat{y} = 5.8734 \times 10^{-8} + 0.85687x$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue	
(Intercept)	5.8734e-08	0.0001175	0.00049987	0.9996	
x	0.85687	0.0081376	105.3	0	

Number of observations: 1696, Error degrees of freedom: 1694

Root Mean Squared Error: 0.00484

R-squared: 0.867, Adjusted R-Squared 0.867

F-statistic vs. constant model: 1.11e+04, p-value = 0

五、模型诊断

1. 查看方差分析表

```
>> tbl = mdl.anova
```

```
>> tb2 = mdl.anova('summary')
```

	SumSq	DF	MeanSq	F	pValue
Total	0.29906	1695	0.00017644		
Model	0.25943	1	0.25943	11088	0
Residual	0.039637	1694	2.3398e-05		

2. 残差分析

```
>> h = mdl.plotResiduals(plottype,Name,Value)
```

plottype参数值说明

'caseorder'	残差序列图
'fitted'	残差与拟合值关系图
'histogram'	残差直方图（默认）
'lagged'	残差与滞后残差关系图
'probability'	正态概率图
'symmetry'	对称图

3. 绘制诊断图（查找强影响点）

```
>> h = mdl.plotDiagnostics(plottype,Name,Value)
```

plottype参数值说明	
'contour'	残差与杠杆值关系图
'cookd'	Cook距离图
'covratio'	去掉1个观测点前后的协方差矩阵的行列式比值图
'dfbetas'	去掉1个观测点前后的系数估计值的差值图
'dffits'	去掉1个观测点前后的拟合值的差值图
'leverage'	杠杆值图（默认）
's2_i'	去掉1个观测点后的均方误差图

4. 查找异常点

利用标准化残差或学生化残差查找异常点

```
>> Res = mdl.Residuals;           % 查询残差值
>> Res_Stu = Res.Studentized;     % 学生化残差
>> Res_Stan = Res.Standardized;   % 标准残差
>> id = find(abs(Res_Stu)>2);      % 查找异常值序号
```

5. 模型改进

去除异常点和不显著项后重新拟合，可对模型做出改进。

```
>> figure;  
>> mdl.plotResiduals('probability');  
>> figure;  
>> mdl.plotDiagnostics('dffits');  
>> mdl2 = LinearModel.fit(ds, 'Exclude', id)  
>> mdl3 = mdl2.removeTerms('1')  
>> figure;  
>> mdl3.plot;
```

模型改进结果

Linear regression model:

$$y \sim x$$

$$\hat{y} = 0.86093x$$

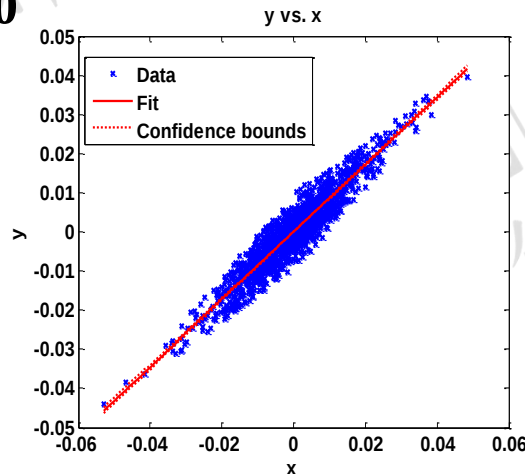
Estimated Coefficients:

	Estimate	SE	tStat	pValue
x	0.86093	0.0066694	129.09	0

Number of observations: 1609,

Error degrees of freedom: 1608

Root Mean Squared Error: 0.00374



六、利用回归模型进行预测

1. 预测

```
>> xnew = linspace(min(x),max(x),30)';  
>> ynew = mdl3.predict(xnew);
```

七、稳健拟合

```
>> mdl4 = LinearModel.fit(ds,'RobustOpts','on')
```

Linear regression model (robust fit):

$$y \sim 1 + x$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	2.5352e-05	0.00010449	0.24263	0.80832
x	0.86759	0.0072366	119.89	0

Number of observations: 1696, Error degrees of freedom: 1694

Root Mean Squared Error: 0.0043

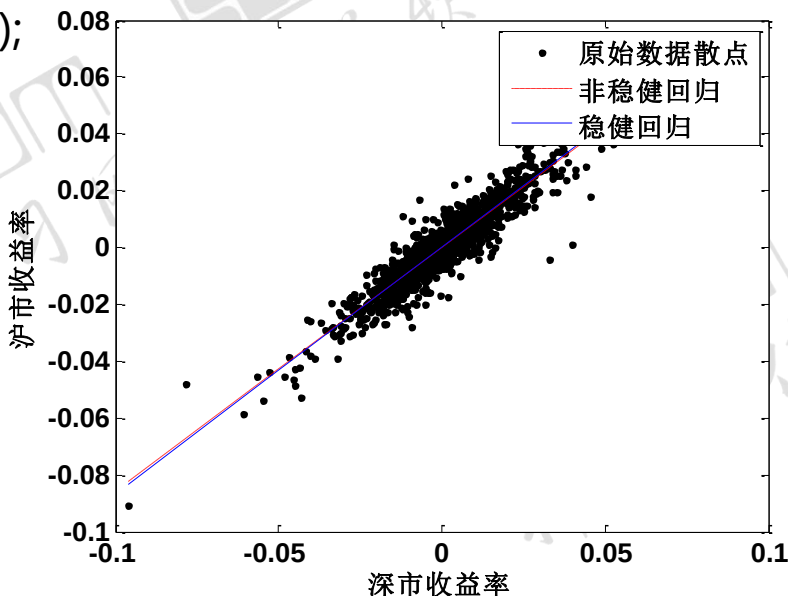
R-squared: 0.895, Adjusted R-Squared 0.895

F-statistic vs. constant model: 1.44e+04, p-value = 0

➤ 非稳健回归和稳健回归的结果对比

```
>> plot(x, y, 'k.')
>> hold on
>> xnew = linspace(min(x),max(x),5)';
>> yhat1 = mdl.predict(xnew);
>> yhat2 = mdl4.predict(xnew);
>> plot(xnew, yhat1, 'r--')
>> plot(xnew, yhat2)
>> legend('原始数据散点',...
    '非稳健回归',...
    '稳健回归');
```

```
>> xlabel('深市收益率')
>> ylabel('沪市收益率')
```



第二节 一元非线性回归

【例12.2-1】头围(head circumference)是反映婴幼儿大脑和颅骨发育程度的重要指标之一，对头围的研究具有非常重要的意义。笔者研究了天津地区1281位儿童（700个男孩，581个女孩）的颅脑发育情况，测量了年龄、头宽、头长、头宽/头长、头围和颅围等指标，测得1281组数据，年龄跨度从7个星期到16周岁，试根据这1281组数据建立头围关于年龄的回归方程。

一、数据的散点图及备选方程

1. 散点图

令 x 表示年龄， y 表示头围。由于 x 和 y 均为一维变量，可以先从 x 和 y 的散点图上直观的观察它们之间的关系，然后再作进一步的分析。

```
>> HeadData = xlsread('examp12_2_1.xls');
```

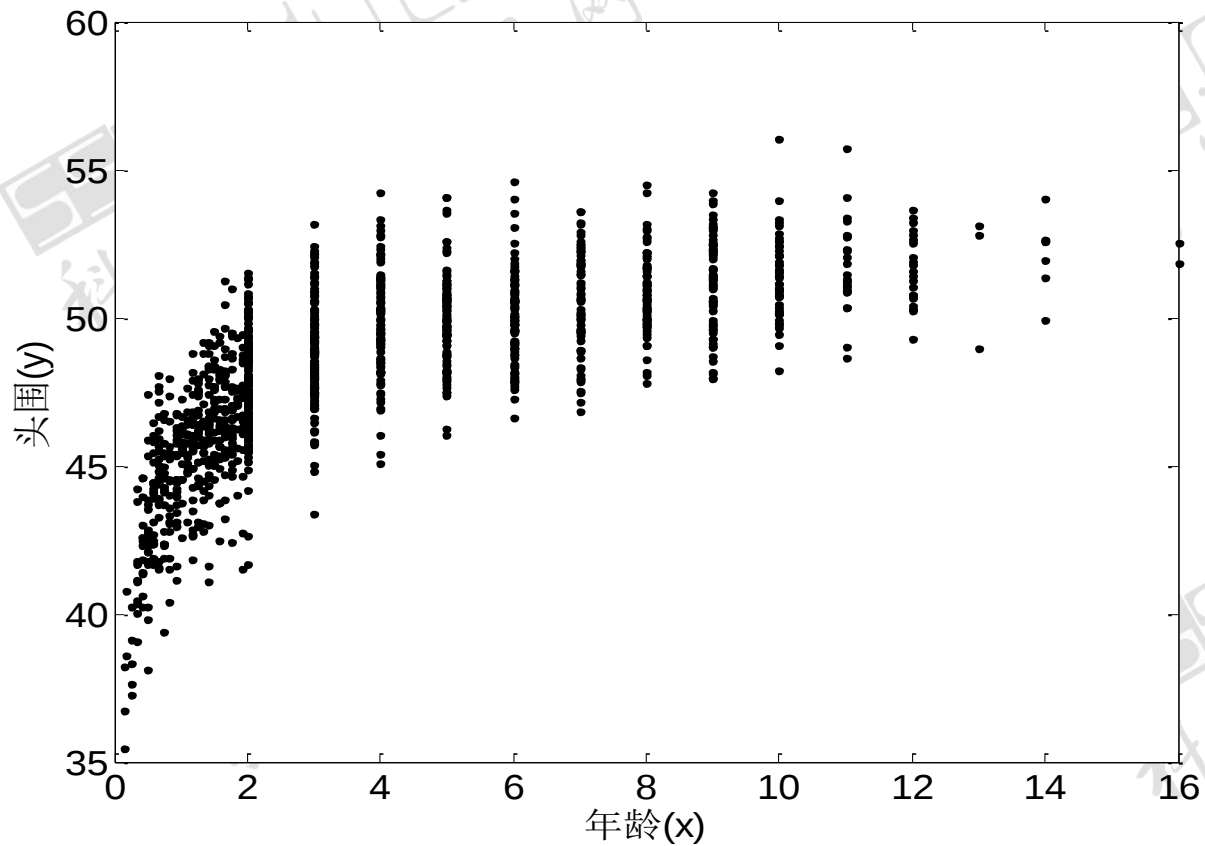
```
>> x = HeadData(:, 4);
```

```
>> y = HeadData(:, 9);
```

```
>> plot(x, y, 'k.')
```

```
>> xlabel('年龄(x)')
```

```
>> ylabel('头围(y)')
```



2. 备选方程

- 负指数函数： $y = \beta_1 e^{\frac{\beta_2}{x + \beta_3}}$
- 双曲线函数： $y = \frac{x + \beta_1}{\beta_2 x + \beta_3}$
- 幂函数： $y = \beta_1 (x + \beta_2)^{\beta_3}$
- Logistic曲线函数： $y = \frac{\beta_1}{1 + \beta_2 e^{-(x + \beta_3)}}$
- 对数函数： $y = \beta_1 + \beta_2 \ln(x + \beta_3)$

二、MATLAB非线性回归模型类

1. 非线性回归模型类的类构造函数

➤ 调用Non LinearModel 函数

```
>> nlm = NonLinearModel
```

➤ 调用 fit方法

```
>> nlm = NonLinearModel.fit(X,y,modelfun,beta0,Name,Value)
```

```
>> nlm = NonLinearModel.fit(ds,modelfun,beta0,Name,Value)
```

2. 非线性回归模型类的类方法

方法名	功能说明
coefCI	系数估计值的置信区间
coefTest	对回归系数进行检验
disp	显示非线性回归分析的结果
feval	利用非线性回归模型进行预测
fit	利用观测数据对非线性回归模型进行拟合
plotDiagnostics	绘制模型诊断图
plotResiduals	绘制非线性回归模型的残差图
plotSlice	绘制通过回归面的切片图
predict	利用非线性回归模型进行预测
random	利用非线性回归模型模拟响应值（因变量值）

三、程序实现

1. 创建数据集

```
>> ds = dataset(x,y)
```

2. 编写回归方程对应的匿名函数或函数文件

➤ 负指数函数： $y = \beta_1 e^{\frac{\beta_2}{x + \beta_3}}$

```
>> modelfun = @(beta, x)beta(1)*exp(beta(2)./(x+beta(3)));
```

3. 数据拟合

```
>> beta0 = [53,-0.2604,0.6276];
```

```
>> nlm = NonLinearModel.fit(ds,modelfun,beta0)
```

4. 拟合结果

Nonlinear regression model:

$$y \sim \text{beta1} * \exp(\text{beta2} / (x + \text{beta3})) \quad \hat{y} = 52.433e^{-\frac{0.2676}{x+0.7907}}$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
beta1	52.433	0.14685	357.04	0
beta2	-0.26758	0.016715	-16.008	1.0417e-52
beta3	0.79072	0.074909	10.556	4.9551e-25

Number of observations: 1281, Error degrees of freedom: 1278

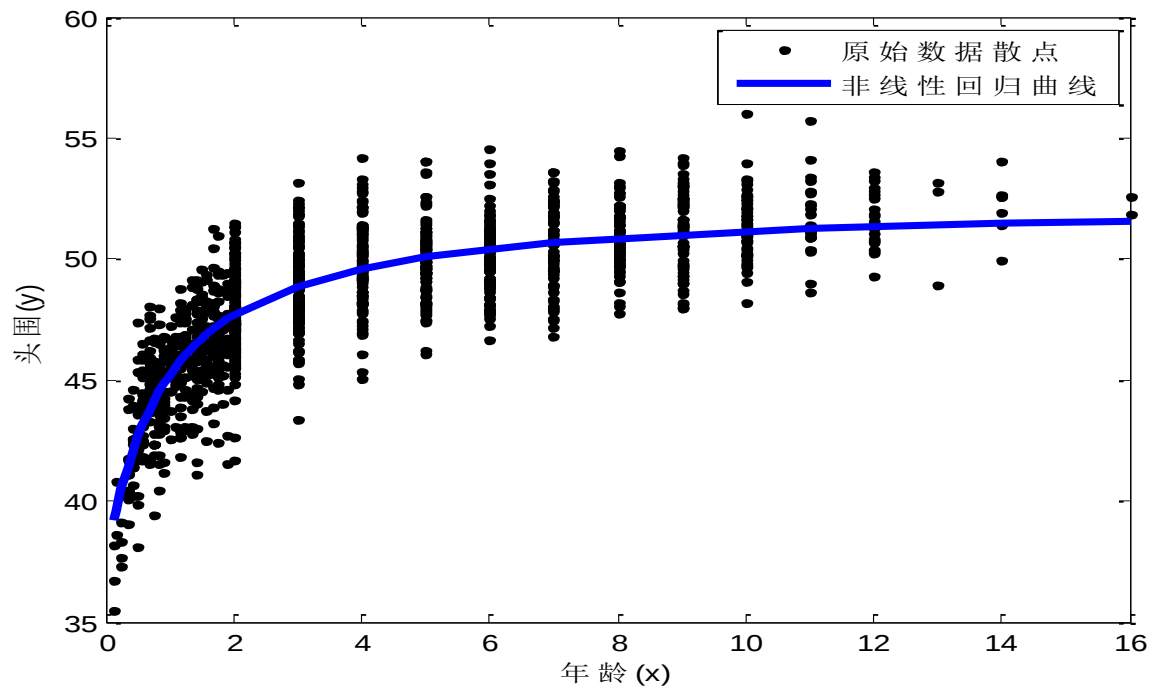
Root Mean Squared Error: 1.65

R-Squared: 0.705, Adjusted R-Squared 0.704

F-statistic vs. zero model: 3.67e+05, p-value = 0

5. 绘制一元非线性回归曲线

```
>> xnew = linspace(0,16,50)';  
>> ynew = nlm.predict(xnew);  
>> figure  
>> plot(x, y, 'k.')  
>> hold on  
>> plot(xnew, ynew, 'linewidth', 3)  
>> xlabel('年龄(x)')  
>> ylabel('头围(y)')  
>> legend('原始数据散点',...  
          '非线性回归曲线')
```

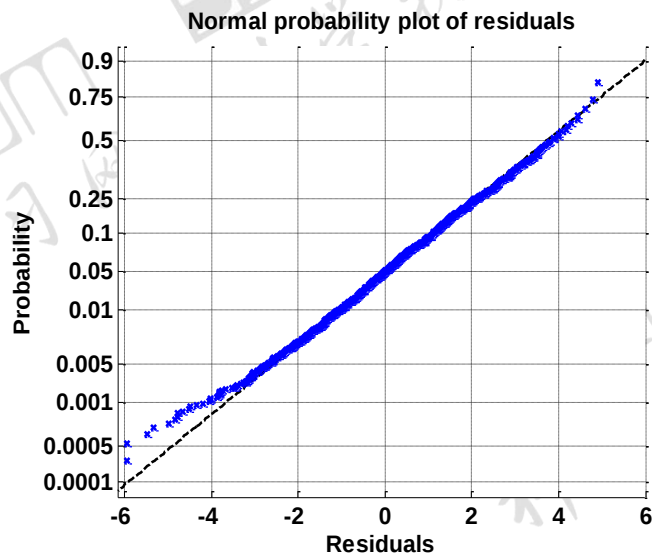
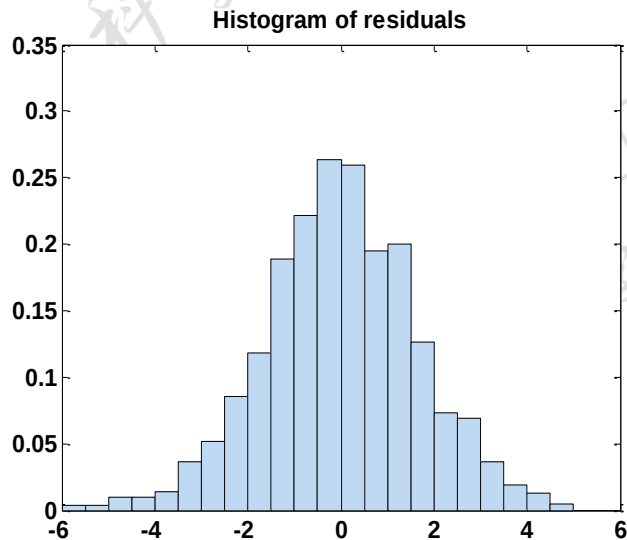


四、模型诊断

1. 绘制残差的直方图和正态概率图

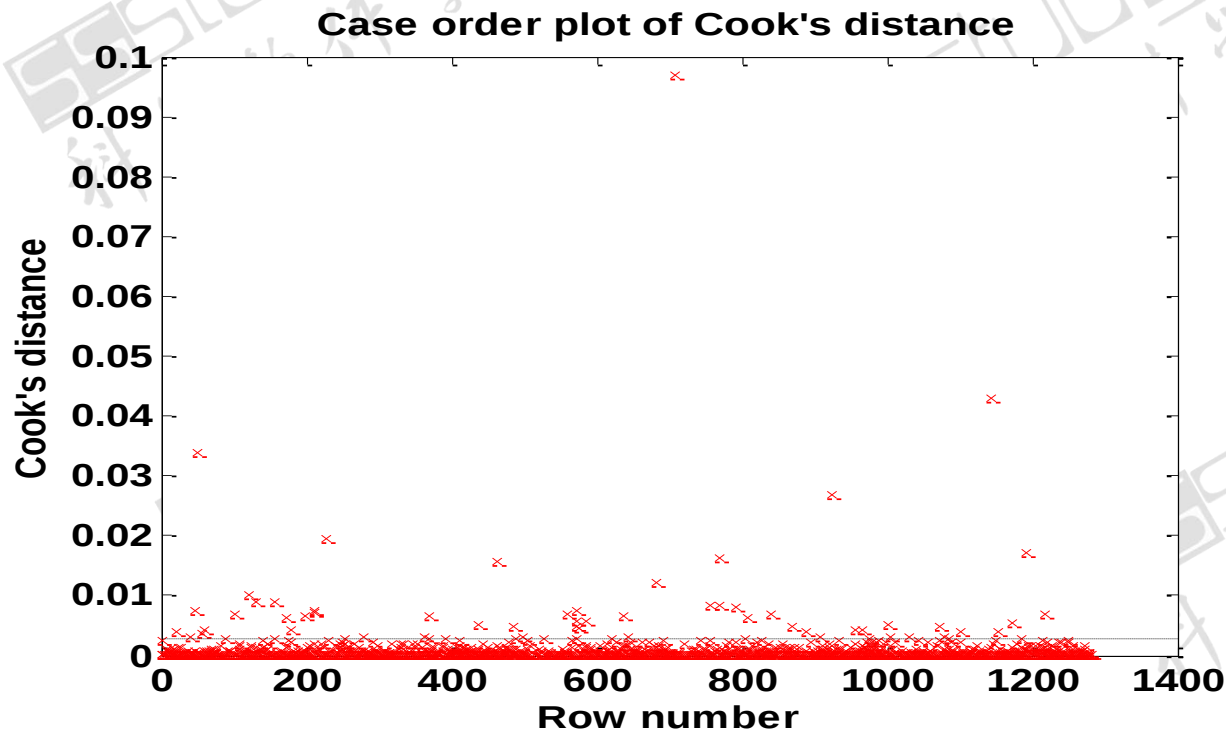
```
>> h = nlm.plotResiduals;
```

```
>> h = nlm.plotResiduals('probability');
```



2. 绘制Cook距离图

```
>> h = nlm.plotDiagnostics('cookd');
```



3. 根据学生化残差查找异常数据

```
>> id = find(abs(nlm.Residuals.Studentized) > 2);
```

4. 模型改进

```
>> nlm2 = NonLinearModel.fit(ds,modelfun,beta0,'Exclude',id)
```

$$\hat{y} = 52.345e^{-\frac{0.259}{x+0.7686}}$$

五、头围平均值的置信区间和观测值的预测区间

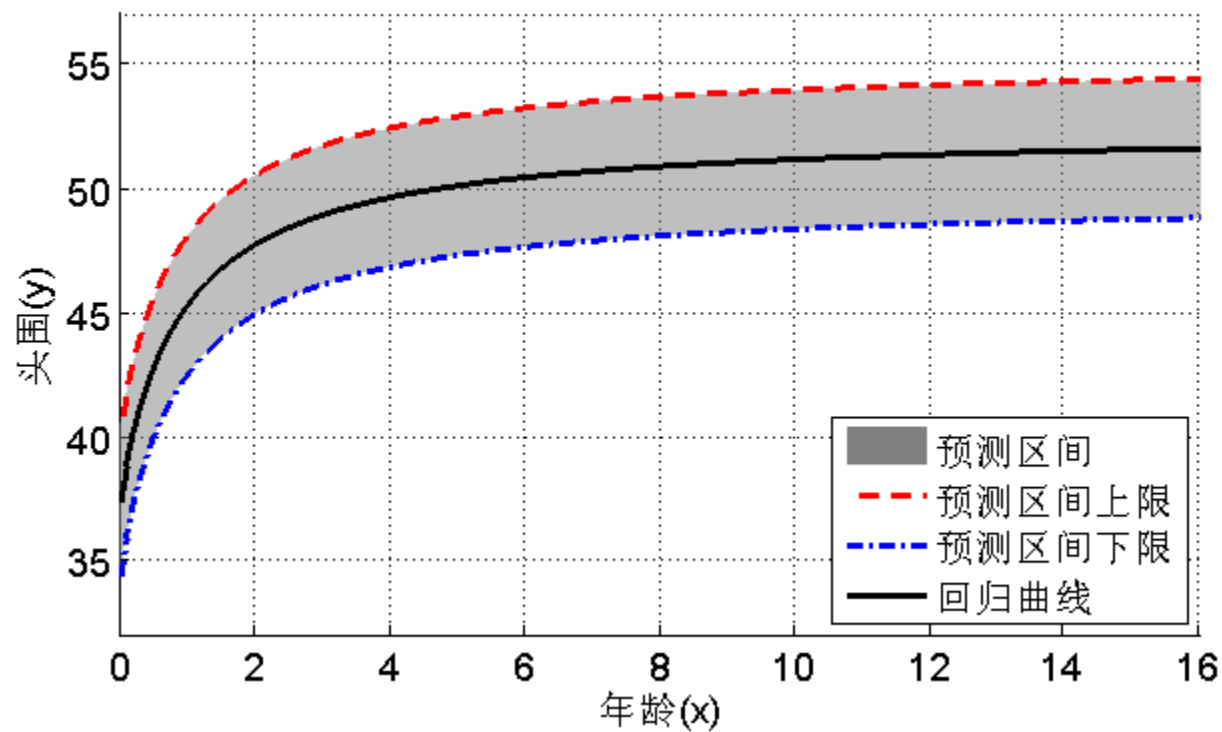
对于 x (年龄) 的一个给定值 x_0 , 相应的 y (头围) 是一个随机变量 , 具有一定的分布。 x 给定时 y 的总体均值的区间估计称为平均值 (或预测值) 的置信区间 , y 的观测值的区间估计称为观测值的预测区间。

求出头围关于年龄的回归曲线后，对于给定的年龄，可以调用`predict`函数求出头围的预测值（即 x 给定时 y 的总体均值）、预测值的置信区间和观测值的预测区间。

- `[yp,ypci] = nlm.predict(xnew,'Prediction','curve');`
- `[yp,ypci] = nlm.predict(xnew,'Prediction','observation');`

➤ 绘制预测区间图

```
>> xnew = [0:0.1:16]';  
>> [yp,ypci] = nlm2.predict(xnew,'Prediction','observation');  
>> figure;  
>> hold on;  
>> h1 = fill([xnew;flipud(xnew)],[ypci(:,2);flipud(ypci(:,1))],[0.5,0.5,0.5]);  
>> set(h1,'EdgeColor','none','FaceAlpha',0.5);  
>> plot(xnew,ypci(:,2),'r--','LineWidth',2);  
>> plot(xnew,ypci(:,1),'b-.','LineWidth',2);  
>> plot(xnew, yp, 'k','linewidth', 2)  
>> grid on  
>> ylim([32, 57])  
>> xlabel('年龄(x)')  
>> ylabel('头围(y)')  
>> h2 = legend('预测区间','预测区间上限','预测区间下限','回归曲线');  
>> set(h2, 'Location', 'SouthEast')
```



第三节 多元线性和广义线性回归

【例12.3-1】在有氧锻炼中，人的耗氧能力 y 是衡量身体状况的重要指标，它可能与以下因素有关：年龄 x_1 （岁），体重 x_2 （kg），1500米跑所用的时间 x_3 （min），静止时心速 x_4 （次/min），跑步后心速 x_5 （次/min）。对24名40至57岁的志愿者进行了测试，测试数据保存在文件examp12_3_1.xls中，试根据这些数据建立耗氧能力 y 与诸因素之间的回归模型。

一、相关性分析

1. 创建数据集

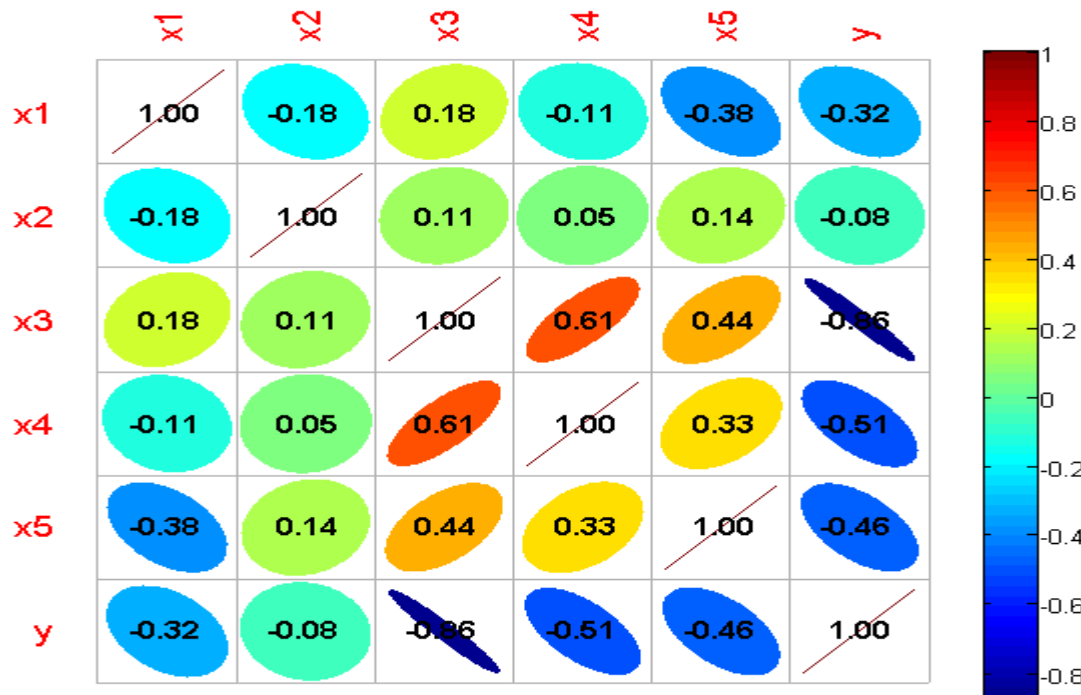
```
>> VarNames = {'x1','x2','x3','x4','x5', 'y'};  
>> ds = dataset('XLSFile','examp12_3_1.xls',...  
                'Range','B1:G25','VarNames', VarNames);
```

2. 调用corrcoef函数计算相关系数矩阵

```
>> R = corrcoef(double(ds));
```

3. 相关系数矩阵的可视化

```
>> matrixplot(R,'FigShap','e','FigSize','Auto', ...  
    'ColorBar','on','XVar', VarNames,'YVar',VarNames);
```



二、创建线性回归模型类作多元线性回归

1. 理论回归方程

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5$$

2. 数据拟合

```
>> mdl = LinearModel.fit(ds)
```

```
% 或 mdl = LinearModel.fit(ds,'y ~ 1+x1+x2+x3+x4+x5')
```

$$\hat{y} = 121.1655 - 0.3471x_1 - 0.0167x_2 - 4.2903x_3 - 0.0399x_4 - 0.1587x_5$$

3. 拟合结果

Linear regression model:

$$y \sim 1 + x1 + x2 + x3 + x4 + x5$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	121.17	17.406	6.961	1.6743e-06
x1	-0.34712	0.14353	-2.4185	0.026406
x2	-0.016719	0.087353	-0.19139	0.85036
x3	-4.2903	1.0268	-4.1784	0.00056473
x4	-0.039917	0.094237	-0.42357	0.67689
x5	-0.15866	0.078847	-2.0122	0.059407

Number of observations: 24, Error degrees of freedom: 18

Root Mean Squared Error: 2.8

R-squared: 0.816, Adjusted R-Squared 0.765

F-statistic vs. constant model: 16, p-value = 4.46e-06

三、模型诊断

1. 查看方差分析表

```
>> tab = mdl.anova
```

	SumSq	DF	MeanSq	F	pValue
x1	45.698	1	45.698	5.849	0.026406
x2	0.2862	1	0.2862	0.036631	0.85036
x3	136.4	1	136.4	17.459	0.00056473
x4	1.4018	1	1.4018	0.17942	0.67689
x5	31.635	1	31.635	4.049	0.059407
Error	140.63	18	7.813		

结论：模型中X2，X4和X5不显著。

2. 多重共线性诊断

```
>> R = corrccoef(double(ds(:,1:5)));
```

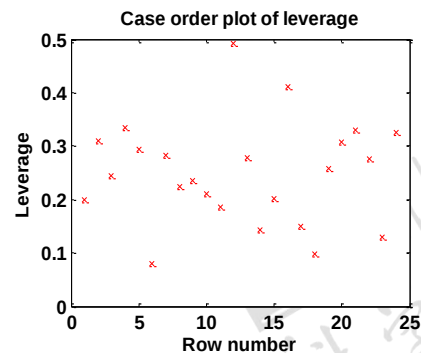
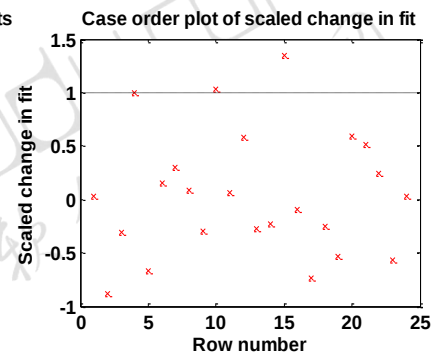
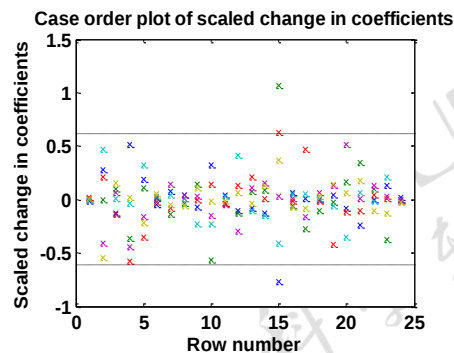
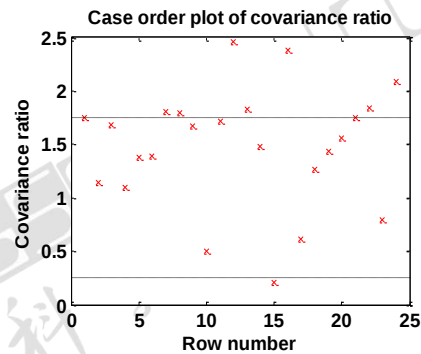
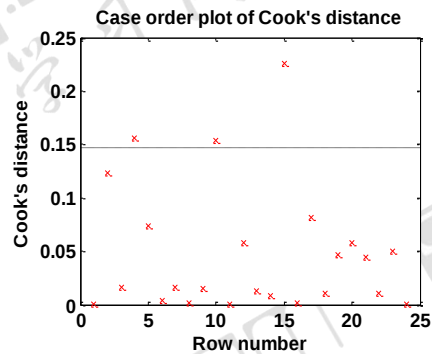
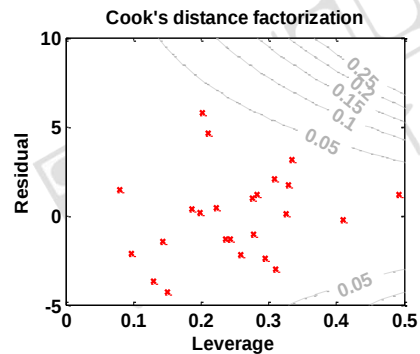
```
>> VIF = max(diag(inv(R)))
```

```
VIF = 2.4044
```

结论：方差膨胀因子 $VIF < 5$ ，自变量间不存在多重共线性。

3. 根据诊断图查找强影响点

```
>> subplot(2,3,1);  
>> mdl.plotDiagnostics('contour');  
>> subplot(2,3,2);  
>> mdl.plotDiagnostics('cookd');  
>> subplot(2,3,3);  
>> mdl.plotDiagnostics('covratio');  
>> subplot(2,3,4);  
>> mdl.plotDiagnostics('dfbetas');  
>> subplot(2,3,5);  
>> mdl.plotDiagnostics('dffits');  
>> subplot(2,3,6);  
>> mdl.plotDiagnostics('leverage');
```



4. 根据学生化残差查找异常点

```
>> id = find(abs(mdl.Residuals.Studentized) > 2);
```

结论：第10，15号观测有些异常。

四、模型改进

1. 去除不显著项 x2 和 x4

$$\hat{y} = b_0 + b_1x_1 + b_3x_3 + b_5x_5$$

```
>> terms = 'x2 + x4';
```

```
>> mdl2 = removeTerms(mdl, terms)
```

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	118.01	14.34	8.2297	7.506e-08
x1	-0.32544	0.12876	-2.5274	0.020023
x3	-4.5694	0.77413	-5.9026	8.976e-06
x5	-0.15609	0.075008	-2.0809	0.050505

$$\hat{y} = 118.01 - 0.3254x_1 - 4.5694x_3 - 0.1561x_5$$

2. 同时去除异常值和不显著项 x2 和 x4

$$\hat{y} = b_0 + b_1x_1 + b_3x_3 + b_5x_5$$

```
>> Model = 'poly10101';
```

```
>> mdl3 = LinearModel.fit(ds,Model,'Exclude',[10 15])
```

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	119.5	11.81	10.118	7.4559e-09
x1	-0.36229	0.11272	-3.2141	0.0048108
x3	-4.0411	0.62858	-6.4289	4.7386e-06
x5	-0.17739	0.05977	-2.9678	0.0082426

$$\hat{y} = 119.5 - 0.3623x_1 - 4.0411x_3 - 0.1774x_5$$

3. 增加非线性项，做完全二次回归

$$\hat{y} = b_0 + \sum_{i=1}^5 b_i x_i + \sum_{i=1}^4 \sum_{j=i+1}^5 b_{ij} x_i x_j + \sum_{i=1}^5 b_{ii} x_i^2$$

```
>> Model = 'poly22222';
```

```
>> mdl4 = LinearModel.fit(ds,Model)
```

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	1804.1	176.67	10.211	0.0020018
x1	-26.768	3.3174	-8.069	0.0039765
x2	-16.422	1.4725	-11.153	0.0015449
x3	-7.2417	17.328	-0.41792	0.70412
x4	1.7071	1.5284	1.1169	0.34543

.....

4. 在完全二次模型基础上做逐步回归

$$\hat{y} = b_0 + \sum_{i=1}^5 b_i x_i + \sum_{i=1}^4 \sum_{j=i+1}^5 b_{ij} x_i x_j + \sum_{i=1}^5 b_{ii} x_i^2$$

```
>> Model = 'poly22222';
```

```
>> mdl5 = LinearModel.stepwise(ds, Model)
```

Estimated Coefficients:

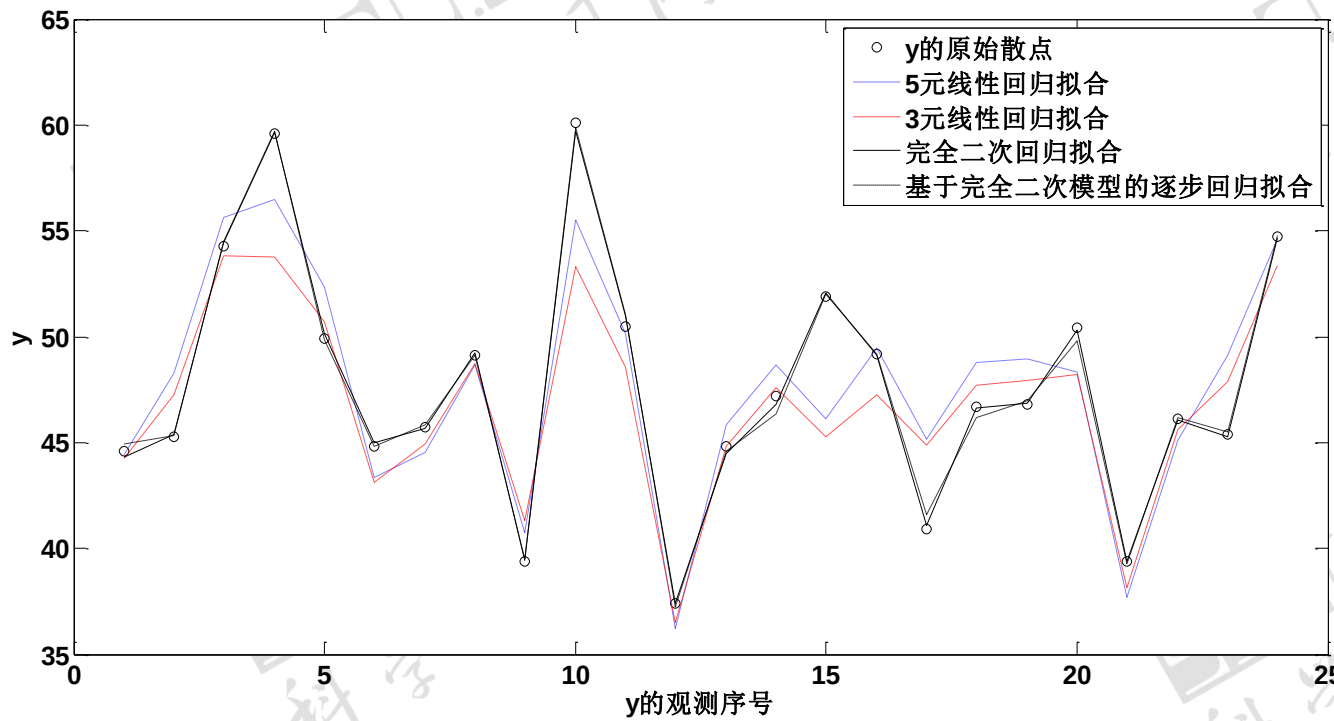
	Estimate	SE	tStat	pValue
(Intercept)	1916.6	106.48	17.999	2.2957e-08
x1	-29.485	1.6156	-18.251	2.0321e-08
x2	-15.841	0.92505	-17.124	3.553e-08
x3	3.3267	4.4986	0.7395	0.47845
x4	0.757	0.43986	1.721	0.11936

.....

五、模型比较

% 绘制拟合效果图

```
>> figure;  
>> plot(double(ds(:,6)), 'ko');  
>> hold on  
>> X = ds(:,1:5);  
>> plot mdl.predict(X, ':');  
>> plot mdl3.predict(X, 'r-');  
>> plot mdl4.predict(X, 'k');  
>> plot mdl5.predict(X, 'k--');  
>> legend('y的原始散点', '5元线性回归拟合', ...  
          '3元线性回归拟合', '完全二次回归拟合', ...  
          '基于完全二次模型的逐步回归拟合');  
>> xlabel('y的观测序号');  
>> ylabel('y');
```



第四节 多元非线性回归——

地震震中位置的确定

一、问题描述

【例12.4-1】2011年4月1日某时在某一地点发生了一次地震，图12.4-1中10个地震观测站点均接收到了地震波，观测数据如表12.4-1所列。假定地震波在各种介质和各个方向的传播速度均相等，并且在传播过程中保持不变。请根据表12.4-1中的数据确定这次地震的震中位置、震源深度以及地震发生的时间（不考虑时区因素，建议时间以分为单位）。

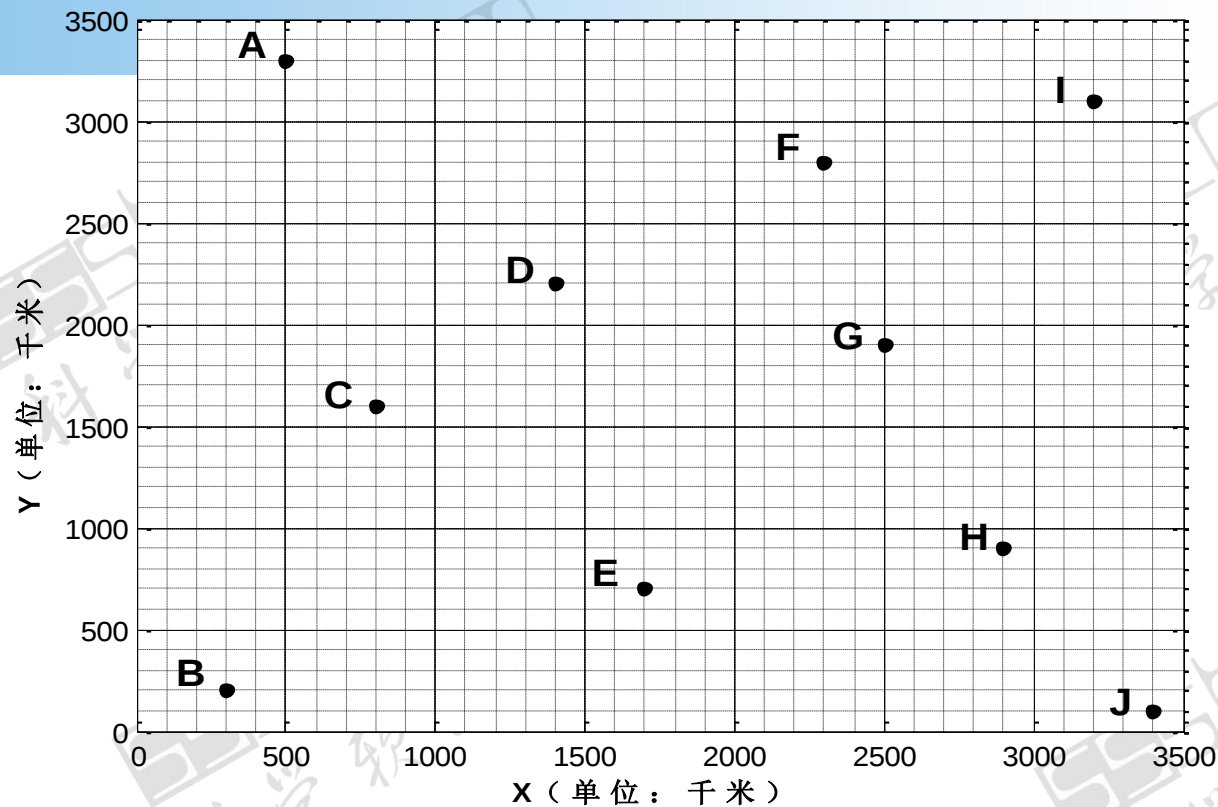


图12.4-1 地震观测站点示意图

表12.4-1 地震观测站坐标及接收地震波时间

地震观测站	横坐标x (千米)	纵坐标y (千米)	接收地震波时间
A	500	3300	4月1日9时21分9秒
B	300	200	4月1日9时19分29秒
C	800	1600	4月1日9时14分51秒
D	1400	2200	4月1日9时13分17秒
E	1700	700	4月1日9时11分46秒
F	2300	2800	4月1日9时14分47秒
G	2500	1900	4月1日9时10分14秒
H	2900	900	4月1日9时11分46秒
I	3200	3100	4月1日9时17分57秒
J	3400	100	4月1日9时16分49秒

二、模型建立

假设震源三维坐标为 (x_0, y_0, z_0) ，这里的 z_0 取正值，设地震发生的时间为2011年4月1日9时 t_0 分，地震波传播速度为 v_0 （单位：**km/s**）。用 $(x_i, y_i, 0)$ ， $i=1, 2, \dots, 10$ 分别表示地震观测站点A—J的三维坐标，用 T_i ， $i=1, 2, \dots, 10$ 表示接收到地震波的时刻，这里的 T_i ， $i=1, 2, \dots, 10$ 表示9时 T_i 分接收到地震波。建立 T 关于 x, y 的二元非线性回归模型如下：

$$T_i = t_0 + \frac{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + z_0^2}}{60v_0} + \varepsilon_i, \quad i = 1, 2, \dots, 10$$

其中 ε_i 为随机误差， x_0, y_0, z_0, v_0, t_0 为模型参数。

三、创建非线性回归模型类作多元非线性回归

1. 数据拟合

% 定义地震观测站位置坐标及接收地震波时间数据矩阵[x,y,Minutes,Seconds]

```
xyt = [500  3300  21  9  
       300  200  19  29  
       800 1600  14  51  
      1400 2200  13  17  
      1700  700  11  46  
      2300 2800  14  47  
      2500 1900  10  14  
      2900  900  11  46  
      3200 3100  17  57  
      3400  100  16  49];
```

```
xy = xyt(:,1:2); Minutes = xyt(:,3); Seconds = xyt(:,4);
```

```
T = Minutes + Seconds/60; b0 = [1000 100 1 1 1];
```

```
modelfun = @(b,x) sqrt((x(:,1)-b(1)).^2+(x(:,2)-b(2)).^2+b(3).^2)/(60*b(4))+b(5);
```

```
mnlm = NonLinearModel.fit(xy,T,modelfun,b0)
```

2. 拟合结果

Nonlinear regression model:

$$y \sim \sqrt{(x_1 - b_1)^2 + (x_2 - b_2)^2 + b_3^2} / (60 \cdot b_4) + b_5$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
b1	2200.5	0.53366	4123.5	1.5922e-17
b2	1399.9	0.48183	2905.4	9.168e-17
b3	35.144	61.893	0.56782	0.5947
b4	2.9994	0.0041439	723.82	9.5533e-14
b5	6.9863	0.02087	334.75	4.515e-12

Number of observations: 10, Error degrees of freedom: 5

Root Mean Squared Error: 0.00591

R-Squared: 1, Adjusted R-Squared 1

F-statistic vs. constant model: 8.3e+05, p-value = 9.75e-15

2. 拟合结果（续）

$$\begin{cases} x_0 &= 2200.5 \\ y_0 &= 1399.9 \\ z_0 &= 35.1 \\ v_0 &= 3 \\ t_0 &= 7 \end{cases}$$

也就是说地震发生的时间为2011年4月1日
09时07分，震中位于 $x_0 = 2200.5, y_0 = 1399.9$
处，震源深度35.1公里。

第五节 曲线（面）拟合工具

一、曲线（面）拟合工具cftool简介

1. cftool函数的用法

- cftool
- cftool(x, y)
- cftool(x, y, z)
- cftool(x, y, [], w)
- cftool(x, y, z, w)
- cftool(filename)
- cftool -v1
- cftool('-v1', x, y)
- cftool('-v1', x, y, w)

2. 界面预览

The screenshot shows the Curve Fitting Tool window with several red boxes and arrows highlighting key features:

- 选择数据** (Select Data): Points to the 'X data' and 'Y data' dropdown menus.
- 选择模型** (Select Model): Points to the 'Custom Equation' dropdown and the equation input field.
- 拟合结果** (Fit Results): Points to the 'Results' pane on the left.
- 拟合效果图** (Fit Effect Diagram): Points to the plot area showing data points and the fitted curve.

Results

General model:
 $f(x) = a \cdot \exp(b/(x+c))$

Coefficients (with 95% confidence bounds):
a = 52.43 (52.14, 52.72)
b = -0.2675 (-0.3003, -0.2347)
c = 0.7904 (0.6435, 0.9373)

Goodness of fit:
SSE: 3485
R-square: 0.7049
Adjusted R-square: 0.7045
RMSE: 1.651

Table of Fits

Fit name	Data	Fit type	SSE	R-square	DFE	Adj R-sq	RMSE	# Coeff	Validation...	Validation...	Validation...
untitled ...	y vs. x	a*exp(b/(x+...	3.4849e+03	0.7049	1278	0.7045	1.6513	3			

二、应用案例

【例12.5-1】仍考虑例12.2-1中的数据，利用曲线拟合工具建立头围关于年龄的回归方程。

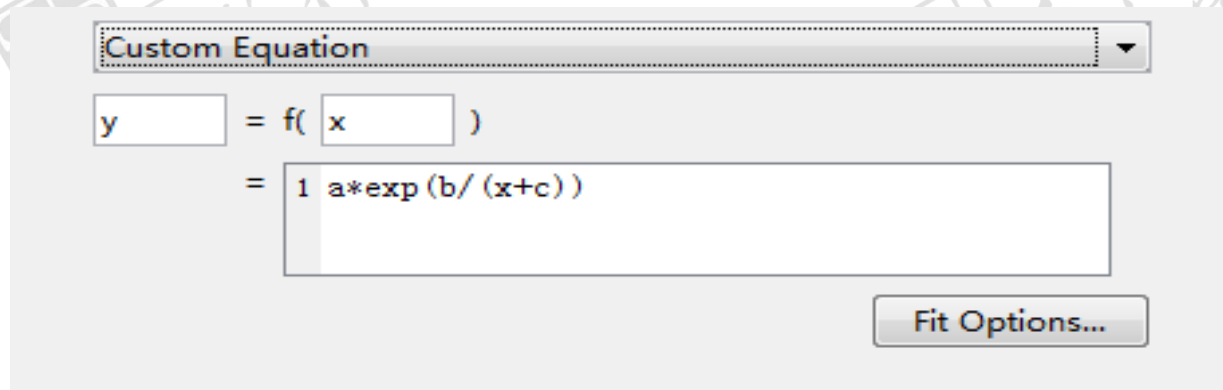
1. 导入数据

```
>> HeadData = xlsread('examp12_2_1.xls');  
>> HeadData = sortrows(HeadData, 4);  
>> x = HeadData(:, 4);  
>> y = HeadData(:, 9);  
>> cftool(x,y);
```

Fit name:	untitled fit 1
X data:	x
Y data:	y
Z data:	(none)
Weights:	(none)

2. 选择模型

➤ 负指数函数： $y = \beta_1 e^{\frac{\beta_2}{x + \beta_3}}$



The screenshot shows a software window titled "Custom Equation". It features a dropdown menu at the top with "Custom Equation" selected. Below this, there are input fields for "y" and "x" in the expression "y = f(x)". A large text box below contains the formula "1 a*exp (b/ (x+c))". At the bottom right, there is a button labeled "Fit Options...".

3. 拟合结果

Results

General model:

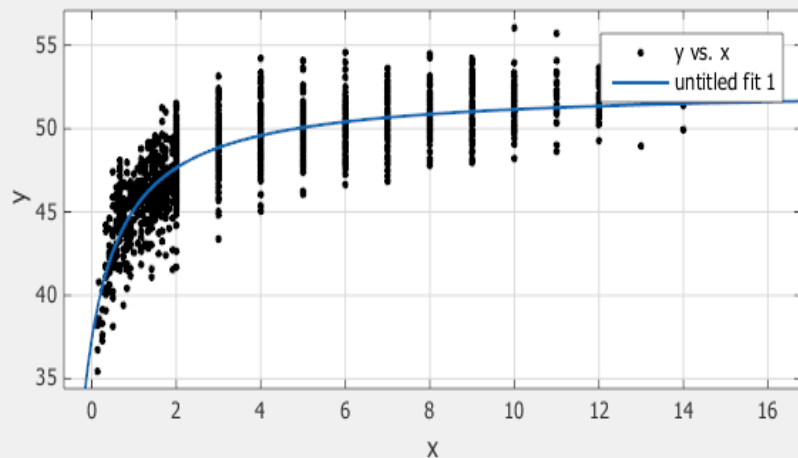
$$f(x) = a \cdot \exp(b/(x+c))$$

Coefficients (with 95% confidence bounds)

a = 52.43 (52.14, 52.72)
b = -0.2676 (-0.3003, -0.2349)
c = 0.7906 (0.6436, 0.9376)

Goodness of fit:

SSE: 3485
R-square: 0.7049
Adjusted R-square: 0.7045
RMSE: 1.651

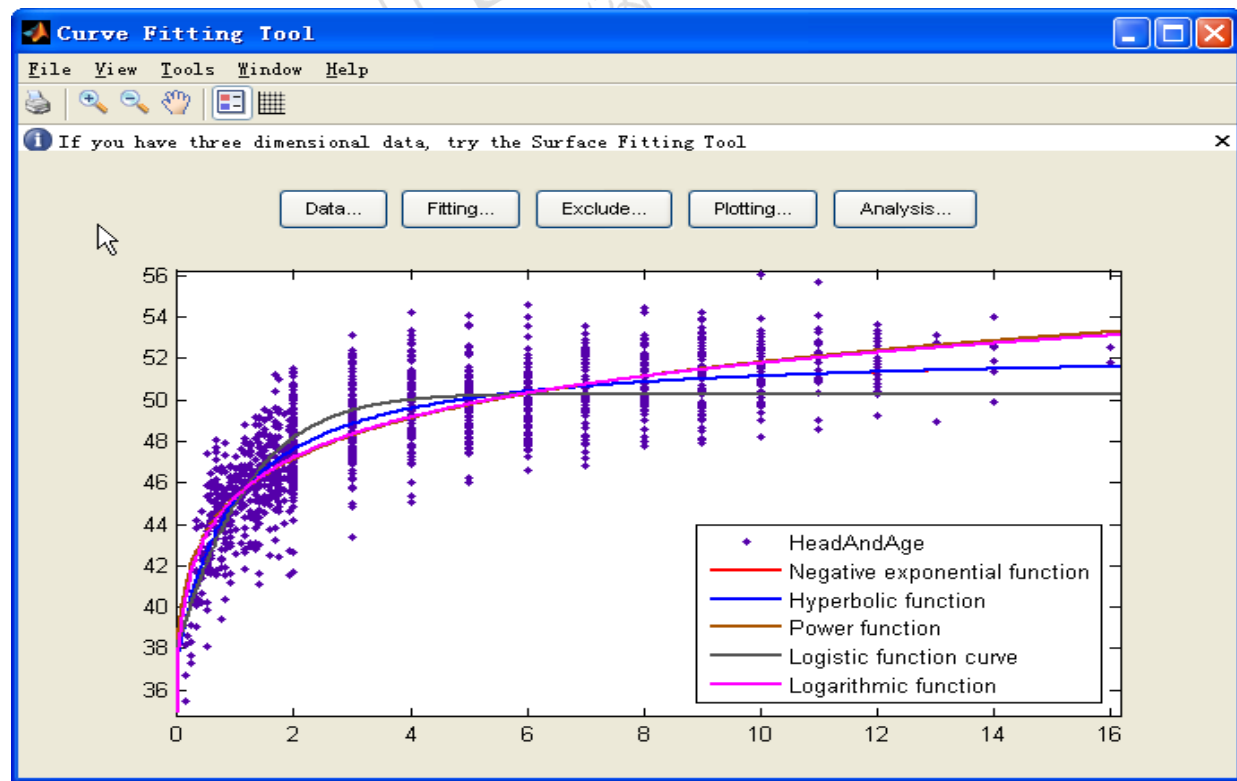


4. 模型比较

拟合较好

- 负指数曲线： $\hat{y} = 52.43e^{-\frac{0.2676}{x+0.7906}}$
- 双曲线： $\hat{y} = \frac{x + 0.6644}{0.01907x + 0.01779}$
- 幂函数曲线： $\hat{y} = 45.22(x + 0.05)^{0.05907}$
- Logistic曲线： $\hat{y} = \frac{50.3}{1 + 32.96e^{-(x+4.634)}}$
- 对数曲线： $\hat{y} = 45.18 + 2.859 \ln(x + 0.05)$

5. 多模型拟合效果对比图



Thank You