

clc % 清屏

clear all; % 删除workplace变量

close all; % 关掉显示图形窗口

format short

pz=[191 36 50 5 162 60

189 37 52 2 110 60

193 38 58 12 101 101

162 35 62 12 105 37

189 35 46 13 155 58

182 36 56 4 101 42

211 38 56 8 101 38

167 34 60 6 125 40

176 31 74 15 200 40

154 33 56 17 251 250

169 34 50 17 120 38

166 33 52 13 210 115

154 34 64 14 215 105

247 46 50 1 50 50

193 36 46 6 70 31

202 37 62 12 210 120

176 37 54 4 60 25

157 32 52 11 230 80

156 33 54 15 225 73

138 33 68 2 110 43];%每一列为一个指标，每一行为一个样本

mu=mean(pz); %求均值

sig=std(pz); %求标准差

rr=corrcoef(pz); %求相关系数矩阵

data=zscore(pz); %数据标准化

n=3; % n 是自变量的个数

m=3; % m 是因变量的个数

x0=pz(:,1:n);y0=pz(:,n+1:end);%定义自变量为前n列，因变量为n+1到m列

e0=data(:,1:n);f0=data(:,n+1:end);%e0为自变量归一化值，f0为因变量归一化值

num=size(e0,1);%求样本点的个数

chg=eye(n); % w 到 w\* 变换矩阵的初始化

for i=1:n

%计算 w，w\* 和t 的得分向量，

matrix=e0'\*f0\*f0'\*e0;

[vec,val]=eig(matrix); %求特征值和特征向量

val=diag(val); %提出对角线元素，即特征值

[val,ind]=sort(val,'descend');%降序排列，ind为排序后原下标序号

w(:,i)=vec(:,ind(1)); %提出最大特征值对应的特征向量

w\_star(:,i)=chg\*w(:,i); %计算w\*的取值

t(:,i)=e0\*w(:,i); %计算成分ti 的得分

alpha=e0'\*t(:,i)/(t(:,i)'\*t(:,i)); %计算alpha\_i

chg=chg\*(eye(n)-w(:,i)\*alpha'); %计算w 到w\*的变换矩阵

e=e0-t(:,i)\*alpha'; %计算残差矩阵

e0=e;%将残差矩阵带进下次循环

%计算ss(i)的值

beta=[t(:,1:i),ones(num,1)]\f0; %求回归方程的系数

beta(end,:)=[]; %删除回归分析的常数项

cancha=f0-t(:,1:i)\*beta; %求残差矩阵

ss(i)=sum(sum(cancha.^2)); %求误差平方和

%计算p(i)

for j=1:num

t1=t(:,1:i);f1=f0;

she\_t=t1(j,:);she\_f=f1(j,:); %把舍去的第j 个样本点保存起来

t1(j,:)=[];f1(j,:)=[]; %删除第j 个观测值

beta1=[t1,ones(num-1,1)]\f1; %求回归分析的系数

beta1(end,:)=[]; %删除回归分析的常数项

cancha=she\_f-she\_t\*beta1; %求残差向量

p\_i(j)=sum(cancha.^2);

end

p(i)=sum(p\_i);

if i>1

Q\_h2(i)=1-p(i)/ss(i-1);

else

Q\_h2(1)=1;

end

if Q\_h2(i)<0.0975

fprintf('提出的成分个数r=%d',i);

fprintf(' ');

fprintf('交叉的有效性=%f',Q\_h2(i));

r=i;

break

end

end

beta\_z=[t(:,1:r),ones(num,1)]\f0; %求Y 关于t 的回归系数

beta\_z(end,:)=[]; %删除常数项

xishu=w\_star(:,1:r)\*beta\_z; %求Y 关于X 的回归系数，且是针对标准数据的回归系数，每一列是一个回归方程

mu\_x=mu(1:n);mu\_y=mu(n+1:end);

sig\_x=sig(1:n);sig\_y=sig(n+1:end);

for i=1:m

ch0(i)=mu\_y(i)-mu\_x./sig\_x\*sig\_y(i)\*xishu(:,i); %计算原始数据的回归方程的常数项

end

for i=1:m

xish(:,i)=xishu(:,i)./sig\_x'\*sig\_y(i); %计算原始数据的回归方程的系数，每一列是一个回归方程

end

sol=[ch0;xish] %显示回归方程的系数，每一列是一个方程，每一列的第一个数是常数项,每一列为一个因变量与自变量们的回归方程

%此为还原为原始变量后的方程

%% 感觉用途不大，用到的时候再查询怎么使用

save mydata x0 y0 num xishu ch0 xish

w1=w(:,1)

w2=w(:,2)

wx1=w\_star(:,1)

wx2=w\_star(:,2)

tx1=t(:,1)'

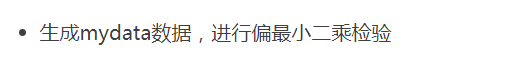
tx2=t(:,2)'

beta\_z %回归系数

xishu%系数矩阵,即未还原原始变量的系数,每一列为一个因变量与自变量的回归方程

%% 用法：分别计算出第四列第五列第六列和前三列的线性回归关系，给出系数，系数以列的方式给出，

%%每一列sol分别为自变量y的常数项系数，x1 x2 x3的系数



clc % 清屏

clear all; % 删除workplace变量

close all; % 关掉显示图形窗口

format short

load('mydata.mat')%mydata为计算偏最小二乘保存的数据集，可以用于检验

%% 更直观的解释各个自变量的作用

figure

bar(xishu')%分别画出三个自变量对三个因变量标准化后回归方程的系数的的长度图

axis tight

hold on

annotation('textbox',[0.26 0.14 0.086 0.07],'String',{'单杠'},'FitBoxToText','off');

annotation('textbox',[0.56 0.14 0.086 0.07],'String',{'弯曲'},'FitBoxToText','off');

annotation('textbox',[0.76 0.14 0.086 0.07],'String',{'跳高'},'FitBoxToText','off');%在指定位置加注释

%% 拟合效果的确定

%所有点都在对角线附近均匀分布，则效果较好

ch0=repmat(ch0,num,1);%repmat起复制矩阵组合为新矩阵的作用

yhat=ch0+x0\*xish; %计算y 的预测值

y1max=max(yhat);

y2max=max(y0);

ymax=max([y1max;y2max]);

cancha=yhat-y0; %计算残差

figure

subplot(2,2,1)

plot(0:ymax(1),0:ymax(1),yhat(:,1),y0(:,1),'\*')

title('单杠成绩预测')

subplot(2,2,2)

plot(0:ymax(2),0:ymax(2),yhat(:,2),y0(:,2),'O')

title('弯曲成绩预测')

subplot(2,1,2)

plot(0:ymax(3),0:ymax(3),yhat(:,3),y0(:,3),'H')

title('跳高成绩预测')

%% 绘制拟合效果图和权重比重图

