MATHWORK

part2

```
format bank;
%%
load fisheriris0.mat%加载文件
length=data(:,1);%获取矩阵数据
width=data(:,2);
%%
figure;
plot(length,width,'k*','MarkerSize',8);%画图
title('鸢尾花数据');%图例
xlabel('长度');
ylabel('宽度');
%%
rng(1);
[idx,C] = kmeans(data,3);%3为簇的数量
%%
% 可视化聚类结果
figure;
gscatter(length,width,idx,'br');
hold on;
plot(C(:,1),C(:,2),'kx', 'MarkerSize',15,'LineWidth',3);
legend('length','width','质心');
title '鸢尾花数据聚类结果';
xlabel '长度';
ylabel '长度';
%%
%欧氏最小距离
% 计算数据点与每个簇中心点的欧氏距离
Points=[5.3,3.7;5.0,3.3;5.1,2.5;5.7,2.8;5.9,3.0];
distances=zeros(5,3);
for i = 1:5
    distances(i, 1) = pdist([Points(i,:);C(1,:)], 'euclidean');
    distances(i, 2) = pdist([Points(i, :); C(2,:)], 'euclidean');
    distances(i, 3) = pdist([Points(i, :); C(3,:)], 'euclidean');
end
% 找到每个数据点的最小距离对应的簇
[~, idx2] = mink(distances, 1, 2);
```

part3

```
load runoff.txt
train1=runoff(1:end-1,1:4)
train1=train1';
train2=runoff(1:end-1,5)
train2=train2';

net = fitnet(9);
net.layers{1}.transferFcn = 'tansig'; % 隐含层的激活函数
net.layers{2}.transferFcn = 'purelin'; % 输出层的激活函数
net.trainparam.show = 50; % 每次循环50次
net.trainParam.epochs = 5000; % 最大循环500次
net.trainparam.goal = 0.0001; % 期望目标误差最小值
```

```
net.trainFcn = 'trainlm'; % 学习规则
net = train(net,train1,train2); % 训练数据

 Y = sim(net,train1) % 预测
hold on
plot(train2,'ro')
plot(Y,'b*')
legend('原始数据','预测值')
hold off;

sample=runoff(end,1:4);
sample=sample';
output=runoff(end,end);

D= sim(net,sample) % 预测
hold on;
plot(output,'ro')
plot(D,'b*')
legend('原始数据','预测值')
```

part3

```
% 有向图的最短路径
s = [1 1 2 2 2 3 3 3 4 5 6]; % start
t = [2 3 4 5 6 4 5 6 7 7 7]; % target
W = [4 2 1 3 3 3 2 3 4 3 1]; % weight
n = max(max(s, t)); % 城市数目
names = {'A' 'B1' 'B2' 'C1' 'C2' 'C3' 'D'};
G = digraph(s,t,W,names);
p = plot(G,'Layout','force','EdgeLabel',G.Edges.Weight);

[path,dist,pred] = shortestpath(G,1,7,'Method','positive')

highlight(p,path,'EdgeColor','r','LineWidth',1.5)
```