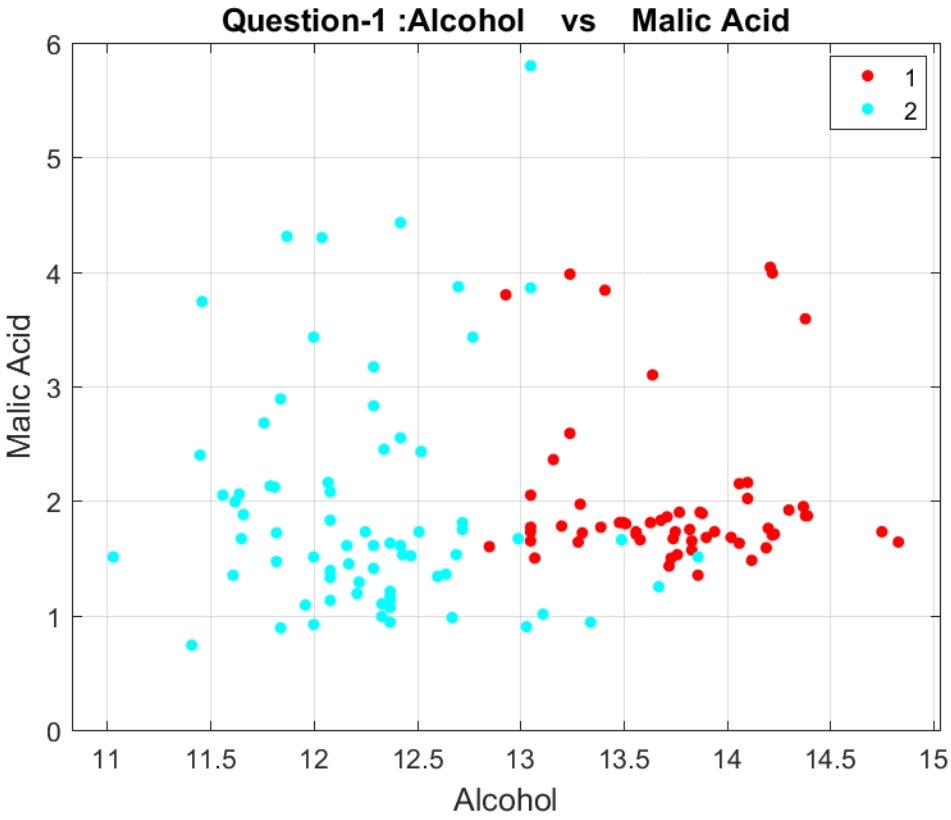
Question-0 (Preprocessing) Remove all row corresponding to the labeled winery 3. After this process, you should have only 2 labels on your data.

```
% LOAD DATASET %
load('wine.mat');

% LOAD VALUES INTO VARIABLES %
wine=double(A);

wine_label = wine(:, 14);
wine_data = wine(:, 1:13);
% DEFINE ATTRIBUTES AND CLASSES %
categories = {'Alcohol'; 'Malic acid'; 'Ash'; 'Alcalinity of ash';
'Magnesium'; 'Total phenols'; 'Flavanoids'; 'Nonflavanoid phenols';
'Proanthocyanins'; 'Color intensitys'; 'Hue'; 'OD280/OD315 of diluted wines';
'Proline'};
classnumber = 3;

% Question-0 (Preprocessing) %
%Remove all row corresponding to the labeled winery 3. After this process,
%you should have only 2 %
idx = (wine_label > 2);
wine(idx,:) = [];
wine_label = wine(:, 14);
wine_data = wine(:, 1:13);
```

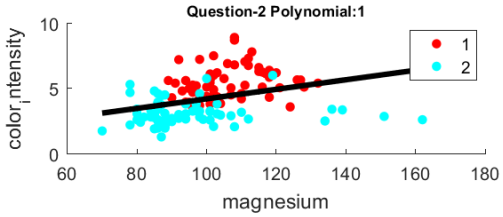
Question-1 (0.25 pts) Load the data and plot (visualize) the data points of wines by their Alcohol (feature 1 in x axis) and Malic acid (feature 2 in y axis).

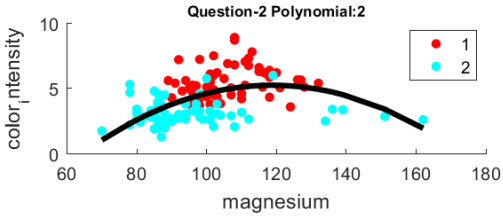
```
%Question-1 (0.25 pts)
% Load the data and plot (visualize) the data points of wines by their
Alcohol (feature 1 in x axis) and
% Malic acid (feature 2 in y axis).
alcohol=wine(:,1);
malic_acid=wine(:,2);
c = linspace(1,100,length(wine(:,14)));
figure
gscatter(alcohol,malic_acid,wine_label)
title('Question-1 :Alcohol    vs    Malic Acid', 'FontSize', 12);
xlabel('Alcohol','FontSize', 12)
ylabel('Malic Acid','FontSize', 12)
grid on
```

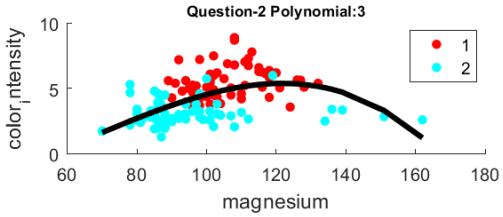

Question-1 :Alcohol vs Malic Acid
1
2
Malic Acid
Alcohol
0
1
2
3
4
5
6
11
11.5
12
12.5
13
13.5
14
14.5
15

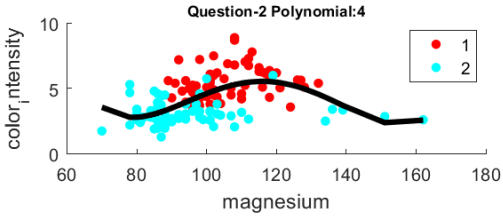
Question-2 (1 pts) Pick Magnesium and Color intensity as your two features and for degrees n =1, …, 10 fit a polynomial of degree n to your data. Plot those fitting lines on the data. You can check the correctness of your solution with MALAB's built-in curve fitting function

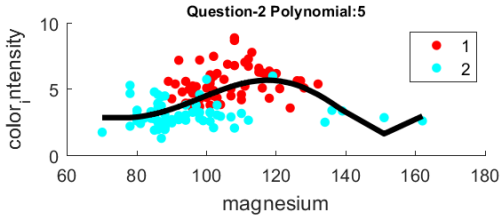
```
% Question-2 (1 pts)
% Pick Magnesium and Color intensity as your two features and for degrees n
=1, …, 10 fit a polynomial
% of degree n to your data. Plot those fitting lines on the data. You can
check the correctness of your
% solution with MALAB's built-in curve fitting function.
magnesium=wine(:,5);
color_intensity=wine(:,10);
figure('Renderer', 'painters', 'Position', [0 0 900 900])

xlabel('Magnesium','FontSize', 8)
ylabel('Color Intensity','FontSize', 8)
grid on


for i=1:10
    p = polyfit(magnesium,color_intensity,i);
    y1 = polyval(p,magnesium);
    B = [magnesium, y1];
    C = sortrows(B,1);

    subplot(5,2,i);
    hold on
    gscatter(magnesium,color_intensity,wine_label)
    plot(C(:,1),C(:,2),'k','LineWidth',3);
    hold off
    str = sprintf('Question-2 Polynomial:%d', i);
    title(str, 'FontSize', 8);
end
```

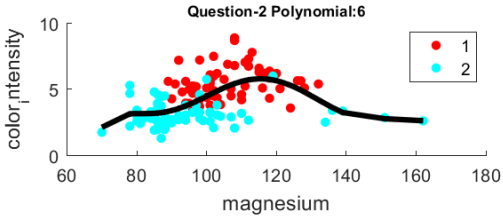

Question-2 Polynomial:1
color_intensity
magnesium
1
2

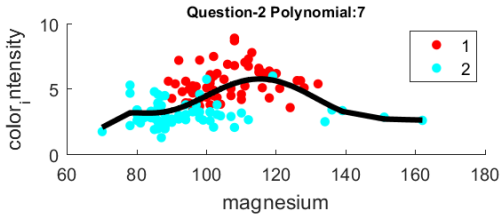

Question-2 Polynomial:2
color_intensity
magnesium
1
2

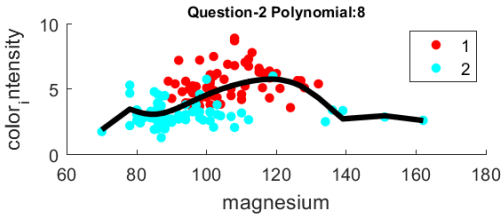

Question-2 Polynomial:3
color_intensity
magnesium
1
2

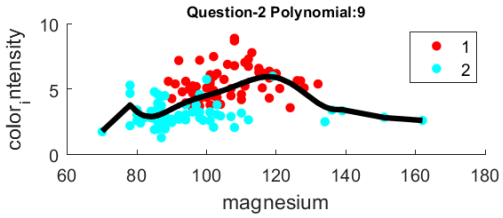

Question-2 Polynomial:4
color_intensity
magnesium
1
2

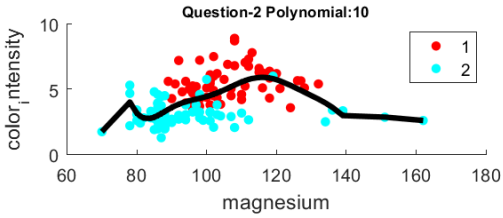

Question-2 Polynomial:5
color_intensity
magnesium
1
2


Question-2 Polynomial:6
color_intensity
magnesium
1
2


Question-2 Polynomial:7
color_intensity
magnesium
1
2

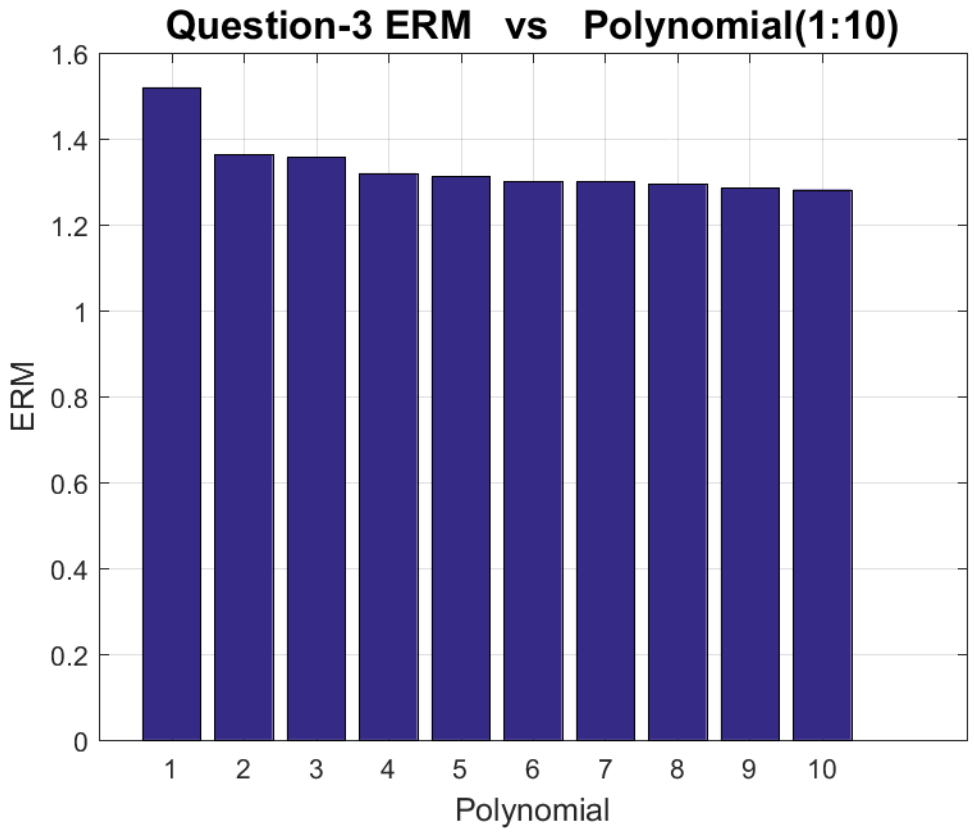

Question-2 Polynomial:8
color_intensity
magnesium
1
2


Question-2 Polynomial:9
color_intensity
magnesium
1
2


Question-2 Polynomial:10
color_intensity
magnesium
1
2

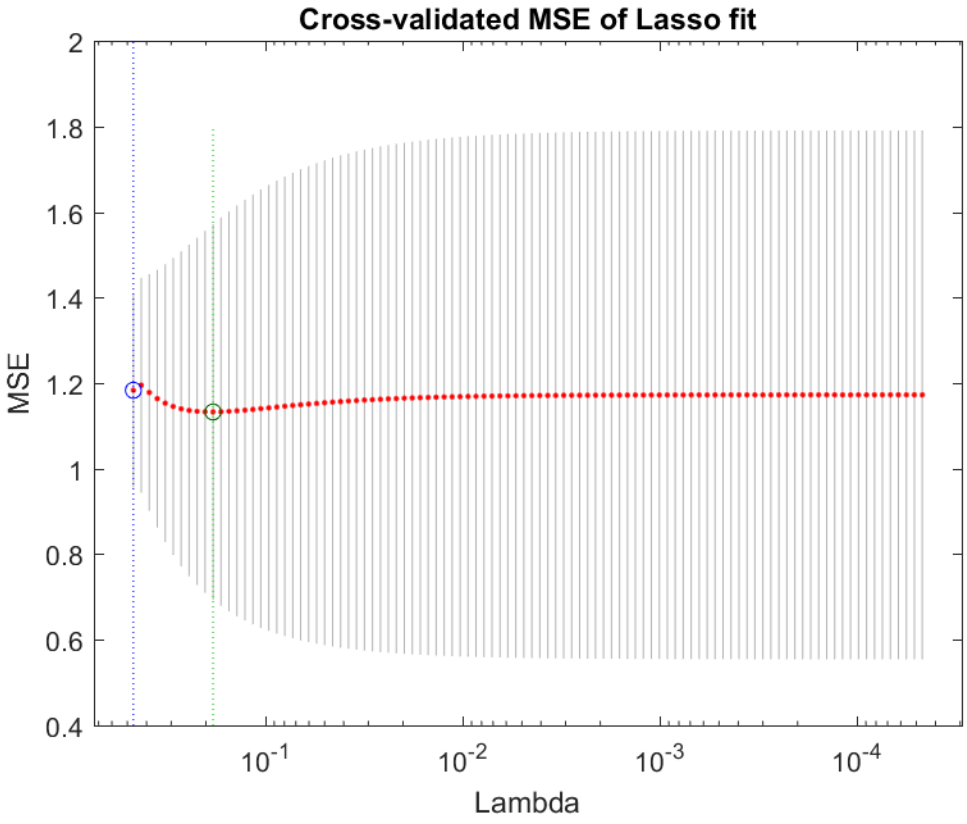
Question-3 (1 pts) For each learned function (n=1, …, 10), compute the empirical square loss (ERM) on data and plot it as a function of n.

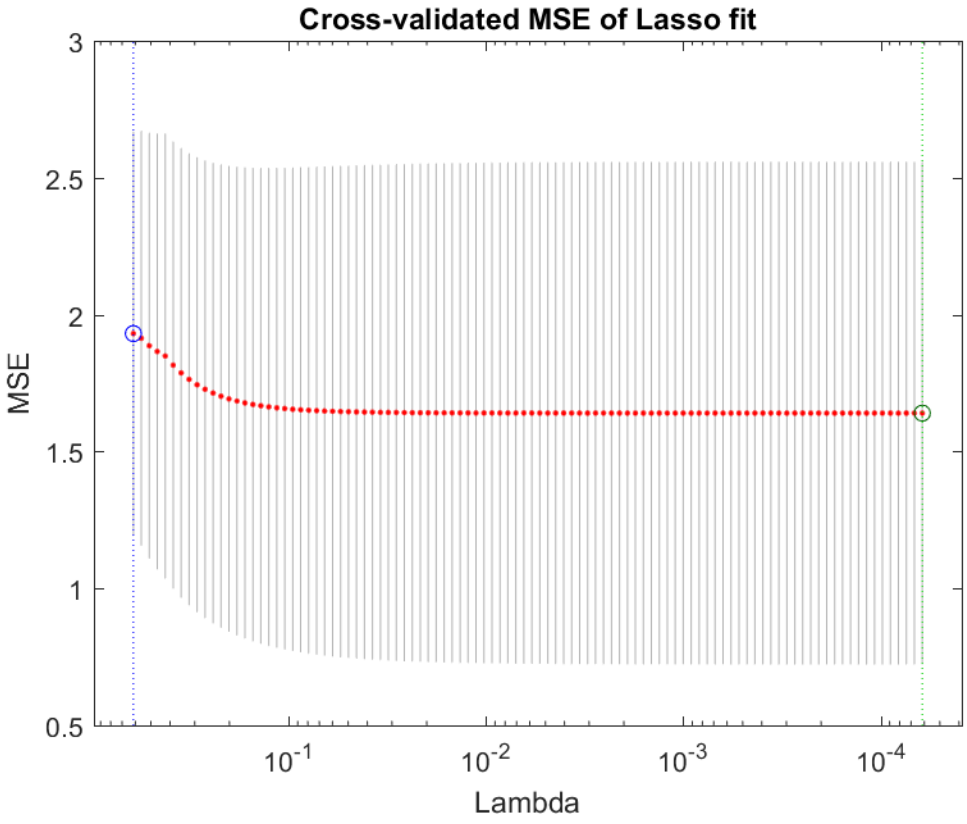
```
% Question-3 (1 pts)
% For each learned function (n=1, …, 10), compute the empirical square loss
(ERM) on data and plot
% it as a function of n.
RMSE=[10];
for i=1:10
    p = polyfit(magnesium,color_intensity,i);
    f = polyval(p,magnesium);
%     T = table(magnesium,color_intensity,f,color_intensity-
f,'VariableNames',{'X','Y','Fit','FitError'})
    RMSE(i)=sqrt(mean((color_intensity-f).^2));
end
figure()
t=1:10;
bar(t,RMSE);
xlabel('Polynomial','FontSize', 12)
ylabel('ERM','FontSize', 12)
title('Question-3 ERM   vs   Polynomial(1:10)', 'FontSize', 15);
grid on
```

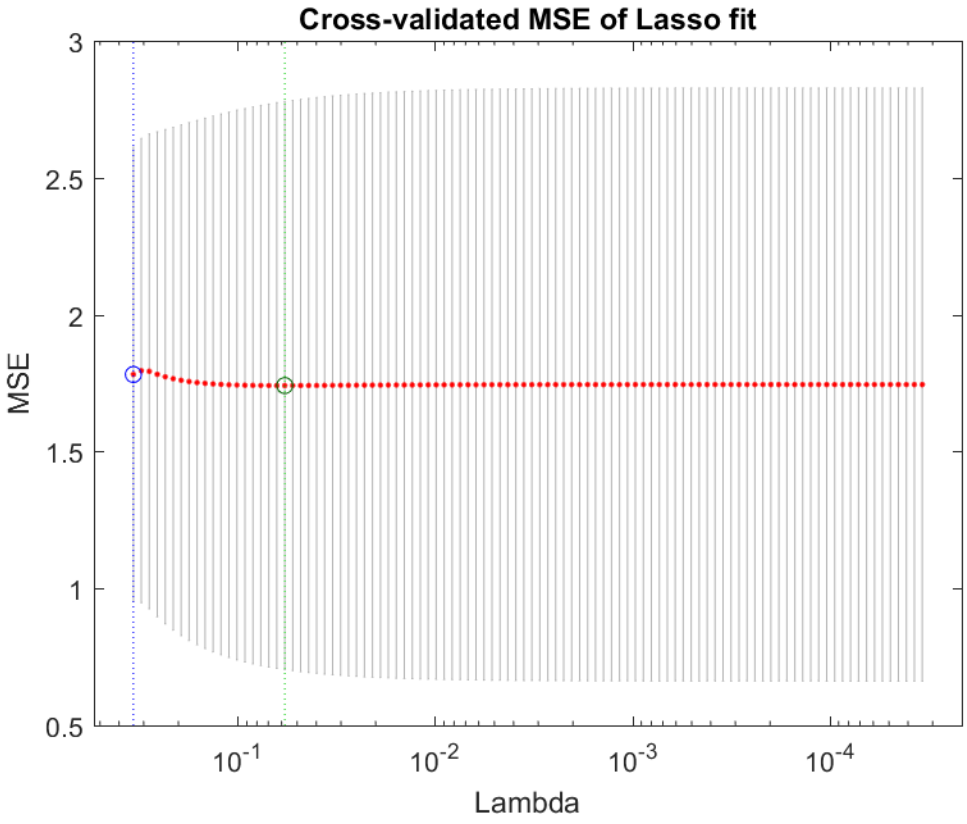


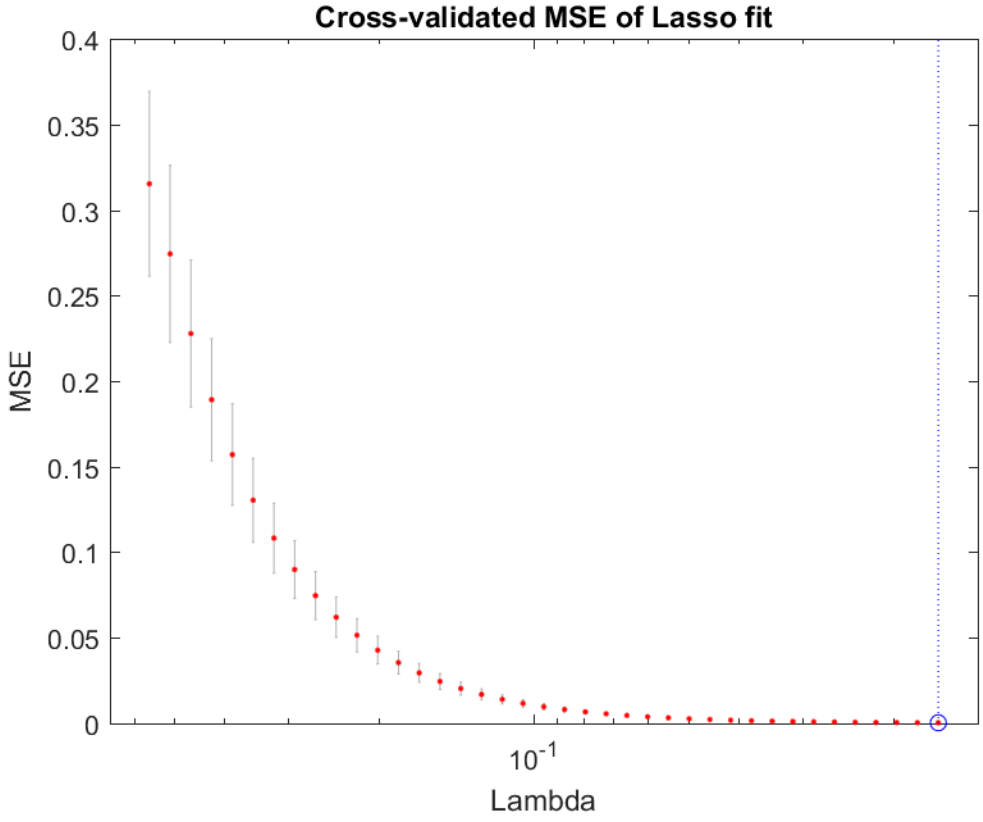
Question-4 (1 pts) Now, fix the n=10 and add a lasso regularization for your predictor of data. Vary the regularization parameter in a loop of 20 and visualize the RLM loss. You can check the correctness of your solution with MALAB's built-in Lasso.

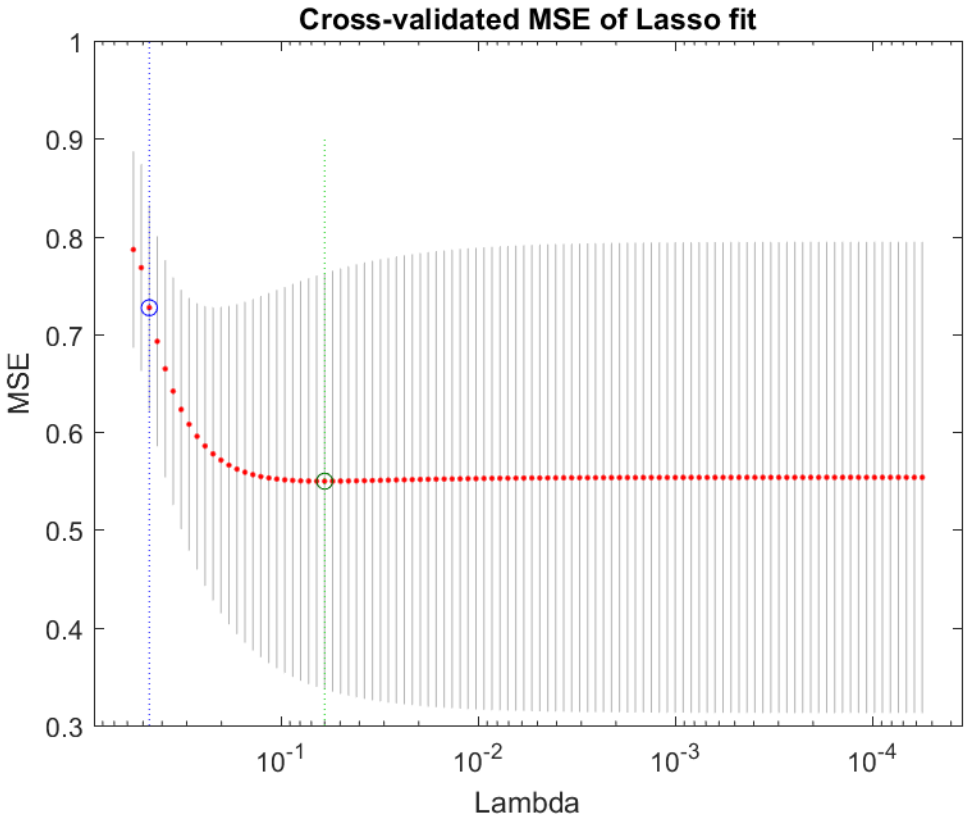
```
% Question-4 (1 pts)
% Now, fix the n=10 and add a lasso regularization for your predictor of
data. Vary the regularization
% parameter in a loop of 20 and visualize the RLM loss. You can check the
correctness of your solution
% with MALAB's built-in Lasso.
figure()
grid on
for i=1:20
    p = polyfit(magnesium,color_intensity,i);
    y1 = polyval(p,magnesium);
    [b,fitinfo] = lasso(magnesium,y1,'CV',10);
    lam = fitinfo.Index1SE;
    fitinfo.MSE(lam);
    hold on
    lassoPlot(b,fitinfo,'PlotType','CV');
    str = sprintf('Question-4 Lasso Regularization Cross Validation fold:%d',
i);
    title(str, 'FontSize',10);
    hold off
end
```

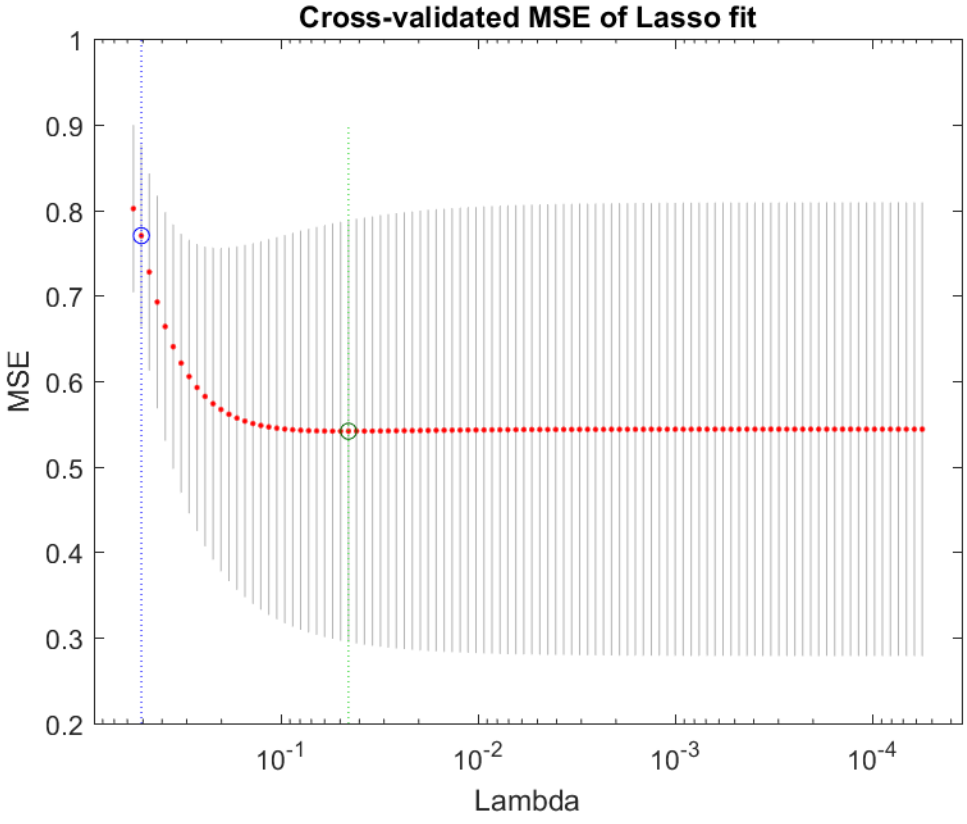

Cross-validated MSE of Lasso fit
MSE
Lambda
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
10-1
10-2
10-3
10-4

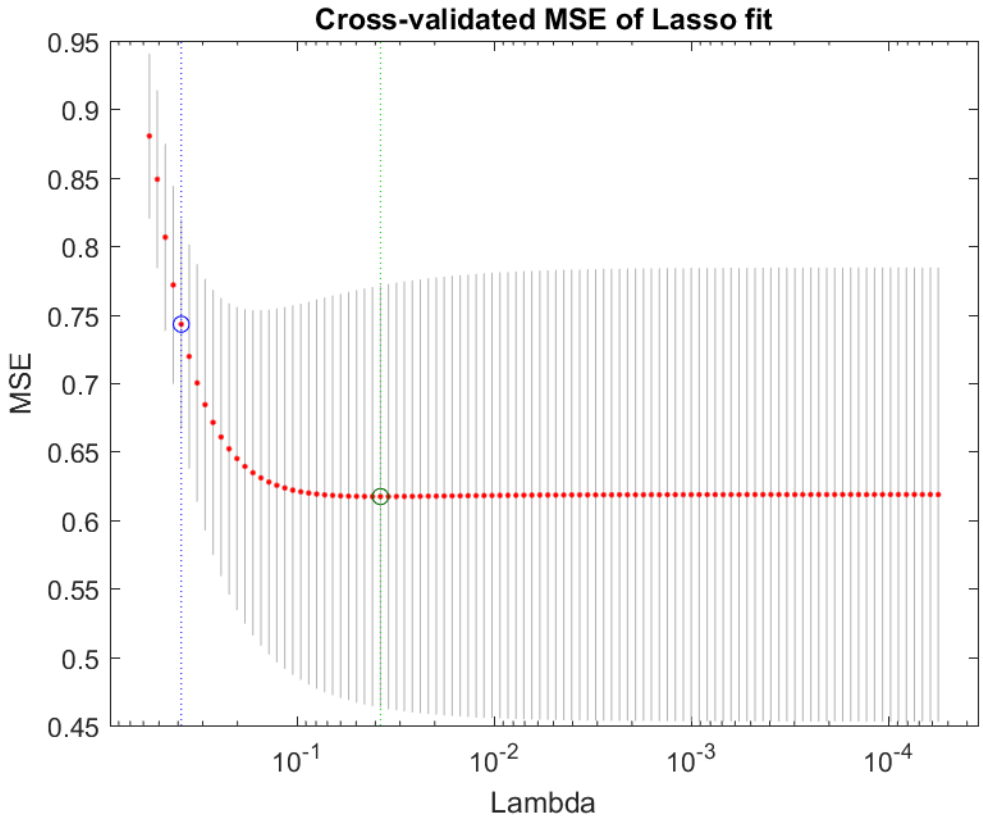

Cross-validated MSE of Lasso fit
MSE
Lambda
3
2.5
2
1.5
1
0.5
10-1
10-2
10-3
10-4

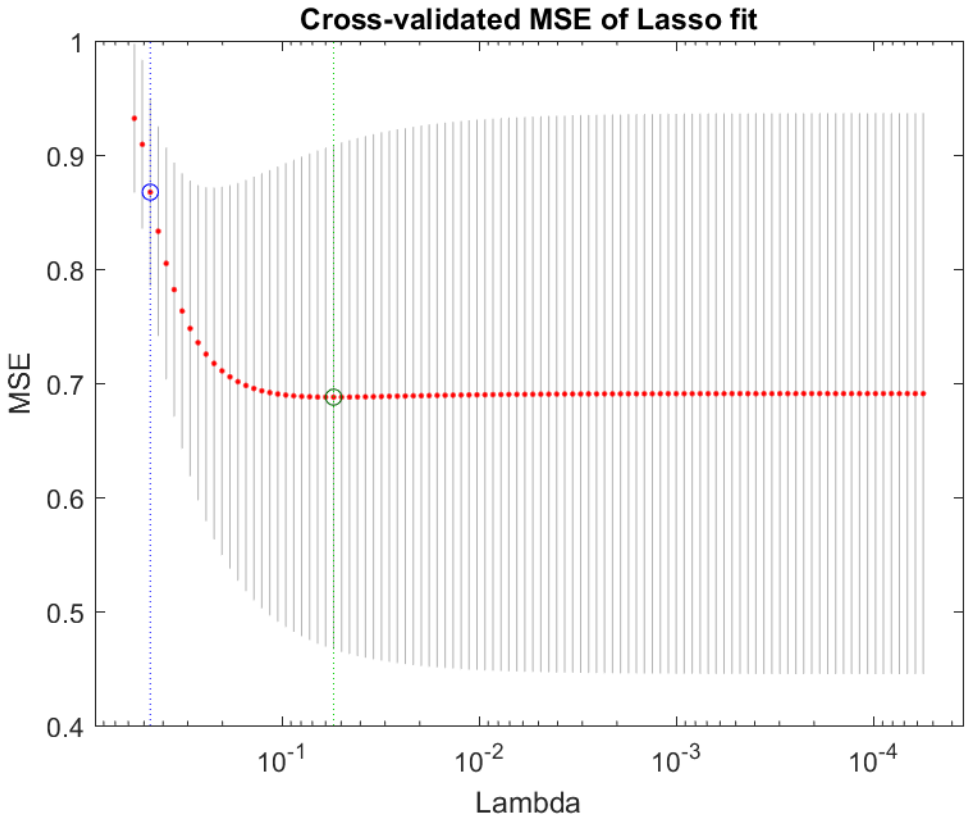

Cross-validated MSE of Lasso fit
MSE
Lambda
3
2.5
2
1.5
1
0.5
10-1
10-2
10-3
10-4

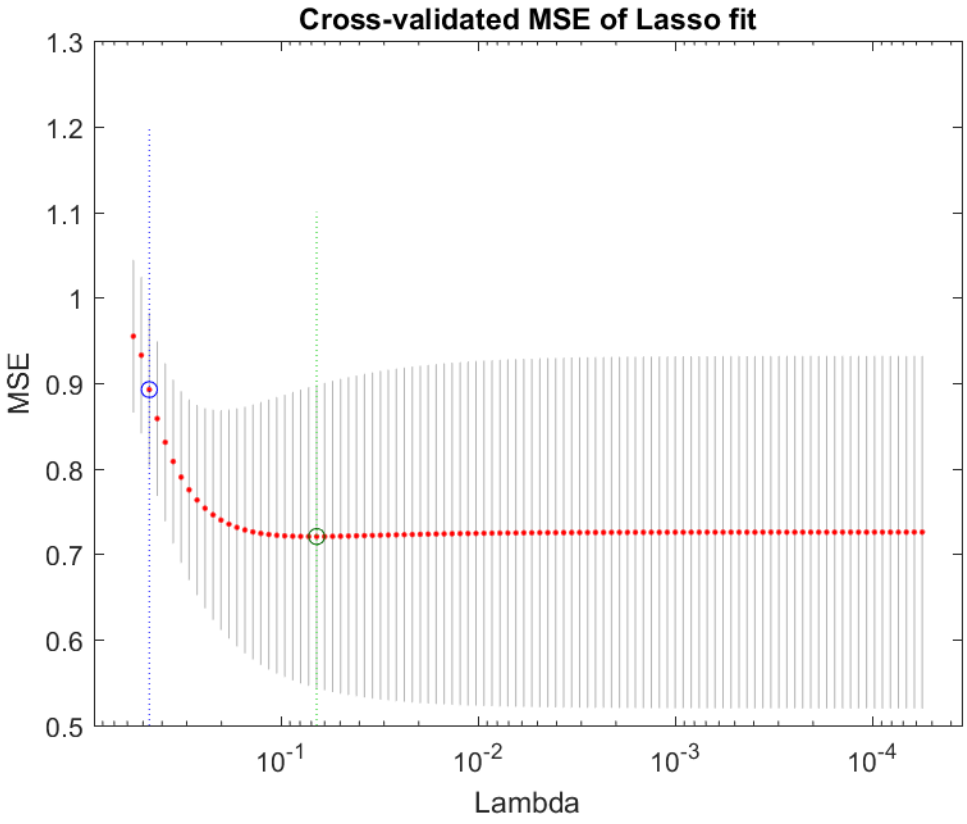

Cross-validated MSE of Lasso fit
MSE
Lambda
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
10-1

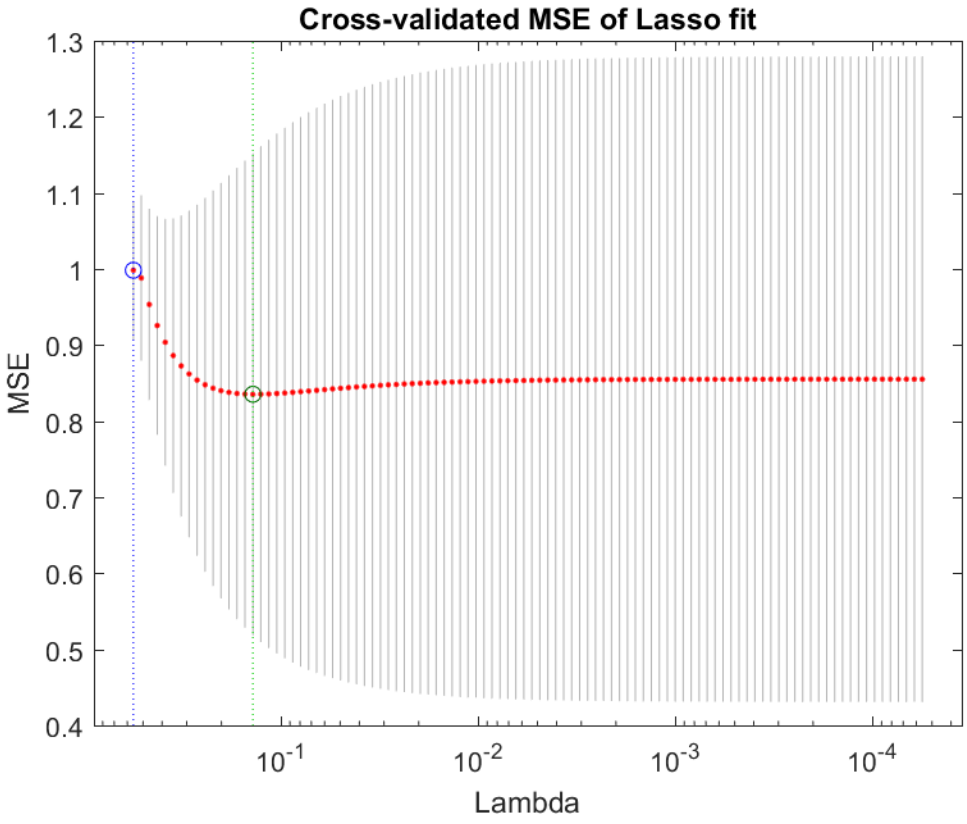

Cross-validated MSE of Lasso fit
MSE
Lambda
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
10-1
10-2
10-3
10-4

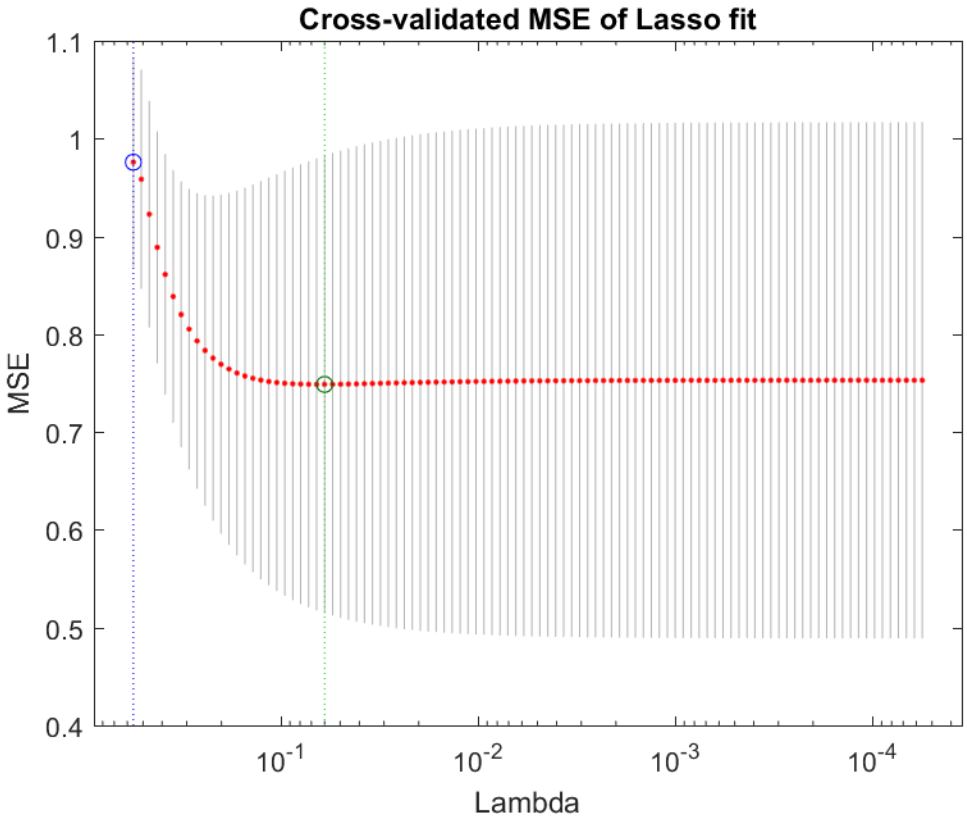

Cross-validated MSE of Lasso fit
MSE
Lambda
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
10-1
10-2
10-3
10-4

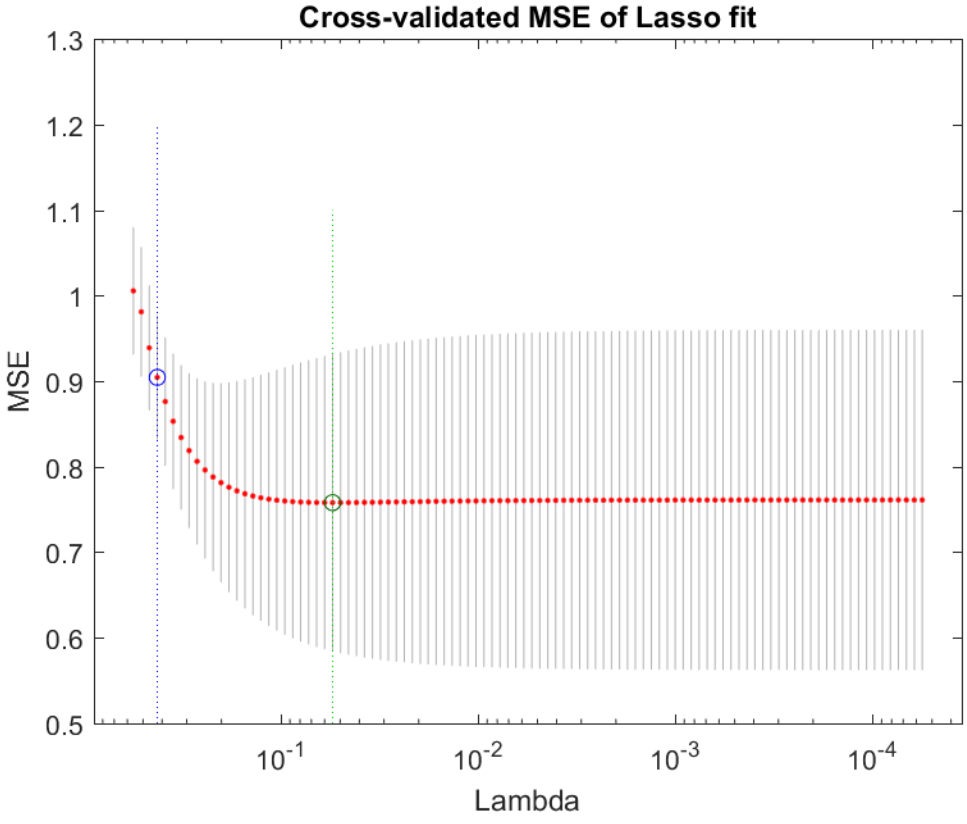

Cross-validated MSE of Lasso fit
MSE
Lambda
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
10-1
10-2
10-3
10-4

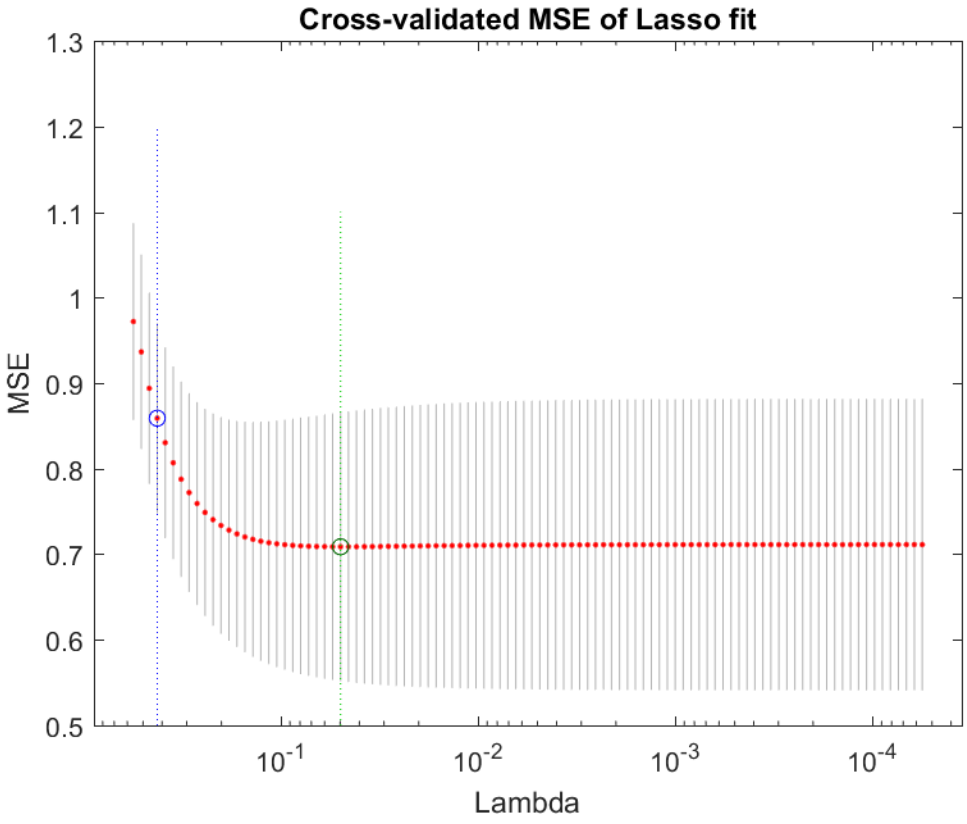

Cross-validated MSE of Lasso fit
MSE
Lambda
1
0.9
0.8
0.7
0.6
0.5
0.4
10-1
10-2
10-3
10-4

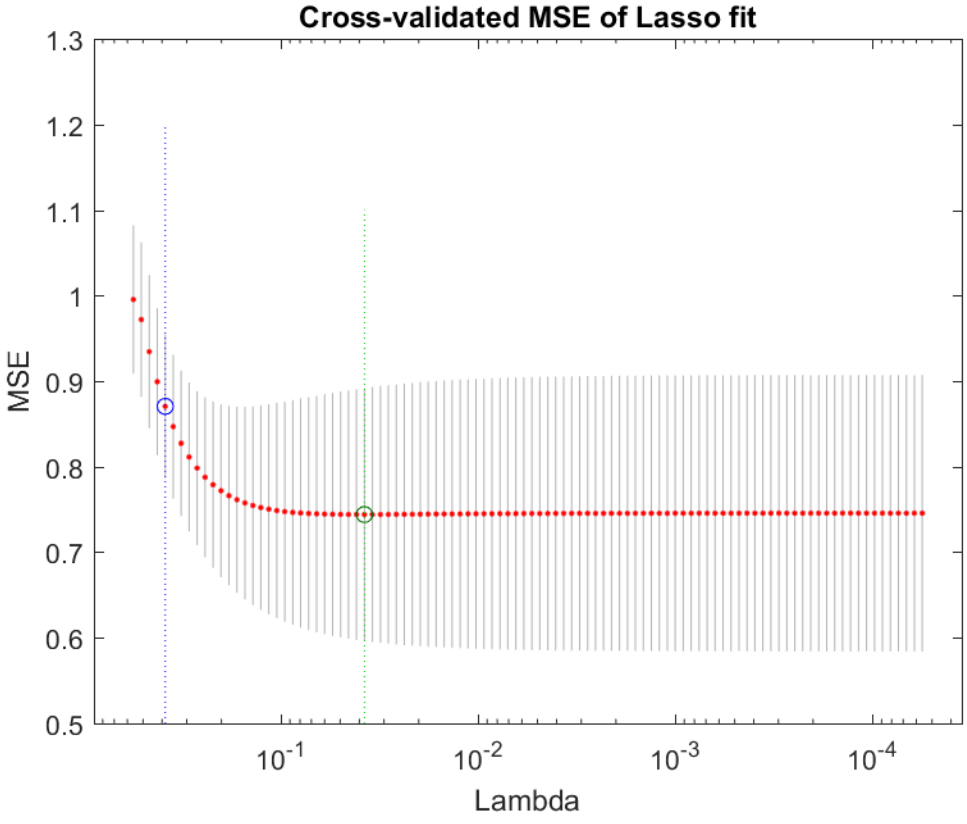

Cross-validated MSE of Lasso fit
MSE
Lambda
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
10-1
10-2
10-3
10-4

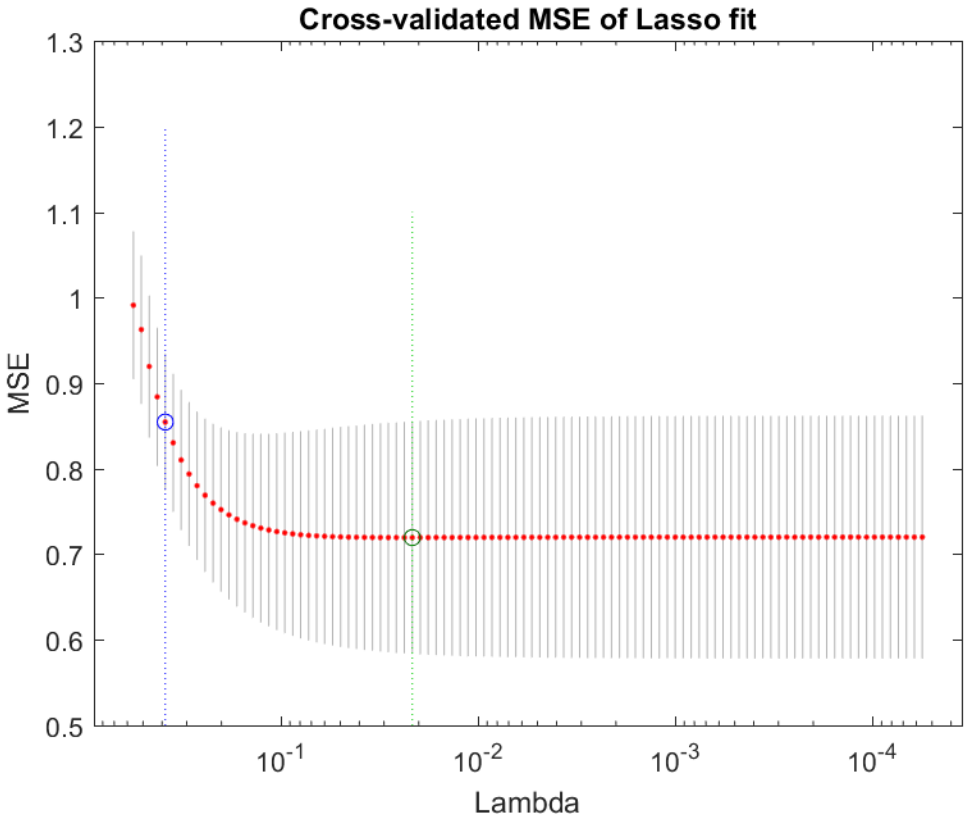

Cross-validated MSE of Lasso fit
MSE
Lambda
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
10-1
10-2
10-3
10-4

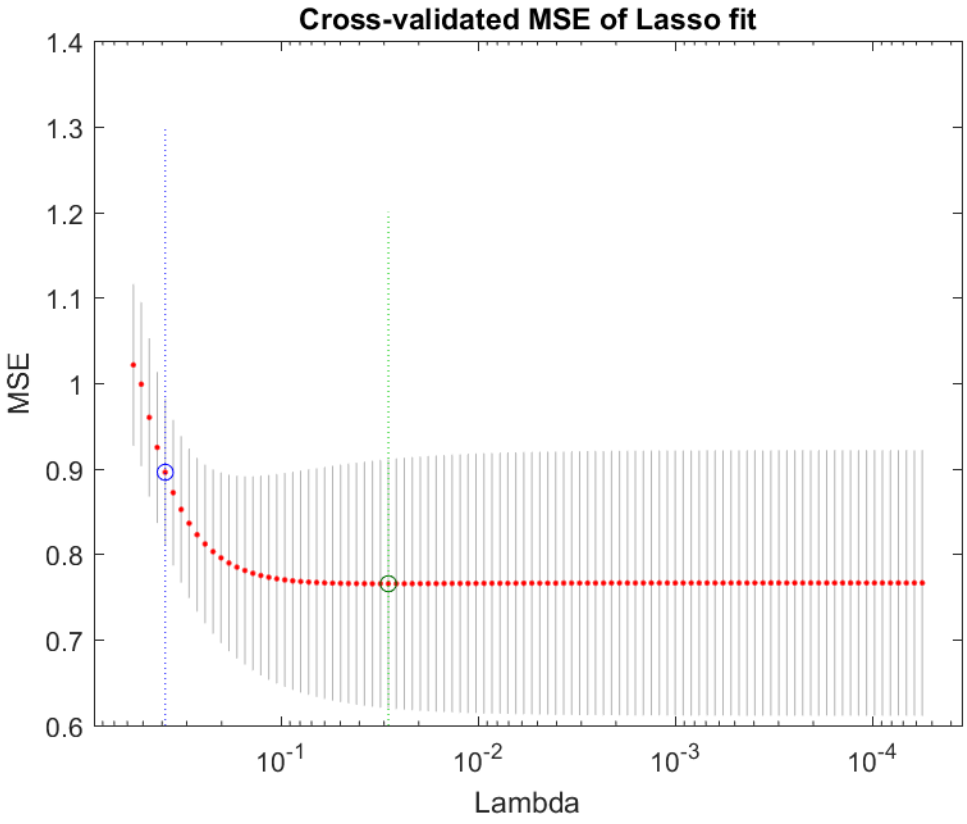

Cross-validated MSE of Lasso fit
MSE
Lambda
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
10-1
10-2
10-3
10-4

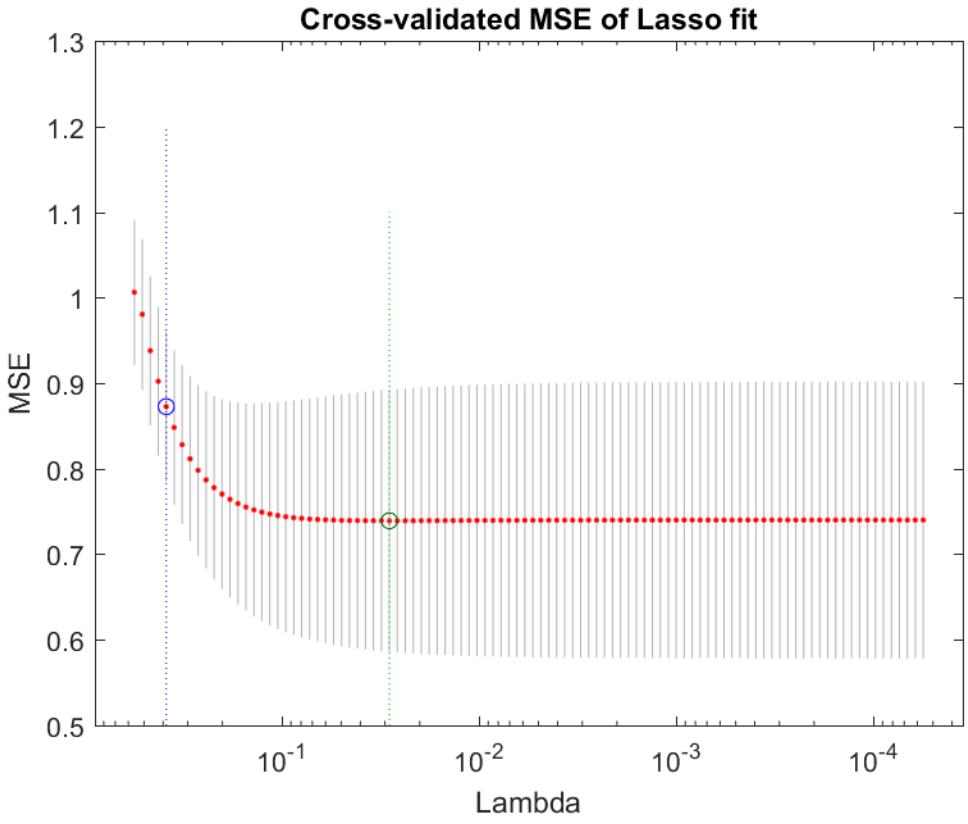

Cross-validated MSE of Lasso fit
MSE
Lambda
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
10-1
10-2
10-3
10-4

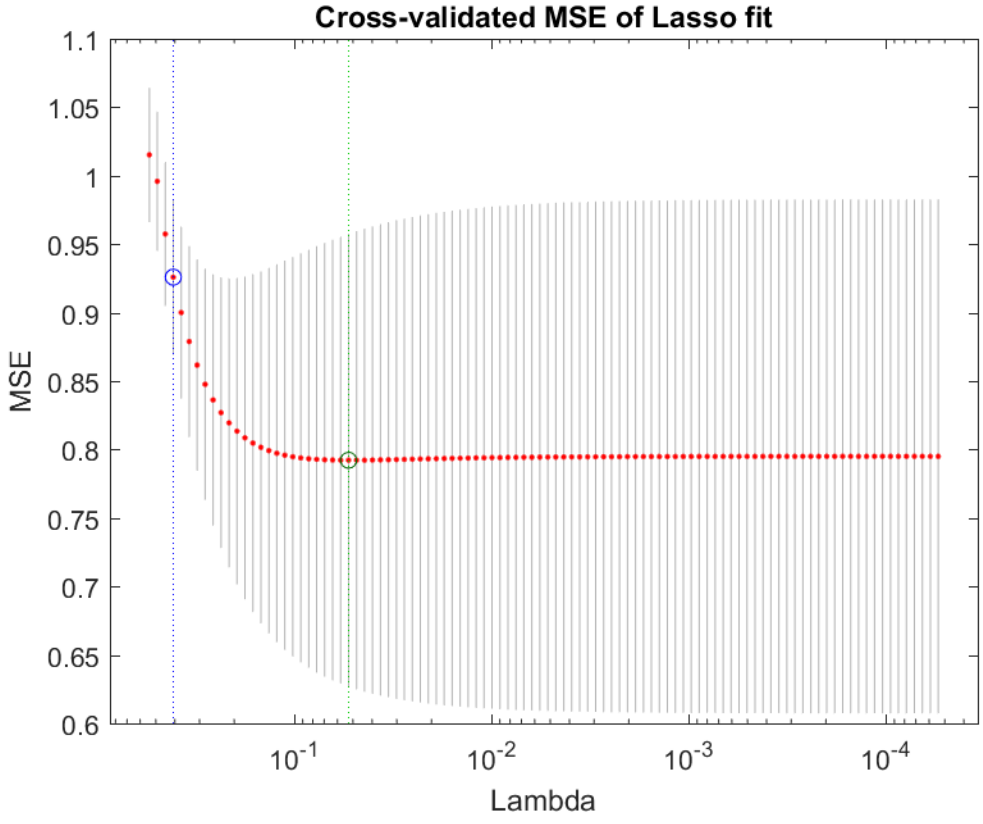

Cross-validated MSE of Lasso fit
MSE
Lambda
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
10-1
10-2
10-3
10-4

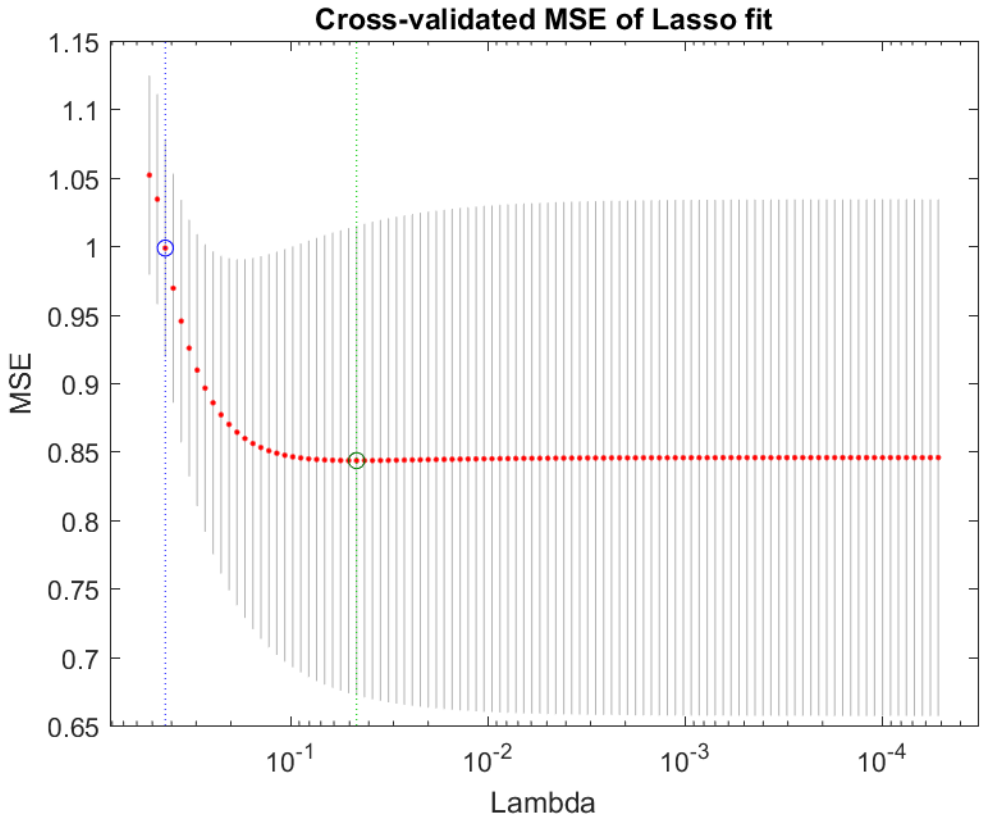

Cross-validated MSE of Lasso fit
MSE
Lambda
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
10-1
10-2
10-3
10-4

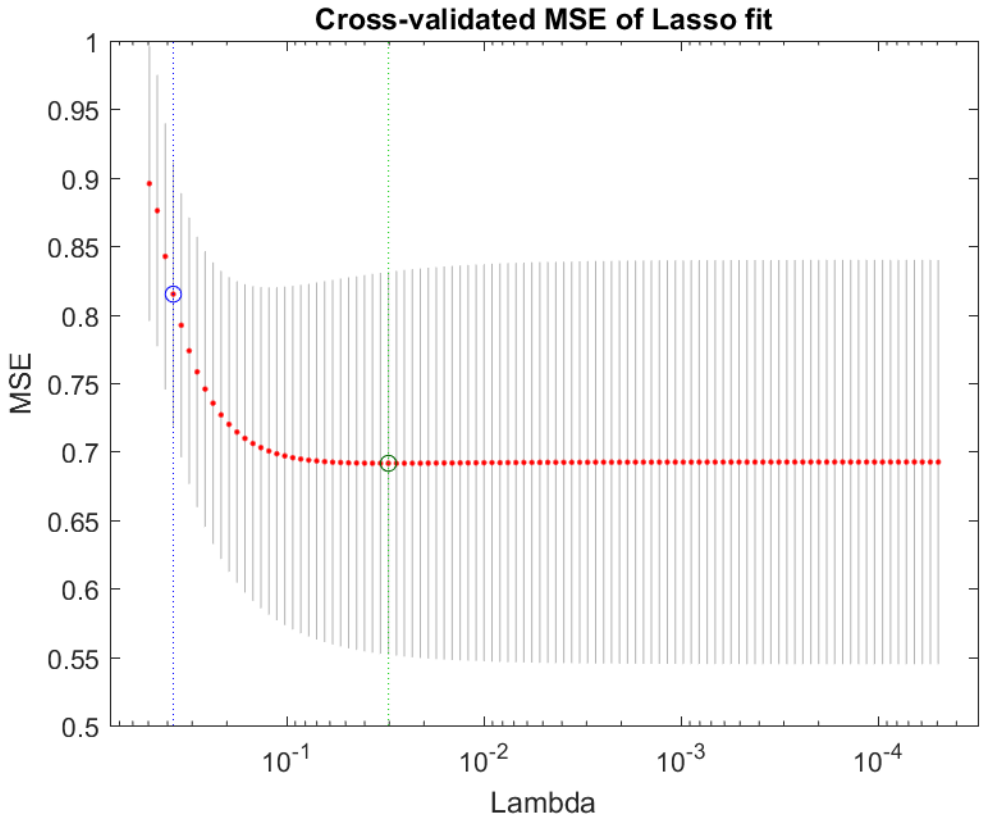

Cross-validated MSE of Lasso fit
MSE
Lambda
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
10-1
10-2
10-3
10-4


Cross-validated MSE of Lasso fit
MSE
Lambda
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
10-1
10-2
10-3
10-4


Cross-validated MSE of Lasso fit
MSE
Lambda
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
10-1
10-2
10-3
10-4

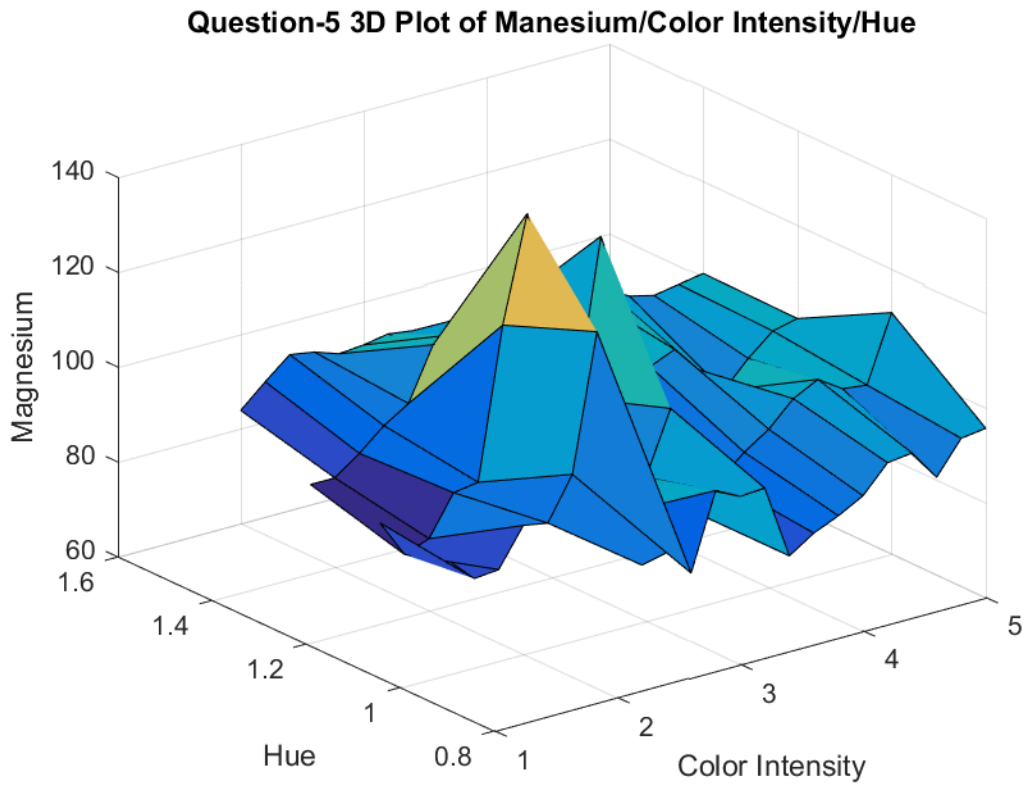

Cross-validated MSE of Lasso fit
MSE
Lambda
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
10-1
10-2
10-3
10-4


Cross-validated MSE of Lasso fit
MSE
Lambda
1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
10-1
10-2
10-3
10-4


Cross-validated MSE of Lasso fit
MSE
Lambda
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
10-1
10-2
10-3
10-4

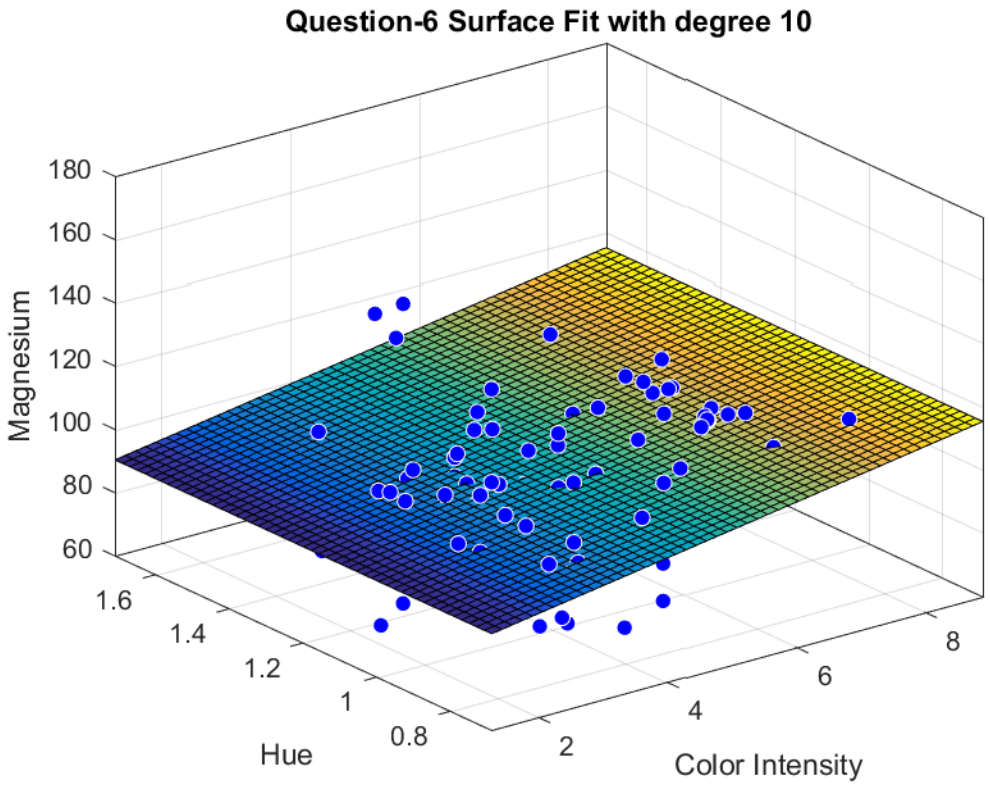
Question-5 (0.25 pts) Now, add a third feature of Hue to your data and plot the three in a 3D plot.

```
% Question-5 (0.25 pts)
% Now, add a third feature of Hue to your data and plot the three in a 3D
plot.
hue=wine(:,11);
A = [magnesium,color_intensity,hue];
x = A(:,2) ; y = A(:,3) ; z = A(:,1) ;

[xq,yq] = meshgrid(0:.2:5, 0:.2:5);   % change these values to suitable ones
for your problem. Maybe use min and max on the x and y data etc.
zq = griddata(x,y,z,xq,yq);
figure; surf(xq,yq,zq);
xlabel('Color Intensity')
ylabel('Hue')
zlabel('Magnesium')
title('Question-5 3D Plot of Manesium/Color Intensity/Hue')
```



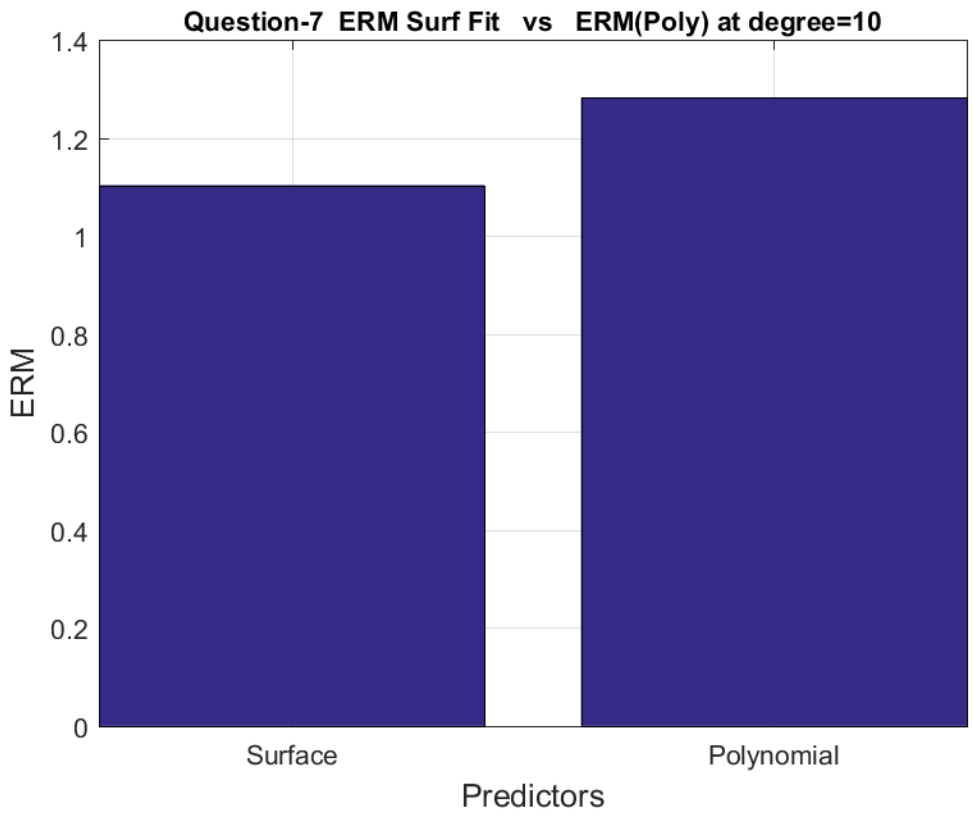
Question-6 (1 pts) For your three selected features, fit a surface to your data of a degree 10.

```
% Question-6 (1 pts)
% For your three selected features, fit a surface to your data of a degree
10.
figure()
surffit = fit( [x, y], z, 'poly10','normalize','on' );
plot(surffit, [x,y], z)
xlabel('Color Intensity')
ylabel('Hue')
zlabel('Magnesium')
title('Question-6 Surface Fit with degree 10')
xi = [5;10;7];
yi = [0.7;0.6;0.5];
```



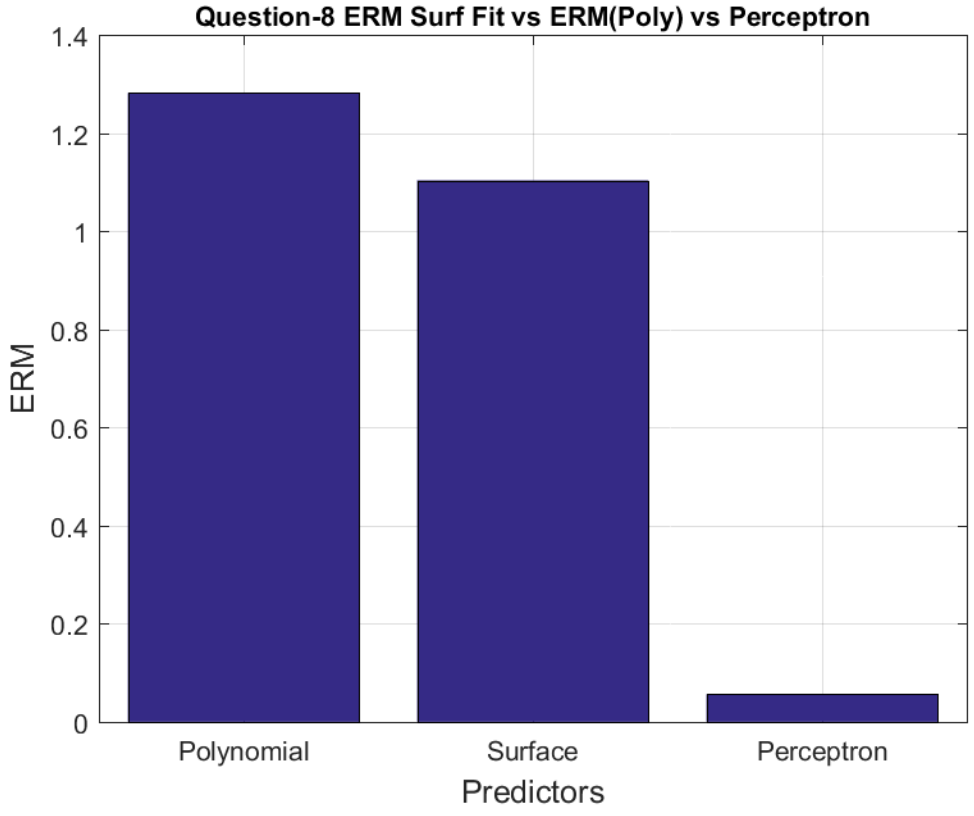
Question-7 (0.5 pts) Compare the ERM loss of your surface (question 6) and line (question 3) predictors.

```
% Question-7 (0.5 pts)
% Compare the ERM loss of your surface (question 6) and line (question 3)
predictors.
se=surffit(xi,yi);
RMSE_surf=sqrt(mean(se.^2))/100;
rmse_comp=[RMSE_surf,RMSE(10)];
figure()
h=bar(rmse_comp);
xlabel('Predictors','FontSize', 12)
ylabel('ERM','FontSize', 12)
title('Question-7  ERM Surf Fit   vs   ERM(Poly) at degree=10', 'FontSize',
10);
l{1}='Surface'; l{2}='Polynomial';
set(gca,'xticklabel', l)
grid on
```



Question-8 (1 bonus pts) Fit the data with a Perceptron classifier and compare the loss with respect to your fitted lines (question-3)

```
% Question-8 (1 bonus pts)
% Fit the data with a Perceptron classifier and compare the loss with respect
to your fitted lines
% (question-3)

magnesium=double(magnesium);
color_intensity=double(color_intensity);
hue=double(hue);

mag_norm=magnesium / max(magnesium);
color_intensity=color_intensity / max(color_intensity);
hue_norm=hue / max(hue);

input = [mag_norm, color_intensity, hue_norm];

numIn = 130;
desired_out = wine_label/max(wine_label);
bias = -1;
coeff = 0.01;
rand('state',sum(100*clock));
weights = -1*2.*rand(4,1);
iterations = 1000;
for i = 1:iterations
     out = zeros(4,1);
     for j = 1:numIn
          y = bias*weights(1,1)+...

input(j,1)*weights(2,1)+input(j,2)*weights(3,1)+input(j,3)*weights(4,1);
          out(j) = 1/(1+exp(-y));
          delta = desired_out(j)-out(j);
          weights(1,1) = weights(1,1)+coeff*bias*delta;
          weights(2,1) = weights(2,1)+coeff*input(j,1)*delta;
          weights(3,1) = weights(3,1)+coeff*input(j,2)*delta;
          weights(4,1) = weights(4,1)+coeff*input(j,3)*delta;
     end
end

rmse_comp=[RMSE(10),RMSE_surf,delta];
figure()
h=bar(rmse_comp);
xlabel('Predictors','FontSize', 12)
ylabel('ERM','FontSize', 12)
title('Question-8 ERM Surf Fit vs ERM(Poly) vs Perceptron', 'FontSize', 10);
l{2}='Surface'; l{1}='Polynomial';   l{3}='Perceptron';
set(gca,'xticklabel', l)
grid on
```


Question-8 ERM Surf Fit vs ERM(Poly) vs Perceptron
ERM
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Polynomial
Surface
Perceptron
Predictors