Contents

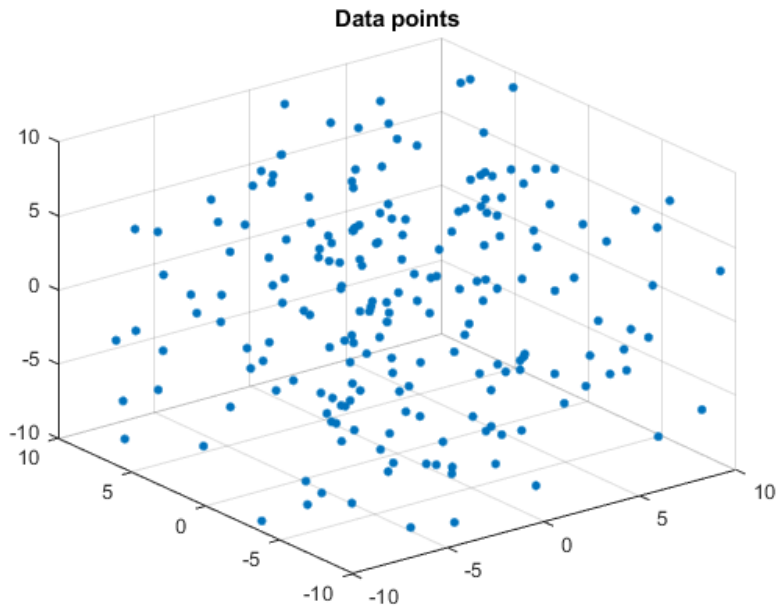

```
clc
clear all
close all
```

Part 1 (Data Generation)

Data generation

```
%Inputs
x = -10 + 20*rand(200,3);
%Weights
w = [-0.8; 2.1; 1.5];
%Noise
vr = 10; % variance of the data
mn = 0; % mean of the data
sd = sqrt(vr);
n = 200; % number of points to be generated
noise = sd*randn(200,1) + mn;
%Outputs
y = x*w + 10 + noise;
figure('Name','Generated Data')
scatter3(x(:,1),x(:,2),x(:,3),200,'.')
title('Data points')

save('data.mat','x','y')
clear all
load('data.mat')
```



Part 2 (Ridge Regression)

Standardizing inputs

```
xtrain = (x(1:100,:) - mean(x(1:100,:)))./std(x(1:100,:));
xtest = (x(101:end,:) - mean(x(1:100,:)))./std(x(1:100,:));
```

Normalizing outputs

```
ytrain = y(1:100,:) - mean(y(1:100,:));
ytest = y(101:end,:) - mean(y(1:100,:));
```

Determining wo (bias term 'b' for ridge)

```
wor = mean(y(1:100,1))
```

```
wor =

    9.8933
```

Learning weights

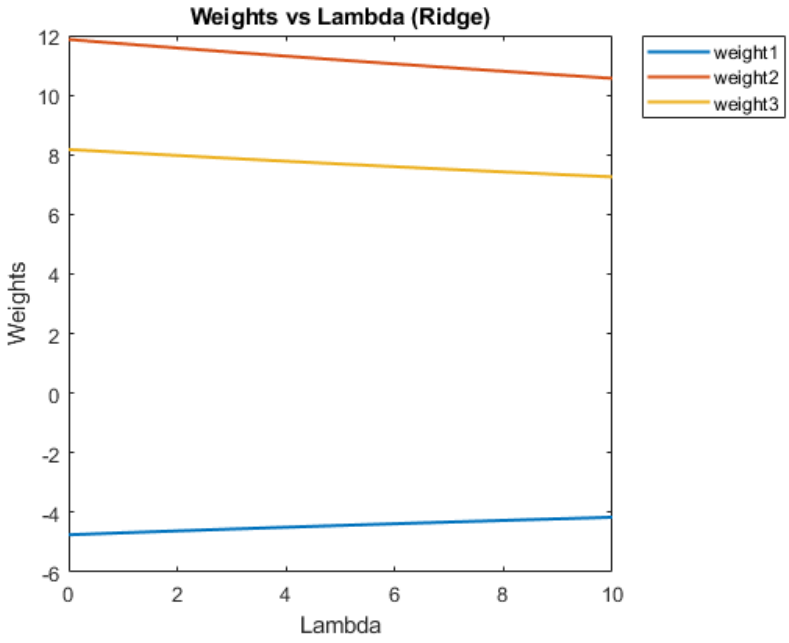
```
w = [0 0 0];
for i = 1:101
    lambda = 0.1*(i-1);
    a = ((xtrain'*xtrain + lambda * eye(max(size(xtrain'*xtrain)))) \ (xtrain'*ytrain))';
    w = [w; a];
end
wr = w(2:end,:);
```

SSE

```
xtest = [ones(max(size(xtrain)),1) xtest];
wts = [wor*ones(max(size(wr)),1) wr];
SSEr = zeros(max(size(wr)),1);
for i = 1:max(size(wr))
    SSEr(i,1) = sum((wts(i,:)*xtest' - ytest').^2);
end
```

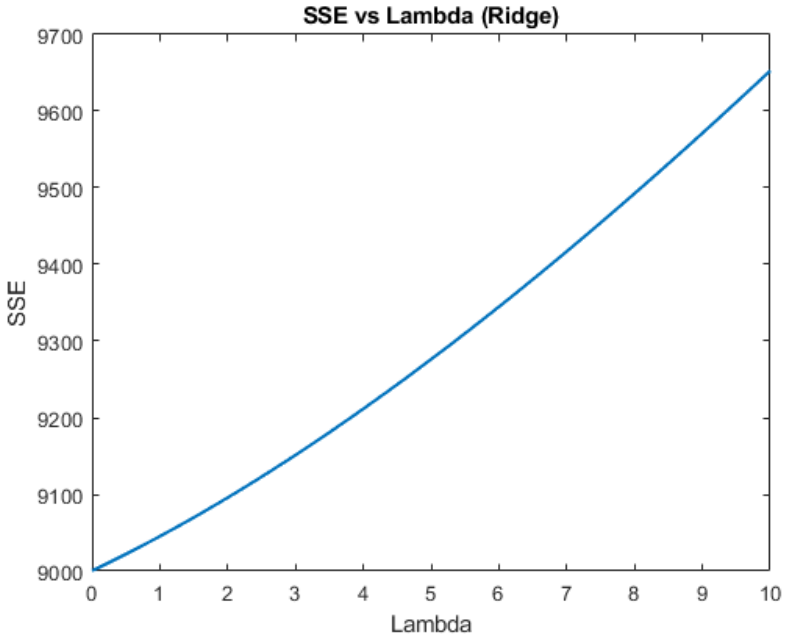
Plots for Weights vs Lambda (Ridge)

```
lambda = (0:0.1:10)';
figure('Name','Weights vs Lambda (Ridge)')
plot(lambda,wr(:,1),'-','LineWidth',1.5)
hold on
plot(lambda,wr(:,2),'-','LineWidth',1.5)
hold on
plot(lambda,wr(:,3),'-','LineWidth',1.5)
title('Weights vs Lambda (Ridge)'); xlabel('Lambda'); ylabel('Weights');
legend('weight1','weight2','weight3','Location','northeastoutside')
hold off
```



Plot for SSE vs Lambda (Ridge)

```
figure('Name','SSE vs Lambda (Ridge)')
plot(lambda,SSEr,'-','LineWidth',1.5)
title('SSE vs Lambda (Ridge)'); xlabel('Lambda'); ylabel('SSE');
```



Part 3 (Lasso Regression)

Standardizing inputs

```
xtrain = (x(1:100,:) - mean(x(1:100,:)))./std(x(1:100,:));
xtest = (x(101:end,:) - mean(x(1:100,:)))./std(x(1:100,:));
```

Normalizing outputs

```
ytrain = y(1:100,:) - mean(y(1:100,:));
ytest = y(101:end,:) - mean(y(1:100,:));
```

Determining wo (bias term 'b' for Lasso)

```
wol = mean(y(1:100,1))
```

```
wol =

    9.8933
```

Overall least square estimate

```
wp = xtrain\ytrain;
```

First value of GE

```
GE = sign(wp)';
```

Learning weights using TIBSHIRANI algorithm

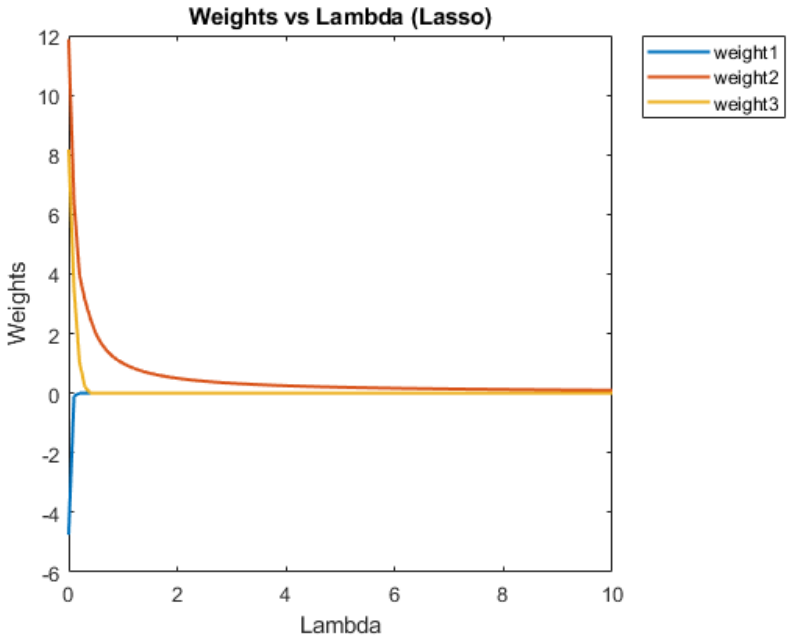
```
eps = 10^-4;
w = [0 0 0];
for i = 1:101
    lambda = 0.1*(i-1);
    t = 1/lambda;
        while sum(abs(wp)) > t+eps
        options = optimoptions(@quadprog,'Display','off');
        [wp] = quadprog(2*xtrain'*xtrain,-2*xtrain'*ytrain,GE,t*ones(size(GE,1),1),[],[],[],[],wp,options);
        GE = [GE;sign(wp)'];
        end
     w = [w; wp'];
end
wl = w(2:end,:);
```

SSE

```
wo = mean(y(1:100,1));
xtest = [ones(max(size(xtest)),1) xtest];
wts = [wo*ones(max(size(wts)),1) wl];
SSEl = zeros(max(size(wts)),1);
for i = 1:max(size(wts))
    SSEl(i,1) = sum((wts(i,:)*xtest' - ytest').^2);
end
```

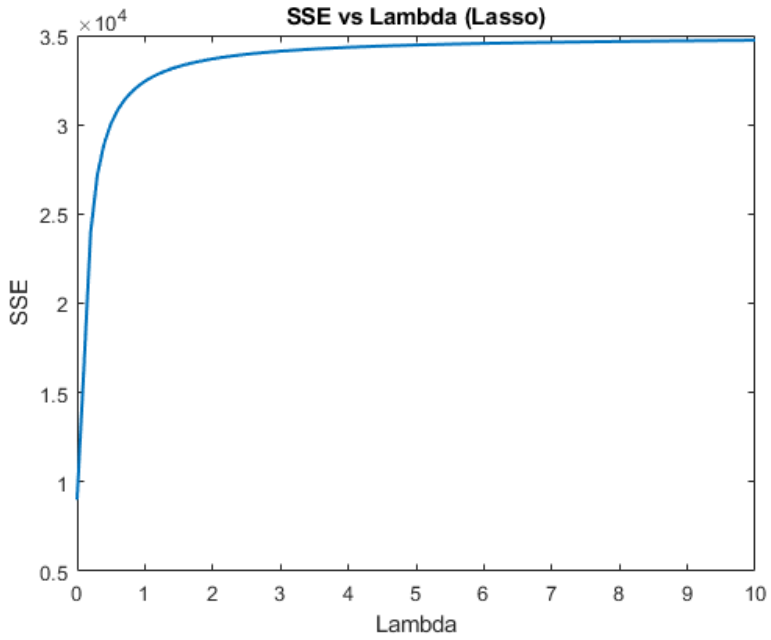
Plots for Weights vs Lambda (Lasso)

```
lambda = (0:0.1:10)';
figure('Name','Weights vs Lambda (Lasso)')
plot(lambda,wl(:,1),'-','LineWidth',1.5)
hold on
plot(lambda,wl(:,2),'-','LineWidth',1.5)
hold on
plot(lambda,wl(:,3),'-','LineWidth',1.5)
title('Weights vs Lambda (Lasso)'); xlabel('Lambda'); ylabel('Weights');
legend('weight1','weight2','weight3','Location','northeastoutside')
hold off
```



Plot for SSE vs Lambda (Lasso)

```
figure('Name','SSE vs Lambda (Lasso)')
plot(lambda,SSEl,'-','LineWidth',1.5)
title('SSE vs Lambda (Lasso)'); xlabel('Lambda'); ylabel('SSE');
```

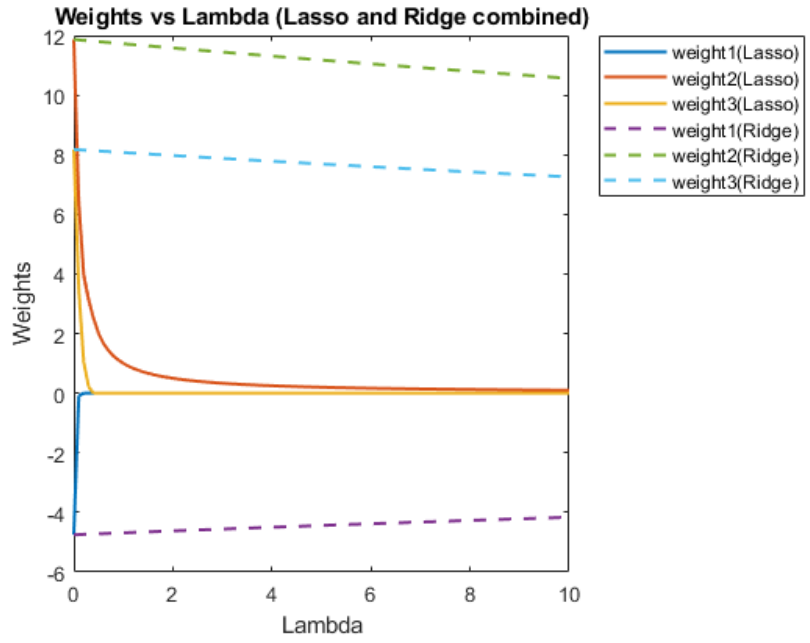


Part 4 (Miscellaneous plots for understanding purpose only)

Plot for Weights vs Lambda (Lasso and Ridge combined)

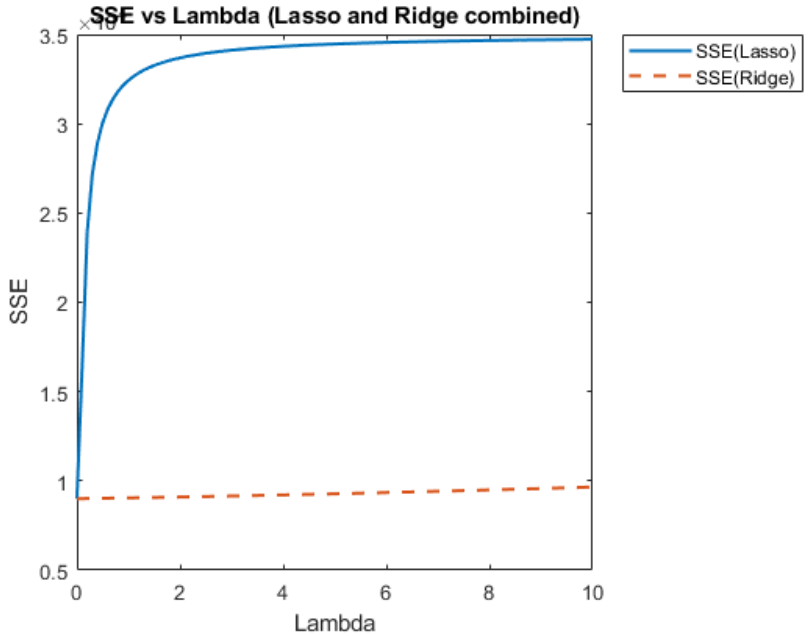
```
figure('Name','Plot for Weights vs Lambda (Lasso and Ridge combined)')
plot(lambda,wl(:,1),'-','LineWidth',1.5)
hold on
plot(lambda,wl(:,2),'-','LineWidth',1.5)
hold on
plot(lambda,wl(:,3),'-','LineWidth',1.5)
hold on
plot(lambda,wr(:,1),'--','LineWidth',1.5)
hold on
plot(lambda,wr(:,2),'--','LineWidth',1.5)
hold on
```

```
plot(lambda,wr(:,3),'--','LineWidth',1.5)
title('Weights vs Lambda (Lasso and Ridge combined)'); xlabel('Lambda'); ylabel('Weights');
legend('weight1(Lasso)','weight2(Lasso)','weight3(Lasso)','weight1(Ridge)','weight2(Ridge)','weight3(Ridge)','Location','northeastoutside')
hold off
```



Plot for SSE vs Lambda(Lasso and Ridge combined)

```
figure('Name','SSE vs Lambda (Lasso and Ridge combined)')
plot(lambda,SSEl,'-','LineWidth',1.5)
hold on
plot(lambda,SSEr,'--','LineWidth',1.5)
title('SSE vs Lambda (Lasso and Ridge combined)'); xlabel('Lambda'); ylabel('SSE');
legend('SSE(Lasso)','SSE(Ridge)','Location','northeastoutside')
hold off
```



Published with MATLAB® R2018a