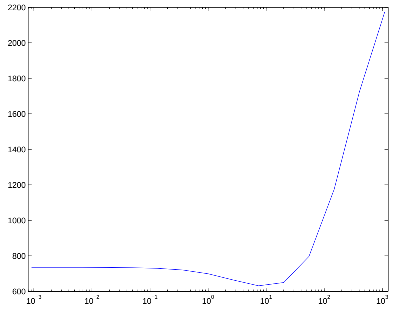
Problem Set 2
Solutions

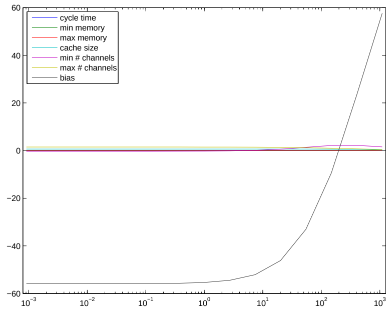
```
% q2 script
load -ascii machine.ascii
x = machine(:,1:(end-1));
y = machine(:,end);
[n,d] = size(x);
x = [x ones(n,1)];
nsplit = 20;

l = exp(-7:1:7);
loss = zeros(size(l));
for j=1:nsplit
    mixup = randperm(n);
    x = x(mixup,:);
    y = y(mixup);
    nn = floor(n*0.2);
    trainx = x(1:nn,:);
    trainy = y(1:nn);
    testx = x((nn+1):end,:);
    testy = y((nn+1):end);
    i = 1;
    for ll = l
        w = myridge(trainx,trainy,ll);
        myy = testx*w;
        err = myy-testy;
        loss(i) = loss(i) + mean(err.*err);
        i = i+1;
    end;
end;
loss = loss/n;

figure
semilogx(l,loss);
figure
ridgeplot(x,y);
```

```
function w = myridge(X,y,lambda)
% w = myridge(X,y,lambda)
% computes ridge regression coefficients
% assumes that last column of X is all 1s

% perform z-normalization
[n,d] = size(X);
mu = mean(X(:,1:d-1)); sigma = std(X(:,1:d-1));
X(:,1:d-1) = (X(:,1:d-1)-repmat(mu,n,1))...
                ./repmat(sigma,n,1);

reg = eye(size(X,2))*lambda;
reg(end,end) = 0.0;
w = (X'*X+reg)\(X'*y);

% transform weights for non-z-normalized data
w(d) = w(d) - w(1:d-1)'*(mu./sigma)';
w(1:d-1) = w(1:d-1)./sigma';
```

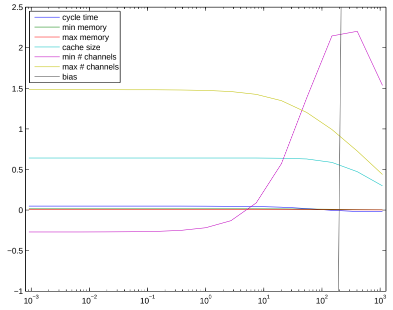
```
function ridgeplot(x,y)

l = exp(-7:1:7);
ws = zeros(size(x,2),length(l));
i = 1;
for ll = l
    ws(:,i) = myridge(x,y,ll);
    i = i+1;
end;
semilogx(repmat(l,[size(ws,1) 1])',ws');
```



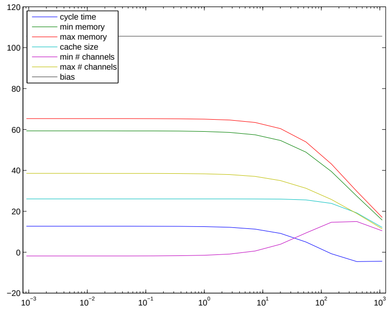
Mean squared error verses lambda



Weights verses lambda



Weights verses lambda (zoomed)



Weights verses lambda if testing data is also z-normalized