function ungradedRandomPlotting(X, y, Xval, yval)	
	% we'll check the cross val
	m = size(X, 1);
	% vectors for storing traing
	error_train = zeros(m,1);
	error_val = zeros(m,1);
	% the lambda
	lambda = 0.01;
	% number of times to loop fo
	loops = 20;
	% pick a random value "loops
	for l=1:loops
	% Concretely, to det
	% i examples, (:1a:)
	% (:1b:) and i examp
	% eters theta using
	% theta on the rando
	% i = number of trai
	for i=1:m
	% ---
	% test set
	% ---
	% (:1a:) ran
randperm(N)生成一个1到N的无重复整数的随机排列	sel = randpe
取前n个	sel = sel(1:
	% create a r

```
        y_sel = y(se

        % (:2:) lear
        theta = trai

        % (:3a:) eva
        [J, grad] =

        % accumulate
        error_train(

        % ---
        % cross vali
        % ---
        % (:1b:) ...
        sel = randpe
        sel = sel(1:
        X_sel = Xval
        y_sel = yval
        % (:3b:) ...
        [J, grad_val

        error_val(i)

    end
end

% finding the average
error_train = error_train ./
error_val = error_val ./ loc

% least but not last, do som
plot(1:m, error_train, 1:m,
xlabel('Number of training e
ylabel('Error');
```

	`axis([0 13 0 100]);`
	`legend('Train', 'Cross Vali`
	`end`

Learning Curve学习曲线

```
function [error_train, error_val] = learningCurve(X, y, Xval, yval, lambda)

% Number of training examples
m = size(X, 1);

% You need to return these values correctly
error_train = zeros(m, 1);
error_val   = zeros(m, 1);

% ---------------------- Sample Solution ----------------------
for i = 1:m,
  theta = trainLinearReg(X(1:i,:), y(1:i,:), lambda);
  error_train(i) = linearRegCostFunction(X(1:i,:), y(1:i,:), theta, 0);
  error_val(i) = linearRegCostFunction(Xval, yval, theta, 0);
end
```

Validation Curve验证曲线

```
function [lambda_vec, error_train, error_val] = validationCurve(X, y, Xval, yval)

% Selected values of lambda (you should not change this)
lambda_vec = [0 0.001 0.003 0.01 0.03 0.1 0.3 1 3 10]';

% You need to return these variables correctly.
error_train = zeros(length(lambda_vec), 1);
error_val = zeros(length(lambda_vec), 1);

for i = 1:length(lambda_vec),
  lambda = lambda_vec(i);
  theta = trainLinearReg(X, y, lambda);
  error_train(i) = linearRegCostFunction(X, y, theta, 0);
  error_val(i) = linearRegCostFunction(Xval, yval, theta, 0);
end
```