1. Linear Regression (75 points)

We will find coefficients A to estimate $X * A \approx Y$, using the provided datasets X and Y. We will compare two approaches least squares and ridge regression.
Least Squares: Set A = inverse(X' * X)*X'*Y
Ridge Regression: Set As = inverse(X'*X + s^2*eye(15))*X'*Y

A(30 points) Solve for the coefficients A (or As) using Least Squares and Ridge Regression with s = 1, 5, 10, 15, 20, 25, 30 (i.e. s will take on one of those 7 values each time you try, say obtaining A05 for s = 5). For each set of coefficients, report the error in the estimate $\hat{Y}$ of Y as norm(Y - X*A,2).

Error A = 605.6547
Error AS 1 = 605.6549
Error AS 5 = 605.7708
Error AS 10 = 607.1568
Error AS 15 = 611.3054
Error AS 20 = 618.4469
Error AS 25 = 627.9439
Error AS 30 = 639.1680

B (30 points): Create three row-subsets of X and Y

X1 = X(1:66,:) and Y1 = Y(1:66)
Error A = 483.0236
Error AS 1= 482.7316
Error AS 5 = 476.4220
Error AS 10 = 462.9247
Error AS 15 = 451.7693
Error AS 20 = 446.5091
Error AS 25 = 446.8578
Error AS 30= 451.7081

X2 = X(34:100,:) and Y2 = Y(34:100)
Error A = 434.7992
Error AS 1 = 434.5322
Error AS 5 = 428.8565
Error AS 10 = 417.0631
Error AS 15 = 406.8163
Error AS 20 = 400.1875
Error AS 25 = 397.0574
Error AS 30 = 397.1167

X3 = [X(1:33,:); X(67:100,:)] and Y3 = [Y(1:33); Y(67:100)]
Error A = 383.0319
Error AS 1 = 382.7135
Error AS 5 = 376.3331

Error AS 10 = 365.6099
Error AS 15 = 359.6731
Error AS 20 = 358.5897
Error AS 25 = 360.6872
Error AS 30 = 365.0511

C (15 points): Which approach works best (averaging the results from the three subsets): Least Squares, or for which value of s using Ridge Regression?

Error A = 433.6182
Error AS 1 = 433.3258
Error AS 5 = 427.2039
Error AS 10 = 415.1992
Error AS 15 = 406.0862
Error AS 20 = 401.7621
Error AS 25 = 401.5342
Error AS 30 = 404.6253

Ridge Regression with s = 25 seems to be the best because the lowest value out of the average is Error AS 25 = 401.5342. The lowest error is the best

2. Orthogonal Matching Pursuit (25 points)

Consider a linear equation W = M*S where M is a measurement matrix filled with random values (although now that they are there, they are no longer random), and W is the output of the sparse signal S when measured by M. Use Orthogonal Matching Pursuit (as described in the notes as Algorithm 18.2.1) to recover the non-zero entries from S. Record the order in which you find each entry and the residual vector after each step.

The non-zero entries are given by the given indexes below and the residual vectors after each step are also given below.

r = [-3 0 0 -3 5 1 1 2 0 -2 -2 -2 -4 1 0 2 1 0 3 -1 -2 -1 -2 1 -2 -2 2 -1 0 3 4 2 1 -1 1 1 -4 2 0 1 0 -1 -1 -3 1 -1 1 2 -2 2 2 0 -1 -2 0 0 -1 -3 0 2 -2 0 -2 1 -1 0 -4 -3 -1 0 3 0 0 0 4 1 3 -3 -1 -3

index = 43

r = [-2 1 1 -2 4 1 2 1 1 -2 -1 -2 -3 2 1 1 0 0 3 -1 -1 0 -1 1 -2 -1 3 -1 0 2 3 1 1 0 0 0 -3 2 1 0 0 0 0 -2 0 0 1 2 -2 1 1 -1 -1 -2 -1 -1 0 -2 -1 2 -2 0 -1 0 -2 0 -3 -2 -2 0 2 0 1 -1 3 1 3 -3 -1 -2]

index = 66

r = [-1 0 1 -1 3 1 1 1 1 -2 -2 -1 -2 1 2 1 -1 0 2 0 -1 0 0 0 -2 -1 3 -2 -1 2 2 1 0 1 0 0 -2 1 0 -1 -1 0 -1 -1 1 1 0 2 -1 0 0 -1 0 -1 0 -1 1 -1 0 2 -1 -1 0 1 -1 -1 -2 -2 -1 0

2 0 0 0 2 1 2 -2 -2 -1]

index =16

r = [0 -1 0 -1 2 2 1 1 1 -1 -1 0 -2 1 1 0 0 0 1 0 0 1 -1 -1 -1 -1 2 -2 -1 1 2 1 1 1 -1 0 -1 1 -1 0 -1 -1 0 -2 1 0 0 1 0 0 0 0 1 0 -1 0 1 -1 1 1 -1 -1 1 1 0 -1 -1 -1 0 -1 1 0 0 0 2 2 2 -1 -1 -1]

index = 65

r = [0 -1 1 -1 1 1 0 1 1 0 0 0 -1 1 0 1 0 0 1 1 -1 1 0 0 0 -1 1 -1 0 1 1 1 0 1 0 0 -1 0 -1 0 0 -1 1 -1 0 1 0 0 -1 1 -1 0 0 -1 0 -1 0 0 1 0 -1 0 0 1 1 0 -1 0 -1 -1 0 0 1 0 1 1 1 -1 0 -1]

index = 89

r = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

The Assignment was done in Matlab where index start at 1 instead of 0. The elements at index 43, 66, 16, 65, and 89 are the non-zero entries from S.