```
% Import data
data_HW3 = readtable("Top 100 Genes & Rand 15 Patients.xlsx")
```

```
data_HW3 = 17×101 table
```

...

	A_23_P342744	A_24_P246891	A_23_P24535	A_23_P75362	A_32_P76399	A_23_P102973
1	'LIX1L'	'NEU4'	'TTC12'	'IFITM10'	'EIF3L'	'DGCR14'
2	'Lix1 homolog...	'sialidase 4'	'tetratricope...	'interferon i...	'eukaryotic t...	'DiGeorge syn...
3	'-0.90039'	'-1.3555'	'0.94774'	'-1.937'	'-0.75801'	'-0.018503'
4	'1.4318'	'-1.7487'	'-1.1996'	'-0.65274'	'-0.18371'	'-0.18308'
5	'0.0033245'	'-1.131'	'0.97946'	'-0.68775'	'-1.0177'	'-0.54962'
6	'-0.093005'	'-1.7834'	'0.92898'	'-2.2042'	'-0.49095'	'-0.5776'
7	'0.20914'	'-2.0248'	'1.5423'	'-0.72764'	'-0.65224'	'-1.0807'
8	'-0.35701'	'-1.5359'	'1.174'	'-0.33638'	'-0.20179'	'-0.33608'
9	'0.15583'	'-0.2542'	'0.97841'	'-0.4683'	'-0.29401'	'-0.52826'
10	'0.8064'	'-1.4967'	'0.81275'	'1.158'	'-0.37563'	'0.13596'
11	'1.1037'	'0.74543'	'-2.0402'	'2.3586'	'1.5234'	'1.1066'
12	'-0.34433'	'-0.56852'	'0.58211'	'-1.2632'	'-1.3063'	'-0.023503'
13	'-0.12072'	'-0.14478'	'1.5174'	'0.4458'	'-0.52952'	'-0.75984'
14	'0.38907'	'1.7826'	'-1.4738'	'0.56259'	'0.1603'	'-0.19535'
15	'0.70957'	'-2.2007'	'-0.82342'	'0.013375'	'1.5027'	'1.2178'
16	'-0.090373'	'-1.2046'	'1.12'	'-0.55248'	'-0.40187'	'0.36356'
17	'0.1623'	'0.06178'	'-1.0095'	'-0.081773'	'-0.30487'	'0.17538'

```
% Using readmatrix correctly loads data as class "double"
data_HW3_2 = readmatrix("Top 100 Genes & Rand 15 Patients.xlsx");
```

```
% Create new X and Y matrices (have to convert class if using readtable)
X = str2double(data_HW3{3:end,1:100});
Y = table2array(data_HW3(3:end,end));
```

Hold Out Validation Model

Using Top 8 Correlated Genes from Training Set

```
% Create X and Y training/test sets from 100 genes
[Xtrain, Ytrain, Xtest, Ytest] = trainTestSplit(X, Y, 0.7);

% Find correlation of all 100 genes and then extract data for top 8 genes
r_100 = corr(Xtrain,Ytrain);
[r_8,index_8] = maxk(abs(r_100),8);
```

```
% Create new training and test predictor data sets with top 8 genes
Xtrain_8 = Xtrain(:,index_8);
Xtest_8 = Xtest(:,index_8);

mdl = fitlm(Xtrain_8,Ytrain)
```

```
mdl =
Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8

Estimated Coefficients:
                   Estimate      SE       tStat      pValue
                   ________    ______    ________    _______

    (Intercept)      636.6     80.007      7.9568    0.01543
    x1               77.314    107.91      0.7165    0.54805
    x2               549.59    412.43      1.3326    0.31422
    x3              -346.73    261.38     -1.3265    0.31587
    x4              -126.57    127.78    -0.99052    0.42631
    x5              -53.78     348.61    -0.15427    0.89156
    x6               33.346    143.95     0.23165    0.83835
    x7               70.303    92.535     0.75974    0.52675
    x8              -171.12    165.26     -1.0355    0.40924


Number of observations: 11, Error degrees of freedom: 2
Root Mean Squared Error: 151
R-squared: 0.956,  Adjusted R-Squared: 0.778
F-statistic vs. constant model: 5.39, p-value = 0.166
```

```
Ypred_norm = predict(mdl,Xtest_8);
r_norm = corr(Ytest,Ypred_norm)
```

```
r_norm = 0.9113
```

```
r2_norm = r_norm^2
```

```
r2_norm = 0.8305
```

```
RMSE = sqrt(mean((Ypred_norm-Ytest).^2))
```

```
RMSE = 249.6432
```

```
avg_error = mean(abs(Ypred_norm-Ytest))
```

```
avg_error = 221.6253
```

Lasso Regression

```
[B1, Fit] = lasso(Xtrain_8,Ytrain,"CV",10);
B1_coeff = B1(:,Fit.Index1SE)
```

```
B1_coeff = 8×1
   10.1998
  179.3113
         0
  -50.6444
         0
    9.1231
```

```
   30.9232
         0
```

```
B1_intercept = Fit.Intercept(Fit.Index1SE)
```

```
B1_intercept = 523.5057
```

```
Ypred_lasso = Xtest_8 * B1_coeff + B1_intercept;

r_lasso = corr(Ypred_lasso,Ytest)
```

```
r_lasso = 0.9684
```

```
r2_lasso =  r_lasso^2
```

```
r2_lasso = 0.9378
```

```
RMSE_lasso = sqrt(mean((Ypred_lasso-Ytest).^2))
```

```
RMSE_lasso = 242.8598
```

```
avg_error_lasso = mean(abs(Ypred_lasso-Ytest))
```

```
avg_error_lasso = 208.5119
```

Stepwise Regression

```
[B2,~,~,~,stats] = stepwisefit(Xtrain_8,Ytrain);
```

```
Initial columns included: none
Step 1, added column 1, p=0.0106827
Step 2, added column 5, p=0.0178958
Final columns included:  1 5
    'Coeff'        'Std.Err.'    'Status'    'P'
    [ 151.7532]    [ 48.3244]    'In'        [0.0138]
    [  -6.5935]    [351.9508]    'Out'       [0.9856]
    [-131.9476]    [224.4169]    'Out'       [0.5750]
    [ -78.7831]    [ 54.3473]    'Out'       [0.1904]
    [-403.7328]    [135.9782]    'In'        [0.0179]
    [ 134.6150]    [ 62.0593]    'Out'       [0.0667]
    [  54.9106]    [ 64.3412]    'Out'       [0.4217]
    [ -75.2122]    [140.4751]    'Out'       [0.6089]
```

```
Ypred_step = Xtest_8*B2 + stats.intercept;
r_stepwise = corr(Ypred_step,Ytest)
```

```
r_stepwise = 0.8694
```

```
r2_stepwise =  r_stepwise^2
```

```
r2_stepwise = 0.7558
```

```
RMSE_stepwise = sqrt(mean((Ypred_step-Ytest).^2))
```

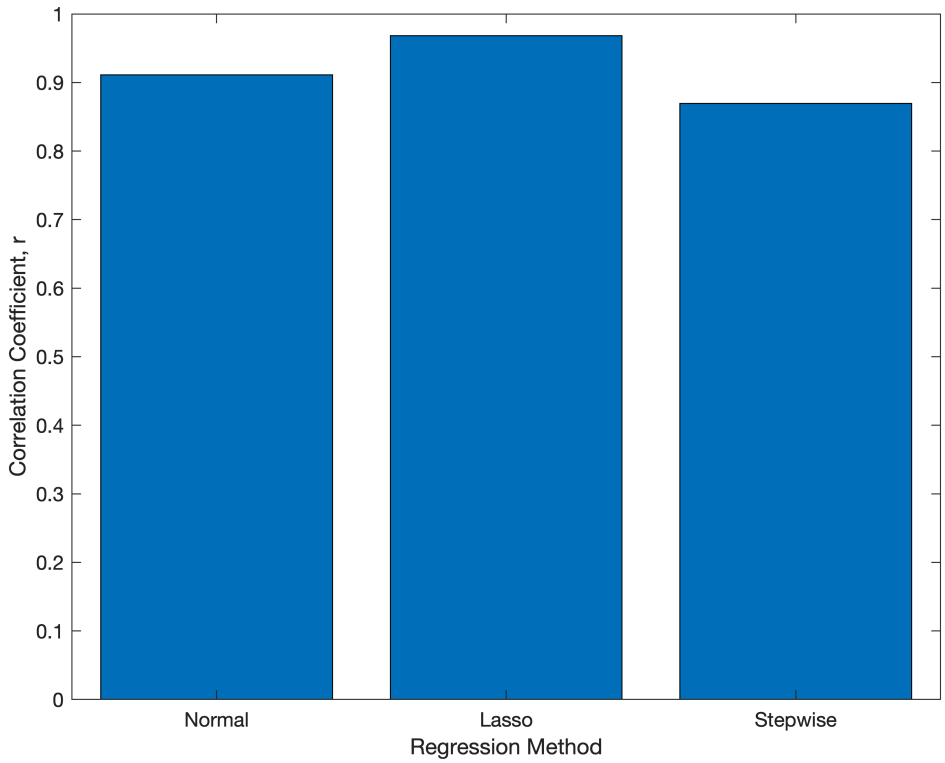
```
RMSE_stepwise = 401.9045
```

```
avg_error_stepwise = mean(abs(Ypred_step-Ytest))
```

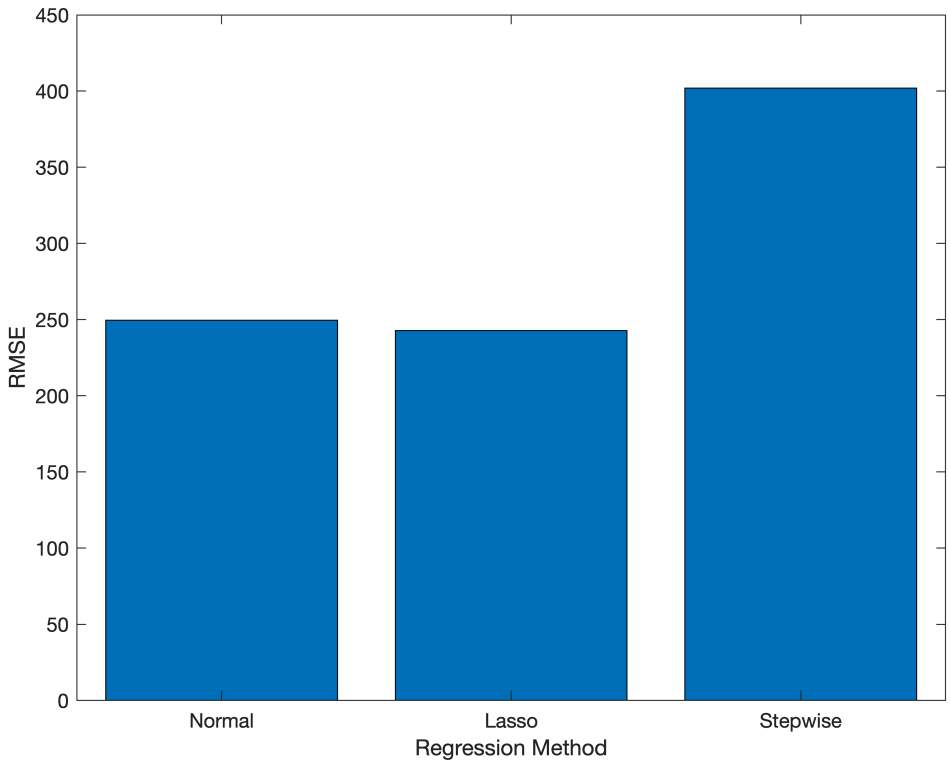
```
avg_error_stepwise = 312.0584
```

Compare Results of 3 Methods

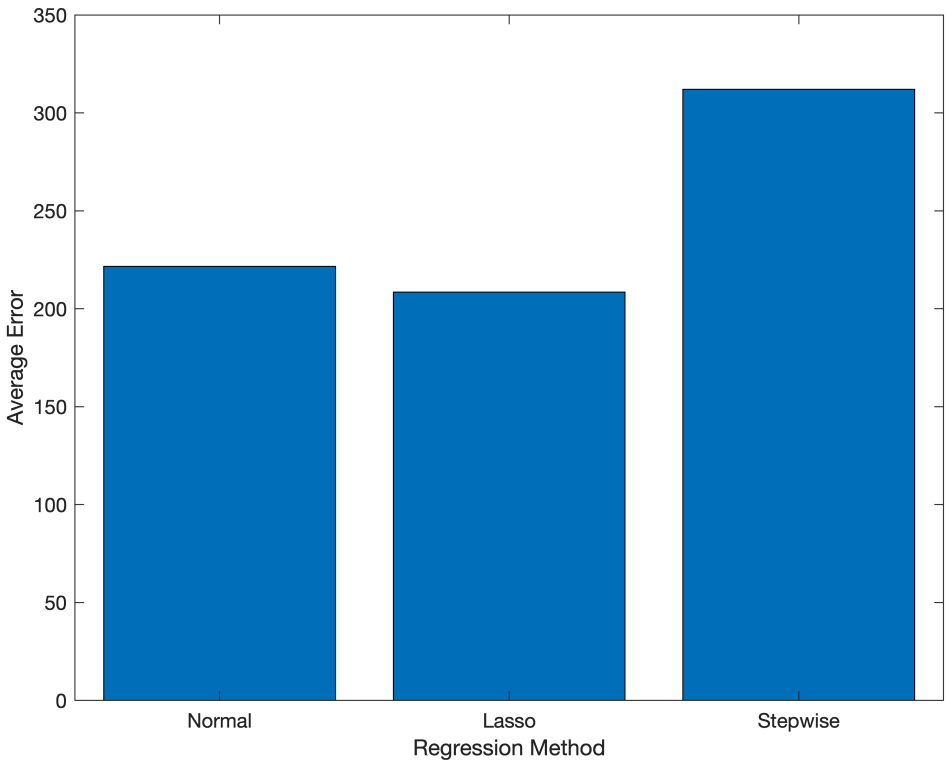
```
% Create labels for bar graphs
x = categorical({'Normal','Lasso','Stepwise'});
x = reordercats(x,{'Normal','Lasso','Stepwise'});

% Correlation bar graph
bar(x,[r_norm,r_lasso,r_stepwise])
xlabel("Regression Method")
ylabel("Correlation Coefficient, r")
```



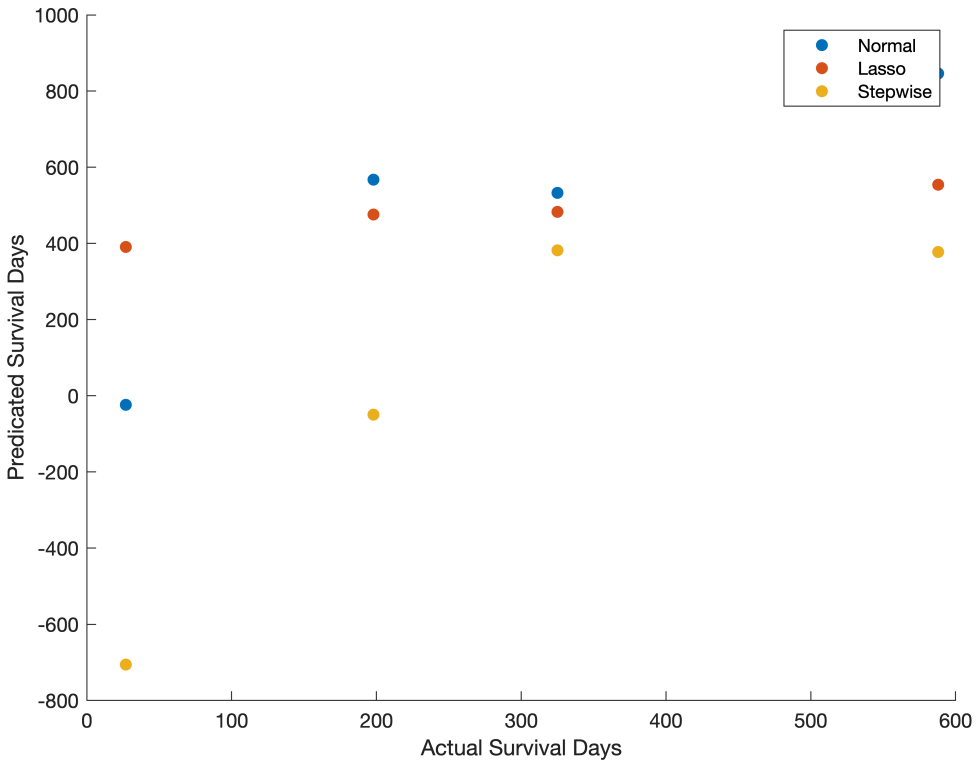

```
% RMSE bar graph
bar(x,[RMSE,RMSE_lasso,RMSE_stepwise])
xlabel("Regression Method")
ylabel("RMSE")
```




```
% Avg Error bar graph
bar(x,[avg_error,avg_error_lasso,avg_error_stepwise])
xlabel("Regression Method")
ylabel("Average Error")
```




```
scatter(Ytest,Ypred_norm,"filled")
xlabel("Actual Survival Days")
ylabel("Predicated Survival Days")
hold on
scatter(Ytest,Ypred_lasso,"filled")
hold on
scatter(Ytest,Ypred_step,"filled")
hold off
legend("Normal","Lasso","Stepwise")
```


Normal
Lasso
Stepwise
Predicated Survival Days
Actual Survival Days
1000
800
600
400
200
0
-200
-400
-600
-800
0
100
200
300
400
500
600