Grading scheme: (feel free to give half point instead of full point if the answer is partial or partially correct)
Problem 1: total of 7 points
Problem 2: total of 5 points

Total number of points 12.

EOAS 510/448 Homework#1

Problem (1)

Code in Matlab: -> students do not need to submit the Matlab codes (though some did)

```
% loading the data corr_data.mat

load 'corr_data.mat';

X=[x x2 x3];
Y=[y y2 y3];

for i=1:3
% Pearson correlation
corr_Pearson(i)=corr(X(:,i),Y(:,i));

% Spearman correlation
corr_Spearman(i)=corr(X(:,i),Y(:,i),'type','Spearman');
end

figure;
subplot(1,3,1)
p = polyfit(x,y,1);
f = polyval(p,x);
plot(x,y,'o',x,f,'r-');
xlabel('x');
ylabel('y');
legend('data','linear fit');

subplot(1,3,2)
p2 = polyfit(x2,y2,1);
f2 = polyval(p2,x2);
plot(x2,y2,'o',x2,f2,'r-');
xlabel('x2');
ylabel('y2');
legend('data','linear fit');

subplot(1,3,3)
p3 = polyfit(x3,y3,1);
f3 = polyval(p3,x3);
plot(x3,y3,'o',x3,f3,'r-');
xlabel('x3');
ylabel('y3');
legend('data','linear fit');
```

Results in Matlab:
>> corr_Pearson

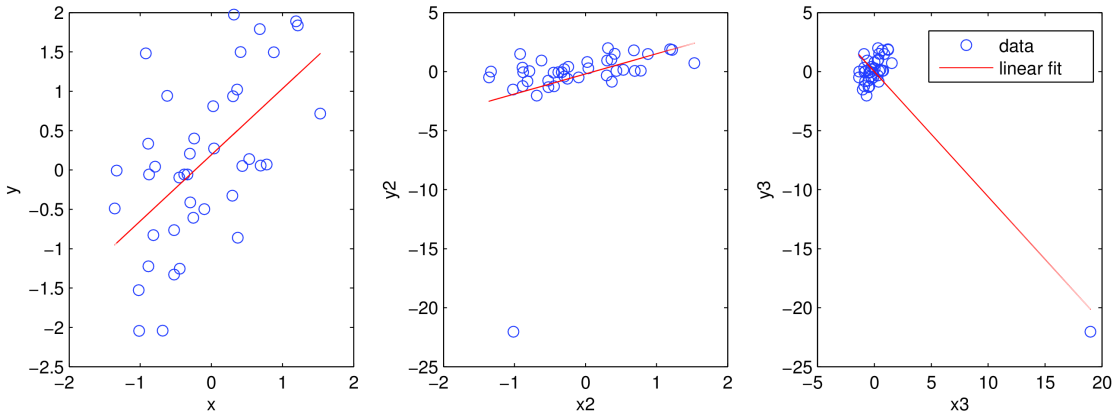
corr_Pearson =

0.5801 0.3397 -0.9010 -> 1 point for showing correct Pearson correlations

>> corr_Spearman

corr_Spearman =

0.5724 0.5724 0.4319 -> 1 point for showing correct Spearman correlations



-> 1 point for each plot plotted correctly (data and regression line), so 3 points in total here

Discussion:
Clearly Spearman correlation is much more resistant to outliers than Pearson correlation. When the data points are plotted in x-y space (figure above), the outliers in case 2 and case 3 are easily seen.
Why was the Spearman correlation unchanged from case 1 (x,y) to case 2 (x2,y2), where the 5th data point in y was shifted downward drastically? The reason is that this data point is already the lowest value in case 1, so there is no change in the ranking of the y data when we turn this data point into an outlier in case 2.

-> 1 point for have pointing out that Spearman correlation is more resistant to outliers than Pearson
-> 1 point for expaining why the Spearman correlation did not change from case 1 (x,y) to case 2 (x2,y2).

Problem (2)

Code in Matlab:

```
load 'MLR.mat';
n=size(y,1);
xall=[x1 x2 x3 x4 x5 x6];

%regression with original variables
X=[ones(n,1) xall];   % don't forget the column of "1"
b=regress(y,X);
y_regr=X*b;

% stepwise regression with original values
[a SE PVAL INMODEL STATS NEXTSTEP HISTORY]=stepwisefit(xall,y);
```

Results in Matlab:

Regression coefficients from multiple linear regression: -> 1 point for showing the results (coefficients)

b =
-820.3593
84.0383
0.3936
-3.3446

-6.6906
0.1828
0.0383

Stepwise: -> 1 point for showing these results (coefficients and which predictors are 'in' or 'out')

```
Initial columns included:  none
Step 1, added column 6, p=1.01879e-09
Step 2, added column 4, p=3.45675e-17
Step 3, added column 3, p=7.44399e-07
Step 4, added column 1, p=0.00532758
Final columns included:  1 3 4 6
    'Coeff'      'Std.Err.'    'Status'    'P'
    [81.6605]    [ 27.4803]    'In'        [    0.0053]
    [ 0.3782]    [  0.2692]    'Out'       [    0.1690]
    [-3.1518]    [  0.4527]    'In'        [4.2694e-08]
    [-6.6926]    [  0.2917]    'In'        [1.1253e-22]
    [ 0.1570]    [  0.2426]    'Out'       [    0.5218]
    [ 0.0402]    [  0.0039]    'In'        [4.5597e-12]

a =

   81.6605
    0.3782
   -3.1518
   -6.6926
    0.1570
    0.0402

>>a0=STATS.intercept  % to get the constant coefficient

a0 =

 -796.5485
```

Stepwise regression says only 4 predictors are significant.
The smaller the "P" values, the higher the significance. In order of significance, the 4 predictors are: x4, x6, x3 and x1. The other way would be to standardize the predictors and sort them according to the magnitude of regression coefficients (see below). -> 1 point for showing the correct order of significance (either by looking at p-values or regression coefficients in the standardized case)

Why are the regression coefficients obtained from regress.m different from those from stepwisefit.m? E.g. the regression coefficient for x1 was 84.0383 from regress.m and 81.6605 from stepwisefit.m. The reason is that the regression coefficient for x1 was obtained when only the significant predictors x1,x3,x4 and x6 were used in the regression. If we rerun regress.m using xall=[x1 x3 x4 x6], we will get the same regression coefficients as stepwisefit.m. -> 1 point if a difference between multiple regression and stepwise regression is discussed

---

Just from the results of multiple linear regression (regress.m above) it is hard to tell which predictor exerts more influence on y. If we check out the mean and standard deviation using mean(x1), std(x1), etc., we find that:

Mean of x1...x6:	10.0003	0.0750	-0.0718	0.0546	-0.1231	-5.5536
Std of x1 ...x6:	0.0055	0.5529	0.6301	0.5061	0.6233	72.1329

Since the range of values of these predictors are quite different it is important to standardize them in order to

investigate the importance each predictor has on y. So we standardize the predictors and repeat the above calculations.

Code in Matlab:

```
% standardizing the predictors (x1-mean(x1))/std(x1), the same done for x2 to x6

mean_xall=mean(xall);
std_xall=std(xall);

xall_mean=repmat(mean_xall,n,1);
xall_std=repmat(std_xall,n,1);

xall_standard=(xall-xall_mean)./xall_std;

%regression with standardized variables
X=[ones(n,1) xall_standard];   % don't forget the column of "1"
b2=regress(y,X);
y_regr=X*b2;

% stepwise regression with standardized values
[a2 SE PVAL INMODEL STATS NEXTSTEP HISTORY]=stepwisefit(xall_standard,y);
```

Results in Matlab:

Regression:

```
b2 =
   19.7222
    0.4586
    0.2176
   -2.1074
   -3.3863
    0.1139
    2.7623
```

Stepwise:

```
Initial columns included:  none
Step 1, added column 6, p=1.01879e-09
Step 2, added column 4, p=3.45675e-17
Step 3, added column 3, p=7.44399e-07
Step 4, added column 1, p=0.00532758
Final columns included:  1 3 4 6
    'Coeff'      'Std.Err.'    'Status'    'P'
    [ 0.4456]    [  0.1499]    'In'        [    0.0053]
    [ 0.2091]    [  0.1488]    'Out'       [    0.1690]
    [-1.9859]    [  0.2852]    'In'        [4.2694e-08]
    [-3.3873]    [  0.1477]    'In'        [1.1253e-22]
    [ 0.0979]    [  0.1512]    'Out'       [    0.5218]
    [ 2.8963]    [  0.2829]    'In'        [4.5597e-12]
```

```
>>a0=STATS.intercept

a0 =

   19.7222
```

Larger magnitude of regression coefficient (in this standardized case) indicates more importance of that

predictor. The “P” values and the magnitude of the regression coefficients tell us that in order of importance, the significant predictors are x4, x6, x3 and x1. -> 1 point if the standardization is performed