Problem 2

2(a) Simulate for Y and X given distribution of epsilon

```
R=5000; N=1000; beta=[1;-1]; sigma=1/2;
for r=1:R
    epsilon(:,r) = sigma*randn([N,1]);      %generate epsilon from normal distribution
    X(:,:,r) = [ones(N,1), randn(N,1)];     %generate X where X1 = 1 and X2 is from normdist
    Y(:,r) = X(:,:,r)*beta+epsilon(:,r);    %calculate Y
end

Ymeanr = mean(Y,1);
Yvarr = nanvar(Y,1);
Y25r = prctile(Y,25);
Y75r = prctile(Y,75);
```

2(b) Calculate averages from simulation in 2(a)

Average of Mean

```
ymean = mean(Ymeanr)
```

```
ymean = 1.0003
```

Average of Variance

```
yvar = mean(Yvarr)
```

```
yvar = 1.2483
```

Average of 25th-percentile

```
y25 = mean(Y25r)
```

```
y25 = 0.2464
```

Average of 75th-percentile

```
y75 = mean(Y75r)
```

```
y75 = 1.7544
```

2(c) Write a function that calculate SSR

```
%fubc_SSR.m
%function[SSR] = func_SSR(beta,Y, X)
%    SSR = transpose(Y-X*beta)*(Y- X*beta)
%end
```

2(d) Minimize sum of square error

```
beta0 = [1,-1];      %assign initial value
```

use fminunc to find beta for each rth simulation

```
%for r=1:R
%[beta(:,r), fval(r)] = fminunc(@(beta) func_SSR(beta,Y(:,r), X(:,:,r)), beta0])
%end
```

however, there is an error returning beta, and thus, I use matrix multiplication to obtain beta instead

The following is the matrix multiplication function "OLS.m"

```
%function[beta] = OLS(Y,X)
%beta = inv(transpose(X)*X)*(transpose(X)*Y);
%end
```

To find beta for rth simulation

```
for r=1:R
    [beta(:,r)] = OLS(Y(:,r), X(:,:,r));    %Using the rth row of Y and the rth page of X
end
```

2(e) Report the average of beta across simulation

```
meanbeta = mean(beta,2)
```

```
meanbeta = 2×1
    0.9998
   -1.0000
```

The first row is beta0 which is close to 1 and the second row is beta which is exactly -1.