```
Homework 3, Part 2

%sets random seed to the same number everytime.
rng(1)

%generates a random matrix and makes it orthogonal with QR decom.
M = rand(100,100);

[Q,R] = qr(M);

% ensuring this is an orthogonal matrix
QtQ = Q*Q';
QQt = Q'*Q;

Qt = Q';
QInv = inv(Q);

%orthogonal matrices are matrices that when multipled by their transpose
% yield the identity matrix, and we can also see that Q' = inverse of Q

U = Q;

%generate a random matrix V and do the same thing
V = rand(100,100);

[V,R] = qr(V);

Sig1 = zeros(100);
Sig2 = zeros(100);
Sig3= zeros(100);

%sets values for sigma matrices as detailed in HW
for i = 1:100
    Sig1(i,i) = 1/i;
    Sig2(i,i) = 1/(i^6);
    Sig3(i,i) = 2^(.6*(1-i));
end

%finds the norm of each sigma
normSig1 = norm(Sig1);
normSig2 = norm(Sig2);
normSig3 = norm(Sig3);

%generates the random b and normalizes it
b = rand(100,1);
b = b/norm(b);

%SVD
A1 = U*Sig1*V';
A2 = U*Sig2*V';
A3 = U*Sig3*V';
```

2c.

```
%Solve with backslash
```

```
x1 = A1\b;
x2 = A2\b;
x3 = A3\b;

%residuals
r1a = abs(A1*x1 - b);
r2a = abs(A2*x2 - b);
r3a = abs(A3*x3 - b);


% We get these residuals because of the condition number, which is listed
below:
% kA1 = 100
% kA2 = 1e12
% kA3 > 1e16
%
% Since we typically lose log10(K(A)) digits, our residuals should be
accurate because when we multiply
% USigV', we are losing that many digits. We lost the expected number of
digits in each answer.
% For 1, we lose 2 digits
% For 2, we lose ~12 digits
% For 3, we lose all of our digits

%takes first column of U as b and normalizes it
b = U(:,1);
b = b/norm(b);

%solve
x1 = A1\b;
x2 = A2\b;
x3 = A3\b;

%residuals
r1b = abs(A1*x1 - b);
r2b = abs(A2*x2 - b);
r3b = abs(A3*x3 - b);

% Rank and Error are displayed in the modified LowRank_SVD
```

2d.

```
% We get residuals of ~0, this is because the other b values introduced
% random error, while when we take the first column of U = b, there is not
% random error.

tol = 1e-8;

LR1 = Lowrank_SVD(A1, tol);
LR2 = Lowrank_SVD(A2, tol);
LR3 = Lowrank_SVD(A3, tol);

rankLR1 = rank(LR1);
rankLR2 = rank(LR2);
rankLR3 = rank(LR3);
```

2e.

```
%displays the rank
display(rankLR1);
display(rankLR2);
display(rankLR3);

%displays the error in the norm
errorLR1 = norm(A1) - norm(LR1);
errorLR2 = norm(A2) - norm(LR2);
errorLR3 = norm(A3) - norm(LR3);

display(errorLR1);
display(errorLR2);
display(errorLR3);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Homework 3 part 3
```

3a.

```
%plot the original graph
t = .01*pi:.01*pi:pi;
figure
plot(t,X)
legend('1','sin(x)','cos(x)','sin(x)cos(x)','sin(x)^2','cos(x)^2')

%%%% part b
alpha1 = pinv(X)*(Y);
alpha2 = pinv(X)*(Y2);
alpha3 = pinv(X)*(Y3);
```

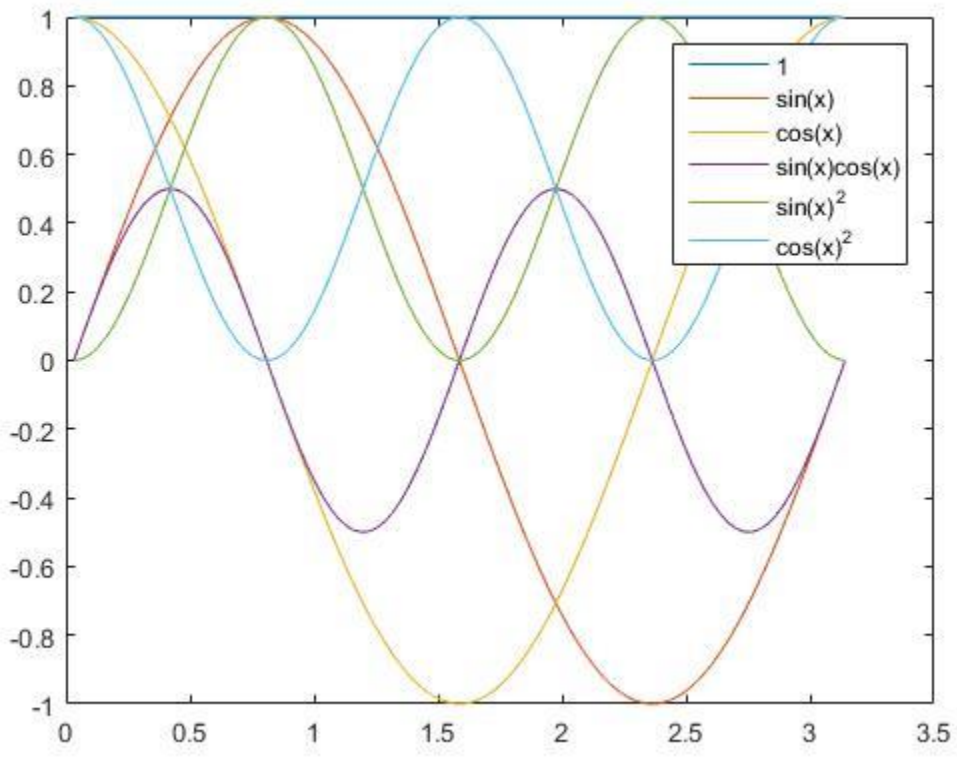
3b.

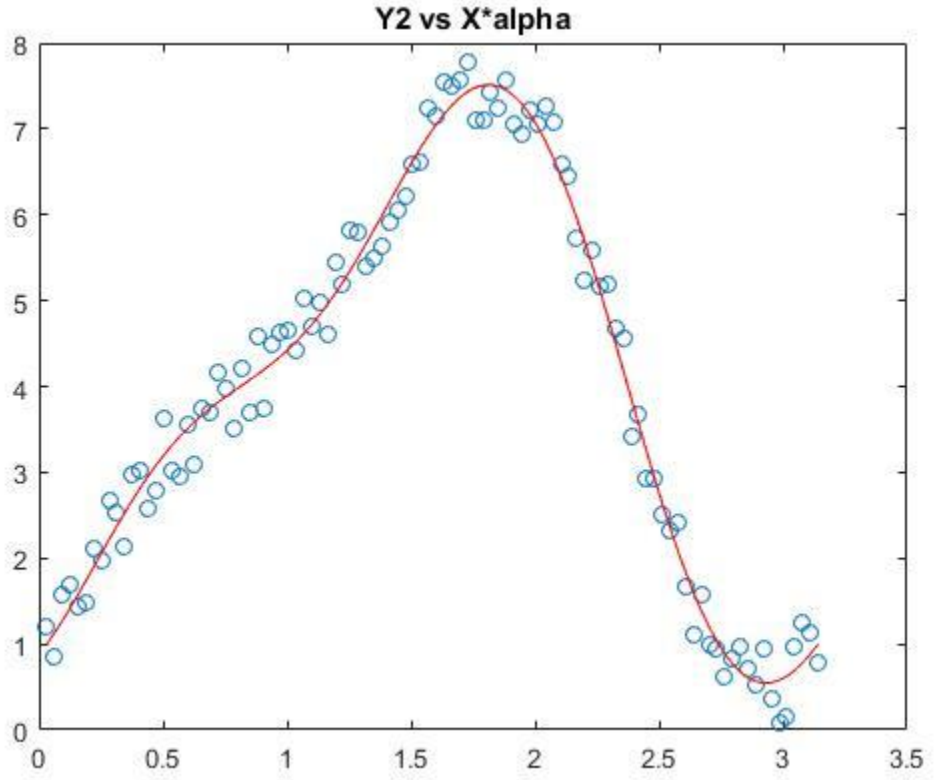
```
% alpha_1 is not the only solution to this equation, but rather the
% solution with the minimum norm

% A1 = reshape(alpha1,[100,1]);
% A2 = reshape(alpha2,[100,1]);
% A3 = reshape(alpha3,[100,1]);

%find the residual
residual_1 = (abs(Y-X*alpha1)).^2;
residual_2 = (abs(Y2-X*alpha2)).^2;
residual_3 = (abs(Y3-X*alpha3)).^2;
```

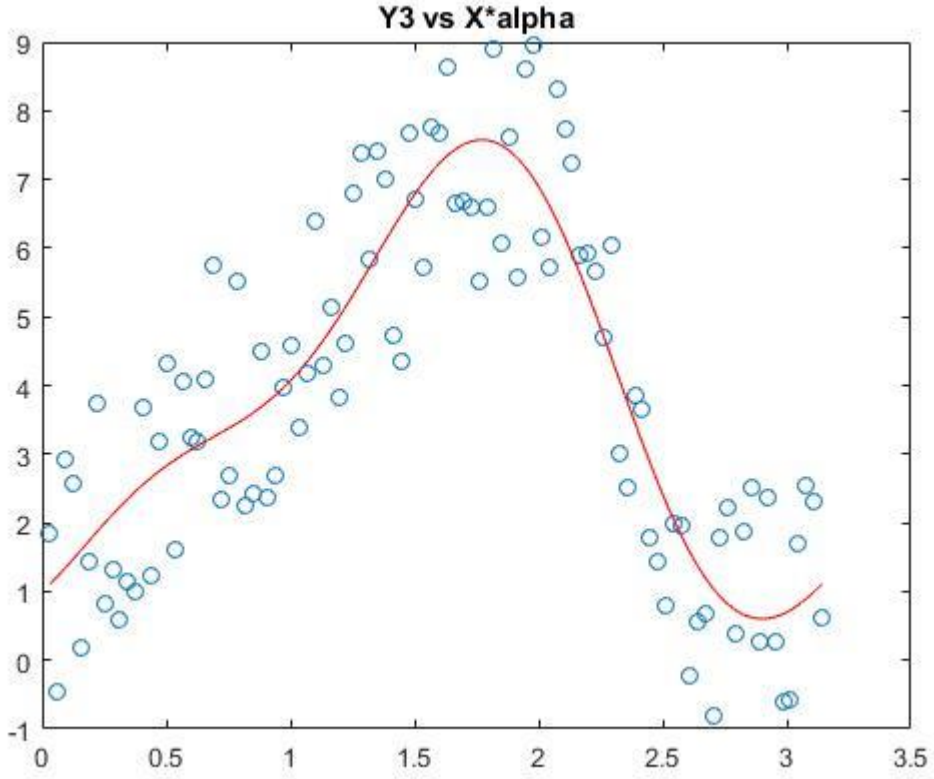
3c.

```
%plots Y2 vs X*aplha2
figure
t2 = .01*pi:.01*pi:pi;
plot(t2, Y2, 'o');
hold on;
plot(t2, X*alpha2, '-r');
title('Y2 vs X*alpha')
```

```
%plots Y3 vs X*alpha3
figure
plot(t2, Y3, 'o');
hold on;
plot(t2, X*alpha3, '-r');
title('Y3 vs X*alpha')
```




Y2 vs X*alpha
8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3
3.5


Y3 vs X*alpha
9
8
7
6
5
4
3
2
1
0
-1
0
0.5
1
1.5
2
2.5
3
3.5