Assignment 9
Juan Silva
ECE 309
Apr. 25, 2018

Main Script

```
% *******************************************************
% Program Name: test.m
% Author: Juan Silva Last Modified: April 25, 2018
% Description: This program will ask the user to enter
% an experiment number and output a graph showing the
% best fit curve for that experiment.
% *******************************************************

clear, clc, close all,
format short, format compact

%Prompt user to enter experiment type.
in = input('Enter an experiment number: ');
[bestCurve, bestFit]  = BestCurve(in);

%Print best curve type from values found in BestCurve function.
fprintf('\t\tThe Best Fit is: %s Curve with SSE of %5.3f.\n', bestCurve,
bestFit)
```

Function Script

```
% *******************************************************
% Program Name: BestCurve.m
% Author: Juan Silva Last Modified: April 25, 2018
% Description: This function will read data from an excel
% sheet, calculate the SSE values using two methods,
% call a function to create a table to display the SSE
% values, and produce the graphs for each case.
% *******************************************************

function [bestCurve, bestFit] = BestCurve(in)


%-----------------------------------
%Part 1 - Obtain Experiment Values
%-----------------------------------


%Initialize x and y from Experiment Data
FILE1 = 'DATA_NewCurveFit.xlsx';     %Load data for Experiments 1 - 3
FILE2 = 'DATA_MyExperiment.xlsx';    %Load data for Experiments 4 - 5

%Assign Experiment data based on user's input number.
switch in
    case 1
         Exp = xlsread(FILE1, 1, 'A:B');    % Experiment # Data
         ExpTitle = 'Experiment 1';         % Update graph title
```

```
        cf = [2.037 2.423 10.64];          % Assign cftool SSEs
        t_num = 1;                         % table #
        p_num = 1;                         % data pair #
    case 2
        Exp = xlsread(FILE1, 1, 'C:D');
        ExpTitle = 'Experiment 2';
        cf = [163.8784 39.9325 142.4113];
        t_num = 1;
        p_num = 2;
    case 3
        Exp = xlsread(FILE1, 1, 'E:F');
        ExpTitle = 'Experiment 3';
        cf = [0.5952 1.15181 0.484];
        t_num = 1;
        p_num = 3;
    case 4
        Exp = xlsread(FILE2, 1, 'B:C');
        ExpTitle = 'Experiment 4';
        t_num = 1;
        p_num = 1;
        cf = [13.75 44.24 23.87];
    case 5
        Exp = xlsread(FILE2, 1, 'D:E');
        ExpTitle = 'Experiment 5';
        t_num = 1;
        p_num = 2;
        cf = [5.381 23.9 14.03];
    otherwise
       disp('Error! Invalid experiment number.')
       return
end

%Contains original x and y values for given experiment.
%For Loop is used to handle varying array sizes
for i=1:length(Exp)
    x(i) = Exp(i,1);
    y(i) = Exp(i,2);
end

%Print table for original data values from specified experiment
fprintf('\n\tx1\t\ty1\n')
temp = [x; y];
fprintf('\t%3.2f\t%3.2f\n',temp)

%Set constant x-range for best curve graphs.
xmin = min(x);
xmax = max(x);
xdense = xmin: 0.1: xmax;



%-------------------------------------
%Part 2 - Solve SSE: Polynomial curve
%-------------------------------------
```

```
%Evaluate polynomial for least-squares fit line.
abc = polyfit(x,y,2);
ynew = polyval(abc, x);

%Polynomial curve: (y = ax^2 + bx + c)
a = abc(1);
b = abc(2);
c = abc(3);

%Set constant y-range for best polynomial curve graph.
ydense1  = a.*xdense.^ 2 + b * xdense + c;

%Polynomial SSE
SSE_poly = sum((ynew - y).^2);

%Polynomial SSE (alternative method)
A = [x.^2; x; x.^0]';
x0 = [a b c]';
b = [y]';

AX_poly = (A*x0 - b)' * (A*x0 - b);



%---------------------------------------
%Part 3 - Solve SSE: Exponential curve
%---------------------------------------



%Modify y from original data to be log(y)
logy = log(y);

%Evaluate exponential for least-squares fit line.
MB = polyfit(x,logy,1);
M = MB(1);
B = MB(2);

%Exponential curve: (y = ce^ax)
C = exp(B);
a = M;

%Set constant y-range for best exponential curves graph.
ynew = C * exp(a * x);
ydense2 = C * exp(a * xdense);

%Exponential SSE
SSE_exp = sum((ynew - y).^2);

%Exponential SSE(alternative method)
A = [x; x.^0]';
x0 = [M B]';
b = [logy]';

AX_exp = (A*x0 - log(b))' * (A*x0 - log(b));
```

```
%--------------------------------
%Part 4 - Solve SSE: Power curve
%--------------------------------


%Modify x from original data to be log(x)
logx = log(x);

%Evaluate power for least-squares fit line.
MB = polyfit(logx,logy,1);
M = MB(1);
B = MB(2);

%Exponential power: (y = cx^a)
C = exp(B);
a = M;

%Set constant y-range for best power curves graph.
ynew = C * x.^a;
ydense3 = C * xdense.^a;

%Power SSE
SSE_pow = sum((ynew - y).^2);

%Power SSE (alternative method)
A = [x; x.^0]';
x0 = [M B]';
b = [logy]';

c = inv(A'*A) * (A'*b);
b2 = exp(c(2));

AX_pow = (A*x0 - log(b2))' * (A*x0 - log(b2));

%Open cftool to verify graph and SSE values
cftool(x,y)


%-------------------------------
%Part 5 - Determine Best Curve
%-------------------------------

%Place SSEs in an array and sort them
%First value is best fit, third value is worst fit
SSE_temp = [SSE_poly SSE_exp SSE_pow];
SSE = sort(SSE_temp);

%Best fit located in SSE(1), first value in SSE array
switch SSE(1)
    case SSE_poly
        bestFit = SSE_poly;                 %Assign best fit
        y_best = ydense1;                   %y values based on curve type
        bestCurve = 'Polynomial';           %Best fit string for "Phrase"
```

```
        leg_best = 'Best Fit: Polynomial';  %Label for legend
    case SSE_exp
        bestFit = SSE_exp;
        y_best = ydense2;
        bestCurve = 'Exponential';
        leg_best = 'Best Fit: Exponential';
    otherwise
        bestFit = SSE_pow;
        y_best = ydense3;
        bestCurve = 'Power';
        leg_best = 'Best Fit: Power';
end

%Next fit located in SSE(2), second value in SSE array
switch SSE(2)
    case SSE_poly
        nextFit = SSE_poly;
        y_next = ydense1;
        nextCurve = 'Polynomial';
        leg_next = 'Next Fit: Polynomial';
    case SSE_exp
        nextFit = SSE_exp;
        y_next = ydense2;
        nextCurve = 'Exponential';
        leg_next = 'Next Fit: Exponential';
    otherwise
        nextFit = SSE_pow;
        y_next = ydense3;
        nextCurve = 'Power';
        leg_next = 'Next Fit: Power';
end

%Worst fit located in SSE(3), third value in SSE array
switch SSE(3)
    case SSE_poly
        worstFit = SSE_poly;
        y_worst = ydense1;
        worstCurve = 'Polynomial';
        leg_worst = 'Worst Fit: Polynomial';
    case SSE_exp
        worstFit = SSE_exp;
        y_worst = ydense2;
        worstCurve = 'Exponential';
        leg_worst = 'Worst Fit: Exponential';
    otherwise
        worstFit = SSE_pow;
        y_worst = ydense3;
        worstCurve = 'Power';
        leg_worst = 'Worst Fit: Power';
end


%---------------------------
%Part 6 - Create the Graph
%---------------------------
```

```
%Plot the graps based on switch cases in Part 5
plot(xdense, y_best, 'g', xdense,  y_next, '--b', xdense, y_worst, ':k', x,
y, 'ro')
grid
title(ExpTitle)
xlabel('x1')
ylabel('y1')
legend(leg_best, leg_next, leg_worst, 'Original Data')

%Call tables function to create a table using SSE values
Tables(SSE_poly, SSE_exp, SSE_pow, AX_poly, AX_exp, AX_pow,p_num,t_num,cf)

end
```

Function Script

```
% ******************************************************
% Program Name: Tables.m
% Author: Juan Silva Last Modified: April 25, 2018
% Description: This function will produce a table using
% SSE values passed from the BestCurve function.
% ******************************************************

function Tables(SSE_poly, SSE_exp, SSE_pow, AX_poly, AX_exp, AX_pow, p_num,
t_num, cf)

fprintf('\n\t\t\t|-----------------------------------------------|\n')
fprintf('\t\t\t|  Poly. Curve  |   Exp. Curve  |  Power Curve  |\n')
fprintf('\t|-------|---------------|---------------|---------------|\n')

%Print Row 1 containing SSE values calculated using Method 1
fprintf('\t|yNew\t|\t%5.3f\t\t|\t%5.3f\t\t|\t%5.3f\t\t|\n', SSE_poly,
SSE_exp, SSE_pow)
fprintf('\t|-------|---------------|---------------|---------------|\n')

%Print Row 2 containing SSE values calculated using Method 2
fprintf('\t|Ax -
b\t|\t%5.3f\t\t|\t%5.3f\t\t|\t%5.3f\t\t|\n',AX_poly,AX_exp,AX_pow)
fprintf('\t|-------|---------------|---------------|---------------|\n')

%Print Row 3 containing SSE values calculated using cftool
fprintf('\t|cftool\t|\t%5.3f\t\t|\t%5.3f\t\t|\t%5.3f\t\t|\n', cf)
fprintf('\t|-------------------------------------------------------|\n')

%Print data types.
fprintf('\t\t\t\t\t\tTable %d, Data Pair %d\n', t_num, p_num)

end
```

Results

Experiment 1:

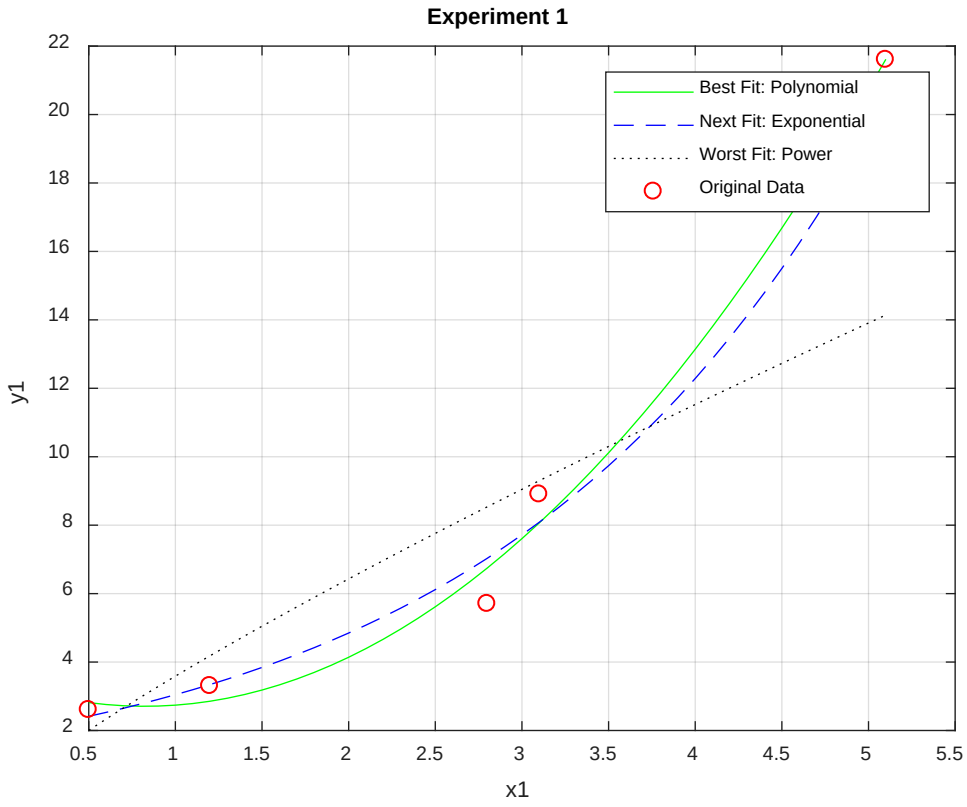
```
Enter an experiment number: 1
```

x1	y1
0.50	2.60
1.20	3.30
2.80	5.70
3.10	8.90
5.10	21.60

	Poly. Curve	Exp. Curve	Power Curve
yNew	2.037	3.710	64.968
Ax – b	2.037	9.180	47.418
cftool	2.037	2.423	10.640

Table 1, Data Pair 1

The Best Fit is: Polynomial Curve with SSE of 2.037.



Experiment 2:

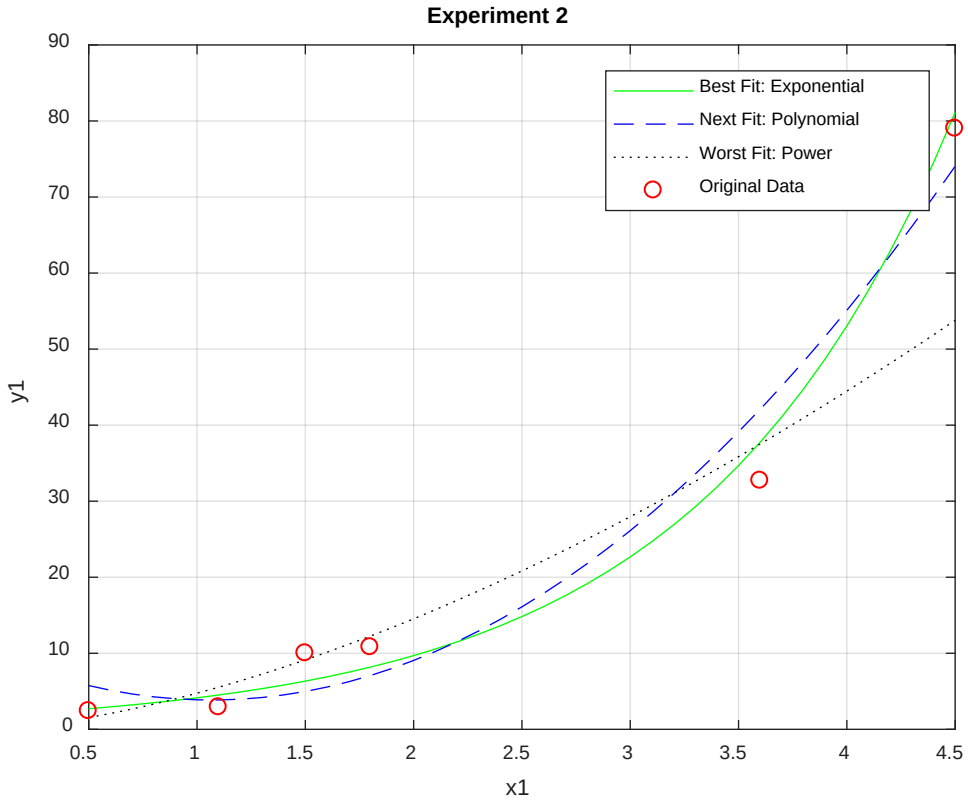
```
Enter an experiment number: 2

      x1    y1
      0.50  2.40
      1.10  2.90
      1.50  10.00
      1.80  10.80
      3.60  32.70
      4.50  79.00
```

	Poly. Curve	Exp. Curve	Power Curve
yNew	163.878	53.065	669.434
Ax – b	163.878	20.120	151.784
cftool	163.878	39.932	142.411

Table 1, Data Pair 2

The Best Fit is: Exponential Curve with SSE of 53.065.



Experiment 3:

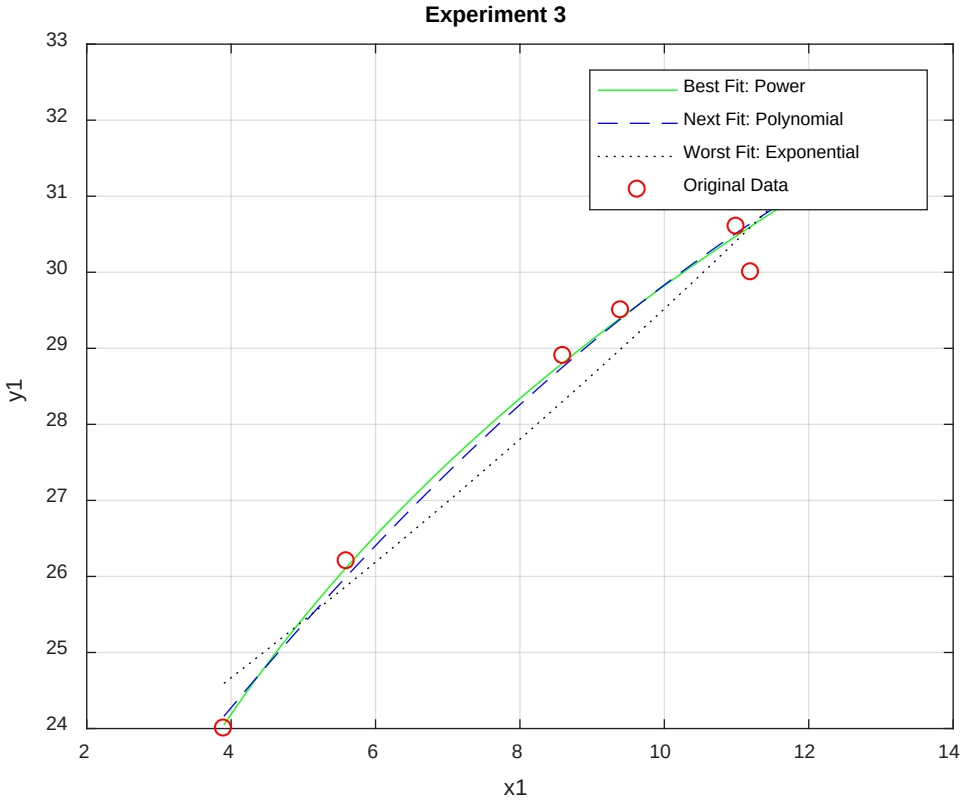
```
Enter an experiment number: 3
```

x1	y1
3.90	24.00
5.60	26.20
9.40	29.50
8.60	28.90
11.00	30.60
11.20	30.00
13.10	32.00

	Poly. Curve	Exp. Curve	Power Curve
yNew	0.595	1.607	0.484
Ax – b	0.595	32.218	26.850
cftool	0.595	1.152	0.484

Table 1, Data Pair 3

The Best Fit is: Power Curve with SSE of 0.484.



Experiment 4:

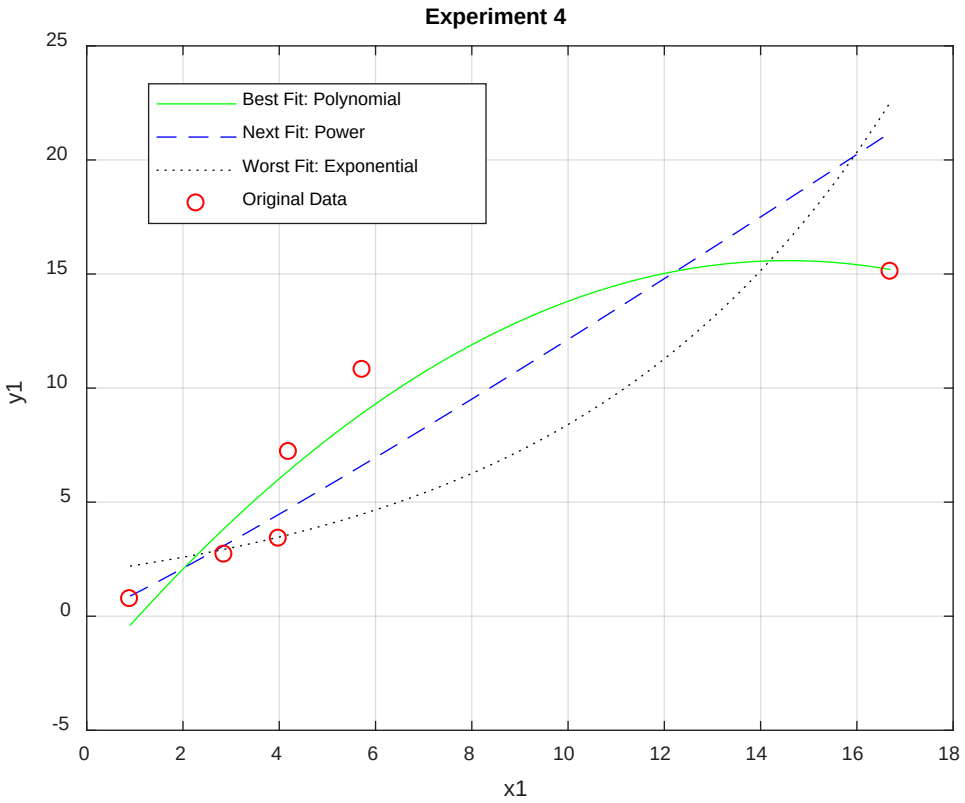
```
Enter an experiment number: 4
```

x1	y1
0.90	0.75
2.86	2.70
3.99	3.40
4.20	7.20
5.73	10.80
16.70	15.10

	Poly. Curve	Exp. Curve	Power Curve
yNew	13.749	110.722	62.271
Ax – b	13.749	21.500	372.793
cftool	13.750	44.240	23.870

Table 1, Data Pair 1

The Best Fit is: Polynomial Curve with SSE of 13.749.



Experiment 5:

```
Enter an experiment number: 5
```

x1	y1
2.40	0.99
2.97	1.60
3.74	2.49
4.05	3.75
4.75	7.50
5.98	7.80
7.30	9.90
9.10	10.40

	Poly. Curve	Exp. Curve	Power Curve
yNew	5.381	63.395	32.062
Ax – b	5.381	47.211	688.342
cftool	5.381	23.900	14.030

Table 1, Data Pair 2

The Best Fit is: Polynomial Curve with SSE of 5.381.