MAT 343 Lab 5 - Jordan Ledbetter

Exercise 1

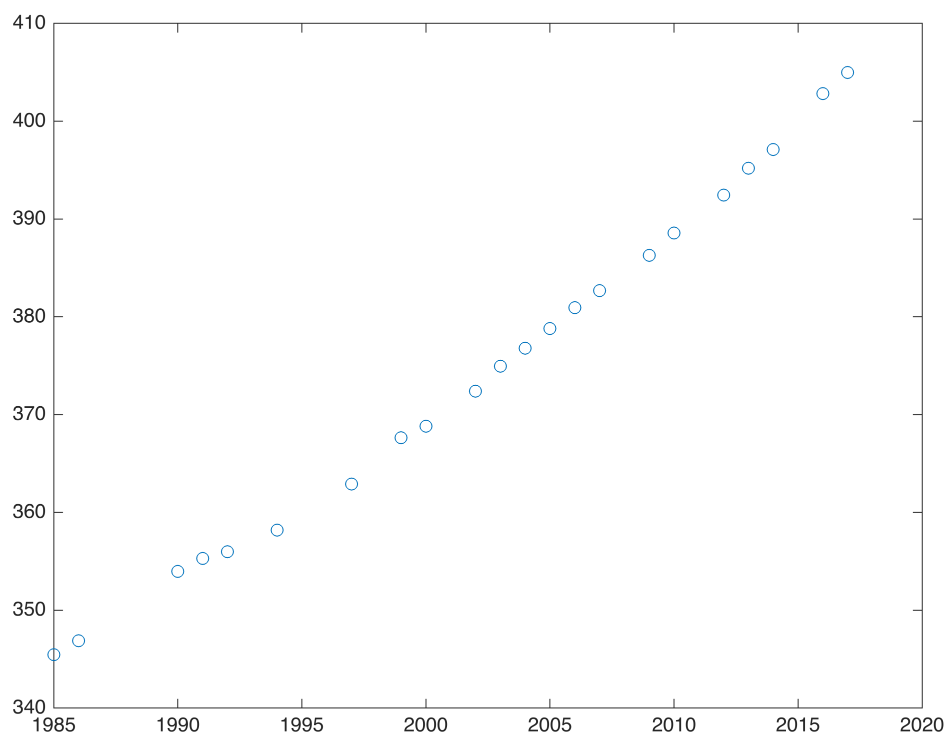
Load the file gco2.dat

```
dat = load('gco2.dat')
```

```
dat = 30×2
   1.9850e+03   3.4546e+02
   1.9850e+03   3.4546e+02
   1.9850e+03   3.4546e+02
   1.9860e+03   3.4688e+02
   1.9900e+03   3.5397e+02
   1.9910e+03   3.5529e+02
   1.9920e+03   3.5599e+02
   1.9920e+03   3.5599e+02
   1.9940e+03   3.5820e+02
   1.9970e+03   3.6290e+02
       ⋮
```

Create the data vectors year and conc and plot the data points.

```
format short e
year = dat(:,1);
conc = dat(:,2);
plot(year, conc, 'o')
```



(a)

Write the code for finding and plotting the best linear fit. Make sure the vector c is displayed.

```
% build the matrix X for a linear model
X = [ones(size(year)), year];

% Solve Normal Equations
z = X'*conc;
S = X'*X;
U = chol(S);
w = U'\z;
c = U\w
```

```
c = 2×1
  -3.2494e+03
   1.8101e+00
```

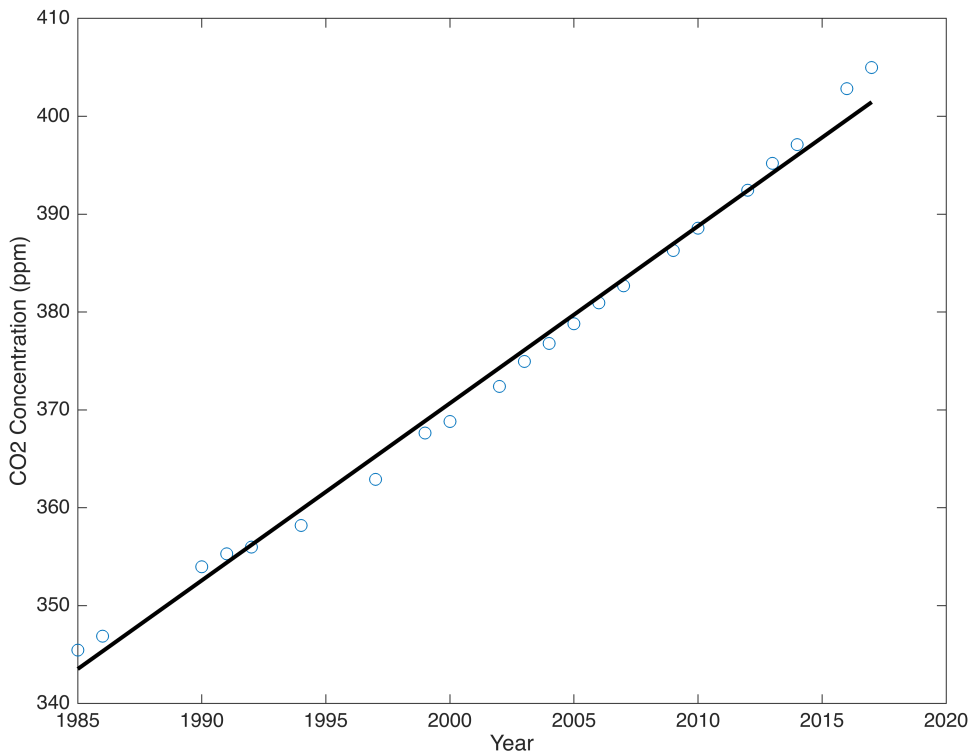
```
fprintf('c1 = %.5e, c2 = %.5e\n', c(1), c(2));
```

```
c1 = -3.24945e+03, c2 = 1.81006e+00
```

```
% define vector for linear function
q = year;

% define linear fit
lFit = c(1) + c(2)*q;
plot(year,conc,'o')

% plot the linear fit with the data points
hold on
plot(q, lFit, 'k', 'LineWidth', 2);
xlabel('Year');
ylabel('CO2 Concentration (ppm)');
hold off
```



(b)

Write the code for finding the quadratic fit and plot the quadratic and linear fit in the same figure. Make sure the vector c is displayed.

```
format short e

% build the matrix X for the quadratic model
X = [ones(length(year),1), year, year.^2];

% Solve Normal Equations
z = X'*conc;
S = X'*X;
U = chol(S);
w = U'\z;
c = U\w
```

```
c = 3×1
   6.1073e+04
  -6.2507e+01
   1.6077e-02
```

```
fprintf('c1 = %.5e, c2 = %.5e, c3 = %.5e\n', c(1), c(2), c(3));
```

```
c1 = 6.10731e+04, c2 = -6.25066e+01, c3 = 1.60773e-02
```

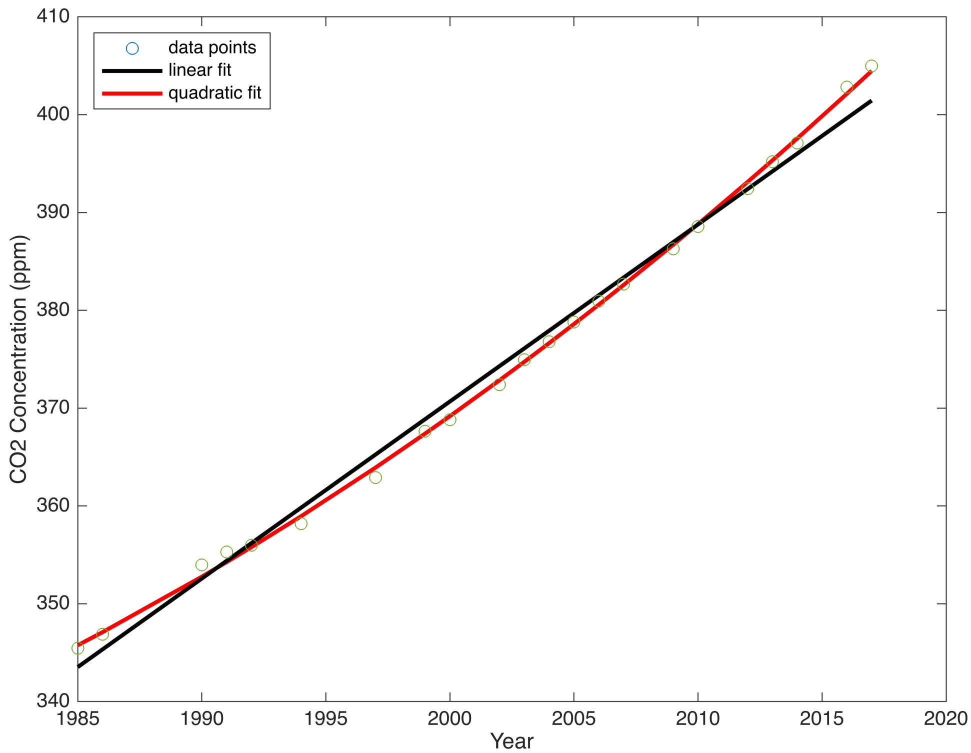
```
% define vector for quadratic function
```

```
q = year;

% define quadratic fit
qFit = c(1) + c(2)*q + c(3)*q.^2;

% plot the quadratic fit with the data points
hold on;
plot(q, qFit, 'r', 'LineWidth', 2);
plot(q, lFit,'k','LineWidth',2);
plot(year,conc,'o');
xlabel('Year');
ylabel('CO2 Concentration (ppm)');

% add a legend
legend('data points','linear fit','quadratic fit','location','northwest');
hold off
```



Question 2

(a)

Enter the vector t (use the colon operator ":") and the vector Y *(note the upper case) as given in the statement of the problem. Plot the data points* (t,Y), *find the best linear fit and plot it together with the points. Make sure the vector c is displayed.*

```
clear
```

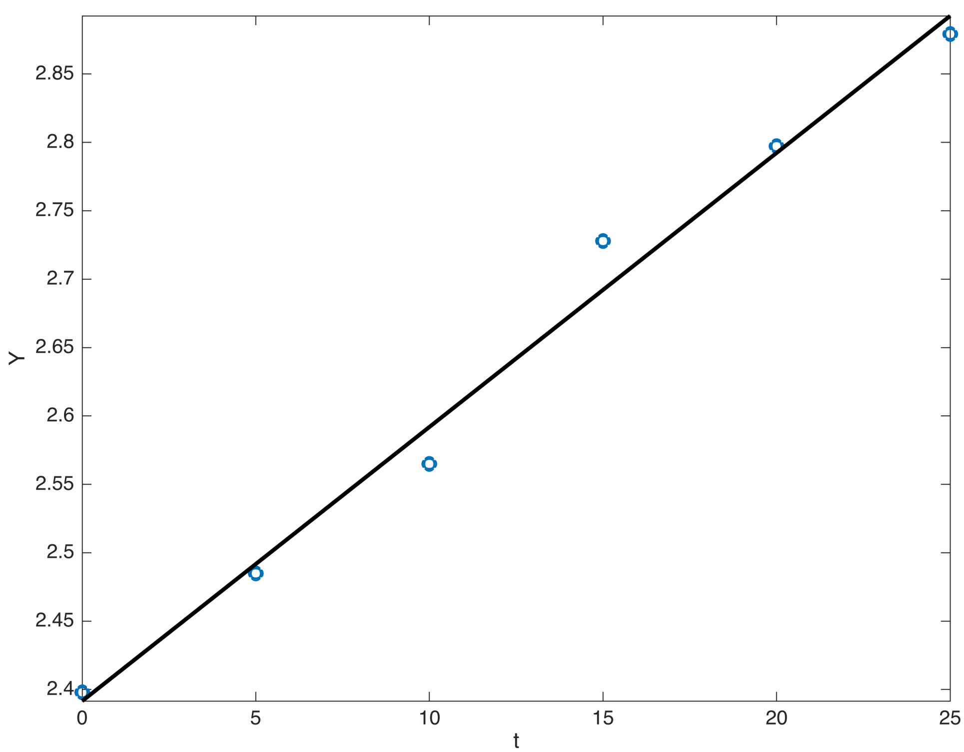
```
clc
% define the vectors
t = 0:5:25;
Y = log([11; 12; 13; 15.3; 16.4; 17.8]);

% compute the coefficients
X=[ones(length(t), 1),t'];
c = X \ Y;
c1 = c(1);
c2 = c(2);

% define range for line of best fit
q = t;

% define line of best fit
fit = c1 + c2 * q;

% plot data points with line of best fit
plot(t, Y, 'o', q, fit, '-k', 'LineWidth', 2);
axis tight;
xlabel('t')
ylabel('Y')
```




```
% display coefficients with 5 digit accuracy
fprintf('c1 = %.5e, c2 = %.5e\n', c1, c2);
```

```
c1 = 2.39155e+00, c2 = 2.00374e-02
```

(b)

Define y (lower case) in terms of Y (upper case), and a and b in terms of `c(1)` and `c(2)` where `c` is the vector from part (a)

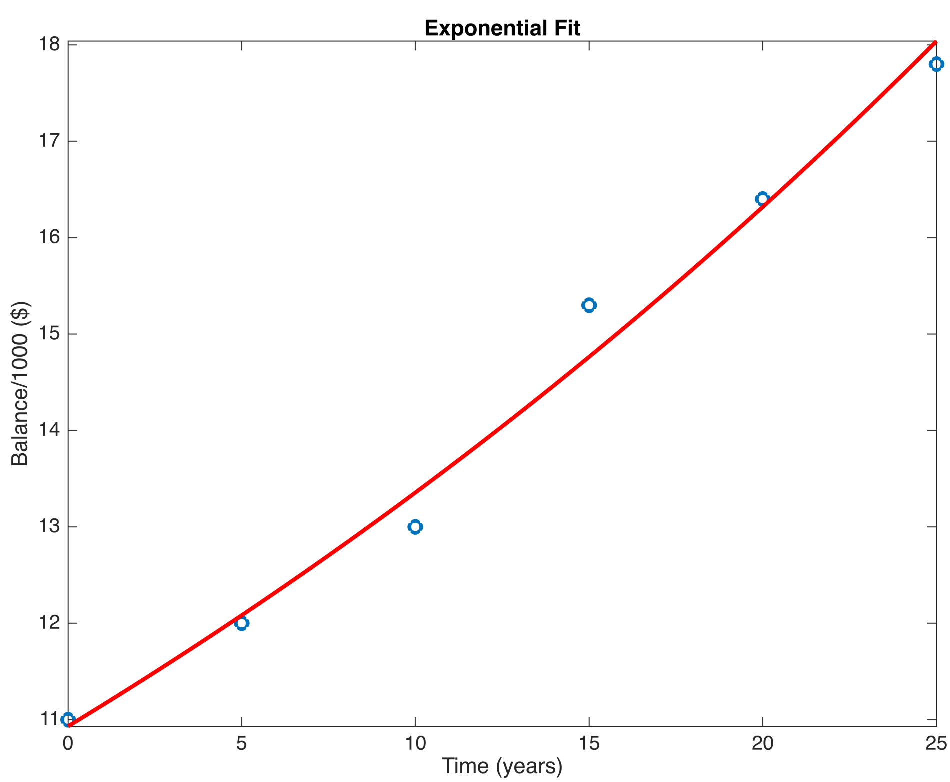
```
% compute the coefficients
y = exp(Y);
a = exp(c1);
b = c2;
```

plot the data points (t,y) together with the exponential fit $y = ae^{bt}$. Use an appropriate vector q to graph the exponential fit.

```
% define range for exponential fit
q = linspace(min(t), max(t), 100);

% define exponential fit
expFit = a * exp(b*q);

% plot the data points and the exponential fit
plot(t, y, 'o', q, expFit, '-r', 'LineWidth', 2);
axis tight;
xlabel('Time (years)');
ylabel('Balance/1000 ($)');
title('Exponential Fit');
```



(c)

Enter the time at which the balance will reach the given amount (make sure you explain how you find the value of t)

By extending the q range and analyzing the line of best fit, it was determined that it would take approximately 30 years when the balance will reach $20,000.

Question 3

Enter the vector m (use the colon operator :), and enter the average temperature Y. Both vectors must be **column** vectors.

```
clear
% define the data, months and average temperatures
m = (1:12)';
Y = [38.8; 40.9; 45.1; 50.5; 58.4; 66.4; 75.6; 74.4; 67.7; 56; 44.1; 38.2];
```

(a)

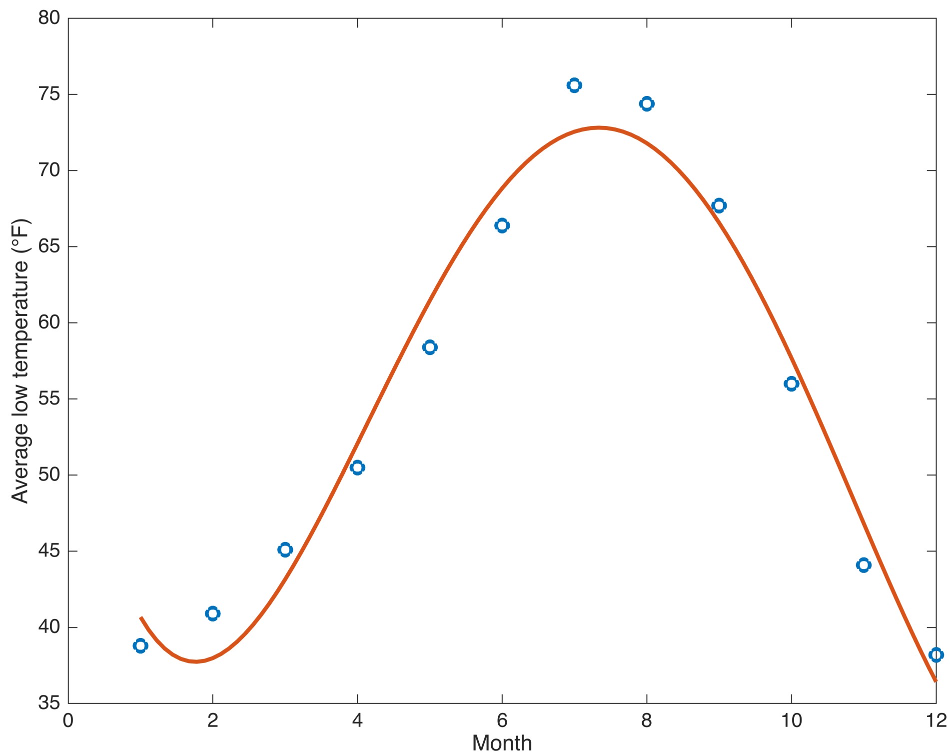
```
% define matrix X
X = [ones(length(m),1), m, m.^2, m.^3, m.^4];

% compute the coefficients
c = (X'*X)\(X'*Y);
disp(c)
```

```
   5.6184e+01
  -2.3763e+01
   9.2535e+00
  -1.0296e+00
   3.4335e-02
```

```
% define the best least square fit
q = linspace(1,12,100)';
y = c(1) + c(2)*q+ c(3)*q.^2 + c(4)*q.^3 + c(5)*q.^4;

% plot the data points with the line of best fit
plot(m,Y,'o',q,y,'LineWidth',2);
xlabel('Month');
ylabel('Average low temperature (°F)');
```



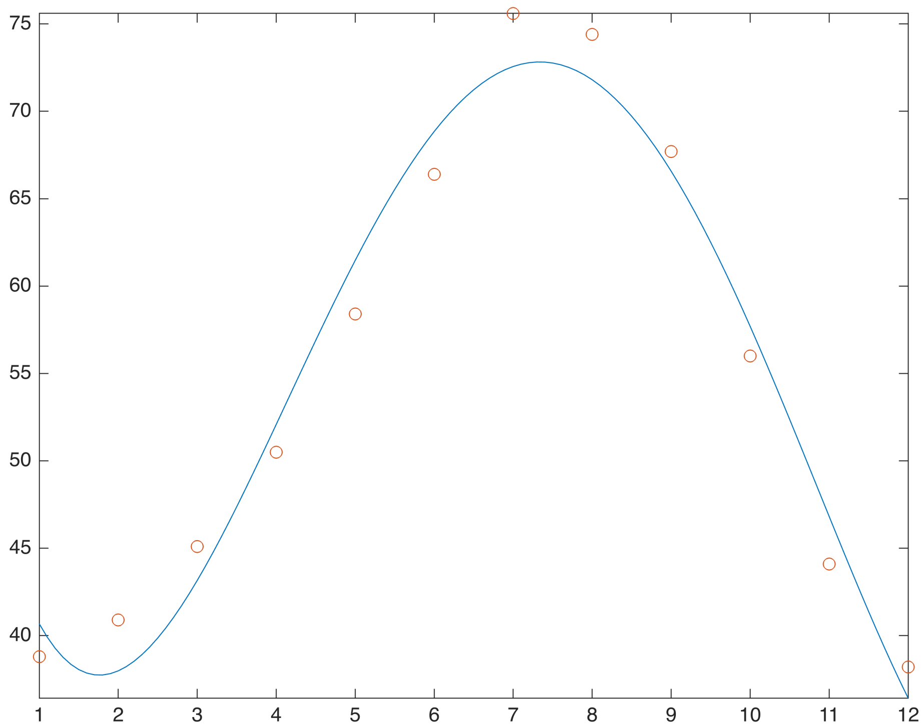
(b)

Enter the given commands

```
c = X \ Y
```

```
c = 5×1
   5.6184e+01
  -2.3763e+01
   9.2535e+00
  -1.0296e+00
   3.4335e-02
```

```
c = c ([5: -1:1]) ;
q = 1:0.1:12;
z = polyval (c , q ) ;
figure
plot (q ,z ,m ,Y ,'o') ;
axis tight
```



How do the values of c compare to the ones you found in part (a)?

Answer: The values of c are in parts a and b are the same; however, the commands in part a provide more flexibility and control over the process In part b, we used 'polyfit' and 'polyval' functions to calculate the coefficients as opposed to part a, where we manually computed the coefficients.

How does the plot compare to the one you found in part (a)?

Answer: The plots in part a and b are the same, except for the range values in part a. In part a, the graph ranges from 35 to 80, where in part b, the graph is plot to fit the data perfectly; therefore, the plot hits every point of range, from 35 to 75.