

Linear Regression of CO₂ Data

Due Time: 23:59, April 16 2017

Earnings: 9% of your final grade

NOTE: Plan to finish a few days early to avoid last minute hardware/software holdups.

The code in this assignment must be your own work. It must not be code taken from another student or written for you by someone else, even if you give a reference to the person you got it from (attribution); if it is not entirely your own work it will be treated as plagiarism and given a fail mark, or less.

Purpose: Fit data using linear regression least squares. Fit to a straight line and a power law function. Test the relative success of the fits by calculating the standard deviation σ in each case from the expression given in course notes:

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (y_i - f(x_i))^2}{N - 2}}$$

Algorithm. Write a program named `ass3` that will enable the user to fit data to a simple function. The application can analyze both linear and exponential data. If you use dynamic memory to hold data, it must be released at the end of the application.

Here is a data set for CO₂ levels in the World at different years.

time - year	1960	1965	1970	1975	1980	1985	1990	1995	2000	2005
CO ₂ level ppm	316.5	320	325	331	338	346	353	360	368.5	378

The data is also supplied in the file `CO2.txt`. It is the same data as above but 1960 has been subtracted from each of the years and 316.5 has been subtracted from each CO₂ level. In other words, 1960 is the start year and levels are measured relative to what it was in that year. This is better data to fit to since it does not accumulate large numbers (and so reduces epsilon cancellation effects):

CO₂.txt

time - year	5	10	15	20	25	30	35	40	45
CO ₂ level ppm	3.5	8.5	14.5	21.5	29.5	36.5	43.5	52	61.5

When the application runs it offers the user two options.

Option 1. Fit to a straight Line: CO₂ level = a*year + b

a and b are the constants calculated from the fit.

A data set to test your code is given in the file `CO2.txt`. It is the level of carbon dioxide in the atmosphere measured at different years between 1960 and 2005. (NOTE: your code must be able to read a file with an arbitrary number of points even though there are only 9 data points in this example).

In this linear case you are fitting the data to the straight line $y = ax + b$, where y represents the CO₂ level and x represents the year. Solve these equations using the code discussed in class to get solutions for a and b. Use the fit to offer the user the option of extrapolating into the future.

Option 2. Fit to a power law CO₂ level = a*year^b

a and b are the constants calculated from the fit.

You are fitting the data to the power law $y = ax^b$ where x and y refer to years and CO₂ levels respectively, as in the linear case. You will use the data in the `CO2.txt` file, as in the linear case.

Your starting point for this is the appropriate transform discussed in class. Make the transform so that the data can be fit to a linear function to calculate c and m for this equation. The way to do this is take the natural logs of y and x in the following way:

$$\ln(y) = \ln(ax^b) = \ln(a) + \ln(x^b) = \ln(a) + b \cdot \ln(x)$$

CST 8233 – W17 - Assignment #3

so using the new variables $Y = \ln(y)$ and $X = \ln(x)$ we now have a straight line equation again :

$$Y = A + b \cdot X$$

where $A = \ln(a)$. You can get a from the fit parameter A since $a = \ln^{-1}(A) = \exp(A)$.

Write the code so that the file is read and a least squares fit is done. Use the equation to give the user the option of extrapolating the CO2 level to time in the future, as in the linear case.

An example of the output of the running application is given at the end. Yours must work identically and produce identical (or very similar) output. When the application terminates it releases all dynamically allocated memory so it does not have a resource leak (or you lose 30%).

See the Marking Sheet for how you can lose marks, but you will lose at least 60% if: it fails to build in Visual Studio 2013, it crashes in normal operation, it doesn't produce the example output.

What to Submit : Use Blackboard to submit this assignment as a zip file (**not RAR**) containing only the source code file(s) `ass3.cpp` (definitely not the entire project!). The name of the zipped folder **must** contain your name as a prefix so that I can identify it, for example using my name the file would be `tyleraAss3CST8233.zip`. It is also vital that you include the Cover Information (as specified in the Submission Standard) as a file header in your source file so the file can be identified as yours. Use comment lines in the file to include the header.

- Before you submit the code, check that it builds and executes in Visual Studio 2013 as you expect - if it doesn't build for me, for whatever reason, you get a deduction of at least 60%.
- Make sure you have submitted the correct file – if I cannot build it because the file is wrong or missing from the zip, even if it's an honest mistake, you get 0.
- Because of the imminent semester end, this assignment cannot be late.

Example Output:

```
*****
Linear Regression of Data - Function Type Menu
1. Linear:      y = a*x + b
2. Power Law:  y = a*x^b
3. Quit
*****
Select an option: 1

Please enter the name of the file to open:CO2.txt

FILE OPENED FOR READING
There are 9 records.

File read into memory

Linear:  y = 1.451x -6.18
Standard Deviation = 1.553

*****
Extrapolation of data Menu
1. Do an extrapolation
2. Return to main menu
*****
Select an option: 1
Input a year (e.g. 2020) to extrapolate to: 2030
y(2030) = 411.9

*****
Extrapolation of data Menu
1. Do an extrapolation
2. Return to main menu
```

CST 8233 – W17 - Assignment #3

```
*****
Select an option: 2
*****
Linear Regression of Data - Function Type Menu
1. Linear:      y = a*x + b
2. Power Law:  y = a*x^b
3. Quit
*****
Select an option: 2

Please enter the name of the file to open:CO2.txt

FILE OPENED FOR READING
There are 9 records.

File read into memory

Power Law:  y = 0.426x^1.31
Standard Deviation = 0.0172

*****
Extrapolation of data Menu
1. Do an extrapolation
2. Return to main menu
*****
Select an option: 1
Input a year to extrapolate to: 2030
y(2.03e+003) = 426

*****
Extrapolation of data Menu
1. Do an extrapolation
2. Return to main menu
*****
Select an option:
```