

Zac Porter

Mini-Project 2

Section I:

The objective of this code is to perform linear regression analysis on a set of x-y data points read from a file. The code accomplishes this by implementing five functions to read the data from a file, calculate the mean of a data array, compute the slope and y-intercept of the linear fit line, calculate the y values corresponding to the x values based on the linear fit parameters, and write the original x-y data along with the computed y values to an output file "output.txt".

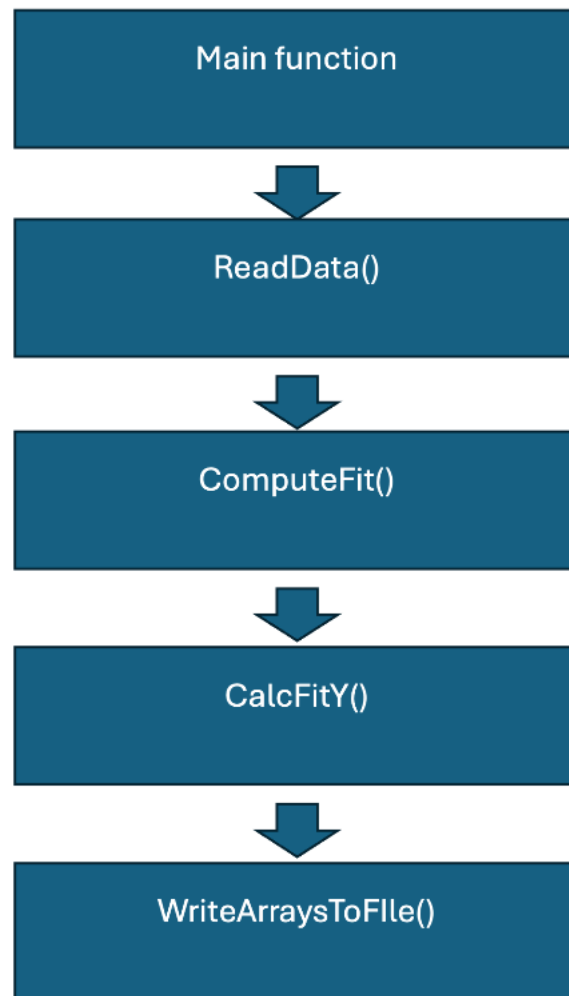
The basic algorithms used in this code can be summarized as follows:

1. ReadData Function: This function reads the x-y data pairs from a file and stores them in a dynamically allocated array.
2. CalcMean Function: This function calculates the mean of a given array of double values.
3. ComputeFit Function: This function calculates the slope (fitm) and y-intercept (fitb) of the linear fit line using the least squares method.
4. CalcFitY Function: This function calculates the y values corresponding to the x values based on the linear fit parameters (fitm and fitb).
5. WriteArraysToFile Function: This function writes the original x-y data along with the computed y values to an output file.

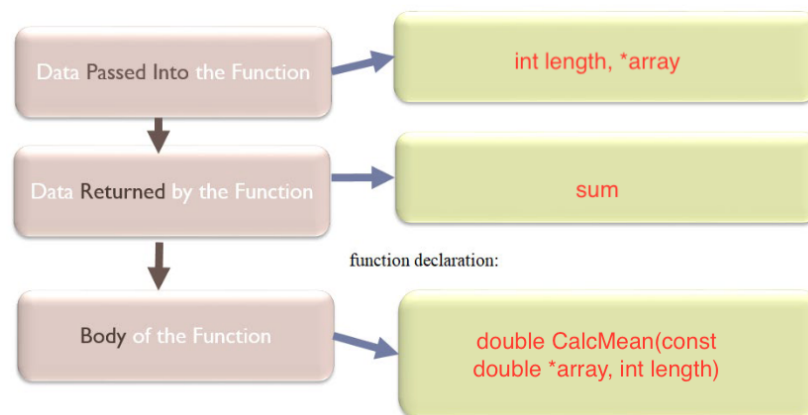
Main Function: The main function orchestrates the execution of the above functions. It reads the data from the file, computes the linear fit parameters, calculates the corresponding y values, and writes the results to an output file. Overall, this code demonstrates the implementation of linear regression analysis using C++ and serves as a practical example for understanding and applying basic statistical techniques in scientific computing.

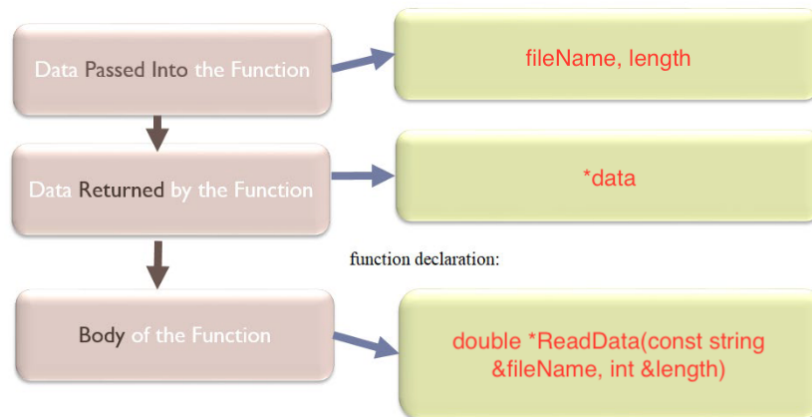
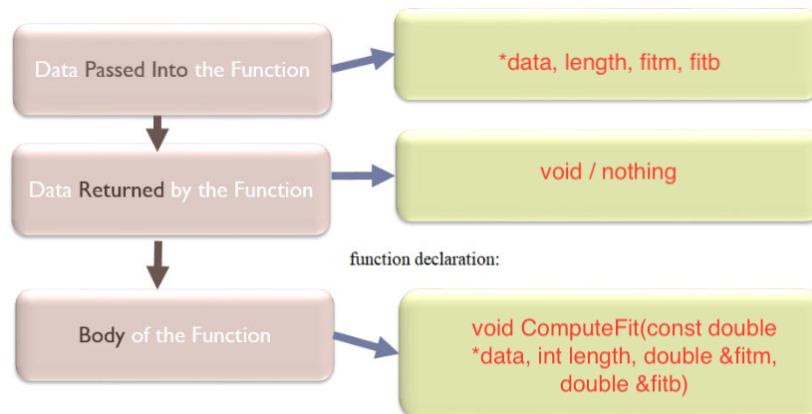
Section II:

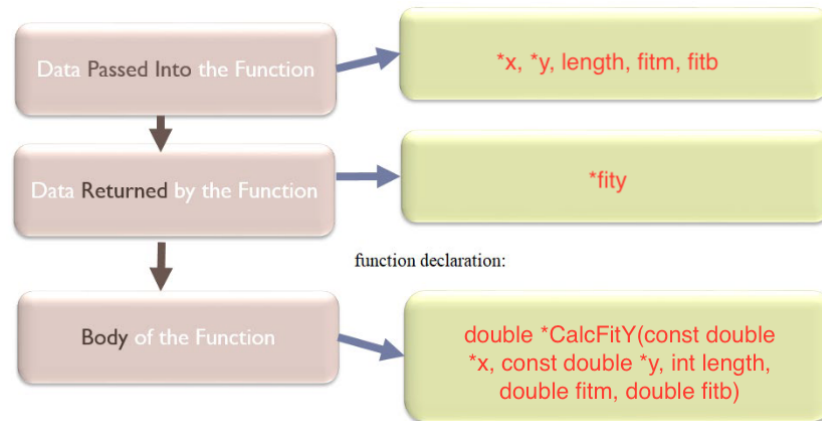
Top Level Design



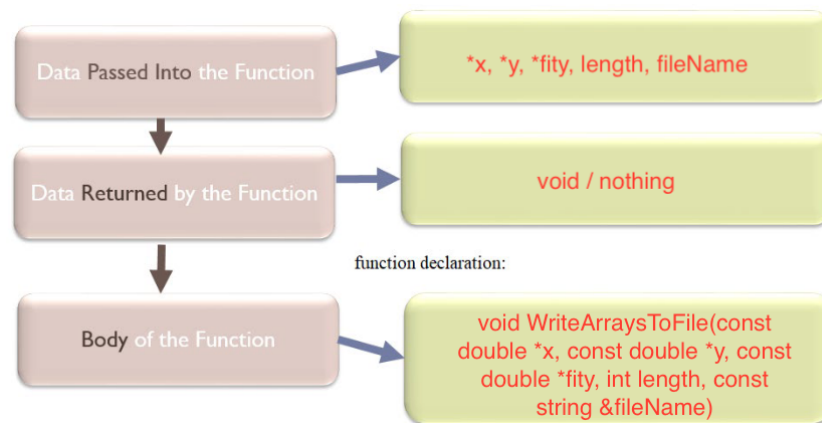
Function 1



Function 2**Function 3****Function 4**



Function 5



Section III:

<https://replit.com/join/ukuyvizsyg-zacporter179>

```

1  #include <fstream>
2  #include <iostream>
3  #include <cmath>
4  using namespace std;
5  //1 FUNCTION TO READ XY AND STORE IN DYNAMIC ARRAY
6  double *ReadData(const string &fileName, int &length){
7      ifstream inFile(fileName);
8      if(!inFile.is_open()) {
9          cout << "Error opening file " << fileName << endl;
10         exit(1);
11     }
12     length = 0;
13     double d1, d2;
14     while (inFile >> d1 >> d2) {
15         ++length;
16     }
17     inFile.clear();
18     inFile.seekg(0, ios::beg);
19     double *data = new double[length * 2]; //TWO COLUMNS
20     for (int i = 0; i < length * 2; i += 2){
21         inFile >> data[i] >> data[i + 1];
22     }
23     inFile.close();
24     return data; //COME BACK TO THIS LATER
25 }
26 //2 FUNCTION TO CALCULATE MEAN
27 double CalcMean(const double *array, int length){
28     double sum = 0.0;
29     for (int i = 0; i < length; ++i){
30         sum += array[i];
31     }
32     return sum / length;
33 }
34 //3 FUNCTION TO CALCULATE FITM AND FITB
35 void ComputeFit(const double *data, int length, double &fitm, double &fitb){
36     double sumx = 0.0, sumy = 0.0, sumxy = 0.0, sumxx = 0.0;
37     for (int i = 0; i < length * 2; i += 2){
38         sumx += data[i];
39         sumy += data[i + 1];
40         sumxy += data[i] * data[i + 1];
41         sumxx += data[i] * data[i];
42     }
43     fitm = (length * sumxy - sumx * sumy) / (length * sumxx - sumx * sumx);
44     fitb = (sumy - fitm * sumx) / length;
45 }
46 //4 FUNCTION TO CALCULATE FITY
47 double *CalcFitY(const double *x, const double *y, int length, double fitm, double
fitb) {
48     double *fity = new double[length];
49     for (int i = 0; i < length; ++i) {
50         fity[i] = fitm * x[i] + fitb;
51     }
52     return fity;
53 }
54 //5 FUNCTION TO WRITE 3 COLUMNS OF DATA TO FILE

```

1	0	1.20089	1.15394
2	0.00101	1.08858	1.15995
3	0.00202	1.27212	1.16597
4	0.00303	1.1093	1.17199
5	0.00404	1.17925	1.17801
6	0.005051	1.16301	1.18403
7	0.006061	1.20402	1.19005
8	0.007071	1.08805	1.19607
9	0.008081	1.15444	1.20209
10	0.009091	1.19427	1.20811
11	0.010101	1.31271	1.21412
12	0.011111	1.14548	1.22014
13	0.012121	1.27694	1.22616
14	0.013131	1.3749	1.23218
15	0.014141	1.3474	1.2382
16	0.015152	1.15781	1.24422
17	0.016162	1.3362	1.25024
18	0.017172	1.26993	1.25626
19	0.018182	1.13223	1.26228
20	0.019192	1.20276	1.26829
21	0.020202	1.2091	1.27431
22	0.021212	1.17536	1.28033
23	0.022222	1.35746	1.28635
24	0.023232	1.17765	1.29237
25	0.024242	1.17326	1.29839
26	0.025253	1.46543	1.30441
27	0.026263	1.27982	1.31043
28	0.027273	1.39116	1.31645
29	0.028283	1.50838	1.32246
30	0.029293	1.5008	1.32848
31	0.030303	1.19932	1.3345
32	0.031313	1.24569	1.34052
33	0.032323	1.35807	1.34654
34	0.033333	1.28826	1.35256
35	0.034343	1.32608	1.35857
36	0.035354	1.55045	1.3646
37	0.036364	1.24573	1.37062
38	0.037374	1.32519	1.37663
39	0.038384	1.32483	1.38265
40	0.039394	1.46663	1.38867
41	0.040404	1.53982	1.39469
42	0.041414	1.29167	1.40071
43	0.042424	1.49421	1.40673
44	0.043434	1.29796	1.41274
45	0.044444	1.37817	1.41876
46	0.045454	1.41956	1.42478
47	0.046465	1.32709	1.43081
48	0.047475	1.30873	1.43682
49	0.048485	1.35428	1.44284
50	0.049495	1.32435	1.44886
51	0.050505	1.58732	1.45488
52	0.051515	1.33176	1.4609
53	0.052525	1.35173	1.46691
54	0.053535	1.52558	1.47293
55	0.054545	1.58848	1.47895
56	0.055556	1.5848	1.48498

```

55 void WriteArraysToFile(const double *x, const double *y, const double *fity, int
length, const string &fileName){
56     ofstream outFile(fileName);
57     if (!outFile.is_open()) {
58         cout << "Error opening file " << fileName << endl;
59         exit(1);
60     }
61     for (int i = 0; i < length; ++i) {
62         outFile << x[i] << " " << y[i] << " " << fity[i] << endl;
63     }
64     outFile.close();
65 }
66 //MAIN
67 int main() {
68     string filename = "numbers.txt";
69     int length;
70     double *data = ReadData(filename, length);
71     double fitm, fitb;
72     ComputeFit(data, length, fitm, fitb);
73     double *x = new double[length];
74     double *y = new double[length];
75     double *fity;
76     for (int i = 0; i < length; ++i) {
77         x[i] = data[i * 2];
78         y[i] = data[i * 2 + 1];
79     }
80     fity = CalcFitY(x, y, length, fitm, fitb);
81     WriteArraysToFile(x, y, fity, length, "output.txt");
82
83     delete[] data;
84     delete[] x;
85     delete[] y;
86     delete[] fity;
87     return 0;
88 }

```

```

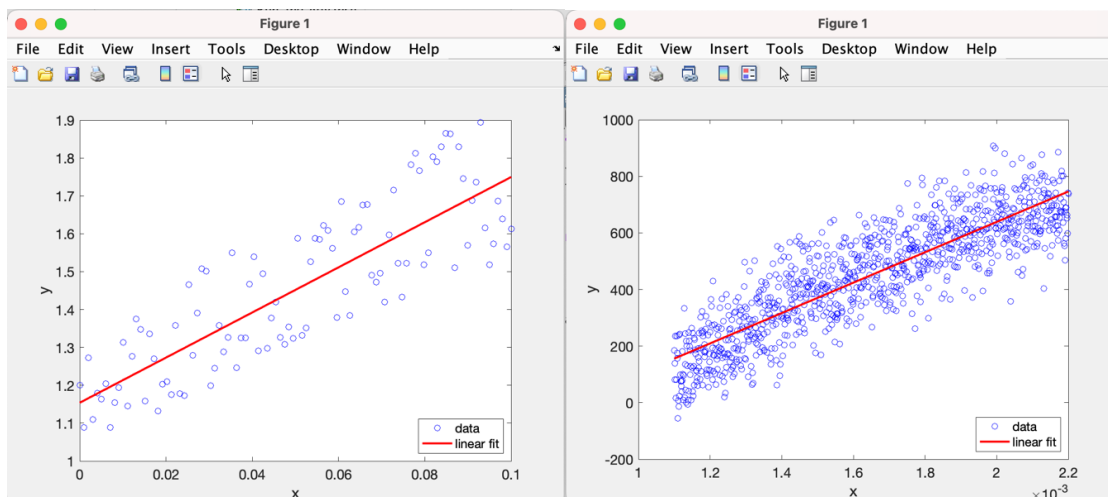
57 0.056566 1.62192 1.49099
58 0.057576 1.60917 1.49701
59 0.058586 1.56178 1.50303
60 0.059596 1.37957 1.50905
61 0.060606 1.68578 1.51507
62 0.061616 1.44677 1.52109
63 0.062626 1.38448 1.5271
64 0.063636 1.60481 1.53312
65 0.064646 1.6165 1.53914
66 0.065657 1.67608 1.54516
67 0.066667 1.67748 1.55118
68 0.067677 1.49068 1.5572
69 0.068687 1.47177 1.56322
70 0.069697 1.49639 1.56924
71 0.070707 1.41978 1.57526
72 0.071717 1.59686 1.58127
73 0.072727 1.71592 1.58729
74 0.073737 1.52286 1.59331
75 0.074747 1.43298 1.59933
76 0.075758 1.5225 1.60535
77 0.076768 1.78299 1.61137
78 0.077778 1.81234 1.61739
79 0.078788 1.76653 1.62341
80 0.079798 1.51872 1.62943
81 0.080808 1.55037 1.63544
82 0.081818 1.80389 1.64146
83 0.082828 1.79049 1.64748
84 0.083838 1.82937 1.6535
85 0.084848 1.86532 1.65952
86 0.085859 1.86369 1.66554
87 0.086869 1.51046 1.67156
88 0.087879 1.82993 1.67758
89 0.088889 1.74595 1.6836
90 0.089899 1.569 1.68961
91 0.090909 1.68819 1.69563
92 0.091919 1.73631 1.70165
93 0.092929 1.89356 1.70767
94 0.093939 1.6155 1.71369
95 0.09495 1.51871 1.71971
96 0.09596 1.57308 1.72573
97 0.09697 1.68613 1.73175
98 0.09798 1.63921 1.73777
99 0.09899 1.56626 1.74378
100 0.1 1.61231 1.7498

```

Section IV:

textCase1:

textCase2:



Section V:

My mini project was to create a C++ program to analyze data using linear regression. It read data from a file, calculated the slope and y-intercept of the best-fit line, computed the corresponding y values, and saved everything to another file. The main challenge was managing my time, especially with exams in other classes due the same week. Despite this, I got through it and successfully completed this project which was a big win for me.