Assignment 5

1301058

Zhang Junming

Exercise 1

Question

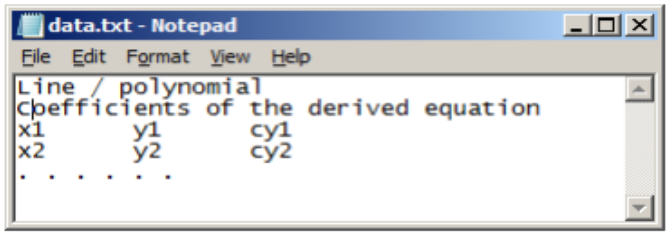
EXERCISE 1 (15 POINTS OUT OF 15)

In this part, a programme is required to fit curves, using a regression analysis (as shown in Appendix), to data sets read from a text file. A set of x,y data pairs are to be read from a file (each line of the file containing an x value and a y value separated by a space or TAB).

- The user should be able to select between fitting a straight line or a second order polynomial equation to the data pairs.
- The coefficients of the best fit curve to these points are then calculated. The programme should produce an output file and show a message to the user when it finishes a fitting process.

The output file should have the following layout

1. The first line of the file indicates the type of equation derived (line or polynomial);
2. The second line of the file contains the coefficients of the best fit curve.
3. From the third line, each line contains an original x value, an original y value and a calculated value of y from the best fit equation for the original x). The values need to be separated by a TAB. The format of the output file :



After you obtain the output data file, **open it with Microsoft Excel to plot the original discrete data and those from the fitting as a curve. You do not need to use your programme to plot the curves!**

Appendix: Curve fitting algorithms

Given a set of x,y data pairs it is often necessary to automatically calculate an equation which gives the best fit line through the data. This type of analysis is known as curve fitting or regression analysis.

The mathematical basis behind curve fitting is quite straightforward. You have a set of data pairs represented by (x1,y1), (x2,y2) ..... (xn, yn). You want an equation which represents this data, the exact type of equation will depend on the way in which the data varies (there is no point trying to fit a straight line to a set of data which is obviously not straight). Given any mathematical relationship between x and y it should be possible to generate a curve fitting algorithm. Two examples are discussed here.

Straight line

The equation for a straight line is: y = ax + b. Suppose that we have n pairs of (x,y) values from the text file, the values of a and b for the best fit line to the data pairs can be obtained from the following equation

$$a = \frac{\begin{vmatrix} b_1 & d_1 \\ b_2 & d_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} \qquad , b = -\frac{\begin{vmatrix} a_1 & d_1 \\ a_2 & d_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$$

where

$$a_1 = \sum_{i=1}^{n} (x_i * x_i), \quad a_2 = \sum_{i=1}^{n} x_i,$$

$$b_1 = \sum_{i=1}^{n} x_i, \qquad \mathrm{b}_2 = \mathrm{n},$$

$$d_1 = -\sum_{i=1}^{n} (x_i * y_i), \qquad d_2 = -\sum_{i=1}^{n} y_i$$

The value of a second order determinant $\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$ is the difference between the two cross products (a_1b_2 - a_2b_1). It is easy to create a C++ function to calculate its value.

Second order polynomial

If we wish to fit the curve: $y = ax^2 + bx + c$ to the data pairs, we can follow a similar approach to that for a straight line. The values of a, b and c for the best fit curve can be calculated from the following equation

$$a = -\frac{\begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}} \quad , \quad b = \frac{\begin{vmatrix} a_1 & c_1 & d_1 \\ a_2 & c_2 & d_2 \\ a_3 & c_3 & d_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}} \quad , \quad c = -\frac{\begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}$$

where

$$a_1 = \sum_{i=1}^{n}(x_i * x_i * x_i * x_i), \quad a_2 = \sum_{i=1}^{n}(x_i * x_i * x_i), \quad a_3 = \sum_{i=1}^{n}(x_i * x_i)$$

$$b_1 = \sum_{i=1}^{n}(x_i * x_i * x_i), \quad b_2 = \sum_{i=1}^{n}(x_i * x_i), \quad b_3 = \sum_{i=1}^{n} x_i$$

$$c_1 = \sum_{i=1}^{n}(x_i * x_i), \quad c_2 = \sum_{i=1}^{n} x_i, \quad c_3 = \mathrm{n}$$

$$d_1 = -\sum_{i=1}^{n}(x_i * x_i * y_i), \quad d_2 = -\sum_{i=1}^{n}(x_i * y_i), \quad d_3 = -\sum_{i=1}^{n} y_i$$

Solving a third order determinant is fairly simple in that the solution can be broken down to utilise a function written to solve second order determinants:

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} - b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} + c_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix}$$

You can use a C++ function to calculate the value of a third order determinant. The 9 elements in the determinant can be passed into the function by a two-dimensional array.

Model Answer

Software Development Process

1. Problem statement

This program is used to analyze and calculate a series of coordinates to fit curve. The way of analyzing called regression analysis and the detail about it has showed on the question part.

Users are asked to input the value of x and y for combining to coordinates and output these coordinates into a text file, after that, choose a way to fit curve (straight line or second order polynomial equation), finally output them to text too.

2. Analysis

Inputs:

1) Coordinates: input a series value of x and y to combine the coordinate
2) Option 1: read the coordinates which store in the text and fit them with a straight line
3) Option 2: read the coordinates which store in the text and fit them with a second order polynomial equation

Outputs:

1) A text file called data which store the entered coordinates
2) A text file called data_straight line which store the regression analysis in the way of straight line
3) A text file called data_second order polynomial which store the regression analysis in the way of second order polynomial

Additional requirements or constraint

The result should plot by using Microsoft Excel

3. Design

1. Add header file iostream and fstream
   <1> using of the ste namespace

2. Write the main function

   <1> int i j k n m - for counting
   <2> int m - store the number of coordinates
   <3> ask user to input the number of coordinates and store the value in m
   <4> double **p - construct a double dimensions array to store coordinates
   <5> allocate memory to this first dimension of array
   <6> setting up a loop to allocate memory to the second dimension of array
   <7> display and ask user to input the value of x and y to create coordinates
   <8> display these coordinates which user entered on srceen
   <9> output these coordinates to a text called data and let the value of x on the left side, the value of y on the right side

   <10> display a menu to ask user to choose how to deal with these data, 1 for straight line and 2 for second order polynomial equation
   <11> setting up a switch
      1- tell user to deal with these coordinates as straight line
         a. open text file data.txt
         b. read these coordinates which store in text data.txt and store them in a new array P1
         c. double a1 a2 b1 b2 d1 d2 - for calculate a and b in regression analysis
         d. using straight line formula to calculate

e. double a and b - coefficients of the derived equation

f. calculate each value of y with y = ax + b and store the value in array cy

g. output the type of regression analysis which is line, the coefficients a and b, the value of x, y and cy to text data_straight line.txt

2- tell user to deal with these coordinates as second order polynomial equation

a. open text file data.txt

b. read these coordinates which store in text data.txt and store them in a new array P2

c. double a1 a2 a3 b1 b2 b3 c1 c2 c3 d1 d2 d3- for calculate a and b in regression analysis

d. using straight line formula to calculate

e. double a and b - coefficients of the derived equation

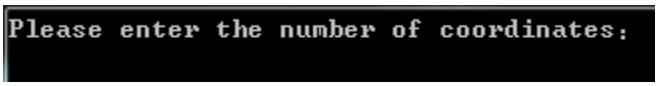
f. calculate each value of y with y = ax^2 + bx +c and store the value in array cy

g. output the type of regression analysis which is line, the coefficients a and b, the value of x, y and cy to text data_second order polynomial.txt

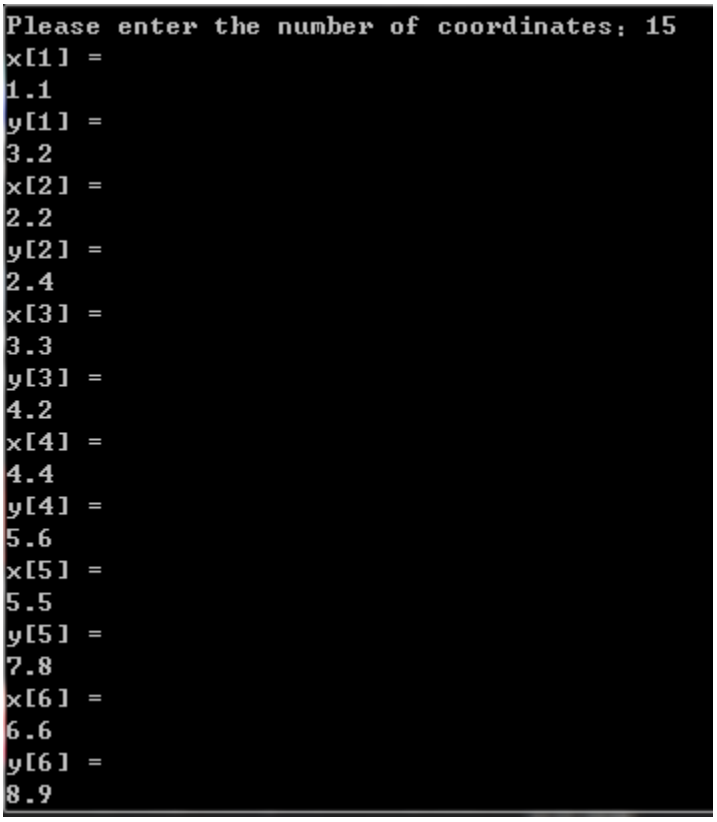
<12> write default case, tell user enter error

<13> return 0

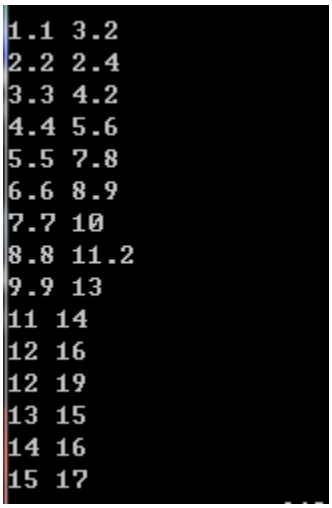
4. **test**:



at start, ask user to confirm how many coordinates will be entered



the process of inputting



after input, the result will be displayed

```
output at a text file named data.txt
Please choose the type of fitting these data
1-straight line
2-second order polynomial equation
```

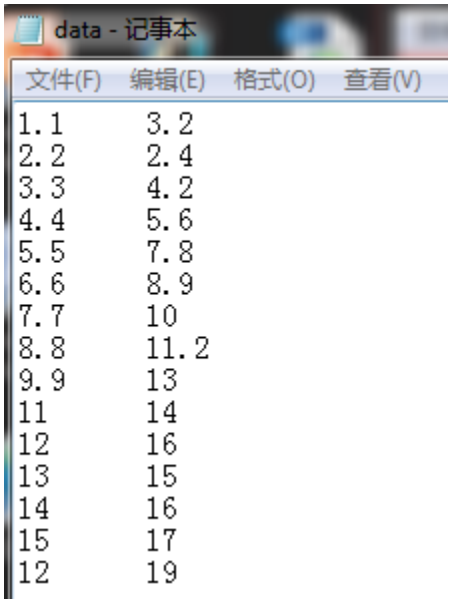
menu

```
1-straight line
2-second order polynomial equation
1
 For straight line
output at a text file named data_straight.txt
```

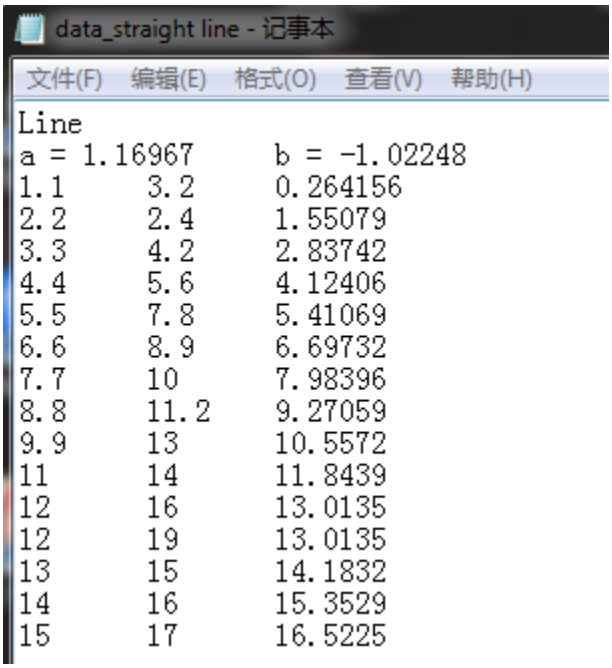
the way of straight

```
1.1 3.2
2.2 2.4
3.3 4.2
4.4 5.6
5.5 7.8
6.6 8.9
7.7 10
8.8 11.2
9.9 13
11 14
12 16
13 15
14 16
15 17
12 19
output at a text file named data.txt
Please choose the type of fitting these data
1-straight line
2-second order polynomial equation
2
 For second order polynomial equation
output at a text file named data_second order polynomial.txt
```

the way of second order polynomial equation



text file of data



text file of straight line

data_second order polynomial - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
polynomial
a = 0.0225508    b = 1.53102      c = -0.00462459
1.1       3.2       1.70679
2.2       2.4       3.47277
3.3       4.2       5.29332
4.4       5.6       7.16845
5.5       7.8       9.09815
6.6       8.9       11.0824
7.7       10        13.1213
8.8       11.2      15.2147
9.9       13        17.3627
11        14        19.5653
12        16        21.6149
13        15        23.7097
14        16        25.8496
15        17        28.0346
12        19        21.6149
```

test file of second order polynomial

Plot in EXCEL

	A	B	C	D
1	x	y	cy1	cy2
2	1.1	3.2	0.264156	1.70679
3	2.2	2.4	1.55079	3.47277
4	3.3	4.2	2.83742	5.29332
5	4.4	5.6	4.12406	7.16845
6	5.5	7.8	5.41069	9.09815
7	6.6	8.9	6.69732	11.0824
8	7.7	10	7.98396	13.1213
9	8.8	11.2	9.27059	15.2147
10	9.9	13	10.5572	17.3627
11	11	14	11.8439	19.5653
12	12	16	13.0135	21.6149
13	13	15	14.1843	23.7097
14	14	16	15.3529	25.8496
15	15	17	16.5225	28.0346
16	12	19	13.0135	21.6149
17				
18