

4.2 Fitting Functions Using *Mathematica*

We will now look at some examples to see how the functions in the package **DataFit** work. Open a new *Mathematica* notebook and select **File** → **Palettes** → **DataFitP** to open the corresponding palette. Remember to move it so that it does not overlap with your notebook window; this ensures that your notebook remains the active window when using the palette.

Always, **the first step is to load the package**. Click on the topmost button to paste `<<DataFit`` into your notebook. Check that the notebook is the active window. (The top bar of the window is colored under Windows 95, striped under the Mac OS.) Move the cursor into the cell containing `<<DataFit`` and click the mouse button. Evaluate the cell and check that you see an `In[#]` tag in front of `<<DataFit``. This is your confirmation that the computational engine is aware of all the functions in the package. If you do not see the `In[#]` tag, check whether the notebook is the active window and repeat the procedure.

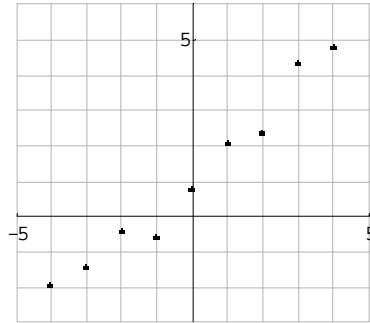
The DataFit palette consists of seven groups of functions besides the topmost button:

<code><<DataFit`</code>	
<code>PlotData</code> <code>ShiftInput</code> <code>ShiftOutput</code>	data visualization and manipulation
<code>FirstUnitDiff</code> <code>SecondUnitDiff</code> <code>PolyFitGraph</code> <code>PolyFitFunc</code>	tools for fitting a polynomial
<code>UnitRatios</code> <code>ExpoFitGraph</code> <code>ExpoFitFunc</code>	tools for fitting an exponential function
<code>LogisticFitGraph</code> <code>LogisticFitFunc</code>	tools for fitting a logistic function
<code>SineFitGraph</code> <code>SineFitFunc</code>	tools for fitting a sine function
<code>FitComp</code>	comparison tool for different types of functions
<code>/.{x → }</code> <code>xplot → { , }</code> <code>yplot → { , }</code>	mechanism to evaluate fitted functions

Before we can use *Mathematica* to find the best fitting function, we have to determine the possible type(s) of function(s) by using the relevant palette functions for graphing and for numerical tests. Then, we can use the appropriate palette function(s) to compute the equation for

the least squares fit function and to see its graph displayed together with the data. For example, in Section 4.1, we wanted to find the best fitting straight line to the given data.

x	y
-4	-1.932
-3	-1.442
-2	-0.408
-1	-0.566
0	0.772
1	2.080
2	2.382
3	4.361
4	4.768



In a new notebook cell, type "A =" (this will give the data table a name). Then select **Input** → **Create Table/Matrix/Palette** from the menu bar. Make sure that "Table" is checked and enter the number of rows (= 10) and the number of columns (= 2). Now you can enter the data, starting with x and y in the first row. Remember to type x and y with quotation marks. Once you have finished entering the data, evaluate the cell.

Our goal is to fit a linear function to this data. Recall that a linear function is a polynomial of degree one. The palette functions **PolyFitGraph** and **PolyFitFunc** fit polynomials to data. Their input is a list of data pairs and the degree of the polynomial. **PolyFitGraph** displays the graph of the fitted function together with the data, whereas **PolyFitFunc** gives the equation of the fitted polynomial. You can check this by using

?PolyFitGraph

`PolyFitGraph[list,n]` fits a polynomial of degree n (linear $n=1$, quadratic $n=2$, cubic $n=3$) to the data given in `list` as pairs of input/output values (the first entry in `list` may consist of the names for the variables). The data and the graph of the fitted function are displayed together.

and

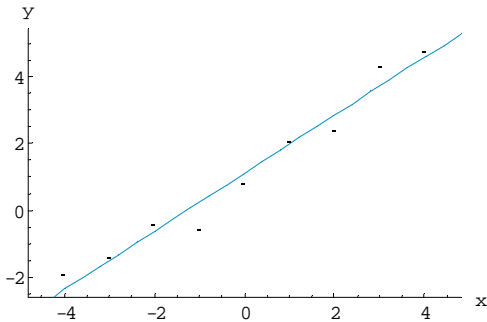
?PolyFitFunc

`PolyFitFunc[list,n]` gives the fitted polynomial of degree n (linear $n=1$, quadratic $n=2$, cubic $n=3$) to the data given in `list` as pairs of input/output values (the first entry in `list` may consist of the names for the variables).

Let's start with the graph of the fitted polynomial. Make sure you see a horizontal line before clicking on the palette button for **PolyFitGraph** (to paste the command into a new cell). Once the function is pasted into the notebook, fill the first place holder with the name of the table and the second place holder with 1 (for linear fit).

```
PolyFitGraph[A,1]
```

Evaluate the cell to see the best fitting straight line displayed together with the original data.



To get the equation for this best fitting linear function use **PolyFitFunc**.

```
PolyFitFunc[A,1]
1.11278+0.873917x
```

Mathematica displays real numbers by default with six significant digits (not to be confused with six decimal places!). We can read off the values for the multiplicative constant $a = 0.873917 \approx 0.874$ and the additive constant $b = 1.11278 \approx 1.113$ to three decimal places. (Note that this is the function used in Activity 4.1.1.)

We will now use the example at the beginning of Chapter 3 to do a full analysis and in the process see how some of the other functions from the **DataFit** palette are used. Here are the data and the questions posed:

Year	1960	1965	1970	1975	1980	1985	1990	1995
Median Age	20.3	20.6	20.8	21.1	22.0	23.3	23.9	24.5

What will the median age of first marriages be in the year 2010? When will the median age of first marriages reach 30 years?

⇒⇒⇒ Please open the *Mathematica* file L4.nb or refer to the section on *Mathematica* lessons.

Now that you have had a full tour of the functions in the DataFit package, here are two activities where you need to find a model (= fitted function) for the given data in order make predictions.

Activity 4.2.1

The table below gives the death rate (number of deaths per thousand people) for Americans at different ages.

Age	Death Rate
40	2.2
45	3.0
50	5.0
55	8.0
60	12.6
65	18.7

- Enter the data into a notebook and graph it using the function **PlotData**. From the shape of the graph determine which type of function should be fitted. If more than one type is possible, list them all. Give reasons for your answer.
- Use the appropriate numerical tool (if available) for each of the function types of part a) to determine whether they fit the data. List any function type(s) that are still good candidates.
- For each type of function listed in part b), find the best fitting function (both graph and functional expression).
- Use the palette function **FitComp** to check how good the best fit is for each function type. If more than one possible function type remained in part c), determine which type is the best one overall.
- Does the function type you fitted make sense in the context of the problem? If several function types were fitted, which one is most likely in the context of the problem? Give reasons for your answer.
- Use the best fitting function that makes sense in the context to predict the death rate for a 5 year old and a 100 year old person. Do your predictions makes sense?

Activity 4.2.2

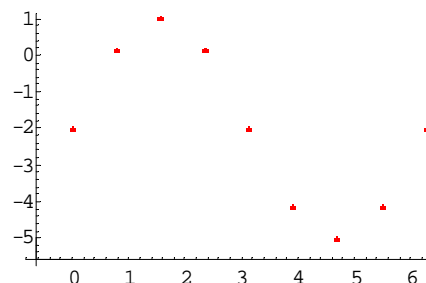
The table below gives the number of European, North American and South American countries issuing postage stamps from 1840 through 1880.

Years	1840	1845	1850	1855	1860	1865	1870	1875	1880
Countries	1	3	9	16	24	30	34	36	37

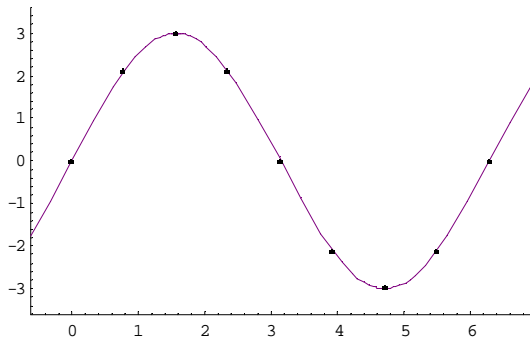
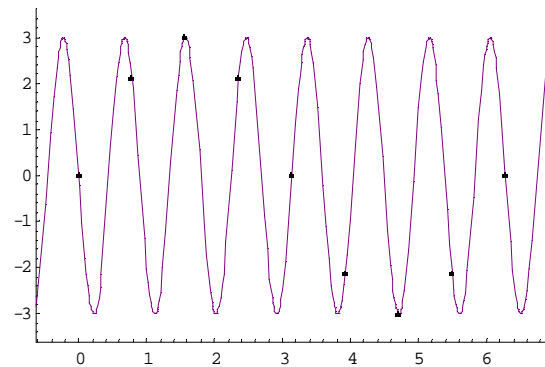
- Identify the meaning of the input and output variable.
- Enter the data in a notebook and graph it using the function **PlotData**. From the graph, determine which function type(s) are possible matches for the data. Give reasons for your answer.
- If numerical tools are available for the function type(s) identified in part b), use them to validate your guess(es). Otherwise, explain from the shape and /or context why a specific function type is appropriate.
- Fit all of the function types identified in part c) to the data. If more than one function type matches, use **FitComp** to compare the fits of the different types.
- Use the best fitted function to predict the number of countries issuing postage stamps in the year 2000. How good is your prediction? (Think of ways you might determine the actual number of stamp issuing countries for the current year.)

In the examples so far, we never had to fit a sine function. For sine functions, the procedure is a little more difficult, as we need to give *Mathematica* more assistance, namely the approximate value of the period p . If the value entered into **SineFitGraph** or **SineFitFunc** is too far off, you may see “strange” results. Here is an example of what might happen - make sure you read off the period correctly and check the resulting fitted graph to see whether it seems appropriate. We will use the data given in the example in Section 3.6, where we determined the period to be **6.283**.

x	$f(x)$
0	-2
0.785	0.121
1.571	1
2.356	0.121
3.142	-2
3.927	-4.121
4.712	-5
5.498	-4.121
6.283	-2



Open a *Mathematica* notebook and enter the data as a table named **A**. We will use **SineFitGraph** with the roughly correct period of 6, and also with an incorrect period of 1. Here are the results:

SineFitGraph[A,6]**SineFitGraph[A,1]**

In both cases we get a sine function that passes through all the data points exactly, i.e., the total squared error is zero in both cases, as can be illustrated using **FitComp**:

```
FitComp[A, {SineFitGraph[A,6], SineFitGraph[A,1]}]
```

```
f1(x) = 3.Sin[1.x]
```

```
Total Error for f1(x) = 0
```

```
f2(x) = -3.Sin[3.14159 - 7.x]
```

```
Total Error for f2(x) = 0
```

What is going on? Even though the sine function in the second graph matches all the data points, the first graph seems more natural, as we tend to connect points smoothly, without major changes between points. However, mathematically the second sine function is just as good a fit as the first, as both match the points exactly. Often, the context of a problem can shed light on the “correct” period. For example, if the data gives the mean temperatures in a city for each month, one would expect a cycle (= period) of 12 months, not one month. The lesson here is that *Mathematica* does exactly what you tell it to do - the program cannot think and decide whether the period given in **SineFitGraph** or **SineFitFunc** makes sense.

To conclude this section on fitting functions to data, the following activity gives a variety of data for you to determine an appropriate model.

Activity 4.2.3

- a) For each set of data, enter the data into a notebook and graph it.
- b) Determine which function types are appropriate using the shape of the graph, numerical tests (if available), and the context.
- c) Fit the type of function that is most appropriate. If several function types are possible use **FitComp** to determine the best fit. Discuss whether you found a good model (= fitted function).

i)

Price (dollars)	Volume (liters)
0.50	0.3
1.00	1.0
1.50	2.25
2.00	4.0
2.50	6.3
3.00	9.2

ii)

Four week period	Soup sales (millions of 16 ounce cans)
1	200
2	215
3	180
4	160
5	150
6	120
7	110
8	100
9	110
10	130
11	155
12	200
13	210
14	200

iii)

Time since drug was given (hours)	Amount of drug remaining
1	575
3	374
5	240
7	153
9	98
11	63

iv)

Length (feet)	Width (feet)
20	25.0
22	27.5
24	30.0
25	31.3
27	31.0
29	30.0
32	27.8
34	25.3

Chapter Review

Key Terms

error	absolute error	squared error
total absolute error	total squared error	method of least squares fit

Short Answers

1. Why do we use either the absolute error or the squared error to measure how much the data values differ from the fitted function?
2. What does the method of least squares provide in the case of a straight line fit?
3. Describe the basic idea of the method of least squares fit.
4. If you are given the equations of several potential functions together with the data, how can you determine which of these functions fits best to the data?
5. How is the total squared error computed?

True - False Questions

- T F 1. The total squared error is the square of the total absolute error.
- T F 2. The method of least squares fit gives the equation or the values of the constants of the best fitting function of a given type.
- T F 3. There may be more than one best fitting function of a given type.
- T F 4. The method of least squares fit works only for straight lines.

Fill in the Blanks

1. The method of least squares fit uses methods from _____ to find the values of the constants of the best fitting function of a given type.
2. The error $\Delta = y - f(x)$ is positive if the data point is _____ the fitted function.