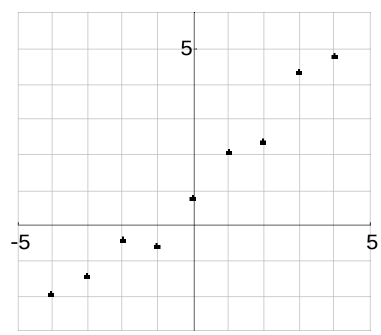
4. Fitting a Function to Data

In the last section we saw how we can determine which type of function fits the shape of the given data. Now we need a way to decide which of the many possible functions of a specific type is closest to the given data. The question now becomes: How do we measure “closeness”? One answer to this question is to use the method of least squares, which will be discussed in Section 4.1. Section 4.2 will give examples on how to use the package DataFit to fit functions to given data with the assistance of the computer program *Mathematica*.

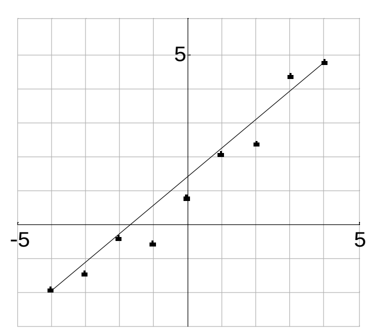
4.1 The Method of Least Squares Fit

We will look at an example to see how the method of least squares fit works. Below is a table of data, together with the corresponding graph:

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



From the graph, it seems that the best function type is linear, thus $f(x) = ax + b$. To draw a straight line based on the data, we can connect for example the two endpoints. This results in the following graph:



For this particular straight line, we can use the two endpoints $(x_1, y_1) = (-4, -1.932)$ and $(x_2, y_2) = (4, 4.768)$, respectively, to compute the values of a and b. Therefore, using the formula for the slope of a straight line,

$$\text{slope } a = \frac{y_1 - y_2}{x_1 - x_2} = \frac{-1.932 - 4.768}{-4 - 4} = \boxed{0.8375}$$

To solve for the intercept *b*, we substitute one of the points and the value of the slope *a* into functional expression of a straight line, $f(x) = a \cdot x + b$, then solve for *b*. Substituting $(-4, -1.932)$ for *x* and *f*(*x*), respectively, as well as *a* = 0.8375 gives

$$-1.932 = 0.837 \cdot (-4) + b.$$

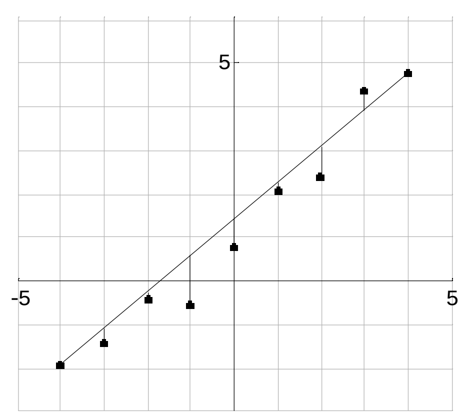
Solving for *b*:

$$-1.932 - (0.8375)(-4) = b$$
$$-1.932 + 3.35 = b$$
$$1.418 = b$$

Thus, the function describing our specific straight line is given by

$$f(x) = 0.8375x + 1.418\,.$$

Once we have computed the equation for the straight line, we can compute how far off the data values are from this line. We use the vertical distance between the data point and the line we drew as measure of the error, as illustrated in the graph below:



To compute this vertical distance, we need to find the output value of the point on the line that is directly above or below a given data point. This means the two points (the one on the line and the data point) have the same value for the independent variable (or input value) *x*. Thus, the first step is to compute the function values for the straight line, using the formula derived above.

Data		Straight line
x	y	$f(x) = 0.8375x + 1.418$
-4	-1.932	-1.932
-3	-1.442	-1.0945
-2	-0.408	-0.257
-1	-0.566	0.5805
0	0.772	1.418
1	2.080	2.2555
2	2.382	3.093
3	4.361	3.9305
4	4.768	4.768

Note that in this example, the data values and the point on the straight line coincide for x = -4 and x = 4. This is no coincidence, as these are the two endpoints which were used to find the equation of the straight line. Whenever we derive the equation of a line that passes through certain points, the function value for the straight line and the output value of the data will coincide for that point.

The second step in the procedure is to compute how far a data value is from the line by computing the *error*, $\Delta = y - f(x)$. This will give positive errors if the data point is above the straight line and negative errors if the data point is below it. If we add the errors for all the input values, cancellations of positive and negative errors can make the total error very small, giving an overly optimistic picture. Thus, we should only look at the magnitude of the error, and not its sign.

The easiest way to accomplish this is by using the *absolute error*, $|\Delta|$, which gives the distance of the data point from the line. Another possibility is to use the *squared error*, Δ^2. Let's try both methods, using the value x = -2 as example:

$$\Delta = y - f(x) = y - f(-2) = -0.408 - (-0.257) = \mathbf{-0.151}$$

$$|\Delta| = |-0.151| = \mathbf{0.151}$$

$$\Delta^2 = (-0.151)^2 = \mathbf{0.022801}$$

Likewise, we can compute the quantities Δ, $|\Delta|$ and Δ^2 for all the input values (as shown in the table below). To compute the *total absolute error* or the *total squared error*, we add up the quantities in the columns for $|\Delta|$ and Δ^2, respectively.

Data		Fitted Function: $f(x) = 0.8375x + 1.418$			
x	y	$f(x)$	$\Delta = y - f(x)$	Absolute error $\lvert\Delta\rvert$	Squared error Δ^2
-4	-1.932	-1.932	0	0	0
-3	-1.442	-1.0945	-0.3475	0.3475	0.12075625
-2	**-0.408**	**-0.257**	**-0.1510**	**0.1510**	**0.02280100**
-1	-0.566	0.5805	-1.1465	1.1465	1.31446225
0	0.772	1.418	-0.6460	0.6460	0.41731600
1	2.080	2.2555	-0.1755	0.1755	0.03080025
2	2.382	3.093	-0.7110	0.7110	0.50552100
3	4.361	3.9305	0.4305	0.4305	0.18533025
4	4.768	4.768	0.0000	0.0000	0.00000000
			Total Error:	3.608	2.59698700

Notice that Δ^2 has twice as many decimal places as Δ. We will keep them for now, but after adding all the errors, we will round our final answer to three decimal places, which is the number of decimal places of the original data.

Recall the **Rule for Rounding**:

> If you want to round to three decimal places, look at the digit at the <u>fourth</u> decimal place to the right of the decimal point. There are two cases to consider. If the fourth digit is between 0 and 4 (inclusive), then round down, i.e., drop all the digits to the right of the third digit. If the fourth digit is between 5 and 9 (inclusive), then round up, i.e., drop the digits to the right of the third digit and increase the third digit by one. (Note that this may result in carry-overs, which have to be completed.)

Applying this rule, the total square error of 2.596<u>9</u>87 in the example above is rounded up to 2.597.

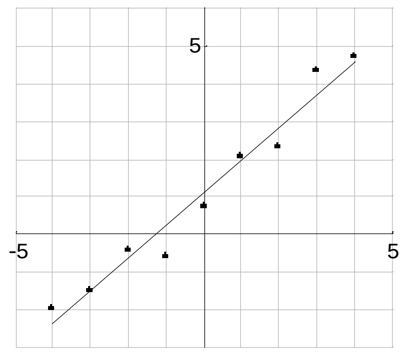
After computing the total squared error, we have to decide whether the fit is good enough. When we tried to find a straight line close to the data, we used one that passed through the two endpoints. However, we could have used other straight lines, such as the one given in Activity 4.1.1 below. As we can see from the graph, this straight line, given by $f(x) = 0.87x + 1.1$, is quite close to all the data points, even though it does not actually pass through any of them. To decide which of the two straight lines is closer overall to the given data, we need to compare their total errors; the function with the smaller total error is the one that fits the data better.

Activity 4.1.1

Here is another possible fit for the given data, namely the line given by $f(x) = 0.87x + 1.1$, whose graph is displayed below together with the data.

a) Complete the table of values.

b) Use the total error to determine whether this straight line is a better fit (i.e., closer to the data) than the one used in the previous example. Give a reason for your answer.



Data		Fitted Function			
x	y	$f(x)$	$\Delta = y - f(x)$	Absolute error $\lvert\Delta\rvert$	Squared Error Δ^2
-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				
			Total Error:		

Using the value of the total errors, we can compare any two straight lines to determine which one fits the given data better. However, there are many potential straight lines; do we have to do this pair-wise comparison ad infinitum? No, we don't. The *method of least squares fit* uses methods from Calculus to find the **unique** values of a and b that give the closest straight line fit to the given data. The least squares fit formulas for a and b are given by

$$a = \frac{n \cdot \left(\sum x \cdot y\right) - \left(\sum x\right)\left(\sum y\right)}{n \cdot \left(\sum x^2\right) - \left(\sum x\right)^2} \text{ and } b = \frac{\left(\sum y\right)}{n} - a \cdot \frac{\left(\sum x\right)}{n}$$

where $n =$ the number of data points and $\sum x$ is an abbreviation for the sum of all the input values. Likewise, $\sum y$ is the sum of all the output values, $\sum x \cdot y$ is the sum of all the products of input and output values, and $\sum x^2$ and $\sum y^2$ are the sums of the squares of the input and output values, respectively.

We will use the data given at the beginning of this section to illustrate the computation of *a* and *b*. Since we need to sum up various quantities (*x*, *y*, $x \cdot y$, x^2, and y^2), it is easiest to make a table. In the first two columns of the table we list the input and output values. Next, we compute the columns containing the products of *x* and *y*, as well as the squares of the input and output values. Finally, each column is summed and the resulting values are substituted into the formula.

x	y	$x \cdot y$	x^2	y^2
-4	-1.932	7.728	16	3.732624
-3	-1.442	4.326	9	2.079364
-2	-0.408	0.816	4	0.166464
-1	-0.566	0.566	1	0.320356
0	0.772	0	0	0.595984
1	2.08	2.08	1	4.3264
2	2.382	4.764	4	5.673924
3	4.361	13.083	9	19.01832
4	4.768	19.072	16	22.73382
$\sum x = \mathbf{0}$	$\sum y = \mathbf{10.015}$	$\sum x \cdot y = \mathbf{52.435}$	$\sum x^2 = \mathbf{60}$	$\sum y^2 = \mathbf{58.64726}$

Since we have 9 data points, *n* = 9. Substituting into the formula, we get

$$a = \frac{n \cdot \left(\sum x \cdot y\right) - \left(\sum x\right)\left(\sum y\right)}{n \cdot \left(\sum x^2\right) - \left(\sum x\right)^2} = \frac{9 \cdot 52.435 - 0 \cdot 10.015}{9 \cdot 60 - 0^2} = \mathbf{0.874}$$

$$b = \frac{\left(\sum y\right)}{n} - a \cdot \frac{\left(\sum x\right)}{n} = \frac{10.015}{9} - 0.874 \cdot \frac{0}{9} = \mathbf{1.113}.$$

As you can see, this formula is tedious to do by hand, but most graphing calculators and computer algebra systems such as *Mathematica* or *Maple* have this function built-in.

The method of least squares fit can be adapted to the case where one wants to fit say a quadratic function. Again, the vertical distances between the data points and the corresponding points on the actual function are measured. Then Calculus methods are used to find the values of the constants that make the error the smallest. The formulas for the constants for quadratic, cubic, exponential, logistic, and sine functions are different from the one given for the straight line fit (= linear function), but the principle is the same.

The *Mathematica* package **DataFit** allows us to compute the best fitting function of a given type for all the types discussed in Chapter 3. We can also display the graph of the best fitting function together with the original data. In addition, we can compare the best fitting functions of different types (such as quadratic and exponential) to find the best fitting function overall.