Data and Error Analysis

Ryan D. Martin

December 24, 2015

Contents

1 Introduction

This book looks at the transition between the physics in textbooks, where everything is exact, to physics in the real world, where nothing can be measured with infinite precision. We thus need a robust framework in which we can apply the laws of physics to the real world, and to test them. We also need a framework that allows us to gather data, understand what the data mean, and then make robust conclusions about what they imply.

1.1 Motivation - "doing good science"

When we "do science", we are really interested in modelling the real world. At the basis of the Scientific Method is the idea that we can form a hypothesis (a "theory") and then test if that hypothesis is wrong (an "experiment").

For example, an experimenter way want to test the hypothesis that the circumference of a circle is $2\pi r$. Suppose that they present the following report of their experiment:

We used a ruler to measure the radius of a circle and obtained 1 cm. We wrapped a string around the circle and determined the length of the string to be 6.4 cm. The theory predicts that the circumference should be 2π(1 cm)=6.28318530718 cm, which is clearly not equal to 6.4 cm. The theory is thus invalidated.

This trivial example already uncovers a number of aspects that are important in "doing good science". Based on what is stated above, can you decide whether the theory is invalidated by the experiment? I certainly cannot, even though I have a strong belief that the theory is not invalidated by this experiment. Here are a few questions that I would want the experimenter to address:

- What are the uncertainties on the measurements of the radius and circumference, and how were they determined?
- What is the precision of the ruler?
- Are the digits presented for the predicted value significant?

- How was the radius determined?
- Can the experiment be performed with a much bigger circle? (why would you ask this?)
- What was the diameter of the string?
- How was the ruler calibrated?
- Where was the value of π taken from?

As you can see, you need to be very precise when you do science. Obviously you should be as precise as possible when performing the measurements, but you should be equally precise in how you report your experimental results. When you write lab reports, or later scientific publications or presentations to your peers, always ask yourself how one could question the validity of your results and then address those concerns.

The subject of this book is "Data and Error Analysis", and the goal is to give you a strong foundation for analysing your data and presenting them[1] in a way that is scientifically useful. I should note that the word "error" in our context is used as a synonym for "uncertainty" rather than "mistake". The idea of "error analysis" is to understand the uncertainties that are involved in the presentation of a result (and to a much lesser extent the tedious "propagation of errors" that are so feared in undergraduate laboratories).

EXAMPLE 1-1: How would you present the results of the experiment attempting to test the hypothesis that the circumference of a circle is $2\pi r$?

Here is a possible way to present and discuss the results:

Experimental procedure and uncertainty determination
We used a stainless steel compass (brand: Acme, model: Pro2000) with a 0.5 mm diameter graphite tip to draw 10 1 cm radius circles onto a set of small post-it notes. The radii of the circles were measured using a stainless steel ruler (brand: Acme, model: RRCatcher2000) with 1 mm graduations. The ruler was used to set the stride of the compass, and was then again used to confirm the radius of the drawn circles (by measuring the distance between the hole made by the compass into the paper and the perimeter of the circles).

The radius of the circles was determined to be 1.0 ± 0.08 cm. We found that the mean measured radius of the 10 circles was 1 cm with a standard deviation of 0.08 cm, which we use as central value and uncertainty. We believe this uncertainty is representative of the fact that the circles have a width of 0.5 mm (due to the size of the graphite tip on the compass) and that the ruler has 1 mm graduations.

We then used pieces of string (brand: Acme, model: CatCatcher2000, diameter: 0.8 mm, material: wool) to measure the circumferences of the 10 circles, by wrapping them around the circles, cutting them to length, and then measuring their lengths with the ruler. We obtained a mean value of 6.4 cm with a standard deviation of 0.12 cm (which we use as the central value and uncertainties on the measurement of the circumference) for the 10 circles. The larger variation in the length of the string as compared to that of the radius is likely due to some stretch in the string, the inability to exactly lay it over the circumference, and the inaccuracy in cutting it to the exact length.

[1] data is the plural of datum, eh

Using the accepted value of π from Wikipedia [2], the predicted value of the circumference is:

$$2\pi(1.0 \pm 0.08)\ cm = (6.3 \pm 0.5)\ cm$$

We find the predicted value above to be consistent with the measured value of (6.4 ± 0.12) cm. The hypothesis that the circumference of a circle is $2\pi r$ is thus verified within the accuracy of this experiment. Unfortunately, we only had small post-it notes, limiting the size of the circles that we could draw to 1 cm radius. We recommend performing this experiment with much larger circles to reduce the relative uncertainties in the measurements of the radius and circumference.

Comment: *Even this more "precise" report on the experiment has holes in it (were the uncertainties in the individual 10 measurements used? What are the 10 values that were measured, were there any outliers? Is using the standard deviation as the uncertainty justified? How was the diameter of the graphite tip and string determined? How was the error in the radius propagated to the uncertainty in the predicted value? Was the ruler calibrated?)*

1.2 Motivation - the case for using computers

As we have seen, an effective way to understand the uncertainty in a measurement is often to perform the measurement multiple times. In realistic scenarios, this can lead to a lot of data to analyse, which can lead to painful calculations if performed by hand. Even calculating the standard deviation of 10 measurements is not particularly pleasant to do with a calculator, even when it has a function to do so. Furthermore, we often collect data in digital form (e.g. a piece of electronic apparatus recording data to files) and it would be an ineffective use of time to then transcribe those data to analyse them by hand when they are already in a form that can be handled by a computer. Finally, the propagation of errors often involves rather painful formulas with derivatives, squares, square roots that can be handled much more easily and with less chances of making mistakes with a simple computer program. Hopefully, I do not need to convince you further that learning a little programming will make your life a lot easier in the long run, as well as open up more job opportunities.

In this course, we will use the python programming language, which is easy to learn and widely accepted. Most data scientists now use python, it is a de-facto requirement in most technology start-up companies and is heavily used throughout physics. The language is also open source and free, and it is extremely easy to find online help due to its wide acceptance. The next chapter gives a small introduction to python, to give you a basis for this course. You are welcome to use python in any form for this course, but we will provide support for using it through Jupyter Notebooks.

[2] don't actually use Wikipedia!

2 Python

This chapter is meant to serve as an introduction to programming in python and to be used as a reference. Do not worry about memorizing all of the commands, this will come automatically with practice.

2.1 The basics

This section covers some of the basics to get you started.

2.1.1 Variables and the assignment operator

In a computer program, one uses "variables" and assigns them values. A variable can be thought of as an address inside of the computer's memory that can store a value. If we want to put a value into the computer's memory, we first create a variable and then "assign" that variable the desired value. We can then access the value that we stored by telling the program that we are interested in the value that is stored in a particular variable. This is the most basic operation in writing a computer program. In python, the syntax is the following:

```
a = 2
```

In the above, we have created a variable, `a`, and assigned it the value of 2. The equal sign is called the **"assignment operator"**, and it always "puts the thing on the right **into** the thing on the left". Notice that python automatically recognizes that we are storing a number into the variable `a`, and python will thus treat `a` like a number. The following code thus has the expected behaviour and the new variable `b` will be equal to the number 4.

```
a = 2
b = 2*a
```

2.1.2 Printing to screen

Now, you may not believe that the variable `b` above is actually equal to 4. Luckily, you can ask the computer program to "print" out the value that is stored in a variable, by using the `print()` function:

```
a = 2
```

```
b = 2*a
print(b)
```

This will print the value of 4 in the output of the program. `print()` is the first example of a function. Functions can take any number of "arguments" and then act on them. In this case, we "passed" the variable `b` to the function `print()` (by putting it into the parenthesis) and then the function took care of printing that to the output. The `print()` function is quite flexible, and can take any number of arguments of different types, separated by commas:

```
a = 2
b = 2*a
print("The value of a is ",a," and the value of b is ",b)
```

In this case, we passed 4 different arguments to the `print()` function. The second and fourth arguments, our variables `a` and `b`, are numbers. The first and third arguments are what we call "strings" (strings of characters), which we must include in quotation marks, so that python doesn't think that these are variables.

2.1.2.1 Formatting numerical output

Python cannot guess how many decimals it makes sense to display when you print a number; often, you want to control this. This can be done using the `format` "method".

```
a = 2
b = 3/2
c = 2/3
print("first={:.1f}, second= {:.2f}, third={:.4f}".format(a,b,c))
```

The `format` method works as follows. You first define a "template string" (in the above case, `"a={:.1f}, b= {:.2f}, c={:.4f}"`) with quotation marks. You then add a `.format()` to the end of it. The `format()` method will look inside of the string for occurrences of curly brackets ({}). Inside of the curly brackets, you can tell it what format you would like for the number: in our case `{:.nf}` means that we would like to display n decimals (.n) of a floating point number (f). Thus, `{:.4f}` displays 4 decimals. The final component, is that one tells `format()` which numbers we actually want to format that way, by passing those numbers as arguments to `format()` in the same order in which we want them to appear. What really happens is that `format()` sequentially replace the curly brackets with the "formatted version" of the numbers that it is given (in our case, the variables a,b,c). The output of the above command is thus:

```
first=2.0, second= 1.50, third=0.6667
```

2.1.3 Comments

As code becomes more complex, it is important to be able to insert some comments so that we can remember what we did (or so that others can understand what your code is doing). In python, comments are preceded with the number (or pound) sign '#'[1], and anything that follows will be ignored by the computer:

```
#Here we have a simple program
a = 2 #a is set to 2
b = 2*a #b is twice a
```

[1] This is not called a hashtag!

```
#Now we print out the result:
print("The value of a is ",a," and the value of b is ",b)
```

Always comment your code. You cannot over comment! This is especially important if you are having your code graded in an assignment; if your code is wrong, but one can understand the intention from the comments, you may get significantly higher grades from partial marks.

2.1.4 Lists

A particularly useful type of variable in python is the "list". This is analogous to "arrays" in other programming languages. A list is used to hold, well, a list of values. For example, let's say that we wish to work with a set of measurements, for example radii of circles from example 1-1, and hold them into a single variable, `radii`, we can use a list as follows:

```
radii=[0.9,1.0,0.95,1.1,1.2,1.0,0.8,0.85,1.05,1.0]
```

Python automatically recognizes that the variable `radii` should be a list, since we are trying to assign a set of 10 comma-separated values to it. We can work with the whole list, but we can also get a particular element out of the list, by using the access operator ('[ ]') and the index of the element:

```
radii=[0.9,1.0,0.95,1.1,1.2,1.0,0.8,0.85,1.05,1.0]
print(radii[2])
```

One thing to note is that the above `print()` function printed out the third element of the list and not the second one. This is because **python indexes the first item in the list with the index 0**. If the list has `n` elements in it, then the last element has the index `n-1`. If you try to ask for an element that is beyond the size of the list, e.g. `radii[10]`, bad things happen. This is because you are really asking for a value stored in the memory of the computer in a location where you have not stored anything.

A useful operation on a list is to ask how many elements are in the list; this is done with the `len()` function:

```
radii=[0.9,1.0,0.95,1.1,1.2,1.0,0.8,0.85,1.05,1.0]
print("We measured ",len(radii)," radii")
```

2.1.4.1 Operations on lists

The code below illustrates a variety of operations that can be performed on lists, such as inserting and removing values. Try it for yourself and make sure you understand what is going on.

```
#a list of numbers
array=[1,2,3,4,5,6]
#len(array) is the number of elements in the list
print("The size of the list is ",len(array))
#lists start a 0
print("Element 1 of the list is ",array[1])
#You can print a list:
print("The list is: ", array)
#You can append an element to the end:
array.append(7)
print("The list is now :",array)
#Remove the 5th element
array.pop(5)
```

```
print("The list is now :",array)
#Remove the last element
array.pop()
print("The list is now :",array)
#Insert the number 9 at position 3
array.insert(3,9)
print("The list is now :",array)
#Reverse the elements in the list
array.reverse()
print("The list is now :",array)
#Sort the list
array.sort()
print("The list is now :",array)
#You can mix variable types in the list (even including other lists)
mixedarray=["hello",42,array]
print("The mixed array ",mixedarray)
print("Some elements from it :", mixedarray[0],
      mixedarray[1],mixedarray[2][3])
#You can slice a list to generate a new list
subarray=array[2:4]
print("The sub list: ",subarray)
```

I will point out that in the second last example, we are accessing a list using two indices. This is because you can nest lists inside of each other. A good analogy is to think of a list as a vector. If each entry in the vector is a vector itself, then we really have a matrix. Specifically, you could build a 2x3 matrix in the following way:

```
matrix = [[1,2,3],[4,5,6]]
print("The first row ",matrix[0])
print("Element 0,2: ",matrix[0][2])
```

The `matrix` list in this case has 2 elements, and each of those elements is a list of 3 values. `matrix[0]` thus refers to the first set of 3 elements, and `matrix[0][2]` to the third element of that set.

2.1.5 Type of variables

We have already seen a few of the different types of variables that python offers (and it typically can infer from the context what type of variable you intend to use). A few common types of variables are:

- **integers:** These are signed integer numbers.
- **floats:** These are signed decimal numbers; they usually cannot be represented exactly and are subject to round off errors
- **strings:** These are "strings" of characters and the value of the variable is always surrounded by quotes (single or double). For example `"someNumber"` is interpreted as a string, whereas `someNumber` is interpreted as a variable (that could be of any type)
- **lists:** These are arrays of variables; each element in the array does not have to be of the same type, and lists can be nested. Elements in a list have a well defined position in the list and can be reference by using their index (e.g. `array[2]` is the third element of a list since the first element has index 0)

- **dictionaries:** These are similar to lists, but instead of referencing elements by a numbered index, they are referenced by a string (which we call the "key"). Technically, the key does not need to be a string, but it has to be "non-mutable"; a string is just the most common usage case. Each key must be unique. Dictionaries use curly braces when they are defined. For example, `students = {'bob':123,'jane':456}`, and are referenced using square braces: `sn=students['bob']`.

- **tuples:** These are values that are grouped together with a parenthesis and can be indexed in a way similar to a list

- **objects:** These are more complex data types that can be user defined, for example, one could define a "student" object with a variety of properties (name, student number, grades) and write code that directly manipulates the objects. Strings and lists are actually objects in python, and that is why they have many available functions to manipulate them.

2.2 Controlling the flow

At this point, we can do some basic calculations and pretty much anything that a calculator can do. However, we cannot do anything very complex without controlling the flow of our program. This section introduces a few key ways to control the flow of the program.

2.2.1 Scope

In python, we can define small little units of code that are only executed in certain conditions. Those units of code are **preceded by a line that ends with a colon (':') and must be indented to the right** compared to the rest of the code. We say that the part of the code that is indented forms a 'scope'; whatever happens in the scope, stays in the scope, just like Vegas. Scopes can be nested within each other. Variable that are defined inside of a scope cease to exist outside the scope. Variables that are defined earlier and outside of the scope, are accessible in the scope.

2.2.2 If-statements

One of the most useful types of scope, is the scope defined by an "if-statement". Code inside the scope of an if-statement is executed only if the if statement is `True`:

```
#define 3 variables:
a = 2
b = 3
c = 4
if b > a: #True
  print("b is bigger than a")
if c < a: #False
  print("c is smaller than a")
if 2*a == c: #True
  print("2a is equal to c")
```

In this case, the code inside the scope of the second if-statement is never executed, because the condition before the colon is False. The greater and lesser than symbols are pretty obvious, but note the use of the double equal sign, `==`, to test if two values are equal. This is because the single

equal sign is already taken (it's the assignment operator), so if we want to test if two things are equal, we must use the "equality operator", `==`.

2.2.3 Loops

The other very useful type of scope is the "loop". This is where computers really excel at doing repetitive tasks. The most common type of loop is the "for" loop, which defines a scope that is executed a specified number of times. A common application is to "loop" through all the elements inside of a list. For example:

```
radii=[0.9,1.0,0.95,1.1,1.2,1.0,0.8,0.85,1.05,1.0]
for r in radii:
  print("r = ",r)
```

Again, the syntax is similar as that for the if-statment. We first have a statement that defines the loop and ends with a colon, followed by a section that is indented to the right (the scope of the for loop). In this case, the `for r in radii` statement means the following: "Create a new variable called `r`, and then execute the code in the scope one time for each value inside of `radii`, assigning that value to `r` each time". This particular example will print all of the values inside of the list.

A useful function for loops is the `range()` function. In its most basic implementation (when passed only 1 argument, `n`) it returns a list of numbers from 0 to `n-1`. This makes is particularly easy to loop through a list using the index of the elements:

```
radii=[0.9,1.0,0.95,1.1,1.2,1.0,0.8,0.85,1.05,1.0]
n = len(radii)
for index in range(n):
  #the variable index will have values 0..9
  print("Element ",index," has value ", radii[index])
```

EXAMPLE 2-1: Use a for loop to create 2 lists that hold the numbers from 1 to 100 and their squares, respectively

```
numbers=[] # an empty list that we will fill
squares=[]
for index in range(100):
  x=index+1 # since the first value is 0
  numbers.append(x)
  squares.append(x*x)
print("the numbers are ",numbers)
print("the squares are ",squares)
```

2.3 Defining your own functions

We have already encountered a few functions; `print()` and `range()`. Functions may take "arguments" (they don't have to), typically would do something (using the arguments if applicable), and may return a value (but they don't have to). Functions are extremely useful, especially as a way to make your code re-usable and to avoid re-writing the same code over and over.

You must first define a function before you can use it. Defining a function is a way for you to

personalize python and give it additional functionality. Let's write a simple function to return the factorial of a number

```
#define the factorial function
def factorial(n):
    fact=1 # keep track of the factorial
    for i in range(n):
        fact = fact*(i+1) #+1 because the first i is zero
    return fact
```

Functions are defined using the `def` statement, followed by the name of the function (`factorial` in this case) and then, if applicable, the arguments that the function needs in parenthesis. The statement ends with a colon and the next line is indented. This defines the scope of the function. To calculate the factorial, we declared a variable that will hold our result (`fact`). We then needed to do a for loop to calculate the factorial. Finally, outside of the scope of the for loop but still inside the scope of the function, we used the `return` statement to pass the calculated number (`fact`) out of the function. The function can then be used as follows:

```
print("3 factorial is ",factorial(3))
print("10 factorial is ",factorial(10))
```

You have now added the functionality of calculating factorials into python!

Functions need not take an argument or return a value, for example:

```
#define a function to print a warning
def printWarning():
  print("Hey, you should be warned here")

#use the function:
printWarning()
```

functions can also return multiple values (as a tuple):

```
#define a function of 1 argument
def squareCube(x):
  #return a tuple
  return x*x, x*x*x

#use the function to assign values to 2 variables
x2 ,x3 =squareCube(5)

print("5 squared is ",x2," 5 cubed is ",x3)
```

In reality, python is not actually returning multiple values, but rather it is bundling the two values into a tuple. In practice, you can however pretend that it is in fact returning multiple comma-separated values.

2.4 Importing modules

Fortunately, many have come before us to write a lot of useful functions, and made them available for us to use. Python comes already bundled with a large amount of functions already written for us, and these are available through "modules". In order to be able to use the functions inside of a module, you "import" the module. This is typically done at the top of your program. For example, the `math` module has a large number of functions for doing math. To use the square root function for example, you would import the math module as follows:

```
import math
a=2
sqrta = math.sqrt(a)
print("the square root of a is ",sqrta)
```

We may also find it useful to change the name of a package when we import it (e.g. to make it shorter if we have to type it a lot). In this case, we can use the `import ... as` format:

```
import math as m
a=2
sina = m.sin(a)
print("the sin of a is ",sina)
```

There are other ways to "import" functions from modules, but the above ways are the most explicit (that is, it is clear when you call the function `math.sqrt()` that the function belongs to the math module).

For completeness, other equivalent ways to achieve the same functionality are:

```
from math import *
a=2
sqrta =sqrt(a) # don't need the math. in this case
print("the square root of a is ",sqrta)
```

which imports all functions from the math module and makes them available directly (without needing the `math.`), or:

```
from math import sqrt
a=2
sqrta =sqrt(a) # don't need the math. in this case
print("the square root of a is ",sqrta)
```

which imports only the `sqrt` function (so for example, `sin(a)`) would not work.

It is very easy to create your own modules. Simply place the definitions of your functions into a file called 'mymodule.py' (where 'mymodule' can be whatever name you choose), and you can then use them by simply typing `import mymodule` (without the '.py'). For example, you may have a file called 'errorAnalysis.py' with the following content:

```
def factorial(n):
    fact=1 # keep track of the factorial
    for i in range(n):
        fact = fact*(i+1) #+1 because the first i is zero
    return fact

def squareCube(x):
  #return a tuple
   return x*x, x*x*x
```

and you can then use these functions in a program:

```
import errorAnalysis

print("8 factorial is ",errorAnalysis.factorial(8))
print("square and cube of 5 are ",errorAnalysis.squareCube(5))
```

There a lot of modules available in python, some are included directly and many can be downloaded. The internet is your friend. If you want to do something, chances are someone has already done it. Don't be afraid to ask the internet!

2.5 Reading and writing data files

Reading and writing data from and to files is very useful, as opposed to directly "hardcoding" your data into a program (as we did above for the radii). For example, you and your lab partner may each have set of measurement that you want to analyse in the same way. In that case, it makes sense to write a common program that can analyse the data, and provide those data to the program from a file. Another obvious scenario is when your experimental apparatus already writes the data to a file; it would make little sense to open the file and type the contents into your program.

Let's say that we have a text file with the following data in it (representing measurements of radius and circumference from an experiment like that in example 1-1), and let's assume the file is named 'radcirc.dat':

```
0.9   5.4
1.0   6.2
0.95   6.2
1.1   5.9
1.2   7.9
1.0   6.0
0.8   5.1
0.85   4.8
1.05   6.7
1.0   6.5
```

The data are in 2 columns, one for radius and one for circumference. We now wish to read these values into our program, compute $2\pi r$ from the radius, compare that with the measured circumference, and then output the difference to a new file, called 'results.dat'. This is done with the following:

```
import math #because we need pi, and it's defined in math

#open the file with the data, need quotation marks on the file name!
dataFile = open('radcirc.dat', 'r') #open in read mode

#open the file to write the output:
resultFile = open('results.dat','w') #open in write mode

#loop over each line in the file:
for line in dataFile:
  #line contains both numbers, so we split them up into list
  lineData = line.split()
  #take each value from the list and put them into new variables
  #we need to convert them to numbers (float) because python assumes
  #that they are strings of characters:
  r=float(lineData[0])
  circ=float(lineData[1])
```

```
  #the predicted circumference is:
  predicted=2*math.pi*r

  #the difference with measured one is:
  difference=predicted - circ
  print(difference, file=resultFile)
```

The file 'results.dat' now contains a single column with the differences between the predicted and measured circumferences. You can see that we used the `print()` function, with an additional "named argument" to specify that we would like to print to the file, rather than to the screen; in this case, we passed our file variable (`resultFile`) to the argument that is named `file`.

If you look at the file 'results.dat', you will find that the numbers have more decimal places that would really be considered significant. We can thus use the `format()` method to clean it up.

```
import math #because we need pi, and it's defined in math

#open the file with the data, need quotation marks on the file name!
dataFile = open('radcirc.dat', 'r') #open in read mode

#open the file to write the output:
resultFile = open('results.dat','w') #open in write mode

#loop over each line in the file:
for line in dataFile:
  #line contains both numbers, so we split them up into list
  lineData = line.split()
  #take each value from the list and put them into new variables
  #we need to convert them to numbers (float) because python assumes
  #that they are strings of characters:
  r=float(lineData[0])
  circ=float(lineData[1])

  #the predicted circumference is:
  predicted=2*math.pi*r

  #the difference with measured one is:
  difference=predicted - circ
  print("{:.2f}".format(difference), file=resultFile)
```

2.6 Module of use in scientific computing

Data science needs in industry have led to the rapid development of python modules for data analysis. Traditionally, programming languages like python, which are "interpreted" on the fly as the computer tries to run the code, are slower at processing numbers than compilable languages like C++. In order to alleviate this issue somewhat, people have programmed modules that provide "pythonic" (easy to program) interfaces for dealing with numbers that, behind the scenes, are much more efficient than regular python code. These interfaces basically figure out what the user is trying to accomplish and then run the calculations using C++ code before passing the result back in python. Some of these interfaces are quite complex and actually allow the code to be run on highly efficient graphical processing units (graphics card); these are highly optimized for matrix algebra and anything where parallel computations are efficient.

The other aspect of data analysis that is often needed is a way to visualize data and make graphs,

and there are a variety of modules for that as well.

A common module (actually, a set of modules) for scientific work in python is "scipy". Scipy (a "stack" of modules), comes with most scientific python packages (such as Anaconda, Python(x,y), etc.) and includes modules for plotting (matplotlib), scientific calculations (scipy), symbolic math (sympy), efficiently using arrays of number (numpy), and others. We will give a brief overview of numpy and matplotlib since they are the most useful in our context.

2.6.1 Working with numbers, numpy

In scientific computing, we work predominantly with numbers, and usually a lot of them (think of a list of measurements). We already saw that the python list is a handy way to hold an array of numbers. The list has a lot of useful functions (e.g. dynamically changing the size of the list, searching) that are not very useful in numerical computation, and this added functionality makes lists inefficient for scientific computing. At the core of the numpy module is a type of array that is much more efficient than lists for dealing with numbers. All of the elements in a numpy array must be of the same type. Numpy arrays can also be mult-dimensional (e.g. a matrix or tensor). There are also a variety of functions that can act on numpy arrays, to fill them, evaluate properties about them, and change their shape. Here are a few examples:

```
# We import numpy (and by convention shorten it to np, since
# we use it a lot)
import numpy as np

# a simple one-dimensional array
a = np.array([1,2,3,4,5,6])

# we can print the array and see it's "shape":
print("The array is ",a," and has shape: ",a.shape, "and the second element is ", a
    [1])
print("The size of the array is ",a.size)

#we can change the shape to be 2D, e.g. 2x3:
a=a.reshape(2,3)
print("The array is :\n",a,"\n and has shape: ",a.shape)

#we access an element of a two d array as expected:
print("The 1,2 element is ",a[1][2])
print("The size of the array is ",a.size)

#we can create an array full of zeroes:
#we specify the shape of the array with a tuple (hence the extra parenthesis)
z=np.zeros( (4,5) ) # 4 by 5 array of zeros
print("Zeroes: \n", z)

#we can create an array full of ones:
#for a 1D array, no need to use a tuple for the shape
o=np.ones(10) # array of 10 times the number 1
print("Ones: ",o)

#an array with numbers in a certain range:
seq1 = np.arange(10,30,5) #numbers from 10 to 30 spaced by 5
print("Sequence 1: ", seq1)

#if using floats, better to use linspace
```

```
seq2 = np.linspace(0,10,11) #11 numbers from 0 to 10
print("Sequence 2: ", seq2)
```

Numpy has a lot of more advanced useful features, such as the ability to directly load a file into an array, to write an array into a file, to calculate properties of the array, and to manipulate arrays. Following, are a few advanced examples for reference.

2.6.1.1 Reading and writing to file

Here is a simple way to read and write a numpy array directly to a file. In this example, it uses the default format, which is to separate entries in the file with a space. One can specify different delimiters, such as commas, if preferred.

```
import numpy as np

#make a 4x5 array from random numbers that are normally distributed
a=np.random.normal(7,1,(4,5)) #mean = 7, stdev = 1, shape=(4,5)
print("a:\n",a)

#now save to file (with 2 decimal places)
#for Windows, need the '\r\n', otherwise, '\n' works on Mac and Linux:
np.savetxt('normaldist.dat',a, fmt='%.2f',newline='\r\n')

#read from file:
b=np.loadtxt('normaldist.dat')
print("b:\n",b)
```

2.6.1.2 Calculating properties of an array

Numpy includes some basic functions to calculate simple properties of arrays (such as the mean of the elements or the sum of the elements):

```
import numpy as np

#make an array of 15 random numbers that are normally distributed
a=np.random.normal(7,1,15) #mean = 7, stdev = 1, shape = 15
print("a:\n",a)

#the average of those numbers and the standard deviation:
print("The mean is ",np.mean(a)," the standard deviation is: ",np.std(a))

#the sum of those numbers:
print("The sum is ",np.sum(a)," and the average is thus: ",np.sum(a)/a.size)
```

2.6.1.3 Operations on arrays

Numpy arrays can be used in computations (e.g. multiplying or summing arrays together) and operated on with numpy functions (such as taking the square root of all elements).

```
import numpy as np

#make two arrays of numbers
a=np.linspace(1,5,5)
b=np.linspace(6,10,5)
print("a: ",a)
print("b: ",b)
```

```
#add arrays together, element by element:
sumAB=a+b
print("sum: ",sumAB)

#multiply, element by element:
productAB=a*b
print("product: ",productAB)

#multiply all elements by a scalar:
twoA=2*a
print("2 times a: ",twoA)

#apply a function to elements:
squareA=np.square(a)
print("a squared: ",squareA)
```

You can even write a function that is able to act on arrays or on regular numbers. For example, the following will add two numbers "in quadrature", or if given two arrays as inputs, it will return an array of the values added in quadrature:

```
import numpy as np

#define a simple function:
def addInQuad (e1,e2):
  return np.sqrt(e1*e1+e2*e2)

print("2 and 3 added in quadrature is ",addInQuad(2,3))

#make two arrays of numbers
a=np.linspace(1,5,5)
b=np.linspace(6,10,5)
print("a: ",a)
print("b: ",b)

#c is an array where each element is the elements of a and b
#added in quadrature
c=addInQuad(a,b)
print("c: ",c)
```

2.6.2 Making plots with matplotlib and pylab

Matplotlib is the de-facto standard modul for plotting in python when doing scientific computing. The pylab module uses the matplotlib module and adds some functionality to give it a Matlab feel. You should note that if you use 'pylab', you are really using matplotlib, and therefore, searching the internet for help with matplotlib might yield the answers that you seek.

2.6.2.1 Scatter plot and plotting a function

Matplotlib can understand numpy arrays. A simple scatter plot can be done with the following code (Fig. 2.1):

```
import numpy as np
import pylab as pl
import math

#create arrays of angles of values from 0 to 2pi
x=np.linspace(0,2*math.pi,50) #values of x
```

```
#use the fact that we can operate element-wise on a whole array:
y=np.sin(x)
z=np.cos(x)

#plot (x,y) and (x,z)
pl.plot(x,y,label="sin(x)") #default is a blue line
pl.plot(x,z,'ro',label="sin(x)") #plot red circles ('ro') instead

#make it pretty:
pl.xlabel("angle in radians")
pl.ylabel("sin or cos")
pl.title("Trig functions")
pl.legend(loc='upper center')
pl.grid(True)

#show the plot:
pl.show()
```

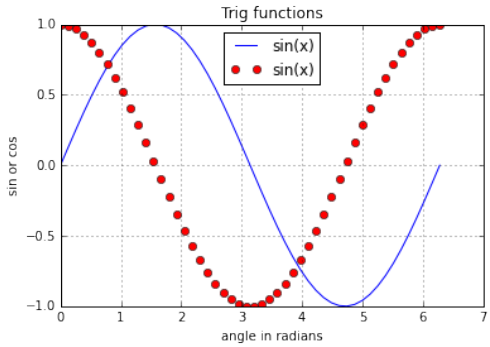

Figure 2.1: Two data sets plotted with legend

2.6.2.2 Plot with error bars

We can also plot with error bars, and include a legend (Fig. 2.2).

```
import numpy as np
import pylab as pl
import math

#plot (x,y)
x=np.linspace(0,2*math.pi,50) # 50 values between 0 and 2pi
y=np.sin(x)

#let's add variable errors in y and fixed errors in x
dy=0.2*np.sqrt(np.abs(y))
dx=0.1*np.ones(x.size)

pl.errorbar(x,y,yerr=dy,xerr=dx, fmt='o')
pl.title("Plot with error bars")
pl.show()
```

2.6.2.3 Plotting histograms

Finally, a particularly useful way to plot data in statistics is the "histogram". This is done very easily with matplotlib (see Fig. 2.3):

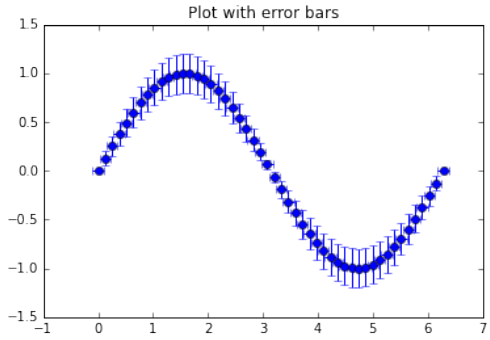

Figure 2.2: Data set with x and y error bars.

```
import numpy as np
import pylab as pl
import math

#create a set of normally distributed random numbers
x=np.random.normal(7,1,1000)

#histogram the values
n,bins,patches=pl.hist(x,bins=40)

#Make it pretty:
pl.xlabel("value")
pl.ylabel("frequency of value")
pl.title("Normally distributed variable")

#Set the axis range
pl.axis([0,12,0,80])

#show the plot:
pl.show()
```

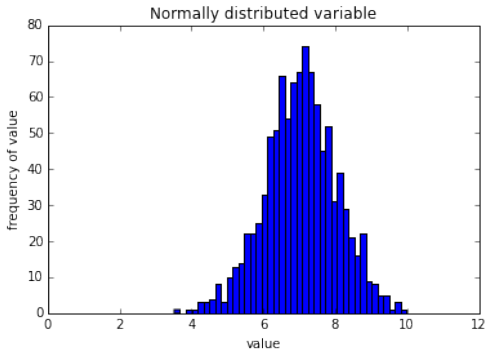

Figure 2.3: Histogram of random values normally distributed with a mean of 7 and standard deviation of 1

The `hist()` function plots the histogram and return a tuple `(n,bins,patches)`. The first element in the tuple (`n`) is an array containing the number of counts in each bin, the second element (`bins`) is an array with the left edge of each bin, and the third element (`patches`) contains information on drawing the actual histogram (e.g. where to place rectangles) and has no analysis use for us.

The `hist()` function takes as first argument the array of values that it should histogram, and it can take a variety of other arguments to control the display of the histogram. A particularly important argument is the `bins` argument, which controls the binning of the histogram. If this is a single number, then python will automatically figure out the range of values in the array and create that many equal bins (40 bins in the example above). One can also specify `bins` by setting it equal to an array of values which correspond to the left side of the bins that we specify (and the last value in the array in the right edge of the highest bin). For example:

```
#histogram the values
n, bins, patches=pl.hist(x, bins=[6.5,7,7.1,7.2,7.3,7.4,7.5,8])
```

would create a histogram with only a subset of the data (from 6.5 to 8), with two wider bins on either side (from 6.5 to 7 and from 7.5 to 8), and a set of smaller bins between 7 and 7.5.

There are also a variety of options for controlling the look of the histogram, and these can be combined onto a single plot, as below, which produces Figure 2.4.

```
import numpy as np
import pylab as pl
import math

#create a set of normally distributed random numbers
x=np.random.normal(7,1,1000)

#histogram the values with different binning and different styles
pl.hist(x,bins=40,alpha=0.5,color='red',histtype='stepfilled',label='40 regular
    bins') #alpha makes it translucent...
pl.hist(x,bins=[6.5,7,7.1,7.2,7.3,7.4,7.5,8],alpha=0.5,color='blue',label='7 uneven
     bins')
#Make it pretty:
pl.xlabel("value")
pl.ylabel("frequency of value")
pl.title("Normally distributed variable - different binning")
pl.legend()
pl.grid()

#Set the axis range
pl.axis([3.5,10.5,0,250])
#show the plot:
pl.show()
```

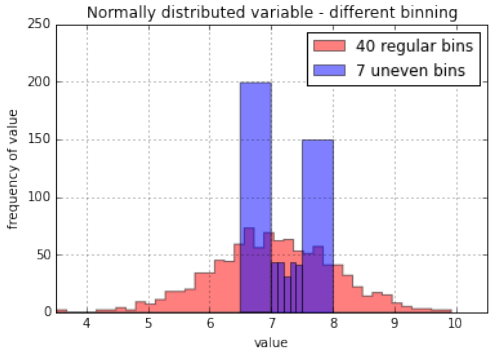

Figure 2.4: Two different histograms of the same data.

We can also normalize the histograms (so that the sum of the bin contents is 1), and we can also plot the "cumulative" histogram (where each bin is the sum of the bin contents that are below it). The cumulative histogram is effectively the integral of a normal histogram. The code below illustrates some of the various possibilities for plotting histogram, and also shows how to make "sub-plots" in python (which does not only apply for histograms).

```
import numpy as np
import pylab as pl
import math
%matplotlib inline

#create a set of normally distributed random numbers
x=np.random.normal(7,1,1000)

#Create different sub plots:
#Available colours by name are the standard "html colours"
pl.subplot(221)
pl.hist(x,bins=40,color='gray',histtype='step')
pl.title("basic histogram")
pl.xlabel("value")
pl.ylabel("frequency of value")

pl.subplot(222)
pl.hist(x,bins=40,color='burlywood',histtype='stepfilled',normed=True,label='40
    bins')
pl.hist(x,bins=10,color='darkslateblue',alpha=0.5,histtype='bar',normed=True,label=
    '10 bins')
pl.title("normalized")
pl.xlabel("value")
pl.ylabel("frequency of value")
pl.legend(loc='upper left',prop={'size':8})

pl.subplot(223)
pl.hist(x,bins=40,color='darkgoldenrod',alpha=0.5,histtype='stepfilled',normed=True
    ,cumulative=True)
pl.title("normalized and cumulative")
pl.xlabel("value")
pl.ylabel("integral of value")

pl.subplot(224)
pl.hist(x,bins=40,color='chocolate',histtype='stepfilled', log=True)
pl.grid()
pl.title("basic - log scale")
pl.xlabel("value")
pl.ylabel("frequency of value")

#show the figure:
pl.tight_layout() #space out the sub plots
pl.show()
```

2.6.3 Sympy: Symbolic computations in python

In later chapters, we will find that we often need to take derivatives of quantities. As in other software packages such as Matlab, Maple, and Mathematica, it is also possible to evaluate derivatives symbolically in python. This is done using the `sympy` ("symbolic python") package.

In order to use symbolic computation, one must first import the sympy package, and then define

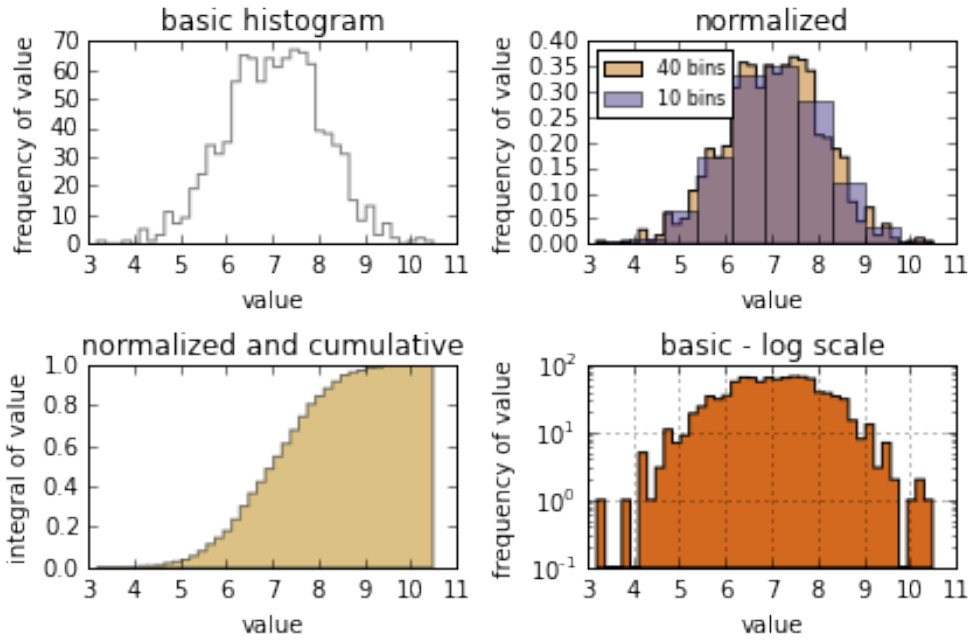

Figure 2.5: Various ways to display a histogram, also showing how to use subplots.

variable that python will represent as symbols. In this example, we define x and y as symbols, as well as a symbolic function $f(x, y) = \sqrt{x^2 + y^2}$, that depends on x and y:

```
import sympy as sym
x,y = sym.symbols('x y')  #declare variables x and y to be used as symbols
f = sym.sqrt(x**2+y**2) #a function of x and y
```

The derivatives of f with respect to x and y can be found easily, and manipulated as symbolic expressions (which we sum below):

```
import sympy as sym
x,y = sym.symbols('x y')  #declare variables x and y to be used as symbols
f = sym.sqrt(x**2+y**2) #a function of x and y
#take the partial derivatives of f
dfdx = sym.diff(f,x)
dfdy = sym.diff(f,y)
print("df/dx=",dfdx,"df/dy=",dfdy)
#add them together
dSum = dfdx+dfdy
print("df/dx+df/dy = ",dSum)
```

Finally, we can also substitute in numerical values using the `subs()` command on an expression and passing it an array of tuples. We can numerically evaluate an expression using `N()`:

```
#substitute in some numbers for x and y
nValue1 = dSum.subs([(x,2),(y,3)])
nValue2 = dSum.subs([(x,4),(y,5)])
print("value 1: {:.2f}, value 2: {:.2f}".format(sym.N(nValue1),sym.N(nValue2)))
```

2.6.4 Curve fitting with scipy optimize

It is often the case that one has a set of data points, (x_i, y_i), and one wants to "fit" the data to a model function, $y = f(x)$, and use the data to determine parameters of the model. For example,

a set of data may be described by a second order polynomial, $f(x) = a + bx + cx^2$, and you may need to use the data to determine the parameters, a, b, and c that best fit the data. This can be done using the scipy.optimize module. The first thing to do is to define a python function to represent our model function:

```
import numpy as np
def model(x,a,b,c):
    return a+b*x+c*x*x
```

where you should note that the dependent variable, x, is the first argument of the model function, whereas the **parameters**, are any number of arguments after the first. This is just seen as a function of x with parameters a, b, and c that we want to determine from the data.

Let us assume that our data is stored into three one-dimensional numpy arrays, `xdata`, `ydata`, and `ydataErr`, holding the x value of the points, the y values of the points, and the uncertainties on the y values, respectively. The algorithm that varies the model parameters can have some difficulties in finding good fit values for the fit parameters, especially if the fit function is complicated. It is thus often necessary to given the fitter an initial guess for the parameters. The fitter is run with the following code:

```
from scipy.optimize import curve_fit
#Guesses for the parameters
A,B,C = 5,50,-3
parGuess = [A,B,C]
pars, pcov = curve_fit(model, xdata, ydata,sigma=ydataErr,p0=parGuess)
```

where we imported only the `curve_fit()` function from the `scipy.optimize` module. The `curve_fit()` function returns a tuple, `pars,pcov`, where `pars` are the best fit values for the parameters that we want and `pcov` is the covariance matrix of the parameters. The square root of the diagonal elements of the covariance matrix correspond to the errors on the parameters from the fit, which we can easily get:

```
#Get the errors out of the covariance matrix
perr = np.sqrt(np.diag(pcov))
#Print out the fitted parameters and errors
print("Fitted paramaters:",
      "\na:","{0:.2f} +/- {1:.2f}".format(pars[0],perr[0]),
      "\nb:","{0:.2f} +/- {1:.2f}".format(pars[1],perr[1]),
      "\nc:","{0:.2f} +/- {1:.2f}".format(pars[2],perr[2]))
```

Below is a full example that creates "fake data", fits those data to the model, and then makes a plot showing the fitted function and the data on the same plot.

```
import numpy as np
import pylab as pl
from math import *
from scipy.optimize import curve_fit #for fitting

#our model (we will also use it to generate data)
def model(x,a,b,c):
    return a+b*x+c*x*x
    #return a+b*np.exp(x/c)

#Let's make some fake data:
#Set the "true" parameters for the model:
A,B,C = 5,50,-3
```

```
#generate x-values of the data:
ndata = 50 #number of data points
xdata = np.linspace(0,10,ndata)

#generate y points that are normally distributed about the model evaluated at xdata
#with a sigma of yerr
yerr=5 #the uncertainty on y values
ydata = np.random.normal(model(xdata,A,B,C),yerr)

#Now fit this to the model:
#Need an array to hold the uncertainties on the y points (all are equal to yerr)
ydataErr=yerr*np.ones(ydata.size)
#Provide parameter guesses:
parGuess=[A,B,C]
pars, pcov = curve_fit(model, xdata, ydata,sigma=ydataErr,p0=parGuess)
#Get the errors out of the covariance matrix
perr = np.sqrt(np.diag(pcov))

#Create some text with the fit results to put into our plot
resultTxt = "Fitted parameters:\n"
parNames = ["a","b","c"]
for i in range(pars.size):
    resultTxt = resultTxt+"{:s}: {:.2f} +/- {:.2f}\n".format(parNames[i],pars[i],
    perr[i])

#Plot the data with error bars and the result of the fit
#Generage a curve from the model and the fitted parameters
yfit = model(xdata,pars[0],pars[1],pars[2])
pl.errorbar(xdata,ydata,yerr,fmt='o',label='data')
pl.plot(xdata,yfit,'r',label='fit')
pl.axis([xdata.min()-1,xdata.max()+1,ydata.min()-yerr-1,1.1*ydata.max()])
pl.legend(loc='upper left')
pl.text(5,50,resultTxt,fontsize=14)
pl.show()
```

The result is shown in Figure 2.6.

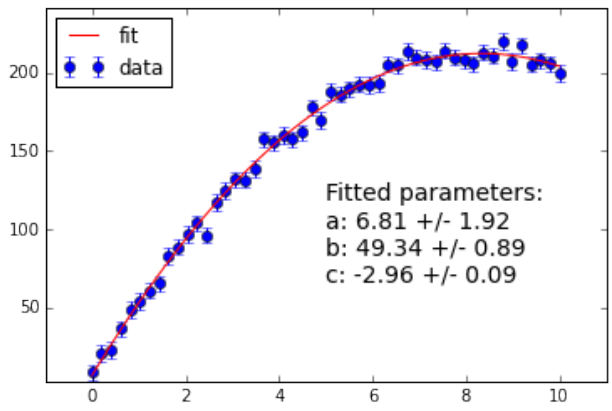

Figure 2.6: Example of fit to a second order polynomial.

3 Determining uncertainties and presenting data

In this chapter we will look at how to determine uncertainties on measurements, how to present those uncertainties, what they mean, and how to determine if two values agree. We will give some basic guidelines that are of use for undergraduate laboratories, and justify most of those guidelines in later chapters.

3.1 The importance of uncertainties

The importance of uncertainties is fundamental in applying the Scientific Method, where one tries to invalidate a hypothesis using data. Generally, we can think of the need to compare two numbers that cannot be determined to infinite precision (e.g. a measurement and a theoretical prediction). We thus need a way to scientifically determine whether two quantities are "consistent" with each other. We will spend quite some time defining what we mean by "consistent" in later chapters. We also need to understand how to present a number; if our "value determined from experiment" involves an operation that leads to many digits (e.g. multiplying by an irrational number), we will have to decide how many digits are really "significant" when performing the comparison.

You may find that you can measure a number to several decimals with an accurate ruler, but then are unable to reproduce that measurement to the same accuracy. Even if the ruler can measure something to a certain number of decimals, the uncertainty in the measurement may not be representative of that precision. For example, consider using a micrometer (which can measure distances of order 10 cm to sub-millimetre accuracy) to measure your height; you will inevitably introduce errors when you try to stack the micrometers atop each other to reach your height. Your final uncertainty should thus be larger than that of the micrometer, even though that is the only instrument that you used.

The need to assign uncertainties to numbers also arises when we try to build things. If you are cutting pieces of wood to build furniture, you will need them to fit together. You will find that you cannot cut them to be exactly the length that you determined in your drawings (even when you take into account the width of the saw blade, use a good square, work very carefully, etc.). You will find that you need to determine the "tolerance" in all of your required dimensions for the parts to fit. If you go to a professional machine shop to have parts manufactured, the machinists will inevitably ask you what tolerance you have on the pieces; it is usually not their job to assemble

your parts for you. This issue becomes one that is quite important in designing parts; it may be relatively easy to design the function of a part compared to determining how much tolerance is required for the parts to fit. Although the issue of tolerance is related to uncertainties, in this book, we will focus more on the aspect of comparing two numbers, but do keep in mind that the concepts usually apply to tolerances in construction.

3.2 Determining uncertainties

3.2.1 What do we mean by "uncertainty"?

You have likely already encountered uncertainties in your labs, but you may not really have thought them through. A typical experiment would be to build a simple balance, where two unequal weights are placed on either side of a lever arm. You may be asked to determine where to place the weights to equilibrate the balance. Inevitably, you will measure the mass of the weights, and if the digital scale displays numbers down to a gram, you would likely quote the weight of one of the masses as, say, m_1=(595.0 $\pm$ 0.5) g. That may seem pretty straightforward, but what do you really mean by (595.0 $\pm$ 0.5) g? Do you mean that the mass of the weight is *guaranteed* to be between 594.5 g and 595.5 g? Does it mean you would bet your life on it? What if you later discovered that the scale reads all weights to be 5 g too light, can you guard yourself against that (since you are apparently willing to bet your life on your measurements)?

The basic answer to these questions is that you cannot just quote the measured value to be m_1=(595.0 $\pm$ 0,5) g! It is *on you*, the scientist, to explain what you mean by m_1=(595.0 $\pm$ 0.5) g. If you mean that you just assigned an uncertainty, without much thought, of 0.5 g because that is half of the smallest digit on the scale, then you have to say that. If you mean that you would bet your life that the true value is between 594.5 g and 595.5 g, then you should say that. If you mean that you are 90% confident that it lies between 594.5 g and 595.5 g, then you should say that (but then, what do you mean by 90% confident?). If you are not confident that the scale reads "true" (e.g. you didn't repeat your measurement with a different scale, or don't know if the scale was calibrated), then you should mention that as well. Again, we see that doing good science is about being precise! Furthermore, doing good experimental physics, is all about *understanding* your uncertainties.

You may have also seen results from opinion polls that try to survey the general population to predict election results. These polls can be quite important in how politicians plan their campaigns (read: where to spend money), and the polls may be quite expensive to conduct, so they typically are motivated to get scientific data. You may, for example, see the following reported in a news article: "Candidate A is leading, favoured by 52% of voters polled, whereas Candidate B has fallen behind and is favoured by only 48% of voters". At the end of the article, there will typically be a statement of the form: "The poll was conducted by BMIP by calling 1000 registered voters and conducting a 10 question survey by phone. The margin of error on the polling numbers are 4%".

We can ask ourselves what they mean by the "margin of error". It certainly makes sense that they cannot get an exact number for the whole electorate by surveying only 1000 people, as there will necessarily be statistical fluctuations in the sample that they surveyed[1]. We can ask ourselves if

[1]We will visit the issue of trying to determine a property of the population by measuring that property on a sample of the population in chapter REF.

the difference between the two candidates really is "significant". If the results really are $(52 \pm 4)\%$ and $(48 \pm 4)\%$, there is substantial overlap between the possible results and a statistical fluctuation cannot be excluded. Do they mean that for Candidate A, the result is definitely between 48% and 56% with an equal probability over that whole range? How did they determine the 4% uncertainty? Why did they ask 10 questions when they only needed one?

Although we will not spend any time on designing opinion polls, we will spend a substantial amount of time trying to understand the uncertainties in statistical quantities. As you can see, a strong understanding of statistical uncertainties is not only useful in physics, but carries out to a wide range of applications in real life, where we are often confronted with dubious conclusions from badly analysed data.

All this being said, when we quote a number as $A \pm \delta A$, we imply that A is quite likely to be in the range $A - \delta A$ to $A + \delta A$ (and we typically specify what we mean by "quite likely"). Generally, we also imply that it is **not** more likely for A to be closer to one of the end points of the range than the other. We always quote the uncertainty, δA, as a positive number.

3.2.2 Systematic uncertainties

Systematic uncertainties are different from what we would call "statistical" or "random" uncertainties. By **random uncertainty**, we mean that a repeated measurement will give variable results that are **equally likely to be higher or lower than the true value**. With a **systematic error, we mean that reapeated measurements will be systematically higher or lower than the true value** (hence the name). The advantage of a random error is that if we make many measurements, we can expect that the average of those measurements will tend to the true value. There are no advantages to systematic errors! They are often difficult to detect and the design of most experiments is often driven by the need to reduce systematic uncertainties, ideally to the level that they are negligible.

For example, imagine that several laboratory groups are measuring weights for an experiment. They all have their own scales, and as good experimentalists, they borrow each other's scales to check their measurements and make sure that their own scale is calibrated. However, all of the scales were calibrated by the TA before the lab, and the TA used the wrong "standard weight" when calibrating all of the scales. This is an example of a systematic uncertainty in all of the mass measurements in the lab that is very difficult to catch. Or maybe the scales are more accurate for weights that are smaller than 100 g, and the various lab groups have used weight well below and well above 100 g; some groups may be affected by the systematic uncertainty, while others not.

Generally, it is not possible to arrive at a prescription of how to determine the systematic uncertainty in a measurement. It is however very important to think very carefully about all possible sources of systematic uncertainty in a measurement. Ideally, this is done before designing the experiment so that the design of the experiment can be improved to minimize the associated uncertainties. One then estimates the magnitude of these uncertainties and conducts the experiment with the knowledge that the most precise result that they can obtain will have an uncertainty at least as big as the systematic uncertainty.

Experiments that are currently searching for dark matter are building large sensitive detectors that

are looking for small flashes of light from dark matter interacting in their detector. The problem is that naturally occurring radioactive decays in the materials used to construct the experiments will also produce flashes of light. Therefore, those experiments are only sensitive to dark matter if the rate of the flashes of light from dark matter is substantially higher than the rate from "backgrounds". Much of the effort in designing these experiments is thus spent at minimizing the backgrounds (e.g. by material selection) so that the sensitivity to dark matter is as high as possible, and not "systematically" limited by a high background rate. The experiments are then conducted until the sensitivity to dark matter becomes limited by the background rate.

3.2.3 Uncertainties from scales - "Half of the smallest division"

In this section, we consider the uncertainty in measurements made with a device that has some sort of scale on it (a ruler, a digital scale, a needle on a dial, etc.). We assume that the measuring device is well calibrated, but in a real application, you should estimate whether you also need to include a systematic uncertainty.

If you have a ruler that has mm graduations on it, you should expect that you can measure a distance to within 1 mm. In typical scenario, you will use the ruler and find that the true value is between two lines on the ruler. In this case, you would typically quote the best estimate of the measurement to correspond to the line on the ruler **that is closest to the distance being measured and an uncertainty that is half of the smallest division on the ruler**. Figure 3.1 shows an example of measuring something with a ruler. Using our prescription, you would quote the grey object to have length of (2.80 ± 0.05) cm. That is, you are reasonably confident that the true length is between 2.75 cm and 2.85 cm, which is a range of 1 mm.

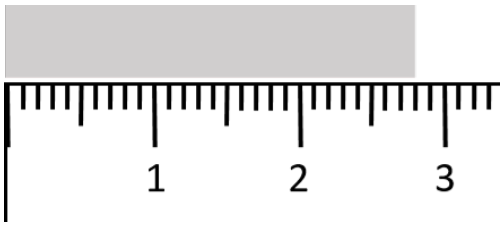

Figure 3.1: Measuring with a ruler.

This general rule ("half of the smallest division") can be used for all measurements devices where numbers are read off a scale, including digital scales. For example, a digital voltmeter may read 12.1 V (0.1 V being the smallest displayed digit) and you would then quote the measurement as (12.10 ± 0.05) V. Sometimes, it is clear that you can provide a measurement with a higher accuracy than half of the smallest division (if the divisions are far apart), but you should only do so cautiously. Even using half of the smallest division may significantly underestimate the uncertainty (for example, you may not be able to get the ruler right up against what you are trying to measure, or what you are trying to measure has a finite width that makes it unclear from which point to measure, or the needle on the dial is fluctuating in position, or the digits on the voltmeter are fluctuating). Often, you can get an idea if your uncertainty is reasonable by repeating the measurement, and you should always do so if you can.

This type of uncertainty, on a direct measurement, barring any systematic uncertainties, can usually be taken to mean that you are certain (100% confident), that the value lies within the range

specified by the uncertainty. If you are absolutely certain that you did not miss any systematic effects (e.g. parallax, non-calibrated instrument, clearly defined what you need to measure), this is the only type of measurement where you may be so bold as to bet your life that the true value is in the quoted range. But again, half of the smallest division may be an underestimate, so be careful when you bet your life on things.

3.2.4 Repeatable measurements - "Mean and uncertainty on the mean"

When you have the opportunity to repeat measurements, you should always do so. If you have several measurements of a single quantity, then how do you define the central value and uncertainty? For example, you may be using a stop watch to measure the period of a pendulum, and obtained several values, say T={2.1 s, 2.2 s, 2.1 s, 2.0 s, 2.3 s, 2.1 s}, so what should you quote as your measured value? It is likely that the true value is somewhere between 2.0 s and 2.3 s, so you could quote the result as (2.15 ± 0.15) s (given by the central value of the range with an uncertainty that spans the whole range). In later chapters, we will motivate a different prescription, but we take it here as given:

When you have a set of N independent measurements, $\{x_1, x_2, x_3, \ldots, x_n\}$ of a quantity, x, the quoted result should be given as $\bar{x} \pm \frac{\sigma_x}{\sqrt{N}}$, where

$$\bar{x} \equiv \frac{1}{N}\sum x_i$$
$$\sigma_x \equiv \sqrt{\frac{1}{N-1}\sum (x_i - \bar{x})^2} \tag{3.1}$$

$\bar{x}$ is the arithmetic average (or mean) of the values, and σ_x is called the standard deviation of the measurements (and requires the mean to be known). The standard deviation is a measure of the average distance between the individual measurements and the mean. It is also representative of the uncertainty on a single measurement. The uncertainty on the mean of the measurements is smaller than the standard deviation, since it is divided by $\sqrt{N}$. It makes sense for uncertainty on the mean to be smaller than that of the individual measurements because it combines the information from multiple measurements. Again, here we assumed that there is no systematic uncertainty that skewed all of the results and that the measurements are independent.

The uncertainty that we obtain using the error on the mean does not guarantee that the true value is within the quoted range. In fact, it is carefully designed (under most circumstances) so that there is a 68% chance that the true value lies within the specified range. We will explain this in chapter REF.

EXAMPLE 3-1: What is the central value and uncertainty on T, if the following measurements were performed: T={2.1 s, 2.2 s, 2.1 s, 2.0 s, 2.3 s, 2.1 s}?

This is done easily in python

```
import numpy as np
from math import *

#load the measurements into a numpy array
Ti = np.array([2.1,2.2,2.1,2.0,2.3,2.1])
```

```
#numpy already knows how to fine the mean and standard deviation:
Tavg = np.mean(Ti)
Tstd = np.std(Ti)
Terr = Tstd/sqrt(5)

print("T = ","{:.2f}".format(Tavg),"+/-","{:.2f}".format(Terr)")
```

which gives $T = (2.13 \pm 0.04)$ s.

3.2.5 Counting experiments - The square root rule

In some experiments, the quantity that you measure is something that you can count. For example, you may want to figure out the average rate of car accidents per month in your town. You may, for example, count that in January, there were 8 accidents in your town. That is a definite number without an obvious uncertainty. However, you would likely be wrong if you claimed that in your town, every month, there will be exactly 8 accidents. So how do you translate your measurement of 8 counts into a more scientific statement, such as "*In my town, I determined the average rate of accidents per month to be 8 $\pm$ X*"?

It turns out that the answer is rather simple: **if the events occur at random times but with a well defined average rate, the uncertainty on the number of counts, N, is the square root of the number of counts, $\sqrt{N}$.**. In our example, you would conclude that the rate of accidents per month is 8 ± 2.8. The square root uncertainty is also constructed such that the true value has approximately a 68% chance of occurring in the quoted range. The value is close to 68% when N is large (bigger than about 75), and reduces as N becomes smaller. We will examine this in more detail in Chapter REF, when we consider the Poisson distribution.

3.3 Overestimating uncertainties is also bad!

With all of these caveats in determining the uncertainties on measurements, you may be tempted to just round up all of your uncertainties to be on the safe side. That way, you will have a good chance of concluding in your lab report that the results agree with the laws of physics! In principle, for an undergraduate laboratory, there will not be much consequences in doing this (you're not really expected to revolutionize our understanding of physics). Hopefully, there is no need to convince you that this is not a very scientific attitude, and you certainly should not think like this! Similarly, it should be obvious that if you really are testing the laws of physics (with the hope of discovering something new), then you really want to have the most precise result possible, with the smallest uncertainties.

If you understand your uncertainties and have defined them precisely, then you would only be tempted to inflate them after the fact, when you see that your results disagrees with the hypothesis that you are testing. If your uncertainties are well defined and your result does not agree with the hypothesis, then you must have missed a systematic effect. In fact, you can tell from the disagreement between your result and the hypothesis how big that systematic effect must be. This is in fact an opportunity to think through the experiment and try to identify where the systematic error arose (or find the mistake in your calculations). This approach of "discovering a systematic" uncertainty can apply in undergraduate laboratories, where you are highly confident that you will

not violate the laws of physics that you are testing (and thus that you can reasonably expect a certain result). However, in "real" experimental physics, one does not have a hypothesis that can be taken as "definitely correct" and this is why systematic uncertainties are particularly dangerous, since there is no guaranteed way to find them. This is also why, in general, you want to make sure your uncertainties are not overestimated, so that others can verify your results as accurately as possible. Please become a good scientist and don't overestimate your uncertainties.

3.4 Reporting uncertainties, significant figures

In the above example (Example 3-1), we asked python to format the output to 2 decimal places. By default, python would have output the result as $T = (2.13333333333 \pm 0.0421637021356)\,\mathrm{s}$. In this case, it would not make sense to present that many decimal places on the central value, when the uncertainty is much bigger than most of them. Similarly, it doesn't make sense to claim that we have determined the uncertainty that precisely either. You should quote results and uncertainties showing only "significant" digits, and in a way that is the most legible (for example by using scientific notation).

Again, we don't have a perfect rule for determining significance of digits, but in general, **your uncertainty should have 1 (and rarely more) "significant figures"**. By significant figure, we mean digits after leading zeros or before trailing zeros. For example, in the number 00019.21000, only 19.21 are "significant", and we say that the measurement is determined to four significant figures. Once the number of significant figures is determined on the uncertainty, one can quote the central value with the same number of figures. When you quote a number, you should ask yourself if you can justify all of the decimals, or if they are instead arbitrary (maybe from a calculation or from a random fluctuation in the apparatus); if you don't think that you can reproduce all of the decimals, you should likely consider them as non-significant and not present them.

Usually, we are interested in evaluating the precision of a result; that is, how big the uncertainty is relative to the central value. This tells us something about how accurate the result may be. The "relative" uncertainty (sometimes called the "fractional uncertainty") is a good measure of the precision, and should generally be quoted with your final result. The **relative uncertainty is quoted as a percent and is given by the uncertainty divided by the central value**. In order to help your audience appreciate the precision of a measurement even without giving the relative uncertainty, you should try to format the number using scientific notation and factoring out the units in a way that it is easy to estimate the relative uncertainty. For a measurement of m=12000 g $\pm$ 23 g, you could format the number as m=(120.00 $\pm$ 0.23) $\times 10^2$ g; that way, it is quite obvious that the relative uncertainty is of order 0.2%.

Generally, if the first significant digit in the uncertainty is small (say, less than 5) and the second digit is large (say, more than 5), then it may make sense to include a second digit since that digit results in an "appreciable change" in the uncertainty. For example if an uncertainty is determined to be 0.15, then it might make sense to quote it as 0.15 instead of 0.2, since the additional decimal changes the uncertainty by 25% (an arguably appreciable change). Conversely, if the uncertainty is determined to be 0.92, then it might as well be quoted as 0.9, since the 0.02 results in a small relative change in the uncertainty. In addition to considering the relative change in the uncertainty itself when deciding whether to include an additional digit, you can consider the change in the relative uncertainty of the measurement. For example, if a measurement is determined to be

10001.12 ± 0.15 (a relative uncertainty of $\frac{0.15}{10001.12} = 0.001\%$), then quoting it as 10001.0 ± 0.2 probably does not make a significant difference in your conclusion. So what is an "appreciable relative change"? That is of course arbitrary, but for high precision measurements anything above a 1% change may be considered appreciable, while in undergraduate laboratories, that threshold could be closer to 5%.

EXAMPLE 3-2: Present the following results using appropriate significant figures and give the relative uncertainties on the results: 2.1124 ± 0.156, 100000 ± 3900, $0.000084341387 \pm 0.000000322331$, 0.000091 ± 0.1

- $2.1124 \pm 0.156 \rightarrow 2.11 \pm 0.16$ *(8% relative) - Include a second digit in the uncertainty since it is an appreciable change in both the uncertainty and the relative uncertainty*
- $100000 \pm 3900 \rightarrow (100 \pm 4) \times 10^3$ *(4% relative) - Change to scientific notation to easily see the size of the relative uncertainty, only use 1 digit on the uncertainty since changing between 3.9% and 4% relative uncertainty is not significant.*
- $0.000084341387 \pm 0.0000003223 \rightarrow (84 \pm 0.3) \times 10^{-6}$ *(0.4% relative) - Scientific notation to make it more legible, only show 1 digit for the uncertainty since the relative uncertainty is already very small.*
- $0.000091 \pm 0.1 \rightarrow 0 \pm 0.1$ *(relative not defined) - The uncertainty being so much bigger than the value, we cannot distinguish it from zero, so we might as well quote it as zero. With a zero central value, we cannot determine the relative uncertainty.*

3.5 Comparing measured quantities

Now that we have some grasp on uncertainties, or at least an understanding that uncertainties are not easy to define accurately, how do we come to conclusions based on measured values? How do we conclude if two numbers with uncertainties agree? Again, we have to remember that we are scientists and that we work within the Scientific Method. We really need to think in terms of "does a measurement exclude a hypothesis?". When comparing two measurements, we must then ask ourselves if one measurement is "incompatible" with the other one.

Suppose that two teams have each measured the mass of a dark matter particle. Team 1 measured the mass to be m=(10.4 ± 1) GeV and Team 2 measured the mass to be m=(12.5 ± 1) GeV. Do the results agree, or should we conclude that there may in fact exist two different dark matter particles? Well, it depends! What is represented by the uncertainties? If both teams are betting their lives that the mass is definitely included within the range that they quote, then the measurements disagree (since the highest compatible value with Team 1 is 11.4 and the lowest value compatible with Team 2 is 11.5). In general however, results are reported with some "degree of confidence" that the true value lies within the specified range. Often, that degree of confidence is around 70% (which we will motivate in chapter REF). If the two teams are both quoting there result with a 70% confidence that the true value lies in their quoted range, there is some room for the value to be outside of the respective ranges and for the two measurements to agree (and indeed, we would conclude that the two measurements, at this level of precision, are consistent with each other).

We can also use the relative difference between the measurements and compare that to the relative

uncertainty on the measurements to get an idea of the agreement in terms of the precision. Team 1 has a relative uncertainty on their measurement of $\frac{1.0}{10.4} = 9.6\% \sim 10\%$, and Team 2 has a relative uncertainty in their measurement of $\frac{1.0}{12.5} = 8\%$; the two measurements thus have similar precision, around 10%. The relative difference between the central value of the two measurements is $\frac{12.5-10.4}{12.5} = 16.8\% \sim 17\%$. In this analysis, given that the difference between the measurements is of order the precision of the measurements (17% vs 10%), one would not conclude that the measurements are incompatible, even if the ranges covered by the uncertainties do not overlap.

As a scientific community, we would want to understand if both teams have correctly represented their uncertainties, we would want to see the experiments repeated to be more precise, and if the difference subsists, we would first assume that there is a systematic uncertainty in at least one of the experiments that is shifting their value one way. Only after all of those issues have been addressed, and if the difference between the two measurements was many times the uncertainty between them, we would consider accepting that there may be two different dark matter particles.

Again, we find that we have to be very precise when comparing results, as the conclusions that we draw are dependent on how the uncertainties are determined. Even worse, the standard that we apply to make a conclusion may well depend on how important that conclusion is. Let's say that instead of measuring the mass of a dark matter particle, Team 1 simply claims that they have finally discovered dark matter (which would certainly earn them a Nobel Prize). In their results, they claim that they expected 5 ± 2 background events, and instead observed 11 events. They claim that the excess of 6 events over the expected background of 5 is a clear indication of dark matter events, as it is larger than the expected background by 3 times the uncertainty on the background rate. Indeed, a measurement of 11 is not very consistent with 5 ± 2, so the hypothesis that they observed an unusually large fluctuation in the background is difficult to support. We also assume that the community has thoroughly analysed their background estimate and agrees that 5 ± 2 is reasonable. As we will see in chapter REF, there is approximately a 0.3% chance that the background actually did fluctuate by 3 times its uncertainty. Since we have to significantly change the laws of physics if the measurement is correct, we may not be satisfied with a 0.3% chance that the measurement is wrong. And indeed, in particle physics, the standard is often that a measurement must disagree with the null hypothesis by 5 times its uncertainty in order to invalidate the null hypothesis (in this case, the null hypothesis is that Team 1 only observed background events).

3.6 Summary

Hopefully this chapter will have convinced you of the following points, which you should typically think about when discussing your results:

- There is no perfect rule to determine the size of uncertainties in a measurement; the goal is to make sure that the uncertainties are justified.

- Overestimating uncertainties is just as bad as underestimating them. If you think your uncertainties are too small, understand why and look for systematic effects.

- If possible, you should perform independent measurements to gauge whether your uncertainties are well estimated.

- It is absolutely critical to present results precisely (explain how uncertainties were determined), otherwise it is impossible to use the results.

- Systematic errors are the plague of experimental physics and the design of experiments should focus on minimizing them.

- Random errors can be minimized by repeating measurements (but some measurements may not be easy to repeat).

- When you quote a value with uncertainty, $x \pm \delta x$, you should mean that it is equally likely for the true value to be on either side of x (we call this a random error).

- Comparing the relative difference between values to the relative uncertainties on the values gives a good idea of whether the values agree.

- The ultimate comparison between numbers with uncertainties involves some degree of subjectivity and depends on: what the uncertainties are defined to be, whether systematic errors have been considered, and what standard needs to be met for a hypothesis to be rejected.

With the caveat that you need to consider systematic uncertainties and that this may underestimate your uncertainty, it is usually safe to use half of the smallest division from an instrument as the uncertainty on a direct measurement from the instrument. With the caveats in mind, you can usually be certain that the true value lies in the quoted range (or more precisely, you should specify the range so that you are certain that the true values is in the range).

We saw that for a set of N independent measurements, $\{x_1, x_2, x_3, \ldots, x_n\}$ of a quantity, x, the quoted result should be given as $\bar{x} \pm \frac{\sigma_x}{\sqrt{N}}$, where

$$\begin{aligned} \bar{x} &\equiv \frac{1}{N} \sum x_i \\ \sigma_x &\equiv \sqrt{\frac{1}{N-1} \sum (x_i - \bar{x})^2} \end{aligned} \tag{3.2}$$

σ_x is called the standard deviation of the measurements and is representative of the uncertainty on a single measurement. $\bar{x}$ is the mean of the measurements, and $\frac{\sigma_x}{\sqrt{N}}$ is the uncertainty on the mean. If the measurements are independent and there are no systematic effects, there is a 68% chance that the true value lies within the specified range.

In a counting experiment, where events occur at random times but with a well defined average rate, the uncertainty on the number that you counted, N, is simply the square root, $\sqrt{N}$. This uncertainty is constructed so that it is the standard deviation that you would get if you measured N many times. If N is large (say bigger than about 80), the range given by $N \pm \sqrt{N}$ will have the true value in it 68% of the time, if N is small, that percentage will be somewhat reduced.

4 Error Propagation

In this chapter, we show how to propagate uncertainties from "direct" measurements to uncertainties in quantities calculated from those measurements.

4.1 The Min-Max method

We will look at the "Min-Max" method as an introduction to propagating uncertainties before looking at a more general method that works in all cases. The Min-Max method is based on considering the maximal change in a calculated quantity that can result from the variation of the measured quantities.

4.1.1 Summing and subtracting

First, we consider the case of summing (or subtracting) two numbers with uncertainties. For example, we may be placing two masses (m_1, m_2) on a balance and need to know their sum, $M = m_1 + m_2$. If each individual mass, m_i, has an uncertainty of σ_{m_i}, then the absolute maximal value that the sum of the masses could have is:

$$M^{max} \equiv (m_1 + \sigma_{m_1}) + (m_2 + \sigma_{m_2}) \tag{4.1}$$

which assume that in each case, the "true" value of the masses was at the maximum end of the range defined by our uncertainty. Similarly, if the "true" value of the masses was at the minimum end of the range, the minimum value that the sum could be is:

$$M^{min} \equiv (m_1 - \sigma_{m_1}) + (m_2 - \sigma_{m_2}) \tag{4.2}$$

Now that we know the range over which M must lie, we can define a value of M with uncertainty, σ_M, to correspond to the center of the range with an uncertainty that allows the range to be covered:

$$\begin{aligned} M &\equiv \frac{1}{2}(M^{max} + M^{min}) \\ \sigma_M &\equiv \frac{1}{2}(M^{max} - M^{min}) \end{aligned} \tag{4.3}$$

Substituting the expressions for M^{max} and M^{min}, we have:

$$\begin{aligned} M &= \frac{1}{2}\left([(m_1+\sigma_{m_1})+(m_2+\sigma_{m_2})]+[(m_1-\sigma_{m_1})+(m_2-\sigma_{m_2})]\right) \\ &= m_1+m_2 \\ \sigma_M &= \frac{1}{2}\left([(m_1+\sigma_{m_1})+(m_2+\sigma_{m_2})]-[(m_1-\sigma_{m_1})+(m_2-\sigma_{m_2})]\right) \\ &= \sigma_{m_1}+\sigma_{m_2} \end{aligned} \tag{4.4}$$

The central value of the calculated quantity, M, is thus given by the sum of m_1 and m_2, and the uncertainty, σ_M, is given by summing the uncertainties σ_{m_1} and σ_{m_2}.

It is easy to show that if there are more than two measurements, say N measurements of quantities x_i, each with uncertainty σ_{x_i}, the best estimate of the sum, X, and its uncertainty, σ_X, are given by:

$$\begin{aligned} X &= \sum_{i=1}^{i=N} x_i \\ \sigma_X &= \sum_{i=1}^{i=N} \sigma_{x_i} \end{aligned} \tag{4.5}$$

EXAMPLE 4-1: Using the Min-Max method, show that the best estimate and uncertainty on the difference of two measurements, x_1 and x_2, with uncertainty σ_{x_1} and σ_{x_2}, are given by $X = x_1 - x_2$ and $\sigma_X = \sigma_{x_2} + \sigma_{x_2}$

The maximum value of the difference is given when x_1 has the biggest value in its range, while x_2 has the smallest values in its range:

$$X^{max} = (x_1+\sigma_{x_1}) - (x_2-\sigma_{x_2})$$

and the minimum value that the difference can have is given by:

$$X^{min} = (x_1-\sigma_{x_1}) - (x_2+\sigma_{x_2})$$

The central value and uncertainties are then:

$$\begin{aligned} X &= \frac{1}{2}(X^{max}+X^{min}) \\ &= \frac{1}{2}\left([(x_1+\sigma_{x_1})-(x_2-\sigma_{x_2})]+[(x_1-\sigma_{x_1})-(x_2+\sigma_{x_2})]\right) \\ &= x_1-x_2 \\ \sigma_X &= \frac{1}{2}(X^{max}-X^{min}) \\ &= \frac{1}{2}\left([(x_1+\sigma_{x_1})-(x_2-\sigma_{x_2})]-[(x_1-\sigma_{x_1})-(x_2+\sigma_{x_2})]\right) \\ &= \sigma_{x_1}+\sigma_{x_2} \end{aligned}$$

as required.

We see that for addition and subtraction, the uncertainties on the individual measurements are always summed, whereas the central value is obtained by summing (or subtracting) the individual measurements as if they had no uncertainty. It should be clear that with this method, the final uncertainty is always "conservative", that is, it is never going to be too small. In fact, it is almost always too big. In our example with the sum of two masses, if our measurements of m_1 and m_2 are truly independent and the errors on the individual measurements are truly random errors, it is very unlikely that the sum is actually at one of the extreme ranges given by the uncertainty. On average, it is much more likely for the error in the measurement of one mass to somewhat offset the error in the measurement of the other (unless there is a systematic bias). Our uncertainty estimated this way is almost always an overestimate of the true uncertainty on the sum, and is in fact an upper limit on the uncertainty.

4.1.2 Multiplication and division

Suppose we measured two sides of a rectangle, l_1 and l_2 with uncertainties σ_{l_1} and σ_{l_2}, and want to know the area of the rectangle, $A = l_1 l_2$, and its uncertainty, σ_A. Multiplication (and division) are best handled by using the relative uncertainties, $\frac{\sigma_{l_1}}{l_1}$ and $\frac{\sigma_{l_2}}{l_2}$. We can write the biggest and smallest possible values for, say, l_1, in terms of the relative uncertainty as:

$$\begin{aligned} l_1^{max} &= l_1 \left(1 + \frac{\sigma_{l_1}}{l_1}\right) \\ l_1^{min} &= l_1 \left(1 - \frac{\sigma_{l_1}}{l_1}\right) \end{aligned} \tag{4.6}$$

The highest possible value of the area of the rectangle is thus:

$$\begin{aligned} A^{max} &= l_1^{max} l_2^{max} \\ &= l_1 \left(1 + \frac{\sigma_{l_1}}{l_1}\right) l_2 \left(1 + \frac{\sigma_{l_2}}{l_2}\right) \\ &= l_1 l_2 \left(1 + \frac{\sigma_{l_1}}{l_1} + \frac{\sigma_{l_2}}{l_2} + \frac{\sigma_{l_1}}{l_1}\frac{\sigma_{l_2}}{l_2}\right) \\ &\approx l_1 l_2 \left(1 + \left[\frac{\sigma_{l_1}}{l_1} + \frac{\sigma_{l_2}}{l_2}\right]\right) \end{aligned} \tag{4.7}$$

where in the last line, we made the approximation that if each relative uncertainty is small, then their product $(\frac{\sigma_{l_1}}{l_1}\frac{\sigma_{l_2}}{l_2})$ is really small and negligible. This is reasonable, for example if each relative uncertainty if of order 10% (0.1), then their product (0.01) is certainly negligible when compared to their sum (0.2). And often, one would have relative uncertainties well below 10%.

The minimum value of the area is found similarly:

$$\begin{aligned} A^{min} &= l_1^{min} l_2^{min} \\ &= l_1 \left(1 - \frac{\sigma_{l_1}}{l_1}\right) l_2 \left(1 - \frac{\sigma_{l_2}}{l_2}\right) \\ &= l_1 l_2 \left(1 - \frac{\sigma_{l_1}}{l_1} - \frac{\sigma_{l_2}}{l_2} - \frac{\sigma_{l_1}}{l_1}\frac{\sigma_{l_2}}{l_2}\right) \\ &\approx l_1 l_2 \left(1 - \left[\frac{\sigma_{l_1}}{l_1} + \frac{\sigma_{l_2}}{l_2}\right]\right) \end{aligned} \tag{4.8}$$

The expressions for A^{min} and A^{max} are symmetric about the central value of $l_1 l_2$, so we can easily write the value of A and its uncertainty as:

$$A \pm \sigma_A = l_1 l_2 \left(1 \pm \left[\frac{\sigma_{l_1}}{l_1} + \frac{\sigma_{l_2}}{l_2}\right]\right) \tag{4.9}$$

We can thus identify that the central value and uncertainty on A are given by:

$$\begin{aligned} A &= l_1 l_2 \\ \sigma_A &= A\left(\frac{\sigma_{l_1}}{l_1} + \frac{\sigma_{l_2}}{l_2}\right) \end{aligned} \tag{4.10}$$

More conveniently, we can see that the relative uncertainty of A is given by summing the relative uncertainties on l_1 and l_2:

$$\frac{\sigma_A}{A} = \frac{\sigma_{l_1}}{l_1} + \frac{\sigma_{l_2}}{l_2} \tag{4.11}$$

EXAMPLE 4-2: Use the Min-Max method to determine the uncertainty on the quotient, $Q = \frac{x_1}{x_2}$, of two numbers, x_1 and x_2, that have uncertainties σ_{x_1} and σ_{x_2}

We proceed in the same way as we did for the product, using the relative uncertainties. The biggest value that the quotient can have is:

$$\begin{aligned} Q^{max} &= \frac{x_1^{max}}{x_2^{min}} \\ &= \frac{x_1\left(1 + \frac{\sigma_{x_1}}{x_1}\right)}{x_2\left(1 - \frac{\sigma_{x_2}}{x_2}\right)} \end{aligned}$$

Again, we have to assume that the relative uncertainties are small, in particular, we make use of the binomial approximation:

$$(1+x)^\alpha \approx 1 + \alpha x$$

which applies when x is small. In our case:

$$\frac{1}{1 - \frac{\sigma_{x_2}}{x_2}} \approx 1 + \frac{\sigma_{x_2}}{x_2}$$

Q^{max} is thus given by:

$$\begin{aligned} Q^{max} &= \frac{x_1}{x_2}\left(1 + \frac{\sigma_{x_1}}{x_1}\right)\left(1 + \frac{\sigma_{x_2}}{x_2}\right) \\ &\approx \frac{x_1}{x_2}\left(1 + \left[\frac{\sigma_{x_1}}{x_1} + \frac{\sigma_{x_2}}{x_2}\right]\right) \end{aligned}$$

where, again, we assumed that the product of the relative uncertainties was negligible. The minimum value

of the quotient is found in a similar way:

$$\begin{aligned} Q^{min} &= \frac{x_1^{min}}{x_2^{max}} \\ &= \frac{x_1\left(1-\frac{\sigma_{x_1}}{x_1}\right)}{x_2\left(1+\frac{\sigma_{x_2}}{x_2}\right)} \\ &\approx \frac{x_1}{x_2}\left(1-\frac{\sigma_{x_1}}{x_1}\right)\left(1-\frac{\sigma_{x_2}}{x_2}\right) \\ &\approx \frac{x_1}{x_2}\left(1-\left[\frac{\sigma_{x_1}}{x_1}+\frac{\sigma_{x_2}}{x_2}\right]\right) \end{aligned}$$

where we again used the binomial expansion and the fact that the product of the relative uncertainties is negligible. We end up with a very similar result as for multiplication, where the quotient can be written as:

$$Q \pm \delta Q = \frac{x_1}{x_2}\left(1 \pm \left[\frac{\sigma_{x_1}}{x_1}+\frac{\sigma_{x_2}}{x_2}\right]\right)$$

and we find that the central value of the quotient is just the quotient of the two numbers without uncertainty and that the relative uncertainty on Q is found by summing the relative uncertainties of the two numbers:

$$\begin{aligned} Q &= \frac{x_1}{x_2} \\ \frac{\delta Q}{Q} &= \frac{\sigma_{x_1}}{x_1}+\frac{\sigma_{x_2}}{x_2} \end{aligned}$$

In summary, for multiplication and division, we sum the relative uncertainties to obtain the relative uncertainty on the result, whereas for summation and subtraction we sum the absolute uncertainties.

We also make the same point for multiplication and division as we did for summation and subtraction: the Min-Max method overestimates the uncertainties. It is very unlikely for the "true" values of the measured quantities to be at the extrema of the uncertainty range, and this method really gives an upper limit on the uncertainty of the result.

4.1.2.1 Uncertainties in functions

We can also apply the formalism of the Min-Max method to propagate the uncertainty in a function, although there may not be a trivial way to do it. If we have a function, $F(x, y)$ of two measured quantities, x and y that have uncertainties σ_x and σ_y, it may not be trivial to find out the particular combination of x and y that maximize and minimize F. If these combinations are found, then F and its uncertainty σ_F, are given by:

$$\begin{aligned} F &\equiv \frac{1}{2}(F^{max}+F^{min}) \\ \sigma_F &\equiv \frac{1}{2}(F^{max}-F^{min}) \end{aligned} \tag{4.12}$$

If F is a monotonic function in both x and y, then the situation is relatively straightforward. For example, if F monotonically *increases* as a function of both x and y (e.g. $F(x, y) = \sqrt{x^2+y^2}$),

then we have the simple case where:

$$F^{max} = F(x^{max}, y^{max})$$
$$F^{min} = F(x^{min}, y^{min})$$

In more complicated situations, it may be necessary to find out the answer numerically, which thankfully is easy with a computer!

EXAMPLE 4-3: Given the function $F(x, y) = \sin(x + \log y)$, and the measured values of $x = 0 \pm 1$ and $y = 0.5 \pm 0.2$, use the Min-Max method to find the uncertainty on F

Since this is not a trivial function, we can solve this numerically using python:

```
import numpy as np
import pylab as pl
from math import *
%matplotlib inline

#The function for which we want to find the uncertainties
def F(x,y):
    return np.sin(x+np.log(y))

#x and y, and their uncertainties:
x,delta_x=0.,1
y,delta_y=0.5,0.2

#generate 50 values within the range for x and y to scan the values of F
xvalues=np.linspace(x-delta_x,x+delta_x,50)
yvalues=np.linspace(y-delta_y,y+delta_y,50)

#initialize Fmin and Fmax outside of the possible range (since -1<F<1)
Fmin,Fmax=10,-10

#Now scan all the values of x and y to find the minimum and maximum of F
xmin,xmax,ymin,ymax=0,0,0,0
for xi in xvalues:
    for yi in yvalues:
        f=F(xi,yi)
        if f<Fmin:
            Fmin=f
            xmin=xi
            ymin=yi
        if f>Fmax:
            Fmax=f
            xmax=xi
            ymax=yi

print("The range of F is between {0:.2f} and {1:.2f}".format(Fmin,Fmax))
print("F is thus given by: {0:.2f} +/- {1:.2f}".format(0.5*(Fmax+Fmin),0.5*(Fmax-
    Fmin)))
print("These occurred at x,y = {0:.2f},{1:.2f}, and {2:.2f},{3:.2f}".format(xmin,
    ymin,xmax,ymax))
```

The code finds that the minimum of $F^{min} = -1.00$ occurs at $(x, y) = (-0.92, 0.52)$, and the maximum, $F^{max} = 0.60$ occurs at $(x, y) = (1.00, 0.70)$, giving $F = -0.20 \pm 0.80$.

4.1.2.2 Summary and comments on the Min-Max method

The Min-Max method can be used to "propagate" the uncertainties of measured quantities through summation, subtraction, multiplication and division. In these cases, it is straightforward to apply as the central value of the result is always given by the central values of the measurements without uncertainties (e.g. the central value of a sum of measurements is the sum of the central values of the measurements). **In the case of summation and subtraction, the uncertainty on the result is given by the sum of the uncertainties on the measurements. In the case of multiplication and division, the relative uncertainty of the result is given by the sum of the relative uncertainties of the measurements**. When wants to evaluate the uncertainty in a function, the same formalism can be applied, but the answer may need to be obtained numerically.

The Min-Max method gives the uncertainty on the result under the "worst case scenario" that the uncertainties on the measurements are at the extreme values allowed in the uncertainty range, in such a way to make the biggest change in the result. If the errors on the measurements are truly random, it is very unlikely for all the measurements to be at the extrema of their allowed range, so the **Min-Max method gives a maximum possible value for the uncertainty on the result**.

4.2 Adding in quadrature

As we discussed, it is quite unlikely that the uncertainty on a calculated number is as large as that given by the Min-Max method. For example, in the case of a sum of masses, when we use the Min-Max uncertainty, we are effectively claiming that we think it is quite possible that both measurements were either high or low, which is quite unlikely in reality. This is however not impossible. If there was a bias in the scale that we used to weigh the masses, it is in fact quite possible that both measurements are indeed at one end of the range that we quoted with the uncertainties. If this were the case, then the Min-Max method gives a reasonable uncertainty on the sum of the masses. However, if there was indeed a systematic effect that resulted in the true value being systematically at the higher or lower end of our quoted range, then we should rethink how we determined the uncertainties.

Most often, the uncertainties that we determine have some sort of "statistical nature" to them. For example, when we quote a measured values as $a \pm \delta a$, we usually mean that a is equally likely to be bigger or smaller than the central value. In fact, when we determine a and δa as the mean and error on the mean of multiple measurements, we are explicitly assuming that the true value is normally distributed about the central value. We will define in later chapters what we really mean by "normally distributed", but it is related to saying that there is a 68% chance that the true value is in the quoted range, and that it is more likely for it to be in the center of the range than at the edges. It turns out that most measurements are normally distributed, which justifies the following procedure for adding uncertainties.

If the the uncertainties on multiple measurements are normally distributed and are independent of each other, then those uncertainties should be added in quadrature, rather than added. Addition in quadrature is performed by adding the squares of the uncertainties and then taking the square root of the sum. For the case of two masses being added together, $M = m_1 + m_2$, the uncertainty on M is given by:

$$\sigma_M = \sqrt{\sigma_{m_1}^2 + \sigma_{m_2}^2} \tag{4.13}$$

This is also true for the case of subtraction. For multiplication and division, instead of adding the relative uncertainties, we add the relative uncertainties in quadrature. For example, if the area of a rectangle is given by $A = l_1 l_2$, then the relative uncertainty in A is given by:

$$\frac{\sigma_A}{A} = \sqrt{\left(\frac{\sigma_{l_1}}{l_1}\right)^2 + \left(\frac{\sigma_{l_2}}{l_2}\right)^2} \tag{4.14}$$

Addition in quadrature is not limited to only 2 terms; if there are a multiple measurements, all of the uncertainties can be squared and added together before taking the square root.

The uncertainties added in quadrature are always smaller or equal than the uncertainties that are found by straight addition (think of the Pythagorean theorem). Adding in quadrature is only correct if the various measurements are independent of each other. That is, the measurement of one quantity does not affect the measurement of another quantity whose uncertainty is being added in quadrature. If possible, one should check to see if one values is systematically dependent on the other, in other words, if one of the measurement is correlated to the other.

4.3 Uncertainty on a function of one variable

We now introduce a more general way to propagate uncertainties. We start by considering how an uncertainty, σ_x on a measurement of x should propagate to an uncertainty σ_F on a function of $F(x)$. We assume that $F(x \pm \sigma_x)$ is the possible Min-Max range that F could have when x changes by $\pm\sigma_x$. Now this only makes sense if σ_x is small; for example, if $F = \sin x$ and $\sigma_x \approx \pi$, then the maximum change in F most definitely does not occur at $x \pm \sigma_x$. However, in the case were σ_x is small enough, we can always be assured that in that small range, $x \pm \sigma_x$, F is continuous and monotonous ("well behaved"). We can thus use a Taylor series about $F(x)$ to estimate $F(x \pm \sigma_x)$ (and only keep the first order term):

$$F(x \pm \sigma_x) \approx F(x) \pm \frac{dF}{dx}\sigma_x + \ldots \tag{4.15}$$

Which tells us that F and its uncertainty are given by:

$$\begin{aligned} F \pm \sigma_F &= F(x) \pm \frac{dF}{dx}\sigma_x \\ \therefore \sigma_F &= \frac{dF}{dx}\sigma_x \end{aligned} \tag{4.16}$$

That is, F is evaluated at the best estimate value for x and the uncertainty, σ_F is given by the derivative of F with respect to x multiplied by σ_x. The derivative is the rate of change of F with

respect to x, so it makes sense that the change in F due to a change in x is given by the derivative.

EXAMPLE 4-4: Show that the uncertainty on $F = ax$ is given by $\sigma_F = a\sigma_x$

The derivative of F with respect to x is:

$$\frac{dF}{dx} = a$$

Hence the uncertainty is given by:

$$\sigma_F = \frac{dF}{dx}\sigma_x = a\sigma_x$$

as required

EXAMPLE 4-5: Show that the uncertainty on $F = \sin\theta$ is given by $\sigma_F = \cos\theta\delta\theta$

The derivative of F with respect to θ is:

$$\frac{dF}{d\theta} = \cos\theta$$

Hence the uncertainty is given by:

$$\sigma_F = \frac{dF}{d\theta}\delta\theta = \cos\theta\delta\theta$$

as required. Note that this only works when θ is expressed in radians!

4.4 The general case: functions of multiple variables

We now consider the case of a function of multiple variables, for example, $F(x, y)$. Again, we estimate $F(x \pm \sigma_x, y \pm \sigma_y)$ as the possible Min-Max range and assume that both σ_x, and σ_y must be small (so that $F(x, y)$ is continuous and monotonous in that range of x and y). We again use the Taylor series formula (and limit it to the first order terms):

$$F(x \pm \sigma_x, y \pm \sigma_y) \approx F(x, y) \pm \left(\frac{\partial F}{\partial x}\sigma_x + \frac{\partial F}{\partial Y}\sigma_y\right) \tag{4.17}$$

where we have now used the partial derivatives[1], since F depends on multiple variables. The central value of F is thus found by evaluating $F(x, y)$ at the central value of x and y. The uncertainty from this expression is given by:

$$\sigma_F = \frac{\partial F}{\partial x}\sigma_x + \frac{\partial F}{\partial Y}\sigma_y \text{ (upper limit)} \tag{4.18}$$

This derivation however assumed that both x and y have changed in the "worst possible way" so as to lead to the biggest possible change in F. It is conceivable that the true values of x and y somewhat offset each other in the change that they cause to F. For example, if $F(x, y) = x + y$,

[1] Recall that when you take the partial derivative with respect to one variable, you treat all the others as constants

and the true value of x is on the high side of $x \pm \sigma_x$ and the true value of y is in the low side of $y \pm \sigma_y$, then the true value of F is actually quite close to $x + y$. Again, if the uncertainties in x and y are truly random and independent, then it is quite unlikely that they lead to the maximum possibly uncertainty in F. Thus, in the case where the uncertainties are random and uncorrelated, it makes more sense to add them in quadrature:

$$\sigma_F = \sqrt{\left(\frac{\partial F}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial F}{\partial Y}\sigma_y\right)^2} \text{ (x and y random and uncorrelated)} \tag{4.19}$$

This formula applies no matter how many variables F depends on; you can just add more terms to the sum in quadrature.

EXAMPLE 4-6: Use the derivative method to evaluate a formula for the uncertainty, σ_F, on $F(x, y) = \frac{x}{y}$ if x and y have uncertainties σ_x and σ_y,

We have:

$$\begin{aligned} F(x,y) &= \frac{x}{y} \\ \therefore \frac{\partial F}{\partial x} &= \frac{1}{y} \\ \therefore \frac{\partial F}{\partial y} &= \frac{-x}{y^2} \end{aligned}$$

The uncertainty on F is thus:

$$\begin{aligned} \sigma_F &= \sqrt{\left(\frac{\partial F}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial F}{\partial Y}\sigma_y\right)^2} \\ &= \sqrt{\left(\frac{1}{y}\sigma_x\right)^2 + \left(\frac{-x}{y^2}\sigma_y\right)^2} \end{aligned}$$

EXAMPLE 4-7: Use the derivative method to evaluate a formula for the uncertainty, σ_F, on $F(x, y) = \sin(x + \ln y)$ if x and y have uncertainties σ_x and σ_y,

We have:

$$\begin{aligned} F(x,y) &= \sin(x + \log y) \\ \therefore \frac{\partial F}{\partial x} &= \cos(x + \ln y) \\ \therefore \frac{\partial F}{\partial y} &= \cos(x + \ln y)\frac{d}{dy}x + \log y = \cos(x + \log y)\frac{1}{y} \end{aligned}$$

The uncertainty on F is thus:

$$\begin{aligned} \sigma_F &= \sqrt{\left(\frac{\partial F}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial F}{\partial Y}\sigma_y\right)^2} \\ &= \sqrt{(\cos(x + \ln y)\sigma_x)^2 + \left(\cos(x + \log y)\frac{1}{y}\sigma_y\right)^2} \end{aligned}$$

EXAMPLE 4-8: Use the derivative method to (numerically) evaluate F and its uncertainty, σ_F, for the case where $F(x, y, z) = \sin(x + \ln y)\cos(z)\sqrt{x^2 + y^2 + 2z}$, where the following values have been measured: $x = 8 \pm 0.1$, $y = 3 \pm 0.2$, and $z = 10 \pm 0.3$

In this case, it is much easier to just use symbolic computation in python.

```
import sympy as sym

#Declare x,y,z and their uncertainties as symbols
x,y,z=sym.symbols('x y z')
dx,dy,dz=sym.symbols('dx dy dz')

#Define our function:
F=(sym.sin(x+sym.ln(y))*sym.cos(z))*(sym.sqrt(x**2+y**2+2*z))

#Get the partial derivatives:
dFdx=sym.diff(F,x)
dFdy=sym.diff(F,y)
dFdz=sym.diff(F,z)

#The quadrature sum is given by:
dF=sym.sqrt((dFdx*dx)**2+(dFdy*dy)**2+(dFdz*dz)**2)

#Create an array of tuples for the numerical values:
values=[(x,8),(dx,0.1),(y,3),(dy,0.2),(z,10),(dz,0.3)]

#Substitute the values into our expressions for F and dF
Fval=F.subs(values)
dFval=dF.subs(values)

#Evaluate F and dF numerically, and print:
print("F = {:.2f} +/- {:.2f}".format(sym.N(Fval),sym.N(dFval)))
```

This gives the result $F = -2.59 \pm 1.02$, which would not have been fun to evaluate by hand! You can easily modify this code to calculate the error on any function.

4.5 Correlated uncertainties

We saw the two extreme cases for combining the uncertainties on measured quantities. In the worst case scenario, when one measurement is completely correlated with the other, the uncertainties should be added together. If the measurements are completely independent from each other, then the uncertainties should be added in quadrature. But what if they are "somewhat" correlated? That is, what if measuring one quantity to be high usually makes the other one high as well, but not always? In the next chapter, we will formalize our definition of the "covariance", σ_{xy}, which is a measure of how correlated two quantities are. If we have two quantities, x and y, that we measure simultaneously N times, giving us two sets of measurements $x_i = \{x_1, x_2, \ldots, x_N\}$ and $y_i = \{y_1, y_2, \ldots, y_N\}$, the covariance, σ_{xy}, is defined to be:

$$\sigma_{xy} \equiv \frac{1}{N-1} \sum_{i=1}^{i=N} (x_i - \bar{x})(y_i - \bar{y}) \tag{4.20}$$

where $\bar{x}$ and $\bar{y}$ are the means of the measurements of x and y, respectively.

Suppose that we are interested in the sum of two masses $M = m_1 + m_2$, and that we have made several measurements of m_1 and m_2 which are tabulated in Table 4.1. For completeness, we also added a third column with the sum of the masses.

m_1	m_2	$m_1 + m_2$
399.3	193.2	592.5
404.6	205.1	609.7
394.6	192.6	587.2
396.3	194.2	590.5
399.6	196.6	596.2
404.9	201.0	605.9
387.4	184.7	572.1
404.9	215.2	620.1
398.2	203.6	601.8
407.2	207.8	615.0

Table 4.1: Measurements of masses and their sum.

We have determined the mean and the standard deviation (Equation 3.1) for the masses to be $\bar{m}_1 = 399.7\,\text{kg}$, $\sigma_{m1} = 6.01\,\text{kg}$, $m_2 = 199.4\,\text{kg}$, and $\sigma_{m2}8.87$ (we kept a non-significant figure here to propagate in our uncertainty calculations). So what would should we quote for M and its uncertainty? There are three obvious options:

1. The mean and error on the mean of the sum value gives (simply using the third column from the table as independent data) $M = 599.1 \pm 4.5$, which is certainly correct.

2. Adding the standard deviations of m_1 and m_2 in quadrature and then dividing by the square root of the number of measurements gives $M = 599.1 \pm 3.39$, which is an underestimate.

3. Adding the standard deviations of m_1 and m_2 and then dividing by the square root of the number of measurements gives $M = 599.10 \pm 4.71$, which is an over estimate.

As expected, the error from the sum in quadrature is smaller than the error from the straight sum. Treating the third column as individual measurements and using the error on the mean from those values must certainly be correct, and should in fact be most representative of the actual variation that we get in the sum. This gives an uncertainty that is larger than the quadrature sum, but not as large as the straight sum. So we have a situation that is closer to the worst case scenario of having to add the uncertainties, but not the worst possible case. We say that the two values of m_1 and m_2 are "correlated".

We can see this graphically very easily by plotting a scatter plot of m_1 and m_2, as in Figure 4.1, where we can see that on average, m_2 is higher than its mean (199.4) when m_1 is higher than its mean (399.7). There is a trend and the values are not randomly distributed. If m_1 and m_2 really were randomly distributed, we would expect the scatter plot to look like an ellipse.

The correct way to propagate the uncertainty in quantities that are correlated is to first calculate

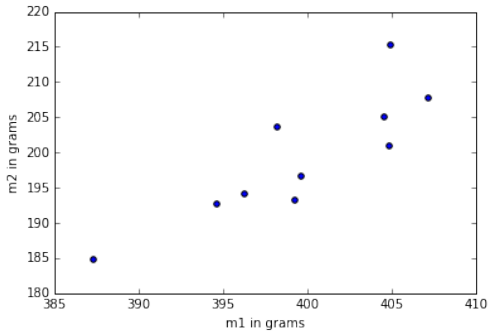

Figure 4.1: Two measurements that are not completely independent of each other.

the correlation, σ_{xy}, and then to use a more general version of equation 4.19:

$$\begin{aligned} F &= F(x,y) \\ \sigma_F &= \sqrt{\left(\frac{\partial F}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial F}{\partial y}\sigma_y\right)^2 + 2\frac{\partial F}{\partial y}\frac{\partial F}{\partial y}\sigma_{xy}} \end{aligned} \tag{4.21}$$

where the partial derivatives are evaluated at the central values of x and x, and σ_{xy} is the covariance between x and y. The formula is easily extended for more than 2 variables, where there will be one cross term for each pair of correlated variables.

Using equation 4.21 for the example of the two masses, the covariance is determined to be 41.10 leading to an uncertainty in the sum of 4.4, which is comparable to the result obtained by treating the third column in Table 4.1 as independent measurements. It is not exact, as the small number of measurements results in a large uncertainty in the estimate of the covariance. It should be noted that we used formula 4.21 to estimate the standard deviation of the sum, and then divided that by the square root of the number of measurements to obtain the error on the mean of the sum. As we will see in a later chapter, the formulas for combining uncertainties assume that the uncertainties are the standard deviations in the measurements, not the errors on the mean.

The code below compares the various computations for estimating the error on the sum of the masses:

```
import numpy as np
from math import *
#Copy and paste the data, add commas by hand
data=np.array([
399.3, 193.2, 592.5,
404.6, 205.1, 609.7,
394.6, 192.6, 587.2,
396.3, 194.2, 590.5,
399.6, 196.6, 596.2,
404.9, 201.0, 605.9,
387.4, 184.7, 572.1,
404.9, 215.2, 620.1,
398.2, 203.6, 601.8,
407.2, 207.8, 615.0])
#Reshape into the correct format
data=data.reshape(10,3)
#Extract the columns:
```

```
m1=data[:,0]
m2=data[:,1]
msum=data[:,2]
#Get the mean and standard deviations of the columns:
m1mean=m1.mean() #mean
m1std=m1.std(ddof=1) #std
m2mean=m2.mean() #mean
m2std=m2.std(ddof=1) #std
msmean=msum.mean() #mean
msstd=msum.std(ddof=1) #std
msemean=msstd/sqrt(msum.size) #error on the mean
#Calculate the covariance
cov=((m1-m1mean)*(m2-m2mean)).sum()/(m1.size-1)
#Print out comparisons of the uncertainties
print("m1 = {:.2f} +/- {:.2f}".format(m1mean,m1std))
print("m2 = {:.2f} +/- {:.2f}".format(m2mean,m2std))
print("covariance = {:.2f}".format(cov))
print("Treat as independent measurements: M = {:.2f} +/- {:.2f} (correct)".format(
    msmean,msemean))
print("Quadrature: M = {:.2f} +/- {:.2f} (underestimate)".format(msmean,sqrt(m1std*
    *2+m2std**2)/sqrt(msum.size)))
print("Sum error: M = {:.2f} +/- {:.2f} (overestimate)".format(msmean,(m1std+m2std)
    /sqrt(msum.size)))
print("With covariance: M = {:.2f} +/- {:.2f} (correct)".format(msmean,sqrt(m1std**
    2+m2std**2+2*cov)/sqrt(msum.size)))
```

the output is:

```
m1 = 399.70 +/- 6.01
m2 = 199.40 +/- 8.87
covariance = 41.10
Treat as independent measurements: M = 599.10 +/- 4.54 (correct)
Quadrature: M = 599.10 +/- 3.39 (underestimate)
Sum error: M = 599.10 +/- 4.71 (overestimate)
With covariance: M = 599.10 +/- 4.54 (correct)
```

4.6 Averaging numbers with uncertainties

Suppose that two different experiments have measured the same quantity, X, and obtained, say $x_1 = 10 \pm 0.1$ and $x_2 = 11 \pm 1$. We assume that both measurements have correctly estimated their uncertainties, and that these each represent a 68% of the true value being within their quoted uncertainty range. How do we combine both of these measurements into a single average measurement? How do we compute the uncertainty on the average measurement?

If we use the derivative formula to get the average, we find:

$$X = \frac{1}{2}(x_1 + x_2) = 10.5$$
$$\sigma_X = \frac{1}{2}\sqrt{\sigma_{x_1}^2 + \sigma_{x_2}^2} = 0.5$$

which does not make much sense. The first measurement, $x_1 = 10 \pm 0.1$, is clearly more precise than the second one, $x_2 = 11 \pm 1$. Since we trust both measurements, our average must somehow include the fact that x_1 is more precise, and we thus expect that it should be closer to 10 than to

11. We also expect that averaging in a less precise result should not "blow up" the uncertainty on our number; at worst it should do nothing to the uncertainty (providing the second measurement is consistent with the first one, which it is in this case). In the extreme case, you can imagine that x_1 is well measured, and x_2 has measured nothing (in fact they were busy doing other things and didn't have time to perform the experiment); in this extreme case, a bad measurement of x_2 should not impact our knowledge of X or the uncertainty that we obtained from the x_1 measurement. If the two measurements are consistent, we expect that the overall uncertainty should decrease.

The correct way to combine multiple measurements with uncertainties is to perform a weighted average of the measurements, where the uncertainties are used as the weights. We will justify this approach in a later chapter. If n measurements of X, $\{x_1, x_2, \dots\}$ each have uncertainties, $\{\sigma_{x_1}, \sigma_{x_2}, \dots\}$, then the average value is given by:

$$\begin{aligned} X &= \frac{\sum_{i=1}^{i=n} w_i x_i}{\sum_{i=1}^{i=n} w_i} \\ \sigma_X &= \frac{1}{\sqrt{\sum_{i=1}^{i=n} w_i}} \\ w_i &\equiv \frac{1}{\sigma_{x_i}^2} \end{aligned}$$

If we use this formula for the example two measurement that we had above, we get:

$$\begin{aligned} w_1 &= \frac{1}{0.1^2} = 100 \\ w_2 &= \frac{1}{1^2} = 1 \\ X &= \frac{\sum_{i=1}^{i=n} w_i x_i}{\sum_{i=1}^{i=n} w_i} = \frac{100 \times 10 + 1 \times 11}{101} = 10.0099 \\ \sigma_X &= \frac{1}{\sqrt{\sum_{i=1}^{i=n} w_i}} = \frac{1}{\sqrt{101}} = 0.09950 \end{aligned}$$

where we kept the extra decimals to show that the uncertainty get a tiny bit smaller, and the central value is shifted very slightly in the direction of x_2.

4.7 Summary

In this chapter, we showed how to propagate the uncertainties from a set of quantities through to results calculated from those quantities. We introduced the Min-Max method as a way to understand the biggest possible change in a calculated quantity resulting from variations in the measured quantities. We also introduced the "derivative" method which applies more generally than the Min-Max method. We found that the uncertainty in a calculated quantity is obtained by summing uncertainties in measured quantities. We argued that a quadrature sum of the uncertainties was usually more appropriate in the case where the measured quantities are independent from each other and have random errors. We argued that a straight sum generally provides an upper limit on the uncertainty in a calculated result.

The uncertainty given by a sum in quadrature generally assumes that the individual uncertainties are normally distributed; that is, that they signify that each measurement has a 68% chance of being in their quoted range. Consequently, the error given by a quadrature sum also implies that the result has a 68% chance of being within the determined uncertainty.

The only formula for propagating uncertainties that one really needs to know is the derivative method. If a function, $F(x_1, x_2, \dots)$, depends on N quantities, $x_1, x_2, \dots$, each with uncertainties $\sigma_{x_1}, \sigma_{x_2}, \dots$, then the best estimate of F is obtained by evaluated F at the best estimate values of the x_i, and the uncertainty on F, σ_F, is given by the following quadratic sum:

$$\sigma_F = \sqrt{\sum_{i=1}^{i-N} \left(\frac{\partial F}{\partial x_i} \sigma_{x_i} \right)^2} \tag{4.22}$$

where $\frac{\partial F}{\partial x_i}$ is the partial derivative of $F(x_1, x_2, \dots)$ with respect to x_i.

When averaging n measurements x_i, each with their own individual uncertainty σ_{x_i}, the average, X, and its uncertainty, σ_X, are given by:

$$\begin{aligned} X &= \frac{\sum_{i=1}^{i=n} w_i x_i}{\sum_{i=1}^{i=n} w_i} \\ \sigma_X &= \frac{1}{\sqrt{\sum_{i=1}^{i=n} w_i}} \\ w_i &\equiv \frac{1}{\sigma_{x_i}^2} \end{aligned}$$

If we have two quantities, x and y, that we measure simultaneously N times, giving us two sets of measurements $x_i = \{x_1, x_2, \dots, x_N\}$ and $y_i = \{y_1, y_2, \dots, y_N\}$, the covariance, σ_{xy}, is defined to be:

$$\sigma_{xy} \equiv \frac{1}{N-1} \sum_{i=1}^{i=N} (x_i - \bar{x})(y_i - \bar{y}) \tag{4.23}$$

where $\bar{x}$ and $\bar{y}$ are the means of the measurements of x and y, respectively. If the covariance is non-zero, then we should take into account and modify the formula from the derivative method:

$$\begin{aligned} F &= F(X, Y) \\ \sigma_F &= \sqrt{\left(\frac{\partial F}{\partial x} \sigma_x \right)^2 + \left(\frac{\partial F}{\partial Y} \sigma_y \right)^2 + 2 \frac{\partial F}{\partial Y} \frac{\partial F}{\partial Y} \sigma_{XY}} \end{aligned} \tag{4.24}$$

where σ_x and σ_y are the standard deviations of the measurements of x and y, respectively. In practice, one should estimate whether including the covariance makes a difference, and include it if it is significant.

5 Statistics - Presenting Data

5.1 Data and Statistics

Statistical analysis is useful in many field, including physics, because virtually any measurement that is performed multiple times will show a variation, either due to the imperfection of the measuring apparatus, because of variations in the samples, or because the quantity truly has a variance (e.g. speed of gas molecules at a specific temperature and pressure).

For example, if you wanted to model the length of 8 month old cats and set out to measure cats, you would get a variation in your measurements: you would get variations between different cat breeds, you would get variations because the cats will have different lengths of hair, you would get variations because some of the cats would move more than others while you try to measure them, and most likely, you will get variations because not all 8 month old cats are identical. Even if you tried to minimize systematic effects (e.g. by shaving and sedating all the cats and working with a single breed), you will still get variations.

The goal of statistics would be to understand if your measurements allow you to infer something about a model for cat length, perhaps that 8 month old cats are on average L +/- dL long. Maybe you would find that the average cat length is different between countries, or different types of cat owners. As a scientist, you want to be assured that you can make rigorous statements about your findings and rigorously test different cat length models even when the data are expected to show a spread.

For the specific case of many measurements of a single value (e.g cat length), there are several common quantities that can be calculated to provide some description of the set (i.e. the distribution) of measurements that were obtained. We start by introducing those quantities (such as the mean and standard deviation), and then describe some useful ways to visualize the data such as histograms.

5.2 Basic statistical parameters

5.2.1 Mean and variance

The mean and variance are the most common quantities to characterize a distribution of measurements. If a set of N measurements, $\{x_1, x_2, \ldots, x_N\}$, of a single quantity were performed, the "sample mean" is defined as:

$$\bar{x} \equiv \frac{1}{N} \sum_{i=1}^{i=N} x_i \tag{5.1}$$

This is simply the arithmetic average. The "sample variance", σ_x^2, is defined as:

$$\text{var} \equiv \sigma_X^2 = \frac{1}{N-1} \sum_{i=1}^{i=N} (x_i - \bar{x})^2 \tag{5.2}$$

and the "sample standard deviation", σ_x, is defined as the square root of the sample variance:

$$\sigma_x \equiv \sqrt{\text{var}} \tag{5.3}$$

While the sample mean gives the average value of the data, the sample variance and standard deviation indicate the average distance between the data and the sample mean. That is, whether the measured values are all really close to the sample mean, or whether they are all spread out.

Note the $(N-1)$ in the definition of the sample variance, instead of N. This is a subtlety that arises from the fact that the formula for the sample variance uses the sample mean as determined from the data. If N is large, there will not be a big difference in the result from including the correct $N-1$.

This subtlety arises due to the fact that the "true" mean is not known, rather it has been estimated from the data. One way to think of this is that there is a large "population" of values, and we have only measured a small sample of them (for example, you would measure a small subset of cats and try to infer something about all cats). It is likely that, by chance, the sample mean that you measured is not exactly the population mean. We think of the sample mean as an "estimator", also referred to as a "statistic", whereas the true population mean can be referred to as a "parameter" of the population (which you are trying to estimate from data). The "population" may represent an actual group of which you have measured a subset, or it could be a theoretical model. That is, your sample mean could be an estimator for the mean of a model that you are using to describe the data.

Similarly, the sample variance is only an estimator for the true variance in the population. It has been shown (Bernoulli's correction), that dividing by N instead of $(N-1)$ would yield a sample variance that is systematically smaller than the true population variance (we would call this a "biased estimator"). Dividing by $(N-1)$ removes the bias.

Note that if the true mean of the population, μ, is known, then the sample variance as an estimator of the population variance is given by:

$$\text{var} \equiv \sigma_x^2 = \frac{1}{N} \sum_{i=1}^{i=N} (x_i - \mu)^2 \tag{5.4}$$

where we now divide by N since we did not use an estimate for the mean of the population.

5.2.2 Error on the mean

The variance and standard deviation are measurements of how spread out the measurements are relative to the mean. We can also determine how well the mean is determined from our sample by defining the "error on the mean", $\sigma_{\bar{x}}$, which is given by the standard deviation divided by the square root of the number of measurements:

$$\sigma_{\bar{x}} = \frac{\sigma_x}{\sqrt{N}} \tag{5.5}$$

The error on the mean related to our estimate of how well we have determined the mean. In particular, it shows that as we make more measurements, the error on the mean goes down (as the square root of the number of measurements); this at least make intuitive sense (why it is a square root is less intuitive).

5.2.3 Mode and Median

The mode and median are also or use to describe data. The mode is the number that occurs the most often (sometimes also called the "most probable value"). The median of the data set is the value for which there are an equal number of values that are bigger and smaller. For a set of measurements that are normally (gaussian) distributed, the mean, mode and median overlap.

EXAMPLE 5-1: Calculate the mean, variance, error on the mean, mode and median of the following measurements: {1.1, 1.2, 1.0, 1.3, 1.4, 1.5, 1.2, 1.3, 1.3, 1.1, 1.0, 1.6 }

We can solve this using a simple python program:

```
import numpy as np
from math import *

xi=np.array([1.1, 1.2, 1.0, 1.3, 1.4,1.5, 1.2, 1.3, 1.3, 1.1, 1.0, 1.6])
N=xi.size

#Calculate the mean:
xmean=0
for x in xi:
    xmean=xmean+x
xmean=xmean/N
print("mean: {:.2f}".format(xmean))
#This agrees with what numpy calculates:
print("mean from np: {:.2f}".format(xi.mean()))


#calculate the variance and standard deviation
var=0
for x in xi:
    var=var+(x-xmean)**2
var=var/(N-1)
std=sqrt(var)
print("variance: {:.3f}".format(var))
print("std. dev.: {:.3f}".format(std))
```

```
#This does not agree with what np calculates (because np uses N in the denominator
    by default)
print("variance from np: {:.3f}".format(xi.var()))
print("std. dev. from np: {:.3f}".format(xi.std()))
#To correctly use the numpy version, specify the " delta degrees of freedom"
print("variance from np (correct): {:.3f}".format(xi.var(ddof=1)))
print("std. dev. from np (correct): {:.3f}".format(xi.std(ddof=1)))

#The error on the mean is:
print("error on the mean: {:.3f}".format(std/sqrt(N)))

#Sort the array to make it easier to get the mode and median:
xi=np.sort(xi)
print(xi)
```

which gives the following output:

```
mean: 1.25
mean from np: 1.25
variance: 0.035
std. dev.: 0.188
variance from np: 0.033
std. dev. from np: 0.180
variance from np (correct): 0.035
std. dev. from np (correct): 0.188
error on the mean: 0.054
[ 1.   1.   1.1  1.1  1.2  1.2  1.3  1.3  1.3  1.4  1.5  1.6]
```

We have also shown how to obtain the values using the built-in numpy functions, rather than having to write the for loops. Note that by default, numpy uses N in the denominator instead of $N-1$ when calculating the sample variance. One needs to specify the "delta degrees of freedom" parameter, ***ddof****, which is the number to subtract from N (in our case it's 1).*

We could also write code to determine the median and mode, but by printing out the array with the values sorted, we can see that the number 1.3 appears the most often, so it is the mode. We can also see that half of the numbers are equal or smaller to 1.2 and that half of the numbers are equal or larger than 1.3; the median is this somewhere between 1.2 and 1.3; we can safely claim that it is 1.25.

5.2.4 Moments, Skewness and Kurtosis

The skewness and kurtosis are higher order "moments" of the data that describe how symmetric the data are with respect to the mode (skewness) and whether the data are grouped tightly at the mode (kurtosis).

A moment of the data is defined as:

$$m_\alpha = \frac{1}{N}\sum_{i=1}^{i=N}(x_i - \bar{x})^\alpha \tag{5.6}$$

The skewness is defined as:

$$\text{skew} = \frac{m_3}{m_2^{3/2}} \tag{5.7}$$

The skewness will be positive if a larger fraction of the measurements are bigger than the mode. It will be negative if most of the measurements are smaller that the mode. Effectively it tells you whether the mean is to the right or the left of the mode. If the skewness is 0, then mean and mode coincide.

The kurtosis is defined as:

$$\text{kurt} = \frac{m_4}{m_2^2} \tag{5.8}$$

If the data are normally distributed about the mean, then the kurtosis is equal to 3. If the data have a kurtosis higher than 3, then the distribution has a sharper peak than a normal distribution. The smallest kurtosis is 1, corresponding to data that are not concentrated about any specific value.

5.2.5 Covariance and correlation

Often, we may measure two different quantities simultaneously multiple times. For example, we may measure both the length and the weight of 8 month old cats, and we probably would expect that the two are "correlated"; that is, that if one quantity is bigger, then the other one tends to be bigger as well. We can quantify how correlated two quantities are by measuring the covariance and the correlation factors.

Let us suppose that we have two quantities, x and y, that we measure simultaneously N times, giving us two sets of measurements $x_i = \{x_1, x_2, \ldots, x_N\}$ and $y_i = \{y_1, y_2, \ldots, y_N\}$. The covariance, σ_{xy}, is defined to be:

$$\sigma_{xy} \equiv \frac{1}{N-1} \sum_{i=1}^{i=N} (x_i - \bar{x})(y_i - \bar{y}) \tag{5.9}$$

where $\bar{x}$ and $\bar{y}$ are the means of the measurements of x and y, respectively. The $N-1$ in the denominator has the same origin as when calculating the variance, and is required to recover the correct formula for the variance if $y = x$. The covariance is thus positive when x and y tend to both be larger or smaller than their mean at the same time, and negative when x and y tend to be in opposite directions from their means at the same time (e.g. x_i bigger than the mean when y_i is smaller than the mean). The covariance is near 0 when a value of x bigger or smaller than $\bar{x}$ does not lead to a value of y that is predictably bigger or smaller than $\bar{y}$. When the covariance is positive we say that x and y are correlated, whereas when it is negative, we say that they are "anti-correlated".

The correlation factor, ρ_{xy}, sometimes also called r or R or "the r-factor", is simply the covariance divided by the standard deviations of x and y, σ_x and σ_y, respectively. This leads to ρ_{xy} being bound between -1 and 1. A correlation factor of 1 means that x and y are perfectly correlated, a value of 0 indicates that they are un-correlated, and a value of -1 that they are perfectly uncorrelated. The correlation factor, ρ_{xy} is given by:

$$\rho_{xy} \equiv \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{5.10}$$

EXAMPLE 5-2: Calculate the covariance and correlation factors for the following 12 measurements of the quantities x and y: $x_i = \{1.1, 1.2, 1.0, 1.3, 1.4, 1.5, 1.2, 1.3, 1.3, 1.1, 1.0, 1.6\}$ and $y_i = \{1.3, 0.9, 0.9, 1.0, 1.6, 1.1, 1.2, 1.2, 1.2, 0.9, 0.9, 1.1\}$

The quantities can easily be evaluated in python using the following code:

```
import numpy as np
import pylab as pl
xi=np.array([1.1, 1.2, 1.0, 1.3, 1.4,1.5, 1.2, 1.3, 1.3, 1.1, 1.0, 1.6])
yi=np.array([1.3, 0.9, 0.9, 1.0, 1.6, 1.1, 1.2, 1.2, 1.2, 0.9, 0.9, 1.1])
xmean=xi.mean()
ymean=yi.mean()
xstd=xi.std(ddof=1)
ystd=yi.std(ddof=1)

covxy=((xi-xmean)*(yi-ymean)).sum()/(xi.size-1)
corxy=covxy/xstd/ystd

text='''
Covariance: {:.2f}
Correlation: {:.1f}'''.format(covxy,corxy)

pl.scatter(xi,yi)
pl.xlabel('x')
pl.ylabel('y')
pl.title("scatter plot of x and y")
pl.text(0.95,1.5,text,fontsize=14)
pl.show()
```

The above code determined the covariance to be 0.02 and the correlation to be 0.42. The values of x and y are thus correlated, and on average, a bigger value of x will result in a larger value of y. A scatter plot of x and y is show in Figure 5.1, highlighting the trend in the two correlated variables.

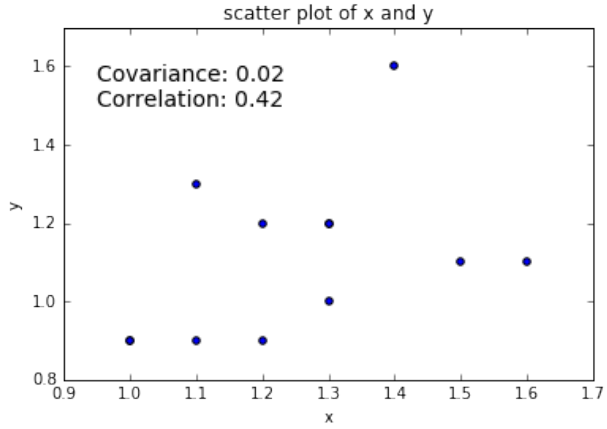

Figure 5.1: Scatter plot of x and y values for different measurements (example 5-2), showing the correlation between the two quantities.

It is important to note that the correlation is a measure of whether two quantities appear to be related in some manner. For example, a positive correlation between x and y indicates that if one measures a high value of x, then one is also likely to measure a high value of y. This does not

mean that a high value of x *causes* a high value of y. "Correlation does not imply causation" is a popular slogan in statistics (strongly emphasized in medical studies). Measuring the correlation should never be confused with trying to model one variable as a function of the other. For example, it is likely that the length of cats is correlated with their weights, but a simple confirmation of a strong correlation factor in no way implies that the length of cats causes them to be heavier. For example, it could be measured that going to sleep with one shoe on is correlated with having a headache in the morning. It would be wrong to conclude that sleeping with a shoe on causes headaches. Perhaps, there is a third variable that leads to both observations, for example, going to bed drunk.

5.3 The histogram

The histogram is a way to visualize a set of measurements of the same quantity. With repeated measurements, we generally obtain a "distribution" of values and the histogram is way to visualize that distribution. In example 5-1, we had the following measurements: x_i ={ 1., 1., 1.1, 1.1, 1.2, 1.2, 1.3, 1.3, 1.3, 1.4, 1.5, 1.6 } (which we sorted) for the value of some parameter, x. One way to visualize these data is to count how many times each value occurs and to make a table, as in Table 5.1 We can now plot this into what we call a "histogram", as in Figure 5.2.

value	frequency
1.0	2
1.1	2
1.2	2
1.3	3
1.4	1
1.5	1
1.6	1

Table 5.1: Frequency of occurrence of each measurement.

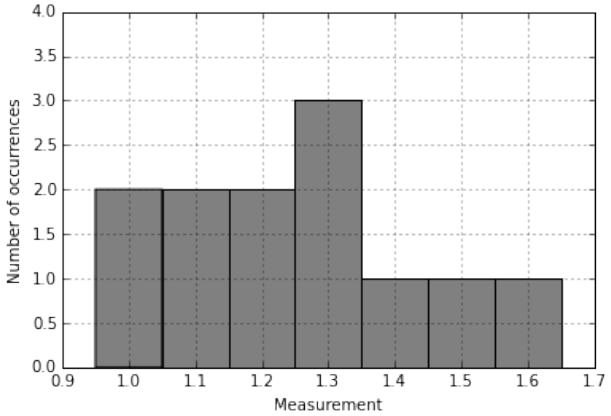

Figure 5.2: A simple histogram of the measurements from Example 5-1 and Table 5.1.

Histograms are drawn as "bar charts", since measurements at different values on the x-axis are unrelated (it would not be meaningful to treat the columns in Table 5.1 as x and y values for

data points and connect them with a line). We call each bar on the chart a "bin", since it "holds" the number of times each measurement occurred. The histogram is a good way to visualize a distribution of values; in our case, we can see that the value 1.3 comes up the most, and that the values below 1.3 come up more often than the ones above (the distribution is asymmetric about the mode). With only 12 measurements, it is not too difficult to sort the values and come to these conclusions without drawing a histogram; in the case where we have many measurements, histograms become an extremely useful visualization (and analysis) tool.

In the case that we just illustrated, we counted the occurrences of specific values, such as 1.3 or 1.2. If we had measured these numbers with additional precision (more decimals), it is likely that they would be each different. In this case, it would not make sense to count how many time each specific value occurs, since each value would occur only once. What we really want to do, is count how many times we get a value that is within a certain range. That is, we define the left and right edge for each bin, and then count how many times we get a value within that range. The histogram in Figure 5.2 was actually defined to have 7 bins: the first bin from 0.95 to 1.05, the second bin from 1.05 to 1.15, etc, in such a way that the numbers 1.0, 1.1, etc ended up at the center of each bin. In Figure 5.3, we have made a histogram of the same data, but using only 3 unequally-sized bins, from 0.9 to 1.1, from 1.1 to 1.4, and from 1.4 to 1.7, which results in an equally valid histogram of the data.

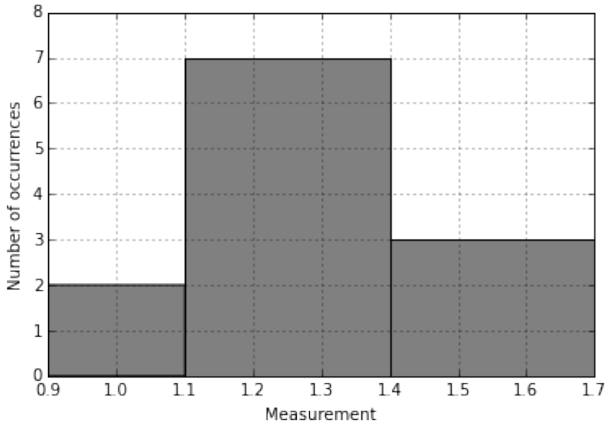

Figure 5.3: The same histogram as in Figure 5.2 but with different size bins.

As you can see, we have some liberty in how we define the "binning" of our histogram. If you make all of the bins really narrow, then they would typically have counts of either 0 or 1. If you make the bins way too wide, then all the measurements will end up into one bin, and the histogram will not help you visualize anything. In general, you will need to play with the binning for plotting data so that the histogram looks reasonable. A good rule of thumb is that the bins should have at least 5 counts in them. Ideally, you should also try to have equally sized bins (rather that varying bin widths as in Figure 5.3), as it makes it easier to analyse the histogram. Refer to section 2.6.2.3 for all of the commands to make histograms in python, as well as the code at the end of this section.

It is sometimes useful to "normalize" a histogram such that the area of the bins of the histogram sums to 1. If each bin, i, contains n_i counts (the bin height) and has a bin width, dx_i, then the

area under a histogram, A, is simply given by summing the area of each rectangular bin:

$$A = \sum_{i=1}^{i=N} n_i dx_i \tag{5.11}$$

In order to normalize a histogram, we must thus divide each bin content by A. If all of the bins in the histogram have equal width, dx, (say all $dx_i = dx$), then the area of the histogram is given by:

$$A = dx \sum_{i=1}^{i=N} n_i \tag{5.12}$$

where $\sum_{i=1}^{i=N} n_i$ is equal to the number of measurements (the sum of the contents of each bin). In Figure 5.2 from above, we had a fixed bin width of 0.1 and a total of 12 measurements; the area is thus given by $A = 1.2$. We can thus "normalize" that histogram by dividing the content of each bin by 1.2, as in Figure 5.4. In python, histograms can easily be normalized by passing the argument `normed=True` to the `hist()` function.

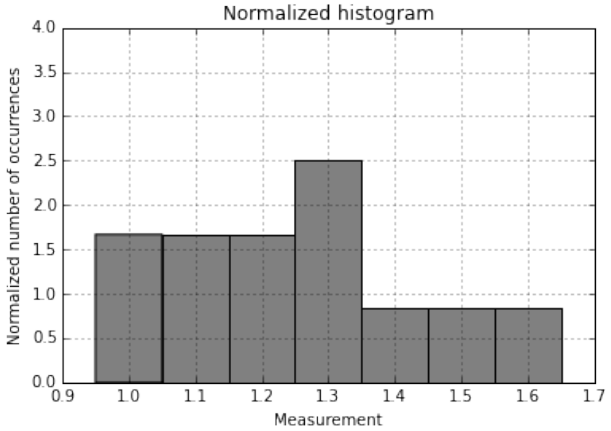

Figure 5.4: The same histogram as in Figure 5.2 but normalized to have an area of 1.

As we will see in the next chapters, a normalized histogram can be thought of as a "probability density function". If we assume that the data in the histogram are representative of the distribution of measurements that we would get if we repeated the measurements, then the normalize bin contents are representative of the probability of obtaining a certain measurement. For example, using Figure 5.4, we could quote that the probability of obtaining a measurement between 1.25 and 1.35 is given by (bin content)$\times$ (bin width) $= 2.5 \times 0.1 = 25\%$. In our original 12 measurements, we had obtained the number 1.3 three times, or 25% of the time.

The example python code below shows how to obtain the histograms shown in this section:

```
import numpy as np
import pylab as pl
#The data:
xi=np.array([1.1, 1.2, 1.0, 1.3, 1.4, 1.5, 1.2, 1.3, 1.3, 1.1, 1.0, 1.6])

#Make a histogram with equal bin width and plot it:
pl.hist(xi, bins=[0.95,1.05,1.15,1.25,1.35,1.45,1.55,1.65],color='gray')
```

```
pl.axis([0.9,1.7,0,4])
pl.grid()
pl.xlabel("Measurement")
pl.ylabel("Number of occurrences")
pl.show()

#Use 3 unequal bins
pl.hist(xi, bins=[0.9,1.1,1.4,1.7],color='gray',normed=False)
pl.axis([0.9,1.7,0,8])
pl.grid()
pl.xlabel("Measurement")
pl.ylabel("Number of occurrences")
pl.show()

#Use original binning, but normalize:
pl.hist(xi, bins=[0.95,1.05,1.15,1.25,1.35,1.45,1.55,1.65],color='gray', normed=
    True)
pl.axis([0.9,1.7,0,4])
pl.grid()
pl.xlabel("Measurement")
pl.ylabel("Number of occurrences")
pl.title("Normalized histogram")
pl.show()
```

5.4 Comparing statistical data to a model

We now have a few tools that can allow us to compare statistical data to a model (or between different data sets). For example, if one researcher has measured the lengths of 8 month old Canadian sphynx cats and another researcher has measured the lengths of 8 month old American sphynx cats, we can think about whether the Canadian and American cats are statistically similar. We could compare the mean lengths of the two samples using the corresponding errors on the mean to see if they agree. If the means agree within their respective errors on the mean, we could still conclude that Canadian and American cats are different if the shapes of the normalized histograms of the measurements were very different (e.g. you could get the same mean, but vastly different distributions).

5.5 Summary

In this chapter, we covered a few useful metrics to characterize a set of N measurements, $\{x_1, x_2, \ldots, x_N\}$, of a single quantity. In particular, we defined the mean:

$$\bar{x} \equiv \frac{1}{N} \sum_{i=1}^{i=N} x_i \tag{5.13}$$

The "sample variance":

$$\text{var} \equiv \sigma^2 = \frac{1}{N-1} \sum_{i=1}^{i=N} (x_i - \bar{x})^2 \tag{5.14}$$

and the "sample standard deviation":

$$\sigma \equiv \sqrt{\text{var}} \tag{5.15}$$

We also introduced the error on the mean:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}} \tag{5.16}$$

If we have a set of simultaneous measurement of two quantities, $x_i = \{x_1, x_2, \ldots, x_N\}$ and $y_i = \{y_1, y_2, \ldots, y_N\}$, the covariance, σ_{xy}, is defined to be:

$$\sigma_{xy} \equiv \frac{1}{N-1} \sum_{i=1}^{i=N} (x_i - \bar{x})(y_i - \bar{y}) \tag{5.17}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of the measurements of x and y, respectively. The correlation factor, ρ_{xy} is defined to be:

$$\rho_{xy} \equiv \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{5.18}$$

where σ_x and σ_y are the sample standard deviations of the measurements of x and y, respectively. The correlation is a measure of whether the quantities x and y tend to be simultaneously big or small when measured.

We finished by introducing the histogram as a way to visualize the distribution of measurements of a single quantity. We saw that we have liberty in how we define the location and sizes of the bins. We saw that by normalizing the area of the histogram, we could approximate a probability density function for the data.

6 Statistics - Distributions

In the previous chapter, we considered a sample of data that consisted of measurements of a certain quantity (e.g. cat length), and defined a few simple variables to describe that sample (e.g. mean, variance). Those were general quantities that can be applied to any sample of measurements.

In many physical situations, the data will follow particular "distributions". That is, if one makes a histogram of the measurements, the shape of that histogram will be well described by a function. Certain physical processes lead to very specific distribution functions, and we will examine a few of those in this chapter.

6.1 The binomial distribution

The binomial distribution is used to describe the result of an experiment that can go in one of two ways (for example, flipping a coin can be either heads or tails). Very generally, a lot of processes can be thought of as binomial. If you are interested in the number 5 coming up with rolling dice, even though there are 6 possible outcome, you could think of it as a binomial problem (you either get 5 or you don't).

The binomial distribution describes **how often we will get k successes in N independent trials, when we expect pN successes**. We call p the probability of getting the "successful" outcome, and reserve a more formal description of probability for a later chapter. For example, if we have a coin, we can think of k as the number of times we get heads in N coin tosses. We can think of p as the fraction of times we get heads (either that we expect or that we measured). If we have a biased coin that, on average, gives 70% heads, then $p = 0.7$ which can be measured by tossing the coin many times and measuring the fraction of time that we get heads (this would be the "frequentist" definition of probability, p).

EXAMPLE 6-1: For an unfair coin that has a $p = 0.7$ of being heads on any given toss, what is the probability of obtaining exactly 2 heads in 3 coin tosses?

If we label each toss by H (heads) or T (tails), then there are three possible sequences of 3 tosses that

give two heads: HHT, HTH, and THH. The probability of obtaining the first sequence (HHT) is given by:

$$P_{HHT} = pp(1-p)$$

p for the first toss, multiplied by p for the second toss and $(1-p)$ for the third toss. The other two sequences (HTH and THH) have the same probability. We only care about obtaining two heads in three tosses (not the order), so the probability of obtaining two heads in 3 tosses is simply the sum of the (identical) probability for each sequence:

$$\begin{aligned} P_{2H} &= P_{HHT} + P_{HTH} + P_{THH} = 3pp(1-p) \\ &= 3 \times 0.7^2 \times 0.3 = 0.441 \end{aligned}$$

Thus, there is a 44.1% chance that with this biased coin, we would obtain exactly 2 heads in 3 tosses.

A general formula for the probability of obtaining k successes in N trials, when the probability of a success is p, can easily be derived. If there are k successes, then there will be $N-k$ failures. One possible way to obtain this is to have the first k trials be successes, and the next $N-k$ trials be failures. The probability of that particular sequence would be:

$$P_{success\,first} = ppp\ldots(1-p)(1-p) = p^k(1-p)^{N-k}$$

However, we do not care about the particular order of the successes, so we must think about all of the possible sequences in which we can have k successes in N trials. This is found by the combinatorial formula for "N choose k":

$$\left(\begin{array}{c} N \\ k \end{array} \right) = \frac{N!}{k!(N-k)!}$$

and corresponds to the number of ways of choosing the k "positions" of the successes in a sequence of N trials. The total probability of obtaining exactly k successes in N trials is thus given by multiplying the probability of a single sequence (e.g. $P_{success\,first}$) by the number of possible sequences:

$$P(k, N, p) = \frac{N!}{k!(N-k)!} p^k (1-p)^{N-k} \tag{6.1}$$

which is called the **binomial probability (or distribution)** function.

Let us suppose that we run a food stand that sells three different kinds of poutines: a classic poutine, a poutine with pulled pork, and a vegetarian poutine (with vegetarian gravy). Our historical data show that 70% of customers purchase a classic poutine, 20% of customers purchase the pork poutine, and 10% purchase the vegetarian. It is one hour before closing time and we are running low on pork; we only have enough pork for 5 poutines and want to know if we should send someone to buy more. From our previous data, we also know that in this hour we should expect about 15 customers. So we can ask ourselves, what is the probability that we will get $k = 5$ customers out of $N = 15$ customers that order pork poutines in the next hour? Using the binomial distribution, we can easily estimate that quantity to be:

$$P(k = 5, N = 15, p = 0.2) = \frac{N!}{k!(N-k)!} p^k (1-p)^{N-k} = 0.103$$

There is thus a 10.3% chance that we will sell exactly 5 pork poutines to the next 15 customers.

However, this is not quite the information that we need in order to know if we should purchase more pork. We really want to know what the chances are of the next 15 customers purchasing 6 or more pork poutines, in which case we would not have enough pork. So we really need to add the probabilities for people to buy 6 or more pork poutines, to get the probability that we do not have enough pork:

$$P_{Not\ enough} = P(k = 6, N = 15, p = 0.2) + P(k = 7, N, p) + \cdots + P(k = 15, N, p)$$

which may be tedious to calculate by hand. An equivalent calculation would be to sum the probabilities of people buying 5 or less poutines, giving the probability that we will have enough pork:

$$P_{Have\ enough} = P(k = 0, N = 15, p = 0.2) + P(k = 1, N, p) + \cdots + P(k = 5, N, p)$$

where it should be clear that $P_{Not\ enough} + P_{Have\ enough} = 1$.

Fortunately, we do not have to calculate this by hand and can use a simple python program. The code below shows how to calculate the sum, but also illustrates that there are built-in functions to calculate these sums, as they come up very often. The `cdf()` functions sums the probabilities of getting k or less, and the `sf` function, which is 1-cdf, gives the probability of getting k or higher.

```
#The binomial distribution
import numpy as np
import pylab as pl
import scipy.stats as stats #for binomial distribution
%matplotlib inline

N=15 # number of customers (trials)
p=0.2 # probability of ordering a pork poutine

print("Prob of exactly 5 pork poutines: {:.3f}".format(stats.binom.pmf(5,N,p)))

#An array to hold all possible outcomes:
allOutcomes=np.arange(0,N+1)
#An array to hold the probabilities corresponding to each outcome:
probOutcomes=stats.binom.pmf(allOutcomes,N,p)

#Can sum the probabilities (or use sf and cdf):
print("Prob of 6 or more (sum): {:.3f}".format(probOutcomes[6:].sum()))
print("Prob of 6 or more (sf): {:.3f}".format(stats.binom.sf(5,N,p)))
print("Prob of 5 or less (sum): {:.3f}".format(probOutcomes[:6].sum()))
print("Prob of 5 or less (cdf): {:.3f}".format(stats.binom.cdf(5,N,p)))

#Make a plot of the probability of each outcome
pl.plot(allOutcomes,probOutcomes,'o',color='black')
pl.xlabel("Number of pork poutines ordered (k)")
pl.ylabel("Prob of ordering")
pl.title("Binomial prob. distribution for N=15, p=0.2")
pl.axis([-1,16,0,0.3])
pl.grid()
pl.show()
```

The output of the code is:

```
Prob of exactly 5 pork poutines: 0.103
Prob of 6 or more (sum): 0.061
```

```
Prob of 6 or more (sf): 0.061
Prob of 5 or less (sum): 0.939
Prob of 5 or less (cdf): 0.939
```

and we can see that there is 94% chance that we do not need to purchase more pork. In this case, we would probably take the 6% chance that we have to tell a customer that we have run out of pork.

We also used the code to create a graph of the probability of k outcomes as a function of k for a fixed value of $N = 15$ and $p = 0.2$, which is shown in Figure 6.1. We call this plot a "distribution" as it describes the distribution of outcomes that we expect. That is, given the values of N and p for our situation, we can use Figure 6.1 to model how many people purchase pork poutines in the hour preceding the closure of our poutine shop over many nights. For example, we would expect that 23% of the time, 2 pork poutines will be ordered in the last hour. The most likely number of pork poutines to be ordered in the last hour is 3, which is equal to pN (0.2×15).

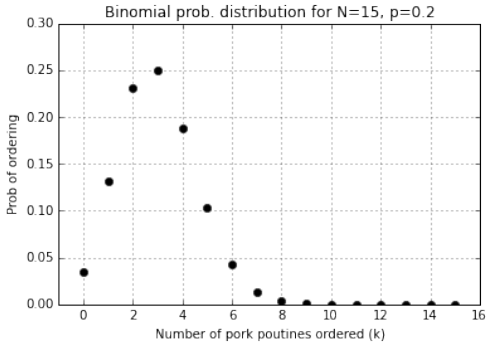

Figure 6.1: Binomial distribution for $N = 15$ and $p = 0.2$.

We can also turn the problem around and test the hypothesis that customers in our poutine store can be modelled by the binomial distribution. To do this, we could collect data to measure, over many nights, the number of customers that bought pork poutine in the hour preceding the store closing. In order to keep things simple, we should measure the number of pork poutines bought by the last 15 customers, rather than in the last hour (since we don't expect exactly 15 customers in the last hour). The data could look like that in table 6.1.

We can plot a normalized histogram of these numbers and overlay the binomial distribution that we obtained in Figure 6.1 to see if they agree. This is done in Figure 6.2 where Table 6.1 was generated with 1000 lines (simulating 100 nights of data taking).

We do not yet have the tools to quantitatively conclude whether the distribution of the data (the histogram) agrees with the predicted distribution from our binomial distribution model, but looking by eye, it does seem that the binomial distribution is a good model for the histogram of the data. As you can see, the histogram itself, when normalized, can be thought of as a probability distribution. With no model, we could generate the histogram using data, and then use that histogram to make statements about the probability of have between, say, 3 and 4 pork poutine orders.

Number of pork poutines ordered by last 15 customers
3
2
0
0
3
2
2
3
2
...

Table 6.1: Sample data on pork poutine orders by the last 15 customers in the store over many nights

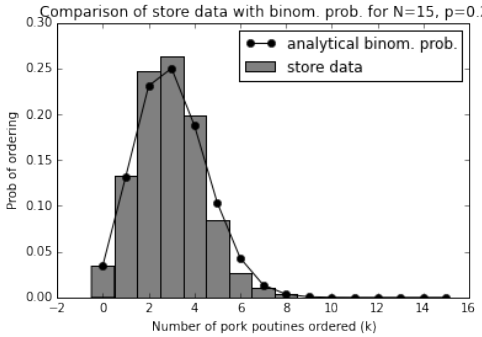

Figure 6.2: Binomial distribution for $N = 15$ and $p = 0.2$ overlaid with simulated data from 1000 measurements where a number of customers (Poisson distributed around 15) bought a number of pork poutines with a binomial distribution with $p = 0.2$.

6.1.1 Statistical properties of the binomial distribution

The mean of measurements that follow the binomial distribution, $\bar{k}$, for N trials with probability p of success is given by:

$$\bar{k} = \sum_{k=0}^{k=N} kP^{binom}(k, N, p) = Np \tag{6.2}$$

and their variance is given by:

$$\sigma^2 = \sum_{k=0}^{k=N} (k - \bar{k})^2 P^{binom}(k, N, p) = Np(1 - p) \tag{6.3}$$

Thus, a simple way to test if a set of measurements are binomially distributed is to calculate their mean and variance to see if they agree with that expected from the binomial distribution.

EXAMPLE 6-2: In a class of 20 students, we have counted over 15 different lectures how many of the students checked their smart phone instead of paying attention to class. Over the 15 different

lectures, the following number of students checked their phone: {8,10,7,6,5,9,5,4,4,5,6,4,5,7,6}. For example, in first lecture, 8 students checked their phone, in the second lecture, 10 students checked their phone, etc. Use some simple statistics to determine whether the number of students checking their phone during a given lecture is consistent with being binomially distributed.

We wish to check that the number of students check their phone follows a binomial distribution. In each lecture, we have $N = 20$ students and we thus wish to see if we can predict, on average, how many of those students will check their phone. We first need to determine the probability of checking their phone, p, which we must estimate from the data. We can estimate p by averaging the fraction of students that check their phone at each lecture. We can then compare the mean and variance of the data with what we expect for a binomial distribution. This is shown in the following python code:

```
import numpy as np

N=20
#put the data into an array
xi=np.array([8,10,7,6,5,9,5,4,4,5,6,4,5,7,6])
#get the average (mean) value of xi/N, which is our estimate of p
p=np.mean(xi/N)
print("p is {:.2f}".format(p))
#We can now compare the mean and variance of the data with what we expect
#for the binomial distribution:
print("data mean: {:.1f}, data variance: {:.1f}".format(xi.mean(),xi.var(ddof=1)))
print("binom. mean: {:.1f}, binom. variance: {:.1f}".format(N*p,N*p*(1-p)))
```

The output is:

```
p is 0.30
data mean: 6.1, data variance: 3.4
binom. mean: 6.1, binom. variance: 4.2
```

The mean from the data and that predicted for a binomial distribution agree exactly, and the variances are difficult to compare without uncertainties, although they appear to be consistent. We should not be surprised that the mean from the binomial prediction is exactly equal to the mean from the data, since we used the data to evaluate p in such a way that Np is precisely equal to the mean of the data!

In order to make a definite statement, we would need more data to see if the variance of the data converges to that predicted for a binomial distribution. If we had other competing models that also predicted a variance, we could see which model predicts the closest variance. Ultimately, it is very difficult to confirm if data come from a specific distribution if there are few data points (imagine how poorly defined the histogram would be).

6.2 The Poisson distribution

The Poisson distribution is used to predict **the probability of observing k successes when we expect n**. One can show that the Poisson distribution is the limiting case of the binomial distribution when N is very large and p is very small, but Np (the mean of the binomial distribution) is finite (see Figure 6.4). The n in the Poisson distribution is equal to Np from the binomial distribution, which is the mean of the binomial distribution. The Poisson distribution is given by:

$$P(k,n) = \frac{n^k e^{-n}}{k!}$$

and is much easier to calculate than the binomial distribution. The Poisson distribution is said to apply to "rare processes", as the probability, p, of a success occurring is small. Note that the

Poisson distribution only has one parameter, n, the expected number of successes.

One common example is to predict the expected number of radioactive decays in a period of time, t, for a given radioactive decay process with "time constant" λ. Generally, we would have a large number of atoms, N, and the probability of a particular atom to decay, p, is very small. However, the average number of decays, Np is finite. We could model this with a binomial distribution, but the Poisson distribution is in this case a very good approximation, and much easier to calculate.

Recall, the radioactive decay law tells us how many un-decayed atoms we should have at time t, $N(t)$, if we started with N_0 at time $t = 0$:

$$N(t) = N_0 e^{-\lambda t} \tag{6.4}$$

The rate of decays (how many decays we expect per unit time) is simply given by the time derivative (in this case we take the absolute value since we are interested in the rate of change of N and do not explicitly care that it is decreasing):

$$\left|\frac{dN}{dt}\right| = \lambda N_0 e^{-\lambda t} = \lambda N \tag{6.5}$$

In a given time interval, T, we thus expect an average of n decays given by:

$$n = \lambda N T \tag{6.6}$$

n is thus our expected number of decays in a period of time T for a radioactive decay constant λ given that we have N un-decayed atoms. We can now use the Poisson probability density function to evaluate the probability of getting k decays in that period of time.

EXAMPLE 6-3: A typical germanium detector is made of ultra pure germanium (100% germanium) and has a mass of 1 kg. We are interested in using a single germanium detector to observe a rare form of radioactive decay called "double-beta decay". The decay only occurs for the isotope ^{76}Ge which has a 7.4% natural abundance. The half-life of the decay is very long and has been determined to be 1.8×10^{21} years. Due to natural radioactivity in our laboratory, we have a background rate in our germanium detector of 10 counts per month, so we need to make sure that we expect a rate for the radioactive decay that is well above that in order to measure it. What is the probability of observing exactly 15 double-beta decays in our detector if we operate it for 1 month? What is the probability that we will see 15 or more events from double-beta decay?

First we must find out how many ^{76}Ge are present in our 1 kg detector. This is given by using the average mass of a germanium atom (72.63 amu) and the natural abundance.

$$N_{^{76}Ge} = 0.074\frac{1000\,g}{72.63\,amu \times 1.66\times10^{-24}\,g/amu} = 6.14\times10^{23}$$

We must then convert the half-life to a decay rate. The half-life is the time that it takes for half of the atoms to decay, so the radioactive decay law gives:

$$N(t) = N_0 e^{-\lambda t}$$
$$\therefore \frac{1}{2}N_0 = N_0 e^{-\lambda t_{1/2}}$$
$$\therefore \lambda = \frac{\ln(2)}{t_{1/2}} = 3.85\times10^{-22}\,yr^{-1}$$

The expected number of decays in a 1 month period ($T = 1/12$ years) is given by:

$$n = N_{^{76}Ge}\lambda T$$
$$n = 6.14 \times 10^{23} \times 3.85 \times 10^{-22}\, yr^{-1} \frac{1\, yr}{12}$$
$$n = 19.7$$

Evaluating the probability from the Poisson distribution with $n = 19.7$ and $k = 15$ gives 5.6%.

In order to get the probability of observing 15 or more events, we must sum the Poisson probabilities for $k = 15, 16, \ldots$ to infinity. The infinite sum can be avoided by instead summing the probabilities for $k = 0, 1, \ldots, 14$ and subtracting that from 1. The result is that the probability of getting 15 or more decays in 1 month is 82.7%.

We thus conclude that we have a good (83%) chance to detect double-beta decay given the requirement that we want to see at least 15 counts so that our signal is distinguishable from the background[1]*. The python code below shows how to calculate everything and uses the function* `sf` *to sum the probabilities of obtaining $k = 15$ counts or more:*

```
import numpy as np
import scipy.stats as stats # to use Poisson
from math import * # to use log

#Need the number of Ge atoms in 1 kg of germanium
mGe=72.63 #amu
amu=1.66054e-24 #amu to grams
abundance=0.074
mexp=1000 #total mass of Ge in grams
NGe=abundance*mexp/(amu*mGe)
print("There are {:.2e} 76Ge atoms".format(NGe))

#from half-life, need decay rate
thalf=1.8e21 #half-life in years
lam=log(2)/thalf #lambda, but cannot call it lambda as that is a python keyword
rate=NGe*lam
print("Lambda is {:.2e}".format(lam))
print("The decay rate is {:.1f} per year".format(rate))
print("The decay rate is {:.1f} per month".format(rate/12))

n=rate/12 #The expected number of decays in 1 month

print("The probability of observing 15 decays in one month is: {:.3f}".format(
    stats.poisson.pmf(15,n)))
print("The probability of observing 15 or more decays in one month is: {:.3f}".
    format(stats.poisson.sf(15,n)))
```

The output is:

```
There are 6.14e+23 76Ge atoms
Lambda is 3.85e-22
The decay rate is 236.3 per year
The decay rate is 19.7 per month
The probability of observing 15 decays in one month is: 0.056
The probability of observing 15 or more decays in one month is: 0.827
```

[1] In reality, achieving a background rate of 10 counts per month in a germanium detector is almost impossible, and it would not be nearly as easy to observe double-beta decay.

Figure 6.3 shows the Poisson distribution as a function of k plotted for different values of the expected number of successes, n. For low values of n, the distribution is asymmetric, but as n becomes bigger, the distribution becomes more symmetric and eventually will approach a bell curve. Figure 6.4 shows a comparison of the Poisson distribution for $n = 3$ along with the binomial distribution with different values of N and p such that $Np = 3$. As p gets smaller, the binomial distribution is indeed seen to approach the Poisson distribution.

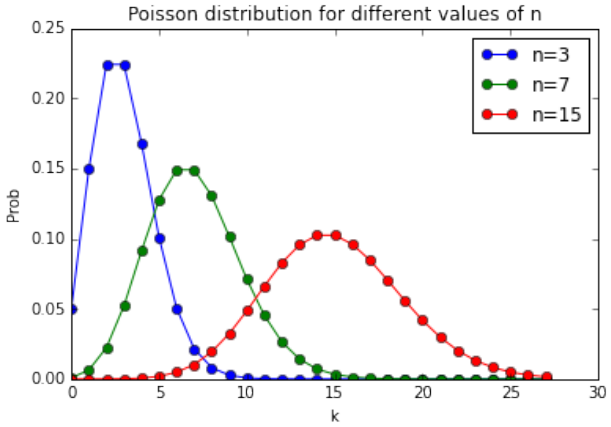

Figure 6.3: Poisson distribution for different values of the expected number of successes.

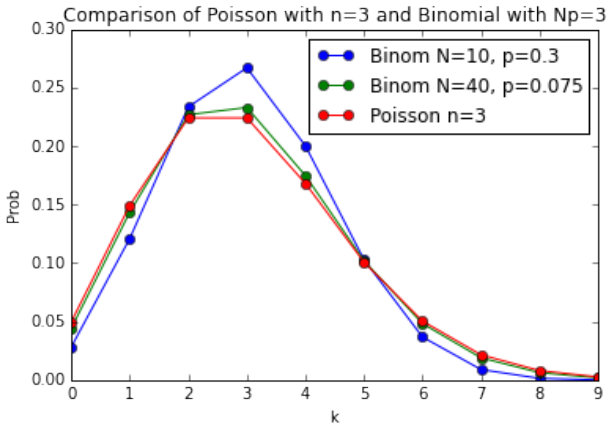

Figure 6.4: Comparison of the Poisson distribution with $n = 3$ and the binomial distribution for $Np = 3$, using different combination of N and p. As p gets smaller, the binomial distribution converges to the Poisson distribution.

The Poisson distribution is extremely useful and appears every time that one carries out experiments that involve counting. This is because one can phrase any experiment where something is to be counted into a problem of asking: "what is the probability that I count k occurrences when I expect n?". For example, what is the chance that 14 babies will be born at the hospital tonight, when on average 10 are born per night? Or, what is the probability that we will have less than 3 accidents this week in Kingston when we usually have 6 accidents per week? Again, all of these scenarios could be modelled by the binomial distribution, but they correspond to cases where p is very small and where the Poisson distribution can be used (the probability of specific mother

giving birth on a given night is relatively small, as is the probability of a specific car getting in an accident). It can sometimes be difficult to reconcile a counting experiment with the concept of a probability in a binomial process being small. Usually, the way to think about it is that there is a small probability for a specific item out of many to be the one that is counted, but a finite number of items will be counted.

When we count something that has intrinsic randomness (as the examples above), the Poisson distribution can help us to understand the uncertainty on a measured number of counts. Since we understand that when we count something, the result we get follows the Poisson distribution, we can very precisely quantify the probability of getting a number that is bigger or smaller than what we obtained. In fact, we can use the Poisson distribution to define a range in which we are, say, 90% confident that the true value lies.

EXAMPLE 6-4: Suppose we spent one week in an all-inclusive vacation in the Caribbean, lounging on the beach and watching tourists get hit on the head by falling coconuts. In one week, we counted that 10 tourists were hit on the head by coconuts. Upon returning to our university, we want to tell our physics professor about our measurement and wish to report with a high degree of confidence (90%) how many tourists get hit by coconuts every week. What range should we quote?

Note that this is a Poisson process, and the probability of a specific coconut hitting a tourist is small, but the number of coconuts that will hit tourists is finite.

We want to find a range over which the number of coconut hits per week covers 90% of the probability. For example, we would like to be able to say that we are 90% confident that every week between 8 and 12 tourists get hit.

We only have 1 data point, which is the 10 hits that we counted in the week that we were there. Our "best estimate" for the expected number of hits each week is thus $n = 10$*. We can tabulate the Poisson probabilities for* $n = 10$*:*

number of hits	Poisson prob
5	0.038
6	0.063
7	0.090
8	0.113
9	0.125
10	0.125
11	0.114
12	0.095
13	0.073
14	0.052
15	0.035
16	0.022

Table 6.2: Poisson probabilities of the number of tourists getting hit by coconuts in a week for an expected number of 10.

We can use different ways to choose the range; one way is to symmetrically expand the range on either side of 10 and add the probabilities until we reach 90%. The following python code does this:

```
import numpy as np
import scipy.stats as stats # to use Poisson
n=10 # the expected number of coconut hits

#Sum the probabilities on either side of n until we reach 90%

prob=stats.poisson.pmf(n,n)#the probability of getting n
i=0
while(prob<0.9):
    i=i+1
    #add in the probability at n-i and n+i
    prob=prob+stats.poisson.pmf(n-i,n)+stats.poisson.pmf(n+i,n)

print("Between {} and {}, the probability is {:.1f}%".format(n-i,n+i,100*prob))
```

The output is:

```
Between 5 and 15, the probability is 92.2%
```

Thus, if we observed 10 coconut hits in one week, we can claim with 92.2% confidence that every week, between 5 and 15 people get hit by coconuts!

We can also easily modify this code to get the common 68% confidence level (replace 0.9 on line 9 with 0.68), in which case we find that with 73.4% confidence, between 7 and 13 people get hit on the head each week. We could quote that, each week 10 ± 3 people get hit with coconuts.

If you recall, we had prescribed in Section 3.2.5 that the uncertainty on a counting measurement should be the square root of the number of counts, and that this would give a 68% chance that the true number is in the quoted range. We have just verified numerically that this is true, as our square root uncertainty would be $\sqrt{10} = 3.2$, close to what we have found.

6.2.1 Statistical properties of the Poisson distribution

The mean of measurements that follow the Poisson distribution for an expected number of successes, n, is simply n:

$$\bar{k} = \sum_{k=0}^{k=\infty} kP^{Poisson}(k,n) = n \tag{6.7}$$

and their variance is given by:

$$\sigma^2 = \sum_{k=0}^{k=\infty} (k-\bar{k})^2 P^{Poisson}(k,n) = n \tag{6.8}$$

The standard deviation (the square root of the variance) is thus given by $\sqrt{n}$. This is the origin of the prescription for the square root uncertainty presented in Section 3.2.5, which as we saw approaches the value of a 68% confidence when n becomes large (for $n = 10$, we saw in example 6-4 that the confidence level was approximately 73.4%).

We can go back to Example 6-2, where we had counted the number of students that had checked their phone during the lectures. From the data, we found a mean of 6.1 students checking their

phone with a variance of 3.4. If the students checking their phones are binomially distributed, then we expected the variance to be 4.2. Now that we know the variance for a Poisson process, we can see if the students are closer to that distribution. The expected number, n, for the Poisson process is also 6.1 (it is equal to the mean of the binomial). The expected variance, if the process of student checking their phones is Poisson, is thus also 6.1. We can see that, in this case, the data (based on their variance) are more consistent with the binomial distribution than the Poisson distribution.

One other interesting property of the Poisson distribution is that if you measure the time interval between two events, the distribution of those times will follow a decaying exponential. That is, you will more often have two Poisson events that follow each other close in time, rather than having a long time between events. For example, if you look at radioactive decays and measure the time between two decays, you will find that two decays happen in rapid succession more often than with a long time interval. A more tangible example is that airplane crashes seem to appear in bunches rather than spread out regularly over time. This is because airplane crashes are rare (Poisson) events, so it is in fact more likely to have them occur closely in time rather than regularly spread out over the year. A mis-understanding of Poisson statistics also leads to people predicting impeding doomsdays when "terrible" events happen in close succession.

6.3 Statistical hypothesis testing

We can now apply our understanding of distributions to statistical hypothesis testing. The general idea is that given the result of a measurement, we want to be able to make a quantitative statement about how likely the result is based on a hypothesis. If the result is very unlikely according to the hypothesis, then we would conclude that the hypothesis is not supported by the data.

For example, a pharmaceutical company may claim that they have developed a new pill that can help people to perform very well on IQ tests . As sceptical scientists, we decide to test whether the pill is actually effective and devise the following test: we choose 16 people, give them a first IQ test, then give them the pill, and finally give them a second, different but equivalent, IQ test. Suppose that we find that 9 out of the 16 people have performed "significantly" better on the second IQ test; can we conclude that the pill works?

We need to state this problem into a hypothesis that predicts the distribution of outcomes, and then evaluate the probability of our outcome (9 out 16 people performing better). If the probability is high, then the hypothesis is supported (note that it is never proven). When possible, one should try to make a "null hypothesis", which is usually easier to model. Typically, the null hypothesis is that "there is no effect".

In our case, our null hypothesis is that the pill has no effect, or rather that the results are from pure chance. We must thus ask ourselves what is the probability that, out of pure chance, 9 out 16 participants would perform better on a second IQ test? If we imagine that the IQ tests have many questions, it is unlikely that someone would get exactly the same score twice in a row. It seems reasonable that it is equally likely that they will perform a little better or a little worse on the second test. There is thus a 50% chance that they will perform a little bit better. We can model our experiment as a binomial problem: we have 16 people (N trials), and we had 9 people do better on the second test (k successes), and under our null hypothesis there is a 50% chance of

performing better on the second test (p). The binomial probability, $P^{binom}(k = 9, N = 16, p = 0.5)$ is easily found to be 17%. There is thus a 17% chance that we would obtain our result even in the pill did not work (or an 83% probability that our null hypothesis is wrong and that the result are not from pure chance).

Is 17% a high enough probability for us to support the null hypothesis that the results are from pure chance? Of course, that is an arbitrary question. One way to think about whether a probability is "high" is to ask yourself if you would risk your life at that probability. Would you get in a car if you knew you had a 17% chance of getting into a fatal accident? Probably not (I hope). At 17%, we would generally accept the null hypothesis as plausible, and conclude that the results are consistent with pure chance. In science, a common "bar" is that the null hypothesis must be no more than 5% plausible for it to be rejected, although depending on how "revolutionary" the result, that bar can be much higher. For example, for discovering a new particle in physics, the null hypothesis (there is no new particle) must usually be no more than 0.00006% likely.

Why would we not think about this the other way around, namely that the hypothesis that the pill works is confirmed with 83% probability, well exceeding our 5% bar? Remember, our model is that there is a 50% chance of doing better on an IQ test simply due to chance. Our model is not "given this pill, there is a x% chance of doing better on the test". The 83% probability cannot be interpreted as the probability that the pill works, which would be a completely different hypothesis. The "opposite" of "the results are from pure chance" is *not* that "the pill works". Perhaps there are other effects that lead to our result that are not the pill itself (maybe the subjects took the pill with some water, were then better hydrated, and performed better, or it was a placebo effect).

We did not quite analyse our data correctly, since we earlier claimed that 9 of the test subjects performed *significantly* better, not just marginally better. The definition of "significantly" would presumably come from the designers of the IQ test. IQ tests are statistical tests in themselves, so let us assume that if someone gets a score of X, then the test is designed such that one can attribute an uncertainty to the score, δX, such that over repeated tests, there is a 68% chance that a given participant scores in the range $X \pm \delta X$. When we claimed that 9 of the test subjects performed significantly better, we really meant that they performed with a score higher than their original score plus the uncertainty.

To take this into account, we must thus use a different probability than $p = 0.5$ in our null hypothesis that subjects would perform significantly better just out of pure chance. In this case, out of pure chance, there is a 32% chance of a participant scoring outside of the range $X \pm \delta X$, and a 50% probability that it is on the higher side of the range. The correct binomial probability to use is thus $p = 0.5 \times 0.32 = 0.16$, and it represents the probability of scoring significantly higher out of pure chance. Evaluating $P^{binom}(k = 9, N = 16, p = 0.16)$ gives 0.02%, which is certainly low. It is thus unlikely that the null hypothesis is correct, in other words, it is unlikely that by pure chance 9 out of 16 participants would score significantly higher on the IQ test. We would then lean to the conclusion that the pill likely has some effect (but as good scientists, we would want to make sure that any other possible effect have been controlled in the test; for example, that there is no placebo effect from the pill, that the water taken with the pill had no effect, etc.).

EXAMPLE 6-5: The 2013 Ontario Road Safety Annual Report shows the following number of fatal car collisions by month: {35,17,26,24,39,27,61,44,37,46,42,31}, which add up to 429 for the year.

Upon looking at the data, a politician decides that their next campaign will be based on banning driving in July since that is a particularly deadly month for driving in Ontario (61 deaths). Can you support the politician's claim that July is a bad month for driving?

We must first develop a null hypothesis, namely that no month is particularly worse for driving. If the data are consistent with the null hypothesis, then there is little reason to support the politician's claim. We thus need to evaluate the probability of having 61 fatal accidents in a particular month under the hypothesis that all months are equal.

In our null hypothesis, we expect the rate of accidents each month to be approximately equal to the average rate. Since we had 429 accidents in the year, we would expect 429/12=36 accidents per month as the average rate. Given that we expect 36 accidents in a month under the null hypothesis, what is the probability of having 61 accidents? This is found using the Poisson distribution. Evaluating $P^{Poisson}(k = 61, n = 36)$*, we find a probability of 0.004%, which is certainly small. The null hypothesis is not supported by the data, so we conclude that it is highly likely that some months are more fatal than others on Ontario roads, based on the 2013 data.*

As good scientists, we would still want to look at the data from other years to see if they also do not support the null hypothesis. In 2012, there were a total of 505 accidents in the year, and 50 accidents in July. Going through the same exercise, gives a probability for the null hypothesis of $P^{Poisson}(k = 50, n = 505/12)$*=2.7%, which is orders of magnitude higher than 2013, although still barely lower than our 5% bar (if that is the bar that we choose). As you can see, you need to be very careful when making claims based on statistical data.*

6.4 Summary

In this chapter, we introduced the concept of a statistical distribution to model the spread of results that arises from processes that have some randomness associated with them. We introduced the binomial distribution to model processes where we expect k successes after N independent trials, when a given success has a probability of p. The binomial distribution gives the probability of having exactly k successes:

$$P^{binom}(k, N, p) = \frac{N!}{k!(N-k)!} p^k (1-p)^{N-k} \tag{6.9}$$

The mean and variance of a quantity that is binomially distributed are given by:

$$\bar{k} = \sum_{k=0}^{k=N} k P^{binom}(k, N, p) = Np \tag{6.10}$$

$$\sigma^2 = \sum_{k=0}^{k=N} (k - \bar{k}) P^{binom}(k, N, p) = Np(1-p) \tag{6.11}$$

We then introduced the Poisson distribution to model the probability of obtaining k successes when we expect n. We saw that the Poisson distribution is the limiting case of the binomial distribution when $n = Np$ and p is small (thus the probability of a success is low). The Poisson distribution is given by:

$$P^{Poisson}(k, n) = \frac{n^k e^{-n}}{k!}$$

The mean and variance of a quantity that is Poisson distributed are given by:

$$\bar{k} = \sum_{k=0}^{k=\infty} kP^{Poisson}(k,n) = n \tag{6.12}$$

$$\sigma^2 = \sum_{k=0}^{k=\infty} (k-\bar{k})P^{Poisson}(k,n) = n \tag{6.13}$$

The variance of the Poisson distribution is equal to its mean, n. The standard deviation is thus $\sqrt{n}$, which is the origin of the square root uncertainty on a number of counts.

Finally, we introduced the concept of statistical hypothesis testing, which involved determining the probability of a particular outcome given a particular hypothesis. We argued that it is usually easier to model the null hypothesis, that is, to determine the probability of the results occurring out of pure chance. We cautioned against drawing conclusions simply based on the probability of the null hypothesis being below a given bar (5% is often used).

7 Statistics - The Normal Distribution

So far, we have examined two types of statistical distributions, the binomial and Poisson distributions. We reserved this chapter to cover one of the most important distributions in physics and science: the normal distribution (or gaussian distribution). The assumption that data follow a normal distribution is what leads to a lot of the rules for error analysis that we introduced in earlier chapters (e.g. adding in quadrature, 68% confidence levels, etc), and this distribution is thus fundamental in understanding many of the topics in error analysis.

The binomial and Poisson distributions are functions of a discrete variable (we used k in the previous chapter as the discrete variable). For example, it only made sense to ask what is the probability of obtaining k heads in N coin tosses if k is an integer. $P^{binom}(k, N, p)$ thus truly represents the probability of obtaining k heads in N tosses. If we sum the probabilities for all of the possible values of k, we get 1, as expected (the probability of having some sort of outcome has to be 1).

The normal distribution is a function of a continuous variable, and we usually use x. The normal distribution is given by:

$$P^{norm}(x, \mu_x, \sigma_x) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \tag{7.1}$$

and has two parameters: the mean, μ_x, and the standard deviation, σ_x. We cannot interpret the normal distribution to give us the probability of obtaining the value x in a given measurement, because x is a continuous variable. We can never measure anything to infinite accuracy, so the probability of getting *exactly* x is always zero. This was not the case for the discrete binomial and Poisson distributions, where one can expect to get exactly a given number of successes. Rather, the normal distribution gives us the probability, $P^{norm}(x, \mu_x, \sigma_x)dx$, of obtaining a value that is between x and $x + dx$, where dx is a small increment in x (small in the sense that $P^{norm}(x + dx, \mu_x, \sigma_x)$ is equal to $P^{norm}(x, \mu_x, \sigma_x)$). The probability, $P^{norm}(x, \mu, \sigma)dx$, is the area underneath the curve $P^{norm}(x, \mu_x, \sigma_x)$ between x and $x + dx$.

Figure 7.1 shows the normal distribution as a function of x for different values of the parameters μ and σ. The normal distribution is symmetric about the mean μ and has a width that is proportional to σ. It should be obvious from Equation 7.1 that the distribution is symmetric about μ. The

factor $\frac{1}{\sigma\sqrt{2\pi}}$ in front of the exponential normalizes the distribution so that the total area under the curve is equal to 1. Just as in the case of the binomial and Poisson distributions, we need the sum of the probabilities of all possible outcomes to be equal to 1. For the normal distribution, we thus require:

$$\sum_{x=-\infty}^{x=+\infty} P^{norm}(x, \mu_x, \sigma_x)dx = 1$$

which of course should be written as an integral since x is a continuous variable:

$$\int_{-\infty}^{+\infty} P^{norm}(x, \mu_x, \sigma_x)dx = 1 \tag{7.2}$$

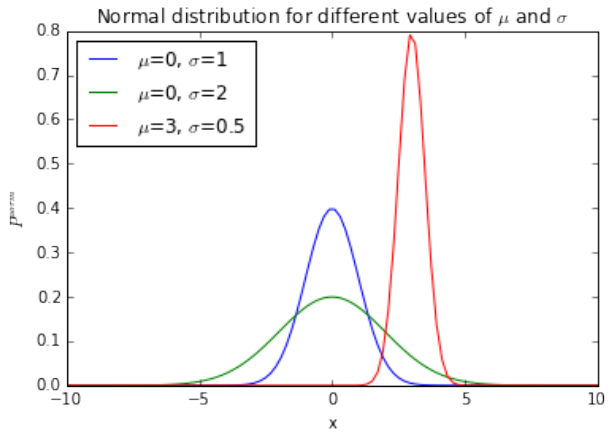

Figure 7.1: Normal distribution for different parameters.

The main reason that a the normal distribution is so important is that both the binomial and Poisson distributions approach the normal distributions when their mean is large. Figure 7.2 shows that as the mean of the binomial distribution increases, the distribution approaches the normal distribution with $\mu_x = Np$ and $\sigma_x = \sqrt{Np(1-p)}$ (given by the mean and standard deviation of the binomial distribution).

Figure 7.3 shows that as the mean of the Poisson distribution, n, increases, the distribution approaches the normal distribution with $\mu_x = n$ and $\sigma_x = \sqrt{n}$. We chose to illustrate the binomial and Poisson distributions with the same mean (6 and 22.5) but different variances.

The normal distribution is much more straightforward to evaluate than the binomial or Poisson distribution, because it a function of a continuous variable, and it does not require evaluating the factorial of a number (calculators struggle with factorials of numbers as small as 100). As we will see, many physical situations are in a regime (e.g Poisson with a large mean) where they can be well approximated by the gaussian distribution. Grades in a class, heights of people, lengths of 8 month old cats, weights of brand new pencils, the rapid fluctuations in stock prices, time spent by patients in the ER, etc. are all normally distributed.

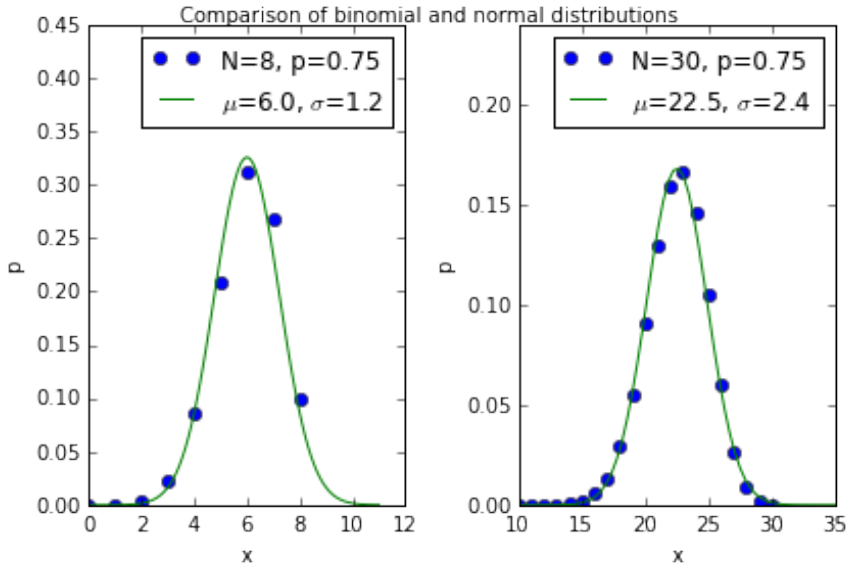

Figure 7.2: Comparison of binomial and normal distributions.

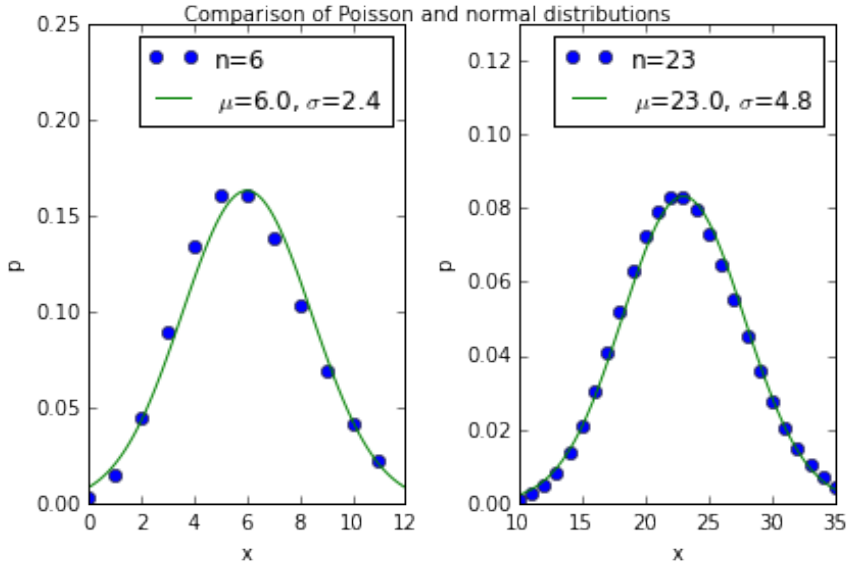

Figure 7.3: Comparison of Poisson and normal distributions.

EXAMPLE 7-1: Compare the probability of obtaining 140 heads in 300 coin tosses (of a fair coin) evaluated using the binomial and normal distributions.

We need to evaluate the binomial probability given by $P^{binom}(k = 140, N = 300, p = 0.5)$*:*

$$P^{binom}(k = 140, N = 300, p = 0.5) = \frac{N!}{k!(N-k)!} p^k (1-p)^{N-k} = 0.046$$

If you try to evaluate this with your calculator, it might crash when trying to evaluate 300!*, so this is best evaluated using a computer (see below).*

The parameters for the normal distribution in terms of the parameters from the binomial distribution are given by $\mu_x = Np = 150$ *and* $\sigma_x = \sqrt{Np(1-p)} = 8.7$*. We can evaluate the probability for the normal distribution,* $P^{norm}(x, \mu_x, \sigma_x)dx$ *using* $x = 150$ *and* $dx = 1$*, which we interpret as the probability of obtaining* x *between 150 and 151, since the probability of obtaining an exact value of* x *is zero. Of course, with* $dx = 1$*, we only really evaluate* $P^{norm}(x, \mu_x, \sigma_x)$ *at the given value of* x*, but we should always remember that we have implicitly chosen* $dx = 1$*, and our result is the probability of having* x *between* x *and* $x + dx$*.*

The simple python program below evaluates the two probabilities:

```
import scipy.stats as stats
from math import sqrt
print("The binomial probability of k=140, N=300, p=0.5 is: {:.5f}".format(stats.
    binom.pmf(140,300,0.5)))
mu=300*0.5
sigma=sqrt(300*0.5*(1-0.5))
print("The gaussian probability of x=140 for mu={} and sigma={:.2f} is: {:.5f}".
    format(mu, sigma, stats.norm.pdf(140,mu, sigma)))
```

with output:

```
The binomial probability of k=140, N=300, p=0.5 is: 0.02367
The gaussian probability of x=140 for mu=150.0 and sigma=8.66 is: 0.02365
```

and the probability from the normal distribution is very close to that from the binomial distribution.

7.1 Properties of the normal distribution

The normal distribution has several properties that make it straightforward to use and understand.

7.1.1 Statistical properties

The mean of measurements that follow the normal distribution with mean μ and standard deviation σ is given by

$$\bar{x} = \int_{-\infty}^{+\infty} x P^{norm}(x, \mu_x, \sigma_x)dx = \mu \tag{7.3}$$

and their variance is given by:

$$\sigma_x^2 = \int_{-\infty}^{+\infty} (x - \bar{x})^2 P^{norm}(x, \mu_x, \sigma_x)dx = \sigma^2 \tag{7.4}$$

That is, the mean and standard deviation of measurements that are normally distributed are precisely the mean and standard deviation of the normal distribution.

7.1.2 Area underneath the normal distribution

As we stated earlier, the total area underneath the normal distribution must equal 1, as it represents the probability of measuring x to be *something*. The area under x_a and x_b represents the probability that x is between x_a and x_b:

$$P(x_a \leq x \leq x_b) = \int_{x_a}^{x_b} P^{norm}(x, \mu_x, \sigma_x)dx \tag{7.5}$$

EXAMPLE 7-2: On average, 25 students skip any given Error Analysis class. Use the Poisson and normal distributions to evaluate the probability that 30 students will skip class on a given day, and compare the results.

This problem should be well modelled by the Poisson distribution, $P^{Poisson}(k, n)$, with the expected number of students skipping class, $n = 25$. Evaluating the probability of having $k = 30$ students gives:

$$P^{Poisson}(k = 30, n = 25) = \frac{n^k e^{-n}}{k!} = 0.0454$$

The corresponding normal distribution is given using $\mu_x = n = 25$ and $\sigma_x = \sqrt{n} = 5$. Recall that the normal distribution does not give the probability of obtaining an exact value, but rather must be used to evaluate the probability of obtaining a value within a certain range. We can choose to evaluate $P^{norm}(x, \mu_x, \sigma_x)dx$ with $x = 30$ and $dx = 1$, so that we can interpret the result as giving the probability of obtaining x between 30 and 31, which is almost equivalent (in interpretation) to the result we get from the Poisson distribution. For this case, with $dx = 1$, we get:

$$P^{norm}(x = 30, \mu_x = 25, \sigma_x = 5)dx = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} = 0.0483$$

If we do the proper integral, we find:

$$\int_{x=30}^{x=31} P^{norm}(x, \mu_x = 25, \sigma_x = 25)dx = 0.0436$$

The probability from the normal distribution when evaluated correctly (with the integral) is within 1 % of the Poisson probability, and within 3% when evaluated with $dx = 1$. As the mean of the distribution becomes bigger, the approximation between the normal and Poisson distribution will improve.

As we will see, it is often useful to compute the probability of obtaining a value within a certain number of standard deviations from the mean of the distribution. For example, the probability of obtaining x between $\mu_x - \sigma_x$ and $\mu_x + \sigma_x$ is given by:

$$P(\mu_x - \sigma_x \leq x \leq \mu_x + \sigma_x) = \int_{\mu_x - \sigma_x}^{\mu_x + \sigma_x} P^{norm}(x, \mu_x, \sigma_x)dx = 0.68 \tag{7.6}$$

which is a fixed number, regardless of the actual values of μ_x and σ_x. This is the origin of our seemingly arbitrary use of 68% as our usual confidence interval. Since many measurements are normally distributed (as we will see in the next section), the use of the standard deviation as the

n, number of σ_x	**Probability of being in range $\mu_x \pm n\sigma_x$**
1	0.682689
2	0.954500
3	0.997300
4	0.999937
5	0.999999

Table 7.1: Probability that a normally distributed measurement will fall within a certain number, n, of standard deviations from the mean.

error (or quoted uncertainty) corresponds to a 68% chance of a value being within one standard deviation of the mean. Table 7.1 shows the probability for a measurement to lie within a certain number of standard deviations from the mean.

As you can see, the probability of being further than $3\sigma_x$ is already quite small, and quickly becomes vanishingly small (as can also be seen in Figure 7.1). In particle physics, it is often required that a measurement have a validity of at least $5\sigma_x$ to be considered a discovery. For example, this would mean that in order to claim that a new particle has been discovered, the probability of the data given the null hypothesis must be at least $5\sigma_x$ from the mean expected if there were no new particle. A new particle may be discovered as a resonant peak in a spectrum; we would thus require that the peak indicating the presence of a particle be at least 5 times larger than the random fluctuations in the spectrum (under the assumption that those fluctuations are normally distributed).

Figure 7.4 shows the normal distribution on the left, calculated with μ_x=20 and σ_x=5. Since this is not the probability of getting a given value of x (you need to take the integral to get a probability), we call this the "probability density function" (or pdf). On the right are the "cumulative probability density function" (cdf) and the "survival fraction" (sf), which is simply 1 minus the cdf. The cdf is the integral of the pdf. Thus, the cdf is a probability. The cdf evaluated at a certain value of x is the integral of the pdf from negative infinity to x, which corresponds to the probability of obtaining a measurement x or smaller. The survival fraction is thus the probability of getting x or bigger. There is no closed form for the cdf or sf corresponding to the normal distribution. Since the normal distribution is symmetric about the mean μ, the cdf evaluated a $x = \mu_x$ is exactly 0.5 (there is a 50% chance of measuring x smaller than the mean).

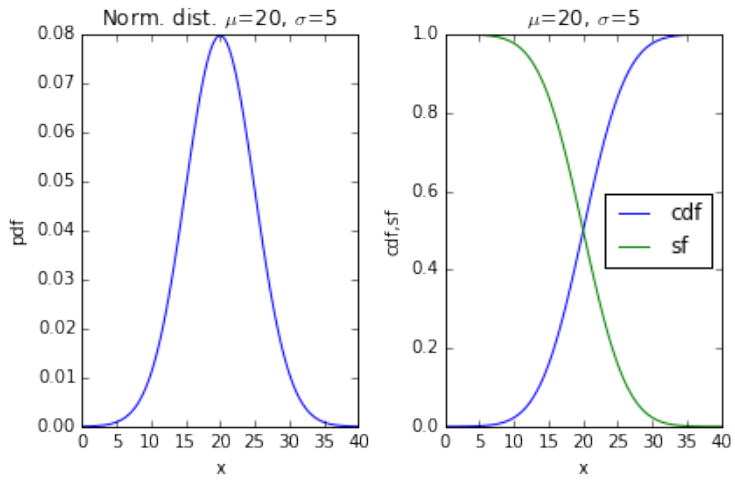

Figure 7.4: Normal probability density function (pdf), cumulative probability density function (cdf) and survival fraction (sf).

EXAMPLE 7-3: An IQ test is designed such that the test results, on a random sample of the population, are normally distributed with a mean of 100 and a standard deviation of 10. What is the probability that a random person will score between 120 and 130? What is the probability that a random person will score 75 or below?

As stated in the problem, the distribution of test results follows a normal distribution with μ_x=100 and σ_x=10. The probability of scoring between 120 and 130 is given by:

$$\begin{aligned} P(120 \leq x \leq 130) &= \int_{120}^{130} P^{norm}(x, \mu_x = 100, \sigma_x = 10)dx \\ &= 1 - P(x < 120) - P(x > 130) \\ &= 1 - CDF(120) - SF(130) \end{aligned}$$

where we have changed the probability to be calculated as 1 minus the probability of being outside of the range (which is easier to calculate using common functions for the cdf and sf). The probability of scoring 75 or lower is given by:

$$\begin{aligned} P(x \leq 75) &= \int_{-\infty}^{75} P^{norm}(x, \mu_x = 100, \sigma_x = 10)dx \\ &= CDF(75) \end{aligned}$$

which is the same as the cdf evaluated at x=75.

These probabilities are easiest to estimate with a computer program, as in the following python code:

```
import scipy.stats as stats

#In order to get the probability of being within a certain range,
#we actually have to calculate the probability of being outside that range!

pBigger130=stats.norm.sf(130,100,10)
pSmaller120=stats.norm.cdf(120,100,10)
print("Prob of scoring between 120 and 130: {:.2f} %".format(100*(1-pBigger130-
    pSmaller120)))
print("Prob of scoring 75 or below: {:.2f} %".format(100*(stats.norm.cdf
    (75,100,10))))
```

The output of the code is:

```
Prob of scoring between 120 and 130: 2.14 %
Prob of scoring 75 or below: 0.62 %
```

7.2 Why most measurements are normally distributed

In this section, we motivate why we expect most measurements to be normally distributed. Suppose that we are measuring some quantity which has a "true" value equal to X, and that our measurement is subject to a small random error ϵ. That is, each time we perform a measurement we have equal probability ($p = 0.5$) of obtaining $X - \epsilon$ or $X + \epsilon$. With this single source of random error, over many measurements, we will obtain $X - \epsilon$, 50% of the time and $X + \epsilon$, the other 50% of the time.

Result	Ways to get result
$X - 2\epsilon$	$-\epsilon - \epsilon$
X	$-\epsilon + \epsilon$ or $+\epsilon - \epsilon$
$X + 2\epsilon$	$+\epsilon + \epsilon$

Table 7.2: Ways to obtain different results when 2 random errors contribute

If we have 2 sources of random error (each equally probable to add or subtract ϵ from the true value X), then we can obtain three possible results: $\{X - 2\epsilon, X, X + 2\epsilon\}$, depending on whether the errors occurred in the same "direction" or not. The possible ways of obtaining the 3 possible results are given in Table 7.2, where we can see that there are 4 possible outcomes, 2 of which are to measure X and two of which are to measure a value that is 2ϵ away. We thus expect to measure X 50% of the time (2 out of 4 possible outcomes), $X - 2\epsilon$, 25% of the time, and $X + 2\epsilon$, 25% of the time.

Another way to think about this is in terms of the binomial distribution. In order to evaluate the probability of obtaining $X + 2\epsilon$, we need to know the probability of both errors being positive. Each error has a probability $p = 0.5$ of being positive, and we would like to know the probability of having $k = 2$ positive errors ("successes") when we add $N = 2$ errors ("trials"). The binomial probability for this case is:

$$P^{binom}(k = 2, N = 2, p = 0.5) = 0.25$$

just as we obtained by counting the possible outcomes in Table 7.2.

Now, if we have N source of random errors, each error being equal to ϵ and having equal probability to push our measured value up or down, the distribution of our measurements will be binomially distributed. If a value x requires that k of the errors be positive and $N - k$ of the errors be negative, then the probability of obtaining x is given by:

$$P(x) = P^{binom}(k, N, p = 0.5)$$

which corresponds to the probability that the correct number of errors, k, are positive. The possible range in values of x is between $X - N\epsilon$ and $X + N\epsilon$. The distribution of our measurements will be symmetric about X and each measurement, x, will have a binomial probability of occurring (evaluated by determining how many errors, k, are needed to be positive). If we have k positive errors, then our measured value, x, is given by:

$$x = X + k\epsilon - (N - k)\epsilon \tag{7.7}$$

$$= X + 2k\epsilon - N\epsilon \tag{7.8}$$

If we repeat our measurement many times, x will have a distribution of values. The standard deviation (variance) of the distribution of values of x will depend on the standard deviation (variance) of the values that we obtain for k (X and N are both constants, so they will not contribute to giving a distribution of values to x)[1]. A standard deviation (i.e. a spread) in the values of k of σ_k will result in a standard deviation (a spread) for values for x given by:

$$\sigma_x = 2\sigma_k\epsilon \tag{7.9}$$

[1]The variance of a sum of uncorrelated quantities is the sum of their variances

k is distributed according to the binomial distribution, which, as you recall, has a standard deviation given by:

$$\sigma_k = \sqrt{Np(1-p)} = \sqrt{N\frac{1}{2}(1-\frac{1}{2})} = \frac{1}{2}\sqrt{N} \tag{7.10}$$

where we have used the fact that $p = 0.5$. The standard deviation of x is thus given by:

$$\sigma_x = 2\sigma_k\epsilon = \epsilon\sqrt{N} \tag{7.11}$$

We now argue that as ϵ becomes infinitely small and N becomes infinitely large, but in a way that $\epsilon\sqrt{N}$ is finite and equal to σ_x, the distribution of the values of x approaches a normal distribution with mean X and standard deviation σ_x. That is, in the limit of an infinite number of infinitely small random errors, we obtain a normal distribution for the values of x with $\mu = X$ and $\sigma = \sigma_x$.

We already saw that the binomial distribution approaches the normal distribution when the mean of the binomial distribution (given by Np) is large. This is true in our case since N approaches infinity while $p = 0.5$ is finite. Remember, N, is the number of small random errors, and the mean of the binomial distribution, Np, is the mean number, k, of positive errors that result in a certain measured value x. This is not the same as requiring that X (the mean of the distribution of the x) be large. Requiring that the mean of the number of positive errors be large has no impact on the mean value of the measured quantities x. We also require that the individual errors, ϵ, become very small. This means that the possible values of x that we can measure (which are spaced by ϵ) become closer and closer together, and eventually approach a continuous distribution.

Figure 7.5 shows how the normal distribution is approached when the number of random errors, N, is increased at the same time as the random error, ϵ is decreased in a way to keep $\epsilon\sqrt{N}$ constant and equal to 0.1. The histograms show the results of 100,000 (simulated) measurements of x with a given N and ϵ. The histograms are normalized, and are seen to approach the normal distribution (black line) as N becomes large. The mean value of the measurements of x was chosen to be $X = 1.0$ (which is small) to illustrate that while N is large, this does not require the mean of the distribution of the x values to be large. The histogram were created by generating random values of k from a binomial distribution with N and $p = 0.5$ and then converting those to the corresponding values of x, given by $x = X + 2k\epsilon - N\epsilon$.

7.2.1 The Central Limit Theorem

Although a full development of the Central Limit Theorem (CLT) is beyond the scope of this book, it is certainly worth mentioning in this context. The CLT states that the distribution of a sum of a large number of independent variables is normally distributed, *regardless* of the distribution that generated the independent variables. This is in fact a remarkable result, and applies to a wide range of quantities, as many things can be thought of as sums.

For example, the final marks from all of the students in a course are usually normally distributed (if there are many students and each students has multiple marks contributing to their final mark). The final mark from a single student is given by the sum of all of the marks that they obtained in the course (usually scaled by some number to get an average, but ultimately, it is still a sum). Each individual mark that the student obtained may not be from a normal distribution (e.g. for a given assignment, the professor could have graded the marks such that they are not normally

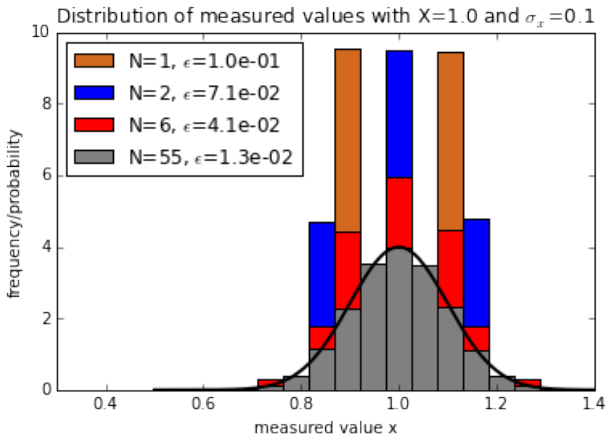

Figure 7.5: Normalized histograms for the results of 100,000 measurements of a value of x that is centred about $X = 1.0$ with a standard deviation of 0.1. The different histograms show how the distribution changes as the number of random errors, N, is increased and the random error of each measurement, ϵ, is decreased in such a was as to keep $\sigma_x = \epsilon\sqrt{N}$ constant. As N increased, the normal distribution, shown by the black line, is approached.

distributed). It is thus quite remarkable that even if the individual assignments marks are not normally distributed, the final marks in the course are! This is the reason that some professors will adjust their final marks with a "bell curve" (a normal distribution!). In effect, the strategy is usually to shift all of the marks so that only a fraction of the students fail (e.g. only those 1 standard deviation below the mean).

Figure 7.6 shows a simulated histogram to illustrate the CTL. In this case, a class of 100 students was simulated. Each student has 20 marks (e.g. from 20 assignments), and for each of those marks, the student has an equal probability of obtaining any mark between 0 and 10 (a flat distribution of marks, certainly not a normal distribution). As you can seen, when we compute the average mark for each student (their "final mark"), and plot the distribution of those marks for all students, we get a distribution that looks like a normal distribution, remarkable!

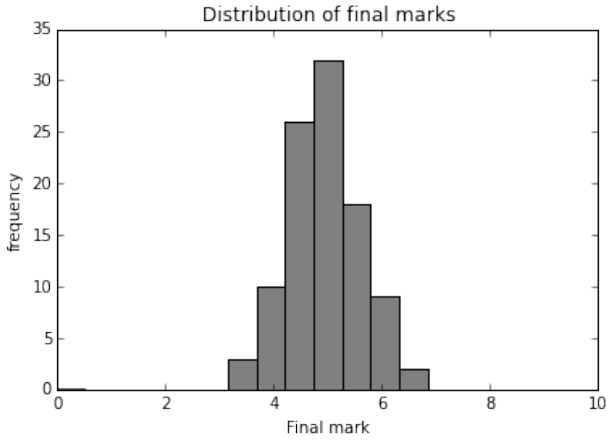

Figure 7.6: Simulated histogram of final marks from 100 students in a course.

7.3 Combining normally distributed quantities

In this section, we examine the distribution of a quantity that depends on one or more normally distributed quantities.

7.3.1 Addition of a number to a normally distributed quantity

Given a quantity, x, that is normally distributed with a probability $P_x^{norm}(x)$, we wish to know the distribution, $P_F(F)$ of a quantity, F, that is given by:

$$F = a + x$$

where a is a constant. Since x is normally distributed, we have:

$$P_x(x) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}}$$

and we want to know the probability, $P_F(F)$, of obtaining a certain value of F. Since $F = a+x$, the probability of obtaining a certain value of F is the same as the probability of obtaining $x = F - a$:

$$\begin{aligned} P_F(F) = P_x(x = F - a) &= \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{((F-a)-\mu_x)^2}{2\sigma_x^2}} \\ &= \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(F-(a+\mu_x))^2}{2\sigma_x^2}} \\ &= \frac{1}{\sigma_F\sqrt{2\pi}} e^{-\frac{(F-\mu_F)^2}{2\sigma_F^2}} \end{aligned}$$

Thus, F is also normally distributed, with the same standard deviation as x, $\sigma_F = \sigma_x$, but with the mean shifted to $\mu_F = x + a$.

7.3.2 Multiplication of a number with a normally distributed quantity

Given a quantity, x, that is normally distributed with a probability $P_x^{norm}(x)$, we wish to know the distribution, $P_F(F)$ of a quantity, F, that is given by:

$$F = ax$$

where a is a constant. Since x is normally distributed, we have:

$$P_x(x) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}}$$

and we want to know the probability, $P_F(F)$, of obtaining a certain value of F. Since $F = ax$, the probability of obtaining a certain value of F is the same as obtaining a value of $x = \frac{F}{a}$:

$$\begin{aligned} P_F(F) = P_x(x = \frac{F}{a}) &= \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(\frac{F}{a}-\mu_x)^2}{2\sigma_x^2}} \\ &= \frac{1}{a\sigma_x\sqrt{2\pi}} e^{-\frac{(F-a\mu_x)^2}{2\sigma_x^2 a^2}} \\ &= \frac{1}{\sigma_F\sqrt{2\pi}} e^{-\frac{(F-\mu_F)^2}{2\sigma_F^2}} \end{aligned}$$

where we had to divide the term out front by a to keep the probability density function normalized. Thus, F, is also normally distributed with a mean of $\mu_F = a\mu_x$ and a standard deviation of $\sigma_F = a\sigma_x$. This is the same result that we obtained in Chapter 4.

7.3.3 Sum of normally distributed quantities

Let x and y be two independent quantities that are normally distributed about μ_x and μ_y, with standard deviations σ_x and σ_y, respectively. We would like to know the expected distribution of their sum, $z = x + y$. We will show that z is normally distributed with a mean $\mu_z = \mu_x + \mu_y$ and a standard deviation $\sigma_z^2 = \sigma_x^2 + \sigma_y^2$. If we interpret the standard deviations as the uncertainties on the measured quantities, then we will effectively have shown that adding in quadrature is the correct method to obtain the uncertainty on the sum of two quantities.

Let x and y be distributed according to probability density functions, $P_x(x)$ and $P_y(y)$, respectively (which we will later set equal to the normal distribution). We also assume that x and y are independent (that is, the probability of a specific value of y does not depend on the value of x, and vice versa). We thus wish to find the probability density function for z, which we will call $P_z(z)$.

We start by assuming that x, y, and z can only take discrete integer values and we will later change to continuous variables. If x is equal to some value k, and $z = x + y$, we must have that $y = z - k$. For a given value $x = k$, then the probability of having a certain value of z is the product of $P_x(x = k)$ and $P_y(y = z - k)$ (i.e. the joint probability of having a specific value of x and a specific value of y):

$$P_z(z|x = k) = P_x(x = k)P_y(y = z - k)$$

where we have indicated, $P_z(z|x = k)$, that this is the probability of a certain value of z *given* that $x = k$. Of course, we do not care what value x has, since k is a completely arbitrary number. We really want to sum this probability over all of the possible values of k to obtain $P_z(z)$ independent of the specific value of x:

$$\begin{aligned} P_z(z) &= \sum_k P_z(z|x = k) \\ P_z(z) &= \sum_k P_x(k)P_y(z - k) \end{aligned} \tag{7.12}$$

which can be applied to any discrete distributions, P_x and P_y, to obtain the distribution of their sum.

EXAMPLE 7-4: When rolling two dice, what is the probability that their sum is 7?

Using our above notation, we are interested in the probability of $P_z(7)$ given the probability distribution for digits from each of the dice. If the number on the individual dice are given by x and y with probabilities $P_x(x)$ and $P_y(y)$, respectively, equation 7.12 gives $P_z(7)$ as:

$$\begin{aligned} P_z(7) &= \sum_{k=1}^{k=6} P_x(k)P_y(7-k) \\ &= P_x(1)P_y(6) + P_x(2)P_y(5) + P_x(3)P_y(4) + P_x(4)P_y(3) + P_x(5)P_y(2) + P_x(6)P_y(1) \end{aligned}$$

If both dice are fair, then $P_x = P_y = \frac{1}{6}$. The probability of $P_z(7)$ is thus $6 \times \frac{1}{6}\frac{1}{6} = \frac{1}{6}$, which can also be found by tabulating all of the possible outcomes of throwing the dice and counting how many of the outcomes sum to 7.

If we are now interested in the continuous case, the sum in equation 7.12 must become an integral:

$$P_z(z) = \int_{k=-\infty}^{k=+\infty} P_x(k)P_y(z-k)dk \tag{7.13}$$

which is valid to obtain the probability of a sum of two continuous independent variables given by probability density functions P_x and P_y. Equation 7.13 is called a "convolution" of the functions P_x and P_y, and appears in many areas of physics and mathematics not related to probabilities. We thus have our formula to compute the functional form of the distribution of the sum of x and y.

For normal distributions, P_x and P_y are given by:

$$\begin{aligned} P_x(x) &= \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \\ P_y(y) &= \frac{1}{\sigma_y\sqrt{2\pi}} e^{-\frac{(y-\mu_y)^2}{2\sigma_y^2}} \end{aligned}$$

Inserting this into equation 7.13:

$$\begin{aligned} P_z(z) &= \int_{k=-\infty}^{k=+\infty} P_x(k)P_y(z-k)dk \\ &= \int_{k=-\infty}^{k=+\infty} \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(k-\mu_x)^2}{2\sigma_x^2}} \frac{1}{\sigma_y\sqrt{2\pi}} e^{-\frac{((z-k)-\mu_y)^2}{2\sigma_y^2}} dk \end{aligned}$$

This can be re-arranged after a substantial amount of algebraic manipulation to:

$$P_z(z) = \int_{k=-\infty}^{k=+\infty} \frac{1}{\sqrt{\sigma_x^2+\sigma_y^2}\sqrt{2\pi}} e^{-\frac{(z-(\mu_x+\mu_y))^2}{2(\sigma_x^2+\sigma_y^2)}} \frac{1}{\frac{\sigma_x\sigma_y}{\sqrt{\sigma_x^2+\sigma_y^2}}\sqrt{2\pi}} e^{-\frac{\left(k-\frac{\sigma_x^2(z-\mu_y)+\sigma_y^2\mu_x}{\sigma_x^2+\sigma_y^2}\right)^2}{2\left(\frac{\sigma_x\sigma_y}{\sqrt{\sigma_x^2+\sigma_y^2}}\right)^2}} dk$$

The first two multiplicative terms do not depend on k, so they can be taken out of the integral:

$$P_z(z) = \frac{1}{\sqrt{\sigma_x^2+\sigma_y^2}\sqrt{2\pi}} e^{-\frac{(z-(\mu_x+\mu_y))^2}{2(\sigma_x^2+\sigma_y^2)}} \int_{k=-\infty}^{k=+\infty} \frac{1}{\frac{\sigma_x\sigma_y}{\sqrt{\sigma_x^2+\sigma_y^2}}\sqrt{2\pi}} e^{-\frac{\left(k-\frac{\sigma_x^2(z-\mu_y)+\sigma_y^2\mu_x}{\sigma_x^2+\sigma_y^2}\right)^2}{2\left(\frac{\sigma_x\sigma_y}{\sqrt{\sigma_x^2+\sigma_y^2}}\right)^2}} dk$$

$$= \frac{1}{\sqrt{\sigma_x^2+\sigma_y^2}\sqrt{2\pi}} e^{-\frac{(z-(\mu_x+\mu_y))^2}{2(\sigma_x^2+\sigma_y^2)}}$$

where we have recognized that the remaining integral corresponded to the integral of a normal distribution, which is equal to 1. We are thus left with $P_z(z)$ given by a normal distribution with mean $\mu_z = \mu_x + \mu_y$ and standard deviation $\sigma_z = \sqrt{\sigma_x^2+\sigma_y^2}$, as anticipated. Thus, adding errors in quadrature is a prescription for obtaining an error on a calculated quantity that corresponds to the standard deviation in that quantity assuming that it is normally distributed.

Although we just proved[2] that the sum of two quantities is normally distributed, we can generalize the result for any function of two variables. If a quantity $F(x,y)$ depends on independent and normally distributed quantities x and y, and the standard deviations on x and y are "small", then we can use a Taylor series to approximate the value of F for x and y near their mean values, μ_x and μ_y:

$$F(x,y) \approx F(\mu_x,\mu_y) + \frac{\partial F}{\partial x}(x-\mu_x) + \frac{\partial F}{\partial y}(y-\mu_y)$$

Since we assumed that the errors in x and y are small, then it is reasonable to assume that x and y will generally be close to their means, μ_x and μ_y and that the Taylor series approximation is valid (i.e. that $x-\mu_x$ and $y-\mu_y$ are small). F can thus be approximated by the sum of three terms; since we know how a sum is distributed, we only need to know how the three terms that make up the sum are distributed.

The first term is constant and will only shift the mean of the distribution of the sum (but does not add to the standard deviation). The second term, $\frac{\partial F}{\partial x}(x-\mu_x)$, is the product of a constant, $\frac{\partial F}{\partial x}$ (evaluated at $x=\mu_x$ and $y=\mu_y$), and a quantity, $x-\mu_x$, that is normally distributed about 0 with a standard deviation of σ_x. The second term is thus normally distributed with a mean of zero and a standard deviation given by $\frac{\partial F}{\partial x}\sigma_x$. The third term is similar, with a mean of zero and a standard deviation of $\frac{\partial F}{\partial y}\sigma_y$. F is thus the sum of a constant term, and two normal distributions with means of 0 and standard deviations of $\frac{\partial F}{\partial x}\sigma_x$ and $\frac{\partial F}{\partial x}\sigma_x$, respectively. F is thus normally distributed with a mean and standard deviations given by:

$$\begin{aligned} \mu_F &= F(\mu_x,\mu_y) \\ \sigma_F^2 &= \left(\frac{\partial F}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial F}{\partial y}\sigma_y\right)^2 \end{aligned} \tag{7.14}$$

just as we had prescribed in Chapter 4. This result is easily extended if F depends on more than two variables.

[2] Minus the lengthy algebra!

7.4 Determining parameters of the normal distribution from data

Given a set of N measurements, $\{x_1, x_2, \ldots, x_N\}$, of a normally distributed quantity, we wish to estimate the "true" value of the mean and standard deviation, μ_x and σ_x, of the normal distribution that describes the data. As we will see, the sample mean and the sample standard deviation are good estimates of the mean and standard deviation of the underlying normal distribution.

7.4.1 The principle of maximum likelihood

Given a probability density function, $P(x, \vec{\alpha})$, we wish to determine the set of parameters (represented as a vector, $\vec{\alpha}$) that best describes a set of data that is modelled by that pdf. For example, the pdf may be that for the normal distribution, $P^{norm}(x, \mu_x, \sigma_x)$, for which we wish to estimate μ_x and σ_x, given a set of N measurements, $\{x_1, x_2, \ldots, x_N\}$.

The "principle of maximum likelihood" is a procedure to determine the value of the parameters $\vec{\alpha}$ for which the "likelihood" of the data is maximized. The likelihood, $L(x|\vec{\alpha})$, is essentially the probability of obtaining a particular data set given a particular set of parameters. For example, the likelihood of obtaining the N data points $\{x_1, x_2, \ldots, x_N\}$ would be given by the joint probability of obtaining each individual data point:

$$\begin{aligned} L(x|\vec{\alpha}) &= P(x_1, \vec{\alpha})P(x_2, \vec{\alpha}) \ldots P(x_N, \vec{\alpha}) \\ &= \prod_{i=1}^{i=N} P(x_i, \vec{\alpha}) \end{aligned} \tag{7.15}$$

Given a specific value of the parameters, $\vec{\alpha}$, we can calculate the likelihood. We can then vary those parameters until the likelihood is maximized, yielding our best estimate of the parameters. Formally, we want to find the set of parameters, $\{\alpha_1, \alpha_2, \ldots\}$, such that:

$$\frac{\partial L}{\partial \alpha_i} = 0 \tag{7.16}$$

since the maximum of L is found when the derivative is zero.

7.4.2 Estimating the mean and standard deviation of a normal distribution from data

In the case of the normal distribution, the likelihood of obtaining a particular set of N measurements, $\{x_1, x_2, \ldots, x_N\}$, given a particular choice of mean, μ_x, and standard deviation, σ_x, is:

$$\begin{aligned} L(x|\mu_x, \sigma_x) &= \prod_{i=1}^{i=N} P^{norm}(x_i, \mu_x, \sigma_x) \\ &= \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x_1-\mu_x)^2}{2\sigma_x^2}} \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x_2-\mu_x)^2}{2\sigma_x^2}} \cdots \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x_N-\mu_x)^2}{2\sigma_x^2}} \\ &= \left(\frac{1}{\sigma_x\sqrt{2\pi}}\right)^N e^{-\frac{1}{2}\left(\frac{(x_1-\mu_x)^2}{\sigma_x^2} + \frac{(x_2-\mu_x)^2}{\sigma_x^2} + \cdots + \frac{(x_N-\mu_x)^2}{\sigma_x^2}\right)} \end{aligned} \tag{7.17}$$

Because we are only interested in the maximum value of the likelihood function, we can work with the logarithm, $\ln(L)$, instead of the likelihood function itself (the logarithm is a monotonically

increasing function and will thus have a maximum at the same value of the parameters as the likelihood function). For convenience, we prefer to *minimize* the negative of the logarithm of the likelihood, instead of maximizing the logarithm of the likelihood. That is, we introduce the "negative log-likelihood", $-\ln(L)$:

$$\begin{aligned}
-\ln(L) &= -\ln\left(\left[\frac{1}{\sigma_x\sqrt{2\pi}}\right]^N e^{-\frac{1}{2}\left(\frac{(x_1-\mu_x)^2}{\sigma_x^2}+\frac{(x_2-\mu_x)^2}{\sigma_x^2}+\cdots+\frac{(x_N-\mu_x)^2}{\sigma_x^2}\right)}\right) \\
&= -\ln\left(\left[\frac{1}{\sigma_x\sqrt{2\pi}}\right]^N\right) + \frac{1}{2}\left(\frac{(x_1-\mu_x)^2}{\sigma_x^2}+\frac{(x_2-\mu_x)^2}{\sigma_x^2}+\cdots+\frac{(x_N-\mu_x)^2}{\sigma_x^2}\right) \\
&= N\ln(\sqrt{2\pi}) + N\ln(\sigma_x) + \frac{1}{2}\sum_{i=1}^{i=N}\frac{(x_i-\mu_x)^2}{\sigma_x^2} \\
&= N\ln(\sigma_x) + \frac{1}{2\sigma_x^2}\sum_{i=1}^{i=N}(x_i-\mu_x)^2 \qquad (7.18)
\end{aligned}$$

where in the last line, we dropped the constant term, $N\ln(\sqrt{2\pi})$, since it will have no influence on minimizing the negative log-likelihood in terms of μ_x or σ_x. Finding the set of parameters that maximize the likelihood function is thus equivalent to finding the parameters that minimize the negative log-likelihood function.

Thus, the best estimate of μ_x from the data, $\hat{\mu}_x$, using the principle of maximum likelihood, is given by the condition:

$$\frac{\partial}{\partial\mu_x}\left(-\ln(L)\right) = 0 \qquad (7.19)$$

and the best estimate of the standard deviation, σ_x, from the data $\hat{\sigma}_x$ is given by:

$$\frac{\partial}{\partial\sigma_x}\left(-\ln(L)\right) = 0 \qquad (7.20)$$

We call $\hat{\mu}_x$ and $\hat{\sigma}_x$ the "maximum likelihood estimates" of the true parameters. We used hats on the quantities to denote that these are estimated from the data as we do not know the "true" values, μ_x and σ_x. Also note that there are methods other than maximum likelihood that can be used to estimate parameters. It certainly makes intuitive sense to think of maximizing the probability of obtaining a given data set, but it is difficult to justify scientifically (e.g. what is really meant by "probability"?). For this reason, you should remember to always specify the method used to estimate your parameters from data.

Taking the derivative of the negative log-likelihood function with respect to μ_x, we have:

$$\begin{aligned}
\frac{\partial}{\partial \mu_x}\left(-\ln(L)\right) &= \frac{\partial}{\partial \mu_x}\left(N\ln(\sigma_x) + \frac{1}{2\sigma_x^2}\sum_{i=1}^{i=N}(x_i-\mu_x)^2\right) \\
&= \frac{1}{2\sigma_x^2}\sum_{i=1}^{i=N}\frac{\partial}{\partial \mu_x}(x_i-\mu_x)^2 \\
&= \frac{1}{2\sigma_x^2}\sum_{i=1}^{i=N}-2(x_i-\mu_x) \\
&= \frac{-1}{\sigma_x^2}\left(\sum_{i=1}^{i=N}x_i - \mu_x\right) \\
&= \frac{1}{\sigma_x^2}\left(N\mu_x - \sum_{i=1}^{i=N}x_i\right)
\end{aligned} \tag{7.21}$$

Setting the above expression to zero, we find that:

$$\begin{aligned}
\frac{1}{\sigma_x^2}\left(N\mu_x - \sum_{i=1}^{i=N}x_i\right) &= 0 \\
\mu_x &= \frac{1}{N}\sum_{i=1}^{i=N}x_i \\
\therefore \hat{\mu}_x &= \bar{x}
\end{aligned} \tag{7.22}$$

Thus, the maximum likelihood estimate of the mean of the normal distribution, $\hat{\mu}_x$, is precisely the sample mean, $\bar{x}$, of the data!

We now derive the formula for the maximum likelihood estimate of the standard deviation:

$$\begin{aligned}
\frac{\partial}{\partial \sigma_x}\left(-\ln(L)\right) &= \frac{\partial}{\partial \sigma_x}\left(N\ln(\sigma_x) + \frac{1}{2\sigma_x^2}\sum_{i=1}^{i=N}(x_i-\mu_x)^2\right) \\
&= \frac{N}{\sigma_x} + \frac{\partial}{\partial \sigma_x}\frac{1}{2\sigma_x^2}\sum_{i=1}^{i=N}(x_i-\mu_x)^2 \\
&= \frac{N}{\sigma_x} - \frac{1}{\sigma_x^3}\sum_{i=1}^{i=N}(x_i-\mu_x)^2
\end{aligned} \tag{7.23}$$

Setting the above expression to zero, we have:

$$\begin{aligned}
\frac{N}{\sigma_x} - \frac{1}{\sigma_x^3}\sum_{i=1}^{i=N}(x_i-\mu_x)^2 &= 0 \\
\sigma_x^2 &= \frac{1}{N}\sum_{i=1}^{i=N}(x_i-\mu_x)^2
\end{aligned} \tag{7.24}$$

which gives us the maximum likelihood estimator for the standard deviation (or rather, the variance). You may now notice that we cannot use our data to actually calculate the estimate of the variance, σ_x^2, because it depends on the "true" mean of the distribution, μ_x, which we do not know! We can only use our estimate of the mean, $\hat{\mu}_x$, in its place. However, when we replace the true mean, μ_x, with its estimate, we cannot be guaranteed that our estimate for the variance, $\hat{\sigma}_x^2$, is unbiased. In fact, we already pointed this out in Chapter 5 where we argued that in order to get an unbiased measure of the variance, we had to replace the N in the denominator by $N-1$. The correct estimate of the variance of the normal distribution from data is given by:

$$\therefore \hat{\sigma}_x^2 = \frac{1}{N-1} \sum_{i=1}^{i=N} (x_i - \hat{\mu}_x)^2 = \sigma^2 \tag{7.25}$$

where σ^2 is the sample variance, as defined in Chapter 5. The maximum likelihood estimate of the standard deviation (which does not have the $N-1$) is biased, although the effect is small even for only moderately large N.

7.4.3 The error on the mean

As we just saw, the sample mean, $\bar{x}$, and sample standard deviations, σ, from a set of N measurements, $\{x_1, x_2, \ldots, x_N\}$ can be used as estimates of the true mean, μ_x, and standard deviation, σ_x, of a normal distribution that describes those measurements. We now determine the uncertainty, $\sigma_{\bar{x}}$, on our estimated mean of the normal distribution. In Chapter 5, we introduced the error on the mean, $\sigma_{\bar{x}}$:

$$\sigma_{\bar{x}} \equiv \frac{\sigma}{\sqrt{N}} \tag{7.26}$$

and we want to show here, that this is the correct uncertainty on our estimate of the mean $\hat{\mu}_x = \bar{x}$ of the normal distribution estimated from our measurements.

In order to "measure" the error in our estimate of the mean, we would need to estimate the mean many times and see how that is distributed. We would thus need to perform our experiment many, say, M times. In the first experiment, we would measure N values: $\{x_1^1, x_2^1, \ldots, x_N^1\}$, and obtain a mean value $\bar{x}^1$. We would then repeat the experiment a second time, and obtain another N measurements: $\{x_1^2, x_2^2, \ldots, 2_N^1\}$ with a new mean value $\bar{x}^2$. We would repeat this M times, until we have obtained M mean values: $\{\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^M\}$. In order to understand our error on the mean, we would want to know how the mean values are distributed.

Since the quantity x that we are measuring is normally distributed, we are guaranteed that the mean values, $\bar{x}^i$, are also normally distributed. This is true by the Central Limit Theorem (even if the x were not normally distributed). We also showed explicitly that the sum of normally distributed quantities is normally distributed (the sample mean, $\bar{x}$ is just a sum multiplied by a constant $\frac{1}{N}$). We thus know that the sample means, $\{\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^M\}$, follow a normal distribution, and we are interested in the standard deviation of that distribution. This is easy to determine using our formula (equation 7.14) for the standard deviation of a function of different quantities that are normally distributed.

The sample mean is given by a function of the various measured quantities:

$$\bar{x} = \bar{x}(x_1, x_2, \ldots, x_N) = \frac{1}{N}\sum_{i=1}^{i=N} x_i = \frac{x_1 + x_2 + \cdots + x_N}{N} \tag{7.27}$$

Each x_i in the sum has the same uncertainty, given by the sample standard deviation, σ. Using our equation 7.14 for the standard deviation of a function of normally distributed values, we have:

$$\sigma_{\bar{x}}^2 = \left(\frac{\partial \bar{x}}{\partial x_1}\sigma\right)^2 + \left(\frac{\partial \bar{x}}{\partial x_2}\sigma\right)^2 + \cdots + \left(\frac{\partial \bar{x}}{\partial x_N}\sigma\right)^2$$

All of the derivatives are the same:

$$\frac{\partial \bar{x}}{\partial x_i} = \frac{1}{N}\frac{\partial}{\partial x_i}(x_1 + x_2 + \ldots x_N) = \frac{1}{N}$$

The standard deviation of the mean is thus given by:

$$\begin{aligned}
\sigma_{\bar{x}}^2 &= \left(\frac{\partial \bar{x}}{\partial x_1}\sigma\right)^2 + \left(\frac{\partial \bar{x}}{\partial x_2}\sigma\right)^2 + \cdots + \left(\frac{\partial \bar{x}}{\partial x_N}\sigma\right)^2 \\
&= \left(\frac{1}{N}\sigma\right)^2 + \left(\frac{1}{N}\sigma\right)^2 + \cdots + \left(\frac{1}{N}\sigma\right)^2 \\
&= N \times \left(\frac{1}{N}\sigma\right)^2 \\
&= \frac{1}{N}\sigma^2
\end{aligned} \tag{7.28}$$

which gives the expected result:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}} \tag{7.29}$$

Figure 7.7 shows an illustration of determining the parameters for a normal distribution using data. The histogram corresponds to 40 measurements that are normally distributed and that were sampled from a normal distribution (shown in black) with a "true" mean of 10 and a "true" standard deviation of 1. The sample mean of the measurements is 9.8 and is close to the true mean. The sample standard deviation of the data is 1.0, which is coincidentally exactly equal to the true standard deviation of the normal distribution. Finally, the error on the mean was found to be 0.2, which is representative of the error in our estimate of the true mean of the distribution. The red line shows a normal distribution drawn with the mean and standard deviation estimated from the data. The vertical lines show the range in the mean estimated from the data that comes from the error on the mean.

The error on the mean is often confused with the standard deviation. Given many data points, the standard deviation is a measure of how spread out the individual points are. The error on the mean is representative of our ability to determine the average value of those points. Even if the points are very spread out (large standard deviation), if we have many of them, we can precisely determine the mean value (small error on the mean).

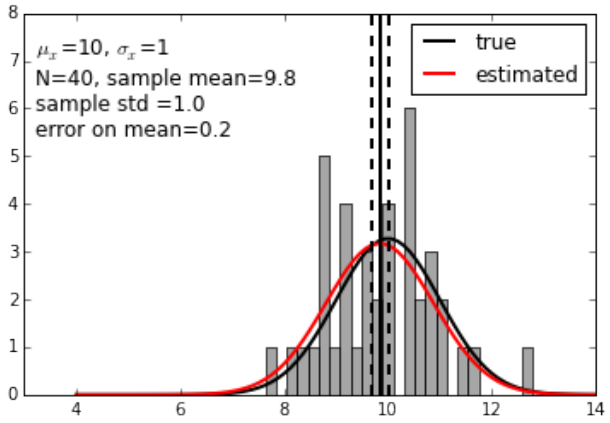

Figure 7.7: Histogram of 40 measurements that originate from a normal distribution (black curve) compared to a normal distribution (red curve) evaluated using the parameters estimated from the data. The vertical lines show the range of the estimated mean given by the error on the mean.

7.5 Averaging multiple measurements with standard errors

Suppose that a quantity, X, has been measured two different times with values $x_a \pm \sigma_a$ and $x_b \pm \sigma_b$. Perhaps the measurements were performed by two different teams, or perhaps they were performed by the same team at different times. Both measurements are quoted with uncertainties that are understood to be "standard" (corresponding to the standard deviation of a normal distribution), and both quoted results may themselves be the mean and standard deviation of multiple measurements. It should appear reasonable that there exists a way to combine the two results into a more accurate estimate, $\hat{X}$, of the unknown value X.

Let us assume that the two measurements, x_a and x_b, are consistent which each other; that is, the difference $|x_a - x_b|$ is not much larger than either σ_a or σ_b. If that were not the case, then we may want to reconsider combining the two measurements, since one of them (or both) is likely affected by a systematic effect. If the two measurements have similar uncertainties, $\sigma_a \approx \sigma_b$, then it would be reasonable to average the two measurements and use our rules for propagating the errors on normally distributed quantities:

$$\begin{aligned} \hat{X} &= \frac{1}{2}(x_a + x_b) \\ \hat{\sigma}_X^2 &= \left(\frac{1}{2}\sigma_a\right)^2 + \left(\frac{1}{2}\sigma_b\right)^2 = \frac{1}{4}(\sigma_a^2 + \sigma_b^2) \\ \therefore \hat{\sigma}_X &= \frac{1}{2}\sqrt{\sigma_a^2 + \sigma_b^2} \end{aligned} \tag{7.30}$$

If the two measurements have the same uncertainty, $\sigma_a = \sigma_b$, then the uncertainty on the average is given by:

$$\hat{\sigma}_X = \frac{1}{2}\sqrt{2\sigma_a^2} = \frac{1}{\sqrt{2}}\sigma_a \tag{7.31}$$

which is smaller than the original uncertainty. Combining these two measurement did indeed reduce the uncertainty in our measurement of X.

Let us suppose that σ_b is much bigger than σ_a; is it still reasonable to average the measurements? First, if the uncertainty on x_b is much bigger, then the measurement x_b is much less precise and it does not make sense to give it "equal weight" as the more precise measurement, x_a, when computing the average. We would expect that the true value of X is closer to x_a than to x_b, so averaging x_a and x_b is intuitively wrong. Second, if we apply this formula, then the resulting uncertainty, $\hat{\sigma}_X$, would be bigger than the original uncertainty on x_a. Combining measurements would result in a less precise estimate of our quantity which does not make sense; we at least intuitively expect that we can only improve our estimate of a given quantity by adding more data [3]. We must thus find a (rigorous) way to include the fact that we should not give as much importance to measurements with large uncertainties when combining measurements.

We can use the principle of maximum likelihood to derive a method for combining measurements whose errors are normally distributed. We assume that the probability for obtaining $x_{a(b)}$ is given

[3] If our two measurements both have small uncertainties and are inconsistent with each other, then we would expect to have a larger uncertainty on the combined measurement. This does occur when we are unable to identify any systematic effect that is leading to the discrepancy and we are effectively forced to include that discrepancy into the uncertainty

by a normal distribution centred at the unknown value, X, with standard deviation $\sigma_{a(b)}$:

$$\begin{aligned} P(x_a) &\propto \frac{1}{\sigma_a} e^{-\frac{(x_a - X)^2}{2\sigma_a^2}} \\ P(x_b) &\propto \frac{1}{\sigma_b} e^{-\frac{(x_b - X)^2}{2\sigma_b^2}} \end{aligned} \tag{7.32}$$

where we have neglected the normalization factor for the probability, since we will only care about maximizing the result (rather than its absolute value as a probability).

We can evaluate the joint probability of obtaining both measurements, given by the product of the two probabilities:

$$\begin{aligned} P(x_a, x_b) &\propto P(x_a) P(x_b) \\ &= \frac{1}{\sigma_a} e^{\frac{(x_a - X)^2}{2\sigma_a^2}} \frac{1}{\sigma_b} e^{\frac{(x_b - X)^2}{2\sigma_b^2}} \\ &= \frac{1}{\sigma_a \sigma_b} e^{-\left(\frac{(x_a - X)^2}{2\sigma_a^2} + \frac{(x_b - X)^2}{2\sigma_b^2}\right)} \end{aligned} \tag{7.33}$$

and we can use this to define our estimate, $\hat{X}$, to be the value of X that maximizes this joint probability (also called the likelihood of our measurements). Again, instead of maximizing the likelihood, we can minimize the negative of the logarithm of the likelihood, given by:

$$\begin{aligned} -\ln P(x_a, x_b) &= -\ln \frac{1}{\sigma_a \sigma_b} e^{-\left(\frac{(x_a - X)^2}{2\sigma_a^2} + \frac{(x_b - X)^2}{2\sigma_b^2}\right)} \\ &= \ln \sigma_a + \ln \sigma_b + \left(\frac{(x_a - X)^2}{2\sigma_a^2} + \frac{(x_b - X)^2}{2\sigma_b^2}\right) \\ &= \ln \sigma_a + \ln \sigma_b + \frac{1}{2}\chi^2 \end{aligned} \tag{7.34}$$

where we have introduced the quantity "chi-squared", χ^2:

$$\chi^2 \equiv \frac{(x_a - X)^2}{\sigma_a^2} + \frac{(x_b - X)^2}{\sigma_b^2} \tag{7.35}$$

Finding the value of X that maximizes the likelihood is thus equivalent to finding the value that minimizes the chi-squared (since the other terms in the log-likelihood do not depend on X and σ_a and σ_b are constants). The task of minimizing a chi-squared comes up often and is sometime called the "method of least squares", since one is trying to find X such that the squared distance to other quantities (x_a and x_b in our case) is minimized.

We can minimize χ^2 analytically in our case, by taking the derivative with respect to X:

$$\begin{aligned} \frac{d\chi^2}{dX} &= \frac{d}{dX}\left(\frac{(x_a - X)^2}{\sigma_a^2} + \frac{(x_b - X)^2}{\sigma_b^2}\right) \\ &= -2\frac{(x_a - X)}{\sigma_a^2} - 2\frac{(x_b - X)}{\sigma_b^2} \end{aligned} \tag{7.36}$$

Setting this equal to zero, we find:

$$\begin{aligned}\frac{(x_a - X)}{\sigma_a^2} + \frac{(x_b - X)}{\sigma_b^2} &= 0 \\ \hat{X}\left(\frac{1}{\sigma_a^2} + \frac{1}{\sigma_b^2}\right) &= \frac{x_a}{\sigma_a^2} + \frac{x_b}{\sigma_b^2}\end{aligned} \tag{7.37}$$

If we introduce "weights", $w_a \equiv \frac{1}{\sigma_a^2}$ and $w_b \equiv \frac{1}{\sigma_b^2}$, we can write this as:

$$\hat{X} = \frac{w_a x_a + w_b x_b}{w_a + w_b} \tag{7.38}$$

and we find that our estimate, $\hat{X}$, obtained by combining the measurements x_a and x_b, is given by the weighted average of x_a and x_b; the weights are given by 1 over the square of the uncertainty on the measurements. If the uncertainty on a measurement is large, then the weight of that measurement will be correspondingly smaller; this is exactly the property that we were after. This result is easily extended when averaging N measurements, $\{x_1, x_2, \ldots, x_N\}$, with corresponding uncertainties, $\{\sigma_1, \sigma_2, \ldots, \sigma_N\}$.

$$\begin{aligned}\hat{X} &= \left(\frac{1}{\sum_{i=1}^{i=N} w_i}\right) \sum_{i=1}^{i=N} w_i x_i \\ w_i &\equiv \frac{1}{\sigma_i^2}\end{aligned} \tag{7.39}$$

Since we now have a function for the weighted average of several quantities, we can apply our error propagation formula to obtain the resulting uncertainty in $\hat{X}$:

$$\hat{\sigma}^2 = \left(\frac{\partial \hat{X}}{\partial x_1}\sigma_1\right)^2 + \left(\frac{\partial \hat{X}}{\partial x_2}\sigma_2\right)^2 + \cdots + \left(\frac{\partial \hat{X}}{\partial x_N}\sigma_N\right)^2 \tag{7.40}$$

The derivatives are all similar:

$$\frac{\partial \hat{X}}{\partial x_j} = \left(\frac{1}{\sum_{i=1}^{i=N} w_i}\right) w_j = \left(\frac{1}{\sum_{i=1}^{i=N} w_i}\right) \frac{1}{\sigma_j^2} \tag{7.41}$$

The uncertainty is thus given by:

$$\begin{aligned}\hat{\sigma}^2 &= \left(\left(\frac{1}{\sum_{i=1}^{i=N} w_i}\right)\frac{1}{\sigma_1^2}\sigma_1\right)^2 + \left(\left(\frac{1}{\sum_{i=1}^{i=N} w_i}\right)\frac{1}{\sigma_2^2}\sigma_2\right)^2 + \cdots + \left(\left(\frac{1}{\sum_{i=1}^{i=N} w_i}\right)\frac{1}{\sigma_N^2}\sigma_N\right)^2 \\ &= \left(\frac{1}{\sum_{i=1}^{i=N} w_i}\right)^2 \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} + \cdots + \frac{1}{\sigma_N^2}\right) \\ &= \left(\frac{1}{\sum_{i=1}^{i=N} w_i}\right)^2 \sum_{i=1}^{i=N} w_i \\ &= \frac{1}{\sum_{i=1}^{i=N} w_i}\end{aligned} \tag{7.42}$$

Therefore, the uncertainty on our estimate of the average, $\hat{X}$, is given by:

$$\hat{\sigma} = \frac{1}{\sqrt{\sum_{i=1}^{i=N} w_i}} \tag{7.43}$$

EXAMPLE 7-5: The ATLAS and CMS experiments at the Large Hadron Collider measured the mass of the Higgs boson by looking for the decay of the Higgs into two gamma rays (amongst other channels), and obtained $m_H^{ATLAS} = 126.02 \pm 0.51\,\text{GeV}$ and $m_H^{CMS} = 124.70 \pm 0.34\,\text{GeV}$, respectively. Both uncertainties contain a contribution from systematic and random errors. Assuming that all of the uncertainties are in fact random (and standard), and that the measurements are considered as being consistent with each other, what is the best estimate and uncertainty of the Higgs mass when combining the results from these two experiments?

First, we compute the weight for each measurement:

$$w^{ATLAS} = \frac{1}{0.51^2} = 3.84$$
$$w^{CMS} = \frac{1}{0.34^2} = 8.65$$

which we then use to compute the weighted average and uncertainty as:

$$\hat{m}_H = \frac{3.84 \times 126.020 + 8.65 \times 124.70}{3.84 + 8.65} = 125.11$$
$$\hat{\sigma} = \frac{1}{\sqrt{3.84 + 8.86}} = 0.28$$

The best estimate from combining these two measurements as if they had random errors and were completely independent, is $m_H = 125.11 \pm 0.28$, which has an uncertainty that is smaller than that of the individual measurements.

7.6 Summary

In this chapter, we introduced the probability density function for the normal distribution:

$$P^{norm}(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{7.44}$$

which has two parameters: the mean (μ) and the standard deviation (σ). The normal (or gaussian) probability density function gives the probability, $P^{norm}(x, \mu, \sigma)dx$, for a continuous variable, x, to be between x and $x + dx$.

The sample mean and sample standard deviation of a set of measurements that are normally distributed coincide with the mean and standard deviation of the normal distribution.

The probability density function (pdf) for the normal distribution is normalized such that the total area under the curve is equal to 1. The area underneath the curve between $\mu - \sigma$ and $\mu + \sigma$ is 0.68. A number drawn at random from the normal distribution thus has a 68% chance of being within 1 standard deviation of the mean.

The binomial and Poisson distribution are both approximated by the normal distribution as their mean value increases. Since many measurements are described by a binomial process (e.g. a large number of small random errors adding up), it follows that most measurements are expected to follow a normal distribution. The Central Limit Theorem furthermore states that the sum of a large number of values drawn from any distribution is also normally distributed. This leads to the normal distribution appearing in many areas of science.

When quoting a number with an uncertainty, e.g. $x \pm \delta x$, it is usually reasonable to define the uncertainty, δx, as representing the standard deviation of a normal distribution with mean x. We thus imply that there is a 68% chance that the true value of x is within the range $x \pm \delta x$.

If the best estimate (x) and uncertainty (δx) in a quantity are the mean and standard deviation of a normal distribution, then we can recover all of the error propagation formulas that we presented in Chapter 4. In particular, we showed that the sum of two normally distributed quantities is normally distributed about the sum of the means of the two values, with a standard deviation that is found by adding in quadrature the standard deviation of the two quantities. We showed that "adding in quadrature" is really the result of combining different quantities that are normally distributed. We showed that for an arbitrary function, $F(x, y)$, of *independent*, and normally distributed values, x and y, that F is also normally distributed with mean and standard deviation given by:

$$\begin{aligned} \mu_F &= F(\mu_x, \mu_y) \\ \sigma_F^2 &= \left(\frac{\partial F}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial F}{\partial y}\sigma_y\right)^2 \end{aligned} \tag{7.45}$$

as long as the error in x and y given by σ_x and σ_y are small.

We showed how to use normally distributed data to determine estimates for the mean and standard deviation of a normal distribution that describes those data. We showed that the sample mean and the square root of the sample variance give the maximum likelihood estimates of the mean and standard deviation of the normal distribution. We showed that the formula for the error on the mean gives the correct error on the estimate of the mean of the normal distribution.

Finally, we use principle of maximum likelihood to show how to combine N measurements $\{x_1, x_2, \ldots, x_N\}$, with corresponding uncertainties, $\{\sigma_1, \sigma_2, \ldots, \sigma_N\}$, into an average measurement, $\hat{X}$, with uncertainty $\hat{\sigma}$:

$$\begin{aligned} \hat{X} &= \left(\frac{1}{\sum_{i=1}^{i=N} w_i}\right) \sum_{i=1}^{i=N} w_i x_i \\ \hat{\sigma} &= \frac{1}{\sqrt{\sum_{i=1}^{i=N} w_i}} \\ w_i &\equiv \frac{1}{\sigma_i^2} \end{aligned} \tag{7.46}$$

which corresponds to a weighted average where each measurement is weighed by the inverse of the square of the uncertainty in the measurement.

8 Statistics - Modelling Data

In this chapter, we examine the vast topic of modelling data. We divide the subject into two topics: parameter estimation and model testing. In parameter estimation, we assume that we have a parametric model that can describe our data, and our goal is to determine the parameters of the model. For example, we may expect a linear relationship between two measured quantities and wish to determine the corresponding constant of proportionality. In model testing, we assume that we have a model, and we wish to verify if the data follow the model. For example, we may have data that look like they are linearly related, but we would like to quantify if that is the case.

8.1 Fitting a straight line

8.1.1 Linear problems

It is often possible to model two quantities in an experiment as being linearly related. For example, suppose that we wish to measure the gravitational constant, g, by measuring the time, t, that it takes for objects to fall a certain distance, x. From Newton's Second Law, we expect to have:

$$x = \frac{1}{2}gt^2 \tag{8.1}$$

which is not a linear relationship between our measured quantities, x and t. However, if every time that we measure t, we then take the square of the value, we find that x is indeed linearly related to t^2 with a slope of $\frac{1}{2}g$.

In such an experiment, you may have a series of measurements as in Table 8.1, where you varied the height, x, from which an object was dropped, and measured the corresponding time, t. As a good experimenter, you performed the measurement several times for each value of x in order to measure the time, t, as the mean of several measurements with an error on the mean time of σ_t. You also repeated the measurements at each height until you obtained an error on the mean time of 0.1 s, which you propagated to t^2. You found that the error on the height, x, was negligible.

Since we varied x while measuring t, and we assumed that the uncertainty in x is small, it makes

height, x (m)	**time, t, (s)**	σ_t **(s)**	t^2 **(s^2)**	$\sigma_{t^2} = \sqrt{2t\sigma_t}$ **(s^2)**
1.0	1.1	0.1	1.2	0.2
2.0	1.3	0.1	1.6	0.3
3.0	1.3	0.1	1.7	0.3
4.0	1.5	0.1	2.2	0.3
5.0	1.5	0.1	2.3	0.3
6.0	1.6	0.1	2.4	0.3
7.0	1.8	0.1	3.1	0.4
8.0	1.8	0.1	3.3	0.4
9.0	1.8	0.1	3.1	0.4
10.0	1.9	0.1	3.7	0.4

Table 8.1: Example measurements of the time, t, that it takes to fall a certain distance, x.

more sense to plot t^2 vs x and model the data as:

$$t^2 = \frac{2}{g}x \tag{8.2}$$

The data from Table 8.1 are plotted in Figure 8.1. The slope of the data in this graph is expected to be $\frac{2}{g}$. We will show in this chapter how to determine the slope and uncertainty on the slope from these data.

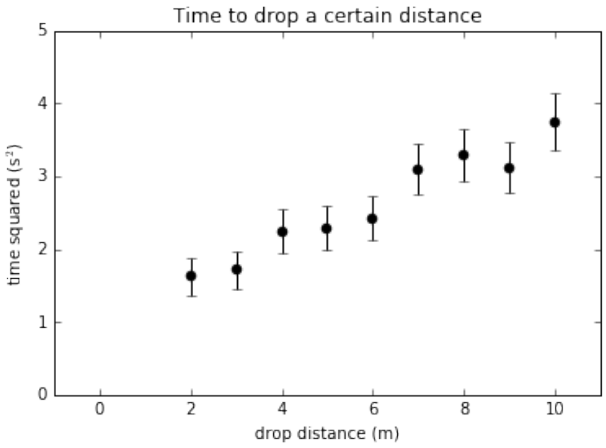

Figure 8.1: Measured time (squared) to fall a certain distance, using data from Table 8.1.

8.1.2 The Method of Least Squares

Given set of N pairs of measurements $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, we wish to determine the best estimates (and uncertainties) for the slope, m, and offset, b, of a straight line:

$$y = mx + b \tag{8.3}$$

that best matches the data. For simplicity, we assume that x can be measured with negligible uncertainty compared to the uncertainties on y. Furthermore, we also assume that each value of y, y_i, have the same uncertainty, σ_y, that corresponds to a standard error.

Equation 8.3 is our model for the data. At a given value of $x = x_i$, we expect that we should measure $y^{exp}(x_i) = mx_i + b$. However, since we cannot measure y exactly (due to random errors in our measurement), we will measure a value y_i that is close to the expected value. If the errors in our measurement are truly random, and we measure y many times for a fixed value of $x = x_i$, then we expect a normal distribution of y_i with a standard deviation, σ_y. Given a value of $x = x_i$, the probability of measuring a specific value, y_i, is given by the normal distribution:

$$P(x_i, y_i) = \frac{1}{\sigma_y\sqrt{2\pi}} e^{-\frac{(y_i - y^{exp}(x_i))^2}{2\sigma_y^2}} \tag{8.4}$$

where the mean of the distribution is given by $y^{exp}(x_i) = mx_i + b$ for the model in equation 8.3.

In order to estimate the parameters of our model, m and b, we use the principle of maximum likelihood to find the value of m and b that yield the highest joint probability (i.e. likelihood) of obtaining our set of measurements. The joint probability of obtaining our data set is given by:

$$P(x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N) \propto \prod_{i=1}^{i=N} \frac{1}{\sigma_y} e^{-\frac{(y_i - y^{exp}(x_i))^2}{2\sigma_y^2}}$$

where we have ignored the normalization constant since we are only interested in the maximal value. As we did in the previous chapter, we take the negative of the logarithm of the likelihood:

$$-\ln P(x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N) = \ln(\sigma_y) + \frac{1}{2}\sum_{i=1}^{i=N} \frac{(y_i - y^{exp}(x_i))^2}{\sigma_y^2} \tag{8.5}$$

Since we are interested in minimizing the negative log-likelihood in terms of m and b, the first term can be ignored, and we can focus on minimizing the chi-squared, χ^2, given by:

$$\chi^2 = \sum_{i=1}^{i=N} \frac{(y_i - y^{exp}(x_i))^2}{\sigma_y^2} \tag{8.6}$$

Before proceeding, we can pause to think of what it means to minimize χ^2. The formula for chi-squared is a sum over all data points of the square distance between the point, y_i, and the model prediction, $y^{exp}(x_i)$. That squared distance is then divided (or scaled) by the uncertainty on the point. Thus, if the points are on average 1 standard deviation away from the model prediction, then they will add approximately 1 unit to the chi-squared. For a good model, we expect the chi-squared to be about equal to the number of points. It should be apparent why the method is called the "method of least squares", since we are effectively minimizing the squared distance between model and data when minimizing the chi-squared.

Inserting our model explicitly for $y^{exp}(x_i)$:

$$\chi^2 = \sum_{i=1}^{i=N} \frac{(y_i - mx_i - b)^2}{\sigma_y^2} \tag{8.7}$$

By taking the derivative of χ^2 with respect to m and b, we can find the values that lead to a

minimum:

$$\frac{d\chi^2}{dm} = \frac{d}{dm}\sum_{i=1}^{i=N}\frac{(y_i - mx_i - b)^2}{\sigma_y^2} = \frac{-2}{\sigma_y^2}\sum_{i=1}^{i=N} x_i(y_i - mx_i - b) \tag{8.8}$$

$$\frac{d\chi^2}{db} = \frac{d}{db}\sum_{i=1}^{i=N}\frac{(y_i - mx_i - b)^2}{\sigma_y^2} = \frac{-2}{\sigma_y^2}\sum_{i=1}^{i=N} (y_i - mx_i - b) \tag{8.9}$$

Setting these two equations equal to zero:

$$\sum_{i=1}^{i=N} x_i(y_i - mx_i - b) = \sum_{i=1}^{i=N} x_i y_i - m\sum_{i=1}^{i=N} x_i^2 - b\sum_{i=1}^{i=N} x_i = 0 \tag{8.10}$$

$$\sum_{i=1}^{i=N} (y_i - mx_i - b) = \sum_{i=1}^{i=N} y_i - m\sum_{i=1}^{i=N} x_i - Nb = 0 \tag{8.11}$$

and rearranging:

$$m\sum_{i=1}^{i=N} x_i^2 + b\sum_{i=1}^{i=N} x_i = \sum_{i=1}^{i=N} x_i y_i \tag{8.12}$$

$$m\sum_{i=1}^{i=N} x_i + Nb = \sum_{i=1}^{i=N} y_i \tag{8.13}$$

which is a system of linear equations that is easily solved for m and b:

$$m = \frac{N\sum xy - \sum x \sum y}{N\sum x^2 - \left(\sum x\right)^2}$$

$$b = \frac{\sum x^2 \sum y - \sum x \sum xy}{N\sum x^2 - \left(\sum x\right)^2} \tag{8.14}$$

where we have omitted the indices on the x_i and y_i and we have implied that the sums are from $i = 1$ to $i = N$. Now that we have formulas for the m and b, we can propagate the errors to obtain:

$$\sigma_m = \sigma_y\sqrt{\frac{N}{N\sum x^2 - \left(\sum x\right)^2}}$$

$$\sigma_b = \sigma_y\sqrt{\frac{\sum x^2}{N\sum x^2 - \left(\sum x\right)^2}} \tag{8.15}$$

EXAMPLE 8-1: Use equations 8.14 and 8.15 to determine the slope and offset for a line that best fit the data in Figure 8.1, and thus determine a value of the gravitational constant g with its uncertainty from the data.

We want to plot the data of t^2 as a function of y, and determine the slope and offset of those data. The slope will be equal to $\frac{2}{g}$ (equation 8.2), and can thus be used to determine our measured value of g. We can use equations 8.14 and 8.15 to determine the slope and offset, and then use error propagation to convert the error in the slope into an error in g. If the error in the slope is σ_m, the error in g, σ_g, will be given by:

$$g = \frac{2}{m}$$

$$\sigma_g = \sqrt{\left(\frac{dg}{dm}\sigma_m\right)^2} = \frac{2}{m^2}\sigma_m$$

The rest of the calculations are done easily in python:

```
import numpy as np
from math import *
import pylab as pl
import scipy.stats as stats
%matplotlib inline

#Copy and paste data from the table in the notes, add commas:
datalist=[
1.0, 1.1, 0.1, 1.2, 0.2,
2.0, 1.3, 0.1, 1.6, 0.3,
3.0, 1.3, 0.1, 1.7, 0.3,
4.0, 1.5, 0.1, 2.2, 0.3,
5.0, 1.5, 0.1, 2.3, 0.3,
6.0, 1.6, 0.1, 2.4, 0.3,
7.0, 1.8, 0.1, 3.1, 0.4,
8.0, 1.8, 0.1, 3.3, 0.4,
9.0, 1.8, 0.1, 3.1, 0.4,
10.0, 1.9, 0.1, 3.7, 0.4,
]
#convert this to a numpy array, and change the shape
data=np.array(datalist)
data=data.reshape(10,5) #changes the shape to get the data in the same format as
    table
#Get the colums out of the table:
x=data[:,0]#This is the first column of data (x)
t2=data[:,3]#The 4th column (t^2)
sigma_t2=data[:,4] #The 5th colum, sigma_t^2

#calculate the slope and offset
#start by calculating all of the sums
N=x.size
x2sum=(x*x).sum()
xt2sum=(x*t2).sum()
xsum=x.sum()
t2sum=t2.sum()

#We will use the mean value of the sigma_t2 as the "error in y"

denominator=N*x2sum-xsum**2
#slope:
m=(N*xt2sum-xsum*t2sum)/denominator
sigma_m=sigma_t2.mean()*sqrt(N/denominator)# using mean error in t^2
#offset:
b=(x2sum*t2sum-xsum*xt2sum)/denominator
sigma_b=sigma_t2.mean()*sqrt(x2sum/denominator)# using mean error in t^2

gmeas=2.0/m
sigma_g=2/m/m*sigma_m

#The fit results
text='''Fit results: t$^2$=mx+b
m= {:.3f} +/- {:.3f}
b= {:.3f} +/- {:.3f}
g= {:.1f} +/- {:.1f} '''.format(m,sigma_m,b,sigma_b,gmeas,sigma_g)

#calculate probability of obtaining a value this far away from the accepted value
dsigma=(9.81-gmeas)/sigma_g
prob=2.0*stats.norm.sf(dsigma)
```

```
print("Probability to get g {:.1f} sigma or further from accepted value: {:.3f}".
    format(dsigma,prob))

#plot:
#The data:
pl.errorbar(x,t2,yerr=sigma_t2,fmt='o',color='black')
#The fit:
pl.plot(x,m*x+b,color='red')
#The fit with m+/-sigma_c
pl.plot(x,(m+sigma_m)*x+b,'--',color='red')
pl.plot(x,(m-sigma_m)*x+b,'--',color='red')

pl.text(0,3,text,fontsize=12)#fit results onto the plot
pl.xlabel("x, drop distance (m)")
pl.ylabel("t$^2$, time squared (s$^2$)")
pl.title("Time to drop a certain distance")
pl.axis([-1,11,0,5])
pl.show()
```

The output is:

```
Probability to get g 2.2 sigma or further from accepted value: 0.027
```

The results of this code, are shown in Figure 8.2. Since our formulas in equations 8.14 and 8.15 assumed that all points had the same size vertical error bar, we used the mean of the values in the last column of table 8.1 as the error. We find that the measurement is $g = 7.5 \pm 1.0\, m/s^2$ *which is more than one standard deviation away from the accepted value of g. The value is* $\frac{9.81-7.5}{1.0}\sigma = 2.2\sigma$ *away from the expected value, which has a 2.7% change of occurring. It is not clear if this is due to pure chance or if there is a systematic effect. Given the low probability of obtaining such a shifted result, one should suspect a systematic effect.*

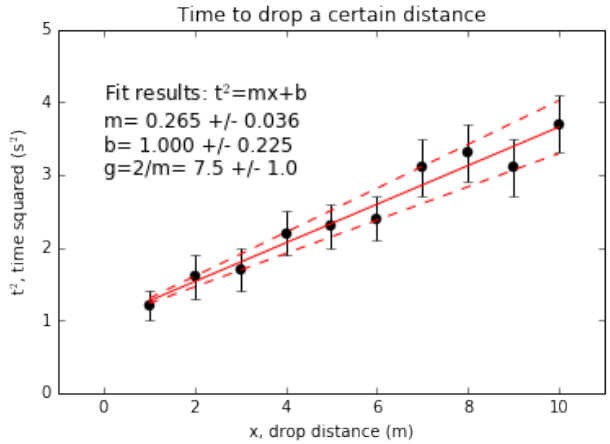

Figure 8.2: Measured time (squared) to fall a certain distance, using data from Table 8.1 and fit (solid red line) using equation 8.14. The slope of the line is also shown for the values of the slope 1 standard deviation away from the mean value.

We should also note that the fit returned a non-zero value for the offset of the fit. Although our model expects the offset to be zero, we have found a value of $b = 1.0 \pm 0.2\, s^2$ *that is inconsistent with zero by* 5σ*. It is thus very likely that there is some systematic offset in the data leading to the time always being measured to be slightly high. Perhaps the person doing the timing systematically stopped the timer late, maybe due to their reaction time. This offset in the timing may also result in a systematic shift of the slope, leading to a systematic shift in the measured value of g.*

We see that even if we expect there to be no offset in the model, it is always best to include it. If there is no real systematic effect, then the offset should fit to a value that is consistent with zero.

The analysis in example 8-1 highlights some interesting aspects of "fitting" a straight line to data using the least squares method (also called "linear regression"). In our physical model, 8.2, we would expect the data to fit to a straight line with a slope and no offset. It is generally true that even if the model does not have an offset, one should always fit a line with an offset to the data. If the data really are consistent with no offset, then the fitted value of the offset will be zero within its uncertainty. If the fitted offset is inconsistent with zero (as it was in the example by 5 standard deviations), then this is a useful tool to uncover a systematic effect. You should thus always include an offset in the model.

8.1.3 Residuals

In order to get an idea of how well our model "fits" the data, it is generally a good idea to look at the "residuals" of the fit. For a data point, (x_i, y_i), with a model $y(x_i)$, the residuals, $R(x_i)$, are defined as:

$$R(x_i) \equiv y_i - y(x_i) \tag{8.16}$$

and correspond to the different between the data and the model prediction. Again, if we assume that the errors in the x_i are negligible, then the error in the residuals are given by:

$$\sigma_{R(x_i)} = \sigma_{y_i} \tag{8.17}$$

That is, the error on the residuals are the same as the error in the data points. If the model is a good representation of the data, then the residuals are expected to be normally distributed around 0. If all of the residuals were positive or negative, then one would question whether the model is a good fit to the data. Similarly, if the residuals show a trend, then one would also suspect that there is an issue with the model. The residuals for the fit from example 8-1 are shown in Figure 8.3 and do not show any obvious trend. Within the uncertainties in the data points, we can conclude that a linear model is a good representation of the data. Note that in this particular experiment, we would expect that friction from drag would likely play an effect for high drop heights, and the residual plot may have helped us to identify this effect. The code to plot the residuals from example 8-1 in python is given by:

```
#The residuals have the same error bars:
pl.errorbar(x,t2-(m*x+b),yerr=sigma_t2,fmt='o',color='black')
pl.xlabel("drop distance (m)")
pl.ylabel("residual from fit (s$^2$)")
pl.title("Residuals from linear fit")
pl.axis([-1,11,-1,1])
pl.show()
```

8.1.4 Correlated parameters

In the least squares fitting for example 8-1, we treated the slope, m, and offset, b, as independent parameters. That is, we explicitly assumed that we could determine the two parameters independently (the formula for determining m does not depend on the value of b). We thus assumed that the chi-squared that we minimized could be minimized independently for the two variables.

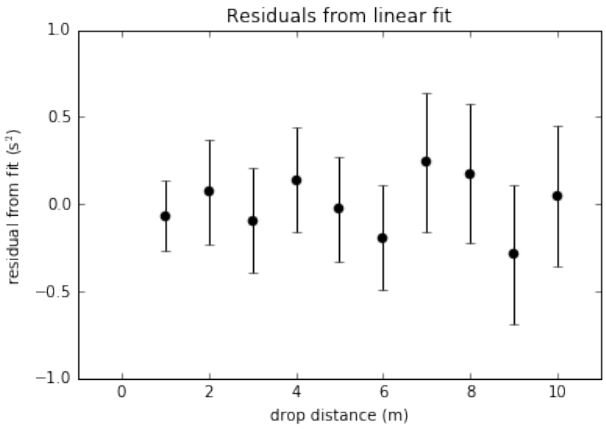

Figure 8.3: Residuals from the fit in Figure 8.2. There is no obvious trend in the residuals and we thus conclude the linear model is a reasonable representation of the data.

However, it is likely the case that as m changes, the corresponding value of b that minimizes the chi-squared changes as well. When this is the case, it makes it difficult to really understand how the errors on m and b need to be combined to understand the error on a quantity, $f(m, b)$, that depends on both m and b (such as for example, $y(x) = mx + b$). As you recall, we can only add errors in quadrature when propagating them if the various quantities with errors are independent of each other. If the value of m that minimizes the chi-squared depends on b, then m and b are not independent.

This is illustrated for the example 8-1 in Figure 8.4 which shows the value of chi-squared as a function of m for different values of b (at the best fit value and the best fit value plus 1 standard deviation). When b is also at its fit value ($b = 1.0$), we see that the minimum of the chi-squared indeed occurs when m is at its best fit value of $m = 0.265$; however, this is no longer the case when $b = 1.2$. We thus say the parameters m and b are correlated (rather, in this case, they are anti-correlated, as a larger value of one parameter leads to a lower optimal value of the other parameter). The χ^2 is also shown in Figure 8.5 as a function of both m and b, and the tilt in the plot is a sign that the parameters are correlated.

The code to make the two plots in python is:

```
#A function to calculate chi-squared for a given set of data, errors on data, and
    model predictions
def chis(ydata,sigma_y,ymodel):
    return (((ydata-ymodel)/sigma_y)**2).sum()

ydata=t2
sigma_y=sigma_t2

#an array of values of b and m centered around the best fit value:
b_vals=np.linspace(b-2*sigma_b,b+2*sigma_b,100)
m_vals=np.linspace(m-2*sigma_m,m+2*sigma_m,100)

#Plot the chi-squared vs m, for 2 different values of b
pl.plot(m_vals,[chis(ydata,sigma_y,m_vals[i]*x+b) for i in range(m_vals.size)],
    label="b={:.1f}".format(b))
pl.plot(m_vals,[chis(ydata,sigma_y,m_vals[i]*x+(b+sigma_b)) for i in range(m_vals.
```

```
    size)],label="b={:.1f}".format(b+sigma_b))
pl.legend(loc='best')
pl.xlabel('m')
pl.ylabel('$\chi^2$')
pl.title('Minimum of $\chi^2$ as a function of m for different values of b')
pl.show()

#Make a 2D plot of chi-squared vs m and b:
mm,bb=np.meshgrid(m_vals,b_vals)
chi2d=pl.zeros(mm.shape)
for i in range(m_vals.size):
    for j in range(b_vals.size):
        chi2d[i,j]=chis(ydata,sigma_y,m_vals[i]*x+b_vals[j])
pl.pcolormesh(mm,bb,chi2d,cmap='RdBu')
pl.axis([m_vals.min(),m_vals.max(),b_vals.min(),b_vals.max()])
pl.xlabel('m')
pl.ylabel('b')
pl.title("$\chi^2$ as a function of m and b")
pl.colorbar()
pl.show()
```

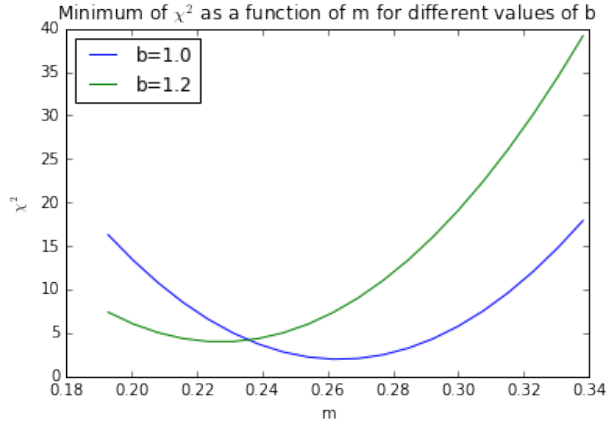

Figure 8.4: Chi-squared as a function of the slope from the fit in Figure 8.2 for different values of the offset. Since the minimum in chi-squared as a function of the slope, m, decreases for a higher value of the offset, b, we can see that the parameters from the fit are anti-correlated.

In order to provide the uncertainty in a quantity that depends on m and b, we need to know how correlated the two parameters are, since we cannot simply add the error in quadrature.

8.2 Non-linear fits

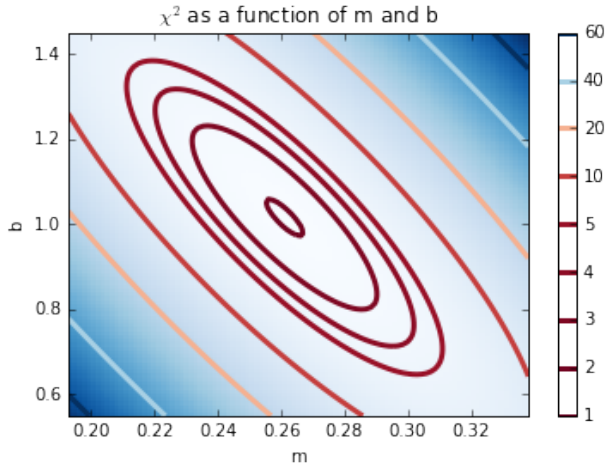

Figure 8.5: Chi-squared as a function of the slope and offset from the fit in Figure 8.2. The "tilt" in the chi-squared map is the result of the anti-correlation between the two parameters.