

## Statistical Tools and Terminology

### 2.1 Introduction

Having classified our materials as being stochastic, we require a family of mathematical tools to represent the distributions of their properties and some suitable numbers to describe these distributions. This chapter provides informally some background to these tools. A real number is called a ‘random variable’ if its value is governed by a well-defined statistical distribution. We begin by defining some general properties of random variables and many of the distributions that we will encounter in subsequent chapters and that we shall use to derive the properties of stochastic fibrous materials. As well as using standard mathematical notation, the use of *Mathematica* to handle statistical functions and generate random data is introduced.

### 2.2 Discrete and Continuous Random Variables

We have identified the difference between stochastic and deterministic processes as being essentially one of uncertainty. Often this uncertainty arises because we do not know enough about the factors that contribute to the state of the process or its outcome. Consider for example the rolling of a fair six-sided die. If we knew enough about the position, orientation in three-dimensions, and velocity of the die at some given point in time, as well as the relevant elastic moduli and coefficients of friction of the die and the surface onto which we are rolling, then we might develop appropriate equations of motion and solve these to compute the precise position at rest of the die and hence predict the number that will be rolled. This is a difficult problem to formulate, let alone solve even if all the equations and variables were known; typically we expect that at least the first three will be unknown. Accordingly, we have uncertainty in our system. In fact, even if we create a machine to roll the die identically for several throws, we expect that different outcomes will result because of the sensitivity to even small uncertainties in the variables. We are

unable therefore to deterministically predict the outcome of a roll and must always be uncertain of any individual event. Despite this uncertainty, we may be confident that the probability of rolling any number is  $\frac{1}{6}$ . Thus, whereas we cannot predict the outcome of an individual roll, we know what all the possible outcomes are and the probability of their occurrence. We can state then the random variable  $x$  which represents the outcome of the roll of a die can take the values 1, 2, 3, 4, 5 and 6 and each outcome has probability  $\frac{1}{6}$ . In the sequel, we shall see that this characterises the random variable  $x$  as being controlled by the discrete uniform probability distribution,  $P(x) = \frac{1}{6}$ .

We consider first the application of statistics to the description of systems where the events within that system or the outcomes of it are *discrete*. This means that each possible event or outcome has a definite probability of occurrence. We have just considered one such process, the rolling of an unbiased die. Another example of a discrete stochastic process is the tossing of a coin where the probability of the outcome being either heads or tails is  $\frac{1}{2}$ . If we assume that the probability of the die coming to rest on one of its edges is infinitesimal, then we may state that the probability of each event is  $\frac{1}{6}$ . Similarly, we know that it is not possible to throw the die and have the uppermost face show, for example,  $4\frac{1}{2}$  spots. So the outcome of rolling the die is a discrete random variable. Examples of discrete random variables that characterise the structure of fibre networks are the number of fibre centres per unit volume or area in the network, or the number of fibres making contact with any given fibre in the structure. As a rule, we can expect to encounter discrete random variables when the feature of interest, experimental conditions permitting, may be counted; the exception to this being where only certain classes of events exist, for example, where a fibre network is formed from a blend of fibres manufactured with precisely known lengths which are known because they have been measured and not because they have been counted.

Consider now the distribution of the weights of eggs produced by free-range hens. The probability that an egg weighs precisely 60 g is very small; as is the probability that it weighs precisely 59.9 g or 60.000001 g. It is much easier, and certainly more meaningful, to state the probability that eggs from these hens weigh between say 55 and 65 g or between 45 and 55 g, *etc.* Clearly, the weights of the eggs differ from the rolling of a die in that we do not have discrete outcomes; the weight of an egg is therefore classified as a *continuous* random variable. Examples of continuous random variables encountered in the description of fibre networks are the area or volume of inter-fibre voids and the lengths of the fibrous ligaments that exist between fibre crossings.

### 2.2.1 Characterising Statistics

Given sample data from a system, *e.g.* the outcomes,  $x_i$  of  $n$  rolls of a die or the weights of  $n$  eggs, we may use statistics to characterise the population. The most common statistics to characterise the distribution are familiar to

most of us through the handling of experimental data. We define them here for completeness.

**Mean:** The mean value of the sample data is given by the sum of all the data divided by the number of observations. For data  $x_1, x_2 \dots x_n$  we denote the mean  $\bar{x}$  and this is given by

$$\bar{x} = \sum_{i=1}^n \frac{x_i}{n} . \quad (2.1)$$

The mean is often termed the *expectation* or, in every-day language, the *average*.

**Mode:** The mode is the value within our sample data that occurs with the greatest frequency. For discrete data, this is found by inspection; for continuous data the mode is estimated from a histogram of the data as the mid point of the tallest column.

**Median:** The median is occasionally used instead of the mean for the characterisation of data that has a histogram that is not symmetric about the mean; such data is described as *skewed*. The median is found by sorting the data by magnitude and selecting the middle observation such that half the observations are numerically greater than the median and half are numerically smaller.

**Variance:** The variance of our data is the mean square difference from the mean, *i.e.* it is the expected value of  $(x_i - \bar{x})^2$ . It is denoted  $\sigma^2(x)$  and given by

$$\sigma^2(x) = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n} . \quad (2.2)$$

For small samples of data, Equation 2.2 will underestimate the variance because, for a sample of size  $n$ , each observation can be independently compared with only  $(n - 1)$  other observations, biasing the calculation of the variance. Accordingly, the unbiased estimate of the variance is given by

$$\sigma^2(x) = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n - 1} , \quad (2.3)$$

and this is typically applied for samples with  $n$  less than about 20.

**Standard Deviation:** The standard deviation is the square root of the variance and it is denoted  $\sigma(x)$ . It is often preferred to the variance as it has the same units as the original data.

$$\sigma(x) = \sqrt{\sigma^2(x)} = \sqrt{\sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n}} \quad (2.4)$$

The unbiased estimate is given by

$$\sigma(x) = \sqrt{\sigma^2(x)} = \sqrt{\sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n-1}}. \quad (2.5)$$

**Coefficient of Variation:** The coefficient of variation is the standard deviation relative to the mean. We denote it  $CV(x)$  and it is given by

$$CV(x) = \frac{\sigma(x)}{\bar{x}}. \quad (2.6)$$

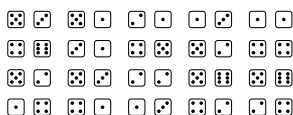
Note that the coefficient of variation is dimensionless and is often reported as a percentage.

We classify the mean, mode and median as measures of location; they provide a measure of the magnitude of the numbers we can expect to characterise our distribution. The variance, standard deviation and coefficient of variation are classified as measures of spread; they provide a measure of how widely distributed the data are in our sample or population and thus can be used to inform how representative our measures of location are of the distribution as a whole.

## Using Characterising Statistics

We illustrate the calculation of these characterising statistics with *Mathematica* by generating a sample of data representing rolls of a pair of unbiased dice using the command **RandomInteger**. This function generates pseudorandom integers with equal probability, so the command **RandomInteger[]** will give an output of either 0 or 1 with the probability of each outcome being  $\frac{1}{2}$ . To represent the roll of a fair six-sided die we use **RandomInteger[1, 6]**.

Consider first the outcomes of rolling a pair of unbiased dice 20 times. The outcomes of the experiment are recorded in the following graphic:



In fact, these dice rolls were simulated in *Mathematica* using **RandomInteger** with the following input:

```
SeedRandom[1]
pairs = RandomInteger[{1, 6}, {20, 2}]
```

which gives the output in list form which corresponds to our graphic:

```
Out[2]= {{5, 3}, {5, 1}, {2, 1}, {1, 3}, {1, 1}, {4, 6}, {3, 1},
         {4, 5}, {5, 2}, {4, 4}, {5, 2}, {5, 3}, {2, 2}, {5, 6},
         {5, 6}, {1, 4}, {4, 1}, {1, 3}, {4, 2}, {2, 4}}
```

Note the use of the command **SeedRandom**. By including this line, *Mathematica* uses the same random seed for each evaluation and we obtain the same value for **pairs** each time we evaluate the code. Each pair of numbers is identified in *Mathematica* by its location in the list, so we can refer to these using the command **Part** or the assignment **[[ ]]**, e.g. ,

```
In[3]:= Part[pairs, 4]
        pairs[[8]]
```

```
Out[3]= {1, 3}
```

```
Out[4]= {4, 5}
```

The values obtained by summing the numbers shown on each pair of dice represent the random variable of interest. For the  $i$ th pair of random numbers, we obtain their sum using **Total[pairs[[i]]]**, and we use the command **Table** to carry this out for all  $i$ :

```
In[5]:= rolls = Table[Total[pairs[[i]]], {i, 1, 20}]
Out[5]= {8, 6, 3, 4, 2, 10, 4, 9, 7, 8, 7, 8, 4, 11, 11, 5, 5, 4, 6, 6}
```

To compute the mean of our dice rolls we need to apply Equation 2.1 and compute the sum of all observations and divide this by the number of observations. *Mathematica* has a built-in command **Mean** to carry out this calculation:

```
In[6]:= Mean[rolls]
Out[6]=  $\frac{32}{5}$ 
```

The result is displayed as an improper fraction, because *Mathematica* has carried out computations on random integers. To convert to the corresponding numerical value, we use **N**:

```
In[7]:= N[%]
Out[7]= 6.4
```

where the symbol **%** refers to the last output. To compute the variance we require the mean square difference from the mean, as given by Equation 2.3. We might compute this explicitly using,

```
In[8]:= Total[(rolls - Mean[rolls]) ^ 2] / 19
      N[%]
Out[8]= 644
      95
Out[9]= 6.77895
```

though again, *Mathematica* has the specific command **Variance** to handle this for us:

```
In[10]:= Variance[rolls]
Out[10]= 644
      95
```

Inevitably, the standard deviation is given by,

```
In[11]:= StandardDeviation[rolls]
      N[%]
Out[11]= 2  $\sqrt{\frac{161}{95}}$ 
Out[12]= 2.60364
```

and is the square root of the variance:

```
In[13]:= TrueQ[StandardDeviation[rolls] ==  $\sqrt{\text{Variance[rolls]}}$ ]
Out[13]= True
```

Importantly in Version 6, *Mathematica* always uses Equations 2.3 and 2.5 to calculate the variance and standard deviation when handling lists. Note that to generate the square root operator in *Mathematica* we use **Ctrl**+2, though we could obtain the square root of the variance using any of the following:

```

Sqrt[Variance[rolls]]
Variance[rolls] ^ (1 / 2)
Variance[rolls]1/2
Power[Variance[rolls], 1 / 2]

```

where in the third example, the superscript is generated using  $\boxed{\text{Ctrl}}+6$ .

For completeness, we calculate the remaining measures of location and spread for our data, as given earlier in this section:

```

In[14]:= Median[rolls]
Commonest[rolls] (* Commonest = mode *)
CVrolls = StandardDeviation[rolls] / Mean[rolls]
N[CVrolls]

Out[14]= 6

Out[15]= {4}

Out[16]=  $\frac{\sqrt{\frac{805}{19}}}{16}$ 

Out[17]= 0.406819

```

The use of the command **Median** is an intuitive choice, but we note that the command **Mode** is used in *Mathematica* in conjunction with commands associated with equation solving and other operations; thus we compute the mode using the command **Commonest**. Note that the output of this command is a list enclosed in braces, { }, in our case this list has length 1, though this need not be the case. Note also the use of the comment enclosed between starred brackets, (\* \*); anything between these characters is not evaluated.

If we change the first line of our code to **SeedRandom**[2] we obtain a different set of observations:

```

In[18]:= SeedRandom[2]
pairs = RandomInteger[{1, 6}, {20, 2}]
rolls = Table[Total[pairs[[i]]], {i, 1, 20}];

Out[19]= {{6, 2}, {3, 3}, {6, 3}, {2, 6}, {6, 1}, {1, 5}, {4, 5},
           {1, 2}, {2, 6}, {2, 6}, {5, 5}, {1, 1}, {5, 5}, {2, 3},
           {4, 4}, {1, 2}, {1, 5}, {3, 3}, {2, 6}, {5, 4}}

```

and the output of **rolls** has been suppressed by ending this line of code with a semi-colon. Calculating the mean, variance and standard deviation as before we have,

```

In[21]:= N[Mean[rolls]]
          N[Variance[rolls]]
          N[StandardDeviation[rolls]]

Out[21]= 6.95

Out[22]= 5.31316

Out[23]= 2.30503

```

On first inspection, it is clear that the calculated mean, variance and standard deviation for our two sets of simulated dice rolls are different. This arises because we have only a limited set of data available to characterise the distribution, *i.e.* we are considering the statistics of two *samples* that we hope are representative of the *population* from which they are drawn. Using different values of **SeedRandom** we have generated independent samples from the population of dice rolls where the probabilities of a given number being shown on the face of each dice are equal. Of course, we might pool the results of our two samples to provide a better estimate of the statistics that characterise the distribution:

```

In[24]:= SeedRandom[1]
          pairs = RandomInteger[{1, 6}, {20, 2}];
          rolls1 = Table[Total[pairs[[i]]], {i, 1, 20}];
          SeedRandom[2]
          pairs = RandomInteger[{1, 6}, {20, 2}];
          rolls2 = Table[Total[pairs[[i]]], {i, 1, 20}];
          pooledrolls = Join[rolls1, rolls2];

```

Note here that the name **pairs** is used twice, so values arising from the first evaluation are overwritten in the *Mathematica* kernel by those from the second evaluation. The command **Join** concatenates the specified lists. The characterising statistics for the pooled data are given in the usual way:

```

In[31]:= Length[pooledrolls]
          N[Mean[pooledrolls]]
          N[Variance[pooledrolls]]
          N[StandardDeviation[pooledrolls]]

Out[31]= 40

Out[32]= 6.675

Out[33]= 5.96859

Out[34]= 2.44307

```

and we observe that our new estimate of the mean is precisely the mean of our two estimates from the independent samples. The estimates of the variance, and hence the standard deviation, lie between those of the two samples, but are not the mean of these estimates as they are calculated on the basis of the new estimate of the mean and a larger sample with  $n = 40$ .

*Mathematica* can handle very large lists very comfortably, so we get a much improved estimate of the characterising statistics using larger  $n$ :

```
In[35]:= SeedRandom[1]
n = 1 000 000;
pairs = RandomInteger[{1, 6}, {n, 2}];
rolls =
  Table[Total[pairs[[i]]], {i, 1, Length[pairs]}];
Mean[N[rolls]]
StandardDeviation[N[rolls]]

Out[39]= 7.00089

Out[40]= 2.41501
```

Note the placing of the command **N** such that the calculations are performed on numerical rather than integer values of the random variable. This speeds up the calculations as illustrated by use of the command **Timing**, which gives the output as a list where the first term is the time taken in seconds for *Mathematica* to perform the calculation:

```
In[41]:= StandardDeviation[rolls] // Timing
N[StandardDeviation[rolls]] // Timing
StandardDeviation[N[rolls]] // Timing

Out[41]= {7.1,  $\frac{3 \sqrt{\frac{375016153}{37037}}}{125}$ }

Out[42]= {8.142, 2.41501}

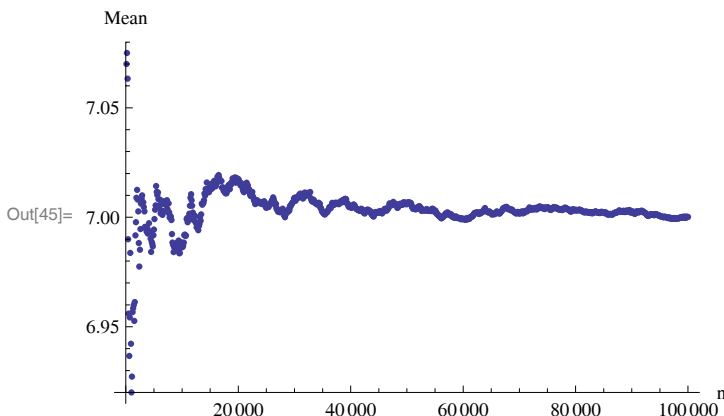
Out[43]= {0.09, 2.41501}
```

So for this example, the calculation of the standard deviation is almost 80 times faster when performed numerically.

Using our list of length 1 million, we can track the dependence of our calculation of the mean and standard deviation on the size of our sample. To do this, we compute the mean and standard deviation for samples of increasing length,  $n$  using the command **Take** to extract elements from the list and the command **Table** to do this for different  $n$ . In the example that follows we compute the mean for samples of length between 100 and 100,000 in steps of 100. The output of **meanrollsn** is a list of sublists, each of length 2,

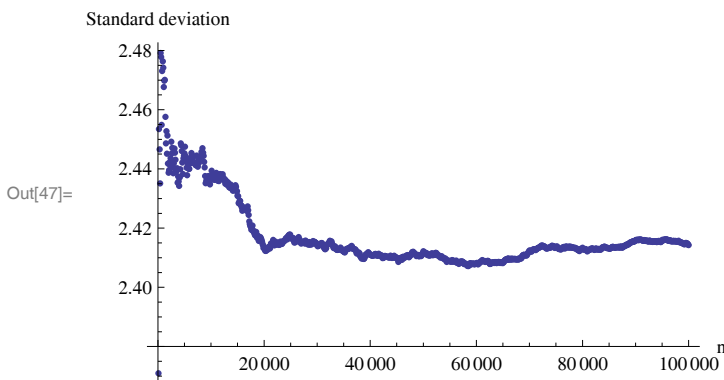
where the first element is the size of the sample,  $n$  and the second element is the mean of that sample. Using **ListPlot** we are able to visualise the quality of our estimate of the mean as we increase the sample size.

```
In[44]:= meanrollsn = Table[{n, Mean[N[Take[rolls, n]]]},
    {n, 100, 100 000, 100}];
ListPlot[meanrollsn, PlotRange → All,
    AxesLabel → {"n", "Mean"}]
```



We use similar code to calculate the standard deviation for different  $n$ :

```
In[46]:= stdrollsn =
    Table[{n, StandardDeviation[N[Take[rolls, n]]]},
    {n, 100, 100 000, 100}];
ListPlot[stdrollsn, PlotRange → All,
    AxesLabel → {"n", "Standard deviation"}]
```



From inspection of the graphical outputs generated using **ListPlot** we can be reasonably confident that a sample size of some tens of thousands will

provide us with a reasonable estimate of the characterising statistics for our distribution. When dealing with a sample of size 1 million, we might consider that the statistics of our sample approach those of the population. As yet, though, we do not know precisely the characterising statistics for the population from which our samples are drawn. Referring back to our simulation of 1 million rolls, we might reasonably assume that the mean of the population is 7 and the standard deviation is about 2.42. Note that if we used **SeedRandom[2]** to simulate a million rolls of a pair of dice, our estimate of the mean would change in the 4th decimal place, whereas that of the standard deviation would differ in the third. We will now consider how we can use probability theory to obtain robust measures of location and spread for statistical populations.

## Theoretical Determination of Characterising Statistics

Numerical approaches of the type used so far are often referred to as Monte Carlo methods and are very useful when theoretical approaches do not lend themselves to closed form solutions. Very often however, statistical theory does allow us to make precise statements about the properties of distributions. We consider first theory describing the problem of rolling a single die and proceed to consider the case of rolling a pair of dice, which we have just considered.

Consider first the rolling of a fair six-sided die. The only possible outcomes are the integers 1 to 6 and each outcome has probability  $\frac{1}{6}$ . Since the family of possible outcomes is limited to these values, we have a discrete random variable and, since all outcomes have the same probability, our random variable has a *discrete uniform distribution*. For random integers  $x$  with  $x_{\min} \leq x \leq x_{\max}$  the probability of a given  $x_i$  is given by

$$P(x) = \begin{cases} 0 & \text{if } x < x_{\min} \\ \frac{1}{1+x_{\max}-x_{\min}} & \text{if } x_{\min} \leq x \leq x_{\max} \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

In *Mathematica* the discrete uniform distribution is input as

```
In[1]:= DiscreteUniformDistribution[{xmin, xmax}]

Out[1]= DiscreteUniformDistribution[{xmin, xmax}]
```

and the probability function is input using

```
In[2]:= PDF[DiscreteUniformDistribution[{xmin, xmax}], x]

Out[2]= 
$$\frac{1}{1 + \text{xmax} - \text{xmin}}$$

```

which corresponds to the second interval of the piecewise function given by Equation 2.7. Note that *Mathematica* is aware of the definition of the distribution for arbitrary  $x$ :

```
In[3]:= PDF[DiscreteUniformDistribution[{1, 6}], x]
Table[PDF[DiscreteUniformDistribution[{1, 6}], x],
      {x, 0, 8}]
PDF[DiscreteUniformDistribution[{1, 6}], 2.2]

Out[3]=  $\frac{1}{6}$ 

Out[4]=  $\left\{0, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, 0, 0\right\}$ 

Out[5]= 0
```

Of course, *Mathematica*'s functions are well defined and have been fully tested. However, when deriving our own probability functions later, we will frequently check that we have accounted for all possible outcomes by ensuring that the probability function sums to 1:

```
In[6]:= Sum[PDF[DiscreteUniformDistribution[{xmin, xmax}], x],
          {x, xmin, xmax}]

Out[6]= 1
```

Having reassured ourselves of this, we can compute the mean using

$$\bar{x} = \sum_{x=x_{\min}}^{x_{\max}} x P(x) \quad (2.8)$$

Note that whereas when handling data, the mean was calculated as the sum of all observations divided by the number of observations, here we compute the product of the value of the observation  $x_i$  and its frequency of occurrence and sum the result for all possible  $x$ . We input this as:

```
In[7]:= xbar = Sum[
          x PDF[DiscreteUniformDistribution[{xmin, xmax}], x],
          {x, xmin, xmax}]

Out[7]=  $\frac{x_{\max} + x_{\min}}{2}$ 
```

Similarly, to compute the variance we require,

$$\bar{x} = \sum_{x=x_{\min}}^{x_{\max}} (x - \bar{x})^2 P(x) \quad (2.9)$$

which we compute using

```
In[8]:= Sum[(x - xbar)^2
           PDF[DiscreteUniformDistribution[{xmin, xmax}], x],
           {x, xmin, xmax}]

Out[8]= 1/(12) (xmax - xmin) (2 + xmax - xmin)
```

For distributions that are predefined in *Mathematica* we can compute these statistics directly, though the output of the command **Variance** requires some manipulation to yield the same form as given by the summing method:

```
In[9]:= Mean[DiscreteUniformDistribution[{xmin, xmax}]]
Variance[DiscreteUniformDistribution[{xmin, xmax}]]
Factor[%]

Out[9]= (xmax + xmin)/2

Out[10]= 1/12 (-1 + (1 + xmax - xmin)^2)

Out[11]= 1/12 (xmax - xmin) (2 + xmax - xmin)
```

In the case of our six-sided die, we have  $x_{\min} = 1$  and  $x_{\max} = 6$  and the mean and variance are given by


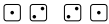









```
In[12]:= Mean[DiscreteUniformDistribution[{1, 6}]]
Variance[DiscreteUniformDistribution[{1, 6}]]

Out[12]= 7/2

Out[13]= 35/12
```

and we observe that the mean, or the expected value, is not a possible outcome. This is an important property of discrete random variables and will shall encounter it in other contexts as we develop theory describing the structure of stochastic fibrous materials.

We return now to the two-dice problem that we considered numerically earlier. The possible outcomes and their probabilities are summarised in Table 2.1. It is immediately clear that the distribution of outcomes is symmetrical about 7, which is the mode of our distribution. We require a probability function that describes our random variable and by inspection we note that

Face value		Permutations	Probability
2		1	$\frac{1}{36}$
3		2	$\frac{2}{36} = \frac{1}{18}$
4		3	$\frac{3}{36} = \frac{1}{12}$
5		4	$\frac{4}{36} = \frac{1}{9}$
6		5	$\frac{5}{36}$
7		6	$\frac{6}{36} = \frac{1}{6}$
8		5	$\frac{5}{36}$
9		4	$\frac{4}{36} = \frac{1}{9}$
10		3	$\frac{3}{36} = \frac{1}{12}$
11		2	$\frac{2}{36} = \frac{1}{18}$
12		1	$\frac{1}{36}$

**Table 2.1.** Permutations and probabilities for outcomes of rolling a pair of unbiased six-sided dice

the probabilities on the right of our graphic are given by<sup>1</sup>

$$P(x) = \begin{cases} \frac{6-|7-x|}{36} & \text{if } 2 \leq x \leq 12 \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

To input this to *Mathematica* we introduce two new commands. Firstly, instead of assigning a variable name to the function, we use the function **SetDelayed** which we input as **:=** such that the right-hand side of our input is not evaluated until called. We also use the command **Piecewise** to assign probability zero for all  $x$  outside the applicable range of our function.

```
In[14]:= P[x_] :=  
Piecewise[{{(6 - Abs[7 - x]) / 36, 2 ≤ x ≤ 12}}, 0]
```

We should check that our probability function yields the required probabilities:

```
In[15]:= Table[P[x], {x, 0, 14}]  
  
Out[15]= {0, 0,  $\frac{1}{36}$ ,  $\frac{1}{18}$ ,  $\frac{1}{12}$ ,  $\frac{1}{9}$ ,  $\frac{5}{36}$ ,  $\frac{1}{6}$ ,  $\frac{5}{36}$ ,  $\frac{1}{9}$ ,  $\frac{1}{12}$ ,  $\frac{1}{18}$ ,  $\frac{1}{36}$ , 0, 0}
```

and check also that we have considered all probabilities:

<sup>1</sup> In the general case, the random variable  $Y = X_1 + X_2$  with  $1 \leq X_1, X_2 \leq X_{\max}$  where  $X_1$  and  $X_2$  are independent discrete random variables taking integer values, has probability function,

$$P(Y) = \frac{X_{\max} - |X_{\max} + 1 - Y|}{X_{\max}^2} .$$

```
In[16]:= Sum[P[x], {x, 2, 12}]
```

```
Out[16]= 1
```

The mean, variance and standard deviation are given by

```
In[17]:= xbar = Sum[x P[x], {x, 2, 12}]
```

```
xvar = Sum[(x - xbar)^2 P[x], {x, 2, 12}]
```

```
xstd = Sqrt[xvar]
```

```
N[%]
```

```
Out[17]= 7
```

```
Out[18]=  $\frac{35}{6}$ 
```

```
Out[19]=  $\sqrt{\frac{35}{6}}$ 
```

```
Out[20]= 2.41523
```

By using the probability function for the outcomes of rolling a pair of unbiased-six sided dice, we are able to make precise statements about the characterising statistics of our distribution. The expected outcome, *i.e.* the mean, is 7; since this outcome has the highest probability and the distribution is symmetrical about the mean, the mode and median are 7 also. The standard deviation of the distribution is  $\sqrt{35/6}$ . We observe that these theoretical measures agree rather closely with those obtained for a million dice rolls.

## 2.3 Common Probability Functions

In the last section we encountered the discrete uniform distribution and identified the *Mathematica* commands to call this distribution and to generate its probability function, its mean, and its variance. The discrete uniform distribution is one of the simplest distributions we are likely to encounter; we have a finite number of permissible outcomes in an interval, and these have equal probability. Before considering continuous random variables, where the number of outcomes in an interval is infinite, we introduce some more probability functions that characterise the distributions of discrete random variables and which we shall use extensively in modelling the structure of fibrous materials.

### 2.3.1 Bernoulli Distribution

The Bernoulli distribution is used to characterise random processes where there are only two possible outcomes. The classical example of such a process is

the tossing of a coin where the outcomes ‘heads’ or ‘tails’ each have probability  $\frac{1}{2}$ , though other examples include observations by researchers of whether cars travelling in rush-hour are occupied by the driver only or by the driver and passengers or whether a random point within a block of sandstone lies with a void or in the solid phase of the material. In this latter case, the probability that the point lies in a void is its porosity,  $\epsilon$  and the probability that the point lies within the solid is  $(1 - \epsilon)$ .

By convention, we denote the outcomes 0 and 1 and often these are taken to classify the outcomes as ‘failure’ and ‘success’, respectively. If the probability of success is  $0 \leq p \leq 1$ , then the probabilities of success and failure are given by

$$P(0) = 1 - p \quad (2.11)$$

$$P(1) = p \quad , \quad (2.12)$$

which can be written as,

$$P(n) = p^n (1 - p)^{1-n} \quad . \quad (2.13)$$

We call the Bernoulli distribution in *Mathematica* using

**BernoulliDistribution[p]**

To obtain the probability function we use

```
In[1]:= PDF[BernoulliDistribution[p], n]

Out[1]= { 1 - p   n == 0
         { p       n == 1
```

which corresponds to Equation 2.13, but it is expressed in piecewise form. Note that the piecewise function given in the output uses the notation ‘==’ for ‘equals’; the single equals sign, ‘=’, as used so far, allows us to set a value to the variable name preceding it.

The mean and variance of the Bernoulli distribution are given by

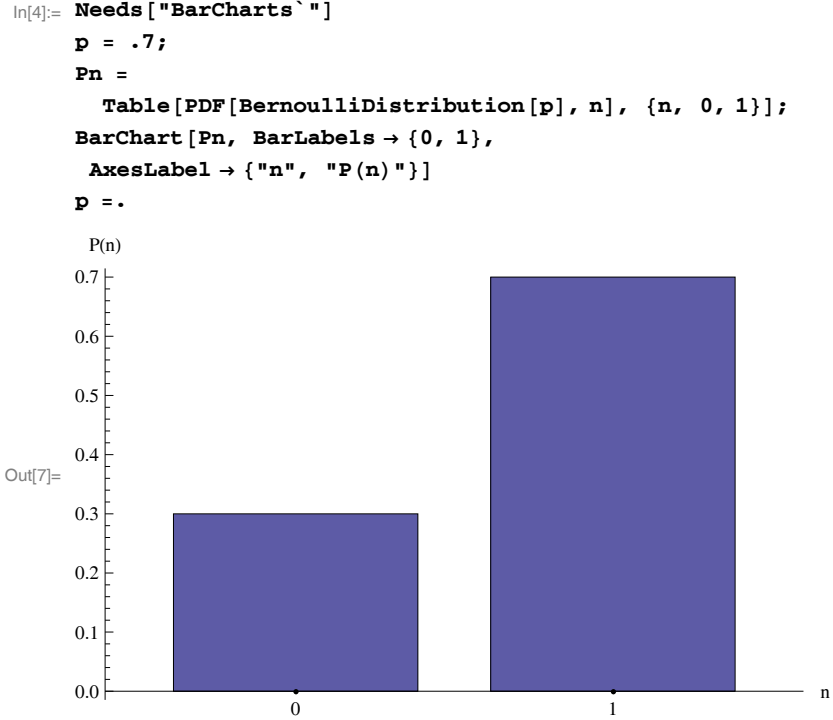
```
In[2]:= Mean[BernoulliDistribution[p]]
        Variance[BernoulliDistribution[p]]

Out[2]= p

Out[3]= (1 - p) p
```

The appropriate graphical representation of a discrete probability function is a bar chart, which we generate using the command **BarChart**. This command is not loaded by default when the *Mathematica* kernel is launched, so we must call the required package using the command **Needs**. The following code calls

the package **BarCharts** and generates a list of Bernoulli probabilities for the case where the probability of success  $p = 0.7$ . This list is then plotted as a bar chart with appropriate axis labels. Note that the last line unsets the value of parameter **p**.



### 2.3.2 Binomial Distribution

Consider now an extension to the examples we considered when introducing the Bernoulli distribution. If we toss a coin  $m$  times, observe  $m$  cars to see if they are carrying passengers, or select  $m$  points at random from within the volume of a block of sandstone to identify if they are in the solid or void phase, we may be interested in how many of these  $m$  *Bernoulli trials* have a particular outcome, *i.e.* how many result in ‘success’ or ‘failure’. The distribution describing this discrete random variable is the binomial distribution. Denoting the number of successes  $0 \leq x \leq m$  for Bernoulli trials with probability of success  $p$ , it has probability function,

$$P(x) = \binom{m}{x} (1-p)^{m-x} p^x \quad (2.14)$$

where  $\binom{m}{x}$  is the binomial coefficient,

$$\binom{m}{x} = \frac{m!}{x!(m-x)!} \quad (2.15)$$

and it is invoked using **Binomial**[**m**,**x**] in *Mathematica*.

To obtain the binomial probability function in *Mathematica* we use

```
In[9]:= PDF[BinomialDistribution[m, p], x]
Out[9]= (1 - p)^(m - x) p^x Binomial[m, x]
```

The mean and variance of the binomial distribution are given by

```
In[10]:= Mean[BinomialDistribution[m, p]]
          Variance[BinomialDistribution[m, p]]
Out[10]= m p
Out[11]= m (1 - p) p
```

To plot the probability function, we again use **BarChart**. To aid investigation of the influence of parameters  $p$  and  $m$  on the distribution, we define a function **bar**[**p**\_, **m**\_] using **SetDelayed** (**:=**).

```
In[12]:= bar[m_, p_] := BarChart[Table[
          PDF[BinomialDistribution[m, p], x], {x, 0, m}],
          BarLabels -> Range[0, m], AxesLabel -> {"x", "P(x)"}]
```

Note that we have nested several *Mathematica* commands, neatening the code; note also the command **Range** which is used here to generate a list representing the labels on the abscissa. This is required because by default **BarChart** labels the first bar, '1', the second '2', *etc.*, yet for our data, the first bar represents the probability of outcome zero, the second the probability of outcome 1, *etc.*

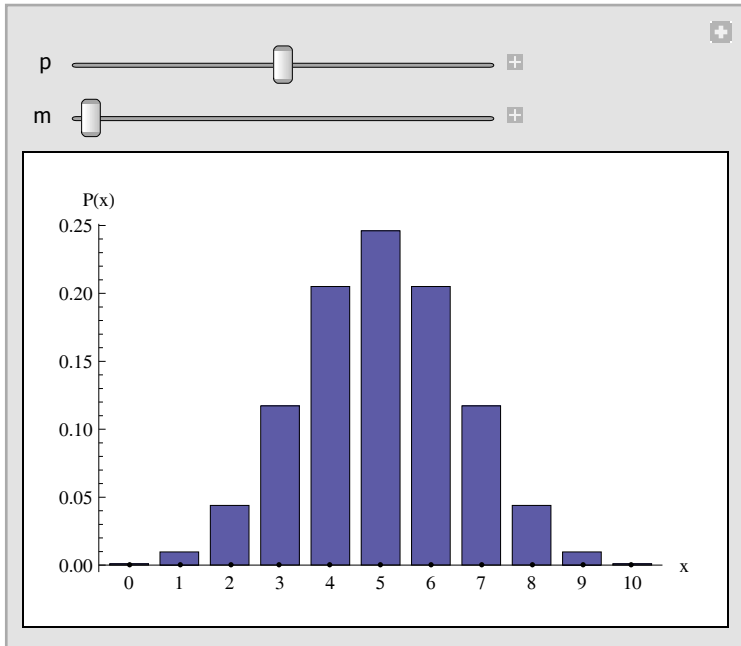
Now, we could investigate the influence of parameter  $p$ , representing the probability of success in a trial, by evaluating, for example,

```
bar[10, .2]
bar[20, .5]
```

*etc.* and generate several bar charts for comparison. As we have mentioned earlier however, *Mathematica* allows us to alter variables interactively using the command **Manipulate**. We generate a bar chart of the probability function for the binomial distribution with two associated sliders allowing us to vary  $p$  and  $m$  using,

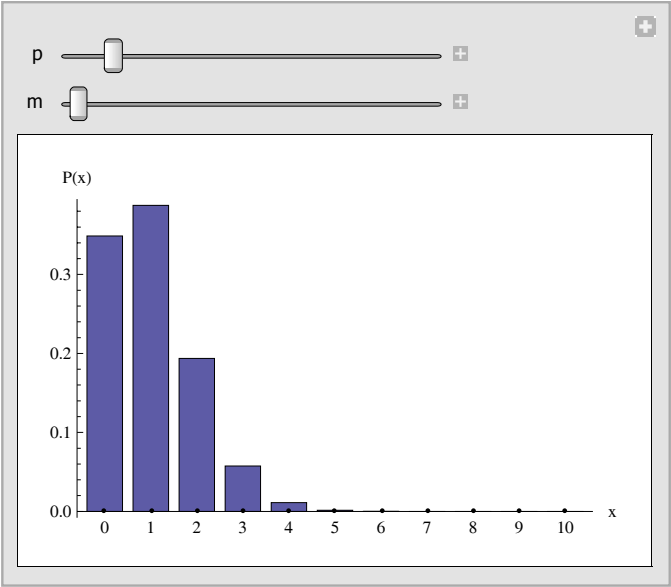
```
In[14]:= Manipulate[bar[m, p], {{p, .5}, 0, 1}, {m, 10, 100, 5}]
```

Out[14]=

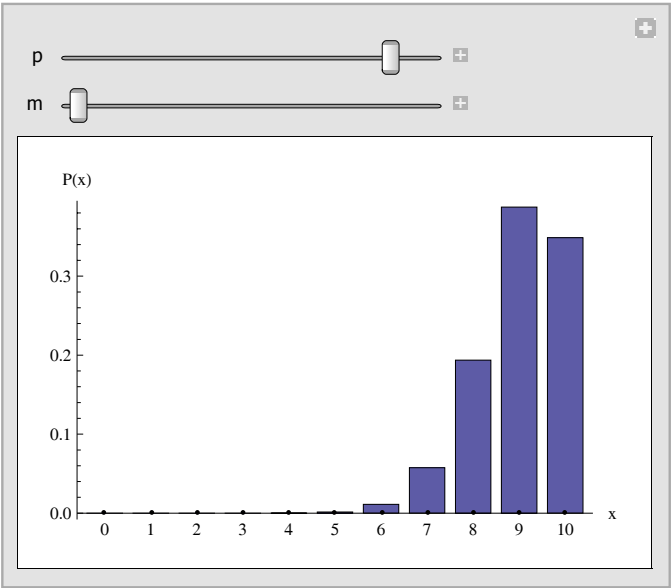


Moving the slider for parameter  $m$  increases the number of bars in our chart, but seemingly has a limited influence on its shape. When moving the slider for parameter  $p$  however, we observe a significant change in the shape of the distribution, moving from a positive skew as  $p$  approaches zero, through apparent symmetry around  $p = 1/2$ , to a negative skew as  $p$  approaches 1. This behaviour is easily understood. When  $p$  is close to 1, successful trials are far more likely than unsuccessful trials and *vice versa* when  $p$  is close to zero. Accordingly, we expect that  $P(x)$  will exhibit a maximum close to  $mp$ .

Out[14]=



Out[14]=



We quantify the asymmetry of the distribution by its *skewness*.

$$Sk(x) = \frac{\sum (x_i - \bar{x})^3 P(x)}{\sigma^3(x)} \quad (2.16)$$

such that for  $Sk(x) > 0$  the distribution exhibits a longer tail to the right than to the left and *vice versa* for  $Sk(x) < 0$ ; when  $Sk = 0$  the distribution is symmetrical about the mean. For our binomial distribution, we have

$$\begin{aligned} \text{In[15]} &:= \text{Skewness[BinomialDistribution[m, p]]} \\ \text{Out[15]} &= \frac{1 - 2p}{\sqrt{m(1-p)p}} \end{aligned}$$

such that the distribution exhibits symmetry when  $p = 1/2$ , has a positive skew when  $p < 1/2$  and a negative skew when  $p > 1/2$ . We observe also that the influence of  $p$  on the magnitude of skewness, and hence on the length of the tails of the distribution, diminishes as  $m$  increases. We observe behaviour consistent with this if we return to the bar chart that we generated with **Manipulate** and move the sliders to vary  $p$  and  $m$ .

### 2.3.3 Poisson Distribution

The Poisson distribution has probability function

$$P(x) = \frac{e^{-\bar{x}} \bar{x}^x}{x!} \quad \text{for } x = 0, 1, 2, 3, \dots \quad (2.17)$$

where  $\bar{x}$  is the mean of the random variable  $x$ . It arises as a limiting case of the binomial distribution when the number of independent trials  $m$  is large and the probability of success  $p$  is small such that  $mp = \bar{x}$ .

The binomial coefficient, and hence the probability distribution function for the binomial distribution can be expressed in terms of the Euler gamma function, **Gamma**:

$$\begin{aligned} \text{In[16]} &:= \text{FunctionExpand[PDF[BinomialDistribution[m, p], x]]} \\ \text{Out[16]} &= \frac{(1-p)^{m-x} p^x \text{Gamma}[1+m]}{\text{Gamma}[1+m-x] \text{Gamma}[1+x]} \end{aligned}$$

The Euler gamma function is an example of a ‘special function’ that we shall encounter in several contexts as we develop our models. It satisfies,

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt. \quad (2.18)$$

The arguments to **Gamma** can take any value, but in the special case of integer arguments, the gamma function returns the factorial, such that

$$\Gamma(z+1) = z! \quad \text{for } z = 0, 1, 2, 3 \dots$$

The *Mathematica* command **FullSimplify** is able to reduce many expressions containing special functions to simpler forms, particularly if the second argument provides some assumptions, *e.g.*

```
In[17]:= FullSimplify[%, {x, m} ∈ Integers && 0 ≤ x < m]
```

$$\text{Out[17]} = \frac{(1-p)^{m-x} p^x m!}{(m-x)! x!}$$

Note that in the above example, we use the symbol  $\in$ , input as Esc **e1** Esc, to specify that the variables  $x$  and  $m$  are elements of the domain **Integers**; the inequality  $\leq$  is input using '**<=**'.

We have noted that the Poisson distribution arises from the binomial distribution when  $mp = \bar{x}$  and for large  $m$ , so we substitute  $\bar{x}/m$  for  $p$  and take the limit as  $m \rightarrow \infty$ . To perform the substitution, we use the command **ReplaceAll**, which we input as '**/.**' To input the arrow for the limit, we use '**->**', so we do,

```
In[18]:= % /. p -> (xbar / m)
Limit[%, m -> ∞]
```

$$\text{Out[18]} = \frac{\left(\frac{xbar}{m}\right)^x \left(1 - \frac{xbar}{m}\right)^{m-x} m!}{(m-x)! x!}$$

$$\text{Out[19]} = \frac{e^{-xbar} xbar^x}{x!}$$

This last expression is the probability function for the Poisson distribution as given by Equation 2.17 and we note that the probability of observing a given integer  $x$  depends only on the expected value  $\bar{x}$ . We can call this probability function directly using

```
In[20]:= PDF[PoissonDistribution[xbar], x]
```

$$\text{Out[20]} = \frac{e^{-xbar} xbar^x}{x!}$$

and we note that the mean and variance of the Poisson distribution are equal:

```
In[21]:= Mean[PoissonDistribution[xbar]]
          Variance[PoissonDistribution[xbar]]

Out[21]= xbar

Out[22]= xbar
```

It follows that the coefficient of variation is  $1/\sqrt{x}$ .

Many physical phenomena are described rather well by the Poisson distribution [53] and it is often considered to be the standard model for random processes. We shall use the distribution several times in the models we derive in the following chapters and will consider it to model pure random processes. Thus, if we partitioned the random networks shown in Figure 1.3 into say  $10 \times 10$  square regions, we expect the number of fibre centres occurring within these regions to be distributed according to the Poisson distribution and so to have variance equal to the mean. The expected number of fibre centres in such regions will be the same for the disperse and clumped networks, but the variance of the number of fibre centres within regions would be less than the mean for the disperse cases, and greater than the mean in the clumped cases.

## 2.4 Common Probability Density Functions

So far, we have considered some of the more common statistical distributions that may be used to characterise discrete random variables. The functions that we have studied give the probability of a given outcome, say  $x$ , such that the probability  $0 \leq P(x) \leq 1$ . In Section 2.2 we observed that many random variables are not discrete, but are continuous. A property of a continuous random variable is that the probability of it having a given value  $x$  is infinitesimal, though the probability that  $x$  lies in a given interval is finite and lies between 0 and 1. The mathematical functions used to describe distributions of continuous variables are called probability *density* functions, whereas those describing the distributions of discrete random variables are called probability functions, or probability *distribution* functions. If we denote the probability density function of a continuous random variable  $x$ ,  $f(x)$ , then the probability that  $x$  lies in the range  $x_1 \leq x < x_2$  is

$$P(x_1 \leq x < x_2) = \int_{x_1}^{x_2} f(x) dx . \quad (2.19)$$

If  $x$  is defined in the domain  $x_{\min} \leq x < x_{\max}$ , then

$$\int_{x_{\min}}^{x_{\max}} f(x) dx = 1 . \quad (2.20)$$

The probability that  $x < X$  is called the cumulative distribution function. It is given by

$$F(X) = \int_{x_{\min}}^X f(x) \, dx . \quad (2.21)$$

The mean is given by

$$\bar{x} = \int_{x_{\min}}^{x_{\max}} x f(x) \, dx , \quad (2.22)$$

and the variance is given by

$$\sigma^2(x) = \int_{x_{\min}}^{x_{\max}} (x - \bar{x})^2 f(x) \, dx . \quad (2.23)$$

It is instructive to compare Equations 2.22 and 2.23 with Equations 2.8 and 2.9, respectively. We proceed by considering some common probability density functions encountered frequently in subsequent chapters.

### 2.4.1 Uniform Distribution

As expected, the uniform distribution is the continuous analogue of the discrete uniform distribution, which we considered in Section 2.2.1. Thus, whereas previously for the discrete random variable  $x_{\min} \leq x \leq x_{\max}$  we required

$$\sum_{i=x_{\min}}^{x_{\max}} P(x) = 1 ,$$

for the continuous random variable distributed uniformly in the same domain we require,

$$\int_{x_{\min}}^{x_{\max}} f(x) \, dx = 1 .$$

Accordingly, the uniform distribution has probability density given by

$$f(x) = \begin{cases} \frac{1}{x_{\max} - x_{\min}} & \text{if } x_{\min} \leq x \leq x_{\max} \\ 0 & \text{otherwise.} \end{cases} \quad (2.24)$$

We obtain this probability density function in *Mathematica* using

```
In[23]:= PDF[UniformDistribution[{xmin, xmax}], x]
```

```
Out[23]= { 1/(xmax-xmin)  xmin ≤ x ≤ xmax
```

The mean and variance are

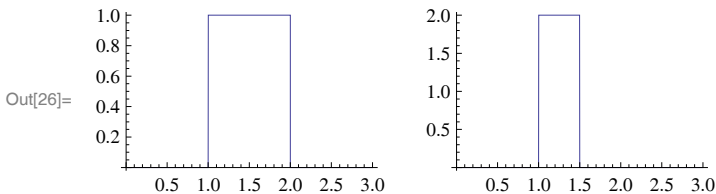
```
In[24]:= Mean[UniformDistribution[{xmin, xmax}]]
Variance[UniformDistribution[{xmin, xmax}]]

Out[24]=  $\frac{x_{\max} + x_{\min}}{2}$ 

Out[25]=  $\frac{1}{12} (x_{\max} - x_{\min})^2$ 
```

In the following plots of the probability density function we use the option **Exclusions -> None** to connect the discontinuities in the function.

```
In[26]:= GraphicsGrid[
  {{Plot[PDF[UniformDistribution[{1, 2}], x],
    {x, 0, 3}, Exclusions -> None],
    Plot[PDF[UniformDistribution[{1, 1.5}], x],
    {x, 0, 3}, Exclusions -> None]}}
```



Out[26]=

The plot on the right is generated for the uniformly distributed random variable in a domain where  $(x_{\max} - x_{\min}) < 1$  such that  $f(x) > 1$ . This is an important feature of probability density functions, whereas probabilities must be between 0 and 1, probability densities can exceed 1.

### 2.4.2 Normal Distribution

Most of us have encountered the classical bell-shaped normal, or Gaussian, distribution. It describes the distribution of data arising in many physical and biological contexts very well. The normal distribution is fully defined by its mean,  $\mu$  and variance  $\sigma^2$  and has probability density given by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (2.25)$$

We call the probability density function, mean and variance in *Mathematica* in the usual way:

```
In[27]:= PDF[NormalDistribution[μ, σ], x]
Mean[NormalDistribution[μ, σ]]
Variance[NormalDistribution[μ, σ]]
```

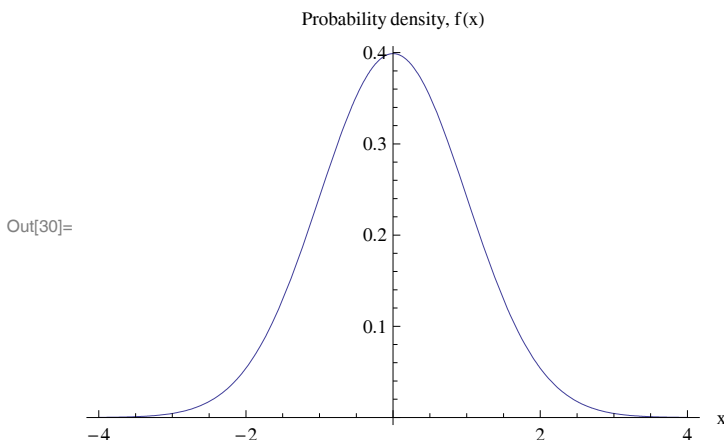
$$\text{Out[27]} = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}$$

$$\text{Out[28]} = \mu$$

$$\text{Out[29]} = \sigma^2$$

The distribution is defined in the domain  $-\infty < x < \infty$  and it is symmetrical about the mean:

```
In[30]:= Plot[PDF[NormalDistribution[0, 1], x], {x, -4, 4},
  AxesLabel -> {"x", "Probability density, f(x)"}]
```



We have already noted that the normal distribution is often found to describe distributions encountered in a wide variety of contexts. This very convenient property of many random variables can be attributed to the central limit theorem. Here we state the central limit theorem in simple terms following Chatfield [14]; detailed discussion and proof of the theorem are given by, *e.g.* Papoulis [119], pp. 278–284.

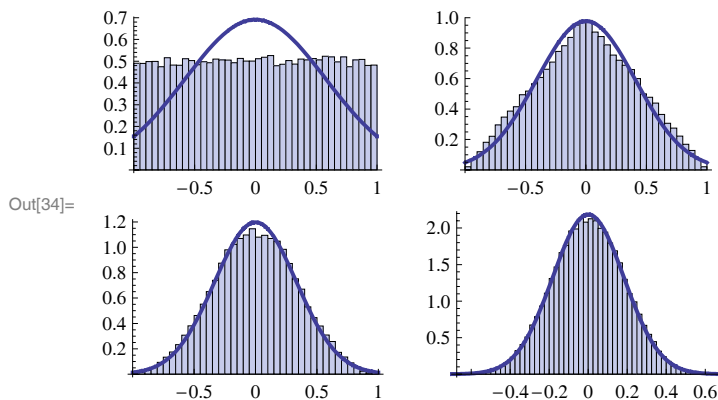
Consider the random variable

$$x = \frac{(x_1 + x_2 + \dots x_n)}{n}$$

where the  $x_i$  are drawn from independent and identical distributions with mean  $\mu$  and variance  $\sigma^2$ . The central limit theorem states that the distribution of  $x$  is approximately normal with mean  $\mu$  and variance  $\sigma^2/n$  and that as  $n$  increases, so does the quality of the approximation. Importantly, the source distribution of the  $x_i$  does not need to be specified; the central limit theorem applies for any source distribution.

The following *Mathematica* code illustrates the central limit theorem by plotting the histogram of 50,000 independent  $x$  where the source distribution is a uniform distribution on the interval  $\{-1, 1\}$  which has mean  $\mu = 0$  and variance  $1/3$ . With each histogram, we plot the probability density function of the normal distribution with the same mean and variance  $1/(3n)$ .

```
In[31]:=  $\sigma$  = StandardDeviation[UniformDistribution[{-1, 1}]];
Needs["Histograms`"]
H[n_] :=
Show[Histogram[Mean[RandomReal[{-1, 1}], {n, 50 000}]],
HistogramCategories -> 50, HistogramScale -> 1],
Plot[PDF[NormalDistribution[0,  $\sigma/\sqrt{n}$ ], x],
{x, -1, 1}, PlotStyle -> Thickness[.015]]]
GraphicsArray[{{H[1], H[2]}, {H[3], H[10]}}]
 $\sigma$  = .
```



The graphic in the top left shows the histogram for our source uniform distribution, which is manifestly not well approximated by the normal distribution. On the top right, we see the histogram for the case when  $n = 2$ . Here, each of our values of  $x$  is the mean of two values drawn randomly from the uniform distribution; the resultant histogram exhibits a symmetrical triangular distribution. On the bottom left,  $n = 3$  and we can see quite clearly that the approximation of the histogram to the normal distribution is improving. In

the final case considered here, as shown on the bottom right,  $n = 10$  and the approximation is very good. Of course, our example is rather static, but it is simple to generate a **Manipulate** object to investigate the influence of  $n$  interactively.

```
Manipulate[H[m], {m, 1, 20, 1}]
```

The code that we have used to demonstrate the central limit theorem introduces a few *Mathematica* commands. **Histogram** operates on a one-dimensional list to count the occurrences of data in intervals, the width of which are determined by the maximum and minimum values occurring in the list and the number of categories; we have specified this using the option **HistogramCategories**, but if this is not specified, the number of categories are chosen automatically. The option **HistogramScale -> 1** scales the heights of the bars so that the area under the histogram is 1, allowing direct comparison with a plot of a probability density function, as we have done here. The uniformly distributed random data in the interval  $\{-1, 1\}$  are generated using **RandomReal** as opposed to the command **RandomInteger** used earlier. Here we have specified an interval, though if we input **RandomReal[ ]** then we would obtain uniformly distributed random numbers between 0 and 1. We have encountered the command **Mean** earlier, but here we take advantage of the way that it operates on a list. In our example, **Mean** operates on a list consisting of  $n$  sublists, each of length 50,000, and computes the mean of the  $i$ th value of each of these sublists to generate a new list of length 50,000, where each element represents a value of  $x$ . The manner in which **Mean** operates on a list of sublists is perhaps best understood by examining the following example:

```
In[36]:= Mean[{{a, b, c}, {d, e, f}, {g, h, i}}]
```

```
Out[36]:=  $\left\{ \frac{1}{3} (a + d + g), \frac{1}{3} (b + e + h), \frac{1}{3} (c + f + i) \right\}$ 
```

We will make considerable use of *Mathematica*'s advanced list handling capabilities when applying Monte Carlo techniques to the modelling of fibre networks, and by understanding precisely how they work, we are able to optimise our code, reducing evaluation times considerably.

### 2.4.3 Lognormal Distribution

The random variable  $x$  has a lognormal distribution if the random variable  $y = \log(x)$  is normally distributed with mean  $\mu$  and variance  $\sigma^2$ . The probability density for the lognormal distribution is given by

$$f(x) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma x} e^{-\frac{(\log(x)-\mu)^2}{2\sigma^2}} & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (2.26)$$

It is obtained using one of the variable transform formulae. We shall encounter variable transform formulae for the sums and products of independent random variables repeatedly in subsequent chapters. The simplest variable transform formula allows us to obtain the probability density of  $y = g(x)$  if the probability density of  $x$  is  $f(x)$ . We state it as follows:

$$f(y) = f(g(x)) \left| \frac{dy}{dx} \right|. \quad (2.27)$$

The second term on the right-hand side of Equation 2.27 is called the Jacobian and takes account of the change in the domain of the random variable such that the resultant probability density integrates to 1 over its domain. Note that to perform this variable transformation, we require the relationship between  $y$  and  $x$  to be one-to-one, *i.e.* each value of  $y$  is associated with only one value of  $x$ . We are interested in the variable  $y = \log(x)$  which is indeed one-to-one so we perform the change of variable by doing,

```
In[37]:= y = Log[x];
PDF[NormalDistribution[μ, σ], y] D[y, x]
y = .
```

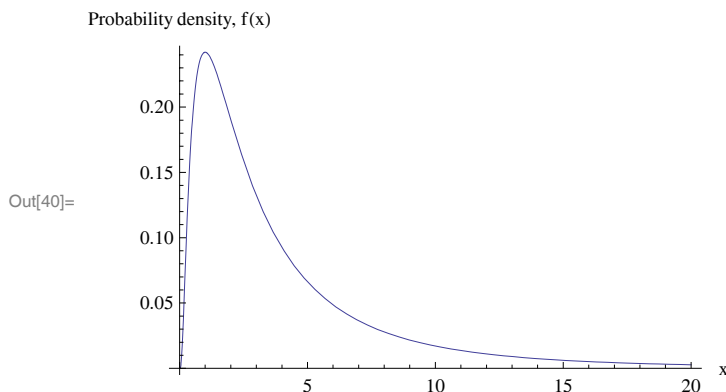
$$\text{Out[38]} = \frac{e^{-\frac{(-\mu + \text{Log}[x])^2}{2\sigma^2}}}{\sqrt{2\pi} \, x \, \sigma}$$

Which is the probability density function for the lognormal distribution as given by Equation 2.26. We call this probability density in *Mathematica* using

```
PDF[LogNormalDistribution[μ, σ], x]
```

The probability density is defined in the domain  $0 \leq x \leq \infty$  and it exhibits a positive skew.

```
In[40]:= Plot[PDF[LogNormalDistribution[1, 1], x], {x, 0, 20},
AxesLabel -> {"x", "Probability density, f(x)}]
```



We obtain the mean and variance in the usual way and from these can calculate the coefficient of variation using **PowerExpand** in conjunction with **FullSimplify**:

```
In[41]:= Mean[LogNormalDistribution[μ, σ]]
          Variance[LogNormalDistribution[μ, σ]]
          FullSimplify[PowerExpand[Sqrt[%] / %%]]
```

Out[41]=  $e^{\mu + \frac{\sigma^2}{2}}$

Out[42]=  $e^{2\mu + \sigma^2} (-1 + e^{\sigma^2})$

Out[43]=  $\sqrt{-1 + e^{\sigma^2}}$

We observe then that the mean and variance of the lognormal distribution are defined in terms of those of the normal distribution,  $\mu$  and  $\sigma^2$ , respectively, and that the coefficient of variation depends only on  $\sigma$ .

We have just seen that the normal distribution arises as a consequence of the central limit theorem when considering the sum of several  $x_i$  drawn from independent and identical distributions. The central limit theorem provides also the explanation for the occurrence of the lognormal distribution. Consider the product of  $n$  independent positive random variables  $x_i$ :

$$x = x_1 x_2 \dots x_n$$

The random variable  $y = \log(x)$  is given by

$$y = \log(x) = \log(x_1) + \log(x_2) + \dots + \log(x_n)$$

Since random variables arising as the sum of independent random variables exhibit a normal distribution, we may state that the random variable  $y = \log(x)$  exhibits a normal distribution and thus the random variable  $x$  exhibits a lognormal distribution.

#### 2.4.4 Exponential distribution

The random variable  $x$  with mean  $\bar{x}$  is said to have an exponential distribution if its probability density is given by

$$f(x) = \begin{cases} \frac{1}{\bar{x}} e^{-\frac{x}{\bar{x}}} & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (2.28)$$

In Section 3.3.2 we will show that the exponential distribution arises as the distribution of intervals between random events as described by the Poisson distribution. For now, we note that the probability density given by Equation 2.28 is a weighted form of the probability of zero events in a point Poisson process where the discrete random variable is the occurrence of events in an interval.

We obtain the exponential probability density and its mean and variance in *Mathematica* using,

```
In[44]:= PDF[ExponentialDistribution[1 / xbar], x]
Mean[ExponentialDistribution[1 / xbar]]
Variance[ExponentialDistribution[1 / xbar]]
```

```
Out[44]= 
$$\frac{e^{-\frac{x}{\text{xbar}}}}{\text{xbar}}$$

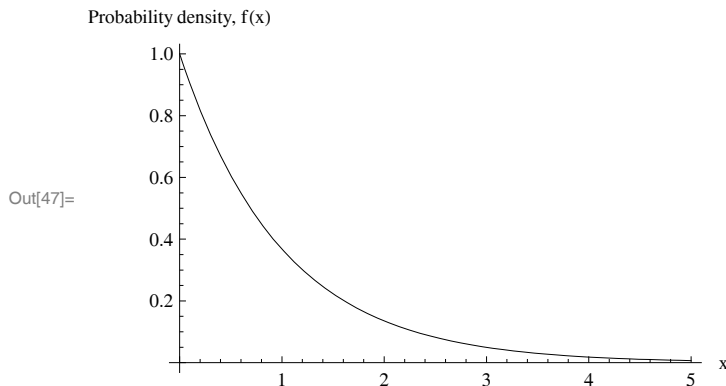
```

```
Out[45]= xbar
```

```
Out[46]= xbar2
```

Note that since the variance of the distribution is the square of the mean, the exponential distribution has constant coefficient of variation  $cv(x) = 1$ . Accordingly, the shape of the distribution is unaltered by the mean which acts as a scaling parameter.

```
In[47]:= Plot[PDF[ExponentialDistribution[1], x], {x, 0, 5},
  AxesLabel -> {"x", "Probability density, f(x)"},
  PlotRange -> All]
```



### 2.4.5 Gamma Distribution

The random variable  $x$  is said to have a gamma distribution if its probability density is given by

$$f(x) = \left(\frac{\alpha}{\beta}\right)^{\alpha} \frac{x^{\alpha-1}}{\Gamma(\alpha)} e^{-\alpha x/\beta}, \quad (2.29)$$

where the term  $\Gamma(\alpha)$  represents the Euler gamma function, which we encountered on page 35. We obtain the probability density, mean, variance and coefficient of variation as previously:

```
In[48]:= PDF[GammaDistribution[α, β / α], x]
Mean[GammaDistribution[α, β / α]]
Variance[GammaDistribution[α, β / α]]
PowerExpand[√% / %%]
```

$$\text{Out[48]} = \frac{e^{-\frac{x\alpha}{\beta}} x^{-1+\alpha} \left(\frac{\beta}{\alpha}\right)^{-\alpha}}{\text{Gamma}[\alpha]}$$

$$\text{Out[49]} = \beta$$

$$\text{Out[50]} = \frac{\beta^2}{\alpha}$$

$$\text{Out[51]} = \frac{1}{\sqrt{\alpha}}$$

So, in the form given in Equation 2.29, the distribution has mean  $\bar{x} = \beta$  and coefficient of variation  $cv(x) = 1/\sqrt{\alpha}$ .

Although the probability density for the gamma distribution is conventionally expressed in terms of parameters  $\alpha$  and  $\beta$ , it can be expressed equally well in terms of the mean and coefficient of variation:

```
In[52]:= PDF[GammaDistribution[1/cv^2, xbar cv^2], x]
Mean[GammaDistribution[1/cv^2, xbar cv^2]]
Variance[GammaDistribution[1/cv^2, xbar cv^2]]
PowerExpand[Sqrt[%]/%]
```

$$\text{Out[52]} = \frac{e^{-\frac{x}{cv^2 \bar{x}}} x^{-1+\frac{1}{cv^2}} (cv^2 \bar{x})^{-\frac{1}{cv^2}}}{\text{Gamma}\left[\frac{1}{cv^2}\right]}$$

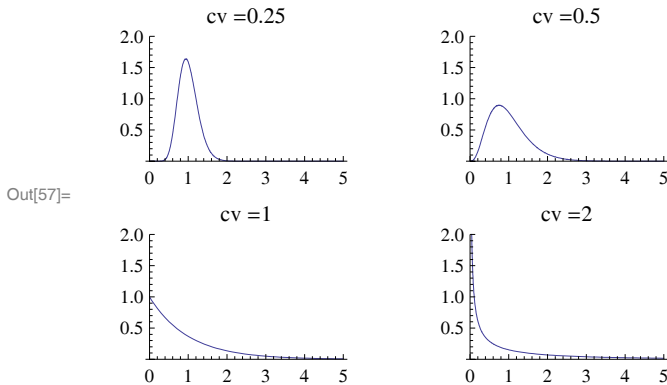
```
Out[53]= xbar
```

```
Out[54]= cv^2 xbar^2
```

```
Out[55]= cv
```

The mean  $\bar{x} = \beta$  acts as a scaling parameter for the gamma distribution and the coefficient of variation  $cv(x) = 1/\sqrt{\alpha}$  controls its shape. We illustrate the influence of the coefficient of variation on the shape of the distribution by comparing plots of the probability density for distributions with unit mean:

```
In[56]:= PlotPDF[cv_] :=
Plot[PDF[GammaDistribution[1/cv^2, cv^2], x],
{x, 0, 5}, PlotRange -> {All, {0, 2}},
PlotLabel -> Row[{"cv =", cv}]]
GraphicsArray[{{PlotPDF[0.25], PlotPDF[0.5]},
{PlotPDF[1], PlotPDF[2]}}]
```



We observe that the distribution exhibits a positive skew and this increases with the coefficient of variation:

```
In[58]:= PowerExpand[Skewness[GammaDistribution[1/cv^2, cv^2]]]
Out[58]= 2 cv
```

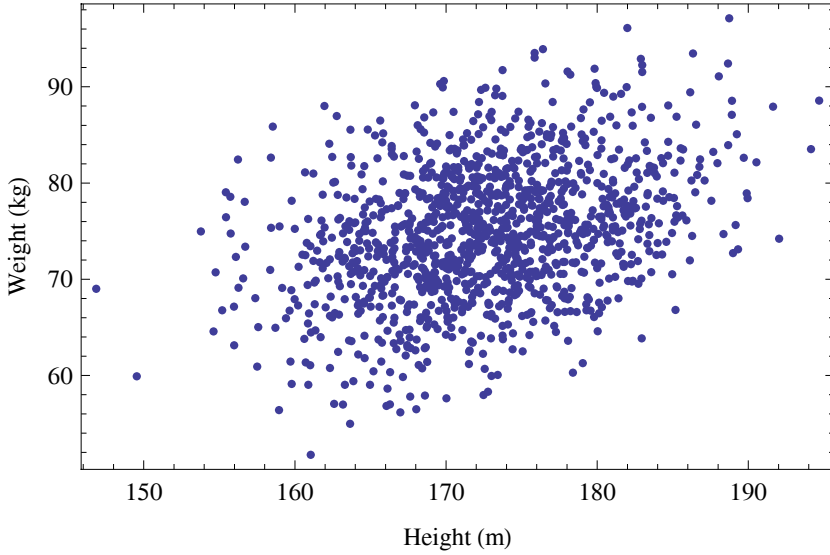
Note that the distribution exhibits a maximum only when  $cv(x) < 1$  and decreases monotonically otherwise.

The gamma distribution includes the exponential distribution as a special case when  $\alpha = 1$ :

```
In[59]:= TrueQ[GammaDistribution[1,  $\beta$ ] ==
           ExponentialDistribution[1 /  $\beta$ ]]
Out[59]= True
```

Recall that the coefficient of variation of the exponential distribution is 1 and that of the gamma distribution is  $1/\sqrt{\alpha}$ . This means that if a random variable has a gamma or exponential distribution, then processes that change the mean will change the standard deviation proportionately and a plot of the standard deviation against the mean will be linear with gradient  $cv(x)$ . Hwang and Hu [69] provide a proof that for independent positive random variables  $x_1, x_2, \dots, x_n$  with a common continuous probability density function, this property of the sample mean  $\bar{x}$  and coefficient of variation  $cv(x)$  being independent is equivalent to the  $x_i$  being drawn from a gamma distribution. Such linearity is common in experimental data characterising the void structure of fibrous materials and Johnston [72, 73] proposes that the gamma distribution describes the pore size distribution of stochastic porous materials in general.

It turns out that the sum of  $n$  independent exponential random variables is a gamma distribution. This is consistent with our remarks concerning the central limit theorem on page 41. The central limit theorem gives  $cv(x, n) = 1/\sqrt{n}$  such that as  $n$  increases,  $cv(x, n)$  decreases. Recall that the skewness of the gamma distribution is twice the coefficient of variation, so as  $n$  increases the distribution becomes increasingly symmetrical and thus, the sum of our independent exponential random variables approaches a normal distribution. As a rule of thumb, we note that the probability densities of the normal and the gamma distribution are similar for coefficients of variation less than about 0.2. Another useful property of the gamma distribution is that products of gamma distributions are themselves well approximated by the gamma distribution.



**Figure 2.1.** Simulated data for typical weight and height distributions of men

## 2.5 Multivariate Distributions

So far we have considered the distributions of single random variables and those arising from combinations of several independent distributions. Very often, we need to consider distributions of *dependent* random variables. For example, if we recorded the heights and weights of students at a university we would expect both variables to exhibit a normal distribution. Equally, we might expect taller students to be heavier than shorter students, but we would not be surprised to find tall students who were lighter than average or short students who were heavier. Thus, we would expect a scatter plot of weight against height for male students to look something like Figure 2.1.

Our example considers two random variables, height and weight, which exhibit a degree of interdependence. As such, we classify the distribution as being bivariate. Since the distributions of height and weight both exhibit a normal distribution, then the data in Figure 2.1 exhibit a bivariate normal distribution. We shall consider this distribution in more detail shortly, but first we note that bivariate distributions often occur in stochastic fibrous materials and in the fibres from which they are formed. For example, the length and diameter of wool fibres [94], man-made mineral fibres [67, 150] and wood-pulp

fibres [85] are often distributed according to bivariate distributions including the bivariate lognormal distribution and the distributions of mass and thickness in near-random fibre mats are well described by the bivariate normal distribution [36, 37].

To handle bivariate distributions, we require some additional statistical descriptors. The *covariance* is a measure of the association between random variables. For the random variables  $x$  and  $y$  it is given by

$$Cov(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) , \quad (2.30)$$

and it is easy to show that

$$Cov(x, y) = \overline{xy} - \bar{x}\bar{y} . \quad (2.31)$$

The units of covariance are the product of those of the constituent variables  $x$  and  $y$ . The covariance of the random variables  $k_x x$  and  $k_y y$  is

$$Cov(k_x x, k_y y) = k_x k_y Cov(x, y) , \quad (2.32)$$

and the variance of the sum of the random variables  $x$  and  $y$  is

$$\sigma^2(x+y) = \sigma^2(x) + 2Cov(x, y) + \sigma^2(y) . \quad (2.33)$$

Often, we seek to standardize the covariance so that it is dimensionless. It is convenient to do this by dividing  $(x_i - \bar{x})$  and  $(y_i - \bar{y})$  by their standard deviations  $\sigma(x)$  and  $\sigma(y)$ , respectively. The resultant expression gives the *correlation* between the variables,  $-1 \leq \rho \leq 1$ :

$$\rho = \frac{1}{n-1} \sum_{i=1}^n \frac{(x_i - \bar{x})}{\sigma(x)} \frac{(y_i - \bar{y})}{\sigma(y)} \quad (2.34)$$

$$= \frac{Cov(x, y)}{\sigma(x)\sigma(y)} , \quad (2.35)$$

where  $\rho$  is called the correlation coefficient.

Negative covariance and correlation tells us that as the value of one variable increases, so that of the other decreases; positive correlation tells us that an increase in one variable is associated with an increase in the other; if the variables are independent, then the correlation and covariance are zero<sup>2</sup>. The size of the correlation coefficient tells us the degree of scatter in a plot such as Figure 2.1, so can be interpreted as providing a measure of the range of spread of  $x$  and  $y$  about a regression line. Often the square of the correlation coefficient, is reported for experimental data; this parameter, typically denoted  $r^2$ , is called the coefficient of determination.

---

<sup>2</sup> The converse is not necessarily true [119].

### 2.5.1 Bivariate Normal Distribution

As we might expect given our earlier discussion, the bivariate distribution encountered most frequently is the bivariate normal, or Gaussian, distribution. The bivariate normal distribution arises when the random variables  $x$  and  $y$  both exhibit a normal distribution and are correlated. This means that if we select any  $x_i$  from a bivariate normal distribution, then we expect the range of  $y_i$  that can be associated with it to be constrained to an extent depending on the correlation  $\rho$ . We state then that  $x$  and  $y$  have *joint* probability density.

For the bivariate normal distribution, the joint probability density of  $x$  and  $y$  with correlation coefficient  $\rho$  is

$$f(x, y) = \frac{1}{2\pi\sigma(x)\sigma(y)\sqrt{1-\rho^2}} e^{-\frac{z}{2(1-\rho^2)}} \quad (2.36)$$

where

$$z = \frac{(x - \bar{x})^2}{\sigma^2(x)} - \frac{2\rho(x - \bar{x})(y - \bar{y})}{\sigma(x)\sigma(y)} + \frac{(y - \bar{y})^2}{\sigma^2(y)}$$

In *Mathematica* we obtain the joint probability density function in this form using the command **MultinormalDistribution** which is in the **MultivariateStatistics** package:

```
In[60]:= Needs["MultivariateStatistics`"]
FullSimplify[PDF[MultinormalDistribution[{xbar, ybar},
  {{σx², ρ σx σy}, {ρ σx σy, σy²}}], {x, y}],
  {σx, σy} > 0 && {σx, σy} ∈ Reals]

Out[61]= 
$$\frac{e^{\frac{(y-ybar)^2 \sigma x^2 - 2 (x-xbar) (y-ybar) \rho \sigma x \sigma y + (x-xbar)^2 \sigma y^2}{2 (-1+\rho^2) \sigma x^2 \sigma y^2}}}{2 \pi \sqrt{1 - \rho^2} \text{Abs}[\sigma x] \text{Abs}[\sigma y]}$$

```

Note that we are considering only the bivariate normal distribution here, but the command **MultinormalDistribution** will handle trivariate and higher orders of multivariate normal distributions. Accordingly, the second argument for an  $n$ -variate distribution is input as an  $n \times n$  array known as the covariance matrix where the elements are the covariances of the variables with each other. From the definition of covariance  $Cov(x, x) = \sigma^2(x)$ , so the first and last terms are simply the variances of  $x$  and  $y$  and the matrix is symmetrical. *Mathematica* returns the mean and variance of the bivariate normal distribution as two element lists, and the covariance as a matrix in the same form as it was input:

```

In[62]:= Mean[MultinormalDistribution[
  {xbar, ybar}, {{σx², ρ σx σy}, {ρ σx σy, σy²}}]]
Variance[MultinormalDistribution[{xbar, ybar},
  {{σx², ρ σx σy}, {ρ σx σy, σy²}}]]
Covariance[MultinormalDistribution[
  {xbar, ybar}, {{σx², ρ σx σy}, {ρ σx σy, σy²}}]]

Out[62]= {xbar, ybar}

Out[63]= {σx², σy²}

Out[64]= {{σx², ρ σx σy}, {ρ σx σy, σy²}}

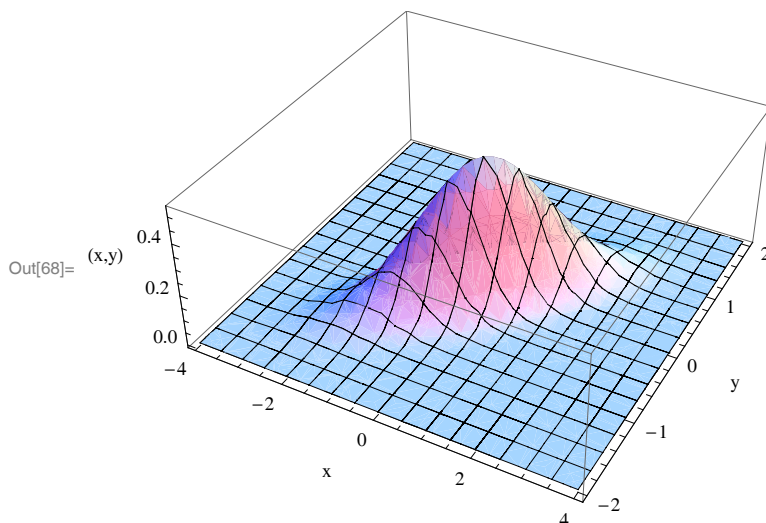
```

The joint probability density is represented by a surface. Here we generate this for a bivariate normal distribution with mean  $\bar{x} = \bar{y} = 0$ , variances  $\sigma^2(x) = 1$ ,  $\sigma^2(y) = \frac{1}{4}$  and correlation  $\rho = 0.8$ . Note the use of the command **Clear** to unset several variables; this command may also be used to clear functions which have been previously defined.

```

In[65]:= σx = 1;
σy = 1/2;
ρ = 0.8;
Plot3D[PDF[MultinormalDistribution[{0, 0},
  {{σx², ρ σx σy}, {ρ σx σy, σy²}}], {x, y}],
  {x, -4, 4}, {y, -2, 2}, PlotRange → All,
  AxesLabel → {"x", "y", "f(x,y)"}]
Clear[σx, σy, ρ]

```



The fraction of the distribution which lies in the interval  $(x + \Delta x, y + \Delta y)$  is

$$P(x, y) = \int_x^{x+\Delta x} \int_y^{y+\Delta y} f(x, y) \, dy \, dx , \quad (2.37)$$

and the joint probability density integrates to unity (*cf.* Equation 2.20):

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \, dy \, dx = 1 . \quad (2.38)$$



<http://www.springer.com/978-1-84800-990-5>

Modelling Stochastic Fibrous Materials with  
Mathematica®

Sampson, W.W.

2009, XI, 277 p., Hardcover

ISBN: 978-1-84800-990-5