

Lecture Notes Accompanying

STAT2003/STAT7003

Mathematical Probability

Semester 1, 2025

by the

Statistics Group

of the School of Mathematics and Physics

21st February 2025

CONTENTS

1	Random Experiments and Probability Models	7
1.1	Random Experiments	7
1.2	Sample Space	12
1.3	Events	13
1.4	Probability	16
1.5	Counting	20
1.6	Conditional probability and independence	26
1.6.1	Product Rule	29
1.6.2	Law of Total Probability and Bayes' Rule	31
1.6.3	Independence	32
2	Random Variables and Probability Distributions	35
2.1	Random Variables	35
2.2	Probability Distribution	37
2.2.1	Discrete Distributions	39
2.2.2	Continuous Distributions	40
2.3	Expectation	41
2.4	Transforms	45
2.5	Some Important Discrete Distributions	47
2.5.1	Bernoulli Distribution	48
2.5.2	Binomial Distribution	48
2.5.3	Geometric distribution	50
2.5.4	Poisson Distribution	52
2.5.5	Hypergeometric Distribution	53
2.6	Some Important Continuous Distributions	54
2.6.1	Uniform Distribution	54
2.6.2	Exponential Distribution	55
2.6.3	Normal, or Gaussian, Distribution	57
2.6.4	Gamma- and χ^2 -distribution	59
3	Generating Random Variables on a Computer	61
3.1	Introduction	61
3.2	Random Number Generation	61
3.3	The Inverse-Transform Method	63

3.4	The Acceptance–Rejection Method	66
3.5	Generating From Commonly Used Distributions	68
4	Joint Distributions	73
4.1	Joint Distribution and Independence	74
4.1.1	Discrete Joint Distributions	74
4.1.2	Continuous Joint Distributions	78
4.2	Expectation	81
4.3	Conditional Distribution	88
4.4	Conditional Expectation	90
5	Functions of Random Variables and Limit Theorems	93
5.1	Functions of Random Variables	93
5.1.1	Sum of Two Random Variables	96
5.1.2	Linear Transformations	97
5.1.3	General Transformations	99
5.2	Jointly Normal Random Variables	101
5.3	Normal Linear Models	104
5.3.1	Linear Regression	105
5.4	Limit Theorems	107
6	Markov Chains	113
6.1	Introduction	113
6.2	Simulating Markov Chains	118
6.3	Limiting Behaviour	120
7	Reliability	123
7.1	Introduction	123
7.2	Structure function	123
7.3	Reliability function	129
7.4	Non-repairable systems	131
7.5	Lifetime distributions	132
A	Exercises and Solutions	135
A.1	Problem Set 1	135
A.2	Answer Set 1	137
A.3	Problem Set 2	139
A.4	Answer Set 2	142
A.5	Problem Set 3	144
A.6	Answer Set 3	147
B	More Exercises	151
C	Summary of Formulas	155

D Python Primer	161
D.1 Getting Started	161
D.2 Python Objects	164
D.3 Types and Operators	165
D.4 Functions and Methods	167
D.5 Modules	168
D.6 Flow Control	170
D.7 Iteration	171
D.8 Classes	173
D.9 Files	175
D.10 NumPy	178
D.10.1 Creating and Shaping Arrays	178
D.10.2 Slicing	180
D.10.3 Array Operations	181
D.10.4 Random Numbers	183
D.11 Matplotlib	184
D.11.1 Creating a Basic Plot	184
D.12 System Calls, URL Access, and Speed-Up	187
Index	189

PREFACE

These notes form a comprehensive, second-year introduction to probability. I will use these notes as a source of reference for STAT2003. You might find it helpful for self-learning. Further examples and exercises will be given at the tutorials and lectures. Any material discussed in the lectures and given in the tutorial problem sets is *fair game* for the exam.

To completely master STAT2003 it is important that you

1. visit the lectures, where I will provide many extra examples;
2. do the tutorial exercises and the exercises in the appendix, which are there to help you with the “technical” side of things; you will learn here to apply the concepts learned at the lectures,
3. carry out random experiments on the computer, in the simulation project. This will give you a better intuition about how randomness works.

All of these will be essential if you wish to understand probability beyond “filling in the formulas”.

Notation and Conventions

Throughout these notes I try to use a uniform notation in which, as a rule, the number of symbols is kept to a minimum. For example, I prefer q_{ij} to $q(i, j)$, X_t to $X(t)$, and $\mathbb{E}X$ to $\mathbb{E}[X]$.

The symbol “:=” denotes “is defined as”. We will also use the abbreviations r.v. for *random variable* and i.i.d. (or iid) for *independent and identically and distributed*.

Numbering

All references to Examples, Theorems, etc. are of the same form. For example, Theorem 1.2 refers to the second theorem of Chapter 1. References to formulas appear between brackets. For example, (3.4) refers to formula 4 of Chapter 3.

Bibliography

- W. Feller (1970). *An Introduction to Probability Theory and Its Applications*, Volume I., 2nd ed., John Wiley & Sons, Hoboken.
- H. Hsu (1997). *Probability, Random Variables & Random Processes*. Shaum’s Outline Series, McGraw–Hill, New York.
- D.P. Kroese and J.C.C. Chan (2014). *Statistical Modeling and Computation*, Springer, New York.
- D.P. Kroese, Z.I. Botev, T. Taimre, R. Vaisman. *Data Science and Machine Learning: Mathematical and Statistical Methods* (2019). Chapman & Hall/CRC Press, Boca Raton.
- S. M. Ross (2005). *A First Course in Probability*, 7th ed., Prentice-Hall, Englewood Cliffs.
- R.Y. Rubinstein and D.P. Kroese, (2016). *Simulation and the Monte Carlo Method*, 3rd ed., John Wiley & Sons, Hoboken.

RANDOM EXPERIMENTS AND PROBABILITY MODELS

1.1 Random Experiments

The basic notion in probability is that of a **random experiment**: an experiment whose outcome cannot be determined in advance, but is nevertheless still subject to analysis.

Examples of random experiments are:

1. tossing a die,
2. measuring the amount of rainfall in Brisbane in January,
3. counting the number of calls arriving at a telephone exchange during a fixed time period,
4. selecting a random sample of fifty people and observing the number of left-handers,
5. choosing at random ten people and measuring their height.

■ **Example 1.1 (Coin Tossing)** The most *fundamental* stochastic experiment is the experiment where a coin is tossed a number of times, say n times. Indeed, much of probability theory can be based on this simple experiment, as we shall see in subsequent chapters. To better understand how this experiment behaves, we can carry it out on a digital computer, for example in Python. The following simple Python program, simulates a sequence of 100 tosses with a fair coin(that is, heads and tails are equally likely), and plots the results in a bar chart.

```

from numpy.random import rand
from numpy import cumsum, arange
import matplotlib.pyplot as plt
n = 100
x = (rand(n) < 0.5).astype(int)
plt.figure(1)
plt.bar(arange(n), x)

```

Here x is a vector with 1s and 0s, indicating Heads and Tails, say. Typical outcomes for three such experiments are given in Figure 1.1.

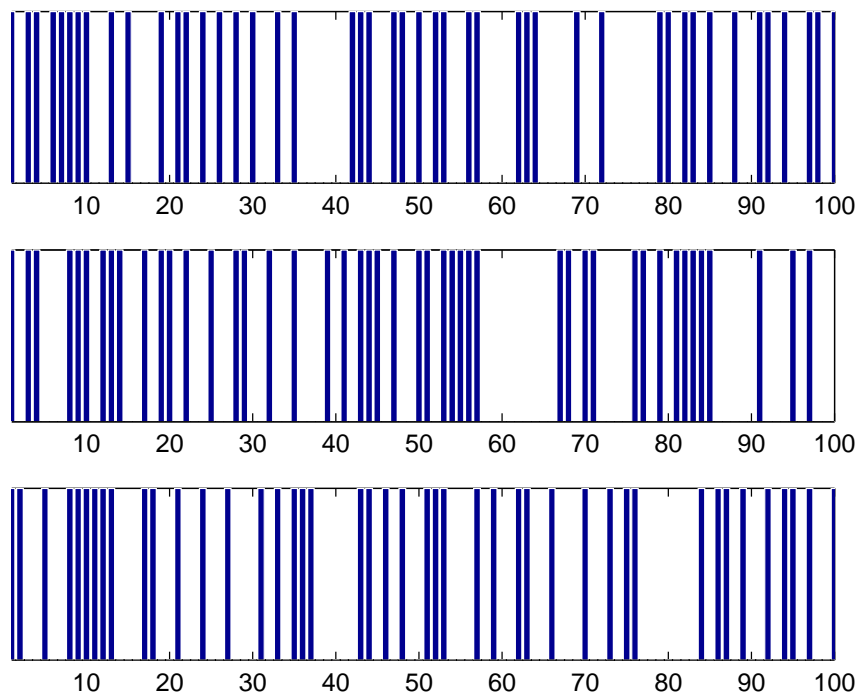


Figure 1.1: Three experiments where a fair coin is tossed 100 times. The dark bars indicate when “Heads” (=1) appears.

We can also plot the average number of “Heads” against the number of tosses. In the same Python program, this is done with a few extra lines of code:

```

y = cumsum(x)
t = arange(n)+1
plt.figure(2)
plt.plot(y/t)

```

The result of three such experiments is depicted in Figure 1.2. Notice that the average

number of Heads seems to converge to $1/2$, but there is a lot of random fluctuation.

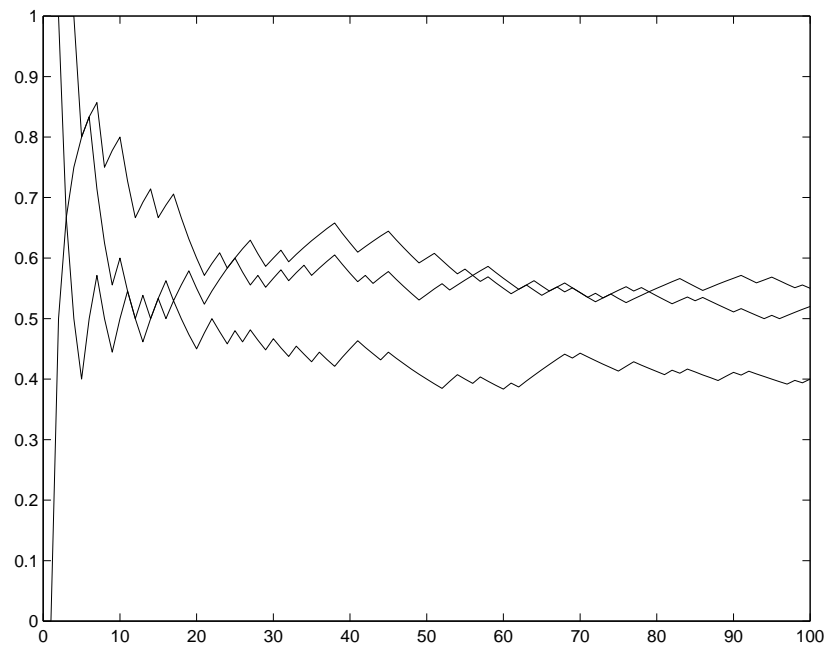


Figure 1.2: The average number of heads in n tosses, where $n = 1, \dots, 100$.

Try it yourself by executing the following code (after the ones above):

```
plt.figure(3)
n=100
t = arange(n)+1
for i in range(n):
    x = (rand(n) < 0.5).astype(int)
    y = cumsum(x)
    plt.plot(y/t, 'k', alpha=0.1)
```

■ **Example 1.2 (Control Chart)** Control charts, see Figure 1.3, are frequently used in manufacturing as a method for *quality control*. Each hour the average output of the process is measured — for example, the average weight of 10 bags of sugar — to assess if the process is still “in control”, for example, if the machine still puts on average the correct amount of sugar in the bags. When the process $> \text{Upper Control Limit}$ or $< \text{Lower Control Limit}$ and an alarm is raised that the process is out of control, e.g., the machine needs to be adjusted, because it either puts too much or not enough sugar in the bags. The question is how to set the control limits, since the random process naturally fluctuates around its “center” or “target” line.

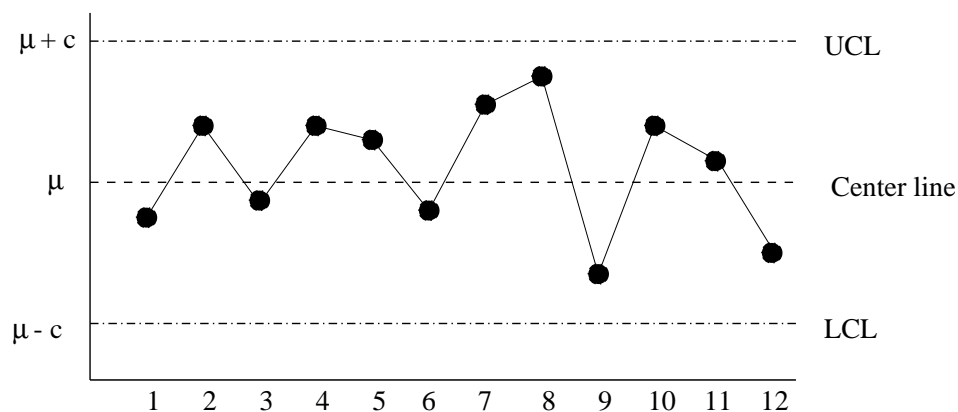


Figure 1.3: Control Chart

■ **Example 1.3 (Machine Lifetime)** Suppose 1000 identical components are monitored for failure, up to 50,000 hours. The outcome of such a random experiment is typically summarised via the cumulative lifetime table and plot, as given in Table 1.1 and Figure 1.3, respectively. Here $\widehat{F}(t)$ denotes the proportion of components that have failed at time t . One question is how $\widehat{F}(t)$ can be modelled via a continuous function F , representing the lifetime distribution of a typical component.

t (h)	failed	$\widehat{F}(t)$	t (h)	failed	$\widehat{F}(t)$
0	0	0.000	3000	140	0.140
750	22	0.022	5000	200	0.200
800	30	0.030	6000	290	0.290
900	36	0.036	8000	350	0.350
1400	42	0.042	11000	540	0.540
1500	58	0.058	15000	570	0.570
2000	74	0.074	19000	770	0.770
2300	105	0.105	37000	920	0.920

Table 1.1: The cumulative lifetime table

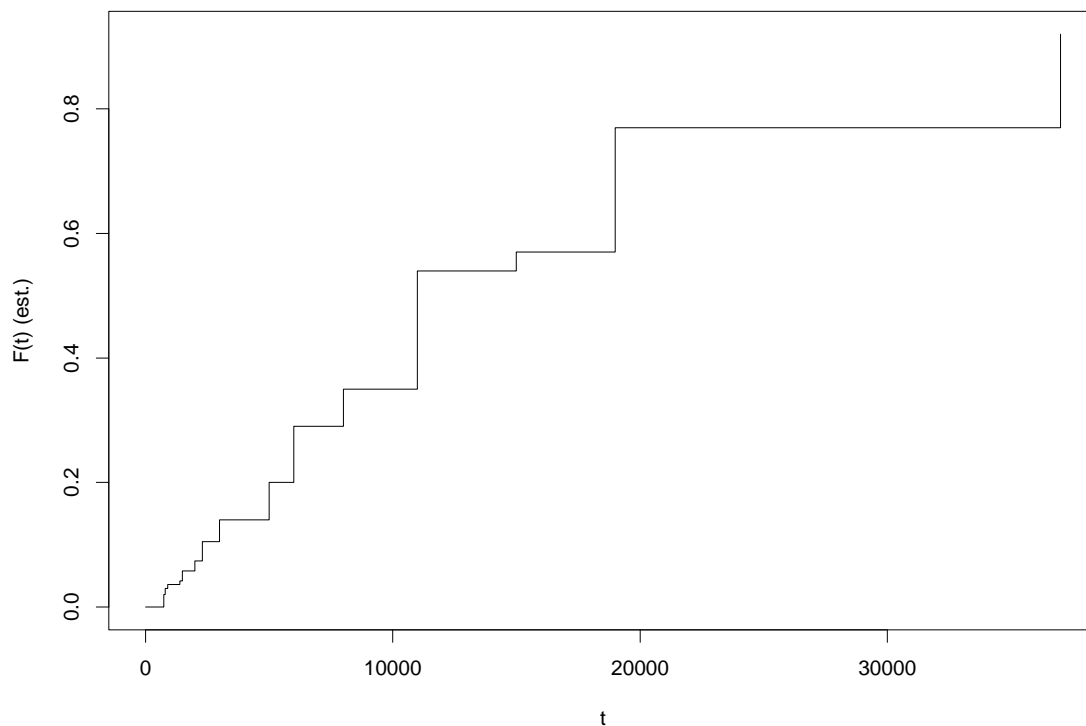


Figure 1.4: The cumulative lifetime table

■ **Example 1.4 (Aeroplane)** A 4-engine aeroplane is able to fly on just one engine on each wing. All engines are unreliable.

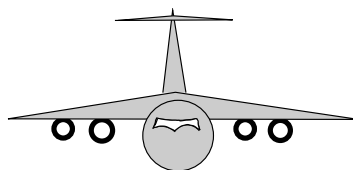


Figure 1.5: A aeroplane with 4 unreliable engines

Number the engines: 1,2 (left wing) and 3,4 (right wing). Observe which engine works properly during a specified period of time. There are $2^4 = 16$ possible outcomes of the experiment. Which outcomes lead to “system failure”? Moreover, if the probability of failure

within some time period is known for each of the engines, what is the probability of failure for the entire system? Again this can be viewed as a random experiment. ■

Below are two more pictures of randomness. The first is a computer-generated “plant”, which looks remarkably like a real plant. The second is real data depicting the number of bytes that are transmitted over some communications link. An interesting feature is that the data can be shown to exhibit “fractal” behaviour, that is, if the data is aggregated into smaller or larger time intervals, a similar picture will appear.

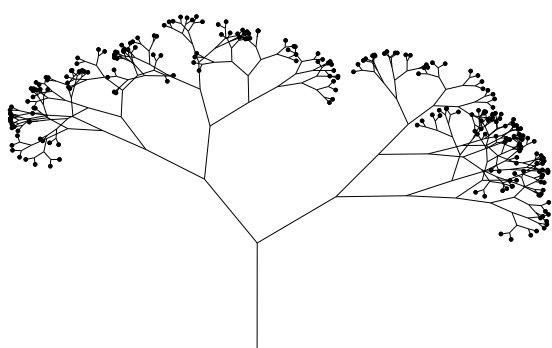


Figure 1.6: Plant growth

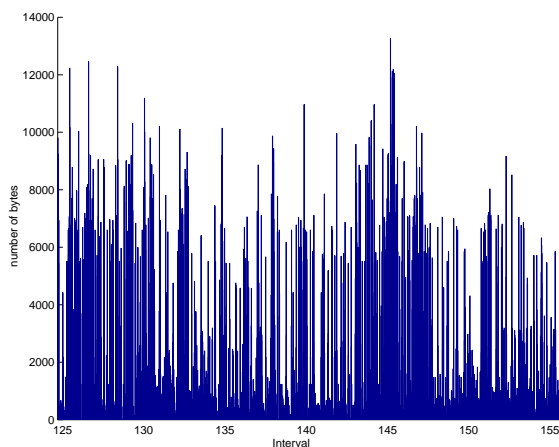


Figure 1.7: Telecommunications data

We wish to describe these experiments via appropriate mathematical models. These models consist of three building blocks: a *sample space*, a set of *events* and a *probability*. We will now describe each of these objects.

1.2 Sample Space

Although we cannot predict the outcome of a random experiment with certainty we usually can specify a set of possible outcomes. This gives the first ingredient in our model for a random experiment.

Definition 1.1: Sample Space

The **sample space** Ω of a random experiment is the set of all possible outcomes of the experiment.

Examples of random experiments with their sample spaces are:

1. Cast two dice consecutively,

$$\Omega = \{(1, 1), (1, 2), \dots, (1, 6), (2, 1), \dots, (6, 6)\}.$$

2. The lifetime of a machine (in days),

$$\Omega = \mathbb{R}_+ = \{ \text{positive real numbers} \}.$$

3. The number of arriving calls at an exchange during a specified time interval,

$$\Omega = \{0, 1, \dots\} = \mathbb{Z}_+.$$

4. The heights of 10 selected people.

$$\Omega = \{(x_1, \dots, x_{10}), x_i \geq 0, i = 1, \dots, 10\} = \mathbb{R}_+^{10}.$$

Here (x_1, \dots, x_{10}) represents the outcome that the length of the first selected person is x_1 , the length of the second person is x_2 , et cetera.

Notice that for modelling purposes it is often easier to take the sample space larger than necessary. For example the actual lifetime of a machine would certainly not span the entire positive real axis. And the heights of the 10 selected people would not exceed 3 metres.

1.3 Events

Often we are not interested in a single outcome but in whether or not one of a *group* of outcomes occurs. Such subsets of the sample space are called **events**. Events will be denoted by capital letters A, B, C, \dots . We say that event A **occurs** if the outcome of the experiment is one of the elements in A .

Examples of events are:

1. The event that the sum of two dice is 10 or more,

$$A = \{(4, 6), (5, 5), (5, 6), (6, 4), (6, 5), (6, 6)\}.$$

2. The event that a machine lives less than 1000 days,

$$A = [0, 1000).$$

3. The event that out of fifty selected people, five are left-handed,

$$A = \{5\}.$$

■ **Example 1.5 (Coin Tossing)** Suppose that a coin is tossed 3 times, and that we “record” every head and tail (not only the number of heads or tails). The sample space can then be written as

$$\Omega = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\} ,$$

where, for example, HTH means that the first toss is heads, the second tails, and the third heads. An alternative sample space is the set $\{0, 1\}^3$ of binary vectors of length 3, e.g., HTH corresponds to (1,0,1), and THH to (0,1,1).

The event A that the third toss is heads is

$$A = \{HHH, HTH, THH, TTH\} .$$

■

Since events are sets, we can apply the usual set operations to them:

1. the set $A \cup B$ (A **union** B) is the event that A *or* B *or* both occur,
2. the set $A \cap B$ (A **intersection** B) is the event that A *and* B both occur,
3. the event A^c (A **complement**) is the event that A does *not* occur,
4. if $A \subset B$ (A is a **subset** of B) then event A is said to *imply* event B .

Two events A and B which have no outcomes in common, that is, $A \cap B = \emptyset$, are called **disjoint** events.

■ **Example 1.6 (Casting Two Dice)** Suppose we cast two dice consecutively. The sample space is $\Omega = \{(1, 1), (1, 2), \dots, (1, 6), (2, 1), \dots, (6, 6)\}$. Let $A = \{(6, 1), \dots, (6, 6)\}$ be the event that the first die is 6, and let $B = \{(1, 6), \dots, (6, 6)\}$ be the event that the second die is 6. Then $A \cap B = \{(6, 1), \dots, (6, 6)\} \cap \{(1, 6), \dots, (6, 6)\} = \{(6, 6)\}$ is the event that both die are 6. ■

It is often useful to depict events in a **Venn diagram**, such as in Figure 1.8

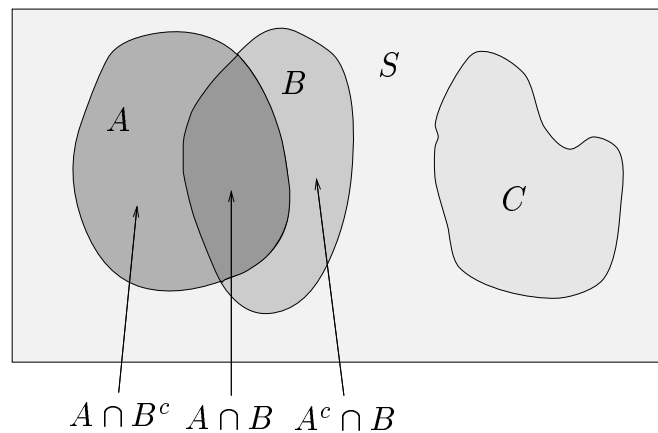


Figure 1.8: A Venn diagram

In this Venn diagram we see:

- (i) $A \cap C = \emptyset$ and therefore events A and C are disjoint.
- (ii) $(A \cap B^c) \cap (A^c \cap B) = \emptyset$ and hence events $A \cap B^c$ and $A^c \cap B$ are disjoint.

■ **Example 1.7 (System Reliability)** In Figure 1.9 three systems are depicted, each consisting of 3 unreliable components. The *series* system works if and only if (abbreviated as iff) *all* components work; the *parallel* system works iff *at least one* of the components works; and the 2-out-of-3 system works iff at least 2 out of 3 components work.

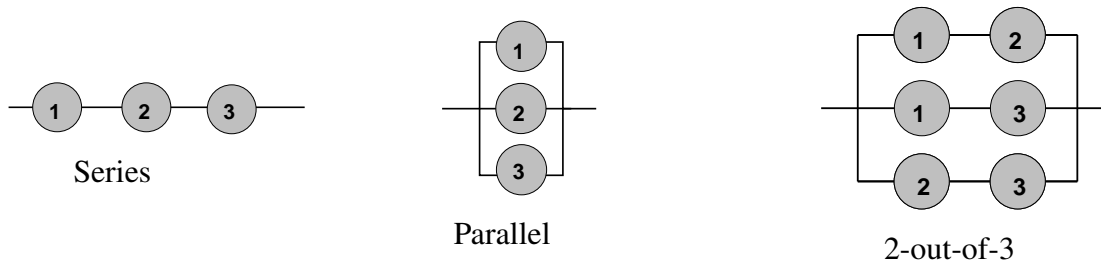


Figure 1.9: Three unreliable systems

Let A_i be the event that the i th component is functioning, $i = 1, 2, 3$; and let D_a, D_b, D_c be the events that respectively the series, parallel and 2-out-of-3 system is functioning. Then,

$$D_a = A_1 \cap A_2 \cap A_3 ,$$

and

$$D_b = A_1 \cup A_2 \cup A_3 .$$

Also,

$$\begin{aligned} D_c &= (A_1 \cap A_2 \cap A_3) \cup (A_1^c \cap A_2 \cap A_3) \cup (A_1 \cap A_2^c \cap A_3) \cup (A_1 \cap A_2 \cap A_3^c) \\ &= (A_1 \cap A_2) \cup (A_1 \cap A_3) \cup (A_2 \cap A_3) . \end{aligned}$$



Two useful results in the theory of sets are the following, due to **De Morgan**: If $\{A_i\}$ is a collection of events (sets) then

$$\left(\bigcup_i A_i \right)^c = \bigcap_i A_i^c \quad (1.1)$$

and

$$\left(\bigcap_i A_i \right)^c = \bigcup_i A_i^c . \quad (1.2)$$

This is easily proved via Venn diagrams. Note that if we interpret A_i as the event that a component works, then the left-hand side of (1.1) is the event that the corresponding parallel system is not working. The right hand is the event that at all components are not working. Clearly these two events are the same.

1.4 Probability

The third ingredient in the model for a random experiment is the specification of the probability of the events. It tells us how *likely* it is that a particular event will occur.

Definition 1.2: Probability Measure

A probability \mathbb{P} is a rule (function) which assigns a positive number to each event, and which satisfies the following **axioms**:

Axiom 1: $\mathbb{P}(A) \geq 0$.

Axiom 2: $\mathbb{P}(\Omega) = 1$.

Axiom 3: For any sequence A_1, A_2, \dots of *disjoint* events we have

$$\mathbb{P}\left(\bigcup_i A_i\right) = \sum_i \mathbb{P}(A_i). \quad (1.3)$$

Axiom 2 just states that the probability of the “certain” event Ω is 1. Property (1.3) is the *crucial* property of a probability, and is sometimes referred to as the **sum rule**. It just states that if an event can happen in a number of different ways *that cannot happen at the same time*, then the probability of this event is simply the sum of the probabilities of the composing events.

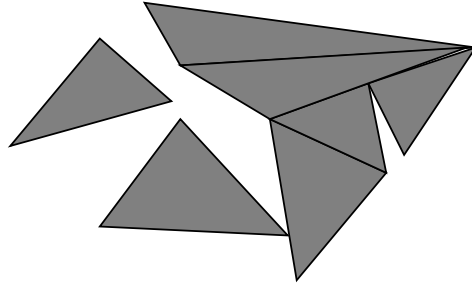


Figure 1.10: The probability measure has the same properties as the “area” measure: the total area of the triangles is the sum of the areas of the individual triangles.

Note that a probability rule \mathbb{P} has exactly the same properties as the common “area measure”. For example, the total area of the union of the triangles in Figure 1.10 is equal to the sum of the areas of the individual triangles. This is how you should interpret property (1.3). But instead of measuring areas, \mathbb{P} measures probabilities.

As a direct consequence of the axioms we have the following properties for \mathbb{P} .

Theorem 1.1: Properties of a Probability Measure

Let A and B be events. Then,

1. $\mathbb{P}(\emptyset) = 0$.
2. $A \subset B \implies \mathbb{P}(A) \leq \mathbb{P}(B)$.
3. $\mathbb{P}(A) \leq 1$.
4. $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$.
5. $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$.

Proof:

1. Since $\Omega = \Omega \cup \emptyset$, we have, by the sum rule, $\mathbb{P}(\Omega) = \mathbb{P}(\Omega) + \mathbb{P}(\emptyset)$, and therefore, by the second axiom, $1 = 1 + \mathbb{P}(\emptyset)$, from which it follows that $\mathbb{P}(\emptyset) = 0$.
2. If $A \subset B$, then $B = A \cup (B \cap A^c)$, where A and $B \cap A^c$ are disjoint. Hence, by the sum rule, $\mathbb{P}(B) = \mathbb{P}(A) + \mathbb{P}(B \cap A^c)$, which is (by the first axiom) greater than or equal to $\mathbb{P}(A)$.
3. This follows directly from property 2 and axiom 2, since $A \subset \Omega$.
4. $\Omega = A \cup A^c$, where A and A^c are disjoint. Hence, by the sum rule and axiom 2: $1 = \mathbb{P}(\Omega) = \mathbb{P}(A) + \mathbb{P}(A^c)$, and thus $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$.

5. Write $A \cup B$ as the disjoint union of A and $B \cap A^c$. Then, $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B \cap A^c)$. Also, $B = (A \cap B) \cup (B \cap A^c)$, so that $\mathbb{P}(B) = \mathbb{P}(A \cap B) + \mathbb{P}(B \cap A^c)$. Combining these two equations gives $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$.

□

We have now completed our model for a random experiment. It is up to the modeller to specify the sample space Ω and probability measure \mathbb{P} which most closely describes the actual experiment. This is not always as straightforward as it looks, and sometimes it is useful to model only certain *observations* in the experiment. This is where *random variables* come into play, and we will discuss these in the next chapter.

■ **Example 1.8 (Fair Die)** Consider the experiment where we throw a fair die. How should we define Ω and \mathbb{P} ?

Obviously, $\Omega = \{1, 2, \dots, 6\}$; and some common sense shows that we should define \mathbb{P} by

$$\mathbb{P}(A) = \frac{|A|}{6}, \quad A \subset \Omega,$$

where $|A|$ denotes the number of elements in set A . For example, the probability of getting an even number is $\mathbb{P}(\{2, 4, 6\}) = 3/6 = 1/2$. ■

In many applications the sample space is *countable*, i.e. $\Omega = \{a_1, a_2, \dots, a_n\}$ or $\Omega = \{a_1, a_2, \dots\}$. Such a sample space is called **discrete**.

The easiest way to specify a probability \mathbb{P} on a discrete sample space is to specify first the probability p_i of each **elementary event** $\{a_i\}$ and then to define

$$\mathbb{P}(A) = \sum_{i: a_i \in A} p_i, \quad \text{for all } A \subset \Omega.$$

This idea is graphically represented in Figure 1.11. Each element a_i in the sample is assigned a probability weight p_i represented by a black dot. To find the probability of the set A we have to sum up the weights of all the elements in A .

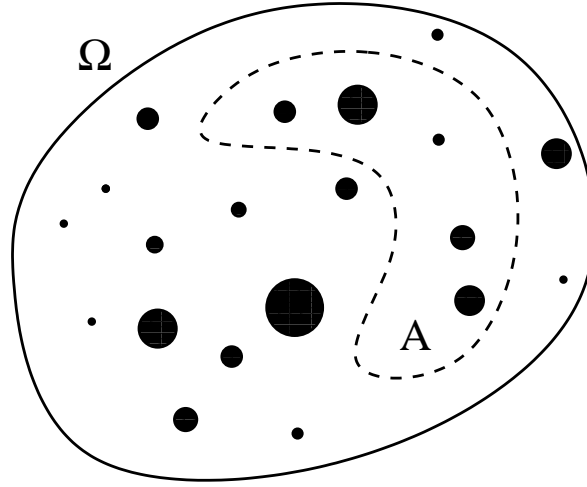


Figure 1.11: A discrete sample space

Again, it is up to the modeller to properly specify these probabilities. Fortunately, in many applications all elementary events are *equally likely*, and thus the probability of each elementary event is equal to 1 divided by the total number of elements in Ω . E.g., in Example 1.8 each elementary event has probability $1/6$.

Because the “equally likely” principle is so important, we formulate it as a theorem.

Theorem 1.2: Equilikely Principle

If Ω has a finite number of outcomes, and all are equally likely, then the probability of each event A is defined as

$$\mathbb{P}(A) = \frac{|A|}{|\Omega|} .$$

Thus for such sample spaces the calculation of probabilities reduces to *counting* the number of outcomes (in A and Ω).

When the sample space is not countable, for example $\Omega = \mathbb{R}_+$, it is said to be **continuous**.

■ **Example 1.9 (Drawing a Point in an Interval)** We draw at random a point in the interval $[0, 1]$. Each point is equally likely to be drawn. How do we specify the model for this experiment?

The sample space is obviously $\Omega = [0, 1]$, which is a continuous sample space. We cannot define \mathbb{P} via the elementary events $\{x\}$, $x \in [0, 1]$ because each of these events must have probability 0 (!). However we can define \mathbb{P} as follows: For each $0 \leq a \leq b \leq 1$, let

$$\mathbb{P}([a, b]) = b - a .$$

This completely specifies \mathbb{P} . In particular, we can find the probability that the point falls into any (sufficiently nice) set A as the *length* of that set. ■

1.5 Counting

Counting is not always easy. Let us first look at some examples:

1. A multiple choice form has 20 questions; each question has 3 choices. In how many possible ways can the exam be completed?
2. Consider a horse race with 8 horses. How many ways are there to gamble on the placings (1st, 2nd, 3rd).
3. Jessica has a collection of 20 CDs, she wants to take 3 of them to work. How many possibilities does she have?
4. How many different throws are possible with 3 dice?

To be able to comfortably solve a multitude of counting problems requires a lot of experience and *practice*, and even then, some counting problems remain exceedingly hard. Fortunately, many counting problems can be cast into the simple framework of drawing balls from an urn, see Figure 1.12.

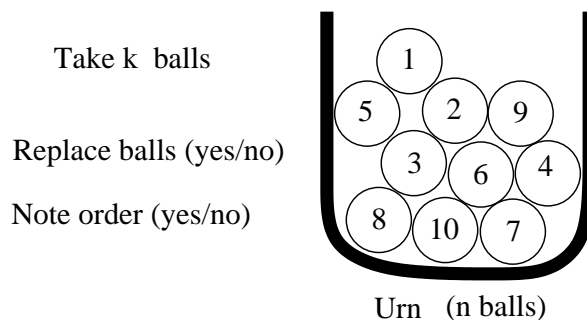


Figure 1.12: An urn with n balls

Consider an urn with n different balls, numbered $1, \dots, n$ from which k balls are drawn. This can be done in a number of different ways. First, the balls can be drawn one-by-one, or one could draw all the k balls at the same time. In the first case the **order** in which the balls are drawn can be noted, in the second case that is not possible. In the latter case we can (and will) still assume the balls are drawn one-by-one, but that the order is not noted. Second, once a ball is drawn, it can either be put back into the urn (after the number is recorded), or left out. This is called, respectively, drawing with and without **replacement**. All in all there

are 4 possible experiments: (ordered, with replacement), (ordered, without replacement), (unordered, without replacement) and (ordered, with replacement). The art is to recognise a seemingly unrelated counting problem as one of these four urn problems. For the 4 examples above we have the following

1. Example 1 above can be viewed as drawing 20 balls from an urn containing 3 balls, noting the order, and with replacement.
2. Example 2 is equivalent to drawing 3 balls from an urn containing 8 balls, noting the order, and without replacement.
3. In Example 3 we take 3 balls from an urn containing 20 balls, not noting the order, and without replacement
4. Finally, Example 4 is a case of drawing 3 balls from an urn containing 6 balls, not noting the order, and with replacement.

Before we proceed it is important to introduce a notation that reflects whether the outcomes/arrangements are ordered or not. In particular, we denote ordered arrangements by *vectors*, e.g., $(1, 2, 3) \neq (3, 2, 1)$, and unordered arrangements by *sets*, e.g., $\{1, 2, 3\} = \{3, 2, 1\}$. We now consider for each of the four cases how to count the number of arrangements. For simplicity we consider for each case how the counting works for $n = 4$ and $k = 3$, and then state the general situation.

Drawing with Replacement, Ordered

Here, after we draw each ball, note the number on the ball, and put the ball back. Let $n = 4, k = 3$. Some possible outcomes are $(1, 1, 1), (4, 1, 2), (2, 3, 2), (4, 2, 1), \dots$ To count how many such arrangements there are, we can reason as follows: we have three positions (\cdot, \cdot, \cdot) to fill in. Each position can have the numbers 1, 2, 3 or 4, so the total number of possibilities is $4 \times 4 \times 4 = 4^3 = 64$. This is illustrated via the following tree diagram:

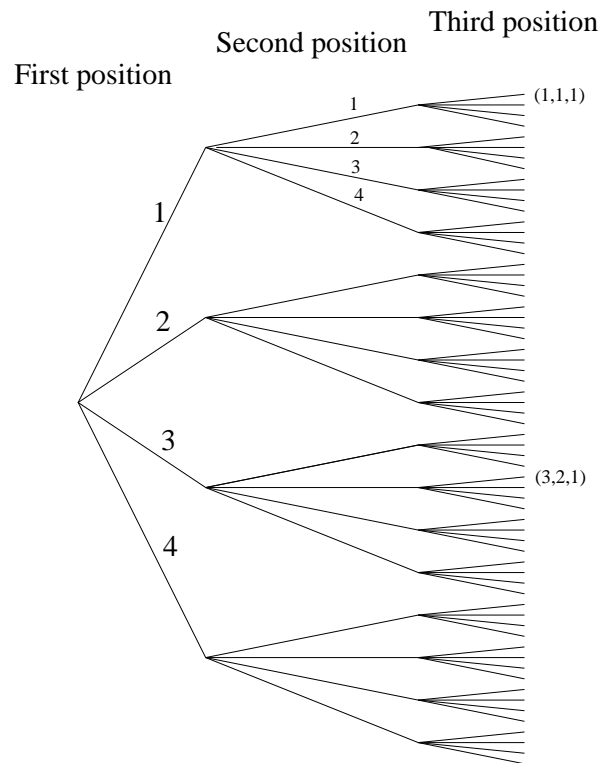


Figure 1.13: Combinatorial tree with duplication

For general n and k we can reason analogously to find:

The number of ordered arrangements of k numbers chosen from $\{1, \dots, n\}$, with replacement (repetition) is n^k .

Drawing Without Replacement, Ordered

Here we draw again k numbers (balls) from the set $\{1, 2, \dots, n\}$, and note the order, but now do not replace them. Let $n = 4$ and $k = 3$. Again there are 3 positions to fill (\cdot, \cdot, \cdot) , but now the numbers cannot be the same, e.g., $(1, 4, 2), (3, 2, 1)$, etc. Such an ordered arrangements called a **permutation** of size k from set $\{1, \dots, n\}$. (A permutation of $\{1, \dots, n\}$ of size n is simply called a permutation of $\{1, \dots, n\}$ (leaving out “of size n ”). For the 1st position we have 4 possibilities. Once the first position has been chosen, we have only 3 possibilities left for the second position. And after the first two positions have been chosen there are 2 positions left. So the number of arrangements is $4 \times 3 \times 2 = 24$ as illustrated in Figure 1.14, which is the same tree as in Figure 1.13, but with all “duplicate” branches removed.

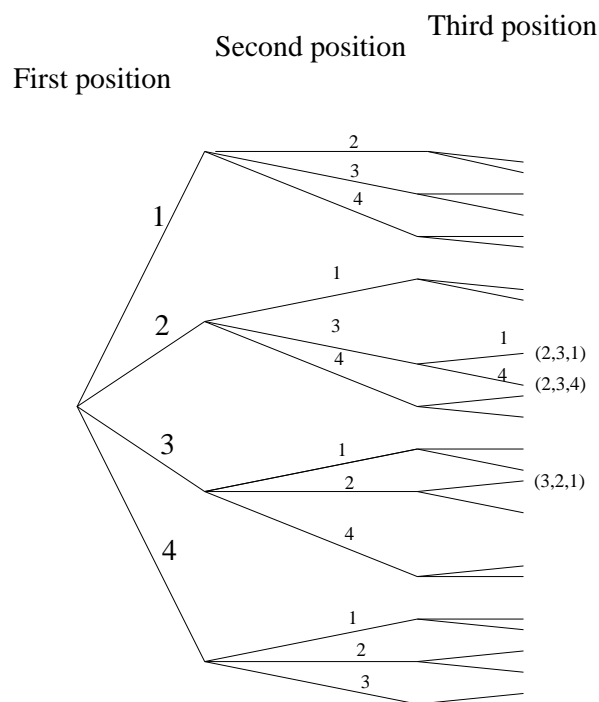


Figure 1.14: Combinatorial tree without duplication

For general n and k we have:

The number of permutations of size k from $\{1, \dots, n\}$ is ${}^n P_k = n(n-1) \cdots (n-k+1)$.

In particular, when $k = n$, we have that the number of ordered arrangements of n items is $n! = n(n-1)(n-2) \cdots 1$, where $n!$ is called **n -factorial**. Note that

$${}^n P_k = \frac{n!}{(n-k)!}.$$

Drawing Without Replacement, Unordered

This time we draw k numbers from $\{1, \dots, n\}$ but do not replace them (no replication), and do not note the order (so we could draw them in one grab). Taking again $n = 4$ and $k = 3$, a possible outcome is $\{1, 2, 4\}$, $\{1, 2, 3\}$, etc. If we noted the order, there would be ${}^n P_k$ outcomes, amongst which would be $(1, 2, 4), (1, 4, 2), (2, 1, 4), (2, 4, 1), (4, 1, 2)$ and $(4, 2, 1)$. Notice that these 6 permutations correspond to the single unordered arrangement $\{1, 2, 4\}$. Such unordered arrangements without replications are called **combinations** of size k from the set $\{1, \dots, n\}$.

To determine the number of combinations of size k simply need to divide nP_k by the number of permutations of k items, which is $k!$. Thus, in our example ($n = 4, k = 3$) there are $24/6 = 4$ possible combinations of size 3. In general we have:

The number of combinations of size k from the set $\{1, \dots, n\}$ is

$${}^nC_k = \binom{n}{k} = \frac{{}^nP_k}{k!} = \frac{n!}{(n-k)!k!}.$$

Note the two different notations for this number. We will use the second one.

Drawing With Replacement, Unordered

Taking $n = 4, k = 3$, possible outcomes are $\{3, 3, 4\}$, $\{1, 2, 4\}$, $\{2, 2, 2\}$, etc. The trick to solve this counting problem is to represent the outcomes in a different way, via an ordered vector (x_1, \dots, x_n) representing how many times an element in $\{1, \dots, 4\}$ occurs. For example, $\{3, 3, 4\}$ corresponds to $(0, 0, 2, 1)$ and $\{1, 2, 4\}$ corresponds to $(1, 1, 0, 1)$. Thus, we can count how many distinct vectors (x_1, \dots, x_n) there are such that the sum of the components is 3, and each x_i can take value 0, 1, 2 or 3. Another way of looking at this is to consider placing $k = 3$ balls into $n = 4$ urns, numbered 1, ..., 4. Then $(0, 0, 2, 1)$ means that the third urn has 2 balls and the fourth urn has 1 ball. One way to distribute the balls over the urns is to distribute $n - 1 = 3$ “separators” and $k = 3$ balls over $n - 1 + k = 6$ positions, as indicated in Figure 1.15.

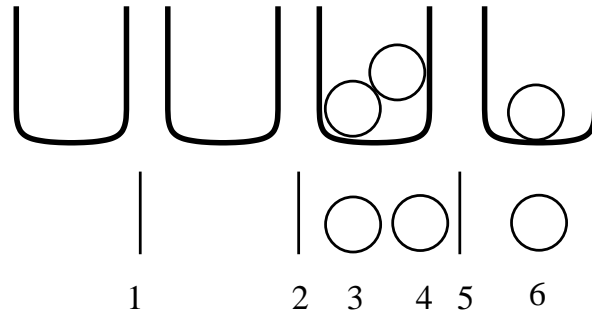


Figure 1.15: distributing k balls over n urns

The number of ways this can be done is equal to the number of ways k positions for the balls can be chosen out of $n - 1 + k$ positions, that is, $\binom{n+k-1}{k}$. We thus have:

The number of different sets $\{x_1, \dots, x_k\}$ with $x_i \in \{1, \dots, n\}, i = 1, \dots, k$ is

$$\binom{n+k-1}{k}.$$

Returning to our original four problems, we can now solve them easily:

1. The total number of ways the exam can be completed is $3^{20} = 3,486,784,401$.
2. The number of placings is ${}^8P_3 = 336$.
3. The number of possible combinations of CDs is $\binom{20}{3} = 1140$.
4. The number of different throws with three dice is $\binom{8}{3} = 56$.

More examples

Here are some more examples. Not all problems can be directly related to the 4 problems above. Some require additional reasoning. However, the counting principles remain the same.

1. In how many ways can the numbers $1, \dots, 5$ be arranged, such as 13524, 25134, etc?
Answer: $5! = 120$.
2. How many different arrangements are there of the numbers $1, 2, \dots, 7$, such that the first 3 numbers are 1,2,3 (in any order) and the last 4 numbers are 4,5,6,7 (in any order)?
Answer: $3! \times 4!$.
3. How many different arrangements are there of the word “arrange”, such as “aarrnge”, “arrngea”, etc?
Answer: Convert this into a ball drawing problem with 7 balls, numbered $1, \dots, 7$. Balls 1 and 2 correspond to ‘a’, balls 3 and 4 to ‘r’, ball 5 to ‘n’, ball 6 to ‘g’ and ball 7 to ‘e’. The total number of permutations of the numbers is $7!$. However, since, for example, $(1,2,3,4,5,6,7)$ is identical to $(2,1,3,4,5,6,7)$ (when substituting the letters back), we must divide $7!$ by $2! \times 2!$ to account for the 4 ways the two ‘a’s and ‘r’s can be arranged. So the answer is $7!/4 = 1260$.
4. An urn has 1000 balls, labelled 000, 001, \dots , 999. How many balls are there that have all number in ascending order (for example 047 and 489, but not 033 or 321)?
Answer: There are $10 \times 9 \times 8 = 720$ balls with different numbers. Each triple of numbers can be arranged in $3! = 6$ ways, and only one of these is in ascending order. So the total number of balls in ascending order is $720/6 = 120$.
5. In a group of 20 people each person has a different birthday. How many different arrangements of these birthdays are there (assuming each year has 365 days)?
Answer: ${}^{365}P_{20}$.

Once we've learned how to count, we can apply the equilikely principle to calculate probabilities:

1. What is the probability that out of a group of 40 people all have different birthdays?

Answer: Choosing the birthdays is like choosing 40 balls with replacement from an urn containing the balls $1, \dots, 365$. Thus, our sample space Ω consists of vectors of length 40, whose components are chosen from $\{1, \dots, 365\}$. There are $|\Omega| = 365^{40}$ such vectors possible, and all are *equally likely*. Let A be the event that all 40 people have different birthdays. Then, $|A| = {}^{365}P_{40} = 365!/325!$. It follows that $\mathbb{P}(A) = |A|/|\Omega| \approx 0.109$, so not very big!

2. What is the probability that in 10 tosses with a fair coin we get exactly 5 Heads and 5 Tails?

Answer: Here Ω consists of vectors of length 10 consisting of 1s (Heads) and 0s (Tails), so there are 2^{10} of them, and all are *equally likely*. Let A be the event of exactly 5 heads. We must count how many binary vectors there are with exactly 5 1s. This is equivalent to determining in how many ways the positions of the 5 1s can be chosen out of 10 positions, that is, $\binom{10}{5}$. Consequently, $\mathbb{P}(A) = \binom{10}{5}/2^{10} = 252/1024 \approx 0.25$.

3. We draw at random 13 cards from a full deck of cards. What is the probability that we draw 4 Hearts and 3 Diamonds?

Answer: Give the cards a number from 1 to 52. Suppose 1–13 is Hearts, 14–26 is Diamonds, etc. Ω consists of unordered sets of size 13, without repetition, e.g., $\{1, 2, \dots, 13\}$. There are $|\Omega| = \binom{52}{13}$ of these sets, and they are all equally likely. Let A be the event of 4 Hearts and 3 Diamonds. To form A we have to choose 4 Hearts out of 13, and 3 Diamonds out of 13, followed by 6 cards out of 26 Spade and Clubs. Thus, $|A| = \binom{13}{4} \times \binom{13}{3} \times \binom{26}{6}$. So that $\mathbb{P}(A) = |A|/|\Omega| \approx 0.074$.

1.6 Conditional probability and independence

How do probabilities change when we know some event $B \subset \Omega$ has occurred? Suppose B has occurred. Thus, we know that the outcome lies in B . Then A will occur if and only if $A \cap B$ occurs, and the relative chance of A occurring is therefore

$$\mathbb{P}(A \cap B)/\mathbb{P}(B).$$

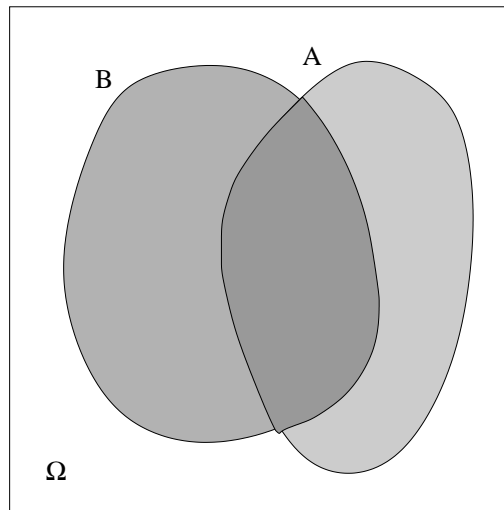


Figure 1.16: What is the conditional probability of A given that B occurs?

This leads to the following definition.

Definition 1.3: Conditional Probability

The **conditional probability** of A given B is:

$$\mathbb{P}(A | B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \quad (1.4)$$

■ **Example 1.10 (Throwing Two Dice)** We throw two dice. Given that the sum of the eyes is 10, what is the probability that one 6 is cast?

Let B be the event that the sum is 10,

$$B = \{(4, 6), (5, 5), (6, 4)\}.$$

Let A be the event that one 6 is cast,

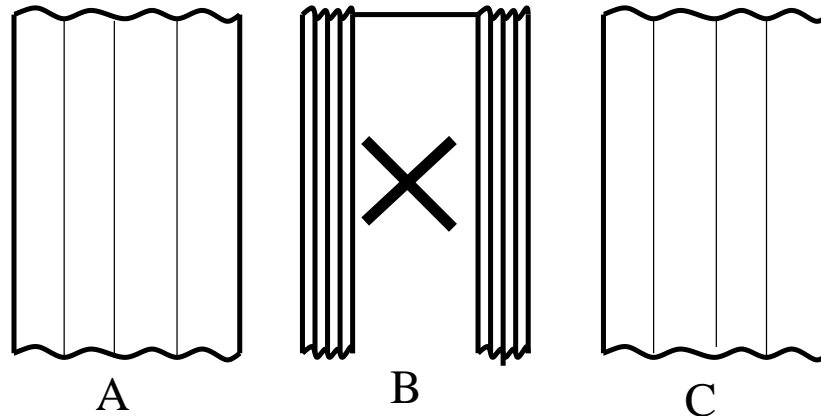
$$A = \{(1, 6), \dots, (5, 6), (6, 1), \dots, (6, 5)\}.$$

Then, $A \cap B = \{(4, 6), (6, 4)\}$. And, since all elementary events are equally likely, we have

$$\mathbb{P}(A | B) = \frac{2/36}{3/36} = \frac{2}{3}.$$

■

■ **Example 1.11 (Monty Hall Problem)** This is a nice application of conditional probability. Consider a quiz in which the final contestant is to choose a prize which is hidden behind one three curtains (A, B or C). Suppose without loss of generality that the contestant chooses curtain A. Now the quiz master (Monty Hall) always opens one of the other curtains: if the prize is behind B, Monty opens C, if the prize is behind C, Monty opens B, and if the prize is behind A, Monty opens B or C with equal probability, e.g., by tossing a coin (of course the contestant does not see Monty tossing the coin!).



Suppose, again without loss of generality that Monty opens curtain B. The contestant is now offered the opportunity to switch to curtain C. Should the contestant stay with his/her original choice (A) or switch to the other unopened curtain (C)?

Notice that the sample space consists here of 4 possible outcomes: Ac : The prize is behind A, and Monty opens C; Ab : The prize is behind A, and Monty opens B; Bc : The prize is behind B, and Monty opens C; and Cb : The prize is behind C, and Monty opens B. Let A , B , C be the events that the prize is behind A, B and C, respectively. Note that $A = \{Ac, Ab\}$, $B = \{Bc\}$ and $C = \{Cb\}$, see Figure 1.17.

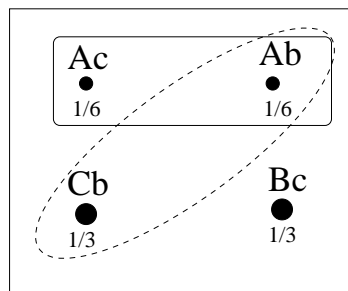


Figure 1.17: The sample space for the Monty Hall problem.

Now, obviously $\mathbb{P}(A) = \mathbb{P}(B) = \mathbb{P}(C)$, and since Ac and Ab are equally likely, we have $\mathbb{P}(\{Ab\}) = \mathbb{P}(\{Ac\}) = 1/6$. Monty opening curtain B means that we have information that

event $\{Ab, Cb\}$ has occurred. The probability that the prize is under A given this event, is therefore

$$\mathbb{P}(A | B \text{ is opened}) = \frac{\mathbb{P}(\{Ac, Ab\} \cap \{Ab, Cb\})}{\mathbb{P}(\{Ab, Cb\})} = \frac{\mathbb{P}(\{Ab\})}{\mathbb{P}(\{Ab, Cb\})} = \frac{1/6}{1/6 + 1/3} = \frac{1}{3}.$$

This is what we expected: the fact that Monty opens a curtain does not give us any extra information that the prize is behind A. So one could think that it doesn't matter to switch or not. But wait a minute! What about $\mathbb{P}(B | B \text{ is opened})$? Obviously this is 0 — opening curtain B means that we know that event B cannot occur. It follows then that $\mathbb{P}(C | B \text{ is opened})$ must be $2/3$, since a conditional probability behaves like any other probability and must satisfy axiom 2 (sum up to 1). Indeed,

$$\mathbb{P}(C | B \text{ is opened}) = \frac{\mathbb{P}(\{Cb\} \cap \{Ab, Cb\})}{\mathbb{P}(\{Ab, Cb\})} = \frac{\mathbb{P}(\{Cb\})}{\mathbb{P}(\{Ab, Cb\})} = \frac{1/3}{1/6 + 1/3} = \frac{2}{3}.$$

Hence, given the information that B is opened, it is twice as likely that the prize is under C than under A. Thus, the contestant should switch!

■

1.6.1 Product Rule

By the definition of conditional probability we have

$$\mathbb{P}(A \cap B) = \mathbb{P}(A) \mathbb{P}(B | A). \quad (1.5)$$

We can generalise this to n intersections $A_1 \cap A_2 \cap \dots \cap A_n$, which we abbreviate as $A_1 A_2 \dots A_n$. This gives the **product rule** of probability (also called *chain rule*).

Theorem 1.3: Product rule

Let A_1, \dots, A_n be a sequence of events with $\mathbb{P}(A_1 \dots A_{n-1}) > 0$. Then,

$$\mathbb{P}(A_1 \dots A_n) = \mathbb{P}(A_1) \mathbb{P}(A_2 | A_1) \mathbb{P}(A_3 | A_1 A_2) \dots \mathbb{P}(A_n | A_1 \dots A_{n-1}). \quad (1.6)$$

Proof: We only show the proof for 3 events, since the $n > 3$ event case follows similarly. By applying (1.4) to $\mathbb{P}(B | A)$ and $\mathbb{P}(C | A \cap B)$, the left-hand side of (1.6) is we have,

$$\mathbb{P}(A) \mathbb{P}(B | A) \mathbb{P}(C | A \cap B) = \mathbb{P}(A) \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)} \frac{\mathbb{P}(A \cap B \cap C)}{\mathbb{P}(A \cap B)} = \mathbb{P}(A \cap B \cap C),$$

which is equal to the left-hand size of (1.6). □

■ **Example 1.12 (Drawing Balls)** We draw consecutively 3 balls from a bowl with 5 white and 5 black balls, without putting them back. What is the probability that all balls will be black?

Solution: Let A_i be the event that the i th ball is black. We wish to find the probability of $A_1A_2A_3$, which by the product rule (1.6) is

$$\mathbb{P}(A_1)\mathbb{P}(A_2|A_1)\mathbb{P}(A_3|A_1A_2) = \frac{5}{10} \frac{4}{9} \frac{3}{8} = 0.083.$$

Note that this problem can also be easily solved by counting arguments, as in the previous section. ■

■ **Example 1.13 (Birthday Problem)** In Section 1.5 we derived by counting arguments that the probability that all people in a group of 40 have different birthdays is

$$\frac{365 \times 364 \times \cdots \times 326}{365 \times 365 \times \cdots \times 365} \approx 0.109. \quad (1.7)$$

We can derive this also via the product rule. Namely, let A_i be the event that the first i people have different birthdays, $i = 1, 2, \dots$. Note that $A_1 \supset A_2 \supset A_3 \supset \cdots$. Therefore $A_n = A_1 \cap A_2 \cap \cdots \cap A_n$, and thus by the product rule

$$\mathbb{P}(A_{40}) = \mathbb{P}(A_1)\mathbb{P}(A_2|A_1)\mathbb{P}(A_3|A_2) \cdots \mathbb{P}(A_{40}|A_{39}).$$

Now $\mathbb{P}(A_k|A_{k-1}) = (365 - k + 1)/365$ because given that the first $k - 1$ people have different birthdays, there are no duplicate birthdays if and only if the birthday of the k -th is chosen from the $365 - (k - 1)$ remaining birthdays. Thus, we obtain (1.7). More generally, the probability that n randomly selected people have different birthdays is

$$\mathbb{P}(A_n) = \frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \cdots \times \frac{365 - n + 1}{365}, \quad n \geq 1.$$

A graph of $\mathbb{P}(A_n)$ against n is given in Figure 1.18. Note that the probability $\mathbb{P}(A_n)$ rapidly decreases to zero. Indeed, for $n = 23$ the probability of having no duplicate birthdays is already less than $1/2$.

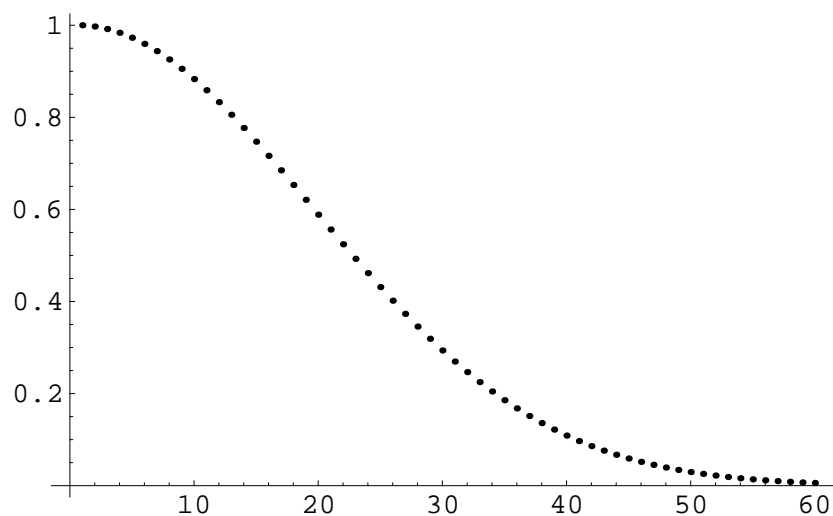


Figure 1.18: The probability of having no duplicate birthday in a group of n people, against n .



1.6.2 Law of Total Probability and Bayes' Rule

Suppose B_1, B_2, \dots, B_n is a **partition** of Ω . That is, B_1, B_2, \dots, B_n are disjoint and their union is Ω , see Figure 1.19

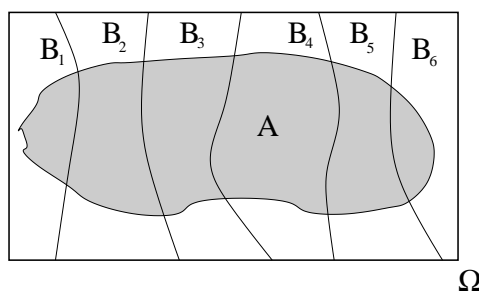


Figure 1.19: A partition of the sample space

Then, by the sum rule, $\mathbb{P}(A) = \sum_{i=1}^n \mathbb{P}(A \cap B_i)$ and hence, by the definition of conditional probability we have

$$\boxed{\mathbb{P}(A) = \sum_{i=1}^n \mathbb{P}(A|B_i) \mathbb{P}(B_i)}$$

This is called the **law of total probability**.

Combining the Law of Total Probability with the definition of conditional probability gives **Bayes' Rule**:

$$\mathbb{P}(B_j|A) = \frac{\mathbb{P}(A|B_j)\mathbb{P}(B_j)}{\sum_{i=1}^n \mathbb{P}(A|B_i)\mathbb{P}(B_i)}$$

■ **Example 1.14 (Three Factories)** A company has three factories (1, 2 and 3) that produce the same chip, each producing 15%, 35% and 50% of the total production. The probability of a defective chip at 1, 2, 3 is 0.01, 0.05, 0.02, respectively. Suppose someone shows us a defective chip. What is the probability that this chip comes from factory 1?

Let B_i denote the event that the chip is produced by factory i . The $\{B_i\}$ form a partition of Ω . Let A denote the event that the chip is faulty. By Bayes' rule,

$$\mathbb{P}(B_1|A) = \frac{0.15 \times 0.01}{0.15 \times 0.01 + 0.35 \times 0.05 + 0.5 \times 0.02} = 0.052 .$$

■

1.6.3 Independence

Independence is a very important concept in probability and statistics. Loosely speaking it models the *lack of information* between events. We say A and B are *independent* if the knowledge that A has occurred does not change the *probability* that B occurs. That is

$$A, B \text{ independent} \Leftrightarrow \mathbb{P}(A|B) = \mathbb{P}(A)$$

Since $\mathbb{P}(A|B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$ an alternative definition of independence is:

$$A, B \text{ independent} \Leftrightarrow \mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$$

This definition covers the case $B = \emptyset$ (empty set). We can extend the definition to arbitrarily many events:

Definition 1.4: Independent Events

The events A_1, A_2, \dots , are said to be **(mutually) independent** if for any k and any choice of distinct indices i_1, \dots, i_k ,

$$\mathbb{P}(A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_k}) = \mathbb{P}(A_{i_1})\mathbb{P}(A_{i_2}) \dots \mathbb{P}(A_{i_k}) .$$

■ **Remark 1.1 (Independence as Model Assumption)** In most cases independence of events is a **model assumption**. That is, we assume that there exists a \mathbb{P} such that certain events are independent. ■

■ **Example 1.15 (A Coin Toss Experiment and the Binomial Law)** We flip a coin n times. We can write the sample space as the set of binary n -tuples:

$$\Omega = \{(0, \dots, 0), \dots, (1, \dots, 1)\}.$$

Here 0 represent Tails and 1 represents Heads. For example, the outcome $(0, 1, 0, 1, \dots)$ means that the first time Tails is thrown, the second time Heads, the third times Tails, the fourth time Heads, etc.

How should we define \mathbb{P} ? Let A_i denote the event of Heads during the i th throw, $i = 1, \dots, n$. Then, \mathbb{P} should be such that the events A_1, \dots, A_n are *independent*. And, moreover, $\mathbb{P}(A_i)$ should be the same for all i . We don't know whether the coin is fair or not, but we can call this probability p ($0 \leq p \leq 1$).

These two rules completely specify \mathbb{P} . For example, the probability that the first k throws are Heads and the last $n - k$ are Tails is

$$\begin{aligned} \mathbb{P}(\{(1, 1, \dots, 1, 0, 0, \dots, 0)\}) &= \mathbb{P}(A_1) \cdots \mathbb{P}(A_k) \cdots \mathbb{P}(A_{k+1}^c) \cdots \mathbb{P}(A_n^c) \\ &= p^k (1 - p)^{n-k}. \end{aligned}$$

Also, let B_k be the event that there are k Heads in total. The probability of this event is the sum the probabilities of elementary events $\{(x_1, \dots, x_n)\}$ such that $x_1 + \dots + x_n = k$. Each of these events has probability $p^k (1 - p)^{n-k}$, and there are $\binom{n}{k}$ of these. Thus,

$$\mathbb{P}(B_k) = \binom{n}{k} p^k (1 - p)^{n-k}, \quad k = 0, 1, \dots, n.$$

We have thus discovered the **binomial distribution**. ■

■ **Example 1.16 (Geometric Law)** There is another important law associated with the coin flip experiment. Suppose we flip the coin until Heads appears for the first time. Let C_k be the event that Heads appears for the first time at the k -th toss, $k = 1, 2, \dots$. Then, using the same events $\{A_i\}$ as in the previous example, we can write

$$C_k = A_1^c \cap A_2^c \cap \dots \cap A_{k-1}^c \cap A_k,$$

so that with the product law and the mutual independence of A_1^c, \dots, A_k we have the **geometric law**:

$$\begin{aligned} \mathbb{P}(C_k) &= \mathbb{P}(A_1^c) \cdots \mathbb{P}(A_{k-1}^c) \mathbb{P}(A_k) \\ &= \underbrace{(1 - p) \cdots (1 - p)}_{k-1 \text{ times}} p = (1 - p)^{k-1} p. \end{aligned}$$

■

RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS

Specifying a model for a random experiment via a complete description of Ω and \mathbb{P} may not always be convenient or necessary. In practice we are only interested in various *observations* (i.e., numerical measurements) of the experiment. We include these into our modelling process via the introduction of *random variables*.

2.1 Random Variables

Formally a **random variable** is a *function* from the sample space Ω to \mathbb{R} . Here is a concrete example.

■ **Example 2.1 (Sum of two dice)** Suppose we toss two fair dice and note their sum. If we throw the dice one-by-one and observe each throw, the sample space is $\Omega = \{(1, 1), \dots, (6, 6)\}$. The function X , defined by $X(i, j) = i + j$, is a random variable, which maps the outcome (i, j) to the sum $i + j$, as depicted in Figure 2.1. Note that all the outcomes in the “encircled” set are mapped to 8. This is the set of all outcomes whose sum is 8. A natural notation for this set is to write $\{X = 8\}$. Since this set has 5 outcomes, and all outcomes in Ω are equally likely, we have

$$\mathbb{P}(\{X = 8\}) = \frac{5}{36}.$$

This notation is very suggestive and convenient. From a non-mathematical viewpoint we can interpret X as a “random” variable. That is a variable that can take on several values, with

certain probabilities. In particular it is not difficult to check that

$$\mathbb{P}(\{X = x\}) = \frac{6 - |7 - x|}{36}, \quad x = 2, \dots, 12.$$

■

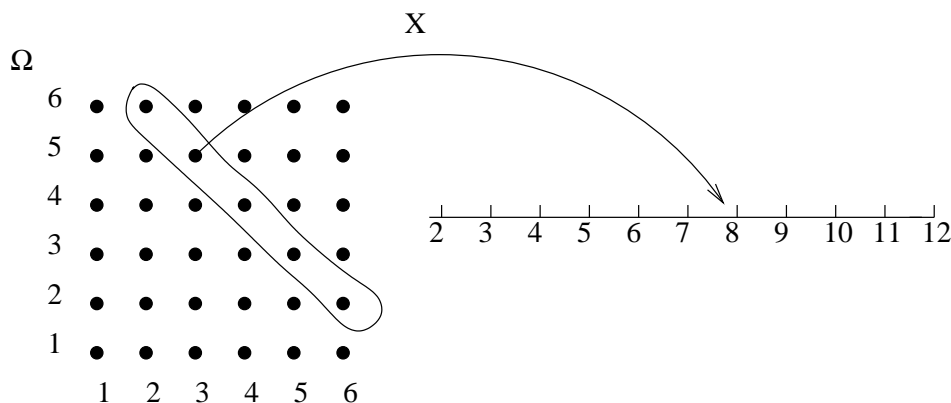


Figure 2.1: A random variable representing the sum of two dice

Although random variables are, mathematically speaking, *functions*, it is often convenient to view random variables as observations of a random experiment that has not yet been carried out. In other words, a random variable is considered as a measurement that becomes available once we carry out the random experiment, e.g., *tomorrow*. However, all the *thinking* about the experiment and measurements can be done *today*. For example, we can specify today exactly the probabilities pertaining to the random variables.

We usually denote random variables with *capital* letters from the last part of the alphabet, e.g. X, X_1, X_2, \dots, Y, Z . Random variables allow us to use natural and intuitive notations for certain events, such as $\{X = 10\}$, $\{X > 1000\}$, $\{\max(X, Y) \leq Z\}$, etc.

■ **Example 2.2 (Coin Flips)** We flip a coin n times. In Example 1.15 we can find a probability model for this random experiment. But suppose we are not interested in the complete outcome, e.g., $(0, 1, 0, 1, 1, 0, \dots)$, but only in the total number of heads (1s). Let X be the total number of heads. X is a “random variable” in the true sense of the word: X could lie anywhere between 0 and n . What we are interested in, however, is the *probability* that X takes certain values. That is, we are interested in the **probability distribution** of X . Example 1.15 now suggests that

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}, \quad k = 0, 1, \dots, n. \quad (2.1)$$

This contains all the information about X that we could possibly wish to know. Example 2.1 suggests how we can justify this mathematically. Define X as the function that assigns to each

outcome $\omega = (x_1, \dots, x_n)$ the number $x_1 + \dots + x_n$. Then clearly X is a random variable in mathematical terms (that is, a function). Moreover, the event that there are exactly k Heads in n throws can be written as

$$\{\omega \in \Omega : X(\omega) = k\}.$$

If we abbreviate this to $\{X = k\}$, and further abbreviate $\mathbb{P}(\{X = k\})$ to $\mathbb{P}(X = k)$, then we obtain exactly (2.1).



We give some more examples of random variables without specifying the sample space.

1. The number of defective transistors out of 100 inspected ones,
2. the number of bugs in a computer program,
3. the amount of rain in Brisbane in June,
4. the amount of time needed for an operation.

The set of all possible values a random variable X can take is called the **range** of X . We further distinguish between discrete and continuous random variables:

Discrete random variables can only take *isolated* values.

For example: a count can only take non-negative integer values.

Continuous random variables can take values in an *interval*.

For example: rainfall measurements, lifetimes of components, lengths, ... are (at least in principle) continuous.

2.2 Probability Distribution

Let X be a random variable. We would like to specify the probabilities of events such as $\{X = x\}$ and $\{a \leq X \leq b\}$.

If we can specify all probabilities involving X , we say that we have specified the **probability distribution** of X .

One way to specify the probability distribution is to give the probabilities of all events of the form $\{X \leq x\}$, $x \in \mathbb{R}$. This leads to the following definition.

Definition 2.1: Cumulative distribution function

The **cumulative distribution function** (cdf) of a random variable X is the function $F : \mathbb{R} \rightarrow [0, 1]$ defined by

$$F(x) := \mathbb{P}(X \leq x), \quad x \in \mathbb{R}.$$

Note that above we should have written $\mathbb{P}(\{X \leq x\})$ instead of $\mathbb{P}(X \leq x)$. From now on we will use this type of abbreviation throughout the course. In Figure 2.2 the graph of a cdf is depicted.

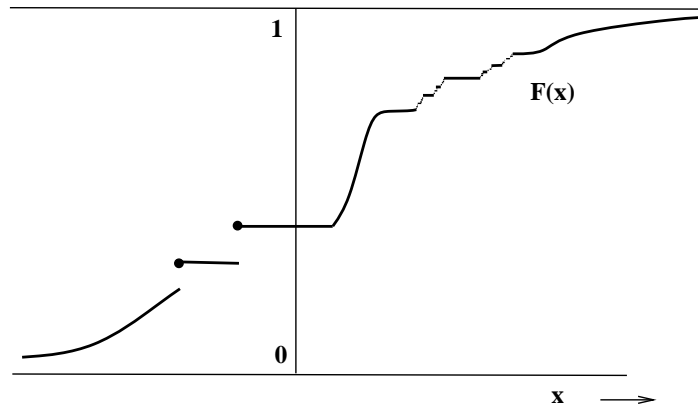


Figure 2.2: A cumulative distribution function

The following properties for F are a direct consequence of the three Axioms for \mathbb{P} .

1. F is right-continuous: $\lim_{h \downarrow 0} F(x + h) = F(x)$,
2. F is increasing: $x \leq y \Rightarrow F(x) \leq F(y)$,
3. $0 \leq F(x) \leq 1$.

Proof: We will prove (1) in STAT3004. For (2), suppose $x \leq y$ and define $A = \{X \leq x\}$ and $B = \{X \leq y\}$. Then, obviously, $A \subset B$ (for example if $\{X \leq 3\}$ then this implies $\{X \leq 4\}$). Thus, by (2) on page 17, $\mathbb{P}(A) \leq \mathbb{P}(B)$, which proves (2). Property (3) follows directly from the fact that $0 \leq \mathbb{P}(A) \leq 1$ for any event A — and hence in particular for $A = \{X \leq x\}$. \square

Any function F with the above properties can be used to specify the distribution of a random variable X . Suppose that X has cdf F . Then the probability that X takes a value in the interval $(a, b]$ (excluding a , including b) is given by

$$\mathbb{P}(a < X \leq b) = F(b) - F(a).$$

Namely, $\mathbb{P}(X \leq b) = \mathbb{P}(\{X \leq a\} \cup \{a < X \leq b\})$, where the events $\{X \leq a\}$ and $\{a < X \leq b\}$ are disjoint. Thus, by the sum rule: $F(b) = F(a) + \mathbb{P}(a < X \leq b)$, which leads to the result above. Note however that

$$\begin{aligned}\mathbb{P}(a \leq X \leq b) &= F(b) - F(a) + \mathbb{P}(X = a) \\ &= F(b) - F(a) + F(a) - \lim_{h \downarrow 0} F(a - h) \\ &= F(b) - \lim_{h \downarrow 0} F(a - h).\end{aligned}$$

In practice we will specify the distribution of a random variable in a different way, whereby we make the distinction between *discrete* and *continuous* random variables.

2.2.1 Discrete Distributions

Definition 2.2: Discrete Distribution

We say that X has a **discrete** distribution if X is a discrete random variable. In particular, for some finite or countable set of values x_1, x_2, \dots we have $\mathbb{P}(X = x_i) > 0$, $i = 1, 2, \dots$ and $\sum_i \mathbb{P}(X = x_i) = 1$. We define the **probability mass function** (pmf) f of X by $f(x) = \mathbb{P}(X = x)$. We sometimes write f_X instead of f to stress that the pmf refers to the random variable X .

The easiest way to specify the distribution of a discrete random variable is to specify its pmf. Indeed, by the sum rule, if we know $f(x)$ for all x , then we can calculate all possible probabilities involving X . Namely,

$$\boxed{\mathbb{P}(X \in B) = \sum_{x \in B} f(x)} \quad (2.2)$$

for any subset B of the range of X .

■ **Example 2.3 (Tossing a Die)** Toss a die and let X be its face value. X is discrete with range $\{1, 2, 3, 4, 5, 6\}$. If the die is fair the probability mass function is given by

x	1	2	3	4	5	6	Σ
$f(x)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	1

■

■ **Example 2.4 (Largest Value of Two Dice)** Toss two dice and let X be the largest face value showing. The pmf of X can be found to satisfy

x	1	2	3	4	5	6	Σ
$f(x)$	$\frac{1}{36}$	$\frac{3}{36}$	$\frac{5}{36}$	$\frac{7}{36}$	$\frac{9}{36}$	$\frac{11}{36}$	1

The probability that the maximum is at least 3 follows directly from the table: $\mathbb{P}(X \geq 3) = \sum_{x=3}^6 f(x) = 32/36 = 8/9$. ■

2.2.2 Continuous Distributions

Definition 2.3: Continuous Distribution

A random variable X is said to have a **continuous distribution** if X is a continuous random variable for which there exists a *positive* function f with *total integral 1*, such that for all a, b

$$\mathbb{P}(a < X \leq b) = F(b) - F(a) = \int_a^b f(u) du. \quad (2.3)$$

The function f is called the **probability density function** (pdf) of X .

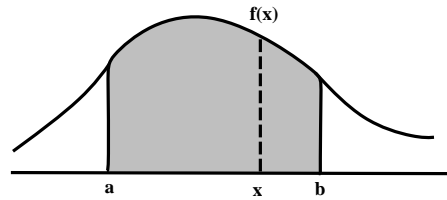


Figure 2.3: Probability density function (pdf)

Note that the corresponding cdf F is simply a *primitive* (also called anti-derivative) of the pdf f . In particular,

$$F(x) = \mathbb{P}(X \leq x) = \int_{-\infty}^x f(u) du.$$

Moreover, if a pdf f exists, then f is the *derivative* of the cdf F :

$$f(x) = \frac{d}{dx} F(x) = F'(x).$$

We can *interpret* $f(x)$ as the “density” that $X = x$. More precisely,

$$\mathbb{P}(x \leq X \leq x + h) = \int_x^{x+h} f(u) du \approx h f(x).$$

However, it is important to realise that $f(x)$ is *not a probability* — it is a probability *density*. In particular, if X is a continuous random variable, then $\mathbb{P}(X = x) = 0$, for all x . Note that this also justifies using $\mathbb{P}(x \leq X \leq x + h)$ above instead of $\mathbb{P}(x < X \leq x + h)$. Although we will use the same notation f for probability mass function (in the discrete case) and probability density function (in the continuous case), it is crucial to understand the difference between the two cases.

■ **Example 2.5 (Drawing Uniformly From an Interval)** Draw a random number from the interval of real numbers $[0, 2]$. Each number is equally possible. Let X represent the number. What is the probability density function f and the cdf F of X ?

Solution: Take an $x \in [0, 2]$. Drawing a number X “uniformly” in $[0, 2]$ means that $\mathbb{P}(X \leq x) = x/2$, for all such x . In particular, the cdf of X satisfies:

$$F(x) = \begin{cases} 0 & x < 0, \\ x/2 & 0 \leq x \leq 2, \\ 1 & x > 2. \end{cases}$$

By differentiating F we find

$$f(x) = \begin{cases} 1/2 & 0 \leq x \leq 2, \\ 0 & \text{otherwise} \end{cases}$$

Note that this density is *constant* on the interval $[0, 2]$ (and zero elsewhere), reflecting that each point in $[0, 2]$ is equally likely. Note also that we have modelled this random experiment using a continuous random variable and its pdf (and cdf). Compare this with the more “direct” model of Example 1.9. ■

Describing an experiment via a random variable and its pdf, pmf or cdf seems much easier than describing the experiment by explicitly specifying the probability space; that is, by specifying (Ω, \mathbb{P}) . In fact, we have not used a specific probability space in the above examples.

2.3 Expectation

Although all the probability information of a random variable is contained in its cdf (or pmf for discrete random variables and pdf for continuous random variables), it is often useful to consider various numerical characteristics of that random variable. One such number is the *expectation* of a random variable; it is a sort of “weighted average” of the values that X can take. Here is a more precise definition.

Definition 2.4: Expectation of a Discrete Random Variable

Let X be a *discrete* random variable with pmf f . The **expectation** (or expected value) of X , denoted by $\mathbb{E}X$, is defined by

$$\mathbb{E}X = \sum_x x \mathbb{P}(X = x) = \sum_x x f(x) .$$

The expectation of X is sometimes written as μ_X .

■ **Example 2.6 (Tossing a Fair Die)** Find $\mathbb{E}X$ if X is the outcome of a toss of a fair die.

Since $\mathbb{P}(X = 1) = \dots = \mathbb{P}(X = 6) = 1/6$, we have

$$\mathbb{E}X = 1\left(\frac{1}{6}\right) + 2\left(\frac{1}{6}\right) + \dots + 6\left(\frac{1}{6}\right) = \frac{7}{2} .$$

Note: $\mathbb{E}X$ is not necessarily a possible outcome of the random experiment as in the previous example. ■

One way to interpret the expectation is as a type of “expected profit”. Specifically, suppose we play a game where you throw two dice, and I pay you out, in dollars, the sum of the dice, X say. However, to enter the game you must pay me d dollars. You can play the game as many times as you like. What would be a “fair” amount for d ? The answer is

$$\begin{aligned} d = \mathbb{E}X &= 2\mathbb{P}(X = 2) + 3\mathbb{P}(X = 3) + \dots + 12\mathbb{P}(X = 12) \\ &= 2\frac{1}{36} + 3\frac{2}{36} + \dots + 12\frac{1}{36} = 7 . \end{aligned}$$

Namely, in the long run the fractions of times the sum is equal to 2, 3, 4, ... are $\frac{1}{36}, \frac{2}{36}, \frac{3}{36}, \dots$, so the average pay-out per game is the weighted sum of 2, 3, 4, ... with the weights being the probabilities/fractions. Thus the game is “fair” if the average profit (pay-out - d) is zero.

Another interpretation of expectation is as a *centre of mass*. Imagine that point masses with weights p_1, p_2, \dots, p_n are placed at positions x_1, x_2, \dots, x_n on the real line, see Figure 2.4.

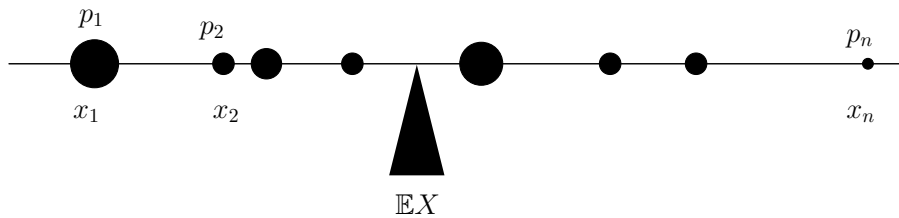


Figure 2.4: The expectation as a centre of mass

Then the centre of mass, the place where we can “balance” the weights, is

$$\text{centre of mass} = x_1 p_1 + \cdots + x_n p_n,$$

which is exactly the expectation of the discrete variable X taking values x_1, \dots, x_n with probabilities p_1, \dots, p_n . An obvious consequence of this interpretation is that for a *symmetric* probability mass function the expectation is equal to the symmetry point (provided the expectation exists). In particular, suppose $f(c + y) = f(c - y)$ for all y , then

$$\begin{aligned} \mathbb{E}X &= c f(c) + \sum_{x>c} x f(x) + \sum_{x<c} x f(x) \\ &= c f(c) + \sum_{y>0} (c + y) f(c + y) + \sum_{y>0} (c - y) f(c - y) \\ &= c f(c) + \sum_{y>0} c f(c + y) + c \sum_{y>0} f(c - y) \\ &= c \sum_x f(x) = c \end{aligned}$$

For continuous random variables we can define the expectation in a similar way:

Definition 2.5: Expectation of a Continuous Random Variable

Let X be a *continuous* random variable with pdf f . The **expectation** (or expected value) of X , denoted by $\mathbb{E}X$, is defined by

$$\mathbb{E}X = \int_x x f(x) dx .$$

If X is a random variable, then a function of X , such as X^2 or $\sin(X)$ is also a random variable. The following theorem is not so difficult to prove, and is entirely “obvious”: the expected value of a function of X is the weighted average of the values that this function can take.

Theorem 2.1

If X is *discrete* with pmf f , then for any real-valued function g

$$\mathbb{E} g(X) = \sum_x g(x) f(x) .$$

Similarly, if X is *continuous* with pdf f , then

$$\mathbb{E} g(X) = \int_{-\infty}^{\infty} g(x) f(x) dx .$$

Proof: We prove it for the discrete case only. Let $Y = g(X)$, where X is a discrete random variable with pmf f_X , and g is a function. Y is again a random variable. The pmf of Y , f_Y satisfies

$$f_Y(y) = \mathbb{P}(Y = y) = \mathbb{P}(g(X) = y) = \sum_{x: g(x)=y} \mathbb{P}(X = x) = \sum_{x: g(x)=y} f_X(x) .$$

Thus, the expectation of Y is

$$\mathbb{E}Y = \sum_y y f_Y(y) = \sum_y y \sum_{x: g(x)=y} f_X(x) = \sum_y \sum_{x: g(x)=y} y f_X(x) = \sum_x g(x) f_X(x)$$

□

■ **Example 2.7 (Expected Square Value for a Fair Die)** Find $\mathbb{E}X^2$ if X is the outcome of the toss of a fair die. We have

$$\mathbb{E}X^2 = 1^2 \frac{1}{6} + 2^2 \frac{1}{6} + 3^2 \frac{1}{6} + \cdots + 6^2 \frac{1}{6} = \frac{91}{6} .$$

■

An important consequence of Theorem 2.1 is that the expectation is “linear”. More precisely, for any real numbers a and b , and functions g and h

1. $\mathbb{E}(aX + b) = a\mathbb{E}X + b$.
2. $\mathbb{E}(g(X) + h(X)) = \mathbb{E}g(X) + \mathbb{E}h(X)$.

Proof: Suppose X has pmf f . Then 1. follows (in the discrete case) from

$$\mathbb{E}(aX + b) = \sum_x (ax + b)f(x) = a \sum_x x f(x) + b \sum_x f(x) = a\mathbb{E}X + b .$$

Similarly, 2. follows from

$$\begin{aligned} \mathbb{E}(g(X) + h(X)) &= \sum_x (g(x) + h(x))f(x) = \sum_x g(x)f(x) + \sum_x h(x)f(x) \\ &= \mathbb{E}g(X) + \mathbb{E}h(X) . \end{aligned}$$

The continuous case is proved analogously, by replacing the sum with an integral. □

Another useful number about (the distribution of) X is the *variance* of X . This number, sometimes written as σ_X^2 , measures the *spread* or dispersion of the distribution of X .

Definition 2.6: Variance

The **variance** of a random variable X , denoted by $\text{Var}(X)$ is defined by

$$\text{Var}(X) = \mathbb{E}(X - \mathbb{E}X)^2 .$$

The square root of the variance is called the **standard deviation**. The number $\mathbb{E}X^r$ is called the r th **moment** of X .

The following important properties for variance hold for discrete or continuous random variables and follow easily from the definitions of expectation and variance.

1. $\text{Var}(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2$
2. $\text{Var}(aX + b) = a^2 \text{Var}(X)$

Proof: Write $\mathbb{E}X = \mu$, so that $\text{Var}(X) = \mathbb{E}(X - \mu)^2 = \mathbb{E}(X^2 - 2\mu X + \mu^2)$. By the linearity of the expectation, the last expectation is equal to the sum $\mathbb{E}(X^2) - 2\mu\mathbb{E}X + \mu^2 = \mathbb{E}X^2 - \mu^2$, which proves 1. To prove 2, note first that the expectation of $aX + b$ is equal to $a\mu + b$. Thus,

$$\text{Var}(aX + b) = \mathbb{E}(aX + b - (a\mu + b))^2 = \mathbb{E}(a^2(X - \mu)^2) = a^2\text{Var}(X) .$$

□

2.4 Transforms

Many calculations and manipulations involving probability distributions are facilitated by the use of *transforms*. We discuss here a number of such transforms.

Definition 2.7: Probability Generating Function

Let X be a *non-negative* and *integer-valued* random variable. The **probability generating function** (PGF) of X is the function $G : [0, 1] \rightarrow [0, 1]$ defined by

$$G(z) := \mathbb{E} z^X = \sum_{x=0}^{\infty} z^x \mathbb{P}(X = x) .$$

■ **Example 2.8 (PGF of the Poisson Distribution)** Let X have pmf f given by

$$f(x) = e^{-\lambda} \frac{\lambda^x}{x!}, \quad x = 0, 1, 2, \dots$$

We will shortly introduce this as the *Poisson* distribution, but for now this is not important. The PGF of X is given by

$$\begin{aligned} G(z) &= \sum_{x=0}^{\infty} z^x e^{-\lambda} \frac{\lambda^x}{x!} \\ &= e^{-\lambda} \sum_{x=0}^{\infty} \frac{(z\lambda)^x}{x!} \\ &= e^{-\lambda} e^{z\lambda} = e^{-\lambda(1-z)}. \end{aligned}$$

■

Knowing only the PGF of X , we can easily obtain the pmf:

$$\mathbb{P}(X = x) = \frac{1}{x!} \frac{d^x}{dz^x} G(z) \Big|_{z=0}, \quad x = 1, 2, \dots$$

Proof: By definition

$$G(z) = z^0 \mathbb{P}(X = 0) + z^1 \mathbb{P}(X = 1) + z^2 \mathbb{P}(X = 2) + \dots$$

Substituting $z = 0$ gives $G(0) = \mathbb{P}(X = 0)$. If we differentiate $G(z)$ once, then

$$G'(z) = \mathbb{P}(X = 1) + 2z \mathbb{P}(X = 2) + 3z^2 \mathbb{P}(X = 3) + \dots$$

Thus, $G'(0) = \mathbb{P}(X = 1)$. Differentiating again, we see that $G''(0) = 2 \mathbb{P}(X = 2)$, and in general the x -th derivative of G at zero is $G^{(x)}(0) = x! \mathbb{P}(X = x)$, which completes the proof. \square

Thus we have the **uniqueness** property: two pmf's are the same if and only if their PGFs are the same.

Another useful property of the PGF is that we can obtain the moments of X by differentiating G and evaluating it at $z = 1$. Differentiating $G(z)$ w.r.t. z gives

$$\begin{aligned} G'(z) &= \frac{d \mathbb{E} z^X}{dz} = \mathbb{E} X z^{X-1}, \\ G''(z) &= \frac{d \mathbb{E} X z^{X-1}}{dz} = \mathbb{E} X(X-1) z^{X-2}, \\ G'''(z) &= \mathbb{E} X(X-1)(X-2) z^{X-3}, \end{aligned}$$

et cetera. If you're not convinced, write out the expectation as a sum, and use the fact that the derivative of the sum is equal to the sum of the derivatives (although we need a little care when dealing with infinite sums).

In particular,

$$\mathbb{E}X = G'(1),$$

and

$$\mathbb{V}\text{ar}(X) = G''(1) + G'(1) - (G'(1))^2.$$

Definition 2.8: Moment Generating Function

The **moment generating function** (MGF) of a random variable X is the function, $M : I \rightarrow [0, \infty)$, given by

$$M(s) = \mathbb{E} e^{sX}.$$

Here I is an open interval containing 0 for which the above integrals are finite for all $s \in I$. In particular, for a discrete random variable with pmf f ,

$$M(s) = \sum_x e^{sx} f(x),$$

and for a continuous random variable with pdf f ,

$$M(s) = \int_x e^{sx} f(x) dx.$$

As for the PGF, the moment generation function has the *uniqueness property*: Two MGFs are the same if and only if their corresponding distribution functions are the same.

Similar to the PGF, the moments of X follow from the derivatives of M :

If $\mathbb{E}X^n$ exists, then M is n times differentiable, and

$$\mathbb{E}X^n = M^{(n)}(0).$$

Hence the name moment generating function: the moments of X are simply found by differentiating. As a consequence, the variance of X is found as

$$\mathbb{V}\text{ar}(X) = M''(0) - (M'(0))^2.$$

■ **Remark 2.1 (Sums of Independent Random Variables)** The transforms discussed here are particularly useful when dealing with **sums** of **independent** random variables. We will return to them in Chapters 4 and 5. ■

2.5 Some Important Discrete Distributions

In this section we give a number of important discrete distributions and list some of their properties. Note that the pmf of each of these distributions depends on one or more parameters; so in fact we are dealing with *families* of distributions.

2.5.1 Bernoulli Distribution

We say that X has a **Bernoulli** distribution with success probability p if X can only assume the values 0 and 1, with probabilities

$$\mathbb{P}(X = 1) = p = 1 - \mathbb{P}(X = 0).$$

We write $X \sim \text{Ber}(p)$. Despite its simplicity, this is one of the most important distributions in probability! It models for example a single coin toss experiment. The cdf is given in Figure 2.5.

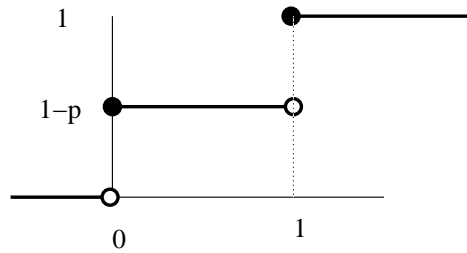


Figure 2.5: The cdf of the Bernoulli distribution

Here are some properties:

1. The expectation is $\mathbb{E}X = 0\mathbb{P}(X = 0) + 1\mathbb{P}(X = 1) = 0 \times (1 - p) + 1 \times p = p$.
2. The variance is $\mathbb{V}\text{ar}(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2 = \mathbb{E}X - (\mathbb{E}X)^2 = p - p^2 = p(1 - p)$. (Note that $X^2 = X$).
3. The PGF is given by $G(z) = z^0(1 - p) + z^1p = 1 - p + zp$.

2.5.2 Binomial Distribution

Consider a sequence of n coin tosses. If X is the random variable which counts the total number of heads and the probability of “head” is p then we say X has a **binomial** distribution with parameters n and p and write $X \sim \text{Bin}(n, p)$. The probability mass function X is given by

$$f(x) = \mathbb{P}(X = x) = \binom{n}{x} p^x (1 - p)^{n-x}, \quad x = 0, 1, \dots, n. \quad (2.4)$$

This follows from Examples 1.15 and 2.2. An example of the graph of the pmf is given in Figure 2.6

Here are some important properties of the Binomial distribution. Some of these properties can be proved more easily after we have discussed multiple random variables.

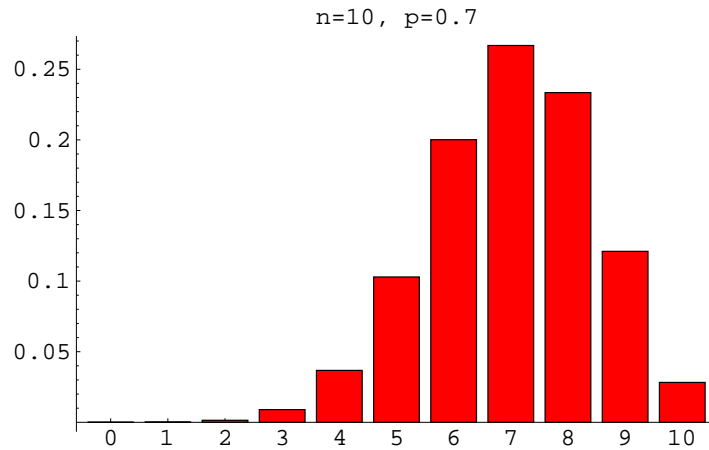


Figure 2.6: The pmf of the Bin(10, 0.7)-distribution

1. The expectation is $\mathbb{E}X = np$. This is a quite intuitive result. The expected number of successes (heads) in n coin tosses is np , if p denotes the probability of success in any one toss. To prove this, one could simply evaluate the sum

$$\sum_{x=0}^n x \binom{n}{x} p^x (1-p)^{n-x},$$

but this is not elegant. We will see in chapter 4 that X can be viewed as a sum $X = X_1 + \dots + X_n$ of n independent $\text{Ber}(p)$ random variables, where X_i indicates whether the i -th toss is a success or not, $i = 1, \dots, n$. Also we will prove that the expectation of such a sum is the sum of the expectation, therefore,

$$\mathbb{E}X = \mathbb{E}(X_1 + \dots + X_n) = \mathbb{E}X_1 + \dots + \mathbb{E}X_n = \underbrace{p + \dots + p}_{n \text{ times}} = np.$$

2. The variance of X is $\mathbb{V}\text{ar}(X) = np(1-p)$. This is proved in a similar way to the expectation:

$$\begin{aligned} \mathbb{V}\text{ar}(X) &= \mathbb{V}\text{ar}(X_1 + \dots + X_n) = \mathbb{V}\text{ar}(X_1) + \dots + \mathbb{V}\text{ar}(X_n) \\ &= \underbrace{p(1-p) + \dots + p(1-p)}_{n \text{ times}} = np(1-p). \end{aligned}$$

3. The probability generating function of X is $G(z) = (1-p+zp)^n$. Again, we can easily prove this after we consider multiple random variables in Chapter 4. Namely,

$$\begin{aligned} G(z) &= \mathbb{E}z^X = \mathbb{E}z^{X_1 + \dots + X_n} = \mathbb{E}z^{X_1} \dots \mathbb{E}z^{X_n} \\ &= (1-p+zp) \times \dots \times (1-p+zp) = (1-p+zp)^n. \end{aligned}$$

However, we can also easily prove it using Newton's binomial formula:

$$(a+b)^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k}.$$

Specifically,

$$G(z) = \sum_{k=0}^n z^k \binom{n}{k} p^k (1-p)^{n-k} = \sum_{k=0}^n \binom{n}{k} (zp)^k (1-p)^{n-k} = (1-p+zp)^n.$$

Note that once we have obtained the PGF, we can obtain the expectation and variance as $G'(1) = np$ and $G''(1) + G'(1) - (G'(1))^2 = (n-1)np^2 + np - n^2p^2 = np(1-p)$.

2.5.3 Geometric distribution

Again we look at a sequence of coin tosses but count a different thing. Let X be the number of tosses needed before the first head occurs. Then

$$\mathbb{P}(X = x) = (1-p)^{x-1} p, \quad x = 1, 2, 3, \dots \quad (2.5)$$

since the only string that has the required form is

$$\underbrace{ttt \dots t}_{x-1} h$$

and this has probability $(1-p)^{x-1} p$. See also Example 1.16 on page 33. Such a random variable X is said to have a **geometric** distribution with parameter p . We write $X \sim \text{Geom}(p)$. An example of the graph of the pdf is given in Figure 2.7

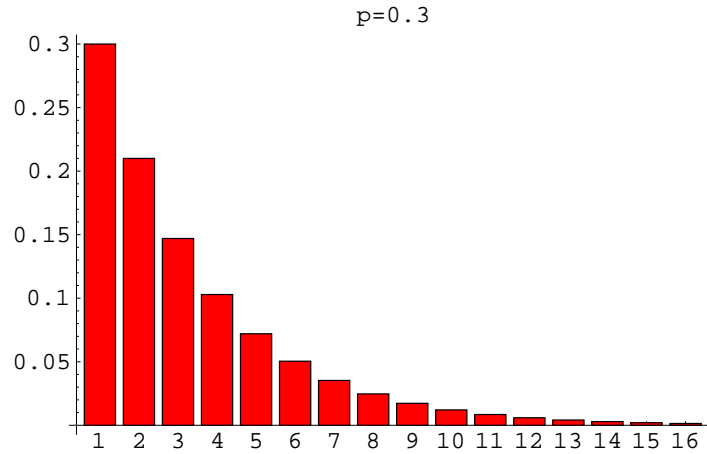


Figure 2.7: The pmf of the Geom(0.3)-distribution

We give some more properties, including the expectation, variance and PGF of the geometric distribution. It is easiest to start with the PGF:

1. The PGF is given by

$$G(z) = \sum_{x=1}^{\infty} z^x p (1-p)^{x-1} = zp \sum_{k=0}^{\infty} (z(1-p))^k = \frac{zp}{1-z(1-p)},$$

using the well-known result for *geometric sums*: $1 + a + a^2 + \cdots = \frac{1}{1-a}$, for $|a| < 1$.

2. The expectation is therefore

$$\mathbb{E}X = G'(1) = \frac{1}{p},$$

which is an intuitive result. We expect to wait $1/p$ throws before a success appears, if successes are generated with probability p .

3. By differentiating the PGF twice we find the variance:

$$\text{Var}(X) = G''(1) + G'(1) - (G'(1))^2 = \frac{2(1-p)}{p^2} + \frac{1}{p} - \frac{1}{p^2} = \frac{1-p}{p^2}.$$

4. The probability of requiring more than k tosses before a success is

$$\mathbb{P}(X > k) = (1-p)^k.$$

This is obvious from the fact that $\{X > k\}$ corresponds to the event of k consecutive failures.

A final property of the geometric distribution which deserves extra attention is the **memory-less property**. Think again of the coin toss experiment. Suppose we have tossed the coin x times without a success (Heads). What is the probability that we need more than y additional tosses before getting a success. The answer is, obviously, the same as the probability that we require more than y tosses if we start from scratch, that is, $\mathbb{P}(X > y) = (1-p)^y$, irrespective of x . The fact that we have already had x failures does not make the event of getting a success in the next trial(s) any more likely. In other words, the coin does not have a memory of what happened, hence the word memoryless property. Mathematically, it means that for any $y, x = 1, 2, \dots$,

$$\mathbb{P}(X > x + y | X > x) = \mathbb{P}(X > y)$$

Proof: By the definition of conditional probability

$$\mathbb{P}(X > x + y | X > x) = \frac{\mathbb{P}(\{X > x + y\} \cap \{X > x\})}{\mathbb{P}(X > x)}.$$

Now, the event $\{X > x + y\}$ is a subset of $\{X > x\}$, hence their intersection is $\{X > x + y\}$. Moreover, the probabilities of the events $\{X > x + y\}$ and $\{X > x\}$ are $(1-p)^{x+y}$ and $(1-p)^x$, respectively, so that

$$\mathbb{P}(X > x + y | X > x) = \frac{(1-p)^{x+y}}{(1-p)^x} = (1-p)^y = \mathbb{P}(X > y),$$

as required. □

2.5.4 Poisson Distribution

A random variable X for which

$$\mathbb{P}(X = x) = \frac{\lambda^x}{x!} e^{-\lambda}, \quad x = 0, 1, 2, \dots, \quad (2.6)$$

(for fixed $\lambda > 0$) is said to have a **Poisson** distribution. We write $X \sim \text{Poi}(\lambda)$. The Poisson distribution is used in many probability models and may be viewed as the “limit” of the $\text{Bin}(n, \lambda/n)$ for large n in the following sense: Consider a coin tossing experiment where we toss a coin n times with success probability λ/n . Let X be the number of successes. Then, as we have seen $X \sim \text{Bin}(n, \lambda/n)$. In other words,

$$\begin{aligned} \mathbb{P}(X = x) &= \binom{n}{x} \left(\frac{\lambda}{n}\right)^x \left(1 - \frac{\lambda}{n}\right)^{n-x} \\ &= \frac{\lambda^x}{x!} \frac{n \times n - 1 \times \dots \times n - x + 1}{n \times n \times \dots \times n} \left(1 - \frac{\lambda}{n}\right)^n \left(1 - \frac{\lambda}{n}\right)^{-x} \end{aligned}$$

As $n \rightarrow \infty$, the second and fourth factors go to 1, and the third factor goes to $e^{-\lambda}$ (this is one of the defining properties of the exponential function). Hence, we have

$$\lim_{n \rightarrow \infty} \mathbb{P}(X = x) = \frac{\lambda^x}{x!} e^{-\lambda},$$

which shows that the Poisson distribution is a limiting case of the binomial one. An example of the graph of its pmf is given in Figure 2.8

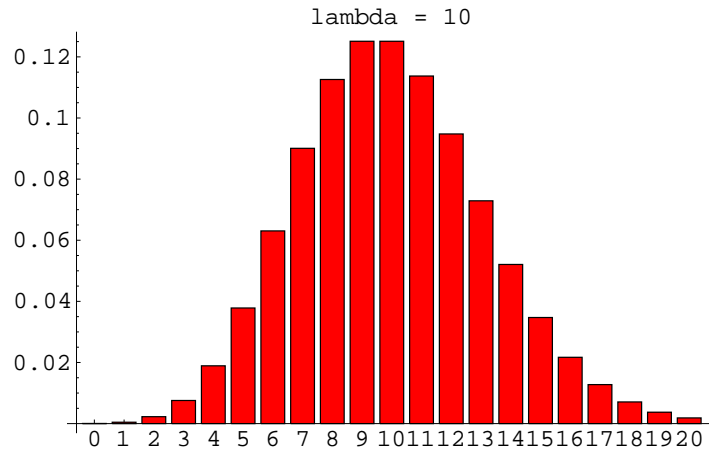


Figure 2.8: The pmf of the $\text{Poi}(10)$ -distribution

We finish with some properties.

1. The PGF was derived in Example 2.8:

$$G(z) = e^{-\lambda(1-z)}.$$

2. It follows that the expectation is $\mathbb{E}X = G'(1) = \lambda$. The intuitive explanation is that the mean number of successes of the corresponding coin flip experiment is $np = n(\lambda/n) = \lambda$.
3. The above argument suggests that the variance should be $n(\lambda/n)(1 - \lambda/n) \rightarrow \lambda$. This is indeed the case, as

$$\text{Var}(X) = G''(1) + G'(1) - (G'(1))^2 = \lambda^2 + \lambda - \lambda^2 = \lambda.$$

Thus for the Poisson distribution the variance and expectation are the same.

2.5.5 Hypergeometric Distribution

We say that a random variable X has a **Hypergeometric distribution** with parameters N , n and r if

$$\mathbb{P}(X = x) = \frac{\binom{r}{x} \binom{N-r}{n-x}}{\binom{N}{n}},$$

for $\max\{0, r + n - N\} \leq x \leq \min\{n, r\}$.

We write $X \sim \text{Hyp}(n, r, N)$. The hypergeometric distribution is used in the following situation.

Consider an urn with N balls, r of which are red. We draw at random n balls from the urn without replacement. The number of red balls amongst the n chosen balls has a $\text{Hyp}(n, r, N)$ distribution. Namely, if we number the red balls $1, \dots, r$ and the remaining balls $r + 1, \dots, N$, then the total number of outcomes of the random experiment is $\binom{N}{n}$, and each of these outcomes is equally likely. The number of outcomes in the event “ x balls are red” is $\binom{r}{x} \times \binom{N-r}{n-x}$ because the x balls have to be drawn from the r red balls, and the remaining $n - x$ balls have to be drawn from the $N - r$ non-red balls. In table form we have:

	Red	Not Red	Total
Selected	x	$n - x$	n
Not Selected	$r - x$	$N - n - r + x$	$N - n$
Total	r	$N - r$	N

■ **Example 2.9 (Hypergeometric Distribution)** Five cards are selected from a full deck of 52 cards. Let X be the number of Aces. Then $X \sim \text{Hyp}(n = 5, r = 4, N = 52)$.

k	0	1	2	3	4	Σ
$\mathbb{P}(X = k)$	0.659	0.299	0.040	0.002	0.000	1

The expectation and variance of the hypergeometric distribution are

$$\mathbb{E}X = n \frac{r}{N}$$

and

$$\mathbb{V}\text{ar}(X) = n \frac{r}{N} \left(1 - \frac{r}{N}\right) \frac{N-n}{N-1}.$$

Note that this closely resembles the expectation and variance of the binomial case, with $p = r/N$. The proofs are given in Chapter 4 (see Examples 4.7 and 4.10, on pages 82 and 86). ■

2.6 Some Important Continuous Distributions

In this section we give a number of important continuous distributions and list some of their properties. Note that the pdf of each of these distributions depends on one or more parameters; so, as in the discrete case discussed before, we are dealing with *families* of distributions.

2.6.1 Uniform Distribution

We say that a random variable X has a **uniform** distribution on the interval $[a, b]$, if it has density function f , given by

$$f(x) = \frac{1}{b-a}, \quad a \leq x \leq b.$$

We write $X \sim \mathcal{U}[a, b]$. X can model a randomly chosen point from the interval $[a, b]$, where each choice is equally likely. A graph of the pdf is given in Figure 2.9.

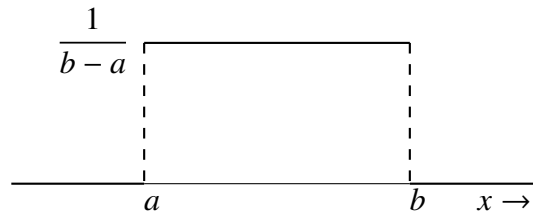


Figure 2.9: The pdf of the uniform distribution on $[a, b]$

We have

$$\mathbb{E}X = \int_a^b \frac{x}{b-a} dx = \frac{1}{b-a} \left[\frac{b^2 - a^2}{2} \right] = \frac{a+b}{2}.$$

This can be seen more directly by observing that the pdf is symmetric around $c = (a + b)/2$, and that the expectation is therefore equal to the symmetry point c . For the variance we have

$$\begin{aligned}\mathbb{V}\text{ar}(X) &= \mathbb{E}X^2 - (\mathbb{E}X)^2 = \int_a^b \frac{x^2}{b-a} dx - \left(\frac{a+b}{2}\right)^2 \\ &= \dots = \frac{(a-b)^2}{12}.\end{aligned}$$

A more elegant way to derive this is to use the fact that X can be thought of as the sum $X = a + (b - a)U$, where $U \sim \mathcal{U}[0, 1]$. Namely, for $x \in [a, b]$

$$\mathbb{P}(X \leq x) = \frac{x-a}{b-a} = \mathbb{P}\left(U \leq \frac{x-a}{b-a}\right) = \mathbb{P}(a + (b-a)U \leq x).$$

Thus, we have $\mathbb{V}\text{ar}(X) = \mathbb{V}\text{ar}(a + (b-a)U) = (b-a)^2 \mathbb{V}\text{ar}(U)$. And

$$\mathbb{V}\text{ar}(U) = \mathbb{E}U^2 - (\mathbb{E}U)^2 = \int_0^1 u^2 du - \left(\frac{1}{2}\right)^2 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}.$$

2.6.2 Exponential Distribution

A random variable X with probability density function f , given by

$$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0 \tag{2.7}$$

is said to have an **exponential** distribution with parameter λ . We write $X \sim \text{Exp}(\lambda)$. The exponential distribution can be viewed as a continuous version of the geometric distribution. Graphs of the pdf for various values of λ are given in Figure 2.10.

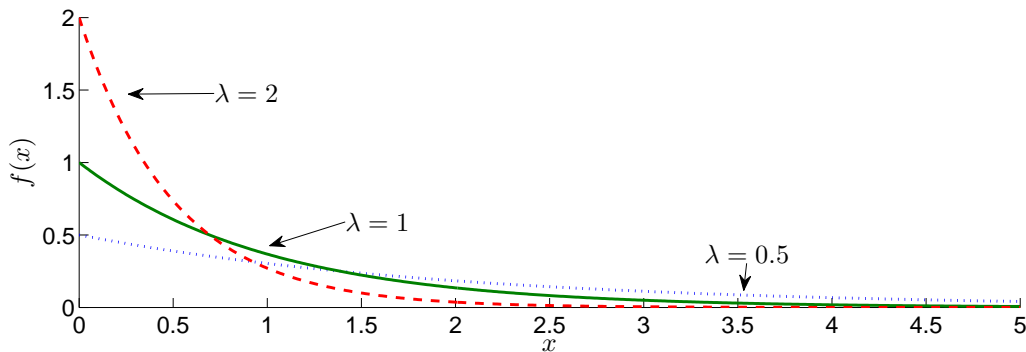


Figure 2.10: The pdf of the $\text{Exp}(\lambda)$ -distribution for various λ .

Here are some properties of the exponential distribution:

1. The moment generating function is

$$\begin{aligned} M(s) &= \int_0^{\infty} e^{sx} \lambda e^{-\lambda x} dx = \lambda \int_0^{\infty} e^{-(\lambda-s)x} dx = \lambda \left[\frac{-e^{-(\lambda-s)x}}{\lambda-s} \right]_{x=0}^{\infty} \\ &= \frac{\lambda}{\lambda-s}, \end{aligned}$$

for $s < \lambda$.

2. From the moment generating function we find by differentiation:

$$\mathbb{E}X = M'(0) = \frac{\lambda}{(\lambda-s)^2} \Big|_{s=0} = \frac{1}{\lambda}.$$

Alternatively, you can use integration by parts to evaluate

$$\mathbb{E}X = \int_0^{\infty} \underbrace{x}_1 \underbrace{\lambda e^{-\lambda x}}_{-e^{-\lambda x}} dx = \left[-xe^{-\lambda x} \right]_0^{\infty} + \int_0^{\infty} e^{-\lambda x} dx = 0 + \left[\frac{-e^{-\lambda x}}{\lambda} \right]_0^{\infty} = \frac{1}{\lambda}.$$

3. Similarly, the second moment is $\mathbb{E}X^2 = M''(0) = \frac{2\lambda}{(\lambda-s)^3} \Big|_{s=0} = 2/\lambda^2$, so that the variance becomes

$$\text{Var}(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2 = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}.$$

4. The cdf of X is given by

$$F(x) = \mathbb{P}(X \leq x) = \int_0^x \lambda e^{-\lambda u} du = \left[-e^{-\lambda u} \right]_0^x = 1 - e^{-\lambda x}, \quad x \geq 0.$$

5. As consequence the tail probability $\mathbb{P}(X > x)$ is exponentially decaying:

$$\mathbb{P}(X > x) = e^{-\lambda x}, \quad x \geq 0.$$

The most important property of the exponential distribution is the following:

Theorem 2.2: Memoryless Property

Let X have an exponential distribution with parameter λ . Then

$$\mathbb{P}(X > x+y | X > x) = \mathbb{P}(X > y), \quad \text{for all } x, y \geq 0. \quad (2.8)$$

Proof: By (1.4)

$$\begin{aligned} \mathbb{P}(X > x+y | X > x) &= \frac{\mathbb{P}(X > x+y, X > x)}{\mathbb{P}(X > x)} = \frac{\mathbb{P}(X > x+y)}{\mathbb{P}(X > x)} \\ &= \frac{e^{-\lambda(x+y)}}{e^{-\lambda x}} = e^{-\lambda y} = \mathbb{P}(X > y), \end{aligned}$$

where in the second equation we have used that the event $\{X > x + y\}$ is contained in the event $\{X > x\}$; hence, the intersection of these two sets is $\{X > x + y\}$. \square

For example, when X denotes the lifetime of a machine, then given the fact that the machine is alive at time x , the remaining lifetime of the machine, i.e., $X - x$, has the same exponential distribution as a completely new machine. In other words, the machine has no memory of its age and does not “deteriorate” (although it will break down eventually).

It is not too difficult to prove that the exponential distribution is the *only* continuous (positive) distribution with the memoryless property.

2.6.3 Normal, or Gaussian, Distribution

The normal (or Gaussian) distribution is the most important distribution in the study of statistics. We say that a random variable has a **normal** distribution with parameters μ and σ^2 if its density function f is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad x \in \mathbb{R}. \quad (2.9)$$

We write $X \sim \mathcal{N}(\mu, \sigma^2)$. The parameters μ and σ^2 turn out to be the expectation and variance of the distribution, respectively. If $\mu = 0$ and $\sigma = 1$ then

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2},$$

and the distribution is known as a **standard normal** distribution. The cdf of this latter distribution is often denoted by Φ , and is tabulated in Appendix B. In Figure 2.11 the probability densities for three different normal distributions have been depicted.

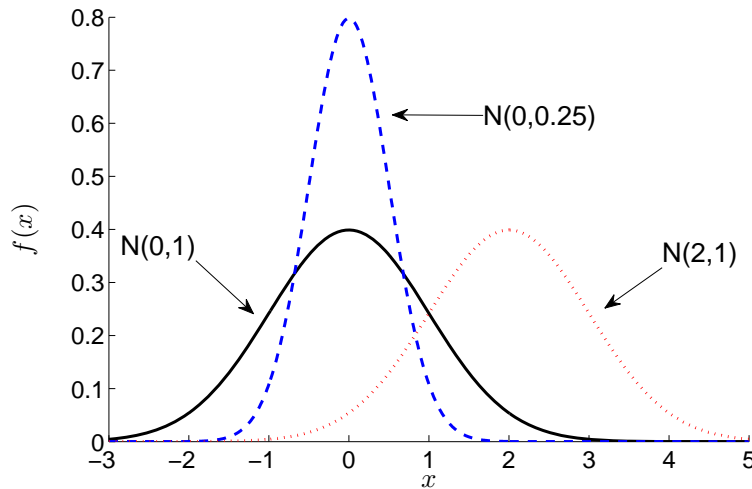


Figure 2.11: Probability density functions for various normal distributions

We next consider some important properties of the normal distribution.

1. If $X \sim \mathcal{N}(\mu, \sigma^2)$, then

$$\frac{X - \mu}{\sigma} \sim \mathcal{N}(0, 1) . \quad (2.10)$$

Thus by subtracting the mean and dividing by the standard deviation we obtain a standard normal distribution. This procedure is called **standardisation**.

Proof: Let $X \sim \mathcal{N}(\mu, \sigma^2)$, and $Z = (X - \mu)/\sigma$. Then,

$$\begin{aligned} \mathbb{P}(Z \leq z) &= \mathbb{P}((X - \mu)/\sigma \leq z) = \mathbb{P}(X \leq \mu + \sigma z) \\ &= \int_{-\infty}^{\mu + \sigma z} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2} dx \\ &= \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy \quad [\text{c.o.v. } y = \frac{x - \mu}{\sigma}] \\ &= \Phi(z) . \end{aligned}$$

Thus Z has a standard normal distribution. \square

Standardisation enables us to express the cdf of any normal distribution in terms of the cdf of the standard normal distribution. This is the reason why only the table for the standard normal distribution is included in traditional statistics textbooks.

2. A trivial rewriting of the standardisation formula gives the following important result: If $X \sim \mathcal{N}(\mu, \sigma^2)$, then

$$X = \mu + \sigma Z, \quad \text{with } Z \sim \mathcal{N}(0, 1) .$$

In other words, any Gaussian (normal) random variable can be viewed as a so-called *affine* (linear + constant) transformation of a standard normal random variable.

3. $\mathbb{E}X = \mu$. This is because the pdf is symmetric around μ .
4. $\text{Var}(X) = \sigma^2$. This is a bit more involved. First, write $X = \mu + \sigma Z$, with Z standard normal. Then, $\text{Var}(X) = \text{Var}(\mu + \sigma Z) = \sigma^2 \text{Var}(Z)$. Hence, it suffices to show that the variance of Z is 1. Consider $\text{Var}(Z) = \mathbb{E}Z^2$ (note that the expectation is 0). We have

$$\mathbb{E}Z^2 = \int_{-\infty}^{\infty} z^2 \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = \int_{-\infty}^{\infty} z \times \frac{z}{\sqrt{2\pi}} e^{-z^2/2} dz$$

By writing the last integrand in this way we can apply partial integration to the two factors to yield

$$\mathbb{E}Z^2 = \left[z \frac{-1}{\sqrt{2\pi}} e^{-z^2/2} \right]_{-\infty}^{\infty} + \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = 1 ,$$

since the last integrand is the pdf of the standard normal distribution.

5. The moment generating function of $X \sim \mathcal{N}(\mu, \sigma^2)$ is given by

$$\mathbb{E}e^{sX} = e^{s\mu + s^2\sigma^2/2}, \quad s \in \mathbb{R}. \quad (2.11)$$

Proof: First consider the moment generation function of $Z \sim \mathcal{N}(0, 1)$. We have

$$\begin{aligned} \mathbb{E}e^{sZ} &= \int_{-\infty}^{\infty} e^{sz} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = e^{s^2/2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-(z-s)^2/2} dz \\ &= e^{s^2/2}, \end{aligned}$$

where the second integrand is the pdf of the $\mathcal{N}(s, 1)$ distribution, which therefore integrates to 1. Now, for general $X \sim \mathcal{N}(\mu, \sigma^2)$ write $X = \mu + \sigma Z$. Then,

$$\mathbb{E}e^{sX} = \mathbb{E}e^{s(\mu + \sigma Z)} = e^{s\mu} \mathbb{E}e^{s\sigma Z} = e^{s\mu} e^{\sigma^2 s^2/2} = e^{s\mu + \sigma^2 s^2/2}.$$

□

More on the Gaussian distribution later, especially the multidimensional cases!

2.6.4 Gamma- and χ^2 -distribution

The **gamma** distribution arises frequently in statistics. Its density function is given by

$$f(x) = \frac{\lambda^\alpha x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)}, \quad x \geq 0, \quad (2.12)$$

where Γ is the Gamma function, defined as

$$\Gamma(\alpha) = \int_0^\infty u^{\alpha-1} e^{-u} du, \quad \alpha > 0.$$

Parameter α is called the **shape** parameter and λ is called the **rate** parameter (and $1/\lambda$ is called the **scale** parameter). We write $X \sim \text{Gamma}(\alpha, \lambda)$.

Of particular importance is following special case: A random variable X is said to have a **chi-square** distribution with n ($\in \{1, 2, \dots\}$) **degrees of freedom** if $X \sim \text{Gamma}(n/2, 1/2)$. We write $X \sim \chi_n^2$. A graph of the pdf of the χ_n^2 -distribution, for various n is given in Figure 2.12.

We mention a few properties of the Γ -function.

1. $\Gamma(a + 1) = a\Gamma(a)$, for $a > 0$.
2. $\Gamma(n) = (n - 1)!$ for $n = 1, 2, \dots$

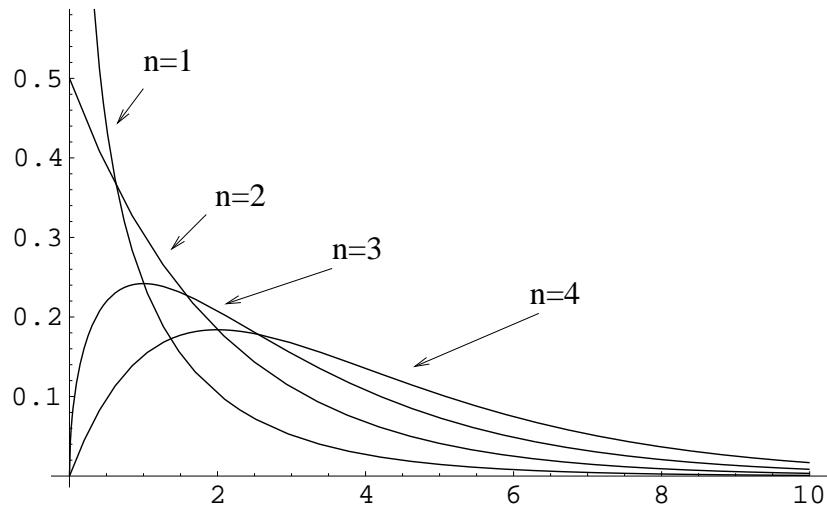


Figure 2.12: Pdfs for the χ_n^2 -distribution, for various degrees of freedom n

3. $\Gamma(1/2) = \sqrt{\pi}$.

The moment generating function of $X \sim \text{Gamma}(\alpha, \lambda)$ is given by

$$\begin{aligned}
 M(s) = \mathbb{E} e^{sX} &= \int_0^\infty \frac{e^{-\lambda x} \lambda^\alpha x^{\alpha-1}}{\Gamma(\alpha)} e^{sx} dx \\
 &= \left(\frac{\lambda}{\lambda - s} \right)^\alpha \int_0^\infty \frac{e^{-(\lambda-s)x} (\lambda-s)^\alpha x^{\alpha-1}}{\Gamma(\alpha)} dx \\
 &= \left(\frac{\lambda}{\lambda - s} \right)^\alpha.
 \end{aligned} \tag{2.13}$$

As a consequence, we have

$$\mathbb{E}X = M'(0) = \frac{\alpha}{\lambda} \left(\frac{\lambda}{\lambda - s} \right)^{\alpha+1} \Big|_{s=0} = \frac{\alpha}{\lambda},$$

and, similarly,

$$\mathbb{V}\text{ar}(X) = \frac{\alpha}{\lambda^2}.$$

GENERATING RANDOM VARIABLES ON A COMPUTER

3.1 Introduction

This chapter deals with the execution of random experiments via the computer, also called **stochastic simulation**. In a typical stochastic simulation, randomness is introduced into simulation models via independent uniformly distributed random variables, also called **random numbers**. These random numbers are then used as building blocks to simulate more general stochastic systems.

3.2 Random Number Generation

In the early days of simulation, randomness was generated by *manual* techniques, such as coin flipping, dice rolling, card shuffling, and roulette spinning. Later on, *physical devices*, such as noise diodes and Geiger counters, were attached to computers for the same purpose. The prevailing belief held that only mechanical or electronic devices could produce “truly” random sequences. Although mechanical devices are still widely used in gambling and lotteries, these methods were abandoned by the computer-simulation community for several reasons: (a) Mechanical methods were too slow for general use, (b) the generated sequences cannot be reproduced and, (c) it has been found that the generated numbers exhibit both bias and dependence. Although certain modern physical generation methods are fast and would pass most statistical tests for randomness (for example, those based on the universal background radiation or the noise of a PC chip), their main drawback remains to be their lack of repeatability. Most of today’s random number generators are not based on physical devices,

but on simple algorithms that can be easily implemented on a computer, are fast, require little storage space, and can readily reproduce a given sequence of random numbers. Importantly, a good random number generator captures all the important statistical properties of true random sequences, even though the sequence is generated by a deterministic algorithm. For this reason, these generators are sometimes called *pseudorandom*.

The most common methods for generating pseudorandom sequences use the so-called *linear congruential generators*. These generate a deterministic sequence of numbers by means of the recursive formula

$$X_{i+1} = aX_i + c \pmod{m}, \quad (3.1)$$

where the initial value, X_0 , is called the *seed*, and the a, c and m (all positive integers) are called the *multiplier*, the *increment* and the *modulus*, respectively. Note that applying the modulo- m operator in (3.1) means that $aX_i + c$ is divided by m , and the remainder is taken as the value of X_{i+1} . Thus, each X_i can only assume a value from the set $\{0, 1, \dots, m-1\}$, and the quantities

$$U_i = \frac{X_i}{m}, \quad (3.2)$$

called *pseudorandom numbers*, constitute approximations to the true sequence of uniform random variables. Note that the sequence $\{X_i\}$ will repeat itself in at most m steps, and will therefore be periodic with period not exceeding m . For example, let $a = c = X_0 = 3$ and $m = 5$; then the sequence obtained from the recursive formula $X_{i+1} = 3X_i + 3 \pmod{5}$ is $X_i = 3, 2, 4, 0, 3$, which has period 4, while $m = 5$. In the special case where $c = 0$, (3.1) simply reduces to

$$X_{i+1} = aX_i \pmod{m}. \quad (3.3)$$

Such a generator is called a *multiplicative congruential generator*. It is readily seen that an arbitrary choice of X_0, a, c and m will not lead to a pseudorandom sequence with good statistical properties. In fact, number theory has been used to show that only a few combinations of these produce satisfactory results. In computer implementations, m is selected as a large prime number that can be accommodated by the computer word size. For example, in a binary 32-bit word computer, statistically acceptable generators can be obtained by choosing $m = 2^{31} - 1$ and $a = 7^5$, provided the first bit is a sign bit. A 64-bit or 128-bit word computer will naturally yield better statistical results.

Most computer languages already contain a built-in pseudorandom number generator. The user is typically requested only to input the initial seed, X_0 , and upon invocation, the random number generator produces a sequence of independent uniform $(0, 1)$ random variables. We, therefore assume in this chapter the availability of a “black box”, capable of producing a stream of pseudorandom numbers. In Python this is provided by the `rand` function of the `numpy.random` module.

■ **Example 3.1 (Generating uniform random variables in Python)** This example illustrates the use of the `rand` function in Python, to generate samples from the $\mathcal{U}(0, 1)$ distribution.

```
from numpy.random import rand, seed
print(rand(1))
```

```
[0.88171397]
```

```
print(rand(5))
```

```
[0.86880489 0.37527345 0.90842683 0.21716382 0.71134196]
```

```
seed(1234)      # setting the random seed to 1234
print(rand(5))
seed(1234)      # resetting the seed to 1234
print(rand(5))  # gives the same sequence
```

```
[0.19151945 0.62210877 0.43772774 0.78535858 0.77997581]
[0.19151945 0.62210877 0.43772774 0.78535858 0.77997581]
```



3.3 The Inverse-Transform Method

In this section we discuss a general method for generating one-dimensional random variables from a prescribed distribution, namely the *inverse-transform method*.

Let X be a random variable with cdf F . Since F is a nondecreasing function, the inverse function F^{-1} may be defined as

$$F^{-1}(y) = \inf\{x : F(x) \geq y\}, \quad 0 \leq y \leq 1. \quad (3.4)$$

(Readers not acquainted with the notion \inf should read \min .) It is easy to show that if $U \sim \mathcal{U}(0, 1)$, then

$$X = F^{-1}(U) \quad (3.5)$$

has cdf F . Namely, since F is invertible and $\mathbb{P}(U \leq u) = u$, we have

$$\mathbb{P}(X \leq x) = \mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x). \quad (3.6)$$

Thus, to generate a random variable X with cdf F , draw $U \sim \mathcal{U}(0, 1)$ and set $X = F^{-1}(U)$. Figure 3.1 illustrates the inverse-transform method given by the following algorithm.

Algorithm 3.3.1: Inverse-Transform Method

1. Generate U from $\mathcal{U}(0, 1)$.
 2. Return $X = F^{-1}(U)$.
-

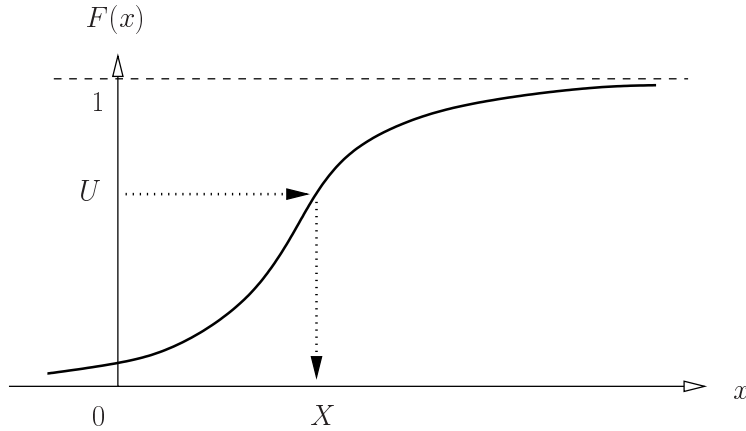


Figure 3.1: The inverse-transform method.

■ **Example 3.2 (Inverse-Transform Method)** Generate a random variable from the pdf

$$f(x) = \begin{cases} 2x, & 0 \leq x \leq 1 \\ 0, & \text{otherwise.} \end{cases} \quad (3.7)$$

The cdf is

$$F(x) = \begin{cases} 0, & x < 0 \\ \int_0^x 2x \, dx = x^2, & 0 \leq x \leq 1 \\ 1, & x > 1. \end{cases}$$

Applying (3.5), we have

$$X = F^{-1}(U) = \sqrt{U}, \quad 0 \leq u \leq 1.$$

Therefore, to generate a random variable X from the pdf (3.7), first generate a random variable U from $\mathcal{U}(0, 1)$, and then take its square root. ■

■ **Example 3.3 (Drawing From a Discrete Distribution)** Let X be a discrete random variable with $\mathbb{P}(X = x_i) = p_i$, $i = 1, 2, \dots$, with $\sum_i p_i = 1$. The cdf F of X is given by $F(x) = \sum_{i: x_i \leq x} p_i$, $i = 1, 2, \dots$; see Figure 3.2. ■

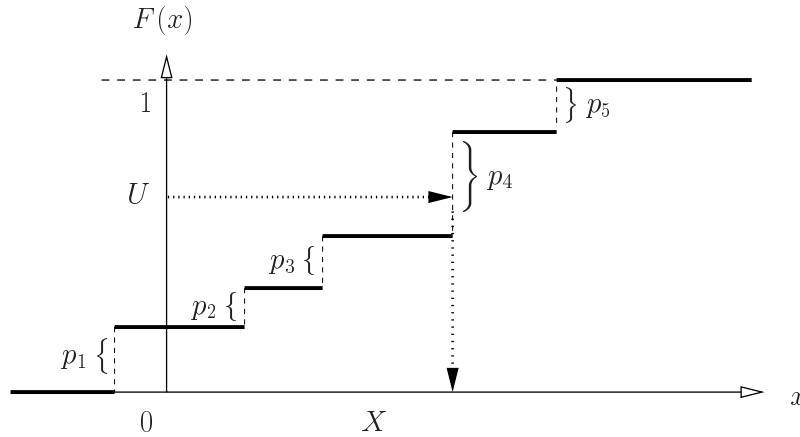


Figure 3.2: The inverse-transform method for a discrete random variable.

The algorithm for generating a random variable from F can thus be written as follows:

Algorithm 3.3.2: The Inverse-Transform Method for a Discrete Distribution

1. Generate $U \sim \mathcal{U}(0, 1)$.
 2. Find the smallest positive integer, k , such that $U \leq F(x_k)$ and return $X = x_k$.
-

In Python simulating numbers $0, 1, \dots, k-1$ according to probabilities p_0, p_1, \dots, p_{k-1} can be done in one line, after importing **numpy** as **np**:

```
np.min(np.where(np.cumsum(p) > np.random.rand()))
```

Here, **p** is the vector of probabilities, such as $[1/3, 1/3, 1/3]$, **np.cumsum** gives the cumulative sum vector, e.g., $[1/3, 2/3, 1]$, **np.where** finds the indices i such that $p_i > r$, where r is some random number, and **np.min** takes the smallest of these indices.

Much of the execution time in Algorithm 3.3.2 is spent in making the comparisons of Step 2. This time can be reduced by using efficient search techniques.

In general, the inverse-transform method requires that the underlying cdf, F , exist in a form for which the corresponding inverse function F^{-1} can be found analytically or algorithmically. Applicable distributions are, for example, the exponential, uniform, Weibull, logistic, and Cauchy distributions. Unfortunately, for many other probability distributions, it is either impossible or difficult to find the inverse transform, that is, to solve

$$F(x) = \int_{-\infty}^x f(t) dt = u$$

with respect to x . Even in the case where F^{-1} exists in an explicit form, the inverse-transform method may not necessarily be the most efficient random variable generation method.

3.4 The Acceptance–Rejection Method

The **acceptance–rejection method** is a general method for simulating random variables due to Stan Ulam and John von Neumann. Suppose that the pdf from which we want to sample is bounded on some finite interval $[a, b]$ and zero outside this interval. Let

$$c = \max\{f(x) : x \in [a, b]\}.$$

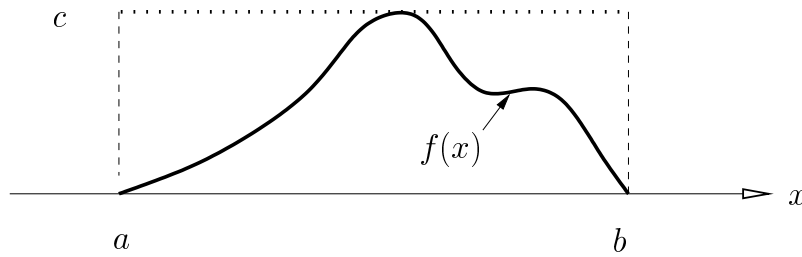


Figure 3.3: A bounded pdf

In this case we can generate $Z \sim f$ in the following way:

1. Generate $X \sim \mathcal{U}(a, b)$.
2. Generate $Y \sim \mathcal{U}(0, c)$.
3. If $Y \leq f(X)$ accept the point (X, Y) and return $Z = X$. Otherwise, reject the point and go back to Step 1.

The idea behind the method is that each random point (X, Y) is uniformly distributed over the rectangle $[a, b] \times [0, c]$. Therefore the accepted pair (X, Y) is uniformly distributed under the graph of f . This implies that the distribution of the accepted values of X has the desired pdf f .

We can generalize this as follows. Let g be a **proposal** pdf that is easy to sample from, and which “majorizes” f in such a way that there is a constant $C \geq 1$ such that $\phi(x) = C g(x) \geq f(x)$ for all x . pdf.

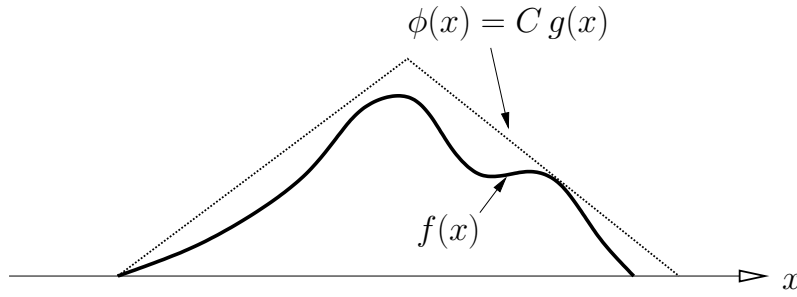


Figure 3.4: The acceptance-rejection method with a majorizing function.

The general acceptance-rejection algorithm can be written as:

Algorithm 3.4.1: Acceptance-Rejection

1. Generate $X \sim g$.
 2. Generate $Y \sim \mathcal{U}(0, C g(X))$.
 3. If $Y \leq f(X)$ return $Z = X$. Otherwise, return to Step 1.
-

The random variable X returned by the algorithm has pdf f .

The *efficiency* of an ARM is defined as

$$\mathbb{P}((X, Y) \text{ is accepted}) = \frac{\text{Area under } f}{\text{Area under } C g} = \frac{1}{C}. \quad (3.8)$$

For an acceptance–rejection method to be of practical interest, the following criteria must be used in selecting the proposal density $g(x)$:

1. It should be easy to generate a random variable from $g(x)$.
2. The efficiency, $1/C$, of the procedure should be large; that is, C should be close to 1 (which occurs when $g(x)$ is close to $f(x)$).

■ **Example 3.4 (Acceptance–Rejection)** To generate a random variable Z from the pdf

$$f(x) = \begin{cases} 2x, & 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

using the acceptance-rejection method, we could take the proposal pdf $g(x) = 1$, ($0 \leq x \leq 1$) and constant $C = 2$. Thus, our proposal distribution is simply the uniform distribution on $[0, 1]$ and C is as small as we can make it, with this proposal. The algorithm becomes:

1. Generate X from $\mathcal{U}(0, 1)$.
2. Generate Y from $\mathcal{U}(0, 2)$, independent of X .
3. If $Y \leq 2X$ return $Z = X$. Otherwise, go to Step 1.

The efficiency is $1/C = 1/2$. That is, we throw away half of the generated sample.



In general, the number of trials, N , until a successful pair (X, Y) is found, has a geometric distribution:

$$\mathbb{P}(N = n) = p(1 - p)^{n-1}, \quad n = 1, \dots, \quad (3.9)$$

with $p = 1/C$.

3.5 Generating From Commonly Used Distributions

We next present algorithms for generating variables from commonly used continuous and discrete distributions. Of the numerous algorithms available we have tried to select those which are reasonably efficient and relatively simple to implement.

Exponential Distribution

We start by applying the inverse-transform method to the exponential distribution. If $X \sim \text{Exp}(\lambda)$, then its cdf F is given by

$$F(x) = 1 - e^{-\lambda x}, \quad x > 0. \quad (3.10)$$

Hence, solving $u = F(x)$ in terms of x gives

$$F^{-1}(u) = -\frac{1}{\lambda} \ln(1 - u).$$

Noting that $U \sim \mathcal{U}(0, 1)$ implies $1 - U \sim \mathcal{U}(0, 1)$, we obtain the following algorithm.

Algorithm 3.5.1: Generation of an Exponential Random Variable

1. Generate $U \sim \mathcal{U}(0, 1)$.
 2. Return $X = -\frac{1}{\lambda} \ln U$ as a random variable from $\text{Exp}(\lambda)$.
-

There are many alternative procedures for generating variables from the exponential distribution. The interested reader is referred to Luc Devroye's book *Non-Uniform Random Variate Generation*, Springer-Verlag, 1986. (The entire book can be downloaded for free.)

Normal (Gaussian) Distribution

If $X \sim \mathcal{N}(\mu, \sigma^2)$, its pdf is given by

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{(x - \mu)^2}{2\sigma^2} \right], \quad -\infty < x < \infty, \quad (3.11)$$

where μ is the mean (or expectation) and σ^2 the variance of the distribution.

Since inversion of the normal cdf is numerically inefficient, the inverse-transform method is not very suitable for generating normal random variables, and some other procedures must be devised instead. We consider only generation from $\mathcal{N}(0, 1)$ (standard normal variables), since any random $Z \sim \mathcal{N}(\mu, \sigma^2)$ can be represented as $Z = \mu + \sigma X$, where X is from $\mathcal{N}(0, 1)$. One of the earliest methods for generating variables from $\mathcal{N}(0, 1)$ is the following, due to Box and Muller. A justification of this method will be given in Chapter 5, see Example 5.8.

Algorithm 3.5.2: Generation of a pair of $\mathcal{N}(0, 1)$ variables (Box–Muller)

1. Generate two independent random variables, U_1 and U_2 , from $\mathcal{U}(0, 1)$.
2. Return two independent standard normal variables, X and Y , via

$$\begin{aligned} X &= (-2 \ln U_1)^{1/2} \cos(2\pi U_2), \\ Y &= (-2 \ln U_1)^{1/2} \sin(2\pi U_2). \end{aligned} \quad (3.12)$$

An alternative is to use the acceptance–rejection method, using an exponential proposal. In particular, Note that in order to generate $Y \sim \mathcal{N}(0, 1)$, one can first generate a nonnegative random variable X from the **half-normal** pdf

$$f(x) = \sqrt{\frac{2}{\pi}} e^{-x^2/2}, \quad x \geq 0, \quad (3.13)$$

and then assign to X a random sign. The validity of this procedure follows from the symmetry of the standard normal distribution about zero.

To generate a random variable X from (3.13), we bound $f(x)$ by $C g(x)$, where $g(x) = e^{-x}$ is the pdf of the $\text{Exp}(1)$. The smallest constant C such that $f(x) \leq C g(x)$ is $\sqrt{2e/\pi}$, see Figure 3.5. The efficiency of this method is thus $\sqrt{\pi/2e} \approx 0.76$.

Bernoulli Distribution

If $X \sim \text{Ber}(p)$, its pmf is of the form

$$f(x) = p^x (1 - p)^{1-x}, \quad x = 0, 1, \quad (3.14)$$

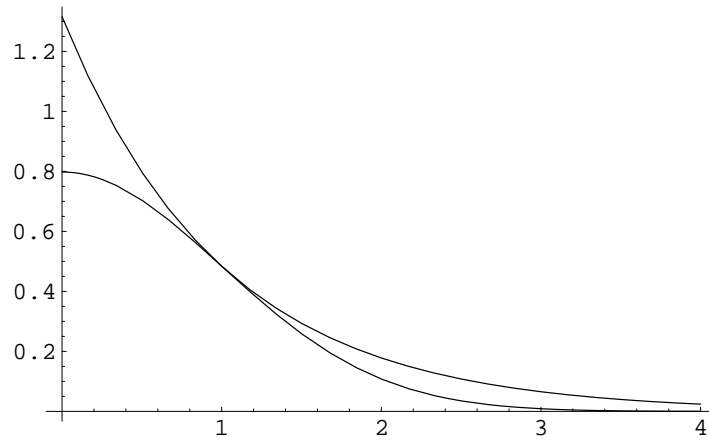


Figure 3.5: Bounding the positive normal density

where p is the success probability. Applying the inverse-transform method, one readily obtains the following generation algorithm:

Algorithm 3.5.3: Generation of a Bernoulli Random Variable

1. Generate $U \sim \mathcal{U}(0, 1)$.
 2. If $U \leq p$, return $X = 1$; otherwise return $X = 0$.
-

In Figure 1.1 on page 8 typical outcomes are given of 100 independent Bernoulli random variables, each with success parameter $p = 0.5$.

Binomial Distribution

If $X \sim \text{Bin}(n, p)$ then its pmf is of the form

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, \dots, n. \quad (3.15)$$

Recall that a binomial random variable X can be viewed as the total number of successes in n independent Bernoulli experiments, each with success probability p ; see Example 1.15. Denoting the result of the i -th trial by $X_i = 1$ (success) or $X_i = 0$ (failure), we can write $X = X_1 + \dots + X_n$ with the $\{X_i\}$ iid $\text{Ber}(p)$ random variables. The simplest generation algorithm can thus be written as follows:

It is worthwhile to note that if $Y \sim \text{Bin}(n, p)$, then $n - Y \sim \text{Bin}(n, 1 - p)$. Hence, to enhance

Algorithm 3.5.4: Generation of a Binomial Random Variable

1. Generate iid random variables X_1, \dots, X_n from $\text{Ber}(p)$.
2. Return $X = \sum_{i=1}^n X_i$ as a random variable from $\text{Bin}(n, p)$.

efficiency, one may elect to generate X from $\text{Bin}(n, p)$ according to

$$X = \begin{cases} Y_1 \sim \text{Bin}(n, p), & \text{if } p \leq \frac{1}{2} \\ Y_2 \sim \text{Bin}(n, 1 - p), & \text{if } p > \frac{1}{2} . \end{cases}$$

Geometric Distribution

If $X \sim \text{Geom}(p)$, then its pmf is of the form

$$f(x) = p(1 - p)^{x-1}, \quad x = 1, 2, \dots \quad (3.16)$$

The random variable X can be interpreted as the the number of trials required until the first success occurs, in a series of independent Bernoulli trials with success parameter p . Note that $\mathbb{P}(X > m) = (1 - p)^m$.

We now present an algorithm which is based on the relationship between the exponential and geometric distributions. Let $Y \sim \text{Exp}(\lambda)$, with λ such that $q = 1 - p = e^{-\lambda}$. Then, $X = \lfloor Y \rfloor + 1$ has a $\text{Geom}(p)$ distribution — here $\lfloor \cdot \rfloor$ denotes the integer part. Namely,

$$\mathbb{P}(X > x) = \mathbb{P}(\lfloor Y \rfloor > x - 1) = \mathbb{P}(Y > x) = e^{-\lambda x} = (1 - p)^x .$$

Hence, to generate a random variable from $\text{Geom}(p)$, we first generate a random variable from the exponential distribution with $\lambda = -\ln(1 - p)$, truncate the obtained value to the nearest integer and add 1.

Algorithm 3.5.5: Generation of a Geometric Random Variable

1. Generate $Y \sim \text{Exp}(-\ln(1 - p))$
2. Return $X = 1 + \lfloor Y \rfloor$ as a random variable from $\text{Geom}(p)$.

JOINT DISTRIBUTIONS

Often a random experiment is described via more than one random variable. Examples are:

1. We select a random sample of $n = 10$ people and observe their lengths. Let X_1, \dots, X_n be the individual lengths.
2. We flip a coin repeatedly. Let $X_i = 1$ if the i th flip is “heads” and 0 else. The experiment is described by the sequence X_1, X_2, \dots of coin flips.
3. We randomly select a person from a large population and measure his/her weight X and height Y .

How can we specify the behaviour of the random variables above? We should not just specify the pdf or pmf of the individual random variables, but also say something about the “interaction” (or lack thereof) between the random variables. For example, in the third experiment above if the height Y is large, we expect that X is large as well. On the other hand, for the first and second experiment it is reasonable to assume that information about one of the random variables does not give extra information about the others. What we need to specify is the **joint distribution** of the random variables.

The theory for multiple random variables is quite similar to that of a single random variable. The most important extra feature is perhaps the concept of *independence* of random variables. Independent random variables play a crucial role in stochastic modelling.

4.1 Joint Distribution and Independence

Let X_1, \dots, X_n be random variables describing some random experiment. We can accumulate the X_i 's into a row vector $\mathbf{X} = (X_1, \dots, X_n)$ or column vector $\mathbf{X} = (X_1, \dots, X_n)^T$ (here T means transposition). \mathbf{X} is called a **random vector**.

Recall that the distribution of a *single* random variable X is completely specified by its cumulative distribution function. Analogously, the joint distribution of X_1, \dots, X_n is specified by the **joint cumulative distribution function** F , defined by

$$F(x_1, \dots, x_n) = \mathbb{P}(\{X_1 \leq x_1\} \cap \dots \cap \{X_n \leq x_n\}) = \mathbb{P}(X_1 \leq x_1, \dots, X_n \leq x_n),$$

If we know F then we can in principle derive any probability involving the X_i 's. Note the abbreviation on the right-hand side. We will henceforth use this kind of abbreviation throughout the notes.

Similar to the 1-dimensional case we distinguish between the case where the X_i are discrete and continuous. The corresponding joint distributions are again called discrete and continuous, respectively.

4.1.1 Discrete Joint Distributions

To see how things work in the discrete case, let's start with an example.

■ **Example 4.1 (Box With Different Dice)** In a box are three dice. Die 1 is a normal die; die 2 has no 6 face, but instead two 5 faces; die 3 has no 5 face, but instead two 6 faces. The experiment consists of selecting a die at random, followed by a toss with that die. Let X be the die number that is selected, and let Y be the face value of that die. The probabilities $\mathbb{P}(X = x, Y = y)$ are specified below.

x	y						Σ
	1	2	3	4	5	6	
1	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{3}$
2	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{9}$	0	$\frac{1}{3}$
3	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	0	$\frac{1}{9}$	$\frac{1}{3}$
Σ	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	1



The function $f : (x, y) \mapsto \mathbb{P}(X = x, Y = y)$ is called the *joint pmf* of X and Y . The following definition is just a generalization of this.

Definition 4.1: Joint Probability Mass Function

Let X_1, \dots, X_n be *discrete* random variables. The function f defined by $f(x_1, \dots, x_n) = \mathbb{P}(X_1 = x_1, \dots, X_n = x_n)$ is called the **joint probability mass function** (pmf) of X_1, \dots, X_n .

We sometimes write f_{X_1, \dots, X_n} instead of f to show that this is the pmf of the random variables X_1, \dots, X_n . Or, if \mathbf{X} is the corresponding random vector, we could write $f_{\mathbf{X}}$ instead.

Note that, by the sum rule, if we are given the joint pmf of X_1, \dots, X_n we can in principle calculate *all possible probabilities* involving these random variables. For example, in the 2-dimensional case

$$\mathbb{P}((X, Y) \in B) = \sum_{(x, y) \in B} \mathbb{P}(X = x, Y = y) ,$$

for any subset B of possible values for (X, Y) . In particular, we can find the pmf of X by summing the joint pmf over all possible values of y :

$$\mathbb{P}(X = x) = \sum_y \mathbb{P}(X = x, Y = y) .$$

The converse is *not* true: from the individual distributions (so-called **marginal** distribution) of X and Y we cannot in general reconstruct the joint distribution of X and Y . We are missing the “dependency” information. E.g., in Example 4.1 we cannot reconstruct the inside of the two-dimensional table if only given the column and row totals.

However, there is one *important exception* to this, namely when we are dealing with *independent* random variables. We have so far only defined what independence is for *events*. The following definition says that random variables X_1, \dots, X_n are independent if the events $\{X_1 \in A_1\}, \dots, \{X_n \in A_n\}$ are independent for any subsets A_1, \dots, A_n of \mathbb{R} . Intuitively, this means that any information about one of them does not affect our knowledge about the others.

Definition 4.2: Independent Random Variables

The random variables X_1, \dots, X_n are called **independent** if for all A_1, \dots, A_n , with $A_i \subset \mathbb{R}$, $i = 1, \dots, n$

$$\mathbb{P}(X_1 \in A_1, \dots, X_n \in A_n) = \mathbb{P}(X_1 \in A_1) \cdots \mathbb{P}(X_n \in A_n) .$$

The following theorem is a direct consequence of the definition above.

Theorem 4.1: Independent Discrete Random Variables

Discrete random variables X_1, \dots, X_n , are independent if and only if

$$\mathbb{P}(X_1 = x_1, \dots, X_n = x_n) = \mathbb{P}(X_1 = x_1) \cdots \mathbb{P}(X_n = x_n), \quad (4.1)$$

for all x_1, x_2, \dots, x_n .

Proof: The necessary condition is obvious: if X_1, \dots, X_n are independent random variables, then $\{X_1 = x_1\}, \dots, \{X_n = x_n\}$ are (mutually) independent events. To prove the sufficient condition, write

$$\mathbb{P}(X_1 \in A_1, \dots, X_n \in A_n) = \sum_{x_1 \in A_1} \cdots \sum_{x_n \in A_n} \mathbb{P}(X_1 = x_1, \dots, X_n = x_n).$$

Then, if (4.1) holds, the multiple sum can be written as

$$\sum_{x_1 \in A_1} \mathbb{P}(X_1 = x_1) \cdots \sum_{x_n \in A_n} \mathbb{P}(X_n = x_n) = \mathbb{P}(X_1 \in A_1) \cdots \mathbb{P}(X_n \in A_n),$$

which implies that X_1, \dots, X_n are independent random variables. \square

■ **Example 4.2 (Box with Same Dice)** We repeat the experiment in Example 4.1 with three ordinary fair dice. What are now the joint probabilities in the table? Since the events $\{X = x\}$ and $\{Y = y\}$ are now independent, each entry in the pmf table is $\frac{1}{3} \times \frac{1}{6}$. Clearly in the first experiment not *all* events $\{X = x\}$ and $\{Y = y\}$ are independent (why not?). ■

■ **Example 4.3 (Coin Flip Experiment)** Consider the experiment where we flip a coin n times. We can model this experiments in the following way. For $i = 1, \dots, n$ let X_i be the result of the i th toss: $\{X_i = 1\}$ means Heads, $\{X_i = 0\}$ means Tails. Also, let

$$\mathbb{P}(X_i = 1) = p = 1 - \mathbb{P}(X_i = 0), \quad i = 1, 2, \dots, n.$$

Thus, p can be interpreted as the probability of Heads, which may be known or unknown. Finally, assume that X_1, \dots, X_n are *independent*.

This completely specifies our model. In particular we can find any probability related to the X_i 's. For example, let $X = X_1 + \cdots + X_n$ be the total number of Heads in n tosses. Obviously X is a random variable that takes values between 0 and n . Denote by A the set of all binary vectors $\mathbf{x} = (x_1, \dots, x_n)$ such that $\sum_{i=1}^n x_i = k$. Note that A has $\binom{n}{k}$ elements. We now have

$$\begin{aligned} \mathbb{P}(X = k) &= \sum_{\mathbf{x} \in A} \mathbb{P}(X_1 = x_1, \dots, X_n = x_n) \\ &= \sum_{\mathbf{x} \in A} \mathbb{P}(X_1 = x_1) \cdots \mathbb{P}(X_n = x_n) = \sum_{\mathbf{x} \in A} p^k (1-p)^{n-k} \\ &= \binom{n}{k} p^k (1-p)^{n-k}. \end{aligned}$$

In other words, $X \sim \text{Bin}(n, p)$. Compare this to what we did in Example 1.15 on page 33. ■

■ **Remark 4.1 (Independence of Random Variables)** If f_{X_1, \dots, X_n} denotes the joint pmf of X_1, \dots, X_n and f_{X_i} the marginal pmf of X_i , $i = 1, \dots, n$, then the theorem above states that independence of the X_i 's is equivalent to

$$f_{X_1, \dots, X_n}(x_1, \dots, x_n) = f_{X_1}(x_1) \cdots f_{X_n}(x_n)$$

for all possible x_1, \dots, x_n . ■

■ **Remark 4.2 (Independence for an Infinite Sequence)** An *infinite* sequence X_1, X_2, \dots of random variables is called independent if for any finite choice of parameters i_1, i_2, \dots, i_n (none of them the same) the random variables X_{i_1}, \dots, X_{i_n} are independent. ■

4.1.1.1 Multinomial Distribution

An important discrete joint distribution is the multinomial distribution. It can be viewed as a generalization of the binomial distribution. First we give the definition, then an example how this distribution arises in applications.

Definition 4.3: Multinomial Distribution

We say that (X_1, X_2, \dots, X_k) has a **multinomial** distribution, with parameters n and p_1, p_2, \dots, p_k , if

$$\mathbb{P}(X_1 = x_1, \dots, X_k = x_k) = \frac{n!}{x_1! x_2! \cdots x_k!} p_1^{x_1} p_2^{x_2} \cdots p_k^{x_k}, \quad (4.2)$$

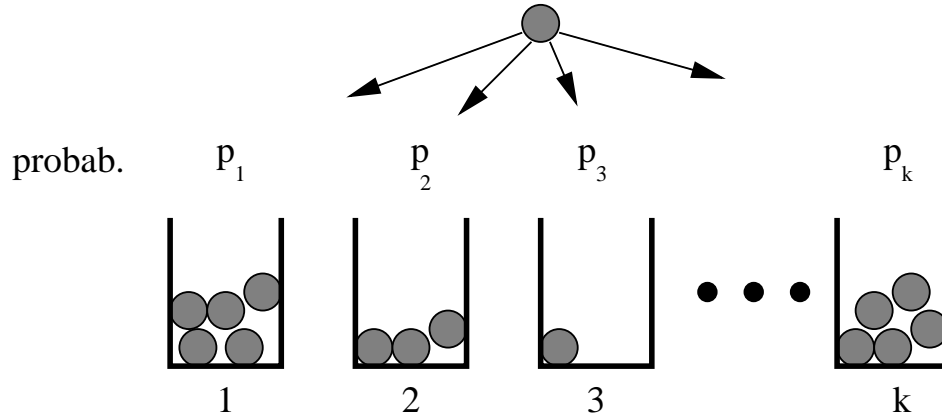
for all $x_1, \dots, x_k \in \{0, 1, \dots, n\}$ such that $x_1 + x_2 + \cdots + x_k = n$. We write $(X_1, \dots, X_k) \sim \text{Mnom}(n, p_1, \dots, p_k)$.

■ **Example 4.4 (Multinomial Distribution)** We independently throw n balls into k urns, such that each ball is thrown in urn i with probability p_i , $i = 1, \dots, k$; see Figure 4.1.

Let X_i be the total number of balls in urn i , $i = 1, \dots, k$. We show that $(X_1, \dots, X_k) \sim \text{Mnom}(n, p_1, \dots, p_k)$. Let x_1, \dots, x_k be integers between 0 and n that sum up to n . The probability that the *first* x_1 balls fall in the first urn, the *next* x_2 balls fall in the second urn, etcetera, is

$$p_1^{x_1} p_2^{x_2} \cdots p_k^{x_k}.$$

To find the probability that there are x_1 balls in the first urn, x_2 in the second, etcetera, we have to multiply the probability above with the number of ways in which we can fill the urns with x_1, x_2, \dots, x_k balls, i.e., $n!/(x_1! x_2! \cdots x_k!)$. This gives (4.2). ■

Figure 4.1: Throwing n balls into k urns.

■ **Remark 4.3 (Binomial Distribution)** Note that for the *binomial* distribution there are only *two* possible urns. Also, note that for each $i = 1, \dots, k$, $X_i \sim \text{Bin}(n, p_i)$. ■

4.1.2 Continuous Joint Distributions

Joint distributions for continuous random variables are usually defined via the joint pdf. The results are very similar to the discrete case discussed in Section 4.1.1. Compare this section also with the 1-dimensional case in Section 2.2.2.

Definition 4.4: Joint Probability Density Function

We say that the continuous random variables X_1, \dots, X_n have a **joint probability density function** (pdf) f if

$$\mathbb{P}(a_1 \leq X_1 \leq b_1, \dots, a_n \leq X_n \leq b_n) = \int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} f(x_1, \dots, x_n) dx_n \cdots dx_1$$

for all a_1, \dots, b_n .

We sometimes write f_{X_1, \dots, X_n} instead of f to show that this is the pdf of the random variables X_1, \dots, X_n . Or, if \mathbf{X} is the corresponding random vector, we could write $f_{\mathbf{X}}$ instead.

We can interpret $f(x_1, \dots, x_n)$ as a continuous analogue of a pmf, or as the “density” that $X_1 = x_1, X_2 = x_2, \dots$, and $X_n = x_n$. For example in the 2-dimensional case:

$$\begin{aligned} \mathbb{P}(x \leq X \leq x+h, y \leq Y \leq y+h) \\ = \int_x^{x+h} \int_y^{y+h} f(u, v) dv du \approx h^2 f(x, y) . \end{aligned}$$

Note that if the joint pdf is given, then in principle we can calculate *all probabilities*. Specifically, in the 2-dimensional case we have

$$\mathbb{P}((X, Y) \in B) = \iint_{(x,y) \in B} f(x, y) dx dy, \quad (4.3)$$

for any subset B of possible values for \mathbb{R}^2 . Thus, the calculation of probabilities is reduced to *integration*.

Similarly to the discrete case, if X_1, \dots, X_n have joint pdf f , then the (individual, or marginal) pdf of each X_i can be found by integrating f over all other variables. For example, in the two-dimensional case

$$f_X(x) = \int_{y=-\infty}^{\infty} f(x, y) dy.$$

However, we usually cannot reconstruct the joint pdf from the marginal pdf's unless we assume that the random variables are *independent*. The definition of independence is exactly the same as for discrete random variables, see Definition 4.2. But, more importantly, we have the following analogue of Theorem 4.1.

Theorem 4.2: Independent Continuous Random Variables

Let X_1, \dots, X_n be continuous random variables with joint pdf f and marginal pdf's f_{X_1}, \dots, f_{X_n} . The random variables X_1, \dots, X_n are independent if and only if, for all x_1, \dots, x_n ,

$$f(x_1, \dots, x_n) = f_{X_1}(x_1) \cdots f_{X_n}(x_n). \quad (4.4)$$

■ **Example 4.5 (Uniform Numbers)** Consider the experiment where we select randomly and independently n points from the interval $[0,1]$. We can carry this experiment out using a calculator or computer, using the *random generator*. On your calculator this means pushing the RAN# or Rand button. Here is a possible outcome, or **realisation**, of the experiment, for $n = 12$.

0.9451226800	0.2920864820	0.0019900900	0.8842189383	0.8096459523
0.3503489150	0.9660027079	0.1024852543	0.7511286891	0.9528386400
0.2923353821	0.0837952423			

A model for this experiment is: Let X_1, \dots, X_n be independent random variables, each with a uniform distribution on $[0,1]$. The joint pdf of X_1, \dots, X_n is very simple, namely

$$f(x_1, \dots, x_n) = 1, \quad 0 \leq x_1 \leq 1, \dots, \quad 0 \leq x_n \leq 1,$$

(and 0 else). In principle we can now calculate any probability involving the X_i 's. For example for the case $n = 2$ what is the probability

$$\mathbb{P}\left(\frac{X_1 + X_2^2}{X_1 X_2} > \sin(X_1^2 - X_2)\right)?$$

The answer, by (4.3), is

$$\iint_A 1 \, dx_1 \, dx_2 = \text{Area}(A),$$

where

$$A = \left\{ (x_1, x_2) \in [0, 1]^2 : \frac{x_1 + x_2^2}{x_1 x_2} > \sin(x_1^2 - x_2) \right\}.$$

(Here $[0, 1]^2$ is the unit square in \mathbb{R}^2).

■

■ **Remark 4.4 (Random Sample)** The type of model used in the previous example, i.e., X_1, \dots, X_n are independent and all have the same distribution, is the most widely used model in statistics. We say that X_1, \dots, X_n is a **random sample** of **size** n , from some given distribution. In Example 4.5 X_1, \dots, X_n is a random sample from a $\mathcal{U}[0, 1]$ -distribution. In Example 4.3 we also had a random sample, this time from a $\text{Ber}(p)$ -distribution. The common distribution of a random sample is sometimes called the **sampling distribution**.

Using the computer we can generate the outcomes of random samples from many (sampling) distributions. In Figure 4.2 the outcomes of a two random samples, both of size 1000 are depicted in a **histogram**. Here the x-axis is divided into 20 intervals, and the number of points in each interval is counted. The first sample is from the $\mathcal{U}[0, 1]$ -distribution, and the second sample is from the $\mathcal{N}(1/2, 1/12)$ -distribution. The Python commands are:

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.random import rand, randn
N=10**3
x = rand(N)
plt.hist(x, bins=20, facecolor='blue', edgecolor='black')

y = 0.5 + randn(N)*np.sqrt(1/12)
plt.hist(y, bins=20, facecolor='blue', edgecolor='black')
```

Note that the true expectation and variance of the distributions are the same. However, the “density” of points in the two samples is clearly different, and follows that shape of the corresponding pdf’s.

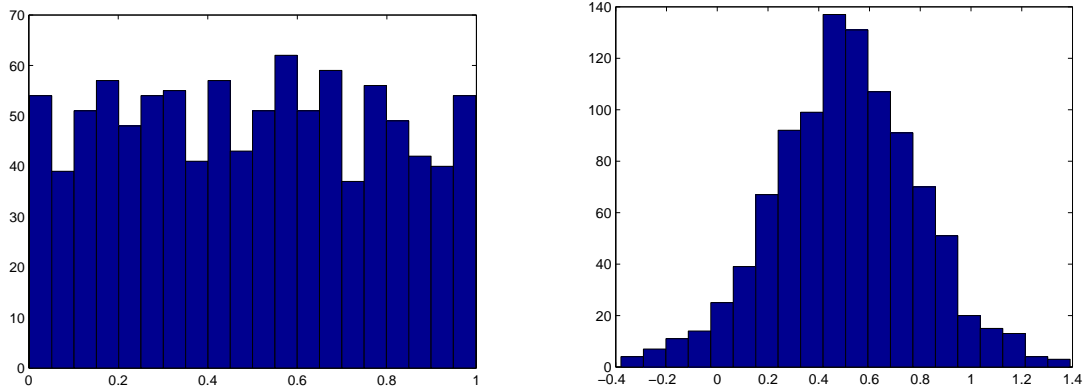


Figure 4.2: A histogram of a random sample of size 1000 from the $\mathcal{U}[0, 1]$ -distribution (left) and the $\mathcal{N}(1/2, 1/12)$ -distribution (right).

4.2 Expectation

Similar to the 1-dimensional case, the expected value of any real-valued function of X_1, \dots, X_n is the weighted average of all values that this function can take. Specifically, if $Z = g(X_1, \dots, X_n)$ then in the discrete case

$$\mathbb{E}Z = \sum_{x_1} \cdots \sum_{x_n} g(x_1, \dots, x_n) f(x_1, \dots, x_n),$$

where f is the joint pmf; and in the continuous case

$$\mathbb{E}Z = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(x_1, \dots, x_n) f(x_1, \dots, x_n) dx_1 \cdots dx_n,$$

where f is the joint pdf.

■ **Example 4.6 (Expectation of the Sum of Two Random Variables)** Let X and Y be continuous, possibly *dependent*, random variables with joint pdf f . Then,

$$\begin{aligned} \mathbb{E}(X + Y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x + y) f(x, y) dx dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f(x, y) dx dy + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y f(x, y) dx dy \\ &= \int_{-\infty}^{\infty} x f_X(x) dx + \int_{-\infty}^{\infty} y f_Y(y) dy \\ &= \mathbb{E}X + \mathbb{E}Y. \end{aligned}$$

The previous example is easily generalized to the following result:

Theorem 4.3

Suppose X_1, X_2, \dots, X_n are discrete or continuous random variables with means $\mu_1, \mu_2, \dots, \mu_n$. Let

$$Y = a + b_1X_1 + b_2X_2 + \dots + b_nX_n$$

where a, b_1, b_2, \dots, b_n are constants. Then

$$\begin{aligned}\mathbb{E}Y &= a + b_1\mathbb{E}X_1 + \dots + b_n\mathbb{E}X_n \\ &= a + b_1\mu_1 + \dots + b_n\mu_n.\end{aligned}$$

■ **Example 4.7 (Expectation of $\text{Bin}(n, p)$ and $\text{Hyp}(n, r, N)$)** We can now prove that the expectation of a $\text{Bin}(n, p)$ random variable is np , without having to resort to difficult arguments. Namely, if $X \sim \text{Bin}(n, p)$, then X can be written as the sum $X_1 + \dots + X_n$ of iid $\text{Ber}(p)$ random variables, see Example 4.3. Thus,

$$\mathbb{E}X = \mathbb{E}(X_1 + \dots + X_n) = \mathbb{E}X_1 + \dots + \mathbb{E}X_n = \underbrace{p + \dots + p}_{n \text{ times}} = np, \quad (4.5)$$

because the expectation of each X_i is p .

Notice that we do not use anywhere the independence of the $\{X_i\}$. Now, let $X \sim \text{Hyp}(n, r, N)$, and let $p = r/N$. We can think of X as the total number of red balls when n balls are drawn from an urn with r red balls and $N - r$ other balls. Without loss of generality we may assume the balls are drawn one-by-one. Let $X_i = 1$ if the i -th balls is red, and 0 otherwise. Then, again $X_1 + \dots + X_n$, and each $X_i \sim \text{Ber}(p)$, but now the $\{X_i\}$ are *dependent*. However, this does not affect the result (4.5), so that the expectation of X is $np = nr/N$. ■

Another important result is the following.

Theorem 4.4

If X_1, \dots, X_n are *independent*, then

$$\mathbb{E}X_1X_2 \dots X_n = \mathbb{E}X_1\mathbb{E}X_2 \dots \mathbb{E}X_n.$$

Proof: We prove it only for the 2-dimensional continuous case. Let f denote the joint pdf of X and Y , and f_X and f_Y the marginals. Then, $f(x, y) = f_X(x)f_Y(y)$ for all x, y . Thus

$$\begin{aligned}\mathbb{E}XY &= \iint xy f(x, y) dx dy = \iint xy f_X(x) f_Y(y) dx dy \\ &= \int x f_X(x) dx \int y f_Y(y) dy = \mathbb{E}X\mathbb{E}Y.\end{aligned}$$

The generalization to the n -dimensional continuous/discrete case is obvious. \square

Theorem 4.4 is particularly handy in combination with transform techniques. We give two examples.

■ **Example 4.8 (Sum of Poisson Random Variables)** Let $X \sim \text{Poi}(\lambda)$, then we saw in Example 2.8 on page 45 that its PGF is given by

$$G(z) = e^{-\lambda(1-z)}. \quad (4.6)$$

Now let $Y \sim \text{Poi}(\mu)$ be independent of X . Then, the PGF of $X + Y$ is given by

$$\mathbb{E}z^{X+Y} = \mathbb{E}z^X \mathbb{E}z^Y = e^{-\lambda(1-z)} e^{-\mu(1-z)} = e^{-(\lambda+\mu)(1-z)}.$$

Thus, by the uniqueness property of the PGF, $X + Y \sim \text{Poi}(\lambda + \mu)$. \blacksquare

■ **Example 4.9 (Sum of Gamma Random Variables)** The MGF of $X \sim \text{Gamma}(\alpha, \lambda)$ is given, see (2.13) on page 60, by

$$\mathbb{E}e^{sX} = \left(\frac{\lambda}{\lambda - s} \right)^\alpha.$$

As a special case, the moment generating function of the $\text{Exp}(\lambda)$ distribution is given by $\lambda/(\lambda - s)$. Now let X_1, \dots, X_n be iid $\text{Exp}(\lambda)$ random variables. The MGF of $S_n = X_1 + \dots + X_n$ is

$$\mathbb{E}e^{sS_n} = \mathbb{E}[e^{sX_1} \dots e^{sX_n}] = \mathbb{E}e^{sX_1} \dots \mathbb{E}e^{sX_n} = \left(\frac{\lambda}{\lambda - s} \right)^n,$$

which shows that $S_n \sim \text{Gamma}(n, \lambda)$. \blacksquare

Definition 4.5: Covariance

The **covariance** of two random variables X and Y is defined as the number

$$\text{Cov}(X, Y) := \mathbb{E}[(X - \mathbb{E}X)(Y - \mathbb{E}Y)].$$

The covariance is a measure for the amount of linear dependency between the variables. If small values of X (smaller than the expected value of X) go together with small values of Y , and at the same time large values of X go together with large values of Y , then $\text{Cov}(X, Y)$ will be *positive*. If on the other hand small values of X go together with large values of Y , and large values of X go together with small values of Y , then $\text{Cov}(X, Y)$ will be *negative*.

For easy reference we list some important properties of the variance and covariance in Table 4.1. The proofs follow directly from the definitions of covariance and variance and the properties of the expectation.

Proof: Properties 1. and 2. were already proved on page 45. Properties 4. and 6. follow directly from the definitions of covariance and variance. Denote for convenience $\mu_X = \mathbb{E}X$ and $\mu_Y = \mathbb{E}Y$.

Table 4.1: Properties of variance and covariance

1.	$\text{Var}(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2.$
2.	$\text{Var}(aX + b) = a^2 \text{Var}(X).$
3.	$\text{Cov}(X, Y) = \mathbb{E}XY - \mathbb{E}X\mathbb{E}Y.$
4.	$\text{Cov}(X, Y) = \text{Cov}(Y, X).$
5.	$\text{Cov}(aX + bY, Z) = a \text{Cov}(X, Z) + b \text{Cov}(Y, Z).$
6.	$\text{Cov}(X, X) = \text{Var}(X).$
7.	$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2 \text{Cov}(X, Y).$
8.	$X \text{ and } Y \text{ indep.} \implies \text{Cov}(X, Y) = 0.$

$$3. \text{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = \mathbb{E}[XY - Y\mu_X - X\mu_Y + \mu_X\mu_Y] = \mathbb{E}XY - \mu_Y\mu_X - \mu_X\mu_Y + \mu_X\mu_Y = \mathbb{E}XY - \mu_X\mu_Y.$$

5. Using property 3. we get

$$\begin{aligned} \text{Cov}(aX + bY, Z) &= \mathbb{E}[(aX + bY)Z] - \mathbb{E}[aX + bY] \mathbb{E}[Z] \\ &= a \mathbb{E}[XZ] + b \mathbb{E}[YZ] - a \mathbb{E}[X] \mathbb{E}[Z] - b \mathbb{E}[Y] \mathbb{E}[Z] \\ &= a\{\mathbb{E}[XZ] - \mathbb{E}[X]\mathbb{E}[Z]\} + b\{\mathbb{E}[YZ] - \mathbb{E}[Y]\mathbb{E}[Z]\} \\ &= a \text{Cov}(X, Z) + b \text{Cov}(Y, Z). \end{aligned}$$

7. By property 6. we have

$$\text{Var}(X + Y) = \text{Cov}(X + Y, X + Y).$$

By property 5. we can expand this to

$$\begin{aligned} \text{Cov}(X + Y, X + Y) &= \text{Cov}(X, X + Y) + \text{Cov}(Y, X + Y) \\ &= \text{Cov}(X, X) + \text{Cov}(X, Y) + \text{Cov}(Y, X) + \text{Cov}(Y, Y), \end{aligned}$$

where we have also used the symmetry property 4. to expand the second argument of the covariance. Now, by properties 4. and 6. we can simplify the above sum to $\text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y)$, which had to be shown.

8. If X and Y are independent, then $\mathbb{E}XY = \mathbb{E}X\mathbb{E}Y$, so that $\text{Cov}(X, Y) = \mathbb{E}XY - \mathbb{E}X\mathbb{E}Y = 0$.

□

A scaled version of the covariance is given by the **correlation coefficient**.

$$\varrho(X, Y) := \frac{\mathbb{C}\text{ov}(X, Y)}{\sqrt{\mathbb{V}\text{ar}(X)} \sqrt{\mathbb{V}\text{ar}(Y)}}.$$

Theorem 4.5: Correlation coefficient

The correlation coefficient always lies between -1 and 1 .

Proof: Let a be an arbitrary real number, and denote the standard deviations of X and Y by σ_X and σ_Y . Obviously the variance of $\pm aX + Y$ is always non-negative. Thus, using the properties of covariance and variance

$$\mathbb{V}\text{ar}(-aX + Y) = a^2 \sigma_X^2 + \sigma_Y^2 - 2a \mathbb{C}\text{ov}(X, Y) \geq 0.$$

So that after rearranging and dividing by $\sigma_X \sigma_Y$, we obtain

$$\varrho(X, Y) \leq \frac{1}{2} \left(\frac{a \sigma_X}{\sigma_Y} + \frac{\sigma_Y}{a \sigma_X} \right).$$

Similarly,

$$\mathbb{V}\text{ar}(aX + Y) = a^2 \sigma_X^2 + \sigma_Y^2 + 2a \mathbb{C}\text{ov}(X, Y) \geq 0,$$

so that

$$\varrho(X, Y) \geq -\frac{1}{2} \left(\frac{a \sigma_X}{\sigma_Y} + \frac{\sigma_Y}{a \sigma_X} \right).$$

By choosing $a = \sigma_Y / \sigma_X$, we see that $-1 \leq \varrho(X, Y) \leq 1$. □

In Figure 4.3 an illustration of the correlation coefficient is given. Each figure corresponds to samples of size 40 from a different 2-dimensional distribution. In each case $\mathbb{E}X = \mathbb{E}Y = 0$ and $\mathbb{V}\text{ar}(X) = \mathbb{V}\text{ar}(Y) = 1$.

As a consequence of properties 2. and 7., we have

Theorem 4.6

Suppose X_1, X_2, \dots, X_n are discrete or continuous *independent* random variables with variances $\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2$. Let

$$Y = a + b_1 X_1 + b_2 X_2 + \dots + b_n X_n$$

where a, b_1, b_2, \dots, b_n are constants. Then

$$\begin{aligned} \mathbb{V}\text{ar}(Y) &= b_1^2 \mathbb{V}\text{ar}(X_1) + \dots + b_n^2 \mathbb{V}\text{ar}(X_n) \\ &= b_1^2 \sigma_1^2 + \dots + b_n^2 \sigma_n^2 \end{aligned}$$

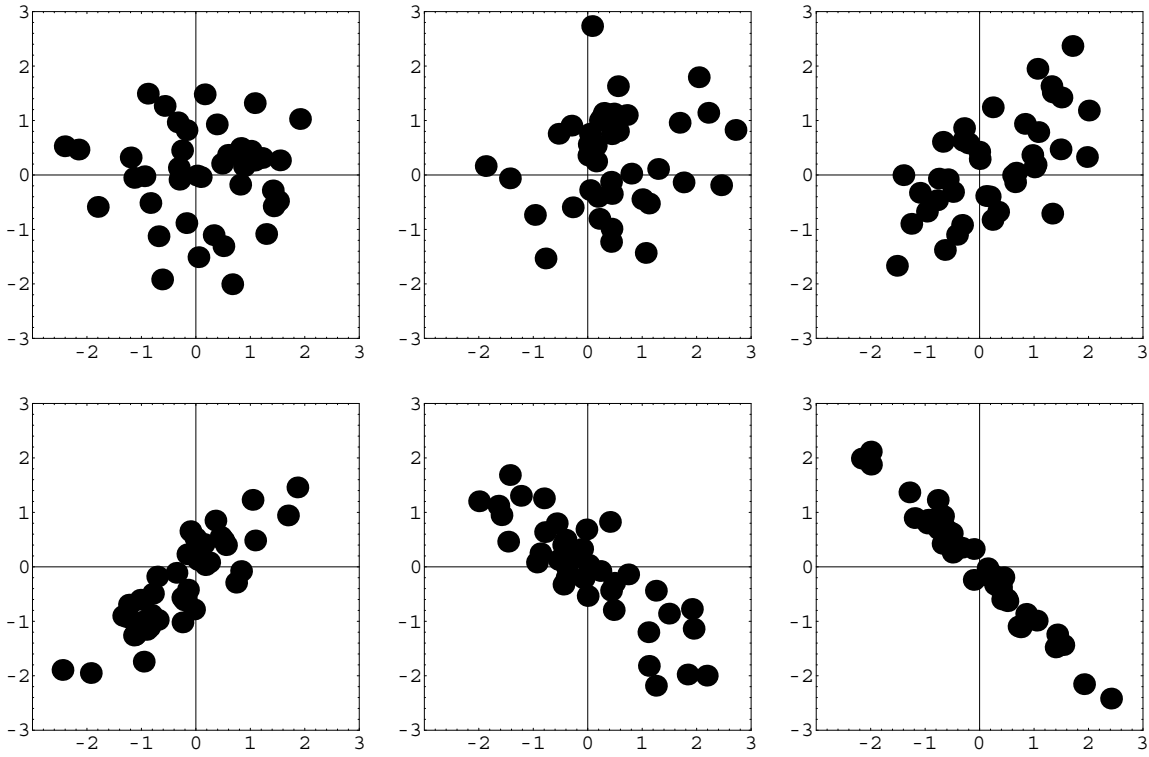


Figure 4.3: *Illustration of correlation coefficient.* Above: $\rho = 0$, $\rho = 0.4$, $\rho = 0.7$. Below: $\rho = 0.9$, $\rho = -0.8$, $\rho = -0.98$.

Proof: By virtue of property 6., and repetitive application of property 5., we have (note that the constant a does not play a role in the variance):

$$\begin{aligned}\text{Var}(Y) &= \mathbb{Cov}(b_1X_1 + b_2X_2 + \cdots + b_nX_n, b_1X_1 + b_2X_2 + \cdots + b_nX_n) \\ &= \sum_{i=1}^n \mathbb{Cov}(b_iX_i, b_iX_i) + 2 \sum_{i < j} \mathbb{Cov}(b_iX_i, b_jX_j) .\end{aligned}$$

Since $\mathbb{Cov}(b_iX_i, b_iX_i) = b_i^2 \text{Var}(X_i)$ and all covariance term are zero because of the independence of X_i and X_j ($i \neq j$), the result follows. \square

■ **Example 4.10 (Variance of $\text{Bin}(n, p)$ and $\text{Hyp}(n, r, N)$)** Consider again Example 4.7 where we derived the expectation of $X \sim \text{Bin}(n, p)$ and $X \sim \text{Hyp}(n, r, N)$ by writing X as

$$X = X_1 + \cdots + X_n$$

of independent (in the binomial case) or dependent (in the hypergeometric case) $\text{Ber}(p)$ random variables, where $p = r/N$ in the hypergeometric case. Using Theorem 4.6, the variance of the binomial distribution follows directly from

$$\text{Var}(X) = \text{Var}(X_1) + \cdots + \text{Var}(X_n) = n\text{Var}(X_1) = np(1 - p).$$

For the hypergeometric case must include the covariance terms as well:

$$\mathbb{V}\text{ar}(X) = \mathbb{V}\text{ar}(X_1) + \cdots + \mathbb{V}\text{ar}(X_n) + 2 \sum_{i < j} \mathbb{C}\text{ov}(X_i, X_j) .$$

By symmetry all the $\mathbb{C}\text{ov}(X_i, X_j)$ are the same ($i \neq j$). Hence,

$$\mathbb{V}\text{ar}(X) = n\mathbb{V}\text{ar}(X_1) + n(n-1)\mathbb{C}\text{ov}(X_1, X_2) .$$

Since $\mathbb{V}\text{ar}(X_1) = p(1-p)$, and $\mathbb{C}\text{ov}(X_1, X_2) = \mathbb{E}X_1X_2 - p^2$, it remains to find $\mathbb{E}X_1X_2 = \mathbb{P}(X_1 = 1, X_2 = 1)$, which is

$$\mathbb{P}(X_1 = 1, X_2 = 1) = \mathbb{P}(X_1 = 1)\mathbb{P}(X_2 = 1 | X_1 = 1) = p \frac{r-1}{N-1} .$$

Simplifying gives,

$$\mathbb{V}\text{ar}(X) = np(1-p) \frac{N-n}{N-1} .$$

■

4.2.0.1 Expectation Vector and Covariance Matrix

Let $\mathbf{X} = (X_1, \dots, X_n)^\top$ be a random vector. Sometimes it is convenient to write the expectations and covariances in vector notation.

Definition 4.6: Expectation Vector and Covariance Matrix

For any random vector \mathbf{X} we define the **expectation vector** as the vector of expectations

$$\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^\top := (\mathbb{E}X_1, \dots, \mathbb{E}X_n)^\top .$$

The **covariance matrix** Σ is defined as the matrix whose (i, j) th element is

$$\mathbb{C}\text{ov}(X_i, X_j) = \mathbb{E}(X_i - \mu_i)(X_j - \mu_j) .$$

If we define the expectation of a vector (matrix) to be the vector (matrix) of expectations, then we can write:

$$\boldsymbol{\mu} = \mathbb{E}\mathbf{X}$$

and

$$\Sigma = \mathbb{E}(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^\top .$$

Note that $\boldsymbol{\mu}$ and Σ take the same role as μ and σ^2 in the 1-dimensional case. We sometimes write $\boldsymbol{\mu}_X$ and Σ_X if we wish to emphasise that $\boldsymbol{\mu}$ and Σ belong to the vector \mathbf{X} .

■ **Remark 4.5 (Positive Semidefinite)** Note that any covariance matrix Σ is a *symmetric* matrix. In fact, it is *positive semidefinite*, i.e., for any (column) vector \mathbf{u} , we have

$$\mathbf{u}^\top \Sigma \mathbf{u} \geq 0.$$

To see this, suppose Σ is the covariance matrix of some random vector \mathbf{X} with expectation vector $\boldsymbol{\mu}$. Write $\mathbf{Y} = \mathbf{X} - \boldsymbol{\mu}$. Then

$$\begin{aligned} \mathbf{u}^\top \Sigma \mathbf{u} &= \mathbf{u}^\top \mathbb{E} \mathbf{Y} \mathbf{Y}^\top \mathbf{u} = \mathbb{E} \mathbf{u}^\top \mathbf{Y} \mathbf{Y}^\top \mathbf{u} \\ &= \mathbb{E} (\mathbf{Y}^\top \mathbf{u})^\top \mathbf{Y}^\top \mathbf{u} = \mathbb{E} (\mathbf{Y}^\top \mathbf{u})^2 \geq 0. \end{aligned}$$

Note that $\mathbf{Y}^\top \mathbf{u}$ is a random variable. ■

4.3 Conditional Distribution

Suppose X and Y are both discrete or both continuous, with joint pmf/pdf $f_{X,Y}$, and suppose $f_X(x) > 0$. The *conditional pdf/pmf* of Y given $X = x$ is defined as

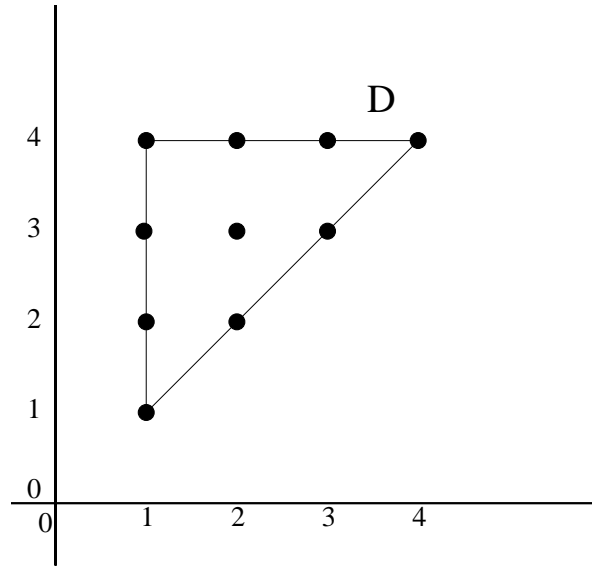
$$f_{Y|X}(y|x) = \frac{f_{X,Y}(x,y)}{f_X(x)}, \text{ for all } y. \quad (4.7)$$

We can interpret $f_{Y|X}(\cdot|x)$ as the pmf/pdf of Y given the information that X takes the value x . For discrete random variables the definition is simply a consequence or rephrasing of the conditioning formula (1.4) on page 27. Namely,

$$f_{Y|X}(y|x) = \mathbb{P}(Y = y | X = x) = \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(X = x)} = \frac{f_{X,Y}(x,y)}{f_X(x)}.$$

■ **Example 4.11 (Uniformly Drawing from a Discrete Triangle)** We draw “uniformly” a point (X, Y) from the 10 points on the triangle D below. Thus, each point is equally likely to be drawn. That is, the joint pmf is

$$f_{X,Y}(x,y) = \mathbb{P}(X = x, Y = y) = \frac{1}{10}, \quad (x,y) \in D,$$



The marginal pmf's of X and Y are

$$f_X(x) = \mathbb{P}(X = x) = \frac{5-x}{10}, \quad x \in \{1, 2, 3, 4\},$$

and

$$f_Y(y) = \mathbb{P}(Y = y) = \frac{y}{10}, \quad y \in \{1, 2, 3, 4\}.$$

Clearly X and Y are not independent. In fact, if we know that $X = 2$, then Y can only take the values $j = 2, 3$ or 4 . The corresponding probabilities are

$$\mathbb{P}(Y = y | X = 2) = \frac{\mathbb{P}(Y = y, X = 2)}{\mathbb{P}(X = 2)} = \frac{1/10}{3/10} = \frac{1}{3}.$$

In other words, the conditional pmf of Y given $X = 2$ is

$$f_{Y|X}(y|2) = \frac{f_{X,Y}(2,y)}{f_X(2)} = \frac{1}{3}, \quad y = 2, 3, 4.$$

Thus, given $X = 2$, Y takes the values 2,3 and 4 with equal probability. ■

When X is *continuous*, we can not longer directly apply (1.4) to define the conditional density. Instead, we define first the **conditional cdf** of Y given $X = x$ as the limit

$$F_{Y|X}(y|x) := \lim_{h \rightarrow 0} F_Y(y | x < X \leq x + h).$$

Now, (1.4) can be applied to $F_Y(y | x < X \leq x + h)$ to yield

$$F_Y(y | x < X \leq x + h) = \frac{\int_{-\infty}^y \int_x^{x+h} f_{X,Y}(u,v) \, du \, dv}{\int_x^{x+h} f_X(u) \, du}$$

Now, for small h the integral $\int_x^{x+h} f_{X,Y}(u, v) du$ is approximately equal to $h f_{X,Y}(x, v)$ plus some small term which goes to zero faster than h . Similarly, $\int_x^{x+h} f_X(u) du \approx h f_X(x)$ (plus smaller order terms). Hence, for $h \rightarrow 0$, the limit of $F_Y(y | x < X \leq x + h)$ is

$$F_{Y|X}(y | X) = \frac{\int_{-\infty}^y f_{X,Y}(x, v) dv}{f_X(x)}.$$

Note that $F_{Y|X}(y | x)$ as a function of y has all the properties of a cdf. By differentiating this cdf with respect to y we obtain the conditional pdf of Y given $X = x$ in the continuous case, which gives the same formula (4.7) as for the discrete case.

Rewriting (4.7), we find the “product rule” for densities:

$$f_{X,Y}(x, y) = f_X(x) f_{Y|X}(y | x),$$

and by integrating out x , we have (in the continuous case) the following of the law of total probability for probability densities:

$$f_Y(y) = \int f_{Y|X}(y | x) f_X(x) dx. \quad (4.8)$$

Conditional pmf’s and pdf’s for more than two random variables are defined analogously. For example, the conditional pmf of X_n given X_1, \dots, X_{n-1} is given by

$$f_{X_n | X_1, \dots, X_{n-1}}(x_n | x_1, \dots, x_{n-1}) = \frac{\mathbb{P}(X_1 = x_1, \dots, X_n = x_n)}{\mathbb{P}(X_1 = x_1, \dots, X_{n-1} = x_{n-1})}.$$

4.4 Conditional Expectation

Since a conditional pmf (pdf) has all the properties of a probability mass (density) function, it makes sense to define the corresponding *conditional expectation* (in the continuous case) as

$$\mathbb{E}[Y | X = x] = \int y f_{Y|X}(y | x) dy.$$

For discrete X and Y we have instead:

$$\mathbb{E}[Y | X = x] = \sum_y y \mathbb{P}(Y = y | X = x).$$

Note that $y \mapsto \mathbb{P}(Y = y | X = x)$ is a genuine pmf for each fixed x .

In both the discrete and continuous case, $\mathbb{E}[Y | X = x]$ is a function of x , say $h(x)$. The corresponding random variable $h(X)$ is written as $\mathbb{E}[Y | X]$. One of the most useful results in

probability is that its expectation is simply the expectation of Y . It simplifies the law of total probability calculations for random variables as in (4.8).

Theorem 4.7: Tower Property

For any pair of random variables X, Y , it holds that

$$\mathbb{E}[\mathbb{E}[Y | X]] = \mathbb{E}[Y]. \quad (4.9)$$

Proof: We prove it for the case where X and Y are jointly continuous. Then, for x such that $f_X(x) > 0$,

$$\mathbb{E}[Y | X = x] = h(x) = \int_{-\infty}^{\infty} y f_{Y|X}(y | x) dy,$$

where $f_{Y|X}(\cdot | x)$ is the pdf of Y given X . It follows that

$$\begin{aligned} \mathbb{E}h(X) &= \int_{-\infty}^{\infty} h(x) f_X(x) dx = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} y f_{Y|X}(y | x) dy \right) f_X(x) dx \\ &= \int_{-\infty}^{\infty} y \left(\underbrace{\int_{-\infty}^{\infty} f_{Y|X}(y | x) f_X(x) dx}_{\text{by (4.8)}} \right) dy = \int_{-\infty}^{\infty} y f_Y(y) dy = \mathbb{E}Y. \end{aligned}$$

□

■ **Example 4.12 (Random Sums)** A useful application of conditional expectation arises when one needs to determine the expected value of a *random* sum of independent and identically distributed (iid) random variables.

Let X_1, X_2, \dots be a sequence of iid random variables with common mean μ , and let

$$S_n = \begin{cases} 0 & \text{if } n = 0 \\ \sum_{i=1}^n X_i & \text{if } n \geq 1. \end{cases}$$

Then, clearly $\mathbb{E}S_n = n\mu$.

Now let N be a random variable which takes values in the non-negative integers. Then,

$$\mathbb{E}S_N = \mathbb{E}(\mathbb{E}[S_N | N]) = \mathbb{E}(N\mu) = \mu \mathbb{E}N.$$

■

FUNCTIONS OF RANDOM VARIABLES AND LIMIT THEOREMS

Suppose X_1, \dots, X_n are the measurements on a random experiment. Often we are interested in certain *functions* of the measurements only, rather than all measurements themselves. For example, if X_1, \dots, X_n are the repeated measurements of the strength of a certain type of fishing line, then what we are really interested in is not the individual values for X_1, \dots, X_n but rather quantities such as the average strength $(X_1 + \dots + X_n)/n$, the minimum strength $\min(X_1, \dots, X_n)$ and the maximum strength $\max(X_1, \dots, X_n)$. Note that these quantities are again random variables. The distribution of these random variables can in principle be derived from the joint distribution of the X_i 's.

5.1 Functions of Random Variables

In this section we list a number of examples of functions of random variables as well as techniques to derive their probability distributions. As functions of discrete random variables are more easy to deal with, the examples will focus on continuous distributions.

■ **Example 5.1 (Affine Transformation)** Let X be a continuous random variable with pdf f_X , and let $Y = aX + b$, where $a \neq 0$. This is called an **affine transformation** of X . We wish to determine the pdf f_Y of Y . We first express the cdf of Y in terms of the cdf of X . Suppose first that $a > 0$. We have for any y

$$F_Y(y) = \mathbb{P}(Y \leq y) = \mathbb{P}(X \leq (y - b)/a) = F_X((y - b)/a) .$$

Differentiating this with respect to y gives $f_Y(y) = f_X((y - b)/a) / a$. For $a < 0$ we get similarly

$f_Y(y) = f_X((y - b)/a) / (-a)$. Thus in general

$$f_Y(y) = \frac{1}{|a|} f_X\left(\frac{y - b}{a}\right) . \quad (5.1)$$

■ **Example 5.2 (Square of a Standard Normal)** Let $X \sim \mathcal{N}(0, 1)$. We wish to determine the distribution of $Y = X^2$. We can use the same technique as in the example above, but note first that Y can only take values in $[0, \infty)$. For $y > 0$ we have

$$\begin{aligned} F_Y(y) &= \mathbb{P}(Y \leq y) = \mathbb{P}(-\sqrt{y} \leq X \leq \sqrt{y}) \\ &= F_X(\sqrt{y}) - F_X(-\sqrt{y}) = 2F_X(\sqrt{y}) - 1 . \end{aligned}$$

Differentiating this with respect to y gives

$$\begin{aligned} f_Y(y) &= 2 f_X(\sqrt{y}) \frac{1}{2\sqrt{y}} = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(\sqrt{y})^2\right) \frac{1}{\sqrt{y}} \\ &= \frac{(1/2)^{1/2} y^{-1/2} e^{-y/2}}{\Gamma(1/2)} . \end{aligned}$$

This is exactly the formula for the pdf of a χ_1^2 -distribution. Thus $Y \sim \chi_1^2$. ■

■ **Example 5.3 (Minimum and Maximum)** Suppose X_1, \dots, X_n are independent and have cdf F . Let $Y = \min(X_1, \dots, X_n)$ and $Z = \max(X_1, \dots, X_n)$. The cdf of Y and Z are easily obtained. First, note that the maximum of the $\{X_i\}$ is less than some number Z if and only if all X_i are less than z . Thus,

$$\mathbb{P}(Z \leq z) = \mathbb{P}(X_1 \leq z, X_2 \leq z, \dots, X_n \leq z) = \mathbb{P}(X_1 \leq z) \mathbb{P}(X_2 \leq z) \cdots \mathbb{P}(X_n \leq z),$$

where the second equation follows from the independence assumption. It follows that

$$F_Z(z) = (F(z))^n .$$

Similarly,

$$\mathbb{P}(Y > y) = \mathbb{P}(X_1 > y, X_2 > y, \dots, X_n > y) = \mathbb{P}(X_1 > y) \mathbb{P}(X_2 > y) \cdots \mathbb{P}(X_n > y),$$

so that

$$F_Y(y) = 1 - (1 - F(y))^n .$$

■ **Example 5.4 (Inverse-Transform Method)** In Chapter 3 we saw an important application of functions of random variables: the inverse-transform method for generating random variables. That is, $U \sim \mathcal{U}(0, 1)$, and let F be continuous and strictly increasing cdf. Then $Y = F^{-1}(U)$ is a random variable that has cdf F . ■

We can use simulation to get an idea of the distribution of a function of one or more random variables, as explained in the following example.

■ **Example 5.5 (Sum of Uniforms)** Let X and Y be independent and both $\mathcal{U}(0, 1)$ distributed. What does the pdf of $Z = X + Y$ look like? Note that Z takes values in $(0, 2)$. The following Python code draws 10,000 times from the distribution of Z and plots a histogram of the data (Figure 5.1):

```
from matplotlib.pyplot import hist
from numpy.random import rand
N=10**4
hist(rand(N)+rand(N), bins=50, edgecolor='black')
```

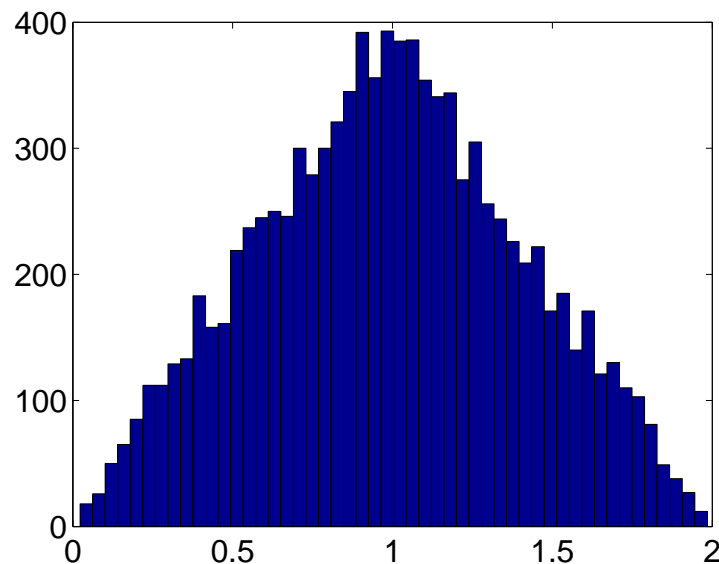


Figure 5.1: Histogram for adding two uniform random variables

This looks remarkably like a triangle. Perhaps the true pdf of $Z = X + Y$ has a triangular shape? This is indeed easily proved. Namely, first observe that the pdf of Z must be symmetrical around 1. Thus to find the pdf, it suffices to find its form for $z \in [0, 1]$. Take such a z and consider the event $\{Z \leq z\}$, which corresponds to (X, Y) lying in the shaded region A of Figure 5.2.

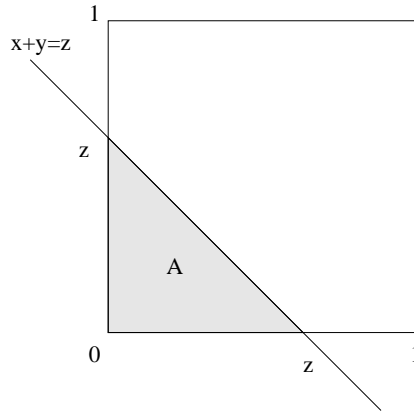


Figure 5.2: The random point (X, Y) must lie in set A

We thus have:

$$F_Z(z) = \mathbb{P}(Z \leq z) = \mathbb{P}((X, Y) \in A) = \iint_A f(x, y) \, dx \, dy = \text{area}(A) = \frac{1}{2} z^2.$$

where we have used the fact that the joint density $f(x, y)$ is equal to 1 on the square $[0, 1] \times [0, 1]$. By differentiating the cdf F_Z we get the pdf f_Z

$$f_Z(z) = z, \quad z \in [0, 1],$$

and by symmetry

$$f_Z(z) = 2 - z, \quad z \in [1, 2],$$

which is indeed a triangular density. If we rescaled the histogram such that the total area under the bars would be 1, the fit with the true distribution would be very good. ■

5.1.1 Sum of Two Random Variables

We can generalize the above procedure to the sum of two random variables with a general joint pdf or pmf $f_{X,Y}$. Let us consider the jointly continuous case only, so X and Y have joint pdf $f_{X,Y}$ and we are interested in finding the pdf of $Z = X + Y$. By conditioning on X , we have (see (4.8)):

$$f_Z(z) = \int_{-\infty}^{\infty} f_{Z|X}(z|x) f_X(x) \, dx.$$

Since $Z = X + Y$, we have

$$\begin{aligned} f_{Z|X}(z|x) &= \lim_{\delta \rightarrow 0} \frac{\mathbb{P}(z \leq Z \leq z + \delta | X = x)}{\delta} \\ &= \lim_{\delta \rightarrow 0} \frac{\mathbb{P}(z - x \leq Y \leq z - x + \delta | X = x)}{\delta} = f_{Y|X}(z - x | x). \end{aligned}$$

Hence, we have found

$$f_Z(z) = \int_{-\infty}^{\infty} f_{Y|X}(z-x|x) f_X(x) dx.$$

In particular, when X and Y are *independent*, we have

$$f_Z(z) = \int_{-\infty}^{\infty} f_Y(z-x) f_X(x) dx = \int_{-\infty}^{\infty} f_X(z-y) f_Y(y) dy,$$

where second equation follows from swapping the roles of X and Y . We say that f_Z is the **convolution** of f_X and f_Y and write this as

$$f_Z = f_X \star f_Y \quad (= f_Y \star f_X).$$

■ **Example 5.6 (Sum of Uniforms (cont.))** Let us derive the pdf of Z for the sum of two independent $\mathcal{U}(0, 1)$ variables via convolution. For $z \in [0, 1]$, we have

$$f_Y(z-x) = \begin{cases} 1 & \text{if } 0 \leq x \leq z \\ 0 & \text{otherwise.} \end{cases}$$

Hence,

$$f_Z(z) = \int_{-\infty}^{\infty} f_Y(z-x) f_X(x) dx = \int_0^1 f_Y(z-x) dx = \int_0^z 1 dx = z.$$

Similarly, for $z \in [1, 2]$, we have

$$f_Y(z-x) = \begin{cases} 1 & \text{if } z-1 \leq x \leq 1 \\ 0 & \text{otherwise,} \end{cases}$$

so that in this case

$$f_Z(z) = \int_{-\infty}^{\infty} f_Y(z-x) f_X(x) dx = \int_0^1 f_Y(z-x) dx = \int_{z-1}^1 1 dx = 2-z.$$

■

5.1.2 Linear Transformations

Let $\mathbf{x} = (x_1, \dots, x_n)^\top$ be a (column) vector in \mathbb{R}^n and A an $(m \times n)$ -matrix. The mapping $\mathbf{x} \mapsto \mathbf{z}$, with

$$\mathbf{z} = A\mathbf{x}$$

is called a **linear transformation**. Now consider a *random* vector $\mathbf{X} = (X_1, \dots, X_n)^\top$, and let

$$\mathbf{Z} = A\mathbf{X}.$$

Then \mathbf{Z} is a random vector in \mathbb{R}^m . Again, in principle, if we know the joint distribution of \mathbf{X} then we can derive the joint distribution of \mathbf{Z} . Let us first see how the expectation vector and covariance matrix are transformed.

Theorem 5.1

If \mathbf{X} has expectation vector $\boldsymbol{\mu}_X$ and covariance matrix Σ_X , then the expectation vector and covariance matrix of $\mathbf{Z} = \mathbf{A}\mathbf{X}$ are respectively given by

$$\boldsymbol{\mu}_Z = \mathbf{A}\boldsymbol{\mu}_X \quad (5.2)$$

and

$$\Sigma_Z = \mathbf{A} \Sigma_X \mathbf{A}^\top . \quad (5.3)$$

Proof: We have $\boldsymbol{\mu}_Z = \mathbb{E}\mathbf{Z} = \mathbb{E}\mathbf{A}\mathbf{X} = \mathbf{A}\mathbb{E}\mathbf{X} = \mathbf{A}\boldsymbol{\mu}_X$ and

$$\begin{aligned} \Sigma_Z &= \mathbb{E}(\mathbf{Z} - \boldsymbol{\mu}_Z)(\mathbf{Z} - \boldsymbol{\mu}_Z)^\top = \mathbb{E}\mathbf{A}(\mathbf{X} - \boldsymbol{\mu}_X)(\mathbf{A}(\mathbf{X} - \boldsymbol{\mu}_X))^\top \\ &= \mathbf{A}\mathbb{E}(\mathbf{X} - \boldsymbol{\mu}_X)(\mathbf{X} - \boldsymbol{\mu}_X)^\top \mathbf{A}^\top \\ &= \mathbf{A} \Sigma_X \mathbf{A}^\top \end{aligned}$$

which completes the proof. \square

From now on assume \mathbf{A} is an *invertible* $(n \times n)$ – *matrix*. If \mathbf{X} has joint density f_X , what is the joint density f_Z of \mathbf{Z} ?

Consider Figure 5.3. For any fixed \mathbf{x} , let $\mathbf{z} = \mathbf{A}\mathbf{x}$. Hence, $\mathbf{x} = \mathbf{A}^{-1}\mathbf{z}$. Consider the n -dimensional cube $C = [z_1, z_1 + h] \times \cdots \times [z_n, z_n + h]$. Let D be the *image* of C under \mathbf{A}^{-1} , i.e., the parallelepiped of all points \mathbf{x} such that $\mathbf{A}\mathbf{x} \in C$. Then,

$$\mathbb{P}(\mathbf{Z} \in C) \approx h^n f_Z(\mathbf{z}) .$$

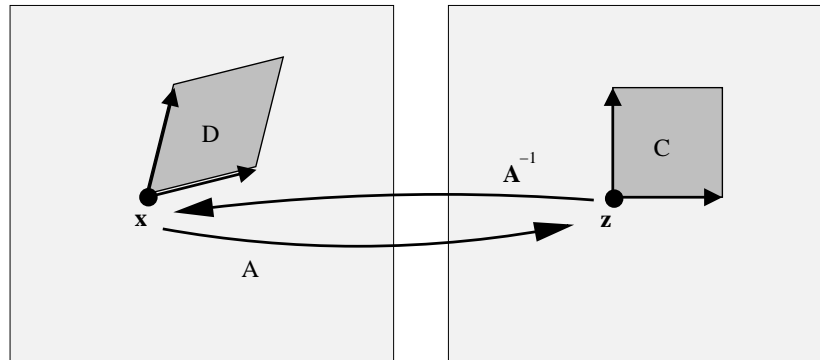


Figure 5.3: Linear transformation

Now recall from linear algebra that any n -dimensional rectangle with “volume” V is transformed into a n -dimensional parallelepiped with volume $V|A|$, where $|A| := |\det(A)|$. Thus,

$$\mathbb{P}(Z \in C) = \mathbb{P}(X \in D) \approx h^n |A|^{-1} f_X(\mathbf{x}) = h^n |A|^{-1} f_X(\mathbf{x})$$

Letting h go to 0 we conclude that

$$f_Z(\mathbf{z}) = \frac{f_X(A^{-1}\mathbf{z})}{|A|}, \quad \mathbf{z} \in \mathbb{R}^n. \quad (5.4)$$

5.1.3 General Transformations

We can apply the same technique as for the linear transformation to general transformations $\mathbf{x} \mapsto \mathbf{g}(\mathbf{x})$, written out:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \mapsto \begin{pmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_n(\mathbf{x}) \end{pmatrix}.$$

For a fixed \mathbf{x} , let $\mathbf{z} = \mathbf{g}(\mathbf{x})$. Suppose \mathbf{g} is invertible, hence, $\mathbf{x} = \mathbf{g}^{-1}(\mathbf{z})$. Any infinitesimal n -dimensional rectangle at \mathbf{x} with volume V is transformed into a n -dimensional parallelepiped at \mathbf{z} with volume $V|\det(J_x(\mathbf{g}))|$, where $J_x(\mathbf{g})$ is the **matrix of Jacobi** at \mathbf{x} of the transformation \mathbf{g} :

$$J_x(\mathbf{g}) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}.$$

Now consider a random column vector $\mathbf{Z} = \mathbf{g}(\mathbf{X})$. Let C be a small cube around \mathbf{z} with volume h^n . Let D be image of C under \mathbf{g}^{-1} . Then, as in the linear case,

$$\mathbb{P}(\mathbf{Z} \in C) \approx h^n f_Z(\mathbf{z}) \approx h^n |\det(J_z(\mathbf{g}^{-1}))| f_X(\mathbf{x}).$$

Hence, we have derived the following transformation rule.

Theorem 5.2: Transformation Rule

Let \mathbf{X} be a continuous random vector with joint pdf f_X , and let \mathbf{g} be an invertible function with Jacobi matrix $J_x(\mathbf{g})$ at \mathbf{x} . Then the random vector $\mathbf{Z} = \mathbf{g}(\mathbf{X})$ has joint pdf f_Z given by

$$f_Z(\mathbf{z}) = \frac{f_X(\mathbf{x})}{|\det(J_x(\mathbf{g}))|} = f_X(\mathbf{x}) |\det(J_z(\mathbf{g}^{-1}))|, \quad \mathbf{z} \in \mathbb{R}^n, \quad (5.5)$$

where $\mathbf{z} = \mathbf{g}(\mathbf{x})$ and $\mathbf{x} = \mathbf{g}^{-1}(\mathbf{z})$.

■ **Remark 5.1 (One-dimensional Transformation Rule)** For the one-dimensional case, $Z = g(X)$, we have that $J_x(g)$ is simply the derivative of g at x ; that is, $g'(x)$. ■

■ **Example 5.7 (Square of a Standard Normal (cont.))** Let $X \sim \mathcal{N}(0, 1)$ and consider the quadratic mapping $g : x \mapsto x^2$ on \mathbb{R} . We cannot directly apply the one-dimensional transformation rule, as g is not invertible. However, this problem can be circumvented by taking the square of the random variable $|X|$ instead. The latter (see (3.13)) has the pdf

$$f_{|X|}(x) = \sqrt{\frac{2}{\pi}} e^{-x^2/2}, \quad x \geq 0.$$

Since X^2 and $|X|^2$ mean the same thing, and the mapping $g : x \mapsto x^2$ is invertible on the interval $[0, \infty)$, we can apply the transformation rule to find the pdf of $Z = g(|X|)$. Taking a $z \geq 0$ and the corresponding $x = \sqrt{z}$, we have

$$f_Z(z) = \frac{f_{|X|}(x)}{g'(x)} = \frac{\sqrt{\frac{2}{\pi}} e^{-x^2/2}}{2x} = \frac{\sqrt{\frac{2}{\pi}} e^{-z/2}}{2\sqrt{z}} = \frac{1}{\sqrt{2\pi z}} e^{-z/2}, \quad z \geq 0.$$

■ **Remark 5.2 (Coordinate Transformation)** In most coordinate transformations it is g^{-1} that is given — that is, an expression for x as a function of z , — rather than g . ■

■ **Example 5.8 (Box–Muller)** Let X and Y be two independent standard normal random variables. (X, Y) is a random point in the plane. Let (R, Θ) be the corresponding polar coordinates. The joint pdf $f_{R,\Theta}$ of R and Θ is given by

$$f_{R,\Theta}(r, \theta) = \frac{1}{2\pi} e^{-r^2/2} r, \text{ for } r \geq 0 \text{ and } \theta \in [0, 2\pi).$$

Namely, specifying x and y in terms of r and θ gives

$$x = r \cos \theta \quad \text{and} \quad y = r \sin \theta. \quad (5.6)$$

The Jacobian of this coordinate transformation is

$$\det \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix} = \begin{vmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{vmatrix} = r.$$

The result now follows from the transformation rule (5.5), noting that the joint pdf of X and Y is $f_{X,Y}(x, y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2}$. It is not difficult to verify that R and Θ are independent, that $\Theta \sim \mathcal{U}[0, 2\pi)$ and that $\mathbb{P}(R > r) = e^{-r^2/2}$. This means that R has the same distribution as \sqrt{V} , with $V \sim \text{Exp}(1/2)$. Namely, $\mathbb{P}(\sqrt{V} > v) = \mathbb{P}(V > v^2) = e^{-v^2/2}$. Both Θ and R are easy to generate, and are transformed via (5.6) to independent standard normal random variables. ■

5.2 Jointly Normal Random Variables

In this section we have a closer look at normally distributed random variables and their properties. Also, we will introduce normally distributed random *vectors*.

It is helpful to view normally distributed random variables as simple transformations of standard normal random variables. For example, let $X \sim \mathcal{N}(0, 1)$. Then, X has density f_X given by

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

Now consider the transformation

$$Z = \mu + \sigma X.$$

Then, by (5.1) Z has density

$$f_Z(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}.$$

In other words, $Z \sim \mathcal{N}(\mu, \sigma^2)$. We could also write this as follows, if $Z \sim \mathcal{N}(\mu, \sigma^2)$, then $(Z-\mu)/\sigma \sim \mathcal{N}(0, 1)$. This **standardisation** procedure was already mentioned in Section 2.6.3.

Let's generalise this to n dimensions. Let X_1, \dots, X_n be independent and standard normal random variables. The joint pdf of $\mathbf{X} = (X_1, \dots, X_n)^\top$ is given by

$$f_{\mathbf{X}}(\mathbf{x}) = (2\pi)^{-n/2} e^{-\frac{1}{2} \mathbf{x}^\top \mathbf{x}}, \quad \mathbf{x} \in \mathbb{R}^n. \quad (5.7)$$

Consider the transformation

$$\mathbf{Z} = \boldsymbol{\mu} + \mathbf{B} \mathbf{X}, \quad (5.8)$$

for some $(m \times n)$ matrix \mathbf{B} . Note that by Theorem 5.1 \mathbf{Z} has expectation vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^\top$. Any random vector of the form (5.8) is said to have a **jointly normal** (or multi-variate normal) distribution. We write $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Suppose \mathbf{B} is an *invertible* $(n \times n)$ -matrix. Then, by (5.4) the density of $\mathbf{Y} = \mathbf{Z} - \boldsymbol{\mu}$ is given by

$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{|\mathbf{B}| \sqrt{(2\pi)^n}} e^{-\frac{1}{2} (\mathbf{B}^{-1}\mathbf{y})^\top \mathbf{B}^{-1}\mathbf{y}} = \frac{1}{|\mathbf{B}| \sqrt{(2\pi)^n}} e^{-\frac{1}{2} \mathbf{y}^\top (\mathbf{B}^{-1})^\top \mathbf{B}^{-1} \mathbf{y}}.$$

We have $|\mathbf{B}| = \sqrt{|\boldsymbol{\Sigma}|}$ and $(\mathbf{B}^{-1})^\top \mathbf{B}^{-1} = (\mathbf{B}^\top)^{-1} \mathbf{B}^{-1} = (\mathbf{B}\mathbf{B}^\top)^{-1} = \boldsymbol{\Sigma}^{-1}$, so that

$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2} \mathbf{y}^\top \boldsymbol{\Sigma}^{-1} \mathbf{y}}.$$

Because \mathbf{Z} is obtained from \mathbf{Y} by simply adding a constant vector $\boldsymbol{\mu}$, we have $f_{\mathbf{Z}}(\mathbf{z}) = f_{\mathbf{Y}}(\mathbf{z} - \boldsymbol{\mu})$, and therefore

$$f_{\mathbf{Z}}(\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{z} - \boldsymbol{\mu})}, \quad \mathbf{z} \in \mathbb{R}^n. \quad (5.9)$$

Note that this formula is very similar to the 1-dimensional case.

■ **Example 5.9 (Bivariate Normal Distribution)** Consider the 2-dimensional case with $\mu = (\mu_1, \mu_2)^\top$, and

$$B = \begin{pmatrix} \sigma_1 & 0 \\ \sigma_2 \varrho & \sigma_2 \sqrt{1 - \varrho^2} \end{pmatrix}. \quad (5.10)$$

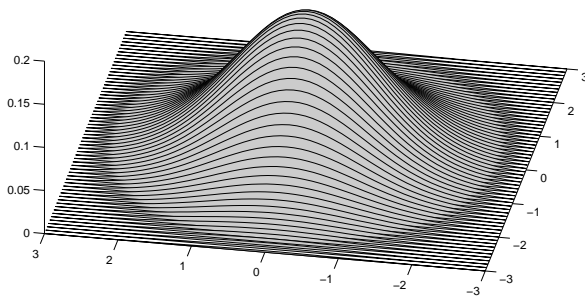
The covariance matrix is now

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \varrho \sigma_1 \sigma_2 \\ \varrho \sigma_1 \sigma_2 & \sigma_2^2 \end{pmatrix}. \quad (5.11)$$

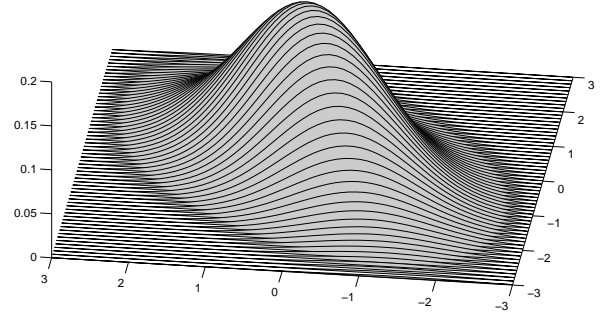
Therefore, the density is

$$f_{\mathbf{Z}}(\mathbf{z}) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\varrho^2}} \exp \left\{ -\frac{1}{2(1-\varrho^2)} \left(\frac{(z_1 - \mu_1)^2}{\sigma_1^2} - 2\varrho \frac{(z_1 - \mu_1)(z_2 - \mu_2)}{\sigma_1\sigma_2} + \frac{(z_2 - \mu_2)^2}{\sigma_2^2} \right) \right\}. \quad (5.12)$$

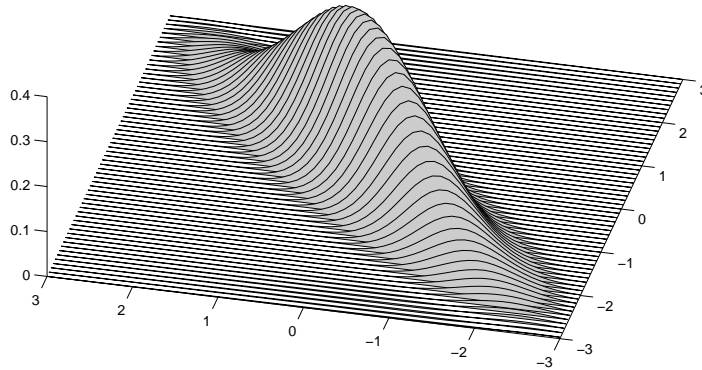
Here are some pictures of the density, for $\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$, and for various ϱ .



$\varrho = 0$



$\varrho = 0.5$



$\varrho = 0.9$

We say that $(Z_1, Z_2)^\top$ has a **bivariate normal** distribution. Note that in this example $\mathbb{E}Z_i = \mu_i, i = 1, 2$. Moreover, since we have chosen B such that the covariance matrix has the form (5.11), we have $\text{Var}(Z_i) = \sigma_i^2, i = 1, 2$, and $\varrho(Z_1, Z_2) = \varrho$. We will see shortly that Z_1 and Z_2 both have normal distributions. ■

Compare the following with property 8 of Table 4.1.

Theorem 5.3

If Z_1 and Z_2 have a jointly normal distribution then

$$\text{Cov}(Z_1, Z_2) = 0 \implies Z_1 \text{ and } Z_2 \text{ are independent.}$$

Proof: If $\text{Cov}(Z_1, Z_2) = 0$, then B in (5.10) is a diagonal matrix. Thus, trivially $Z_1 = \sigma_1 X_1$ and $Z_2 = \sigma_2 X_2$ are independent. \square

One of the most (if not the most) important properties of the normal distribution is that affine combinations (that is, linear combinations plus a constant) of independent normal random variables are normally distributed. Here is a more precise formulation.

Theorem 5.4: Affine Combinations of Normals

If $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, independently, for $i = 1, 2, \dots, n$, then

$$Y = a + \sum_{i=1}^n b_i X_i \sim \mathcal{N}\left(a + \sum_{i=1}^n b_i \mu_i, \sum_{i=1}^n b_i^2 \sigma_i^2\right). \quad (5.13)$$

Proof: The easiest way to prove this is by using moment generating functions. First, recall that the MGF of a $\mathcal{N}(\mu, \sigma^2)$ -distributed random variable X is given by

$$M_X(s) = e^{\mu s + \frac{1}{2} \sigma^2 s^2}.$$

Let M_Y be the moment generating function of Y . Since X_1, \dots, X_n are independent, we have

$$\begin{aligned} M_Y(s) &= \mathbb{E} \exp\{as + \sum_{i=1}^n b_i X_i s\} \\ &= e^{as} \prod_{i=1}^n M_{X_i}(b_i s) \\ &= e^{as} \prod_{i=1}^n \exp\{\mu_i(b_i s) + \frac{1}{2} \sigma_i^2 (b_i s)^2\} \\ &= \exp\{sa + s \sum_{i=1}^n b_i \mu_i + \frac{1}{2} \sum_{i=1}^n b_i^2 \sigma_i^2 s^2\}, \end{aligned}$$

which is the MGF of a normal distribution of the form (5.13). \square

■ **Remark 5.3 (Distribution of Normal Affine Combinations)** Note that from Theorems 4.3 and 4.6 we had already established the expectation and variance of Y in (5.13). But we have now found that the *distribution* is normal. ■

■ **Example 5.10 (Ball Bearings)** A machine produces ball bearings with a $\mathcal{N}(1, 0.01)$ diameter (cm). The balls are placed on a sieve with a $\mathcal{N}(1.1, 0.04)$ diameter. The diameter of the balls and the sieve are assumed to be independent of each other.

Question: What is the probability that a ball will fall through?

Answer: Let $X \sim \mathcal{N}(1, 0.01)$ and $Y \sim \mathcal{N}(1.1, 0.04)$. We need to calculate $\mathbb{P}(Y > X) = \mathbb{P}(Y - X > 0)$. But, $Z := Y - X \sim \mathcal{N}(0.1, 0.05)$. Hence

$$\mathbb{P}(Z > 0) = \mathbb{P}\left(\frac{Z - 0.1}{\sqrt{0.05}} > \frac{-0.1}{\sqrt{0.05}}\right) = \Phi(0.447) \approx 0.67,$$

where Φ is the cdf of the $\mathcal{N}(0, 1)$ -distribution. ■

5.3 Normal Linear Models

One of the simplest and most important models for statistical learning is the **normal linear model**, where a random **response variable** Y is explained via an **explanatory variable** (vector) \mathbf{x} via a linear relationship plus a normal error term:

Definition 5.1: Normal Linear Model

In a *normal linear model* the response Y depends on a p -dimensional explanatory variable $\mathbf{x} = [x_1, \dots, x_p]^\top$, via the linear relationship

$$Y = \mathbf{x}^\top \boldsymbol{\beta} + \varepsilon, \quad (5.14)$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

Note that (5.14) is a model for a single pair (\mathbf{x}, Y) . The model for the training set $\{(\mathbf{x}_i, Y_i)\}$ is simply that each Y_i satisfies (5.14) (with $\mathbf{x} = \mathbf{x}_i$) and that the $\{Y_i\}$ are independent. Gathering all responses in the vector $\mathbf{Y} = [Y_1, \dots, Y_n]^\top$, we can write

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (5.15)$$

where $\boldsymbol{\varepsilon} = [\varepsilon_1, \dots, \varepsilon_n]^\top$ is a vector of iid copies of ε , so $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$, and \mathbf{X} is the so-called *model matrix*, with rows $\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top$.

Consequently, \mathbf{Y} can be written as $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \sigma \mathbf{Z}$, where $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, so that $\mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n)$. It follows from (5.9) that its joint density is given by

$$f_Y(\mathbf{y}) = (2\pi\sigma^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2}. \quad (5.16)$$

The main question is now how to find a good guess (estimate) of β based on the observed data. In the least-squares method an estimate is obtained by minimizing the square distance $\|y - X\beta\|^2$ between the vectors of observed responses, y , and the predicted ones, $X\beta$.

We will see in STAT3001 that the optimal solution to this optimization problem satisfies that matrix equation

$$X^T X \beta = X^T y.$$

These are called the **normal equations**.

5.3.1 Linear Regression

The most basic linear model involves a linear relationship between the response and a single explanatory variable. In particular, we have measurements $(x_1, y_1), \dots, (x_n, y_n)$ that lie approximately on a straight line, as in Figure 5.4.

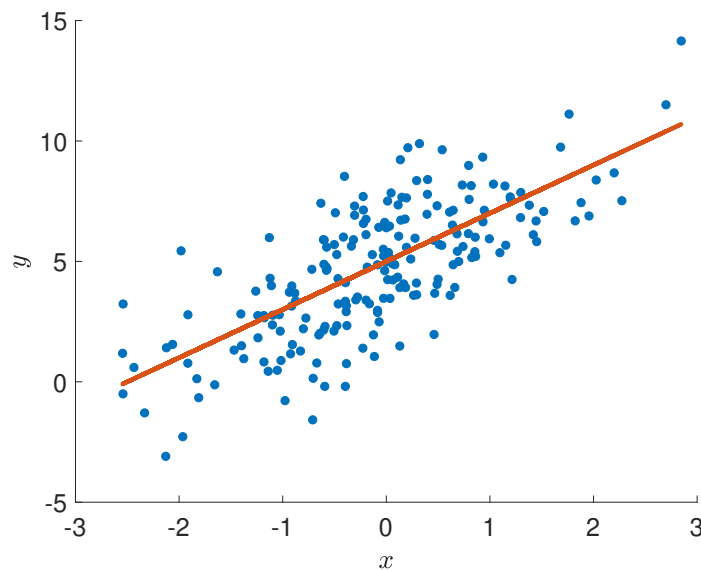


Figure 5.4: Data from a simple linear regression model.

A simple model for these data is that the $\{x_i\}$ are fixed and variables $\{Y_i\}$ are random such that

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i = 1, \dots, n, \quad (5.17)$$

for certain *unknown* parameters β_0 and β_1 . The $\{\varepsilon_i\}$ are assumed to be independent with expectation 0 and unknown variance σ^2 . The unknown line

$$y = \underbrace{\beta_0 + \beta_1 x}_{g(x|\beta)} \quad (5.18)$$

is called the *regression line*. Thus, we view the responses as random variables that would lie exactly on the regression line, were it not for some “disturbance” or “error” term represented by the $\{\varepsilon_i\}$. The extent of the disturbance is modeled by the parameter σ^2 . The model in (5.17) is called *simple linear regression*. This model can easily be extended to incorporate more than one explanatory variable, as follows.

Definition 5.2: Multiple Linear Regression Model

In a *multiple linear regression model* the response Y depends on a d -dimensional explanatory vector $\mathbf{x} = [x_1, \dots, x_d]^\top$, via the linear relationship

$$Y = \beta_0 + \beta_1 x_1 + \dots + \beta_d x_d + \varepsilon, \quad (5.19)$$

where $\mathbb{E} \varepsilon = 0$ and $\text{Var} \varepsilon = \sigma^2$.

Thus, the data lie approximately on a d -dimensional affine hyperplane

$$y = \underbrace{\beta_0 + \beta_1 x_1 + \dots + \beta_d x_d}_{g(\mathbf{x}|\boldsymbol{\beta})},$$

where we define $\boldsymbol{\beta} = [\beta_0, \beta_1, \dots, \beta_d]^\top$. The function $g(\mathbf{x}|\boldsymbol{\beta})$ is linear in $\boldsymbol{\beta}$, but not linear in the feature vector \mathbf{x} , due to the constant β_0 . However, *augmenting* the feature space with the constant 1, the mapping $[1, \mathbf{x}^\top]^\top \mapsto g(\mathbf{x}|\boldsymbol{\beta}) := [1, \mathbf{x}^\top] \boldsymbol{\beta}$ becomes linear in the feature space and so (5.19) becomes a *linear model*. Most software packages for regression include 1 as a feature by default.

Note that in (5.19) we only specified the model for a single pair (\mathbf{x}, Y) . The model for n independent data points $\{(\mathbf{x}_1, Y_1), \dots, (\mathbf{x}_n, Y_n)\}$ is simply that each Y_i satisfies (5.19) (with $\mathbf{x} = \mathbf{x}_i$) and that the $\{Y_i\}$ are independent. Setting $\mathbf{Y} = [Y_1, \dots, Y_n]^\top$, we can write the multiple linear regression model for the training data compactly as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (5.20)$$

where $\boldsymbol{\varepsilon} = [\varepsilon_1, \dots, \varepsilon_n]^\top$ is a vector of iid copies of ε and \mathbf{X} is the *model matrix* given by

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{x}_1^\top \\ 1 & \mathbf{x}_2^\top \\ \vdots & \vdots \\ 1 & \mathbf{x}_n^\top \end{bmatrix}.$$

■ **Example 5.11 (Multiple Linear Regression Model)** Figure 5.5 depicts a realization of the multiple linear regression model

$$Y_i = x_{i1} + x_{i2} + \varepsilon_i, \quad i = 1, \dots, 100,$$

where $\varepsilon_1, \dots, \varepsilon_{100} \sim_{\text{iid}} \mathcal{N}(0, 1/16)$. The fixed feature vectors (vectors of explanatory variables) $\mathbf{x}_i = [x_{i1}, x_{i2}]^\top$, $i = 1, \dots, 100$ lie in the unit square.

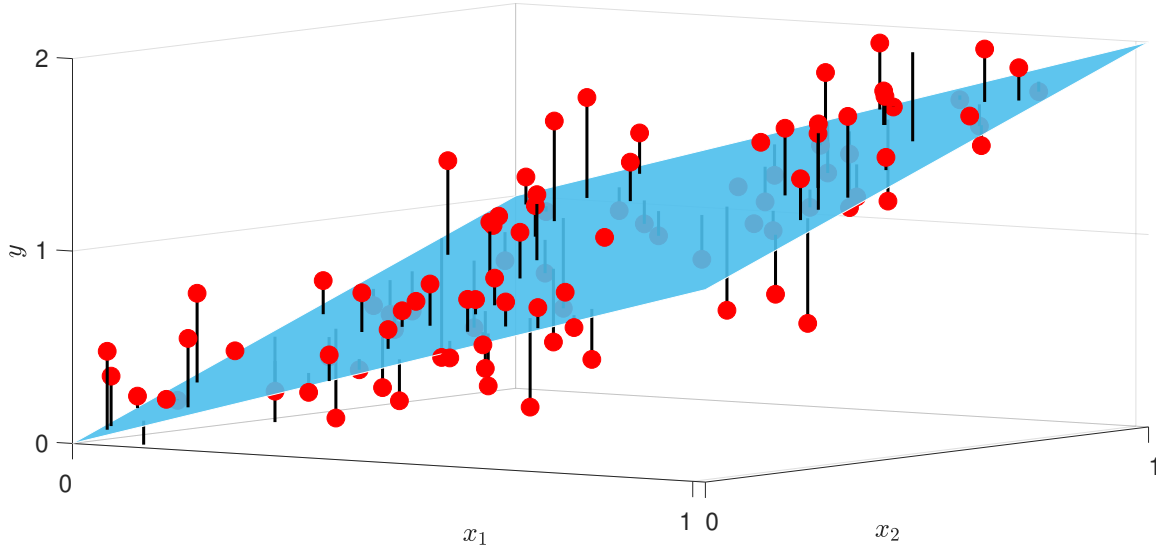


Figure 5.5: Data from a multiple linear regression model.

5.4 Limit Theorems

In this section we briefly discuss two of the main results in probability: the Law of Large Numbers (LLN) and the Central Limit Theorem (CLT). Both are about sums of independent random variables.

Let X_1, X_2, \dots be independent and identically distributed random variables. For each n let

$$S_n = X_1 + \dots + X_n.$$

Suppose $\mathbb{E}X_i = \mu$ and $\text{Var}(X_i) = \sigma^2$. We assume that both μ and σ^2 are finite. By the rules for expectation and variance we know that

$$\mathbb{E}S_n = n \mathbb{E}X_1 = n\mu$$

and

$$\text{Var}(S_n) = n \text{Var}(X_1) = n\sigma^2.$$

Moreover, if the X_i have moment generating function M , then the MGF of S_n is simply given by

$$\mathbb{E}e^{s(X_1 + \dots + X_n)} = \mathbb{E}e^{sX_1} \dots \mathbb{E}e^{sX_n} = [M(s)]^n.$$

The law of large numbers roughly states that S_n/n is close to μ , for large n . Here is a more precise statement.

Theorem 5.5: (Weak) Law of Large Numbers

If X_1, \dots, X_n are independent and identically distributed with expectation μ , then for all $\varepsilon > 0$

$$\lim_{n \rightarrow \infty} \mathbb{P} \left(\left| \frac{S_n}{n} - \mu \right| \geq \varepsilon \right) = 0 .$$

Proof: First, for any $z > 0$, and any positive random variable Z we have

$$\begin{aligned} \mathbb{E}Z &= \int_0^z t f(t) dt + \int_z^\infty t f(t) dt \geq \int_z^\infty t f(t) dt \\ &\geq \int_z^\infty z f(t) dt = z \mathbb{P}(Z \geq z) , \end{aligned}$$

from which follows immediately the following **Markov inequality**: if $Z \geq 0$, then for all $z > 0$,

$$\mathbb{P}(Z \geq z) \leq \frac{\mathbb{E}Z}{z} . \quad (5.21)$$

Now take $Z = (S_n/n - \mu)^2$ and $z = \varepsilon^2$. Then,

$$\mathbb{P}((S_n/n - \mu)^2 \geq \varepsilon^2) \leq \frac{\mathbb{E}(S_n/n - \mu)^2}{\varepsilon^2}$$

The left-hand side of the above equation can also be written as $\mathbb{P}(|S_n/n - \mu| \geq \varepsilon)$, and the right-hand side is equal to the variance of S_n/n , which is σ^2/n . Combining gives,

$$\mathbb{P}(|S_n/n - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{n\varepsilon^2} ,$$

for any $\varepsilon > 0$. As $n \rightarrow \infty$ the quotient $\frac{\sigma^2}{n\varepsilon^2}$ tends to zero, and therefore $\mathbb{P}(|S_n/n - \mu| \geq \varepsilon)$ goes to zero as well, which had to be shown. \square

There is also a **strong law of large numbers**, which implies the weak law, but is more difficult to prove. It states the following:

$$\mathbb{P} \left(\lim_{n \rightarrow \infty} \frac{S_n}{n} = \mu \right) = 1 ,$$

as $n \rightarrow \infty$, meaning that the set of outcomes ω such that $\frac{S_n(\omega)}{n} \rightarrow \mu$, has probability one. In other words if we were to run a computer simulation, then all paths that we would simulate would converge to μ .

The Central Limit Theorem says something about the approximate *distribution* of S_n (or S_n/n). Roughly it says this:

*The sum of a large number of iid random variables has approximately a **normal** distribution*

Here is a more precise statement.

Theorem 5.6: Central Limit Theorem

If X_1, \dots, X_n are independent and identically distributed with expectation μ and variance $\sigma^2 < \infty$, then for all $x \in \mathbb{R}$,

$$\lim_{n \rightarrow \infty} \mathbb{P} \left(\frac{S_n - n\mu}{\sigma \sqrt{n}} \leq x \right) = \Phi(x),$$

where Φ is the cdf of the standard normal distribution.

In other words, S_n has approximately a normal distribution with expectation $n\mu$ and variance $n\sigma^2$.

Proof: (Sketch) Without loss of generality assume $\mu = 0$ and $\sigma = 1$. This amounts to replacing X_n by $(X_n - \mu)/\sigma$. A Taylor-expansion of the MGF around $s = 0$ yields

$$M(s) = \mathbb{E} e^{sX_1} = 1 + s\mathbb{E}X_1 + \frac{1}{2}s^2 \mathbb{E}X_1^2 + o(s^2) = 1 + \frac{1}{2}s^2 + o(s^2),$$

where $o(\cdot)$ is a function for which $\lim_{x \downarrow 0} o(x)/x = 0$. Because the X_1, X_2, \dots are i.i.d., it follows that the MGF of S_n/\sqrt{n} satisfies

$$\begin{aligned} \mathbb{E} \exp \left(s \frac{S_n}{\sqrt{n}} \right) &= \mathbb{E} \exp \left(\frac{s}{\sqrt{n}} (X_1 + \dots + X_n) \right) = \prod_{i=1}^n \mathbb{E} \exp \left(\frac{s}{\sqrt{n}} X_i \right) \\ &= M^n \left(\frac{s}{\sqrt{n}} \right) = \left[1 + \frac{s^2}{2n} + o\left(\frac{s^2}{n}\right) \right]^n. \end{aligned}$$

For $n \rightarrow \infty$ this converges to $e^{s^2/2}$, which is the MGF of the standard normal distribution. Thus, it is plausible that the cdf of S_n/\sqrt{n} converges to Φ . To make this argument rigorous, one needs to show that convergence of the moment generating function implies convergence of the cdf. Moreover, since for some distributions the MGF does not exist in a neighbourhood of 0, one needs to replace the MGF in the argument above with a more flexible transform, namely the Fourier transform, also called characteristic function: $r \mapsto \mathbb{E} e^{irX}$, $r \in \mathbb{R}$. \square

To see the CLT in action consider Figure 5.6. The first picture shows the pdf's of S_1, \dots, S_4 for the case where the X_i have a $\mathcal{U}[0, 1]$ distribution. The second show the same, but now for an $\text{Exp}(1)$ distribution. We clearly see convergence to a bell shaped curve.

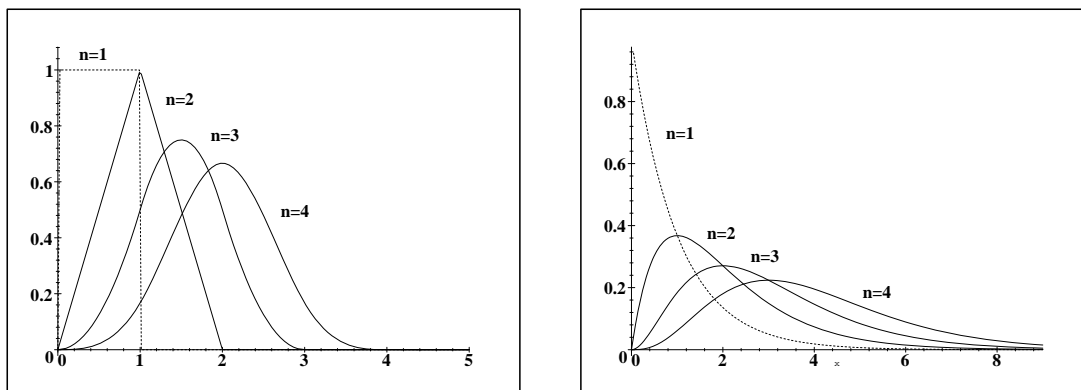


Figure 5.6: Illustration of the CLT for the uniform and exponential distribution

The CLT is not restricted to continuous distributions. For example, Figure 5.7 shows the cdf of S_{30} in the case where the X_i have a Bernoulli distribution with success probability $1/2$. Note that $S_{30} \sim \text{Bin}(30, 1/2)$, see Example 4.3.

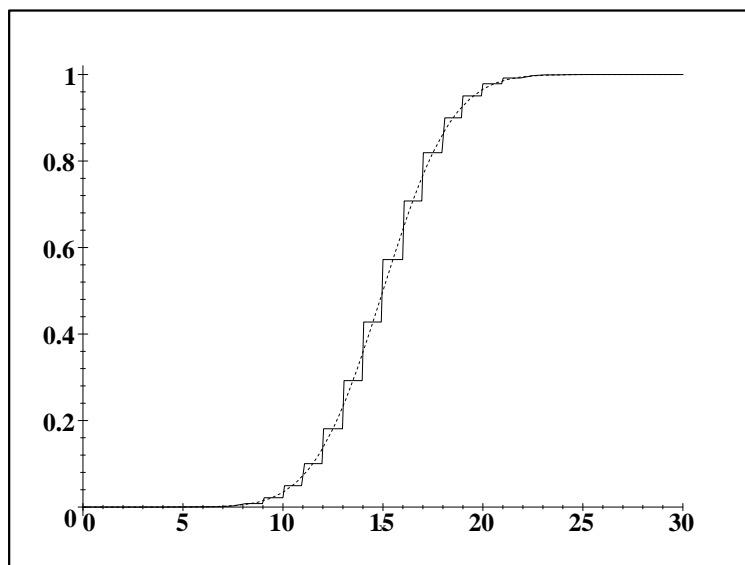


Figure 5.7: The cdf of a Bin(30, 1/2)-distribution and its normal approximation.

In general we have:

Theorem 5.7

Let $X \sim \text{Bin}(n, p)$. For large n we have

$$\mathbb{P}(X \leq k) \approx \mathbb{P}(Y \leq k),$$

where $Y \sim \mathcal{N}(np, np(1-p))$. As a rule of thumb, the approximation is accurate if both np and $n(1-p)$ are larger than 5.

MARKOV CHAINS

6.1 Introduction

Markov chains are very important tools for modelling random phenomena. In this chapter we will have a brief glimpse at these processes, to whet our appetite for more advanced treatments such as STAT3004. Let us start with a motivating example.

■ **Example 6.1 (Stepping Stones)** Suppose there are 5 rocks in a pond on which you can step, see Figure 6.1. The arrows indicate from which stone you can step to another. Suppose that from your current position you choose your next stone with equal probability amongst the possible stone. For example, if, at after a number of steps you are on stone 5, you can step onto stone 4 or stone 6, with probability $1/2$ each. Suppose you start from position (stone) 1. Where would you be after 10 steps? What would the probability be that you end up back in position 1 after 10 steps?

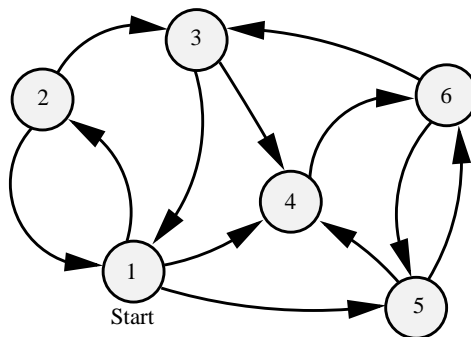


Figure 6.1: Stepping Stones

To analyse this problem, we introduce the random variables X_0, X_1, X_2, \dots that indicate the current position after n steps. By definition $X_0 = 1$, because we start at position 1. What can we say about the probability distribution of the random variables X_1, X_2, \dots ? Clearly they are not independent. For example, knowing $X_9 = 5$ gives information about X_{10} (namely X_{10} is either 4 or 6 with probability $1/2$). On the other hand, to predict where we will be at time 10, we only need to know our position at time 9. Any additional information, e.g., where we were at times $1, 2, \dots, 8$ is *irrelevant*. Thus, for this particular example we can say that “the future is conditionally independent of the pasts, given the present”. This is the defining property of Markov chains. ■

Definition 6.1: Markov Chain

A collection of random variables X_0, X_1, X_2, \dots is called a **Markov chain** with **state space** \mathcal{E} if

$$\mathbb{P}(X_{n+1} = x_{n+1} | X_0 = x_0, \dots, X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n) \quad (6.1)$$

for all $x_0, \dots, x_{n+1}, \in \mathcal{E}$ and $n \in \mathbb{N}$.

We restrict ourselves to Markov chains for which the conditional probability

$$\mathbb{P}(X_{n+1} = j | X_n = i), \quad i, j \in \mathcal{E} \quad (6.2)$$

is independent of the time n and for which the state space \mathcal{E} is discrete (countable), like in Example 6.1. The probabilities in (6.2) are called the *(one-step) transition probabilities* of $X := \{X_n, n \geq 0\}$. The distribution of X_0 is called the *initial distribution* of the Markov chain. The one-step transition probabilities and the initial distribution completely specify the distribution of X . Namely, we have by the product rule (1.6) and the Markov property (6.1)

$$\begin{aligned} \mathbb{P}(X_0 = x_0, \dots, X_n = x_n) \\ &= \mathbb{P}(X_0 = x_0) \mathbb{P}(X_1 = x_1 | X_0 = x_0) \cdots \mathbb{P}(X_n = x_n | X_0 = x_0, \dots, X_{n-1} = x_{n-1}) \\ &= \mathbb{P}(X_0 = x_0) \mathbb{P}(X_1 = x_1 | X_0 = x_0) \cdots \mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}). \end{aligned}$$

■ **Example 6.2 (Stepping Stones (Continued))** The probability of simulating the “path” $\{X_0 = 1, X_1 = 2, X_2 = 3, X_3 = 4, X_4 = 6, X_5 = 5\}$ is: $1 \times 1/3 \times 1/2 \times 1/2 \times 1 \times 1/2 = 1/24$. ■

Since \mathcal{E} is countable, we can arrange the one-step transition probabilities in an array. This array is called the *(one-step) transition matrix* of X . We denote it usually by P . For example, when $\mathcal{E} = \{0, 1, 2, \dots\}$ the transition matrix P has the form

$$\begin{array}{c} \text{to} \\ \text{from} \end{array} \begin{pmatrix} p_{00} & p_{01} & p_{02} & \cdots \\ p_{10} & p_{11} & p_{12} & \cdots \\ p_{20} & p_{21} & p_{22} & \cdots \\ \vdots & \vdots & & \end{pmatrix}.$$

Note that the elements in every row are positive and sum up to unity.

■ **Example 6.3 (Stepping Stones (Continued))** For the stepping stone example the transition matrix is

$$P = \begin{pmatrix} 0 & 1/3 & 0 & 1/3 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \end{pmatrix}.$$

Another convenient way to describe a Markov chain X is through its *transition graph*. States are indicated by the nodes of the graph, and a strictly positive (> 0) transition probability p_{ij} from state i to j is indicated by an arrow from i to j with weight p_{ij} . For the stepping stone example, the transition graph would give the graph of Figure 6.1 with the addition of weights on the arrows (arcs). For example the arrow from 1 to 2, would have weight $1/3$.

■ **Example 6.4 (Random walk on the integers)** Let p be a number between 0 and 1. A random walk on \mathbb{Z} , with parameter p , is the Markov chain X with state space \mathbb{Z} and transition matrix P , given by

$$p_{i,i+1} = p, \quad p_{i,i-1} = q = 1 - p, \quad \text{for all } i \in \mathbb{Z}.$$

Let X start at 0, thus $\mathbb{P}(X_0 = 0) = 1$. The corresponding transition graph is given in Figure 6.2. Starting at 0, we take subsequent steps to the right with probability p and to the left with probability q .

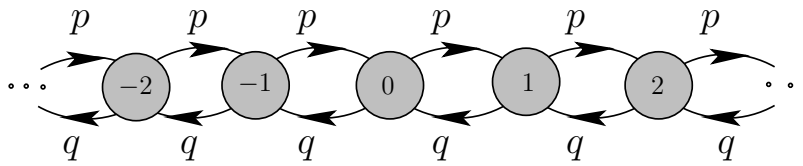


Figure 6.2: Transition graph for the random walk.

We shall show next now how to calculate the probability that, starting from state i at some (discrete) time n , we arrive in j at (discrete) time $n + m$, that is, the probability $\mathbb{P}(X_{n+m} = j \mid X_n = i)$. For clarity let us assume that $\mathcal{E} = \{1, 2, \dots, r\}$, for some fixed r , so that P is an $r \times r$ matrix. For $n = 0, 1, 2, \dots$, define the *row* vector

$$\pi^{(n)} = (\mathbb{P}(X_n = 1), \dots, \mathbb{P}(X_n = r)).$$

We call $\pi^{(n)}$ the **distribution vector**, or simply *distribution*, of X at time n and $\pi^{(0)}$ the *initial distribution* of X . The following result shows that the n -step probabilities can be simply found by *matrix multiplication*.

Theorem 6.1

The distribution of X at time n is given by

$$\pi^{(n)} = \pi^{(0)} P^n, \quad (6.3)$$

for all $n = 0, 1, \dots$ (Here P^0 denotes the identity matrix.)

Proof: The proof is by induction. The equality (6.3) holds for $n = 0$, by definition. Suppose it is true for some $n = 0, 1, \dots$. We have

$$\mathbb{P}(X_{n+1} = k) = \sum_{i=1}^r \mathbb{P}(X_{n+1} = k \mid X_n = i) \mathbb{P}(X_n = i).$$

But (6.3) is assumed to be true for n , so $\mathbb{P}(X_n = i)$ is the i -th element of $\pi^{(0)} P^n$. Moreover, $\mathbb{P}(X_{n+1} = k \mid X_n = i)$ is the (i, k) -th element of P . Therefore, for every k

$$\sum_{i=1}^r \mathbb{P}(X_{n+1} = k \mid X_n = i) \mathbb{P}(X_n = i) = \sum_{i=1}^r P(i, k) (\pi^{(0)} P^n)(i),$$

which is just the k -th element of $\pi^{(0)} P^{n+1}$. This completes the induction step and thus the theorem is proved. \square

By taking $\pi^{(0)}$ as the i -th unit vector, e_i , the n -step transition probabilities can be found as $\mathbb{P}(X_n = j \mid X_0 = i) = (e_i P^n)(j) = P^n(i, j)$, which is the (i, j) -th element of matrix P^n . Thus to find the n -step transition probabilities we just have to compute the n -th power of P .

■ **Example 6.5 (Stepping Stones (Continued))** Below I have listed some powers of the matrix P . We see for example that the probability of going from stone 1 to stone 6 in two steps is $1/2$, and the probability of returning to 1 in 10 steps is 0.1049 (approximately). Observe that as n grows, $P^n \rightarrow P^\infty$ for some matrix P^∞ that has equal rows. The interpretation of $P^\infty(i, j)$ is the probability of being in j “far away in the future”, starting from i . That this probability does not depend on i makes sense: if we keep stepping on the stones for a long time, it does not matter where we started from.

```
np.set_printoptions(precision=4)
for n in [1, 2, 5, 10, 20, 100]:
    print(np.linalg.matrix_power(P, n), '\n')
```

```
[ [0.    0.3333 0.    0.3333 0.3333 0.    ]
  [0.5    0.    0.5    0.    0.    0.    ]
  [0.5    0.    0.    0.5    0.    0.    ]
  [0.    0.    0.    0.    0.    1.    ]
  [0.    0.    0.    0.5    0.    0.5   ]
  [0.    0.    0.5    0.    0.5    0.    ] ]]
```

```
[ [0.1667 0.    0.1667 0.1667 0.    0.5   ]
  [0.25   0.1667 0.    0.4167 0.1667 0.    ]
  [0.    0.1667 0.    0.1667 0.1667 0.5   ]
  [0.    0.    0.5    0.    0.5    0.    ]
  [0.    0.    0.25   0.    0.25   0.5   ]
  [0.25   0.    0.    0.5    0.    0.25  ] ]]
```

```
[ [0.0694 0.0509 0.1597 0.162  0.1968 0.3611]
  [0.1597 0.0278 0.1181 0.3125 0.1319 0.25  ]
  [0.0764 0.0556 0.1181 0.1944 0.1597 0.3958]
  [0.1042 0.    0.2917 0.1667 0.25   0.1875]
  [0.0521 0.0417 0.2083 0.125  0.2292 0.3438]
  [0.1458 0.0347 0.0938 0.3056 0.1285 0.2917] ]]
```

```
[ [0.1049 0.0346 0.1581 0.2202 0.1779 0.3044]
  [0.1004 0.0296 0.1848 0.2025 0.1944 0.2882]
  [0.1095 0.0324 0.1598 0.226  0.1773 0.295  ]
  [0.0872 0.0384 0.1693 0.1899 0.1901 0.3249]
  [0.1013 0.0369 0.1543 0.2165 0.177  0.3142]
  [0.1039 0.0291 0.1817 0.2088 0.1916 0.285  ] ]]
```

```
[ [0.1006 0.0336 0.1683 0.2097 0.1851 0.3027]
  [0.101  0.0337 0.1675 0.2106 0.1845 0.3026]
  [0.1006 0.0337 0.1681 0.2097 0.185  0.303  ]
  [0.1011 0.0334 0.1685 0.2103 0.185  0.3016]
  [0.1007 0.0336 0.1685 0.2096 0.1852 0.3024]
  [0.1009 0.0337 0.1675 0.2105 0.1845 0.3028] ]]
```

```
[ [0.1008 0.0336 0.1681 0.2101 0.1849 0.3025]
  [0.1008 0.0336 0.1681 0.2101 0.1849 0.3025]
  [0.1008 0.0336 0.1681 0.2101 0.1849 0.3025]
  [0.1008 0.0336 0.1681 0.2101 0.1849 0.3025]
  [0.1008 0.0336 0.1681 0.2101 0.1849 0.3025]
  [0.1008 0.0336 0.1681 0.2101 0.1849 0.3025] ]]
```



6.2 Simulating Markov Chains

We now discuss how to simulate a Markov chain $X_0, X_1, X_2, \dots, X_n$. To generate a Markov chain with initial distribution $\pi^{(0)}$ and transition matrix P we first generate X_0 from $\pi^{(0)}$. Then, given $X_0 = x_0$, generate X_1 from the conditional distribution of X_1 given $X_0 = x_0$; in other words, generate X_1 from the x_0 -th row of P . Suppose $X_1 = x_1$. Then, generate X_2 from the x_1 -st row of P , etcetera. The algorithm for a general discrete-state Markov chain with one-step transition matrix P and initial distribution vector $\pi^{(0)}$ is as follows:

Algorithm 6.2.1: Generating a Markov Chain

1. Draw X_0 from the initial distribution $\pi^{(0)}$. Set $n = 0$.
 2. Draw X_{n+1} from the distribution corresponding to the X_n -th row of P .
 3. Set $n = n + 1$ and go to 2.
-

■ **Example 6.6 (Stepping stones (continued))** Here is how you can simulate the stepping stone Markov chain in Python:

```
import numpy as np
import matplotlib.pyplot as plt
n = 101
P = np.array([[0, 1/3, 0, 1/3, 1/3, 0],
              [0.5, 0, 0.5, 0, 0, 0],
              [0.5, 0, 0, 0.5, 0, 0],
              [0, 0, 0, 0, 0, 1],
              [0, 0, 0, 0.5, 0, 0.5],
              [0, 0, 0.5, 0, 0.5, 0]])
x = np.array(np.ones(n, dtype=int))
x[0] = 0
for t in range(0, n-1):
    x[t+1] = np.min(np.where(np.cumsum(
        P[x[t], :]) > np.random.rand()))
x = x + 1 #add 1 to all elements of the vector x
plt.plot(np.array(range(0, n)), x, 'o')
plt.plot(np.array(range(0, n)), x, '--')
plt.show()
```

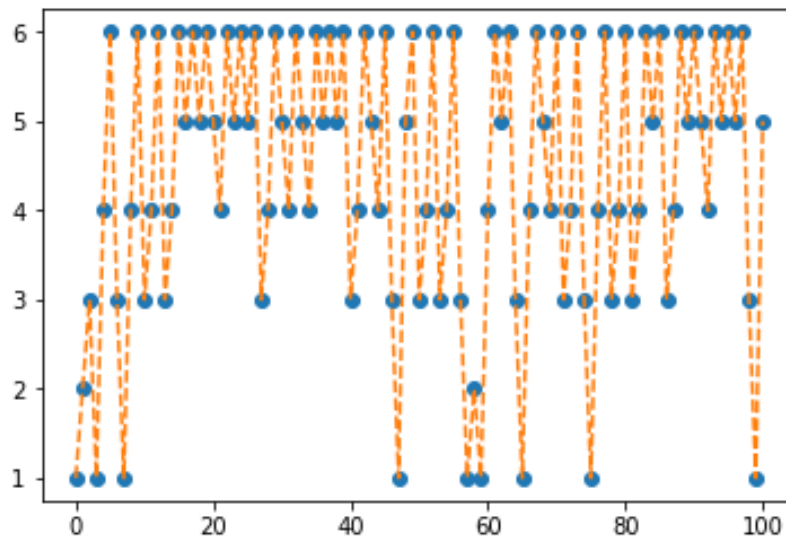
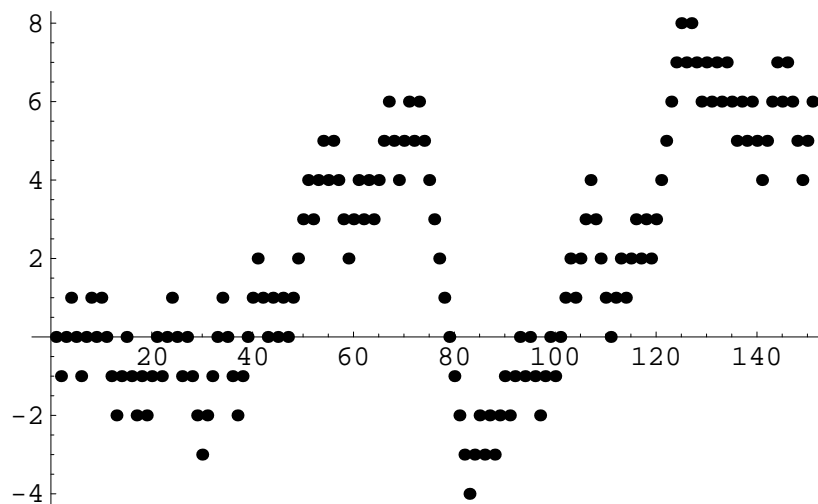


Figure 6.3: A path of the stepping stone Markov chain.

■ **Example 6.7 (Random Walk on the Integers)** Consider the random walk on the integers in Example 6.4. Let $X_0 = 0$ (that is, we start at 0). Suppose the chain is at some time n in state i . Then, in step 2 of Algorithm 6.2.1 we simply need to draw from a 2-point distribution with mass p and q at $i + 1$ and $i - 1$ respectively. In other words, we draw $I_n \sim \text{Ber}(p)$ and set $X_{n+1} = X_n + 2I_n - 1$. Figure 6.4 gives a typical sample path for the case where $p = q = 1/2$.

Figure 6.4: Random walk on the integers, with $p = q = 1/2$.

6.3 Limiting Behaviour

In Example 6.5 we saw that the stepping stone Markov chain exhibits “limiting behaviour”, in the sense that the n -step transition matrix P^n converges to a matrix P^∞ as $n \rightarrow \infty$. Moreover, this P^∞ has all rows equal to some row vector π . The limiting behaviour of Markov chains as $n \rightarrow \infty$, i.e., after the process has been going on for a very long time, is of considerable interest and importance. The results are often much simpler and are quite useful in approximating real situations. It can be shown that for general Markov chains (satisfying some mild conditions)

$$\lim_{n \rightarrow \infty} P^n(i, j) = \pi_j, \quad (6.4)$$

for some number $0 \leq \pi_j \leq 1$. When the $\{\pi_j\}$ sum up to 1, they form the **limiting distribution** of the Markov chain. The following theorem gives a method for obtaining limiting distributions. Here we assume for simplicity that $\mathcal{E} = \{0, 1, 2, \dots\}$. The limiting distribution is identified with the row vector $\pi = (\pi_0, \pi_1, \dots)$.

Theorem 6.2

The limiting distribution π , if it exists, is uniquely determined by the solution of

$$\pi = \pi P, \quad (6.5)$$

with $\pi_j \geq 0$ and $\sum_j \pi_j = 1$. Conversely, if there exists a unique positive row vector π satisfying (6.5) and summing up to 1, then π is the limiting distribution of the Markov chain.

Proof: (Sketch). For the case where \mathcal{E} is finite, the result is simply a consequence of (6.3). Namely, with $\pi^{(0)}$ being the i -th unit vector, we have

$$P^{n+1}(i, j) = (\pi^{(0)} P^n P)(j) = \sum_{k \in \mathcal{E}} P^n(i, k) P(k, j).$$

Letting $n \rightarrow \infty$, we obtain (6.5) from (6.4), provided that we can change the order of the limit and the summation. To show uniqueness, suppose that another vector y , with $y_j \geq 0$ and $\sum_j y_j = 1$, satisfies $y = yP$. Then it is easy to show by induction that also $y = yP^n$, for every n . Hence letting $n \rightarrow \infty$, we obtain for every j

$$y_j = \sum_i y_i \pi_j = \pi_j,$$

since the y_j 's sum up to unity. We omit the proof of the converse statement. \square

■ **Example 6.8 (Stepping Stones (Continued))** To find the limiting distribution π we need to solve the matrix equation (6.5), or equivalently

$$\pi(P - I) = \mathbf{0},$$

which in turn is equivalent to

$$(P^T - I)\pi^T = \mathbf{0}^T,$$

where T denotes transposition. In other words, the column vector π^T lies in the null-space of the matrix $P^T - I$. In Python, we can find such a vector via:

```
from scipy.linalg import null_space
v = null_space(P.T - np.eye(6))
```

This vector does not sum up to 1, but this is easily fixed via:

```
v = v/sum(v)
```

Printing the transpose of v gives the row vector π :

```
[[0.1008 0.0336 0.1681 0.2101 0.1849 0.3025]]
```

■

■ **Example 6.9 (Random Walk on the Positive Integers)** This is a slightly different random walk than we had before in Example 6.4. Let X be a random walk on $\mathcal{E} = \{0, 1, 2, \dots\}$ with transition matrix

$$P = \begin{pmatrix} q & p & 0 & \dots \\ q & 0 & p & 0 & \dots \\ 0 & q & 0 & p & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix},$$

where $0 < p < 1$ and $q = 1 - p$. X_n could represent for example the number of customers that are waiting in a queue at time n .

Note that all states can be reached from each other. The equation $\pi = \pi P$ becomes

$$\pi_0 = q\pi_0 + q\pi_1,$$

$$\pi_1 = p\pi_0 + q\pi_2,$$

$$\pi_2 = p\pi_1 + q\pi_3,$$

$$\pi_3 = p\pi_2 + q\pi_4,$$

etc. We can solve this set of equation sequentially. If we let $r = p/q$, then we can express the π_1, π_2, \dots in terms of π_0 and r as

$$\pi_j = r^j \pi_0, \quad j = 0, 1, 2, \dots$$

If $p < q$, then $r < 1$ and $\sum_{j=0}^{\infty} \pi_j = \frac{1}{1-r} \pi_0$, and by choosing $\pi_0 = 1 - r$, we can make the sum $\sum \pi_j = 1$. Hence, for $r < 1$ we have found the limiting distribution $\pi = (1 - r)(1, r, r^2, r^3, \dots)$ for this Markov chain. On the other hand, when $p \geq q$, then $\sum \pi_j$ is either 0 or infinite, and hence the limiting distribution does not exist. ■

Noting that $\sum_j p_{ij} = 1$, we can rewrite (6.5) as the system of equations

$$\sum_j \pi_i p_{ij} = \sum_j \pi_j p_{ji}, \quad \text{for all } i \in \mathcal{E}. \quad (6.6)$$

These are called the *global balance equations*. We can interpret (6.5) as the statement that the “probability flux” out of i is balanced with the probability flux into i . An important generalisation, which follows directly from (6.6), states that the same balancing of probability fluxes holds for an arbitrary set \mathcal{A} . That is, for every set \mathcal{A} of states we have

$$\sum_{i \in \mathcal{A}} \sum_{j \notin \mathcal{A}} \pi_i p_{ij} = \sum_{i \in \mathcal{A}} \sum_{j \notin \mathcal{A}} \pi_j p_{ji}. \quad (6.7)$$

RELIABILITY

7.1 Introduction

Reliability theory studies the behaviour of unreliable systems, such as oil refineries, aeroplanes, alarm systems, computer systems, etcetera. Such systems consist of a great number of subsystems and components that can fail. It is often easier to describe the failure behaviour of the smaller components. The question is then how this relates to the failure behaviour of the system. In this part of the course we will address a number of important *quantitative* reliability methods, i.e. methods that quantify the failure behaviour of the system in terms of probabilities, expectations and functions.

7.2 Structure function

An often used mathematical model to describe “unreliable” systems is the following: Consider a system which consists of n **components**. Each component is either **functioning** or **failed**. Suppose that the state of the system also can only be either functioning or failed. We wish to express the state of the system in term of the state of the components.

We can do this neatly by defining binary random variables $x_i, i = 1, \dots, n$ which represent the state of the components: $x_i = 1$ if the i th component works, and 0 else. The state of the system s , say, is again a binary random variable (1 if the system works and 0 else). We now assume that s is completely determined by the vector $\mathbf{x} = (x_1, \dots, x_n)$ of component states. In other words, we assume that there exists a function $\varphi : \{0, 1\}^n \rightarrow \{0, 1\}$ such that

$$s = \varphi(\mathbf{x}) .$$

This function is called the **structure function** or *system function* of the system.

To determine the structure function of the system it often helps to have a graphical representation of system.

■ **Example 7.1 (Aeroplane)** A 4-engine aeroplane is able to fly on just one engine on each wing.

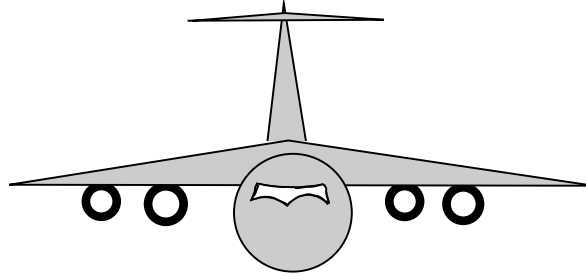


Figure 7.1: An aeroplane with 4 engines

Number the engines 1,2 (left wing) and 3,4 (right wing). We can represent the system graphically in the following way:

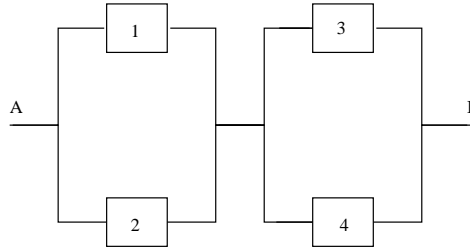


Figure 7.2: The system works if there is a “path” from A to B

The structure function can be given via the “truth table” below (1 = working, 0 = failed)

x_1	x_2	x_3	x_4	s	x_1	x_2	x_3	x_4	s
0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	0	1	0	1
0	0	1	1	0	1	0	1	1	1
0	1	0	0	0	1	1	0	0	0
0	1	0	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1

We can write this as (check this yourself)

$$\varphi(\mathbf{x}) = (1 - (1 - x_1)(1 - x_2))(1 - (1 - x_3)(1 - x_4)).$$

■

Series, Parallel and k -out-of- n system

A system that only functions when *all* components function is called a **series** system. A graphical representation is given below.

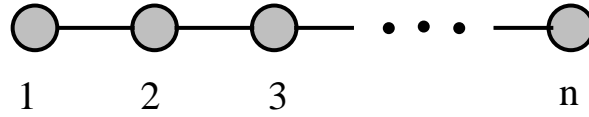


Figure 7.3: Series system

The structure function is obviously given by

$$\varphi(\mathbf{x}) = \min\{x_1, \dots, x_n\} = x_1 \cdots x_n = \prod_{i=1}^n x_i.$$

A system which functions as long as at least one component functions is called a **parallel** system. A graphical representation is given below.

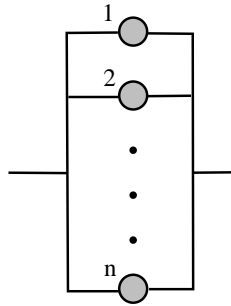


Figure 7.4: Parallel system

The system function is given by

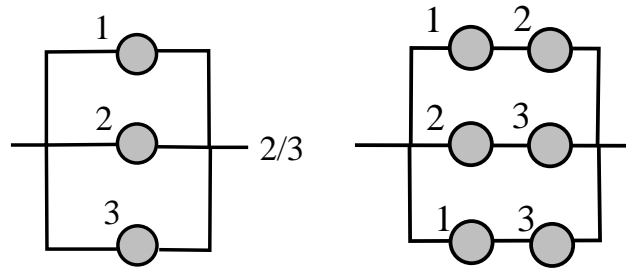
$$\varphi(\mathbf{x}) = \max\{x_1, \dots, x_n\} = 1 - (1 - x_1) \cdots (1 - x_n).$$

Sometimes we write

$$1 - (1 - x_1) \cdots (1 - x_n) = \prod_{i=1}^n x_i.$$

A k -out-of- n system is a system which works if and only if at least k of the n components are functioning. Of course the parallel and series systems are special examples of k -out-of- n systems.

A graphical representations for 2-out-of-3 system is given below. Notice that the last representation has *duplicate* nodes.



Exercise 7.1 Determine the structure function of a k -out-of- n system.

Exercise 7.2 Cities A,B and C are connected to each other and to the water supply W via unreliable pipes (1,2,3,4,5). Can you find an easy expression for the structure function? Note:

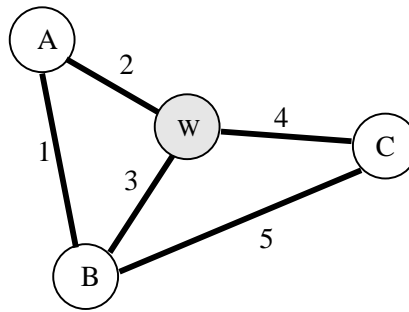


Figure 7.5: The system works if the cities A,B,C can be supplied by W.

this is different type of graphical representation than in the plane example, because here the *links* are unreliable.

We now show a number of techniques that can be used to simplify the quest for the structure function.

Modules

Often a system consists of combinations of series and parallel structures. The determination of the structure function for such systems can be handled in stages. The following example explains the procedure. Consider the top system of Figure 7.6. We can view the system as consisting of component 1 and *modules* 1^* and 2^* . This gives the second system in Figure 7.6. Similarly, we can view this last system as a series system consisting of component 1 and module 1^{**} (last system in Figure 7.6). Now define s as the state of the system, z_1 as the state of module 1^{**} , y_i the state of module i^* and x_i the state of component i , then, working “backwards”, we have

$$s = x_1 z_1,$$

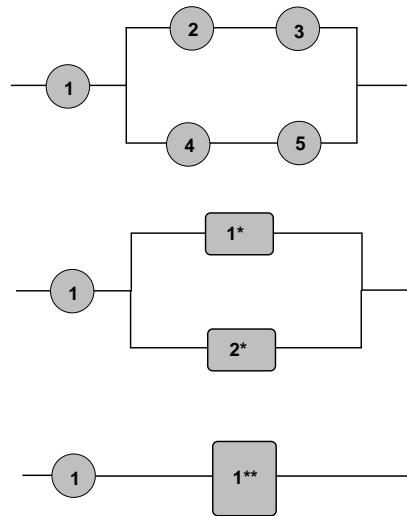


Figure 7.6: Decomposition into modules

$$z_1 = 1 - (1 - y_1)(1 - y_2),$$

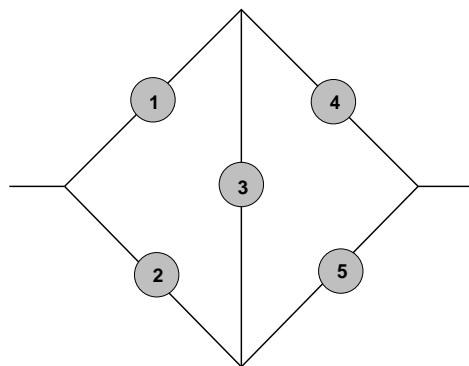
$$y_1 = x_2 x_3 \quad \text{and} \quad y_2 = x_4 x_5.$$

And successive substitution gives

$$\varphi(\mathbf{x}) = s = x_1(1 - (1 - x_2 x_3)(1 - x_4 x_5)).$$

Pivot method

We again explain this method using an example. Look at the **bridge** system below. It cannot be reduced to a set of series and parallel networks. Suppose we *pivot* around component 3.



That is, we consider the system both for the case where component 3 is working and not working. For both cases the reduced structure function is easy to determine. The original structure function is now given by

$$\varphi(\mathbf{x}) = x_3 z_1 + (1 - x_3) z_2,$$

where z_1 is the state of the system when component 3 is working and z_2 the state of the system when component 3 is not working.

Exercise 7.3 Determine z_1 and z_2 and complete the structure function.

Method of paths and cuts

We assume from now on that the structure function is **monotone**: for all vectors \mathbf{x} and \mathbf{y}

$$\mathbf{x} < \mathbf{y} \Rightarrow \varphi(\mathbf{x}) \leq \varphi(\mathbf{y}),$$

where $\mathbf{x} < \mathbf{y}$ means that $x_i \leq y_i$ for all i and $x_i < y_i$ for at least one i .

A **minimal path vector** (MPV) is a vector \mathbf{x} such that

$$\varphi(\mathbf{x}) = 1,$$

and

$$\varphi(\mathbf{y}) = 0 \text{ for all } \mathbf{y} < \mathbf{x}.$$

A **minimal cut vector** (MCV) is a vector \mathbf{x} such that

$$\varphi(\mathbf{x}) = 0,$$

and

$$\varphi(\mathbf{y}) = 1 \text{ for all } \mathbf{y} > \mathbf{x}.$$

The **minimal path set** corresponding to the MPV \mathbf{x} is the set of indices i for which $x_i = 1$. The **minimal cut set** corresponding to the MCV \mathbf{x} is the set of indices i for which $x_i = 0$.

■ **Example 7.2 (Bridge System)** For the bridge system the minimal path sets are

$$\{1, 4\}, \{2, 5\}, \{1, 3, 5\}, \{2, 3, 4\},$$

and the minimal cuts sets are

$$\{1, 2\}, \{4, 5\}, \{1, 3, 5\}, \{2, 3, 4\}.$$

The minimal path vector corresponding to the minimal path set $\{1, 4\}$ is the vector $(1, 0, 0, 1, 0)$. The minimal cut vector corresponding to the minimal cut set $\{1, 2\}$ is the vector $(0, 0, 1, 1, 1)$. ■

The use of minimal paths sets and cut sets is that these completely determine the structure function. In particular, let P_1, \dots, P_m be the minimal path sets and K_1, \dots, K_k be the minimal cut sets of a system with structure function φ . Then,

$$\varphi(\mathbf{x}) = \bigcup_{j=1}^m \prod_{i \in P_j} x_i,$$

and

$$\varphi(\mathbf{x}) = \prod_{j=1}^k \bigcup_{i \in K_j} x_i.$$

The first equation is explained by observing that the system works if and only if there is at least one minimal path set with all components working. Similarly, the second equation means that the system working if and only if at least one component is working for each of the cut sets.

Exercise 7.4 Use the method of paths and cuts to calculate the structure function for the bridge system. Give a graphical representation of the bridge system as (a) a series system of parallel modules, (b) a parallel system of series modules. Note that you have to use duplicate nodes.

7.3 Reliability function

Until now we only considered systems for which the component states were *deterministic*. However, more often than not, only the component reliabilities are known. Literally, the reliability of a component is the probability that the component will perform a required function under stated conditions for a stated period of time.

To put it in more mathematical terms, consider a system with n components and structure function φ . The state of each component i is now a *random variable* X_i , with

$$X_i = \begin{cases} 1 & \text{w.p. } p_i \\ 0 & \text{w.p. } 1 - p_i \end{cases}, \quad i = 1, \dots, n$$

The $\{p_i\}$ are the component reliabilities. We can gather them into a vector $\mathbf{p} := (p_1, \dots, p_n)$. The question is now: what is the reliability of the system, i.e., the probability that the system works. Using the structure function, this is

$$\mathbb{P}(\text{System works}) = \mathbb{P}(\varphi(\mathbf{X}) = 1) = \mathbb{E} \varphi(\mathbf{X}),$$

where \mathbf{X} is the random *vector* (X_1, \dots, X_n) . Note that the system reliability cannot, in general, be determined from the component reliabilities, *unless* X_1, \dots, X_n are *independent*.

We will assume this from now on!

Under the above assumption, $\mathbb{P}(\varphi(\mathbf{X}) = 1)$ can be expressed in terms of p_1, \dots, p_n . The function $r : \mathbf{p} \mapsto \mathbb{P}(\varphi(\mathbf{X}) = 1)$ is called the **reliability function** of the system.

■ **Example 7.3 (Reliability Functions)** We give some examples of reliability functions. Note that, to derive the reliability function, we may use similar techniques as used in the derivation of the structure function.

For the **series** we simply have

$$r(\mathbf{p}) = \mathbb{E}X_1 \cdots X_n = \prod_{i=1}^n p_i.$$

And similarly for the **parallel** system:

$$r(\mathbf{p}) = 1 - \prod_{i=1}^n (1 - p_i).$$

Note that the impression could arise that the reliability function is the same as the structure function. Although the algebraic expressions can be similar in some cases (as above) the two functions can never be the same (why not?)

For the **k -out-of- n** system, with $p_i = p$ we have

$$r(\mathbf{p}) = \mathbb{P}\left(\sum_{i=1}^n X_i \geq k\right) = \sum_{i=k}^n \binom{n}{i} p^i (1-p)^{n-i}.$$

Finally, for the **bridge system** we have

$$\begin{aligned} \varphi(\mathbf{X}) &= 1 - (1 - X_1 X_3 X_5)(1 - X_2 X_3 X_4) \\ &\quad \times (1 - X_2 X_5)(1 - X_1 X_4) \end{aligned}$$

When we take the expectation of this, we may not simply replace X_i with p_i (why not?)

However, using the fact that $X_i = X_i^2 = X_i^3$, we may write

$$\begin{aligned} \varphi(\mathbf{X}) &= X_1 X_3 X_5 + X_2 X_3 X_4 + X_2 X_5 \\ &\quad + X_1 X_4 - X_1 X_2 X_3 X_5 - X_1 X_2 X_4 X_5 \\ &\quad - X_1 X_3 X_4 X_5 - X_1 X_2 X_3 X_4 \\ &\quad - X_2 X_3 X_4 X_5 + 2X_1 X_2 X_3 X_4 X_5. \end{aligned}$$

Because here all the terms are products of independent r.v.'s we have

$$\begin{aligned} r(\mathbf{p}) &= \mathbb{E} \varphi(\mathbf{X}) = p_1 p_3 p_5 + p_2 p_3 p_4 + p_2 p_5 \\ &\quad + p_1 p_4 - p_1 p_2 p_3 p_5 - p_1 p_2 p_4 p_5 \\ &\quad - p_1 p_3 p_4 p_5 - p_1 p_2 p_3 p_4 \\ &\quad - p_2 p_3 p_4 p_5 + 2p_1 p_2 p_3 p_4 p_5. \end{aligned}$$



Reliability bounds

It should be noted that for complex systems, consisting of a many components, with little structure, it can be exceedingly difficult to derive the reliability function. In fact determining the structure function for general systems can shown to be an \mathcal{NP} -hard problem, for those of you who have heard of this term in combinatorics. However, it is sometimes possible to find good bounds for the reliability function. One such bound is given here.

Consider a monotone system with structure function φ . Let K_1, \dots, K_k be the minimal cut sets. Let E_i be the event that all the components in K_i fail. Then, the reliability of the system is

$$r(\mathbf{p}) = 1 - \mathbb{P}\left(\bigcup_{i=1}^k E_i\right).$$

A useful general result in probability is the principle of **Inclusion and Exclusion**, which states that the probability of the union of any sequence of events E_1, E_2, \dots , is equal to

$$\sum_i \mathbb{P}(E_i) - \sum_{i < j} \mathbb{P}(E_i \cap E_j) + \sum_{i < j < k} \mathbb{P}(E_i \cap E_j \cap E_k) - \dots.$$

Since successive approximations alternate around the true value, we can bound the reliability function.

The first term is the ‘rare event approximation; it holds when all p ’s are close to 1.

7.4 Non-repairable systems

In the previous section we introduced randomness into the model for a unreliable system. In this section we introduce the *time* aspect.

Consider a system with n components and structure function φ . Suppose each component functions from time 0, until a random time T_i . Assume all the T_i ’s are *independent*. Let $X_i(t)$ be the state of component i at time t . The corresponding state vector, which now depends on the time, is denoted by $\mathbf{X}(t)$. Suppose T_i has cdf F_i :

$$F_i(t) = \mathbb{P}(T_i \leq t) = \mathbb{P}(X_i(t) = 0).$$

Now let T be the failure time of the system. What is the cdf F of T ? We have,

$$F(t) = \mathbb{P}(T \leq t) = \mathbb{P}(\varphi(\mathbf{X}(t)) = 0).$$

The reliability of the system at time t is given by

$$\mathbb{P}(\varphi(X(t)) = 1) = r(1 - F_1(t), \dots, 1 - F_n(t)).$$

Hence,

$$F(t) = 1 - r(1 - F_1(t), \dots, 1 - F_n(t)).$$

This is how we can obtain the lifetime distribution (and therefore expected lifetime, etc.) of a system without repair.

Exercise 7.5 A Christmas tree is illuminated by a string of 20 light bulbs. The lifetime of each light bulb has an exponential distribution with an expectation of 1000 hours. Determine the lifetime distribution of the system. What is the expected lifetime of the system?

7.5 Lifetime distributions

Although in this course we focus on the *modeling* aspects of reliability, I cannot leave without saying something about the statistical aspects. In particular, we can have different *descriptors* of the lifetime distribution apart from the cdf of the distribution.

For example, let T be the lifetime of a component with cdf F . The **survival function** R of T is defined by

$$R(t) = \mathbb{P}(T > t) = 1 - F(t), \quad t \geq 0.$$

Exercise 7.6 Show that the expectation of T , called the *mean time to failure* (MTTF), is given by

$$\mathbb{E}T = \int_0^\infty R(t) dt.$$

Another descriptor is the **failure rate** or **hazard rate** of the component. For this, assume that F has density f . The failure rate of the component is the function h defined by

$$h(t) = \frac{f(t)}{1 - F(t)}, \quad t \geq 0.$$

We can interpret this as follows. For small ε , the probability that the component will fail in the interval $[t, t + \varepsilon]$ given that the component is still alive at time t is (approximately) $h(t)\varepsilon$. That is,

$$\mathbb{P}(T \in [t, t + \varepsilon] | T > t) \approx \frac{\varepsilon f(t)}{1 - F(t)}.$$

Thus, in the true sense of the word, $h(t)$ gives the “rate” of failure at time t . When $h(t)$ is increasing in t , the component is said to have an **Increasing Failure Rate** distribution. In

this case the component gradually gets “older”. When $h(t)$ is decreasing in t , the component is said to have an **Decreasing Failure Rate** distribution. In this case the component gradually gets “younger”. Can this happen? Think about it.

The failure rate completely determines the survival function R (and hence f and F), namely

$$R(t) = e^{-\int_0^t h(u) du}.$$

Exercise 7.7 Show this.

The exponential distribution is often used to model the lifetime T of a component. Note that when T has an exponential distribution, then the failure rate is *constant*. Thus, if we assume that T has an exponential distribution, then we are in fact assuming that the machine *does not age*! This observation also follows from the **memoryless** property of the exponential distribution. Check this yourself.

Weibull distribution

Apart from the exponential distribution, the distribution that is most widely used to describe lifetimes of components is the Weibull distribution. We say that a random variable T has a **Weibull** distribution with parameter **shape** parameter β and **rate** parameter λ , if it has density function f , given by

$$f(t) = \begin{cases} \lambda^\beta \beta t^{\beta-1} \exp\{-(\lambda t)^\beta\}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

Here are some properties:

- The cdf is given by

$$F(t) = 1 - \exp\{-(\lambda t)^\beta\}, \quad t \geq 0.$$

- The survival function of T is given by

$$R(t) = \exp\{-(\lambda t)^\beta\}, \quad t \geq 0.$$

- The Weibull distribution reduces to exponential when $\beta = 1$.

- $\mathbb{E}T = \frac{\Gamma(1 + \frac{1}{\beta})}{\lambda}$, where Γ is the gamma function.

- The Weibull family includes both increasing and decreasing failure rates.

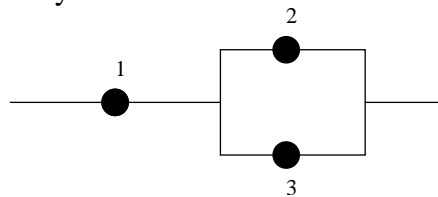
$$h(t) = \lambda^\beta \beta t^{\beta-1}, \quad t \geq 0.$$

EXERCISES AND SOLUTIONS

A.1 Problem Set 1

1. We toss a fair coin three times.
 - (a) Find the sample space, if we observe the exact sequences of Heads (= 1) and Tails (= 0).
 - (b) Find the sample space, if we observe only the total number of Heads.
2. Assign a probability to all elementary events in the sample spaces 1.(a) and 1.(b).
3. We randomly select 3 balls from an urn with 365 balls, numbered $1, \dots, 365$, noting the order.
 - (a) How many possible outcomes of the experiment are there, if we put each ball back into the urn before we draw the next?
 - (b) Answer the same question as above, but now if we *don't* put the balls back.
 - (c) Calculate the probability that in case (a) we draw 3 times the same ball.
4. Let $\mathbb{P}(A) = 0.9$ and $\mathbb{P}(B) = 0.8$. Show that $\mathbb{P}(A \cap B) \geq 0.7$.
5. What is the probability that none of 54 people in a room share the same birthday?
6. Consider the experiment of throwing 2 fair dice.
 - (a) Find the probability that both dice show the same face.
 - (b) Find the same probability, using the extra information that the sum of the dice is not greater than 4.

7. We draw 3 cards from a full deck of cards, noting the order. Number the cards from 1 to 52.
 - (a) Give the sample space. Is each elementary event equally likely?
 - (b) What is the probability that we draw 3 Aces?
 - (c) What is the probability that we draw 1 Ace, 1 King and 1 Queen?
 - (d) What is the probability that we draw no pictures (no A,K,Q,J)?
8. We draw at random a number in the interval $[0,1]$ such that each number is “equally likely”. Think of the *random generator* on you calculator.
 - (a) Determine the probability that we draw a number less than $1/2$.
 - (b) What is the probability that we draw a number between $1/3$ and $3/4$?
 - (c) Suppose we do the experiment two times (independently), giving us two numbers in $[0,1]$. What is the probability that the sum of these numbers is greater than $1/2$? Explain your reasoning.
9. Select at random 3 people from a large population. What is the probability that they all have the same birthday?
10. We draw 4 cards (at the same time) from a deck of 52, not noting the order. Calculate the probability of drawing one King, Queen, Jack and Ace.
11. In a group of 20 people there are three brothers. The group is separated at random into two groups of 10. What is the probability that the brothers are in the same group?
12. How many binary vectors are there of length 20 with exactly 5 ones?
13. We draw at random 5 balls from an urn with 20 balls (numbered $1, \dots, 20$), without replacement or order. How many different possible combinations are there?
14. In a binary transmission channel, a 1 is transmitted with probability $2/3$ and a 0 with probability $1/3$. The conditional probability of receiving a 1 when a 1 was sent is 0.95, the conditional probability of receiving a 0 when a 0 was sent is 0.90. Given that a 1 is received, what is the probability that a 1 was transmitted?
15. Consider the following system. Each component has a probability 0.1 of failing. What is the probability that the system works?



16. Two fair dice are thrown and the smallest of the face values, Z say, is noted.

(a) Give the pmf of Z in table form:

z	*	*	*	...
$\mathbb{P}(Z = z)$	*	*	*	...

(b) Calculate the expectation of Z .

17. In a large population 40% votes for A and 60% for B. Suppose we select at random 10 people. What is the probability that in this group exactly 4 people will vote for A?

18. We select “uniformly” a point in the unit square: $\{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}$. Let Z be the largest of the coordinates. Give the cdf and pdf of Z and draw their graphs.

19. A continuous random variable X has cdf F given by,

$$F(x) = \begin{cases} 0, & x < 0 \\ x^3, & x \in [0, 1] \\ 1 & x > 1. \end{cases}$$

(a) Determine the pdf of X .

(b) Calculate $\mathbb{P}(1/2 < X < 3/4)$.

(c) Calculate $\mathbb{E}[X]$.

A.2 Answer Set 1

1. (a) $\Omega = \{(0, 0, 0), \dots, (1, 1, 1)\}$.

(b) $\Omega = \{0, 1, 2, 3\}$.

2. (a) $\mathbb{P}(\{(x, y, z)\}) = 1/8$ for all $x, y, z \in \{0, 1\}$.

(b) $\mathbb{P}(\{0\}) = 1/8, \mathbb{P}(\{1\}) = 3/8, \mathbb{P}(\{2\}) = 3/8, \mathbb{P}(\{3\}) = 1/8$.

3. (a) $|\Omega| = 365^3$.

(b) $|\Omega| = 365 \times 364 \times 363$.

(c) $365/365^3 = 1/365^2$.

4. $\mathbb{P}(A \cap B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cup B)$. Since $\mathbb{P}(A \cup B) \leq 1$, we have $\mathbb{P}(A \cap B) \geq \mathbb{P}(A) + \mathbb{P}(B) - 1 = 1.7 - 1 = 0.7$.

5. $\frac{365 \times 364 \times \dots \times (365 - 54 + 1)}{365^{54}} \approx 0.016$.

6. (a) $1/6$.

(b) Let A be the event that the dice show the same face, and B the event that the sum is not greater than 4. Then $B = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (3, 1)\}$, and $A \cap B = \{(1, 1), (2, 2)\}$. Hence, $\mathbb{P}(A|B) = 2/6 = 1/3$.

7. (a) $\Omega = \{(1, 2, 3), \dots, (52, 51, 50)\}$. Each elementary event is equally likely.
 (b) $\frac{4 \times 3 \times 2}{52 \times 51 \times 50}$.
 (c) $\frac{12 \times 8 \times 4}{52 \times 51 \times 50}$.
 (d) $\frac{36 \times 35 \times 34}{52 \times 51 \times 50}$.
8. (a) $1/2$.
 (b) $5/12$.
 (c) $7/8$.
9. $1/365^2$, see question 3 (c).
10. $|\Omega| = \binom{52}{4}$ (all equally likely outcomes. Note that the outcomes are represented as (unordered sets), e.g., $\{1, 2, 3, 4\}$. Let A be the event of drawing one K, Q, J and Ace each. Then, $|A| = \binom{4}{1} \times \binom{4}{1} \times \binom{4}{1} \times \binom{4}{1} = 4^4$. Thus, $\mathbb{P}(A) = 4^4 / \binom{52}{4}$.
11. Suppose we choose 10 people to go in group 1 (the rest go in group 2). The total number of ways this can be done is $\binom{20}{10}$. Let A be the event that the brothers belong to the same group. The number of ways in which they can be chosen into group 1 is: $\binom{17}{7}$. The number of ways they can be chosen into group 2 is the same, $\binom{17}{10} = \binom{17}{7}$. Thus, $\mathbb{P}(A) = 2\binom{17}{7} / \binom{20}{10}$.
12. $\binom{20}{5}$, because we have to choose the 5 positions for the 1s, out of 20 positions.
13. $\binom{20}{5}$
14. Let B be the event that a 1 was sent, and A the event that a 1 is received. Then, $\mathbb{P}(A|B) = 0.95$, and $\mathbb{P}(A^c|B^c) = 0.90$. Thus, $\mathbb{P}(A^c|B) = 0.05$ and $\mathbb{P}(A|B^c) = 0.10$. Moreover, $\mathbb{P}(B) = 2/3$ and $\mathbb{P}(B^c) = 1/3$. By Bayes:

$$\mathbb{P}(B|A) = \frac{\mathbb{P}(A|B)\mathbb{P}(B)}{\mathbb{P}(A|B)\mathbb{P}(B) + \mathbb{P}(A|B^c)\mathbb{P}(B^c)} = \frac{0.95 \times \frac{2}{3}}{0.95 \times \frac{2}{3} + 0.10 \times \frac{1}{3}}$$

15. Let A_i be the event that component i works, $i = 1, 2, 3$, and let A be the event that the system works. We have $A = A_1 \cap (A_2 \cup A_3)$. Thus, by the independence of the A'_i s:

$$\begin{aligned} \mathbb{P}(A) &= \mathbb{P}(A_1) \times \mathbb{P}(A_2 \cup A_3) \\ &= \mathbb{P}(A_1) \times [\mathbb{P}(A_2) + \mathbb{P}(A_3) - \mathbb{P}(A_2 \cap A_3)] \\ &= \mathbb{P}(A_1) \times [\mathbb{P}(A_2) + \mathbb{P}(A_3) - \mathbb{P}(A_2)\mathbb{P}(A_3)] \\ &= 0.9 \times [0.9 + 0.9 - 0.81] \end{aligned}$$

16. (a)

z	1	2	3	4	5	6
$\mathbb{P}(Z = z)$	11/36	9/36	7/36	5/36	3/36	1/36

$$(b) \mathbb{E}[Z] = 1 \times 11/36 + 2 \times 9/36 + 3 \times 7/36 + 4 \times 5/36 + 5 \times 3/36 + 6 \times 1/36.$$

17. Let X be the number that vote for A. Then $X \sim \text{Bin}(10, 0.4)$. Hence, $\mathbb{P}(X = 4) = \binom{10}{4}(0.4)^4(0.6)^6$.

18. The region where the largest coordinate is less than z is a square with area z^2 (make a picture). Divide this area by the area of the unit square (1), to obtain $\mathbb{P}(Z \leq z) = z^2$, for all $z \in [0, 1]$. Thus,

$$F(z) = \mathbb{P}(Z \leq z) = \begin{cases} 0 & z < 0 \\ z^2 & 0 \leq z \leq 1 \\ 1 & z > 1. \end{cases}$$

Differentiate to get the pdf:

$$f(z) = \begin{cases} 2z & 0 \leq z \leq 1 \\ 0 & \text{otherwise} . \end{cases}$$

19. (a)

$$f(x) = \begin{cases} 3x^2 & 0 \leq x \leq 1 \\ 0 & \text{otherwise} . \end{cases}$$

$$(b) \int_{1/2}^{3/4} f(x) dx = F(3/4) - F(1/2) = (3/4)^3 - (1/2)^3.$$

$$(c) \mathbb{E}[X] = \int_0^1 x \cdot 3x^2 dx = 3 \int_0^1 x^3 dx = 3/4.$$

A.3 Problem Set 2

1. In a binary communication channel, 0s and 1s are transmitted with equal probability. The probability that a 0 is correctly received (as a 0) is 0.95. The probability that a 1 is correctly received (as a 1) is 0.99. Suppose we receive a 0, what is the probability that, in fact, a 1 was sent?
2. Throw two fair dice one after the other.
 - (a) What is the probability that the second die is 3, given that the sum of the dice is 6?
 - (b) What is the probability that the first die is 3 and the second not 3?
3. We flip a fair coin 20 times.
 - (a) What is the probability of exactly 10 Heads?
 - (b) What is the probability of 15 or more Heads?

4. We toss two fair dice until their sum is 12.
 - (a) What is probability that we have to wait exactly 10 tosses?
 - (b) What is the probability that we do not have to wait more than 100 tosses?
5. We independently throw 10 balls into one of 3 boxes, numbered 1, 2 and 3, with probabilities $1/4$, $1/2$ and $1/4$, respectively.
 - (a) What is the probability that box 1 has 2, box 2 has 5 and box 3 has 3 balls?
 - (b) What is the probability that box 1 remains empty?
6. Consider again the experiment where we throw two fair dice one after the other. Let the random variable Z denote the sum of the dice.
 - (a) How is Z defined as a function?
 - (b) What is the pmf of Z ?
7. Consider the random experiment of question 4. Let X be the number of tosses required until the sum of the dice is 12. Give the pmf of X .
8. We draw at random and uniformly a point from the interior of a circle with radius 4. Let R be the distance of this point to the centre of the circle.
 - (a) What is the probability that $R > 2$?
 - (b) What is the pdf of R ?
9. Let $X \sim \text{Bin}(4, 1/2)$. What is the pmf of $Y = X^2$?
10. Let $X \sim \mathcal{U}[0, 1]$. What is the pdf of $Y = X^2$?
11. Let $X \sim \mathcal{N}(0, 1)$, and $Y = 1 + 2X$. What is the pdf of Y ?
12. Let $X \sim \mathcal{N}(0, 1)$. Find $\mathbb{P}(X \leq 1.4)$ from the table of the $\mathcal{N}(0, 1)$ distribution. Also find $\mathbb{P}(X > -1.3)$.
13. Let $Y \sim \mathcal{N}(1, 4)$. Find $\mathbb{P}(Y \leq 3)$, and $\mathbb{P}(-1 \leq Y \leq 2)$.
14. If $X \sim \text{Exp}(1/2)$ what is the pdf of $Y = 1 + 2X$? Sketch the graph.
15. We draw at random 5 numbers from $1, \dots, 100$, *with replacement* (for example, drawing number 9 twice is possible). What is the probability that exactly 3 numbers are even?
16. We draw at random 5 numbers from $1, \dots, 100$, *without replacement*. What is the probability that exactly 3 numbers are even?

17. A radioactive source of material emits a radioactive particle with probability $1/100$ in each second. Let X be the number of particles emitted in one hour. X has approximately a Poisson distribution with what parameter? Draw (with the use of a computer) or sketch the pmf of X .
18. An electrical component has a lifetime X that is exponentially distributed with parameter $\lambda = 1/10$ per year. What is the probability the component is still alive after 5 years?
19. A random variable X takes the values 0, 2, 5 with probabilities $1/2, 1/3, 1/6$, respectively. What is the expectation of X ?
20. A random variable X has expectation 2 and variance 1. Calculate $\mathbb{E}[X^2]$.
21. We draw at random a 10 balls from an urn with 25 red and 75 white balls. What is the expected number of red balls amongst the 10 balls drawn? Does it matter if we draw the balls with or without replacement?
22. Let $X \sim \mathcal{U}[0, 1]$. Calculate $\text{Var}(X)$.
23. If $X \sim \mathcal{U}[0, 1]$, what is the expectation of $Y = 10 + 2X$?
24. We repeatedly throw two fair dice until two sixes are thrown. What is the expected number of throws required?
25. Suppose we divide the population of Brisbane (say 1,000,000 people) randomly in groups of 3.
 - (a) How many groups would you expect there to be in which all persons have the same birthday?
 - (b) What is the probability that there is at least one group in which all persons have the same birthday?
26. An electrical component has an exponential lifetime with expectation 2 years.
 - (a) What is the probability that the component is still functioning after 2 years?
 - (b) What is the probability that the component is still functioning after 10 years, given it is still functioning after 7 years?
27. Let $X \sim \mathcal{N}(0, 1)$. Prove that $\text{Var}(X) = 1$. Use this to show that $\text{Var}(Y) = \sigma^2$, for $Y \sim \mathcal{N}(\mu, \sigma^2)$.
28. let $X \sim \text{Exp}(1)$. Use the Moment Generating Function to show that $\mathbb{E}[X^n] = n!$.
29. Explain how to generate random numbers from the $\text{Exp}(10)$ distribution. Sketch a graph of the scatterplot of 10 such numbers.

30. Explain how to generate random numbers from the $\mathcal{U}[10, 15]$ distribution. Sketch a graph of the scatterplot of 10 such numbers.
31. Suppose we can generate random numbers from the $\mathcal{N}(0, 1)$ distribution, e.g., via Matlab's `randn` function. How can we generate from the $\mathcal{N}(3, 9)$ distribution?

A.4 Answer Set 2

1. $\frac{0.01 \times \frac{1}{2}}{0.01 \times \frac{1}{2} + 0.95 \times \frac{1}{2}}$ (Bayes' rule).
2. (a) $\frac{1}{5}$ (conditional probability, the possible outcomes are $(1, 5), (2, 4), \dots, (5, 1)$. In only one of these the second die is 3).
(b) $\frac{1}{6} \times \frac{5}{6}$ (independent events).
3. (a) $\binom{20}{10}/2^{20} \approx 0.176197$.
(b) $\sum_{k=15}^{20} \binom{20}{k}/2^{20} \approx 0.0206947$.
4. (a) $\left(\frac{35}{36}\right)^9 \frac{1}{36} \approx 0.021557$. (geometric formula)
(b) $1 - \left(\frac{35}{36}\right)^{100} \approx 0.94022$.
5. (a) $\frac{10!}{2!5!3!} \left(\frac{1}{4}\right)^2 \left(\frac{1}{2}\right)^5 \left(\frac{1}{4}\right)^3 = \frac{315}{4096} \approx 0.076904$. (multinomial)
(b) $(3/4)^{10} \approx 0.0563135$.
6. (a) $Z((x, y)) = x + y$ for all $x, y \in \{1, 2, \dots, 6\}$.
(b) Range: $S_X = \{2, 3, \dots, 12\}$. Pmf: $\mathbb{P}(X = x) = \frac{6-|x-7|}{36}$, $x \in S_X$.
7. $\mathbb{P}(X = x) = \left(\frac{35}{36}\right)^{x-1} \frac{1}{36}$, $x \in \{1, 2, \dots\}$. (geometric formula)
8. (a) $\frac{\pi 16 - \pi 4}{\pi 16} = \frac{3}{4}$.
(b) First, the cdf of R is $F_R(r) = \pi r^2 / (\pi 16) = r^2 / 16$, $r \in (0, 4)$. By differentiating we obtain the pdf $f(r) = \frac{r}{8}$, $0 < r < 4$.
9. $S_Y = \{0, 1, 4, 9, 16\}$. $\mathbb{P}(Y = k^2) = \mathbb{P}(X = k) = \frac{\binom{4}{k}}{16}$, $k = 0, 1, \dots, 4$.
10. $S_Y = [0, 1]$. For $0 \leq y \leq 1$ we have $\mathbb{P}(Y \leq y) = \mathbb{P}(X^2 \leq y) = \mathbb{P}(X \leq \sqrt{y}) = \sqrt{y}$. Thus, $f_Y(y) = \frac{1}{2\sqrt{y}}$, $0 < y \leq 1$.
11. $Y \sim \mathcal{N}(1, 4)$. (affine transformation of a normal r.v. gives again a normal r.v.)

12. $\mathbb{P}(X \leq 1.4) = \Phi(1.4) \approx 0.9192$. $\mathbb{P}(X > -1.3) = \mathbb{P}(X < 1.3)$ (by symmetry of the pdf — make a picture). $\mathbb{P}(X < 1.3) = \mathbb{P}(X \leq 1.3) = \Phi(1.3) \approx 0.9032$.
13. $\mathbb{P}(Y \leq 3) = \mathbb{P}(1+2X \leq 3)$, with X standard normal. $\mathbb{P}(1+2X \leq 3) = \mathbb{P}(X \leq 1) \approx 0.8413$.
 $\mathbb{P}(-1 \leq Y \leq 2) = \mathbb{P}(-1 \leq X \leq 1/2) = \mathbb{P}(X \leq 1/2) - \mathbb{P}(X \leq -1) = \Phi(1/2) - (1 - \Phi(1)) \approx 0.6915 - (1 - 0.8413) = 0.5328$.
14. $f_Y(y) = f_X((y-1)/2)/2 = e^{-(y-1)/4}/4$, $y \geq 1$.
15. Let X be the total number of “even” numbers. Then, $X \sim \text{Bin}(5, 1/2)$. And $\mathbb{P}(X = 3) = \binom{5}{3}/32 = 10/32 = 0.3125$
16. Let X be the total number of “even” numbers. Then $X \sim \text{Hyp}(5, 50, 100)$. Hence
 $\mathbb{P}(X = 3) = \frac{\binom{50}{3}\binom{50}{2}}{\binom{100}{5}} = 6125/19206 \approx 0.318911$.
17. $\lambda = 3600/100 = 36$.
18. $e^{-5/10} \approx 0.6065$.
19. $3/2$.
20. 5
21. It does matter for the distribution of the number of red balls, which is $\text{Bin}(10, 1/4)$ if we replace and $\text{Hyp}(10, 25, 100)$ if we don’t replace. However the expectation is the same for both cases: 2.5.
22. $\text{Var}(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2$. By symmetry $\mathbb{E}X = 1/2$. And $\mathbb{E}X^2 = \int_0^1 x^2 dx = 1/3$. So $\text{Var}(X) = 1/12$.
23. $10 + 2\mathbb{E}X = 11$.
24. 36 (expectation for the $\text{Geom}(1/36)$ distribution).
25. Let N be the number of groups in which each person has the same birthday. Then $N \sim \text{Bin}(333333, 1/365^2)$. Hence (a) $\mathbb{E}N \approx 2.5$, and (b) $\mathbb{P}(N > 0) = 1 - \mathbb{P}(N = 0) = 1 - (1 - 1/365^2)^{333333} \approx 0.92$. [Alternatively N has approximately a $\text{Poi}(2.5)$ distribution, so $\mathbb{P}(N = 0) \approx e^{-2.5}$, which gives the same answer 0.92.]
26. First note $\lambda = 1/2$. Let X be the lifetime.
- (a) $\mathbb{P}(X > 2) = e^{2/2} = e^{-1} \approx 0.368$.
- (b) $\mathbb{P}(X > 10 | X > 7) = \mathbb{P}(X > 3) = e^{-1.5} \approx 0.223$ (memoryless property).
27. $\text{Var}(X) = \mathbb{E}X^2 - \mathbb{E}X = \mathbb{E}X^2 = \int_{-\infty}^{\infty} \frac{x^2 e^{-x^2/2}}{\sqrt{2\pi}} dx = \int_{-\infty}^{\infty} x \frac{xe^{-x^2/2}}{\sqrt{2\pi}} dx = \int_{-\infty}^{\infty} \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx = 1$. Note that we have used partial integration in the fifth equality. For general $Y \sim \mathcal{N}(\mu, \sigma^2)$, write $Y = \mu + \sigma X$, so that $\text{Var}(Y) = \text{Var}(\mu + \sigma X) = \sigma^2 \text{Var}(X) = \sigma^2$.

28. $M(s) = \lambda/(\lambda - s)$. Differentiate: $M'(s) = \lambda/(\lambda - s)^2$, $M''(s) = 2\lambda/(\lambda - s)^3$, ..., $M^{(n)}(s) = n! \lambda/(\lambda - s)^{n+1}$. Now apply the moment formula.
29. Draw $U \sim \mathcal{U}(0, 1)$. Return $X = -\frac{1}{10} \ln U$.
30. Draw $U \sim \mathcal{U}(0, 1)$. Return $X = 10 + 5U$.
31. Draw $X \sim \mathcal{N}(0, 1)$. Return $Y = 3 + 3X$.

A.5 Problem Set 3

- Consider the random experiment where we draw independently n numbers from the interval $[0, 1]$; each number in $[0, 1]$ being equally likely to be drawn. Let the independent and $\mathcal{U}[0, 1]$ -distributed random variables X_1, \dots, X_n represent the numbers that are to be drawn.
 - Let M be the smallest of the n numbers, and \bar{X} the average of the n numbers. Express M and \bar{X} in terms of X_1, \dots, X_n .
 - Determine the pdf of M .
 - Give the expectation and variance of \bar{X} .
- The joint pmf of X and Y is given by the table

x	y			
	1	3	6	8
2	0	0.1	0.1	0
5	0.2	0	0	0
6	0	0.2	0.1	0.3

- Determine the (marginal) pmf of X and of Y .
 - Are X and Y independent?
 - Calculate $\mathbb{E}[X^2 Y]$.
- Explain how, in principle we can calculate

$$\mathbb{P}\left(\frac{X_1^2 + \sin(X_2)}{X_1^2 X_2} > 1\right),$$

if we know the joint pdf of X_1 and X_2 .

- Suppose X_1, X_2, \dots, X_n are independent random variables, with cdfs F_1, F_2, \dots, F_n , respectively. Express the cdf of $M = \max(X_1, \dots, X_n)$ in terms of the $\{F_i\}$.

5. Let X_1, \dots, X_6 be the weights of 6 people, selected from a large population. Suppose the weights have a normal distribution with a mean of 75 kg and a standard deviation of 10 kg. What do $Y_1 = 6X_1$ and $Y_2 = X_1 + \dots + X_6$ represent, physically? Explain why Y_1 and Y_2 have different distributions. Which one has the smallest variance?
6. Let $X \sim \text{Bin}(100, 1/4)$. Approximate, using the CLT, the probability $\mathbb{P}(20 \leq X \leq 30)$.
7. Let X and Y be independent and $\text{Exp}(1)$ distributed. Consider the coordinate transformation

$$x = uv, \quad y = u - uv \quad (\text{thus } u = x + y \text{ and } v = x/(x + y)).$$

Let $U = X + Y$ and $V = X/(X + Y)$.

- (a) Determine the Jacobian of the above coordinate transformation.
 - (b) Determine the joint pdf of U and V .
 - (c) Show that U and V are independent.
8. A random vector (X, Y) has joint pdf f , given by

$$f(x, y) = 2e^{-x-2y}, \quad x > 0, y > 0.$$
 - (a) Calculate $\mathbb{E}[XY]$.
 - (b) Calculate the covariance of $X + Y$ and $X - Y$.
9. Consider the random experiment where we make repeated measurements of the voltage across a resistor in an electric circuit. Let X_1, \dots, X_n be the voltage readings. We assume that the X_1, \dots, X_n are independent and normally distributed with the same (unknown) mean μ and (unknown) variance σ^2 . Suppose x_1, \dots, x_n are the outcomes of the random variables X_1, \dots, X_n . Let $\bar{X} = (X_1 + \dots + X_n)/n$.
 - (a) How would you estimate the unknown parameter μ if you had the data x_1, \dots, x_n ?
 - (b) Show that $\mathbb{E}\bar{X} = \mu$.
 - (c) Show that $\text{Var}\bar{X}$ goes to 0 as $n \rightarrow \infty$.
 - (d) What is the distribution of \bar{X} ?
 - (e) Discuss the implications of (b) and (c) for your estimation of the unknown μ .
10. Let $X \sim \text{Bin}(100, 1/2)$. Approximate, using the CLT, the probability $\mathbb{P}(X \geq 60)$.
11. Let X have a uniform distribution on the interval $[1, 3]$. Define $Y = X^2 - 4$. Derive the probability density function (pdf) of Y . Make sure you also specify where this pdf is zero.
12. Let $X \sim \mathcal{U}(1, 3)$. Define $Y = 4X + 5$. What is the distribution of Y ?

13. Let $Y \sim \mathcal{N}(2, 5)$.
- Sketch the pdf and cdf of Y .
 - Calculate $\mathbb{P}(Y \leq 5)$. [Use the table for the cdf of the $\mathcal{N}(0, 1)$ -distribution.]
 - Let $Z = 3Y - 4$. Determine $\mathbb{P}(1 \leq Z \leq 5)$. [Use the table for the cdf of the $\mathcal{N}(0, 1)$ -distribution.]
14. Some revision questions. Please make sure you can comfortably answer the questions below *by heart*.
- Give the formula for the pmf of the following distributions:
 - $\text{Bin}(n, p)$,
 - $\text{Geom}(p)$,
 - $\text{Poi}(\lambda)$.
 - Give the formula for the pdf of the following distributions:
 - $\mathcal{U}(a, b)$,
 - $\text{Exp}(\lambda)$,
 - $\mathcal{N}(0, 1)$.
 - Give examples of random experiments where the distributions in (a) and (b) above occur.
15. Random variables X_1, X_2, \dots are independent and have a standard normal distribution. Let $Y_1 = X_1$, $Y_2 = X_1 + X_2$, and, generally, $Y_n = X_1 + \dots + X_n$, $n = 1, 2, \dots$
- Sketch a typical outcome of X_1, X_2, \dots, X_n and Y_1, Y_2, \dots, Y_n for large n .
 - Determine $\mathbb{E}Y_n$ and $\text{Var}(Y_n)$.
 - Determine $\text{Cov}(Y_m, Y_n)$, $m \leq n$.
16. A lift can carry a maximum of 650 kg. Suppose that the weight of a person is normally distributed with expectation 75 kg and standard deviation 10 kg. Let Z_n be the total weight of n randomly selected persons.
- Determine the probability that $Z_8 \geq 650$.
 - Determine n such that $\mathbb{P}(Z_n \geq 650) \leq 0.01$.
17. The thickness of a printed circuit board is required to lie between the specification limits of $0.150 - 0.004$ and $0.150 + 0.004$ cm. A machine produces circuit boards with a thickness that is normally distributed with mean 0.151 cm and standard deviation 0.003 cm.
- What is the probability that the thickness X of a circuit board which is produced by this machine falls within the specification limits?

- (b) Now consider the mean thickness $\bar{X} = (X_1 + \cdots + X_{25})/25$ for a batch of 25 circuit boards. What is the probability that this batch mean will fall within the specification limits? Assume that X_1, \dots, X_{25} are independent random variables with the same distribution as X above.
18. We draw n numbers independently and uniformly from the interval $[0,1]$ and note their sum S_n .
- (a) Draw the graph of the pdf of S_2 .
- (b) What is approximately the distribution of S_{20} ?
- (c) Calculate the probability that the *average* of the 20 numbers is greater than 0.6.
19. Consider the following game: You flip 10 fair coins, all at once, and count how many Heads you have. I'll pay you out the squared number of Heads, in dollars. However, you will need to pay me some money in advance. How much would you prepare to give me if you could play this game as many times as you'd like?

A.6 Answer Set 3

1. (a) $M = \min(X_1, \dots, X_n)$ and $\bar{X} = (X_1 + \cdots + X_n)/n$.
- (b) $\mathbb{P}(M > x) = \mathbb{P}(X_1 > x, \dots, X_n > x) = (\text{by indep. of } X_1, \dots, X_n) \mathbb{P}(X_1 > x) \cdots \mathbb{P}(X_n > x) = (1 - x)^n$, for all $0 \leq x \leq 1$. Hence the cdf of M is given by $F_M(x) = 1 - (1 - x)^n$, $0 \leq x \leq 1$. Consequently, the pdf of M is given by $f_M(x) = n(1 - x)^{n-1}$, $0 \leq x \leq 1$.
- (c) $\mathbb{E}\bar{X} = \mathbb{E}\left(\frac{X_1 + \cdots + X_n}{n}\right) = \frac{1}{n}(\mathbb{E}X_1 + \cdots + \mathbb{E}X_n) = \frac{1}{n}n\mathbb{E}X_1 = \frac{1}{2}$.
 $\mathbb{V}\text{ar}(\bar{X}) = \mathbb{V}\text{ar}\left(\frac{X_1 + \cdots + X_n}{n}\right) \underbrace{=}_{\text{by indep.}} \frac{1}{n^2}(\mathbb{V}\text{ar}X_1 + \cdots + \mathbb{V}\text{ar}X_n) = \frac{1}{n^2}n\mathbb{V}\text{ar}(X_1) = \frac{1}{12n}$.
2. (a) $\mathbb{P}(X = 2) = 0.2$, $\mathbb{P}(X = 5) = 0.2$, $\mathbb{P}(X = 6) = 0.6$.
 $\mathbb{P}(Y = 1) = 0.2$, $\mathbb{P}(Y = 3) = 0.3$, $\mathbb{P}(Y = 6) = 0.2$, $\mathbb{P}(Y = 8) = 0.3$.
- (b) No. For example $\mathbb{P}(X = 2, Y = 1) = 0 \neq \mathbb{P}(X = 2) \cdot \mathbb{P}(Y = 1)$
- (c) $\mathbb{E}[X^2Y] = 0.1(2^2 \cdot 3) + 0.1(2^2 \cdot 6) + 0.2(5^2 \cdot 1) + 0.2(6^2 \cdot 3) + 0.1(6^2 \cdot 6) + 0.3(6^2 \cdot 8) = 138.2$.
3. By integrating the joint pdf over the region $A = \{(x_1, x_2) : \frac{x_1^2 + \sin(x_2)}{x_1^2 x_2} > 1\}$.
4. $\mathbb{P}(M \leq m) = \mathbb{P}(X_1 \leq x_1, \dots, X_n \leq x_n) = \mathbb{P}(X_1 \leq x_1) \cdots \mathbb{P}(X_n \leq x_n) = F_1(m) \cdots F_n(m)$.
5. Y_2 represents the sum of the weights of 6 different people, whereas Y_1 represents 6 times the weight of 1 person. The expectation of both Y_1 and Y_2 is 6×75 . However, the variance of Y_1 is $6^2 \mathbb{V}\text{ar}(X_1) = 3600$, whereas the variance of Y_2 is 6 times smaller: $6 \mathbb{V}\text{ar}(X_1) = 600$. Thus, $Y_1 \sim \mathcal{N}(450, 3600)$ and $Y_2 \sim \mathcal{N}(450, 600)$.

6. $\mathbb{P}(20 \leq X \leq 30) \approx \mathbb{P}(20 \leq Y \leq 30)$, with $Y \sim \mathcal{N}(100 \times \frac{1}{4}, 100 \times \frac{1}{4} \times \frac{3}{4}) = \mathcal{N}(25, 75/4)$. We have

$$\mathbb{P}(20 \leq Y \leq 30) = \mathbb{P}\left(\frac{20 - 25}{\sqrt{75/4}} \leq Z \leq \frac{30 - 25}{\sqrt{75/4}}\right) = \mathbb{P}(-1.1547 \leq Z \leq 1.1547),$$

where $Z \sim \mathcal{N}(0, 1)$. The cdf of the standard normal distribution in 1.1547 is $\mathbb{P}(Z \leq 1.1547) = \Phi(1.1547) = 0.875893$. Hence, $\mathbb{P}(-1.1547 \leq Z \leq 1.1547) = \Phi(1.1547) - (1 - \Phi(1.1547)) = 2\Phi(1.1547) - 1 = 0.752$. [The exact answer is 0.796682.]

7. (a) The Jacobian is

$$\left| \det \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix} \right| = \left| \det \begin{pmatrix} v & u \\ 1 - v & -u \end{pmatrix} \right| = |-u| = u.$$

- (b) The joint pdf of U and V follows from the transformation rule:

$$f_{U,V}(u, v) = f_{X,Y}(x, y) u = e^{-(x+y)} u = e^{-u} u,$$

for $u \geq 0$ and $0 \leq v \leq 1$.

- (c) By integrating over u we see that V is uniformly distributed over $[0, 1]$, and by integrating over v we find that $f_U(u) = ue^{-u}$, $u \geq 0$. Thus, the joint pdf of U and V is equal to the marginal pdfs of U and V , and hence U and V are independent.
8. Note that $f(x, y)$ can be written as the product of $f_1(x) = e^{-x}$, $x \geq 0$ and $f_2(y) = 2e^{-2y}$, $y \geq 0$. It follows that X and Y are independent random variables, and that $X \sim \text{Exp}(1)$ and $Y \sim \text{Exp}(2)$.

- (a) Because X and Y are independent: $\mathbb{E}[XY] = \mathbb{E}[X] \times \mathbb{E}[Y] = 1 \times 1/2 = 1/2$.

- (b) $\mathbb{Cov}(X + Y, X - Y) = \mathbb{Cov}(X, X) - \mathbb{Cov}(X, Y) + \mathbb{Cov}(Y, X) - \mathbb{Cov}(Y, Y) = \mathbb{Var}(X) - \mathbb{Var}(Y) = 1 - 1/4 = 3/4$.

9. (a) Take the average $\bar{x} = (x_1 + \cdots + x_n)/n$.

- (b) $\mathbb{E}\bar{X} = \mathbb{E}[(X_1 + \cdots + X_n)/n] = \frac{1}{n}(\mathbb{E}[X_1] + \cdots + \mathbb{E}[X_n]) = \frac{1}{n}n\mu = \mu$.

- (c) $\mathbb{Var}(\bar{X}) = \mathbb{Var}[(X_1 + \cdots + X_n)/n] = \frac{1}{n^2}(\mathbb{Var}[X_1] + \cdots + \mathbb{Var}[X_n]) = \frac{1}{n^2}n\sigma^2 = \sigma^2/n$. This goes to 0 as $n \rightarrow \infty$.

- (d) $\mathcal{N}(\mu, \sigma^2/n)$.

- (e) The larger n is, the more accurately μ can be approximated with \bar{x} .

10. Similar to question 6: $\mathbb{P}(X \geq 60) \approx \mathbb{P}(Y \geq 60)$, with $Y \sim \mathcal{N}(50, 25)$. Moreover, $\mathbb{P}(Y \geq 60) = \mathbb{P}(Z \geq (60 - 50)/5) = \mathbb{P}(Z \geq 2) = 1 - \Phi(2) = 0.02275$, where $Z \sim \mathcal{N}(0, 1)$.

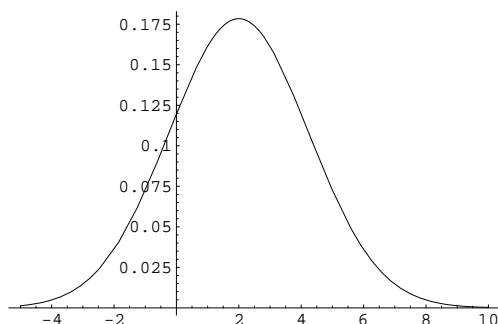
11. First draw the graph of the function $y = x^2 - 4$ on the interval $[1, 3]$. Note that the function is increasing from -3 to 5 . To find the pdf, first calculate the cdf:

$$F_Y(y) = \mathbb{P}(Y \leq y) = \mathbb{P}(X^2 - 4 \leq y) = \mathbb{P}(X \leq \sqrt{y+4}) = F_X(\sqrt{y+4}), \quad -3 \leq y \leq 5.$$

Now take the derivative with respect to y :

$$f_Y(y) = \frac{d}{dy} F_X(\sqrt{y+4}) = f_X(\sqrt{y+4}) \times \frac{1}{2\sqrt{y+4}} = \frac{1}{4\sqrt{y+4}}, \quad -3 \leq y \leq 5.$$

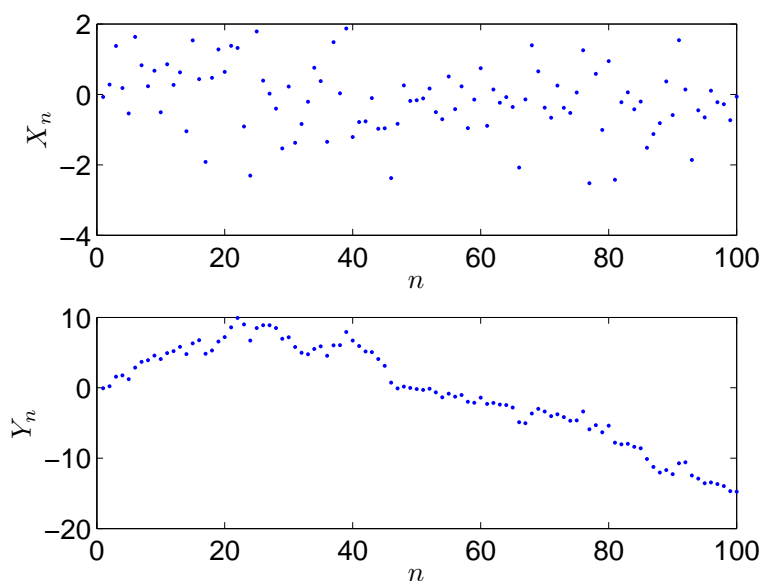
12. $\mathcal{U}(9, 17)$.

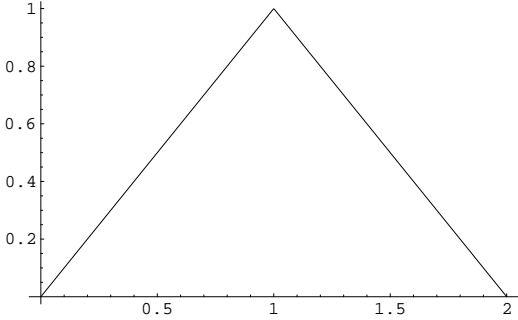


13. (a) $\mathbb{P}(Y \leq 5) = \Phi((5-2)/\sqrt{5}) \approx 0.9101$.
 (c) $Z \sim \mathcal{N}(3 \times 2 - 4, 3^2 \times 5) = \mathcal{N}(2, 45)$. $\mathbb{P}(1 \leq Z \leq 5) = \mathbb{P}((1-2)/\sqrt{45} \leq V \leq (5-2)/\sqrt{45})$, with $V \sim \mathcal{N}(0, 1)$. The latter probability is equal to $\Phi(3/\sqrt{45}) - (1 - \Phi(1/\sqrt{45})) \approx 0.2319$.

14. See the notes.

15. (a)



- (b) $\mathbb{E}[Y_n] = n\mathbb{E}[X_1] = 0$, and $\text{Var}(Y_n) = n\text{Var}(X_1) = n$.
- (c) $\text{Cov}(Y_m, Y_n) = \text{Cov}(Y_m, Y_m + X_{m+1} + \cdots + X_n) = \text{Var}(Y_m) + 0 = m$.
16. (a) $Z_8 \sim \mathcal{N}(8 \times 75, 8 \times 100) = \mathcal{N}(600, 800)$. $\mathbb{P}(Z_8 \geq 650) = 1 - \mathbb{P}(Z_8 \leq 650) = 1 - \Phi((650 - 600)/\sqrt{800}) = 1 - \Phi(1.7678) \approx 0.0385$.
- (b) $\mathbb{P}(Z_n \geq 650) = 1 - \Phi((650 - n75)/\sqrt{n100})$. For $n = 8$ the probability is 0.0385. For $n = 7$ it is much smaller than 0.01. So the largest such n is $n = 7$.
17. (a) $\mathbb{P}(0.150 - 0.004 \leq X \leq 0.150 + 0.004) = \mathbb{P}((0.150 - 0.004 - 0.151)/0.003 \leq Z \leq (0.150 + 0.004 - 0.151)/0.003) = \mathbb{P}(-1.66667 \leq Z \leq 1) = \Phi(1) - (1 - \Phi(1.66667)) \approx 0.794$, where $Z \sim \mathcal{N}(0, 1)$.
- (b) Note first that $\bar{X} \sim \mathcal{N}(0.151, (0.003)^2/25)$. Thus, $\mathbb{P}(0.150 - 0.004 \leq \bar{X} \leq 0.150 + 0.004) = \mathbb{P}((0.150 - 0.004 - 0.151)/(0.003/5) \leq Z \leq (0.150 + 0.004 - 0.151)/(0.003/5)) = \mathbb{P}(-1.66667 \times 5 \leq Z \leq 5) = \Phi(5) - (1 - \Phi(8.3333)) \approx 1$.
18. (a) 
- (b) $\mathcal{N}(10, 20/12)$, because the expectation of $\mathcal{U}(0, 1)$ is $1/2$ and the variance is $1/12$.
- (c) $\mathbb{P}(\bar{X} > 0.6) = \mathbb{P}(X_1 + \cdots + X_{20} > 12) \approx \mathbb{P}(Y > 12)$, with $Y \sim \mathcal{N}(10, 20/12)$. Now, $\mathbb{P}(Y > 12) = 1 - \mathbb{P}(Y \leq 12) = 1 - \Phi((12 - 10)/\sqrt{20/12}) = 1 - \Phi(1.5492) \approx 0.0607$.
19. The payout is X^2 , with $X \sim \text{Bin}(10, 1/2)$. The expected payout is $\mathbb{E}X^2 = \text{Var}(X) + (\mathbb{E}X)^2 = 2.5 + 5^2 = 27.5$. So, if you pay less than 27.5 dollars in advance, your expected profit per game is positive.

MORE EXERCISES

1. Two fair dice are thrown and the sum of the face values, Z say, is noted.

(a) Give the pmf of Z in table form:

z	*	*	*	...
$\mathbb{P}(Z = z)$	*	*	*	...

(b) Calculate the variance of Z .

(c) Consider the game, where a player throws two fair dice, and is paid $Y = (Z - 7)^2$ dollars, with Z the sum of face values. To enter the game the player is required to pay 5 dollars.

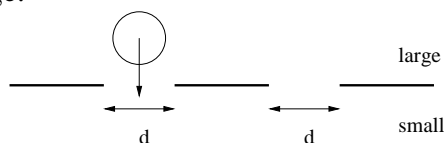
What is the expected profit (or loss) of the player, if he/she plays the game 100 times (each time paying 5 dollars to play)?

2. Consider two electrical components, both with an expected lifetime of 3 years. The lifetime of component 1, X say, is assumed to have an exponential distribution, and the lifetime of component 2, Y say, is modeled via a normal distribution with a standard deviation of $1/2$ years.

(a) What is the probability that component 1 is still functioning after 4.5 years, given that it still works after 4 years?

(b) What is the probability that component 2 is still functioning after 4.5 years, given that it still works after 4 years?

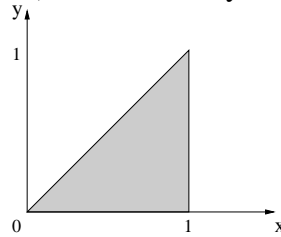
3. A sieve with diameter d is used to separate a large number of blueberries into two classes: small and large.



Suppose the diameters of the blueberries are normally distributed with an expectation $\mu = 1$ (cm) and a standard deviation $\sigma = 0.1$ (cm).

- (a) How large should the diameter of the sieve be, so that the proportion of large blueberries is 30%?
- (b) Suppose that the diameter is chosen such as in (a). What is the probability that out of 1000 blueberries, fewer than 280 end up in the “large” class?

4. We draw a random vector (X, Y) non-uniformly from the triangle $(0, 0) - (1, 0) - (1, 1)$



in the following way: First we draw X uniformly on $[0, 1]$. Then, given $X = x$ we draw Y uniformly on $[0, x]$.

- (a) Give the conditional pdf of Y given $X = x$. Specify where this conditional pdf is 0. [3]
 - (b) Find the joint pdf of X and Y .
 - (c) Calculate the pdf of Y and sketch its graph.
5. We draw n numbers independently and uniformly from the interval $[0, 1]$ and note their sum S_n .
- (a) Draw the graph of the pdf of S_2 .
 - (b) What is approximately the distribution of S_{20} ? [4]
 - (c) Calculate the probability that the *average* of the 20 numbers is greater than 0.6.
6. Two fair dice are thrown and the smallest of the face values, Z say, is noted.

- (a) Give the probability mass function (pmf) of Z in table form:

z	*	*	*	...
$\mathbb{P}(Z = z)$	*	*	*	...

- (b) Calculate the expectation of $1/Z$.
- (c) Consider the game, where a player throws two fair dice, and is paid Z dollars, with Z as above. To enter the game the player is required to pay 3 dollars.
What is the expected profit (or loss) of the player, if he/she plays the game 100 times (each time paying 3 dollars to play)?

7. Let U and V be independent random variables, with $\mathbb{P}(U = 1) = \mathbb{P}(V = 1) = 1/4$ and $\mathbb{P}(U = -1) = \mathbb{P}(V = -1) = 3/4$. Define $X = U/V$ and $Y = U + V$.

(a) Give the joint pmf of X and Y .

(b) Calculate $\mathbb{Cov}(X, Y)$.

8. In a binary transmission channel, a 1 is transmitted with probability $1/4$ and a 0 with probability $3/4$. The conditional probability of receiving a 1 when a 1 was sent is 0.90, the conditional probability of receiving a 0 when a 0 was sent is 0.95.

(a) What is the probability that a 1 is received?

(b) Given that a 1 is received, what is the probability that a 1 was transmitted?

9. Consider the probability density function (pdf) given by

$$f(x) = \begin{cases} 4e^{-4(x-1)}, & x \geq 1, \\ 0 & x < 1. \end{cases}$$

(a) If X is distributed according to this pdf f , what is the expectation of X ?

(b) Specify how one can generate a random variable X from this pdf, using a random number generator that outputs $U \sim \mathcal{U}(0, 1)$.

10. A certain type of electrical component has an exponential lifetime distribution with an expected lifetime of $1/2$ year. When the component fails it is immediately replaced by a second (new) component; when the second component fails, it is replaced by a third, etc. Suppose there are 10 such identical components. Let T be the time that the last of the components fails.

(a) What is the expectation and variance of T ?

(b) Approximate, using the central limit theorem and the table of the standard normal cdf, the probability that T exceeds 6 years.

(c) What is the exact distribution of T ?

SUMMARY OF FORMULAS

1. **Sum rule:** $\mathbb{P}(\bigcup_i A_i) = \sum_i \mathbb{P}(A_i)$,
when A_1, A_2, \dots are disjoint.
2. $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$.
3. $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$.
4. **Cdf** of X : $F(x) = \mathbb{P}(X \leq x)$, $x \in \mathbb{R}$.
5. **Pmf** of X : (discrete r.v.) $f(x) = \mathbb{P}(X = x)$.
6. **Pdf** of X : (continuous r.v.) $f(x) = F'(x)$.
7. For a discrete r.v. X :

$$\mathbb{P}(X \in B) = \sum_{x \in B} \mathbb{P}(X = x).$$

8. For a continuous r.v. X with pdf f :

$$\mathbb{P}(X \in B) = \int_B f(x) \, dx.$$

9. In particular (continuous), $F(x) = \int_{-\infty}^x f(u) \, du$.
10. Similar results 7-8 hold for random vectors; e.g.,

$$\mathbb{P}((X, Y) \in B) = \iint_B f_{X,Y}(x, y) \, dx \, dy.$$

11. Marginal from joint pdf: $f_X(x) = \int f_{X,Y}(x, y) \, dy$.

12. Important discrete distributions:

Distr.	pmf	$x \in$
Ber(p)	$p^x(1-p)^{1-x}$	$\{0, 1\}$
Bin(n, p)	$\binom{n}{x} p^x(1-p)^{n-x}$	$\{0, 1, \dots, n\}$
Poi(λ)	$e^{-\lambda} \frac{\lambda^x}{x!}$	$\{0, 1, \dots\}$
Geom(p)	$p(1-p)^{x-1}$	$\{1, 2, \dots\}$
Hyp(n, r, N)	$\frac{\binom{r}{x} \binom{N-r}{n-x}}{\binom{N}{n}}$	$\{0, \dots, n\}$

13. Important continuous distributions:

Distr.	pdf	$x \in$
$\mathcal{U}[a, b]$	$\frac{1}{b-a}$	$[a, b]$
Exp(λ)	$\lambda e^{-\lambda x}$	\mathbb{R}_+
Gamma(α, λ)	$\frac{\lambda^\alpha x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)}$	\mathbb{R}_+
$\mathcal{N}(\mu, \sigma^2)$	$\frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma}\right)^2}$	\mathbb{R}

14. **Conditional probability:** $\mathbb{P}(A | B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$.

15. **Law of total probability:**

$$\mathbb{P}(A) = \sum_{i=1}^n \mathbb{P}(A | B_i) \mathbb{P}(B_i),$$

with B_1, B_2, \dots, B_n a partition of Ω .

16. **Bayes' Rule:** $\mathbb{P}(B_j | A) = \frac{\mathbb{P}(B_j) \mathbb{P}(A | B_j)}{\sum_{i=1}^n \mathbb{P}(B_i) \mathbb{P}(A | B_i)}$.

17. **Product rule:**

$$\mathbb{P}(A_1 \cdots A_n) = \mathbb{P}(A_1) \mathbb{P}(A_2 | A_1) \cdots \mathbb{P}(A_n | A_1 \cdots A_{n-1}).$$

18. **Memoryless property** (Exp and Geom distribution):

$$\mathbb{P}(X > s + t | X > s) = \mathbb{P}(X > t), \forall s, t.$$

19. **Independent events:** $\mathbb{P}(A \cap B) = \mathbb{P}(A) \mathbb{P}(B)$.

20. **Independent random variables:** (discrete)

$$\mathbb{P}(X_1 = x_1, \dots, X_n = x_n) = \prod_{k=1}^n \mathbb{P}(X_k = x_k).$$

21. **Independent random variables:** (continuous)

$$f_{X_1, \dots, X_n}(x_1, \dots, x_n) = \prod_{k=1}^n f_{X_k}(x_k).$$

22. **Expectation** (discrete): $\mathbb{E}X = \sum_x x \mathbb{P}(X = x).$

23. (of function) $\mathbb{E} g(X) = \sum_x g(x) \mathbb{P}(X = x)$.
24. **Expectation** (continuous): $\mathbb{E}X = \int x f(x) dx$.
25. (of function) $\mathbb{E} g(X) = \int g(x) f(x) dx$,
26. Similar results 22–25 hold for random vectors.
27. **Expected sum** : $\mathbb{E}(aX + bY) = a \mathbb{E}X + b \mathbb{E}Y$.
28. **Expected product** (only if X, Y independent):
 $\mathbb{E}[X Y] = \mathbb{E}X \mathbb{E}Y$.
29. **Markov inequality**: $\mathbb{P}(X \geq x) \leq \frac{\mathbb{E}X}{x}$.
30. **$\mathbb{E}X$ and $\text{Var}(X)$ for various distributions:**

	$\mathbb{E}X$	$\text{Var}(X)$
Ber(p)	p	$p(1-p)$
Bin(n, p)	np	$np(1-p)$
Geom(p)	$\frac{1}{p}$	$\frac{1-p}{p^2}$
Poi(λ)	λ	λ
Hyp(n, pN, N)	np	$np(1-p)\frac{N-n}{N-1}$
$\mathcal{U}(a, b)$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Exp(λ)	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Gamma(α, λ)	$\frac{\alpha}{\lambda}$	$\frac{\alpha}{\lambda^2}$
$\mathcal{N}(\mu, \sigma^2)$	μ	σ^2

31. **n -th moment**: $\mathbb{E}X^n$.
32. **Covariance**: $\text{cov}(X, Y) = \mathbb{E}(X - \mathbb{E}X)(Y - \mathbb{E}Y)$.
33. **Properties of Var and Cov :**

$$\begin{aligned}
 &\text{Var}(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2. \\
 &\text{Var}(aX + b) = a^2 \text{Var}(X). \\
 &\text{cov}(X, Y) = \mathbb{E}XY - \mathbb{E}X \mathbb{E}Y. \\
 &\text{cov}(X, Y) = \text{cov}(Y, X). \\
 &\text{cov}(aX + bY, Z) = a \text{cov}(X, Z) + b \text{cov}(Y, Z). \\
 &\text{cov}(X, X) = \text{Var}(X). \\
 &\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2 \text{cov}(X, Y). \\
 &X \text{ and } Y \text{ independent} \implies \text{cov}(X, Y) = 0.
 \end{aligned}$$

34. **Probability Generating Function (PGF)**:

$$G(z) := \mathbb{E}z^N = \sum_{n=0}^{\infty} \mathbb{P}(N = n)z^n, \quad |z| < 1.$$

35. PGFs for various distributions:

Ber(p)	$1 - p + zp$
Bin(n, p)	$(1 - p + zp)^n$
Geom(p)	$\frac{zp}{1 - z(1 - p)}$
Poi(λ)	$e^{-\lambda(1 - z)}$

36. $\mathbb{P}(N = n) = \frac{1}{n!} G^{(n)}(0)$. (n -th derivative, at 0)

37. $\mathbb{E}N = G'(1)$

38. $\text{Var}(N) = G''(1) + G'(1) - (G'(1))^2$.

39. Moment Generating Function (MGF):

$$M(s) = \mathbb{E} e^{sX} = \int_{-\infty}^{\infty} e^{sx} f(x) dx ,$$

$s \in I \subset \mathbb{R}$, for r.v.'s X for which all moments exist.

40. MGFs for various distributions:

$\mathcal{U}(a, b)$	$\frac{e^{bs} - e^{as}}{s(b - a)}$
Gamma(α, λ)	$\left(\frac{\lambda}{\lambda - s}\right)^\alpha$
$\mathcal{N}(\mu, \sigma^2)$	$e^{s\mu + \sigma^2 s^2 / 2}$

41. **Moment property:** $\mathbb{E}X^n = M^{(n)}(0)$.

42. $M_{X+Y}(t) = M_X(t) M_Y(t)$, $\forall t$, if X, Y independent.

43. If $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, $i = 1, 2, \dots, n$ (independent),

$$a + \sum_{i=1}^n b_i X_i \sim \mathcal{N}\left(a + \sum_{i=1}^n b_i \mu_i, \sum_{i=1}^n b_i^2 \sigma_i^2\right).$$

44. Conditional pmf/pdf

$$f_{Y|X}(y|x) := \frac{f_{X,Y}(x,y)}{f_X(x)}, \quad y \in \mathbb{R}.$$

45. The corresponding **conditional expectation** (discrete case):

$$\mathbb{E}[Y|X = x] = \sum_y y f_{Y|X}(y|x).$$

46. **Linear transformation:** $f_Z(z) = \frac{f_X(A^{-1}z)}{|A|}$.

47. **General transformation:** $f_Z(z) = \frac{f_X(\mathbf{x})}{|J_{\mathbf{x}}(g)|}$, with $\mathbf{x} = g^{-1}(z)$, where $|J_{\mathbf{x}}(g)|$ is the Jacobian of g evaluated at \mathbf{x} .

48. Pdf of the **multivariate normal** distribution:

$$f_{\mathbf{Z}}(\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{z}-\boldsymbol{\mu})}.$$

Σ is the covariance matrix, and $\boldsymbol{\mu}$ the mean vector.

49. If \mathbf{X} is a column vector with independent $\mathcal{N}(0, 1)$ components, and B is a matrix with $\Sigma = BB^T$ (such a B can always be found), then $\mathbf{Z} = \boldsymbol{\mu} + B\mathbf{X}$ has a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ .

50. **Weak Law of Large Numbers:**

$$\lim_{n \rightarrow \infty} \mathbb{P}\left(\left|\frac{S_n}{n} - \mu\right| > \varepsilon\right) = 0, \quad \forall \varepsilon.$$

51. **Strong Law of Large Numbers:**

$$\mathbb{P}\left(\lim_{n \rightarrow \infty} \frac{S_n}{n} = \mu\right) = 1.$$

52. **Central Limit Theorem:**

$$\lim_{n \rightarrow \infty} \mathbb{P}\left(\frac{S_n - n\mu}{\sigma \sqrt{n}} \leq x\right) = \Phi(x),$$

where Φ is the cdf of the standard normal distribution.

53. **Normal Approximation to Binomial:** If $X \sim \text{Bin}(n, p)$, then, for large n , $\mathbb{P}(X \leq k) \approx \mathbb{P}(Y \leq k)$, where $Y \sim \mathcal{N}(np, np(1-p))$.

Other Mathematical Formulas

1. Factorial. $n! = n(n-1)(n-2) \cdots 1$. Gives the number of *permutations* (orderings) of $\{1, \dots, n\}$.
2. Binomial coefficient. $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. Gives the number *combinations* (no order) of k different numbers from $\{1, \dots, n\}$.
3. Newton's binomial theorem: $(a+b)^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k}$.
4. Geometric sum: $1 + a + a^2 + \cdots + a^n = \frac{1-a^{n+1}}{1-a}$ ($a \neq 1$).
If $|a| < 1$ then $1 + a + a^2 + \cdots = \frac{1}{1-a}$.
5. Logarithms:
 - (a) $\ln(xy) = \ln x + \ln y$.

(b) $e^{\ln x} = x$.

6. Exponential:

(a) $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$.

(b) $e^x = \lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n$.

(c) $e^{x+y} = e^x e^y$.

7. Differentiation:

(a) $(f + g)' = f' + g'$,

(b) $(fg)' = f'g + fg'$,

(c) $\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$.

(d) $\frac{d}{dx} x^n = n x^{n-1}$.

(e) $\frac{d}{dx} e^x = e^x$.

(f) $\frac{d}{dx} \ln(x) = \frac{1}{x}$.

8. Chain rule: $(f(g(x)))' = f'(g(x)) g'(x)$.

9. Integration: $\int_a^b f(x) dx = [F(x)]_a^b = F(b) - F(a)$, where $F' = f$.

10. Integration by parts: $\int_a^b f(x) G(x) dx = [F(x) G(x)]_a^b - \int_a^b F(x) g(x) dx$. (Here $F' = f$ and $G' = g$.)

11. Jacobian: Let $\mathbf{x} = (x_1, \dots, x_n)$ be an n -dimensional vector, and $g(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_n(\mathbf{x}))$ be a function from \mathbb{R}^n to \mathbb{R}^n . The *matrix of Jacobi* is the matrix of partial derivatives: $(\partial g_i / \partial x_j)$. The corresponding determinant is called the *Jacobian*. In the neighbourhood of any fixed point, g behaves like a *linear transformation* specified by the matrix of Jacobi at that point.

12. Γ function: $\Gamma(\alpha) = \int_0^\infty u^{\alpha-1} e^{-u} du$, $\alpha > 0$. $\Gamma(\alpha+1) = \alpha \Gamma(\alpha)$, for $\alpha \in \mathbb{R}_+$. $\Gamma(n) = (n-1)!$ for $n = 1, 2, \dots$. $\Gamma(1/2) = \sqrt{\pi}$.

PYTHON PRIMER

Python has become the programming language of choice for many researchers and practitioners in data science and machine learning. This appendix gives a brief introduction to the language. As the language is under constant development and each year many new packages are being released, we do not pretend to be exhaustive in this introduction. Instead, we hope to provide enough information for novices to get started with this beautiful and carefully thought-out language.

D.1 Getting Started

The main website for Python is

<https://www.python.org/>,

where you will find documentation, a tutorial, beginners' guides, software examples, and so on. It is important to note that there are two incompatible “branches” of Python, called Python 3 and Python 2. Further development of the language will involve only Python 3, and in this appendix (and indeed the rest of the book) we only consider Python 3. As there are many interdependent packages that are frequently used with a Python installation, it is convenient to install a distribution — for instance, the *Anaconda* Python distribution, available from

<https://www.anaconda.com/>.

The *Anaconda* installer automatically installs the most important packages and also provides a convenient interactive development environment (IDE), called *Spyder*.



Use the *Anaconda Navigator* to launch *Spyder*, *Jupyter notebook*, install and update packages, or open a command-line terminal.

To get started¹, try out the Python statements in the input boxes that follow. You can either type these statements at the IPython command prompt or run them as (very short) Python programs. The output for these two modes of input can differ slightly. For example, typing a variable name in the console causes its contents to be automatically printed, whereas in a Python program this must be done explicitly by calling the `print` function. Selecting (highlighting) several program lines in *Spyder* and then pressing function key² F9 is equivalent to executing these lines one by one in the console.

In Python, data is represented as an *object* or relation between objects (see also Section D.2). Basic data types are numeric types (including integers, booleans, and floats), sequence types (including strings, tuples, and lists), sets, and mappings (currently, dictionaries are the only built-in mapping type).

Strings are sequences of characters, enclosed by single or double quotes. We can print strings via the `print` function.

```
print("Hello World!")
```

```
Hello World!
```

For pretty-printing output, Python strings can be formatted using the `format` function. The bracket syntax `{i}` provides a placeholder for the *i*-th variable to be printed, with 0 being the first index. Individual variables can be formatted separately and as desired; formatting syntax is discussed in more detail in Section D.9.

```
print("Name:{1} (height {2} m, age {0})".format(111, "Bilbo", 0.84))
```

```
Name:Bilbo (height 0.84 m, age 111)
```

Lists can contain different types of objects, and are created using square brackets as in the following example:

```
x = [1, 'string', "another string"] # Quote type is not important
```

```
[1, 'string', 'another string']
```

Elements in lists are indexed starting from 0, and are *mutable* (can be changed):

¹We assume that you have installed all the necessary files and have launched *Spyder*.

²This may depend on the keyboard and operating system.

```
x = [1,2]
x[0] = 2 # Note that the first index is 0
x
[2,2]
```

In contrast, tuples (with round brackets) are *immutable* (cannot be changed). Strings are immutable as well.

```
x = (1,2)
x[0] = 2
TypeError: 'tuple' object does not support item assignment
```

Lists can be accessed via the *slice* notation `[start:end]`. It is important to note that `end` is the index of the first element that will *not* be selected, and that the first element has index 0. To gain familiarity with the slice notation, execute each of the following lines.

```
a = [2, 3, 5, 7, 11, 13, 17, 19, 23]
a[1:4] # Elements with index from 1 to 3
a[:4]  # All elements with index less than 4
a[3:]  # All elements with index 3 or more
a[-2:] # The last two elements
[3, 5, 7]
[2, 3, 5, 7]
[7, 11, 13, 17, 19, 23]
[19, 23]
```

An *operator* is a programming language construct that performs an action on one or more operands. The action of an operator in Python depends on the type of the operand(s). For example, operators such as `+`, `*`, `-`, and `%` that are arithmetic operators when the operands are of a numeric type, can have different meanings for objects of non-numeric type (such as strings).

```
'hello' + 'world' # String concatenation
'helloworld'
```

```
'hello' * 2 # String repetition
'hellohello'
```

```
[1,2] * 2 # List repetition
```

```
[1, 2, 1, 2]
```

```
15 % 4    # Remainder of 15/4
```

```
3
```

Some common Python operators are given in Table D.1.

D.2 Python Objects

As mentioned in the previous section, data in Python is represented by objects or relations between objects. We recall that basic data types included strings and numeric types (such as integers, booleans, and floats).

As Python is an object-oriented programming language, functions are objects too (everything is an object!). Each object has an identity (unique to each object and immutable — that is, cannot be changed — once created), a type (which determines which operations can be applied to the object, and is considered immutable), and a value (which is either mutable or immutable). The unique identity assigned to an object `obj` can be found by calling `id`, as in `id(obj)`.

Each object has a list of *attributes*, and each attribute is a reference to another object. The function `dir` applied to an object returns the list of attributes. For example, a string object has many useful attributes, as we shall shortly see. Functions are objects with the `__call__` attribute.

A class (see Section D.8) can be thought of as a template for creating a custom type of object.

```
s = "hello"
d = dir(s)
print(d, flush=True)    # Print the list in "flushed" format
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__',
... (many left out) ... 'replace', 'rfind',
'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split',
'splitlines', 'startswith', 'strip', 'swapcase', 'title',
'translate', 'upper', 'zfill']
```

Any attribute `attr` of an object `obj` can be accessed via the *dot notation*: `obj.attr`. To find more information about any object use the `help` function.

```
s = "hello"
help(s.replace)
```

```
replace(...) method of builtins.str instance
  S.replace(old, new[, count]) -> str
```

Return a copy of S with all occurrences of substring old replaced by new. If the optional argument count is given, only the first count occurrences are replaced.

This shows that the attribute `replace` is in fact a function. An attribute that is a function is called a *method*. We can use the `replace` method to create a new string from the old one by changing certain characters.

```
s = 'hello'
s1 = s.replace('e', 'a')
print(s1)
```

```
hallo
```



In many Python editors, pressing the TAB key, as in `objectname.<TAB>`, will bring up a list of possible attributes via the editor's autocompletion feature.

D.3 Types and Operators

Each object has a *type*. Three basic data types in Python are `str` (for string), `int` (for integers), and `float` (for floating point numbers). The function `type` returns the type of an object.

```
t1 = type([1,2,3])
t2 = type((1,2,3))
t3 = type({1,2,3})
print(t1,t2,t3)
```

```
<class 'list'> <class 'tuple'> <class 'set'>
```

The *assignment* operator, `=`, assigns an object to a variable; e.g., `x = 12`. An *expression* is a combination of values, operators, and variables that yields another value or variable.



Variable names are case sensitive and can only contain letters, numbers, and underscores. They must start with either a letter or underscore. Note that reserved words

such as True and False are case sensitive as well.

Python is a dynamically typed language, and the type of a variable at a particular point during program execution is determined by its most recent object assignment. That is, the type of a variable does not need to be explicitly declared from the outset (as is the case in C or Java), but instead the type of the variable is determined by the object that is currently assigned to it.

It is important to understand that a variable in Python is a *reference* to an object — think of it as a label on a shoe box. Even though the label is a simple entity, the *contents* of the shoe box (the object to which the variable refers) can be arbitrarily complex. Instead of moving the contents of one shoe box to another, it is much simpler to merely move the label.

<pre>x = [1,2] y = x # y refers to the same object as x print(id(x) == id(y)) # check that the object id's are the same y[0] = 100 # change the contents of the list that y refers to print(x)</pre>
True [100,2]

<pre>x = [1,2] y = x # y refers to the same object as x y = [100,2] # now y refers to a different object print(id(x) == id(y)) print(x)</pre>
False [1,2]

Table D.1 shows a selection of Python operators for numerical and logical variables.

Table D.1: Common numerical (left) and logical (right) operators.

+	addition	~	binary NOT
-	subtraction	&	binary AND
*	multiplication	^	binary XOR
**	power		binary OR
/	division	==	equal to
//	integer division	!=	not equal to
%	modulus		

Several of the numerical operators can be combined with an assignment operator, as in `x += 1` to mean `x = x + 1`. Operators such as `+` and `*` can be defined for other data types as

well, where they take on a different meaning. This is called operator *overloading*, an example of which is the use of `<List> * <Integer>` for list repetition as we saw earlier.

D.4 Functions and Methods

Functions make it easier to divide a complex program into simpler parts. To create a *function*, use the following syntax:

```
def <function name>(<parameter_list>):  
    <statements>
```

A function takes a list of input variables that are references to objects. Inside the function, a number of statements are executed which may modify the objects, but not the reference itself. In addition, the function may return an output object (or will return the value `None` if not explicitly instructed to return output). Think again of the shoe box analogy. The input variables of a function are labels of shoe boxes, and the objects to which they refer are the contents of the shoe boxes. The following program highlights some of the subtleties of variables and objects in Python.



Note that the statements within a function must be indented. This is Python's way to define where a function begins and ends.

```
x = [1,2,3]  
  
def change_list(y):  
    y.append(100) # Append an element to the list referenced by y  
    y[0]=0       # Modify the first element of the same list  
    y = [2,3,4]  # The local y now refers to a different list  
                # The list to which y first referred does not change  
    return sum(y)  
  
print(change_list(x))  
print(x)
```

```
9  
[0, 2, 3, 100]
```

Variables that are defined inside a function only have *local scope*; that is, they are recognized only within that function. This allows the same variable name to be used in different functions without creating a conflict. If any variable is used within a function, Python first checks if the variable has local scope. If this is not the case (the variable has not been defined

inside the function), then Python searches for that variable outside the function (the global scope). The following program illustrates several important points.

```
from numpy import array, square, sqrt

x = array([1.2, 2.3, 4.5])

def stat(x):
    n = len(x)          #the length of x
    meanx = sum(x)/n
    stdx = sqrt(sum(square(x - meanx))/n)
    return [meanx, stdx]

print(stat(x))
```

[2.6666666666666665, 1.3719410418171119]

1. Basic math functions such as `sqrt` are unknown to the standard Python interpreter and need to be imported. More on this in Section D.5 below.
2. As was already mentioned, indentation is crucial. It shows where the function begins and ends.
3. No semicolons³ are needed to end lines, but the first line of the function definition (here line 5) must end with a colon (:).
4. Lists are not arrays (vectors of numbers), and vector operations cannot be performed on lists. However, the `numpy` module is designed specifically with efficient vector/matrix operations in mind. On the second code line, we define `x` as a vector (`ndarray`) object. Functions such as `square`, `sum`, and `sqrt` are then applied to such arrays. Note that we used the default Python functions `len` and `sum`. More on `numpy` in Section D.10.
5. Running the program with `stat(x)` instead of `print(stat(x))` in line 11 will not show any output in the console.



To display the complete list of built-in functions, type (using double underscores) `dir(__builtin__)`.

D.5 Modules

A Python *module* is a programming construct that is useful for organizing code into manageable parts. To each module with name `module_name` is associated a Python file

³Semicolons can be used to put multiple commands on a single line.

`module_name.py` containing any number of definitions, e.g., of functions, classes, and variables, as well as executable statements. Modules can be imported into other programs using the syntax: `import <module_name> as <alias_name>`, where `<alias_name>` is a short-hand name for the module.

When imported into another Python file, the module name is treated as a *namespace*, providing a naming system where each object has its unique name. For example, different modules `mod1` and `mod2` can have different `sum` functions, but they can be distinguished by prefixing the function name with the module name via the dot notation, as in `mod1.sum` and `mod2.sum`. For example, the following code uses the `sqrt` function of the `numpy` module.

```
import numpy as np
np.sqrt(2)

1.4142135623730951
```

A Python *package* is simply a directory of Python modules; that is, a collection of modules with additional startup information (some of which may be found in its `__path__` attribute). Python's built-in module is called `__builtins__`. Of the great many useful Python modules, Table D.2 gives a few.

Table D.2: A few useful Python modules/packages.

<code>datetime</code>	Module for manipulating dates and times.
<code>matplotlib</code>	MATLAB TM -type plotting package
<code>numpy</code>	Fundamental package for scientific computing, including random number generation and linear algebra tools. Defines the ubiquitous <code>ndarray</code> class.
<code>os</code>	Python interface to the operating system.
<code>pandas</code>	Fundamental module for data analysis. Defines the powerful <code>DataFrame</code> class.
<code>pytorch</code>	Machine learning library that supports GPU computation.
<code>scipy</code>	Ecosystem for mathematics, science, and engineering, containing many tools for numerical computing, including those for integration, solving differential equations, and optimization.
<code>requests</code>	Library for performing HTTP requests and interfacing with the web.
<code>seaborn</code>	Package for statistical data visualization.
<code>sklearn</code>	Easy to use machine learning library.
<code>statsmodels</code>	Package for the analysis of statistical models.

The `numpy` package contains various subpackages, such as `random`, `linalg`, and `fft`. More details are given in Section D.10.



When using *Spyder*, press Ctrl+I in front of any object, to display its help file in a separate window.

As we have already seen, it is also possible to import only specific functions from a module using the syntax: `from <module_name> import <fnc1, fnc2, ...>`.

```
from numpy import sqrt, cos
sqrt(2)
cos(1)
```

```
1.4142135623730951
0.54030230586813965
```

This avoids the tedious prefixing of functions via the (alias) of the module name. However, for large programs it is good practice to always use the prefix/alias name construction, to be able to clearly ascertain precisely which module a function being used belongs to.

D.6 Flow Control

Flow control in Python is similar to that of many programming languages, with conditional statements as well as `while` and `for` loops. The syntax for `if-then-else` flow control is as follows.

```
if <condition1>:
    <statements>
elif <condition2>:
    <statements>
else:
    <statements>
```

Here, `<condition1>` and `<condition2>` are logical conditions that are either `True` or `False`; logical conditions often involve comparison operators (such as `==`, `>`, `<=`, `!=`). In the example above, there is one `elif` part, which allows for an “else if” conditional statement. In general, there can be more than one `elif` part, or it can be omitted. The `else` part can also be omitted. The colons are essential, as are the indentations.

The `while` and `for` loops have the following syntax.

```
while <condition>:
    <statements>
```

```
for <variable> in <collection>:
    <statements>
```

Above, <collection> is an iterable object (see Section D.7 below). For further control in for and while loops, one can use a `break` statement to exit the current loop, and the `continue` statement to continue with the next iteration of the loop, while abandoning any remaining statements in the current iteration. Here is an example.

```
import numpy as np
ans = 'y'
while ans != 'n':
    outcome = np.random.randint(1,6+1)
    if outcome == 6:
        print("Hooray a 6!")
        break
    else:
        print("Bad luck, a", outcome)
    ans = input("Again? (y/n) ")
```

D.7 Iteration

Iterating over a sequence of objects, such as used in a for loop, is a common operation. To better understand how iteration works, we consider the following code.

```
s = "Hello"
for c in s:
    print(c, '*', end= ' ')
```

```
H * e * l * l * o *
```

A string is an example of a Python object that can be iterated. One of the methods of a string object is `__iter__`. Any object that has such a method is called an *iterable*. Calling this method creates an *iterator* — an object that returns the next element in the sequence to be iterated. This is done via the method `__next__`.

```
s = "Hello"
t = s.__iter__() # t is now an iterator. Same as iter(s)
print(t.__next__() ) # same as next(t)
print(t.__next__() )
print(t.__next__() )
```

```
H
e
l
```

The inbuilt functions `next` and `iter` simply call these corresponding double-underscore functions of an object. When executing a `for` loop, the sequence/collection over which to iterate must be an iterable. During the execution of the `for` loop, an iterator is created and the `next` function is executed until there is no next element. An iterator is also an iterable, so can be used in a `for` loop as well. Lists, tuples, and strings are so-called *sequence* objects and are iterables, where the elements are iterated by their index.

The most common iterator in Python is the *range* iterator, which allows iteration over a range of indices. Note that `range` returns a *range* object, not a list.

```
for i in range(4,20):
    print(i, end=' ')
print(range(4,20))
```

4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
range(4,20)



Similar to Python's slice operator `[i : j]`, the iterator `range(i, j)` ranges from *i* to *j*, *not including* the index *j*.

Two other common iterables are sets and dictionaries. Python *sets* are, as in mathematics, unordered collections of unique objects. Sets are defined with curly brackets `{ }`, as opposed to round brackets `()` for tuples, and square brackets `[]` for lists. Unlike lists, sets do not have duplicate elements. Many of the usual set operations are implemented in Python, including the union `A | B` and intersection `A & B`.

```
A = {3, 2, 2, 4}
B = {4, 3, 1}
C = A & B
for i in A:
    print(i)
print(C)
```

2
3
4
{3, 4}

A useful way to construct lists is by *list comprehension*; that is, by expressions of the form

`<expression> for <element> in <list> if <condition>`

For sets a similar construction holds. In this way, lists and sets can be defined using very similar syntax as in mathematics. Compare, for example, the mathematical definition of the

sets $A := \{3, 2, 4, 2\} = \{2, 3, 4\}$ (no order and no duplication of elements) and $B := \{x^2 : x \in A\}$ with the Python code below.

```
setA = {3, 2, 4, 2}
setB = {x**2 for x in setA}
print(setB)
listA = [3, 2, 4, 2]
listB = [x**2 for x in listA]
print(listB)
```

```
{16, 9, 4}
[9, 4, 16, 4]
```

A *dictionary* is a set-like data structure, containing one or more **key:value** pairs enclosed in curly brackets. The keys are often of the same type, but do not have to be; the same holds for the values. Here is a simple example, storing the ages of Lord of the Rings characters in a dictionary.

```
DICT = {'Gimly': 140, 'Frodo': 51, 'Aragorn': 88}
for key in DICT:
    print(key, DICT[key])
```

```
Gimly 140
Frodo 51
Aragorn 88
```

D.8 Classes

Recall that objects are of fundamental importance in Python — indeed, data types and functions are all objects. A *class* is an object type, and writing a class definition can be thought of as creating a template for a new type of object. Each class contains a number of attributes, including a number of inbuilt methods. The basic syntax for the creation of a class is:

```
class <class_name>:
    def __init__(self):
        <statements>
    <statements>
```

The main inbuilt method is `__init__`, which creates an *instance* of a class object. For example, `str` is a class object (string class), but `s = str('Hello')` or simply `s = 'Hello'`, creates an instance, `s`, of the `str` class. Instance attributes are created during initialization

and their values may be different for different instances. In contrast, the values of class attributes are the same for every instance. The variable `self` in the initialization method refers to the current instance that is being created. Here is a simple example, explaining how attributes are assigned.

```
class shire_person:
    def __init__(self, name): # initialization method
        self.name = name     # instance attribute
        self.age = 0         # instance attribute
        address = 'The Shire' # class attribute

print(dir(shire_person)[1:5], '...', dir(shire_person)[-2:])
# list of class attributes

p1 = shire_person('Sam')    # create an instance
p2 = shire_person('Frodo') # create another instance
print(p1.__dict__)          # list of instance attributes

p2.race = 'Hobbit'         # add another attribute to instance p2
p2.age = 33                # change instance attribute
print(p2.__dict__)

print(getattr(p1, 'address')) # content of p1's class attribute

['__delattr__', '__dict__', '__dir__', '__doc__', '...']
['__weakref__', 'address']
{'name': 'Sam', 'age': 0}
{'name': 'Frodo', 'age': 33, 'race': 'Hobbit'}
The Shire
```

It is good practice to create all the attributes of the class object in the `__init__` method, but, as seen in the example above, attributes can be created and assigned everywhere, even outside the class definition. More generally, attributes can be added to any object that has a `__dict__`.



An “empty” class can be created via

```
class <class_name>:
    pass
```

Python classes can be derived from a parent class by *inheritance*, via the following syntax.

```
class <class_name>(<parent_class_name>):
    <statements>
```

The derived class (initially) inherits all of the attributes of the parent class.

As an example, the class `shire_person` below inherits the attributes `name`, `age`, and `address` from its parent class `person`. This is done using the `super` function, used here to refer to the parent class `person` without naming it explicitly. When creating a new object of type `shire_person`, the `__init__` method of the parent class is invoked, and an additional instance attribute `Shire_address` is created. The `dir` function confirms that `Shire_address` is an attribute only of `shire_person` instances.

```
class person:
    def __init__(self, name):
        self.name = name
        self.age = 0
        self.address = ' '

class shire_person(person):
    def __init__(self, name):
        super().__init__(name)
        self.Shire_address = 'Bag End'

p1 = shire_person("Frodo")
p2 = person("Gandalf")
print(dir(p1)[:1], dir(p1)[-3:])
print(dir(p2)[:1], dir(p2)[-3:])
```

```
['Shire_address'] ['address', 'age', 'name']
['__class__'] ['address', 'age', 'name']
```

D.9 Files

To write to or read from a file, a file first needs to be opened. The `open` function in Python creates a file object that is iterable, and thus can be processed in a sequential manner in a `for` or `while` loop. Here is a simple example.

```
fout = open('output.txt', 'w')
for i in range(0, 41):
    if i%10 == 0:
        fout.write('{:3d}\n'.format(i))
fout.close()
```

The first argument of `open` is the name of the file. The second argument specifies if the file is opened for reading (`'r'`), writing (`'w'`), appending (`'a'`), and so on. See `help(open)`. Files are written in text mode by default, but it is also possible to write in binary mode. The above program creates a file `output.txt` with 5 lines, containing the strings 0, 10, ..., 40. Note that if we had written `fout.write(i)` in the fourth line of the code above, an error

message would be produced, as the variable `i` is an integer, and not a string. Recall that the expression `string.format()` is Python's way to specify the format of the output string.

The formatting syntax `{:3d}` indicates that the output should be constrained to a specific width of three characters, each of which is a decimal value. As mentioned in the introduction, bracket syntax `{i}` provides a placeholder for the `i`-th variable to be printed, with 0 being the first index. The format for the output is further specified by `{i:format}`, where `format` is typically⁴ of the form:

`[width][.precision][type]`

In this specification:

- `width` specifies the minimum width of output;
- `precision` specifies the number of digits to be displayed after the decimal point for a floating point values of type `f`, or the number of digits before *and* after the decimal point for a floating point values of type `g`;
- `type` specifies the type of output. The most common types are `s` for strings, `d` for integers, `b` for binary numbers, `f` for floating point numbers (floats) in fixed-point notation, `g` for floats in general notation, `e` for floats in scientific notation.

The following illustrates some behavior of formatting on numbers.

```
{:5d}'.format(123)
'{:.4e}'.format(1234567890)
'{:.2f}'.format(1234567890)
'{:.2f}'.format(2.718281828)
'{:.3f}'.format(2.718281828)
'{:.3g}'.format(2.718281828)
'{:.3e}'.format(2.718281828)
'{0:3.3f}; {2:~.4e};'.format(123.456789, 0.00123456789)
```

```
' 123 '
'1.2346e+09'
'1234567890.00 '
'2.72 '
'2.718 '
'2.72 '
'2.718e+00 '
'123.457; 1.2346e-03; '
```

The following code reads the text file `output.txt` line by line, and prints the output on the screen. To remove the newline `\n` character, we have used the `strip` method for strings, which removes any whitespace from the start and end of a string.

⁴More formatting options are possible.


```
fin = open('output.txt','r')
for line in fin:
    line = line.strip()    # strips a newline character
    print(line)
fin.close()
```

```
0
10
20
30
40
```

When dealing with file input and output it is important to always close files. Files that remain open, e.g., when a program finishes unexpectedly due to a programming error, can cause considerable system problems. For this reason it is recommended to open files via *context management*. The syntax is as follows.

```
with open('output.txt', 'w') as f:
    f.write('Hi there!')
```

Context management ensures that a file is correctly closed even when the program is terminated prematurely. An example is given in the next program, which outputs the most-frequent words in Dicken's *A Tale of Two Cities*, which can be downloaded from the book's GitHub site as [ataleof2cities.txt](#).

Note that in the next program, the file `ataleof2cities.txt` must be placed in the current working directory. The current working directory can be determined via `import os` followed by `cwd = os.getcwd()`.

```
numline = 0
DICT = {}
with open('ataleof2cities.txt', encoding="utf8") as fin:
    for line in fin:
        words = line.split()
        for w in words:
            if w not in DICT:
                DICT[w] = 1
            else:
                DICT[w] += 1
        numline += 1

sd = sorted(DICT, key=DICT.get, reverse=True) #sort the dictionary

print("Number of unique words: {}".format(len(DICT)))
print("Ten most frequent words:\n")
print("{:8} {}".format("word", "count"))
print(15*'-')
```

```
for i in range(0,10):
    print("{:8} {}".format(sd[i], DICT[sd[i]]))
```

Number of unique words: 19091

Ten most frequent words:

word	count
the	7348
and	4679
of	3949
to	3387
a	2768
in	2390
his	1911
was	1672
that	1650
I	1444

D.10 NumPy

The package NumPy (module name **numpy**) provides the building blocks for scientific computing in Python. It contains all the standard mathematical functions, such as **sin**, **cos**, **tan**, etc., as well as efficient functions for random number generation, linear algebra, and statistical computation.

```
import numpy as np    #import the package
x = np.cos(1)
data = [1,2,3,4,5]
y = np.mean(data)
z = np.std(data)
print('cos(1) = {:1.8f}   mean = {1}   std = {2}'.format(x,y,z))
```

```
cos(1) = 0.54030231   mean = 3.0   std = 1.4142135623730951
```

D.10.1 Creating and Shaping Arrays

The fundamental data type in **numpy** is the **ndarray**. This data type allows for fast matrix operations via highly optimized numerical libraries such as LAPACK and BLAS; this in contrast to (nested) lists. As such, **numpy** is often essential when dealing with large amounts of quantitative data.

ndarray objects can be created in various ways. The following code creates a $2 \times 3 \times 2$ array of zeros. Think of it as a 3-dimensional matrix or two stacked 3×2 matrices.

```
A = np.zeros([2,3,2]) # 2 by 3 by 2 array of zeros
print(A)
print(A.shape)      # number of rows and columns
print(type(A))      # A is an ndarray
```

```
[[[ 0.  0.]
   [ 0.  0.]
   [ 0.  0.]]

 [[ 0.  0.]
   [ 0.  0.]
   [ 0.  0.]]]
(2, 3, 2)
<class 'numpy.ndarray'>
```

We will be mostly working with 2D arrays; that is, ndarrays that represent ordinary matrices. We can also use the `range` method and lists to create ndarrays via the `array` method. Note that `arange` is `numpy`'s version of `range`, with the difference that `arange` returns an ndarray object.

```
a = np.array(range(4))      # equivalent to np.arange(4)
b = np.array([0,1,2,3])
C = np.array([[1,2,3],[3,2,1]])
print(a, '\n', b, '\n', C)
```

```
[0 1 2 3]
[0 1 2 3]
[[1 2 3]
 [3 2 1]]
```

The dimension of an ndarray can be obtained via its `shape` method, which returns a tuple. Arrays can be reshaped via the `reshape` method. This does not change the current ndarray object. To make the change permanent, a new instance needs to be created.

```
a = np.array(range(9)) #a is an ndarray of shape (9,)
print(a.shape)
A = a.reshape(3,3)     #A is an ndarray of shape (3,3)
print(a)
print(A)
```

```
[0 1 2 3 4 5 6 7 8]
(9,)
[[0, 1, 2]
 [3, 4, 5]
 [6, 7, 8]]
```



One shape dimension for **reshape** can be specified as -1 . The dimension is then inferred from the other dimension(s).

The 'T' attribute of an ndarray gives its transpose. Note that the transpose of a “vector” with shape $(n,)$ is the same vector. To distinguish between column and row vectors, reshape such a vector to an $n \times 1$ and $1 \times n$ array, respectively.

```
a = np.arange(3)    #1D array (vector) of shape (3,)
print(a)
print(a.shape)
b = a.reshape(-1,1) # 3x1 array (matrix) of shape (3,1)
print(b)
print(b.T)
A = np.arange(9).reshape(3,3)
print(A.T)
```

```
[0 1 2]
(3,)
[[0]
 [1]
 [2]]
[[0 1 2]]
[[0 3 6]
 [1 4 7]
 [2 5 8]]
```

Two useful methods of joining arrays are **hstack** and **vstack**, where the arrays are joined horizontally and vertically, respectively.

```
A = np.ones((3,3))
B = np.zeros((3,2))
C = np.hstack((A,B))
print(C)
```

```
[[ 1.  1.  1.  0.  0.]
 [ 1.  1.  1.  0.  0.]
 [ 1.  1.  1.  0.  0.]]
```

D.10.2 Slicing

Arrays can be sliced similarly to Python lists. If an array has several dimensions, a slice for each dimension needs to be specified. Recall that Python indexing starts at '0' and ends at 'len(obj)-1'. The following program illustrates various slicing operations.

```
A = np.array(range(9)).reshape(3,3)
print(A)
print(A[0])      # first row
print(A[:,1])    # second column
print(A[0,1])    # element in first row and second column
print(A[0:1,1:2]) # (1,1) ndarray containing A[0,1] = 1
print(A[1:,-1])  # elements in 2nd and 3rd rows, and last column
```

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
[0 1 2]
[1 4 7]
1
[[1]]
[5 8]
```

Note that ndarrays are mutable objects, so that elements can be modified directly, without having to create a new object.

```
A[1:,1] = [0,0] # change two elements in the matrix A above
print(A)
```

```
[[0, 1, 2]
 [3, 0, 5]
 [6, 0, 8]]
```

D.10.3 Array Operations

Basic mathematical operators and functions act *element-wise* on ndarray objects.

```
x = np.array([[2,4],[6,8]])
y = np.array([[1,1],[2,2]])
print(x+y)
```

```
[[ 3,  5]
 [ 8, 10]]
```

```
print(np.divide(x,y)) # same as x/y
```

```
[[ 2.  4.]
 [ 3.  4.]]
```

```
print(np.sqrt(x))
```

```
[[1.41421356  2.          ]
 [2.44948974  2.82842712]]
```

In order to compute matrix multiplications and compute inner products of vectors, `numpy`'s `dot` function can be used, either as a method of an `ndarray` instance or as a method of `np`.

```
print(np.dot(x,y))
```

```
[[10, 10]
 [22, 22]]
```

```
print(x.dot(x))    # same as np.dot(x,x)
```

```
[[28, 40]
 [60, 88]]
```

Since version 3.5 of Python, it is possible to multiply two `ndarrays` using the `@ operator` (which implements the `np.matmul` method). For matrices, this is similar to using the `dot` method. For higher-dimensional arrays the two methods behave differently.

```
print(x @ y)
```

```
[[10 10]
 [22 22]]
```

NumPy allows arithmetic operations on arrays of different shapes (dimensions). Specifically, suppose two arrays have dimensions (m_1, m_2, \dots, m_p) and (n_1, n_2, \dots, n_p) , respectively. The arrays or shapes are said to be *aligned* if for all $i = 1, \dots, p$ it holds that

- $m_i = n_i$, or
- $\min\{m_i, n_i\} = 1$, or
- either m_i or n_i , or both are missing.

For example, shapes $(1, 2, 3)$ and $(4, 2, 1)$ are aligned, as are $(2, ,)$ and $(1, 2, 3)$. However, $(2, 2, 2)$ and $(1, 2, 3)$ are not aligned. NumPy “duplicates” the array elements across the smaller dimension to match the larger dimension. This process is called *broadcasting* and is carried out without actually making copies, thus providing efficient memory use. Below are some examples.

```
import numpy as np
A= np.arange(4).reshape(2,2) # (2,2) array

x1 = np.array([40,500])      # (2,) array
x2 = x1.reshape(2,1)        # (2,1) array
```

```
print(A + x1) # shapes (2,2) and (2,)
print(A * x2) # shapes (2,2) and (2,1)
```

```
[[ 40 501]
 [ 42 503]]
[[   0   40]
 [1000 1500]]
```

Note that above `x1` is duplicated row-wise and `x2` column-wise. Broadcasting also applies to the matrix-wise operator `@`, as illustrated below. Here, the matrix `b` is duplicated across the third dimension resulting in the two matrix multiplications

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 4 & 5 \\ 6 & 7 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}.$$

```
B = np.arange(8).reshape(2,2,2)
b = np.arange(4).reshape(2,2)
print(B@b)
```

```
[[[ 2  3]
   [ 6 11]]

 [[10 19]
  [14 27]]]
```

Functions such as `sum`, `mean`, and `std` can also be executed as methods of an `ndarray` instance. The argument `axis` can be passed to specify along which dimension the function is applied. By default `axis=None`.

```
a = np.array(range(4)).reshape(2,2)
print(a.sum(axis=0)) #summing over rows gives column totals
```

```
[2, 4]
```

D.10.4 Random Numbers

One of the sub-modules in `numpy` is `random`. It contains many functions for random variable generation.

```
import numpy as np
np.random.seed(123) # set the seed for the random number generator
x = np.random.random() # uniform (0,1)
```

```
y = np.random.randint(5,9) # discrete uniform 5,...,8
z = np.random.randn(4)     # array of four standard normals
print(x,y, '\n', z)
```

```
0.6964691855978616 7
[ 1.77399501 -0.66475792 -0.07351368  1.81403277]
```

For more information on random variable generation in **numpy**, see

<https://docs.scipy.org/doc/numpy/reference/random/index.html>.

D.11 Matplotlib

The main Python graphics library for 2D and 3D plotting is **matplotlib**, and its subpackage **pyplot** contains a collection of functions that make plotting in Python similar to that in MATLAB.

D.11.1 Creating a Basic Plot

The code below illustrates various possibilities for creating plots. The style and color of lines and markers can be changed, as well as the font size of the labels. Figure D.1 shows the result.

sqrtplot.py

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0, 10, 0.1)
u = np.arange(0,10)
y = np.sqrt(x)
v = u/3
plt.figure(figsize = [4,2]) # size of plot in inches
plt.plot(x,y, 'g--')        # plot green dashed line
plt.plot(u,v,'r.')          # plot red dots
plt.xlabel('x')
plt.ylabel('y')
plt.tight_layout()
plt.savefig('sqrtplot.pdf',format='pdf') # saving as pdf
plt.show()                  # both plots will now be drawn
```

The library **matplotlib** also allows the creation of subplots. The scatterplot and histogram in Figure D.2 have been produced using the code below. When creating a histogram there are several optional arguments that affect the layout of the graph. The number of bins is

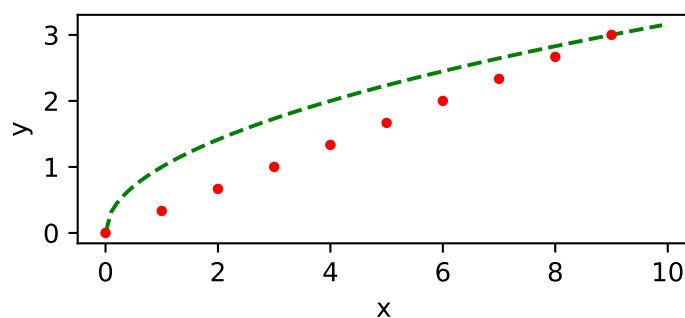


Figure D.1: A simple plot created using pyplot.

determined by the parameter `bins` (the default is 10). Scatterplots also take a number of parameters, such as a string `c` which determines the color of the dots, and `alpha` which affects the transparency of the dots.

histscat.py

```
import matplotlib.pyplot as plt
import numpy as np
x = np.random.randn(1000)
u = np.random.randn(100)
v = np.random.randn(100)
plt.subplot(121)          # first subplot
plt.hist(x, bins=25, facecolor='b')
plt.xlabel('X Variable')
plt.ylabel('Counts')
plt.subplot(122)          # second subplot
plt.scatter(u, v, c='b', alpha=0.5)
plt.show()
```

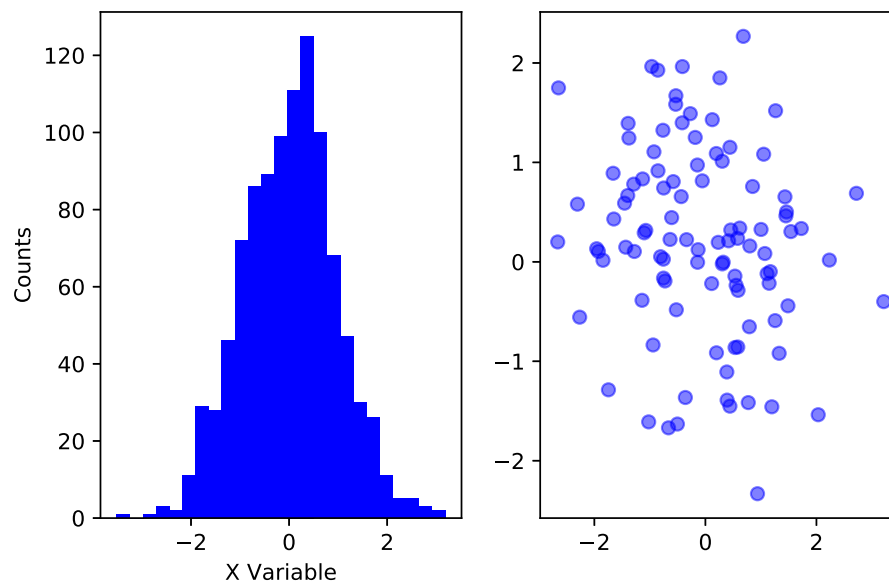


Figure D.2: A histogram and scatterplot.

One can also create three-dimensional plots as illustrated below.

surf3dscat.py

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

def npdf(x,y):
    return np.exp(-0.5*(pow(x,2)+pow(y,2)))/np.sqrt(2*np.pi)

x, y = np.random.randn(100), np.random.randn(100)
z = npdf(x,y)

xgrid, ygrid = np.linspace(-3,3,100), np.linspace(-3,3,100)

Xarray, Yarray = np.meshgrid(xgrid,ygrid)
Zarray = npdf(Xarray,Yarray)

fig = plt.figure(figsize=plt.figaspect(0.4))
ax1 = fig.add_subplot(121, projection='3d')
ax1.scatter(x,y,z, c='g')
ax1.set_xlabel('$x$')
ax1.set_ylabel('$y$')
ax1.set_zlabel('$f(x,y)$')

ax2 = fig.add_subplot(122, projection='3d')
```

```
ax2.plot_surface(Xarray, Yarray, Zarray, cmap='viridis',
                 edgecolor='none')
ax2.set_xlabel('$x$')
ax2.set_ylabel('$y$')
ax2.set_zlabel('$f(x,y)$')

plt.show()
```

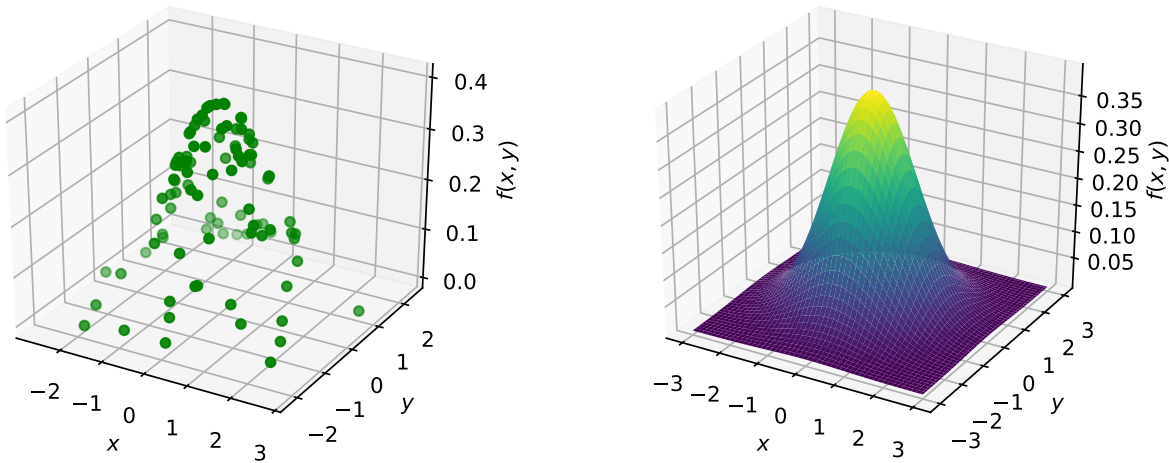


Figure D.3: Three-dimensional scatter- and surface plots.

D.12 System Calls, URL Access, and Speed-Up

Operating system commands (whether in Windows, MacOS, or Linux) for creating directories, copying or removing files, or executing programs from the system shell can be issued from within Python by using the package **os**. Another useful package is **requests** which enables direct downloads of files and webpages from URLs. The following Python script uses both. It also illustrates a simple example of exception handling in Python.

misc.py

```
import os
import requests
for c in "123456":
    try:
        os.mkdir("MyDir"+ c) # if it does not yet exist
    except:                  # make a directory
        pass                # otherwise
                            # do nothing
```

```

uname = "https://github.com/DSML-book/Programs/tree/master/Appendices/
        Python Primer/"
fname = "ataleof2cities.txt"
r = requests.get(uname + fname)
print(r.text)
open('MyDir1/ato2c.txt', 'wb').write(r.content) #write to a file
                                                # bytes mode is important here

```

The package **numba** can significantly speed up calculations via smart compilation. First run the following code.

jitex.py

```

import timeit
import numpy as np
from numba import jit
n = 10**8

#@jit
def myfun(s,n):
    for i in range(1,n):
        s = s+ 1/i
    return s

start = timeit.time.clock()
print("Euler's constant is approximately {:.8f}".format(
        myfun(0,n) - np.log(n)))
end = timeit.time.clock()
print("elapsed time: {:.32f} seconds".format(end-start))

```

```

Euler's constant is approximately 0.57721566
elapsed time: 5.72 seconds

```

Now remove the # character before the @ character in the code above, in order to activate the “just in time” compiler. This gives a 15-fold speedup:

```

Euler's constant is approximately 0.57721566
elapsed time: 0.39 seconds

```

INDEX

- acceptance–rejection method, 66
- aligned arrays (Python), 182
- anaconda (Python), 161
- assignment operator (Python), 165
- attributes (Python), 164
- Bayes’ rule, 32
- Bernoulli
 - distribution, 69
- Bernoulli distribution, 48
- binomial
 - distribution
 - generation, 70
- binomial distribution, 33, 48
- birthday problem, 30
- bivariate normal distribution, 102
- Box–Muller method, 69
- broadcasting (Python), 182
- Central Limit Theorem (CLT), 109
- chain rule, 29
- chi-square distribution, 59, 94
- class (Python), 173
- coin flip experiment, 33, 36, 48, 76
- coin tossing, 14
- conditional
 - expectation, 90
 - pdf/pmf, 88
- conditional probability, 27
- context management (Python), 177
- convolution, 97
- correlation coefficient, 85
- covariance, 83
 - properties, 83
- covariance matrix, 87, 98, 101
- cumulative distribution function (cdf), 38
- De Morgan, 16
- dictionary (Python), 173
- disjoint events, 14
- disjoint events, 16
- distribution
 - Bernoulli, 69
 - binomial, 70
 - discrete, 39, 64
 - exponential, 68, 83
 - gamma, 83
 - geometric, 71
 - normal, 69

- uniform, 62
- dot notation (Python), 164
- event, 13
 - elementary, 18
- expectation, 42, 43, 81
 - properties, 45, 82
- expectation vector, 87, 98, 101
- exponential
 - distribution, 83
 - generation, 68
- exponential distribution, 55
- function (Python), 167
- gamma
 - distribution, 83
- gamma distribution, 59
- Gaussian distribution, 57
- geometric distribution, 33, 50
 - generation, 71
- global balance equations, 122
- halfnormal, 69
- hypergeometric distribution, 53
- immutable (Python), 163
- independence
 - of events, 32
 - of random variables, 75, 77, 79
- inheritance (Python), 174
- initial distribution, 114, 116
- instance (Python), 173
- inverse-transform method, 63, 68
- iterable (Python), 171
- iterator (Python), 171
- Jacobi
 - matrix of $-$, 99
- joint cdf, 74
- joint distribution, 74, 98
- joint pdf, 78
- joint pmf, 75
- jointly normal distribution, 101
- Law of Large Numbers (LLN), 108
- law of total probability, 31
- limiting distribution
 - of Markov chain, 120
- linear congruential generator, 62
- linear transformation, 97
- list comprehension (Python), 172
- marginal distribution, 75, 79
- Markov
 - chain
 - generation, 118
 - limiting behaviour, 120
 - inequality, 108
- Markov chain, 114
- matplotlib (Python), 184–187
- matrix multiplication (Python), 182
- method (Python), 165
- model
 - matrix, 104, 106
 - multiple linear regression, 106
- module (Python), 168
- moment, 45
- moment generating function, 47
- Monty Hall problem, 28
- multinomial distribution, 77
- multiple linear regression, 106
- multiplicative congruential generator, 62
- mutable (Python), 162
- namespace (Python), 169
- normal
 - distribution
 - generation, 69
- normal distribution, 57, 81, 103
- object (Python), 162
- operator (Python), 163
- overloading (Python), 167
- partition, 31
- Poisson distribution, 52
- probability (measure), 16
- probability density function (pdf), 40

- probability distribution, 37
- probability mass function (pmf), 39
- product rule, 114
- proposal pdf, 66
- pseudorandom number, 62
- random
 - number generation, 61
 - numbers (Python), 183
 - walk, 115, 121
- random experiment, 7
- random sample, 80
- random variable, 35
 - continuous, 37, 40
 - discrete, 37
- random vector, 74, 97
- random walk, 119
- range (Python), 172
- reference (Python), 166
- regression
 - line, 106
- sample space, 12
 - discrete, 18
- sampling distribution, 80
- seed, 62
- sequence object (Python), 172
- set (Python), 172
- simple linear regression, 106
- slice (Python), 163
- standard deviation, 45
- standard normal distribution, 57
- standardisation, 101
- state space, 114
- sum rule, 16
- target pdf, 66
- transformation rule, 99
- transition
 - graph, 115
 - matrix, 114
 - n -step, 116
 - probability, 114
- type (Python), 165
- uniform
 - distribution, 62
- uniform distribution, 54, 80
- variance, 45
 - properties, 83
- Venn diagram, 14