Bayesian Time-Series Econometrics

Book 3 - mathematical background

Romain Legrand

Third edition

Bayesian Time-Series Econometrics

Cover illustration: Thomas Bayes (d. 1761) in Terence O'Donnell, History of Life Insurance in Its Formative Years (Chicago: American Conservation Co:, 1936), p. 335.

To my wife, Mélanie.

To my sons, Tristan and Arnaud.

Contents

CHAPTER S

Set theory

s.1 Elementary concepts

In mathematics, sets are used to describe groups of objects. Formally:

definition s.1: a **set** is a collection of elements.

The elements in question can be anything, but typically they are mathematical objects such as numbers or symbols. To denote a set, it is customary to list its elements within curly brackets `{}`.

example s.1: the set A containing the numbers 1, 2 and 3 is denoted by $A = \{1,2,3\}$.

Sets can be defined in a more systematic way by describing their elements. This is done by using indifferently the notations | or : which stand for "such that".

example s.2: the set containing the numbers that are smaller than 3 can be denoted by $A = \{x|x < 3\}$ or $A = \{x : x < 3\}$.

To indicate that some element x is a member of set A, one writes $x \in A$, which reads "x belongs to A". Conversely, to denote the fact that x is not a member of A, one writes $x \notin A$.

example s.3: if A={1,2,3}, then $2 \in A$, but $5 \notin A$.

Two fundamental concepts in set theory are that of subsets and supersets:

definition s.2: given two sets A and B, A is a **subset** of B, denoted by $A \subseteq B$, if every member of A is also a member of B. If A is a subset of B, then B is a **superset** of A.

For instance:

example s.4: let $A = \{2,4\}$ and $B = \{1,2,3,4,5\}$. Then $A \subseteq B$. A is a subset of B, and B is a superset of A.

example s.5: let $A = \{x : x < 3\}$, the set of numbers smaller than 3, and let $B = \{x : x < 5\}$, the set of numbers smaller than 5. Then $A \subseteq B$. A is a subset of B, and B is a superset of A.

example s.6: let $A = \{1,2,3\}$ and $B = \{1,2,3\}$, so that $A = B$. Then $A \subseteq B$ and $B \subseteq A$. A and B are at the same time subset and superset of each other. Thus, subsets and supersets include the case of equal sets.

There exist two sets of special interest, called the empty set and the universal set.

definition s.3: the **empty set** is the set that contains no element. It is denoted by $\varnothing$.

At the other end of the spectrum, the universal set is defined as:

definition s.4: the **universal set** is the set that contains all possible elements, in a given context.

Any set we might consider is a subset of the universal set. The empty set and the universal sets are also important for the definition of the notion of complement. The latter is defined as:

definition s.5: If A is some set, then the **complement** of A, denoted by A^c, is the set containing all the elements of the universal set that are not in A. Formally, if A is some set, and X denotes the universal set, then $A^c = \{x \in X : x \notin A\}$.

For example:

example s.7: if we let the universal set be $X = \{1,2,3,4,5,6,7,8\}$, and A be the set $A = \{2,4,5\}$ then the complement of A is $A^c = \{1,3,6,7,8\}$.

example s.8: let $A = \{x : x < 3\}$, the set of numbers smaller than 3. Then if the universal set X is the set of all numbers, $A^c = \{x : x \geq 3\}$, the set of numbers greater than or equal to 3.

In examples s.7 and s.8, the universal set was explicitly described. Most of the time however the universal set is only implicit and used as an underlying element defining the complement of a set A as "everything that is not in A".

s.2 Unions and intersections

Operations on sets are realised through the concepts of unions and intersections. Set unions are defined as follows:

definition s.6: let A and B be two sets; the **union** of A and B, denoted by $A \cup B$, is the set of all elements that are either in A or in B (or in both). Formally: $A \cup B = \{x : x \in A \text{ or } x \in B\}$.

For example:

example s.9: let $A = \{1,2,3\}$ and $B = \{3,4,5\}$; then $A \cup B = \{1,2,3,4,5\}$.

example s.10: let $A = \{x : 2 < x < 6\}$, the set of numbers comprised between 2 and 6, and let $B = \{x : 4 < x < 8\}$, the set of numbers comprised between 4 and 8. Then $A \cup B = \{x : 2 < x < 8\}$, the set of numbers comprised between 2 and 8.

The counterpart of the concept of union is that intersection. Set intersection is defined as follows:

definition s.7: let A and B be two sets; the **intersection** of A and B, denoted by $A \cap B$, is the set of all elements that are both in A and in B. Formally: $A \cap B = \{x : x \in A \text{ and } x \in B\}$.

For example:

example s.11: let $A = \{1,2,3\}$ and $B = \{3,4,5\}$; then $A \cap B = \{3\}$.

example s.12: let $A = \{x : 2 < x < 6\}$, the set of numbers comprised between 2 and 6, and let $B = \{x : 4 < x < 8\}$, the set of numbers comprised between 4 and 8. Then $A \cap B = \{x : 4 < x < 6\}$, the set of numbers comprised between 4 and 6.

Two sets which have no elements in common are called disjoint sets:

definition s.8: two sets A and B are **disjoint** if they have no element in common, that is, if $A \cap B = \varnothing$.

For example:

example s.13: let $A = \{1,2,3\}$ and $B = \{4,5,6\}$; then $A \cap B = \varnothing$, so A and B are disjoint.

Notations for multiple unions and intersections can be used to avoid cumbersome writing.

definition s.9: let $A_1, A_2, A_3, \ldots, A_n$ be some sets. Then the **multiple union** of those sets is denoted by: $\bigcup_{i=1}^{n} A_i = A_1 \cup A_2 \cup A_3 \cup \ldots \cup A_n$.

Also:

definition s.10: let $A_1, A_2, A_3, \ldots, A_n$ be some sets. Then the **multiple intersection** of those sets is denoted by: $\bigcap_{i=1}^{n} A_i = A_1 \cap A_2 \cap A_3 \cap \ldots \cap A_n$.

For example:

example s.14: let $A_1 = \{1,2,3,4,5\}$, $A_2 = \{2,3,4,5,6\}$ and $A_3 = \{3,4,5,6,7\}$; then:
$\bigcup_{i=1}^{3} A_i = \{1,2,3,4,5,6,7\}$ and $\bigcap_{i=1}^{3} A_i = \{3,4,5\}$.

example s.15: let $A_1 = \{x : 2 < x < 6\}$, $A_2 = \{x : 3 < x < 7\}$ and $A_3 = \{x : 4 < x < 8\}$; then:
$\bigcup_{i=1}^{3} A_i = \{x : 2 < x < 8\}$ and $\bigcap_{i=1}^{3} A_i = \{x : 4 < x < 6\}$.

s.3 Countable and uncountable sets

The notion of countability plays an important role in statistical theory, in particular when discussing random variables. Indeed, it is countability which determines the nature of random variables, discrete or continuous. Countable sets of outcomes produce discrete random variables, while uncountable sets of outcome result in continuous random variables. Before discussing this concept formally, it is useful to introduce some very famous sets.

definition s.11: the set of **natural numbers**, denoted by $\mathbb{N}$, is the set of positive whole numbers (or counting numbers). That is, $\mathbb{N} = \{1,2,3,\ldots\}$.

Some textbooks also include 0 in the natural numbers. Most commonly however 0 is excluded, and this choice is retained here.

A natural extension of the natural numbers is the set of integer numbers:

definition s.12: the set of **integers**, denoted by $\mathbb{Z}$, is the set of all whole numbers, positive, negative and zero. That is, $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$.

The set of integers provides a convenient way to denote the set of natural numbers plus zero:

definition s.13: the set of **non-negative integers**, denoted by $\mathbb{Z}^*$, is the set of all non-negative whole numbers. That is, $\mathbb{Z}^* = \{0, 1, 2, 3, \ldots\}$.

It is clear that $\mathbb{Z}^* = 0 \cup \mathbb{N}$, and thus represents only a shortcut notation. With the set of integers, it is possible to define the set of rational numbers:

definition s.14: the set of **rational numbers** , denoted by $\mathbb{Q}$, is the set of all numbers which can be written as the quotient (or ratio) of two integers. That is, $\mathbb{Q} = \{\frac{x}{y} : x \in \mathbb{Z},\ y \in \mathbb{Z},\ y \neq 0\}$.

It may seem at first that the set of rational numbers can describe any possible number. But this is not true: certain numbers like $\sqrt{2}$ or π for instance cannot be written as the ratio of two integers, and are hence not rational numbers. This leads to the following definition:

definition s.15: an **irrational number** is a number which cannot be written as the ratio of two integers.

Irrational numbers are important because they provide the final element required to define the set of real numbers. Loosely speaking, one can see the set of real numbers as the set containing all numbers. The formal definition goes as follows:

definition s.16: the set of **real numbers**, denoted by $\mathbb{R}$, is the set of all rational and irrational numbers.

From the above definitions, it should be clear that the natural, integer, rational and real numbers represent nested sets of numbers, namely: $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}$.

As a preliminary to the incoming discussion on countability, it is also useful to introduce the notion of finiteness:

definition s.17: a set A is **finite** if there exists some natural number $n \in \mathbb{N}$ such that the cardinality (number of elements) of A is equal to n. A set which is not finite is **infinite** .

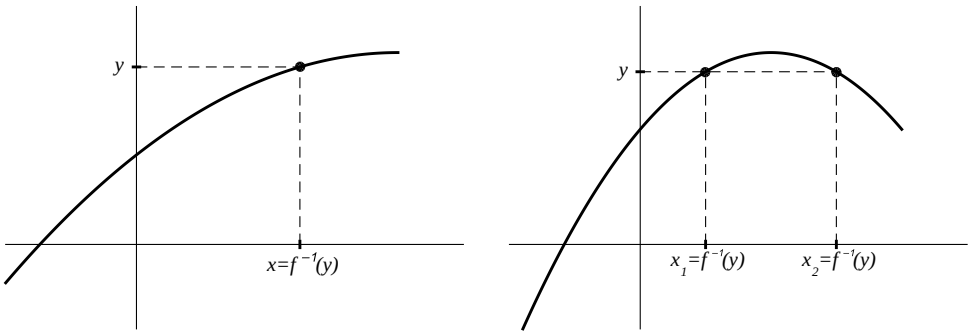
A trivial way to reformulate the above definition is to state that a set is finite if it contains a finite number of elements. Otherwise, it is infinite. For example:

example s.16: the set $A = \{5, 6, 7\}$ is finite, since its cardinality is 3 (it contains 3 elements); on the other hand, $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ and $\mathbb{R}$ are examples of infinite sets.

It is now possible to introduce the notion of countability of a set.

definition s.18: a set A is **countable** if there exists a bijection from A to $\mathbb{N}$ (or some subset of $\mathbb{N}$). A set which is not countable is said to be **uncountable** .

Let us clarify this definition. First, a bijection is a function such that to each value x of the domain corresponds a unique value $f(x)$ of the codomain, and vice versa such that to each value y of the codomain corresponds a unique value $f^{-1}(y)$ of the domain. Figure s.1 makes the point. The function displayed on panel (a) is a bijection since for each possible value y of the codomain corresponds a unique value $x = f^{-1}(y)$, and vice versa. The function displayed on panel (b) on the other hand is not a bijection since for a given value y of the codomain correspond two possible values $f^{-1}(y) = x_1$ and x_2.



(a) a bijective function (b) a non-bijective function

Figure s.1: bijective and non-bijective functions

It is now possible to go back to the definition of a countable set. Simply speaking, it says that a set is countable if its elements can be enumerated. In other words, a set A is countable if one can create a list of its elements, and assign to each of these elements a unique position in this list ("1st element of the list", "2nd element of the list", and so on).

The use of a bijective function simply represents a formal way to draw the list, the domain being the elements of A, and the codomain being the position in the list (1,2,3 and so on, hence the set $\mathbb{N}$ for the codomain). As the function is bijective, it guarantees that each element in the set is associated to a unique position in the list, and vice versa that each position in the list corresponds to a single element in A. It does not matter if the list is infinite. Infinite sets result in infinite lists, in which case the function spans the whole of $\mathbb{N}$. On the other hand, finite sets result in a finite list and hence span only some subset of $\mathbb{N}$.

With this definition, it is possible to discuss the countability of $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$, starting with the set of natural numbers $\mathbb{N}$:

property s.1: the set of natural numbers $\mathbb{N}$ is countable.

proof: to prove the result, it must be possible to create a list of the elements in $\mathbb{N}$, and assign to each of these elements a unique position in this list. In the case of the natural numbers, this is quite trivial since it amounts to creating a mapping from $\mathbb{N}$ to $\mathbb{N}$. The resulting list is displayed in Table s.1:

List of elements in $\mathbb{N}$ (domain of the injection: $\mathbb{N}$)	Position in the list (codomain of the injection: $\mathbb{N}$)
1	1
2	2
3	3
4	4
⋮	⋮

Table s.1: bijection for the countability of $\mathbb{N}$

Next, establish the countability of the set of integers $\mathbb{Z}$:

property s.2: the set of integer numbers $\mathbb{Z}$ is countable.

proof: again, to prove the result, one creates a list of the elements in $\mathbb{Z}$ and assign to each of these elements a unique position in this list. This is hardly more complicated than in the case of the natural numbers $\mathbb{N}$. Zero must be included in the list, and because $\mathbb{Z}$ also include the negative whole numbers, the enumeration must alternate between positive and negative values. The resulting list is displayed in Table s.2:

List of elements in $\mathbb{Z}$ (domain of the injection: $\mathbb{Z}$)	Position in the list (codomain of the injection: $\mathbb{N}$)
0	1
1	2
-1	3
2	4
-2	5
3	6
-3	7
⋮	⋮

Table s.2: bijection for the countability of $\mathbb{Z}$

Next, consider the set of rational numbers $\mathbb{Q}$:

property s.3: the set of rational numbers $\mathbb{Q}$ is countable.

proof: to prove the result, create a list of the elements in $\mathbb{Q}$, and assign to each of these elements a unique position in this list. This is a bit more complicated for $\mathbb{Q}$ than it is for $\mathbb{N}$ and $\mathbb{Z}$. The strategy consists in identifying all possible fractions $\frac{a}{b}$, with $a, b \in \mathbb{N}$ and then select those fractions in a systematic way to ensure all rational numbers are covered in the process. To do so, the following table is used:

	1		**2**		**3**		**4**		**5**	**...**
1	$\frac{1}{1}$		$\frac{1}{2}$	$\longrightarrow$	$\frac{1}{3}$		$\frac{1}{4}$	$\longrightarrow$	$\frac{1}{5}$	...
	$\downarrow$	$\nearrow$		$\swarrow$		$\nearrow$		$\swarrow$		
2	$\frac{2}{1}$		$\frac{2}{2}$		$\frac{2}{3}$		$\frac{2}{4}$		$\frac{2}{5}$	...
		$\swarrow$		$\nearrow$		$\swarrow$				
3	$\frac{3}{1}$		$\frac{3}{2}$		$\frac{3}{3}$		$\frac{3}{4}$		$\frac{3}{5}$	...
	$\downarrow$	$\nearrow$		$\swarrow$						
4	$\frac{4}{1}$		$\frac{4}{2}$		$\frac{4}{3}$		$\frac{4}{4}$		$\frac{4}{5}$	...
		$\swarrow$								
5	$\frac{5}{1}$		$\frac{5}{2}$		$\frac{5}{3}$		$\frac{5}{4}$		$\frac{5}{5}$	...
⋮	⋮		⋮		⋮		⋮		⋮	

Table s.3: ordered pairs (*a*,*b*) of natural numbers

It should be clear that this process records any possible positive rational number: $\frac{a}{b}$ will be found in row a, column b of the table. Following the arrow path then ensures that all the entries are covered at some point of the enumeration. What remains to do to complete the list of rational numbers $\mathbb{Q}$ is to include the entry $0 = \frac{0}{1}$, add the negative counterpart of each positive fraction, and get rid of the duplicates (for instance, $\frac{2}{2}$ and $\frac{1}{1}$ are the same number). This produces the following list of rational numbers:

List of elements in $\mathbb{Q}$ (domain of the injection: $\mathbb{Q}$)	Position in the list (codomain of the injection: $\mathbb{N}$)
$\frac{0}{1}$	1
$\frac{1}{1}$	2
$-\frac{1}{1}$	3
$\frac{2}{1}$	4
$-\frac{2}{1}$	5
$\frac{1}{2}$	6
$-\frac{1}{2}$	7
$\frac{1}{3}$	6
$-\frac{1}{3}$	7
$\frac{3}{1}$	8
$-\frac{3}{1}$	9
⋮	⋮

Table s.4: bijection for the countability of $\mathbb{Q}$

Consider finally $\mathbb{R}$, the set of real numbers. As stated previously, the real numbers extend the rational numbers by integrating both rational and irrational numbers. It may seem intuitively that most numbers can be written as rational numbers, so that irrational numbers represent an exception. In fact, the contrary is true: most numbers cannot be written as rational numbers, and there are considerably more real numbers than rational numbers. The real numbers are in fact so many that it is not possible to count them. This establishes the next result:

property s.4: the set of real numbers $\mathbb{R}$ is uncountable.

proof: the proof relies on the so-called Cantor diagonal argument. It proceeds by contradiction: it assumes that $\mathbb{R}$ is countable, and then shows that this assumption cannot be true. So, suppose it is possible to draw a list of all real numbers. Then the list of real numbers with integer part 0 would look like this:

List of elements in $\mathbb{R}$ (domain of the injection: $\mathbb{R}$)	Position in the list (codomain of the injection: $\mathbb{N}$)
$0\,.\,\boldsymbol{x_{11}}x_{12}x_{13}x_{14}\ldots$	1
$0\,.\,x_{21}\boldsymbol{x_{22}}x_{23}x_{24}\ldots$	2
$0\,.\,x_{31}x_{32}\boldsymbol{x_{33}}x_{34}\ldots$	3
$0\,.\,x_{41}x_{42}x_{43}\boldsymbol{x_{44}}\ldots$	4
$\vdots$	$\vdots$

Table s.5: bijection for the countability of $\mathbb{R}$ (assumption)

Now consider the real number $y = 0.y_1y_2y_3y_4\ldots$ constructed in the following way: y_1 is any digit except x_{11}, y_2 is any digit except x_{22}, and in general y_n is any digit except x_{nn} (the bold diagonal terms in Table s.5). Then clearly y is not equal to any number in the list since it has at least one digit that differs with each number. Therefore, the assumed list of real numbers cannot be complete, which results in a contradiction.

This concludes the discussion on the countability of the major sets of numbers. Some additional results on countability are now introduced.

property s.5: let A be some finite set; then A is countable.

proof: because A is finite, it contains n elements, for some $n \in \mathbb{N}$. Following, it is possible to associate to each of the n elements in A a unique natural number between 1 and n. This defines a bijection from A to a subset of $\mathbb{N}$, hence A is countable.

Though finiteness implies countability, finite and countable are not equivalent notions. Many infinite sets are countable, for instance $\mathbb{N}, \mathbb{Z}$ and $\mathbb{Q}$, as previously established. The next results discuss the countability of subsets.

property s.6: let A be some countable set; if $B \subseteq A$, then B is countable. In other words, the subset of a countable set is itself countable.

proof: only a sketch of the proof is provided. Because B is a subset of A, every element in B also lies in A. Also, because A is countable, there exists a bijection from A to $\mathbb{N}$. This means that for each element in A, there exists a unique associated natural number. Then for each element of B, consider the corresponding element in A, and the corresponding associated natural number from the bijection. Doing so, one defines a bijection from B to a subset of $\mathbb{N}$, hence B is countable.

To introduce the final result on subsets, it is necessary to define first the notions of closed and open intervals:

definition s.19: let a and b be two real numbers; then the **closed interval** $[a,b]$ is the set $[a,b] = \{x \in \mathbb{R} : a \leq x \leq b\}$, and the **open interval** (a,b) is the set $(a,b) = \{x \in \mathbb{R} : a < x < b\}$.

Roughly speaking, a closed interval is an interval which includes its endpoints, while an open interval excludes them. Intervals need not be fully closed or open, yet. One can also find the half-open (or half-closed) intervals $[a,b) = \{x \in \mathbb{R} : a \leq x < b\}$ and $(a,b] = \{x \in \mathbb{R} : a < x \leq b\}$. The final result of this chapter discusses the countability of intervals:

property s.7: let a and b be two real numbers with $a < b$; then the closed interval $[a,b]$, the open interval (a,b) and the half-open intervals $[a,b)$ and $(a,b]$ are uncountable.

proof: the easiest way to prove the above result is to rely, again, on the Cantor diagonal argument. First, one notes that for every interval $[a,b],(a,b),[a,b)$ or $(a,b]$ it is possible to define some subset (c,d) such that all elements in (c,d) share the same integer part z and the same first n decimals $d_1,d_2,\ldots,d_n$. In other words, $(c,d)=\{x\in\mathbb{R}: x=z\,.\,d_1d_2\ldots d_n<x<z\,.\,d_1d_2\ldots(d_n+1)\}$. For instance, for the closed interval $[a,b]=\{x\in\mathbb{R}: 2.34\leq x\leq 2.37\}$, it is possible to define $(c,d)=\{x\in\mathbb{R}: 2.35<x<2.36\}$, with integer part $z=2$, and decimal parts $d_1=3$ and $d_2=5$. The strategy then consists in using the Cantor diagonal argument on the sub-interval (c,d). Assume hence that (c,d) is countable, so that it is possible to draw a list of all real numbers on (c,d). This list would look like this:

List of elements in (c,d) (domain of the injection: (c,d))	Position in the list (codomain of the injection: $\mathbb{N}$)
$z\,.\,d_1d_2\ldots d_n\mathbf{x_{1(n+1)}}x_{1(n+2)}x_{1(n+3)}x_{1(n+4)}\cdots$	1
$z\,.\,d_1d_2\ldots d_nx_{2(n+1)}\mathbf{x_{2(n+2)}}x_{2(n+3)}x_{2(n+4)}\cdots$	2
$z\,.\,d_1d_2\ldots d_nx_{3(n+1)}x_{3(n+2)}\mathbf{x_{3(n+3)}}x_{3(n+4)}\cdots$	3
$z\,.\,d_1d_2\ldots d_nx_{4(n+1)}x_{4(n+2)}x_{4(n+3)}\mathbf{x_{4(n+4)}}\cdots$	4
$\vdots$	$\vdots$

Table s.6: bijection for the countability of (c,d) (assumption)

Now consider the real number $y=z.d_1d_2\ldots d_ny_{(n+1)}y_{(n+2)}y_{(n+3)}\ldots$ constructed in the following way: $y_{(n+1)}$ is any digit except $x_{1(n+1)}$, $y_{(n+2)}$ is any digit except $x_{2(n+2)}$, and in general $y_{(n+i)}$ is any digit except $x_{i(n+i)}$ (the bold diagonal terms in Table s.6). Then clearly y is not equal to any number in the list since it has at least one digit that differs with each number. Therefore, the assumed list of (c,d) cannot be complete, which results in a contradiction. Hence (c,d) is uncountable, so that the intervals $[a,b],(a,b),[a,b)$ and $(a,b]$ are also uncountable.

CHAPTER M

Matrix algebra

m.1 Elementary concepts

At the basis of matrix algebra lies a class of objects called matrices. A matrix is defined as follows:

definition m.1: a **matrix** is a rectangular array of numbers or symbols.

For example:

example m.1:

Let: $A = \begin{pmatrix} 3 & 0 & -2 \\ -1 & 2 & 3 \\ 0 & 3 & 1 \end{pmatrix}$ $B = \begin{pmatrix} 2 & 1 \\ 0 & 0 \\ -1 & 4 \\ 7 & 3 \end{pmatrix}$ $C = \begin{pmatrix} -5 & 0 & 3 \\ 1 & -3 & 0 \end{pmatrix}$

A, B and C are examples of matrices.

It is conventional to use regular brackets () to wrap the elements of a matrix, but sometimes square brackets [] and even curly brackets {} can also be used.

It is customary to describe a matrix by its dimension, namely its number of rows and columns. For a matrix with m rows and n columns, one uses the notation "$m \times n$", which reads "m by n".

example m.2:

Let: $A = \begin{pmatrix} 3 & 0 & -2 \\ -1 & 2 & 3 \\ 0 & 3 & 1 \end{pmatrix}$ $B = \begin{pmatrix} 2 & 1 \\ 0 & 0 \\ -1 & 4 \\ 7 & 3 \end{pmatrix}$ $C = \begin{pmatrix} -5 & 0 & 3 \\ 1 & -3 & 0 \end{pmatrix}$

A is a 3×4 matrix, B is a 4×2 matrix, while C is a 2×3 matrix.

It is useful to introduce some specific terminologies about matrix dimension.

definition m.2: a matrix of dimension $m \times 1$ is called a **column vector** ; a matrix of dimension $1 \times m$ is called a **row vector** ; a matrix of dimension 1×1 is called a **scalar** .

As the name suggests, a column vector is a matrix made of a single column, while a row vector is a matrix made of a single row. A scalar is simply an individual number, which is equivalent to a 1×1 matrix. By convention, the word "vector" is often used as a shortcut to designate a column vector. On the other hand, the full expression "row vector" is usually employed in order to avoid any ambiguity.

example m.3:

Let: $A = \begin{pmatrix} -1 & 3 & 0 \\ 4 & -2 & 6 \end{pmatrix}$ $b = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$ $c = \begin{pmatrix} 4 & 3 & -2 & 0 \end{pmatrix}$ $d = \begin{pmatrix} 2 \end{pmatrix}$.

A is a matrix, b is a (column) vector, c is a row vector, and d is a scalar.

As can be seen from example m.3, it is customary to use capital blocks to denote matrices, and lower blocks to denote vectors and scalars.

Sometimes, it is useful to designate specific elements in a matrix.

definition m.3: the (i, j) **entry** of a matrix is the number found in row i, column j of this matrix.

The convention to denote entries is to use the name of the matrix written in lower case, and associate to it the index (i, j) of the entry as a subscript. For instance, if A is a matrix, then the entry (i, j) of A will be written as a_{ij}.

example m.4:

Let: $A = \begin{pmatrix} 8 & 6 & 0 & 3 \\ 5 & 7 & -4 & 0 \\ 9 & 3 & -3 & -5 \end{pmatrix}$

Then $a_{12} = 6$, $a_{22} = 7$ and $a_{34} = -5$.

In general, it is possible to express a $m \times n$ matrix in terms of its entries as:

$$B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{pmatrix}$$

m.2 Matrix operations: addition

Similarly to numbers, operations can be defined on matrices. The most basic of these operations is matrix addition:

definition m.4: let A and B be two matrices of similar dimension $m \times n$; then the **matrix addition** of A and B is the $m \times n$ matrix $A + B$ such that $(a+b)_{ij} = a_{ij} + b_{ij}$. In other words:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{pmatrix} = \begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \dots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & \dots & a_{2n}+b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \dots & a_{mn}+b_{mn} \end{pmatrix}$$

Matrix addition exists only if the matrices involved are of similar dimension, that is, share the same number of rows and columns. Otherwise, it is not defined.

example m.5:

Let: $A = \begin{pmatrix} 2 & 3 & 0 \\ -1 & 4 & -3 \end{pmatrix}$ $B = \begin{pmatrix} 1 & 0 & 8 \\ -2 & 3 & 1 \end{pmatrix}$ $C = \begin{pmatrix} -3 & 4 \\ 7 & -1 \end{pmatrix}$

Then: $A+B = \begin{pmatrix} 2 & 3 & 0 \\ -1 & 4 & -3 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 8 \\ -2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 2+1 & 3+0 & 0+8 \\ -1-2 & 4+3 & -3+1 \end{pmatrix} = \begin{pmatrix} 3 & 3 & 8 \\ -3 & 7 & -2 \end{pmatrix}$

On the other hand, the operations $A+C$ and $B+C$ are not defined since the matrix dimensions don't agree.

Matrix addition has the following properties:

property m.1: let A and B be matrices such that $A+B$ is defined; then $A+B = B+A$ (commutative property).

property m.2: let A, B and C be matrices such that $A+B+C$ is defined; then $(A+B)+C = A+(B+C)$ (associative property).

m.3 Matrix operations: subtraction

Matrix subtraction is simply the counterpart of matrix addition.

definition m.5: let A and B be two matrices of similar dimension $m \times n$; then the **matrix subtraction** of A and B is the $m \times n$ matrix $A-B$ such that $(a-b)_{ij} = a_{ij} - b_{ij}$. In other words:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} - \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{pmatrix} = \begin{pmatrix} a_{11}-b_{11} & a_{12}-b_{12} & \dots & a_{1n}-b_{1n} \\ a_{21}-b_{21} & a_{22}-b_{22} & \dots & a_{2n}-b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}-b_{m1} & a_{m2}-b_{m2} & \dots & a_{mn}-b_{mn} \end{pmatrix}$$

Similarly to matrix addition, matrix subtraction exists only if the matrices involved are of similar dimension. Otherwise, it is not defined.

example m.6:

Let: $A = \begin{pmatrix} 5 & -2 & 1 \\ -3 & 3 & 0 \end{pmatrix}$ $B = \begin{pmatrix} 0 & 7 & 4 \\ 4 & 1 & 2 \end{pmatrix}$ $C = \begin{pmatrix} 2 & -1 \\ 5 & 1 \end{pmatrix}$

Then: $A-B = \begin{pmatrix} 5 & -2 & 1 \\ -3 & 3 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 7 & 4 \\ 4 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 5-0 & -2-7 & 1-4 \\ -3-4 & 3-1 & 0-2 \end{pmatrix} = \begin{pmatrix} 5 & -9 & -3 \\ -7 & 2 & -2 \end{pmatrix}$

On the other hand, the operations $A-C$ and $B-C$ are not defined since the matrix dimensions don't agree.

m.4 Matrix operations: multiplication

Matrix multiplication constitutes the next step after matrix addition and matrix subtraction. The simplest version of matrix multiplication is the scalar multiplication:

definition m.6: let a be some scalar, and let B be some $m \times n$ matrix; then the **scalar multiplication** aB is the $m \times n$ matrix such that $(ab)_{ij} = a \times b_{ij}$. In other words:

$$a \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{np} \end{pmatrix} = \begin{pmatrix} ab_{11} & ab_{12} & \dots & ab_{1n} \\ ab_{21} & ab_{22} & \dots & ab_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ab_{m1} & ab_{m2} & \dots & ab_{mn} \end{pmatrix}$$

For instance:

example m.7:

Let: $\quad a = 3 \quad B = \begin{pmatrix} 2 & -1 & -2 \\ 1 & 3 & 0 \end{pmatrix}$

The scalar multiplication aB is given by:

$$aB = (3) \begin{pmatrix} 2 & -1 & -2 \\ 1 & 3 & 0 \end{pmatrix} = \begin{pmatrix} 3 \times 2 & 3 \times (-1) & 3 \times (-2) \\ 3 \times 1 & 3 \times 3 & 3 \times 0 \end{pmatrix} = \begin{pmatrix} 6 & -3 & -6 \\ 3 & 9 & 0 \end{pmatrix}$$

Multiplication with the scalar multiplication works much like matrix addition or subtraction: the operation is realised on pairwise elements of the two matrices. This simple logic only applies when the first matrix is a scalar. Matrix multiplication in general is more complicated, and is defined as follows:

definition m.7: let A be some $m \times n$ matrix, and let B be some $n \times k$ matrix; then the **matrix product** AB is the $m \times k$ matrix such that $(ab)_{ij} = \sum_{h=1}^{n} a_{ih} b_{hj}$. In other words:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1k} \\ b_{21} & b_{22} & \dots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nk} \end{pmatrix} = \begin{pmatrix} \sum_{h=1}^{n} a_{1h} b_{h1} & \sum_{h=1}^{n} a_{1h} b_{h2} & \dots & \sum_{h=1}^{n} a_{1h} b_{hk} \\ \sum_{h=1}^{n} a_{2h} b_{h1} & \sum_{h=1}^{n} a_{2h} b_{h2} & \dots & \sum_{h=1}^{n} a_{2h} b_{hk} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{h=1}^{n} a_{mh} b_{h1} & \sum_{h=1}^{n} a_{mh} b_{h2} & \dots & \sum_{h=1}^{n} a_{mh} b_{hk} \end{pmatrix}$$

The definition implies that for a matrix product AB to be defined, A must be $m \times n$, and B must be $n \times k$. In other words, the number of columns of the first matrix must be equal to the number of rows of the second matrix. Otherwise, the product is not defined.

example m.8:

Let: $\quad A = \begin{pmatrix} -1 & 2 & 6 \\ 0 & -2 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 4 & -1 \\ 3 & 2 \\ -2 & 4 \end{pmatrix} \quad C = \begin{pmatrix} -2 & 3 \\ 1 & 2 \end{pmatrix}$

The matrix product AB is defined, since A has 3 columns and B has 3 rows. Similarly, the product BC is defined, since B has 2 columns and C has 2 rows. The matrix product AC is not defined however, since A has 3 columns while C has 2 rows.

When A is $m \times n$ and B is $n \times k$, the product AB is well defined. In this case, the resulting matrix is $n \times k$. That is, the matrix resulting from the product AB has a number of rows equal to the number of rows of A, and a number of columns equal to the number of columns of B.

example m.9:

Let: $A = \begin{pmatrix} 2 & -2 & 3 \\ 1 & 0 & -4 \end{pmatrix}$ $B = \begin{pmatrix} 2 \\ -1 \\ 5 \end{pmatrix}$.

A has 3 columns and B has 3 rows. Hence the product AB is defined. Since A has 2 rows and B has 1 column, the matrix resulting from the product AB will be of dimension 2×1.

The final step consists in computing the product itself. When the product AB is defined, the entry of row i, column j of AB is obtained by calculating the product of row i of A with column j of B.

example m.10:

Let: $A = \begin{pmatrix} 3 & 1 & -1 \\ 3 & -2 & 2 \end{pmatrix}$ $B = \begin{pmatrix} 0 & 1 \\ 4 & -1 \\ 6 & 0 \end{pmatrix}$

A has 3 columns and B has 3 rows, hence the product AB is defined. A has 2 rows and B has 2 columns, hence the matrix resulting from the product AB is of dimension 2×2.

The entry of row 1, column 1 of the product AB is obtained by multiplying row 1 of matrix A with column 1 of matrix B: $(ab)_{11} = \sum_{h=1}^{3} a_{1h} b_{h1} = 3 \times 0 + 1 \times 4 - 1 \times 6 = -2$

Similarly, the entry of row 1, column 2 of the product AB is obtained by multiplying row 1 of matrix A with column 2 of matrix B: $(ab)_{12} = \sum_{h=1}^{3} a_{1h} b_{h2} = 3 \times 1 + 1 \times (-1) - 1 \times 0 = 2$.

Continuing in a similar fashion for the two remaining entries, the complete product obtains as:

$$AB = \begin{pmatrix} 3 & 1 & -1 \\ 3 & -2 & 2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 4 & -1 \\ 6 & 0 \end{pmatrix} = \begin{pmatrix} -2 & 2 \\ 4 & 5 \end{pmatrix}$$

Matrix product is not commutative: in general $AB \neq BA$. In fact, BA may not even be defined, even though AB is.

A number of convenient properties apply to scalar multiplication and matrix products, and are now introduced to conclude this section:

property m.3: let a be some scalar, and let B and C be matrices such that BC is defined.
Then $a(BC) = (aB)C = B(aC) = (BC)a$ (associative property of scalar multiplication).

property m.4: let a be some scalar, and let B and C be matrices such that $B+C$ is defined.
Then $a(B+C) = aB + aC = Ba + Ca = (B+C)a$ (distributive property of scalar multiplication).

property m.5: let A, B and C be matrices such that ABC is defined.
Then $ABC = (AB)C = A(BC)$ (associative property of matrix product).

property m.6: let A, B and C be matrices such that AB, AC and $B+C$ are defined.
Then $A(B+C) = AB + AC$ (left distributivity).

property m.7: let A, B and C be matrices such that AC, BC and $A+B$ are defined.
Then $(A+B)C = AC + BC$ (right distributivity).

m.5 Matrix operations: inversion

Strictly speaking, division is not defined for matrices. The closest equivalent is the concept of matrix inversion. Before discussing inversion however, it is necessary to introduce an important type of matrices.

definition m.8: the **identity matrix** of size n, denoted by I or sometimes I_n to stress the dimension, is the $n \times n$ matrix that has 1 entries on its main diagonal, and 0 entries everywhere else. In other words:

$$I_n = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

For instance:

example m.11: The identity matrices of size 2 and 4 are given by:

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The remarkable property of the identity matrix is that any matrix pre or post-multiplied by it is left unchanged. That is, if A is a $m \times n$ matrix, then $I_m\, A = A$ and $A\, I_n = A$. In this sense, the identity matrix represents the equivalent of a multiplication by 1 in the case of scalars.

example m.12:

Let: $\quad A = \begin{pmatrix} 0 & -1 & 4 \\ 7 & -1 & -3 \end{pmatrix}$

Then, computing the products, one can verify that:

$$I_2\, A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 4 \\ 7 & -1 & -3 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 4 \\ 7 & -1 & -3 \end{pmatrix}$$

and

$$A\, I_3 = \begin{pmatrix} 0 & -1 & 4 \\ 7 & -1 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 4 \\ 7 & -1 & -3 \end{pmatrix}$$

It is then possible to introduce the concept of matrix inverse:

definition m.9: let A be some $n \times n$ matrix. Then if it exists, the **inverse** of A, denoted by A^{-1}, is the $n \times n$ matrix such that $AA^{-1} = A^{-1}A = I_n$.

The following example illustrates the definition:

example m.13:

Let: $\quad A = \begin{pmatrix} -2 & -5 \\ 1 & 3 \end{pmatrix}$

Then the inverse of A is given by:

$$A^{-1} = \begin{pmatrix} -3 & -5 \\ 1 & 2 \end{pmatrix}$$

Indeed, it is immediate to check that:

$$AA^{-1} = \begin{pmatrix} -2 & -5 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} -3 & -5 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

and

$$A^{-1}A = \begin{pmatrix} -3 & -5 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} -2 & -5 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The next example illustrates the equivalence between scalar division and matrix inversion:

example m.14: one of the main interest of matrix inversion lies the resolution of systems of linear equations. Consider the following system of linear equations:

$x_1 + 3x_3 = 1$
$2x_2 - x_3 = 2$
$x_1 + 3x_2 + x_3 = 3$

This system can be reformulated in matrix form as:

$$\begin{pmatrix} 1 & 0 & 3 \\ 0 & 2 & -1 \\ 1 & 3 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ 3 \end{pmatrix}$$

Or $Ax = b$, with:

$$A = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 2 & -1 \\ 1 & 3 & 1 \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \qquad b = \begin{pmatrix} 4 \\ 2 \\ 3 \end{pmatrix}$$

If both sides of the system are pre-multiplied by A^{-1}, one obtains:

$A^{-1}Ax = A^{-1}b \quad \Rightarrow \quad Ix = A^{-1}b \quad \Rightarrow \quad x = A^{-1}b$

In other words, the value of x that satisfies the system of equations can be obtained directly from the inverse A^{-1}. It is readily verifiable that A^{-1} is given by:

$$A^{-1} = \begin{pmatrix} -5 & -9 & 6 \\ 1 & 2 & -1 \\ 2 & 3 & -2 \end{pmatrix}$$

Following:

$$x = A^{-1}b \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -5 & -9 & 6 \\ 1 & 2 & -1 \\ 2 & 3 & -2 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -5 \\ 2 \\ 2 \end{pmatrix}$$

This example illustrates the relation between matrix inversion and the standard scalar division. Multiplying a scalar by its inverse results in a value of 1. Much the same way, when a matrix is multiplied by its inverse, the result is the identity matrix. So, when the system $Ax = b$ is pre-multiplied by the inverse A^{-1}, it turns A into the identity matrix, effectively eliminating it from the left-hand side, leaving only x remaining. In a way, the operation effectively "divides" both sides of the system by A.

There exist different ways to calculate the inverse of a matrix. One method that is commonly used is based on the concepts of determinant and adjoint of a matrix.

definition m.10: let A be some $n \times n$ invertible matrix. Then there exists a number called the **determinant** of A and denoted by $|A|$, and a $n \times n$ matrix called the **adjoint** of A and denoted by $adj(A)$ such that:
$A^{-1} = \frac{1}{|A|} adj(A)$

Computing the determinant and the adjoint of A typically requires complicated calculations and is beyond the scope of this manual. The general methodology is thus not developed here, but for the sake of illustration the formulas are provided for the simple case where A is 2×2.

property m.8: let A be a 2×2 invertible matrix, so that $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$.

Then $|A| = a_{11}a_{22} - a_{21}a_{12}$ and $adj(A) = \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$.

It is not necessarily the case that an inverse exists for a given matrix. This is in fact related to the concept of determinant. A square matrix which has a determinant equal to zero cannot be inverted. On the other hand, any non-zero determinant implies the possibility of inversion.

definition m.11: let A be some $n \times n$ matrix. If $|A| = 0$, then A is said to be **singular** and it cannot be inverted. If $|A| \neq 0$, then a well-defined inverse A^{-1} exists.

For instance:

example m.15:

Let: $A = \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix}$

Then $|A| = 1 \times 5 - 3 \times 2 = -1$. Because $|A| \neq 0$, the inverse of A exists.

Also, $adj(A) = \begin{pmatrix} 5 & -2 \\ -3 & 1 \end{pmatrix}$.

Then: $A^{-1} = \frac{1}{|A|} adj(A) = \frac{1}{-1} \begin{pmatrix} 5 & -2 \\ -3 & 1 \end{pmatrix} = \begin{pmatrix} -5 & 2 \\ 3 & -1 \end{pmatrix}$

It can then be readily verified that:

$$AA^{-1} = \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} -5 & 2 \\ 3 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

and

$$A^{-1}A = \begin{pmatrix} -5 & 2 \\ 3 & -1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

To conclude this section, a number of common results about matrix inverses and determinants are introduced.

property m.9: let A be some $n \times n$ invertible matrix; then the inverse A^{-1} is unique. (uniqueness of matrix inverse)

property m.10: let A be some $n \times n$ invertible matrix; then $\left(A^{-1}\right)^{-1} = A$. (inverse of matrix inverse)

property m.11: let a be some scalar and B be some $n \times n$ matrix; then $(aB)^{-1} = a^{-1}B^{-1}$. (inverse of scalar multiplication)

property m.12: let A and B be two $n \times n$ invertible matrices; then $(AB)^{-1} = B^{-1}A^{-1}$. (inverse of matrix product)

property m.13: let A and D be two invertible matrices, and let B and C be matrices with compliant dimensions; then:
$(A + BDC)^{-1} = A^{-1} - A^{-1}B(D^{-1} + CA^{-1}B)^{-1}CA^{-1}$. (Sherman-Woodbury-Morrison identity)

property m.14: let a be some scalar and B be some $n \times n$ matrix; then $|aB| = a^n|B|$. (determinant of scalar multiplication)

property m.15: let A and B be two $n \times n$ matrices; then $|AB| = |A||B|$. (determinant of matrix product)

property m.16: let A be some $n \times n$ invertible matrix; then $|A^{-1}| = |A|^{-1}$. (determinant of matrix inverse)

The next two results are less standard, but they can prove occasionally useful in Bayesian statistics.

property m.17: let A be some $m \times n$ matrix, and B be some $n \times m$ matrix, so that AB and BA are defined. Then $|I_m + AB| = |I_n + BA|$. (Sylvester's determinant identity)

property m.18: let A be some $m \times n$ matrix, and B be some $n \times m$ matrix, so that AB and BA are defined; also, let C be any $m \times m$ invertible matrix. Then $|C + AB| = |C||I_n + BC^{-1}A|$. (generalisation of Sylvester's determinant identity).

m.6 Matrix operations: transposition

Matrix transposition represents a very common operation in matrix algebra. It is formally defined as follows.

definition m.12: let A be some $m \times n$ matrix; then the **transpose** of A, denoted by A' or A^T, is the $n \times m$ matrix such that row i of A becomes column i of A', for $i = 1, 2, \ldots, m$.

In other words, transposing a matrix means interchanging its rows with its columns, or equivalently, fliping the matrix over its main diagonal. For instance:

example m.16:

Let: $A = \begin{pmatrix} 3 & 8 & -9 \\ 1 & 0 & 4 \end{pmatrix}$ $b = \begin{pmatrix} 1 \\ -2 \\ 0 \\ 7 \end{pmatrix}$

Then: $A' = \begin{pmatrix} 3 & 1 \\ 8 & 0 \\ -9 & 4 \end{pmatrix}$ $b' = \begin{pmatrix} 1 & -2 & 0 & 7 \end{pmatrix}$

Matrix transposes have a number of convenient properties.

property m.19: let a be some scalar. Then $a' = a$.

property m.20: let A be some matrix. Then $(A')' = A$.

property m.21: let A and B be matrices such that $A + B$ is defined. Then $(A + B)' = A' + B'$.

property m.22: let a be some scalar and B be some matrix. Then $(aB)' = aB'$.

property m.23: let A and B be matrices such that AB is defined. Then $(AB)' = B'A'$.

property m.24: let A be some some $n \times n$ invertible matrix. Then $(A^{-1})' = (A')^{-1}$.

property m.25: let A be some some $n \times n$ matrix. Then $|A'| = |A|$.

m.7 Some special matrices

This section introduces a number of special matrices that are commonly encountered in matrix algebra in general, and in statistics in particular. The presentation starts with a very basic concept:

definition m.13: a **square matrix** is a matrix which has as many rows as columns.

In other words, a square matrix is a matrix of dimension $n \times n$. Some occurrences of square matrices have already been introduced. The identity matrix for instance is a square matrix. Also, only square matrix can be inverted.

example m.17:

Let: $A = \begin{pmatrix} 1 & 0 \\ -3 & 2 \end{pmatrix}$ $B = \begin{pmatrix} 1 & 6 & -3 \\ -2 & 0 & 4 \end{pmatrix}$

Then A is square, while B is not.

Another important concept is that of diagonal:

definition m.14: let A be some matrix; the **main diagonal** of A is the collection of entries a_{ij} of A with $i = j$.

The main diagonal of a matrix A is thus the collection of entries $a_{11}, a_{22}, \ldots, a_{nn}$, where n is the smallest dimension of A (number of rows or columns).

example m.18:

Let: $A = \begin{pmatrix} -2 & 3 & 1 \\ -3 & 1 & 0 \end{pmatrix}$ $B = \begin{pmatrix} 1 & -7 & 6 \\ 0 & -5 & 2 \\ 0 & -1 & 2 \\ 8 & 1 & -3 \end{pmatrix}$

Then the main diagonal of A consists in entries $a_{11} = -2$ and $a_{22} = 1$, while the main diagonal of B is made of entries $b_{11} = 1$, $b_{22} = -5$ and $b_{33} = 2$.

The concept of main diagonal naturally extends to that of a diagonal matrix.

definition m.15: a square matrix A is **diagonal** if the entries outside its main diagonal are all zeros.

For instance:

example m.19:

Let: $A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7 \end{pmatrix}$ $B = \begin{pmatrix} -3 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ $C = \begin{pmatrix} 6 & 0 & 0 \\ 0 & -2 & 0 \\ 1 & 0 & -3 \end{pmatrix}$.

A is a diagonal matrix. B is not diagonal since it is not square. C is not diagonal since one entry outside the main diagonal is non-zero ($c_{31} = 1$).

The simplicity of diagonal matrices makes them trivial to invert:

property m.26: let A be a $n \times n$ invertible diagonal matrix. Then the inverse of A is the diagonal matrix such that: $a_{ii}^{-1} = 1/a_{ii}$, for $i = 1, 2, \ldots, n$.

In other words, the inverse of a diagonal matrix obtain from the inverse of its main diagonal. For instance:

example m.20:

Let: $A = \begin{pmatrix} 3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 5 \end{pmatrix}$ then $A^{-1} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & -\frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{5} \end{pmatrix}$

Another useful property of diagonal matrices is the simplicity of their determinants:

property m.27: Let A be a $n \times n$ diagonal matrix. Then the determinant of A is the product of the terms on its main diagonal, so that: $|A| = \prod_{i=1}^{n} a_{ii}$.

Another class of important matrices based on the main diagonal is the class of triangular matrices.

definition m.16: a square matrix A is **lower triangular** if the entries above its main diagonal are all zeros; a square matrix A is **upper triangular** if the entries below its main diagonal are all zeros.

For instance:

example m.21:

Let: $A = \begin{pmatrix} 1 & 0 & 0 \\ 2 & -3 & 0 \\ 5 & 7 & -8 \end{pmatrix}$ $B = \begin{pmatrix} -1 & 4 & -2 \\ 0 & 1 & 2 \\ 0 & 0 & -8 \end{pmatrix}$.

Then A is a lower triangular matrix and B is an upper triangular matrix.

Similarly to diagonal matrices, the determinant of triangular matrices is easy to calculate.

property m.28: let A be a $n \times n$ lower or upper triangular matrix. Then the determinant of A is the product of the terms on its main diagonal, so that: $|A| = \prod_{i=1}^{n} a_{ii}$.

Another class of special matrices is that of symmetric matrices:

definition m.17: a square matrix A is a **symmetric matrix** if $A = A'$.

Hence, as the name indicates, a symmetric matrix is a matrix which is symmetric around its main diagonal. For instance:

example m.22:

Let: $A = \begin{pmatrix} -2 & -1 & 4 \\ -1 & 3 & 5 \\ 4 & 5 & 0 \end{pmatrix}$.

Then A is a symmetric matrix.

A concept closely related to that of symmetric matrix is that of positive definiteness:

definition m.18: let A be some square, $n \times n$ matrix; then A is **positive definite** if for any vector x of dimension n, one has $x'Ax > 0$.

For instance:

example m.23:

Let: $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

Then: $x'Ax = \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2x_1^2 - 2x_2x_1 + 2x_2^2 = x_1^2 + (x_1 - x_2)^2 + x_2^2$

All the terms in the sum are positive, so the sum is positive and $x'Ax > 0$ for any vector x. Hence A is positive definite.

Matrices that are both symmetric and positive definite have interesting properties. For this reason, they are used extensively in statistics. One such property is the decomposition of any symmetric and positive definite matrix into lower triangular matrices:

definition m.19: let A be a symmetric and positive definite matrix; the **Cholesky factor** of A is the lower triangular matrix G such that $GG' = A$.

For instance:

example m.24:

Let: $A = \begin{pmatrix} 9 & -6 \\ -6 & 5 \end{pmatrix}$

Then $G = \begin{pmatrix} -3 & 0 \\ 2 & 1 \end{pmatrix}$ is the Cholesky factor of A. Indeed, it is immediate to check that:

$GG' = \begin{pmatrix} -3 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} -3 & 2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 9 & -6 \\ -6 & 5 \end{pmatrix} = A$

Cholesky factors exist for any symmetric positive definite matrix. This is stated in the next property:

property m.29: let A be a symmetric and positive definite matrix; then there exists a Cholesky factor of A, and this Cholesky factor is unique.

An alternative decomposition for a symmetric and positive definite matrix is the triangular factorisation:

definition m.20: let A be a symmetric and positive definite matrix; the **triangular factorisation** of A consists in the pair of matrices F and L such that $FLF' = A$, with F a lower triangular matrix with ones on the main diagonal, and L a diagonal matrix.

For instance:

example m.25:

Let: $A = \begin{pmatrix} 2 & -4 \\ -4 & 11 \end{pmatrix}$

Then the pair of matrices $F = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}$ and $L = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$ represent the triangular factorisation of A.

Indeed, it is immediate to check that:

$FLF' = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & -4 \\ -4 & 11 \end{pmatrix} = A$

Similarly to Cholesky factorisation, triangular factorisation exists for any symmetric positive definite matrix:

property m.30: let A be a symmetric and positive definite matrix; then there exists a triangular factorisation of A, and this triangular factorisation is unique.

Often, one works with the inverse of triangular factorisation matrices. In this respect, the following result proves very useful:

property m.31: let A be some $n \times n$ invertible lower triangular matrix with ones on the main diagonal; then its inverse A^{-1} is also lower triangular with ones on the main diagonal, and is given by:
$A^{-1} = I_n - B + B^2 + \ldots + (-1)^{(n-1)} B^{(n-1)}$,
where B is a lower triangular matrix with zeros on the main diagonal and $b_{ij} = a_{ij}$ below the diagonal.

proof: A can be written as $A = I_n + B$. Also, the definition of B implies that B^n is the $n \times n$ zero matrix. Following:
$A\left(I_n - B + B^2 + \ldots + (-1)^{n-1} B^{n-1})\right) = (I_n + B)\left(I_n - B + B^2 + \ldots + (-1)^{n-1} B^{n-1})\right)$
$\left(I_n - B + B^2 + \ldots + (-1)^{n-1} B^{n-1})\right) + \left(B - B^2 + B^3 + \ldots + (-1)^{n-2} B^{n-1} + (-1)^{n-1} B^n)\right)$
$= I_n$.
Hence, from the definition of a matrix inverse, $I_n - B + B^2 + \ldots + (-1)^{(n-1)} B^{(n-1)}) = A^{-1}$.
To prove the first part of the assertion, note that the term $-B + B^2 + \ldots + (-1)^{n-1} B^{n-1}$ is a summation of terms which are of powers of B. Because of the definition of B, the summation is a lower triangular matrix with zeros on the main diagonal. Following, the full term $I_n - B + B^2 + \ldots + (-1)^{n-1} B^{n-1}$ is lower triangular with ones on the main diagonal.

The following example illustrates this property:

example m.26:

Let: $A = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 2 & 3 & 1 \end{pmatrix}$ then $B = \begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 2 & 3 & 0 \end{pmatrix}$ $B^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -6 & 0 & 0 \end{pmatrix}$

Following:

$$A^{-1} = I_3 - B + B^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 2 & 3 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -6 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -8 & -3 & 1 \end{pmatrix}$$

One can check that:

$$AA^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 2 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -8 & -3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

To conclude, the details of the calculations involved in the estimation of the Cholesky and triangular factorisations are developed. This part can be skipped if one is not interested in computational details.

property m.32: let A be some $n \times n$ symmetric positive definite matrix; then its Cholesky factor G can be estimated from:

$g_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} g_{jk}^2}$ (diagonal term of column j)

$g_{ij} = \frac{1}{g_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} g_{ik}g_{jk}\right)$ (for $i > j$, terms below the diagonal of column j)

proof: the Cholesky decomposition of A implies that $GG' = A$. Developing the involved matrices yields:

$$\begin{pmatrix} g_{11} & 0 & \dots & 0 \\ g_{21} & g_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \dots & g_{nn} \end{pmatrix} \begin{pmatrix} g_{11} & g_{21} & \dots & g_{n1} \\ 0 & g_{22} & \dots & g_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & g_{nn} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Then, computing the product on the left-hand side and ignoring the upper triangular part of A because of symmetry:

$$\begin{pmatrix} g_{11}^2 & & & \\ g_{21}g_{11} & g_{21}^2 + g_{22}^2 & & \\ \vdots & \vdots & \ddots & \\ g_{n1}g_{11} & g_{n1}g_{21} + g_{n2}g_{22} & \dots & \sum_{k=1}^{n} g_{nk}^2 \end{pmatrix} = \begin{pmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Column 1 yields:

$$\begin{aligned} g_{11}^2 = a_{11} \quad &\Rightarrow \quad g_{11} = \sqrt{a_{11}} \\ g_{21}g_{11} = a_{21} \quad &\Rightarrow \quad g_{21} = \tfrac{a_{21}}{g_{11}} \\ &\vdots \\ g_{n1}g_{11} = a_{n1} \quad &\Rightarrow \quad g_{n1} = \tfrac{a_{n1}}{g_{11}} \end{aligned}$$

Column 2 yields:

$$\begin{aligned} g_{21}^2 + g_{22}^2 = a_{22} \quad &\Rightarrow \quad g_{22} = \sqrt{a_{22} - g_{21}^2} \\ g_{31}g_{21} + g_{32}g_{22} = a_{32} \quad &\Rightarrow \quad g_{32} = \tfrac{1}{g_{22}}(a_{32} - g_{31}g_{21}) \\ &\vdots \\ g_{n1}g_{21} + g_{n2}g_{22} = a_{n2} \quad &\Rightarrow \quad g_{n2} = \tfrac{1}{g_{22}}(a_{n2} - g_{n1}g_{21}) \end{aligned}$$

Going on this way, one obtains that in general:

$$g_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} g_{jk}^2} \qquad g_{ij} = \frac{1}{g_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} g_{ik}g_{jk}\right)$$

The same kind of result can be obtained for triangular factorisation:

property m.33: let A be some $n \times n$ symmetric positive definite matrix; then its triangular factorisation matrices F and L can be estimated from:

$l_{jj} = a_{jj} - \sum_{k=1}^{j-1} f_{jk}^2 l_{kk}$ (terms of the L matrix)

$f_{ij} = \frac{1}{l_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} f_{ik}f_{jk}l_{kk}\right)$ (terms of the F matrix)

What is the best alternative between Cholesky and triangular factorization? On the one hand, triangular factorization is more robust numerically because it does not involve square root calculations. On the other hand, Cholesky factorization involves fewer operations and is usually faster. Which one is best thus often depends on the context. Also, the two methods are related, so that using one makes it easy to recover the other. This is stated in the next property:

property m.34: let A be some $n \times n$ symmetric positive definite matrix, so that there exists a Cholesky factor G such that $GG' = A$, and a pair of triangular factorisation matrices F and L such that $FLF' = A$. Then G, F and L are linked by the following relations:

$g_{jj} = \sqrt{l_{jj}}$ (diagonal term of column j)
$g_{ij} = f_{ij}\sqrt{l_{jj}}$, (for $i > j$, terms below the diagonal of column j)

proof: as $GG' = A$ and $FLF' = A$, then $GG' = FLF'$. Also, because L is a diagonal matrix, it is possible to define its square root as the diagonal matrix $L^{1/2}$ whose main diagonal entries are $\sqrt{l_{11}}, \sqrt{l_{22}}, \ldots$, so that $L^{1/2}L^{1/2} = L$. Following, $GG' = FL^{1/2}L^{1/2}F' = FL^{1/2}(L^{1/2})'F' = (FL^{1/2})(FL^{1/2})'$, and thus $G = FL^{1/2}$.

Developing the involved matrices yields:

$$\begin{pmatrix} g_{11} & 0 & \ldots & 0 \\ g_{21} & g_{22} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \ldots & g_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ f_{21} & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \ldots & 1 \end{pmatrix} \begin{pmatrix} \sqrt{l_{11}} & 0 & \ldots & 0 \\ 0 & \sqrt{l_{22}} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sqrt{l_{nn}} \end{pmatrix}$$

Then, computing the product on the right-hand side:

$$\begin{pmatrix} g_{11} & 0 & \ldots & 0 \\ g_{21} & g_{22} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \ldots & g_{nn} \end{pmatrix} = \begin{pmatrix} \sqrt{l_{11}} & 0 & \ldots & 0 \\ f_{21}\sqrt{l_{11}} & \sqrt{l_{22}} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1}\sqrt{l_{11}} & f_{n2}\sqrt{l_{22}} & \ldots & \sqrt{l_{nn}} \end{pmatrix}$$

The correspondence between pairwise entries establishes the result.

A final remark to conclude this section: the Cholesky factorization can also be used as an efficient way to invert positive definite matrices. This is stated in the next property:

property m.35: let A be some $n \times n$ invertible, symmetric and positive definite matrix. Then the inverse of A can be obtained from:
$A^{-1} = (G^{-1})'\, G^{-1}$
where G is the Cholesky factor of A.

Indeed, because $A = GG'$, then $A^{-1} = (GG')^{-1} = (G')^{-1}G^{-1} = (G^{-1})'\, G^{-1}$. The benefit of this procedure over regular inversion is that inverting the lower triangular Cholesky factor G is considerably cheaper than inverting A directly, thanks to what is known as back substitution. The total calculation time is thus significantly reduced with this method.

m.8 Kronecker products

An alternative to the standard matrix product is the so-called Kronecker product, defined as follows:

definition m.21: let A be a $m \times n$ matrix, and B be a $p \times q$ matrix; the **Kronecker product** of A and B, denoted by $A \otimes B$, is the $mp \times nq$ matrix given by:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{pmatrix}$$

For instance:

example m.27:

$$\text{Let:} \qquad A = \begin{pmatrix} 2 & -3 \\ 0 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 5 & -2 \\ -1 & 0 & 7 \\ -4 & 8 & 0 \end{pmatrix}$$

Then:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix}$$

$$= \begin{pmatrix} (2)\begin{pmatrix} 1 & 5 & -2 \\ -1 & 0 & 7 \\ -4 & 8 & 0 \end{pmatrix} & (-3)\begin{pmatrix} 1 & 5 & -2 \\ -1 & 0 & 7 \\ -4 & 8 & 0 \end{pmatrix} \\ (0)\begin{pmatrix} 1 & 5 & -2 \\ -1 & 0 & 7 \\ -4 & 8 & 0 \end{pmatrix} & (1)\begin{pmatrix} 1 & 5 & -2 \\ -1 & 0 & 7 \\ -4 & 8 & 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 2 & 10 & -4 & -3 & -15 & 6 \\ -2 & 0 & 14 & 3 & 0 & -21 \\ -8 & 16 & 0 & 12 & -24 & 0 \\ 0 & 0 & 0 & 1 & 5 & -2 \\ 0 & 0 & 0 & -1 & 0 & 7 \\ 0 & 0 & 0 & -4 & 8 & 0 \end{pmatrix}$$

Unlike the regular matrix product which may not be defined, the kronecker product is always defined for any pair of matrices A and B. Nevertheless, similarly to standard matrix products, Kronecker products are not commutative: in general $A \otimes B \neq B \otimes A$.

Kronecker products have the following properties:

property m.36: let a be some scalar and B be some matrix; then $aB = a \otimes B = B \otimes a = Ba$.

property m.37: let A and B be two matrices; then $(A \otimes B)' = A' \otimes B'$.

property m.38: let A and B be two invertible matrices; then $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.

property m.39: let A, B and C be matrices such that $A + B$ is defined; then $A \otimes C + B \otimes C = (A + B) \otimes C$.

property m.40: let A, B and C be matrices such that $B + C$ is defined; then $A \otimes B + A \otimes C = A \otimes (B + C)$.

property m.41: let a be some scalar, and B and C be two matrices; then:
$a(B \otimes C) = (aB) \otimes C = B \otimes (aC) = (B \otimes C)a$.

property m.42: let A, B, C and D matrices such that AC and BD are defined; then:
$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$.

property m.43: let A be some $n \times n$ matrix, and B be some $k \times k$ matrix; then $|A \otimes B| = |A|^k |B|^n$.

m.9 Matrix rank

The rank of a matrix is related to the notion of linear independence:

definition m.22: let A be some matrix; the **rank** of A, denoted by $rank(A)$, is the maximum number of linearly independent rows (or columns) in A.

For instance:

example m.28:

Let: $A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & -1 & 3 \\ 4 & -3 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 3 & 5 \\ 2 & -1 & 3 \\ 4 & -3 & 5 \end{pmatrix}$

Then $rank(A) = 3$ since all the columns (and rows) of A are linearly independent.

On the other hand, $rank(B) = 2$ since there are only two linearly independent columns in B. Indeed, the third column of B is a linear combination of the first two columns:

$$\begin{pmatrix} 5 \\ 3 \\ 5 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ -1 \\ -3 \end{pmatrix}$$

The rank of a matrix has the following properties:

property m.44: let A be some matrix; then the number of linearly independent rows of A is equal to the number of linearly independent columns of A. In other words, the row rank of A is equal to the column rank of A.

property m.45: let A be some $n \times m$ matrix; then $rank(A) \leq min(n,m)$. In other words, the rank of a matrix is at most equal to the minimum between the number of row and the number of columns.

The notion of full rank is defined as follows.

definition m.23: let A be some matrix; if all the rows (or columns) of A are linearly independent, then A is said to have **full rank** .

For instance:

example m.29:

Let A and B be defined as in example m.28. Then A has full rank, while B hasn't.

The notion of rank is closely related to the notion of invertibility, as stated in the next property:

property m.46: let A be some square, $n \times n$ matrix; then A is invertible if and only if A has full rank.

m.10 Matrix trace

In linear algebra, the trace of a square matrix is defined as:

definition m.24: let A be a $n \times n$ square matrix; the **trace** of A, denoted by $tr(A)$, is the sum of its main diagonal entries, so that:
$tr(A) = a_{11} + a_{22} + \ldots + a_{nn} = \sum_{i=1}^{n} a_{ii}$

For instance:

example m.30:

Let: $A = \begin{pmatrix} 0 & 2 & -6 \\ 9 & 2 & 1 \\ 7 & 5 & -3 \end{pmatrix}$

Then:

$tr(A) = a_{11} + a_{22} + a_{33} = 0 + 2 - 3 = -1$

Traces have the following properties:

property m.47: let a be some scalar; then $a = tr(a)$.

property m.48: let a be some scalar and B be some matrix; then $tr(aB) = a\, tr(B)$.

property m.49: let A and B be two $n \times n$ square matrices; then $tr(A+B) = tr(A) + tr(B)$.

property m.50: let A, B, and C be matrices such that the products ABC, CAB and BCA all result in square matrices; then $tr(ABC) = tr(CAB) = tr(BCA)$ (cyclical property).

m.11 Matrix vectorization

In linear algebra, vectorization is used to convert matrices into vectors:

definition m.25: let A be a $m \times n$ matrix; the **vectorization** of A, denoted by $vec(A)$, is the $mn \times 1$ column vector obtained by rearranging the columns of A on top of each other:
$vec(A) = \begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \\ \vdots \\ a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix}$

For instance:

example m.31:

Let: $A = \begin{pmatrix} 1 & 3 & 5 \\ 0 & -4 & 9 \end{pmatrix}$ then $vec(A) = \begin{pmatrix} 1 \\ 0 \\ 3 \\ -4 \\ 5 \\ 9 \end{pmatrix}$

Matrix vectorization has the following properties:

property m.51: let A and B be matrices such that $A+B$ is defined; then $vec(A+B) = vec(A) + vec(B)$.

property m.52: let A and B be matrices such that $A'B$ is a square matrix; then:
$vec(A)'vec(B) = vec(B)'vec(A) = tr(A'B) = tr(AB') = tr(B'A) = tr(BA')$.

property m.53: let a be some column vector; then: $a = vec(a')$.

property m.54: let A, B and C be matrices such that ABC is defined; then $vec(ABC) = (C' \otimes A)vec(B)$.

property m.55: let A, B, C, D, E and F be matrices such that A is $n \times n$ and symmetric, D is $m \times m$, and B, C, E and F are $m \times n$; then:
$tr\big(A^{-1}(B-C)'D^{-1}(E-F)\big) = \big(vec(B)-vec(C)\big)'(A \otimes D)^{-1}\big(vec(E)-vec(F)\big)$.

proof:
$$
\begin{aligned}
& tr\big(A^{-1}(B-C)'D^{-1}(E-F)\big) \\
&= tr\big((B-C)'D^{-1}(E-F)A^{-1}\big) && \text{(m.50)} \\
&= vec(B-C)' \times vec\big(D^{-1}(E-F)A^{-1}\big) && \text{(m.52)} \\
&= vec(B-C)' \times \big((A^{-1} \otimes D^{-1})vec(E-F)\big) && \text{(m.54)} \\
&= \big(vec(B)-vec(C)\big)'(A^{-1} \otimes D^{-1})\big(vec(E)-vec(F)\big) && \text{(m.51)} \\
&= \big(vec(B)-vec(C)\big)'(A \otimes D)^{-1}\big(vec(E)-vec(F)\big) && \text{(m.38)}
\end{aligned}
$$

m.12 Eigenvalues and eigenvectors

Eigenvalues and eigenvectors provide a general way to decompose square matrices. They prove occasionally useful for their relations with matrix determinants and stability analysis.

definition m.26: let A be a $n \times n$ square matrix; let v be a $n \times 1$ vector and λ be a scalar such that $Av = \lambda v$; then v is an **eigenvector** of A, and λ is an **eigenvalue** of A associated to this eigenvector.

For instance:

example m.32:

Let $A = \begin{pmatrix} 4 & -3 \\ 2 & -1 \end{pmatrix}$

Then $v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is an eigenvector of A, and $\lambda_1 = 1$ is the associated eigenvalue.

Indeed, it is straightforward to check that:

$Av_1 = \begin{pmatrix} 4 & -3 \\ 2 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\lambda_1 v_1 = (1)\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Then, $v_2 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ is the second eigenvector of A, and its associated eigenvalue is $\lambda_2 = 2$. It is possible to check that:

$Av_2 = \begin{pmatrix} 4 & -3 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$ and $\lambda_2 v_2 = (2) \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$

How eigenvalues and eigenvectors are computed is beyond the scope of this manual. It is however useful to outline a few properties:

property m.56: let A be a $n \times n$ matrix, and let $\lambda_1, \lambda_2, \ldots, \lambda_n$ denote the n eigenvalues of A; then the trace of A is equal to the sum of its n eigenvalues, namely:

$tr(A) = \sum_{i=1}^{n} \lambda_i$.

property m.57: let A be a $n \times n$ matrix, and let $\lambda_1, \lambda_2, \ldots, \lambda_n$ denote the n eigenvalues of A; then the determinant of A is equal to the product of its n eigenvalues, namely:

$|A| = \prod_{i=1}^{n} \lambda_i$.

property m.58: let A be a $n \times n$ matrix and let t be some scalar; if λ is an eigenvalue of A , then $\lambda + t$ is an eigenvalue of $A + tI_n$.

proof: because λ is an eigenvalue of A, there exists some eigenvector v such that $Av = \lambda v$. Then:
$Av = \lambda v \Rightarrow Av + tv = \lambda v + tv \Rightarrow Av + tI_n v = \lambda v + tv \Rightarrow (A + tI_n)v = (\lambda + t)v$
Therefore, by definition, $\lambda + t$ is an eigenvalue of $A + tI_n$.

property m.59: let A be a $n \times n$ matrix; then the determinant of $I_n + A$ is equal to the product of the eigenvalues of A plus 1:

$|I_n + A| = \prod_{i=1}^{n} (1 + \lambda_i(A))$, where $\lambda_i(A)$ denotes the i^{th} eigenvalue of A.

proof:
$|I_n + A|$
$= \prod_{i=1}^{n} \lambda_i(I_n + A)$ (m.56), where $\lambda_i(I_n + A)$ denotes the i^{th} eigenvalue of $I_n + A$
$= \prod_{i=1}^{n} (1 + \lambda_i(A))$ (m.57) in the case $t = 1$

One of the main use of eigenvalues and eigenvectors is what is known as the diagonalization of matrices. Consider some matrix A of dimension $n \times n$. Then, using eigenvalues and eigenvectors, one can write: $Av_1 = \lambda_1 v_1, Av_1 = \lambda_2 v_2, \cdots, \lambda_n v_n$. These n solutions can be written as a single compact system of matrices as:

$AV = V\Lambda$

with:

$V = \begin{pmatrix} \vdots & \vdots & & \vdots \\ v_1 & v_2 & \ldots & v_n \\ \vdots & \vdots & & \vdots \end{pmatrix}$ and $\Lambda = \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{pmatrix}$.

Post-multiplying both sides by V^{-1} yields:

$A = V\Lambda V^{-1}$

This operation diagonalizes the matrix A by the way of matrix Λ. Its main application in statistics is related to stability analysis, through the following two results:

property m.60: let A be a $n \times n$ square matrix, and let V and Λ be the associated matrices of eigenvectors and eigenvalues; then:

$A^k = V\Lambda^k V^{-1}$

with $\Lambda^k = \begin{pmatrix} \lambda_1^k & 0 & \dots & 0 \\ 0 & \lambda_2^k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n^k \end{pmatrix}$

proof: $A^k = (V\Lambda^k V^{-1})^k = \underbrace{V\Lambda V^{-1} \times V\Lambda V^{-1} \times \ldots \times V\Lambda V^{-1}}_{k \text{ times}} = V \underbrace{\Lambda \times \Lambda \times \ldots \times \Lambda}_{k \text{ times}} V^{-1} = V\Lambda^k V^{-1}$

property m.61: let A be a $n \times n$ square matrix, and let V and Λ be respectively the associated matrices of eigenvectors and eigenvalues; if the n eigenvalues of A are all smaller than one in absolute value, namely $|\lambda_1| < 1, |\lambda_2| < 1, \ldots, |\lambda_n| < 1$, then:

$\lim_{k\to\infty} A^k = 0$

proof: from property m.59, $A^k = V\Lambda^k V^{-1}$. For $i = 1, 2, \ldots, n$, if $|\lambda_i| < 1$, then $\lim_{k\to\infty} \lambda_i^k = 0$. Following:

$$\lim_{k\to\infty} \Lambda^k = \lim_{k\to\infty} \begin{pmatrix} \lambda_1^k & 0 & \dots & 0 \\ 0 & \lambda_2^k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n^k \end{pmatrix} = \begin{pmatrix} \lim_{k\to\infty} \lambda_1^k & 0 & \dots & 0 \\ 0 & \lim_{k\to\infty} \lambda_2^k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lim_{k\to\infty} \lambda_n^k \end{pmatrix} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

Hence $\lim_{k\to\infty} A^k = V \lim_{k\to\infty} \Lambda^k V^{-1} = V0V^{-1} = 0$

m.13 Matrix definiteness

To discuss definiteness, one first needs the notion of quadratic form.

definition m.27: let A be a $n \times n$ square matrix and b an n-dimensional vector; the **quadratic form** of A is the scalar x such that:

$x = b'Ab$

For instance:

example m.33:

Let $A = \begin{pmatrix} 4 & -3 \\ 2 & -1 \end{pmatrix}$ and $b = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$

Then the quadratic form of A and b is given by:

$x = b'Ab = \begin{pmatrix} -2 & 1 \end{pmatrix} \begin{pmatrix} 4 & -3 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} -2 \\ 1 \end{pmatrix} = 17$

It is then possible to define positive definiteness:

definition m.28: let A be a $n \times n$ symmetric matrix; A is **positive definite** if for any non-zero n-dimensional vector b the quadratic form $x = b'Ab$ is strictly positive (that is, $x = b'Ab > 0$). A is **positive semi-definite** if instead $b'Ab \geq 0$ for any non-zero n-dimensional vector b.

For instance:

example m.34:

Let $A = \begin{pmatrix} 4 & -2 \\ -2 & 3 \end{pmatrix}$

Then for any vector $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, the quadratic form of A and b is given by:

$$x = b'Ab = \begin{pmatrix} b_1 & b_2 \end{pmatrix} \begin{pmatrix} 4 & -2 \\ -2 & 3 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = 2b_1^2 + 2(b_1 - b_2)^2 + b_2^2$$

The quadratic form involves only positive combinations of square terms, and is thus strictly positive for any non-zero vector b. Therefore, A is positive definite.

The notion of positive definiteness is important in statistics because variance-covariance matrices are always positive definite. As a consequence, certain statistical distributions used to generate variance-covariance matrices like the inverse Wishart distribution involve positive definite matrices both for their parameters and generated values.

At the opposite of positive definiteness is the notion of negative definiteness.

definition m.29: let A be a $n \times n$ symmetric matrix; A is **negative definite** if for any non-zero n-dimensional vector b the quadratic form $x = b'Ab$ is strictly negative (that is, $x = b'Ab < 0$). A is **negative semi-definite** if instead $b'Ab \leq 0$ for any non-zero n-dimensional vector b.

A matrix which is neither positive definite nor negative definite is said to be indefinite.

There exists a close relation between definiteness and the eigenvalues of a matrix, as stated by the next result.

property m.62: let A be a $n \times n$ symmetric matrix; A is positive definite (respectively positive semi-definite) if all its eigenvalues are positive (respectively non-negative).

property m.63: let A be a $n \times n$ symmetric matrix; A is negative definite (respectively negative semi-definite) if all its eigenvalues are negative (respectively non-positve).

m.14 Partitioned matrices

Partitioned matrices provide a convenient representation. They are defined as follows.

definition m.30: a **partitioned matrix** is a matrix that has been partitioned into a set of submatrices by indicating subgroups (or "blocks") of rows and or columns.

For instance:

example m.35:

$$\text{Let } A = \left(\begin{array}{ccc|c} 2 & -4 & 0 & 3 \\ 1 & 2 & -5 & 7 \\ \hline -1 & 0 & 1 & 6 \end{array}\right) = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right)$$

with:

$$A_{11} = \begin{pmatrix} 2 & -4 & 0 \\ 1 & 2 & -5 \end{pmatrix} \qquad A_{12} = \begin{pmatrix} 3 \\ 7 \end{pmatrix} \qquad A_{21} = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \qquad A_{22} = \begin{pmatrix} 6 \end{pmatrix}$$

It is worth noting that the partitioning is only an interpretation, or a visualization of the original matrix. The representation may prove however quite convenient because the usual rules of matrix addition and multiplication directly apply to partitioned matrices, provided the submatrices are of appropriate dimensions. This is stated in the next properties.

property m.64: let A and B be some matrices partitioned as:

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) \qquad B = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right)$$

Then, provided the dimensions of the submatrices agree, the addition is defined as:

$$A + B = \left(\begin{array}{c|c} A_{11} + B_{11} & A_{12} + B_{12} \\ \hline A_{21} + B_{21} & A_{22} + B_{22} \end{array}\right)$$

property m.65: let A and B be some matrices partitioned as:

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) \qquad B = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right)$$

Then, provided the dimensions of the submatrices agree, the mutiplication is defined as:

$$A \times B = \left(\begin{array}{c|c} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ \hline A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{array}\right)$$

The logic is similar if the matrix is partitioned into more blocks. In general, matrix partitionning can significantly improve the visualisation of large block matrices, as well as their interractions with other large matrices.

m.15 Matrix derivatives

The calculation of maximum likelihood estimates often requires the derivatives of expressions involving matrices. The computation of such matrix derivatives is non-trivial and typically involves entrywise developments. We take a shortcut here and present only the final results. Most properties introduced in this section can be found in Dhrymes (2013).

property m.66: let A be some matrix and x be some vector such that Ax is defined.

Then $\dfrac{\partial Ax}{\partial x} = A$

property m.67: let A be some matrix and z and x be some vector such that $z'Ax$ is defined.

Then $\dfrac{\partial z'Ax}{\partial z} = x'A'$

property m.68: let A be some matrix and z and x be some vector such that $z'Ax$ is defined.
Then $\frac{\partial z'Ax}{\partial x} = z'A$

property m.69: let A be some matrix and x be some vector such that $x'Ax$ is defined.
Then $\frac{\partial x'Ax}{\partial x} = x'(A + A')$

property m.70: let A be some square matrix matrix with positive determinant.
Then $\frac{\partial log|A|}{\partial A} = (A^{-1})'$

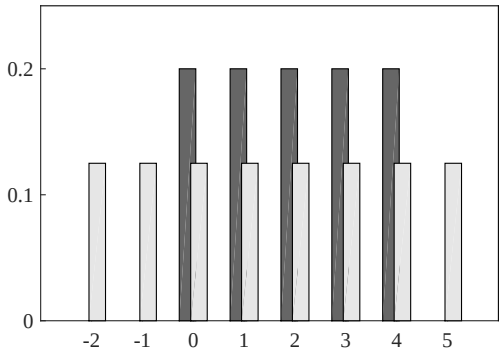
property m.71: let A be some square and invertible matrix matrix.
Then $\frac{\partial A^{-1}}{\partial A} = A^{-1} \times A^{-1}$

property m.72: let A, X and B be some matrice such that AXB is defined.
Then $\frac{\partial tr(AXB)}{\partial X} = A'B'$

CHAPTER D

Statistical distributions

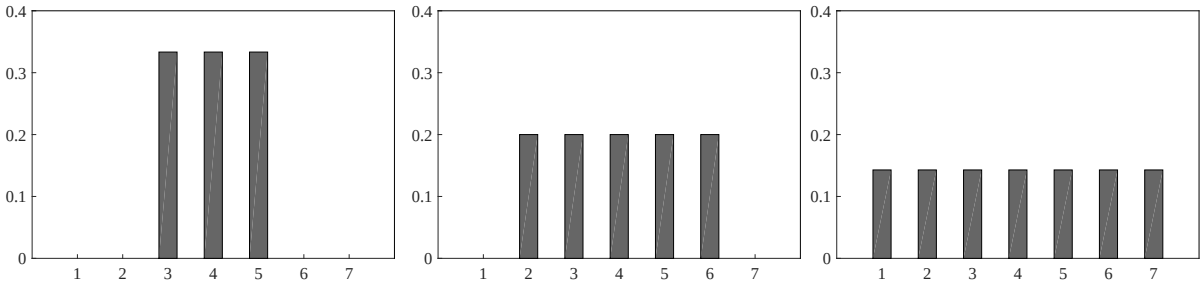
d.1. Discrete uniform



Type:	discrete
Notation:	$x \sim U(a,b)$
Parameters:	a (integer, lower bound of the support) b (integer, upper bound of the support, $b > a$)
Support:	$x \in \{a, a+1, \ldots, b-1, b\}$
pmf:	$f(x\|a,b) = \frac{1}{k}$ $\quad k = b-a+1$ (number of outcomes)
Kernel:	$f(x\|a,b) \propto \frac{1}{k}$
Normalizing constant:	$c = 1$
Mean:	$\frac{a+b}{2}$
Variance:	$\frac{k^2-1}{12}$
Median:	$\frac{a+b}{2}$
Mode:	any $x \in \{a, a+1, \ldots, b-1, b\}$
Diffuse distribution:	set $a \to -\infty$ and $b \to \infty$
Related distributions:	–

Table d.1: Summary of the Discrete uniform distribution

The uniform distribution represents one of the simplest discrete distributions. It is used in the case of experiments with k outcomes, all equally likely. This includes for instance the outcome of a fair die roll, or the number obtained from a roulette game. The distribution is straightforward: the mean is found halfway of the support, and the variance increases with the spread of the distribution, i.e. the number of outcomes. This is illustrated by Figure d.1.



(a) k=3, mean=4, variance=2/3 (b) k=5, mean=4, variance=2 (c) k=7, mean=4, variance=4

Figure d.1: Variance of discrete uniform distributions

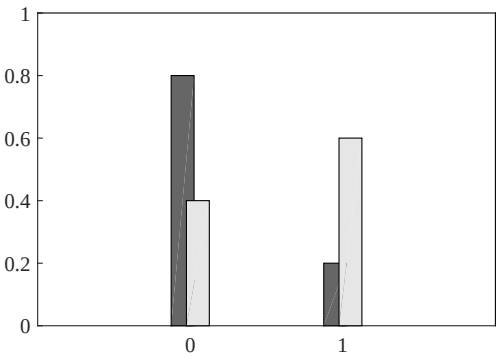
Generating pseudo random numbers from the discrete uniform distribution is easy, as long as one can create pseudo random numbers from the continuous uniform distribution. This is illustrated by the following algorithm.

algorithm d.1: random number generator for the discrete uniform distribution

1. draw a random number u from the continuous uniform distribution: $u \sim U(0,1)$.
2. set $x = \lfloor a + (b+1-a)u \rfloor$, where $\lfloor . \rfloor$ denotes the integer part of x.

Then x is a random draw from $x \sim U(a,b)$.

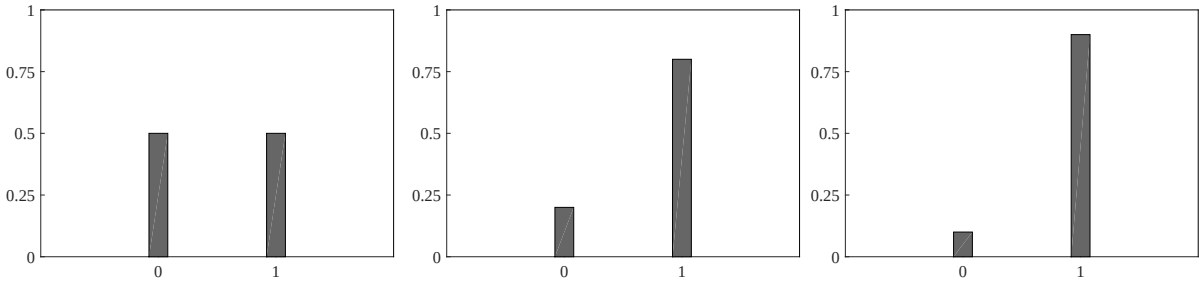
d.2. Bernoulli



Type:	discrete
Notation:	$x \sim Bern(p)$
Parameters:	p (success probability, $0 \leq p \leq 1$)
Support:	$x \in \{0,1\}$
pmf:	$f(x\|p) = p^x(1-p)^{1-x}$
Kernel:	$f(x\|p) \propto p^x(1-p)^{1-x}$
Normalizing constant:	$c = 1$
Mean:	p
Variance:	$p(1-p)$
Median:	$\begin{cases} 0, & \text{if } p \leq 0.5 \\ 1, & \text{if } p > 0.5 \end{cases}$
Mode:	$\begin{cases} 0, & \text{if } p \leq 0.5 \\ 1, & \text{if } p > 0.5 \end{cases}$
Diffuse distribution:	$f(x\|p) \propto 1$
Related distributions:	Discrete uniform: if $x_1 \sim Bern(0.5)$, then $x_1 \sim U(0,1)$ Binomial: if $x_1, x_2, \ldots, x_n$ are $i.i.d \sim Bern(p)$, then $\sum_{i=1}^{n} x_i \sim Bin(p,n)$

Table d.2: Summary of the Bernoulli distribution

The Bernoulli distribution is used in situations where the considered random experiment can only produce two outcomes. Typical examples are the outcome of a coin flip (heads or tails), the gender of a baby (male or female), the success at an exam (pass or fail), and so on. The outcomes are labelled as "success", in which case the variable takes the value of 1, or "failure", in which case the variable takes the value of 0. The probability of success is given by p, implying that the probability of failure is $1-p$. The parameter also determines the variance of the distribution, the maximum variance being reached when $p = 0.5$, and declining as p approaches 0 or 1. This is illustrated by Figure d.2:



(a) p=0.5, variance=0.25 (b) p=0.2, variance=0.16 (c) p=0.1, variance=0.09

Figure d.2: Variance of Bernoulli distributions

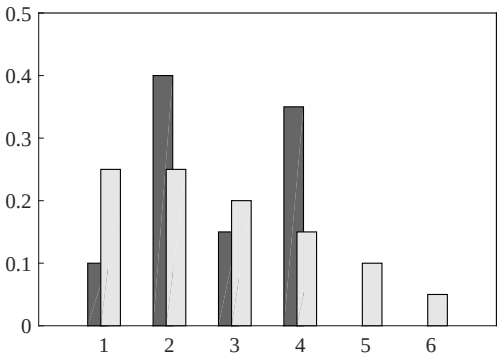
The following algorithm can be used to generate pseudo random numbers from the Bernoulli distribution.

algorithm d.2: random number generator for the Bernoulli distribution

1. draw a random number u from the continuous uniform distribution: $u \sim U(0,1)$.
2. if $u \leq p$, set $x = 1$; otherwise, set $x = 0$.

Then x is a random draw from $x \sim Bern(p)$.

d.3. Categorical



Type:	discrete
Notation:	$x \sim Cat(p_1, p_2, \ldots, p_k)$
Parameters:	$p_1, p_2, \ldots, p_k$ (outcome probabilities, $p_i > 0 \; \forall i = 1, 2, \ldots, k$, and $\sum_{i=1}^{k} p_i = 1$)
Support:	$x \in \{1, 2, \ldots, k\}$
pmf:	$f(x\|p_1, p_2, \ldots, p_k) = \prod_{i=1}^{k} p_i^{\mathbb{1}(x=i)}$ $\mathbb{1}(.)$ denotes the indicator function
Kernel:	$f(x\|p_1, p_2, \ldots, p_k) \propto \prod_{i=1}^{k} p_i^{\mathbb{1}(x=i)}$
Normalizing constant:	$c = 1$
Mean:	$\sum_{i=1}^{k} i p_i$
Variance:	$\sum_{i=1}^{k} i^2 p_i - (\sum_{i=1}^{k} i p_i)^2$
Median:	i such that $\sum_{j=1}^{i-1} p_j \leq 0.5$ and $\sum_{j=1}^{i} p_j \geq 0.5$
Mode:	i such that $p_i = max(p_1, p_2, \ldots, p_k)$
Diffuse distribution:	$f(x\|p_1, p_2, \ldots, p_k) \propto 1$
Related distributions:	Uniform: if $x \sim Cat(\frac{1}{k}, \ldots, \frac{1}{k})$, then: $x \sim U(1, k)$ Bernoulli: if $x \sim Cat(p_1, p_2)$, then $x \sim Bern(p_1)$ Multinomial: if $x_1, x_2, \ldots, x_n$ are $i.i.d \sim Cat(p_1, p_2, \ldots, p_k)$, then: $\sum_{i=1}^{n} x_i \sim Mun(p_1, p_2, \ldots, p_k, n)$

Table d.3: Summary of the Categorical distribution

The categorical distribution represents a generalization of the Bernoulli distribution. While Bernoulli outcomes are restricted to be binary, the categorical distribution expands the number of possible outcomes to k. Typical applications are the outcome of rolling a 6-face die, or the mark obtained at an exam (A, B, C, D, E or F). The different outcomes are labelled as $1, 2, \ldots, k$, the numbers representing the different categories.

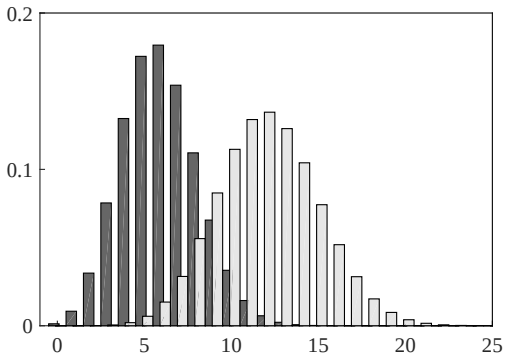
Pseudo random numbers from the categorical distribution can be easily generated from the following algorithm.

algorithm d.3: random number generator for the categorical distribution

1. draw a random number u from the continuous uniform distribution: $u \sim U(0,1)$.
2. if $u \leq p_1$, set $x = 1$; if $p_1 < u \leq p_1 + p_2$, set $x = 2$, and so on. In general:
   if $\sum_{j=1}^{i-1} p_j \leq u \leq \sum_{j=1}^{i} p_j$, set $x = i$, for $i = 1, \ldots, k$.

Then x is a random draw from $x \sim Cat(p_1, p_2, \ldots, p_k)$.

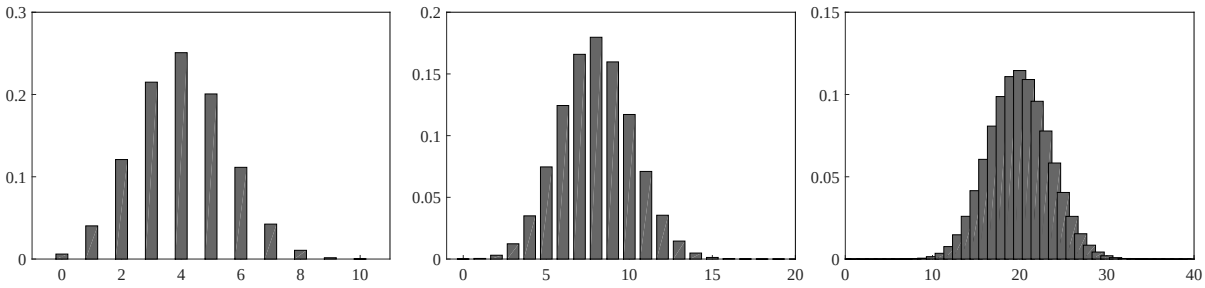
d.4. Binomial



Type:	discrete
Notation:	$x \sim Bin(n,p)$
Parameters:	n (integer, number of trials) p (probability of success for each trial, $0 \leq p \leq 1$)
Support:	$x \in \{1,\ldots,n\}$
pmf:	$f(x\|n,p) = \binom{n}{x} p^x (1-p)^{n-x}$ $\quad$ $\binom{n}{x} = \frac{n!}{x!(n-x)!}$
Kernel:	$f(x\|n,p) \propto \frac{1}{x!(n-x)!} p^x (1-p)^{n-x}$
Normalizing constant:	$c = n!$
Mean:	np
Variance:	$np(1-p)$
Median:	$\lfloor np \rfloor$ $\quad$ $\lfloor . \rfloor$ denotes the floor function
Mode:	$\lfloor (n+1)p \rfloor$
Diffuse distribution:	$f(x\|n,p) \propto 1$
Related distributions:	Bernoulli: if $x \sim Bin(1,p)$, then $x \sim Bern(p)$ Normal: if $x \sim Bin(n,p)$ with $np \geq 5$ and $n(1-p) \geq 5$, then approximately: $x \sim N(np, np(1-p))$ Poisson: if $x \sim Bin(n,p)$ with $n \geq 100$ and $np \leq 10$, then approximately: $x \sim Pois(np)$

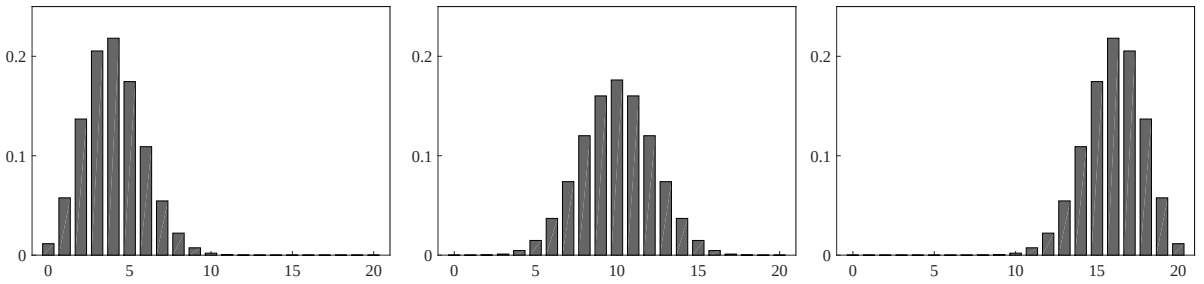
Table d.4: Summary of the Binomial distribution

The Binomial distribution considers the number of succesful outcomes from n independent Bernoulli experiments. In this respect it is closely related to the Bernoulli distribution, and the success probability p of the Bernoulli distribution determines the mass function of the Binomial distribution, along with its moments. There are two characteristic features of the binomial distribution. First, the mean and the variance of the distribution are increasing with the number of trials n. Second, the skewness of the distribution is determined by the probability of success p: values lower than 0.5 generate positive skewness, while values greater than 0.5 imply negative skewness, the distribution being symmetric at $p = 0.5$. This is illustrated by Figures d.3 and d.4:



(a) $n = 10$,mean=4, variance=2.4 (b) $n = 20$,mean=8, variance=4.8 (c) $n = 40$,mean=16, variance=9.6

Figure d.3: Mean and variance of Binomial distributions ($p = 0.4$)



(a) $p = 0.2$ (b) $p = 0.5$ (c) $p = 0.8$

Figure d.4: Skewness of Binomial distributions ($n = 20$)

Finally, the Binomial distribution is related to both the normal and Poisson distributions when the number of trials n becomes large enough. When the probability of success p is sufficiently close to 0.5, the Binomial distribution provides a discrete approximation to the normal distribution, while if p is sufficiently small, the Binomial distribution approximates the Poisson distribution.

As a conclusion to this section, the following algorithm introduces the procedure to generate pseudo random numbers from the Binomial distribution. It makes direct use of the definition of the binomial distribution as the sum of n independent Bernoulli trials:

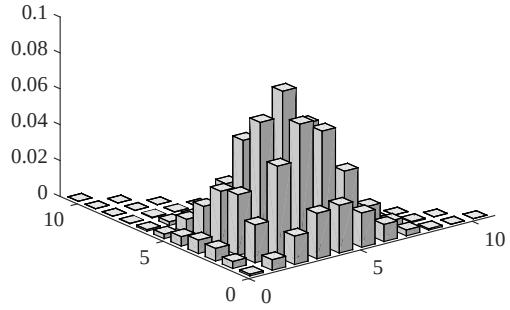
algorithm d.4: random number generator for the Binomial distribution

1. draw independently n numbers $x_1, \ldots, x_n$ from: $x_i \sim Bern(p)$, $i = 1, \ldots, n$.

2. set $x = x_1 + \ldots + x_n$.

Then x is a random draw from $x \sim Bin(n, p)$.

d.5. Multinomial

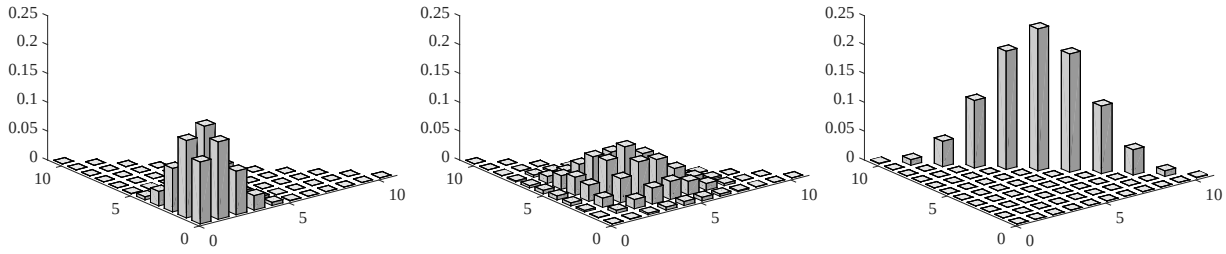


Type:	discrete
Notation:	$x \sim Mun(n, p_1, \ldots, p_k)$
Parameters:	n (integer, number of trials) $p_1, \ldots, p_k$ (outcome probabilities, $p_i > 0 \; \forall i = 1, 2, \ldots, k$, and $\sum_{i=1}^{k} p_i = 1$)
Support:	$x_1, \ldots, x_k \in \{1, \ldots, n\}$, with $\sum_{i=1}^{k} x_i = n$
pmf:	$f(x_1, \ldots, x_k \mid n, p_1, \ldots, p_k) = \frac{n!}{x_1! \ldots x_k!} p_1^{x_1} \ldots p_k^{x_k}$
Kernel:	$f(x_1, \ldots, x_k \mid n, p_1, \ldots, p_k) \propto \frac{1}{x_1! \ldots x_k!} p_1^{x_1} \ldots p_k^{x_k}$
Normalizing constant:	$c = n!$
Mean:	$\mathbb{E}(x_i) = np_i$
Variance:	$Var(x_i) = np_i(1 - p_i)$, $Cov(x_i, x_j) = -np_i p_j$
Median:	$\lfloor np_i \rfloor$ for variable x_i $\quad$ $\lfloor . \rfloor$ denotes the floor function
Mode:	$\lfloor (n+1)p_i \rfloor$ for variable x_i
Diffuse distribution:	$f(x_1, \ldots, x_k \mid n, p_1, \ldots, p_k) \propto 1$
Related distributions:	Binomial: if $x \sim Mun(n, p_1, p_2)$, then $x \sim Bin(n, p_1)$ Categorical: if $x \sim Mun(1, p_1, \ldots, p_k)$, then $x \sim Cat(p_1, \ldots, p_k)$

Table d.5: Summary of the Multinomial distribution

The multinomial distribution generalizes the Binomial distribution to the case of experiments with k possible outcomes. Concretely, the multinomial distribution considers the outcome of repeated categorical experiments, much the same way the Binomial considers the outcome of repeated Bernoulli experiments.

The marginal distributions of the k variables $x_1,\ldots,x_k$ are Binomial: $x_i \sim Bin(n,p_i)$. Following, all the properties of the Binomial distribution apply to the individual variables x_i, including the fact that the mean and variance increase with the number of experiments n. On the other hand, the covariance between any two variables x_i and x_j is always negative, and the correlation tends to -1 as the success probabilities of variables p_i approaches $1-p_j$. This is illustrated by Figure d.5:



(a) p_1, p_2=0.1, $Corr(x_1, x_2 = -0.11)$ (b) p_1, p_2=0.3, $Corr(x_1, x_2 = -0.43)$ (c) p_1, p_2=0.5, $Corr(x_1, x_2 = -1)$

Figure d.5: Correlation of Multinomial distributions ($k = 3, n = 10$)

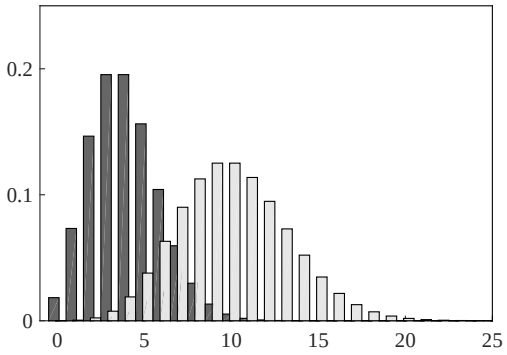
It is easy to generate pseudo random numbers from the Multinomial distribution, making direct use of the definition of the multinomial distribution as the sum of n independent categorical trials.

algorithm d.5: random number generator for the Multinomial distribution

1. generate n numbers $z_1,\ldots,z_n$ from: $z_j \sim Cat(p_1,\ldots,p_k)$, $j = 1,\ldots,n$.
2. for $i = 1,\ldots,k$, set x_i as the number of times z_j was equal to i, that is, the number of times z_j was a success for category i.

Then $x = (x_1,\ldots,x_k)$ is a random draw from $x \sim Mun(n, p_1,\ldots,p_k)$.

d.6. Poisson

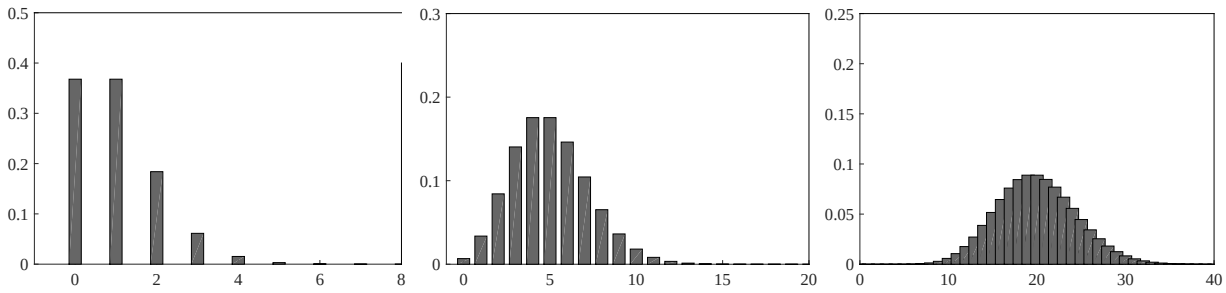


Type:	discrete
Notation:	$x \sim Pois(\lambda)$
Parameters:	λ (intensity parameter, scalar with $\lambda > 0$)
Support:	$x \in \mathbb{Z}^*$ $\quad$ $\mathbb{Z}^* = \{0, 1, 2, \cdots\}$, the set of non-negative integers
pmf:	$f(x\|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$
Kernel:	$f(x\|\lambda) \propto \frac{\lambda^x}{x!}$
Normalizing constant:	$c = e^{-\lambda}$
Mean:	λ
Variance:	λ
Median:	$\approx \left\lfloor \lambda + \frac{1}{3} - \frac{0.02}{\lambda} \right\rfloor$ $\quad$ $\lfloor . \rfloor$ denotes the floor function
Mode:	$\lfloor \lambda \rfloor$
Diffuse distribution:	set $\lambda \to \infty$
Related distributions:	Normal: if $x \sim Pois(\lambda)$ with $\lambda \geq 1000$, then approximately $x \sim N(\lambda, \lambda)$

Table d.6: Summary of the Poisson distribution

The Poisson distribution considers the number of occurrences of a given event in a specified interval of time or distance. Because the number of occurrences is typically assmed to be small, the Poisson distribution is sometimes refered to as the "law of small numbers". The occurrences are also independent, namely, the occurrence of one event does not affect the probability of occurrence of a second event. Typical examples are the number of calls reaching a call center in a minute, or the number of car accidents over a 200 kilometers portion of highway.

The distribution is characterized by a unique intensity parameter λ which represents both the mean and the variance of the distribution. Following, rare events (small values of λ) consistently result in small number of occurrences, while more likely events (large values of λ) are characterized by more variability and allow for both small and large numbers of occurrences. This is represented by Figure d.6:



(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 20$

Figure d.6: Mean and variance of Poisson distributions

If λ is small enough, one can generate pseudo-random Poisson numbers easily with the following algorithm.

algorithm d.6: random number generator for the Poisson distribution, $\lambda \leq 30$

1. set $p = F = e^{-\lambda}$, and $z = 0$.
2. draw a random number u from $u \sim U(0,1)$.
3. if $u > F$: set $z = z+1$, $p = \frac{\lambda p}{z}$, $F = F + p$ and repeat step 3;
   else, if $u \leq F$, set $x = z$.

Then x is a random draw from $x \sim Pois(\lambda)$.

This algorithm is based upon inversion by sequential search. It is algorithm 3 in Kemp and Kemp (1991). It is fast for small values of λ. However the number of computations increases linearly with λ, so that when λ becomes large the algorithm becomes excessively slow. It also become numerically unstable because of the very small value of the $e^{-\lambda}$ term. For large values, a better alternative is algorithm 8 in Kemp and Kemp (1991). It uses a unidirectional search from the mode and looks considerably more complex. However, the different steps all rely on basic operations, which keeps the algorithm fast.

algorithm d.7: random number generator for the Poisson distribution, $\lambda > 30$

preliminary phase

1. set the first series of preliminary functions:
$q_r(\lambda) = (2\pi\lambda)^{-1/2}\left(1 - \frac{1}{12\lambda + \frac{1}{2} + \frac{293}{720\lambda}}\right)$
$G_r(\lambda) = \frac{1}{2} + \frac{2}{3}(2\pi\lambda)^{-1/2}\left(1 - \frac{\frac{23}{15}}{12\lambda + \frac{15}{14} + \frac{\frac{30557}{4508}}{12\lambda + \frac{138134432}{105880005}}}\right)$

2. decompose λ into $\lambda = \bar{\lambda} + \alpha$, where $\bar{\lambda}$ is an integer and $-0.5 \leq \alpha \leq 0.5$, so that $\bar{\lambda} = \lfloor \lambda \rfloor$ when $\alpha \geq 0$, and $\bar{\lambda} = \lfloor \lambda \rfloor + 1$ when $\alpha < 0$. This implies: $\bar{\lambda} = \lfloor \lambda + 0.5 \rfloor$ and $\alpha = \lambda - \bar{\lambda}$. Also, set $c = (2\pi\bar{\lambda})^{-1/2}$.

3. set the second series of preliminary functions:
$p_r(\bar{\lambda}, \alpha) = q_r(\bar{\lambda})\left(\frac{\bar{\lambda} + \frac{2\alpha}{3} - \frac{\alpha^2}{4} - \frac{\alpha^2}{18\bar{\lambda}}}{\bar{\lambda} + \frac{2\alpha}{3} + \frac{\alpha^2}{4} - \frac{\alpha^2}{18\bar{\lambda}}}\right)$
$F_r(\bar{\lambda}, \alpha) = G_r(\bar{\lambda}) - \alpha q_r(\bar{\lambda})\left(\frac{\bar{\lambda} + \frac{\alpha}{2} - \frac{\alpha^2}{60} - \frac{\alpha^2}{20\bar{\lambda}}}{\bar{\lambda} + \frac{\alpha}{2} + \frac{3\alpha^2}{20} - \frac{\alpha^2}{20\bar{\lambda}}}\right)$
It should be clear that when λ is an integer, $\lambda = \bar{\lambda}$, $\alpha = 0$ and consequently $p_r(\bar{\lambda}, \alpha) = q_r(\lambda)$ and $F_r(\bar{\lambda}, \alpha) = G_r(\lambda)$.

4. calculate $p_r(\bar{\lambda}, \alpha)$.

5. draw a random number u from $u \sim U(0,1)$.

squeeze phase

6. if $u \leq 0.5$, go directly to step 9; else, if $u \geq 0.5 + \frac{7c}{6}$, go directly to step 12; else:

7. calculate $F_r(\bar{\lambda}, \alpha)$.

8. if $u > F_r(\bar{\lambda}, \alpha)$, go directly to step 12; else:

downward search phase

9. if $u < p_r(\bar{\lambda}, \alpha)$, set $x = \bar{\lambda}$ and stop; else:

10. set $p = p_r(\bar{\lambda}, \alpha)$.

11. for $i = 0$ to $\bar{\lambda} - 1$:
set $u = u - p$, and $p = \frac{(\bar{\lambda} - i)p}{\lambda}$;
if $u < p$, set $x = \bar{\lambda} - i - 1$ and stop.

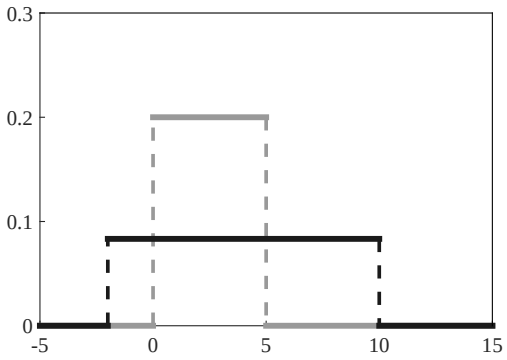
upward search phase

12. set $u = 1 - u$, and $p = p_r(\bar{\lambda}, \alpha)$.

13. for $i = \bar{\lambda} + 1$ to λ_{max} :
set $p = \frac{p\lambda}{i}$.
if $u < p$, set $x = i$ and stop.
set $u = u - p$.
λ_{max} is an upper bound factor for the upward search. Following the recommendations of Kemp and Kemp (1991), λ_{max} is set to $2\bar{\lambda} + 30$.

Then x is a random draw from $x \sim Pois(\lambda)$.

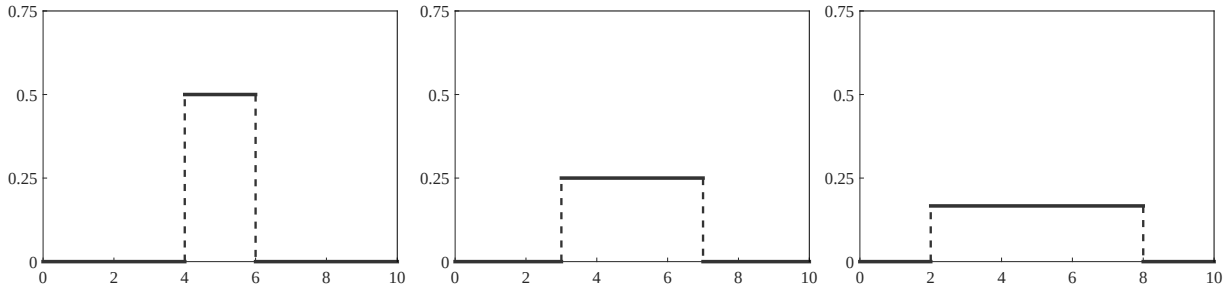
d.7. Uniform



Type:	continuous
Notation:	$x \sim U(a,b)$
Parameters:	a (scalar, lower bound of the support) b (scalar, upper bound of the support, $b > a$)
Support:	$x \in [a,b]$
pdf:	$f(x\|a,b) = \frac{1}{b-a}$
Kernel:	$f(x\|a,b) \propto 1$
Normalizing constant:	$c = \frac{1}{b-a}$
Mean:	$\frac{a+b}{2}$
Variance:	$\frac{(b-a)^2}{12}$
Median:	$\frac{a+b}{2}$
Mode:	any $x \in [a,b]$
Diffuse distribution:	set $a \to -\infty$ and $b \to \infty$
Related distributions:	–

Table d.7: Summary of the Uniform distribution

The uniform distribution is the continuous counterpart to the discrete uniform distribution. It assumes constant probability over its support, the closed interval $[a,b]$. The distribution is straightforward: the mean and median are found half-way of the support, and the variance of the distribution increases as the support enlarges, as illustrated by Figure d.7:



(a) Variance= $\frac{1}{3}$ (b) Variance= $\frac{4}{3}$ (c) Variance= $\frac{9}{3}$

Figure d.7: Variance of the uniform distribution (mean=5)

In practical applications, the uniform distribution is often used whenever one wants to remain agnostic about the outcome of an experiment. This is reflected by the flat probability attributed to all the possible outcomes. For instance, a uniform distribution over $[0,1]$ can be used to model an agnostic belief about the probability that a coin yields "heads".

The uniform distribution also represents one of the most important statistical distributions because it constitutes the basis of virtually every algorithm used to generate random numbers from any other distribution. It is thus crucial to generate uniform numbers efficiently.

For years, a class of algorithms known as the linear congruential generator algorithms were used. Those algorithms are easy to understand and run fast, but they repeat themselves after a given period, and it is possible to show that the numbers that are produced are not effectively random, but lie on a finite number of hyperplanes. For these reasons, more efficient algorithms have been developed. The current standard is an algorithm known as the Mersenne twister, developed by Matsumoto and Nishimura (1998). In comparison with the linear congruential generator algorithms, the Mersenne twister benefits from longer periods, and the numbers produced offer better randomness properties. The algorithm is fairly complicated and it thus not introduced in details here. Most mathematical software applications like Matlab, R or NumPy integrate built-in functions for this algorithm.

To provide an intuition of how uniform numbers can be easily generated, the linear congruential generator algorithm is introduced. This is only for the sake of pedagogy, as again the Mersenne twister represents a better alternative.

algorithm d.8: random number generator for the uniform distribution (linear congruential generator)

1. set the following integer values:
   m, with $m > 0$: the modulus
   a, with $0 < a < m$: the multiplier
   c, with $0 \leq c < m$: the increment
   x_0, with $0 \leq x_0 < m$: the seed or initial value

2. generate any quantity of random numbers from the following recurrence relation:
   $x_n = (ax_{n-1} + c) \ mod \ m$
   where "$(ax_{n-1} + c) \ mod \ m$" means: "divide $(ax_{n-1} + c)$ by m, and take the remainder".

Then $x_1, x_2, \ldots$ are random draws from $x \sim U(0,1)$.

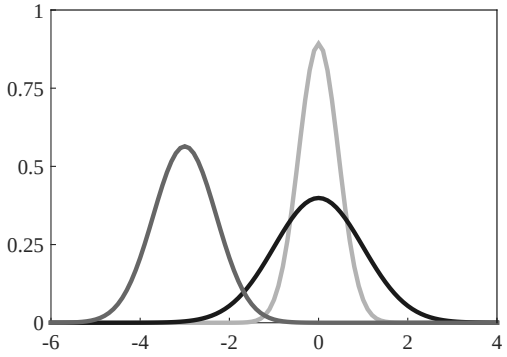
The preceding algorithm produces random numbers from $x \sim U(0,1)$. It is then trivial to use those numbers to generate random numbers from a general distribution $x \sim U(a,b)$, using the following algorithm:

algorithm d.9: random number generator for the uniform distribution

1. draw a random number u from $u \sim U(0,1)$.
2. set $x = a + u(b-a)$.

Then x is a random draw from $x \sim U(a,b)$.

d.8. Normal

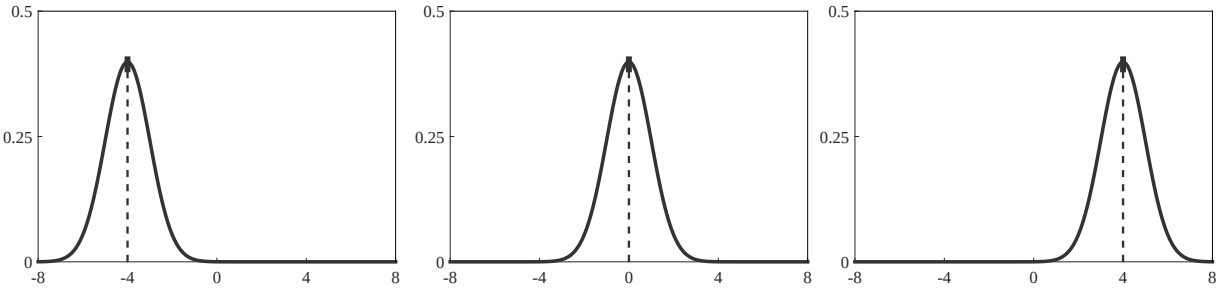


Type:	continuous
Notation:	$x \sim N(\mu, \sigma)$
Parameters:	μ (mean, scalar) σ (variance, scalar with $\sigma > 0$)
Support:	$x \in \mathbb{R}$
pdf:	$f(x\|\mu,\sigma) = (2\pi\sigma)^{-1/2} exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma}\right)$
Kernel:	$f(x\|\mu,\sigma) \propto exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma}\right)$
Normalizing constant:	$c = (2\pi\sigma)^{-1/2}$
Mean:	μ
Variance:	σ
Median:	μ
Mode:	μ
Diffuse distribution:	set $\mu = 0$ and $\sigma \to \infty$ (proper distribution) or set $f(x\|\mu,\sigma) \propto 1$ (improper distribution)
Related distributions:	–

Table d.8: Summary of the normal distribution

The normal distribution represents by far the most important distribution in statistics. One reason is the remarkable result known as the central limit theorem. Another reason is the shape of the distribution which makes it an attractive candidate for many types of random variables. First, the support of the distribution ranges from $-\infty$ to $+\infty$, making the normal distribution suitable for random variables taking any real value. Second, the bell shape of the density function implies that the bulk of probabilities are concentrated around the mean, making extreme events (values far away from the mean) unlikely. Finally, the distribution is symmetric around the mean, reflecting the fact that many random experiments behave similarly on both sides of the mean. Following, the normal distribution can be used for a wide range of real-life phenomena. Typical examples are the distribution of adult heights, the distribution of marks on tests, or stock market returns.

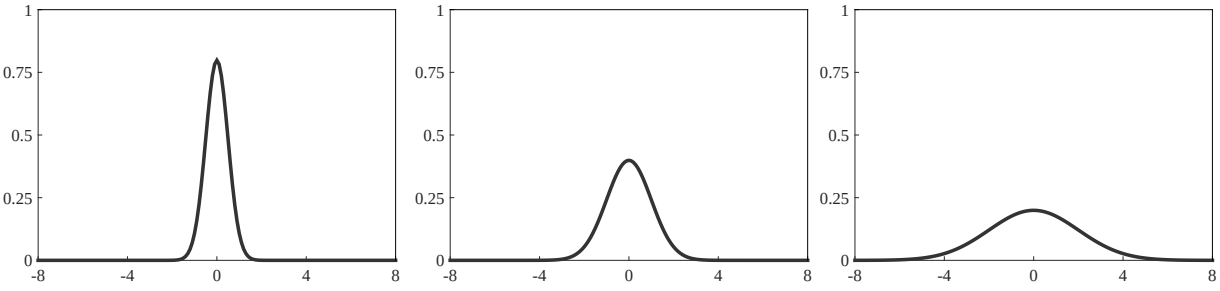
The normal distribution is characterised by two parameters. The first is the mean parameter μ. By changing μ, the distribution shifts rightward or leftward, without affecting the general shape, as shown by Figure d.8:



(a) mean: $\mu = -4$ (b) mean: $\mu = 0$ (c) mean: $\mu = 4$

Figure d.8: Mean of the normal distribution (unit variance)

The second parameter is the variance σ. Unlike most textbooks, this manual uses σ and not σ^2 to denote the variance. This makes notations more consistent with other distributions (in particular the other normal and student distributions), and avoid the ambiguity of treating the square superscript as a notation or as an actual mathematical operator. For a given mean μ, larger values of σ increase the spread and the flatness of the distribution, resulting in higher variance. This is illustrated by Figure d.9:



(a) variance: $\sigma = 0.25$ (b) variance: $\sigma = 1$ (c) variance: $\sigma = 4$

Figure d.9: Variance of the normal distribution (zero mean)

The special case where the mean μ is equal to 0 and the variance σ is equal to 1 is known as the standard normal distribution. From the standard normal distribution, is easy to construct a normal distribution with arbitrary mean and variance by using the following property, known as the affine property of the normal distribution:

property d.1: let x be a normally distributed random variable: $x \sim N(\mu, \sigma)$, and let $y = ax + b$. Then: $y \sim N(a\mu + b, a^2\sigma)$.

A plethora of different algorithms is available to generate pseudo random normal numbers. In practice, it is not necessary to code any of them since all mathematical softwares are equipped with built-in functions to generate random normal numbers, though different softwares use different methods. For instance, Matlab uses the Ziggurat Method introduced by Marsaglia and Tsang (2000b), NumPy uses the Box-Muller approach from Box and Muller (1958), and R implements an inversion procedure identifying cumulative densities with Wichura (1988). The algorithm proposed here is simple and relies on the polar method proposed by Marsaglia and Bray (1964).

algorithm d.10: random number generator for the standard normal distribution

1. draw two random numbers u_1 and u_2 from $u_1, u_2 \sim U(-1,1)$.
2. set $u_3 = u_1^2 + u_2^2$; if $u_3 \geq 1$, go back to step 1; else:
3. define: $x_1 = u_1\sqrt{\frac{-2ln(u_3)}{u_3}}$ and $x_2 = u_2\sqrt{\frac{-2ln(u_3)}{u_3}}$

Then x_1 and x_2 are random draws from $x_1, x_2 \sim N(0,1)$.

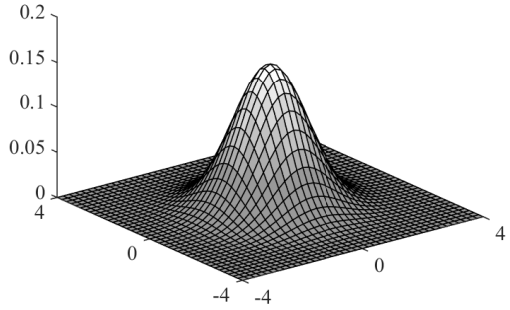
Once one can generate draws from $x \sim N(0,1)$, it becomes easy to generate random numbers from an arbitrary normal distribution $x \sim N(\mu, \sigma)$, using property d.1.

algorithm d.11: random number generator for the normal distribution

1. draw z from the standard normal distribution: $z \sim N(0,1)$.
2. set $x = \sqrt{\sigma}z + \mu$.

Then x is a random draw from $x \sim N(\mu, \sigma)$.

d.9. Multivariate normal

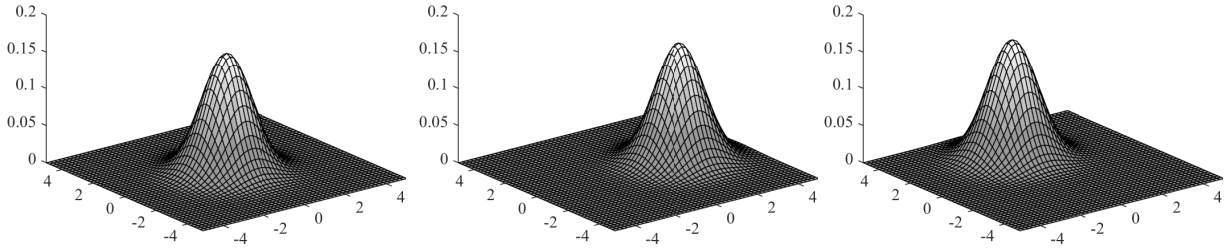


Type:	continuous
Notation:	$x \sim N(\mu, \Sigma)$
Parameters:	μ (n-dimensional mean vector) Σ ($n \times n$ variance-covariance matrix, symmetric and positive definite)
Support:	$x \in \mathbb{R}^n$, the set of $n \times 1$ vectors of real numbers
pdf:	$f(x\|\mu,\Sigma) = (2\pi)^{-n/2}\|\Sigma\|^{-1/2} exp\left(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right)$
Kernel:	$f(x\|\mu,\Sigma) \propto exp\left(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right)$
Normalizing constant:	$c = (2\pi)^{-n/2}\|\Sigma\|^{-1/2}$
Mean:	μ
Variance:	Σ
Median:	μ
Mode:	μ
Diffuse distribution:	set $\mu = 0$ and $\Sigma = \sigma I_n$, with σ a scalar such that $\sigma \to \infty$ (proper distribution) or set $f(x\|\mu,\Sigma) \propto 1$ (improper distribution)
Related distributions:	Normal: if $n = 1$, then $x \sim N(\mu, \sigma)$ (univariate normal)

Table d.9: Summary of the Multivariate normal distribution

The multivariate normal distribution represents a generalization of the one-dimensional normal distribution to n-dimensional random vectors. It is used to model the joint distribution of several normal random variables, possibly correlated. For instance, the height and weight of the adult population of a given country are both approximately distributed, and certainly correlated.

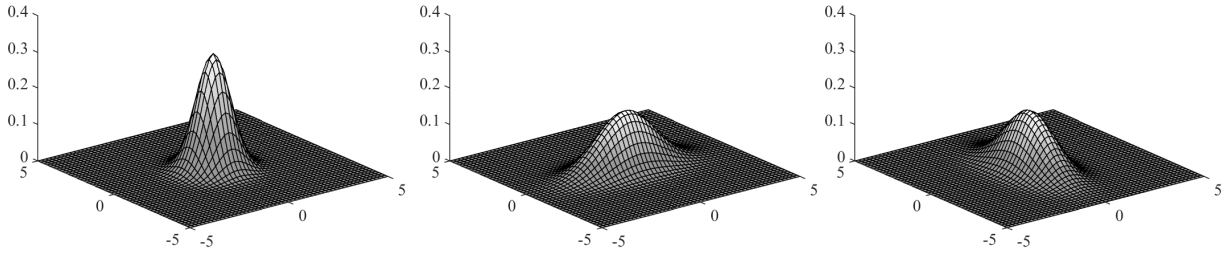
The parallel with the univariate normal distribution is straightforward. The mean of the joint distribution is determined by the n-dimensional vector μ. A change in μ_j (the j^{th} entry of μ) switches the distribution rightward or leftward in the j^{th} dimension, leaving the other dimensions unaffected. This is illustrated by figure d.10.



(a) $\mu_1 = 0, \mu_2 = 0$ (b) $\mu_1 = 2, \mu_2 = 0$ (c) $\mu_1 = 0, \mu_2 = 2$

Figure d.10: Mean of the multivariate normal distribution ($n = 2$, unit variance)

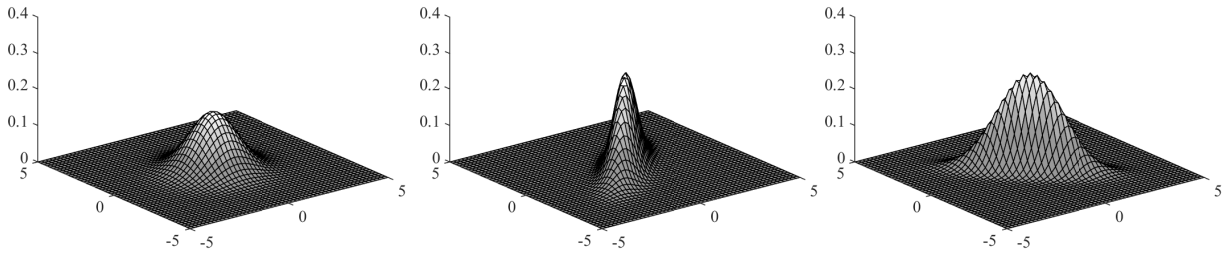
A similar logic applies to the variance of the distribution: an increase in σ_{jj} (the j^{th} diagonal entry of the variance-covariance matrix Σ) results in a larger spread of the distribution in dimension j, leaving the spread of other dimensions unchanged. This is illustrated by Figure d.11:



(a) $\sigma_{11} = 0.5, \sigma_{22} = 0.5$ (b) $\sigma_{11} = 2, \sigma_{22} = 0.5$ (c) $\sigma_{11} = 0.5, \sigma_{22} = 2$

Figure d.11: Variance of the multivariate normal distribution ($n = 2$, mean=0)

A specificity of the multivariate normal compared to the univariate normal is the possible existence of correlation between the different variables. This is defined by the covariance (off-diagonal) entries of Σ. When correlation is positive, the two variables tend to produce similar values and the density function is oriented upward. When correlation is negative, the two variables tend to produce opposite values and the density function is oriented downward. This is illustrated by Figure d.12:



(a) correlation=0 (b) correlation=0.8 (c) correlation=-0.8

Figure d.12: Correlation of the multivariate normal distribution ($n = 2$, mean=0, variance=1)

The multivariate normal distribution has a number of convenient properties. Because this distribution is used extensively in Bayesian analysis, it is useful to detail some of those properties. The first property represents the multivariate generalisation of the affine property of the Normal distribution. It says that linear combinations of normal random variables are also normal. This is stated in the following property.

property d.2: let x be a random variable with: $x \sim N(\mu, \Sigma)$. Let A be some matrix and b be some vector such that $y = Ax + b$ is defined. Then:
$y \sim N(A\mu + b\,,\, A\Sigma A')$

The second property is very useful and relates to the marginal distributions of a multivariate normal distribution. It states that the marginal distributions of a multivariate normal distribution are themselves normal.

property d.3: let x be a random variable with: $x \sim N(\mu, \Sigma)$. Let x, μ and Σ be partitioned the following way:

$$x = \begin{pmatrix} x_1 \\ \hline x_2 \\ \hline \vdots \\ \hline x_p \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \hline \mu_2 \\ \hline \vdots \\ \hline \mu_p \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix} \qquad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \dots & \Sigma_{1p} \\ \hline \Sigma_{21} & \Sigma_{22} & \dots & \Sigma_{2p} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \Sigma_{p1} & \Sigma_{p2} & \dots & \Sigma_{pp} \\ n_1 & n_2 & \dots & n_p \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix}$$

with $n_1 + n_2 + \ldots + n_p = n$.

Then $x_i \sim N(\mu_i, \Sigma_{ii})$, for all $i = 1, 2, \ldots, p$.

A corollary obtains when the partition is realised at the entry level, for then every individual entry of a multivariate normal distribution follows a univariate normal distribution:

property d.4: let x be a random variable with: $x \sim N(\mu, \Sigma)$. Then:
$x_i \sim N(\mu_i, \Sigma_{ii})$, for all $i = 1, 2, \ldots, n$.

The converse is not generally true: random variables which are individually normally distributed are not necessarily jointly normal. This will be the case however if the random variables are independent, and this is stated in the next property.

property d.5: let $x_1, x_2, \ldots, x_p$ be p independent multivariate random variables with:
$x_i \sim N(\mu_i, \Sigma_{ii})$.
Then:
$x \sim N(\mu, \Sigma)$, with:

$$x = \begin{pmatrix} x_1 \\ \hline x_2 \\ \hline \vdots \\ \hline x_p \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \hline \mu_2 \\ \hline \vdots \\ \hline \mu_p \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix} \qquad \Sigma = \begin{pmatrix} \Sigma_{11} & 0 & \dots & 0 \\ \hline 0 & \Sigma_{22} & \dots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \dots & \Sigma_{pp} \\ n_1 & n_2 & \dots & n_p \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix}$$

and $n = n_1 + n_2 + \ldots + n_p$.

The next property is mostly used to prove certain formulas of the Kalman filter:

property d.6: let x be a random variable with: $x \sim N(\mu, \Sigma)$. Let x, μ and Σ be partitioned the following way:

$$x = \begin{pmatrix} x_1 \\ \hline x_2 \end{pmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \hline \mu_2 \end{pmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix} \qquad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \\ n_1 & n_2 \end{pmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix} \qquad n_1 + n_2 = n.$$

Then $x_1 | x_2 \sim N(\hat{\mu}, \hat{\Sigma})$, with $\hat{\mu} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$ and $\hat{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$.

Finally, the following algorithms introduce the procedures to generate pseudo random numbers from the multivariate normal distribution. First, the next algorithm considers the generation of random numbers from the standard multivariate normal distribution.

algorithm d.12: random number generator for the standard multivariate normal distribution

1. draw p random numbers $x_1, \ldots, x_n$ from the standard normal distribution: $x_i \sim N(0,1)$.
2. organise these n values in a n-dimensional column vector x.

Then from property d.5, $x \sim N(0, I_n)$.

The next algorithm develops the procedure to draw from an arbitrary multivariate normal distribution:

algorithm d.13: random number generator for the multivariate normal distribution

1. calculate any matrix G such that $GG' = \Sigma$. In practice, G is often chosen to be the Cholesky factor of Σ.
2. draw a random vector z from the standard multivariate normal distribution: $z \sim N(0, I_n)$.
3. set $x = \mu + Gz$.

Then from property d.2, x is a random draw from $x \sim N(\mu, \Sigma)$.

d.10. Matrix normal

Type:	continuous
Notation:	$X \sim MN(M, \Sigma, \Omega)$
Parameters:	M ($n \times m$ location matrix) Σ ($n \times n$ scale matrix, symmetric and positive definite) Ω ($m \times m$ scale matrix, symmetric and positive definite)
Support:	$X \in \mathbb{R}^{n \times m}$, the set of $n \times m$ matrices of real numbers
pdf:	$f(X\|M,\Sigma,\Omega) = (2\pi)^{-nm/2}\|\Sigma\|^{-m/2}\|\Omega\|^{-n/2} exp\left(-\frac{1}{2} tr\left[\Omega^{-1}(X-M)'\Sigma^{-1}(X-M)\right]\right)$
Kernel:	$f(X\|M,\Sigma,\Omega) \propto exp\left(-\frac{1}{2} tr\left[\Omega^{-1}(X-M)'\Sigma^{-1}(X-M)\right]\right)$
Normalizing constant:	$c = (2\pi)^{-nm/2}\|\Sigma\|^{-m/2}\|\Omega\|^{-n/2}$
Mean:	M
Variance:	$Var(vec(X)) = \Omega \otimes \Sigma$
Median:	M
Mode:	M
Diffuse distribution:	set $M = 0$, $\Sigma = \sigma I_n$ and $\Omega = \omega I_m$, with σ and ω scalars such that $\sigma, \omega \to \infty$ (proper distribution) or set $f(X\|M,\Sigma,\Omega) \propto 1$ (improper distribution)
Related distributions:	Multivariate normal: if $m = 1$, then $x \sim N(\mu, \Sigma)$ Normal: if $n = 1$ and $m = 1$, then $x \sim N(\mu, \sigma)$

Table d.10: Summary of the Matrix normal distribution

The matrix normal distribution represents a generalization of the multivariate normal distribution to $n \times m$ random matrices. It is a rather uncommon distribution, and deriving its properties is not trivial. Most of the treatment in this section comes from Gupta and Nagar (2000), chapter 2.

Compared to its univariate and multivariate counterparts, the matrix normal distribution adds an additional column dimension of size m. The location matrix M represents the mean of the distribution. Also, just the same way the Σ matrix represents the row covariances for the multivariate normal distribution, the Ω matrix defines the column covariances for the matrix normal. When $m = 1$, the matrix normal degenerates into a multivariate normal distribution, and when $m = n = 1$, it becomes a simple normal distribution.

There exists a general equivalence between the matrix normal and multivariate normal distribution. Some authors actually use this feature as a definition for the matrix normal distribution. This is stated in the following property:

property d.7: the random variable X is a random variable with: $X \sim MN(M, \Sigma, \Omega)$ if and only if $vec(X)$ is a random variable with: $vec(X) \sim N(vec(M), \Omega \otimes \Sigma)$.

The second property represents the equivalent of the affine property for the multivariate normal distribution.

property d.8: let X be a random variable with: $X \sim MN(M, \Sigma, \Omega)$; let A, B and C be matrices such that $AXB + C$ is defined, with A and B of maximum rank n and m respectively. Then:

$AXB + C \sim MN(AMB + C,\ A\Sigma A', B'\Omega B)$

The third property is related to the marginal distributions of matrix normal random variables:

property d.9: let X be a random variable with: $X \sim MN(M, \Sigma, \Omega)$. Let X, M, Σ and Ω be partitioned the following ways:

$$X = \left(\begin{array}{c|c|c|c} X_{11} & X_{12} & \dots & X_{1q} \\ \hline X_{21} & X_{22} & \dots & X_{2q} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline X_{p1} & X_{p2} & \dots & X_{pq} \end{array}\right) \begin{array}{l} n_1 \\ n_2 \\ \vdots \\ n_p \end{array} \qquad M = \left(\begin{array}{c|c|c|c} M_{11} & M_{12} & \dots & M_{1q} \\ \hline M_{21} & M_{22} & \dots & M_{2q} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline M_{p1} & M_{p2} & \dots & M_{pq} \end{array}\right) \begin{array}{l} n_1 \\ n_2 \\ \vdots \\ n_p \end{array}$$

$$\begin{array}{cccc} m_1 & m_2 & \dots & m_q \end{array} \qquad \begin{array}{cccc} m_1 & m_2 & \dots & m_q \end{array}$$

$$\Sigma = \left(\begin{array}{c|c|c|c} \Sigma_{11} & \Sigma_{12} & \dots & \Sigma_{1p} \\ \hline \Sigma_{21} & \Sigma_{22} & \dots & \Sigma_{2p} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \Sigma_{p1} & \Sigma_{p2} & \dots & \Sigma_{pp} \end{array}\right) \begin{array}{l} n_1 \\ n_2 \\ \vdots \\ n_p \end{array} \qquad \Omega = \left(\begin{array}{c|c|c|c} \Omega_{11} & \Omega_{12} & \dots & \Omega_{1q} \\ \hline \Omega_{21} & \Omega_{22} & \dots & \Omega_{2q} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \Omega_{q1} & \Omega_{q2} & \dots & \Omega_{qq} \end{array}\right) \begin{array}{l} m_1 \\ m_2 \\ \vdots \\ m_q \end{array}$$

$$\begin{array}{cccc} n_1 & n_2 & \dots & n_p \end{array} \qquad \begin{array}{cccc} m_1 & m_2 & \dots & m_q \end{array}$$

with $n_1 + n_2 + \ldots + n_p = n$, and $m_1 + m_2 + \ldots + m_q = m$.

Then $X_{ij} \sim MN(M_{ij},\ \Sigma_{ii},\ \Omega_{jj})$, for all partitions $i = 1, 2, \ldots, p$, and all partitions $j = 1, 2, \ldots, q$.

This property states that any subset of a matrix normal distribution is itself matrix normal, with the mean and variance parameters defined in accordance with the considered partition. When the partition is realised at the entry level, one obtains that every individual entry of a matrix normal distribution follows a univariate normal distribution.

property d.10: let X be a random variable with: $X \sim MN(M, \Sigma, \Omega)$. Then:
$x_{ij} \sim N(m_{ij}, \sigma_{ii}\omega_{jj})$, for all $i = 1, 2, \ldots, p$, and all $j = 1, 2, \ldots, q$.

Finally, the following algorithms introduce the procedures to generate pseudo random numbers from the matrix normal distribution. First, the next algorithm considers the generation of random numbers from the standard matrix normal distribution.

algorithm d.14: random number generator for the standard matrix normal distribution

1. draw a nm-dimensional vector x from the standard multivariate normal distribution: $x \sim N(0, I_{nm})$.
2. rearrange x into the $n \times m$ matrix X.

Then from property d.7, X is a random draw from $X \sim MN(0, I_n, I_m)$.

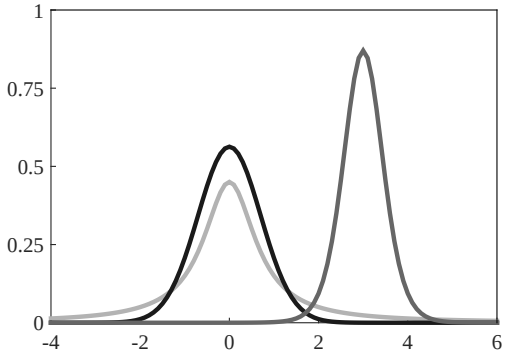
The next algorithm develops the procedure to draw from an arbitrary matrix normal distribution:

algorithm d.15: random number generator for the matrix normal distribution

1. calculate any matrix G such that $GG' = \Sigma$, and any matrix H such that $HH' = \Omega$. In practice, G and H are often chosen to be the Cholesky factors of Σ and Ω.
2. draw a random matrix Z from $Z \sim MN(0, I_n, I_m)$.
3. calculate $X = M + GZH'$.

Then from property d.8, X is a random draw from $X \sim MN(M, \Sigma, \Omega)$.

d.11. Student

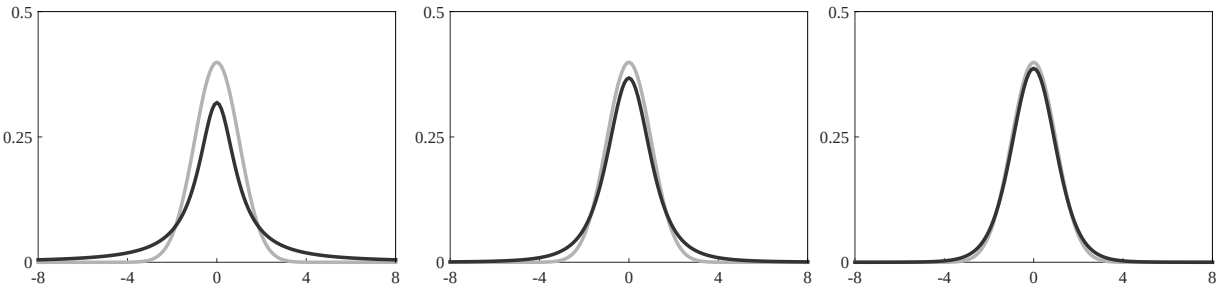


Type:	continuous
Notation:	$x \sim T(\mu, \sigma, \nu)$
Parameters:	μ (location, scalar) σ (scale, scalar with $\sigma > 0$) ν (degrees of freedom, scalar with $\nu > 0$)
Support:	$x \in \mathbb{R}$
pdf:	$f(x\|\mu,\sigma,\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}(\nu\pi\sigma)^{-1/2}\left(1+\frac{1}{\nu}\frac{(x-\mu)^2}{\sigma}\right)^{-(\nu+1)/2}$ $\Gamma(z)$ is the Gamma function, with $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$
Kernel:	$f(x\|\mu,\sigma,\nu) \propto \left(1+\frac{1}{\nu}\frac{(x-\mu)^2}{\sigma}\right)^{-(\nu+1)/2}$
Normalizing constant:	$c = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}(\nu\pi\sigma)^{-1/2}$
Mean:	μ for $\nu > 1$, else undefined
Variance:	$\frac{\nu}{\nu-2}\sigma$ for $\nu > 2$, else undefined
Median:	μ
Mode:	μ
Diffuse distribution:	set $\mu = 0$ and $\sigma \to \infty$ (proper distribution) or set $f(x\|\mu,\sigma,\nu) \propto 1$ (improper distribution)
Related distributions:	Normal: if $x \sim T(\mu,\sigma,\nu)$ and $\nu \to \infty$, then approximately $x \sim N(\mu,\sigma)$

Table d.11: Summary of the Student distribution

The student distribution (sometimes called the Student's t distribution or simply the t distribution) shares much with the normal distribution. It is symmetric around its mean represented by the location parameter μ, and it is also bell-shaped. One fundamental difference with the normal distribution is that the Student distribution has a fat tail. This means the the peak of the distribution is less pronounced, while the tails of the distribution (the extremities) are thicker. As a consequence, values far away from the mean have a higher probability to happen than with the normal distribution. This makes the Student distribution suitable for random variables with higher probabilities of rare events. Typical applications are found in finance, to model for instance classes of assets for which high returns or losses occur more frequently than for other assets.

How fat the tails of the distribution are is determined by the parameter ν called the degrees of freedom. The smaller ν, the fatter the tails and the higher the probability of obtaining values far away from the mean. Conversely, the larger ν, the more limited will be the fatness of the tails. A famous property of the Student distribution is that as $\nu \to \infty$ the Student distribution $T(\mu, \sigma, \nu)$ converges to a Normal distribution $N(\mu, \sigma)$. The limiting case $\nu \to \infty$ can then be interpreted as a situation where the additional fatness of the tail has been completely eliminated, turning the Student distribution into its Normal counterpart. This is illustrated by Figure d.13.



(a) $\nu = 1$ (b) $\nu = 3$ (c) $\nu = 8$

Figure d.13: Fatness of the student distribution $T(0,1,\nu)$ (grey curve is $N(0,1)$)

The final parameter σ known as the scale parameter is not directly interpretable as the variance of the distribution. While larger values of σ do imply a larger variance, the complete variance value is determined by both σ and the degrees of freedom ν. Smaller values of ν imply a larger variance of the distribution, consistently with the fact that the degrees of freedom determine the fatness of the distribution. Only when $\nu \to \infty$ does the variance tend to σ, the variance of the Normal distribution. This once again reflects convergence of the Student distribution to the Normal distribution as $\nu \to \infty$.

Similarly to the normal distribution, the Student distribution has the following affine property:

property d.11: let x be a random variable with: $x \sim T(\mu, \sigma, \nu)$, and let $y = ax + b$. Then:
$y \sim T(a\mu + b, a^2\sigma, \nu)$.

The following algorithm introduces the procedure to generate pseudo random numbers from the standard Student distribution.

algorithm d.16: random number generator for the standard Student distribution

1. draw a random number s from $s \sim IG(\frac{\nu}{2}, \frac{\nu}{2})$.
2. draw a random number x from $x \sim N(0, s)$.

Then x is a random draw from $x \sim T(0, 1, \nu)$.

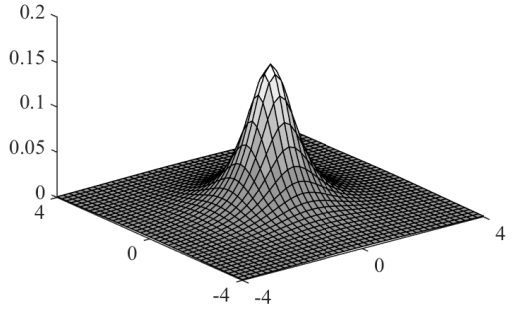
The next algorithm develops the procedure to draw from an arbitrary Student distribution:

algorithm d.17: random number generator for the Student distribution

1. draw a random number z from $z \sim T(0, 1, \nu)$.
2. set $x = \sqrt{\sigma} z + \mu$.

Then from property d.11, x is a random draw from $x \sim T(\mu, \sigma, \nu)$.

d.12. Multivariate Student



Type:	continuous
Notation:	$x \sim T(\mu, \Sigma, \nu)$
Parameters:	μ (n-dimensional location vector) Σ ($n \times n$ scale matrix, symmetric and positive definite) ν (degrees of freedom, scalar with $\nu > 0$)
Support:	$x \in \mathbb{R}^n$, the set of $n \times 1$ vectors of real numbers
pdf:	$f(x\|\mu,\Sigma,\nu) = \frac{\Gamma\left(\frac{\nu+n}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}(\nu\pi)^{-n/2}\|\Sigma\|^{-1/2}\left(1+\frac{1}{\nu}(x-\mu)'\Sigma^{-1}(x-\mu)\right)^{-(\nu+n)/2}$ $\Gamma(z)$ is the Gamma function, with $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$
Kernel:	$f(x\|\mu,\Sigma,\nu) \propto \left(1+\frac{1}{\nu}(x-\mu)'\Sigma^{-1}(x-\mu)\right)^{-(\nu+n)/2}$
Normalizing constant:	$c = \frac{\Gamma\left(\frac{\nu+n}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}(\nu\pi)^{-n/2}\|\Sigma\|^{-1/2}$
Mean:	μ for $\nu > 1$, else undefined
Variance:	$\frac{\nu}{\nu-2}\Sigma$ for $\nu > 2$, else undefined
Median:	μ
Mode:	μ
Diffuse distribution:	set $\mu = 0$ and $\Sigma = \sigma I_n$, with σ a scalar such that $\sigma \to \infty$ (proper distribution) or set $f(x\|\mu,\Sigma,\nu) \propto 1$ (improper distribution)
Related distributions:	Student: if $n = 1$, then $x \sim T(\mu,\sigma,\nu)$ Multivariate normal: if $x \sim T(\mu,\Sigma,\nu)$ and $\nu \to \infty$ then approximately: $x \sim N(\mu,\Sigma)$

Table d.12: Summary of the multivariate Student distribution

The multivariate Student distribution generalises the Student distribution to n-dimensional random vectors, much the same way the multivariate normal generalises the univariate normal to random vectors. Its mean is given by the location vector μ, while its variance and covariance are proportional to the diagonal and off-diagonal terms of Σ, respectively. A change in any of these terms results in a change of the shape of the density in a way that is similar to the multivariate normal.

The degrees of freedom ν affect the distribution just like they do for the univariate case: a smaller value of ν increases the fatness of the tails, making values close to the mean less likely to occur, and values further away more likely. Also, similarly to the univariate case, when $\nu \to \infty$, the multivariate Student distribution converges to the multivariate normal distribution.

Similarly to the univariate case, there exists an affine property for the multivariate Student distribution:

property d.12: let x be a random variable with: $x \sim T(\mu, \Sigma, \nu)$. Let A be some matrix and b be some vector such that $y = Ax + b$ is defined. Then:
$y \sim T(A\mu + b\,,\, A\Sigma A'\,,\, \nu\,)$

The second property states that the marginal distributions of a multivariate Student distribution are themselves Student:

property d.13: let x be a random variable with: $x \sim T(\mu, \Sigma, \nu)$. Let x, μ and Σ be partitioned the following way:

$$x = \begin{pmatrix} x_1 \\ \hline x_2 \\ \hline \vdots \\ \hline x_p \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \hline \mu_2 \\ \hline \vdots \\ \hline \mu_p \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix} \qquad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \dots & \Sigma_{1p} \\ \hline \Sigma_{21} & \Sigma_{22} & \dots & \Sigma_{2p} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \Sigma_{p1} & \Sigma_{p2} & \dots & \Sigma_{pp} \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix}$$
$$\begin{matrix} n_1 & n_2 & \dots & n_p \end{matrix}$$

with $n_1 + n_2 + \ldots + n_p = n$.

Then $x_i \sim T(\mu_i, \Sigma_{ii}, \nu)$, for all $i = 1, 2, \ldots, p$.

A corollary obtains when the partition is realised at the entry level, for then every individual entry of a multivariate Student distribution follows a univariate Student distribution:

property d.14: let x be a random variable with: $x \sim T(\mu, \Sigma, \nu)$. Then:
$x_i \sim T(\mu_i, \Sigma_{ii}, \nu) \quad$, $\quad$ for all $i = 1, 2, \ldots, n$.

The following algorithm introduces the procedure to generate pseudo random numbers from the standard multivariate Student distribution.

algorithm d.18: random number generator for the standard multivariate Student distribution

1. draw a random number s from $s \sim IG(\frac{\nu}{2}, \frac{\nu}{2})$.
2. draw a random vector x from $x \sim N(0, sI_n)$.

Then $x \sim T(0, I_n, \nu)$.

The next algorithm develops the procedure to draw from an arbitrary multivariate Student distribution:

algorithm d.19: random number generator for the multivariate Student distribution

1. calculate any matrix G such that $GG' = \Sigma$. In practice, G is often chosen to be the Cholesky factor of Σ.
2. draw a random number z from the standard multivariate Student distribution: $z \sim T(0, I_n, \nu)$.
3. set $x = \mu + Gz$.

Then from property d.12, x is a random draw from $x \sim T(\mu, \Sigma, \nu)$.

d.13. Matrix Student

Type:	continuous
Notation:	$X \sim MT(M, \Sigma, \Omega, \nu)$
Parameters:	M ($n \times m$ location matrix) Σ ($n \times n$ scale matrix, symmetric and positive definite) Ω ($m \times m$ scale matrix, symmetric and positive definite) ν (degrees of freedom, scalar with $\nu > 0$)
Support:	$X \in \mathbb{R}^{n \times m}$, the set of $n \times m$ matrices of real numbers
pdf:	$f(X\|M,\Sigma,\Omega,\nu) = \frac{\Gamma_n\left(\frac{\nu+n+m-1}{2}\right)}{\Gamma_n\left(\frac{\nu+n-1}{2}\right)}(\nu\pi)^{-nm/2}\|\Omega\|^{-n/2}\|\Sigma\|^{-m/2} \times \left\|I_m + \frac{1}{\nu}\Omega^{-1}(X-M)'\Sigma^{-1}(X-M)\right\|^{-(\nu+n+m-1)/2}$ $\Gamma_n(z)$ is the multivariate Gamma function, with $\Gamma_n(z) = \pi^{n(n-1)/4}\prod_{i=1}^{n}\Gamma\left(z + \frac{1-i}{2}\right)$
Kernel:	$f(X\|M,\Sigma,\Omega,\nu) \propto \left\|I_m + \frac{1}{\nu}\Omega^{-1}(X-M)'\Sigma^{-1}(X-M)\right\|^{-(\nu+n+m-1)/2}$
Normalizing constant:	$c = \frac{\Gamma_n\left(\frac{\nu+n+m-1}{2}\right)}{\Gamma_n\left(\frac{\nu+n-1}{2}\right)}(\nu\pi)^{-nm/2}\|\Omega\|^{-n/2}\|\Sigma\|^{-m/2}$
Mean:	M
Variance:	$Var(vec(X)) = \frac{\nu}{\nu-2}(\Omega \otimes \Sigma)$ for $\nu > 2$, else undefined
Median:	M
Mode:	M
Diffuse distribution:	set $M = 0$, $\Sigma = \sigma I_n$ and $\Omega = \omega I_m$, with σ and ω scalars such that $\sigma, \omega \to \infty$ (proper distribution) or set $f(X\|M,\Sigma,\Omega,\nu) \propto 1$ (improper distribution)
Related distributions:	Multivariate Student: if $m = 1$, then $X \sim T(\mu, \Sigma, \nu)$ Student: if $n = 1$ and $m = 1$, then $x \sim T(\mu, \sigma, \nu)$ Matrix normal: if $X \sim MT(M,\Sigma,\Omega,\nu)$ and $\nu \to \infty$ then approximately: $X \sim MN(M,\Sigma,\Omega)$

Table d.13: Summary of the matrix Student distribution

The matrix Student distribution generalises the multivariate Student distribution to $n \times m$-dimensional matrices, much the same way the matrix normal distribution generalises the multivariate normal to normal random matrices. There exist several different parameterisations for the matrix Student distribution (see for instance Dickey (1967), Box and Tiao (1973) and Gupta and Nagar (2000) for an overview of the different formulations), which complicates the analysis. The parameterisations retained in this manual is that of Gupta and Nagar (2000), slightly adapted to make it consistent with the formulation of the other Student and normal distributions. See Appendix at the end of the section for additional details.

The location parameter M represents the mean of the distribution, while the row and column covariances obtains from the two scale matrices Σ and Ω. The degrees of freedom ν define the fatness of the tails, lower values of ν implying higher probabilities in the tails. Reducing the column dimension m to 1 turns the distribution into a multivariate Student, while reducing both the column and rows dimensions m and n to 1 collapses it to a univariate Student dimension. Finally, as the degrees of freedom ν tends to infinity, the matrix Student distribution converges to a matrix normal distribution.

Similarly to the other Student distributions, there exists an affine property for the matrix Student distribution.

property d.15: let X be a random variable with: $X \sim MT(M,\Sigma,\Omega,\nu)$; let A,B and C be matrices such that $AXB+C$ is defined, with A and B of maximum rank n and m respectively. Then:
$AXB+C \sim MT(AMB+C, A\Sigma A', B'\Omega B, \nu)$.

The next property derives the marginal distributions of the matrix Student distribution.

property d.16: let X be a random variable with: $X \sim MT(M,\Sigma,\Omega,\nu)$. Let X, M, Σ and Ω be partitioned the following ways:

$$X = \begin{pmatrix} X_{11} & X_{12} & \dots & X_{1q} \\ X_{21} & X_{22} & \dots & X_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ X_{p1} & X_{p2} & \dots & X_{pq} \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix} \qquad M = \begin{pmatrix} M_{11} & M_{12} & \dots & M_{1q} \\ M_{21} & M_{22} & \dots & M_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ M_{p1} & M_{p2} & \dots & M_{pq} \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix}$$
$$\begin{matrix} m_1 & m_2 & \dots & m_q \end{matrix} \qquad \begin{matrix} m_1 & m_2 & \dots & m_q \end{matrix}$$

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \dots & \Sigma_{1p} \\ \Sigma_{21} & \Sigma_{22} & \dots & \Sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{p1} & \Sigma_{p2} & \dots & \Sigma_{pp} \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_p \end{matrix} \qquad \Omega = \begin{pmatrix} \Omega_{11} & \Omega_{12} & \dots & \Omega_{1q} \\ \Omega_{21} & \Omega_{22} & \dots & \Omega_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_{q1} & \Omega_{q2} & \dots & \Omega_{qq} \end{pmatrix} \begin{matrix} m_1 \\ m_2 \\ \vdots \\ m_q \end{matrix}$$
$$\begin{matrix} n_1 & n_2 & \dots & n_p \end{matrix} \qquad \begin{matrix} m_1 & m_2 & \dots & m_q \end{matrix}$$

with $n_1+n_2+\ldots+n_p = n$, and $m_1+m_2+\ldots+m_q = m$.

Then $X_{ij} \sim MT(M_{ij},\Sigma_{ii},\Omega_{jj},\nu)$, for all partitions $i = 1,2,\ldots,p$, and all partitions $j = 1,2,\ldots,q$.

A corollary obtains when the partition is realised at the entry level, for then every individual entry of a multivariate Student distribution follows a univariate Student distribution:

property d.17: let X be a random variable with: $X \sim MT(M,\Sigma,\Omega,\nu)$. Then:
$x_{ij} \sim T(m_{ij}, \Sigma_{ii} \times \Omega_{jj}, \nu)$, for all $i = 1,2,...,n$, and all $j = 1,2,...,m$.

The next property is a simple one, but it is useful for the purpose of numerical computations:

property d.18: let X be a random variable with: $X \sim MT(M,\Sigma,\Omega,\nu)$. Then:
$X' \sim MT(M',\Omega,\Sigma,\nu)$.

The following algorithm introduces the procedure to generate pseudo random numbers from the standard matrix Student distribution.

algorithm d.20: random number generator for the standard matrix Student distribution

If $m \leq n$:

1. draw a $m \times m$ random matrix Φ from $\Phi \sim IW(\nu + m - 1, \nu I_m)$.
2. draw a random matrix X from $X \sim MN(0, I_n, \Phi)$.

Else, if $m > n$:

1. draw a $n \times n$ random matrix Φ from $\Phi \sim IW(\nu + n - 1, \nu I_n)$.
2. draw a random matrix Y from $Y \sim MN(0, I_m, \Phi)$.
3. set $X = Y'$.

Then $X \sim MT(0, I_n, I_m, \nu)$.

The separate treatment of the two cases $m \leq n$ and $m > n$ guarantees that the Inverse Wishart draw is realised on the smallest dimension of X, in order to maximise efficiency. The case $m > n$ is obtained by direct application of property d.18. The final algorithm develops the procedure to draw from an arbitrary matrix Student distribution.

algorithm d.21: random number generator for the matrix Student distribution

1. calculate any matrix G such that $GG' = \Sigma$, and any matrix H such that $HH' = \Omega$. In practice, G and H are often chosen to be the Cholesky factors of Σ and Ω.
2. draw a random matrix Z from $Z \sim MT(0, I_n, I_m, \nu)$.
3. calculate $X = M + GZH'$.

Then from property d.15, X is a random draw from $X \sim MT(M, \Sigma, \Omega, \nu)$.

Appendix: details of the derivations for the matrix Student distribution

The main reference for the matrix Student definition is Gupta and Nagar (2000), chapter 4. These authors provide the following definition for the density of the matrix Student definition:

The $n \times m$ random matrix X is said to have a matrix variate t-distribution with parameters M, Σ, Ω and ν if its p.d.f is given by:

$$f(X|M,\Sigma,\Omega,\nu) = \frac{\Gamma_n\left(\frac{\nu+n+m-1}{2}\right)}{\Gamma_n\left(\frac{\nu+n-1}{2}\right)} \pi^{-nm/2} |\Omega|^{-n/2} |\Sigma|^{-m/2} \left|I_m + \Omega^{-1}(X-M)'\Sigma^{-1}(X-M)\right|^{-(\nu+n+m-1)/2}$$

The are two main shortcomings associated with this definition. First, it is not consistent with the standard definitions of the multivariate and univariate Student definitions. That is, setting the column dimension m to 1 will not produce the multivariate Student density given in Table d.12, and reducing both the column dimension m and the row dimension n to 1 will not produce the univariate Student density given in Table d.11. Second, the distribution will not properly converge to its normal counterpart. Indeed, in the limiting case where the degrees of freedom ν tend to infinity, the Student distributions (univariate and multivariate) converge to a normal distribution. That is, as $\nu \to \infty$, one has $T(\mu,\sigma,\mu) \to N(\mu,\sigma)$ and $T(\mu,\Sigma,\mu) \to N(\mu,\Sigma)$. One would then also expect that $\nu \to \infty$ implies $MT(M,\Sigma,\Omega,\mu) \to MN(M,\Sigma,\Omega)$, but with the definition provided by Gupta and Nagar (2000) this is not the case due to improper formulation of the degrees of freedom.

For these reasons, this manual substitutes the following formulation for the density:

$$f(X|M,\Sigma,\Omega,\nu) = \frac{\Gamma_n\left(\frac{\nu+n+m-1}{2}\right)}{\Gamma_n\left(\frac{\nu+n-1}{2}\right)} (\nu\pi)^{-nm/2} |\Omega|^{-n/2} |\Sigma|^{-m/2} \left|I_m + \tfrac{1}{\nu}\Omega^{-1}(X-M)'\Sigma^{-1}(X-M)\right|^{-(\nu+n+m-1)/2}$$

Unlike the formulation of Gupta and Nagar (2000), the above formulation is consistent with the other classes of Student distributions, and it does converge properly to the matrix normal distribution when $\nu \to \infty$. However it is not a standard formulation of the distribution and its properties have not been studied extensively. By contrast, Gupta and Nagar (2000) provide a thorough treatment of the distribution under their formulation. Fortunately, it is trivial to create an equivalence between the two definitions, by noting the following fact:

$$\begin{aligned}
f(X|M,\Sigma,\Omega,\nu) &= \frac{\Gamma_n\left(\frac{\nu+n+m-1}{2}\right)}{\Gamma_n\left(\frac{\nu+n-1}{2}\right)} (\nu\pi)^{-nm/2} |\Omega|^{-n/2} |\Sigma|^{-m/2} \left|I_m + \tfrac{1}{\nu}\Omega^{-1}(X-M)'\Sigma^{-1}(X-M)\right|^{-(\nu+n+m-1)/2} \\
&= \frac{\Gamma_n\left(\frac{\nu+n+m-1}{2}\right)}{\Gamma_n\left(\frac{\nu+n-1}{2}\right)} \pi^{-nm/2} |\nu\Omega|^{-n/2} |\Sigma|^{-m/2} \left|I_m + (\nu\Omega)^{-1}(X-M)'\Sigma^{-1}(X-M)\right|^{-(\nu+n+m-1)/2} \\
&\quad \text{(using properties m.11 and m.14)} \\
&= \frac{\Gamma_n\left(\frac{\nu+n+m-1}{2}\right)}{\Gamma_n\left(\frac{\nu+n-1}{2}\right)} \pi^{-nm/2} |\tilde{\Omega}|^{-n/2} |\Sigma|^{-m/2} \left|I_m + \tilde{\Omega}^{-1}(X-M)'\Sigma^{-1}(X-M)\right|^{-(\nu+n+m-1)/2} \\
&\quad \text{defining } \tilde{\Omega} = \nu\Omega
\end{aligned}$$

It can then be seen that the formulation adopted in this manual is equivalent to the matrix Student distribution of Gupta and Nagar (2000) parameterised as $f(X|M,\Sigma,\tilde{\Omega},\nu)$.

Following, all the properties developed in Gupta and Nagar (2000) apply to the present definition after a trivial adjustment in the definition of the parameters. For instance, Theorem 4.3.1 in Gupta and Nagar (2000) states:

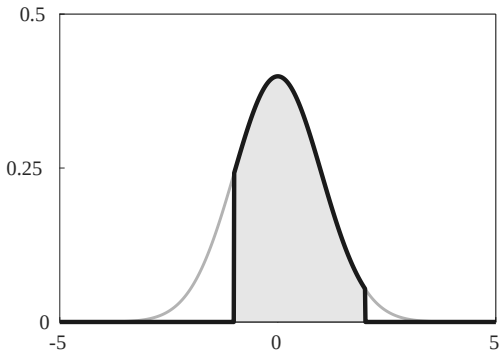
$Var(vec(X)) = \frac{1}{n-2}(\Omega \otimes \Sigma)$

Using $\tilde{\Omega}$ in place of Ω, one obtains that for the definition used in this manual:

$Var(vec(X)) = \frac{1}{\nu-2}(\tilde{\Omega} \otimes \Sigma) = \frac{1}{\nu-2}(\nu\Omega \otimes \Sigma) = \frac{\nu}{\nu-2}(\Omega \otimes \Sigma)$ (using property m.41)

Any other property can be derived in a similar fashion.

d.14. Truncated normal

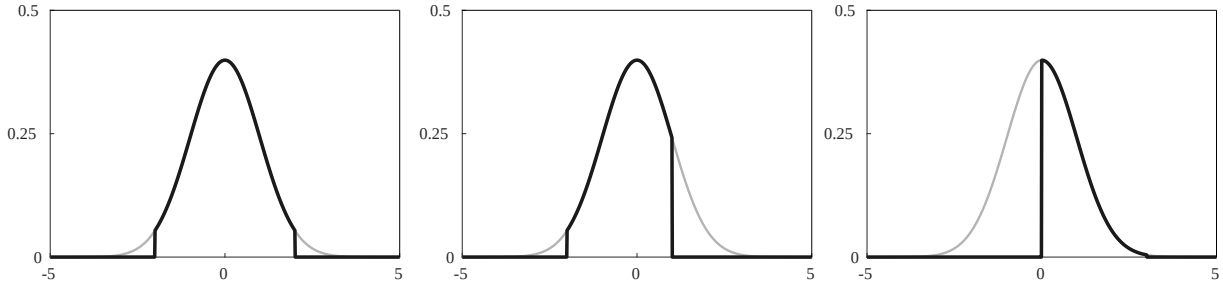


Type:	continuous
Notation:	$x \sim \bar{N}(\mu, \sigma, a, b)$
Parameters:	μ (location, scalar) σ (scale, scalar with $\sigma > 0$) a (lower bound, scalar) b (upper bound, scalar with $b \geq a$)
Support:	$x \in [a, b]$
pdf:	$f(x\|\mu,\sigma,a,b) = (\Phi(\beta) - \Phi(\alpha))^{-1} (2\pi\sigma)^{-1/2} exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma}\right) \mathbb{1}(a \leq x \leq b)$ $\Phi(x)$ is the cumulative distribution function of the standard normal distribution $\alpha = (a-\mu)/\sqrt{\sigma}$ $\quad$ $\beta = (b-\mu)/\sqrt{\sigma}$
Kernel:	$f(x\|\mu,\sigma) \propto exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma}\right) \mathbb{1}(a \leq x \leq b)$
Normalizing constant:	$c = (\Phi(\beta) - \Phi(\alpha))^{-1} (2\pi\sigma)^{-1/2}$
Mean:	$\mu - \sqrt{\sigma}\,\frac{\phi(\beta)-\phi(\alpha)}{\Phi(\beta)-\Phi(\alpha)}$ $\phi(x)$ is the probability density function of the standard normal distribution
Variance:	$\sigma\left(1 - \frac{\beta\phi(\beta)-\alpha\phi(\alpha)}{\Phi(\beta)-\Phi(\alpha)} - \left[\frac{\phi(\beta)-\phi(\alpha)}{\Phi(\beta)-\Phi(\alpha)}\right]^2\right)$
Median:	$\mu + \sqrt{\sigma}\,\Phi^{-1}\left(\frac{\Phi(\alpha)+\Phi(\beta)}{2}\right)$
Mode:	a if $\mu < a$, μ if $a \leq x \leq b$, b if $\mu > b$
Diffuse distribution:	–
Related distributions:	Normal: if $a = -\infty$ and $b = \infty$, then $x \sim N(\mu, \sigma)$

Table d.14: Summary of the truncated normal distribution

As the name indicates, the truncated normal distribution is similar to the normal distribution except that its support is truncated. This can be useful to model real life phenomena that roughly follow a normal distribution, but over a finite support. For instance, to model the probability that a fliiped coin yields "heads", one may want to use a normal distribution centered at 0.5, but with a truncation over the $[0,1]$ interval.

The truncated normal distribution is defined by 4 parameters. The lower bound a and the upper bound b define the limits of the support. By playing over these parameters, one can handle a wide variety of shapes and turn the truncated normal distribution into a very flexible device. This is illustrated in Figure d.14.



(a) Symmetric (b) Asymmetric (c) Downward-sloping

Figure d.14: Different shapes of the truncated normal distribution

Unlike the normal distribution, the parameters μ and σ do not represent the mean and variance of the truncated normal distribution. The truncation acts as a disturbances that shifts these parameters by an amount determined by the parameters a and b.

There exist many different algorithms to generate pseudo-random numbers from the truncated normal distribution. Their efficiency usually depends on where the truncation is defined, and thus which part of the normal distribution the algorithm must sample from (the centre or the tails). The algorithm proposed here is due to Robert (1995). It is quite efficient, whatever the way the truncation is applied.

algorithm d.22: random number generator for the truncated normal distribution

1. draw w from $w \sim U(a,b)$.
2. compute:
   $z = exp(-w^2/2)$ if $0 \in [a,b]$
   $z = exp((b^2-w^2)/2)$ if $b < 0$
   $z = exp((a^2-w^2)/2)$ if $a > 0$
3. draw u from $u \sim U(0,1)$, and set $x = w$ if $u < z$; otherwise return to step 1.

Then x is a random draw from $x \sim \bar{N}(0,1,a,b)$.

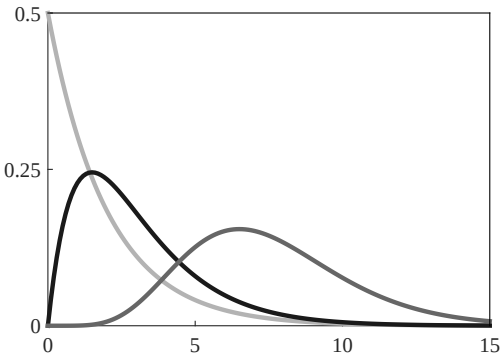
To draw from an arbitrary distribution $\bar{N}(\mu,\sigma,a,b)$, the following algorithm can be used.

algorithm d.23: random number generator for the truncated normal distribution

1. define $\alpha = (a-\mu)/\sqrt{\sigma}$ and $\beta = (b-\mu)/\sqrt{\sigma}$.
2. draw z from $z \sim \bar{N}(0,1,\alpha,\beta)$.
3. set $x = \mu + \sqrt{\sigma}z$.

Then x is a random draw from $x \sim \bar{N}(\mu,\sigma,a,b)$.

d.15. Gamma

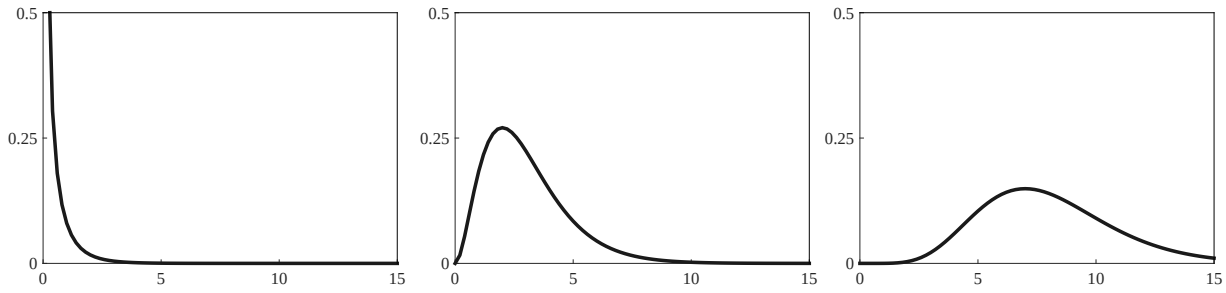


Type:	continuous
Notation:	$x \sim G(a,b)$
Parameters:	a (shape, scalar with $a > 0$) b (scale, scalar with $b > 0$)
Support:	$x \in [0,\infty)$
pdf:	$f(x\|a,b) = \frac{b^{-a}}{\Gamma(a)} x^{a-1} exp\left(-\frac{x}{b}\right)$ $\Gamma(z)$ is the Gamma function, with $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$
Kernel:	$f(x\|a,b) \propto x^{a-1} exp\left(-\frac{x}{b}\right)$
Normalizing constant:	$c = \frac{b^{-a}}{\Gamma(a)}$
Mean:	ab
Variance:	ab^2
Median:	$\approx ab\frac{3a-0.8}{3a+0.2}$
Mode:	$(a-1)b$ for $a \geq 1$
Diffuse distribution:	set $a \to 0$ and $b \to \infty$ (proper distribution) or set $f(x\|a,b) \propto \frac{1}{x}$ (improper distribution)
Related distributions:	Exponential: if $x \sim G(1,1/\lambda)$, then $x \sim Exp(\lambda)$ Chi-squared: if $x \sim G(\nu/2,2)$, then $x \sim \chi^2(\nu)$ Inverse gamma: if $x \sim G(a,b)$, then $1/x \sim IG(a,1/b)$ Normal: if $x \sim G(a,b)$, and $a \to \infty$, then approximately $x \sim N(ab,ab^2)$

Table d.15: Summary of the gamma distribution

The gamma distribution takes only positive values. In this respect it can be used to model any random event resulting in positive quantities, which gives it a very wide range of applications. It can be used for instance to model physical quantities (the amount of rainfall in a given country over a year), time durations (the amount of time before a factory machine defects), amounts of money (the amount of insurance claims after a natural disaster), and so on.

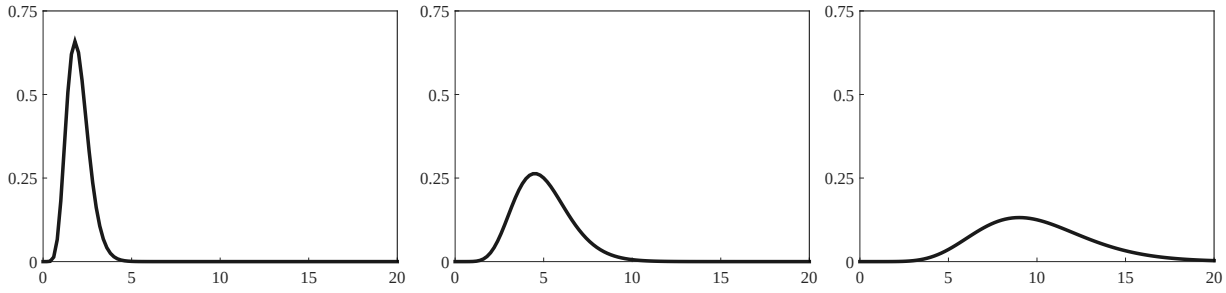
The distribution is defined by two parameters: the shape parameter a, and the scale parameter b. The shape parameter a determines the overall shape of the distribution. Smaller values of a increase positive skewness, attributing more probability to values around the origin and less weight to values further away. As a gets larger, the distribution gets more and more bell-shaped and symmetric. This is depicted in Figure d.15:



(a) $a = 0.2$ (b) $a = 3$ (c) $a = 8$

Figure d.15: Impact of the shape parameter a on the gamma distribution (scale $b = 1$)

The scale parameter b on the other hand represents the overall scale of the function. Smaller values of b squeeze the distribution while larger values of b stretch it, without affecting the shape of the distribution. This is illustrated in Figure d.16:



(a) $b = 0.2$ (b) $b = 0.5$ (c) $b = 1$

Figure d.16: Impact of the scale parameter b on the gamma distribution (shape $a = 10$)

A useful consequence of the effect of the scale parameter is the following property.

property d.19: let x be a random variable with: $x \sim G(a,b)$, and let $c > 0$ be some scalar. Then: $cx \sim G(a,cb)$.

Also, combining the effects of the two parameters a and b, it is easy to generate a gamma distribution with any desired pair of values for the mean and variance. This is stated in the following property:

property d.20: let x be a random variable with: $x \sim G(a,b)$. Let μ and σ respectively denote any desired mean and variance for the distribution (with $\mu > 0$ and $\sigma > 0$). Then these values can be obtained by defining:

$a = \frac{\mu^2}{\sigma}$ and $b = \frac{\sigma}{\mu}$.

The gamma distribution is related to a number of other distributions that appear either as limiting cases or as special cases of the gamma distribution. For instance, for very large values of a the gamma distribution approximates the normal distribution. Other well-known distributions appear as special cases of the gamma. The exponential distribution with rate λ is a gamma distribution with $a = 1$ and $b = \frac{1}{\lambda}$, while the Chi-squared distribution with degrees of freedom ν is a gamma distribution with $a = \frac{\nu}{2}$ and $b = 2$. Because these distributions arise a special cases of the gamma distribution, their properties can be directly derived from those of the gamma distribution.

There exist a number of algorithms to generate pseudo-random numbers from the Gamma distribution. The following algorithm is due to Marsaglia and Tsang (2000a). It is widely used for its efficiency and simplicity.

algorithm d.24: random number generator for the gamma distribution $(a \geq 1)$

1. set $d = a - 1/3$ and $c = 1/\sqrt{9d}$.
2. generate $x \sim N(0,1)$.
3. generate $v = 1 + cx$.
4. if $v > 0$, set $v = v^3$ and generate $u \sim U(0,1)$; otherwise go back to 2.
5. if $u < 1 - 0.0331x^4$, set $y = dv$; then $y \sim G(a,1)$.
6. else, if $log(u) < 0.5x^2 + d(1 - v + log(v))$, set $y = dv$; then $y \sim G(a,1)$.
7. else, go back to 2.

The algorithm only works whenever $a \geq 1$. If $a < 1$, Marsaglia and Tsang (2000a) propose a simple transformation.

algorithm d.25: random number generator for the gamma distribution $(a < 1, b = 1)$

1. generate a random number z from $z \sim G(a+1,1)$, using algorithm d.24.
2. generate $u \sim U(0,1)$.
3. define $y = zu^{1/a}$; then $y \sim G(a,1)$.

Finally, to obtain a random number from a gamma distribution with arbitrary b value, the following algorithm is used.

algorithm d.26: random number generator for the gamma distribution (a, b)

1. generate a random number z from $z \sim G(a,1)$.
2. set $x = by$.

Then from property d.19, $x \sim G(a,b)$.

d.16. Wishart

Type:	continuous
Notation:	$X \sim W(\nu, S)$
Parameters:	ν (degrees of freedom, scalar with $\nu \geq n$) S ($n \times n$ scale matrix, symmetric and positive definite)
Support:	$X \in S^n_{++}$, the set of $n \times n$ positive definite matrices
pdf:	$f(X\|\nu,S) = \frac{2^{-\nu n/2}}{\Gamma_n\left(\frac{\nu}{2}\right)}\|S\|^{-\nu/2}\|X\|^{(\nu-n-1)/2} exp\left(-\frac{1}{2}tr\left\{XS^{-1}\right\}\right)$
Kernel:	$f(X\|\nu,S) \propto \|X\|^{(\nu-n-1)/2} exp\left(-\frac{1}{2}tr\left\{XS^{-1}\right\}\right)$
Normalizing constant:	$c = \frac{2^{-\nu n/2}}{\Gamma_n\left(\frac{\nu}{2}\right)}\|S\|^{-\nu/2}$
Mean:	νS
Variance:	$Var(x_{ij}) = \nu(s^2_{ij} + s_{ii}s_{jj})$
Median:	no simple analytical form
Mode:	$(\nu - n - 1)S$ for $\nu \geq n+1$, else undefined
Diffuse distribution:	set $\nu = n$ and $S = sI_n$, with s a scalar such that $s \to \infty$ (proper distribution) or set $f(X\|\nu,S) \propto \|X\|^{-(n+1)/2}$ (improper distribution)
Related distributions:	Gamma: if $n = 1$, then $X \sim G\left(\frac{\nu}{2}, \frac{S}{2}\right)$ Inverse Wishart: if $X \sim W(\nu,S)$, then $X^{-1} \sim IW(\nu, S^{-1})$

Table d.16: Summary of the Wishart distribution

The Wishart distribution generalizes the Gamma distribution to $n \times n$ positive definite matrices. It is characterised by two parameters, the degrees of freedom ν, and the scale S. The degrees of freedom ν are comparable to the shape parameter a of the gamma distribution, and determine the overall shape of the density function. The interpretation of scale matrix S is similar to its scalar counterpart b in the gamma distribution: an increase in the values of S increase the spread of the distribution by stretching the density, while smaller values in S reduce it. Also, consistently with the gamma distribution which produces only positive scalar values, the Wishart distribution only produces positive definite matrices.

Because its support is the set of positive definite matrices with positive diagonal terms and unrestricted off-diagonal terms, typical applications of the Wishart distribution consist in the analysis of the distribution of variance-covariance matrices.

When the degrees of freedom ν is integer, it is possible to define the Wishart distribution directly as follows.

property d.21: let A be a $n \times \nu$ matrix of independently drawn standard normal random numbers: $a_{ij} \sim N(0,1)$. Let $X = AA'$. Then: $X \sim W(\nu, I_n)$.

The Wishart distribution has the following affine property:

property d.22: let X be a $n \times n$ matrix with: $X \sim W(\nu, S)$. Let A be a matrix of maximum rank n such that AXA' is defined. Then: $AXA' \sim W(\nu, ASA')$.

the following algorithms introduce different procedures to generate pseudo random numbers from the Wishart distribution. When the degrees of freedom ν is integer, a first option consists in using brute strength, using property d.21 as a direct definition of a Wishart draw:

algorithm d.27: random number generator for the Wishart distribution, ν integer

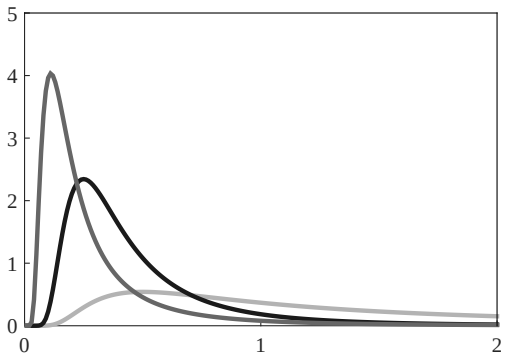
1. generate a $n \times \nu$ matrix A of independent standard normal random numbers: $a_{ij} \sim N(0,1)$.
2. set $Z = AA'$; then from property d.21, $Z \sim W(\nu, I_n)$.
3. calculate any matrix G such that $GG' = S$. In practice, G is often chosen to be the Cholesky factor of S.
4. set $X = GZG'$; then from property d.22, $Z \sim W(\nu, S)$.

This algorithm can only be used for integer degrees of freedom. It also becomes slow whenever ν is integer but large. In this case, one has to rely on alternative methods. The following algortihm is due to Bartlett (1934). It is known as the Bartlett decomposition of the Wishart distribution.

algorithm d.28: random number generator for the general Wishart distribution

1. initiate the matrix A as a $n \times n$ matrix of zeros.
2. diagonal terms: for $i = 1, 2, \ldots, n$, generate $a_{ii} = \sqrt{z}$, with $z \sim \chi^2(\nu + 1 - i)$.
3. off-diagonal terms: for $i = 1, 2, \ldots, n$ and $j < i$, generate a_{ij} from $a_{ij} \sim N(0,1)$.
4. set $Z = AA'$; then $Z \sim W(\nu, I_n)$.
5. calculate any matrix G such that $GG' = S$. In practice, G is often chosen to be the Cholesky factor of S.
6. set $X = GZG'$; then from property d.22, $Z \sim W(\nu, S)$.

d.17. Inverse gamma



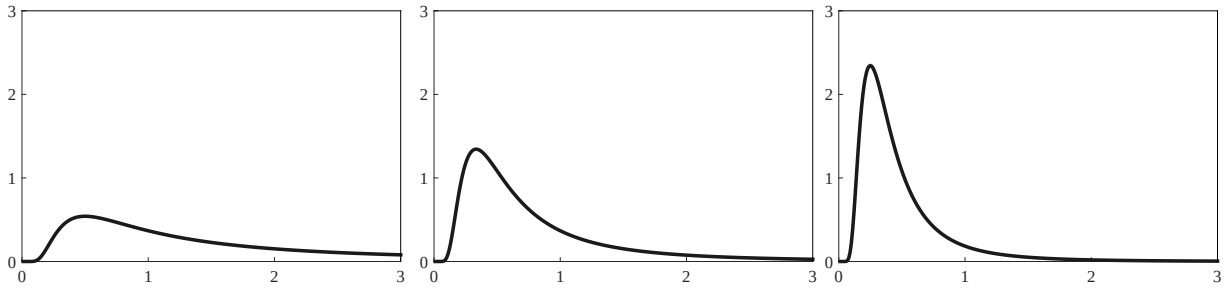
Type:	continuous
Notation:	$x \sim IG(a,b)$
Parameters:	a (shape, scalar with $a > 0$) b (scale, scalar with $b > 0$)
Support:	$x \in [0,\infty)$
pdf:	$f(x\|a,b) = \frac{b^a}{\Gamma(a)} x^{-a-1} exp\left(-\frac{b}{x}\right)$ $\Gamma(z)$ is the Gamma function, with $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$
Kernel:	$f(x\|a,b) \propto x^{-a-1} exp\left(-\frac{b}{x}\right)$
Normalizing constant:	$c = \frac{b^a}{\Gamma(a)}$
Mean:	$\frac{b}{a-1}$ for $a > 1$
Variance:	$\frac{b^2}{(a-1)^2(a-2)}$ for $a > 2$
Median:	$\approx \frac{b(3a+0.2)}{a(3a-0.8)}$
Mode:	$\frac{b}{a+1}$
Diffuse distribution:	set $a \to 0$ and $b \to 0$ (proper distribution) or set $f(x\|a,b) \propto \frac{1}{x}$ (improper distribution)
Related distributions:	Gamma: if $x \sim IG(a,b)$, then $1/x \sim G(a,1/b)$

Table d.17: Summary of the inverse gamma distribution

The inverse gamma distribution can be directly defined as the distribution that obtains from the reciprocal of the gamma distribution.

property d.23: let x be a random variable with: $x \sim G(a,b)$. Then: $\frac{1}{x} \sim IG(a, \frac{1}{b})$.

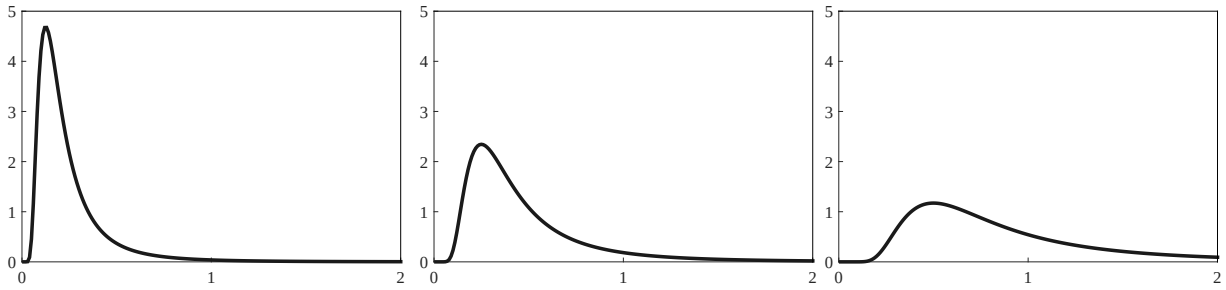
Similarly to the gamma distribution, the support of the inverse gamma distribution consists in the set of positive real numbers. The shape parameter a also determines the overall shape of the distribution. As a increases, the distribution concentrates higher probabilities on small values. This is the converse of the regular gamma distribution which attributes more weight to small values when a decreases, and this results directly from the inverse gamma being the reciprocal of the gamma distribution. This is illustrated by Figure d.17:



(a) $a = 0.2$ (b) $a = 0.5$ (c) $a = 1$

Figure d.17: Impact of the shape parameter a on the inverse gamma distribution (scale $b = 1$)

The second parameter of the distribution, the scale parameter b, determines the overall spread of the function. Its behaviour is similar to that of the regular gamma distribution: smaller values of b squeeze the distribution, while larger values stretch it, without affecting the shape of the distribution. This is illustrated by Figure d.18:



(a) $b = 0.5$ (b) $b = 1$ (c) $b = 2$

Figure d.18: Impact of the scale parameter b on the inverse gamma distribution (shape $a = 3$)

Similarly to the gamma distribution, it is possible to play on the impact of the shape and scale parameters a and b to implement any desired pair of values for the mean and variance of the distribution.

property d.24: let x be a random variable with: $x \sim IG(a,b)$. Let μ and σ respectively denote any desired mean and variance for the distribution (with $\mu > 0$ and $\sigma > 0$). Then these values can be obtained by defining:

$$a = \frac{\mu^2}{\sigma} + 2 \qquad \text{and} \qquad b = \mu\left(\frac{\mu^2}{\sigma} + 1\right).$$

The following algorithm introduces the procedures to generate pseudo random numbers from the inverse gamma distribution.

algorithm d.29: random number generator for the inverse gamma distribution

1. draw a random number z from $z \sim G\left(a, \frac{1}{b}\right)$.
2. set $x = \frac{1}{z}$.

Then from property d.23, $x \sim IG(a,b)$.

d.18. Inverse Wishart

Type:	continuous
Notation:	$X \sim IW(\nu, S)$
Parameters:	ν (degrees of freedom, scalar with $\nu \geq n$) S ($n \times n$ scale matrix, symmetric and positive definite)
Support:	$X \in S^n_{++}$, the set of $n \times n$ positive definite matrices
pdf:	$f(X\|\nu,S) = \frac{2^{-\nu n/2}}{\Gamma_n\left(\frac{\nu}{2}\right)}\|S\|^{\nu/2}\|X\|^{-(\nu+n+1)/2} exp\left(-\frac{1}{2}tr\left\{X^{-1}S\right\}\right)$
Kernel:	$f(X\|\nu,S) \propto \|X\|^{-(\nu+n+1)/2} exp\left(-\frac{1}{2}tr\left\{X^{-1}S\right\}\right)$
Normalizing constant:	$c = \frac{2^{-\nu n/2}}{\Gamma_n\left(\frac{\nu}{2}\right)}\|S\|^{\nu/2}$
Mean:	$\frac{S}{\nu-n-1}$ $\quad \nu > n+1$
Variance:	$Var(x_{ij}) = \frac{(\nu-n+1)s_{ij}^2+(\nu-n-1)s_{ii}s_{jj}}{(\nu-n)(\nu-n-1)^2(\nu-n-3)}$ $\quad \nu > n+3$
Median:	no simple analytical form
Mode:	$\frac{S}{\nu+n+1}$
Diffuse distribution:	set $\nu = n$ and $S = sI_n$, with s a scalar such that $s \to 0$ (proper distribution) or set $f(X\|\nu,S) \propto \|X\|^{-(n+1)/2}$ (improper distribution)
Related distributions:	Inverse gamma: if $n = 1$, then $X \sim IG\left(\frac{\nu}{2}, \frac{S}{2}\right)$ Wishart: if $X \sim IW(\nu,S)$, then $X^{-1} \sim W(\nu, S^{-1})$

Table d.18: Summary of the Inverse Wishart distribution

The inverse Wishart generalises the inverse gamma distribution to $n \times n$ positive definite matrices, much the same way the Wishart distribution generalises the gamma distribution. The relation linking the inverse Wishart to the Wishart distribution is similar to that relating the gamma to the inverse gamma: the inverse Wishart is the distribution that obtains when taking the inverse of a Wishart distribution.

property d.25: let X be a random variable with: $X \sim W(\nu, S)$. Then: $X^{-1} \sim IW(\nu, S^{-1})$.

The distribution is characterised by two parameters: the degrees of freedom ν, and the scale matrix S. The degrees of freedom ν represent the overall shape of the distribution, while the scale matrix S on the other hand determines the spread of the distribution: small values of S squeeze the distribution, while larger values stretch it.

There also exists an affine property for the inverse Wishart distribution.

property d.26: let X be a $n \times n$ matrix with: $X \sim IW(\nu, S)$. Let A be a matrix of maximum rank n such that AXA' is defined. Then: $AXA' \sim IW(\nu, ASA')$.

The following algorithms introduce the procedures to generate pseudo random numbers from the inverse Wishart distribution, making use of the definition of the inverse Wishart distribution as the reciprocal of the Wishart distribution. The first approach applies to integer degrees of freedom, using the brute strength strategy.

algorithm d.30: random number generator for the inverse Wishart distribution, ν integer

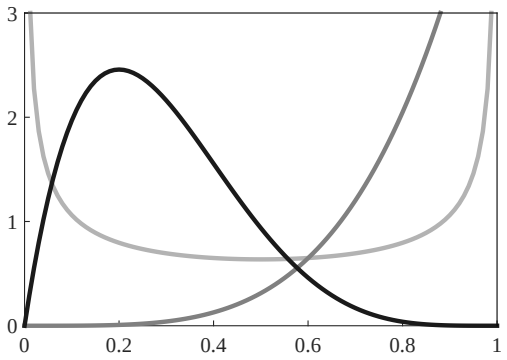
1. generate a $n \times \nu$ matrix A of independent standard normal random numbers: $a_{ij} \sim N(0,1)$.
2. set $Z = (AA')^{-1}$; then from property d.25, $Z \sim IW(\nu, I_n)$.
3. calculate any matrix G such that $GG' = S$. In practice, G is often chosen to be the Cholesky factor of S.
4. set $X = GZG'$; then from property d.26, $Z \sim IW(\nu, S)$.

When the degrees of freedom ν is large or not integer, one switches instead to the Bartlett decomposition.

algorithm d.31: random number generator for the general inverse Wishart distribution

1. initiate the matrix A as a $n \times n$ matrix of zeros.
2. diagonal terms: for $i = 1, 2, \ldots, n$, generate $a_{ii} = \sqrt{z}$, with $z \sim \chi^2(\nu + 1 - i)$.
3. off-diagonal terms: for $i = 1, 2, \ldots, n$ and $j < i$, generate a_{ij} from $a_{ij} \sim N(0,1)$.
4. set $Z = (AA')^{-1}$; then $Z \sim IW(\nu, I_n)$.
5. calculate any matrix G such that $GG' = S$. In practice, G is often chosen to be the Cholesky factor of S.
6. set $X = GZG'$; then from property d.26, $Z \sim IW(\nu, S)$.

d.19. Beta

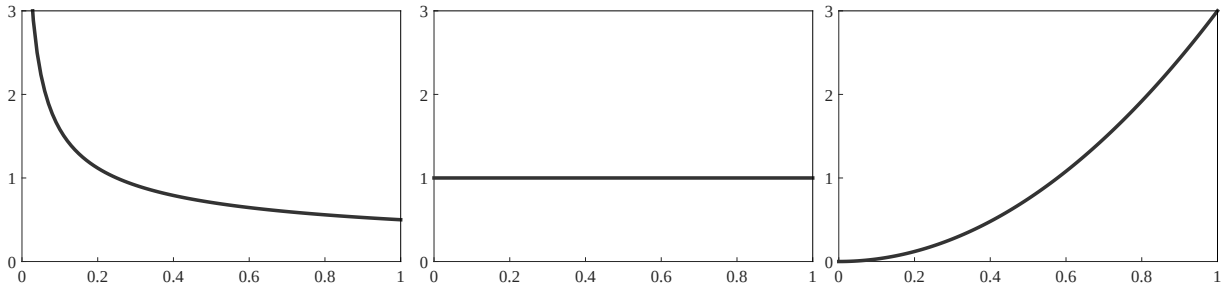


Type:	continuous
Notation:	$x \sim Beta(a,b)$
Parameters:	a (shape, scalar with $a > 0$) b (shape, scalar with $b > 0$)
Support:	$x \in [0,1]$
pdf:	$f(x\|a,b) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}$ $B(z,w)$ is the Beta function, with $B(z,w) = \frac{\Gamma(z)\Gamma(w)}{\Gamma(z+w)}$
Kernel:	$f(x\|a,b) \propto x^{a-1}(1-x)^{b-1}$
Normalizing constant:	$c = \frac{1}{B(a,b)}$
Mean:	$\frac{a}{a+b}$
Variance:	$\frac{ab}{(a+b)^2(a+b+1)}$
Median:	$\approx \frac{a-1/3}{a+b-2/3}$ for $a,b > 1$
Mode:	$\frac{a-1}{a+b-2}$ for $a,b > 1$
Diffuse distribution:	set $a \to 0$ and $b \to 0$ (proper distribution)
Related distributions:	Uniform: if $x \sim Beta(1,1)$, then $x \sim U(0,1)$

Table d.19: Summary of the Beta distribution

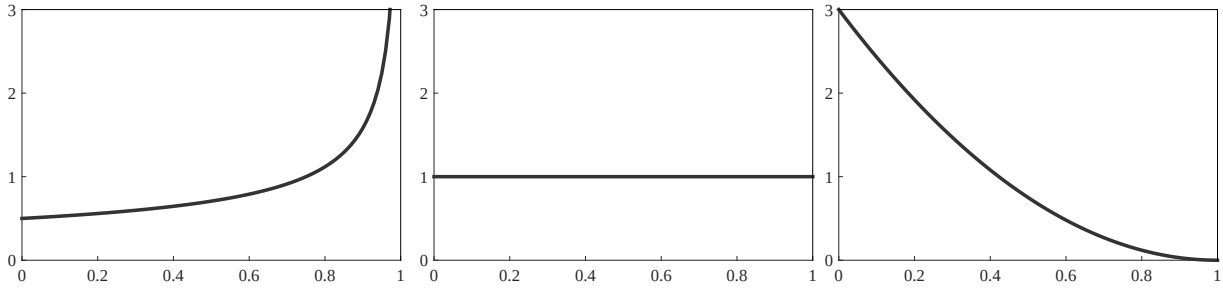
The Beta distribution is a continuous distribution taking values over the closed interval $[0,1]$. In this respect, it constitutes a natural candidate for any model representing probabilities or percentages. Typical applications include the probability of success for binary experiments (e.g. probability of obtaining "heads" at a coin toss), and the estimation of percentages (e.g. percentage of students who will pass the next examination).

The distribution is defined by two shape parameters a and b. The shape parameter a determines the behaviour of the distribution on the left, and the shape parameter b determines its behaviour on the right. Values of a or b below 1 curve the associated distribution tail upward, while values above 1 curve it downward. Values of 1 represent the neutral case, and when both a and b take a unit value, the Beta distribution degenerates into a uniform distribution. These features are illustrated by Figures d.19 and d.20:



(a) $a = 0.5$ (b) $a = 1$ (c) $a = 3$

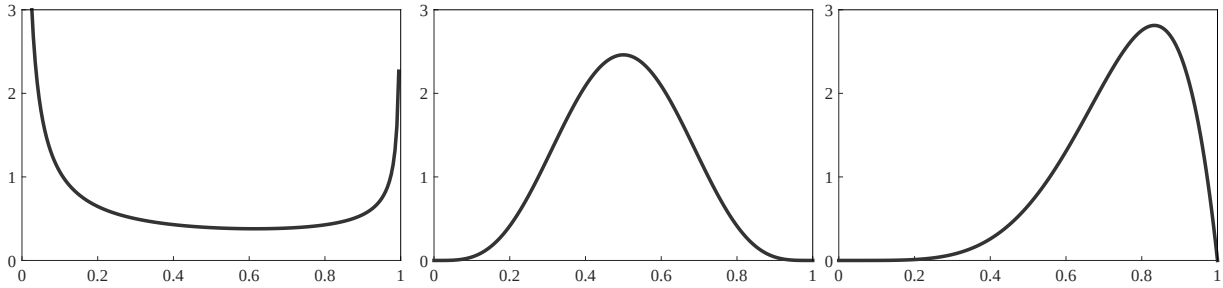
Figure d.19: Impact of the shape parameter a on the left tail (shape $b = 1$)



(a) $b = 0.5$ (b) $b = 1$ (c) $b = 3$

Figure d.20: Impact of the shape parameter b on the right tail (shape $a = 1$)

By playing on the two parameters a and b, a wide variety of shapes is available for the distribution. When $a = b$, the distribution is symmetric, while otherwise it is skewed. It is skewed to the left for $a < b$, and skewed to the right for $a > b$, as illustrated by Figure d.21:



(a) $a = 0.2, b = 0.5$ (b) $a = 5, b = 5$ (c) $a = 6, b = 2$

Figure d.21: Combinations of shapes a and b on the distribution

Playing on the parameters a and b, it is possible to generate a Beta distribution satisfying a pair of desired values for the mean and variance, if these values are compatible with the distribution. This is stated in the following property:

property d.27: let x be a random variable with: $x \sim Beta(a,b)$. Let μ and σ respectively denote a pair of desired values for the distribution mean and variance (with $0 < \mu < 1$ and $0 < \sigma < 0.25$). If this pair of values is compatible with the distribution, it can be obtained by defining:

$a = \frac{\bar{\mu} - \sigma(1+\bar{\mu})^2}{\sigma(1+\bar{\mu})^3}$ and $b = a\bar{\mu}$, $\bar{\mu} \equiv \frac{1-\mu}{\mu}$

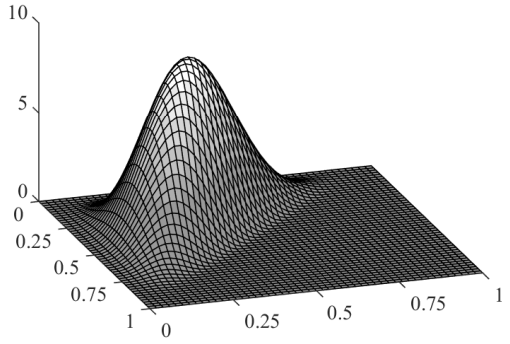
the following algorithm introduces the procedure to generate pseudo random numbers from the Beta distribution. The algorithm is standard, its motivation can be found for instance in Forbes et al. (2011).

algorithm d.32: random number generator for the Beta distribution

1. generate y from $y \sim G(a,1)$.
2. generate z from $z \sim G(b,1)$.
3. set $x = \frac{y}{y+z}$.

Then $x \sim Beta(a,b)$.

d.20. Dirichlet

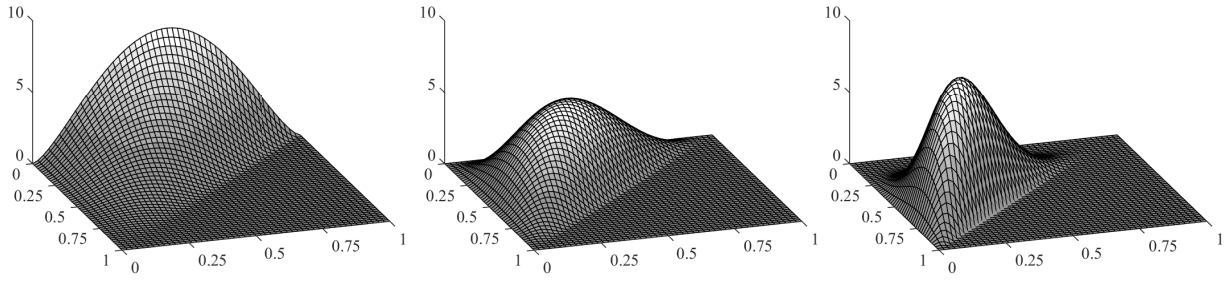


Type:	continuous
Notation:	$x \sim D(a_1, \ldots, a_k)$
Parameters:	$a_1, \ldots, a_n$ (concentration, scalars with $a_i > 0$, $i = 1, \ldots, n$)
Support:	$x_1, \ldots, x_n \in [0,1]$, with $\sum_{i=1}^{n} x_i = 1$
pdf:	$f(x_1, \ldots, x_n \mid a_1, \ldots, a_n) = \frac{1}{B(a_1, \ldots, a_n)} \prod_{i=1}^{n} x_i^{a_i - 1}$ $B(z_1, \ldots, z_n)$ is the multivariate Beta function, with $B(z_1, \ldots, z_n) = \frac{\Gamma(z_1) \ldots \Gamma(z_k)}{\Gamma(z_1 + \ldots + z_n)}$
Kernel:	$f(x_1, \ldots, x_n \mid a_1, \ldots, a_n) \propto \prod_{i=1}^{n} x_i^{a_i - 1}$
Normalizing constant:	$c = \frac{1}{B(a_1, \ldots, a_n)}$
Mean:	$\mathbb{E}(x_i) = \frac{a_i}{a}$ $\qquad a = \sum_{i=1}^{n} a_i$
Variance:	$Var(x_i) = \frac{a_i(a - a_i)}{a^2(a+1)}$
Median:	$\approx \frac{a_i - 1/3}{a - 2/3}$, for $a_i > 1$
Mode:	$\frac{a_i - 1}{a - n}$, for $a_i > 1$
Diffuse distribution:	set $a_1, \ldots, a_n \to 0$
Related distributions:	Beta: if $x \sim D(a_1, a_2)$, then $x \sim Beta(a_1, a_2)$

Table d.20: Summary of the Dirichlet distribution

The Dirichlet distribution generalizes the Beta distribution to n-dimensional random vectors. While the Beta distribution can be interpreted as producing probabilities or percentages of binary experiments, the Dirichlet distribution expands this settings to experiments with n different outcomes or categories, labelled as $x_1, \ldots, x_n$. For instance, a typical application of the Beta consists in determining the probability of success for a coin flip, which represents a binary experiment with two outcomes: heads (success) and tails (failure). A Dirichlet expansion might consist in studying the outcome of a 6-face die roll, determining the probabilities of obtaining any of the faces. The 6 faces then constitute the 6 categories considered by the distribution. Percentages can be treated in a similar way. While the Beta distribution can be used to determine the percentage of student that will pass or fail the next examination (binary experiment), the Dirichlet distribution can be used to determine the percentage of students corresponding to different grade categories (A, B, C, D, E and F, representing 6 categories).

The Dirichlet distribution is consistent with the Beta distribution. While the Beta deals with binary experiments and relies on a set of 2 shape parameters a and b, the Dirichlet considers n-categorical events with a set of n concentration parameters $a_1, \ldots, a_n$. The interpretation of these parameters is similar to that of the Beta distribution. Smaller values of a_i increase the concentration of probabilities on variable x_i, curving the distribution upward on the x_i axis at the expense of other variables. Larger values of a_i attribute less weight to probabilities on x_i and curve the distribution downward on the x_i axis. This is illustrated by Figure d.22:



(a) $a_1 = 1$ (b) $a_1 = 2.5$ (c) $a_1 = 5$

Figure d.22: Impact of concentration a_1 on the distribution ($n = 3, a_2 = a_3 = 2.5$)

The following algorithm introduces the procedure to generate pseudo random numbers from the Dirichlet distribution. Motivations can be found for instance in Forbes et al. (2011):

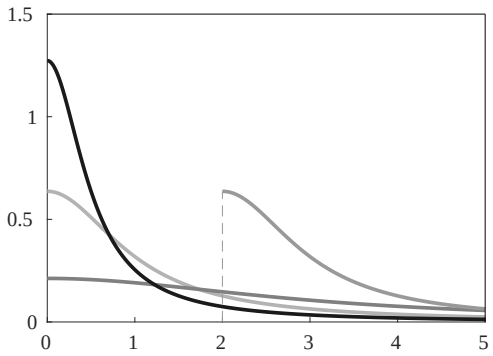
algorithm d.33: random number generator for the Dirichlet distribution

1. generate $z_1, \ldots, z_n$ from $z_i \sim G(a_i, 1)$, $i = 1, \ldots, n$.

2. for $i = 1, \ldots, n$, set $x_i = \frac{z_i}{z}$, with $z = \sum_{i=1}^{n} z_i$.

Then $x = (x_1, \ldots, x_n)$ is a random draw from $x \sim D(a_1, \ldots, a_n)$.

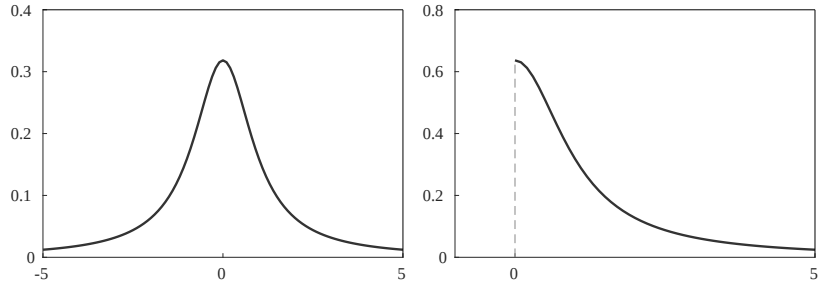
d.21. Half-Cauchy



Type:	continuous
Notation:	$x \sim \mathcal{C}^{+}(\mu, \sigma)$
Parameters:	μ (location, scalar) σ (scale, scalar with $\sigma > 0$)
Support:	$x \in [\mu, \infty)$
pdf:	$f(x\|\mu,\sigma) = \frac{2}{\pi\sigma} \frac{\mathbb{1}(x \geq \mu)}{1+(x-\mu)^2/\sigma^2}$
Kernel:	$f(x\|\mu,\sigma) \propto \frac{\mathbb{1}(x \geq \mu)}{1+(x-\mu)^2/\sigma^2}$
Normalizing constant:	$c = \frac{2}{\pi\sigma}$
Mean:	undefined
Variance:	undefined
Median:	$\mu + \sigma$
Mode:	μ
Diffuse distribution:	set $\sigma \to \infty$
Related distributions:	–

Table d.21: Summary of the half-Cauchy distribution

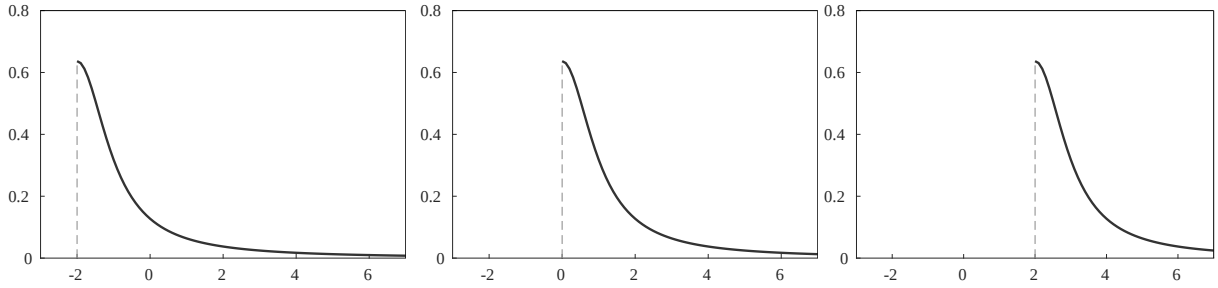
The half-Cauchy distribution is a continuous distribution that takes values over the support $[\mu, +\infty]$. It builds on the Cauchy distribution, a continuous bell-shaped distribution with support $[-\infty, +\infty]$, keeping only the values beyond the location parameter μ, as illustrated in Figure d.23:



(a) Cauchy (b) Half-Cauchy

Figure d.23: Cauchy and corresponding half-Cauchy distribution, $\mu = 0$

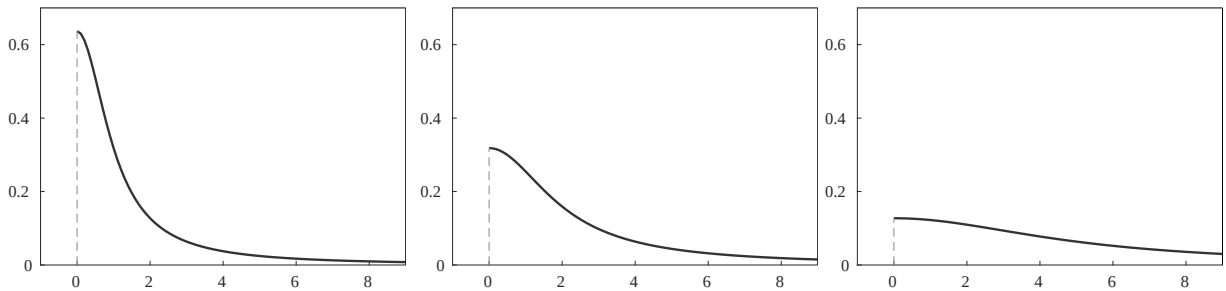
The half-Cauchy distribution is characterized by two parameters: the location μ, and the scale σ. The location μ can be any real value. It shifts the distribution leftwards (for negative values) or rightwards (for positive values) from 0. This is illustrated in Figure d.24:



(a) $\mu = -2$ (b) $\mu = 0$ (c) $\mu = 2$

Figure d.24: Impact of the location parameter μ

The scale parameter σ is a positive scalar which determines the spread of the distribution. The larger σ, the flatter the distribution, and the less informative it is. At large values of σ the distribution becomes almost flat and uniform over the $[\mu + \infty]$ orthant. This is illustrated in Figure d.25:



(a) $\sigma = 1$ (b) $\sigma = 2$ (c) $\sigma = 5$

Figure d.25: Impact of the scale parameter σ

The typical use of the half-Cauchy distribution is the case $\mu = 0$, for parameters taking only positive values. The half-Cauchy then becomes a substitute to the inverse Gamma distribution, with a number of convenient properties. First, the half Cauchy has its mode at 0, making it suitable for the modelling of tiny values. Also, as a special case of the family of Student distributions, the half-Cauchy is fat-tailed, allowing for better modelling of intermediate and large values.

In general, the half-Cauchy displays better behaviour than the inverse Gamma distribution whenever extreme values (small or large) are involved. The strong concentration of the inverse Gamma distribution makes it ill-behaved in this case, and makes it too sensitive to its parameterization. It can be showned that the usual uninformative values of the inverse Gamma distribution can prove in fact quite informative in a context of extreme values (Gelman (2006)). In this context, the half-Cauchy distribution usually achieves better predictive performance than the inverse Gamma (Yanyan (2017)).

The problem of the half-Cauchy distribution is that its probability density function is not conjugate with the usual distributions such as the normal family of distributions. For this reason, it is often necessary to rely on an alternative hierarchical formulation that uses the following result (Wand et al. (2011)):

property d.28: let x be a random variable with: $x \sim IG(1/2, 1/a)$, and let a be itself a random variable with: $a \sim IG(1/2, 1/\sigma^2)$. Then: $\mu + \sqrt{x} \sim \mathcal{C}^+(\mu, \sigma)$.

One can thus equivalently postulate a half-Cauchy distribution, or use a hierarchical inverse Gamma conditional on a second inverse Gamma distribution. This way conjugacy becomes easier, the inverse Gamma distribution being for instance conjugate with the widely used normal distribution.

CHAPTER K

Time-varying parameters

In this chapter we introduce a very important class of parameters: the dynamic (also called time-varying) parameters. Such parameters occur frequently in econometric models, and because their treatment is significantly more involving than the usual static parameters, they are worth dedicating a full chapter. We start by introducing some elementary concepts and the notion of state-space model, then develop on the canonical methodology of Kalman filtering. The Kalman filter is then extended to the Bayesian framework, and an applied example comes to illustrate the methodology at work. We finally introduce the precision sampler, an alternative to the Kalman filter, and conclude by outlining the pros and cons of the two methods.

k.1 Elementary concepts

This chapter is fundamentally concerned with dynamic parameters:

definition k.1: a **dynamic parameter**, also called a **time-varying parameter** is a parameter that takes a different value at each time period. It is opposed to a **static parameter** that takes the same value over all periods.

A dynamic parameter is typically denoted by θ_t, where the subscript t makes it unambiguous that the value depends on the time period. A static counterpart is on the contrary denoted by θ to make it explicit that the parameter is period-invariant.

example k.1: a dynamic linear regression
Consider the canonical linear regression in the context of time-series:

$$y_t = x_t\beta + \varepsilon_t \qquad \varepsilon_t \sim N(0,\sigma)$$

In this model, the vector of regression coefficients β is a static parameter. Consider now a more general linear regression of the form:

$$y_t = x_t\beta_t + \varepsilon_t \qquad \varepsilon_t \sim N(0,\sigma)$$

In this model, β_t is a dynamic parameter. It can take a different value at each period.

To make a dynamic model tractable, we usually need to characterize the evolution of the dynamic parameter θ_t over time.

definition k.2: let θ_t be a dynamic parameter; the **law of motion** of θ_t is the rule that sets the current value θ_t of the dynamic parameter as a function of its past values $\theta_{t-1}, \theta_{t-2}\cdots$, and possibly other parameters.

Consider again the linear regression example:

example k.1 (continued): Assume β_t is wharacterized by the following law of motion:
$\beta_t = (1-\gamma)\bar{\beta} + \gamma\beta_{t-1} + \zeta_t \qquad \zeta_t \sim N(0,Z) \qquad t = 1,\cdots,T$

γ is an autoregressive coefficient with $0 < \gamma < 1$ which guarantees the stability of the system. Stationarity then implies that the unconditional mean and variance of the process are $\mathbb{E}(\beta_t) = \bar{\beta}$ and $Var(\beta_t) = \frac{1}{1-\gamma^2}Z$. This simple law of motion thus defines β_t as a stationary process that fluctuates around an equilibrium value of $\bar{\beta}$, the extent to which β_t can vary over periods being determined by the error variance matrix Z.

k.2 State-space models

A convenient representation to estimate dynamic parameters is the so-called state-space system. We first provide the definition, then develop on the intuitions.

definition k.3: a **state-space model** is a system of the form:

$x_t = A_t z_t + \epsilon_t$	$\epsilon_t \sim N(0,\Omega_t)$	**observation equation**
$z_t = c_t + B_t z_{t-1} + \xi_t$	$\xi_t \sim N(0,\Upsilon_t)$	**state equation**

where x_t is an n-dimensional vector of **observed variable**, and z_t is a k-dimensional vector of so-called **state variable**. c_t, A_t and B_t are $k \times 1$, $n \times k$ and $k \times k$ vectors and matrices of coefficients. Finally, ϵ_t and ξ_t are independent vectors of Gaussian disturbances of respective dimension n and k, with associated variance-covariance matrices Ω_t and Υ_t.

Let us take a closer look at the system. The state variable z_t represents the unobserved dynamic variable that we want to estimate. It corresponds in fact to the dynamic parameter θ_t introduced in section k.1. The state equation in definition k.3 is then just a standard law of motion for z_t. In this general definition we allow A_t, Ω_t, c_t, B_t and Υ_t to be period-specific, but in practice these parameters can also be static.

Assume for now that only the state equation is available. Then predicting z_t is straightforward and can be accomplished recursively from the state equation since it is just a simple VAR(1) model. The whole point of state-space systems is that we can do better than this by exploiting the additional information obtained from the observation equation. In this equation, x_t is the vector of observed variables, that is, the variable whose value is known. The observation equation therefore sets a relation between z_t, the unobserved dynamic variable of interest, and x_t, an observed variable with known value.

We can exploit this relation to assess how good (or poor) the prediction from the state equation is. Indeed, assume that we form some prediction for z_t from the state equation, and that we then use this prediction in the observation equation. If the left-hand side of the observation equation (the observed value) is close to the right-hand-side (its forecast using z_t), then we may conlude that our initial prediction for z_t was accurate as it is supported by the observation equation. Conversely, if our prediction for z_t appears inconsistent with the observation equation, then it may represent a poor forecast and requires some correction in the light of the additional information provided by the observed variable.

This suggests a simple methodology to generate optimal predictions for z_t. First, get an initial prediction for z_t from the state equation; then compare the observed variable x_t with its forecast obtained from z_t in the observation equation; eventually use the discrepancy from the observation equation to correct the initial prediction for z_t. Intuitively, there is a role for the covariance matrices Ω_t and Υ_t in the correction step. A small variance in Υ_t (resp. Ω_t) implies an accurate prediction for the state (resp. observed) variable and thus a small (resp. large) correction from the observation equation.

This simple procedure in 3 steps constitutes the basis of the Kalman filter discussed in the incoming section. Before we formally introduce the Kalman filter, we illustrate the state-space formulation with our dynamic linear regression example.

example k.1 (continued): Consider again the dynamic linear regression:

$y_t = x_t\beta_t + \varepsilon_t \qquad \varepsilon_t \sim N(0,\sigma)$

It has law of motion:

$\beta_t = (1-\gamma)\bar{\beta} + \gamma\beta_{t-1} + \zeta_t \qquad \zeta_t \sim N(0,Z)$

A state-space representation obtains by noting that the first equation gives the observation equation while the law of motion constitutes the state equation, with the state-space parameters defined as:

$x_t = y_t \qquad z_t = \beta_t \qquad \epsilon_t = \varepsilon_t \qquad \xi_t = \zeta_t$

$A_t = x_t \qquad \Omega_t = \sigma \qquad c_t = (1-\gamma)\bar{\beta} \qquad B_t = \gamma \qquad \Upsilon_t = Z$

k.3 The Kalman filter

The Kalman filter is nothing more than a mathematical formalization of the intuitive procedure of state prediction, observation prediction, and state correction developed in previous section. It is a completely mechanical procedure made of 6 steps applied recursively to all sample periods $t = 1,\cdots,T$.

Before we introduce the Kalman filter, we need the following notations:

$x_{t|s} = \mathbb{E}(x_t|x_1,\cdots,x_s) \qquad z_{t|s} = \mathbb{E}(z_t|x_1,\cdots,x_s) \qquad \Omega_{t|s} = Var(x_t|x_1,\cdots,x_s) \qquad \Upsilon_{t|s} = Var(z_t|x_1,\cdots,x_s)$

We now detail the 6 Kalman steps without proving them (full derivations can be found in appendix).

algorithm k.1: The Kalman filter

For $t = 1,\cdots,T$, run the following steps:

step 1. state, prediction: $z_{t|t-1} = c_t + B_t z_{t-1|t-1}$

step 2. state, error variance: $\Upsilon_{t|t-1} = B_t\Upsilon_{t-1|t-1}B_t' + \Upsilon_t$

step 3. observed, prediction: $x_{t|t-1} = A_t z_{t|t-1}$

step 4. observed, error variance: $\Omega_{t|t-1} = A_t\Upsilon_{t|t-1}A_t' + \Omega_t$

step 5. state, prediction correction: $z_{t|t} = z_{t|t-1} + \Phi_t(x_t - x_{t|t-1})$

step 6. state, error variance correction: $\Upsilon_{t|t} = \Upsilon_{t|t-1} - \Phi_t\Omega_{t|t-1}\Phi_t'$

with : $\Phi_t = \Upsilon_{t|t-1}A_t'\Omega_{t|t-1}^{-1}$

Let us take a closer look at the algorithm. Steps 1 and 2 correspond to the state prediction stage, using only the state equation. They provide the optimal forecast and forecast error for z_t when only the state equation is known. Steps 3 and 4 consider the observation equation, computing the optimal prediction and prediction error for x_t, given the prediction for z_t. Steps 5 and 6 finally represent the correction that is applied to the initial state prediction and prediction error, once the discrepancy between x_t and its prediction is known. The term Φ_t provides the scale of the correction and can be seen to depend positively on $\Upsilon_{t|t-1}$ (imprecise state predictions yield large corrections) and negatively on $\Omega_{t|t-1}$ (accurate observation predictions suggest high confidence in the observed gap and thus large corrections).

The initial period (period $t = 1$) is particular as it involves the initial conditions $z_{0|0}$ and $\Upsilon_{0|0}$. Depending on the context, these initial conditions may or may not be known. When that is not the case, we can make the assumption that $z_{0|0} = 0$ and $\Upsilon_{0|0} = 0$, then define c_1 and Υ_1 to obtain a meaningful behaviour at the initiation of the recursion. To see this, let us return to the dynamic linear regression model:

example k.1 (continued): Consider again the dynamic linear regression:

$y_t = x_t\beta_t + \varepsilon_t \qquad \varepsilon_t \sim N(0,\sigma)$

Reformulate its law of motion as:

$\beta_t = (1-\gamma)\bar{\beta} + \gamma\beta_{t-1} + \zeta_t \qquad \zeta_t \sim N(0,Z) \qquad t = 2,\cdots,T$

$\beta_t = \bar{\beta} + \zeta_t \qquad \zeta_t \sim N(0,\tau Z) \qquad t = 1$

The expectation of the process at the initial period is simply set to the unconditional mean $\bar{\beta}$. The parameter $\tau > 1$ acts as an initial multiplier for the process variance. It is usually set to a large value such as $\tau = 5$ or 10. It contributes to align the initial period variance of the state with its long-run value (remember that $Var(\beta_t) = \frac{1}{1-\gamma^2}Z$). Also, setting a large initial variance permits significant correction from the observation equation, reducing the impact the impact of the initial value $\bar{\beta}$. In terms of state-space system, the above formulation implies:

$c_1 = \bar{\beta} \qquad \Upsilon_1 = \tau Z$

k.4 Application: has the Phillips curve changed in Australia?

As an illustration of the use of the Kalman filter, we propose a simple case study on the Phillips curve and its evolution in Australia. The exercise follows classical studies on the subject such as Blanchard et al. (2015).

In its simplest formulation, the Phillips curve writes as:

$\pi_t = \beta_{1,t}(u_t - u_t^n) + \beta_{2,t}\pi_t^e + \varepsilon_t \qquad \varepsilon_t \sim N(0,\sigma)$

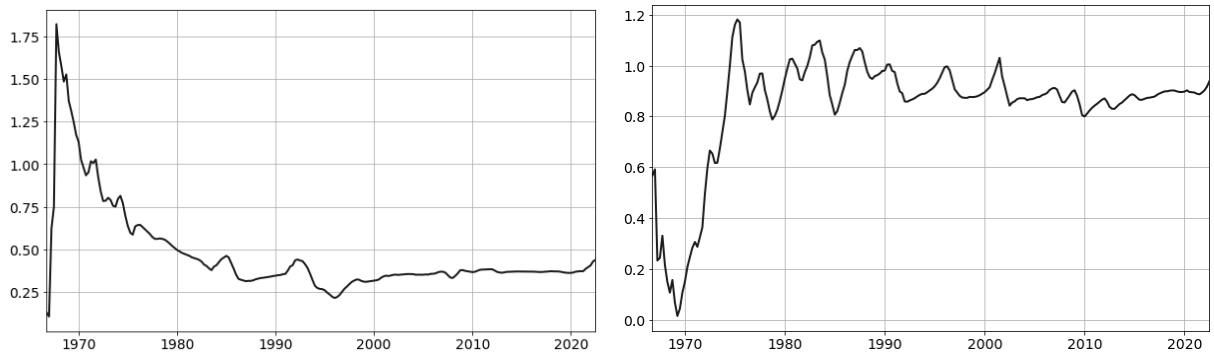
where u_t and u_t^n respectively denote the actual and natural rates of unemployment, and π_t^e is expected inflation at period t. The traditional Phillips curve postulates an inverse relation between inflation and unemployment, and thus the existence of a short term trade-off between the price level and real activity. It is however widely admitted that the Phillips curve trade-off may have weakened over the last decades. The case of Australia is especially relevant since in 1993 the RBA formally adopted an inflation targetting system which contributed to anchor inflation expectations and weaken the Phillips curve trade-off. Prior to this the RBA was conducting a discretionary policy, resulting in significantly more volatile inflation expectations.

To run the exercise, we estimate the Phillips curve on a quarterly sample running from 1966 to 2022. Data comes from the OECD and comprises the Australian CPI as well as a series of harmonized unemployment rate. In this crude exercise the natural level of unemployment is obtained by fitting a linear trend on the full unemployment sample, and inflation expectation is taken as previous year realised inflation.

We first estimate a static version of the model by maximum likelihood and obtain $\hat{\beta} = (-0.36 \;\; 0.94)$ and $\hat{\sigma} = 5.18$. The negative coefficient on the Phillips curve slope seems to confirm the existence of a trade-off in inflation and output, but does not reflect possible changes over the sample period. The value of roughly 1 for the sensitivity to inflation implies a so-called "accelerationist" Phillips curve where inflation expectations play a major role in the determination of actual inflation.

We then estimate the dynamic model from the time-varying linear regression introduced in example k.1, using the Kalman filter methodology developed in section k.3. For the model parameters, we set γ to 0.95, which implies a high degree of inertia but a stationary process for the coefficients. We set $\bar{\beta} = \hat{\beta}$ and $\Omega_t = 0.2\ \hat{\sigma}$ to reflect the fact that we expect the time-varying model to provide a better fit than its static counterpart. We set $\Upsilon_t = (0.02\ I_2)^2$, implying roughly a 5% coefficient variation from one period to another. Finally, the parameter τ is set to 10000, a large value justified by the short sample (224 observations) which necessitates an immediate correction of the initial condition.

Algorithm k.1 is then run for the model. The estimates for β_1 and β_2 are displayed in Figure k.1. The results confirm the two main conclusions of Blanchard et al. (2015). First, short-run inflation expectations have become more stable since the early 1970s, as shown by the increase in β_2 in panel (b). Sensitivity to inflation expectations has then become more stable since the early 1980's. Second, the slope of the Phillips curve β_1 has flattened over time, with nearly all of the decline taking place from the late-1960s to the mid 1990s, and the coefficient remaining roughly constant since then. Interestingly, the period of stability coincide with the adoption of the inflation targetting policy by the RBA. This brief exercise therefore confirms the weakening of the Phillips curve in developed economies such as Australia since the 1970s.



(a) Phillips curve slope β_1 (b) sensitivity to expected inflation β_2

Figure k.1: evolution of Phillips curve in Australia

k.5 Bayesian estimation of state-space models

The Kalman filter is suitable in a frequentist approach. In a Bayesian context however, one must derive the full posterior distribution of the state variable for all sample periods, conditional on the observed variables. Formally, one seeks to derive:

$$\pi(z|x) = \pi(z_1, \cdots, z_T | x_1, \cdots, x_T)$$

The posterior $\pi(z|x)$ can be recovered from a state-space formulation of the dynamic parameter. The state equation then represents the prior distribution (our prior belief about the behaviour of the dynamic parameter), while the observation equation provides the data information (the likelihood function) which, once observed, will update the prior belief into a posterior distribution. We follow here the classical approach proposed by Carter and Kohn (1994), which basically integrates the Kalman filter into a Gibbs sampling framework. As a first step,we rewrite the joint posterior $\pi(z|y)$ as (details of the derivations can be found in Appendix):

$$\pi(z|x) = \pi(z_T|x_T) \prod_{t=1}^{T-1} \pi(z_t|z_{t+1}, x_t)$$

In other words, it is possible to obtain a draw from $\pi(z|x)$ by first sampling a value from $\pi(z_T|x_T)$, and then recursively sampling values from $\pi(z_t|z_{t+1}, y_t)$, for $t = T-1, T-2, \cdots, 1$. This supposes yet that one can first recover the distributions $\pi(z_T|x_T)$ and $\pi(z_t|z_{t+1}, x_t)$. Carter and Kohn (1994) notice that this is easily done thanks to the Kalman filter. Concretely, the authors propose a two-step procedure. The first step constitutes the forward pass of the algorithm, which is just a regular Kalman filter. This step yields the distribution $\pi(z_T|x_T)$. The second step represents the backward pass, which obtains the distributions $\pi(z_t|z_{t+1}, x_t)$ from the forward pass elements.

We now introduce the algorithm. The details for the derivations of the formulas in the backward pass can be found in Appendix. We also need the following notations:

$\bar{z}_{t|s} = \mathbb{E}(z_t|z_s, x_1, \cdots, x_s) \qquad \bar{\Upsilon}_{t|s} = Var(z_t|z_s, x_1, \cdots, x_s)$

algorithm k.2: Gibbs sampling algorithm for state-space models (Carter-Kohn)

1. **Forward pass, Kalman filter:**

   for $t = 1, \cdots, T$, run the following steps:

   step 1. state, prediction: $z_{t|t-1} = c_t + B_t z_{t-1|t-1}$

   step 2. state, error variance: $\Upsilon_{t|t-1} = B_t' \Upsilon_{t-1|t-1} B_t + \Upsilon_t$

   step 3. observed, prediction: $x_{t|t-1} = A_t z_{t|t-1}$

   step 4. observed, error variance: $\Omega_{t|t-1} = A_t \Upsilon_{t|t-1} A_t' + \Omega_t$

   step 5. state, prediction correction: $z_{t|t} = z_{t|t-1} + \Phi_t (x_t - x_{t|t-1})$

   step 6. state, error variance correction: $\Upsilon_{t|t} = \Upsilon_{t|t-1} - \Phi_t \Omega_{t|t-1} \Phi_t'$

   with : $\Phi_t = \Upsilon_{t|t-1} A_t' \Omega_{t|t-1}^{-1}$

2. **Final period sampling:**

   sample z_T from $\pi(z_T|x_T) \sim N(z_{T|T}, \Upsilon_{T|T})$

3. **Backward pass:**

   for $t = T-1, T-2, \cdots, 1$, run the following steps:

   step 1. state, correction: $\bar{z}_{t|t+1} = z_{t|t} + \Xi_t (z_{t+1} - z_{t+1|t})$

   step 2. state, error variance correction: $\bar{\Upsilon}_{t|t+1} = \Upsilon_{t|t} - \Xi_t B_{t+1} \Upsilon_{t|t}$

   with : $\Xi_t = \Upsilon_{t|t} B_{t+1}' \Upsilon_{t+1|t}^{-1}$

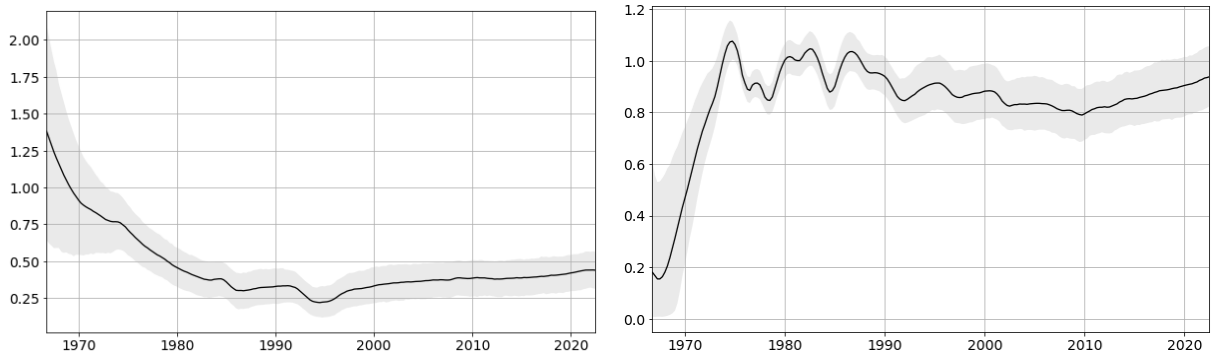
   step 3. sample z_t from: $\pi(z_t|z_{t|t+1}, y_t) \sim N(\bar{z}_{t|t+1}, \bar{\Upsilon}_{t|t+1})$

Using the algorithm creates a series of draws from $\pi(z_T|x_T)$, $\pi(z_{T-1}|z_T, x_{T-1})$, $\cdots$, $\pi(z_2|z_3, x_2)$, $\pi(z_1|z_2, x_1)$ which jointly constitute a draw from $\pi(z|x)$.

k.6 Application: revisiting the Phillips curve for Australia

We return to the Phillips curve example and apply now the Carter-Kohn algorithm to obtain the Bayesian estimates of the regression coefficients over the sample periods. The algorithm is run for 2000 iterations, using the same setting as in section k.4 with the regular Kalman filter. The results are displayed in Figure k.2. The solid line denotes the median, and the shaded bands provide the empricial 95% credibility intervals around the point estimates.

The results are consistent with those obtained in section k.6. In particular, they confirm the two main conclusions of the exercise: an increased sensitivity to inflation expectations since the late 1970s (panel (b)); and a weakening of the trade-off between the price level and real activity, with a stabilization of the Phillips curve slope in the early 1990s.



(a) Phillips curve slope β_1 (b) sensitivity to expected inflation β_2

Figure k.2: evolution of Phillips curve in Australia (Carter-Kohn algorithm)

Note the two benefits of the Bayesian approach. First, the initial jumps of the coefficient observed at the beginning of the sample in Figure k.1 are absent of the Bayesian estimates. Working with posterior distributions and loose priors for the initial period considerably moderate the impact of the initial conditions, eliminating the presence of a transiton path over the first few observations. Second, the Gibbs sampler provides a full numerical posterior distribution, which makes it trivially simple to obtain credibility intervals around the estimates.

k.7 An alternative Bayesian approach: the precision sampler

The Carter and Kohn algorithm exploits the Kalman filter approach to derive the posterior distributions of dynamic parameters in a Bayesian context. While classical, the methodology is sophisticated and may look hard to monitor. An elegant alternative for Bayesian models is the so-called precision sampler. The methodology is also classical and is detailed in textbooks such as Greenberg (2008). It recently regained in popularity with the contributions of Chan and Jeliazkov (2009) and Chan and Eisenstat (2018) and is now widely used in dynamic models.

The precision sampler essentially consists in a joint reformulation of the dynamic parameters, which reduces a multiple period problem into a standard static estimation. To see this, consider the state-space formulation k.3 and note that the observation equation can be stacked for all periods $t = 1, 2, \cdots, T$ as:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_T \end{pmatrix} = \begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_T \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_T \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_T \end{pmatrix} \qquad \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_T \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \Omega_1 & 0 & \cdots & 0 \\ 0 & \Omega_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Omega_T \end{pmatrix} \right)$$

Or, in compact formulation:

$$x = Az + \epsilon \qquad \epsilon \sim N(0, \Omega)$$

This formulation gathers the state variables $z_1, \cdots, z_T$ for all periods in a single vector z that can thus be treated as a single static parameter. Adopting a similar strategy, the state equation reformulates for all periods $t = 1, 2, \cdots, T$ at once as:

$$\begin{pmatrix} I & 0 & \cdots & 0 \\ -B_2 & I & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -B_T & I \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_T \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_T \end{pmatrix} + \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_T \end{pmatrix} \qquad \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_T \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \Upsilon_1 & 0 & \cdots & 0 \\ 0 & \Upsilon_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Upsilon_T \end{pmatrix} \right)$$

where we have made use of the assumption $z_0 = 0$ in the state-space formulation. Rewriting compactly:

$$F\ z = c + \xi \qquad\qquad \xi \sim N(0, \Upsilon)$$

Finally, inverting F on the left-hand side:

$$z = F^{-1}\ c + F^{-1}\ \xi \qquad\qquad \xi \sim N(0, \Upsilon)$$

The vector z is thus expressed as the sum of a constant (the term $F^{-1}\ c$) and a linear combination of Gaussian disturbances (the term $F^{-1}\ \xi$). It then follows immediately from property d.2 that:

$$\pi(z) \sim N(F^{-1}\ c,\ F^{-1}\Upsilon F^{-1\prime})$$

We effectively reformulated a multiple period problem into a static one. The compact formulation of the observation equation gives us the likelihood function, while that of the state equation provides the prior distribution for z. Combining the two with Bayes rule and completing the squares, the posterior immediately obtains as:

$$\pi(z|x) \sim N(\bar{B}, \bar{\Upsilon}) \qquad \bar{\Upsilon}^{-1} = (F'\Upsilon^{-1}F + A'\Omega^{-1}A) \qquad \bar{B} = \bar{\Upsilon}(F'\Upsilon^{-1}c + A'\Omega^{-1}x)$$

The critical stage comes with the inversion of the precision matrix $\bar{\Upsilon}^{-1}$, required to calculate $\bar{B}$. As this matrix is very large, inversion can become numerically challenging. Fortunately, it is possible to exploit the specific structure of $\bar{\Upsilon}^{-1}$. Indeed, note that the $F'\Upsilon^{-1}F$ part in $\bar{\Upsilon}^{-1}$ is given by:

$$F'\Upsilon^{-1}F = \begin{pmatrix} \Upsilon_1^{-1} + B_2'\Upsilon_2^{-1}B_2 & -B_2'\Upsilon_2^{-1} & & & & \\ -\Upsilon_2^{-1}b_2 & \Upsilon_2^{-1} + B_3'\Upsilon_3^{-1}B_3 & -B_3'\Upsilon_3^{-1} & & & \\ & -\Upsilon_3^{-1}B_3 & \Upsilon_3^{-1} + B_4'\Upsilon_4^{-1}B_4 & -B_4'\Upsilon_4^{-1} & & \\ & & \ddots & \ddots & \ddots & \\ & & & & \Upsilon_{T-1}^{-1} + B_T'\Upsilon_T^{-1}B_T & -B_T'\Upsilon_T^{-1} \\ & & & & -\Upsilon_T^{-1}B_T & \Upsilon_T^{-1} \end{pmatrix}$$

Also, the $A'\Omega^{-1}A$ part of $\bar{\Upsilon}^{-1}$ is given by:

$$A'\Omega^{-1}A = \begin{pmatrix} A_1'\Omega_1^{-1}A_1 & & & \\ & A_2'\Omega_2^{-1}A_2 & & \\ & & \ddots & \\ & & & A_T'\Omega_T^{-1}A_T \end{pmatrix}$$

These two matrices are (block) banded, that is, they contain only a small number of non-zero elements concentrated in a narrow band around the main diagonal. Following, the full precision matrix $\bar{\Upsilon}^{-1}$ is also banded. This fact is key as there exist specific algorithms that can efficiently operate on such banded matrices, especially for the purpose of Cholesky factorization.

The next step consists in noting that a draw from $\pi(z|x) \sim N(\bar{B}, \bar{\Upsilon})$ can be obtained by computing:

$$z = \bar{B} + \bar{\xi} \qquad\qquad \bar{\xi} \sim N(0, \bar{\Upsilon})$$

Denote by G the lower Cholesky factor of the precision matrix $\bar{\Upsilon}^{-1}$. That is, G is a lower triangular matrix such that $GG' = \bar{\Upsilon}^{-1}$. Note then that $\bar{\Upsilon} = (GG')^{-1} = G^{-1\prime}G^{-1}$, which implies in turn that a draw from $\pi(z|x) \sim N(\bar{B}, \bar{\Upsilon})$ can be obtained by computing:

$$z = G^{-1\prime}G^{-1}(F'\Upsilon^{-1}c + A'\Omega^{-1}x) + G^{-1\prime}\zeta \qquad\qquad \zeta \sim N(0, I)$$

Then, factoring $G^{-1\prime}$:

$$z = G^{-1\prime}\left[G^{-1}(F'\Upsilon^{-1}c + A'\Omega^{-1}x) + \zeta\right] \qquad\qquad \zeta \sim N(0, I)$$

This is where the precision sampler comes handy. The first trick consists in exploiting the banded nature of $\bar{\Upsilon}^{-1}$ and use specific block-banded algorithms to compute efficiently the Cholesky factor G. Such algorithms are routinely provided by numerical softwares like Matlab or Numpy. The second trick consists in noting that G is triangular so that G^{-1} and $G^{-1\prime}$ can be computed efficiently by forward and backward substitution. This way no explicit inversion is required and one can sample directly from the the precision matrix $\bar{\Upsilon}^{-1}$ (hence the name "precision sampler").

This eventually yields the following algorithm:

algorithm k.3: Gibbs sampling algorithm for state-space models (precision sampler)

1. obtain the precision matrix $\bar{\Upsilon}^{-1}$ from:
   $\bar{\Upsilon}^{-1} = (F'\Upsilon^{-1}F + A'\Omega^{-1}A)$
2. compute G, the Cholesky factor of $\bar{\Upsilon}^{-1}$ such that $GG' = \bar{\Upsilon}^{-1}$, using an efficient algorithm for banded matrices.
3. sample ζ from $\zeta \sim N(0,I)$.
4. solve for $z = G^{-1\prime}\left[G^{-1}(F'\Upsilon^{-1}c + A'\Omega^{-1}x) + \zeta\right]$ efficiently by forward and backward substitution.

The precision sampler thus reduces a complicated dynamic problem into a regular static Bayesian estimation. It also exploits the specific banded nature of the precision matrix to make sampling efficient.

example k.1 (continued): Consider again the dynamic linear regression:

$$y_t = x_t\beta_t + \varepsilon_t \qquad \varepsilon_t \sim N(0,\sigma)$$

This rewrites in compact form as:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{pmatrix} = \begin{pmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & x_T \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_T \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_T \end{pmatrix} \qquad \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_T \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma & 0 & \cdots & 0 \\ 0 & \sigma & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma \end{pmatrix} \right)$$

Or:

$$y = X\beta + \varepsilon \qquad \varepsilon \sim N(0,\sigma I_T)$$

It follows immediately that the likelihood function obtains as:

$$f(y|\beta,\sigma) = (2\pi\sigma)^{-n/2} \exp\left(-\frac{1}{2}\frac{(y-X\beta)'(y-X\beta)}{\sigma}\right)$$

Assume again for simplicity that σ is known and set to $\sigma = \hat{\sigma}$, the maximum likelihood estimate. Then only β remains to be estimated. To compute its prior, consider again its law of motion:

$$\begin{aligned} &\beta_t = (1-\gamma)\bar{\beta} + \gamma\beta_{t-1} + \zeta_t && \zeta_t \sim N(0,Z) && t = 2,\cdots,T \\ &\beta_t = \bar{\beta} + \zeta_t && \zeta_t \sim N(0,\tau Z) && t = 1 \end{aligned}$$

It rewrites in compact form as:

$$\begin{pmatrix} I_k & 0 & \cdots & 0 \\ -\gamma I_k & I_k & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -\gamma I_k & I_k \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_T \end{pmatrix} = \begin{pmatrix} \bar{\beta} \\ (1-\gamma)\bar{\beta} \\ \vdots \\ (1-\gamma)\bar{\beta} \end{pmatrix} + \begin{pmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_T \end{pmatrix} \qquad \begin{pmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_T \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \tau Z & 0 & \cdots & 0 \\ 0 & Z & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & Z \end{pmatrix} \right)$$

Or:

$$(H \otimes I_k)\beta = c + \zeta \qquad \zeta \sim N(0, I_\tau \otimes Z)$$

with:

$$H = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -\gamma & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -\gamma & 1 \end{pmatrix} \qquad c = \begin{pmatrix} \bar{\beta} \\ (1-\gamma)\bar{\beta} \\ \vdots \\ (1-\gamma)\bar{\beta} \end{pmatrix} \qquad \zeta = \begin{pmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_T \end{pmatrix} \qquad I_\tau = \begin{pmatrix} \tau & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Inverting the expression yields $\beta = (H \otimes I_k)^{-1}c + (H \otimes I_k)^{-1}\zeta$. Eventually using property d.2 and rearranging, the prior distribution for β obtains as:

$$\pi(\beta) \sim N(b, V) \qquad b = (H^{-1} \otimes I_k)c \qquad V = (H' I_{-\tau} H)^{-1} \otimes Z \qquad I_{-\tau} = I_\tau^{-1}$$

Following:

$$\pi(\beta) = (2\pi)^{-k/2}|V|^{-1/2}\exp\left(-\frac{1}{2}(\beta - b)'V^{-1}(\beta - b)\right)$$

Using Bayes rule and following the usual steps of completing the squares, it comes immediately that the posterior for β is $\pi(\beta|y) \sim N(\bar{b}, \bar{V})$, with:

$$\bar{V} = (V^{-1} + \sigma^{-1}X'X)^{-1} \qquad \bar{b} = \bar{V}(V^{-1}b + \sigma^{-1}X'y)$$

Note the particularly appealing form of $V^{-1} = H' I_{-\tau} H \otimes Z^{-1}$ which only requires a single inversion of the low-dimensional matrix Z. With these elements, one can eventually use algorithm k.3 to sample values from $\pi(\beta|y)$.

The time-varying regression has been converted into a static Bayesian regression with standard posterior formulas. Using point estimates and credibility intervals over the posterior, one can immediately recover the representation in Figure k.2.

k.8 Kalman filter and precision sampler: a discussion

The Carter-Kohn algorithm and the precision sampler represent two different methods to sample from the posterior distribution of dynamic parameters. It then seems natural to determine which of the two approaches is most suitable for a Bayesian researcher. On the theoretical side, the precision sampler may look more appealing. It converts a complex time-varying formulation into a regular static problem. Bayes rule then applies without further ado. By contrast, the Carter-Kohn algortihm and its construction based on Kalman filtering is arguably more abstract and looks like a "blackbox" based mostly on matrix algebra.

Beyond these aspects, however, the key criterion is numerical efficiency. Following Chan and Jeliazkov (2009), the precision sampler has been widely accepted as the most efficient approach and the old Carter and Kohn (1994) methodology has been deemed to be outdated. To verify this fact, we run a very general exercise. We start from the state-space model formulation in definition k.3 and check which of the two methdologies is faster at generating posterior draws for the state variable.

Specifically, we start from given values of x_t, A_t, B_t, c_t, Ω_t, and Υ_t and calculate how much time it takes with both methods to produce a posterior draw $\pi(z_1, \cdots, z_T|x_1, \cdots, x_T)$. The parameter values used for the exercise are numerically simulated. We set Ω_t and Υ_t to be time-invariant, as this is the case in most econometrics applications. Also, in a attempt to be fair, we try to run each methodology in the most efficient way. For instance, the precision sampler works with the precision matrix

$\bar{\Upsilon}^{-1} = (F'\Upsilon^{-1}F + A'\Omega^{-1}A)$ and the posterior mean $\bar{B} = \bar{\Upsilon}(F'\Upsilon^{-1}c + A'\Omega^{-1}x)$. To compute Υ^{-1} and Ω^{-1}, we first invert the smaller (and time-invariant) matrices Υ_t and Ω_t, then reconstitute $\Upsilon^{-1} = I_T \otimes \Upsilon_t^{-1}$ and $\Omega^{-1} = I_T \otimes \Omega_t^{-1}$, where I_T is taken as the sparse identity matrix of dimension T to make the kronecker product as fast as possible. The products $F'\Upsilon^{-1}$ and $A'\Omega^{-1}$ are also pre-computed since they are redundant in the precision and posterior mean matrices.

The exercise is conducted for different values of n (the dimension of the observed variable), k (the dimension of the time-varying parameter) and T (the sample size), each time for 1000 replications of the Gibbs sampler. Table k.1 reports the results. The table does not report the absolute running times, but rather the ratio of the running times between the precision sampler and the Carter-Kohn algorithm. Values below 1 thus imply that the precision sampler is the most efficient approach, while values above 1 (in bold) indicate that the Carter-Kohn algorithm is the fastest method.

	$n=1$	$n=3$	$n=5$	$n=10$	$n=15$	$n=20$	$n=30$
$k=1$	0.36	0.13	0.14	0.18	0.30	0.23	0.31
$k=3$	0.13	0.10	0.12	0.15	0.16	0.23	0.33
$k=5$	0.27	0.23	0.21	0.24	0.31	0.34	0.56
$k=10$	0.77	0.58	0.56	0.62	0.73	0.79	0.98
$k=15$	**1.55**	**1.19**	**1.25**	**1.30**	**1.37**	**1.39**	**1.57**
$k=20$	**2.65**	**2.11**	**1.98**	**2.21**	**2.32**	**2.38**	**2.66**
$k=30$	**3.42**	**3.04**	**3.07**	**3.10**	**3.15**	**3.19**	**2.71**

(a) Comparison for $T = 100$

	$n=1$	$n=3$	$n=5$	$n=10$	$n=15$	$n=20$	$n=30$
$k=1$	0.67	0.17	0.18	0.21	0.27	0.27	0.37
$k=3$	0.16	0.12	0.13	0.16	0.20	0.25	0.36
$k=5$	0.29	0.22	0.23	0.25	0.33	0.37	0.52
$k=10$	0.83	0.62	0.63	0.71	0.75	0.81	0.95
$k=15$	**1.69**	**1.35**	**1.29**	**1.37**	**1.45**	**1.46**	**1.60**
$k=20$	**2.70**	**2.20**	**2.35**	**2.25**	**2.43**	**2.42**	**2.56**
$k=30$	**3.47**	**3.10**	**3.14**	**3.13**	**3.24**	**3.22**	**2.88**

(b) Comparison for $T = 500$

	$n=1$	$n=3$	$n=5$	$n=10$	$n=15$	$n=20$	$n=30$
$k=1$	**2.09**	0.61	0.61	0.57	0.60	0.48	0.49
$k=3$	0.40	0.28	0.28	0.29	0.32	0.34	0.43
$k=5$	0.51	0.37	0.38	0.37	0.45	0.46	0.59
$k=10$	**1.01**	0.73	0.71	0.81	0.84	0.89	**1.01**
$k=15$	**1.74**	**1.39**	**1.40**	**1.45**	**1.50**	**1.50**	**1.56**
$k=20$	**2.96**	**2.39**	**2.42**	**2.36**	**2.51**	**2.63**	**2.67**
$k=30$	**4.21**	**3.74**	**3.27**	**3.83**	**3.85**	**3.82**	**3.40**

(c) Comparison for $T = 2000$

Table k.1: Computational efficiency: Carter-Kohn algorithm V.S. precision sampler

The table reveals some interesting results. First and foremost, the precision sampler is not necessarily the most efficient methodology. The main criterion to discriminate between the two methods appears to be the dynamic parameter dimension k. Below $k = 15$, the precision sampler consistently beats the Carter-Kohn algorithm. The smaller the state dimension k, the better the performance compared to the Kalman filtering approach. Conversely, as k gets larger than 15 the precision sampler gets more and more inefficient, becoming in fact very inefficient for large values of k.

To understand this, we need additional insights on the computational cost of the precision sampler. The algorithm essentially consists in creating a sparse precision matrix, then computing its Cholesky factor taking advantage of its banded structure before eventually inverting it from back and forward substitution. Boyd and Vandenberghe (2004, p.590) show that the total computational cost of these operations is $c(\bar{\Upsilon}^{-1}) + Tk^2$ flops, where $c(\bar{\Upsilon}^{-1})$ denotes the cost for creating the precision matrix $\bar{\Upsilon}^{-1}$. Hence there are two potential sources of inefficiency for the algorithm: the computational cost $c(\bar{\Upsilon}^{-1})$ that can be potentially large; and the dimension k of the dynamic parameter that renders the precision sampler inefficient at a *quadratic* rate.

Contributions like Chan and Jeliazkov (2009), Chan (2013) and Chan and Eisenstat (2018) all conclude to the efficiency of the precision sampler. There are yet two issues with the way they deal with the problem. First, these papers only consider small models. For instance, Chan and Jeliazkov (2009) use a tiny VAR with 4 variables and a single lag so that $k = 16$. For larger models, the algorithm would quickly become inefficient. Second, the discussions systematically ignore the computational cost $c(\bar{\Upsilon}^{-1})$ and focus only on the factorization/inversion part of the algorithm. Yet this cost is an important component of the methodology and must be taken into account to produce meaningful efficiency comparisons. When done correctly, the comparison is less favourable to the precision sampler, as shown in Table k.1.

The second conclusion from the exercise is that the impact of n (the dimension of the observed variable) is relatively marginal. The tables do show a small loss of efficiency of the precision sampler as n increases, but the effect remains small overall. This is intuitive as n does not intervene directly in the factorization/inversion cost of Tk^2 flops, and thus only impact the algorithm indirectly through the computation cost $c(\bar{\Upsilon}^{-1})$ of the precision matrix.

The third conclusion is that the sample size T adversely impacts the performance of the precision sampler. The effect is stronger at small dimensions of k and n and attenuates at larger dimensions. The efficiency loss is not due to the factorization/inversion part that costs Tk^2 flops and hence shows that the precision sampler is linear in cost in the sample size T. Rather, it is due to the relative increase in the compution cost $c(\bar{\Upsilon}^{-1})$, as larger samples involve more (inefficient) algebraic computations to generate the sparse precision matrix $\bar{\Upsilon}^{-1}$.

Finally, the case $k = 1$ and $n = 1$ deserves specific attention. This configuration systematically improves the relative performance of the Carter-Kohn algorithm. For $T = 2000$, it even makes the Carter-Kohn approach more than twice as efficient as the precision sampler. How can this be? The answer is that $k = 1$ and $n = 1$ creates a very specific situation where the Kalman filter becomes entirely scalar-valued. Following, all linear algebra operations are replaced by regular scalar operations: Cholesky factorization becomes a square root, matrix inversion turns into a simple division, and so on. Because these base operations are extremely fast, the performance of the Kalman filter is greatly enhanced while the precision sampler still faces the costs of dealing with large sparse matrices.

A word of caution: these results may be to some extent computer-, software- and model-dependent. There are ways to improve at the margin the performance of one algorithm or the other, and this may slightly affect the conclusions. For instance, Chan and Jeliazkov (2009) assume that the matrices Υ_t are diagonal. This reduces the band width of the precision matrix $\bar{\Upsilon}^{-1}$ and mildly improves the efficiency of the precision sampler.

Nevertheless, two major conclusions remain:

1. The precision sampler gets inefficient in k at a quadratic rate.
2. At some dimension around $k = 15$, the Carter-Kohn approach becomes in fact more efficient.

The final question is thus: how likely is it that k is equal or larger then 15? The answer is: it depends. To see this, consider a few possible cases:

1. A dynamic linear regression like the one in example k.1. In this case $n = 1$ and $k = p$, the number of regressors. As it is quite common in econometrics to deal with models with $p < 15$, the precision sampler may be efficient in this case.

2. A Bayesian VAR. Considering even a small model with $n = 5$ endogenous variables, a constant and 6 lags, one would obtain $k = 155$ VAR coefficients in the model. In this case, the precision sampler would become extremely inefficient, and possibly even intractable as it would imply days or weeks to produce just 1000 replications.

3. A univariate model with stochastic volatility. In this case, the time-varying error variance implies $n = 1$ and $k = 1$. The precision sampler is efficient, but the Carter-Kohn algorithm becomes scalar-valued. If T is sufficiently large, the precision sampler may still become less efficient than the Kalman filter approach.

In conclusion, this section tried to build a systematic approach to both methods. The final picture is more mixed than the usual belief of greater efficiency of the precision sampler. Both algorithms can be efficient depending on the context, and using one or the other requires some understanding of their characteristics.

Appendix: details of the derivations of the Kalman filter

proof of step 1:

$$
\begin{aligned}
& z_{t|t-1} \\
= \; & \mathbb{E}(z_t|x_{t-1}) \\
= \; & \mathbb{E}(c_t + B_t z_{t-1} + \xi_t|x_{t-1}) \\
= \; & \mathbb{E}(c_t|x_{t-1}) + B_t\,\mathbb{E}(z_{t-1}|x_{t-1}) + \mathbb{E}(\xi_t|x_{t-1}) \\
= \; & c_t + B_t z_{t-1|t-1}
\end{aligned}
$$

proof of step 2:

$$
\begin{aligned}
& z_t - z_{t|t-1} \\
= \; & (c_t + B_t z_{t-1} + \xi_t) - (c_t + B_t z_{t-1|t-1}) \\
= \; & B_t(z_{t-1} - z_{t-1|t-1}) + \xi_t
\end{aligned}
$$

Following:

$$
\begin{aligned}
& \Upsilon_{t|t-1} \\
= \; & \mathbb{E}\left[(z_t - z_{t|t-1})(z_t - z_{t|t-1})'\right] \\
= \; & \mathbb{E}\left[(B_t(z_{t-1} - z_{t-1|t-1}) + \xi_t)((z_{t-1} - z_{t-1|t-1})'B_t' + \xi_t')\right] \\
= \; & \mathbb{E}\left[B_t(z_{t-1} - z_{t-1|t-1})(z_{t-1} - z_{t-1|t-1})'B_t' + \xi_t\xi_t' + 2B_t(z_{t-1} - z_{t-1|t-1})\xi_t'\right] \\
= \; & B_t\,\mathbb{E}\left[(z_{t-1} - z_{t-1|t-1})(z_{t-1} - z_{t-1|t-1})'\right]B_t' + \mathbb{E}\left[\xi_t\xi_t'\right] + 2B_t\,\mathbb{E}\left[(z_{t-1} - z_{t-1|t-1})\xi_t'\right] \\
= \; & B_t\Upsilon_{t-1|t-1}B_t' + \Upsilon_t
\end{aligned}
$$

proof of step 3:

$$
\begin{aligned}
& x_{t|t-1} \\
= \; & \mathbb{E}(x_t|x_{t-1}) \\
= \; & \mathbb{E}(A_t z_t + \epsilon_t\,|x_{t-1}) \\
= \; & A_t\,\mathbb{E}(z_t|x_{t-1}) + \mathbb{E}(\epsilon_t\,|x_{t-1}) \\
= \; & A_t z_{t|t-1}
\end{aligned}
$$

proof of step 4:

$$
\begin{aligned}
& x_t - x_{t|t-1} \\
= \; & (A_t z_t + \epsilon_t) - (A_t z_{t|t-1}) \\
= \; & A_t(z_t - z_{t|t-1}) + \epsilon_t
\end{aligned}
$$

Following:

$$
\begin{aligned}
& \Omega_{t|t-1} \\
= \; & \mathbb{E}\left[(x_t - x_{t|t-1})(x_t - x_{t|t-1})'\right] \\
= \; & \mathbb{E}\left[(A_t(z_t - z_{t|t-1}) + \epsilon_t)((z_t - z_{t|t-1})'A_t' + \epsilon_t')\right] \\
= \; & \mathbb{E}\left[(A_t(z_t - z_{t|t-1})(z_t - z_{t|t-1})'A_t' + \epsilon_t\epsilon_t' + 2A_t(z_t - z_{t-1|t-1})\,\epsilon_t'\right] \\
= \; & A_t\,\mathbb{E}\left[(z_t - z_{t|t-1})(z_t - z_{t|t-1})'\right]A_t' + \mathbb{E}\left[\epsilon_t\epsilon_t'\right] + 2A_t\,\mathbb{E}\left[(z_t - z_{t|t-1})\,\epsilon_t'\right] \\
= \; & A_t\Upsilon_{t|t-1}A_t' + \Omega_t
\end{aligned}
$$

proof of steps 5 and 6:

These two steps are difficult to prove. We want to derive $z_{t|t}$ and $\Upsilon_{t|t}$ which respectively represent the mean and variance of the distribution $\pi(z_t|x_t)$. The difficult task consists in deriving the distribution $\pi(z_t|x_t)$. To do so, we need a first intermediary result that states that $\pi(z_t|x_t) = \pi((z_t|x_{t-1})|(x_t|x_{t-1}))$. This is established from:

$$\begin{aligned}
&\pi(z_t|x_t) \\
=\ &\pi(z_t|x_t, x_{t-1}) = \frac{\pi(z_t, x_t, x_{t-1})}{\pi(x_t, x_{t-1})} = \frac{\pi(z_t, x_t, x_{t-1})}{\pi(x_{t-1})} \frac{\pi(x_{t-1})}{\pi(x_t, x_{t-1})} = \frac{\pi(z_t, x_t|x_{t-1})}{\pi(x_t|x_{t-1})} = \frac{\pi(z_t|x_{t-1}, x_t|x_{t-1})}{\pi(x_t|x_{t-1})} \\
=\ &\pi((z_t|x_{t-1})|(x_t|x_{t-1}))
\end{aligned}$$

This may not look very helpful as we replaced the simple expression $\pi(z_t|x_t)$ with the more complicated expression $\pi((z_t|x_{t-1})|(x_t|x_{t-1}))$. As will become clear shortly however, this is the formulation we need to obtain what we want. The next step consists in obtaining the joint distribution of $z_t|x_{t-1}$ and $x_t|x_{t-1}$. By definition, we have $z_t|x_{t-1} \sim N(z_{t|t-1}, \Upsilon_{t|t-1})$ and $x_t|x_{t-1} \sim N(x_{t|t-1}, \Omega_{t|t-1})$. As the two variables are Gaussian, their joint distribution is Gaussian as well. And since we know their mean and variances, it only remains to determine their covariance to complete the joint distribution. To do so, note that:

$$\begin{aligned}
&\Psi_{t|t-1} \\
=\ &\mathbb{E}\left[(z_t - z_{t|t-1})(x_t - x_{t|t-1})'\right] \\
=\ &\mathbb{E}\left[(z_t - z_{t|t-1})((z_t - z_{t|t-1})'A_t' + \epsilon_t')\right] \\
=\ &\mathbb{E}\left[(z_t - z_{t|t-1})((z_t - z_{t|t-1})'\right] A_t' + \mathbb{E}\left[(z_t - z_{t|t-1})\,\epsilon_t'\right] \\
=\ &\Upsilon_{t|t-1}A_t'
\end{aligned}$$

Therefore, the full joint distribution is given by:

$$\begin{pmatrix} z_t|x_{t-1} \\ x_t|x_{t-1} \end{pmatrix} \sim N\left(\begin{bmatrix} z_{t|t-1} \\ x_{t|t-1} \end{bmatrix}, \begin{bmatrix} \Upsilon_{t|t-1} & \Psi_{t|t-1} \\ \Psi_{t|t-1}' & \Omega_{t|t-1} \end{bmatrix}\right)$$

We now use the second intermediary result that we need, which is property d.6 relative to the conditional distributions of multivariate normal distributions. From this property, it follows immediately that:

$$\pi(z_t|x_t) = \pi((z_t|x_{t-1})|(x_t|x_{t-1})) = N(z_{t|t}, \Upsilon_{t|t})$$

with:

$$z_{t|t} = z_{t|t-1} + \Psi_{t|t-1}\Omega_{t|t-1}^{-1}(x_t - x_{t|t-1}) \qquad \Upsilon_{t|t} = \Upsilon_{t|t-1} - \Psi_{t|t-1}\Omega_{t|t-1}^{-1}\Psi_{t|t-1}'$$

Finally, simplify the terms:

$$z_{t|t} = z_{t|t-1} + \Psi_{t|t-1}\Omega_{t|t-1}^{-1}(x_t - x_{t|t-1}) = z_{t|t-1} + \Upsilon_{t|t-1}A_t'\Omega_{t|t-1}^{-1}(x_t - x_{t|t-1}) = z_{t|t-1} + \Phi_t(x_t - x_{t|t-1})$$

with:

$$\Phi_t = \Upsilon_{t|t-1}A_t'\Omega_{t|t-1}^{-1}$$

Similarly:

$$\begin{aligned}
\Upsilon_{t|t} &= \Upsilon_{t|t-1} - \Psi_{t|t-1}\Omega_{t|t-1}^{-1}\Psi_{t|t-1}' = \Upsilon_{t|t-1} - \Upsilon_{t|t-1}A_t'\Omega_{t|t-1}^{-1}A_t\Upsilon_{t|t-1}' = \Upsilon_{t|t-1} - \Phi_t A_t \Upsilon_{t|t-1}' \\
&= \Upsilon_{t|t-1} - \Phi_t\Omega_{t|t-1}\Omega_{t|t-1}^{-1}A_t\Upsilon_{t|t-1}' = \Upsilon_{t|t-1} - \Phi_t\Omega_{t|t-1}\Phi_t'
\end{aligned}$$

Appendix: details of the derivations of the conditional distribution for the dynamic parameter

The joint distribution is:

$$\pi(z|y) = \pi(z_1, \cdots, z_T | y_1, \cdots, y_T)$$

Obtain first a conditional formulation for period $t = 1$. Note that the joint density $\pi(z|y)$ can rewrite as:

$$\begin{aligned}
& \pi(z_1, \cdots, z_T | y_1, \cdots, y_T) \\
= & \frac{\pi(z_1, \cdots, z_T, y_1, \cdots, y_T)}{\pi(y_1, \cdots, y_T)} \\
= & \frac{\pi(z_1, z_2, \cdots, z_T, y_1, \cdots, y_T)}{\pi(z_2, \cdots, z_T, y_1, \cdots, y_T)} \frac{\pi(z_2, \cdots, z_T, y_1, \cdots, y_T)}{\pi(y_1, \cdots, y_T)} \\
= & \pi(z_1 | z_2, \cdots, z_T, y_1, \cdots, y_T) \, \pi(z_2, \cdots, z_T | y_1, \cdots, y_T)
\end{aligned}$$

Using the Markovian property that implies that $z_3, \cdots, z_T$ and $y_2, \cdots, y_T$ become irrelevant to determine z_1 once z_2 and y_1 are known, this rewrites:

$$\pi(z|y) = \pi(z_1 | z_2, y_1) \, \pi(z_2, \cdots, z_T | y_2, \cdots, y_T)$$

Proceeding similarly with period $t = 2$, one obtains:

$$\pi(z_2, \cdots, z_T | y_2, \cdots, y_T) = \pi(z_2 | z_3, y_2) \, \pi(z_3, \cdots, z_T | y_3, \cdots, y_T)$$

And thus the joint density $\pi(z|y)$ rewrites:

$$\pi(z|y) = \pi(z_1 | z_2, y_1) \, \pi(z_2 | z_3, y_2) \, \pi(z_3, \cdots, z_T | y_3, \cdots, y_T)$$

Continuing recursively for $t = 3, \cdots, T$, one eventually obtains:

$$\pi(z|y) = \pi(z_1 | z_2, y_1) \, \pi(z_2 | z_3, y_2) \, \cdots \, \pi(z_{T-1} | z_T, y_{T-1}) \, \pi(z_T | y_T)$$

Or:

$$\pi(z|y) = \pi(z_T | y_T) \prod_{t=1}^{T-1} \pi(z_t | z_{t+1}, y_t)$$

Appendix: details of the derivations of the additional steps for the Carter-Kohn algorithm

The proof follows the same line as steps 5 and 6 for the Kalman filter. First, we need to show that $\pi(z_t|z_{t+1},x_t) = \pi\left((z_t|x_t)|(z_{t+1}|x_t)\right)$. This is established from:

$$\begin{aligned}
&\pi(z_t|z_{t+1},x_t) \\
=\ & \frac{\pi(z_t,z_{t+1},x_t)}{\pi(z_{t+1},x_t)} = \frac{\pi(z_t,z_{t+1},x_t)}{\pi(x_t)}\frac{\pi(x_t)}{\pi(z_{t+1},x_t)} = \frac{\pi(z_t,z_{t+1}|x_t)}{\pi(z_{t+1}|x_t)} = \frac{\pi(z_t|x_t,z_{t+1}|x_t)}{\pi(z_{t+1}|x_t)} \\
=\ & \pi\left((z_t|x_t)|(z_{t+1}|x_t)\right)
\end{aligned}$$

The next step consists in obtaining the joint distribution of $z_t|x_t$ and $z_{t+1}|x_t$. By definition, we have $z_t|x_t \sim N(z_{t|t},\Upsilon_{t|t})$ and $z_{t+1}|x_t \sim N(z_{t+1|t},\Upsilon_{t+1|t})$. As the two variables are Gaussian, their joint distribution is Gaussian as well. And since we know their mean and variances, it only remains to determine their covariance to complete the joint distribution. To do so, note that from step 2 of the Kalman filter:

$$\begin{aligned}
&\Psi_{t|t+1} \\
=\ & \mathbb{E}\left[(z_t - z_{t|t})(z_{t+1} - z_{t+1|t})'\right] \\
=\ & \mathbb{E}\left[(z_t - z_{t|t})((z_t - z_{t|t})'B_{t+1}' + \xi_{t+1}')\right] \\
=\ & \mathbb{E}\left[(z_t - z_{t|t})(z_t - z_{t|t})'\right]B_{t+1}' + \mathbb{E}\left[(z_t - z_{t|t})\xi_{t+1}'\right] \\
=\ & \Upsilon_{t|t}B_{t+1}'
\end{aligned}$$

Therefore, the full joint distribution is given by:

$$\begin{pmatrix} z_t|x_t \\ z_{t+1}|x_t \end{pmatrix} \sim N\left(\begin{bmatrix} z_{t|t} \\ z_{t+1|t} \end{bmatrix}, \begin{bmatrix} \Upsilon_{t|t} & \Psi_{t|t+1} \\ \Psi_{t|t+1}' & \Upsilon_{t+1|t} \end{bmatrix}\right)$$

We now use property d.6 to obtain:

$$\pi(z_t|z_{t+1},x_t) = \pi\left((z_t|x_t)|(z_{t+1}|x_t)\right) = N(\bar{z}_{t|t+1},\bar{\Upsilon}_{t|t+1})$$

with:

$$\bar{z}_{t|t+1} = z_{t|t} + \Psi_{t|t+1}\Upsilon_{t+1|t}^{-1}(z_{t+1} - z_{t+1|t}) \qquad \bar{\Upsilon}_{t|t+1} = \Upsilon_{t|t} - \Psi_{t|t+1}\Upsilon_{t+1|t}^{-1}\Psi_{t|t+1}'$$

Finally, simplify the terms:

$$\bar{z}_{t|t+1} = z_{t|t} + \Psi_{t|t+1}\Upsilon_{t+1|t}^{-1}(z_{t+1} - z_{t+1|t}) = z_{t|t} + \Upsilon_{t|t}B_{t+1}'\Upsilon_{t+1|t}^{-1}(z_{t+1} - z_{t+1|t}) = z_{t|t-1} + \Xi_t(z_{t+1} - z_{t+1|t})$$

with:

$$\Xi_t = \Upsilon_{t|t}B_{t+1}'\Upsilon_{t+1|t}^{-1}$$

Similarly:

$$\bar{\Upsilon}_{t|t+1} = \Upsilon_{t|t} - \Psi_{t|t+1}\Upsilon_{t+1|t}^{-1}\Psi_{t|t+1}' = \Upsilon_{t|t} - \Upsilon_{t|t}B_{t+1}'\Upsilon_{t+1|t}^{-1}B_{t+1}\Upsilon_{t|t}' = \Upsilon_{t|t} - \Xi_t B_{t+1}\Upsilon_{t|t}$$

Bibliography

Bartlett, M. S. (1934). On the theory of statistical regression. *Proceedings of the Royal Society of Edinburgh*, pages 260–283.

Blanchard, O., Cerutti, E., and Summers, L. (2015). Inflation and activity – two explorations and their monetary policy implications. Working Papers 2015/230, International Monetary Fund.

Box, G. E. P. and Muller, M. E. (1958). A note on the generation of random normal deviates. *The annals of mathematical statistics*, 29(2).

Box, G. E. P. and Tiao, G. C. (1973). *Bayesian Inference in Statistical Analysis*. Addision-Wesley.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Carter, C. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553.

Chan, J. (2013). Moving average stochastic volatility models with application to inflation forecast. *Journal of Econometrics*, 176(2):162–172.

Chan, J. and Eisenstat, E. (2018). Bayesian model comparison for time-varying parameter VARs with stochastic volatility. *Journal of Applied Econometrics*, 33(4):509–532.

Chan, J. and Jeliazkov, I. (2009). Efficient simulation and integrated likelihood estimation in state space models. *International Journal of Mathematical Modelling and Numerical Optimisation*, 1:101–120.

Dhrymes, P. J. (2013). *Mathematics for Econometrics*. Springer, New York, 4^{th} edition.

Dickey, J. M. (1967). Matricvariate generalizations of the multivariate t distribution and the inverted multivariate t distribution. *Annals of Mathematical Statistics*, 8:511–518.

Forbes, C., Evans, M., Hastings, N., and Peacock, B. (2011). *Statitical distributions*. John Wiley & Sons, Inc., 4^{th} edition.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3):515–534.

Greenberg, E. (2008). *Introduction to Bayesian Econometrics*. Cambridge University Press, 2 edition.

Gupta, A. K. and Nagar, D. K. (2000). *Matrix variate distributions*. Monographs and surveys in pure and applied mathematics. Chapman & Hall/CRC.

Kemp, C. D. and Kemp, A. W. (1991). Poisson random variate generation. *Journal of the Royal Statistical Society*, 40(1):143–158.

Marsaglia, G. and Bray, T. A. (1964). A convenient method for generating normal variables. *SIAM Review*, 6(3):260–264.

Marsaglia, G. and Tsang, W. W. (2000a). A simple method for generating gamma variables. *ACM Transactions on Mathematical Software*, 26(3).

Marsaglia, G. and Tsang, W. W. (2000b). The ziggurat method for generating random variables. *Journal of Statistical Software*, 5(8).

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30.

Robert, C. P. (1995). Simulation of truncated normal variables. *Statistics and Computing*, (5):121–125.

Wand, M. P., Ormerod, J. T., Padoan, S. A., and Fruhwirth, R. (2011). Mean field variational bayes for elaborate distributions. *Bayesian Analysis*, 6:847–900.

Wichura, M. J. (1988). algorithm as 241: the percentage points of the normal distribution. *Journal of the Royal Statistical Society*, 37(3):477–484.

Yanyan, S. (2017). Investigating a weakly informative prior for item scale hyperparameters in hierarchical 3pno irt models. *Frontiers in Psychology*, 8.

Subject index