Chapter 0

関数

The Function (and other mathematical and computational preliminaries)

予備知識

> Later generations will regard
> Mengenlehre [set theory] as a disease
> from which one has recovered.
>
> attributed to Poincáre

The basic mathematical concepts that inform our study of vectors and matrices are sets, sequences (lists), functions, and probability theory.

This chapter also includes an introduction to Python, the programming language we use to (i) model the mathematical objects of interest, (ii) write computational procedures, and (iii) carry out data analyses.

斜体：専門用語

専門用語　記号　押し付ける　列挙する　数え上げる

波カッコ

0.1 Set terminology and notation

集合

要素.

The reader is likely to be familiar with the idea of a *set*, a collection of mathematical objects in which each object is considered to occur at most once. The objects belonging to a set are its *elements*. We use curly braces to indicate a set specified by explicitly enumerating its elements. For example, $\{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$ is the set of suits in a traditional deck of cards. The order in which elements are listed is not significant; a set imposes no order among its elements.

The symbol $\in$ is used to indicate that an object belongs to a set (equivalently, that the set *contains* the object). For example, $\heartsuit \in \{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$.

One set S_1 is *contained in* another set S_2 (written $S_1 \subseteq S_2$) if every element of S_1 belongs to S_2. Two sets are equal if they contain exactly the same elements. A convenient way to prove that two sets are equal consists of two steps: (1) prove the first set is contained in the second, and (2) prove the second is contained in the first.

実数

A set can be infinite. In Chapter 1, we discuss the set $\mathbb{R}$, which consists of all real numbers, and the set $\mathbb{C}$, which consists of all complex numbers.

複素数

If a set S is not infinite, we use $|S|$ to denote its *cardinality*, the number of elements it contains. For example, the set of suits has cardinality 4.

基数（濃度）

(カーテシアン) 積

0.2 Cartesian product

デカルト積

One from column A, one from column B.

The *Cartesian product* of two sets A and B is the set of all pairs (a, b) where $a \in A$ and $b \in B$.

Example 0.2.1: For the sets $A = \{1, 2, 3\}$ and $B = \{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$, the Cartesian product is

$$\{(1,\heartsuit),(2,\heartsuit),(3,\heartsuit),(1,\spadesuit),(2,\spadesuit),(3,\spadesuit),(1,\clubsuit),(2,\clubsuit),(3,\clubsuit),(1,\diamondsuit),(2,\diamondsuit),(3,\diamondsuit)\}$$

Quiz 0.2.2: What is the cardinality of $A \times B$ in Example 0.2.1 (Page 2)?

> **Answer**
>
> $|A \times B| = 12.$

Proposition 0.2.3: For finite sets A and B, $|A \times B| = |A| \cdot |B|$.

Quiz 0.2.4: What is the cardinality of $\{1, 2, 3, \ldots, 10, J, Q, K\} \times \{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$?

13 4

$|13| \times |4| = |13 \times 4| = 52$

> **Answer**
>
> We use Proposition 0.2.3. The cardinality of the first set is 13, and the cardinality of the second set is 4, so the cardinality of the Cartesian product is $13 \cdot 4$, which is 52.

The Cartesian product is named for René Descartes, whom we shall discuss in Chapter 6.

0.3 The function

$\heartsuit \in \{ \quad \}$
$D = \{ \quad \}$

Mathematicians never die—they just lose function.

像 原像

割り当て

Loosely speaking, a function is a rule that, for each element in some set D of possible inputs, assigns a possible output. The output is said to be the *image* of the input under the function and the input is a *pre-image* of the output. The set D of possible inputs is called the *domain* of the function. (定義域)

Formally, a *function* is a (possibly infinite) set of pairs (a, b) no two of which share the same first entry.

(a_1, b_1)
(a_2, b_2)

×2

Example 0.3.1: The doubling function with domain $\{1, 2, 3, \ldots\}$ is

$$\{(1, 2), (2, 4), (3, 6), (4, 8), \ldots\}$$

input output

The domain can itself consist of pairs of numbers.

Example 0.3.2: The multiplication function with domain $\{1, 2, 3, \ldots\} \times \{1, 2, 3, \ldots\}$ looks something like this:

$$\{((1, 1), 1), ((1, 2), 2), \ldots, ((2, 1), 2), ((2, 2), 4), ((2, 3), 6), \ldots\}$$

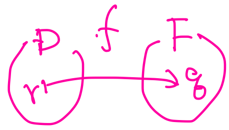
$q \xrightarrow{f} r$
$r = f(q)$

For a function named f, the image of q under f is denoted by $f(q)$. If $r = f(q)$, we say that *q maps to r under f*. The notation for "q maps to r" is $q \mapsto r$. (This notation omits specifying the function; it is useful when there is no ambiguity about which function is intended.)

It is convenient when specifying a function to specify a *co-domain* for the function. The co-domain is a set from which the function's output values are chosen. Note that one has some leeway in choosing the co-domain since not all of its members need be outputs.

終域
余域

domain co-domain



The notation

$$f : D \longrightarrow F$$

means that f is a function whose domain is the set D and whose *co-domain* (the set of possible outputs) is the set F. (More briefly: "a function from D to F", or "a function that maps D to F.")

シーザー 暗号システム

Example 0.3.3: Caesar was said to have used a cryptosystem in which each letter was replaced with the one three steps forward in the alphabet (wrapping around for X,Y, and Z).[a] Thus the plaintext MATRIX would be encrypted as the cyphertext PDWULA. The function that maps each plaintext letter to its cyphertext replacement could be written as

暗号文

$$A \mapsto D, B \mapsto E, C \mapsto F, D \mapsto G, W \mapsto Z, X \mapsto A, Y \mapsto B, Z \mapsto C$$

This function's domain and co-domain are both the alphabet $\{A, B, \ldots, Z\}$.

[a]Some imaginary historians have conjectured that Caesar's assasination can be attributed to his use of such a weak cryptosystem.

Example 0.3.4: The cosine function, *cos*, maps from the set of real numbers (indicated by $\mathbb{R}$) to the set of real numbers. We would therefore write

image ~~値域~~ range

$$\cos : \mathbb{R} \longrightarrow \mathbb{R}$$

Of course, the outputs of the cos function do not include all real numbers, only those between -1 and 1.

The *image* of a function f is the set of images of all domain elements. That is, the image of f is the set of elements of the co-domain that actually occur as outputs. For example, the image of Caesar's encryption function is the entire alphabet, and the image of the cosine function is the set of numbers between -1 and 1.

product (積)

Example 0.3.5: Consider the function *prod* that takes as input a pair of integers greater than 1 and outputs their product. The domain (set of inputs) is the set of pairs of integers greater than 1. We choose to define the co-domain to be the set of all integers greater than 1. The image of the function, however, is the set of *composite* integers since no domain element maps to a prime number.

0.3.1 Functions versus procedures, versus computational problems

There are two other concepts that are closely related to functions and that enter into our story, and we must take some care to distinguish them.

- A *procedure* is a precise description of a computation; it accepts *inputs* (called *arguments*) and produces an output (called the *return value*).

  Example 0.3.6: This example illustrates the Python syntax for defining procedures:

  ```
  def mul(p,q): return p*q
  ```

  In the hope of avoiding confusion, we diverge from the common practice of referring to procedures as "functions".

- A *computational problem* is an input-output specification that a procedure might be required to satisfy.