# Functions

Last modified: 2026-11-12 14:55:20 (PST)

*Remark.* Random variables, densities, and measures are functions. This page collects the definitions of functions, sequences, and countable sets that the Morrison Lab's probability and statistics notes build on. The functions here go between the sets defined on the Sets page.

**Definition 1** (Function, map, mapping, transformation)**.** A **function** $f$ from a set $A$ to a set $B$, (Wikipedia contributors 2026j) written $f : A \to B$, assigns to each element $a \in A$ exactly one element $f(a) \in B$.
A function is also called a **map** (or **mapping**) or a **transformation**; on this site the three words mean the same thing, although some authors reserve "transformation" for a function from a set to itself.

**Example 1** (Doubling a die roll)**.** The rule $f(a) = 2a$ defines a function $f : \{1, 2, 3, 4, 5, 6\} \to \mathbb{R}$, because it assigns exactly one real number to each possible roll; for example, $f(3) = 6$. A rule that assigned both 2 and $-2$ to the roll 1 would not be a function, because a function assigns exactly one value to each element.

**Definition 2** (Argument, input)**.** An **argument** (or **input**) of a function $f : A \to B$ is an element $a \in A$ that $f$ is applied to.

**Example 2** (Arguments of doubling a die roll)**.** For the function $f(a) = 2a$ in Example 1, the roll 3 is an argument of $f$, because $f$ can be applied to it. The number 7 is not an argument of $f$, because it is not an element of $\{1, 2, 3, 4, 5, 6\}$.

**Definition 3** (Value, output)**.** The **value** (or **output**) of a function $f : A \to B$ at an argument $a \in A$ is the element $f(a) \in B$ that $f$ assigns to $a$.

**Example 3** (Values of doubling a die roll)**.** For the function $f(a) = 2a$ in Example 1, the value of $f$ at the argument 3 is $f(3) = 6$, and the value at the argument 5 is $f(5) = 10$.

**Definition 4** (Relation)**.** A **relation** $\sim$ on a set $S$ (Wikipedia contributors 2026a) is a rule that, for each ordered pair of elements $a, b \in S$, says whether $a \sim b$ holds ("$a$ is related to $b$").

**Example 4** (Relations on the real numbers)**.** $=$, $<$ and $\leq$ are relations on $\mathbb{R}$. For $<$, the pair $(2, 5)$ is related, since $2 < 5$, but the pair $(5, 2)$ is not, since $5 < 2$ is false: the order of the pair matters.

**Definition 5** (Domain)**.** The **domain** (Wikipedia contributors 2026g) of a function $f : A \to B$ is the set $A$ of elements that $f$ assigns values to.

**Definition 6** (Codomain)**.** The **codomain** (Wikipedia contributors 2026e) of a function $f : A \to B$ is the set $B$ that its values are required to lie in.

**Definition 7** (Image)**.** The **image** (Wikipedia contributors 2026m) of a function $f : A \to B$ is the set of values that $f$ actually takes:

$$f(A) \stackrel{\text{def}}{=} \{f(a) : a \in A\}$$

For a single element $a \in A$, the value $f(a)$ is also called the **image of** $a$ under $f$.

**Definition 8** (Preimage, inverse image)**.** Let $f : A \to B$ be a function, and let $S$ be a subset of $B$. The **preimage** (Wikipedia contributors 2026m) (also called the **inverse image**) of $S$ under $f$ is the set of elements of $A$ whose values are in $S$:

$$f^{-1}(S) \stackrel{\text{def}}{=} \{a \in A : f(a) \in S\}$$

**Example 5** (The preimage of a set of doubled die rolls)**.** For the function $f : \{1, 2, 3, 4, 5, 6\} \to \mathbb{R}$ with $f(a) = 2a$ from Example 1, the preimage of $\{4, 6\}$ is $\{2, 3\}$, because $f(2) = 4$ and $f(3) = 6$. The preimage of $\{3\}$ is the empty set, because $2a = 3$ has no solution among the die rolls.

**Definition 9** (Range)**.** The word **range** of a function $f : A \to B$ has two conflicting meanings, and sources use one or the other:

- the image of $f$, the set $f(A)$ of values that $f$ actually takes;
- the codomain of $f$, the set $B$ that its values are required to lie in.

To avoid the ambiguity, these notes write "image" or "codomain" instead, and recommend that you do too (Wikipedia contributors 2026r).

**Example 6** (Two ranges of a squaring function)**.** For $f : \mathbb{R} \to \mathbb{R}$ with $f(a) = a^2$, "the range of $f$" could mean the image $[0, \infty)$ or the codomain $\mathbb{R}$. The image is always a subset of the codomain, because every value $f(a)$ lies in the codomain.

*Remark* 1 (Image and range). Many sources call the image the "range", but others use "range" for the codomain. Definition 9 gives the two meanings and the terms these notes use instead.

**Example 7** (Domain, codomain, and image of the doubled die roll)**.** For the function $f : \{1, 2, 3, 4, 5, 6\} \to \mathbb{R}$ with $f(a) = 2a$ from Example 1:

- the domain is $\{1, 2, 3, 4, 5, 6\}$;
- the codomain is $\mathbb{R}$;
- the image is $\{2, 4, 6, 8, 10, 12\}$.

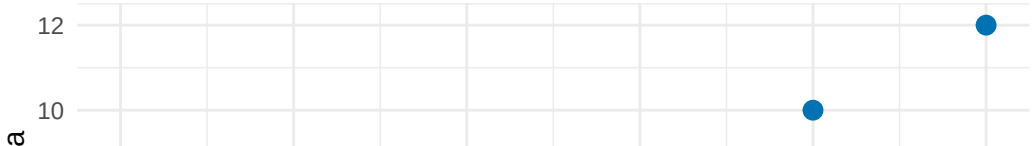
The image is a subset of the codomain, and here a much smaller one: most real numbers, such as 3 and $-1$, are not values of $f$.

```r
rolls <- data.frame(roll = 1:6, double = 2 * (1:6))
levels_shown <- data.frame(
  height = c(3, 4, 6),
  hits = c(
    "preimage is empty",
    "preimage contains a roll",
    "preimage contains a roll"
  )
)

preimage_plot <- ggplot2::ggplot() +
  ggplot2::geom_hline(
    data = levels_shown,
    ggplot2::aes(yintercept = height, linetype = hits),
    colour = "#D55E00"
  ) +
  ggplot2::geom_point(
    data = rolls,
    ggplot2::aes(x = roll, y = double),
    size = 3,
    colour = "#0072B2"
  ) +
  ggplot2::scale_x_continuous(breaks = 1:6) +
  ggplot2::scale_y_continuous(breaks = c(2, 3, 4, 6, 8, 10, 12)) +
  ggplot2::scale_linetype_manual(
    values = c(
      "preimage contains a roll" = "solid",
      "preimage is empty" = "dashed"
    )
  ) +
  ggplot2::labs(
    x = "Input, a (the roll)",
    y = "Output, f(a) = 2a",
    linetype = NULL
  ) +
  ggplot2::theme_minimal() +
  ggplot2::theme(legend.position = "bottom")

if (knitr::is_html_output()) {
  htmltools::div(
    role = "img",
    `aria-label` = fig_alt,
    plotly::ggplotly(preimage_plot) |>
      plotly::layout(legend = list(orientation = "h", y = -0.2))
  )
} else {
  preimage_plot
}
```





**Definition 10** (Graph of a function)**.** The **graph** (Wikipedia contributors 2026l) of a function $f : A \to B$ is the set of ordered pairs $(a, f(a))$, one for each element $a$ of its domain:

$$\{(a, f(a)) : a \in A\}$$

**Example 8** (The graph of a function on three points)**.** Let $f : \{-1, 0, 2\} \to \mathbb{R}$ with $f(a) = a^2$. Then $f(-1) = 1$, $f(0) = 0$, and $f(2) = 4$, so the graph of $f$ is $\{(-1, 1), (0, 0), (2, 4)\}$. The pair $(1, 1)$ is not in the graph, because 1 is not in the domain of $f$.

*Remark* 2 (The graph of a real function is a curve)*.* For a function $f : \mathbb{R} \to \mathbb{R}$, the graph is a curve in the plane, with $a$ on the horizontal axis and $f(a)$ on the vertical axis.

*Remark* 3 (What a function is fundamentally)*.* Definition 1 defines what a function *does*: it assigns an output to each input. In formal set theory, what a function *is* is described in terms of ordered pairs and its graph (Definition 10): the set of pairs $G = \{(a, f(a)) : a \in A\} \subseteq A \times B$. Authors define a function formally in one of two ways:

1. **As its graph** (Halmos 1974, sec. 8, p. 30): a function is a subset of $A \times B$ containing exactly one pair $(a, b)$ for each $a \in A$, where $b = f(a)$, and the statement $f(a) = b$ is notation for $(a, b) \in f$. Under this convention, the codomain $B$ is specified alongside the graph in the declaration $f : A \to B$.
2. **As a triple** $(A, B, G)$: the function consists of the domain $A$, the codomain $B$, and the graph $G \subseteq A \times B$. This explicit packaging retains the codomain $B$ as part of the function itself, distinguishing functions that have the same graph but different codomains (such as a function mapping into $\mathbb{R}$ versus into integers).

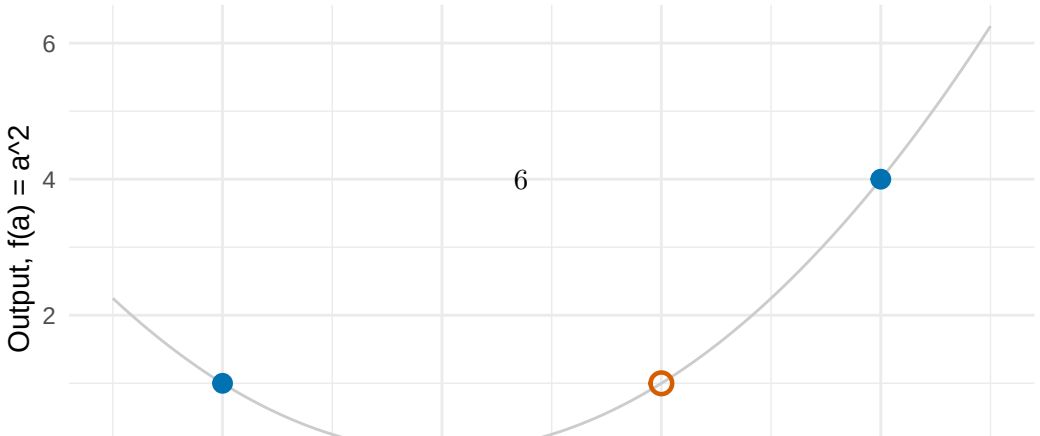
For example, the die-doubling function in Example 1 has the graph of six ordered pairs $\{(1, 2), (2, 4), (3, 6), (4, 8), (5, 10), (6, 12)\}$.
This distinction clarifies what a function is and is not:

- **Not an algebraic formula:** A function is not fundamentally an equation or algebraic expression. While many functions are specified by a formula (such as $f(x) = x^2$), many others cannot be written in closed form (such as empirical measurements or a lookup table), yet every function has a well-defined set of input-output pairs.
- **Not a curve:** A curve or plot is a geometric visualization of a function whose domain and codomain are subsets of $\mathbb{R}$, but functions can map between arbitrary sets (such as sets of words, matrices, or probability distributions).

```r
graph_points <- data.frame(a = c(-1, 0, 2), value = c(-1, 0, 2)^2)
parabola <- data.frame(a = seq(-1.5, 2.5, length.out = 201))
parabola$value <- parabola$a^2

graph_three_points_plot <- ggplot2::ggplot() +
  ggplot2::geom_line(
    data = parabola,
    ggplot2::aes(x = a, y = value),
    colour = "grey80"
  ) +
  ggplot2::geom_point(
    data = graph_points,
    ggplot2::aes(x = a, y = value, colour = "in the graph"),
    size = 3
  ) +
  ggplot2::geom_point(
    data = data.frame(a = 1, value = 1),
    ggplot2::aes(x = a, y = value, colour = "not in the graph"),
    shape = 1,
    size = 3,
    stroke = 1.2
  ) +
  ggplot2::scale_colour_manual(
    values = c("in the graph" = "#0072B2", "not in the graph" = "#D55E00")
  ) +
  ggplot2::scale_x_continuous(breaks = -1:2) +
  ggplot2::labs(x = "Input, a", y = "Output, f(a) = a^2", colour = NULL) +
  ggplot2::theme_minimal() +
  ggplot2::theme(legend.position = "bottom")

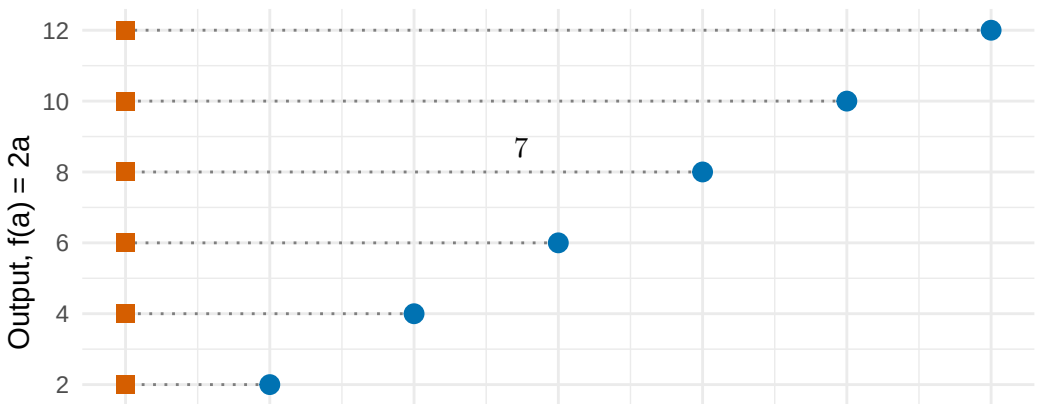
if (knitr::is_html_output()) {
  htmltools::div(
    role = "img",
    `aria-label` = fig_alt,
    plotly::ggplotly(graph_three_points_plot) |>
      plotly::layout(legend = list(orientation = "h", y = -0.2))
  )
} else {
  graph_three_points_plot
}
```



```r
rolls <- data.frame(roll = 1:6, double = 2 * (1:6))

doubling_die_plot <- ggplot2::ggplot(rolls) +
  ggplot2::aes(x = roll, y = double) +
  ggplot2::geom_segment(
    ggplot2::aes(xend = 0, yend = double),
    linetype = "dotted",
    colour = "grey50"
  ) +
  ggplot2::geom_point(
    ggplot2::aes(colour = "graph of f"),
    size = 3
  ) +
  ggplot2::geom_point(
    ggplot2::aes(x = 0, colour = "image of f"),
    shape = 15,
    size = 3
  ) +
  ggplot2::scale_x_continuous(breaks = 0:6, limits = c(0, 6)) +
  ggplot2::scale_y_continuous(breaks = seq(0, 12, 2), limits = c(0, 12)) +
  ggplot2::scale_colour_manual(
    values = c("graph of f" = "#0072B2", "image of f" = "#D55E00")
  ) +
  ggplot2::labs(
    x = "Input, a (the roll)",
    y = "Output, f(a) = 2a",
    colour = NULL
  ) +
  ggplot2::theme_minimal() +
  ggplot2::theme(legend.position = "bottom")

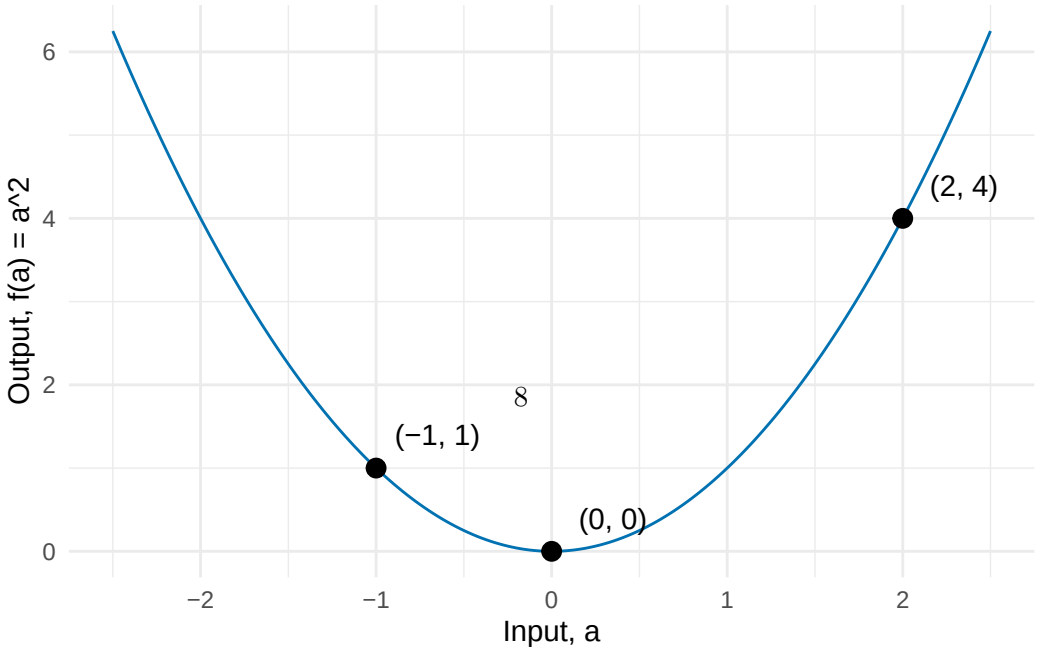
if (knitr::is_html_output()) {
  htmltools::div(
    role = "img",
    `aria-label` = fig_alt,
    plotly::ggplotly(doubling_die_plot) |>
      plotly::layout(legend = list(orientation = "h", y = -0.2))
  )
} else {
  doubling_die_plot
}
```



```r
curve_data <- data.frame(a = seq(-2.5, 2.5, length.out = 201))
curve_data$value <- curve_data$a^2
marked <- data.frame(a = c(-1, 0, 2), value = c(-1, 0, 2)^2)
marked$label <- sprintf("(%d, %d)", marked$a, marked$value)

graph_real_function_plot <- ggplot2::ggplot() +
  ggplot2::geom_line(
    data = curve_data,
    ggplot2::aes(x = a, y = value),
    colour = "#0072B2"
  ) +
  ggplot2::geom_point(
    data = marked,
    ggplot2::aes(x = a, y = value),
    size = 3
  ) +
  ggplot2::geom_text(
    data = marked,
    ggplot2::aes(x = a, y = value, label = label),
    nudge_x = 0.35,
    nudge_y = 0.4
  ) +
  ggplot2::labs(x = "Input, a", y = "Output, f(a) = a^2") +
  ggplot2::theme_minimal()

if (knitr::is_html_output()) {
  htmltools::div(
    role = "img",
    `aria-label` = fig_alt,
    plotly::ggplotly(graph_real_function_plot)
  )
} else {
  graph_real_function_plot
}
```



- **Sets of pairs with unique first entries:** Any subset of $A \times B$ in which no two distinct pairs share the same first element defines a function from the set of those first elements to $B$.

In computer science, viewing a function as a set of key-value pairs corresponds directly to a **dictionary**, **lookup table**, or **hash map**. Evaluating an expression computes outputs on demand; caching computed outputs $(a, f(a))$ on demand to return on future calls without recomputing is known as **memoization**.

**Definition 11** (Composition)**.** Let $g : A \to B$ and $f : B \to C$ be functions. The **composition** (Wikipedia contributors 2026i) of $f$ and $g$ is the function $f \circ g : A \to C$ defined by

$$(f \circ g)(a) \stackrel{\text{def}}{=} f(g(a)) \quad \text{for each } a \in A.$$

**Definition 12** (Inner function)**.** In a composition $f \circ g$ (Definition 11), $g$ is the **inner function**, applied first.

**Definition 13** (Outer function)**.** In a composition $f \circ g$ (Definition 11), $f$ is the **outer function**, applied to the result.

**Example 9** (The order of composition matters)**.** Let $g : \mathbb{R} \to \mathbb{R}$ with $g(x) = x + 1$, and $f : \mathbb{R} \to \mathbb{R}$ with $f(x) = x^2$.

- $$\begin{aligned}(f \circ g)(2) &= f(g(2)) \\ &= f(3) \\ &= 9 :\end{aligned}$$

  $g$ is the inner function, and $f$ is the outer function.

- $$\begin{aligned}(g \circ f)(2) &= g(f(2)) \\ &= g(4) \\ &= 5 :\end{aligned}$$

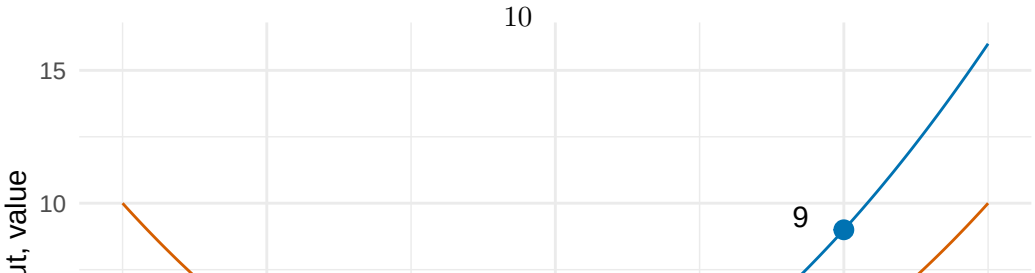
  now $f$ is the inner function, and $g$ is the outer function.

Since $9 \neq 5$, $f \circ g$ and $g \circ f$ are different functions.

```r
x_grid <- seq(-3, 3, length.out = 241)
fg_label <- "f(g(x)) = (x + 1)^2"
gf_label <- "g(f(x)) = x^2 + 1"
compositions <- data.frame(
  x = rep(x_grid, 2),
  composition = rep(c(fg_label, gf_label), each = length(x_grid)),
  value = c((x_grid + 1)^2, x_grid^2 + 1)
)
at_two <- data.frame(
  x = 2,
  composition = c(fg_label, gf_label),
  value = c(9, 5)
)

composition_plot <- ggplot2::ggplot() +
  ggplot2::geom_line(
    data = compositions,
    ggplot2::aes(x = x, y = value, colour = composition)
  ) +
  ggplot2::geom_point(
    data = at_two,
    ggplot2::aes(x = x, y = value, colour = composition),
    size = 3
  ) +
  ggplot2::geom_text(
    data = at_two,
    ggplot2::aes(x = x, y = value, label = value),
    nudge_x = -0.3,
    nudge_y = 0.5
  ) +
  ggplot2::scale_colour_manual(values = c("#0072B2", "#D55E00")) +
  ggplot2::labs(x = "Input, x", y = "Output, value", colour = NULL) +
  ggplot2::theme_minimal() +
  ggplot2::theme(legend.position = "bottom")

if (knitr::is_html_output()) {
  htmltools::div(
    role = "img",
    `aria-label` = fig_alt,
    plotly::ggplotly(composition_plot) |>
      plotly::layout(legend = list(orientation = "h", y = -0.2))
  )
} else {
  composition_plot
}
```



**Definition 14** (Inverse function)**.** Let $f : A \to B$ be a function. A function $g : B \to A$ is the **inverse function** (Wikipedia contributors 2026o) of $f$, written $f^{-1}$, if both

- $g(f(a)) = a$ for every $a \in A$, and
- $f(g(b)) = b$ for every $b \in B$.

A function that has an inverse function is **invertible**.

*Remark* 4 ($f^{-1}$ is not $1/f$)*.* The $-1$ in $f^{-1}$ does not mean a reciprocal: $f^{-1}(b)$ is the input that $f$ maps to $b$, not $\frac{1}{f(b)}$. For example, with $f(x) = 2x$ on $\mathbb{R}$, $f^{-1}(4) = 2$, but $\frac{1}{f(4)} = \frac{1}{8}$.

**Example 10** (An invertible function, and one that is not)**.**

- Let $f : \mathbb{R} \to \mathbb{R}$ with $f(x) = 2x + 1$. Its inverse function is $f^{-1}(y) = \frac{y-1}{2}$. For example, $f(3) = 7$ and

$$\begin{aligned} f^{-1}(7) &= \frac{7-1}{2} \\ &= 3. \end{aligned}$$

- Let $h : \mathbb{R} \to \mathbb{R}$ with $h(x) = x^2$. Then $h(-2) = 4$ and $h(2) = 4$, so an inverse function would need $h^{-1}(4)$ to equal both $-2$ and $2$. So $h$ is not invertible.

**Definition 15** (Elementwise application of a function to a vector)**.** Let $f : A \to B$ be a function (definition), and let $\underset{\sim}{x} = (x_1, x_2, \ldots, x_n)^\top$ be a vector whose entries all lie in $A$. The **elementwise application** of $f$ to $\underset{\sim}{x}$, written $f^{\odot}(\underset{\sim}{x})$, is the vector of the values of $f$ at the entries of $\underset{\sim}{x}$:

$$f^{\odot}(\underset{\sim}{x}) = (f(x_1), f(x_2), \ldots, f(x_n))^\top.$$

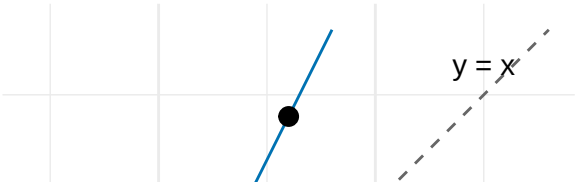
It is a function $f^{\odot} : A^n \to B^n$, where $A^n$ is the $n$-fold Cartesian product of $A$ with itself (definition).

```r
x_f <- seq(-2, 4, length.out = 101)
x_inverse <- seq(-3, 9, length.out = 101)
lines_data <- data.frame(
  x = c(x_f, x_inverse, x_inverse),
  y = c(2 * x_f + 1, (x_inverse - 1) / 2, x_inverse),
  curve = rep(
    c("f", "inverse of f", "y = x"),
    times = c(length(x_f), length(x_inverse), length(x_inverse))
  )
)
marked <- data.frame(x = c(3, 7), y = c(7, 3))
labels_data <- data.frame(
  x = c(1.2, 7, 7.5),
  y = c(5.2, 0.6, 8.2),
  label = c("f", "inverse of f", "y = x")
)

inverse_function_plot <- ggplot2::ggplot() +
  ggplot2::geom_line(
    data = lines_data,
    ggplot2::aes(x = x, y = y, colour = curve, linetype = curve)
  ) +
  ggplot2::geom_point(data = marked, ggplot2::aes(x = x, y = y), size = 3) +
  ggplot2::geom_text(
    data = labels_data,
    ggplot2::aes(x = x, y = y, label = label)
  ) +
  ggplot2::coord_fixed(xlim = c(-3, 9), ylim = c(-3, 9)) +
  ggplot2::scale_colour_manual(
    values = c("f" = "#0072B2", "inverse of f" = "#D55E00", "y = x" = "grey40")
  ) +
  ggplot2::scale_linetype_manual(
    values = c("f" = "solid", "inverse of f" = "solid", "y = x" = "dashed")
  ) +
  ggplot2::labs(x = "Input, x", y = "Output, y") +
  ggplot2::theme_minimal() +
  ggplot2::theme(legend.position = "none")

if (knitr::is_html_output()) {
  htmltools::div(
    role = "img",
    `aria-label` = fig_alt,
    plotly::ggplotly(inverse_function_plot)
  )
} else {
  inverse_function_plot
}
```





```r
x_square <- seq(-3, 3, length.out = 101)
square_data <- data.frame(x = x_square, y = x_square^2)
marked <- data.frame(x = c(-2, 2), y = c(4, 4))

not_invertible_plot <- ggplot2::ggplot() +
  ggplot2::geom_hline(yintercept = 4, colour = "grey40", linetype = "dotted") +
  ggplot2::geom_line(
    data = square_data,
    ggplot2::aes(x = x, y = y),
    colour = "#0072B2"
  ) +
  ggplot2::geom_point(data = marked, ggplot2::aes(x = x, y = y), size = 3) +
  ggplot2::annotate("text", x = -2.8, y = 4.6, label = "y = 4") +
  ggplot2::scale_x_continuous(breaks = -3:3) +
  ggplot2::labs(x = "Input, x", y = "Output, h(x) = x^2") +
  ggplot2::theme_minimal()

if (knitr::is_html_output()) {
  htmltools::div(
    role = "img",
    `aria-label` = fig_alt,
    plotly::ggplotly(not_invertible_plot)
  )
} else {
  not_invertible_plot
}
```

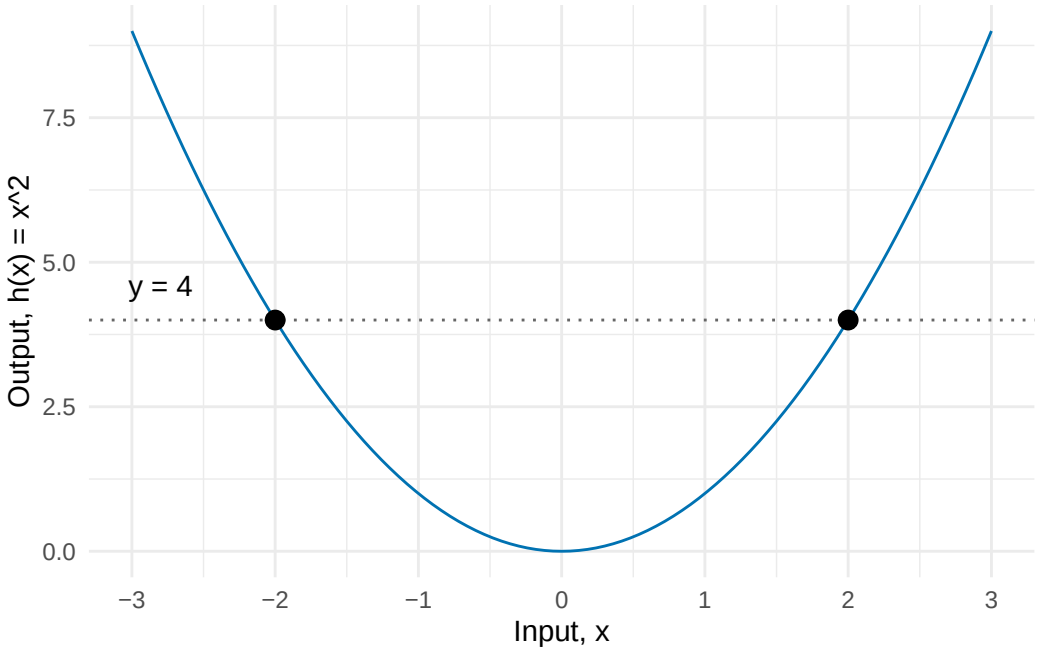



Figure 7: The horizontal line $y = 4$ meets the graph of $h(x) = x^2$ twice, at $x = -2$ and $x = 2$, so $h$ has no inverse function.

**Example 11** (The exponential and logistic functions applied to a vector)**.** Let $\underset{\sim}{x} = (-1, 0, 2)^\top$. With $\exp(t) = e^t$ and the logistic function $\text{expit}(t) = \frac{1}{1+e^{-t}}$,

$$\exp^{\odot}(\underset{\sim}{x}) = \left(e^{-1}, e^0, e^2\right)^\top,$$
$$\text{expit}^{\odot}(\underset{\sim}{x}) = \left(\text{expit}(-1), \text{expit}(0), \text{expit}(2)\right)^\top.$$

The code evaluates both, using R's `exp()` and `plogis()`, which apply to a vector entry by entry.

```
x <- c(-1, 0, 2)
exp_x <- exp(x)
expit_x <- plogis(x)
exp_x
#> [1] 0.3678794 1.0000000 7.3890561
expit_x
#> [1] 0.2689414 0.5000000 0.8807971
```

So $\exp^{\odot}(\underset{\sim}{x}) \approx (0.3679,\ 1,\ 7.3891)$ and $\text{expit}^{\odot}(\underset{\sim}{x}) \approx (0.2689,\ 0.5,\ 0.8808)$. Each entry of the output depends only on the entry of $\underset{\sim}{x}$ in the same position.

*Remark* 5 (Notation and software for elementwise application)*.*

- The superscript $\odot$ keeps $f^{\odot}$ apart from $f$ itself. Where a function $f$ takes a whole vector as its argument, as in the multivariable calculus of vector calculus, $f(\underset{\sim}{x})$ means $f$ evaluated at the vector, which is generally not the vector of its values at the entries.
- If $f$ has parameters, as in $f_\beta$, the same parameters are used at every entry.
- Software usually applies a scalar function to a vector this way, without a special symbol: R and NumPy evaluate `exp(x)` entry by entry. The matrix exponential of a square matrix is a different function from $\exp^{\odot}$.

**Definition 16** (Operator)**.** An **operator** (Wikipedia contributors 2026q) is a function whose arguments and values are both functions or both vectors.

**Example 12** (The derivative as an operator)**.** Let $D$ send each differentiable function $f : \mathbb{R} \to \mathbb{R}$ to its derivative function $Df = f'$. Then $D$ is an operator, because its arguments and its values are functions. For example, $D$ sends $f(x) = x^2$ to the function $(Df)(x) = 2x$.

*Remark* 6 (Other uses of "operator")*.* Some authors call every function an operator, and others reserve the word for functions whose domain and codomain are the same set, as in the linear operator.

**Definition 17** (Functional)**.** A **functional** (Wikipedia contributors 2026k) is a function whose arguments are functions or vectors and whose values are real numbers.

**Example 13** (Evaluation, integration and expectation as functionals)**.**

- Evaluation at 0 sends each function $f : \mathbb{R} \to \mathbb{R}$ to the number $f(0)$.
- The definite integral sends each continuous function $f : [0, 1] \to \mathbb{R}$ to the number $\int_0^1 f(x)\,dx$. For $f(x) = x^2$, the value is $\frac{1}{3}$.
- The expectation sends each random variable, which is a function, to a real number, its population mean.
- Summing the entries sends each vector $\underset{\sim}{x} \in \mathbb{R}^n$ to the number $\sum_{i=1}^n x_i$.

Each takes a function or a vector as its argument and returns a number, so each is a functional. The derivative operator in Example 12 is not a functional, because its values are functions, not numbers.

**Definition 18** (Sequence)**.** A **sequence** in a set $S$ (Wikipedia contributors 2026s) is a function $a : \mathbb{N} \to S$ from the natural numbers to $S$. Its value $a(n)$ is written $a_n$, and the sequence is written $(a_n)$ or $a_1, a_2, a_3, \ldots$.

**Definition 19** (Term of a sequence)**.** The value $a_n$ of a sequence $(a_n)$ (Definition 18) is called its $n$th **term**.

**Example 14** (Sequences)**.**

- $a_n = \frac{1}{n}$ is the sequence $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, ...$; its third term is $a_3 = \frac{1}{3}$.
- $b_n = (-1)^n$ is the sequence $-1, 1, -1, 1, ...$: a sequence can repeat values.
- The die rolls $4, 1, 4$ form a finite sequence of length 3, with $a_1 = 4$, $a_2 = 1$, and $a_3 = 4$. As a set, $\{4, 1, 4\} = \{1, 4\}$, but as a sequence, the order and the repeats matter.

**Definition 20** (Finite sequence)**.** Let $S$ be a set. A **finite sequence** of length $n$ is a function from $\{1, \dots, n\}$ to $S$, written $a_1, \dots, a_n$.

**Definition 21** (Closed under an operation)**.** A set $A$ is **closed under** (Wikipedia contributors 2026d) an operation, such as addition of numbers or union of sets, if applying the operation to elements of $A$ always gives an element of $A$. This property is called closure under the operation.

**Example 15** (Closed and not closed)**.**

- The even integers are closed under addition: for example, $2 + 4 = 6$ is even, and $2m + 2n = 2(m + n)$ is even for all integers $m$ and $n$.
- The odd integers are not closed under addition: 1 and 3 are odd, but $1 + 3 = 4$ is even.
- The collection of all subsets of a set $S$ is closed under union: the union of two subsets of $S$ is again a subset of $S$.

**Definition 22** (Logistic function, sigmoid, expit)**.** The **logistic function**, also called the **sigmoid** or **expit**, is the function $\text{expit} : \mathbb{R} \to (0, 1)$ with (Wikipedia contributors 2026p)

$$\text{expit}(\eta) \stackrel{\text{def}}{=} \frac{1}{1 + \mathrm{e}^{-\eta}}.$$

```r
n <- 1:10
sequence_labels <- c("a_n = 1/n", "b_n = (-1)^n")
sequences <- data.frame(
  n = rep(n, 2),
  sequence = factor(
    rep(sequence_labels, each = length(n)),
    levels = sequence_labels
  ),
  term = c(1 / n, (-1)^n)
)

sequences_plot <- ggplot2::ggplot(sequences) +
  ggplot2::aes(x = n, y = term) +
  ggplot2::geom_point(size = 2.5, colour = "#0072B2") +
  ggplot2::facet_wrap(~sequence, scales = "free_y") +
  ggplot2::scale_x_continuous(breaks = n) +
  ggplot2::labs(x = "Index, n", y = "Term value") +
  ggplot2::theme_minimal()

if (knitr::is_html_output()) {
  htmltools::div(
    role = "img",
    `aria-label` = fig_alt,
    plotly::ggplotly(sequences_plot)
  )
} else {
  sequences_plot
}
```

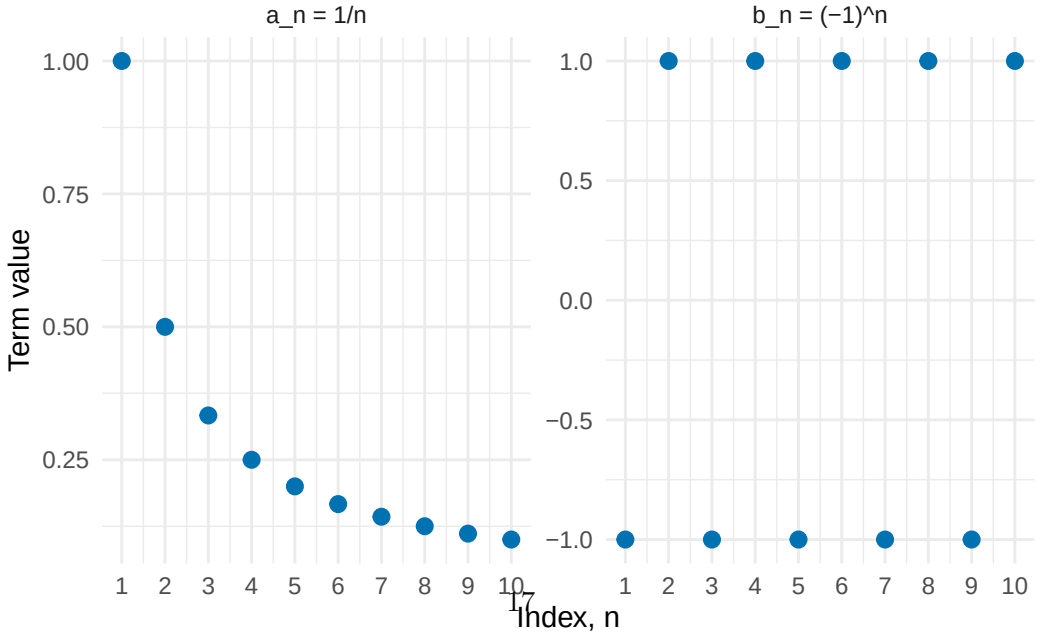



Figure 8: The first ten terms of the sequences $a_n = 1/n$ and $b_n = (-1)^n$ of Example 14, plotted against $n$. The terms of $b_n$ take only two values.

**Example 16** (Values of the logistic function)**.** The logistic function at $\eta = -2$, $\eta = 0$ or $\eta = 2$ has these values:

```
eta <- c(-2, 0, 2)
data.frame(eta = eta, sigmoid = 1 / (1 + exp(-eta)))
#>   eta   sigmoid
#> 1  -2 0.1192029
#> 2   0 0.5000000
#> 3   2 0.8807971
```

The value at $\eta = 0$ is $1/(1 + \mathrm{e}^{-0}) = 1/2$. The values at $\pm 2$ add to 1; each lies in $(0, 1)$.

## Countable sets

**Definition 23** (Finite set)**.** A set $A$ is **finite** (Wikipedia contributors 2026h) if it is empty, or if, for some natural number $n$, its elements can be listed as a finite sequence (Definition 20) $a_1, \ldots, a_n$ in which each element of $A$ appears exactly once.

**Definition 24** (Cardinality)**.** For a finite set $A$ (Definition 23), the number $n$ in such a list is the **cardinality** (Wikipedia contributors 2026c) of $A$, its number of elements, written $|A|$; the empty set has cardinality $|\emptyset| = 0$.

*Remark* 7 (Two meanings of $|\cdot|$). Applied to a set, the bars $|A|$ mean the cardinality of $A$. Applied to a number, $|x|$ means its absolute value, or its modulus when $x$ is complex. For example, $|\{-3\}| = 1$, but $|-3| = 3$.

**Example 17** (Finite and infinite sets)**.**

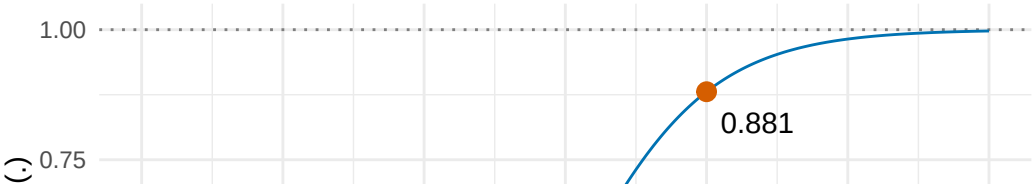
- $\{2, 4, 6\}$ is finite, with $|\{2, 4, 6\}| = 3$.
- $\{1, 1, 2\} = \{1, 2\}$, so $|\{1, 1, 2\}| = 2$: a repeated listing does not count twice.
- $\emptyset$ is finite, with $|\emptyset| = 0$.
- The natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$ are infinite: for any $n$, a list $a_1, \ldots, a_n$ of natural numbers misses the natural number $a_1 + \cdots + a_n + 1$, which is larger than every number in the list.

```r
# the Greek letter eta, written as a code point to keep this file ASCII
eta_symbol <- intToUtf8(0x3b7)
eta_grid <- seq(-6, 6, length.out = 241)
logistic_curve <- data.frame(eta = eta_grid, sigmoid = plogis(eta_grid))
logistic_points <- data.frame(eta = c(-2, 0, 2), sigmoid = plogis(c(-2, 0, 2)))
logistic_points$label <- sprintf("%.3f", logistic_points$sigmoid)

logistic_function_plot <- ggplot2::ggplot() +
  ggplot2::geom_hline(
    yintercept = c(0, 1),
    linetype = "dotted",
    colour = "grey50"
  ) +
  ggplot2::geom_line(
    data = logistic_curve,
    ggplot2::aes(x = eta, y = sigmoid),
    colour = "#0072B2"
  ) +
  ggplot2::geom_point(
    data = logistic_points,
    ggplot2::aes(x = eta, y = sigmoid),
    size = 3,
    colour = "#D55E00"
  ) +
  ggplot2::geom_text(
    data = logistic_points,
    ggplot2::aes(x = eta, y = sigmoid, label = label),
    nudge_x = 0.2,
    nudge_y = -0.06,
    hjust = 0
  ) +
  ggplot2::scale_x_continuous(breaks = seq(-6, 6, 2)) +
  ggplot2::labs(
    x = paste0("Log-odds, ", eta_symbol),
    y = paste0("Probability, logistic(", eta_symbol, ")")
  ) +
  ggplot2::theme_minimal()

if (knitr::is_html_output()) {
  htmltools::div(
    role = "img",
    `aria-label` = fig_alt,
    plotly::ggplotly(logistic_function_plot)
  )
} else {
  logistic_function_plot
}
```





**Definition 25** (Infinite set)**.** A set that is not finite (Definition 23) is **infinite** (Wikipedia contributors 2026n).

**Definition 26** (Countable set)**.** A set is **countable** (Wikipedia contributors 2026f) if it is finite (Definition 23), or if its elements can be listed as a sequence (Definition 18) $a_1, a_2, a_3, ...$ in which every element appears.

**Definition 27** (Countably infinite set)**.** A set is **countably infinite** if it is countable and not finite.

**Example 18** (Countable and uncountable sets)**.**

- $\{1, 2, 3, 4, 5, 6\}$ is countable, because it is finite.
- The non-negative integers $\{0, 1, 2, ...\}$ are countably infinite: the sequence $0, 1, 2, ...$ lists them.
- The integers are countably infinite too: the sequence $0, 1, -1, 2, -2, 3, -3, ...$ lists every one of them.

**Theorem 1** (The unit interval is not countable)**.** *The interval* $[0, 1]$ *is not countable.*

*Remark* 8 (Cantor's diagonal argument). The standard proof is Cantor's diagonal argument: given any sequence of numbers in $[0, 1]$, it builds a number in $[0, 1]$ whose decimal expansion differs from the $n$th number's in the $n$th digit (its $n$th decimal place, one of $0, 1, \ldots, 9$), so the sequence misses it (Wikipedia contributors 2026b). One way to choose the $n$th digit is to use 5, unless the $n$th number's $n$th digit is 5, in which case use 4. For example, if the sequence starts $0.1234\ldots$, $0.3579\ldots$, $0.2468\ldots$, the diagonal digits are 1, 5, and 6, so the new number starts $0.545\ldots$, which differs from the first number in the first digit, from the second in the second digit, and from the third in the third digit.

## References


Halmos, Paul R. 1974. *Naive Set Theory.* Undergraduate Texts in Mathematics. Springer. https://doi.org/10.1007/978-1-4757-1645-0.

Wikipedia contributors. 2026a. *Binary Relation — Wikipedia, the Free Encyclopedia.* https://en.wikipedia.org/wiki/Binary_relation.

Wikipedia contributors. 2026b. *Cantor's Diagonal Argument — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Cantor%27s_diagonal_argument.

Wikipedia contributors. 2026c. *Cardinality — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Cardinality.

Wikipedia contributors. 2026d. *Closure (Mathematics) — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Closure_(mathematics).

Wikipedia contributors. 2026e. *Codomain — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Codomain.

Wikipedia contributors. 2026f. *Countable Set — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Countable_set.

Wikipedia contributors. 2026g. *Domain of a Function — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Domain_of_a_function.

Wikipedia contributors. 2026h. *Finite Set — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Finite_set.

Wikipedia contributors. 2026i. *Function Composition — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Function_composition.

Wikipedia contributors. 2026j. *Function (Mathematics) — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Function_(mathematics).

Wikipedia contributors. 2026k. *Functional (Mathematics) — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Functional_(mathematics).

Wikipedia contributors. 2026l. *Graph of a Function — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Graph_of_a_function.

Wikipedia contributors. 2026m. *Image (Mathematics) — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Image_(mathematics).

Wikipedia contributors. 2026n. *Infinite Set — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Infinite_set.

Wikipedia contributors. 2026o. *Inverse Function — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Inverse_function.

Wikipedia contributors. 2026p. *Logistic Function — Wikipedia, the Free Encyclopedia*. https:

//en.wikipedia.org/wiki/Logistic_function.

Wikipedia contributors. 2026q. *Operator (Mathematics) — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Operator_(mathematics).

Wikipedia contributors. 2026r. *Range of a Function — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Range_of_a_function.

Wikipedia contributors. 2026s. *Sequence — Wikipedia, the Free Encyclopedia*. https://en.wikipedia.org/wiki/Sequence.