Mathematics for Data Science I
MATH 1004

Lewis Mitchell
School of Mathematical Sciences
mailto:lewis.mitchell@adelaide.edu.au

August 7, 2019

Preface

This course is about the mathematics underlying modern data science: linear algebra, probability, and some calculus. These form the foundation of most of the tools used by practising data scientists, including statistical analysis, mathematical modelling, and machine learning. While we'll spend much of our time working on understanding the mathematical fundamentals, we'll also look at how they each appear in real-world data science examples.

Apart from teaching the mathematics, the course is also concerned with an essential part of being a practising data scientist: computer programming and dealing with data computationally. Computer labs will therefore introduce Python, one of the foremost computer programming languages used in data science, and how to use it to analyse data and perform computational mathematics.

Plan 35+ scheduled classes, five written assignments (25%), 6 tutorials and 6 computer labs (5% participation total), one final examination (70%).

Contents

Bibliography

Graham, R. L., Knuth, D. E. & Patashnik, O. (1994), *Concrete Mathematics: A foundation fro computer science*, 2nd edition edn, Addison-Wesley.

Morris, C. C. & Stark, R. M. (2015), *Fundamentals of Calculus*, Wiley. Available online from the Library website.

Stewart, J. (2012), *Calculus*, 7th edn, Brooks/Cole. Available in the library (hardcopy).

1 Preliminaries

Contents

1.1 What is data science?

Our first piece of data science! What is it when you search?

Data science can mean many different things to different people, all the way from "a multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data", to "statistics on a Mac." In May 2019 a Google search for "what is data science?" (with the quotes) gives 341,000 results. Accordingly, let's adopt a definition that blends a few different fields, following a now-classic Venn diagram from the famous data scientist Drew Conway (see Figure 1.1).

There's also the version made by another famous data scientist, Hilary Mason (Figure 1.2).

I prefer Conway's original diagram, partly because it avoids the "nerd" framing, but mainly because it includes the "substantive (domain) expertise" set. Data science techniques can be applied in many different domains, and good practising data scientists find themselves needing to learn a little bit about a lot of things.

In this course we'll focus mainly on the "math and statistics" parts of both diagrams, with a bit of "hacking" mixed in. Therefore, we won't have fully become data scientists by the end of this course, but we'll be on strong foundations, and as we continue to gain "substantive expertise" in other fields we'll be well on our way.

1.2 Notation

Through the Venn diagrams in Figure 1.1 and Figure 1.2 we've already introduced the concept of *sets*, so let's start there.

A set is a collection of objects called *elements*. The notation $\{x \mid \ldots\}$ or $\{x : \ldots\}$ is read "the set of objects x such that ...".

What follows is a summary of notation commonly used.

$x \in A$: the object x is an element of the set A

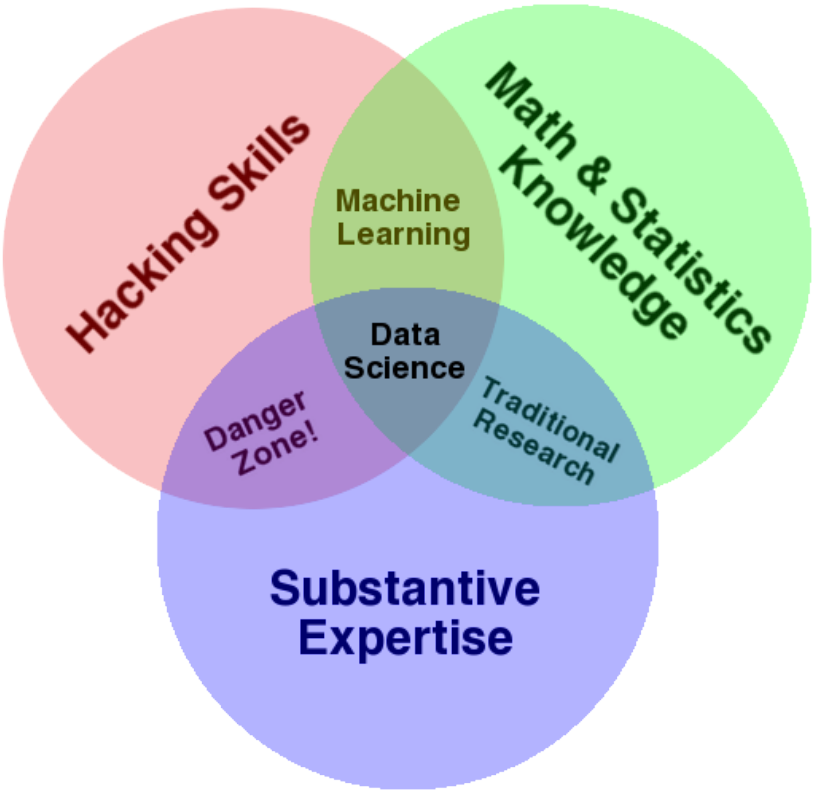


Figure 1.1: Drew Conway's data science Venn diagram.

$\emptyset$: the *empty* set, that is, the set with no elements at all

$A \subseteq B$: the set A is contained in B, that is, every element of A is also an element of B. This does not exclude the possibility that $A = B$. We say A is a subset of B.

$A \subset B$: A is a proper subset of B, meaning A is contained B but they are not equal.

$A \cup B$: the *union* of the sets A and B: $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$

$A \cap B$: the *intersection* of the sets A and B: $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$

$A \backslash B$: the *difference* of the sets A and B: $A \backslash B = \{x \mid x \in A \text{ but } x \notin B\}$

We can depict these basic set operations using Venn diagrams. For example, the following diagrams show $A \cap B$ and $A \cup B$:

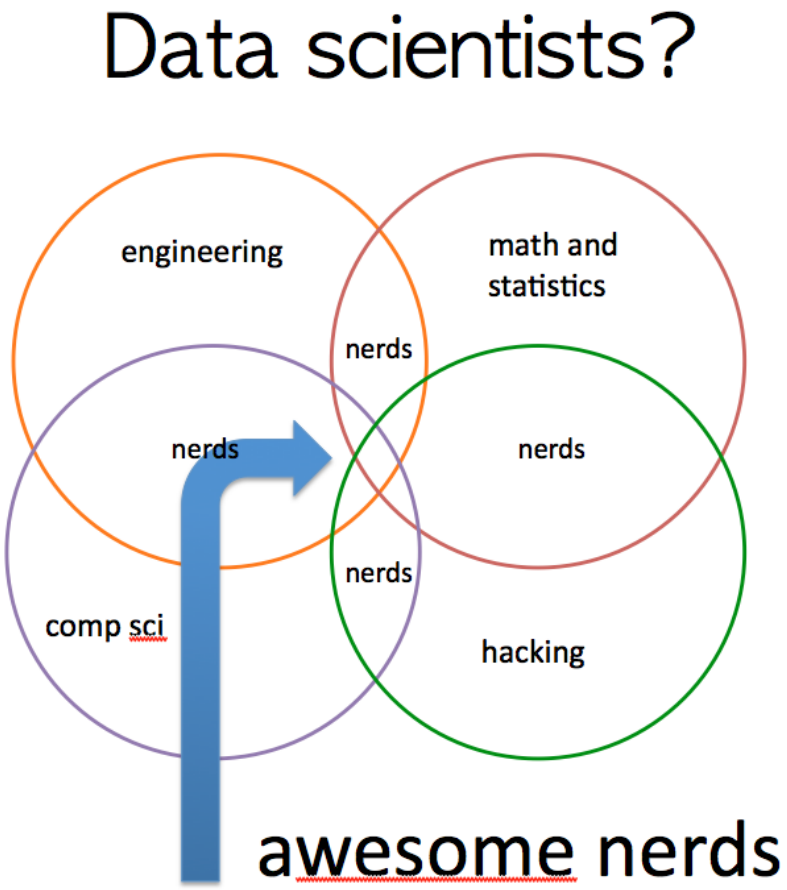
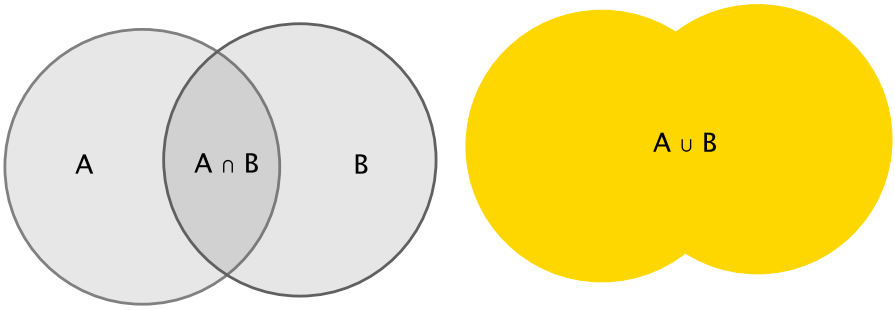


Figure 1.2: Hilary Mason's data science Venn diagram.



Example 1.1. Let $A = \{1, \pi, \mathfrak{m}, 23, \&\}$ and $B = \{1, \mathfrak{k}, \$, \mathfrak{m}, \pi^2\}$. Describe $A \cup B$, $A \cap B$ and $A \setminus B$.

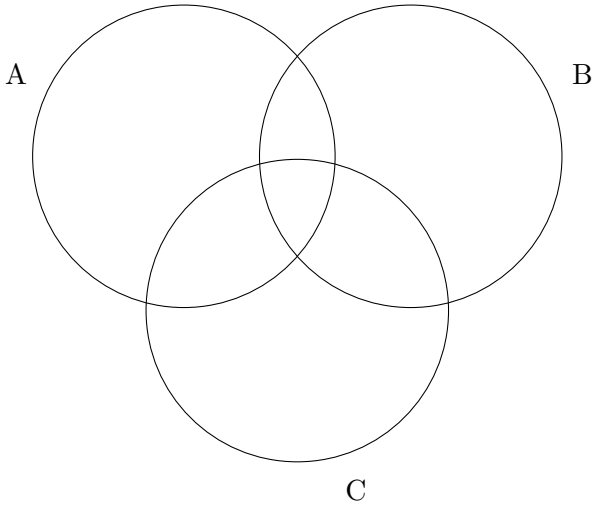
Example 1.2. What is $\{1, 2, 3\} \cup \{2, 4, 6\}$? What about $\{1, 2, 3\} \cap \{2, 4, 6\}$? Or $\{1, 2, 3\} \setminus \{2, 4, 6\}$?

Example 1.3. Let A, B and C be sets. Consider the following sets and determine which ones describe the same sets (Venn diagrams may be helpful).

1. (i) $A \cap (B \cup C)$ (ii) $(A \cap B) \cup C$ (iii) $(A \cap B) \cup (A \cap C)$

2. (i) $(A \setminus B) \cap (A \setminus C)$ (ii) $(A \setminus B) \cup (A \setminus C)$ (iii) $A \setminus (B \cap C)$.

A template for a Venn diagram with three sets is given below:



Sets of Numbers

In this course we will deal with various types of numbers,

The natural numbers, $\mathbb{N}$: The positive whole numbers $\{1, 2, 3, 4 \ldots\}$.

The integers, $\mathbb{Z}$: All positive and negative whole numbers $\{\ldots -3, -2, -1, 0, 1, 2, 3, \ldots\}$. Note that $\mathbb{N} \subset \mathbb{Z}$.

The rational numbers, $\mathbb{Q}$: These are fractions, such as $\frac{1}{2}$, $-\frac{5}{4}$, 2 (since $2 = \frac{2}{1}$). Note that $\mathbb{Z} \subset \mathbb{Q}$.

The real numbers, $\mathbb{R}$: This set includes all of the rationals, as well as other numbers which can't be represented as fractions, such as $\sqrt{2}$ and π. The best way to think of it is by forming a number line where you fill in all the gaps between rational numbers. We have $\mathbb{Q} \subset \mathbb{R}$.

The complex numbers, $\mathbb{C}$: An extension of the real numbers to incorporate $\sqrt{-1}$, we won't focus on these in this course.

The symbol $\forall$ means "for all".
The symbol $\exists$ means "there exists".

Often we use statements such as "if A is true then B is true", which may write as $A \Rightarrow B$. If we have both $A \Rightarrow B$ and $B \Rightarrow A$ then we say "A is true if and only if B is true". This is often written as "A is true iff B is true", where "iff" is an abbreviation for "if and only if". This can also be written as $A \Leftrightarrow B$.

Example 1.4. Work out what the following expressions mean and hence decide whether they are true:
(i) $\forall\, x \in \mathbb{Q},\ 2x \in \mathbb{Z}$
(ii) $\exists\, x \in \mathbb{Q},\ 2x \in \mathbb{Z}$

Example 1.5. Describe the elements of the following sets:
(i) $\{x \in \mathbb{R} \mid 0 < x \leqslant 1\} \cap \{z \in \mathbb{R} \mid 2z \in \mathbb{Z}\}$

(ii) $\{\frac{1}{2}, 1, \frac{3}{2}, 2\} \cap \mathbb{Q} \setminus \mathbb{Z}$.

Example 1.6. Are each of the following true or false?

(i) $\forall\, x \in \mathbb{Z},\ \frac{x}{2} \in \mathbb{Z}$

(ii) $\exists\, x \in \mathbb{Z},\ \frac{x}{2} \in \mathbb{Z}$

1.3 Estimation

Much of the time when practising data science we'll be taking some dataset (often, but not always, represented as numbers), performing some statistical/mathematical operations on it, and coming up with some number in answer to some question we've asked of the dataset. But how do you know if the number you've computed is reasonable? A fundamental skill is learning how to perform a *sanity check* on numerical values you calculate in order to be sure that they're reasonable. One classic method to do this is to use a technique called *Fermi estimation*, which is essentially a back-of-the-envelope calculation. For example, consider the question:

For example, you'd be surprised by the number of times we see negative probabilities when marking exams!

How many piano tuners are there in Adelaide?

Fermi's original problem used Chicago instead of Adelaide.

This may seem like a very tricky question to answer, and indeed to get an exact figure would require a lot of effort (like a census, or a lot of Googling followed by manual checking), but we can easily get an approximation as follows:

1. There are approximately $N = 1,000,000$ people living in Adelaide.
2. On average, there are say $n_h = 2$ people per household.
3. The fraction of houses with pianos is roughly $f_p = 1/20 = 0.05$.
4. Pianos need to be tuned, say, $n_t = 1$ time per year.

5. Say it takes about $\mathsf{T} = 2$ hours to tune a piano, including travel time, meaning that a piano tuner can tune around 4 pianos in a typical 8-hour day.

6. A typical piano tuner works 5 days a week, 50 weeks per year.

(My confidence about each of these guesses gets lower as we go along, but remember we're only after a rough approximation here.)

From points 1-3 we can approximate that there are

$$\frac{\mathsf{N}}{\mathfrak{n}_{\mathfrak{h}}} \times \mathfrak{f}_{\mathfrak{p}} = 25,000$$

pianos in Adelaide, and from point 4 that there are the same number of tunings done each year. Points 5-6 then tell us that the average piano tuner can tune 1,000 pianos in a year, and so to tune all the pianos there must be around 25 piano tuners in Adelaide!

Now, in May 2019, a quick check of the Yellow Pages tells me that this is a little bit of an underestimate, and there are actually 82 piano tuning & piano repair businesses in Adelaide. But our guess was not far off, and most importantly, if someone had come to us and suggested there are 2000 piano tuners in Adelaide, or 2, we could have pretty quickly decided that they were talking rubbish.

This method 'works' because by breaking the problem down into smaller pieces, we can make estimates for each term which are close to correct. And if those estimates are roughly equally likely to be under- or overestimates, the errors will have a tendency to cancel each other out. The moral is that when we're presented with a numerical value, or when we compute one ourselves, we should always step back and think: does this value make *sense*? Fermi estimates are a great way to do a reality check.

General approach to Fermi estimation:

1. Decide on a suitable proxy: e.g., a proxy for piano tuners is the population of Adelaide

2. Break down into smaller pieces (find differentiated clusters): From the population of Adelaide, we broke down to households, because we expect there'll only be one piano per household.

3. Do calculations and round off: Notice that in general we're only after an *order of magnitude*, so we don't care much about the difference between say 1×10^5 and 2×10^5. But we do care about the difference between 10^5 and 10^6. So we'll generally round to the nearest order of magnitude to be able to do some quick calculations.

4. Validate! The aim of all of this is to sanity-check numerical values. So after calculating a number it's good to think if it

makes sense. We'd be pretty sure we'd have made a mistake if we calculated 100,000 piano tuners, for example.

Example 1.7. How many WhatsApp users are there in Australia?

2 Fundamentals

Contents

2.1 A motivating example

OK, maybe this isn't the most modern of DS/ML, but it's an immediately understandable one.

To motivate a few of the upcoming topics (functions and Taylor series, but also many of the later topics as well including linear algebra and probability), let's take an actual data science/machine learning example. This is a very famous one, both from history and as the introductory tutorial on the online data science/ML community website `www.kaggle.com`: understanding and predicting who survived the Titanic disaster.

The key question for us is

Who survived the Titanic?

By which I mean: which types of passengers survived? We hear about women and children being the first to the lifeboats, but was this actually the case? Were the survivors more likely to be:

- women?
- children?
- crew?
- rich people?

This is the sort of question we might be able to answer with some data. Figure 2.1 shows a snippet of a *dataframe* coming from the Kaggle dataset, containing some data on a few of the passengers. The column `Survived` shows whether the passenger survived (1) or not (0).

```
df1 = pd.read_csv('../titanic/train.csv')
df1.tail()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

Figure 2.1: Python code and dataframe snippet from the Titanic dataset.

We can see a lot of potentially useful information here for answering our question, including the Passenger Class of the person's ticket (`Pclass`), their `Sex`, `Age`, and how much they paid for their ticket in pounds (`Fare`). To focus on the *continuous variables* age and fare, we can start to get a feel for who was more likely to have survived the Titanic by making some histograms, as shown in Figure 2.2.

Even though it's a number, `Pclass` is an *ordinal* variable.

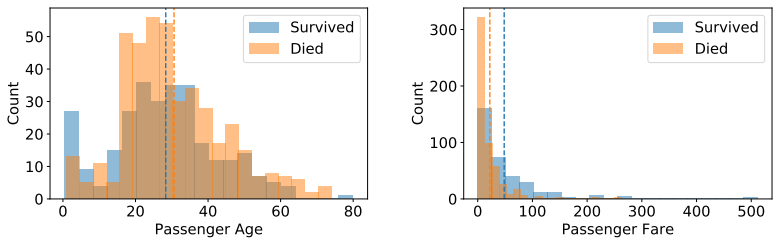


Figure 2.2: Left: Histogram of passenger ages. Right: Histogram of passenger fares. Both are segmented by the passengers' survival status.

These histograms already start to tell the story that young passengers were in general not much more likely to survive than older passengers, whereas passengers who paid more may have been more likely to survive than those who paid less. But it's fairly difficult to distinguish – what we would like here is a statistical model which shows the probability of survival as a function of age or fare. Figure 2.3 shows one such model.

The model being fit here is *logistic regression*, a fundamental statistical machine learning tool.

This now tells an interesting story: younger passengers were slightly more likely to survive, but not by a lot. (All the probabilities shown here are less than 0.5.) On the other hand, richer passengers were *much* more likely to survive than poorer passengers were. The model makes this story very easy to see!

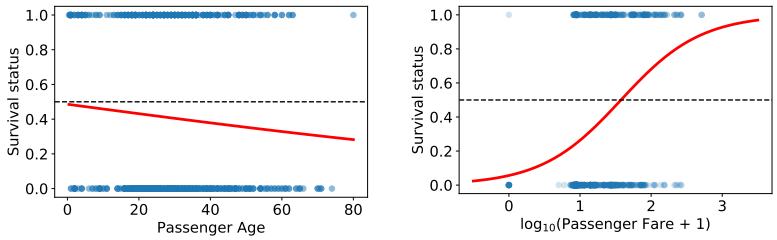


Figure 2.3: Left: Model showing survivial probability as a function of passenger age. Right: Model showing survivial probabiity as a function of a transformation of passenger fare. The blue dots show the raw data.

This model is a function of the form:

$$f(x) = \frac{1}{1 + e^{-(ax+b)}} \tag{2.1}$$

where $a, b \in \mathbb{R}$. These numbers get estimated automatically as part of the statistical modelling process (which we don't need to go into here). At left x represents the passenger age, and at right it represents $\log_{10}$ of the passenger fare, plus 1. (This transformation of the data was motivated by the shape of the histogram in Figure 2.2.) So really, the function at right is a function f of another function $g(x) = \log_{10}(x + 1)$, or $f(g(x))$.

Why did I need to add 1 before taking the logarithm?

This example has introduced a number of concepts, and ties together many of the major concepts for this course!

1. I've introduced *functions* only very loosely above, and a rule (the function-of-a-function $f(g(x))$ for manipulating them. Before we go anywhere we immediately need to make our definitions of these funamental objects more concrete, which we do in the next section.

2. How does a computer estimate something so unusual as (2.1)? The answer lies in *Taylor series*, where we will then proceed.

3. Representing the data from the dataframe above so that our model can be fit requires *linear algebra* at a very deep level.

4. The model that's being fit here is a *probabilistic* model, which necessitates an understanding of probability.

5. Computationally, the way this model actually gets "fit" to the data involves gradient descent, which is built upon *calculus*.

We'll get to each of these in turn. First, because we've introduced one already in (2.1), we need to talk about functions and their inverses.

2.2 Functions

Mathematics provides us with the means to study and manipulate a huge range of types of relationship which can be expressed in the form of a *function*:

$$y = f(x).$$

However, in order to use these tools, we first need to make our ideas about functions as rules relating variables a little more precise. To do this, we begin by introducing the following definitions.

Definition 2.1. *A real valued* function $f : \mathcal{D} \to \mathcal{R}$ *is a rule (or set of rules) which assigns to each element* $x \in \mathcal{D}$ *a unique real number denoted by* $f(x)$*. The set of possible values of the independent variable,* $\mathcal{D}$*, is called the* domain*. We define the* range *of* f*,* $\mathcal{R}$*, to be the set of possible values of the output of* f *- i.e.,*

$$\mathcal{R} = \mathcal{R}(f) = \{y \mid y = f(x) \text{ for some } x \in \mathcal{D}\}.$$

As we concern ourselves only with real-valued functions of a real variable in this course, both the domain and range will be a subset of the reals $\mathbb{R}$.

The key idea of this definition is just that: *a function is like a machine for turning input numbers (from the domain) into output numbers (that form the range).* To determine what numbers form the domain, you should think about ways to break the machine. Any number that breaks the machine can't be in the domain. For example:

Example 2.2. If we write $f(x) = \sqrt{x+2}$, and do not specify the domain, then implicitly we mean

$$f(x) = \sqrt{x+2}, \quad x \geqslant -2;$$

that is, the domain is $[-2, \infty)$—the largest set of real numbers for which $f(x)$ is real-valued.

Basic functions

Polynomials :

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

is a polynomial function of degree n (if $a_n \neq 0$; each of the $a_0, a_1, \ldots, a_n$ is a (constant) real number).

By convention, each polynomial $p(x)$ has domain $\mathbb{R}$ (unless otherwise specified).

Rational Functions : If $p(x)$ and $q(x)$ are polynomial functions

$$r(x) = \frac{p(x)}{q(x)} \quad \text{is a } \textit{rational function}.$$

The domain of $r(x)$, is $\mathcal{D} = \{x \mid q(x) \neq 0\}$.

Absolute Value Function : The absolute value function $f(x) = |x|$ is defined by a two part rule:

$$f(x) = |x| = \begin{cases} x & \text{if } x \geqslant 0, \\ -x & \text{if } x < 0 \end{cases}$$

Domain $\mathcal{D} = \mathbb{R}$; the range is $\{x \mid x \geqslant 0\} = [0, \infty)$.

Data science & machine learning functions

There are plenty more rules that satisfy the definition of being a function and which you may not be familiar with. Here are a few that are commonly used in data science and machine learning.

The Greatest Integer Function : $h(x) = \lfloor x \rfloor$ = the greatest integer less than or equal to x. $h : \mathbb{R} \to \mathbb{R}$.

The domain is $\mathbb{R}$; the range is $\mathbb{Z}$ (the integers).

This is an example of a "*step function*"
Example 2.3. Graph of greatest integer function.

Logistic function : This is the type of function being plotted in Figure 2.3. The logistic function $f(x) = \text{logistic}(x)$ is defined by:

$$\text{logistic}(x) = \frac{1}{1 + e^{-(ax+b)}} = \frac{e^{ax+b}}{1 + e^{ax+b}}$$

for constants $a, b \in \mathbb{R}$. This function is essential in *logistic regression.*

Logit function : Very much connected to the logistic function is the *logit* function, in that they are inverse functions of each other (as we will discover in the next sections). It is defined as

$$\text{logit}(x) = \log\left(\frac{x}{1-x}\right)$$

for $a = 1$, $b = 0$.

Once we cover inverse functions, a good exercise would be to derive the more general logit function for general a and b.

ReLU and softplus : The rectified linear unit (ReLU) function is a common activation function in neural networks. It is defined as

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x \geqslant 0, \\ 0 & \text{if } x < 0. \end{cases}$$

The domain and range are the same as for $|x|$.

A smooth approximation to this function is the *softplus* function:

$$\text{softplus}(x) = \log\left(1 + e^{x}\right).$$

Definition 2.4 (even and odd functions)**.**

- *a function* f *is* even *if* $f(-x) = f(x)$;
- *a function* f *is* odd *if* $f(-x) = -f(x)$*;*

for all x *in its domain.*

Emmy Noether proved symmetries underpin physical laws.

Even functions are symmetric about the y-axis (reflection) (Stewart 2012, pp.17–18).

Odd functions are symmetric with respect to the origin (point/rotation).

2.3 Constructing new functions from old (Stewart 2012, §1.3)

Recall we shift a functions by adding/subtracting a constant from the function or its argument (Stewart 2012, p.36). Also, we stretch a function vertically or horizontally through multiplying or dividing, by some constant, the function or its argument (Stewart 2012, p.37).

Combining two functions f, g we form new functions (Stewart 2012, p.39):

the sum $f + g$	$(f + g)(x) = f(x) + g(x)$
difference $f - g$	$(f - g)(x) = f(x) - g(x)$
product $f \cdot g$	$(f \cdot g)(x) = f(x)g(x)$
quotient $\frac{f}{g}$	$\left(\frac{f}{g}\right)(x) = \frac{f(x)}{g(x)}$

The domain of $f + g$, $f - g$, $f \cdot g$ is the intersection of the domains of f and g, $\mathcal{D}(f) \cap \mathcal{D}(g)$.

For $\frac{f}{g}$ we must also exclude the numbers x such that $g(x) = 0$.

Definition 2.5 (composition of functions)**.** *Given any two functions* f *and* g (Stewart 2012, p.40)*,*

$$(f \circ g)(x) = f(g(x)) \quad \textit{and} \quad (g \circ f)(x) = g(f(x))$$

These are almost always not *equal.*

Domains of $f \circ g$ and $g \circ f$ In general the domain of $f \circ g$ (or $g \circ f$) depends on both the domain of f and g. The composition $f \circ g$ has domain $\{x \in \mathcal{D}(g) \mid g(x) \in \mathcal{D}(f)\}$.

2.4 Inverse functions (Stewart 2012, §6.1)

Definition 2.6 ((Stewart 2012, §6.1, p.384)). *A function* $f : \mathcal{D} \to \mathcal{R}$ *($\mathcal{D}, \mathcal{R}$, the domain and range of* f *are subsets of* $\mathbb{R}$*) is* one-to-one, 1–1, *if*

$$\text{for any } x_1, x_2 \in \mathcal{D}, \text{ if } x_1 \neq x_2 \text{ then } f(x_1) \neq f(x_2).$$

Equivalently, f *is 1–1 provided*

$$\text{if } f(x_1) = f(x_2) \text{ then } x_1 = x_2.$$

Horizontal Line Test A function f is 1–1 if and only if any horizontal line meets the graph of f *at most once*.

Example 2.7. The inverse function of the logistic function

$$f(x) = \frac{e^x}{1 + e^x}$$

is the *logit* function

$$f^{-1}(x) = \log\left(\frac{x}{1-x}\right)$$

Method for finding f^{-1}:

1. Write $y = f(x)$;
2. Solve this equation for x; $x = f^{-1}(y)$;
3. *Interchange* x and y;

and then $y = f^{-1}(x)$.

Note

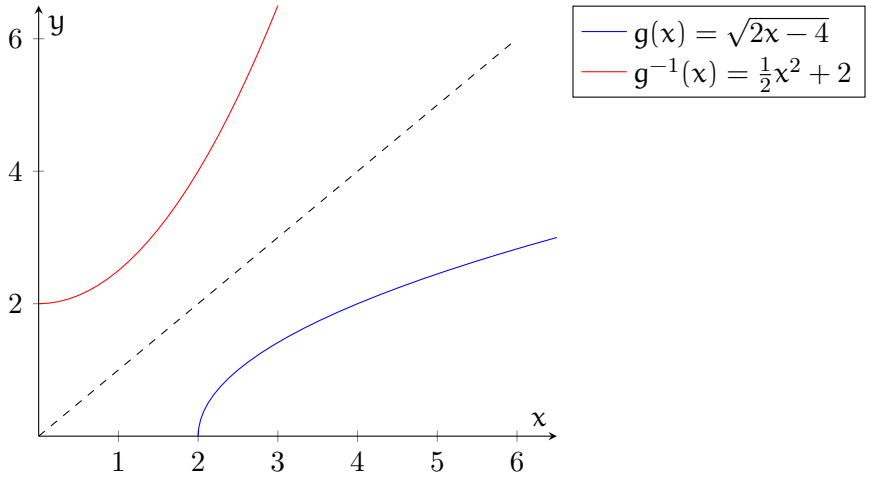
1. If $y = f(x)$ then $x = f^{-1}(y)$ and if $x = f^{-1}(y)$ then $y = f(x)$. That is, $y = f(x)$ if and only if $x = f^{-1}(y)$.
2. f^{-1} does *not* denote $\frac{1}{f}$.
3. f must be 1–1 in order to define f^{-1}.
   Example 2.8. If $f(x) = x^2$, $x \in \mathbb{R}$, then $f(2) = f(-2) = 4$. If f^{-1} existed, what is $f^{-1}(4)$? Is it 2 or -2? Conclusion: f^{-1} cannot exist.

   In general, if f is not 1–1 then somewhere $f(x_1) = f(x_2) = y$, and then what is $f^{-1}(y)$? x_1? or x_2? For f^{-1} to be a function, $f^{-1}(y)$ must be a unique number.

The graph of the inverse If f has an inverse f^{-1}, then the graph of $y = f^{-1}(x)$ is obtained by reflecting the graph of $y = f(x)$ in the line $y = x$.

As $y = f(x)$ if and only if $x = f^{-1}(y)$: if (a, b) is on the graph of f, then (b, a) is on the graph of f^{-1}.

Example 2.9. $g(x) = \sqrt{2x-4}$



2.5 Series and sums

Have you ever wondered how a computer or your calculator can calculate something like $\sin(x)$ or $\log(x)$? Computers are fairly 'dumb' machines, in that they can really only do *arithmetic operations*: addition, subtraction, multiplication, and division. How then do they evaluate anything non-arithmetic such as $\sqrt{}$, $\tan(\,)$, or $\exp(\,)$? The answer lies in *series approximations* of functions. These are absolutely fundamental not just for calculations, but for the sorts of approximations underlying many machine learning methods in general.

2.5.1 Summation notation

(Graham et al. 1994, Ch. 2)

We'll be working a lot with sums such as

$$1 + 2 + 3 + \ldots + (n-1) + n,$$

which is the sum of the first n natural numbers. The ... means 'continue the pattern of the surrounding terms'. In general we talk about sums like

$$a_1 + a_2 + \ldots + a_n,$$

where each a_k is a *term* that needs to be defined somehow. Instead of writing out the '...' each time, we shorten the above using *summation notation*

$$\sum_{k=1}^{n} a_k = a_1 + a_2 + \ldots + a_n, \tag{2.2}$$

where the $\sum$ bit means 'sum the a_k's, where k is an integer from 1 to n (inclusive)', or 'sum over k, from 1 to n'.

We can take the 'sum over' idea further:

$$\sum_{n=1}^{n} a_k = \sum_{1 \leqslant k \leqslant n} a_k.$$

Here the RHS means 'sum a_k over all integers between 1 and n', and is equal to the same sum as Equation 2.2.

Properties of $\sum$:

1. $\sum_{i=m}^{n} ca_i = c\sum_{i=m}^{n} a_i$, for c a constant;
2. $\sum_{i=m}^{n}(a_i + b_i) = \sum_{i=m}^{n} a_i + \sum_{i=m}^{n} b_i$;
3. $\sum_{i=m}^{n}(a_i - b_i) = \sum_{i=m}^{n} a_i - \sum_{i=m}^{n} b_i$.
4. But $\sum_{i=m}^{n}(a_i \times b_i) \quad \neq \quad (\sum_{i=m}^{n} a_i) \times (\sum_{i=m}^{n} b_i)$;
5. and $\sum_{i=m}^{n} \frac{a_i}{b_i} \quad \neq \quad \frac{\sum_{i=m}^{n} a_i}{\sum_{i=m}^{n} b_i}$.

Points 1 and 2 are very important – they tell us that (finite) summation is a *linear operator*:

Definition 2.10 (Linear operator)**.** *An operator* L *is* linear *if for all functions* f *and* g*, and every scalar* $c \in \mathbb{R}$*,*

$$L(cf) = cL(f)$$
$$L(f + g) = L(f) + L(g).$$

Linear operators are a central theme throughout this course – we will encounter a number of other important operations throughout which share this same characteristic, and will call out this theme explicitly!

Some important sums are those of constants, linear and squares.

1. $\sum_{i=1}^{n} 1 = \underbrace{1 + \cdots + 1}_{n} = n.$
2. $\sum_{j=0}^{n} ar^j = a + ar + ar^2 + \cdots + ar^n = a\frac{1-r^{n+1}}{1-r}$ is the geometric sum.
3. $\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$
4. $\sum_{i=1}^{n} i^2 = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} = \frac{n}{6}(2n^2 + 3n + 1) = \frac{n}{6}(2n+1)(n+1)$

Evaluate $\sum_{i=3}^{10}(i+2)^2$.

Further examples and exercises:

- (Stewart 2012, pg. A37-8)

2.5.2 Proof by induction

The derivations of the "important sums" above may have seemed a little bit like they came out of nowhere, which is often the case with results about sums. There are a huge range of approaches to calculating sums, and we won't cover all of them here (Graham et al. 1994, Ch. 2 goes into great depth the variety), but one relatively straightforward approach (which also comes up a lot in future mathematics courses related to data science) is *Proof by Mathematical Induction*. While this is a non-constructive method of proof (i.e., you have to have guessed the correct formula already), it's still quite nice because it's relatively procedural. The procedure is as follows.

Definition 2.11 (Principle of Mathematical Induction)**.** *Consider a statement* $P(n)$ *that is to be proved.*

1. *Basis step: show that* $P(a)$ *is true.*
2. *Inductive step: assume* $P(k)$ *is true, use this to prove* $P(k+1)$*.*

Then $P(n)$ *is true for all integers* $n \geqslant a$*.*

(Graham et al. 1994, pg. 3) has a nice quote which describes the essential idea of mathematical induction:

Just like these notes, Concrete Mathematics has wide margins with lots of side notes! In fact this quote appears in the margin itself.

> *Mathematical induction proves that we can climb as high as we like on a ladder, by proving that we can climb onto the bottom rung (the basis) and that from each rung we can climb up to the next one (the induction).*

Example 2.12. Prove that

$$\sum_{j=1}^{n} \frac{1}{j(j+1)} = \frac{n}{n+1}.$$

Example 2.13. Prove that

$$\sum_{j=1}^{n} 2^{j-1} = 2^n - 1.$$

We can prove the results about "interesting sums" 3 and 4 above similarly using induction. These are good exercises for the reader!

2.5.3 Multiple summation

We might sometimes see something like

$$\begin{aligned}\sum_{1 \leqslant (j,k) \leqslant 3} a_j b_k = &a_1 b_1 + a_1 b_2 + a_1 b_3 \\ &+ a_2 b_1 + a_2 b_2 + a_2 b_3 \\ &+ a_3 b_1 + a_3 b_2 + a_3 b_3\end{aligned}$$

and this is the same as

$$\sum_{j=1}^{3}\left(\sum_{k=1}^{3} a_j b_k\right).$$

In this course, we'll generally use the latter notation for multiple sums like this, but you should be aware of the former notation as well (and similarly for regular sums).

These types of summations can often be simplified using the following rules:

Definition 2.14. *1. Associativity:*

$$\sum_{j\in J}\sum_{k\in K} a_{j,k} = \sum_{k\in K}\sum_{j\in J} a_{j,k}$$

2. Distributivity:

$$\sum_{j\in J,k\in K} a_j b_k = \left(\sum_{j\in J} a_k\right)\left(\sum_{k\in K} b_k\right)$$

(Note: these are specific to *finite* sums!)

So our example can be simplified to

$$\sum_{1\leqslant(j,k)\leqslant 3} a_j b_k = \left(\sum_{j=1}^{3} a_j\right)\left(\sum_{k=1}^{3} b_k\right).$$

Example 2.15. Compute

$$\sum_{i=1}^{3}\sum_{j=1}^{2}(i-j).$$

2.5.4 Infinite series

The final piece we need before being able to talk fully about Taylor series is what it means to add an infinite number of terms, i.e., to compute an *infinite series*.

An infinite series is an expression of the form

$$\sum_{n=1}^{\infty} a_n = a_1 + a_2 + \cdots + a_i + \cdots$$

where the a_i are real numbers.

The N*th partial sum* S_N is the sum of the first N terms

$$S_N = a_1 + a_2 + \cdots + a_N$$

We say the infinite series $\sum_{n=1}^{\infty} a_n$ is *convergent* with *sum* S provided

$$S = \lim_{N\to\infty} S_N.$$

If $\lim_{N\to\infty} S_N$ does not exist we say that the series *diverges*.

What does $\lim_{N\to\infty} S_N$ mean? It is like $\lim_{x\to\infty} f(x)$ but n takes only integer values. So we mean that the partial sums S_N get arbitrarily close to the value S when N gets sufficiently large (i.e. as $N \to \infty$).

Note that there is a precise definition of a limit as x tends to infinity. This is not part of the present course but we give it here.

Definition 2.16. $S_N (N = 1, 2, 3, \cdots)$ *has limit* S *provided that for any* $\epsilon > 0$ *there exists a positive number* M *such that* $|S_N - S| < \epsilon$ *whenever* $n > M$.

Example 2.17. The **p- series** $\sum_{n=1}^{\infty} \frac{1}{n^p}$ is convergent if $p > 1$ and divergent if $p \leqslant 1$.

This result depends on the Integral Test for series which is not covered in this course. This is nonetheless the most useful result to remember regarding infinite series!

Example 2.18. The series $\sum_{n=1}^{\infty} (-1)^n$ diverges.

The Geometric Series Consider the series

$$\sum_{n=0}^{\infty} x^n = 1 + x + x^2 + \dots.$$

This is called the *geometric series* with common ratio x.

$$S_N = \frac{1 - x^{N+1}}{1 - x} = \frac{1}{1 - x} - \frac{x^{N+1}}{1 - x}.$$

Consider what happens as $N \to \infty$. If $|x| < 1$ then $x^{N+1} \to 0$ so $S_N \to \frac{1}{1-x}$, i.e.

$$\sum_{n=0}^{\infty} x^n = \frac{1}{1 - x} \quad \text{if} \quad |x| < 1.$$

If $|x| > 1$ then the term $x^{N+1} \to \pm\infty$ and so $\sum_{n=0}^{\infty} x^n$ diverges.

Sometimes we encounter a geometric series of the form

$$\sum_{n=0}^{\infty} ax^n = a + ax + ax^2 + \cdots = a(1 + x + x^2 + \dots).$$

This diverges if $|x| > 1$ and converges to $a/(1-x)$ if $|x| < 1$.

There are a number of tests used to see if a given series is convergent. We look at one useful test.

2.5.5 The Ratio Test

Consider a series $\sum_{n=0}^{\infty} a_n$, with each $a_n \neq 0$, such that

$$r = \lim_{n\to\infty} \left| \frac{a_{n+1}}{a_n} \right|$$

either exists or is infinite.
Then

if	$r < 1$	the series converges,
if	$r > 1$	the series diverges,
if	$r = 1$	the ratio test is inconclusive.

Example 2.19. Prove that $\sum_{n=0}^{\infty} \frac{(-1)^n n}{2^n}$ converges.

Example 2.20. Find if $\sum_{n=1}^{\infty} \frac{2^n}{n^2}$ converges or diverges.

Further examples and exercises:

- (Stewart 2012, pg. 758-63)
- (Morris & Stark 2015, pg. 261-2, Ex. 9.4, Q18-23)

2.6 Taylor series

2.6.1 Approximating functions by polynomials

2.6.2 Taylor polynomials and Taylor's theorem

2.6.3 Taylor, Maclaurin, and power series

??: