

Basic mathematical symbols and stuff

What do all these pictures and letters represent?



WORKBOOK

Presented by

Dr Chris Jackson

Reliability engineer



Copyright © 2022 Christopher Jackson

All rights reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means, including photocopying or other electronic or mechanical methods without the prior written permission of the author (Christopher Jackson), except as permitted under Section 107 or 108 of the 1976 United States Copyright Act and certain other non-commercial uses permitted by copyright law.

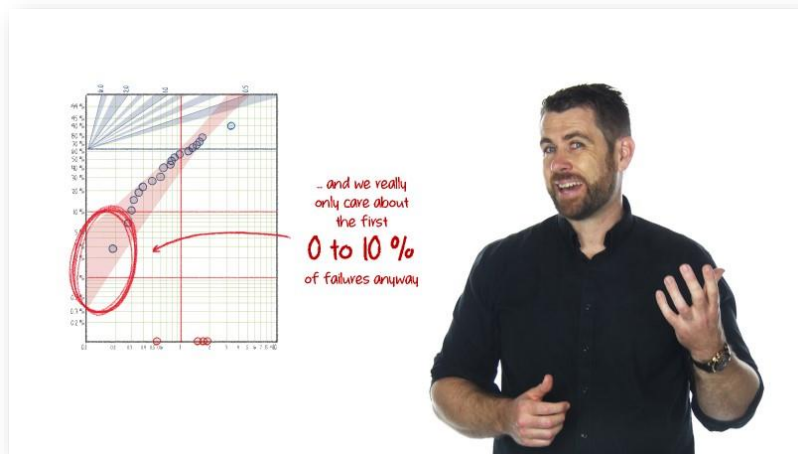
For permission requests, write to Christopher Jackson using the details on the following pages.

Limit of Liability/Disclaimer of Warranty: While the author has used his best efforts in preparing this document, he makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. The author shall not be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

Who is your teacher? ... Dr Chris Jackson

Dr Jackson holds a Ph.D. and MSc in Reliability Engineering from the University of Maryland's Center of Risk and Reliability. He also holds a BE in Mechanical Engineering, a Masters of Military Science from the Australian National University (ANU) and has substantial experience in the field of leadership, management, risk, reliability and maintainability. He is a Certified Reliability Engineer (CRE) through the American Society of Quality (ASQ) and a Chartered Professional Engineer (CPEng) through Engineers Australia.

Dr Jackson is the cofounder of IS4 online learning, was the inaugural director of the University of California, Los Angeles (UCLA's) Center of Reliability and Resilience Engineering and the founder of its Center for the Safety and Reliability of Autonomous Systems (SARAS). He has had an extensive career as a Senior Reliability Engineer in the Australian Defence Force, and is the Director of Acuitas Reliability Pty Ltd.



He has supported many complex and material systems develop their reliability performance and assisted in providing reliability management frameworks that support business outcomes (that is, make more money). He has been a reliability management consultant to many companies across industries ranging from medical devices to small satellites. He has also led initiatives in complementary fields such as systems engineering and performance-based management frameworks (PBMFs). He is the author of two reliability engineering books, co-author of another, and has authored several journal articles and conference papers.

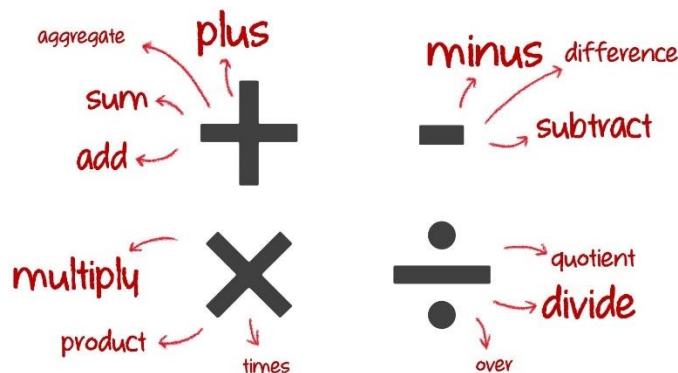
Arithmetic and other math stuff

This is where some people get a little bit scared ... because we are going to start talking about mathematics! This is where our **ponderous professor** loves to ruin your day. But ... we do need to indulge him a little bit. So we will try and limit the extent our **ponderous professor** tries to overwhelm us. We will go through basic mathematics, keep it simple, and limit it to the stuff we need to know to be good **reliability engineers**.



what is ARITHMETIC?

Arithmetic is what we associate with 'mathematics.' **Arithmetic** deals with numbers and how we apply what we call **operations** to them. The most common **arithmetic operations** are illustrated below.



These **operations** are examples of **binary operations**.

a BINARY OPERATION a rule for combining two values to create a new value.

A visual representation of the addition 5 + 3 = 8. On the left, there are 5 blue circles and 3 red circles. In the middle is a plus sign. To the right of the plus sign is an equals sign, followed by 8 grey circles. The numbers 5 and 3 are written above their respective groups of circles.

A visual representation of the subtraction 8 - 4 = 4. On the left, there are 8 grey circles. In the middle is a minus sign. To the right of the minus sign is an equals sign, followed by 4 grey circles. The numbers 8 and 4 are written above their respective groups of circles.

A visual representation of the multiplication 4 x 5 = 20. On the left, there is a 4x5 grid of blue circles. In the middle is a multiplication sign. To the right of the multiplication sign is an equals sign, followed by the number 20. The numbers 4 and 5 are written to the left and above the grid respectively.

A visual representation of the division 20 ÷ 2 = 10. On the left, there are 20 blue circles. In the middle is a division sign. To the right of the division sign is an equals sign, followed by 10 blue circles. The numbers 20 and 2 are written above and below the division sign respectively.

Some **binary operations** (like addition and multiplication) are **commutative** - which means it doesn't matter the order of the numbers. Other **binary operations** (like subtraction and division) are not **commutative** - which means the order of the numbers does matter.

FUNCTIONS ... but different ones from a 'reliability' perspective

We talked about how a **function** is what a product, system or service is intended to do in order to satisfy a **requirement**. But when it comes to mathematics, a **function** has a different meaning.

a FUNCTION relates an input to an output.

We often represent a **function** with a capital '*F*' and the **input** to this **function** with the lower case '*x*' to give ... $F(x)$. The reason we use '*x*' is to represent the fact that it is a **variable** that could take any number of possible values. We can define any **function** we want. For example:

$$f(x) = x + 2$$

This represents a **function** that takes any **input**, and **outputs** a number that is 2 higher than the **input**. For example:

$$f(0) = 0 + 2 = 2$$

$$f(1) = 1 + 2 = 3$$

$$f(2) = 2 + 2 = 4$$

We can use **arithmetic** to define any **function** we want. But when we do, there needs to be rules.

$$f(x) = 3 \times x + 2$$

... we ALWAYS multiply and divide
BEFORE we add or subtract

For example, with this new **function**:

$$f(2) = 3 \times 2 + 2$$

$$f(2) = 6 + 2$$

$$f(2) = 8$$

Let's say that we actually do want to ADD before we MULTIPLY. We can use brackets to represent which part of our **function** we want to evaluate first.

$$f(x) = 3 \times (x + 2)$$

... now we ADD first

For example, with this new **function**:

$$f(2) = 3 \times (2 + 2)$$

$$f(2) = 3 \times 4$$

$$f(2) = 12$$

As a convention, it is not uncommon to omit multiplication signs entirely. For example, a **function** with a multiplication sign is shown below on the left. The same **function** with the multiplication sign omitted is shown below on the right.

$$f(x) = 3 \times (x + 2)$$

$$f(x) = 3(x + 2)$$

INVERSE FUNCTIONS

As we know, a **function** relates an **input** to an **output**. An **inverse function** takes that **output** and relate it back to the **input**. Let's go back to this simple **function**:

$$f(x) = x + 2$$

Let's evaluate this **function** for value of $x = 2$, and allocate its **output** to variable y ($y = 4$).

$$f(2) = 2 + 2 = 4$$

The **inverse function** is denoted F^{-1} and essentially turns y back into x .

$$f^{-1}(y) = y - 2 = x$$

$$f^{-1}(4) = 4 - 2$$

$$f^{-1}(4) = 2$$

We use **inverse functions** a lot in **reliability engineering**.

NOTATION

NOTATION is a series or system of written symbols used to represent numbers, amounts, or elements in something such as music or mathematics

We need to use **notation** a lot in mathematics, statistics and engineering ... otherwise we will exhaust ourselves trying to write things out in full!

For example, let's look at a scenario where we need to add lots of numbers together.

$$1.34 + 2.58 + 4.59 + 10.09 + 7.01 + 8.13$$

We often use the uppercase Greek letter sigma (Σ) to represent the addition of lots of numbers. The expression above is equivalent to the expression below.

$$\Sigma (1.34, 2.58, 4.59, 10.09, 7.01, 8.13)$$

This makes things a *little* bit simpler. But we can go much further.

Let's say that we are studying some sort of metric which might have different values ... depending on when we look at it, or the environment around it. Perhaps we are examining the thickness of an oxide coating from unit to unit. Perhaps we are analyzing the number of units shipped this month. Perhaps we are examining the **Time to Failure (TTF)** of a number of prototypes, meaning that there will be a different **TTF** value for each prototype. The metric you measure can be anything that helps you make a **decision**.

Now let's represent this metric with ' x_i .' If we are (for example) examining **TTF**, then x_1 would be the **TTF** of the 1st prototype, x_2 would be the **TTF** of the 2nd prototype and so on.

Let's say that we want to add lots of these observed **TTFs** together.

$$x_1 + x_2 + x_3 + x_4 + x_5 + \dots$$

We can simplify this expression a little bit:

$$\sum (x_1, x_2, x_3, x_4, x_5, \dots)$$

But now we can *really* simplify it this expression using the following notation.

$$\sum_{\forall i} x_i$$

... noting that ' $\forall i$ ' is how we say that we are interested in *all possible values of i* . Which means we add every single **TTF** we have observed. But if we are only interested in adding a certain number of these variables, we can write:

$$\sum_{i=1}^n x_i$$

For example, if we want to add the first four **TTFs**, we can write:

$$\sum_{i=1}^4 x_i = x_1 + x_2 + x_3 + x_4$$

This **notation** makes it a lot simpler to represent the addition of lots of numbers. We can also incorporate this in a Microsoft Excel spreadsheet, using the following function:



= sum (A1 : A10) ... = sum (A1 , B2 , B3)

EXERCISE

What is the sum of

$$x_1 = 12.4, x_2 = 13.7, \text{ and } x_3 = 15.1?$$

Firstly, let's count how many numbers we are adding.

$$n =$$

Let's create the 'simplified' notation of this expression.

$$\sum_{i=1}^{\quad} \quad = \quad + \quad + \quad = \quad$$

We can also do a similar thing for multiplication. Let's say that we want to multiply lots of these observed **TTFs** together.

$$x_1 \times x_2 \times x_3 \times x_4 \times x_5 \times \dots$$

Instead of using the uppercase Greek letter sigma (Σ), we use the uppercase Greek letter pi (Π). We can now simplify this expression a little bit:

$$\prod (x_1, x_2, x_3, x_4, x_5, \dots)$$

We can again simplify expressions for multiplication expressions like this

$$\prod_{\forall i} x_i \quad \dots \quad \prod_{i=1}^n x_i$$

For example, if we want to multiply the first four **TTFs**, we can write:

$$\prod_{i=1}^4 x_i = x_1 \times x_2 \times x_3 \times x_4$$

The Microsoft Excel spreadsheet function for multiplication is:



= product (A1:A10) ...
= product (A1,B2,B3)

EXERCISE

What is the product of

$$x_1 = 1.5, x_2 = 3.8 \text{ and } x_3 = 2.7?$$

Firstly, let's count how many numbers we are multiplying.

$$n =$$

Let's create the 'simplified' notation of this expression.

$$\prod_{i=1}^{\quad} \quad = \quad + \quad + \quad = \quad$$

EXPONENTS

We can simplify some stuff even more! Let's just say we are multiplying the same number three times. We can write:

$$\prod_{i=1}^3 x = x \times x \times x$$

Another way of writing this is:

$$\prod_{i=1}^3 x = x \times x \times x = \dots \quad x^3$$

We can say that we have 'raised x to the power of 3.' We call '3' the **exponent**.

the EXPONENT refers to the number of times a number is multiplied by itself.

We use the **exponent** a lot in **reliability engineering** as well.

RADICAL or RADIX

An **exponential function** looks something like:

$$f(x) = x^k$$

The **inverse exponential function** involves a symbol called a **radical** or **radix** ($\sqrt{\quad}$).

$$f^{-1}(y) = \sqrt[k]{y}$$

For example, we understand how we multiply 4 by itself once.

$$\begin{array}{ccc} & 4 & \\ 4 & \begin{array}{|c|c|c|c|} \hline \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \hline \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \hline \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \hline \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \hline \end{array} & \begin{array}{l} 4 \times 4 \\ = 4^2 \\ = 16 \end{array} \end{array}$$

So using **exponential notation**:

$$4^2 = 16$$

And going the other way:

$$4 = \sqrt{16}$$

... noting that if the **exponent** is 2, we omit it from the **radix** or **radical sign**. This is what we refer to as the **square root**.

$$x = \sqrt{x^2}$$

ALGEBRA

Algebra is a LOT like **arithmetic**. The key difference is that **algebra** involves at least one 'unknown' number that we represent with a symbol.

$$\text{arithmetic ... } 2 + 3 = 5$$

$$\text{algebra ... } 2 + 3 = x$$

With our second expression, we can 'solve' one side to reveal that the value of our 'unknown' number is 5. But in practice, **algebra** can be a little more complicated than this.

Let's look at the following expression we call an **equation**.

$$3 \times x = 12 \times (x - 3)$$

The reason we call this an **equation** is because it involves an 'equals sign.' This means the part of the expression on one side is the same as the part of the expression on the other. This also means that anything we do to one side of the expression, we have to do to the other.

So we go through steps to simplify this **equation** to find the one and only value of 'x' that makes this expression hold true.

Step #1 ... divide both sides by 12

$$3 \times x = 12 \times (x - 3) \quad \rightarrow \quad 0.25 \times x = x - 3$$

Step #2 ... subtract $0.25 \times x$ from both sides

$$0.25 \times x = x - 3 \quad \rightarrow \quad 0 = (0.75 \times x) - 3$$

Step #3 ... add 3 to both sides

$$0 = (0.75 \times x) - 3 \quad \rightarrow \quad 3 = 0.75 \times x$$

Step #4 ... divide both sides by 0.75

$$3 = 0.75 \times x \quad \rightarrow \quad 4 = x$$

MATRICES

And we need to know what a **matrix** is as well. **Matrices** is the plural of matrix.

a MATRIX is a rectangular array of numbers.

We usually represent a **matrix** with square brackets. We can add any two **matrices** together ... as long as they have the same number of columns and rows. Each corresponding element of the **matrix** is added together.

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

We can also multiply **matrices** ... in a really *weird* way. When we multiply two **matrices**, we look at each row in the first **matrix** and each column in the second. We multiply the first elements of each row and column, multiply the second elements of each row and column, and keep going until they have all been multiplied. We then add them together to get the corresponding element in our output matrix

$$\begin{bmatrix} 1 & 3 \\ -2 & 4 \end{bmatrix} \times \begin{bmatrix} 4 & -3 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ -4 & 14 \end{bmatrix}$$

(1 × 4) + (3 × 1) = 7

We keep going until we have found each element in our output matrix.

$$\begin{bmatrix} 1 & 3 \\ -2 & 4 \end{bmatrix} \times \begin{bmatrix} 4 & -3 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ -4 & 14 \end{bmatrix}$$

(1 × -3) + (3 × 2) = 3

$$\begin{bmatrix} 1 & 3 \\ -2 & 4 \end{bmatrix} \times \begin{bmatrix} 4 & -3 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ -4 & 14 \end{bmatrix}$$

(-2 × 4) + (4 × 1) = -4

$$\begin{bmatrix} 1 & 3 \\ -2 & 4 \end{bmatrix} \times \begin{bmatrix} 4 & -3 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ -4 & 14 \end{bmatrix}$$

(-2 × -3) + (4 × 2) = 14

FACTORIALS, COMBINATIONS and PERMUTATIONS

Sometimes we need to be able to talk about how we can combine and order certain things. And this often comes with some sort of **equation**. Let's just say we want to multiply a whole bunch of whole numbers or integers, starting at 1 and going up to a number you choose (let's call this '*n*').

$$1 \times 2 \times 3 \times \dots \times n = \prod_{i=1}^n i = n!$$

We call this a **factorial**, and we represent it with the symbol '!'

a FACTORIAL is the product of all integers less than or equal to a number.

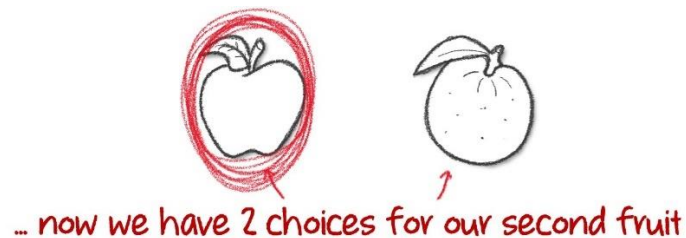
Factorials are really useful when we want to look at **combinations** of things. Let's say we have three pieces of fruit: an apple, a banana and an orange.



How many ways can you 'select' these three pieces of fruit if they are all in the same fruit bowl and you pick one at a time? Well, for our first selection, we have three options. Let's say we choose the banana.



Now for our second choice. We now only have two options. Let's say we go with the apple.



And for our third and final choice ...

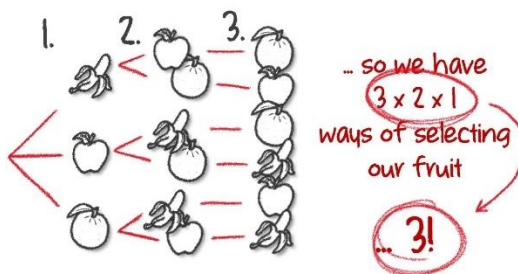


So this is the sequence of fruit we selected:



So how many different ways *could* we have selected this fruit?

Well, we know that we have three choices for our first fruit, two choices for our second fruit, and one choice for our third fruit.



Each way of selecting fruit is called a **permutation**. A **permutation** includes an order or sequence. But what happens if we have 10 pieces of fruit, and only select 3 ... how do we calculate the total number of **permutations**? Well ... we use the following **equation** that uses **factorials**:

$$n \times (n - 1) \times (n - 2) \times \dots \times (n - k) = \frac{n!}{(n - k)!} = P_k^n$$

where P_k^n is the total number of possible **permutations** of k distinct objects from a total of n distinct objects.

If the order or sequence of things doesn't matter, then we are dealing with a **combination**. For example, we have identified six **permutations** of fruit above when we select three pieces of fruit from a total of three. But there is only one **combination**, because for **combinations** the order doesn't matter.



And how do we calculate the total number of possible **combinations**? We use the following equation:

$$\frac{n!}{(n - k)! k!} = \binom{n}{k} = C_k^n$$

where $\binom{n}{k}$ or C_k^n is the total number of possible **combinations** of k distinct objects from a total of n distinct objects.

Basic Probability and Stuff

Human beings are really bad at dealing with uncertainty and randomness. We all have different risk profiles, biases, emotions, gambling debts and so on. We 'gamble' on our life every day. And it can sometimes be about perspective.

For example, Russian Roulette is a very dangerous game where a single round is loaded into the cylinder of a revolver and then spun (so that the position of the round is effectively random). The 'player' then shoots in the direction of him or herself. There is a one in six chance of nothing bad happening. So the following is true ...

... for each 'turn' you play in Russian Roulette, you will probably be FINE!

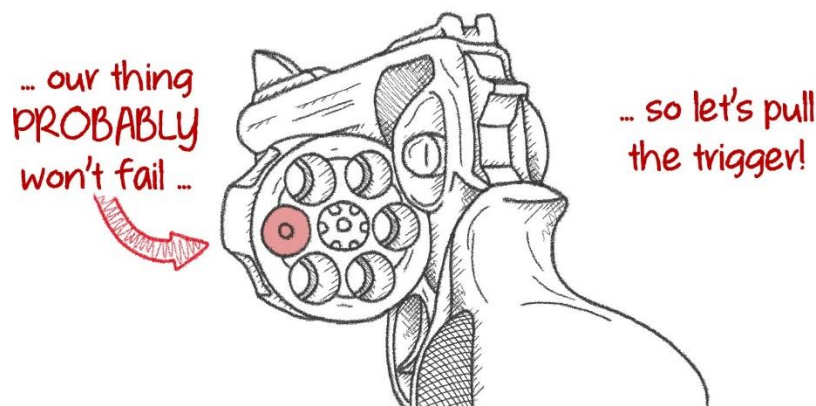
However, playing Russian Roulette is a very, very, very, very bad idea. Even if for each turn, you will 'probably' be fine. But how many times have we been advised by someone that it is OK to do something, because ...

... you will probably be FINE!

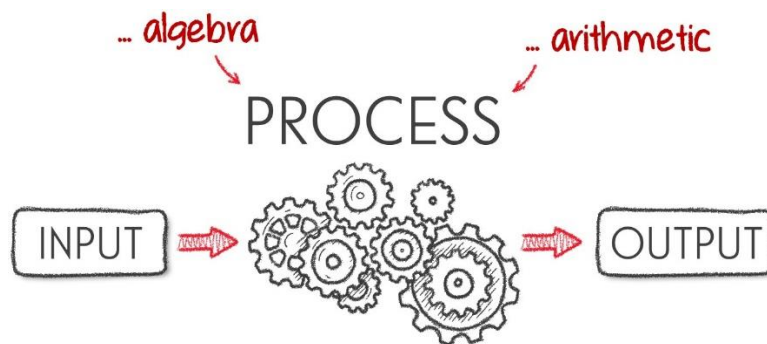
Reliability is a measure of **probability**. And we can delude ourselves that **reliability** is higher than it actually is. Why? Because everything we make will *probably be FINE*. If we manufacture a new product, and 70 % of them manage to be functional at the end of the warranty period, then the following is true:

... each of our product will PROBABLY survive the warranty period.

But if you need to effect **warranty action** on 30 % of your products, you will lose a lot of money and potentially go bankrupt.



As a rule, humans work much better with non-random processes. This means that for every **input**, we like it when there is only one possible **output**. Which is how **arithmetic** and **algebra** work.



But **random processes** can have multiple different **outputs** for the same **input**.



This infuriates us sometimes ... especially if you are an engineer or someone with a technical mind. But **reliability engineers** need to embrace this reality. And we need to remember that **just because something is random, doesn't mean it is not predictable**. And by extension, we can observe a **random process** and deduce what is affecting our **random process**. Which means we have the power to control that **random process**.

And the **random process** we are most interested in is **failure**.

when we talk RANDOM PROCESSES, we talk about EVENTS

an EVENT is a set of outcomes of a process to which a PROBABILITY is assigned.

You can define any **event** you want. Turning on your computer and it working is an **event**. Rain at 2 pm in the afternoon is an **event**. Your smart lock **failing** is an **event**. The consequence of that **failure** is an **event**. We define **events** based on whatever it is we are trying to get a better understanding of.

PROBABILITY is a measure of the likeliness or likelihood that an EVENT will occur

And this is where that randomness comes into play. If we toss a coin, then spotting a 'heads' is an **event**. The **probability** of this event **occurring** is one half, 0.5 or 50 % (whatever works for you).

Remember in our [Arithmetic and other math stuff lesson](#) where we talked about [notation](#)? Do you remember why we like **notation**? It is because it can simplify the stuff we write.

For example, we might simply state that ...

event A is the event of spotting a head when I toss or flip a coin.

So instead of having to say or write 'the event of spotting a head when I toss or flip a coin' every time, we only have to say or write 'event A.' And this **event** (like all others) can be assigned a **probability**.

The **notation** for the **probability** that event A occurring is ...

$\Pr(A)$... and sometimes just $P(A)$

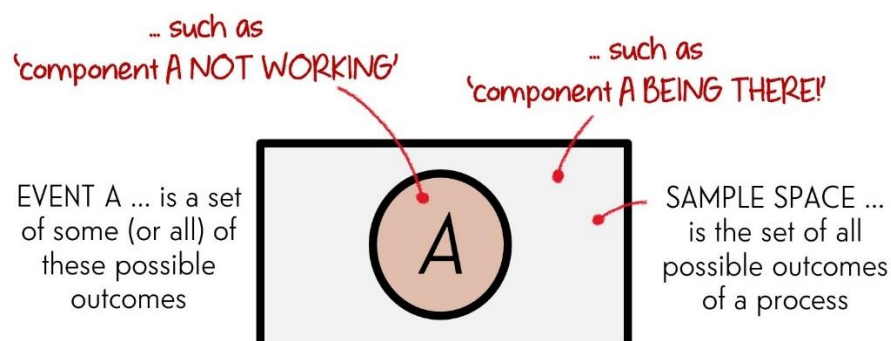
So if $\Pr(A) = 1$ it means that event A will CERTAINLY occur. There is no chance of it not occurring. And $\Pr(A) = 0$ means that event A will NEVER occur.

Reliability engineers are all about the **event** we call '**success**.' **Success** is the opposite of **failure**. So if we define event S as any **event** where our thing has not **failed** ... we have essentially defined **reliability**!

Reliability, $R = \Pr(S)$

VENN DIAGRAMS

Remember that humans are really bad at understanding random processes, **probability**, and making 'good' decisions in regard to them? **Venn diagrams** are useful ways of visualizing **events** to help make it (a little bit) easier for us. A basic example of a **Venn diagram** is illustrated below.



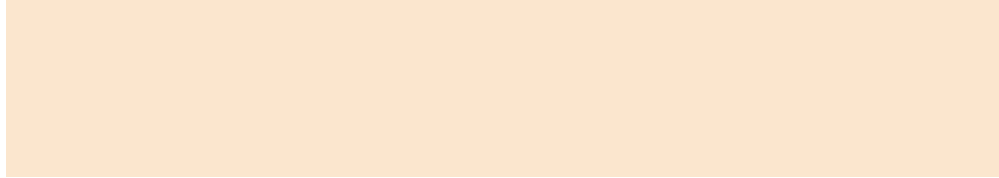
And remember, event A can be anything you define it to be.

EXERCISE



Let's say we are looking at the
PROCESS of rolling a dice.

What is the SAMPLE SPACE?



Let's make this a little easier.

Let's use the uppercase letter X to replace the phrase ...

... *the number I roll on my dice.*

One way of helping us define the sample space is using this symbol:

\in

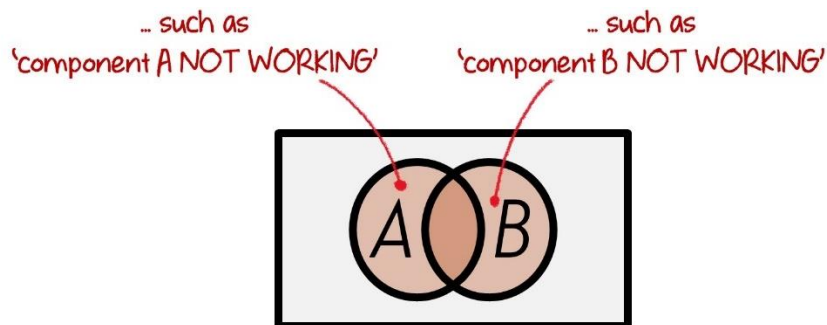
This symbol literally means 'is an element of'.

So define the SAMPLE SPACE of X using the following notation:

$\{X \in \square, \square, \square, \square, \square, \square\}$

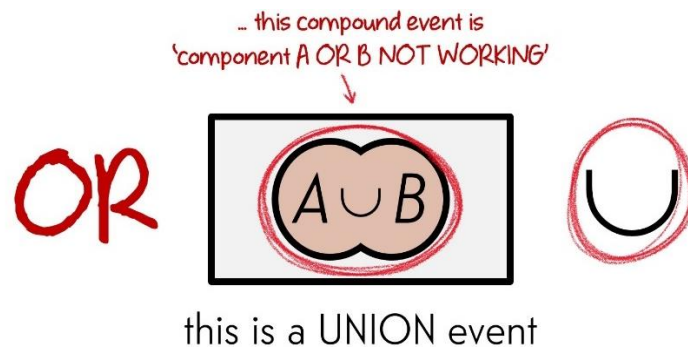
COMPOUND EVENTS (... or COMPOSITIONS)

Random processes can involve more than one **event**. **Venn diagrams** can represent them too.

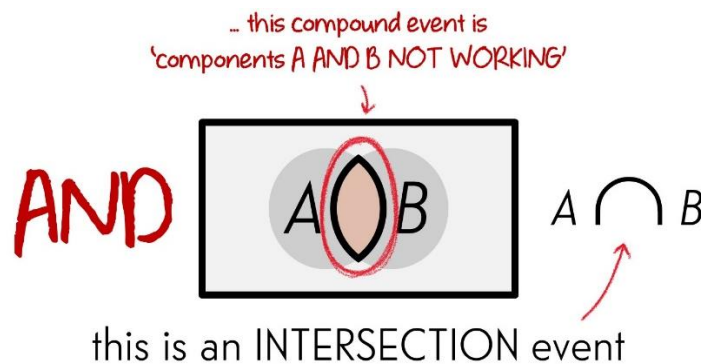


You can see that (according to our **Venn diagram**) it is possible for just event A to occur, just event B to occur, BOTH events A and B to occur, or NEITHER events A nor B occurring.

We can create a whole new **event** based on these two **events**. And any **event** that is based on other **events** is what we call a **compound event** or **composition**.



But there is another type of **compound event** as well.



And there is another type of **event** we should mention here. It is not a **compound event**, as it is based on just one **event**.



We use **compound events** a lot in **reliability engineering**. For example, if we have a **parallel (redundant) system** of two pumps, then our system **fails** if pump *A* AND pump *B* fails (**intersection**). If we have a **series system**, then our system **fails** if pump *A* OR pump *B* fails (**union**). And of course, **success** is NOT **failure** (**complement**).

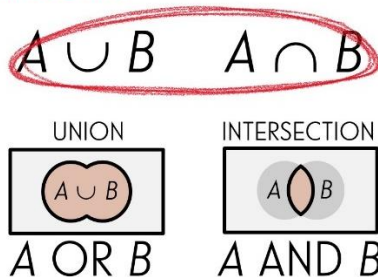
EVENTS have BINARY OPERATIONS as well

We talked about **binary operations** in our [Arithmetic and other math stuff](#) lesson.

a *BINARY OPERATION* a rule for combining two values to create a new value.

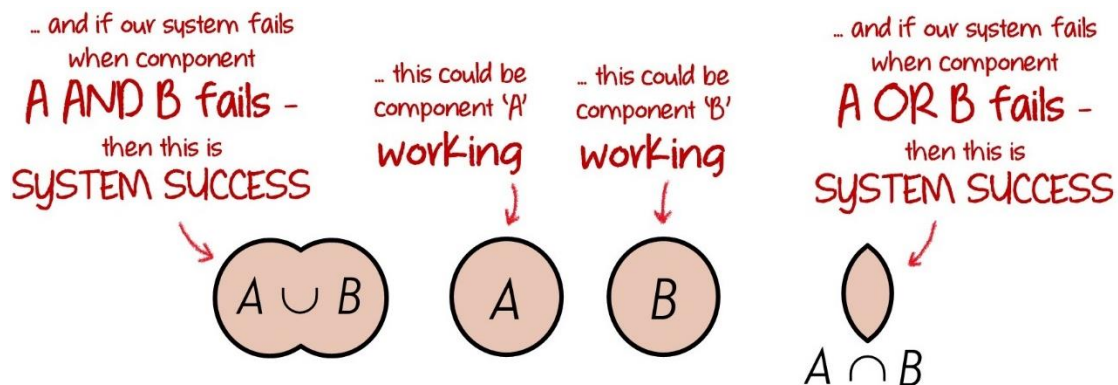
And we have just looked at some **binary operations** as they apply to **events**.

... so these are BINARY OPERATIONS



And as we talked about in the previous section, these **compound events** or **compositions** that we create using **binary operations** are very important for **reliability engineering**.

Let's take look at **reliability** from the perspective of **success events**. Let's say we have a system with two components (A and B).



The **compound event** or **composition** on the left represents a **parallel system**, and you can see that because it is 'bigger' it represents the fact that **reliability** increases when we have **redundancy**. Conversely, the **compound event** or **composition** on the right represents a **series system** and is correspondingly smaller.

If we focus on the size of each **event**, which is analogous to **probability**, we can see that

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$$

EXERCISE



Let's go back to our dice ...

We are going to define 'event A ' as an event where our dice rolls a 1. We are also going to define 'event B ' as an event where our dice rolls a 2.

Complete the following table.

EVENT A ...	ROLLING 1
EVENT B ...	ROLLING 2
$\Pr(A)$	
\bar{A}	
$\Pr(\bar{A})$	
$\Pr(A \cup B)$	
$\Pr(A \cap B)$	

Now complete the following table with the new definitions of events A and B .

EVENT A ...	ROLLING 3
EVENT B ...	ROLLING AN ODD NUMBER
$\Pr(A)$	
\bar{A}	
$\Pr(\bar{A})$	
$\Pr(A \cup B)$	
$\Pr(A \cap B)$	

then we have **MUTUALLY EXCLUSIVE** or **DISJOINT SETS**

So what is a **set**?

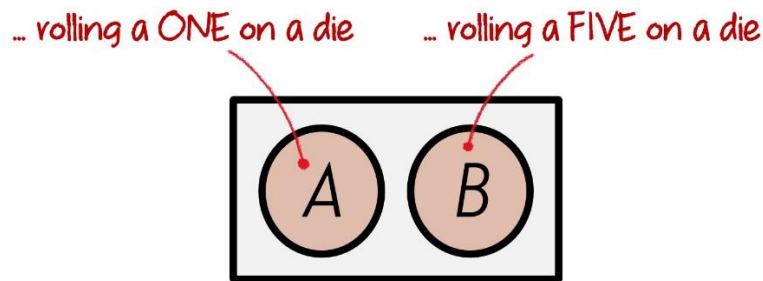
a SET is a collection of elements.

When we are talking about **events**, then the term **set** refers to a **collection of events**. And the **set** can also be an **event** (remember ... we can define pretty much anything happening as an **event**.)

In the exercise above, we defined event *B* as the event of 'rolling an odd number.' This is also a **set** of the following three **events**: rolling a 1, rolling a 3 and rolling a 5.

MUTUALLY EXCLUSIVE or DISJOINT SETS are sets that do not have any common elements.

An example of two **mutually exclusive** or **disjoint sets** represented on a **Venn diagram** is illustrated below.



MUTUALLY EXCLUSIVE or **DISJOINT SETS**

You can see that neither **event** overlaps in the **Venn diagram**. So what does that say about the **intersection event**? We can write that:

$$A \cap B = \emptyset$$

The symbol ' \emptyset ' is described in a couple of different ways by textbooks (... and **ponderous professors**). Most of the time, it really doesn't matter when it comes to **reliability engineering**.

Some textbooks will say that \emptyset is an **EMPTY set** that contains NO ELEMENTS. This makes intuitive sense to us. But some other textbooks will say that \emptyset is a **NULL set** whose elements are negligibly small.

Either way, the intersection of two **MUTUALLY EXCLUSIVE events** is ' \emptyset ' ... or whatever that means.



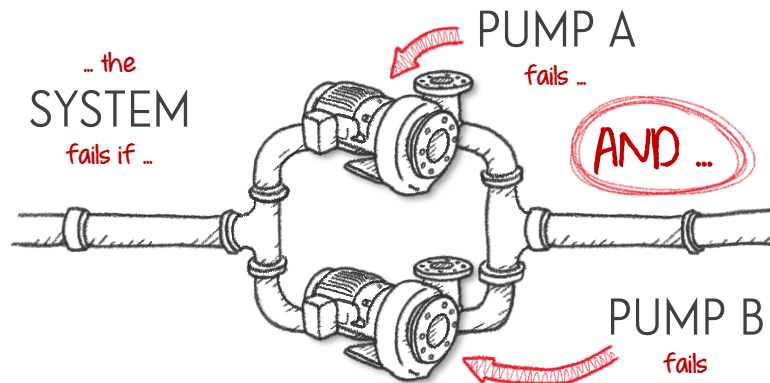
STATISTICALLY INDEPENDENT EVENTS

In **reliability engineering**, we like to assume that **events** are **statistically independent**.

STATISTICALLY INDEPENDENT EVENTS are events where the occurrence of one does not affect the occurrence of the other.

The reason we like **statistically independent events** is that it makes the mathematics easier! So what do **statistically independent events** mean for **reliability engineers**? It starts with **failure**. **Failures** are **statistically independent** if (1) the **failure** of one item doesn't influence the **failure** of another and (2) the **failures** of multiple items are not influenced by the same factors (like humidity, temperature, or the same incompetent maintenance crew).

So let's go back to why we like **statistically independent events**. Think about a two-component **parallel system**.



If events A and B are the **failures** of pump A and B respectively ... AND they are **statistically independent**, then we can write ...

$$\Pr(A \cap B) = \Pr(A) \times \Pr(B)$$

Which means we can also write ...

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A) \times \Pr(B)$$

In our [Fault tolerance](#) lesson, we talked about **common cause failure**. And unfortunately for customers, users and **reliability engineers**, this means that a lot of **failures** are not **statistically independent**. And when that is the case, we can't use the equations above. We instead need to individually investigate our system, analyse **common cause failure**, and create a more tailored model.

This is not a huge problem for **series systems**. But it is a significant concern for **parallel systems** where we want **redundant** components to be functional after a primary component fails.

Experience has shown that around 10 % of failures have a common cause.

EXERCISE



Let's go back to our dice ...

We are going to define 'event A ' as an event where our dice rolls a 1. We are also going to define 'event B ' as an event where our dice rolls a 2 **IN THE NEXT THROW**.

Complete the following table.

EVENT A ...	ROLLING 1
EVENT B ...	ROLLING 2 IN THE NEXT THROW
Are these events mutually exclusive?	
Are these events statistically independent?	

Complete the following table based on the new definitions for events A and B .

EVENT A ...	ROLLING 4
EVENT B ...	ROLLING AN ODD NUMBER
Are these events mutually exclusive?	
Are these events statistically independent?	

... and there is thing called RARE EVENT APPROXIMATION

We are always looking for shortcuts. And sometimes that extends to **reliability engineers!**

Think of any **event** that we call 'event *A*.' It might be a tornado. It might be a tsunami. It might be a meteor strike. If the **probability** of event *A* occurring is really small, then event *A* is what we call a **rare event**.

Now think of another **event**, that we call 'event *B*.' It might be a one in 100-year flood. It might be an earthquake. It might be a doctor's appointment actually being on time. If the **probability** of event *B* occurring is really small, then event *B* is also a **rare event**.

If the **probabilities** of events *A* and *B* are really small, then the **probabilities** of event *A* AND event *B* occurring (at least at the same time) is really, really small. This could be (for example) an **event** where our electricity generation plant experiences a tornado on the SAME DAY it is struck by an earthquake. This is beyond unlikely.

Another way of writing this is ...

$$\Pr(A \cap B) \rightarrow 0 \text{ ... if events } A \text{ and } B \text{ are } \mathbf{rare \text{ events}}.$$

This can really simplify some of our **equations** above. Remember that the **probability** of a **union** of events *A* and *B* is ...

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$$

$$\Pr(A \cup B) \approx \Pr(A) + \Pr(B) \text{ ... if events } A \text{ and } B \text{ are } \mathbf{rare \text{ events}}.$$

If events *A* and *B* refer to the **failure** of components *A* and *B* respectively, AND we believe that events *A* and *B* are **rare events**, then we can use this approximation of system **failure probability** for **parallel systems**. One 'good' thing about **rare event approximation** is that we always underestimate system **reliability**. This means that we don't inadvertently believe our system is more **reliable** than it actually is.

Rare event approximation is sometimes used when we model really complex systems that have thousands of components. Even computers can have trouble accurately calculating **failure probabilities** because there are so many times it needs to multiply component **failure probabilities**. But ... in most applications you don't have to (nor should you) do this!

... then there is **CONDITIONAL PROBABILITY**

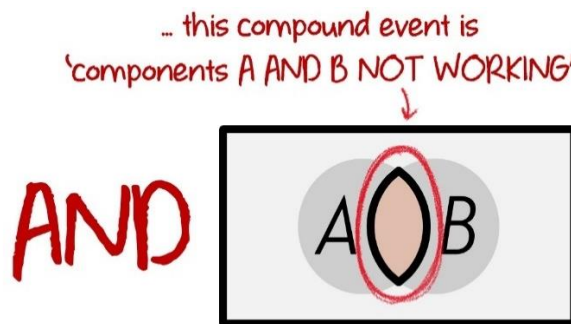
CONDITIONAL PROBABILITY is the probability of an event occurring given another event has occurred

Because **common cause failure** is (unfortunately) everywhere, we need to talk about **conditional probability**. And this is how we write the **conditional probability** of an event ...

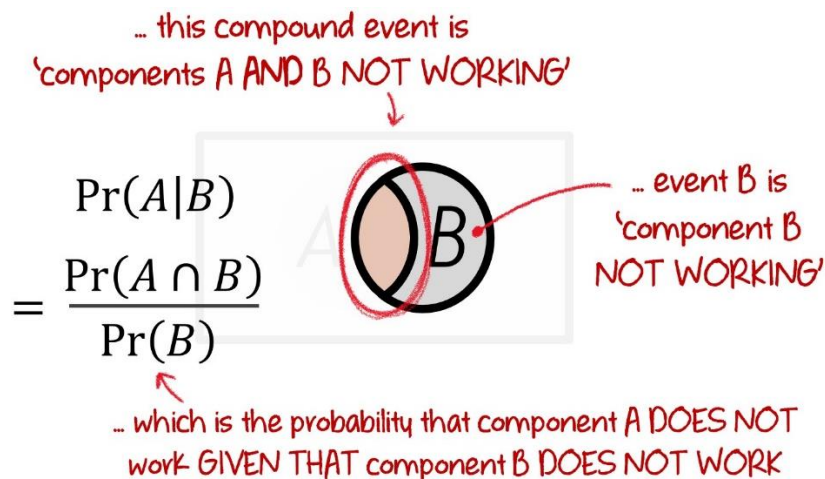
the **CONDITIONAL PROBABILITY** of ... $\Pr(A|B)$

... of event A occurring ... given event B occurred

Let's go back to our **Venn diagram** that illustrates the **intersection** of **two events**, $\Pr(A \cap B)$.



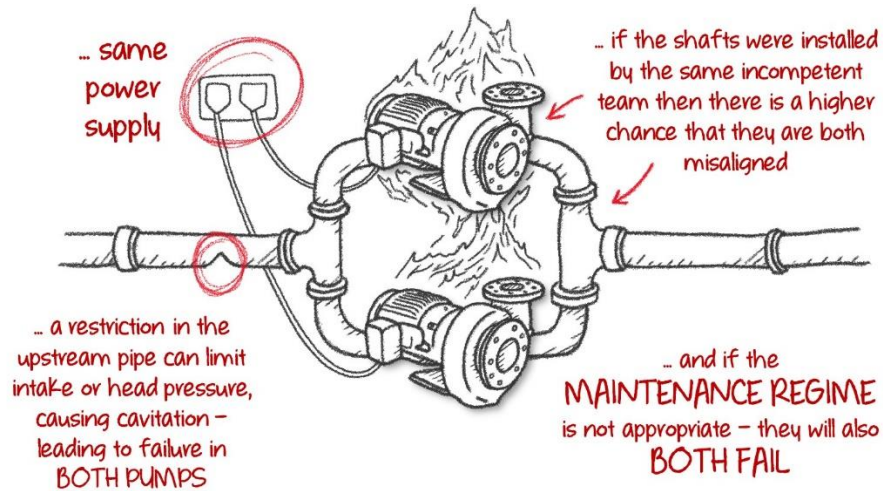
Another way of looking at this is from the perspective of **conditional probability**.



If events A and B are **statistically independent**, then event B occurring has no influence on event A occurring. So the **probability** of event A occurring won't change if event B occurs or doesn't occur. In other words ...

$\Pr(A|B) = \Pr(A)$... if events A and B are **statistically independent**

But things are different for **common cause failures**.



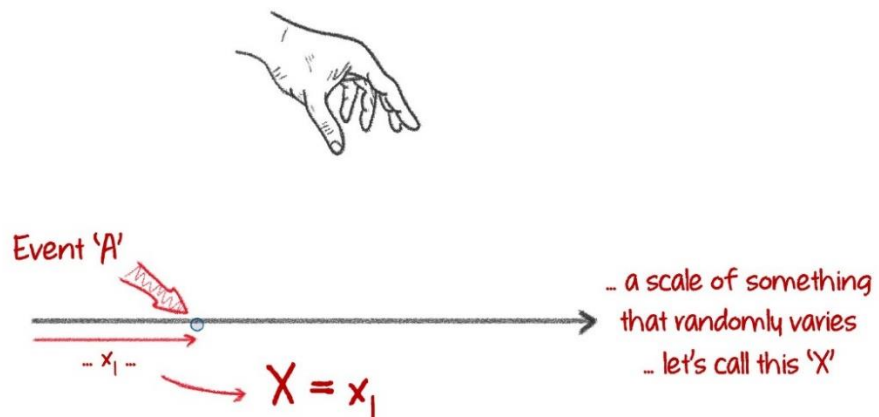
In this scenario, $\Pr(A|B)$ is dependent on whatever it is that drives **common cause failure**. And you will need to investigate it if you believe this could be an issue for your system.

... so let's simplify further

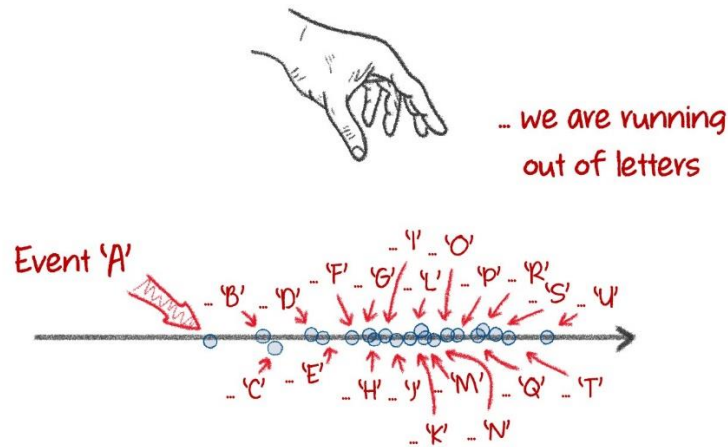
We are now going to introduce something we call the 'random hand of failure.' This hand simply represents the **random process** we are looking at. Because we are looking at the process of **failure**, the hand represents material variations, different ways our system is used, changing environmental metrics and everything else that will control how likely it is for our system to **fail**.

Let's use the upper-case letter ' X ' to represent the **random variable** that our random hand of failure generates. X could be **Time to Failure (TTF)**, cycles to failure, number of demands and so on.

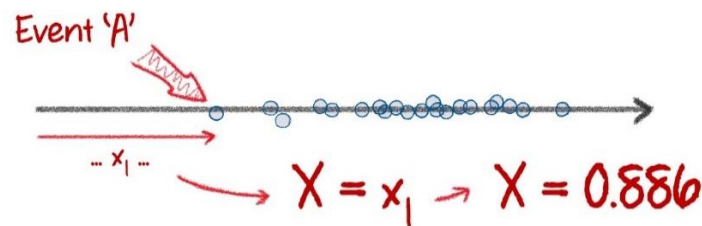
Let's say that the first **random variable** our 'random hand of failure' throws has a particular value, x_1 . Up until now, we have been defining this as an **event**. Which have been represented with letters. So we could say that event A is the event where our random hand of failure generates a random variable with a value x_1 .



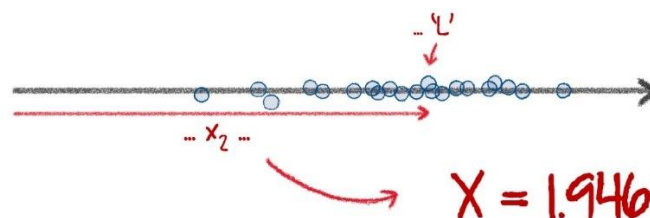
But using letters is problematic. Let's say our 'random hand of failure' generates 21 **random variable** values ... or **failure events**.



There is a much simpler way! Let's look at event A again, where our **random variable** has a value of x_1 , or (in this example) 0.886.



And we can do this for any one of our **failure events**. For example, event L is ...



So event A is simply the **event** where $X = 0.886$. And event L is simply the **event** where $X = 1.946$. So we can write **probability** as follows ...

$$\Pr(A) = \Pr(X = 0.886)$$

$$\Pr(L) = \Pr(X = 1.946)$$

So in general, **reliability engineers** (and lots of other people who need to do **probability** and statistics) often write **probability** of random outcomes generally as follows ...

$$\Pr(X = x)$$

... what this 'random variable' is ...
like 'time to failure' or 'number
of failures in one year'

... what the value of
the random variable is
for this event

For example, if we define X to be the (random) number we roll on a dice, and we know that there is a one in six chance that we roll a one, we can write ...

$$\Pr(X = 1) = \frac{1}{6}$$

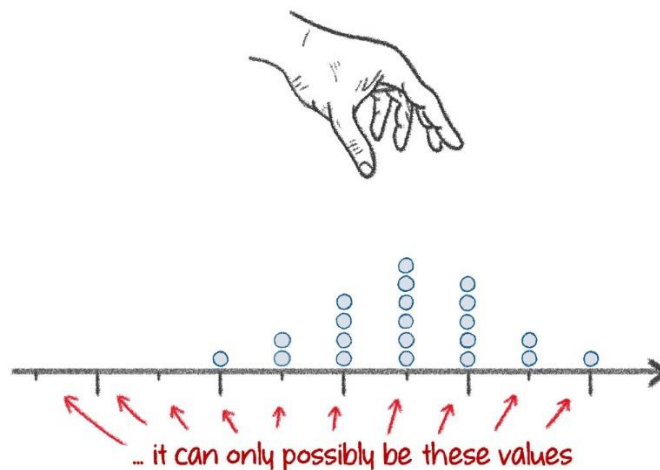
... probably need to start talking about RANDOM VARIABLES

a RANDOM (or STOCHASTIC) VARIABLE is a variable whose value is subject to a random process or chance.

Random variables are all around us. We know plenty of **random variables**. Because we are talking about **reliability**, we need to be really familiar with **random variables**. And there are different types of them.

*a **DISCRETE** RANDOM VARIABLE is a variable whose value is subject to a random process or chance **AND** is one of a set of specified values.*

The number we roll on a dice is an example of a **discrete random variable**, as its possible values are limited. Other examples include the number of **failures** we observe during a test or in an interval of usage, the number of defective ball bearings in a batch, or the number of missiles that correctly impact a target when 10 are launched.



And there is another type of random variable.

a **CONTINUOUS** RANDOM VARIABLE is a variable whose value is subject to a random process or chance **AND** can occur over a number of intervals.

An example of a **continuous random variable** is **Time to Failure (TTF)**, noting that this can be expressed in terms of operating hours, cycles, distance, or anything else that drives the **VITAL FEW failure mechanisms** of your system.

