Content

Basic Terminologies

1. Experiment

- It is basically an activity which I'm trying to do.

> **Let's say I have this mathematical equation**

$a^2 + b^2 + 2ab$

where: a = 3 and b = 4

$3^2 + 4^2 + 2(3)(4)$ = 49

- We are 100% sure that the result of this equation will be 49 only. It cannot be 50 or 48.

This type of experiment is called **Deterministic Experiments** where we can **determine** the exact output, like in this case.

Here are few more examples:

1. Flipping a coin
   - When we flip a coin, there are two possible outcomes: it can land `heads` or `tails`.
   - The outcome can be either of these, each time we perform the experiment.
2. Rolling a six-sided die
   - When we roll the die, the outcome is uncertain, and there are many possible results.
   - The die can land on any of the six faces.
3. Cricket Match
   - Suppose there is a match going on between 2 teams and we don't know what can be the result.
   - Either your team can lose or win. So the outcome is uncertain.

In all of these above examples, we can notice one common thing.

> **Q. Can we determine the outcome of all these experiments?**

No.

Because the outcomes are uncertain. These types of experiments are known as **Probabilistic Experiments**.

Let's continue with the experiment of "Rolling a six sided die" and look at the possible results of an experiment.

Experiment: Rolling a die

2. Outcomes

- Suppose we roll a six sided die and we want to know the possible `Outcomes`.
- We know that we could get any digit out of the 6 digits. So, an outcome could be : {1} or {2} or {3} or {4} or {5} or {6}

3. Sample Space

- It is the collection of all the possible outcomes of the experiment.

So the **sample space** for this experiment will be: **{1, 2, 3, 4, 5, 6}**

4. Events

We know that sample space for die is {1,2,3,4,5,6}.

If we say,

> **An Even number is rolled / While rolling a die, an even number has occurred**

- Then the possible outcomes will be: `{2, 4, 6}`

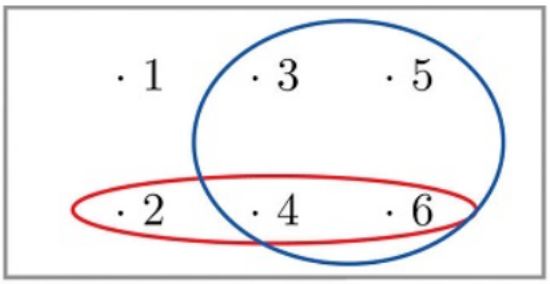
This is known as an **`Event`**.
Any subset of sample space is an event.

- {2, 4, 6} is a subset of sample space.

"**An Even number is rolled**" is an event here and its output is E = {2, 4, 6}, where E denotes an Event.

> **Q1. What are the possible outcomes when a dice is rolled and a number greater than two has occurred?**

- For this Event, outcome will be E = {3, 4, 5, 6}



Here is a graphical representation of a sample space and events

- Here the **sample space** S is represented by a rectangle which is {1, 2, 3, 4, 5, 6}
- **Outcomes** are represented as points within the rectangle which is {1}, {2}, {3}, {4}, {5}, {6}
- **Events** are represented as ovals that enclose the outcomes that compose them.
    - we have two events, $E1$: {2, 4, 6} which is an event for "Even number is rolled"
    - $E2$: {3, 4, 5, 6} which is an event for "A number greater than 2 rolled"

Now let's see few experiments.

Experiment 1: Tossing a single coin

> **Q1. If we toss a single coin then what can be the Possible Outcomes for this experiment?**

- Either we can get **Heads**
- Or we can get **Tails**

Therefore, our outcome becomes: **{H}, {T}**

The **Sample Space** for this experiment will be S **= {H, T}**

> **Based on this sample space, what possible Events can be defined?**

Getting Heads while tossing a coin,

- then our event will be E = {H}

Getting Tails while tossing a coin,

- then our event will be E = {T}

> **Q2. Suppose the given subset is itself {H,T}. Can we define this as an Event or not?**

Yes, It is an event.

- We discussed earlier that any subset of a Sample Space is an Event.
- Also an entire set is a subset of itself so this is a valid event.

> **Q3. So how can we frame this event?**

It is the "**Event of getting Either Heads or Tails**".

> **Q4. Consider the empty set as the given subset denoted by { }. Is it a valid event?**

- We know that, an empty set is a subset of every set. An empty set is therefore a subset of sample space
- It is a valid subset
- So by going with the definition of an Event, we can conclude that this is a valid event.

This can be represented as the "**Event of getting neither Heads nor Tails**".

> **Q5. Is it possible if we toss a coin and get nothing?**

No, it is not possible.

- Therefore, we will have an **Empty set** here

- As we know an empty set is a subset of sample space, therefore it is an Event.

But, the probability of getting a Null Set (No outcome) is Zero.

As it is not possible to toss the coin and don't get any output. we will either gets a head or a tail.

> **Q6. How many subsets can be formed from the sample space?**

There is one formula to find the number of subsets : 2^N

- where N = number of elements in sample space

For the above experiment, number of elements in the sample sapace is 2 {H,T}, So N = 2

- Therefore the number of subsets will be $2^2 = 4$
- Subsets will be **{ {H}, {T}, {H,T}, { } }**

From this, we can conclude that an empty set is also considered as a valid subset.

Experiment 2: Tossing 2 coins simultaneously

> **Q1. If we toss 2 coins simultaneously then what can be the Possible Outcomes for this experiment?**

Explanation :

- We can get **Heads** and **Heads**
- We can get **Heads** and **Tails**
- We can get **Tails** and **Heads**
- And lastly, we can get **Tails** and **Tails**

Therefore the unique possible **outcomes** will be:

- {HH}, {HT}, {TH}, {TT}

Now, the collection of all the outcomes will be the **Sample Space** :

- {HH, HT, TH, TT}

> **Again, based on this sample space, what are the possible Events that can be defined?**

Let's define few events:

The Event of getting either one head:
means one of the outcomes from the 2 tosses should be Head

- {HH, HT, TH}
    - Getting both heads is also valid, getting heads on first coin is also valid and getting heads on the second coin is also valid.

> **Q2. Can we define the above event in a different ways such that the outcome of the event remains the same?**

We can define the above event in 2 other ways:

- **Event of getting at most one tail**
  which means maximum of one Tail only
    - {HH, HT, TH} ,this is the valid outcome

- **Event of getting at least one head**

  which means minimum of one head and maximun 2 as we have 2 coins
    - in this case also {HH, HT, TH} this is the valid outcome

Therefore we can define the above event in 3 different ways

Another event:

Event of getting the same outcomes in both the coins

- {HH, TT} is the valid outcome as either we get both heads or both tails

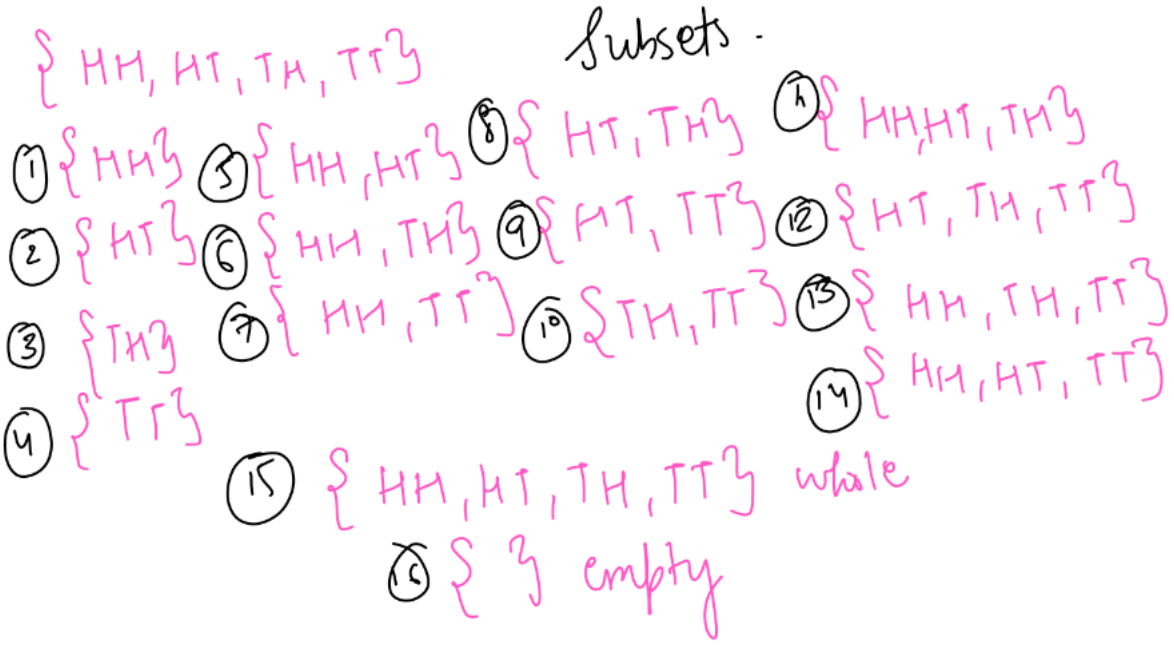
These are some few events, we can define many more.

> **Q3. How many subsets can be formed for this experiment?**

As we know

subset = 2^N, here $N = 4$

- then number of subsets will be $2^4 = 16$



Set Operations

Let's recall the Experiment which we defined above i.e. "Rolling a die"

For this experiment we are aware that,

Sample space is $\{1, 2, 3, 4, 5, 6\}$

- We can also represent this as a **Universe** or **Universal Set**
- Universal set is the collection of all possible sets

Now let's define some events:

- Mohit bets that he will get an odd number
  - So the output of this **Event** will be $A = \{1, 3, 5\}$
- Rakesh bets that he will get either 1, 5 OR 6
  - So the output of this **Event** will be $B = \{1, 5, 6\}$
- Abhishek bets that he will get an Even number
  - So the output of this **Event** will be $C = \{2, 4, 6\}$

Let's discuss few questions with respect to the above events.

Intersection

> **Q1. In which condition, both Mohit and Rakesh will win their bets?**

They will win their bets when we get a number 1 or 5 on a die.

- If we get number 3, then only Mohit will win his bet as 3 only occurs in his Event.
- And if we get number 6, then only Rakesh will win his bet as 6 only occurs in his event.

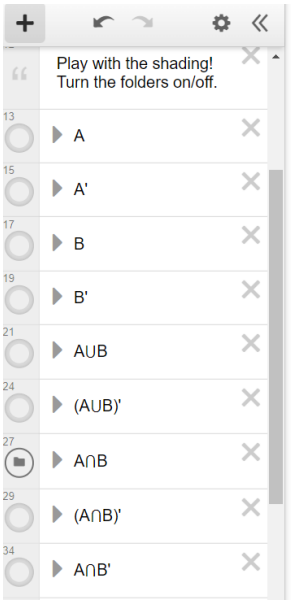
We want a number which occurs in both of their events

- Therefore $\{1, 5\}$ is the possible outcome such that both Mohit and Rakesh will win their bets

This is known as an **Intersection** of two events.

- It is denoted as $A \cap B$
- Intersection means **members belonging to both A `AND` B**
    - If we perform intersection on Events A and B then we will get only those elements that are present in both the events

      i.e. $A \cap B = \{1, 5\}$



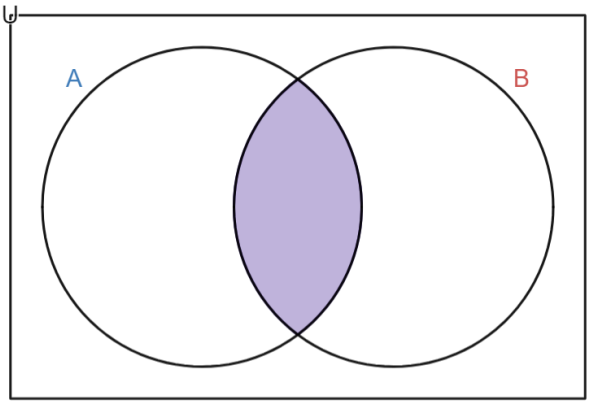


***Image source:** https://www.desmos.com/calculator/nynlqmtuu2*

Union

Now the next question,

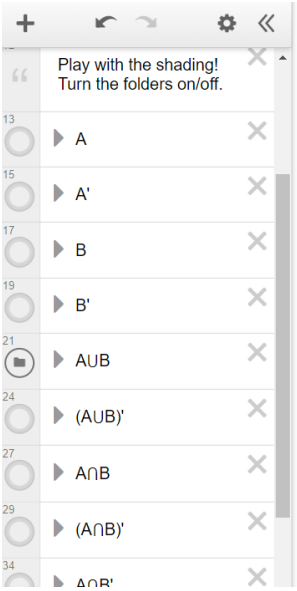
> **Q1. When either Mohit or Rakesh will win their bets?**

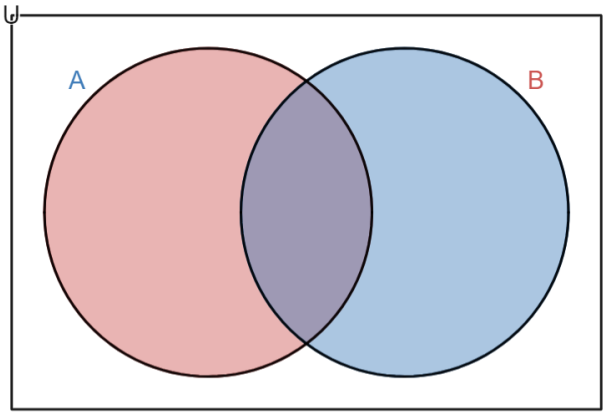
If we get any number out of 1, 3, 5 or 6

- Possible outcomes of this event: $\{1, 3, 5, 6\}$

This is known as **Union** of Two events A and B

- It is denoted by $A \cup B$
- **Union** means **members belonging to either A `OR` B**
- After performing UNION on two events, it'll combine their outcomes together
    - i.e. $A \cup B = \{1, 3, 5, 6\}$





Complement

> **Q1. When will Mohit lose his bet?**

If we get 2, 4 or 6 because these numbers don't occur in the outcome of Mohit's Event.

- This is known as **complement** of Event A
  - It is denoted by A' or A^c

We can define it as the set that contains all the elements Except the elements of A.

- i.e. $A' = \{2, 4, 6\}$

We can also represent it in this way

- $A' = U - A$
  (represents all the elements minus elements of A)

Similarly we know that Rakesh will lose his bet when we get a 2, 4 or 3

- Hence $B' = \{2, 3, 4\}$

> **Q2. When will Mohit win AND Rakesh lose his bet?**

Mohit will win his bet and Rakesh will lose his bet When we will get a **3** on the dice

- As it is the only outcome which doesn't occur in **Rakesh's event**

There is no new set operation for this, we already know the operations.

To denote this, we will need to use a combination of these operations.

- Essentially we want all elements of A, such that they are not present in B
- This can be written as: $A \cap B' = \{3\}$

> **Q3. Similarly, we can find out when will Rakesh win and Mohit lose his bet**

- $B \cap A' = \{6\}$

> **Q4. What about when BOTH Mohit and Rakesh will lose their bets?**
>
> - $A' \cap B' = \{2, 4\}$

Mutually Exclusive Events (Disjoint Events)

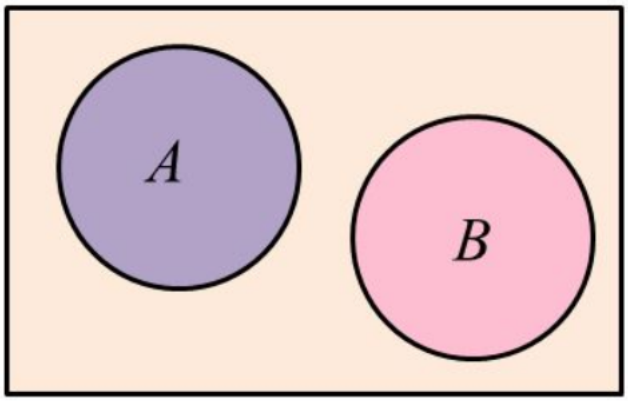
> **Q1. What will be the output of $A \cap C$?**

We will have an empty set $\{\ \}$ which can also be represented by $\emptyset$

Because there are no common elements in Set A and Set C

Or it implies that **both the events can't occur on the same time** means we can't get an **Even number and a Odd number** at the same time on the dice.

- So, when two events cannot occur at the same time or simultaneously then these types of events are known as `Mutually Exclusive Events` or `Disjoint Events`



A and B are mutually exclusive

Exhaustive Events

> **Q1. What will be the output of $A \cup B \cup C$?**

Our events are:

- $A = \{1, 3, 5\}$
- $B = \{1, 5, 6\}$
- $C = \{2, 4, 6\}$
  - Therefore $A \cup B \cup C$ = combined elements of Event A, B, C
  - $A \cup B \cup C$ = {1, 2, 3, 4, 5, 6}

We can observe, that all these events when combined together, it covers all the possible outcomes of our experiment i.e **{1, 2, 3, 4, 5, 6}**.

- These types of events are known as `Exhaustive Events`
- Therefore, events `A and B and C are exhaustive events` because, between them, they encompass all the possible outcomes of rolling the die.

Non Mutually Exclusive Events (Joint Events)

Suppose we define 2 more Events:

- **Event D**: Rolling an even number (2, 4, or 6).
- **Event E**: Rolling a number greater than 3 = (4, 5, or 6).

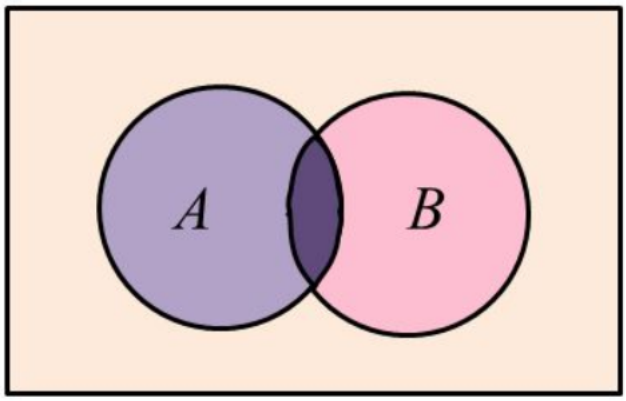
> **Q1. Can we say that Events D and E are mutually exclusive?**

While rolling the die,

we can roll a number that is both even and greater than 3, which means both **events D and E can occur simultaneously**.

- For instance, if the die shows a 4 or a 6, it fulfils the criteria for both events D and E.
- Therefore, events D and E are Not mutually exclusive.

These type of events are known as `non-mutually exclusive` or `joint events`



A and B are not mutually exclusive

Independent Events

While non-mutually exclusive events allow for overlap, where more than one event can occur, independent events focus on how the occurrence of one event **may or may not affect** the likelihood or outcome of another event

Suppose we have 2 two events:

- **Event A**: Rolling an even number (2, 4, or 6)
- **Event B**: Flipping a coin and getting heads

> **Q1. Are these two events Independent or not?**

YES, these events are `independent Events` because

- The outcome of rolling the die (**Event A**) `does not affect the outcome` of flipping the coin (**Event B**), and vice versa.

They are unrelated events that are occurring independently.

- If we get some number on a die, then it'll not affect the chances of getting Heads or Tails while tossing the coin

These types of events are known as `Independent Events`

And if two events A and B are independent, then the probability of happening of both A and B is:

- $P(A \cap B) = P(A) * P(B)$

In case of Disjoint events, $P(A \cap B)$ **= 0** , as A Intersect B = { }

- **So, if the Events are Independent they cannot be Mutually Exclusive or Disjoint and vice a versa**

In the upcoming lectures, we will see how to derive this formula and also prove this claim.

How to calculate Probability

Now if I want to calculate the Probability of the particular event let's say event A, then we can calculate using this.

$$Probability = \frac{Outcomes\ in\ set\ A}{Total\ Outcomes\ in\ Entire\ Sample\ Space}$$

Now, let's take a random Experiment whose **outcome** could be **{1} or {2} or {3} or {4} or {5} or {6}**, then the **Sample Space** will be **{1, 2, 3, 4, 5, 6}**

Let's define some events:

1. A **= {2, 4, 6}**

> **Q1. What will be the probability of Event A?**

- By looking into the formula = $\frac{Possible\ oucomes}{Total\ outcomes}$
- Possible outcomes of event A = 3 and total Outcome in sample space = 6

So, $P(A) = \frac{3}{6}$

2. B = **{1, 2}**
   - Similarly Probability of Event B will be $P(B)$ = $\frac{2}{6}$
3. C = **{1, 4, 5, 6}**
   - and Probability of Event C will be $P(C)$ = $\frac{4}{6}$

Addition Rule

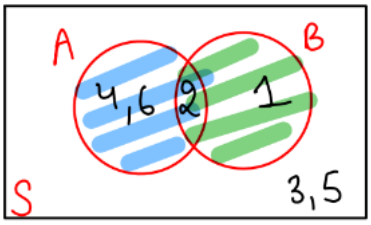
> **Q1. What will be the Probability of $P(A \cup B)$?**

First we need to find $A \cup B$ which is **{1, 2, 4, 6}**

- So by the formula of probability $P(A \cup B)$ will be = $\frac{|A \cup B|}{|S|}$ = $\frac{|\{1,2,4,6\}|}{|\{1,2,3,4,5,6\}|}$ = $\frac{4}{6}$

Where, $|A \cup B|$ = Number of elements(cardinality) of $(A \cup B)$ set,
and $|S|$ = Number of elements in Sample Space

If we want to represent using venn Diagram:



> **Q2. What will be Probability of $P(A \cap B)$?**

$A \cap B$ will be **{2}**

- So by the formula of probability $P(A \cap B)$ will be = $\frac{|\{2\}|}{|\{1,2,3,4,5,6\}|}$ = $\frac{1}{6}$

So by looking into Venn diagram, we observe that AUB means addition of all the elements of **Set A** and **Set B**

- We can also notice in set A we have {2, 4, 6} and in set B we have {1, 2}
- While adding the outcomes of the sets, {2} is occuring twice, which is nothing but $A \cap B$, so we have to subtract it once from our addition, as we want unique outcomes only (Since a set can only have distinct elements).

So the formula for $P(A \cup B)$ can we written as:

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

This is known as **`Addition Rule`**. This is for Joint Events

In case of **Disjoint Events**

- the intersection of $A \cap B = \{\ \}$ so, $P(A \cap B) = 0$
  - therefore, $P(A \cup B) = P(A) + P(B)$

Experiment 3: Sachin Tendulkar ODI records for India

Problem Statement:

We have a dataset containing Sachin Tendulkar's ODI cricket career stats, including various performance metrics and the outcomes of matches.

```
!gdown 1TwgJSuiUW8j3_tXsy6B8YwAzAY0tX4Jk
```

```
Downloading...
From: https://drive.google.com/uc?id=1TwgJSuiUW8j3_tXsy6B8YwAzAY0tX4Jk
To: /content/Sachin_ODI.csv
100% 26.4k/26.4k [00:00<00:00, 47.5MB/s]
```

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df_sachin = pd.read_csv("Sachin_ODI.csv")
```

```
df_sachin.head()
```

	runs	NotOut	mins	bf	fours	sixes	sr	Inns	Opp	Ground	Date	Winner	Won	century
0	13	0	30	15	3	0	86.66	1	New Zealand	Napier	1995-02-16	New Zealand	False	False
1	37	0	75	51	3	1	72.54	2	South Africa	Hamilton	1995-02-18	South Africa	False	False
2	47	0	65	40	7	0	117.50	2	Australia	Dunedin	1995-02-22	India	True	False
3	48	0	37	30	9	1	160.00	2	Bangladesh	Sharjah	1995-04-05	India	True	False
4	4	0	13	9	1	0	44.44	2	Pakistan	Sharjah	1995-04-07	Pakistan	False	False

Each columns represents different features and each row represents a particular match

```
# shape of the dataset

df_sachin.shape
```

```
(360, 14)
```

Sample Space of the experiment consists 360 matches.

Q1. Out of 360 matches, how many matches did India Won?

```
df_sachin["Won"].value_counts()
```

```
True     184
False    176
Name: Won, dtype: int64
```

Solution 1:

True = 184

It indicates that out of the 360 matches, India have won 184 matches and Loose 176 matches

Q2. A match is randomly chosen, what is the probability that India have won that match?

```
# Probablility of India winning : Possible outcome in event (winning) / Total outcome in sample space = 184/360

184/360
```

```
0.5111111111111111
```

Solution:

If we chose a match at a random from this dataset, then there is a **51% chance** that India have won that match

Let's calculate this using the formula of probability, we know:

probability = $\frac{Possible\ Outcomes\ in\ an\ event}{Total\ Outcomes\ in\ an\ Entire\ Sample\ Space}$

Here we want the possible outcomes of India winning a match (WON = True)

Entire sample space will be our entire dataset

```
# find the rows where India have won and store into new dataframe
df_won=df_sachin.loc[df_sachin["Won"]==True]
```

```
# calculate the number of True values which is our possible outcome
df_won.shape[0]
```

```
184
```

```
# We can also look at the length using len()
len(df_won)
```

- So, probability

  = $\frac{number\ of\ matches\ won}{total\ number\ of\ matches}$

```
prob_winning=len(df_won)/len(df_sachin)
prob_winning
```

```
0.5111111111111111
```

Conclusion: :

If a match is randomly chosen, there is **51%** chance that India have won that match.

Q3. A match is chosen at a random, what is the probability that Sachin has scored a Century in that match?

Solution 3:

First let's count the **number of centuries**, Sachin has scored

```
# using value_counts()

df_sachin["century"].value_counts()
```

```
False    314
True      46
Name: century, dtype: int64
```

Out of 360 matches, Sachin has scored 46 Centuries.

so, probability of Sachin scoring a century will be:

```
46/360
```

```
0.12777777777777777
```

Similarly we can calculate the probability using second approach too

- We can create a new dataframe which will only contains the data of those matches where Sachin has scored a century

```
df_century=df_sachin.loc[df_sachin["century"]==True]
len(df_century)
```

```
46
```

Then by using the formula of probability we can calculate the probability of Sachin scoring a century

```
prob_century=len(df_century)/len(df_sachin)
prob_century
```

```
0.12777777777777777
```

Conclusion:

If we chose a random match, there is **12.77% chance** that Sachin has scored a century in that match

Cross Tab:

Now,

Let's find out how many matches India have won when Sachin has `scored a century` and

How many matches India have won when sachin `didn't score a century`.

> **Q1. Can we achieve this task and obtain all these values at once?**

```
df_sachin[["century","Won"]].value_counts()
```

```
century  Won
False    False    160
         True     154
True     True      30
         False     16
dtype: int64
```

> **Q2. What does these values represent?**

`Century` is representing rows, and `Won` is representing columns

- The first row in the **Century (False)** represents Sachin **didn't score a century**.
- The second row in the **Century (True)** represents Sachin has **scored a century**.

- The first column in the **Won (False)** represents India has **lost the match.**
- The second column in the **Won (True)** represents India has **won the match**

Now, if we take first row and first column (False and False), the value is **160**

- means, India have lost 160 matches when Sachin didn't score a century

If we take first row and second column (False and True), the value is **154**

- means, India have Won 154 matches when Sachin didn't score a century

Similarly, we see that

- India have won only 30 matches when Sachin has scored a century and
- India lost only 16 matches when Sachin has scored a century.

We are able to achieve the task at once using values_counts() but the representation is not great and is not properly aligned.

Cross Tab and contingency table

> **Q1. Do you remember pivot table from DAV-1 Libraries module?**

- There is a function called `pd.crosstab()` , which accepts parameters **index** and **columns**.

```
pd.crosstab(index=df_sachin["century"],
            columns=df_sachin["Won"],
            margins=True)
```

Won	False	True	All
century			
False	160	154	314
True	16	30	46
All	176	184	360

What we did using `.valuecounts()` at above, `pd.crosstab()` did the same thing but converted the output into nice tabular format

- **Century** is taken as the **index** and **Won** is taken as **columns**

> **Q2. Now what are the values representing here?**

- These values are giving combinations of **index** and **columns**
  - Wherever **century** is `False` and **Won** is `False` , we get the number of such rows in the dataset or `Frequency` and same for the `True` values

- **160** is the number of rows or frequency representing that number of matches where Sachin didn't score a century (`False`) and India have lost that match (`False`)
- Similarly **154** representing that number of macthes where Sachin didn't score a century (`False`) and India Won that match (`True`)
- **16** represents the number of matches where Sachin has scored a century (`True`) and India have Lost that match (`False`)
- **30** represents the number of matches where Sachin has scored a century (`True`) and India have Won that match (`True`)

This table is also known as `Contingency Table`

> When we do **Margins = True** we get **All**, both in rows and columns, what it represents?

- The values of **All** in a ROW represnts the **Total Value** of each columns (False, True, All)
- The values of **All** in a COLUMN represents the **Total Value** of each rows (False, True, All)

- **176** represents total number of matches we **LOST** (Won -> False)
- **184** represents total number of matches we **WON** (Won -> True)
- **314** represents total number of matches/rows where Sachin **DIDN'T score** a Century (century -> False)
- **346** represents total number of matches/rows where Sachin **scored** a Century (century -> True)
- **360** represents the entire **Sample Space**

We can calculate probabilities using the contingency table.

Q4. A match is chosen at a random. What is the probability that Sachin has scored a century in that match and India have won that match?

Solution 4:

```
pd.crosstab(index=df_sachin["century"],
            columns=df_sachin["Won"],
            margins=True)
```

Won	False	True	All
century			
False	160	154	314
True	16	30	46
All	176	184	360

```
# prob of winning and century
# Won -> True, century -> True

30/360
```

```
0.08333333333333333
```

Second Approach:

Can we calculate this probability in a traditional hard coded way?

```
mask = (df_sachin["Won"]==True) & (df_sachin["century"]==True)
mask
```

```
0      False
1      False
2      False
3      False
4      False
       ...
355    False
356    False
357    False
358    False
359    False
Length: 360, dtype: bool
```

```
df_win_and_century=df_sachin.loc[mask ]
len(df_win_and_century)
```

```
30
```

```
prob_win_and_century=len(df_win_and_century)/len(df_sachin)
prob_win_and_century
```

```
0.08333333333333333
```

Conclusion :

There is **8% chance** that Sachin has scored a century and India have won that match if we choose a random match

This tells us, that **contingency table** is more convenient to calculate probabilities rather than hard coded the every single line

Conclusion of the Problem statement:

Let's have a look how is Sachin's batting can or cannot impact the winning chances of India

1. Out of the **360** matches that Sachin has played, India have **won 184** matches and Loose 176 matches.
2. So, if we choose any match at a random from Sachin's ODI career, there is a **51%** chance that India have won that match.
3. Now, If we choose a random match from Sachin's ODI career, there is **12.77%** chance that Sachin has scored a century in that match.
4. We know if a random match is choosen, there is 12.77% chance that Sachin has scored a century but

   there is **only 8%** chance India have won that match.

   - we can conclude that the chances of India, Winning a match is more when Sachin didn't score a century (what an amazing insight)

Finally,

We can conclude that, if we pick a random match where Sachin played, India's win percentage is 51%. There is 12.77% chance of Sachin scoring a century in that match, and there is only 8% chance that in that match Sachin scores a century as well as India have won that match

Content

- Conditional Probability
- Multiplication Rule
- Marginal and Joint Probability
- Law of Total Probability
- Baye's Theorem
    - Prior, Posterior and Likelihood Probabilities

WhatsApp Autocomplete Example

Conditional probability is a very important concept to understand.

In our daily life, all of us can see direct examples of conditional probability. Lets look at one of them.

Suppose we are typing a message text on WhatsApp. We have typed 2 words of the sentence: `How are`

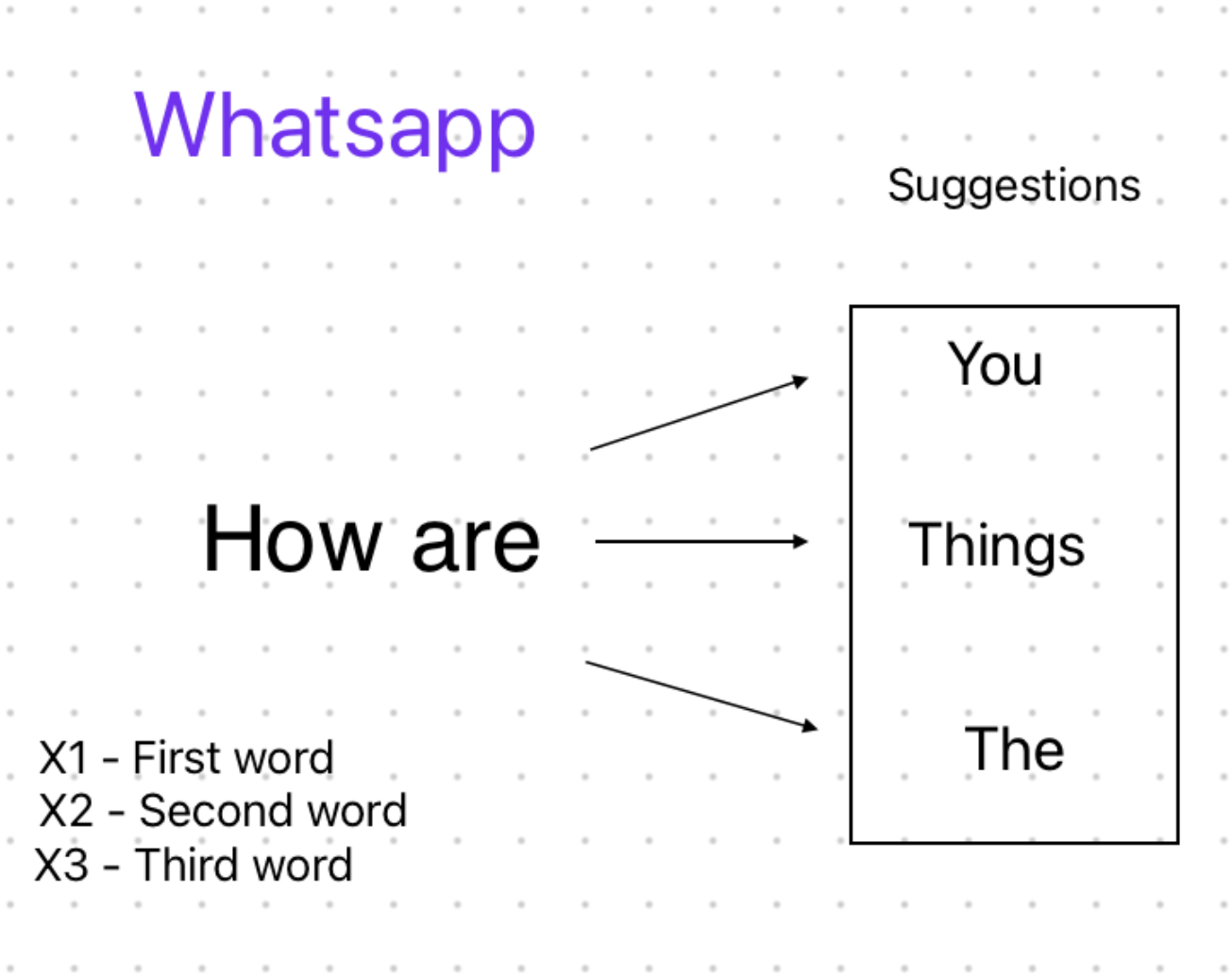
We can notice that at the top of our keyboard, we encounter 3 suggestive words. Like `you`, `things` and `the`.

Though it is not a gaurantee that we need to use one of these words next, but if we look at them, they are very very probable to be used.

Is that magic? How did they know which words we may want to use next?

Let's assign a simple notations

- Let x_1 represents the first word
- Let x_2 represents the second word
- Let x_3 represents the third word



Now, we have given the following information to the keyboard:

- x_1 = "How"
- x_2 = "are"

Now internally, the algorithm needs to compute the probability for a word w that belongs in the dictionary, given the information about words x_1 and x_2.

Consider this structure: $P(A|B)$

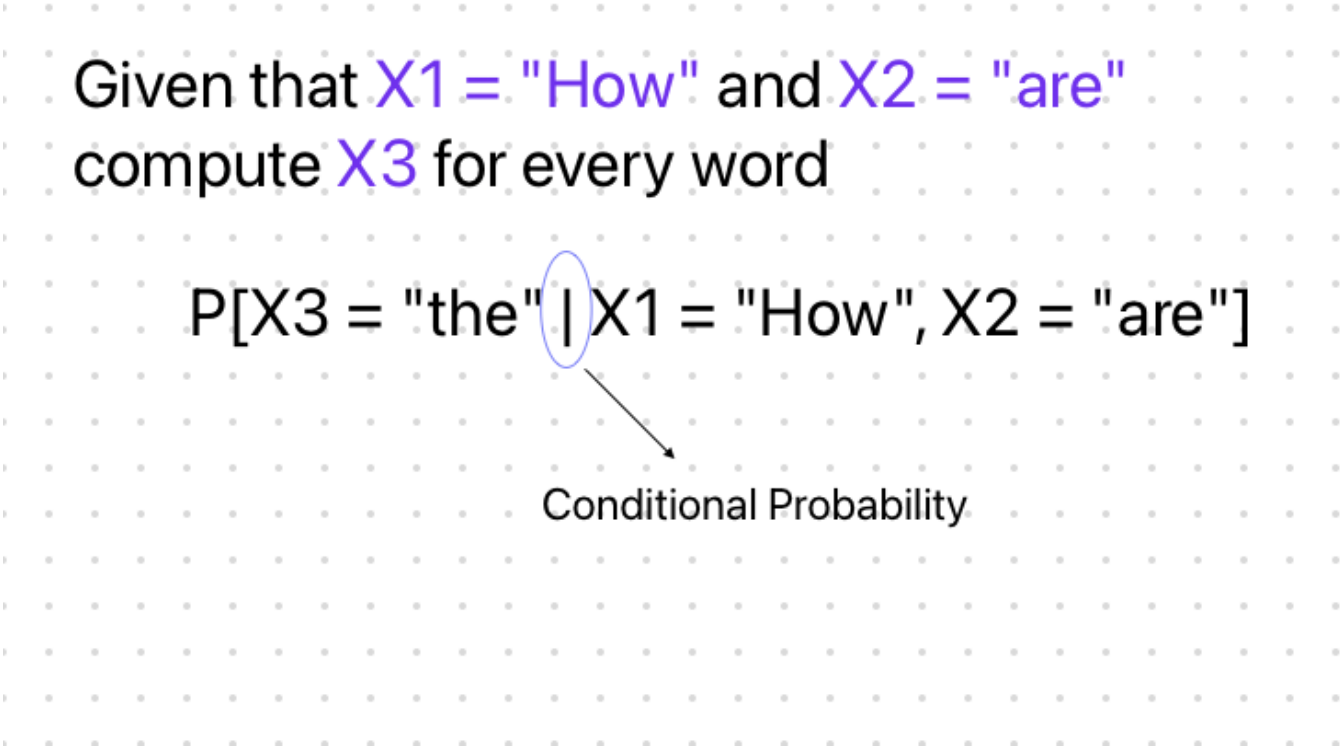
- Here, A represents the event whose probability we are trying to find
- B represents the events that have already happened / information given to us
- The vertical line $|$ represents conditional probability

Therefore, we can represent it as:

$$P(x_3 = w | x_1 = \text{" } How \text{ "} \ \ and \ x_2 = \text{" } are \text{ "})$$

We understand that the algorithm somehow manages to calculate this probability.

It then presents its findings, i.e. the words that are most likely to occur (having maximum probability) given that we have seen the words x_1 and x_2.

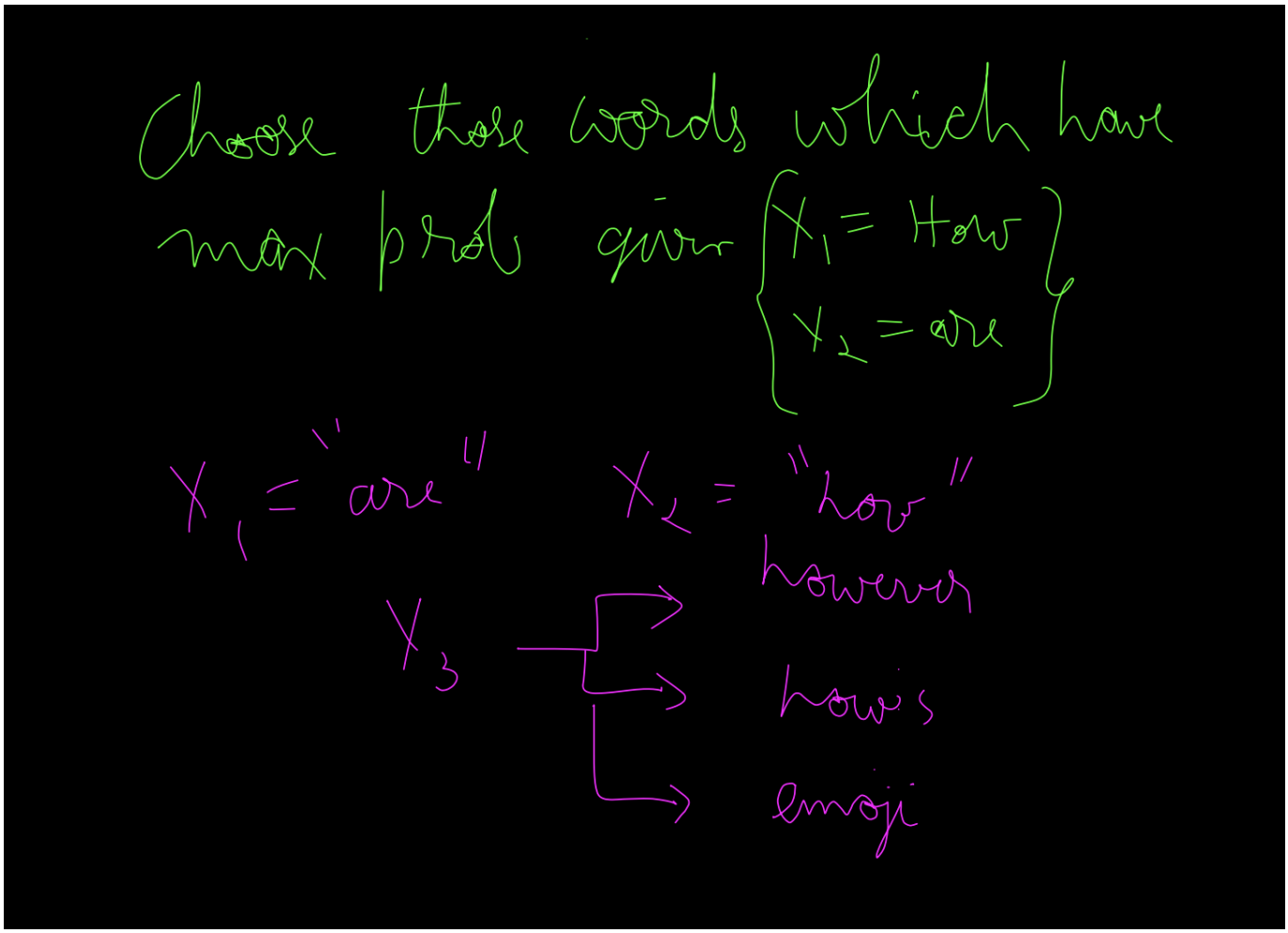


Note:

- The sequence is also important here.
- `you`, `things`, `the` are the top suggestions when x_1 = "How" and x_2 = "are".
- It would suggest different words if the case was x_1 = "are" and x_2 = "How"

Since this is not a sequence of words used very often, it might not give good suggestions here.

Auto complete is another example.



Conditional Probability

Probability of Event A, given Event B has already happened, is equivalent to the probability of A∩B , divided by probability of event B

i.e. $P(A|B) = \frac{P(A \cap B)}{P(B)}$

This equation is known as the **Conditional Probability Formula**

Multiplication Rule

Let's analyse this further,

From the above formula we will get:

$P(A \cap B) = P(A|B).P(B)$

In probability and statistics, this is known as the **Product / Multiplication Rule**.

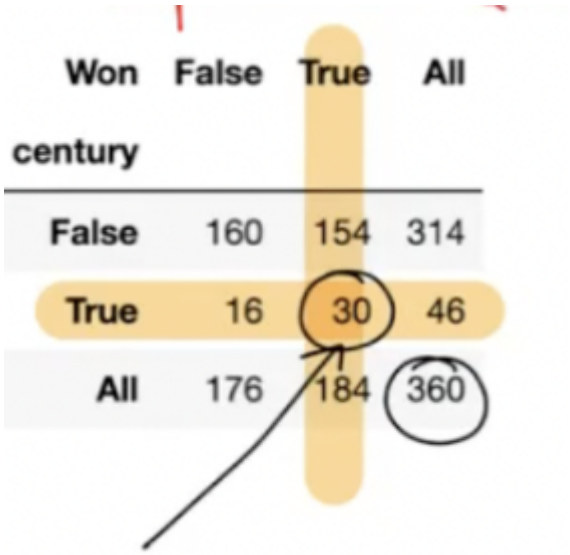
Similarly, we can expand $P(B \cap A) = P(B|A).P(A)$

Marginal and Joint Probabilities

Experiment: Sachin Tendulkar batting for India

Let's define the events happening here:

- W : Sachin's team winning the match
- C: Sachin scoring a century



Won	False	True	All
century			
False	160	154	314
True	16	30	46
All	176	184	360

1) Marginal Probability

Let's answer a few questions based on this contingency table

> **Q1.What is the probability that Sachin's team wins the match?**

We need to find $P(W) = \frac{No\ of\ matches\ won\ by\ Sachin}{Total\ no\ of\ matches} = \frac{184}{360}$

> **Q2.What is the probability of Sachin scoring a century?**

$P(C) = \frac{No\ of\ matches\ with\ century}{Total\ no\ of\ matches} = \frac{46}{360}$

Similarly, we can calculate $P(W^C)$ and $P(C^C)$ as well.

All of these probability values are known as **Marginal Probability**

- It is the probability of an event irrespective of the outcome of other variable.
- For instance, consider $P(W)$

    - It denotes the total probability of Sachin's team winning the match, considering both possibilities that Sachin may or may not score a century.
- It is not conditioned on another event. It may be thought of as an **unconditional probability.**
- Other example:
    - Probability that a card drawn is a 4 : P(four)=1/13.
        - This includes the possibility of the 4 being a spades, heart, club or diamond.
    - Probability that a card drawn is spades : P(spades)=1/4.

2) Joint Probability

Now let's look at the second type of probability values, by answering the following questions.

> **Q1.What is the probability that Sachin's team wins AND he scores a century?**

We need to find $P(W \cap C) = \frac{30}{360}$

> **Q2.What is the probability that Sachin scored a century AND his team wins?**

We need to find $P(C \cap W)$

This will be the same as $P(C \cap W) = P(W \cap C) = \frac{30}{360}$

> **Q3.What is the probability that Sachin scores a century AND his team loses?**

$P(W^C \cap C) = \frac{16}{360}$

Similarly, we can find $P(W^C \cap C^C)$ and $P(W \cap C^C)$

Note:

- Here we calculated the likelihood of two events occurring **together** and at the same point in time.
- This type of probability value is known as **Joint Probability**.
- And it is represented as we saw: $P(A \cap B)$
    - Where, A and B are 2 events.
    - It is read as `Probability that event A and B happen at same time`.
- Other Example: the probability that a card is a four and red = P(four and red) = 2/52

The third kind of probability value, we've just studied, i.e. **Conditional Probability**.

Let's answer a few questions on this also

> **Q1.What is the probability that Sachin's team wins the match given that he scored a century?**

Since it is given that he scores a century, our subset reduces to the second row.

Now since we want to find the prob of team winning among these matches, our probability becomes: $P(W|C) = \frac{30}{46}$

> **Q2.What is the probability that Sachin scores a century, given that his team has won the match?**

As per the given extra information, our subset reduces to the second column.

So among these 184 matches, where India won, Sachin scored a century in only 30 matches.

Therefore $P(C|W) = \frac{30}{184}$

Similarly, we can be asked to calculate other conditional probabilities such as: $P(W|C^C)$, $P(W^C|C)$, $P(C|W^C)$, etc.

Double-click (or enter) to edit

> **Q.How can we find the values of Marginal Probability?**

Law of Total Probability

- If we re-arrange the formula of conditional probability $P(A|B) = \frac{P(A \cap B)}{P(B)}$, we will get get:

  $P(A \cap B) = P(A|B) * P(B)$

This is known as **Law of Total Probability**

Total Probability Law Generic Formula

- Mathematically, The Law of Total Probability is stated as follows:

$P(A) = \sum_{i=1}^{n} P(A \mid B_i)P(B_i)$

Let's have a look into example

```
Example: Email Spam Detection
```

The Law of Total Probability helps combines the information from multiple scenarios or conditions to arrive at a comprehensive probability estimate, making it a valuable tool in various data science and machine learning applications.

Formulas learned so far

1) Conditional Probability:

- $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$

2) Multiplication Rule:

- $P(A \cap B) = P(A \mid B) \cdot P(B)$

3) Law of Total Probability:

- $P(A) = \sum_{i=1}^{n} P(A \mid B_i) P(B_i)$

Let's jump to new concept

Baye's Theorem

- $P(A|B) = \frac{P(B|A).P(A)}{P(B)}$

This equation that we used here is known as the **Bayes Theorem.**

Quick Derivation of Bayes Theorem

From the questions we have solved so far,

> **Q1. Can we say that $P(A \cap B) = P(B \cap A)$?**

We know that $A \cap B$ and $B \cap A$ represent the same subset, i.e. the common elements between A and B.

And, from the Multiplication Rule we can expand them as:

- $P(A \cap B) = P(A|B).P(B)$
- $P(B \cap A) = P(B|A).P(A)$

Since the LHS of both these equations is same, we can equate the RHS also.

$$P(A|B).P(B) = P(B|A).P(A)$$

Dividing both sides by $P(B)$,

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

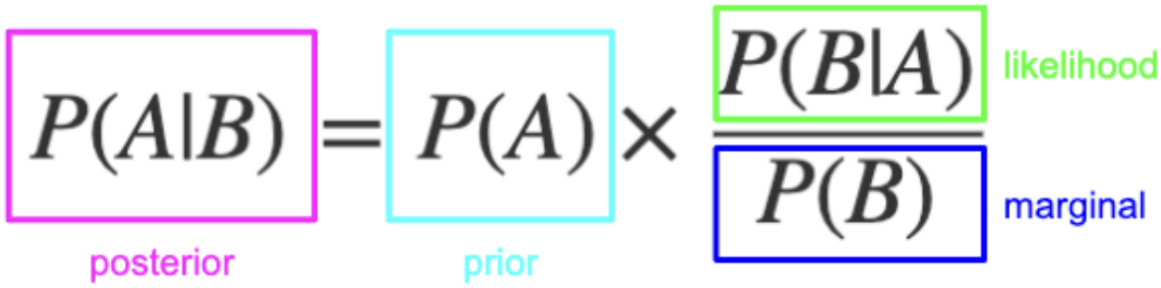
This is exactly the equation of Baye's Theorem.

Let's take a closer look at the Bayes equation.

Prior, Posterior and Likelihood Probabilities

It consists of 4 parts:

- **Posterior probability** (updated probability after the evidence is considered)
- **Prior probability** (the probability before the evidence is considered)
- **Likelihood** (probability of the evidence, given the belief is true)
- **Marginal probability** (probability of the evidence, under any circumstance)



The equation: Posterior = Prior x (Likelihood over Marginal probability)

To understand this better, let's think in a different context.

Consider 2 events:

- **Hypothesis** (which can be true or false), and
- **Evidence** (which can be present or absent).

Therefore, we can write bayes theorem as follows:

$$P(Hypothesis \mid Evidence) = P(Hypothesis) \times \frac{P(Evidence \mid Hypothesis)}{P(Evidence)}$$

Let's understand the different terms here.

- **Posterior probability**
  - The Bayes' Theorem lets you calculate the posterior (or "updated") probability.
  - It is the conditional probability of the **hypothesis being true, if the evidence is present.**
  - $P(Hypothesis|Evidence)$
- **Prior Probability**
  - Can be perceived as your **belief in the hypothesis before seeing the new evidence.**
  - Therefore, if we have a strong belief in the hypothesis already, the prior probability will be large.
  - $P(Hypothesis)$
- **Likelihood**
  - The prior is multiplied by a fraction.
  - Think of this as the "strength" of the evidence.
  - The posterior probability is greater when the top part (numerator) is big, and the bottom part (denominator) is small.
  - The numerator is the likelihood.
  - It is the conditional probability of the **evidence being present, given the hypothesis is true.**
  - This is not the same as the posterior!!
    $P(Evidence|Hypothesis) \neq P(Hypothesis|Evidence)$
- **Marginal Probability**
  - Notice the denominator of this fraction.
  - It is the marginal probability of the evidence. $P(Evidence)$
  - That is, it is the **probability of the evidence being present, whether the hypothesis is true or false.**
  - We can find it using Total Probability Law
  - The smaller the denominator, the more "convincing" the evidence

Content

- **Problem Solving**
- **Mini Case Study**

Formulas learnt so far

Let's recall all the formulas that we have learned so far,

1. **Conditional probability:** $P[A|B] = \frac{P[A \cap B]}{P[B]}$
2. From conditional probability we will get,

   $P[A \cap B] = P[A|B] * P[B]$

   which is known as **Multiplication Rule**
3. **Bayes Theorem:** $P[A|B] = \frac{P[B|A] * P[B]}{P[A]}$
4. **Law of total probability:** $P(A) = \sum_{i=1}^{n} P(A \mid B_i)P(B_i)$
5. **Independent Events:** $P[A \cap B] = P[A] * P[B]$

Quick Derivation of the formula of an Independent Events

Experiment: **Tossing a Coin Followed by rolling a Die**

> **Q1. What is the Sample Space for this event?**

- {(H,1), (H,2), (H,3), (H,4), (H,5), (H,6), (T,1), (T,2), (T,3), (T,4), (T,5), (T,6)}

Let's define some events for this experiment:

1. **A: Event of getting heads**
   - Then **P(A)** will be : $\frac{6}{12}$ (Heads can be occur 6 times out of 12 times)
2. **B: Event of getting 3 on a die**
   - Then **P(B)** will be : $\frac{2}{12}$ (3 can be occur two times out of 12 times)

> **Q2. Calculate the probability of an event of getting Heads `and` 3 on a die**

We want to calculate $P(A \cap B)$ here,

There is only one such outcome possible (H3) out of 12 total outcomes so $A \cap B$ = {(H,3)}

- Therefore, $P(A \cap B) = \frac{1}{12}$

> **Q3. Calculate the probability of getting heads given that 3 has occured on a die.**

In this question, we need to calculate $P(A|B)$

We know $P(A|B)$ = $\frac{P(A \cap B)}{P(B)}$

- Replacing $P(A \cap B)$ and $P(B)$ with their values, we get,

  $P(A|B)$ = $\frac{1}{2}$

We can notice that value of $P(A|B) = P(A) = \frac{1}{2}$

Let's verify it by calculating $P(B|A)$ also:

$P(B|A) = \frac{P(A \cap B)}{P(A)}$

$P(B|A) = \frac{1}{6} = P(B)$

Therefore, we can observe that **Event A is Independent of Event B if**,

- $P(A|B) = P(A)$ **and** $P(B|A) = P(B)$

We can write $P(A|B)$ = $\frac{P(A \cap B)}{P(B)}$ as $P(A \cap B) = P(A|B) \ * \ P(B)$

From above calculation, we saw that $P(A|B)$ = $P(A)$

Therefore,

$P(A \cap B) = P(A) \ * \ P(B)$

Hence, proved

Now let's verify one claim.

Claim: If A and B are mutually Exclusive then A and B are not independent.

We know that if A and B are mutually exclusive or Disjoint events:

- $A \cap B = \{\}$
  Note : $A \cap B$ is a null/empty set as A and B can't occur at the same time
- So, $P(A \cap B) = 0$

But in the case of independent events:

- $P(A \cap B) = P(A) * P(B)$ (we just saw above)

In the case of mutually exclusive events $P(A \cap B)$ is not equal to $P(A) * P(B)$, as A and B are not independent.

Therefore, the claim is proven: If A and B are mutually exclusive, then A and B are not independent.

Alternate Method : Using the conditional probability formula:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

For Disjoint events:

- $P(A \cap B) = 0$
  - So, $P(A|B) = \frac{0}{P(B)} = 0$

For independent Events:

- $P(A \cap B) = P(A) * P(B)$
  - So, $P(A|B) = \frac{P(A) * P(B)}{P(B)} = P(A)$

As we can see in both the events $P(A|B)$ is different

Hence, we can conclude that :

If A and B are mutually Exclusive then A and B are not independent.

Double-click (or enter) to edit

Example: 1

```
In a university, 30% of faculty members are females. Of the female faculty members, 60% have a PHD. Of the male faculty members, 40%
- What is the probability that a randomly chosen faculty member is a female and has PHD?
- What is the probability that a randomly chosen faculty member is a male and has PHD?
- What is the probability that a randomly chosen faculty member has a PHD?
- What is the probability that a randomly chosen PHD holder is female?
```

Explanation:

Given,

- Female faculty members = 30%
  - Out of this 30% members, 60% have PHD

- Male faculty members = 100 - 30 = 70%
    - Out of this 70% members, 40% have PHD

Let's define probabilities:

- probability that a randomly chosen faculty member is a female i.e. $P(F) = 0.3$
    - Given that faculty member is a Female, the probability that she has a PHD is i.e $P(phd \mid F) = 0.6$
- probability that a randomly chosen faculty member is a Male i.e. $P(M) = 0.7$
    - Given that faculty member is a Male, the probability that he has a PHD is i.e $P(phd \mid M) = 0.4$

Answering questions:

> **Q1. What is the probability that a randomly chosen faculty member is a female and has PHD?**

We know `AND` means intersection, here we want to find $P(phd \cap F)$

- Using the formula of conditional probability,

  $P(phd \mid F) = \frac{P(phd \cap F)}{P(F)}$

  So, $P(phd \cap F) = P(phd \mid F) * P(F)$

  Adding values into the equation
    - $P(phd \cap F) = 0.6 * 0.3 = 0.18$

Conclusion:

The probability that a randomly chosen faculty member is a female and has PHD is **0.18**

Similarly,

> **Q2. What is the probability that a randomly chosen faculty member is a male and has PHD?**

- Using the formula of conditional probability,

  $P(phd \mid M) = \frac{P(phd \cap M)}{P(M)}$

  so, $P(phd \cap M) = P(phd \mid M) * P(M)$

  Adding values into the equation
    - $P(phd \cap M) = 0.4 * 0.7 = 0.28$

Conclusion:

The probability that a randomly chosen faculty member is a male and has PHD is **0.28**

> **Q3. What is the probability that a randomly chosen faculty member has a PHD?**

We have 2 approaches to solve this question.

Approach 1:

- Here, we need to find the probability that If I choose a random person, then he/she have a PHD, no matter whether the person is MALE or FEMALE. i.e. $P(phd)$
- We can add $P(phd \cap F) + P(phd \cap M)$ as it'll give me $P(phd)$
- $P(phd) = P(phd \cap F) + P(phd \cap M)$

  adding values into the equation
    - $P(phd) = 0.18 + 0.28 = 0.46$

Approach 2:

- As we know, we can write $P(phd \cap F)$ **as a** $P(phd \mid F) * P(F)$ because, $P(phd \mid F) = \frac{P(phd \cap F)}{P(F)}$

  Here comes the Law of total probability in picture

- For Male also, we can write $P(phd \cap M)$ **as a** $P(phd \mid M) * P(M)$

Replacing these values in the equation,

- $P(phd) = [P(phd \mid F) * P(F)] + [P(phd \mid M) * P(M)]$
  - $P(phd) = [0.6 * 0.3] + [0.4 * 0.7]$

    = $P(phd) = 0.46$

Conclusion:

The probability that a randomly chosen faculty member has a PHD is **0.46**

> **Q4. What is the probability that a randomly chosen PHD holder is female?**

Here, we are already given that the randomly chosen person is PHD holder and we need to find the probability of this person being Female. We need to find: $P(F \mid phd)$

Using the **formula of conditional probability:**

- $P(F \mid phd) = \frac{P(phd \cap F)}{P(phd)}$

  Replace the $P(phd \cap F)$ with $P(phd \mid F) * P(F)$,

  and $P(phd)$ with $[P(phd \mid F) * P(F)] + [P(phd \mid M) * P(M)]$

Final forumla will be:

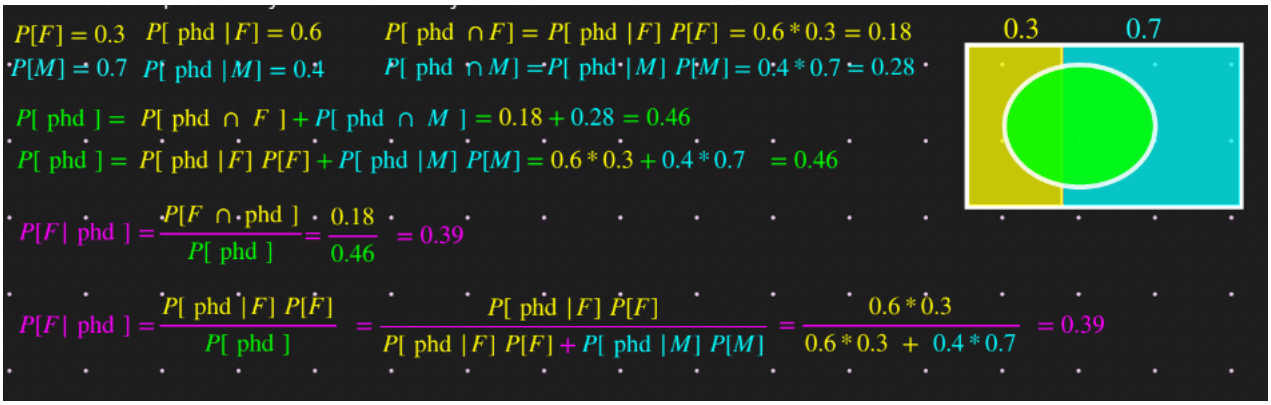
- $P(F \mid phd) = \frac{P(phd \mid F) * P(F)}{[P(phd \mid F) * P(F)] + [P(phd \mid M) * P(M)]}$
  - $P(F \mid phd) = \frac{0.6*0.3}{[0.6*0.3]+[0.4*0.7]}$
  - $P(F \mid phd) = 0.39$

Conclusion:

The probability that a randomly chosen PHD holder is female is **0.39**

There is an alternative approach to solve this question, called **tree based approach**

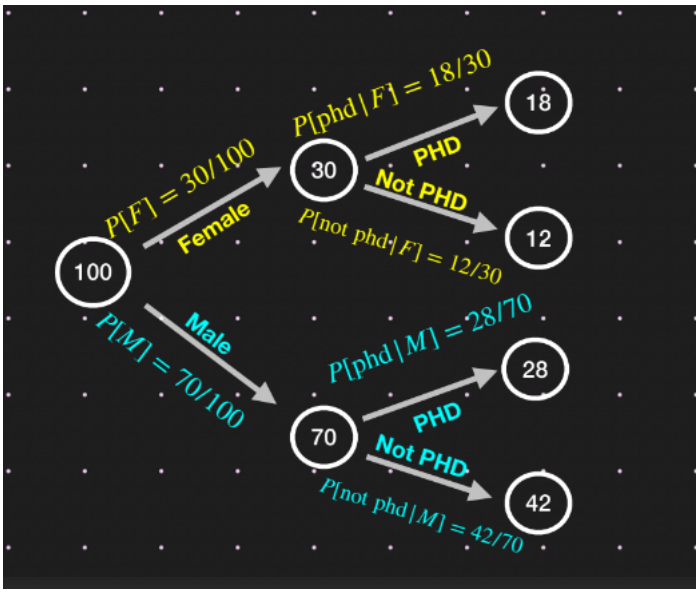
Let's solve this question with tree based approach.



Tree based approach:

Let's assume there are 100 faculty members. Now among these 100 faculty members,

They can be divided into two parts, they can be either male or female.



Explanation of the structure of the Tree:

> **Q1. How many of them are female and how many of them are Male?**

Female : 30% of 100 = 30 (as $P(F) = 0.3$)

We can further seggregate the female part into 2 part:

- Female **AND having** a PHD : 60% of 30 = 18
  - We can represent it as $P(phd \mid F) = 0.6$
- Female **AND NOT having** a PHD : 30 - 18 = 12
  - We can represent it as $P(phd' \mid F) = 1 - P(phd \mid F) = 0.4$

Same for the Male:

Male : 70% of 100 = 70 (as $P(M) = 0.7$)

- Male **AND having** a PHD : 40% of 70 = 28
  - We can represent it as $P(phd \mid M) = 0.4$
- Male **AND NOT having** a PHD : 70 - 28 = 42
  - We can represent it as $P(phd' \mid M) = 1 - P(phd \mid M) = 0.6$

The structure of tree is ready.

Now let's solve the questions

> **Q1. What is the probability that a randomly chosen faculty member is a female and has PHD?**

Let's see how we can easily solve this using tree based approach

We want faculty member and PHD

- From our tree diagram, we can see that there are **18 faculty members who are Female and has PHD**.
  - So $P(F \cap phd)$ = 18/100 = 0.18

    We can observe tha we are getting the same answer but how conviniently we are able to solve this problem with this approach

> **Q2. What is the probability that a randomly chosen faculty member is a male and has PHD?**

Following the same approach as above

- $P(M \cap phd)$ = 28/100 = 0.28

> **Q3. What is the probability that a randomly chosen faculty member has a PHD?**

Here we want to find **total number of faculties having PHD**, it doesn't matter whether the member is male or female

- It will be (18 + 28)/100 = 0.46

> **Q4. What is the probability that a randomly chosen PHD holder is female?**

We have 2 ways to reach the PHD, one through FEMALE and one through MALE

- Now, we need the member **who already has PHD but is a female**.

  It'll be $\frac{18}{18+28}$ = 0.39

> **Q5. What is the probability that a randomly chosen PHD holder is male?**

Following the same approach as above

- $P(M \mid phd) = \frac{28}{18+28}$ = 0.6

We can see how conviniently and easily we are able to solve all the questions using this Tree based approach

Let's implement this on a real life case study.

Kerala Flood Case Study

Double-click (or enter) to edit

Problem Statement:

- The following dataset records monthly rainfall data for the Indian state of Kerala from 1901 to 2018.
- Kerala is known for its vulnerability to annual monsoons, often resulting in significant floods.
- This dataset contains the monthly rainfall index for Kerala and also records whether a flood occurred during a particular month or not.

Your objective is to leverage conditional probability and Bayes' theorem to gain deep insights into the patterns and factors contributing to the occurrence of floods in Kerala.

```
!wget --no-check-certificate https://drive.google.com/uc?id=1Mp2bQl5QJ6O2tcezb0ceBQIn8vW5us0N -O kerala.csv
```

```
--2024-01-31 14:29:12--  https://drive.google.com/uc?id=1Mp2bQl5QJ6O2tcezb0ceBQIn8vW5us0N
Resolving drive.google.com (drive.google.com)... 142.251.162.102, 142.251.162.113, 142.251.162.101, ...
Connecting to drive.google.com (drive.google.com)|142.251.162.102|:443... connected.
HTTP request sent, awaiting response... 303 See Other
Location: https://drive.usercontent.google.com/download?id=1Mp2bQl5QJ6O2tcezb0ceBQIn8vW5us0N [following]
--2024-01-31 14:29:12--  https://drive.usercontent.google.com/download?id=1Mp2bQl5QJ6O2tcezb0ceBQIn8vW5us0N
Resolving drive.usercontent.google.com (drive.usercontent.google.com)... 172.217.193.132, 2607:f8b0:400c:c03::84
Connecting to drive.usercontent.google.com (drive.usercontent.google.com)|172.217.193.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10300 (10K) [application/octet-stream]
Saving to: 'kerala.csv'

kerala.csv          100%[===================>]  10.06K  --.-KB/s    in 0s

2024-01-31 14:29:12 (34.0 MB/s) - 'kerala.csv' saved [10300/10300]
```

```
# Import libraries
import numpy as np
import pandas as pd
```

```
# Read the data
df = pd.read_csv("kerala.csv")
df.head()
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL RAINFALL	FLOODS
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4	3248.6	YES
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5	3326.6	YES
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0	3271.2	YES
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	3129.7	YES
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	2741.6	NO
5	KERALA	1906	26.7	7.4	9.9	59.4	160.8	414.9	954.2	442.8	131.2	251.7	163.1	86.0	2708.0	NO
6	KERALA	1907	18.8	4.8	55.7	170.8	101.4	770.9	760.4	981.5	225.0	309.7	219.1	52.8	3671.1	YES
7	KERALA	1908	8.0	20.8	38.2	102.9	142.6	592.6	902.2	352.9	175.9	253.3	47.9	11.0	2648.3	NO
8	KERALA	1909	54.1	11.8	61.3	93.8	473.2	704.7	782.3	258.0	195.4	212.1	171.1	32.3	3050.2	YES
9	KERALA	1910	2.7	25.7	23.3	124.5	148.8	680.0	484.1	473.8	248.6	356.6	280.4	0.1	2848.6	NO

```
df.shape
```

```
(118, 16)
```

Let's calculate average rainfall for each month over the years

> **Q. What is the average rainfall for each month over the years**

```
# Calculate the average rainfall for each month
cols = ['JAN', 'FEB', 'MAR', 'APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC']

monthly_avg = df[cols].mean()
monthly_avg
```

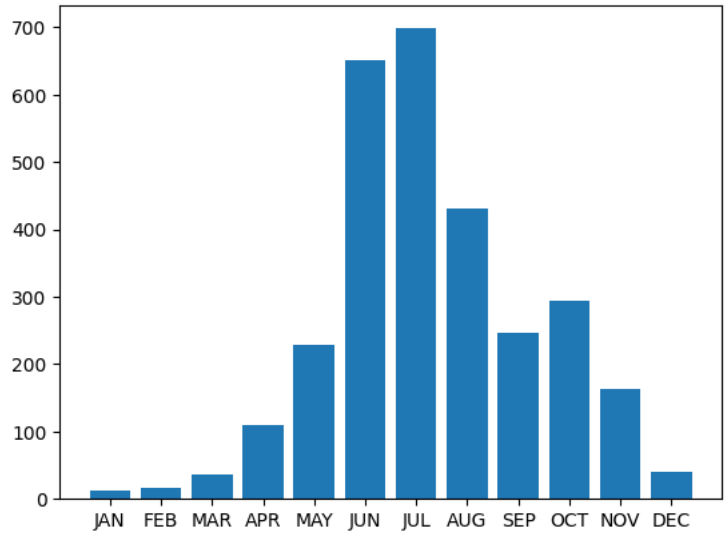
```
JAN     12.218644
FEB     15.633898
MAR     36.670339
APR    110.330508
MAY    228.644915
JUN    651.617797
JUL    698.220339
AUG    430.369492
SEP    246.207627
OCT    293.207627
NOV    162.311017
DEC     40.009322
dtype: float64
```

Let's visualise this data:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
x=monthly_avg.index
y=monthly_avg

plt.bar(x,y)
```

```
<BarContainer object of 12 artists>
```



We can make few **conclusions** here:

- The data reveals significant seasonal variation in rainfall.
    - For instance, the months of **June and July** have the **highest average rainfall**, on an average. This suggests that these two months are typically the wettest in the region.
    - The months of **January and February** have the **lowest average rainfall**, these are typically driest months.
    - The rainfall in **August and September** is still relatively high but begins to decline
    - The months of October, November, and December have moderate to low average rainfall, with **October** having the **highest average** of the three.

You can see **October** has a **higher average rainfall than September**, which may seem counterintuitive, as it should be decling only.

There are two monsoon seasons in Kerala, **one during Jun-Aug, Other during Oct**.

To understand this and uncover the reasons behind it, we can check their yearly trends

Let's look into the statistics of this dataset

```
df.describe()
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	
count	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.00
mean	1959.500000	12.218644	15.633898	36.670339	110.330508	228.644915	651.617797	698.220339	430.369492	246.207627	293.20
std	34.207699	15.473766	16.406290	30.063862	44.633452	147.548778	186.181363	228.988966	181.980463	121.901131	93.70
min	1901.000000	0.000000	0.000000	0.100000	13.100000	53.400000	196.800000	167.500000	178.600000	41.300000	68.50
25%	1930.250000	2.175000	4.700000	18.100000	74.350000	125.050000	535.550000	533.200000	316.725000	155.425000	222.12
50%	1959.500000	5.800000	8.350000	28.400000	110.400000	184.600000	625.600000	691.650000	386.250000	223.550000	284.30
75%	1988.750000	18.175000	21.400000	49.825000	136.450000	264.875000	786.975000	832.425000	500.100000	334.500000	355.15
max	2018.000000	83.500000	79.000000	217.200000	238.000000	738.800000	1098.200000	1526.500000	1398.900000	526.700000	567.90

Here:

1. "mean" is representing **average value** for each column
    - For instance, we can see that Average Annual rainfall is around **2925.405085** (mean of Annual Rainfall)
2. "min" and "max" is representing **Minimum** and **Maximum** value for each column

There are lot of other few statistics, which we will see later in this module

Now,
Let's try to visualise the spread of our entire dataset using Box plot

```
columns = df.columns.tolist()
columns
```

```
['SUBDIVISION',
 'YEAR',
 'JAN',
 'FEB',
 'MAR',
 'APR',
 'MAY',
 'JUN',
 'JUL',
 'AUG',
 'SEP',
 'OCT',
 'NOV',
 'DEC',
 ' ANNUAL RAINFALL',
 'FLOODS']
```

We want only months column

```
df2 = df[columns[1:14]]
df2.head()
```

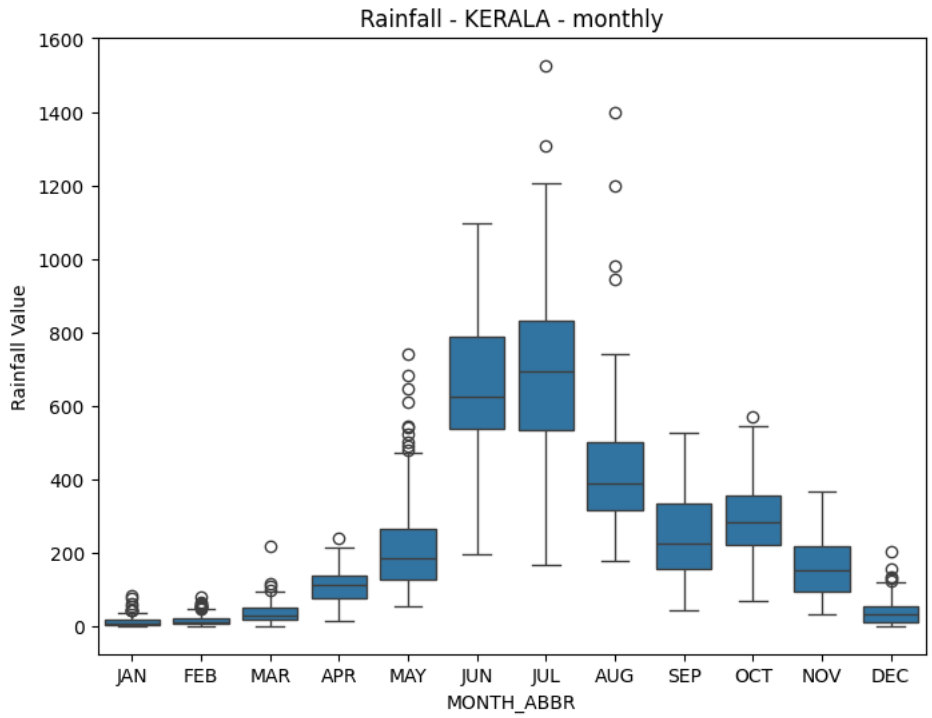
	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
0	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4
1	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5
2	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0
3	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3
4	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2

```
df3  = pd.melt(df2,
               id_vars = ['YEAR'],
               value_vars = ['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC'],
               var_name='MONTH_ABBR', value_name='VALUE')


df3.head()
```

	YEAR	MONTH_ABBR	VALUE
0	1901	JAN	28.7
1	1902	JAN	6.7
2	1903	JAN	3.2
3	1904	JAN	23.7
4	1905	JAN	1.2

Let's plot the Box plot

```
fig, ax = plt.subplots(1, 1, figsize=(8, 6))
sns.boxplot(data=df3, x='MONTH_ABBR', y=df3.VALUE, ax=ax)
ax.set_ylabel('Rainfall Value')
ax.set_title('Rainfall - KERALA - monthly')

plt.show()
```



Conclusions

We can clearly see that the raninfall is started to rise and is at peak in the month of june and july,

Then again started to decline but there is again rise in month of October.

Through which we can conclude that Kerala has 2 rainy seasons

Let's plot the Annual rainfall and try to see the yearly trend

```
df.columns
```

```
Index(['SUBDIVISION', 'YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC', ' ANNUAL RAINFALL', 'FLOODS'],
      dtype='object')
```

As you can see there is an extra space in the start of column "Annual rainfall". It is like this: ' ANNUAL RAINFALL'

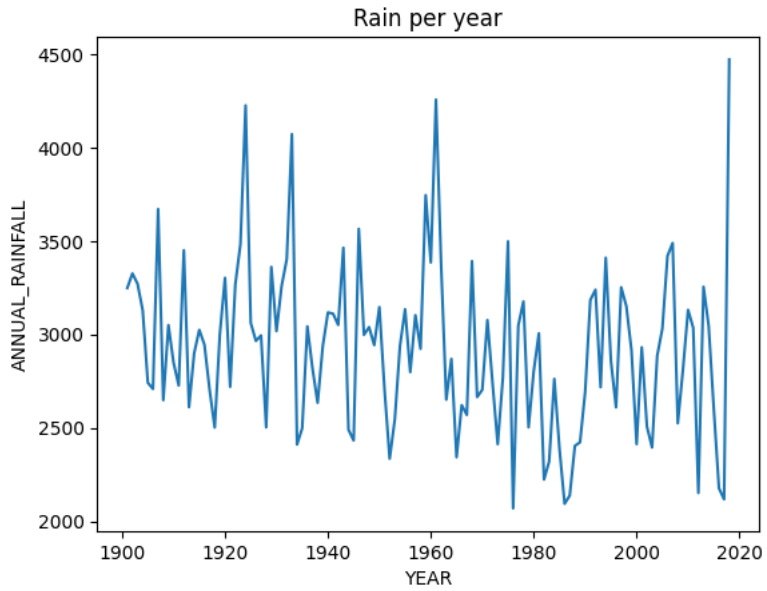
Let's rename this column

```
df.columns = [c.replace(' ANNUAL RAINFALL', 'ANNUAL_RAINFALL') for c in df.columns]

df.head()
```

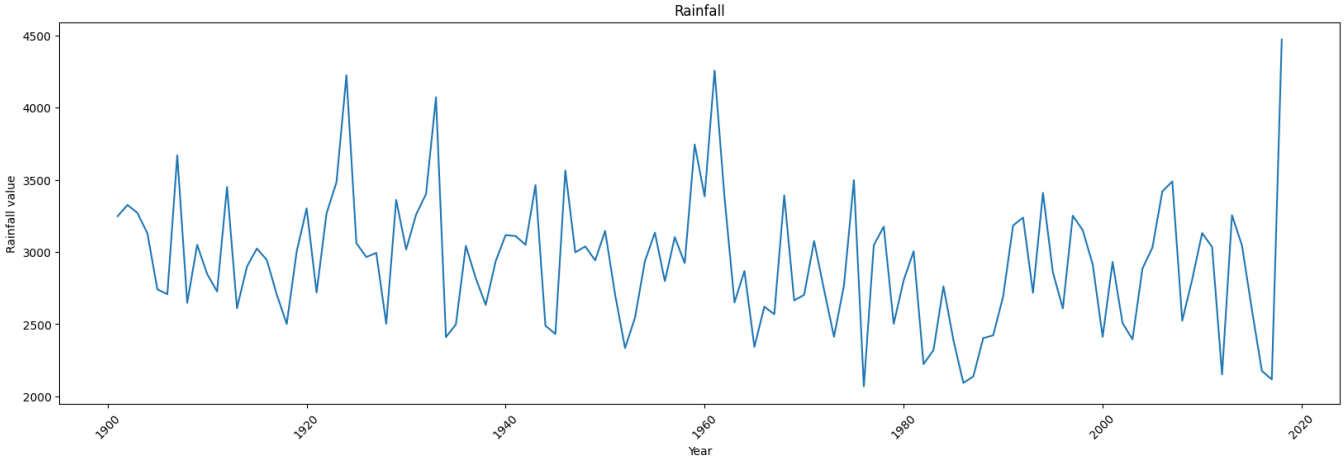
	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL_RAINFALL	FLOODS
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4	3248.6	YES
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5	3326.6	YES
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0	3271.2	YES
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	3129.7	YES
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	2741.6	NO

```
sns.lineplot(data=df,
             x="YEAR",
             y="ANNUAL_RAINFALL")
plt.title("Rain per year")
```

```
Text(0.5, 1.0, 'Rain per year')
```




```
plt.figure(figsize=(20,6))
plt.plot(df['YEAR'], df['ANNUAL_RAINFALL'])
plt.xlabel('Year')
plt.ylabel('Rainfall value')
plt.title('Rainfall')
plt.xticks(rotation=45)
plt.show()
```



As we are done with our analysis, we came to conclusion that the important features in this dataset are "JUN", "JUL", "OCT" , "ANNAUL_RAINFALL", "FLOODS"

because in these months only we have seen the peak of the rainfall which can be one of the major source of causing the flood

Impactful Columns

```
df.columns
```

```
Index(['SUBDIVISION', 'YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC', 'ANNUAL_RAINFALL', 'FLOODS'],
      dtype='object')
```

```
impactful_columns = ['YEAR', 'JUN', 'JUL', 'OCT', 'ANNUAL_RAINFALL', 'FLOODS']

impactful_columns
```

```
['YEAR', 'JUN', 'JUL', 'OCT', 'ANNUAL_RAINFALL', 'FLOODS']
```

Now, I want to label the months column with 0 and 1

- 0: will represents low rainfall
- 1: will represents heavy rainfall

Similarly for "ANNUAL_RAINFALL" column:

- 0: will represents low rainfall in that particular year
- 1: will represents heavy rainfall in that particular year

> **Q. But how much rainfall index is considered as a heavy ranifall?**

One of the parameter is using the **Median** values of these columns.

If their individual **rainfall index value > median value** then it'll we considered as **heavy rainfall** and vice a versa

```
# new dataset containing only impactful columns

data = df[impactful_columns]

data.head()
```

	YEAR	JUN	JUL	OCT	ANNUAL_RAINFALL	FLOODS
0	1901	824.6	743.0	266.9	3248.6	YES
1	1902	390.9	1205.0	358.4	3326.6	YES
2	1903	558.6	1022.5	354.1	3271.2	YES
3	1904	1098.2	725.5	328.1	3129.7	YES
4	1905	850.2	520.5	383.5	2741.6	NO

```
# let's calculate the median of columns and set as their threshold value

threshold_jun = data['JUN'].median().astype(int)
threshold_jul = data['JUL'].median().astype(int)
threshold_oct = data['OCT'].median().astype(int)
threshold_ar = data['ANNUAL_RAINFALL'].median().astype(int)

threshold_jun, threshold_jul, threshold_oct, threshold_ar
```

```
(625, 691, 284, 2934)
```

```
thresholds = {
    'JUN': 625,
    'JUL': 691,
    'OCT': 284,
    'ANNUAL_RAINFALL': 2934
}

# Convert columns to binary based on thresholds
for col, threshold in thresholds.items():
    data[col] = (data[col] > threshold).astype(int)

data.head()
```

```
<ipython-input-21-ce625c741022>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
  data[col] = (data[col] > threshold).astype(int)
```

	YEAR	JUN	JUL	OCT	ANNUAL_RAINFALL	FLOODS
0	1901	1	1	0	1	YES
1	1902	0	1	1	1	YES
2	1903	0	1	1	1	YES
3	1904	1	1	1	1	YES
4	1905	1	0	1	0	NO

```
data.shape
```

```
(118, 6)
```

We are done with some analysis, now our dataset is ready to solve probability related questions

> **We will going to create contigency tables to compare "FLOODS" column with every column**

```
pd.crosstab(index = data['JUN'],
            columns = data['FLOODS'],
            margins=True,
            margins_name='Total')
```

FLOODS	NO	YES	Total
JUN			
0	42	16	58
1	16	44	60
Total	58	60	118

Explanation of contingency table:

index=data['JUN']:

- This specifies the variable that will be used as the **row index** of the contingency table.
- In this case, it's the 'JUN' column, which represents heavy rainfall in June (with values 0 or 1).

columns=data['FLOODS']:

- This specifies the variable that will be used as the **column index** of the contingency table.
- It's the 'FLOODS' column, which represents flooding (with values "YES" or "NO").

margins=True:

- The margins parameter, when set to True, includes row and column margins (totals) in the contingency table.

and `JUN` is representing the rows

- There are 60 records with "1" (indicating rainfaill index exceed threshold)
- There are 58 records with "0" (indicating rainfaill index is less than threshold)

Here, `FLOODS` is represeting the column

- There are 60 records with "YES" (indicating floods)
- There are 58 records with "NO" (indicating no floods)

Now, there are few observations we can make based on this output:

1. There is a strong association between the conditions in June ("JUN") met (JUN = 1) and the occurrence of floods (FLOODS = YES). (Frequency = 44)
    - When it rained more than threshold (JUN = 1), there is a higher likelihood of flood occurring.

2. There is weak association between the conditions in June not met (JUN = 0) and the occurrence of floods ("FLOODS"). (Frequency = 16)
    - When it rained less than threshold (JUN = 0), there is a very low chance of flood occurring.

Q1. Calculate the Probability of flood given that rainfall in June is greater than the median june rainfall value (threshold for heavy rainfall)

Question Explanation:

Let A represents : Flood
B represents: heavy rain in June

We need to calculate $P(A|B)$ i.e. $\frac{P(A \cap B)}{P(B)}$

Solution Approach 1:

We can obtain these values using contingency table and put those values into the formula.

Here we need to compare "FLOODS" and "JUN" column.

```
pd.crosstab(data['JUN'],
            data['FLOODS'],
            margins=True,
            margins_name='Total')
```

FLOODS	NO	YES	Total
JUN			
0	42	16	58
1	16	44	60
Total	58	60	118

Now, $P(A \cap B)$ = Probability of Flood occuring `AND` heavy rainfall in JUNE

As we know in the contingency table, FLOODS = YES represents that flood has occured and JUN = 1 means heavy rainfall.

We need to check value where FLOODS = YES and JUN = 1 which is **44**

Then by the formula of condititonal probability we can feed this data

```
# probability of high rainfall in June P(J)
# P(J) = possible outcomes in june having heavy rainfall / total outcomes

P_J = (16+44)/(42+16+16+44)

# now, P(A and B) (Flood = YES and Jun = 1)

P_F_and_J = 44/(42+16+16+44)

#, so our probability of flood occurring given that the high rainfall occured in June will be

P_F_J = P_F_and_J / P_J

print(f'P(J) : {P_J}')
print(f'P(F AND J) : {P_F_and_J}')
print(f'P(F|J): {P_F_J}')
```

```
P(J) : 0.5084745762711864
P(F AND J) : 0.3728813559322034
P(F|J): 0.7333333333333334
```

Approach 2: using normalize attribute

Explanation of Normalize attribute:

Rather putting all the values in the formula and then calculate the probability

We can just pass one more attribute in pd.crosstab() function which will divide all values by the sum of values.

- This is the probability only, as in probability we divide `possible outcome / total outcome (sum of all values)`

Parameter is : **normalize = ' '**

- Without this attribute, the contingency table will show the raw counts of occurrences for each combination of variables.
- It will not be normalized, and the values in the table will represent counts.

Here we can pass these strings in this attribute:

normalize='index' or **normalize='columns'** :

- The normalize attribute specifies how the values in the contingency table should be normalized.
  - When set to **'index'**, it calculates conditional probabilities based on rows, treating each row as a separate condition.
  - When set to **'columns'**, it calculates conditional probabilities based on columns, treating each column as the condition we are focusing on.
- This means that each row in the table is divided by the sum of its row, making each row's values sum up to 1, representing conditional probabilities.

Same with the column

In this case:

By setting **normalize='index'**,

- the code calculates conditional probabilities within each row.
- Each value in the table represents the probability of the corresponding event (FLOODS) given the value of 'JUN' in that row.

The row sums up to 1, ensuring that it reflects the conditional probabilities.

In summary,

setting normalize='index' in pd.crosstab allows you to calculate and visualize conditional probabilities based on the specified row variable ('JUN' in this case),

making it easier to assess the impact of one variable on another.

```
pd.crosstab(index = data['JUN'],
            columns = data['FLOODS'],
            margins=True,
            normalize='index')
```

FLOODS	NO	YES
JUN		
0	0.724138	0.275862
1	0.266667	0.733333
All	0.491525	0.508475

The values in the table represent the conditional probabilities, where each cell contains the probability of the corresponding outcome (FLOODS) given the condition in June (JUN).

> Then the probability of flood occurring given that the heavy rainfall occured in June will be:

- In the cell at row 1, column 1, the value **0.73333** represents the conditional probability of flooding (FLOODS = YES) given that high rainfall occurred in June (JUN = 1).

Conclusion:

```
So, there is 73.33% chance of Floods when there is a heavy rainfall in June
```

As we can see by calculating using formula also, we are getting the same answer as using directly conditional probability using normalize = 'index'

Now, let's jump into the next question

Q2. Calculate the Probability of flood given that rainfall in July is greater than the median July rainfall value (threshold for heavy rainfall)?

Solution

We are already aware of using formula based approach, so We will solve this using contingency table

Let A represents : Flood
B represents: heavy rain in JULY

We need to calculate $P(A|B)$ i.e. $\frac{P(A \cap B)}{P(B)}$

```
pd.crosstab(index = data['JUL'],
            columns = data['FLOODS'],
            margins=True,
            normalize='index')
```

FLOODS	NO	YES
JUL		
0	0.644068	0.355932
1	0.338983	0.661017
All	0.491525	0.508475

The values in the table represent the conditional probabilities, where each cell contains the probability of the corresponding outcome (FLOODS) given the condition in July (JUL).

> Then the probability of flood occurring given that the heavy rainfall occured in July will be:

- in the cell at row 1, column 1, the value **0.661017** represents the conditional probability of flooding (FLOODS = YES) given that high rainfall occurred in July (JUL = 1).

Conclusion:

```
So, there is 66.1% chance of Floods when there is a heavy rainfall in July
```

Let's solve the next question

Q3. Given that there is a flooding, calculate the probability that heavy rainfall has occurred in July (more than threshold value)?

Here we want to find $P(July = 1|Flood = YES)$

We are already aware of using formula based approach, so We will solve this using contingency table

Before proceeding,

> **Q. In this question, which string will be passed inside normalize=' ' attribute? 'index' or 'columns'**

In this question, we should normalize the contingency table along the columns

- As we want to find the conditional probability of **high rainfall in July (JUL = 1) given that there was flooding (FLOODS = YES)**,

We want to see how the 'JUL' column behaves when there is flooding.

Solution:

```
pd.crosstab(index = data['JUL'],
            columns = data['FLOODS'],
            margins=True,
            normalize='columns')
```

FLOODS	NO	YES	All
JUL			
0	0.655172	0.35	0.5
1	0.344828	0.65	0.5

Conclusion:

The probability that high rainfall occurred in July (JUL = 1) given flooding (FLOODS = YES) is **0.65**.

- **This means that when there is flooding, there is a 65% chance of heavy rainfall in July.**

Q4.Calculate the probability of flood given that june and july rainfall was greater than their median rainfall value?

Solution:

We want to find $P(Flood = Yes|\ june = 1\ and\ Jul = 1)$

Here, we can pass multiple columns in the **pd.crosstab()**

```
pd.crosstab(index = [data['JUN'], data['JUL']],
            columns = data['FLOODS'],
            margins=True,
            normalize='index')
```

FLOODS		NO	YES
JUN	JUL		
0	0	0.862069	0.137931
	1	0.586207	0.413793

Conclusion

Frequency (JUN = 1, JUL = 1, FLOODS = YES) = 0.9000000

Content

- **Combinatorics**
- **Permutations**
- **Permutation : Generic Approach**
- **Combinations**

> **Suppose we have 2 True/False questions. In how many ways can they be solved?**

For question 1, we have two possible outcomes:

- True
- False

Similarly, for question 2 as well.

> **Since we need to solve both questions, will we add or multiply their number of possible outcomes?**

We will multiply since we have to solve question **1 AND 2**

Instead, if we had to solve question **1 OR 2**, we would've added.

This is because, when considering `Event 1 AND Event 2`, we are talking about two **independent events**.

- Solving question 1 is independent from solving question 2
- Hence, we need to multiply to consider their combined effect.

Therefore, we can solve them in $2 * 2 = 4$ ways:

- True, True
- True, False
- False, True
- False, False

Permutation and Combination

> **What is a permutation?**

- When talking about permutations, we mean **arrangement of objects**.
- Therefore, as with arranging objects, the most important thing is **order** is which they are arranged.

- This means that $(i, j) \neq (j, i)$

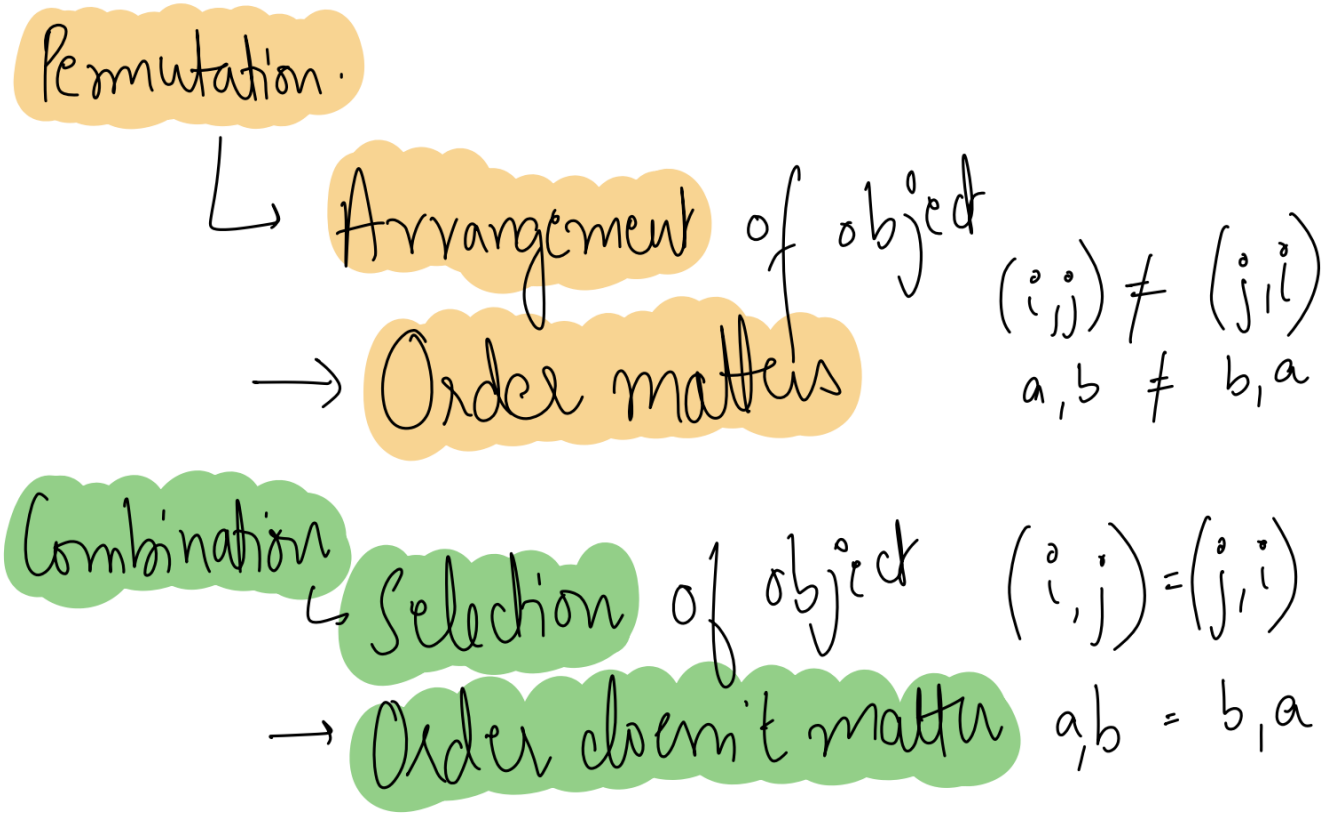
Formal Definition: `A permutation is an arrangement of items or elements in a specific order, where the order of the arrangement matters.`

The second aspect is **Combinations**

> **What is a combination?**

- Combination is **Selection of objects**.
- Over here, the order of objects **does not matter**.
  - This means that $(i, j) = (j, i)$

  Formal Definition: `A combination is selection of items or elements where the order of the arrangement does not matter.`



Permutation : Generic Formula

> **Q1. How would we arrange N object, given that there only 3 slots?**

Since there are 3 slots for `N` objects, the no. of ways in which we can arrange them is ${}^{N}P_3$

i.e. ${}^{N}P_3 = N.(N-1).(N-2)$

> **Q2. How would we arrange N object, given that there only 4 slots?**

$^{N}P_4 = N.(N-1).(N-2).(N-3)$

We can observe a pattern between the no. of slots/blanks, and the last term of the above expressions

> **Q3. Then how would we arrange N object, given that there are `k` slots available?**

This can be found using:

$$^{N}P_k = N(N-1)(N-2)(N-3)....(N-(k-1)) = N(N-1)(N-2)(N-3)....(N-k$$

Let's re-write this equation by multiplying and dividing by same expression, as:

$$^{N}P_k = N(N-1)(N-2)(N-3)....(N-(k-1)) = N(N-1)(N-2)(N-3)....(N-k$$

As we know, we can write this in the form of factorial as: $^{N}P_k = \frac{N!}{(N-k)!}$

Combinations

- Combinations, in simple terms, are all the different ways you can choose a certain number of items from a group, where the order in which you pick them doesn't matter.
- It's like making a sandwich with different ingredients – the combination is the unique mix of ingredients you choose, regardless of the order you add them.

In the language of combinatorics, this number of ways of **selecting** is known as **Combination**.

Similarly, we can write the **general formula** for combinations in terms of permutations as:
$^{n}C_k = \frac{^{n}P_k}{k!}$

We can further expand it as: $^{n}C_k = \frac{n!}{k!(n-k)!}$

Content

- **Descriptive Statistics**
  - Measures of Central Tendency
    - Mean
    - Median
    - Mode
  - Measures of Variability
    - Range
    - Variance
    - Standard Deviation
- **Inferential Statistics**
- **Weighted Average**
- **Inter Quartile Range**
  - Quartile
  - Percentile
  - Box Plot
- **IQR implementation on real life dataset**
- **Random Variables**
  - Discrete RV
  - Continuous RV
- **Distribution Functions**
  - Histogram
  - Probability Mass Function (PMF)
  - Probability Density Function (PDF)
  - Cumulative Distribution Function (CDF)

There are 2 types of Statistics:

1. Descriptive Statistics

The word descriptive means "**DESCRIBE**"

Descriptive statistics involve summarizing and presenting data in a meaningful way, providing a clear and concise overview of a dataset.

Example: Let say you are driving a car and you look at your dashboard.

- The speedometer shows the speed of your car at the moment is 65 km/hr. So, it is simply describing speed.
- This Speedometer simply describes an event that a vehicle is moving at a certain speed so it is an example of descriptive statistics.

2. Inferential Statistics

Inferential statistics, on the other hand, involve making predictions, inferences, or drawing conclusions about a larger population based on a sample of data.

Continuing the example:

- The car's speedometer displays the current speed but **doesn't predict your arrival time** because it depends on various factors like distance and traffic.
- What Google Maps will do here, is estimate arrival time based on data and assumptions, but it's only sometimes 100% accurate.
- This prediction is an example of inferential statistics, as it draws conclusions from real-world scenarios
- It is trying to **"infer"** something. It is concluding out of it. So, it is inferential statistics.

Conclusion:

- Descriptive statistics summarizes data
- Inferential statistics draws conclusions based on the observations.

These two are the essential branches of statistics.

Let's explore descriptive statistics

Measures of Central Tendency

In statistics, we often use measures to understand and describe a set of data.

Three common measures are:

1. **Mean**
   - The Mean is the average of all data points
2. **Median**
   - The Median is the middle value when the data is sorted.
3. **Mode**
   - It is the observation with the highest frequency

Example-1: Data Scientist's Salaries

```
Suppose we are looking for a data scientist job at FAANG.
The sample of salaries is taken and recorded as [30L, 30L, 35L, 40L, 40L].
What will be the salary we would be expecting?
```

Approach:

1) Mean

- Mean will be (30 + 30 + 35 + 40 + 40)/5 = **35 lakhs**
  - $\mu = \frac{\sum X}{N}$

  Where,
  - μ = population mean
  - $\sum X$ = sum of each value in the population
  - N = number of values in the population

So, the mean salary in this sample is 35 lakhs. We might negotiate our expected salary around this figure.

Suppose, a **new candidate comes** in the context and **his salary is 3 crores**.

- **New mean** will become = (30 + 30 + 35 + 40 + 40 + 300)/6 = **79 lakhs**
- The mean salary dramatically increased to 79 lakhs because of the new candidate's exceptionally high salary.

We can observe that this **new candidate is an outlier** in the data which is affecting the mean value.

2) Median

Here comes the concept of **"Median"** to measure central tendency instead of measuring it using **Mean**.

Before the new candidate joined, the Median was 35L. This means that 35L was the center value when the salaries were sorted in ascending order:

- **Original Salaries:** [30L, 30L, 35L, 40L, 40L]
- **Sorted:** [30L, 30L, 35L, 40L, 40L]
    - **Median** = $35L$ (the middle value)

After the new candidate with a significantly higher salary arrived (300L), the new Median became 37.5 lakhs:

- **New Salaries**: [30L, 30L, 35L, 40L, 40L, 300L]
- **Sorted**: [30L, 30L, 35L, 40L, 40L, 300L]
    - **New Median** = (35L + 40L) / 2 = 75L / 2 = $37.5L$.

Formula:

The median would be:

$(\frac{n+1}{2})th$ observation's value

- For the **even number of observations** the median would be:

$$\frac{(\frac{n}{2})th\ observation + (\frac{n}{2}+1)th\ observation}{2}$$

So, it would be suitable to negotiate at 37.5 lakhs.

There is a **huge difference in the new mean and new median**.

Conclusion:

- The outliers dramatically affect the Mean but the Median remains more robust and closer to the typical value of the dataset.
- Which concludes that **Median is more robust to outliers**

3) Mode

It is the observation with the highest frequency. It is **most occurring data point** in the dataset.

Suppose the data points are recorded as - [90, 90, 90, 80, 90, 70, 95, 90]

- The mode will be **90**.
- Remember, sometimes if there are no data points that repeat, then we can implies that there is no mode

There can also be **more than one mode** in the dataset.

Suppose the data points are recorded as - [2, 2, 3, 3, 4]

- We can call this **Bi-modal** with 2 and 3 as the modes

Weighted Average: Reflecting Importance

- In Weighted Average, each data point is assigned a weight that represents its **importance** or relevance.
- We **multiply each data point by its corresponding weight**, **sum these products**, and **then divide by the total weight**.

Example: Calculating GPA

In real life, a common application of weighted average is calculating Grade Point Average (GPA) for students.

Consider a student's course list for a semester:

SUBJECT	CREDIT	GRADE
Math	3	5
History	4	4
Chemistry	3	5
English	2	3

To calculate the GPA:

1. Calculate the weighted score for each course by multiplying the credit by the numerical grade.
2. Sum up all the weighted scores.
3. Divide the total weighted score by the total credits.

Weighted Average will be:

- **For Math:** $3(CREDIT) * 5(GRADE) = 15$
- **For History:** $4 * 4 = 16$
- **For Chemistry:** $3 * 5 = 15$
- **For English:** $2 * 3 = 6$

$$GPA = \frac{Total\ Weighted\ Score}{Total\ Credits}$$

$$= \frac{52}{17} = 3.05$$

Conclusion:

So, the student's GPA for this semester is 3.05

Measures of Variability

Three common measures of variability are

1. Range
2. Variance
3. Standard Deviation

Let's discuss Range

Range

Range is nothing but **Maximum value - Minimum value**

Suppose the Salaries of some employees in a company are : [30, 30, 35, 40, 40]

- Here, the range of the salary will be **40-30 = 10**

It describes the overall spread of the data that the difference between maximum and minimum values is 10.

> **Q1. What will happen if there is an "Outlier" in the data?**

- Let the salaries be: [30, 30, 35, 40, 40, 300]

  New range will be **300 - 30 = 270**
- As we can see one outlier can destroy the range of the dataset.

We can conclude that **Range of the data is also not robust to the outliers like the Mean**

To solve this issue, statisticians came up with the metric called "**Inter Quartile Range**".

Inter Quartile Range

IQR is the metric that provides a robust way to measure the spread of a dataset.

- The IQR is the range between the first quartile (Q1) and the third quartile (Q3) of a dataset.
- Means, $IQR = Q3 - Q1$

> What is Quartiles?

Quartiles

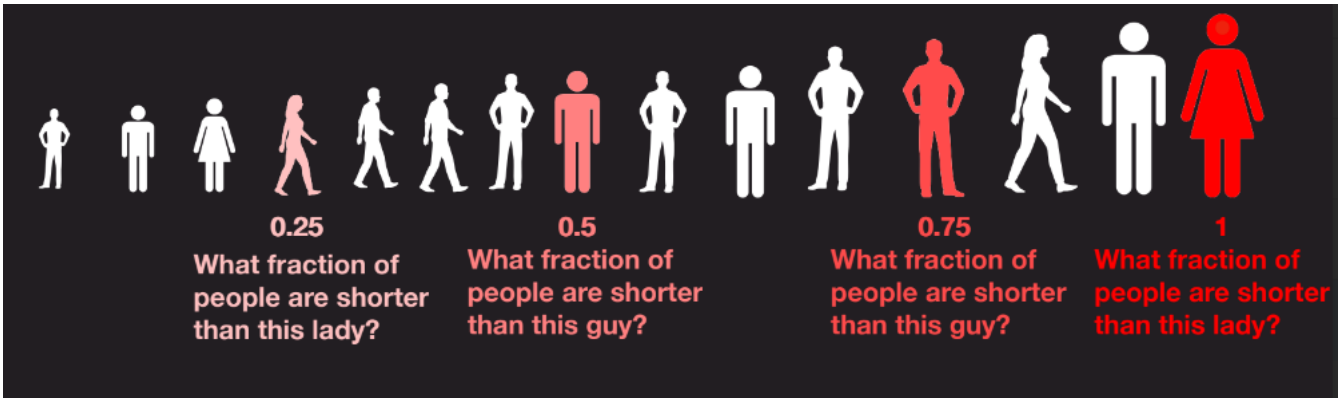
It is the value which divides the dataset into four equal parts.

There are three quartiles, **Q1, Q2, and Q3**.

- Q1 represents the 25th percentile, meaning that 25% of the data falls below this value.
- Q2 is the median and represents the 50th percentile, dividing the data into two equal halves.
- Q3 represents the 75th percentile, meaning that 75% of the data falls below this value.

Suppose we have this data with us,

> **What each values are representing here?**



- 0.25 represents Q1 or 25th percentile:
  - It means that 0.25% of people are shorter than this lady
- 0.5 represents Q2 or 50th percentile:
  - It means that 0.5% or half of the people in the dataset are shorter than this guy
- 0.75 represents Q3 or 75th Percentile:
  - It means that 0.75% of people are shorter than this guy
- Maximum or 1:
  - It means that 100% of people are shorter than this lady or she is the tallest person in the dataset.

> Q.What is percentile?

Percentile

A value that tells us that some **"p%" observations are less than that value**

- Let's say the value occurring at 50 Percentile is 68. We can conclude that 50% of the data is less than 68

One more example:

> Suppose you scored 99 percentile in your 10th boards, what does this mean?

- It indicates that **99% of students scored less marks than you**.

The graphical representation of a dataset's summary statistics, including the median, quartiles, and potential outliers. Box plot comes into the picture

Box Plot

It provides a visual way to understand the distribution and spread of data.

Box:

- The box itself represents the interquartile range (IQR),
  It is divided into two parts, the lower (bottom) quartile (Q1) and the upper (top) quartile (Q3).
- The length of the box is determined by the range between Q1 and Q3.

Line (Median):

- Inside the box, a line or bar is drawn that represents the median, which is the middle value of the dataset when it's ordered.

Whiskers:

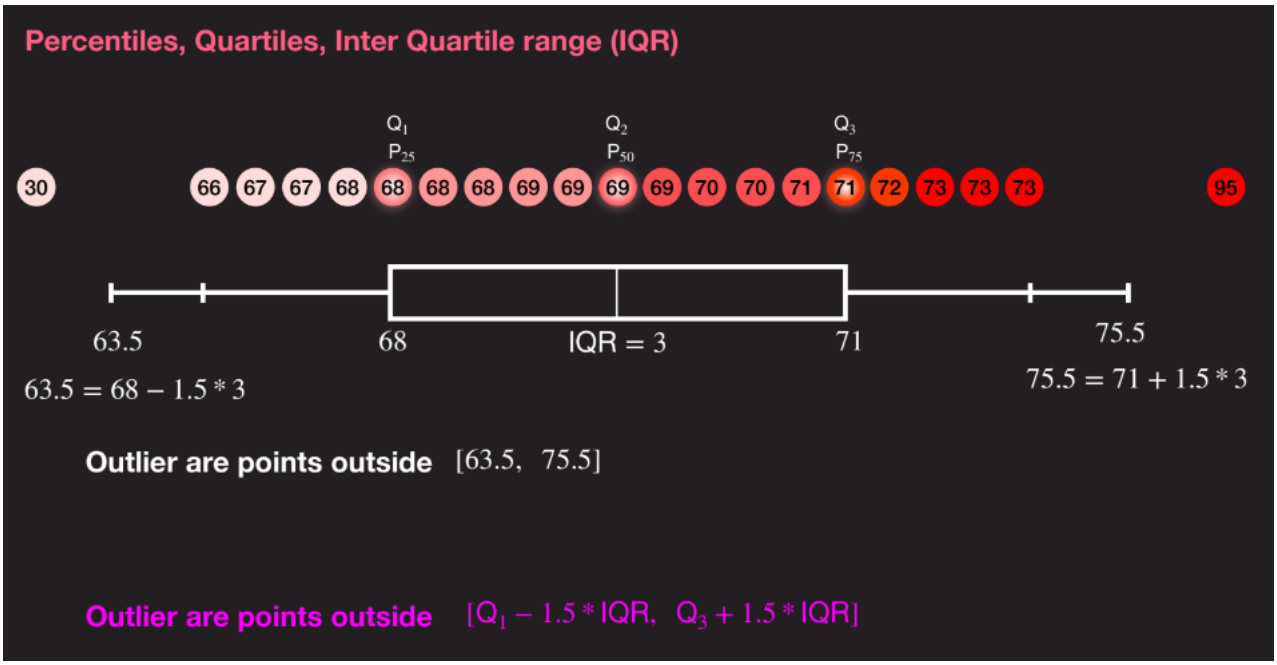
- Two lines, or "whiskers," extend from the box in both directions.
  - The Value of Lower whiskers is determined by Q1 - 1.5(IQR). It is the minimum value of the range
  - The Value of Upper whiskers is determined by Q3 + 1.5(IQR). It is the maximum value of the range

Outliers:

- Data points that fall outside the whiskers are considered outliers.

Example

We have a sorted data, the box plot of the data will look like this



IQR implementation on a real-life dataset

Problem Statement:

When we talk about these two players:

1. Sehwag
2. Rahul Dravid

We all know that Sehwag has **aggressive batting style**

While Rahul Dravid **plays patiently**, with no risk and stands on the crease like a "Wall"

- Let's analyse both of their matches and try to find some insights about their range of scores.
- We will use IQR here to calculate the range of their scores accurately and will also try to find if they have any "Outlier" scores in their careers.

We will conclude that out of these two batsman, who is the more consistent batsman?.

Let's start with Sehwag's matches

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/130/original/sehwag.csv?1684996594 -O sehwag.csv
```

```
--2023-10-31 06:00:12--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/130/original/sehwag.csv?1684
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 18.172.139.94, 18.172.139.46, 18.172.139.210,
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|18.172.139.94|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18584 (18K) [text/plain]
Saving to: 'sehwag.csv'

sehwag.csv          100%[===================>]  18.15K  --.-KB/s    in 0.002s

2023-10-31 06:00:13 (11.0 MB/s) - 'sehwag.csv' saved [18584/18584]
```

```
sehwag = pd.read_csv("sehwag.csv")
sehwag.head()
```

	Runs	Mins	BF	4s	6s	SR	Pos	Dismissal	Inns	Unnamed: 9	Opposition	Ground	Start Date	Unnamed: 13
0	1	5	2	0	0	50.00	7	lbw	1	NaN	v Pakistan	Mohali	1 Apr 1999	ODI # 1427
1	19	18	24	0	1	79.16	6	caught	1	NaN	v Zimbabwe	Rajkot	14 Dec 2000	ODI # 1660
2	58	62	54	8	0	107.40	6	bowled	1	NaN	v Australia	Bengaluru	25 Mar 2001	ODI # 1696
3	2	7	7	0	0	28.57	6	caught	2	NaN	v Zimbabwe	Bulawayo	27 Jun 2001	ODI # 1730
4	11	19	16	1	0	68.75	6	not out	2	NaN	v West Indies	Bulawayo	30 Jun 2001	ODI # 1731

```
sehwag["Runs"].describe()
```

```
count    245.000000
mean      33.767347
std       34.809419
min        0.000000
25%        8.000000
50%       23.000000
75%       46.000000
max      219.000000
Name: Runs, dtype: float64
```

We want to find the range of his scores

> Let's find Quartiles first on the "Runs" column.

So Q1, Q2 and Q3 will be

```
# 25th percentile or Q1
p_25 = np.percentile(sehwag["Runs"], 25)
p_25
```

```
8.0
```

This value indicates that 25% of all the values present in the dataset for Sehwag's run is less than 8

We can also say,

Out of all the matches that Shewag played, in 25% of those matches, he scored less than 8 runs.

```
#50th percentile or Q2, also "Median"
p_50 = np.percentile(sehwag["Runs"], 50)
p_50
```

```
23.0
```

This indicates that in 50% of the matches, he scored less than 23 runs

```
#75th percentile or Q3
p_75 = np.percentile(sehwag["Runs"], 75)
p_75
```

```
46.0
```

This indicates that in 75% of the matches, he scored less than 46 runs

> So, IQR will be?

We know IQR = Q3 - Q1

```
# Inter Quartile Range
iqr_sehwag = p_75 - p_25
iqr_sehwag
```

```
38.0
```

```
normal_range = (sehwag["Runs"].max() - sehwag["Runs"].min())

normal_range
```

```
219
```

We can observe the difference here,

IQR is 38 which means that middle 50% of the data lies in the range of 38.

So more than 50% of the time, Sehwag scores in the range of 38 runs

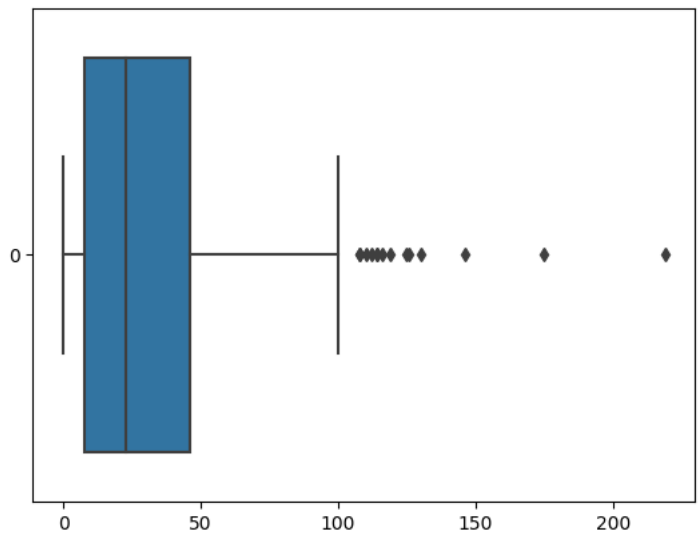
- On the other hand, the **normal range is very high i.e. 219** which is certainly not a good range to consider.
- We can observe one thing that **there in an Outlier present in the data** means in some matches he has scored so many runs like more than 300 in a single match

This is why the range is getting affected by the outlier

Let's plot the box plot to visualise the spread of the data

```
sns.boxplot(data=sehwag["Runs"], orient="h")
```

```
<Axes: >
```



We can see that Q1, Q2, and Q3 values lie within the box and we can also see whiskers on both the sides of box which is the limit.

We already saw how to calculate the lower whisker and upper whisker

All the values outside the limit are considered "Outlier"

```
# upper limit = Q3 + 1.5 * IQR

upper = 46 + 1.5*(iqr_sehwag)
upper
```

```
103.0
```

Here, we cannot have values on the left side of the lower whisker as the batsman cannot score less than 0 runs.

So all the outliers will be present on the right side of the upper whisker

```
# all the values greater than upper is outlier
outliers_sehwag = sehwag[sehwag["Runs"]>upper]
len(outliers_sehwag)
```

```
14
```

```
14/245
```

```
0.05714285714285714
```

Conclusion:

Here we can observe that **5.7% values from the dataset are outliers**.

This means we can conclude that 5.7 or ~6% times Sehwag has scored more than the IQR which is 38 runs

Now let's have a same process into Dravid's stats

Cricket - Dravid

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/131/original/dravid.csv?1684996749 -O dravid.csv
```

```
--2023-10-31 06:00:13--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/131/original/dravid.csv?1684
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 18.172.139.94, 18.172.139.46, 18.172.139.210,
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|18.172.139.94|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24177 (24K) [text/plain]
Saving to: 'dravid.csv'

dravid.csv          100%[===================>]  23.61K  --.-KB/s    in 0.001s

2023-10-31 06:00:13 (32.8 MB/s) - 'dravid.csv' saved [24177/24177]
```

```
dravid = pd.read_csv("dravid.csv")
```

```
dravid["Runs"].describe()
```

```
count    318.000000
mean      34.242138
std       29.681822
min        0.000000
25%       10.000000
50%       26.000000
75%       54.000000
max      153.000000
Name: Runs, dtype: float64
```

```
#25th percentile or Q1
per_25 = np.percentile(dravid["Runs"], 25)
per_25
```

```
10.0
```

This indicates that in 25% of the matches, he scored less than 10 runs

```
#50th percentile or Q2 , also "Median"
per_50 = np.percentile(dravid["Runs"], 50)
per_50
```

```
26.0
```

This indicates that in 50% of the matches, he scored less than 26 runs

```
#75th percentile or Q3
per_75 = np.percentile(dravid["Runs"], 75)
per_75
```

```
54.0
```

This indicates that in 75% of the matches, he scored less than 54 runs

```
# Inter Quartile Range
iqr_dravid = per_75 - per_25
iqr_dravid
```

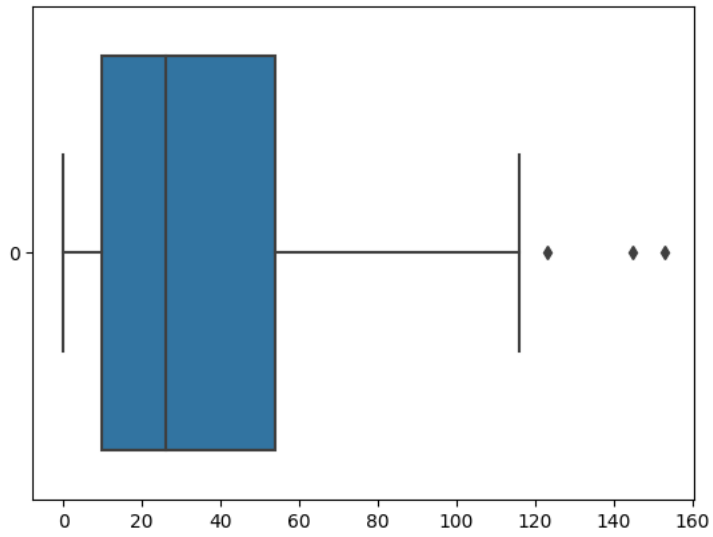
```
44.0
```

```
normal_range = (dravid["Runs"].max() - dravid["Runs"].min())

normal_range
```

```
153
```

```
sns.boxplot(data=dravid["Runs"], orient="h")
```

```
<Axes: >
```




```
# upper limit = Q3 + 1.5 * IQR

upper_dravid = per_75 + 1.5*(iqr_dravid)
upper_dravid
```

```
120.0
```

```
# all the values greater than upper is outlier
outliers_dravid = dravid[dravid["Runs"]>upper_dravid]
len(outliers_dravid)/len(dravid)
```

```
0.009433962264150943
```

```
outliers_dravid['Runs'].shape
```

```
(3,)
```

```
dravid.shape
```

```
(318, 14)
```

Conclusion

Here we can observe that **0.9% values from the dataset are outliers**.

This means we can conclude that 0.9% times Dravid has scored more than the IQR which is 44 runs

So we can conclude that in Sehwag case there is **6% outliers** and in Dravid's case there are only **0.9% outliers**

which shows that "**Dravid was more consistent than Sehwag**"

Random Variable (RV):

A random variable is a situation/event/experiment, for which we are not certain about the outcome.

It is a way to assign numbers to the outcomes of such events.

They can further be divided into 2 types:

- Discrete Random Variable
- Continuous Random Variable

Examples of Discrete Random Variable

Here, we can count the number of possible outcomes.

> **1. Coin Toss**

Let's consider a coin toss.

What are its possible outcomes?

Heads and Tails.

- There is no other possible other than this.

Hence, we can represent its outcomes as a random variable, that can take values: $\{H, T\}$

> **2. Throw of a dice**

- Let's assign a random variable, "X," to represent the outcome of the die roll.
- So, a throw of dice can be represented as: $X = \{1, 2, 3, 4, 5, 6\}$, depending on the outcome of the roll.
- It can not have an outcome lesser than 1, or greater than 6
- Or even, any decimal value between 1 and 2
- Hence it is also discrete RV

Examples of Continuous Random Variable

Here, we cannot count the number of possible outcomes. They are infinite.

> **1. Height of students in a class**

- Suppose the lowest student height in the class is: 4.5 feet
- Suppose the highest student height is: 5.9 feet

Now, we can have students that have height as

- 4.511 feet
- 4.92 feet
- 5.8555 feet

So, we have an infinite number of possible height values between 4.5 and 5.9 feet. We cannot count them

Whereas, we could count the number of possibilities in a coin toss or dice throw.

> **Other examples of Continuous RV can be:**
>
> - Temperature of a room
> - Time taken to complete a task
> - Distance travelled

...etc

Distribution Functions

Probability Density Function (PDF):

- The PDF is a function that describes the probability density of a continuous random variable over its range.
- The term "density" here is similar to how tightly data is packed around a specific point, like cars on a road.

Probability Mass Function (PMF):

- The PMF is a function that describes the probability of a discrete random variable taking on a specific value.

Cumulative Distribution Function (CDF):

- The CDF is a function that gives the probability that a random variable is less than or equal to a specified value.

Let's implement this using a height dataset

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/126/original/weight-height.csv?1684995383 -O weigh
```

```
--2023-10-31 06:00:13--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/126/original/weight-height.
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 18.172.139.94, 18.172.139.46, 18.172.139.210,
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|18.172.139.94|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 428120 (418K) [text/plain]
Saving to: 'weight-height.csv'

weight-height.csv   100%[===================>] 418.09K  --.-KB/s    in 0.06s

2023-10-31 06:00:13 (6.84 MB/s) - 'weight-height.csv' saved [428120/428120]
```

```
df_hw = pd.read_csv("weight-height.csv")
```

```
df_hw.head()
```

	Gender	Height	Weight
0	Male	73.847017	241.893563
1	Male	68.781904	162.310473
2	Male	74.110105	212.740856
3	Male	71.730978	220.042470
4	Male	69.881796	206.349801

```
df_hw.describe()
```

	Height	Weight
count	10000.000000	10000.000000
mean	66.367560	161.440357
std	3.847528	32.108439
min	54.263133	64.700127
25%	63.505620	135.818051
50%	66.318070	161.212928
75%	69.174262	187.169525
max	78.998742	269.989699

We will going to work on the single column for now

```
df_height = df_hw["Height"]
df_height.head()
```

```
0    73.847017
1    68.781904
2    74.110105
3    71.730978
4    69.881796
Name: Height, dtype: float64
```

```
# minimum height
min_height = df_height.min()
min_height
```

```
54.2631333250971
```

```
# maximum height
max_height = df_height.max()
max_height
```

```
78.9987423463896
```

```
total = len(df_height)
total
```

```
10000
```

When we talk about probability, we try to construct the Distribution plots.

Cumulative Distribution Function (CDF), Probability Mass Function (PMF), and Probability Density Function (PDF) are all related to random variables and are used to describe the probability distribution of random variables.

First, let's see what is random variable

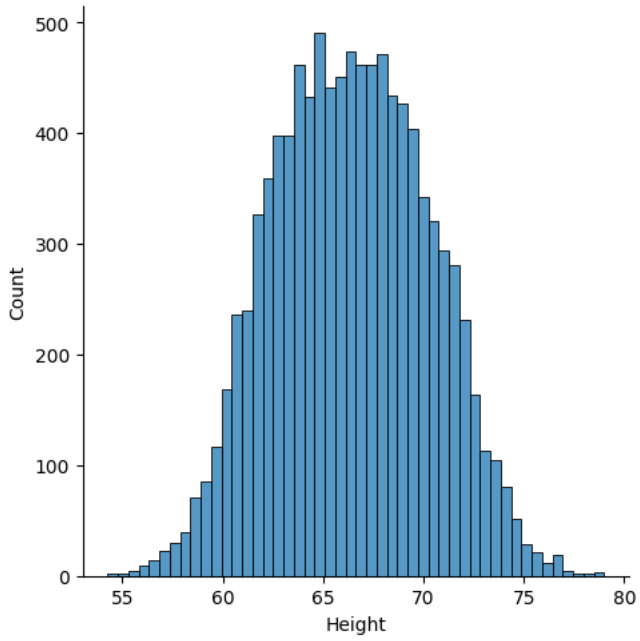
To plot this type of distribution we generally use Histograms or Distribution plots

Histogram

It is a graphical representation of a dataset's distribution, showing the frequency or probability of different values within the data.

```
sns.displot(df_height)
```

```
<seaborn.axisgrid.FacetGrid at 0x7c3b394ab130>
```



> Q.What we can understand from this distribution?

- Each bar in the histogram represents one of the intervals or ranges,
- The height of the bar indicates the frequency or number of data points falling within that interval.

Count:

- It indicates the "**frequency**", which means in the particular bar or range of height, how many values are there.
    - We can observe that, around 500 people have their height in the range of 63 - 65 (that on bar)

This is what histograms or distribution plots tell about the data

1. Probability Mass Function (PMF)

The PMF is a function that describes the probability of a discrete random variable taking on a specific value.

It associates each possible value of the random variable with its probability of occurrence.

Example: Rolling a Fair Six-Sided Die

- If we have a discrete random variable X representing the outcome of rolling a fair six-sided die

Possible outcome is: 1, 2, 3, 4, 5, 6. This is discrete random variable

- The PMF might look like $P(X = 1) = \frac{1}{6}, P(X = 2) = \frac{1}{6}$, and so on.

2. Probability Density Function (PDF)

PDF is used for continuous random variables, as opposed to PMF, which is for discrete variables.

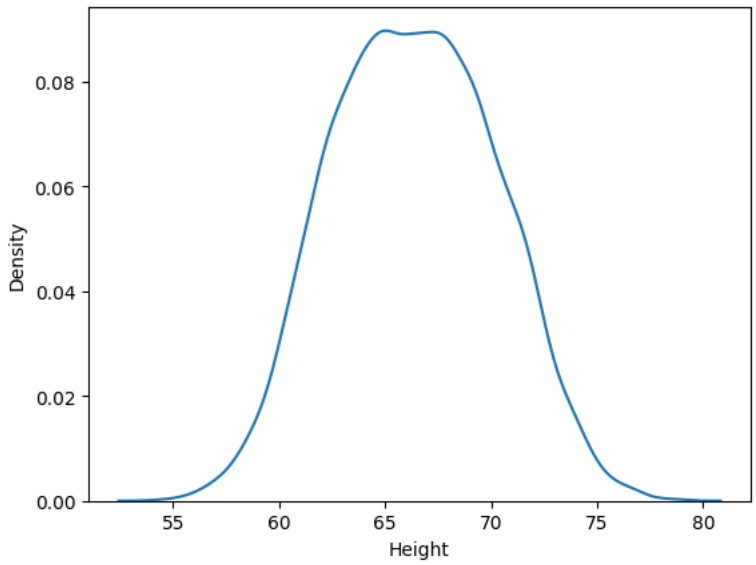
If we want to find the probability of a specific value which is continuous random variable within the given range then we will use PDF.

- It doesn't provide the probability of a specific value but gives us the probability of the RV falling within a certain interval.
- For example, what are the chances that the next height we chose will fall between 62 and 65

We can visualize a PDF by using distribution plots like histograms or KDE (Kernel Density Estimation) plots.

```
sns.kdeplot(df_height)
```

```
<Axes: xlabel='Height', ylabel='Density'>
```



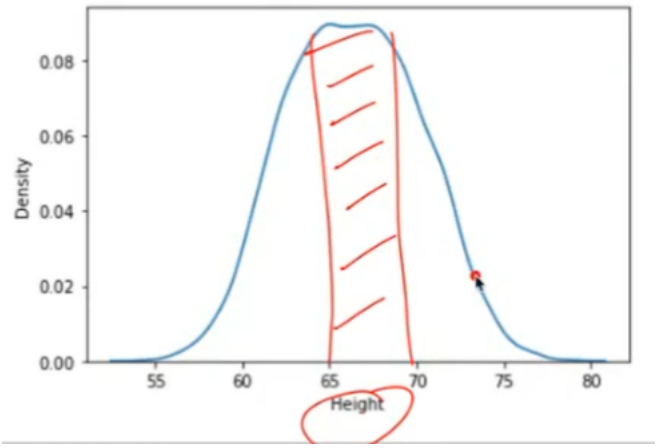
Example:

If we have a continuous random variable Y representing the height of people in a population,

The PDF might represent the probability that a randomly chosen person has a height within a certain range, such as between 65 and 70.

- We will find out the area under that interval to find the probability

```
<AxesSubplot:xlabel='Height', ylabel='Density'>
```



> **Q. Can we find the probability of exact values using PDF?**

- As we know Probability Distribution Function (PDF) works with continuous random variables
- It represents the likelihood of a random variable falling within a particular interval, not at a precise point.
- So, the PDF doesn't assign probabilities to exact values since there are infinitely many possible values within a continuous range.

In other words, we compute the probability that X falls in a range [a, b].

In summary, the PDF represents probabilities as density over an interval rather than exact values

3. Cumulative Distribution Function (CDF)

The Cumulative Distribution Function (CDF) describes the probability that a random variable takes on a value less than or equal to a given value.

In the context of this dataset, in CDF, we talk about fractions of people who are less than the given height

- Let's say we take 60 inches, then what fraction of the people have less than or equal to this value? This fraction is calculated using CDF
- It gives us the cumulative probability up to a certain point.

Example:

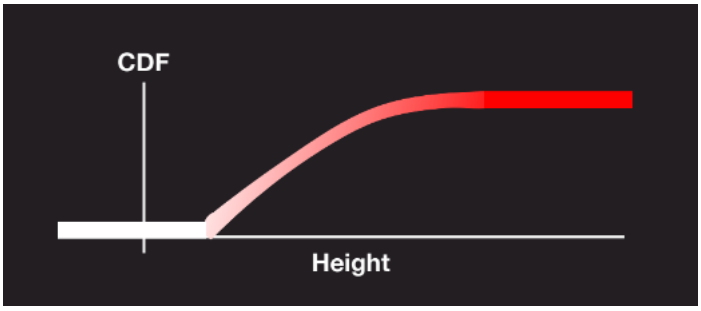
If we have a random variable Z representing the number of heads in three coin tosses,

The CDF would tell us the probability that Z is less than or equal to a certain number, like $P(Z \leq 2)$.

> **How to calculate CDF?**

The **CDF is calculated by accumulating the probabilities for each height value**.

- As we move along the X-axis (height values) on the CDF graph, we're essentially adding up the probabilities
- It shows how likely it is to find someone with a height less than or equal to that value.



- The CDF graph typically starts at 0% on the Y-axis (probability) when height is at its minimum (in our dataset)
- It ends at 100% when height is at its maximum.

- The curve starts at the left and gradually climbs towards the right.
- The steepness of the curve at a particular point represents how quickly the probability is accumulating

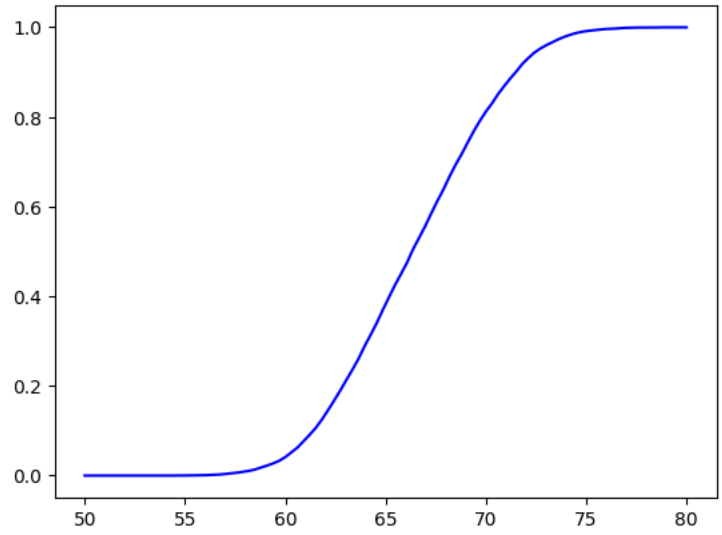
Conclusion

So, the PDF shows us the probability of a specific height, while the CDF shows us the probability of heights up to a certain value in our dataset.

Let's plot the CDF graph for this dataset manually

```
# CDF: Cumulative distribution function

# will take 100 values between the range of 50 and 80 inclusively using np.linpace
x_values = np.linspace(50, 80, 100)

# Will contain fraction of people shorter than x
y_values = []

for x in x_values:

    # find out people shorter than x
    people_shorter_than_x = df_height[df_height <= x]

    # find out number of such people
    num_people_shorter_than_x = len(people_shorter_than_x)

    # How many fraction of people are shorter than x so dividing it by total value
    fraction_people_shorter_than_x = num_people_shorter_than_x / total

    # Appending into the y_values list
    y_values.append(fraction_people_shorter_than_x)

# plotting the CDF
plt.plot(x_values, y_values, c="b")
```

```
[<matplotlib.lines.Line2D at 0x7c3b3c1d0490>]
```



- This Curve is called "**Cumulative Distribution Function**".
- It is a function which takes the x value and returns the y value
  - $f(x) = y$

It is the inverse of the percentile means

- **Percentile will take input as 25 and give output as 63.5** means
  - 25% of the people are shorter than 63.5
- While **CDF will take input as 63.5 and give output as 0.25** means
  - if we want to find how many people are having height less than or equal to 63.5 i.e. 25% of people.

Conclusion :

In summary, the relationships are as follows:

- The PMF is used for discrete random variables.
- The PDF is used for continuous random variables.
- The CDF is used for both discrete and continuous random variables to provide cumulative probabilities.

These functions are essential tools in probability and statistics for describing and understanding the behaviour of random variables.

Variance

Variance, measures the spread or dispersion of the values of a random variable around its mean.

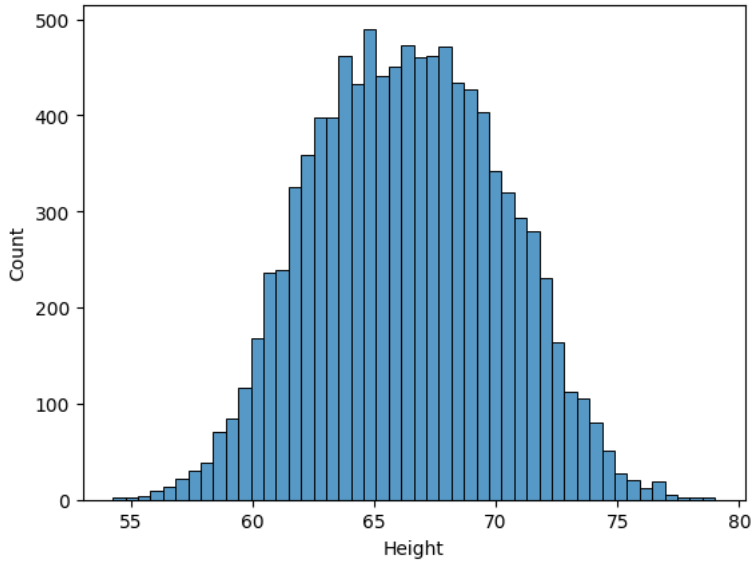
It quantifies how much **individual values deviate from the mean**.

- A **higher variance** indicates that the values are more spread out from the mean.
- While a **lower variance** suggests that the values are closer to the mean.

We can plot the histogram to visualise the spread or distribution of the data

```
sns.histplot(df_height)
```

```
<Axes: xlabel='Height', ylabel='Count'>
```



Let's explore another way to measure error

Defining Error:

$$Error = (Actual\ Height - Guessed\ Height)^2$$

> **Q1. Now, to minimize this error, what's the best approach?**

In our Height Guessing Game, we've seen that aiming for the mean (μ) height is the key. Means, Guessed height should be the mean value.

- $Error = (H_1 - \mu)^2$ (guessing for 1 time)
- It is also known as **Mean Squared Error**

Imagine we're playing the game 10 times, guessing the mean height each time:

Error1 = $(H1 - \mu)^2$

Error2 = $(H2 - \mu)^2$

Error3 = $(H3 - \mu)^2$

...

Error10 = $(H10 - \mu)^2$

To find the overall error, we can sum up these individual errors and then divide by the number of guesses, which gives us the variance:

Variance Calculation:

Variance = (Error1 + Error2 + Error3 + ... + Error10) / 10

- $Variance = \frac{(H_1-\mu)^2+(H_2-\mu)^2+(H_3-\mu)^2+.....+(H_{10}-\mu)}{10}^2$

> **Q2. So, if the variance is low, what does that mean?**

It implies that most of our guesses are incredibly accurate.

In general, variance quantifies how spread out, the data values are from the average (mean) value.

It assesses the average squared difference between data points and the mean.

The formula for calculating variance for n data points is:

Variance Calculation Formula:

$$variance = \sigma^2 = \frac{\sum_{i=1}^{n}(H_i - \mu)^2}{n}$$

- σ2 is the population variance.
- H_i is the ith data point.
- μ is the population mean.
- n is the number of data points in the population.

Now that we have a clear understanding of variance and how it measures the spread or dispersion of data points,

Let's look into another essential concept closely related to variance. It's called "**Standard Deviation**"

Standard Deviation

Let's introduce an even more practical and commonly used statistic - the "Standard Deviation."

While variance quantifies the dispersion of data, standard deviation is derived from variance and offers a more interpretable measure.

- The standard deviation represents how much individual data points deviate from the mean or average value.
- It gives us a clear sense of the typical or expected amount of variation in our dataset.
    - In simple words, it represents that how far is our data point from the mean (μ)

Standard Deviation Formula:

The standard deviation, can be calculated by taking the square root of the variance:

$$SD = \sqrt{variance}$$

- $$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(H_i - \mu)^2}{n}}$$

Interpretation:

- A lower standard deviation signifies that data points tend to be close to the mean, indicating **less variability**.
- Conversely, a higher standard deviation indicates greater data dispersion, suggesting **more variability** within the dataset.

Content

- Empirical vs Theoretical Probability
- Expectations
- Binomial Distribution
- Bernoulli Distribution

Case study on Empirical vs Theoretical Probability

Casino Case Study

Problem Statement:

- A bag has **3 Red** and **2 Blue** balls.

  We pick a ball, write its colour, and **put it back** in the bag. This is done **4 times** in total.

  If all 4 times, the **Red balls** was drawn, we **win Rs 150**.

  Otherwise we **lose Rs 10**.

> **Question : Would engaging in this game result in a profit or loss for us?**

Whether we end up gaining or losing will depend on how many red balls are drawn.

How many number of red balls can we expect to be drawn?

Let's define a random variable X to denote the number of red balls drawn

> **1. What are the possible outcomes of X?**

X can be equal to $0, 1, 2, 3$ or 4

Note that X is a discrete random variable.

Empirical Approach

Let's try to estimate whether we will have a profit or loss after playing this game, using probabilities.

For this, we will simulate this situation in Python code.

```
import math as m
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

There is an important function that we need to know about

np.random.choice()

Suppose we provide a list of possible outcomes to this function, for a simple event, say, a coin toss

It has 2 possible outcomes:

- Head
- Tails

This function will randomly choose one of these possible outcomes, with an equal probability and return it.

```
np.random.choice(['H', 'T'])
```

```
'T'
```

> **2. What do you think is the probability of getting Heads or getting a Tails?**

It should be 0.5

Now imagine, we modify the input list of possible outcomes to `[H, T, T]`

```
np.random.choice(["H","T","T"])
```

```
'T'
```

> **3. What will be the probability of getting heads or getting tails now?**

Now, P(Heads) should be: $\frac{1}{3}$

And, P(Tails) should be: $\frac{2}{3}$

> **4. What if we want to toss this coin twice?**

We have a paramter `size` that we can use define the size of the random outcomes from the passed list.

```
np.random.choice(["H","T","T"],size=2)
```

```
array(['T', 'H'], dtype='<U1')
```

So here,

- It has chosen this first `H` randomly from `[H, T, T]`
- And then it has again randomly chosen the second `T` from `[H, T, T]`

Now that we've understood `np.random.choice()`

> **How can we simulate the given Casino problem?**

Since we have,

- 3 red balls and
- 2 blue balls

We can represent the possible outcomes as a list `["R", "R", "R", "B", "B"]`

```
np.random.choice(["R","R","R","B","B"])
```

```
'B'
```

Since the ball is being drawn 4 times, we set `size = 4`

Now since we'd like to have a count of number of red balls, let's store the random result in a variable `rolls`

```
#Code to be shared to learners
rolls=np.random.choice(["R","R","R","B","B"],size=4)
rolls
```

```
array(['R', 'B', 'B', 'R'], dtype='<U1')
```

We know that it is possible to get 7 heads on 10 coin tosses, when using a fair coin.

So how would one go about proving that P(Heads) = 0.5 for a fair coin?

This process of simulating the experiment, and repeating it multiple times, is done in an effort to calculate probability value (of getting heads in this example).

This value is known as **Empirical Probability**.

- The idea is make estimates **using real-world data/observations**

np.count_nonzero()

There is another function in numpy **`np.count_nonzero()`**

It will return the no of non-zero elements in the list we provide to it.

```
np.count_nonzero([2, 1, 0, 0, 0, 3, 4, 5])
```

```
5
```

This function can be used in another way, for our purposes.

Here in this case study, we need to count the number of red ball drawns so we will create a boolean mask of red ball

> **1. What will `rolls == "R"` return?**

```
rolls == "R"
```

```
array([ True, False, False,  True])
```

This means that anything other than `"R"` will become `False`.

This function will treat `False` as `0`, if the passed list contains booleans.

So we can use this mask, to count the number of red balls in our list `rolls`

```
#Code to be shared to learners
np.count_nonzero(rolls=="R")
```

```
2
```

- We already know how to simulate a ball draw
- Let's store the no of reds observed in a variable `num_red`
- And store this value for all 10,000 simulations into a list `red_values`

```
red_values=[]

for person in range(10000):
    rolls=np.random.choice(["R","R","R","B","B"],size=4)
    num_red=np.count_nonzero(rolls=="R")
    red_values.append(num_red)
pd.value_counts(red_values)
```

```
3    3552
2    3394
1    1496
4    1281
0     277
dtype: int64
```

```
# red_values
```

Let's do a `.value_counts()` to see the frequency of values values it contains.

```
pd.value_counts(red_values,normalize=True)
```

```
3    0.3552
2    0.3394
1    0.1496
4    0.1281
0    0.0277
dtype: float64
```

We are aware that passing `normalize=True` in `value_counts()` gives us the result in percentage of their occurence.

We can see that the probability of drawing 3 red balls is 0.3552, 2 red balls is 0.3394 and so on..

> **Based on this data, how many red balls we will get on an average based on simulations we have done 10,000 times?**

```
# This is empirical value
np.mean(red_values)
```

```
2.4064
```

To obtain theoretical value, we should perform the simulations numerous times.

- With an increasing number of trials, the result approaches the theoretical value, though an exact match may not be achieved.

To prove this claim let's look into one simple example of coin toss.

Experiment : Coin Toss

We know, theoretically the probability of getting head and tails in a coin toss is 0.5

Let's try to prove it:

```
pd.value_counts(np.random.choice(["H", "T"], size=10), normalize=True)
```

```
T    0.8
H    0.2
dtype: float64
```

We can see that if we perform the simulations for limited trials, result is not matching theoretical value.

Let's run the simulation for numerous times

```
pd.value_counts(np.random.choice(["H", "T"], size=1000000), normalize=True)
```

```
H    0.500142
T    0.499858
dtype: float64
```

As predicted, we are getting values much closer to the theoretical result of 0.5

Expectation using Empirical Approach

We got the following value counts

```
pd.value_counts(red_values)
```

```
3    3552
2    3394
1    1496
4    1281
0     277
dtype: int64
```

And these values yielded us a mean of `2.4064`

```
np.mean(red_values)
```

```
2.4064
```

> **How do we think, this mean was calculated from these frequency values?**

As we learnt in the last class, this was calculated as a result of **Weighted Average**

So, for the given frequency count, we can see that this is calculated as:

$$Mean = \frac{4(1281)+3(3552)+2(3394)+1(1496)+0(277)}{1281+3552+3394+1496+277} = \frac{4(1281)+3(3552)+2(3394)+1(1496)+0(277)}{10000}$$

```
(4*(1281) + 3*(3552) + 2*(3394) + 1*(1496) + 0*(277)) / (10000)
```

```
2.4064
```

Now that we've verified this,

Let's represent the same equation in a slightly different format.

$$Mean = 4\frac{1281}{10000} + 3\frac{3552}{10000} + 2\frac{3394}{10000} + 1\frac{1496}{10000} + 0\frac{277}{10000}$$

If we closely look at the value counts table, we will see that this can be represented as the following formula:

$$E(X) = \Sigma_i X_i * P(X = X_i)$$

where

- X was our random variable that denotes the no of red balls drawn.
- $P(X = X_i)$ represents the probability of X getting a value of X_i
- $E(X)$ is known as the **Expected value** of the random variable X

Let's define it formally:

`Expectation of a random variable X, is the weighted average of the values that X takes, with the weights being the probabilities.`

Until now, we simulated the event 10,000 times, and found an expected value of random variable X using the data observed.

This is known as the **Empirical Approach** of solving the problem.

Theoretical Approach

Now, let's solve this case study using theoretical approach and observe the difference in the result

Let's look at the problem statement once more.

Problem Statement:

A bag has **3 Red** and **2 Blue** balls.

we pick a ball, write its colour, and **put it back** in the bag. This is done **4 Times** in total.

If all 4 times, the **Red balls** was drawn, we **win Rs 150**.

Otherwise we **lose Rs 10**.

> **Question: Would engaging in this game result in a profit or loss for us?**

Let's define 2 events:

- R: Drawing a red ball
- B: Drawing a blue ball

> **1. What would be the probability of obtaining a red ball once?**

$P(R) = \frac{3}{5}$

Similarly, we know that $P(B) = \frac{2}{5}$

> **2. What is the probability of drawing a red ball twice?**

$P(RR) = \frac{3}{5} * \frac{3}{5}$

> **3. What is the probability of drawing a red ball followed by a blue ball?**

$P(RB) = \frac{3}{5} * \frac{2}{5}$

These values are easy to evaluate when we are drawing the balls just twice.

In our case study, we are **drawing it 4 times**. Let's consider that case.

Like before, we define X as a random variable that denotes the no of red balls drawn.

> **4. What would be the probability of obtaining 1 red ball?**

For $X = 1$, we can have 4 possible cases as drawn below:

- BBBR
- BBRB
- BRBB
- RBBB

Let's look at the probability value of each of these individual cases:

- Case 1: $\frac{2}{5} * \frac{2}{5} * \frac{2}{5} * \frac{3}{5}$
- Case 2: $\frac{2}{5} * \frac{2}{5} * \frac{3}{5} * \frac{2}{5}$

and so on

So we can see that for all these 4 cases, we can write their probability as: $(\frac{2}{5})^3 * (\frac{3}{5})^1$

Since there are 4 such cases, we write the total probability of $X = 1$ as:

$P(X = 1)$ = case 1 OR case 2 OR case 3 OR case 4

$P(X = 1) = 4 * (\frac{2}{5})^3 * (\frac{3}{5})^1$

> **5. What would be the probability of getting 2 red balls out of the 4 balls drawn?**

Let's look at the different orientations possible for $X = 2$.

- We have 6 possibilities.

Let's look at the probability of each of these orientations:

- Case 1: $\frac{2}{5} * \frac{2}{5} * \frac{3}{5} * \frac{3}{5}$

... and so on

So, at the end of the day, we know that probability for each of these individual cases would be: $(\frac{2}{5})^2 * (\frac{3}{5})^2$

Since either of these 6 cases are possible, the total probability becomes:

$$P(X = 2) = 6 * (\frac{2}{5})^2 * (\frac{3}{5})^2$$

Conclusion

> **1. Can we write this 4 and 6 in a different format?**

Recall the combinatorics lecture.

We know that

- $4 = {}^4C_1$
- $6 = {}^4C_2$

> **2. With this in mind, when we take a look at the results of $P(X = 1)$ and $P(X = 2)$, can we derive some general expression?**

So, $P(X = k) = {}^4C_k \ (\frac{3}{5})^k \ (\frac{2}{5})^{4-k}$

Notice that here, `4` is nothing but the no of times a ball was drawn from the bag, i.e. **no of trials**

We can use this derived equation to find probability for all valid values of the random variable X:

- $P(X = 0) = {}^4C_0 \ (\frac{3}{5})^0 \ (\frac{2}{5})^4$
- $P(X = 1) = {}^4C_1 \ (\frac{3}{5})^1 \ (\frac{2}{5})^3$
- $P(X = 2) = {}^4C_2 \ (\frac{3}{5})^2 \ (\frac{2}{5})^2$
- $P(X = 3) = {}^4C_3 \ (\frac{3}{5})^3 \ (\frac{2}{5})^1$
- $P(X = 4) = {}^4C_4 \ (\frac{3}{5})^4 \ (\frac{2}{5})^0$

Now we've understood this in theory, but

> **1. How can we compute this in code?**

We will use built-in functions of the `math.comb()` library.

```
import math
```

> **2. How will we find the value of ${}^4C_0$?**

```
math.comb(4, 0)
```

```
1
```

As we can see this gave us the result of $\frac{4!}{0!*(4-0)!}$

Similarly, we can find ${}^4C_1$ as:

```
math.comb(4, 1)
```

```
4
```

Let's evaluate the probability values $P(X)$ for all possible values of $X = \{0, 1, 2, 3, 4\}$

```
# P(X=0)
math.comb(4,0)* (3/5)**0 * (2/5)**4
```

```
0.025600000000000005
```

```
# P(X=1)
math.comb(4,1)* (3/5)**1 * (2/5)**3
```

```
0.15360000000000004
```

```
# P(X=2)
math.comb(4,2)* (3/5)**2 * (2/5)**2
```

```
0.3456000000000001
```

```
# P(X=3)
math.comb(4,3)* (3/5)**3 * (2/5)**1
```

```
0.34559999999999996
```

```
# P(X=4)
math.comb(4,4)* (3/5)**4 * (2/5)**0
```

```
0.1296
```

Let's compare these probability results to what we evaluated through the Empirical approach

Notice that these values are very close.

As discussed earlier, **if we increase the no of simulations, the observed result would be more and more closer to these theoretical values**.

Hence, proved.

```
[ ]  pd.value_counts(red_values,normalize=True)
```

```
3    0.3552
2    0.3394
1    0.1496
4    0.1281
0    0.0277
dtype: float64
```

Binomial Distribution

Binomial distribution is a **discrete probability distribution** of the number of successes in `n` **independent** experiments sequence.

A Binomial trial will always have **two possible outcomes**:

- Success / Win
- Failure / Loss

We defined a **discrete random variable** X that denoted number of red balls drawn.

- Note that the event of drawing a ball is independent.
- X will be called a **Binomial RV**

Also, we were given some parameters in our problem, let's define them:

- n: No of independent trials
  - In our example, we draw balls 4 times, hence $n = 4$
- p: Probability of success in one trial
  - In our example, this denotes the probability of drawing a red ball, hence $p = \frac{3}{5}$
  - Therefore, $(1 - p)$ becomes the probability of failure in each trial (i.e. drawing a blue ball, in this example)

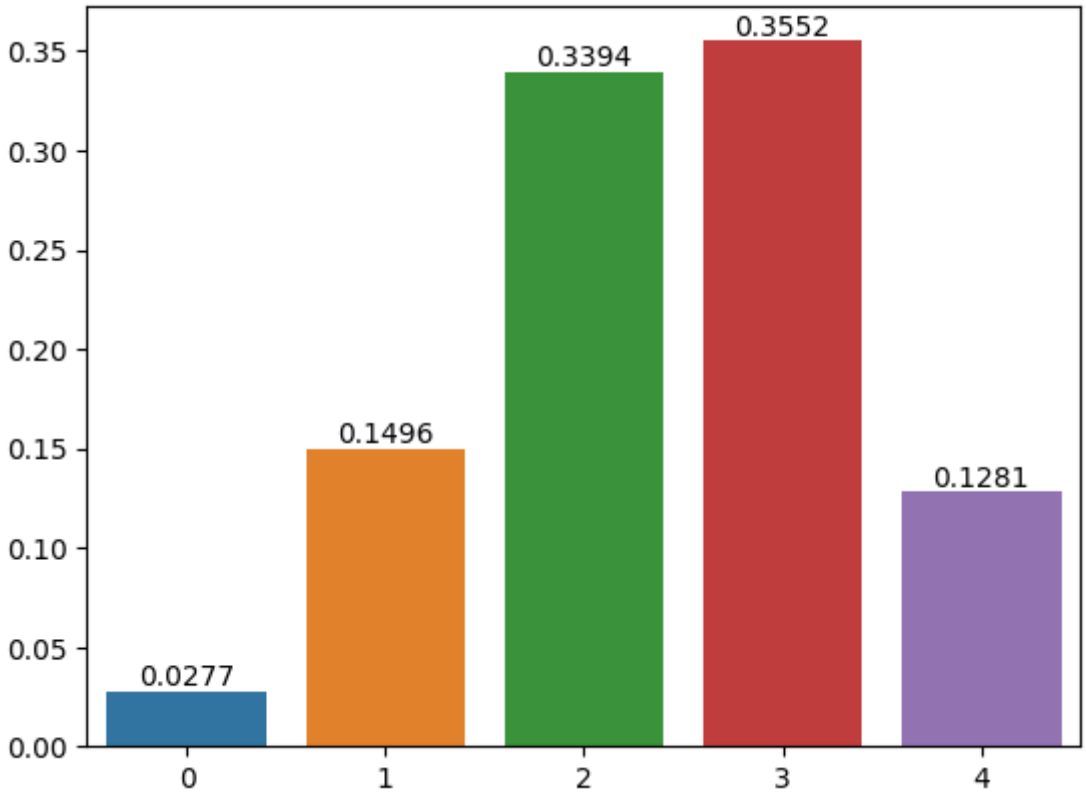
Using these parameters, we can re-write the equation we derived in general form:
$P(X = k) = {}^{n}C_{k}\ (p)^{k}\ (1 - p)^{n-k}$

Let's plot our calculated values to see what Binomial distribution looks like.

```
x = pd.value_counts(red_values, normalize=True)
x
```

```
3    0.3552
2    0.3394
1    0.1496
4    0.1281
0    0.0277
dtype: float64
```

```
ax = sns.barplot(x = x.index, y = x.values)

for i in ax.containers:
  ax.bar_label(i,)
```



This is the **Probabillity Mass Function (PMF)** of our given Binomial experiment, which is called as Binomial Probability Distribution

- The graph shows the probability of obtaining each possible number of successes (k) in n trials.
- The height of each bar represents the probability of that particular outcome.
- The sum of all the probabilities equals 1.

The `scipy.stats.binom` library gives us a built in function that eases the calculation of PMF values (i.e. value of $P(X)$ for specific values of X).

Instead of using the formula $P(X = k) = {}^{n}C_{k}\ (p)^{k}\ (1-p)^{n-k}$, we can directly use this function to calculate the PMF value.

We just need to specify the 3 parameters:

- n
- k
- p

```
from scipy.stats import binom
```

```
prob_0_red = binom.pmf(n=4,p=3/5,k=0)
prob_0_red
```

```
0.025599999999999994
```

```
prob_1_red = binom.pmf(n=4,p=3/5,k=1)
prob_1_red
```

```
0.15359999999999996
```

```
prob_2_red = binom.pmf(n=4,p=3/5,k=2)
prob_2_red
```

```
0.3456
```

```
prob_3_red = binom.pmf(n=4,p=3/5,k=3)
prob_3_red
```

```
0.3456000000000001
```

```
prob_4_red = binom.pmf(n=4,p=3/5,k=4)
prob_4_red
```

```
0.1296
```

Notice that these values are the same as what we calculated using `math.comb`

Expectation using theoretical approach

> ### How will we calculate the theoretical expectation value?

We know the formula: $E(X) = \Sigma_i X_i P(X = X_i)$

- Here, we saw that we can calculate the probability values using `scipy.stats.binom`
- And that random variable $X = \{0, 1, 2, 3, 4\}$

```
expectation_theoretical= (0*prob_0_red) + (1*prob_1_red) + (2*prob_2_red) + (3*pr
expectation_theoretical
```

```
2.4000000000000004
```

Note that this is close to the **Empirical Expected value** we calculated.

Alternately, there is a built-in function to find this expected value in `stats.binom`

Here, we need to pass the following arguments to `args`:

- n, and
- p

```
binom.expect(args=(4,3/5))
```

```
2.4000000000000004
```

Let's define another random variable Y that denotes the amount of money won/lost through gambling.

- Therefore, possible values of Y : $\{150, -10\}$

Let's create a table for this random variable Y, with it's posibble values and corresponding probabilities.

- Case of winning Rs 150 ($Y = 150$)
  - $P(Y = 150)$ would be the same as $P(X = 4)$
- Case of loosing Rs 10 ($Y = -10$)
  - $P(Y = -10) = P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3) = 1 - P(X = 4)$

```
# P(Y=150)
prob_4_red
```

```
0.1296
```

```
#P(Y = -10)
1 - prob_4_red
```

```
0.8704000000000001
```

> **What would be expected value of Y?**

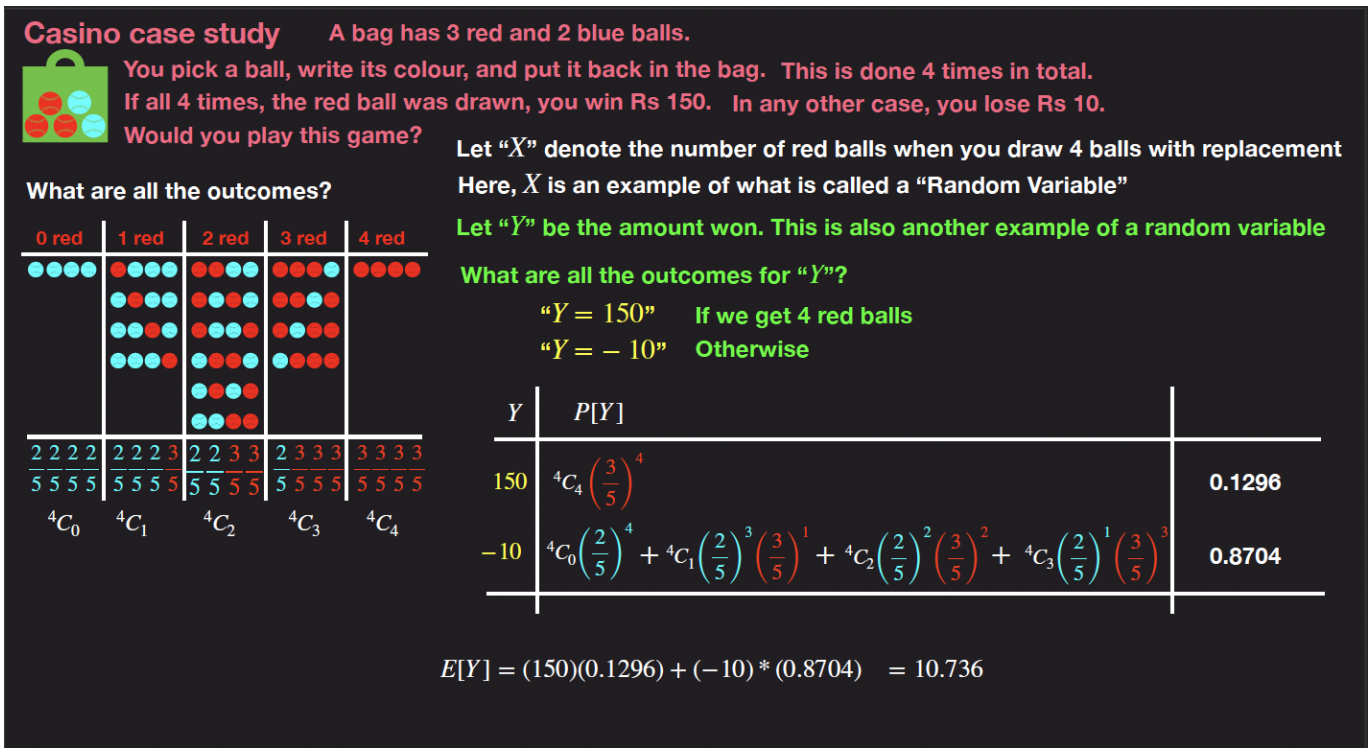
$$E(Y) = \Sigma_i Y_i P(Y = Y_i) = (150 * 0.1296) + (-10 * 0.8704000000000001)$$

```
expected_y = (150*0.1296) + (-10*0.8704000000000001)
expected_y
```

```
10.735999999999997
```

Conclusion of the case study:

This value means that if we play many many times, at the end of the day, **we are expected to have profit of Rs 10.736**



Conditions of Binomial Experiment

1. The experiment must consist of a **fixed number of trials (n)**, with only 2 possible outcomes: Success or Failure

2. Individual trials are **identical and independent**.

3. The random variable denotes the number of success in `n` trials.

Bernoulli Trials

Essentially, it is the **special case** of Binomial trial, where `n = 1`

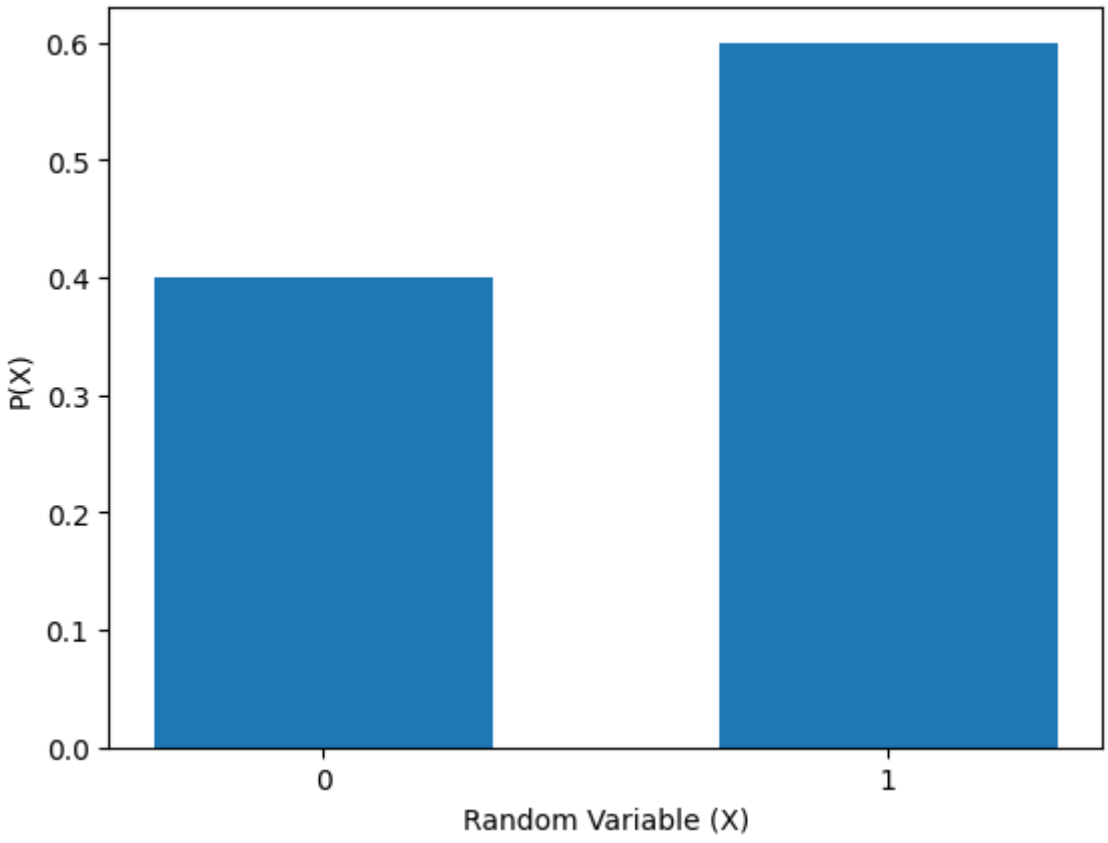
Hence, it must also follow the condition that there must be only 2 possible outcomes:

- Success, or
- Failure

Let's plot this, to see what **Bernoulli distribution** looks like.

```
x = [0, 1]
y = [2/5, 3/5]

plt.bar(x, y, width=0.6, tick_label=["0", "1"])
plt.xlabel("Random Variable (X)")
plt.ylabel("P(X)")
```

```
Text(0, 0.5, 'P(X)')
```



> **To summarize, what is the difference between Binomial and Bernoulli distribution?**

- Bernoulli deals with the outcome of the single trial of the event, whereas Binomial deals with the outcome of the multiple trials of the single event.
- Hence, we can define **Binomial distribution** in another way:

  `It is the collection of Bernoulli trials for the same event, i.e., it contains more than 1 Bernoulli event for the same scenario for which the Bernoulli trial is calculated.`

Dice Example

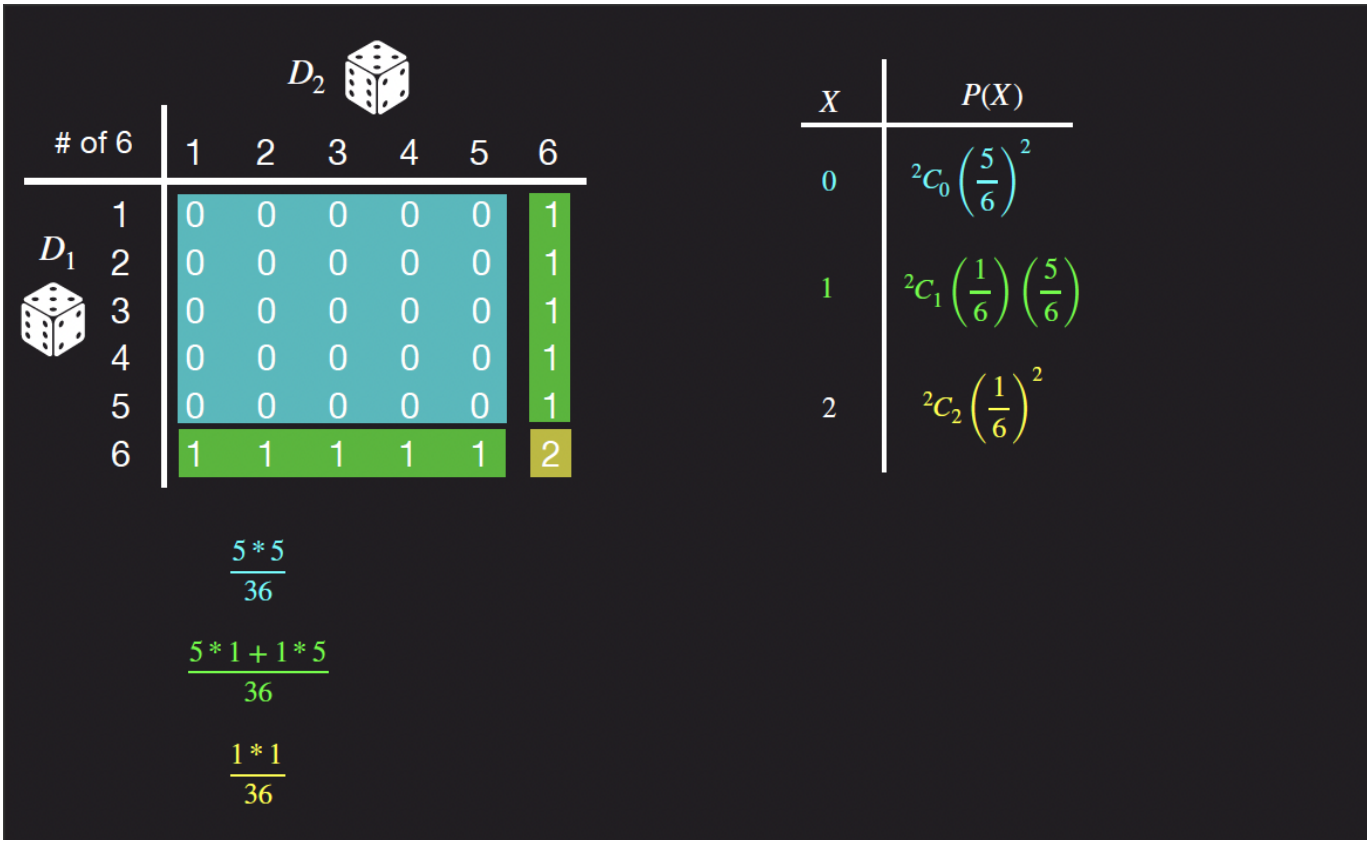
```
You toss 2 dice. If both dice are 6, you get Rs 2. Else, if one dice is 6, you get Rs 1. Otherwise, you do not get anything.
```

Let's define a random variable X that represents the amount of money won.

- Hence, it can take the values: $X = \{0, 1, 2\}$

Answer the following questions.

> **What is the probability of getting the following?**
>
> - Rs 0
> - Rs 1
> - Rs 2



> **Finding $P(X = 0)$**

From the table we can see that we will get 0 Rs for 25 outcomes, hence $P(X = 0) = \frac{25}{36}$

> **Finding $P(X = 1)$**

From the table, $P(X = 1) = \frac{10}{36}$

> **Finding $P(X = 2)$**

From the table, $P(X = 2) = \frac{1}{36}$

lets see if we can obtain the same answers using the Binomial formula

Before we get to solving, let's define the parameters:

- **What will be the value of `n`?**
  - Since we are throwing 2 dice, $n = 2$
- **What will be the value of `p`?**
  - `p` is defined as the probability of succes in one trial
  - So how do we define success here?

- Obtaining a 6
- Therefore, p = probability of getting a 6 in a single dice roll, i.e. $p = \frac{1}{6}$

We know the Binomial formula is: $P(X = k) = {}^{n}C_{k}\ (p)^{k}\ (1-p)^{n-k}$

Therefore,

- $P(X = 0) = {}^{2}C_{0}\ (\frac{1}{6})^{0}\ (\frac{5}{6})^{2} = 1 * 1 * \frac{25}{36} = \frac{25}{36}$
- $P(X = 1) = {}^{2}C_{1}\ (\frac{1}{6})^{1}\ (\frac{5}{6})^{1} = 2 * \frac{1}{6} * \frac{5}{6} = \frac{10}{36}$
- $P(X = 2) = {}^{2}C_{2}\ (\frac{1}{6})^{2}\ (\frac{5}{6})^{0} = 1 * \frac{1}{36} * 1 = \frac{1}{36}$

These are the exact answers we got using the table above!!

Alternately, we could've evaluated the binomial formula using code as:

```
binom.pmf(n=2,p=1/6,k=0)
```

```
0.6944444444444443
```

Now answer the second question.

> **What is the expected value of money won?**

We can find this using the formula: $E(X) = \Sigma_{i} X_{i} P(X = X_{i})$

$= (0 * \frac{25}{36}) + (1 * \frac{10}{36}) + (2 * \frac{1}{36})$

$= \frac{1}{3}$

Alternately, we can use the `stats.binom.expect()` function

```
binom.expect(args=(2,1/6))
```

```
0.33333333333333326
```
