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:** $P[A|B] = \dfrac{P[A \cap B]}{P[B]}$
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:** $P[A|B] = \dfrac{P[B|A] * P[B]}{P[A]}$
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:

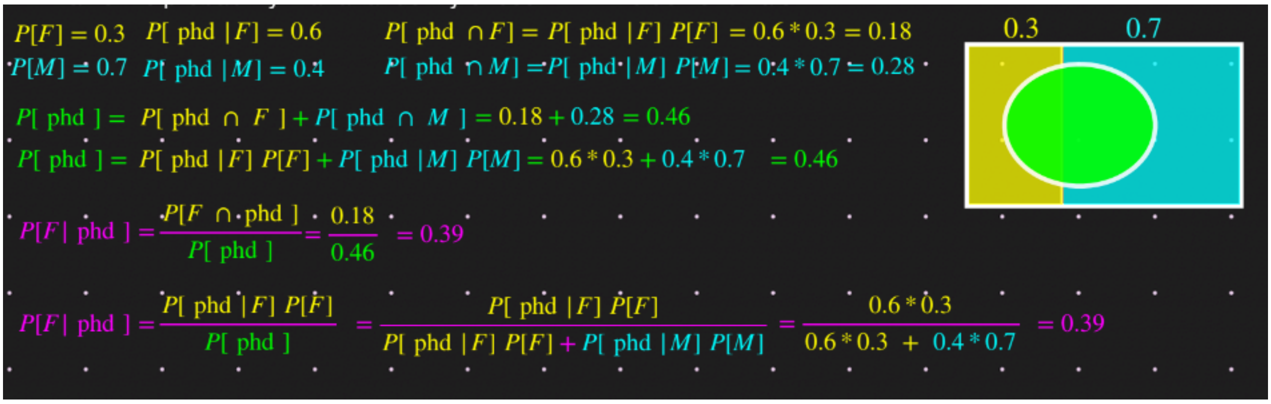
- $P(F \mid phd) = \dfrac{P(phd \mid F) * P(F)}{[P(phd \mid F) * P(F)] + [P(phd \mid M) * P(M)]}$
  - $P(F \mid phd) = \frac{0.6*0.3}{[0.6*0.3]+[0.4*0.7]}$
  - $P(F \mid phd) = 0.39$

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.



$P[F] = 0.3 \quad P[\text{ phd } | F] = 0.6 \qquad P[\text{ phd } \cap F] = P[\text{ phd } | F]\ P[F] = 0.6 * 0.3 = 0.18$

$P[M] = 0.7 \quad P[\text{ phd } | M] = 0.4 \qquad P[\text{ phd } \cap M] = P[\text{ phd } | M]\ P[M] = 0.4 * 0.7 = 0.28$

$P[\text{ phd }] = P[\text{ phd } \cap F] + P[\text{ phd } \cap M] = 0.18 + 0.28 = 0.46$

$P[\text{ phd }] = P[\text{ phd } | F]\ P[F] + P[\text{ phd } | M]\ P[M] = 0.6 * 0.3 + 0.4 * 0.7 = 0.46$

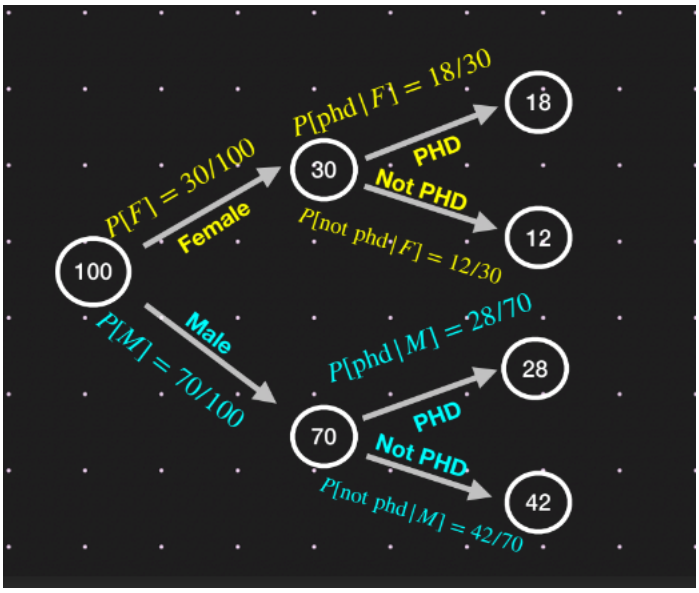
$P[F | \text{ phd }] = \frac{P[F \cap \text{phd }]}{P[\text{ phd }]} = \frac{0.18}{0.46} = 0.39$

$P[F | \text{ phd }] = \frac{P[\text{ phd } | F]\ P[F]}{P[\text{ phd }]} = \frac{P[\text{ phd } | F]\ P[F]}{P[\text{ phd } | F]\ P[F] + P[\text{ phd } | M]\ P[M]} = \frac{0.6 * 0.3}{0.6 * 0.3 + 0.4 * 0.7} = 0.39$

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

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.

```
!gdown 1Mp2bQl5QJ6O2tcezb0ceBQIn8vW5us0N
```

```
Downloading...
From: https://drive.google.com/uc?id=1Mp2bQl5QJ6O2tcezb0ceBQIn8vW5us0N
To: /content/kerala.csv
100% 10.3k/10.3k [00:00<00:00, 25.4MB/s]
```

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

```
# Read the data
df = pd.read_csv("/content/kerala.csv")
df.head(10)
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5
5	KERALA	1906	26.7	7.4	9.9	59.4	160.8	414.9	954.2	442.8	131.2	251.7
6	KERALA	1907	18.8	4.8	55.7	170.8	101.4	770.9	760.4	981.5	225.0	309.7
7	KERALA	1908	8.0	20.8	38.2	102.9	142.6	592.6	902.2	352.9	175.9	253.3
8	KERALA	1909	54.1	11.8	61.3	93.8	473.2	704.7	782.3	258.0	195.4	212.1
9	KERALA	1910	2.7	25.7	23.3	124.5	148.8	680.0	484.1	473.8	248.6	356.6

```
df.shape
```

```
(118, 16)
```

> **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
```

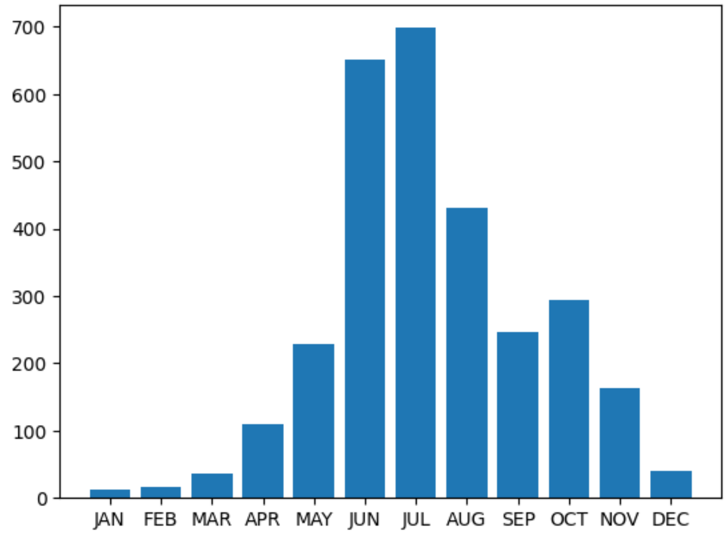
```
JAN     12.218644
FEB     15.633898
MAR     36.670339
APR    110.330508
MAY    228.644915
JUN    651.617797
JUL    698.220339
AUG    430.369492
SEP    246.207627
OCT    293.207627
NOV    162.311017
DEC     40.009322
dtype: float64
```

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.

We can observe **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	
count	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000
mean	1959.500000	12.218644	15.633898	36.670339	110.330508	228.644915	651.617
std	34.207699	15.473766	16.406290	30.063862	44.633452	147.548778	186.181
min	1901.000000	0.000000	0.000000	0.100000	13.100000	53.400000	196.800
25%	1930.250000	2.175000	4.700000	18.100000	74.350000	125.050000	535.550
50%	1959.500000	5.800000	8.350000	28.400000	110.400000	184.600000	625.600
75%	1988.750000	18.175000	21.400000	49.825000	136.450000	264.875000	786.975
max	2018.000000	83.500000	79.000000	217.200000	238.000000	738.800000	1098.200

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

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()
```

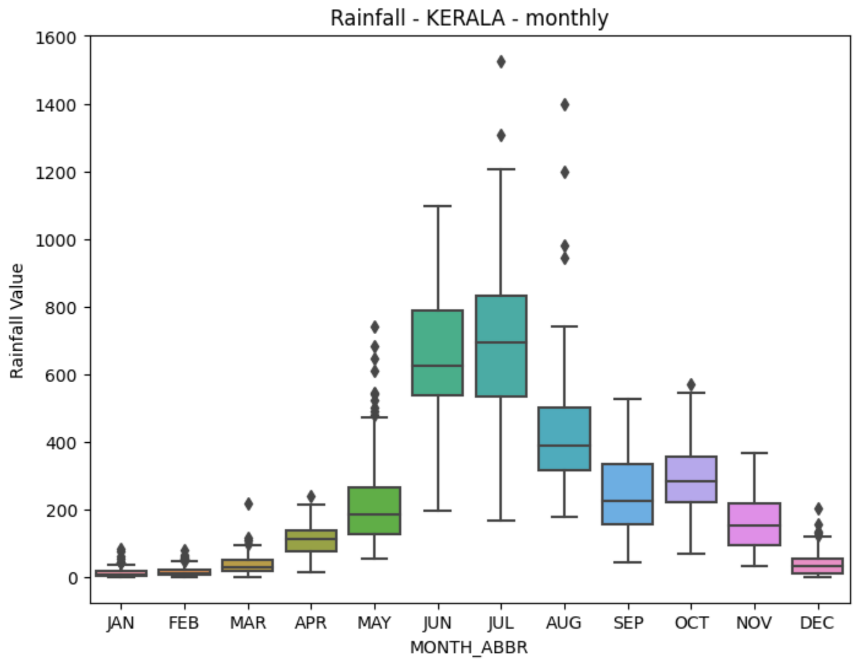
	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
0	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4
1	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5
2	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0
3	1904	23.7	3.0	32.2	71.5	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()
```



Conclusion:

- 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 we can notice there is an extra space in the start of column "Annual rainfall". It is like this: ' ANNUAL RAINFALL'

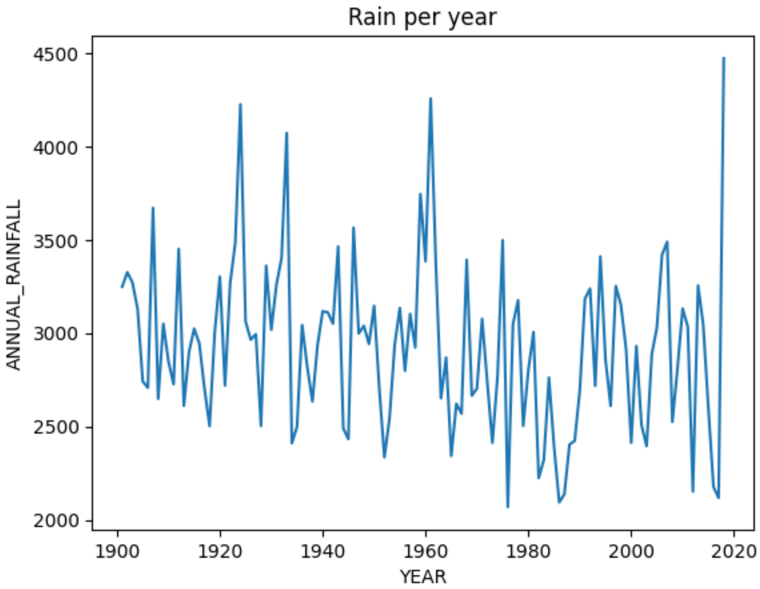
Let's rename this column

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

df.head()
```

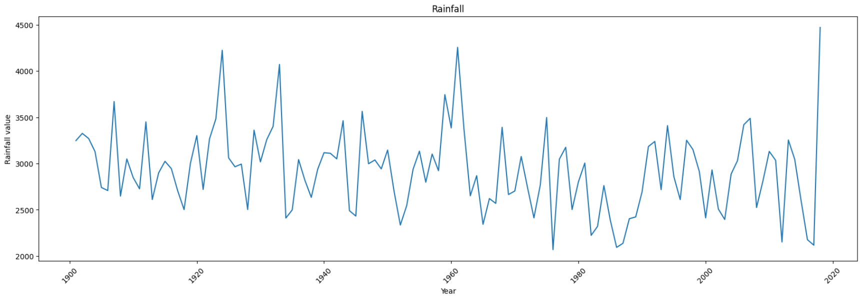
	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5

```
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-106-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/s
  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

```
# if all these condition will satisty then there is a possibility of FLood. So it'll show "YES" in FLOODS column

# If any of the 'and' conditions didn't satisy then FLOODS column will show "NO"

mask = ((data['JUN']==1) | (data['JUL']==1) | (data['OCT']==1) & (data['ANNUAL_RAINFALL']==1))

mask
```

```
0       True
1       True
2       True
3       True
4       True
       ...
113     True
114    False
115    False
116    False
117     True
Length: 118, dtype: bool
```

```
data = data[mask]

data.head()
```

	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
```

```
(90, 6)
```

```
# data[data['JUN'] == 1].value_counts()
```

```
# data[data['JUL'] == 1].value_counts()
```

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	17	13	30
1	16	44	60
Total	33	57	90

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.

`JUN` is representing the rows

- There are 60 records with "1" (indicating rainfaill index exceed threshold)
- There are 30 records with "0" (indicating rainfaill index is less than threshold)

Here, `FLOODS` is represeting the column

- There are 57 records with "YES" (indicating floods)
- There are 33 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.

Let's solve some questions.

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	17	13	30
1	16	44	60
Total	33	57	90

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)/(17+13+16+44)

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

P_F_and_J = 44/(17+13+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) : 0.6666666666666666
P(F AND J) : 0.4888888888888889
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 us 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	0.566667	0.433333
1	0.266667	0.733333
All	0.366667	0.633333

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	0.419355	0.580645
1	0.338983	0.661017
All	0.366667	0.633333

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	0.393939	0.315789	0.344444
1	0.606061	0.684211	0.655556

Conclusion:

The probability that high rainfall occurred in July (JUL = 1) given flooding (FLOODS = YES) is **~0.68**.

- **This means that when there is flooding, there is a 68% 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	0.000000	1.000000
	1	0.586207	0.413793
1	0	0.433333	0.566667
	1	0.100000	0.900000
All		0.366667	0.633333

Conclusion

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

There is **90%** chance of flood given that heavy rainfall in both june and july