KURUNJI VENKATRAMANA GOWDA POLYTECHNIC SULLIA-574327

5TH SEMESTER

AI/ML WEEK-4

Week: 4

Probability:

Probability Definition

Probability is the measure of the likelihood of an event to occur. In the case of events, we can't predict with total certainty. We can only predict the cancer of an event to occur i.e. how likely it is to happen. Probability ranges between 0 to 1 in which 0 indicates the event to be an impossible one and 1 indicates a certain event.

Conditional Probability Definition

Conditional probability is defined as **the likelihood of an event or outcome occurring, based on the occurrence of a previous event or outcome**. Conditional probability is calculated by multiplying the probability of the preceding event by the updated probability of the succeeding, or conditional, event.

Conditional Probability Formula

Mathematically this can be represented as,

Where P(A|B) represents the probability of occurrence of A given B has occurred.

N(A ∩ B) is the number of elements common to both A and B.

N(B) is the number of elements in B and it cannot be equal to zero.

P(A|B) = N(A∩B)/N(B)

Let N represent the total number of elements in the sample space.

⇒P(A|B)

=

N(A∩B)NN(B)N

Since N(A ∩ B)/N and N(B)/N denotes the ratio of the number of favourable outcomes to the total number of outcomes, therefore, it indicates the probability.

Therefore, N(A ∩ B)/N can be written as P(A ∩ B) and N(B)/N as P(B).

⇒ P(A|B) = P(A ∩ B)/P(B)

Therefore, $P(A \cap B) = P(B)\ P(A|B)$ if $P(B) \neq 0$

$= P(A)\ P(B|A)$ if $P(A) \neq 0$

Similarly, the probability of occurrence of B when A has already occurred is

given by, P(B|A) = P(B ∩ A)/P(A)

Problem 1

You toss a fair coin three times:

a. What is the probability of three heads, HHHHHH?
b. What is the probability that you observe exactly one heads?
c. Given that you have observed *at least* one heads, what is the probability that you observe at least two heads?

- **Solution** o We assume that the coin tosses are independent.

a. P(HHH)=P(H)·P(H)·P(H)=0.53=18P(HHH)=P(H)·P(H)·P(H) =0.53=18.
b. To find the probability of exactly one heads, we can write

P(One heads)P(One heads)
=P(HTTUTHTUTTH)=P(HTTUTHTUTTH)

=P(HTT)+P(THT)+P(TTH)=P(HTT)+P(THT)+P(TTH)

=18+18+18=18+18+18

=38=38.

c.
d. Given that you have observed *at least* one heads, what is the probability that you observe at least two heads? Let A1A1 be the event that you observe at least one heads, and

A2A2 be the event that you observe at least two heads.

Then

A_1=S−{TTT}, and P(A_1)=78;A1=S−{TTT}, and P(A1)=78;

A_2={HHT,HTH,THH,HHH}, and

P(A_2)=48.A2={HHT,HTH, THH,HHH}, and P(A2)=48.

Thus, we can write

P(A_2|A_1)P(A2|A1) =P(A_2∩A_1)P(A_1)=P(A2∩A1)P(A1)

$=P(A_2)P(A_1)=P(A2)P(A1)$

$=48.87=47=48.87=47.$

Problem 2:-

A box contains three coins: two regular coins and one fake two-headed coin ($P(H)=1P(H)=1$),

- You pick a coin at random and toss it. What is the probability that it lands heads up?
- You pick a coin at random and toss it, and get heads. What is the probability that it is the two-headed coin?

- **Solution** ^{o} This is another typical problem for which the law of total probability is useful. Let C_1C1 be the event that you choose a regular coin, and let C_2C2 be the event that you choose the twoheaded coin. Note that C_1C1 and C_2C2 form a partition of the sample space. We already know that

$$P(H|C_1)=0.5,P(H|C1)=0.5, \quad P(H|C_2)=1.P(H|C2)=1.$$

a. Thus, we can use the law of total probability to write

$$P(H)P(H) =P(H|C_1)P(C_1)+P(H|C_2)P(C_2)=P(H|C1)P(C1)+P(H|C2)P(C2)$$

$$=12.23+1.13=12.23+1.13$$

$$=23=23.$$

b.

c. Now, for the second part of the problem, we are interested in $P(C_2|H)P(C2|H)$. We use Bayes' rule

$$P(C_2|H)P(C2|H) =P(H|C_2)P(C_2)P(H)=P(H|C2)P(C2)P(H)$$

$$=1._{1323}=1.1323$$

$$=12=12.$$

Joint Probability Definition

In simple words it is the likelihood of a certain event. A statistical measure that calculates the likelihood of two events occurring together and at the same point in time is called Joint probability.

Let A and B be the two events, joint probability is the probability of event B occurring at the same time that event A occurs.

Formula for Joint Probability

Notation to represent the joint probability can take a few different forms. The following formula represents the joint probability of events with intersection.

P (AB) where,

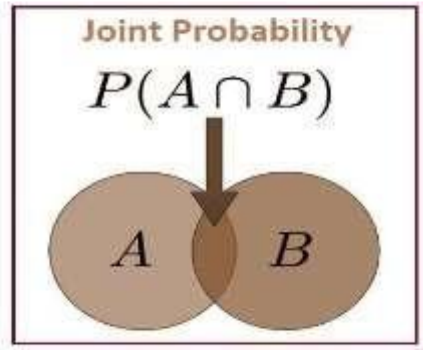
A, B= Two events

- The formula for joint probability is given by P (A ∩ B) = P (A) * P (B) Here, P (A ⋂ B) is the joint probability of events A and B.

P(A and B),P(AB)=The joint probability of A and B

The symbol "∩" in a joint probability is called an intersection. The probability of event A and event B happening is the same thing as the point where A and B

intersect. Hence, the joint probability is also called the intersection of two or more events. We can represent this relation using a Venn diagram as shown below.



Joint Probability Examples

Example: Find the probability that the number three will occur twice when two dice are rolled at the same time.

Solution:

Number of possible outcomes when a

die is rolled = 6 i.e. {1, 2, 3, 4, 5, 6}

Let A be the event of occurring 3 on first die and B be the event of occurring 3 on the second die.

Both the dice have six possible outcomes, the probability of a three occurring on each die is 1/6.

P(A) =1/6

P(B)=1/6

P(A,B) = 1/6 x 1/6 = 1/36

Bayes' theorem:

"Bayes' Theorem states that the conditional probability of an event, based on the occurrence of another event, is equal to the likelihood of the second event given the first event multiplied by the probability of the first event".

theorem describing how the conditional probability of each of a set of possible causes for a given observed outcome can be computed from knowledge of the probability of each cause and the conditional probability of the outcome of each cause.

- Bayes theorem is used to determine conditional probability.
- When two events A and B are independent, $P(A|B) = P(A)$ and $P(B|A) = P(B)$
- Conditional probability can be calculated using the Bayes theorem for continuous random variables.

Bayes Theorem Formula for Events

The formula for events derived from the definition of conditional probability is:

$P(A|B)=P(B|A)P(A)P(B),P(B)\neq0$

Where:

- $P(A|B)$ – the probability of event A occurring, given event B has occurred
- $P(B|A)$ – the probability of event B occurring, given event A has occurred
- $P(A)$ – the probability of event A
- $P(B)$ – the probability of event B

Example 1: Amy has two bags. Bag I has 7 red and 2 blue balls and bag II has 5 red and 9 blue balls. Amy draws a ball at random and it turns out to be red. Determine the probability that the ball was from the bag I using the Bayes theorem.

Solution: Let X and Y be the events that the ball is from the bag I and bag II, respectively. Assume A to be the event of drawing a red ball. We know that the probability of choosing a bag for drawing a ball is 1/2, that is, P(X) = P(Y) = 1/2

Since there are 7 red balls out of a total of 11 balls in the bag I,

therefore, P(drawing a red ball from the bag I) = P(A|X) = 7/11

Similarly, P(drawing a red ball from bag II) = P(A|Y) = 5/14

We need to determine the value of P(the ball drawn is from the bag I given that it is a red ball), that is, P(X|A). To determine this we will use Bayes Theorem. Using Bayes theorem, we have the following:

P(X|A)=P(A|X)P(X)P(A|X)P(X)+P(A|Y)P(Y)P(X|A)=P(A|X)P(X)P(
A
|X)P(X)+P(A|Y)P(Y)

= [((7/11)(1/2))/(7/11)(1/2)+(5/14)(1/2)]

= 0.64

Answer: Hence, the probability that the ball is drawn is from bag I is 0.64

Example 2: A man is known to speak the truth 3/4 times. He draws a card and reports it is king. Find the probability that it is actually a king.

Solution:

Let E be the event that the man reports that king is drawn from the pack of cards

A be the event that the king is drawn

B be the event that the king is not drawn.

Then we have P(A) = probability that king is drawn = 1/4

P(B) = probability that king is drawn = 3/4

P(E/A) = Probability that the man says the truth that king is drawn when actually king is drawn = P(truth) = 3/4

P(E/B)= Probability that the man lies that king is drawn when actually king is drawn = P(lie) = 1/4

Then according to Bayes theorem, the probability that it is actually a king = P(A/E)

=P(A)P(E|A)P(A)P(E|A)+P(B)P(E|B)P(A)P(E|A)P(A)P(E|A)+P(B)P(E|B)

= [1/4 × 3/4] ÷[(1/4 × 3/4) + (1/4 × 3/4)]

= 3/16 ÷12/16

= 3/16 × 16/12

=1/2 = 0.5

Answer: Thus the probability that the drawn card is actually a king = 0.5

Probability Distributions:

--Discreate

--Continuous

--Central Limit Theorem

Discreate Probability;

A discrete probability distribution **counts occurrences that have countable or finite outcomes**. This is in contrast to a continuous distribution, where outcomes can fall anywhere on a continuum. Common examples of discrete distribution include the binomial, Poisson, and Bernoulli distributions.

A **discrete probability distribution** is the probability distribution of a random variable that can take on only a countable number of values

It is given by X G(p). The formula for the pmf is given as follows: $\mathbf{P(X = x) = (1 - p)^{x} p}$, where p is the success probability of the trial. Poisson distribution is a discrete probability distribution that is widely used in the field of finance.

What is a discrete probability function?

A discrete probability function is **a function that can take a discrete number of values (not necessarily finite)**. This is most often the non-negative integers or some subset of the non-negative integers.

Problem1:

I roll two dice and observe two numbers XX and YY.

a. Find RX,RYRX,RY and the PMFs of XX and YY.

b. Find P(X=2,Y=6)P(X=2,Y=6).

c. Find P(X>3|Y=2)P(X>3|Y=2).

d. Let Z=X+YZ=X+Y. Find the range and PMF of ZZ.

e. Find P(X=4|Z=8)P(X=4|Z=8).

- **Solution**

o

a. We have RX=RY={1,2,3,4,5,6}RX=RY={1,2,3,4,5,6}.

Assuming the dice are fair, all values are equally likely so PX(k)={160for

k=1,2,3,4,5,6otherwisePX(k)={16for k=1,2
,3,4,5,60otherwise

Similarly for YY,

PY(k)={160for k=1,2,3,4,5,6otherwisePY(k)={16for k=1,2
,3,4,5,60otherwise

b. Since XX and YY are independent random variables, we can write

P(X=2,Y=6)P(X=2,Y=6)
=P(X=2)P(Y=6)=P(X=2)P(Y=6)

=16 16=136=1616=136.

c.

d. Since XX and YY are independent, knowing the value of YY does not impact the probabilities for XX,

$P(X>3|Y=2)$P(X>3|Y=2) $=P(X>3)$=P(X>3)

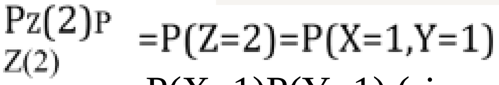
$=P_X(4)+P_X(5)+P_X(6)$=PX(4)+PX(5)+PX(6)

=16+16+16=12=16+16+16=12.

e.

f. First, we have

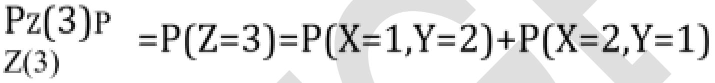
$R_Z=\{2,3,4,...,12\}$RZ={2,3,4,...,12}. Thus, we need to find $P_Z(k)$PZ(k) for $k=2,3,...,12$k=2,3,...,12. We have


$P_Z(2)$P Z(2) =P(Z=2)=P(X=1,Y=1)

=P(Z=2)=P(X=1,Y=1)

=P(X=1)P(Y=1) (since X and Y are independent)=P(X=1)P(Y=1) (since X and Y are independent)

. .
=1616=136=1616=136;


$P_Z(3)$P Z(3) =P(Z=3)=P(X=1,Y=2)+P(X=2,Y=1)

=P(Z=3)=P(X=1,Y=2)+P(X=2,Y=1)

=P(X=1)P(Y=2)+P(X=2)P(Y=1)=P(X=1)P(Y=2)+P(X=2)P(Y=1)

=16.16+16.16=118=1616+16.16=118;

$P_Z(4)$P
=P(Z=4)=P(X=1,Y=3)+P(X=2,Y=2)+P(X=3,Y=1)=P(Z=4)=P(X=1,Y=3 Z(4)) +P(X=2,Y=2)+P(X=3,Y=1) =3.136=112=3.136=112.

g.

We can continue similarly:

$P_Z(5)$PZ(5) =436=19=436=19;

$P_Z(6)$PZ(6) =536=536;

$P_Z(7)$PZ(7) =636=16=636=16;

$P_Z(8)$PZ(8) =536=536;

$P_Z(9)$PZ(9) =436=19=436=19; $P_Z(10)$PZ(10)
=336=112=336=112; $P_Z(11)$PZ(11)
=236=118=236=118;

$P_Z(12)$PZ(12) =136=136.

h.

It is always a good idea to check our answers by verifying that $\sum_{z\in R_Z}P_Z(z)=1$∑z∈RZPZ(z)=1. Here, we have

$\sum_{z \in R_Z} P_Z(z)$∑z∈RZPZ(z)

$=_{136}+_{236}+_{336}+_{436}+_{536}+_{636}$=136+236+336+436+536+636

$+_{536}+_{436}+_{336}+_{236}+_{136}$+536+436+336+236+136

$=1$=1.

i.

j. Note that here we cannot argue that XX and ZZ are independent. Indeed, ZZ seems to completely depend on XX, $Z=X+Y$Z=X+Y. To find the conditional probability $P(X=4|Z=8)$P(X=4|Z=8), we use the formula for conditional probability

$P(X=4|Z=8)$P(X=4|Z=8) =P(X=4,Z=8)P(Z=8)=P(X=4,Z=8)P(Z=8)

=P(X=4,Y=4)P(Z=8)=P(X=4,Y=4)P(Z=8)

=P(X=4)P(Y=4)P(Z=8) (since X and Y are independent)=P(X=4)P(Y=4)P(Z=8) (since X and Y are independent)

$_{16 \cdot 16536}$16·16536

$=_{15}$=15.

Problem2:

The number of customers arriving at a grocery store is a Poisson random variable. On average $10$10 customers arrive per hour. Let XX be the number of customers arriving from $10am$10am to $11{:}30am$11:30am. What is $P(10<X\leq 15)$P(10<X≤15)?

- **S olution**
  - We are looking at an interval of length $1.5$1.5 hours, so the number of customers in this interval is X $\text{Poisson}(\lambda=1.5\times 10=15)$X Poisson(λ=1.5×10=15). Thus,

$P(10<X\leq15)^{P(1}$

$=\sum 15k=11 P_X(k)=\sum k=1115PX(k)\ 0<X\leq15)$

$=\sum 15k=11e_{-15}15_k k!=\sum k=1115e-1515kk!$

$=e_{-15}[15_{11}11!+15_{12}12!+15_{13}13!+15_{14}14!+15_{15}15!]=e-15[151111!+15121$
$2!+151313!+151414!+151515!]$

$=0.4496$

Continues probability:

A continuous distribution **describes the probabilities of a continuous ransdom variable's possible values**. A continuous random variable has an infinite and uncountable set of possible values (known as the range). The mapping of time can be considered as an example of the continuous probability distribution. s

A continuous distribution describes the probabilities of a continuous random variable's possible values. A continuous random variable has an infinite and uncountable set of possible values (known as the range). The mapping of time can be considered as an example of the continuous probability distribution. It can be from 1 second to 1 billion seconds, and so on.

Continuous probability distribution: A probability distribution in which the random variable X can take on any value (is continuous). Because there are infinite values that X could assume, the probability of X taking on any one specific value is zero. Therefore we often speak in ranges of values (**p(X>0) = . 50**).

Problem1 :

Let XX be a random variable with PDF given by

$fx(x)=\{cx_2 0|x|\leq1 otherwise fX(x)=\{cx2|x|\leq10otherwise$

a. Find the constant Cc.

b. Find EXEX and Var(X)(X).

c. Find P(X≥12)P(X≥12).

• **Solution**

o

a. To find Cc, we can use ∫∞−∞fX(u)du=1∫−∞∞fX(u)du=1:

11

=∫∞−∞fX(u)du=∫−

∞∞fX(u)du

=∫1−1cu2du=∫−

11cu2du

=23C.=23c.

b.

Thus, we must have C=32c=32.

c. To find EXEX, we can write

EXEX =∫1−1ufX(u)du=∫−11ufX(u)du

=32∫1−1u3du=3

=0.=0. 2∫−11u3du

d.

In fact, we could have guessed EX=0EX=0 because the PDF is symmetric around X=0x=0. To find Var(X)(X), we have

Var(X)Var(X)

=EX2−(EX)2=EX2=EX2−(EX)2=EX2

=∫1−1u2fX(u)du=∫−11u2fX(u)du

=32∫1−1u4du=32∫−11u4du

=35.=35.

e.

f. To find $P(X\geq 12)$P(X≥12), we can write

$P(X\geq 12)=32\int_{1\,12}x2dx=716.$P(X≥12)=32∫121x2dx=716.

Problem 2 :

Let XX be a continuous random variable with PDF given by $fx(x)=12e_{-|x|}$,for all x∈ R.fX(x)=12e−|x|,for all x∈R.

If $Y=X_2$Y=X2, find the CDF of YY.

- **Solution**
  - First, we note that $R_Y=[0,\infty)$RY=[0,∞). For y $[0,\infty)$y [0,∞), we have

$F_Y(y)$FY(y) $=P(Y\leq y)$=P(Y≤y)

$=P(X_2\leq y)$=P(X2≤y)

$=P(-y\surd\leq X\leq y\surd)$=P(−y≤X≤y)

$=\int_{y\surd-y\surd}12e_{-|x|}dx$=∫−yy12e−|x|dx

$=\int_{y\surd 0}e_{-x}dx$=∫0ye−xdx $=1-e_{-y\surd}$.=1−e−y.

- Thus,
  - $F_Y(y)=\{1-e_{-y\surd}0y\geq 0$otherwise

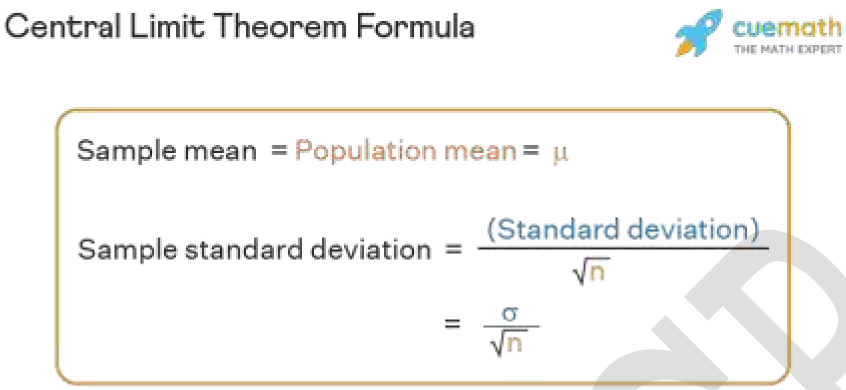
Central limit theorem

“The central limit theorem states that if you have a population with mean μ and standard deviation σ and take sufficiently large random samples from the population with replacement , then the distribution of the sample means will be approximately normally distributed”.

Applications of Central Limit Theorem

This **helps in analyzing data in methods like constructing confidence intervals**. One of the most common applications of CLT is in election polls. To calculate the percentage of persons supporting a candidate which are seen on news as confidence intervals.

The central limit theorem gives a formula for the sample mean and the sample standard deviation when the population mean and standard deviation are known. This is given as follows: **Sample mean = Population mean = μ μ Sample standard deviation = (Population standard deviation) / √n = σ / √n**.



Central Limit Theorem Formula

cuemath
THE MATH EXPERT

Sample mean = Population mean = μ

$$\text{Sample standard deviation} = \frac{\text{(Standard deviation)}}{\sqrt{n}} = \frac{\sigma}{\sqrt{n}}$$

- **Example 1:**

Suppose the mean age of people living in a town is 45 years and the standard deviation is 10. What will be the mean and variance of ages for sample sizes 20 and 49?

Solution: When n = 20, the central limit theorem cannot be applied as the sample size needs to be greater than or equal to 30.

When n = 49. The sample mean will be 45.

Sample standard deviation = σ√nσn = 10 / 7 = 1.43

Sample variance = 1.43^2 = 2.045

Answer: a) For n = 49, Mean = 45, Variance = 2.045

- **Example 2:**

In a study, it was reported that the mean of mobile users is 30 years and the standard deviation is 12. Taking a sample size of 100 what is the mean and standard deviation for the sample mean ages of tablet users?

Solution: Since the sample mean will tend to the population mean, thus, mean is 30.

The sample standard deviation is σ√nσn = 12 / 10 = 1.2

Answer: Mean = 30, Standard deviation = 1.2

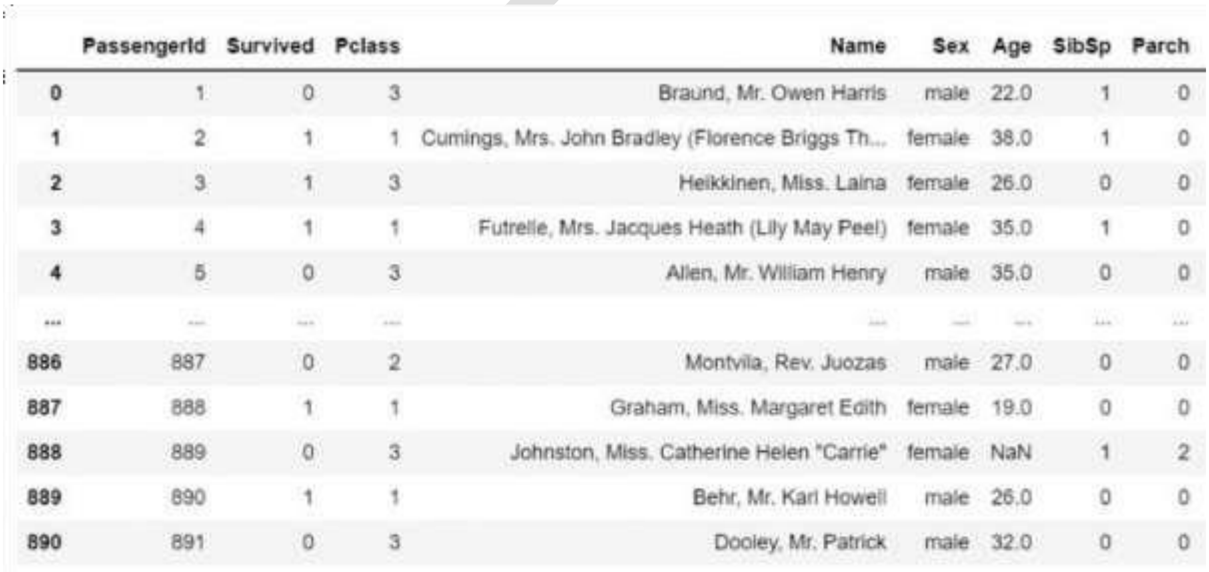
Exploratory Data Analysis

- Exploratory data analysis is applied to **investigate** the data and **summarize** the key insights.
- It will give you the basic understanding of your data, it's **distribution**, null values and much more.
- You can either explore data using graphs or through some python **functions.**
- There will be two type of analysis. **Univariate and Bivariate.** In the univariate, you will be analyzing a single attribute. But in the bivariate, you will be analyzing an attribute with the target attribute.

- In the **non-graphical approach**, you will be using functions such as shape, summary, describe, isnull, info, datatypes and more.
- In the **graphical approach**, you will be using plots such as scatter, box, bar, density and correlation plots.

Load the Data

Well, first things first. We will load the titanic dataset into python to perform EDA.



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0
...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0

```
#Load the required libraries import pandas
as pd import numpy as np import seaborn
as sns

#Load the data
df = pd.read_csv('titanic.csv')


#View the data df.head()
```

Copy

Our data is ready to be explored!

---

1. Basic information about data - EDA

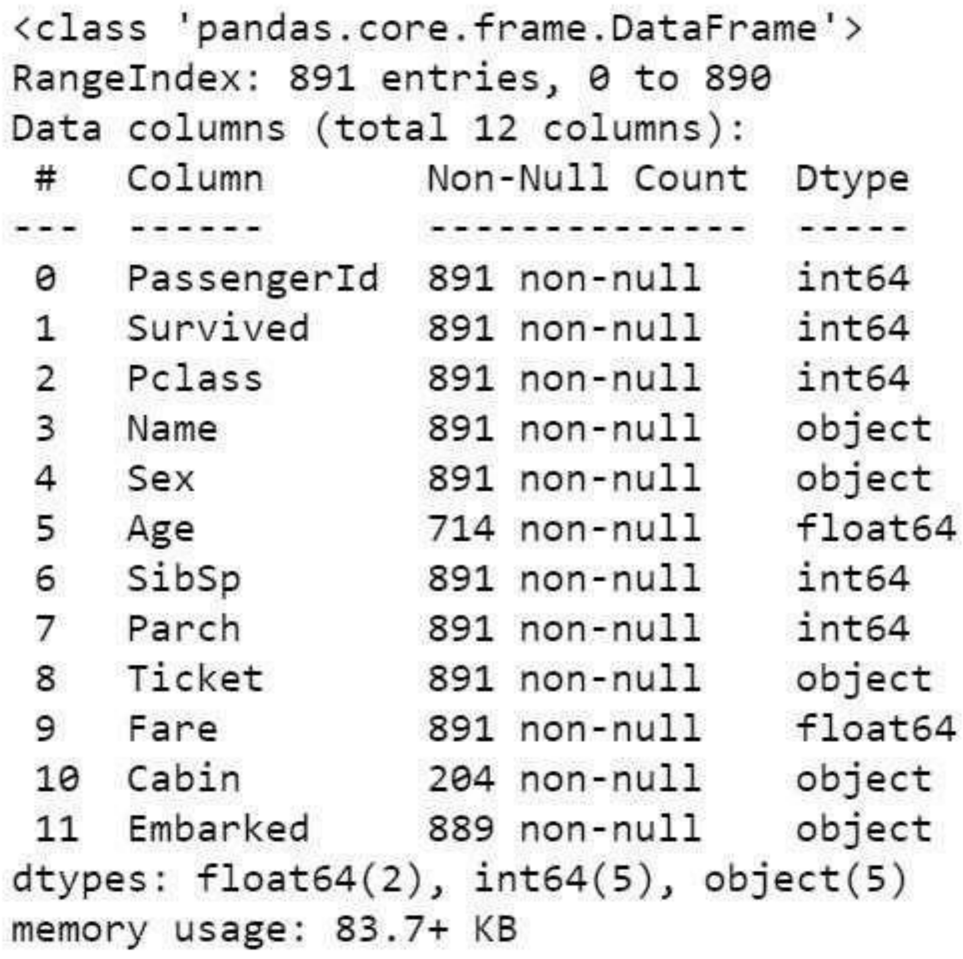
The df.info() function will give us the basic information about the dataset.
For any data, it is good to start by knowing its information.
Let’s see how it works with our data.

```
#Basic information  df.info()

#Describe the data

df.describe()
```

Copy



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

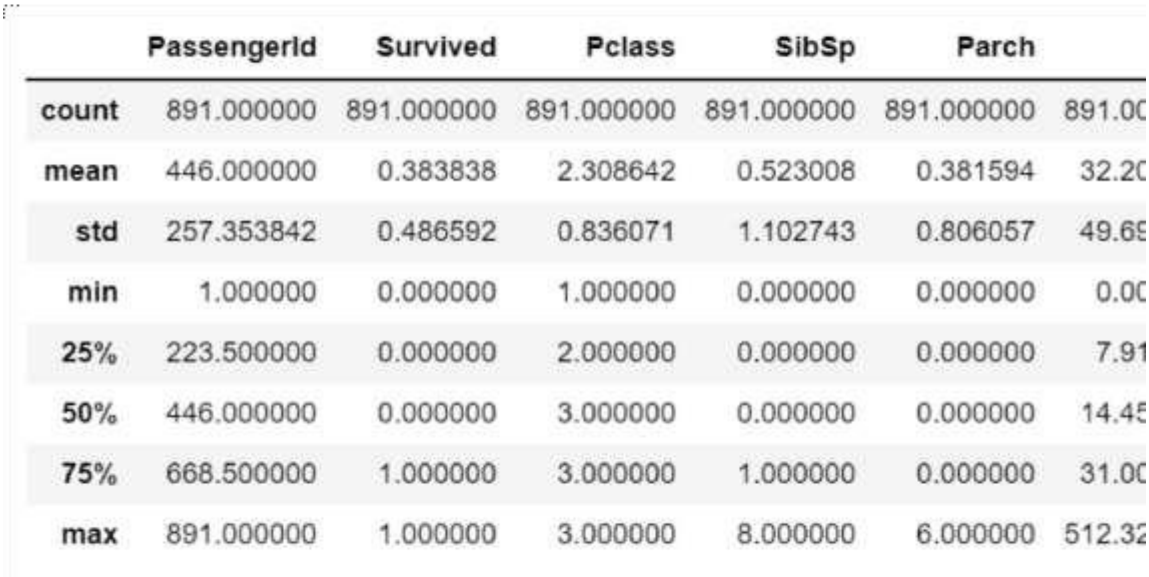
- **Describe the data - Descriptive statistics.**

Using this function, you can see the number of null values, datatypes, and memory usage as shown in the above outputs along with descriptive statistics.

2. Duplicate values

You can use the df.duplicate.sum() function to the sum of duplicate value present if any. It will show the number of duplicate values if they are present in the data.

```
#Find the duplicates
```



	PassengerId	Survived	Pclass	SibSp	Parch	
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.00
mean	446.000000	0.383838	2.308642	0.523008	0.381594	32.20
std	257.353842	0.486592	0.836071	1.102743	0.806057	49.69
min	1.000000	0.000000	1.000000	0.000000	0.000000	0.00
25%	223.500000	0.000000	2.000000	0.000000	0.000000	7.91
50%	446.000000	0.000000	3.000000	0.000000	0.000000	14.45
75%	668.500000	1.000000	3.000000	1.000000	0.000000	31.00
max	891.000000	1.000000	3.000000	8.000000	6.000000	512.32

```
df.duplicated()
.sum()
```

Copy

0

Well, the function returned ‘0’. This means, there is not a single duplicate value present in our dataset and it is a very good thing to know.

3. Unique values in the data

You can find the number of unique values in the particular column using unique() function in python.

```
#unique values

df['Pclass'].unique()

df['Survived'].unique()  df['Sex'].unique()
```

Copy

```
array([3, 1, 2],
dtype=int64)
  array([0, 1], dtype=int64)
array(['male', 'female'], dtype=object)
```

Copy

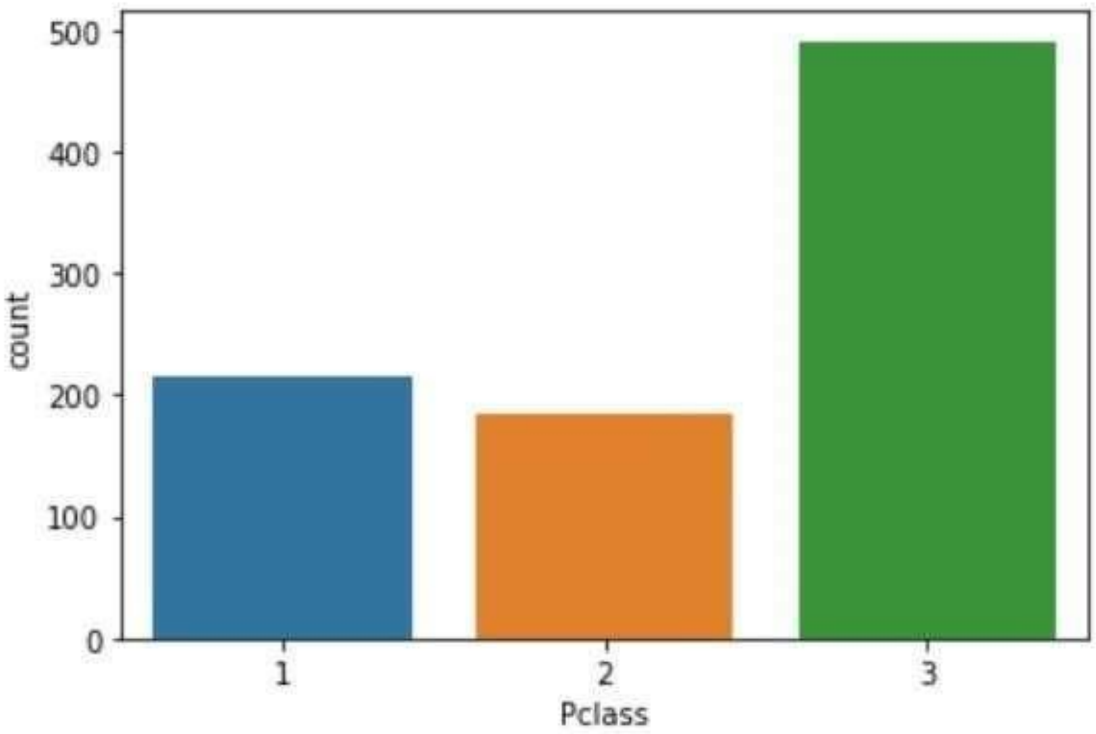
The unique() function has returned the unique values which are present in the data and it is pretty much cool!

4. Visualize the Unique counts

Yes, you can visualize the unique values present in the data. For this, we will be using the seaborn library. You have to call the sns.countlot() function and specify the variable to plot the count plot.

```
#Plot the unique values
sns.countplot(df['Pclass']).unique()
```

Copy



That's great! You are doing good. It is as simple as that. Though EDA has two approaches, a blend of graphical and non-graphical will give you the bigger picture altogether.

5. Find the Null values

Finding the null values is the most important step in the EDA. As I told many a time, ensuring the quality of data is paramount. So, let's see how we can find the null values.

```
#Find null values
df.isnull().s um()
```

Copy

```
PassengerId      0
```

```
Survived        0
Pclass         0
Name          0
Sex          0
Age         177
SibSp         0
Parch         0
Ticket        0
Fare         0
Cabin        687 Embarked        2
dtype: int64
```

Copy

Oh no, we have some null values in the **‘Age’** and **‘Cabin’** variables. But, don’t worry. We will find a way to deal with them soon.

6. Replace the Null values

Hey, we got a `replace()` function to replace all the null values with a specific data. It is too good!

```
#Replace null values

df.replace(np.nan,'0',in
place = True)

#Check the changes now
df.isnull().sum()
```

Copy

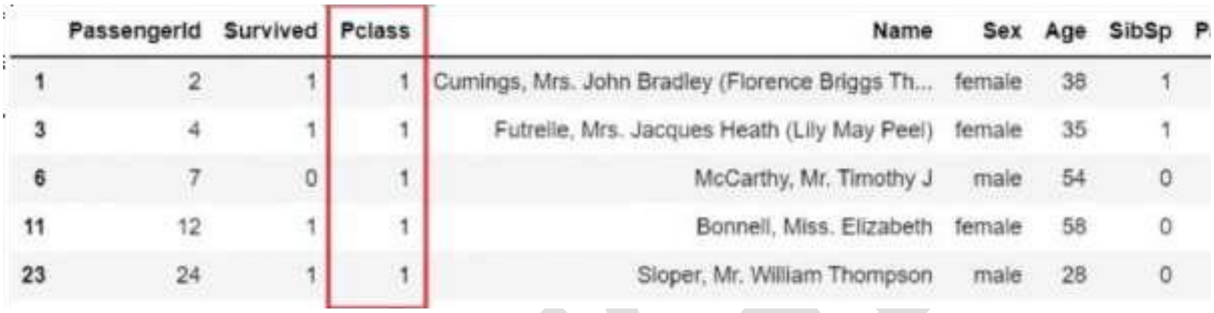
```
PassengerId    0
Survived       0
Pclass        0
Name          0
Sex          0
Age          0
SibSp         0
Parch         0
Ticket        0
Fare         0
Cabin         0
Embarked       0
```

Whoo! That’s awesome. It is very easy to find and replace the null values in the data as shown. I have used 0 to replace null values. You can even opt for more meaningful methods such as mean or median.

7. Know the datatypes

Knowing the datatypes which you are exploring is very important and an easy process too. Let’s see how it works.

#



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	P
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	
6	7	0	1	McCarthy, Mr. Timothy J	male	54	0	
11	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	
23	24	1	1	Sloper, Mr. William Thompson	male	28	0	

Copy

```
PassengerId      int64
Survived         int64
Pclass           int64
Name             object
Sex              object
Age              object
SibSp            int64
Parch            int64
Ticket           object
Fare             float64
Cabin            object Embarked
object   dtype: object
```

Copy

That’s it. You have to use the dtypes function for this a shown and you will get the datatypes of each attribute.

8. Filter the Data

Yes, you can filter the data based on some logic.

```
#Filter data

df[df['Pclass']==1
].head()
```

Copy

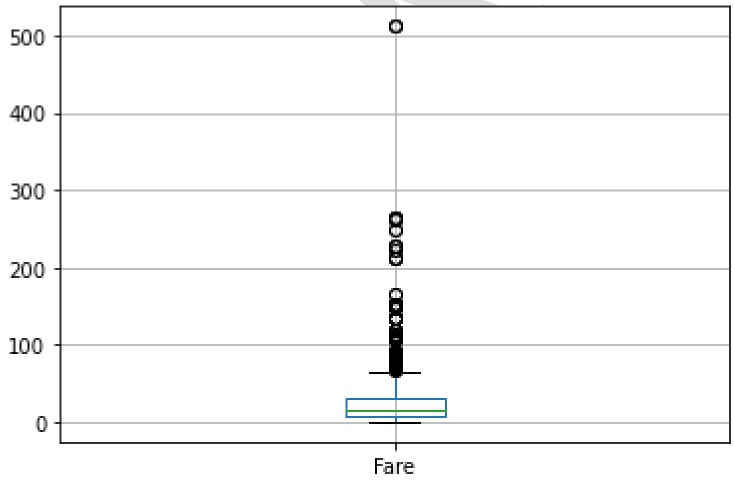
You can see that the above code has returned only data values that belong to class 1.

9. A quick box plot

You can create a box plot for any numerical column using a single line of code.

```
#Boxplot
df[['Fare']].boxplot()
```

Copy



10. Correlation Plot - EDA

Finally, to find the correlation among the variables, we can make use of the correlation function. This will give you a fair idea of the correlation strength between different variables.

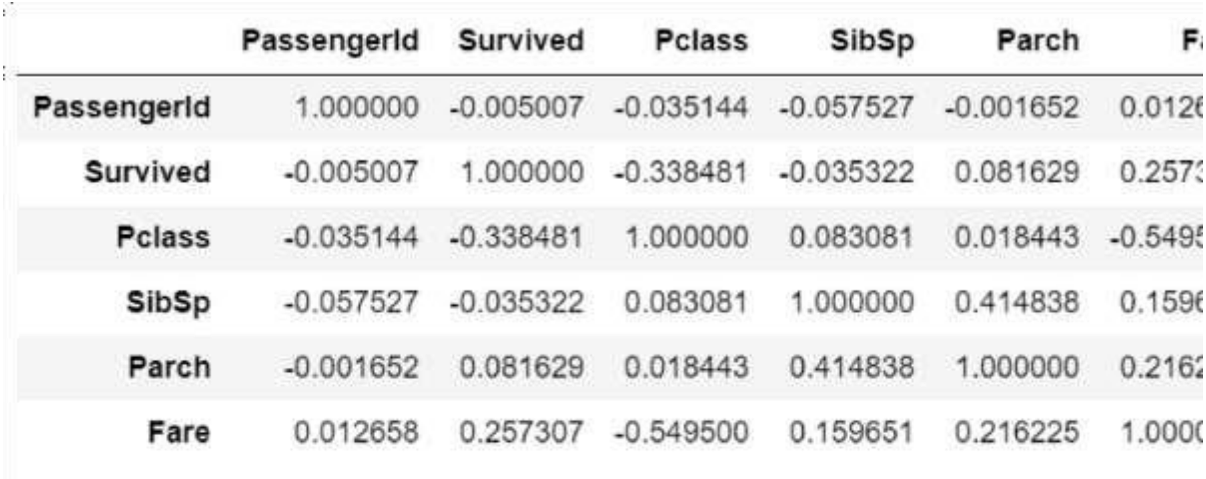
```
#Correlation    df.corr()
```

Copy

This is the correlation matrix with the range from +1 to -1 where +1 is highly and positively correlated and -1 will be highly negatively correlated.

You can even visualize the correlation matrix using seaborn library as shown below.

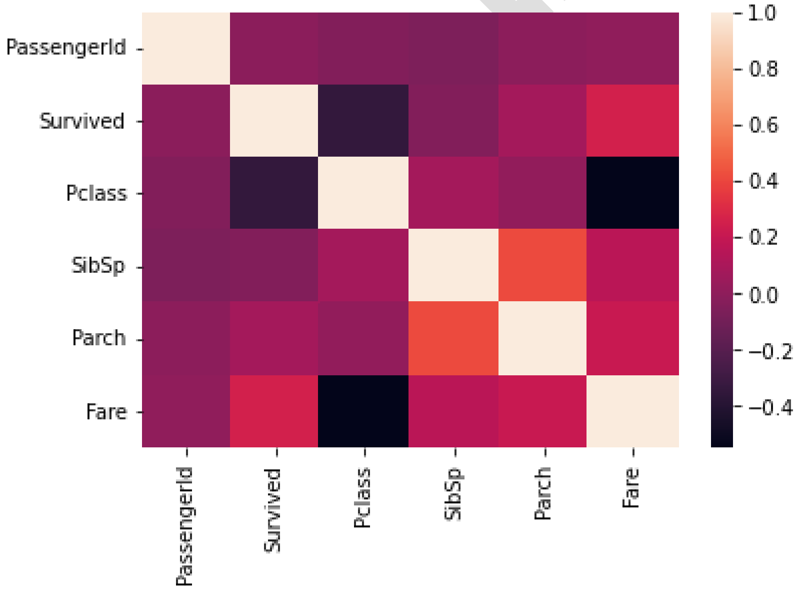
```
#Correlation plot
```



	PassengerId	Survived	Pclass	SibSp	Parch	F
PassengerId	1.000000	-0.005007	-0.035144	-0.057527	-0.001652	0.0126
Survived	-0.005007	1.000000	-0.338481	-0.035322	0.081629	0.2573
Pclass	-0.035144	-0.338481	1.000000	0.083081	0.018443	-0.5495
SibSp	-0.057527	-0.035322	0.083081	1.000000	0.414838	0.1596
Parch	-0.001652	0.081629	0.018443	0.414838	1.000000	0.2162
Fare	0.012658	0.257307	-0.549500	0.159651	0.216225	1.0000

```
sns.heatmap(df.
corr())
```

Copy



▪ EDA goals and benefits

Primary and Secondary Goals

The primary goal of EDA is to maximize the analyst's insight into a data set and into the underlying structure of a data set, while providing all of the specific items that an analyst would want to extract from a data set, such as:

- a good-fitting, parsimonious model
- a list of outliers
- a sense of robustness of conclusions
- estimates for parameters
- uncertainties for those estimates
- a ranked list of important factors
- conclusions as to whether individual factors are statistically significant
- optimal settings

Insight into the Data

Insight implies detecting and uncovering underlying structure in the data. Such underlying structure may not be encapsulated in the list of items above; such items serve as the specific targets of an analysis, but the real insight and "feel" for a data set comes as the analyst judiciously probes and explores the various subtleties of the data. The "feel" for the data comes almost exclusively from the application of various graphical techniques, the collection of which serves as the window into the essence of the data. Graphics are irreplaceable--there are no quantitative analogues that will give the same insight as well-chosen graphics.

To get a "feel" for the data, it is not enough for the analyst to know what is in the data; the analyst also must know what is not in the data, and the only way to do that is to draw on our own human pattern-recognition and comparative abilities in the context of a series of judicious graphical techniques applied to the data.

.Univariate Data Analysis

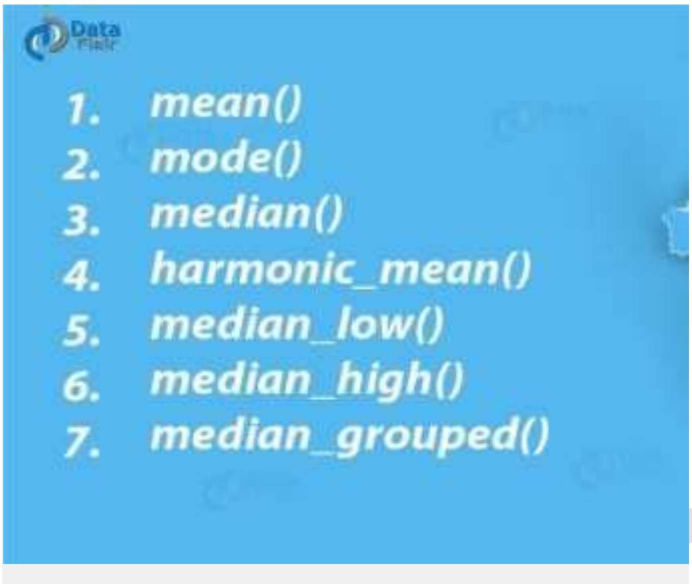
With data analysis, we use two main statistical methods- *Descriptive* and *Inferential*.

- **Descriptive statistics** uses tools like mean and standard deviation on a sample to summarize data.

- **Inferential statistics**, on the other hand, looks at data that can randomly vary, and then draw conclusions from it.

Some such variations include observational errors and sampling variation.

> imp statis a s



1. mean()
2. mode()
3. median()
4. harmonic_mean()
5. median_low()
6. median_high()
7. median_grouped()

Python – *CentTend*

a.

Descriptive Statistics in Python

Python Descriptive Statistics process describes the basic features of data in a study. It delivers summaries on the sample and the measures and does not use the data to learn about the population it represents.

Under descriptive statistics, fall two sets of properties- *central tendency* and *dispersion*. Python Central tendency characterizes one central value for the entire distribution.

Measures under this include mean, median, and mode. Python Dispersion is the term for a practice that characterizes how apart the members of the distribution are from the center and from each other. Variance/Standard Deviation is one such measure of variability.

Python Descriptive Statistics

a.mean()

We have seen what central tendency or central location is. Now let’s take a look at all the functions Python caters to us to calculate the central tendency for a distribution. For this, let’s import the *Python statistics* module.

This function returns the arithmetic average of the data it operates on. If called on an empty container of data, it raises a StatisticsError.

```
>>> nums=[1,2,3,5,7,9]

>>> st.mean(nums)

4.5

>>> st.mean([-2,-4,7]) #Negative numbers

0.3333333333333333

>>> from fractions import Fraction as fr

>>> st.mean((fr(3,4),fr(5,7),fr(2,1))) #Fractions

Fraction(97, 84)

>>> st.mean({1:"one",2:"two",3:"three"}) #Keys from a dictionary

2
```

Do you know the difference between Python Modules vs Packages

b. mode()

This function returns the most common value in a set of data. This gives us a great idea of where the center lies.

```
>>> nums=[1,2,3,5,7,9,7,2,7,6]

>>> st.mode(nums)

7
>>> st.mode(['A','B','b','B','A','B'])

‘B’
```

c. median()

For data of odd length, this returns the middle item; for that of even length, it returns the average of the two middle items.

```
>>> st.median(nums) #(5+6)/2

5.5
```

d. harmonic_mean()

This function returns the harmonic mean of the data. For three values a, b, and c, the harmonic mean is-
3/(1/a + 1/b +1/c)
It is a measure of the center; one such example would be speed.

```
>>> st.harmonic_mean([2,4,9.7])

3.516616314199396
```

For the same set of data, the arithmetic mean would give us a value of 5.233333333333333.

e. median_low()

When the data is of an even length, this provides us the low median of the data. Otherwise, it returns the middle value.

```
>>> st.me
dian_l
ow([1,
2,4])

2

>>> st.median_low([1,2,3,4])

2
```

f. median_high()

Like median_low, this returns the high median when the data is of an even length. Otherwise, it returns the middle value.

```
>>>
st.med
ian_hig
h([1,2,
4]) 2

>>> st.median_high([1,2,3,4])

3
```

Let’s Learn CGI Programming in Python with Functions and Modules

g. median_grouped()

This function uses interpolation to return the median of grouped continuous data. This is the 50th percentile.

```
>>> st.me
dian(
[1,3,
3,5,7
]) 3

>>> st.median_grouped([1,3,3,5,7],interval=1)

3.25

>>> st.median_grouped([1,3,3,5,7],interval=2)

3.5
```

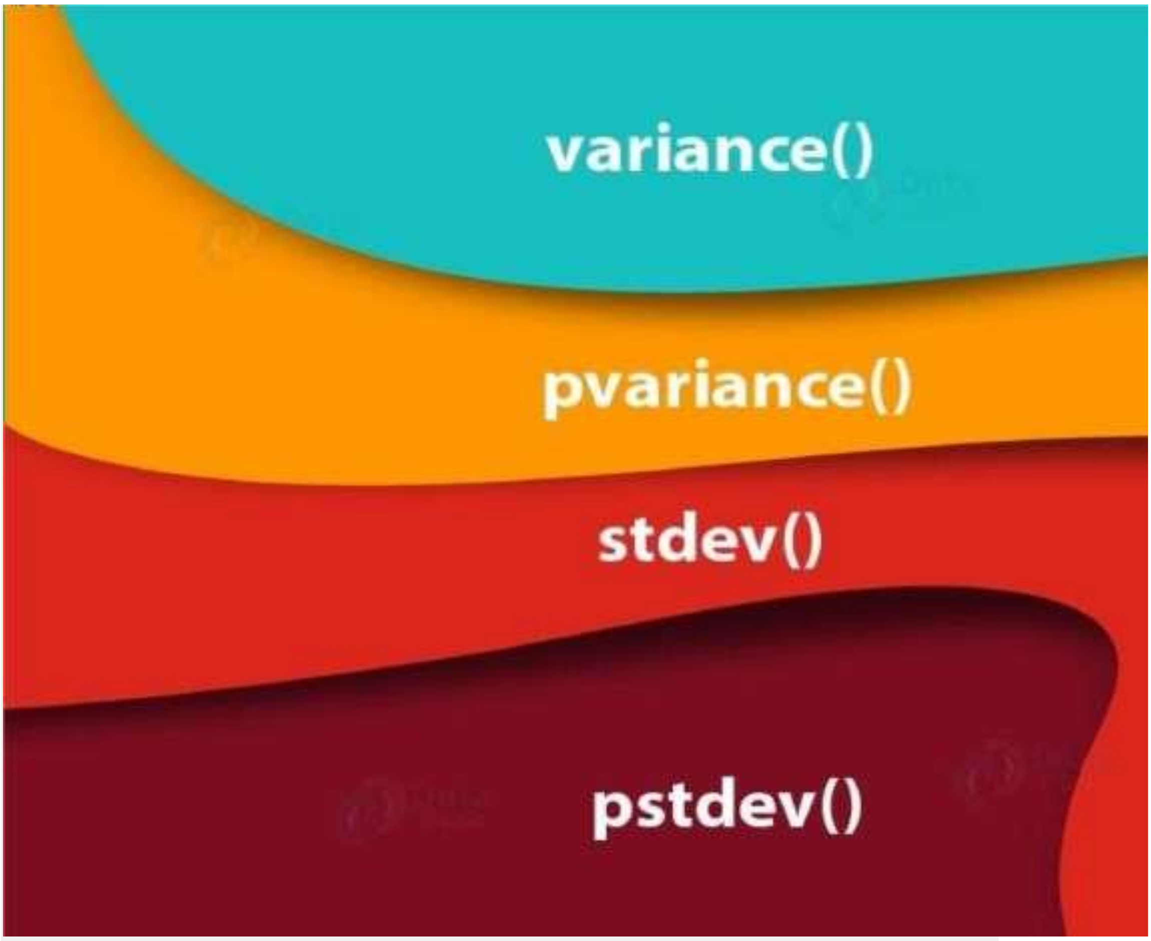
Python Descriptive Statistics – Dispersion in Python

Dispersion/spread gives us an idea of how the data strays from the typical value.

a. variance()

This returns the variance of the sample. This is the second moment about the mean and a larger value denotes a rather spread-out set of data. You can use this when your data is a sample out of a population.

```
>>> st.variance(nums)
```



Python Descriptive Statistics – Dispersion

```
7.433333333333334
```

b. pvariance()

This returns the population variance of data. Use this to calculate variance from an entire population.

```
>>> st.pvariance(nums)

6.69
```

c. stdev()

This returns the standard deviation for the sample. This is equal to the square root of the sample variance.

```
>>> st.stdev(nums)

2.7264140062238043
```

Read about Python Namespace and Variable Scope – Local and Global

Variables

d. pstdev()

This returns the population standard deviation. This is the square root of population variance.

```
>>> st.pstdev(nums)
```

2.5865034312755126 The *statistics* module defines one exception- ***exception*** **statistics.StatisticsError** This is a subclass of ValueError.

pandas with Descriptive Statistics in Python

We can do the same things using pandas too-

```
>>> import pandas as pd

>>> df=pd.DataFrame(nums)

>>> df.mean() 04.9dtyp
```

```
e:float64
```

Follow this to know more about Python Pandas

```
>>>df.mode()
0
7
>>> df.std() #Standard deviation
0
2.726414dtype
: fl
```

```
o
a
t
6
4
>
>
>
d
f
.
s
k
e
w
(
)
0 -0.115956 #The distribution is symmetric
dtype: float64
```

A value less than -1 is skewed to the left; that greater than 1 is skewed to the right. A value between -1 and 1 is symmetric.

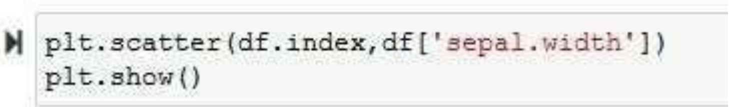
So, this was all about Python Descriptive Statistics Tutorial.
Hope you like our explanation

Univariate enumerative Plots :

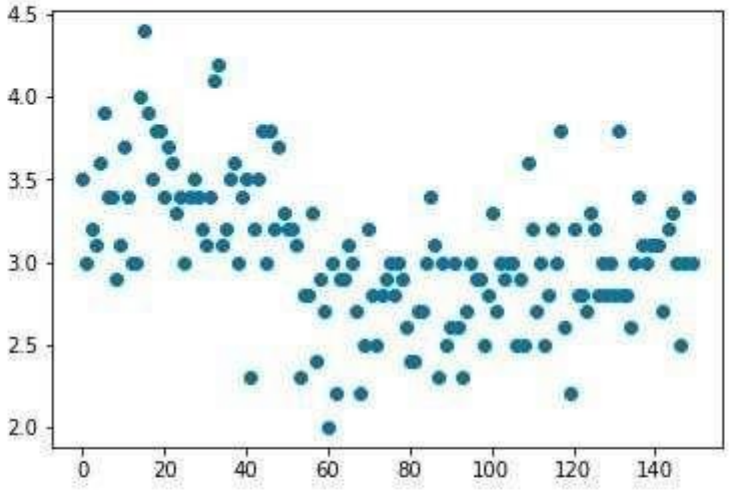
These plots enumerate/show every observation in data and provide information about the distribution of the observations on a single data variable. We now look at different enumerative plots.

1. UNIVARIATE SCATTER PLOT :

This plots different observations/values of the same variable corresponding to the index/observation number. Consider plotting of the variable ‘sepal length(cm)’ :

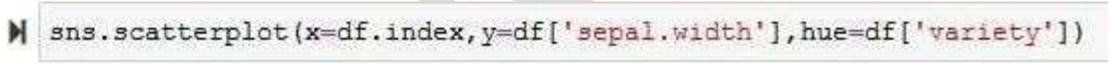


```
plt.scatter(df.index,df['sepal.width'])
plt.show()
```

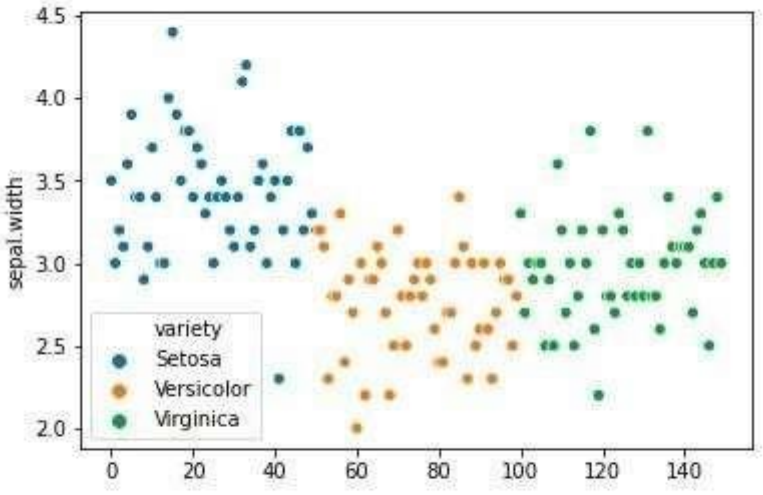


Use the *plt.scatter()* function of matplotlib to plot a univariate scatter diagram. The scatter() function requires two parameters to plot. So, in this example, we plot the variable ‘sepal.width’ against the corresponding observation number that is stored as the index of the data frame (df.index).

Then visualize the same plot by considering its variety using the *sns.scatterplot()* function of the seaborn library.



```
sns.scatterplot(x=df.index,y=df['sepal.width'],hue=df['variety'])
```

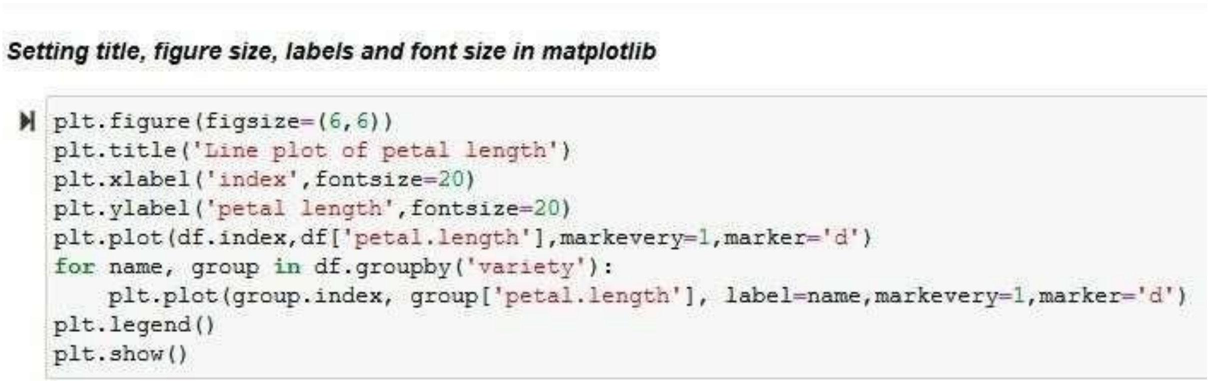


One of the interesting features in seaborn is the ‘hue’ parameter. In seaborn, the hue parameter determines which column in the data frame should be used for color encoding. This helps to differentiate between the data values according to the categories they belong to. The hue parameter takes the grouping variable as it's input using which it will produce points with different colors. The variable passed onto 'hue' can be either categorical or numeric, although color mapping will behave differently in the latter case.

Note:Every function has got a wide variety of parameters to play with to produce better results. If one is using Jupyter notebook, the various parameters of the function used can be explored by using the 'Shift+Tab' shortcut.

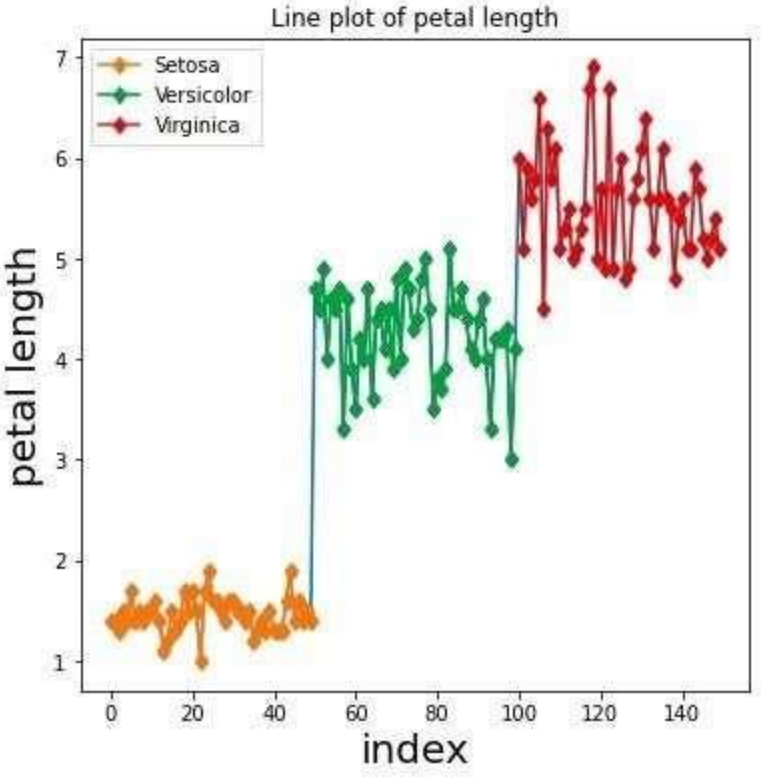
2. LINE PLOT (with markers) :

A line plot visualizes data by connecting the data points via line segments. It is similar to a scatter plot except that the measurement points are ordered (typically by their x-axis value) and joined with straight line segments.



Setting title, figure size, labels and font size in matplotlib

```
plt.figure(figsize=(6,6))
plt.title('Line plot of petal length')
plt.xlabel('index',fontsize=20)
plt.ylabel('petal length',fontsize=20)
plt.plot(df.index,df['petal.length'],markevery=1,marker='d')
for name, group in df.groupby('variety'):
    plt.plot(group.index, group['petal.length'], label=name,markevery=1,marker='d')
plt.legend()
plt.show()
```



The matplotlib *plt.plot()* function by default plots the data using a line plot.

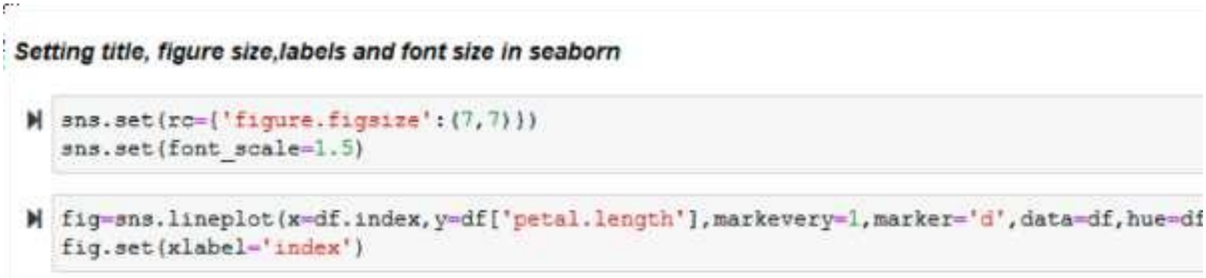
Previously, we discussed the hue parameter of seaborn. Though there is no such automated option available in matplotlib, one can use the *groupby()* function of pandas which helps in plotting such a graph.

Note:In the above illustration, the methods to set title, font size, etc in matplotlib are also implemented.

— Explanation of the functions used :

- *plt.figure(figsize=())* : To set the size of figure
- *plt.title()* : To set title
- *plt.xlabel() / plt.ylabel()* : To set labels on X-axis/Y-axis
- *df.groupby( )* : To group the rows of the data frame according to the parameter passed onto the function

- The groupby() function returns the data frames grouped by the criterion variable passed and the criterion variable.
- The *for loop* is used to plot each data point according to its variety.
- *plt.legend()*: Adds a legend to the graph (Legend describes the different elements seen in the graph).
- *plt.show()* : to show the plot.



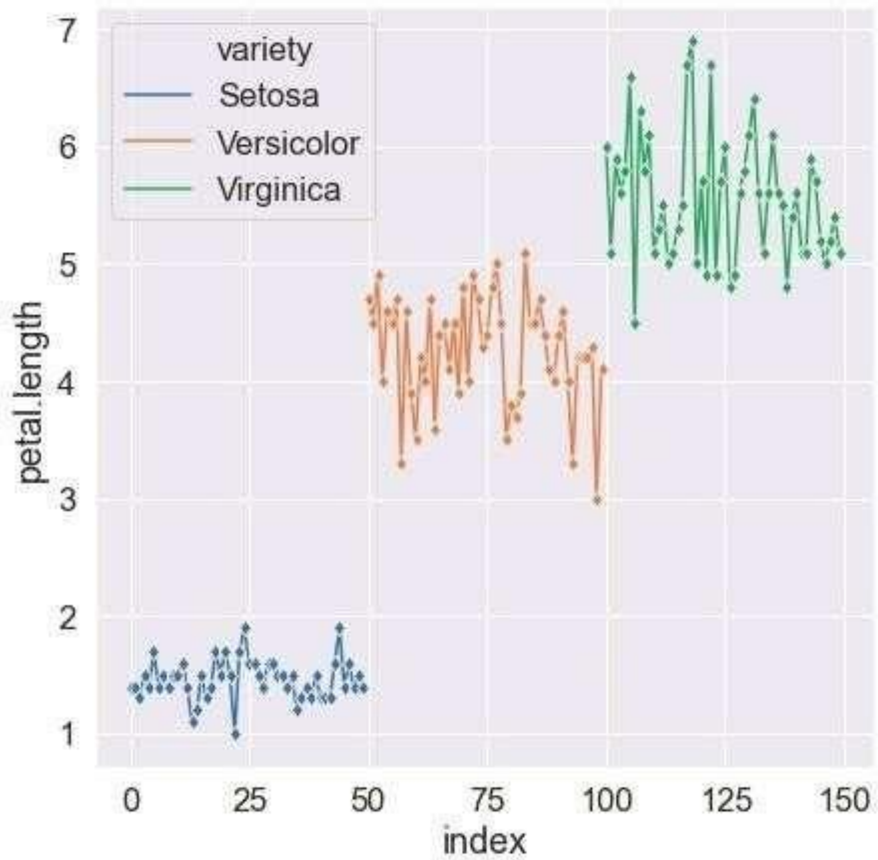
Setting title, figure size,labels and font size in seaborn

```
sns.set(rc={'figure.figsize':(7,7)})
sns.set(font_scale=1.5)
```

```
fig=sns.lineplot(x=df.index,y=df['petal.length'],markevery=1,marker='d',data=df,hue=df
fig.set(xlabel='index')
```

The ‘markevery’ parameter of the function *plt.plot()* is assigned to ‘1’ which means it will plot every 1st marker starting from the first data point. There are various marker styles which we can pass as a parameter to the function.

The *sns.lineplot()* function can also visualize the line plot.



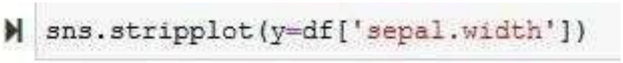
In seaborn, the labels on axes are automatically set based on the columns that are passed for plotting. However if one desires to change it, it is possible too using the set() function.

Note: There are often cases wherein one would want to explore how the distribution of a single continuous variable is affected by a second categorical variable. The seaborn library provides a variety of plots that help perform such types of comparisons between uni-variate distributions. Three such plots are discussed in this article : Strip plot, Swarm plot (under enumerative plots), and Violin plot (under summary plots). The hue parameter mentioned in above plots is also for similar use.

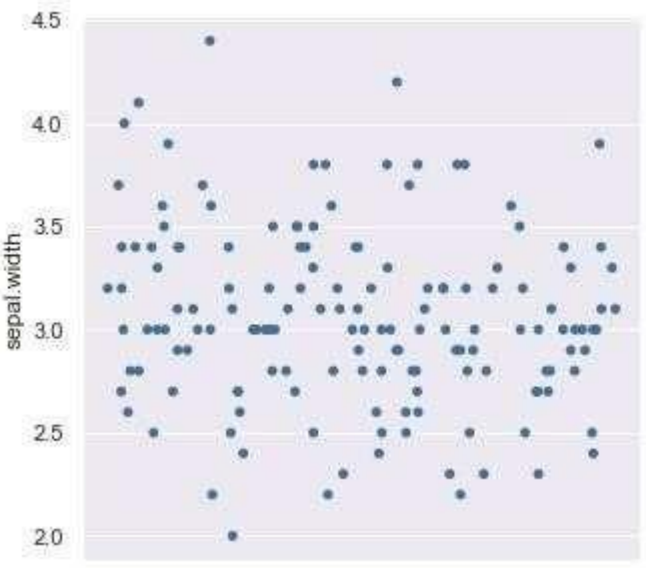
3. *STRIP PLOT :*

The strip plot is similar to a scatter plot. It is often used along with other kinds of plots for better analysis. It is used to visualize the distribution of data points of the variable.

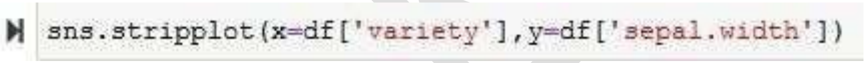
The *sns.striplot ( )* function is used to plot a strip-plot :



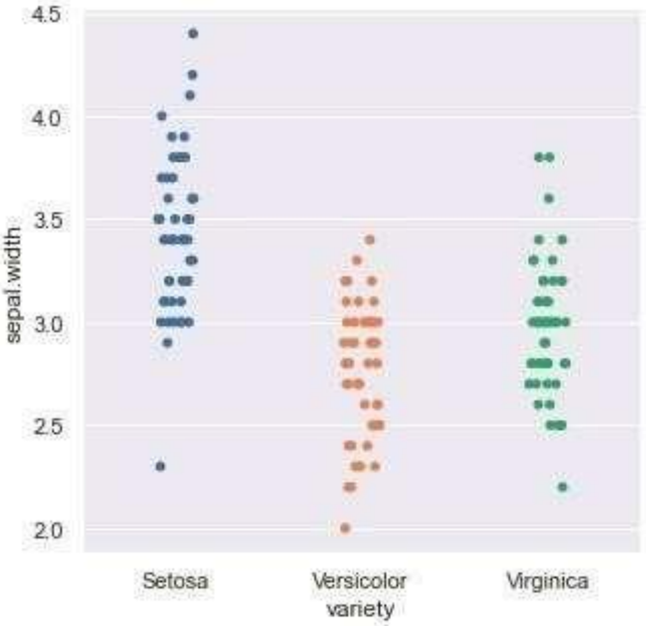
```
sns.stripplot(y=df['sepal.width'])
```



It also helps to plot the distribution of variables for each category as individual data points. By default, the function creates a vertical strip plot where the distributions of the continuous data points are plotted along the Y-axis and the categories are spaced out along the X-axis. In the above plot, categories are not considered. Considering the categories helps in better visualization as seen in the below plot.



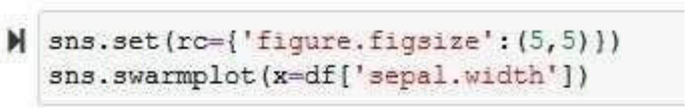
```
sns.stripplot(x=df['variety'],y=df['sepal.width'])
```



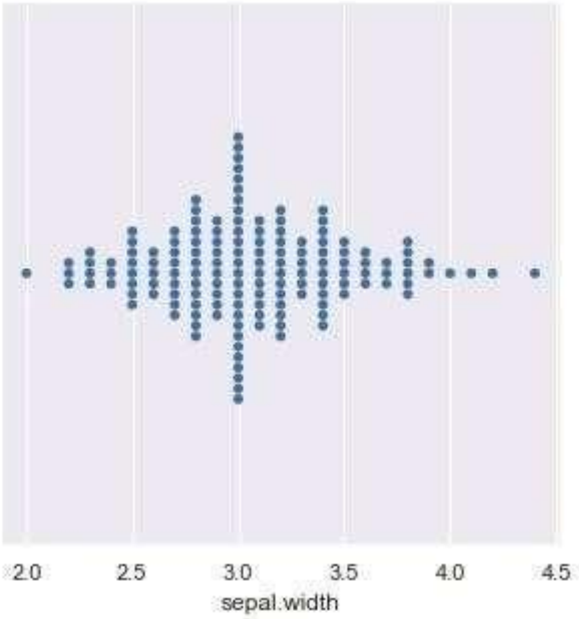
4. *SWARM PLOT :*

The swarm-plot, similar to a strip-plot, provides a visualization technique for univariate data to view the spread of values in a continuous variable. The only difference between the stripplot and the swarm-plot is that the swarm-plot spreads out the data points of the variable automatically to avoid overlap and hence provides a better visual overview of the data.

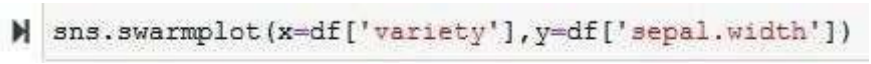
The *sns.swarmplot( )* function is used to plot a swarm-plot :



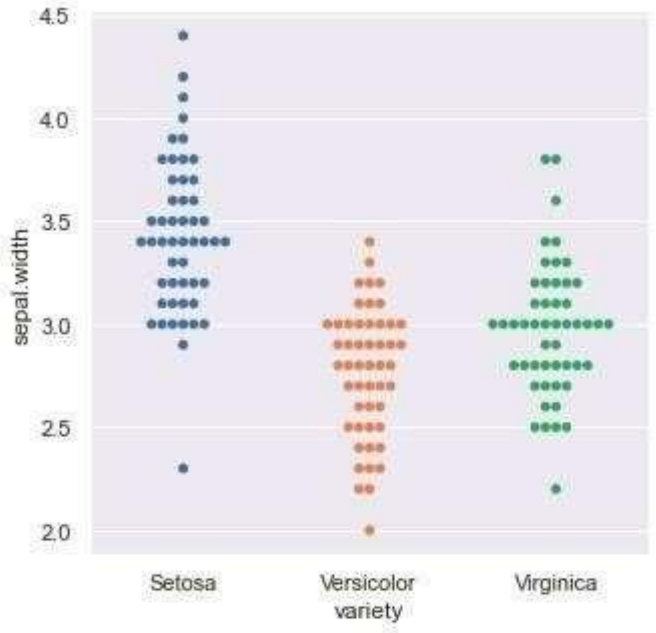
```
sns.set(rc={'figure.figsize':(5,5)})
sns.swarmplot(x=df['sepal.width'])
```



Distribution of the variable ‘sepal.width’ according to the categories :



```
sns.swarmplot(x=df['variety'],y=df['sepal.width'])
```



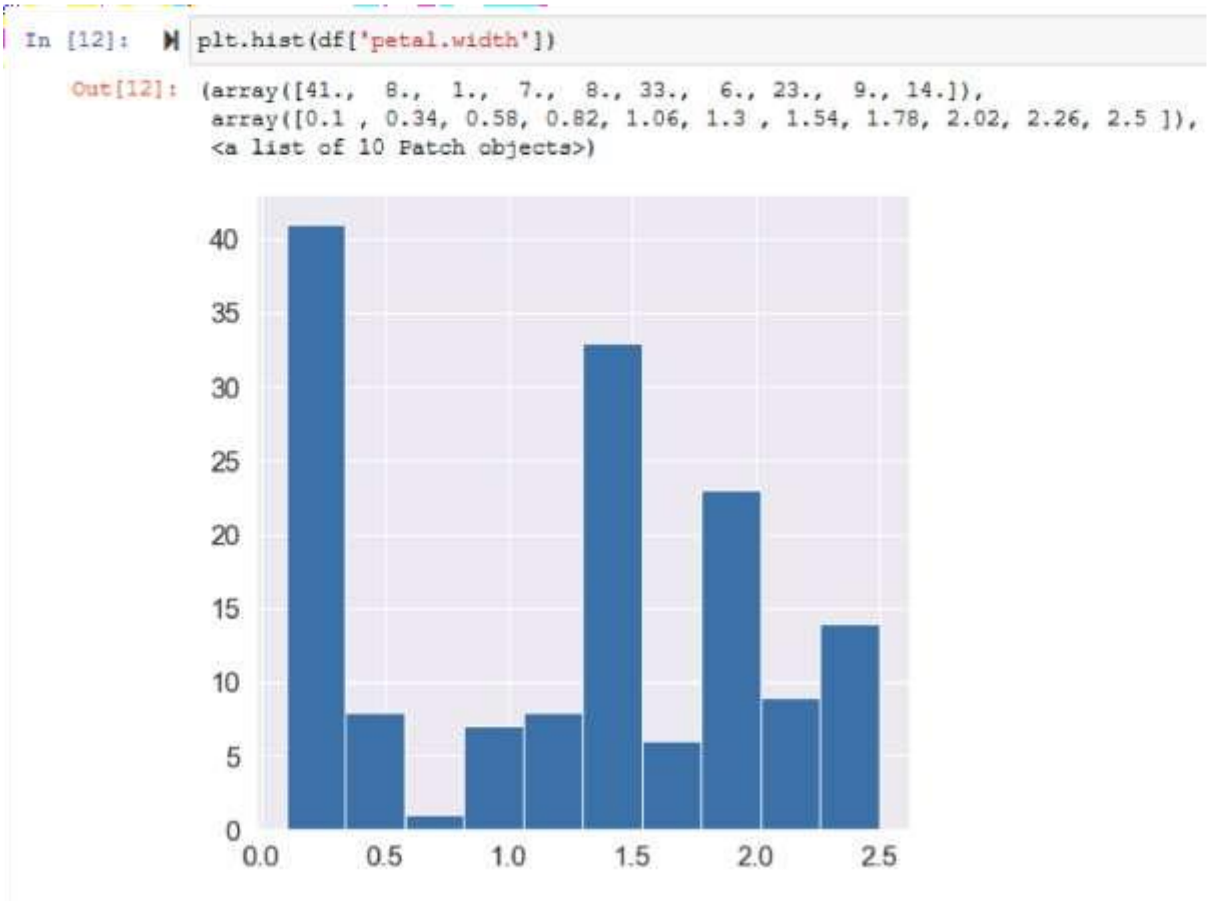
Uni-variate summary plots :

These plots give a more concise description of the location, dispersion, and distribution of a variable than an enumerative plot. It is not feasible to retrieve every individual data value in a summary plot, but it helps in efficiently representing the whole data from which better conclusions can be made on the entire data set.

5. HISTOGRAMS :

Histograms are similar to bar charts which display the counts or relative frequencies of values falling in different class intervals or ranges. A histogram displays the shape and spread of continuous sample data. It also helps us understand the skewness and kurtosis of the distribution of the data.

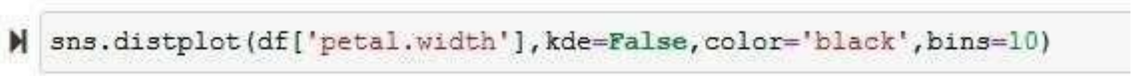
Plotting histogram using the matplotlib *plt.hist()* function :



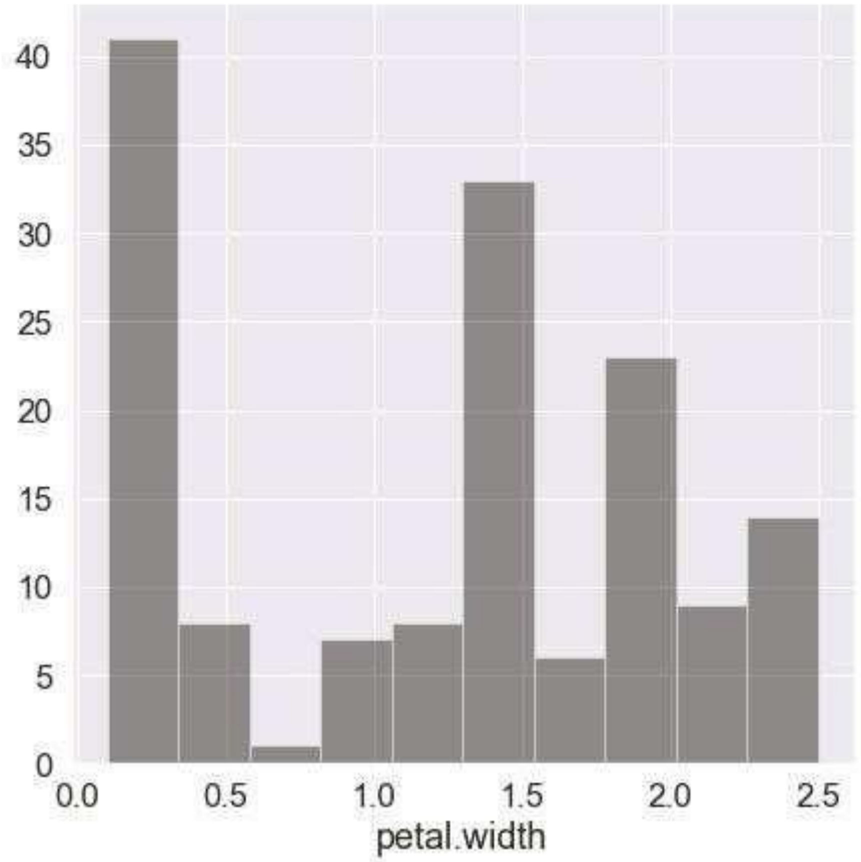
```
In [12]:  plt.hist(df['petal.width'])

Out[12]: (array([41.,  8.,  1.,  7.,  8., 33.,  6., 23.,  9., 14.]),
          array([0.1 , 0.34, 0.58, 0.82, 1.06, 1.3 , 1.54, 1.78, 2.02, 2.26, 2.5 ]),
          <a list of 10 Patch objects>)
```

The seaborn function *sns.distplot()* can also be used to plot a histogram.



```
sns.distplot(df['petal.width'],kde=False,color='black',bins=10)
```

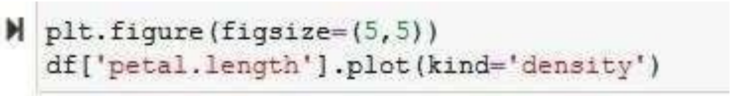


The kde (kernel density) parameter is set to False so that only the histogram is viewed. There are many parameters like bins (indicating the number of bins in histogram allowed in the plot), color, etc; which can be set to obtain the desired output.

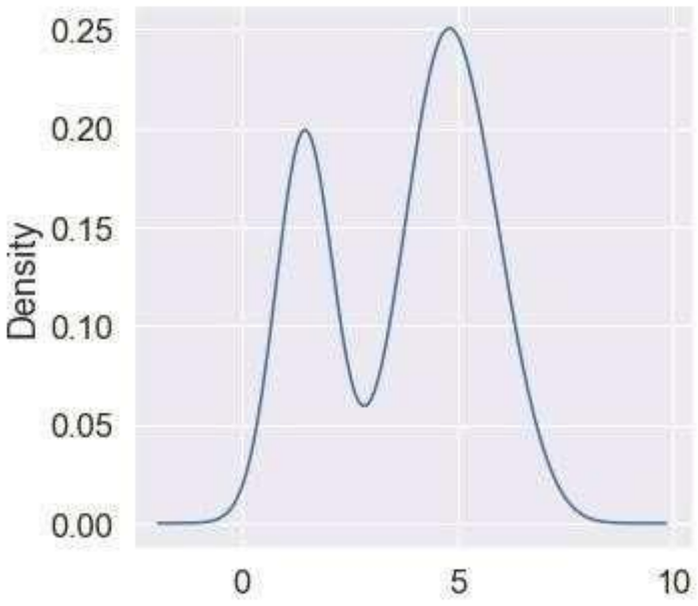
6. DENSITY PLOTS :

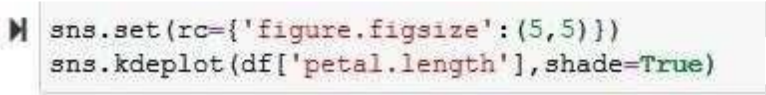
A density plot is like a smoother version of a histogram. Generally, the kernel density estimate is used in density plots to show the probability density function of the variable. A continuous curve, which is the kernel is drawn to generate a smooth density estimation for the whole data.

Plotting density plot of the variable ‘petal.length’ :

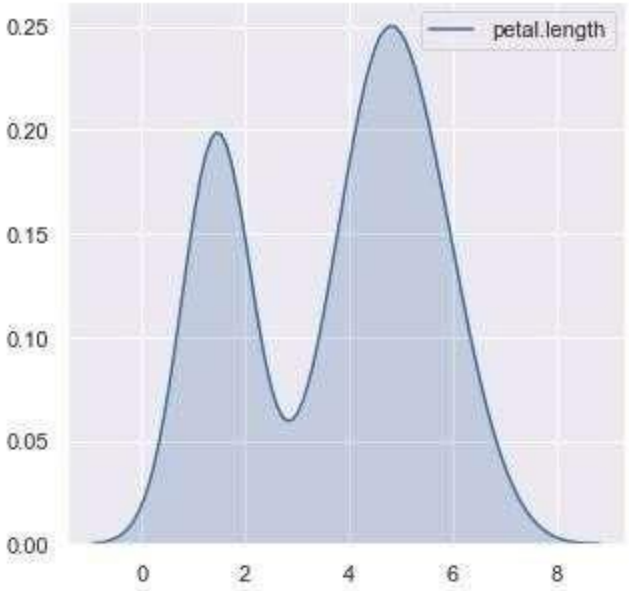


```
plt.figure(figsize=(5,5))
df['petal.length'].plot(kind='density')
```





```
sns.set(rc={'figure.figsize':(5,5)})
sns.kdeplot(df['petal.length'],shade=True)
```



we use the pandas *df.plot()* function (built over matplotlib) or the seaborn library's *sns.kdeplot()* function to plot a density plot . Many features like shade, type of distribution, etc can be set using the parameters available in the functions. By default, the kernel used is Gaussian (this produces a Gaussian bell curve). Also, other graph smoothing techniques/filters are applicable.

7. RUG PLOTS :

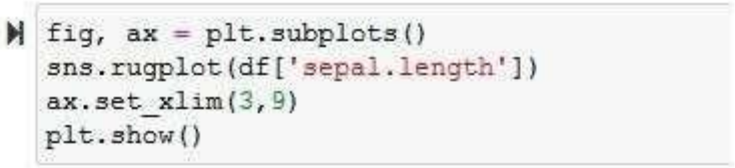
A rug plot is a very simple, but also an ideal legitimate, way of representing a distribution. It consists of vertical lines at each data point. Here, the height is arbitrary. The density of the distribution can be known by how dense the tickmarks are.

The connection between the rug plot and histogram is very direct: a histogram just creates bins along with the range of the data and then draws a bar with height equal to the number of ticks in each bin. In a rug plot, all of the data points are plotted on a single axis, one tick mark or line for each one.

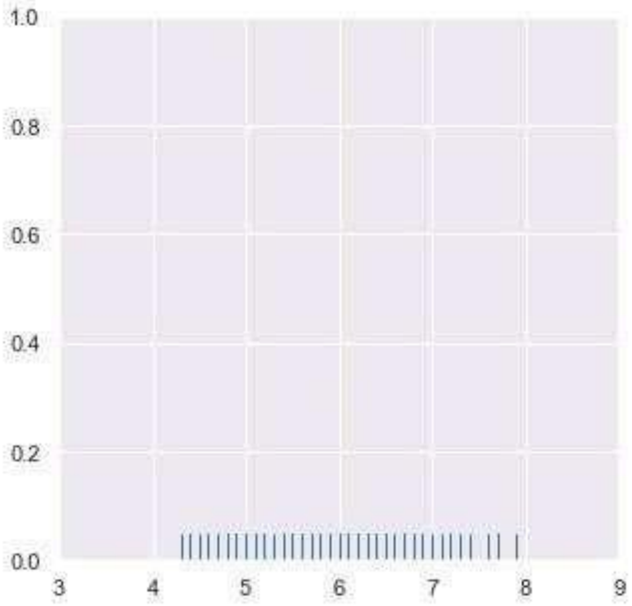
Compared to a marginal histogram, the rug plot suffers somewhat in terms of readability of the distribution, but it is more compact in its representation of the data. A rug is a very short, long display of point symbols, one for each distinct value. Often a vertical pipe symbol | is used to minimize overlap. Rug plot may not be considered as a primary plot choice, but it can be a good supporter plot in certain circumstances.

Plotting the rugs of variable ‘sepal .length’ :

RUG PLOT



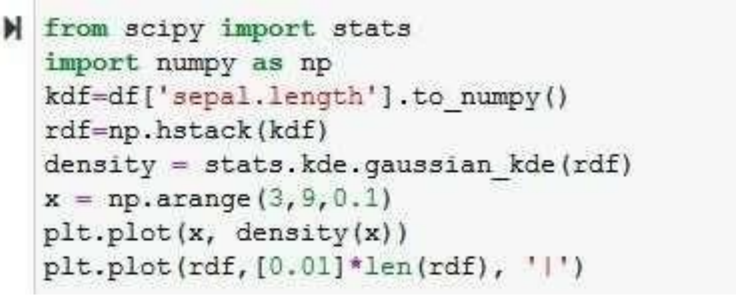
```
fig, ax = plt.subplots()
sns.rugplot(df['sepal.length'])
ax.set_xlim(3,9)
plt.show()
```



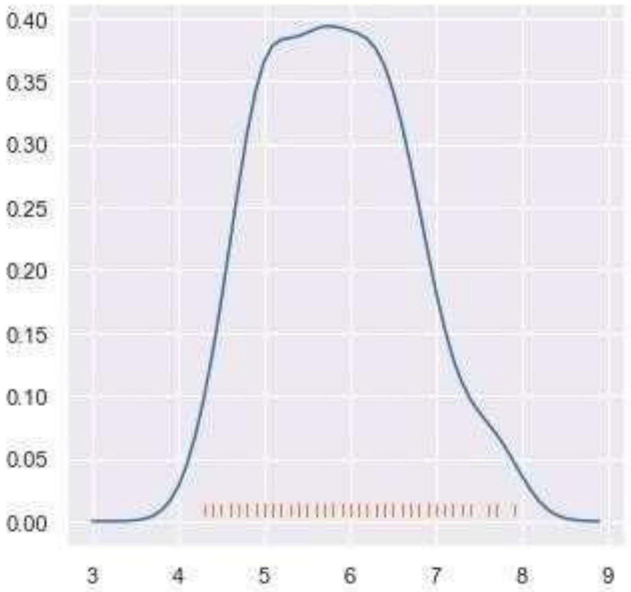
Note :In a few cases, there may be a need to set the range of the values in each axis. In the above illustration, plt.subplots() function that returns a figure object and axes object. Using the axes object ‘ax’ that is passed on to the set_xlim() method, the range of values to be considered on the X-axis is set.

A kernel density estimate can be plotted along with the rugs which can provide a better understanding of the data.

In matplotlib, there is no direct function to create a rug plot. So, scipy.stats module is used to create the required kernel density distribution which is then plotted using the *plt.plot()* function along with the rugs.



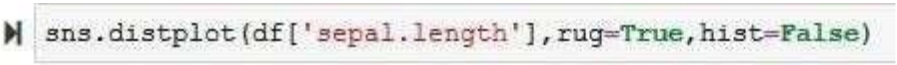
```
from scipy import stats
import numpy as np
kdf=df['sepal.length'].to_numpy()
rdf=np.hstack(kdf)
density = stats.kde.gaussian_kde(rdf)
x = np.arange(3,9,0.1)
plt.plot(x, density(x))
plt.plot(rdf,[0.01]*len(rdf), '|')
```



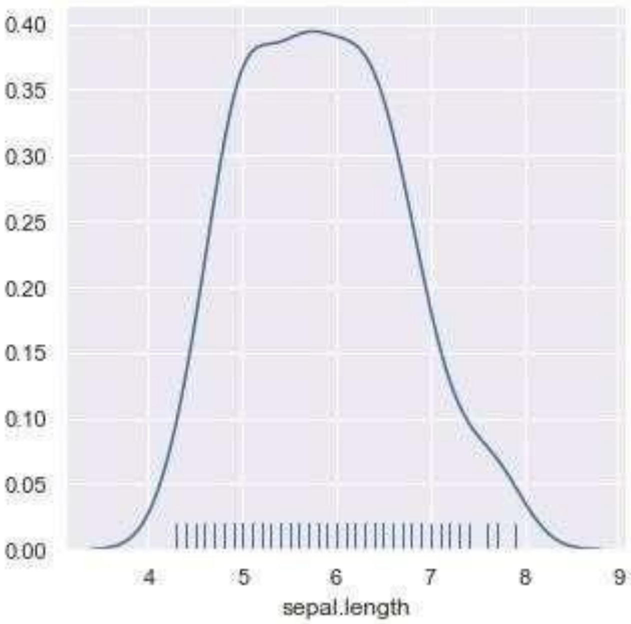
Explanation of the methods used :

The *kde.gaussian_kde( )* function generates a kernel-density estimate using Gaussian kernels. In the current case of univariate data, this function takes a 1D array as an input data set. To get the required 1-D array, firstly the function *to_numpy()* is used to convert the data frame into numpy array and then use the *np.hstack()* function to stack the sequence of input arrays horizontally (i.e. column-wise) to make a single array. This 1-D array 'rdf' was then passed as input to the *kde.gaussian_kde()* function. The range of values to be considered along with the step size was specified using the *np.arange()* function. Then, the *plt.plot()* is used to obtain the plot.

Seaborn library provides a direct and easier function to visualize such a plot with many parameters to play with.

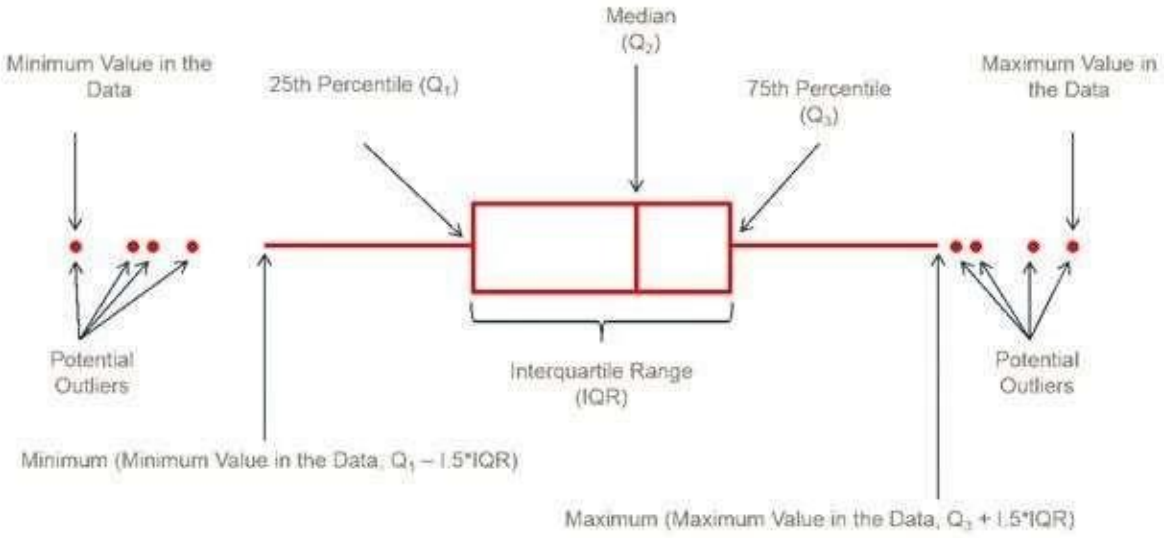


```
sns.distplot(df['sepal.length'],rug=True,hist=False)
```



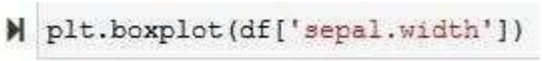
8. BOX PLOTS :

A box-plot is a very useful and standardized way of displaying the distribution of data based on a five-number summary (minimum, first quartile, second quartile(median), third quartile, maximum). It helps in understanding these parameters of the distribution of data and is extremely helpful in detecting outliers.

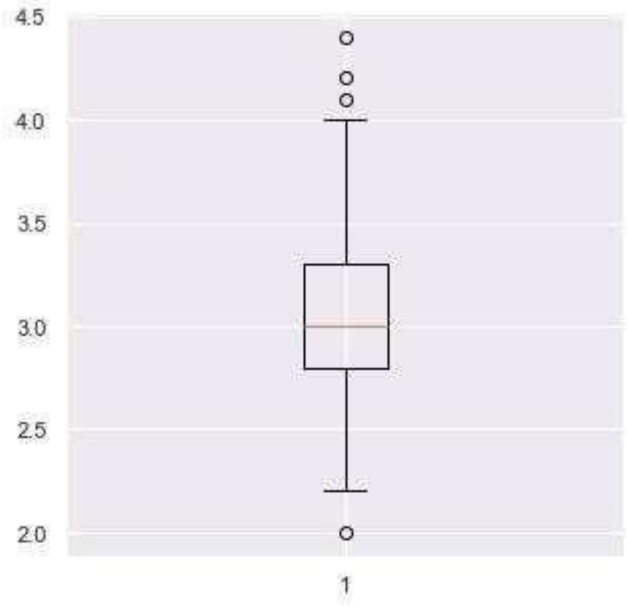

Median (Q_2)
Minimum Value in the Data
25th Percentile (Q_1)
75th Percentile (Q_3)
Maximum Value in the Data
Potential Outliers
Interquartile Range (IQR)
Potential Outliers
Minimum (Minimum Value in the Data, Q_1 – 1.5*IQR)
Maximum (Maximum Value in the Data, Q_3 + 1.5*IQR)

(Source:leansigmacorporation.com)

Plotting box plot of variable ‘sepal.width’ :



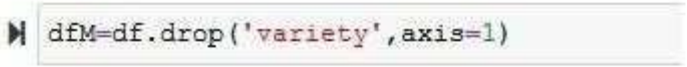
```
plt.boxplot(df['sepal.width'])
```



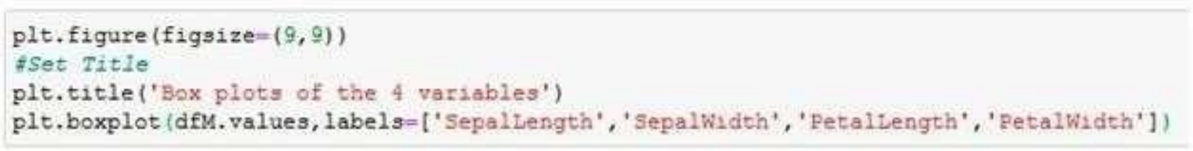
Plotting box plots of all variables in one frame :

Since the box plot is for continuous variables, firstly create a data frame without the column ‘variety’. Then drop the column from the DataFrame using the *drop( )* function and specify axis=1 to indicate it.

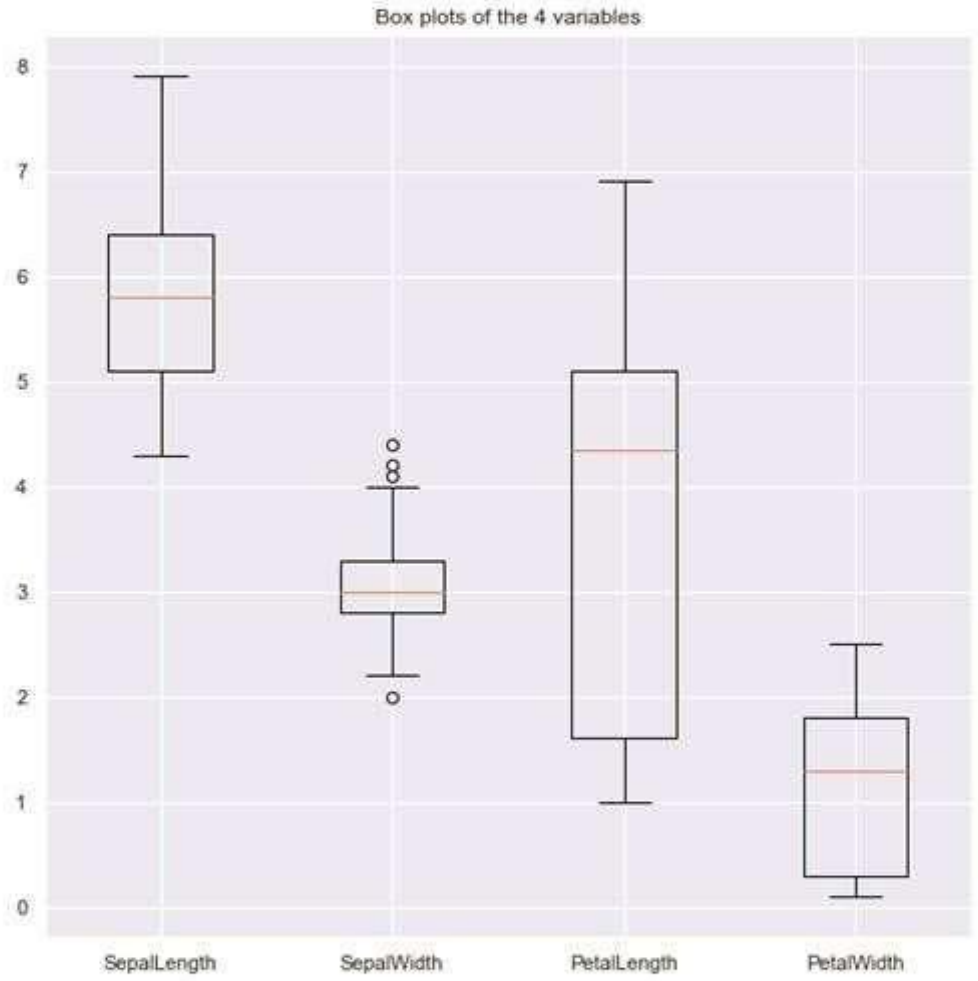
Removing the column with categorical variables



```
dfM=df.drop('variety',axis=1)
```

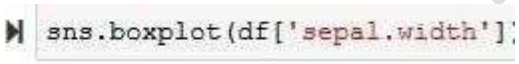


```
plt.figure(figsize=(9,9))
#Set Title
plt.title('Box plots of the 4 variables')
plt.boxplot(dfM.values,labels=['SepalLength','SepalWidth','PetalLength','PetalWidth'])
```

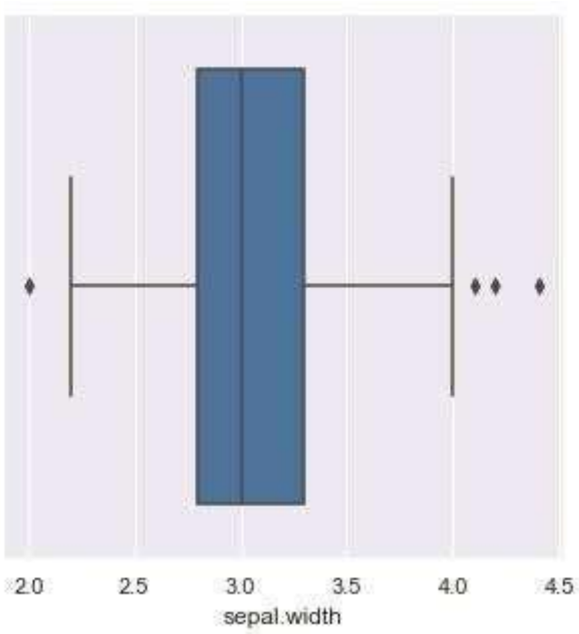


In matplotlib, mention the labels separately to display it in the output.

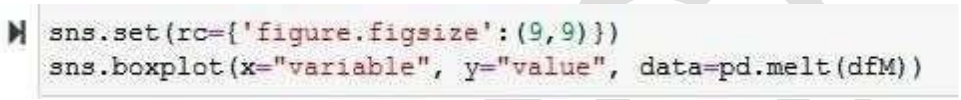
The plotting box plot in seaborn :



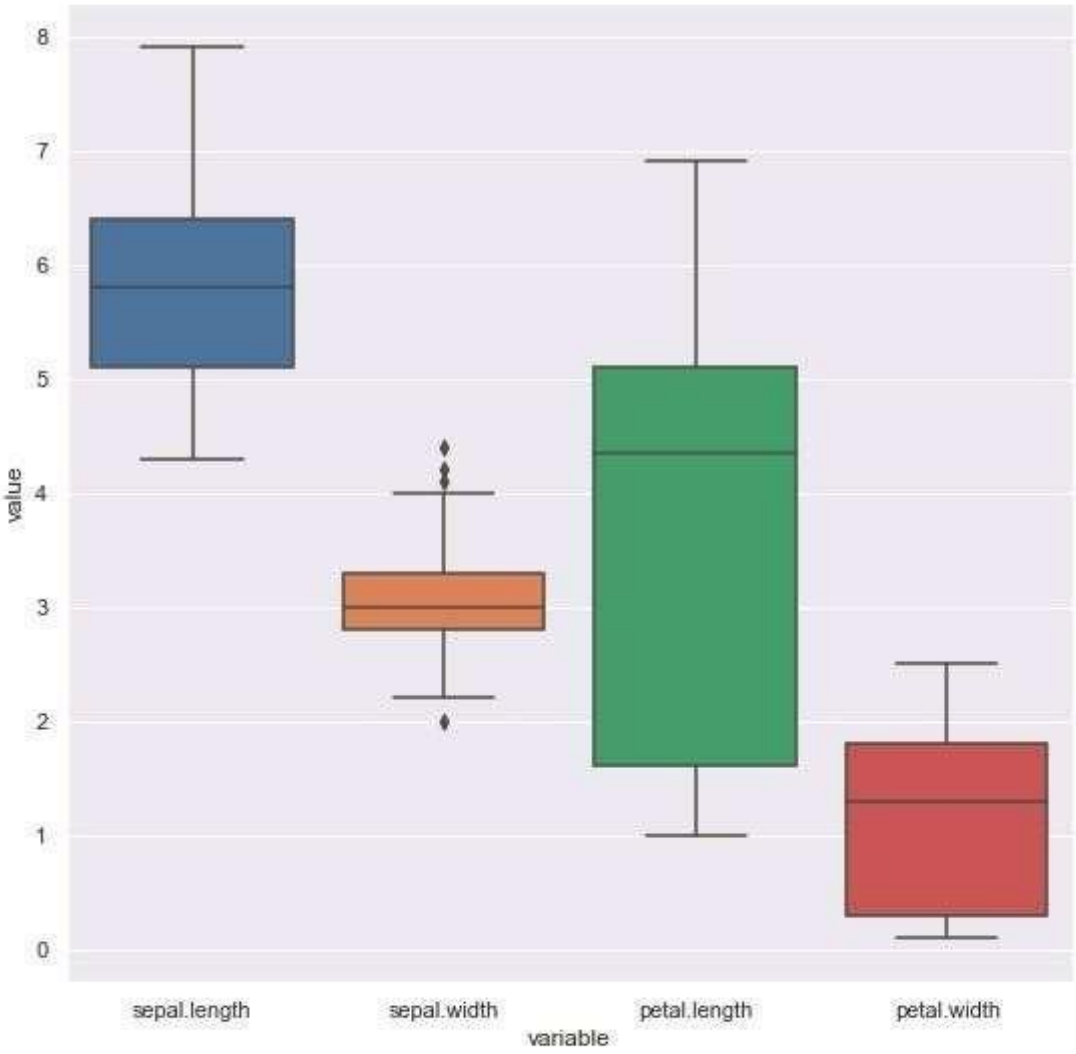
```
sns.boxplot(df['sepal.width'])
```



Plotting the box plots of all variables in one frame :



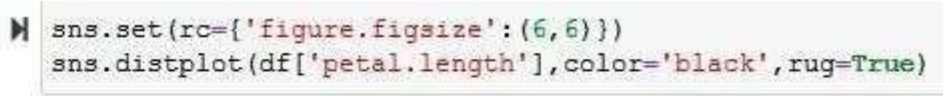
```
sns.set(rc={'figure.figsize':(9,9)})
sns.boxplot(x="variable", y="value", data=pd.melt(dfM))
```



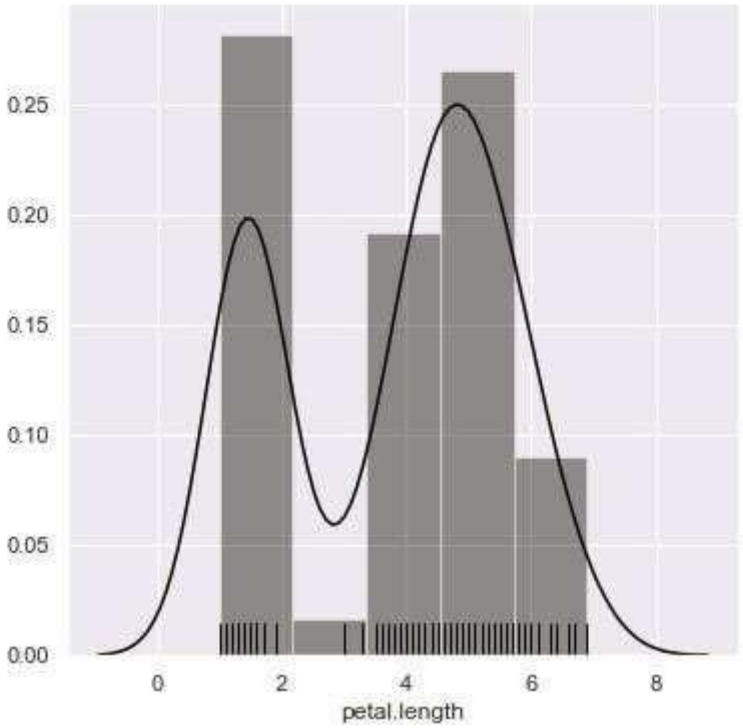
Apply the pandas function *pd.melt()* on the modified data frame which is then passed onto the *sns.boxplot()* function.

9. distplot() :

The *distplot()* function of seaborn library was earlier mentioned under rug plot section. This function combines the matplotlib *hist()* function with the seaborn *kdeplot()* and *rugplot()* functions.

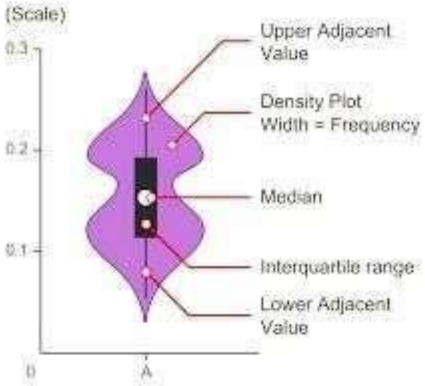


```
sns.set(rc={'figure.figsize':(6,6)})
sns.distplot(df['petal.length'],color='black',rug=True)
```

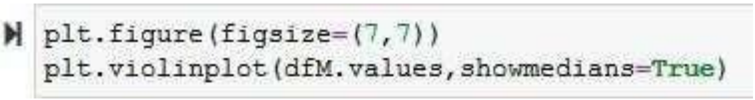


10. VIOLIN PLOTS :

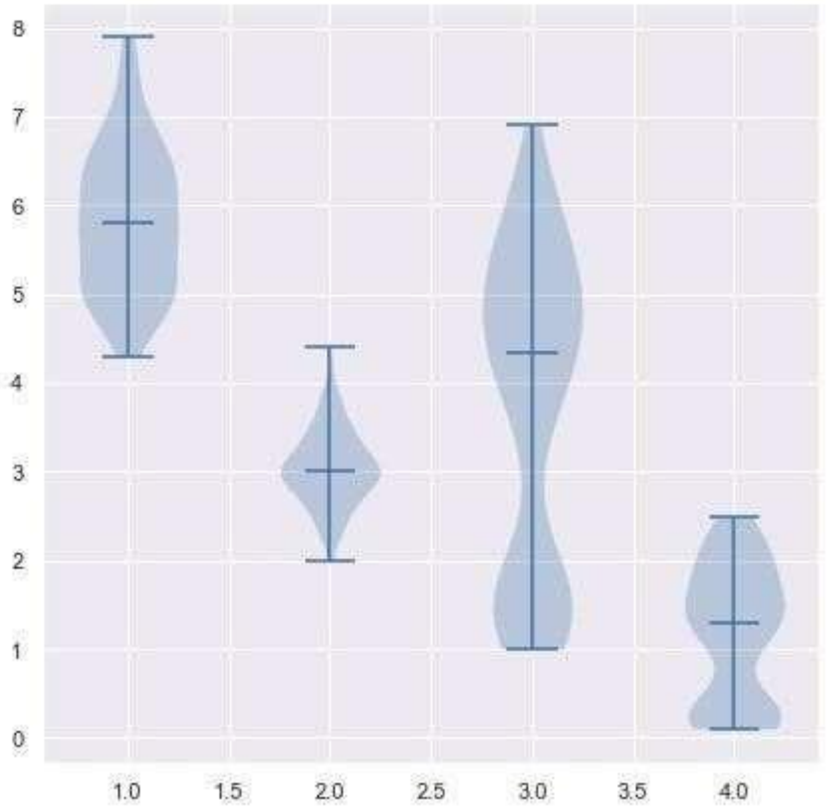
The Violin plot is very much similar to a box plot, with the addition of a rotated kernel density plot on each side. It shows the distribution of quantitative data across several levels of one (or more) categorical variables such that those distributions can be compared.



(Source : r-bloggers.com)

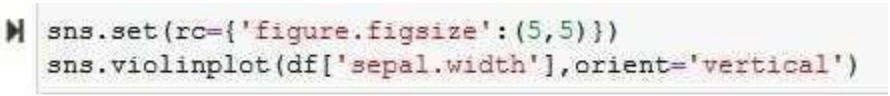


```
plt.figure(figsize=(7,7))
plt.violinplot(dfM.values,showmedians=True)
```

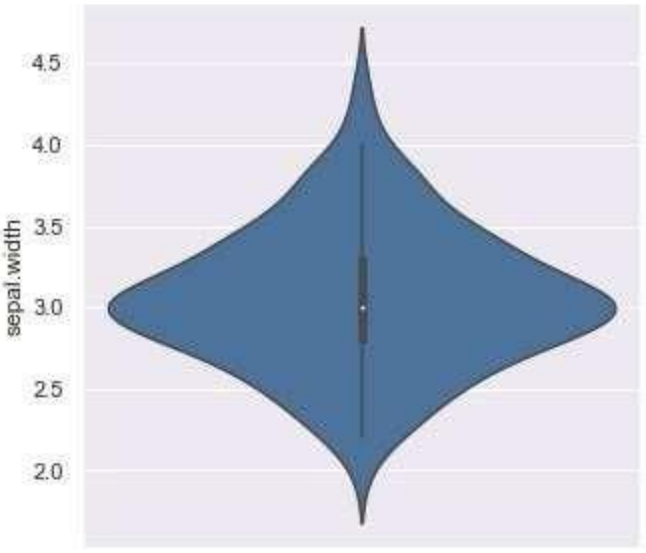


We use *plt.violinplot( )* function. The Boolean parameter ‘showmedians’ is set to True, due to which the medians are marked for every variable. The violin plot helps to understand the estimated density of the variable.

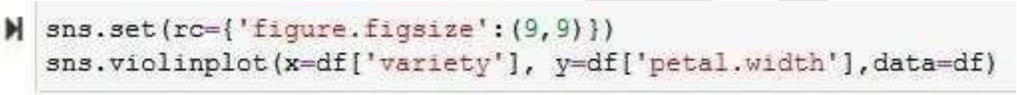
In the seaborn library too, the function used to plot a violin plot is similar.



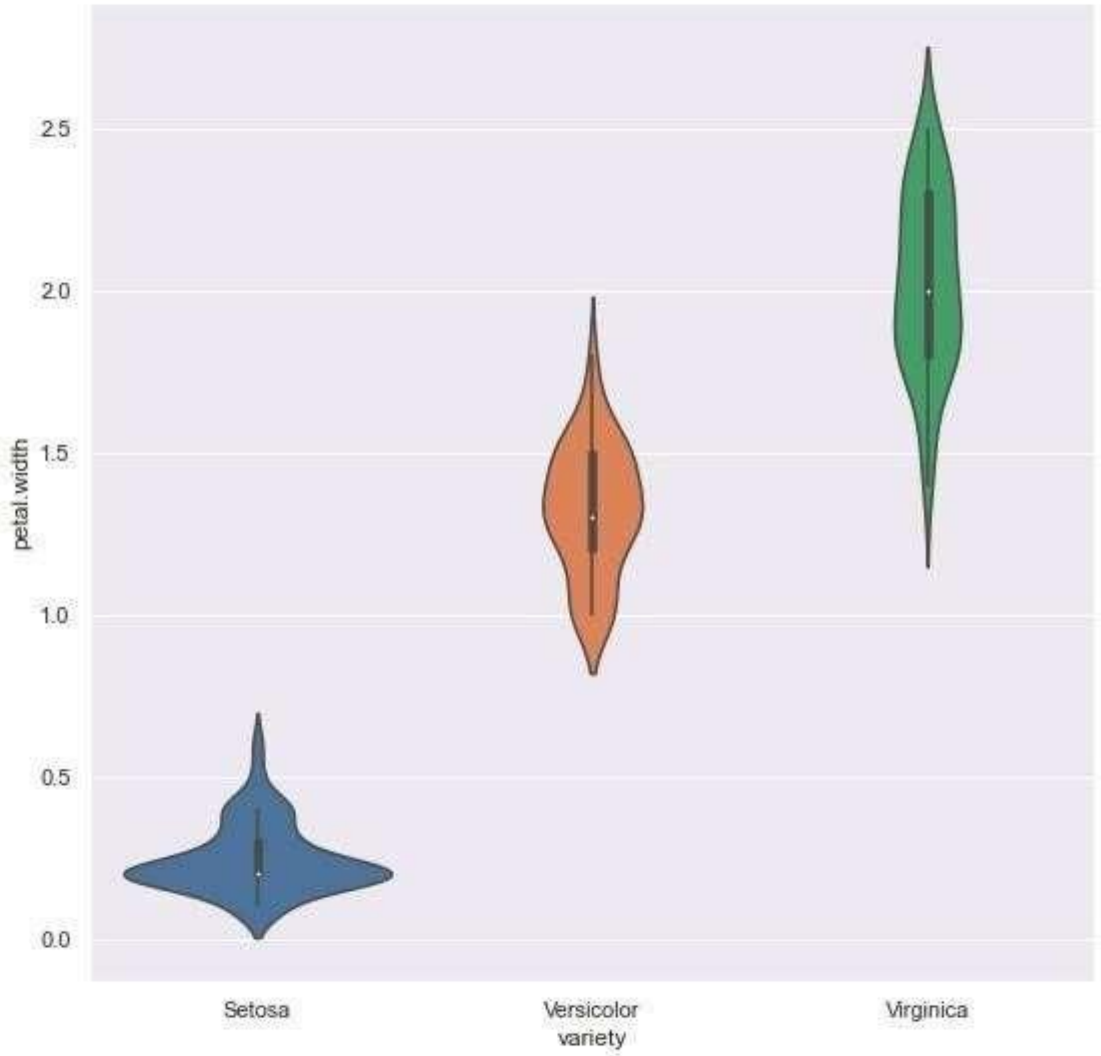
```
sns.set(rc={'figure.figsize':(5,5)})
sns.violinplot(df['sepal.width'],orient='vertical')
```



Comparing the variable ‘sepal.width’ according to the ‘variety’ of species mentioned in the dataset :



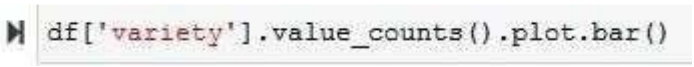
```
sns.set(rc={'figure.figsize':(9,9)})
sns.violinplot(x=df['variety'], y=df['petal.width'],data=df)
```



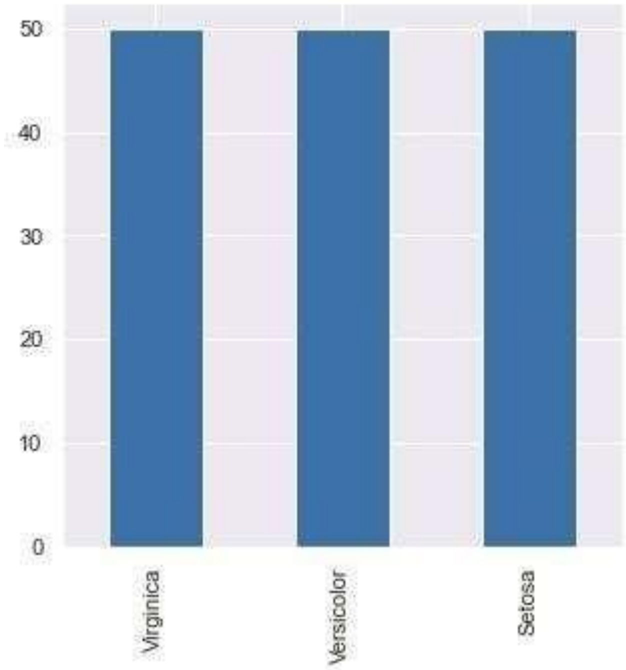
VISUALIZING CATEGORICAL VARIABLES :

11. BAR CHART :

The bar plot is a univariate data visualization plot on a twodimensional axis. One axis is the category axis indicating the category, while the second axis is the value axis that shows the numeric value of that category, indicated by the length of the bar. The *plot.bar()* function plots a bar plot of a categorical variable. The *value_counts()* returns a series containing the counts of unique values in the variable.

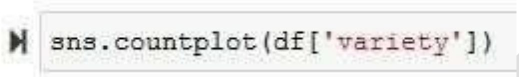


```
df['variety'].value_counts().plot.bar()
```

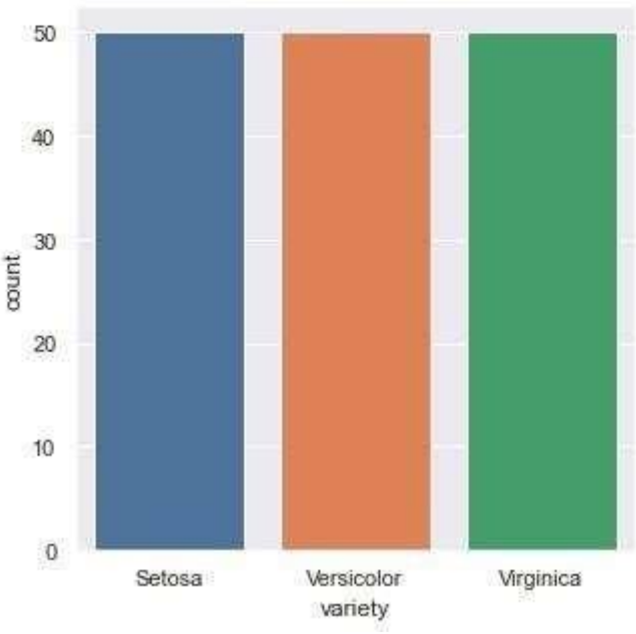


The *countplot()* function of the seaborn library obtains a similar bar plot. There is no need to separately calculate the count when using the *sns.countplot()* function.

Since the variety is equally distributed, we obtain bars with equal heights.



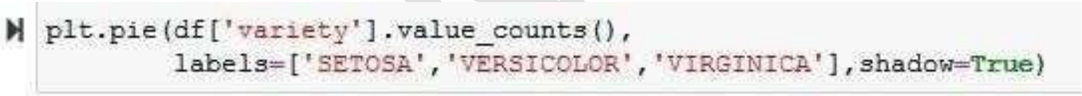
```
sns.countplot(df['variety'])
```



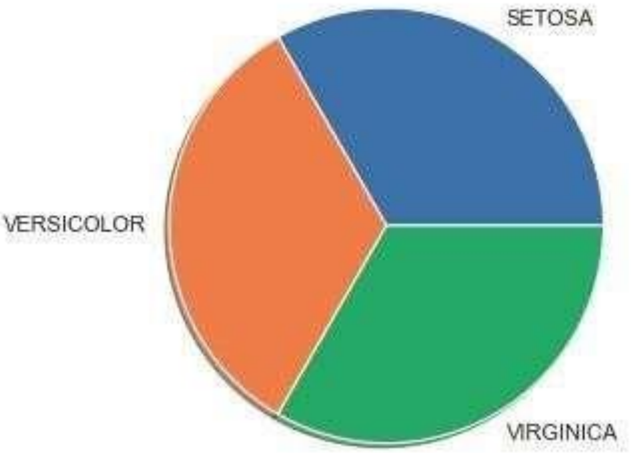
12. PIE CHART :

A pie chart is the most common way used to visualize the numerical proportion occupied by each of the categories.

Use the *plt.pie()* function to plot a pie chart. Since the categories are equally distributed, divide the sections in the pie chart is equally. Then add the labels by passing the array of values to the ‘labels’ parameter.

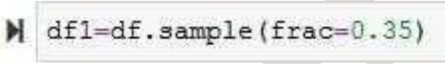


```
plt.pie(df['variety'].value_counts(),
        labels=['SETOSA','VERSICOLOR','VIRGINICA'],shadow=True)
```

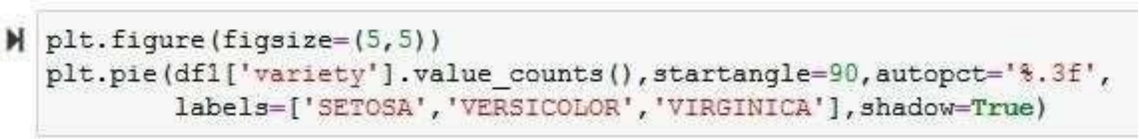


A random sample can be created using the *DataFrame.sample( )* function. The frac parameter of *sample()* function indicates the fraction of axis items to return.

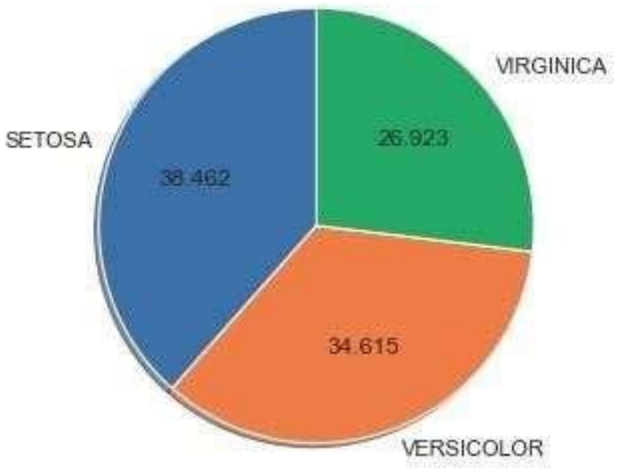
The ‘startangle’ parameter of the *pie()* function rotates everything counterclockwise at a specific angle. Further, the default value for startangle is 0. The ‘autopct’ parameter enables one to display the percentage value using Python string formatting.



```
df1=df.sample(frac=0.35)
```



```
plt.figure(figsize=(5,5))
plt.pie(df1['variety'].value_counts(),startangle=90,autopct='%.3f',
        labels=['SETOSA','VERSICOLOR','VIRGINICA'],shadow=True)
```



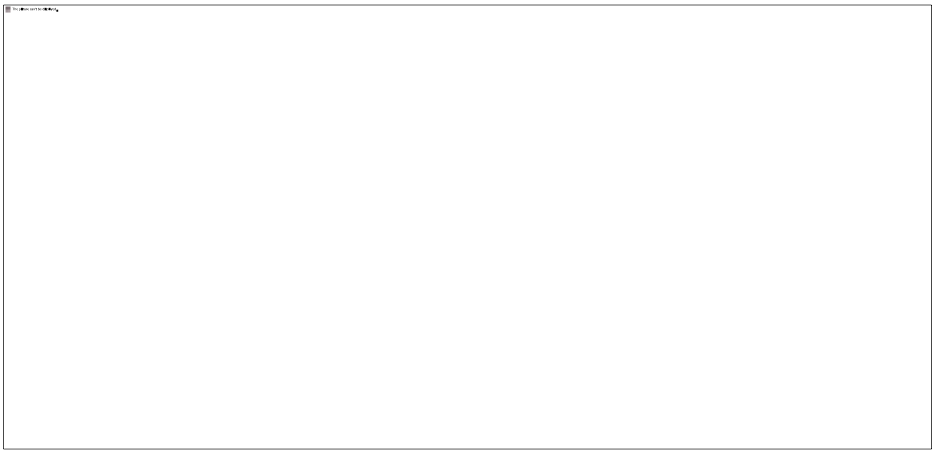
Most of the methods that help in the visualization of univariate data have been outlined in this article. As stated before the ability to see the structure and information carried by the data lies in its visual presentation.

What is Hypothesis Testing

Any data science project starts with exploring the data. When we perform an analysis on a sample through exploratory data analysis and inferential statistics we get information about the sample. Now, we want to use this information to predict values for the entire population.

Errors while making decisions

There are two possible types of error we could commit while performing hypothesis testing.



1) **Type1 Error** – This occurs when the null hypothesis is true but we reject it.The probability of type I error is denoted by alpha (α). Type 1 error is also known as the level of significance of the hypothesis test

2) **Type 2 Error** – This occurs when the null hypothesis is false but we fail to reject it. The probability of type II error is denoted by beta (β) What exactly is a test statistic?

A test statistic describes how closely the distribution of your data matches the distribution predicted under the null hypothesis of the statistical test you are using.

The **distribution** of data is how often each observation occurs, and can be described by its central tendency and variation around that central tendency. Different statistical tests predict different types of distributions, so it's important to choose the right statistical test for your hypothesis.

The test statistic summarizes your observed data into a single number using the central tendency, variation, sample size, and number of predictor variables in your statistical model.

Generally, the test statistic is calculated as the pattern in your data (i.e. the correlation between variables or difference between groups) divided by the variance in the data (i.e. the standard deviation).

ExampleYou are testing the relationship between temperature and flowering date for a certain type of apple tree. You use a long-term data set that tracks temperature and flowering dates from the past 25 years by randomly sampling 100 trees every year in an experimental field.

- **Null hypothesis:** There is no correlation between temperature and flowering date.

- **Alternate hypothesis:** There is a correlation between temperature and flowering date.

To test this hypothesis you perform a regression test, which generates a *t*value as its test statistic. The *t*-value compares the observed correlation between these variables to the null hypothesis of zero correlation.

Types of test statistics

Below is a summary of the most common test statistics, their hypotheses, and the types of statistical tests that use them.

Different statistical tests will have slightly different ways of calculating these test statistics, but the underlying hypotheses and interpretations of the test statistic stay the same.

***t*-value**	**Null:** The means of two groups are equal **Alternative:** The means of two groups are not equal	- *T*-test - Regression tests
***z*-value**	**Null:** The means of two groups are equal **Alternative:**The means of two groups are not equal	- *Z*-test
***F*-value**	**Null:** The variation among two or more groups is greater than or equal to the variation between the groups **Alternative:** The variation among two or more groups is smaller than the variation between the groups	- ANOVA - ANCOVA - MANOVA
X^2-value	**Null:** Two samples are independent **Alternative:** Two samples are not independent (i.e. they are correlated)	

Types of test statistics
Below is a summary of the most common test statistics, their

hypotheses, and the types of statistical tests that use them.

Different statistical tests will have slightly different ways of calculating these test statistics, but the underlying hypotheses and interpretations of the test statistic stay the same.

***t*-value**	**Null:** The means of two groups are equal **Alternative:** The means of two groups are not equal	• *T*-test • Regression tests
***z*-value**	**Null:** The means of two groups are equal **Alternative:**The means of two groups are not equal	• *Z*-test
***F*-value**	**Null:** The variation among two or more groups is greater than or equal to the variation between the groups **Alternative:** The variation among two or more groups is smaller than the variation between the groups	• ANOVA • ANCOVA • MANOVA
***X*^{2}-value**	**Null:** Two samples are independent **Alternative:** Two samples are not independent (i.e. they are correlated)	

Interpreting test statistics

For any combination of sample sizes and number of predictor variables, a statistical test will produce a predicted distribution for the test statistic. This shows the most likely range of values that will occur if your data follows the null hypothesis of the statistical test.

The more extreme your test statistic – the further to the edge of the range of predicted test values it is – the less likely it is that your data could have been generated under the null hypothesis of that statistical test.

The agreement between your calculated test statistic and the predicted values is described by the *p*-**value**. The smaller the *p*value, the less likely your test statistic is to have occurred under the null hypothesis of the statistical test.

Because the test statistic is generated from your observed data, this ultimately means that the smaller the *p*-value, the less likely it is that your data could have occurred if the null hypothesis was true.

ExampleYour calculated *t*-value of 2.36 is far from the expected range of *t*-values under the null hypothesis, and the *p*-value is < 0.01. This means that you would expect to see a *t*-value as large or larger than 2.36 less than 1% of the time if the true relationship between temperature and flowering dates was 0.

Therefore, it is statistically unlikely that your observed data could have occurred under the null hypothesis. Using a significance threshold of 0.05, you can say that the result is **statistically significant**.

Interpreting test statistics

For any combination of sample sizes and number of predictor variables, a statistical test will produce a predicted distribution for the test statistic. This shows the most likely range of values that will occur if your data follows the null hypothesis of the statistical test.

The more extreme your test statistic – the further to the edge of the range of predicted test values it is – the less likely it is that your data could have been generated under the null hypothesis of that statistical test.

The agreement between your calculated test statistic and the predicted values is described by the *p*-**value**. The smaller the *p*value, the less likely your test statistic is to have occurred under the null hypothesis of the statistical test.

Because the test statistic is generated from your observed data, this ultimately means that the smaller the *p*-value, the less likely it is that your data could have occurred if the null hypothesis was true.

Example Your calculated *t*-value of 2.36 is far from the expected range of *t*-values under the null hypothesis, and the *p*-value is < 0.01. This means that you would expect to see a *t*-value as large or larger than 2.36 less than 1% of the time if the true relationship between temperature and flowering dates was 0.

Therefore, it is statistically unlikely that your observed data could have occurred under the null hypothesis. Using a significance threshold of 0.05, you can say that the result is **statistically significant**.

What is Multivariate Analysis?

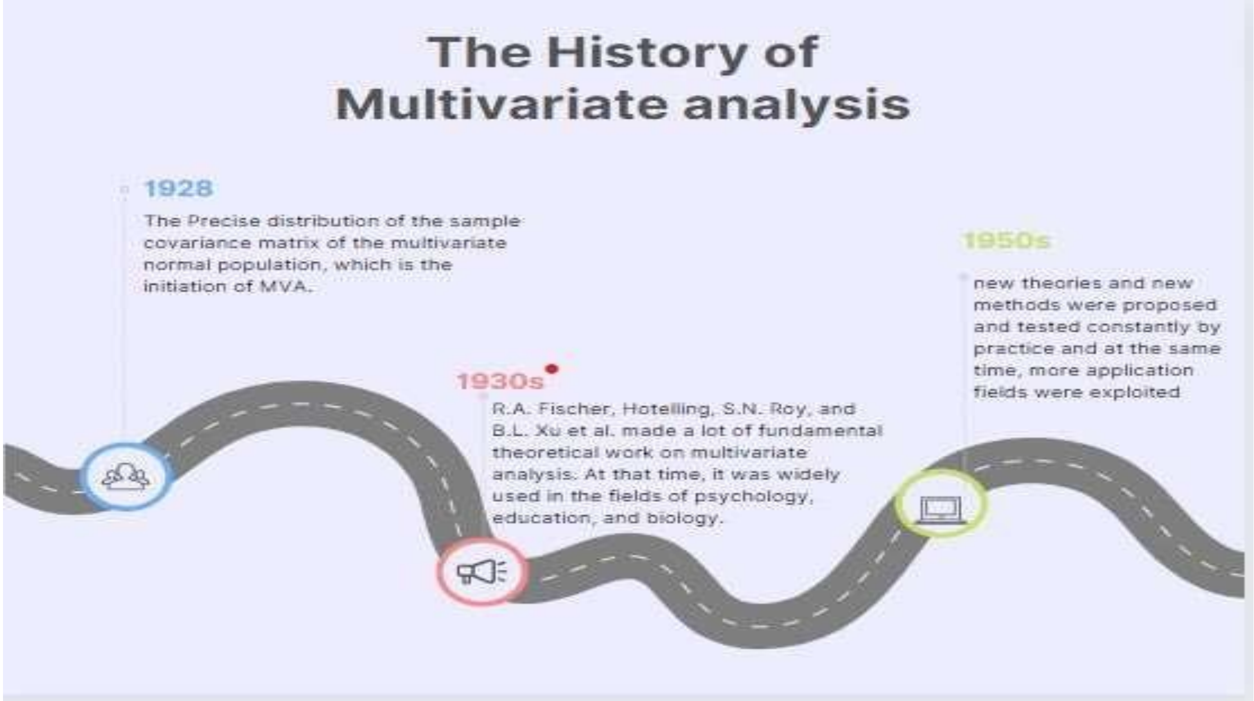
Multivariate Analysis is defined as a process of involving multiple dependent variables resulting in one outcome. This explains that the majority of the problems in the real world are Multivariate. For example, we cannot predict the weather of any year based on the season. There are multiple factors like pollution, humidity, precipitation, etc. Here, we will introduce you to multivariate analysis, its history, and its application in different fields. Also, take up a Multivariate Time Series Forecasting In R to learn more about the concept.

The History of Multivariate analysis

In 1928, Wishart presented his paper. The Precise distribution of the sample covariance matrix of the multivariate normal population, which is the initiation of MVA.

In the 1930s, R.A. Fischer, Hotelling, S.N. Roy, and B.L. Xu et al. made a lot of fundamental theoretical work on multivariate analysis. At that time, it was widely used in the fields of psychology, education, and biology.

In the middle of the 1950s, with the appearance and expansion of computers, multivariate analysis began to play a big role in geological, meteorological. Medical and social and science. From then on, new theories and new methods were proposed and tested constantly by practice and at the same time, more application fields were exploited. With the aids of modern computers, we can apply the methodology of multivariate analysis to do rather complex statistical analyses.



The History of Multivariate analysis

1928

The Precise distribution of the sample covariance matrix of the multivariate normal population, which is the initiation of MVA.

1930s

R.A. Fischer, Hotelling, S.N. Roy, and B.L. Xu et al. made a lot of fundamental theoretical work on multivariate analysis. At that time, it was widely used in the fields of psychology, education, and biology.

1950s

new theories and new methods were proposed and tested constantly by practice and at the same time, more application fields were exploited

Multivariate analysis: An overview

Suppose a project has been assigned to you to predict the sales of the company. You cannot simply say that ‘X’ is the factor which will affect the sales.

We know that there are multiple aspects or variables which will impact sales. To analyze the variables that will impact sales majorly, can only be found with multivariate analysis. And in most cases, it will not be just one variable.

Like we know, sales will depend on the category of product, production capacity, geographical location, marketing effort, presence of the brand in the market, competitor analysis, cost of the product, and multiple other variables. Sales is just one example; this study can be implemented in any section of most of the fields.

Multivariate analysis is used widely in many industries, like healthcare. In the recent event of COVID-19, a team of data scientists predicted that Delhi would have more than 5 lakh COVID-19 patients by the end of July 2020. This analysis was based on multiple variables like government decision, public behavior, population, occupation, public transport, healthcare services, and overall immunity of the community. Check out Multivariate Time Series on Covid Data for more information.

As per the Data Analysis study by Murtaza Haider of Ryerson university on the coast of the apartment and what leads to an increase in cost or decrease in cost, is also based on multivariate analysis. As per that study, one of the major factors was transport infrastructure. People were thinking of buying a home at a location which provides better transport, and as per the analyzing team, this is one of the least thought of variables at the start of the study. But with analysis, this came in few final variables impacting outcome.

Multivariate analysis is part of Exploratory data analysis. Based on MVA, we can visualize the deeper insight of multiple variables.

There are more than 20 different methods to perform multivariate analysis and which method is best depends on the type of data and the problem you are trying to solve.

Multivariate analysis (**MVA**) is a Statistical procedure for analysis of data involving more than one type of measurement or observation. It may also mean solving problems where more than one dependent variable is analyzed simultaneously with other variables.

Advantages and Disadvantages of Multivariate Analysis

Advantages

- The main advantage of multivariate analysis is that since it considers more than one factor of independent variables that influence the variability of dependent variables, the conclusion drawn is more accurate.
- The conclusions are more realistic and nearer to the real-life situation.

Disadvantages

- The main disadvantage of MVA includes that it requires rather complex computations to arrive at a satisfactory conclusion.
- Many observations for a large number of variables need to be collected and tabulated; it is a rather time-consuming process.

Classification Chart of Multivariate Techniques

Selection of the appropriate multivariate technique depends upon-

a) Are the variables divided into independent and dependent classification?

b) If Yes, how many variables are treated as dependents in a single analysis?

c) How are the variables, both dependent and independent measured?

Multivariate analysis technique can be classified into two broad categories viz., This classification depends upon the question: are the involved variables dependent on each other or not?

If the answer is yes: We have **Dependence methods.**
If the answer is no: We have **Interdependence methods.**

Dependence technique: Dependence Techniques are types of multivariate analysis techniques that are used when one or more of the variables can be identified as dependent variables and the remaining variables can be identified as independent.

Multivariate Analysis of Covariance (MANCOVA)

Multivariate analysis of covariance (MANCOVA) is a statistical technique that is the extension of analysis of covariance (ANCOVA). Basically, it is the multivariate analysis of variance (MANOVA) with a covariate(s).). In MANCOVA, we assess for statistical differences on multiple continuous dependent variables by an independent grouping variable, while controlling for a third variable called the covariate; multiple covariates can be used, depending on the sample size. Covariates are added so that it can reduce error terms and so that the analysis eliminates the covariates' effect on the relationship between the independent grouping variable and the continuous dependent variables.

Questions answered:

Do the various school assessments vary by grade level after controlling for gender?

Do the rates of graduation among certain state universities differ by degree type after controlling for tuition costs?

Which diseases are better treated, if at all, by either X drug or Y drug after controlling for length of disease and participant age?

Discover How We Assist to Edit Your Dissertation Chapters

Assumptions:

In multivariate analysis of covariance (MANCOVA), all assumptions are the same as in MANOVA, but one more additional assumption is related to covariate:

1. **Independent Random Sampling:** MANCOVA assumes that the observations are independent of one another, there is not any pattern for the selection of the sample, and that the sample is completely random.
2. **Level and Measurement of the Variables:** MANCOVA assumes that the independent variables are categorical and the dependent variables are continuous or scale variables. Covariates can be either continuous, ordinal, or dichotomous.
3. **Absence of multicollinearity:** The dependent variables cannot be too correlated to each other. Tabachnick & Fidell (2012) suggest that no correlation should be above *r* = .90..
4. **Normality:** Multivariate normality is present in the data.
5. **Homogeneity of Variance:** Variance between groups is equal.
6. **Relationship between covariate(s) and dependent variables**: in choosing what covariates to use, it is common practice to assess if a statistical relationship exists between the covariate(s) and the dependent variables; this can be done through correlation analyses.

Key concepts and terms:

- **Levene's Test of Equality of Variance:** Used to examine whether or not the variance between independent variable groups are equal; also known as homogeneity of variance Non-significant values of Levene's test indicate equal variance between groups.
- **Box's M Test:** Used to know the equality of covariance between the groups. This is the equivalent of a multivariate homogeneity of variance. Usually, significance for this test is determined at α = .001 because this test is considered highly sensitive.
- **Partial eta square:** Partial eta square (η^2) shows how much variance is explained by the independent variable. It is used as the effect size for the MANOVA model.

- **Post hoc test:** If there is a significant difference between groups, then post hoc tests are performed to determine where the significant differences lie (i.e., which specific independent variable level significantly differs from another).
- **Multivariate F-statistics:** The F- statistic is derived by essentially dividing the means sum of the square (*SS*) for the source variable by the source variable mean error (*ME or MSE)*.
- **Covariate:** a Covariate is basically a control variable, which is uncorrelated with the independent variables and correlated with the dependent variables. Covariates areused to reduce the error term.
  SPSS: The following steps have to be performed for multivariate analysis of covariance (MANCOVA):
- **SPSS:** the MANCOVACan be performed using the analysis menu, selecting the “GLM” option, and then choosing the “Multivariate” option from the GLM option.

Linear algebra using Python:

Scalars

Scalars are single numbers and are an example of a 0th-order tensor. In mathematics it is necessary to describe the set of values to which a scalar belongs. The notation x∈R states that the (lowercase) scalar value x is an element of (or member of) the set of real-valued numbers, R.

There are various sets of numbers of interest within machine learning. N represents the set of positive integers (1,2,3,…). Z represents the integers, which include positive, negative and zero values. Q represents the set of *rational* numbers that may be expressed as a fraction of two integers.

Vectors

Vectors are ordered arrays of single numbers and are an example of 1storder tensor. Vectors are members of objects known as **vector spaces**. A vector space can be thought of as the entire collection of *all* possible vectors of a particular length (or dimension). The three-dimensional realvalued vector space, denoted by

R3 is often used to represent our realworld notion of three-dimensional space mathematically.

More formally a vector space is an n-dimensional Cartesian product of a set with itself, along with proper definitions on how to add vectors and multiply them with scalar values. If all of the scalars in a vector are real-valued then the notation x∈Rn states that the (boldface lowercase) vector value x is a member of the n-dimensional vector space of real numbers, Rn.

Sometimes it is necessary to identify the *components* of a vector explicitly. The ith scalar element of a vector is written as xi. Notice that this is nonbold lowercase since the element is a scalar. An n-dimensional vector itself can be explicitly written using the following notation:

x=[x1x2⋮xn]

Given that scalars exist to represent values why are vectors necessary? One of the primary use cases for vectors is to represent physical quantities that have both a *magnitude* and a *direction*. Scalars are only capable of representing magnitudes.

For instance scalars and vectors encode the difference between the *speed* of a car and its *velocity*. The velocity contains not only its speed but also its direction of travel. It is not difficult to imagine many more physical quantities that possess similar characteristics such as gravitational and electromagnetic forces or wind velocity.

In machine learning vectors often represent *feature vectors*, with their individual components specifying how important a particular feature is. Such features could include relative importance of words in a text document, the intensity of a set of pixels in a two-dimensional image or historical price values for a cross-section of financial instruments.

Matrices

Matrices are rectangular arrays consisting of numbers and are an example of 2nd-order tensors. If m and n are positive integers, that is m,n∈N then the m×n matrix contains mn numbers, with m rows and n columns.

If all of the scalars in a matrix are real-valued then a matrix is denoted with uppercase boldface letters, such as A∈Rm×n. That is the matrix lives in a m×n-dimensional real-valued vector space. Hence matrices are really vectors that are just written in a two-dimensional table-like manner.

Its components are now identified by two indices i and j. i represents the index to the matrix row, while j represents the index to the matrix column.
Each component of A is identified by aij.

The full m×n matrix can be written as:

A=[a11a12a13…a1na21a22a23…a2na31a32a33…a3n⋮⋮⋮⋱⋮am1am2am3…amn]

It is often useful to abbreviate the full matrix component display into the following expression:

A=[aij]m×n

Where aij is referred to as the (i,j)-element of the matrix A. The subscript of m×n can be dropped if the dimension of the matrix is clear from the context.

Note that a *column vector* is a size m×1 matrix, since it has m rows and 1 column. Unless otherwise specified all vectors will be considered to be column vectors.

Matrices represent a type of function known as a linear map. Based on rules that will be outlined in subsequent articles, it is possible to define multiplication operations between matrices or between matrices and vectors. Such operations are immensely important across the physical sciences, quantitative finance, computer science and machine learning.

Matrices can encode geometric operations such as rotation, reflection and transformation. Thus if a collection of vectors represents the vertices of a three-dimensional geometric model in Computer Aided Design software then multiplying these vectors individually by a pre-defined rotation matrix will output new vectors that represent the locations of the rotated vertices. This is the basis of modern 3D computer graphics.

In deep learning neural network weights are stored as matrices, while feature inputs are stored as vectors. Formulating the problem in terms of linear algebra allows compact handling of these computations. By casting the problem in terms of tensors and utilising the machinery of linear algebra, rapid training times on modern GPU hardware can be obtained.

Tensors

The more general entity of a tensor encapsulates the scalar, vector and the matrix. It is sometimes necessary—both in the physical sciences and machine learning—to make use of tensors with order that exceeds two.

In theoretical physics, and general relativity in particular, the Riemann curvature tensor is a 4th-order tensor that describes the local curvature of spacetime. In machine learning, and deep learning in particular, a 3rdorder tensor can be used to describe the intensity values of multiple channels (red, green and blue) from a two-dimensional image.

Tensors will be identified in this series of posts via the boldface sans-serif notation, A. For a 3rd-order tensor elements will be given by aijk, whereas for a 4th-order tensor elements will be given by aijkl.

Gradients

- A gradient is a derivative of a function that has more than one input variable. It is a term used to refer to the derivative of a function from the perspective of the field of linear algebra. Specifically when linear algebra meets calculus, called vector calculus. The gradient is the generalization of the derivative to multivariate functions.

What is the the gradient vector of the following function?

f(x,y,z)=18xyz+50x2z+15y2 Possible Answers:

∇xf(x)=⎡⎣⎢⎢18xz+30y18yz+100xz18xy+50x2⎤⎦⎥⎥

∇xf(x)=⎡⎣⎢⎢18yz+100xz18xz+30y18xy+50x2⎤⎦⎥⎥

∇xf(x)=⎡⎣⎢018xy+50x20⎤⎦⎥

∇xf(x)=⎡⎣⎢⎢1118yz+100xz⎤⎦⎥⎥

∇xf(x)=⎡⎣⎢⎢18xz+30y18xy+50x218yz+100xz⎤⎦⎥⎥

Correct answer:

∇xf(x)=⎡⎣⎢⎢18yz+100xz18xz+30y18xy+50x2⎤⎦⎥⎥

Explanation:

Recall that

∇xf(x)=⎡⎣⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢∂f∂x1∂f∂x2∂f∂x3⋮∂f∂xn⎤⎦⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥

All we need to do is calculate 3 partial derivatives, and put them into this form.

∂f∂x=18yz+100xz

∂f∂y=18xz+30y

∂f∂z=18xy+50x2

Put these into vector form to get

∇xf(x)=⎡⎣⎢⎢18yz+100xz18xz+30y18xy+50x2⎤⎦⎥⎥

Report an Error

Example Question #2 : The Gradient

Find the gradient vector of the following function.

f(x,y)=x2y2+10ln(xy) Possible Answers:

F⎣⎢⎢1+10x1+10y⎦⎥⎥⎥

F⎣⎢2xy22yx2⎤⎦⎥

F⎣⎢⎢2xy2+10x2yx2+10y⎤⎦⎥⎥

F⎣⎢⎢10x10y⎤⎦⎥⎥

Correct answer:
F⎣⎢⎢2xy2+10x2yx2+10y⎤⎦⎥⎥

Explanation

To find the gradient vector, we need to find the partial derivatives in respect to x and y.

∂f∂x=2xy2+10x

∂f∂y=2yx2+10y

Then our final answer looks like

F⎣⎢⎢2xy2+10x2yx2+10y⎤⎦⎥⎥

Report an Error

Determine the gradient, ∇, of the functionf(x,y)=5xcos(y)

Possible Answers:

[5cos(y)−5xsin(y)]

[5cos(y)−5xsin(y)]

[0−5sin(y)−5sin(y)−5xcos(y)]

[−5sin(y)]

Correct answer:

[5cos(y)−5xsin(y)]

Explanation:

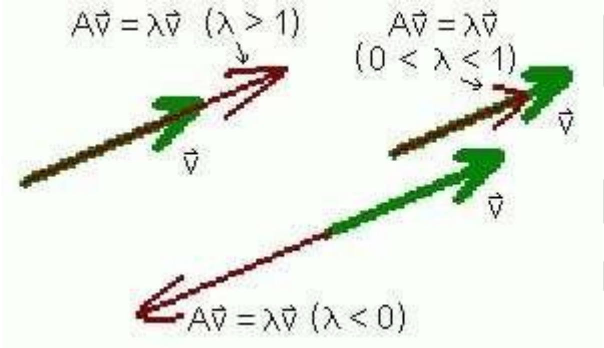
The gradient of a function is a vector of first derivatives taken withrespect to its constituent variables. The form is as follows:F⎡⎣⎢⎢⎢⎢⎢⎢dfdx1dfdx2...dfdxn⎤⎦⎥⎥⎥⎥⎥⎥Considering our function: f(x,y)=5xcos(y)And utilizing derivative rules:(f∘g)′=(f′∘g)⋅g′d[cos(u)]=−sin(u)dud[au]=auduln(a)∇f=[5cos(y)−5xsin(y)]

EIGENVALUES AND EIGENVECTORS

Page

We review here the basics of computing eigenvalues and eigenvectors. Eigenvalues and eigenvectors play a prominent role in the study of ordinary differential equations and in many applications in the physical sciences. Expect to see them come up in a variety of contexts!

Definitions


$A\vec{v} = \lambda\vec{v}$ $(\lambda > 1)$
$A\vec{v} = \lambda\vec{v}$
$(0 < \lambda < 1)$
$\vec{v}$
$\vec{v}$
$\vec{v}$
$A\vec{v} = \lambda\vec{v}$ $(\lambda < 0)$

Let AA be an n×nn×n matrix. The number λλ is an **eigenvalue** of AA if there exists a non-zero vector vv such that

Av=λv.Av=λv.

In this case, vector vv is called an **eigenvector** of AA corresponding to λλ.

Computing Eigenvalues and Eigenvectors

We can rewrite the condition Av=λvAv=λv as

(A−λI)v=0.(A−λI)v=0.

where II is the n×nn×n identity matrix. Now, in order for a *non-zero* vector vv to satisfy this equation, A–λIA–λI must *not* be invertible.

Otherwise, if A–λIA–λI has an inverse,

(A–λI)$^{-1}$(A–λI)vv==(A–λI)$^{-1}$00.(A–λI)−1(A–λI)v=(A–λI)−10v=0.

But we are looking for a non-zero vector vv. That is, the determinant of $\mathrm{A}–\lambda\mathrm{I}$A– λI must equal 0. We call $\mathrm{p}(\lambda)=\det(\mathrm{A}–\lambda\mathrm{I})$p(λ)=det(A–λI) the **characteristic polynomial** of AA. The eigenvalues of AA are simply the roots of the characteristic polynomial of AA.

Example

Let $\mathrm{A}=[2-1-4-1]$A=[2−4−1−1]. Then

$$\mathrm{p}(\lambda)====\det[2-\lambda-1-4-1-\lambda](2-\lambda)(-1-\lambda)-(-4)(-1)\lambda_2-\lambda-6(\lambda-3)(\lambda+2).$$ p(λ)

=det[2−λ−4−1−1−λ]=(2−λ)(−1−λ)−(−4)(−1)=λ2−λ−6=(λ−3)(λ+2).

Thus, $\lambda_1=3$λ1=3 and $\lambda_2=-2$λ2=−2 are the eigenvalues of AA.

To find eigenvectors $\mathrm{v}=\left[\begin{matrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{matrix}\right]$v=[v1v2⋮vn] corresponding to an eigenvalue λλ, we simply solve the system of linear equations given by $(\mathrm{A}-\lambda\mathrm{I})\mathrm{v}=0.$(A−λI)v=0.

Example

The matrix $\mathrm{A}=[2-1-4-1]$A=[2−4−1−1] of the previous example has eigenvalues $\lambda_1=3$λ1=3 and $\lambda_2=-2$λ2=−2. Let's find the eigenvectors corresponding to $\lambda_1=3$λ1=3. Let $\mathrm{v}=[v_1 v_2]$v=[v1v2]. Then $(\mathrm{A}-3\mathrm{I})\mathrm{v}=0$(A−3I)v=0 gives us

$$[2-3-1-4-1-3][v_1 v_2]=[00],$$ [2−3−4−1−1−3][v1v2]=[00],

from which we obtain the duplicate equations

$$-v_1-4v_2-v_1-4v_2==00.$$ −v1−4v2=0−v1−4v2=0.

If we let $v_2=t$v2=t, then $v_1=-4t$v1=−4t. All eigenvectors corresponding to $\lambda_1=3$λ1=3 are multiples of $[-41]$[−41] and thus the eigenspace corresponding to $\lambda_1=3$λ1=3 is given by the span of $[-41]$[−41]. That is, $\{[-41]\}${[−41]} is a **basis** of the eigenspace corresponding to $\lambda_1=3$λ1=3.

Repeating this process with $\lambda_2=-2$λ2=−2, we find that

$$4v_1-4V_2-v_1+v_2==00$$ 4v1−4V2=0−v1+v2=0

If we let $v_2=t$v2=t then $v_1=t$v1=t as well. Thus, an eigenvector corresponding to $\lambda_2=-2$λ2=−2 is $[11]$[11] and the eigenspace corresponding to $\lambda_2=-2$λ2=−2 is given by the span of $[11]$[11]. $\{[11]\}${[11]} is a basis for the eigenspace corresponding to $\lambda_2=-2$λ2=−2.

In the following example, we see a two-dimensional eigenspace.

Example

Let $\mathrm{A}=\left[\begin{matrix} 54-481-4168-11 \end{matrix}\right]$A=[5816418−4−4−11].

Then $\mathrm{p}(\lambda)=\det\left[\begin{matrix} 5-\lambda 4-481-\lambda-4168-11-\lambda \end{matrix}\right]=(\lambda-1)(\lambda+3)_2$p(λ)=det[5−λ8164 1−λ8−4−4−11−λ]=(λ−1)(λ+3)2 after some algebra!

Thus, $\lambda_1=1$λ1=1 and $\lambda_2=-3$λ2=−3 are the eigenvalues of AA.

Eigenvectors v=F⎣⎢⎢v1v2v3⎤⎦⎥⎥v=[v1v2v3] corresponding to λ1=1λ1=1 must satisfy 4v14v1−4v1+–8v24v2++–16v38v312v3===000.4v1+8v2+16v3=04v1+8v3=0−4v1–4v2–12v3=0.

Letting v3=tv3=t, we find from the second equation that v1=−2tv1=−2t, and then v2=−tv2=−t. All eigenvectors corresponding to λ1=1λ1=1 are multiples of F⎣⎢⎢−2−11⎤⎦⎥⎥[−2−11], and so the eigenspace corresponding to λ1=1λ1=1 is given by the span of F⎣⎢⎢−2−11⎤⎦⎥⎥[−2−11]. ⎧⎩⎨⎪⎪⎪⎪F⎣⎢⎢−2−11⎤⎦⎥⎥⎫⎭⎬⎪⎪⎪⎪ {[−2−11]} is a basis for the eigenspace corresponding to λ1=1λ1=1.

Eigenvectors corresponding to λ2=−3λ2=−3 must satisfy

8v14v1−4v1++–8v24v24v2++–16v38v38v3===000.8v1+8v2+16v3=04v1+4v2+8v3=0−4v1–4v2–8v3=0.

The equations here are just multiples of each other! If we let v3=tv3=t and v2=sv2=s, then v1=−s−2tv1=−s–2t. Eigenvectors corresponding to λ2=−3λ2=−3 have the form F⎣⎢⎢−110⎤⎦⎥⎥s+F⎣⎢⎢−201⎤⎦⎥⎥t.[−110]s+[−201]t.

Thus, the eigenspace corresponding to λ2=−3λ2=−3 is two-dimensional and is spanned by F⎣⎢⎢−110⎤⎦⎥⎥[−110] and F⎣⎢⎢−201⎤⎦⎥⎥[−201]. ⎧⎩⎨⎪⎪⎪⎪F⎣⎢⎢−110⎤⎦⎥⎥,F⎣⎢⎢−201⎤⎦⎥⎥⎫⎭⎬⎪⎪⎪⎪ {[−110],[−201]} is a basis for the eigenspace corresponding to λ2=−3λ2=−3. **Notes**

- Eigenvalues and eigenvectors can be complex-valued as well as real-valued.
- The dimension of the eigenspace corresponding to an eigenvalue is less than or equal to the multiplicity of that eigenvalue.
- The techniques used here are practical for 2×22×2 and 3×33×3 matrices. Eigenvalues and eigenvectors of larger matrices are often found using other techniques, such as iterative methods.

Key Concepts

Let AA be an n×nn×n matrix. The eigenvalues of AA are the roots of the characteristic polynomial p(λ)=det(A–λI).p(λ)=det(A–λI).

For each eigenvalue λλ, we find eigenvectors v=F⎣⎢⎢⎢⎢v1v2⋮vn⎤⎦⎥⎥⎥⎥v=[v1v2⋮vn] by solving the linear system

(A–λI)v=0.(A–λI)v=0.

The set of all vectors vv satisfying Av=λvAv=λv is called the **eigenspace** of AA corresponding to λλ.

1. JEE

2. IIT JEE Study Material
3. Normalized and Decomposition of Eigenvectors

PreviousNext

Normalization and Decomposition of Eigenvectors

In linear algebra, Eigenvector is a special part of vectors containing a system of linear equations. Eigenvalues and eigenvectors feature prominently in the analysis of linear transformations such as in the field of stability analysis, atomic orbitals, matrix diagonalization, vibration analysis and many more. In this section will study about eigenvectors definition, vectors normalization and decomposition of eigenvectors.

Formal Definition of Eigen Vector

A nonzero vector that is mapped by a given linear transformation of a vector space onto a vector that is the product of a scalar multiplied by the original vector. Eigenvector of a square matrix is defined as a non-vector by which given matrix is multiplied, and is equal to a scalar multiple of that vector.

Explanation:

Let us suppose that A is an n x n square matrix and if v be a non-zero vector, then product of matrix A and vector v is defined as produced of a scalar quantity λ, and the given vector, such that:

Av = λv where v is eigenvector and λ is the scalar quantity termed as the eigenvalue associated with the given matrix A.

Normalized Eigenvector

In problems related to finding eigenvectors, we often come across with computation of normalized eigenvectors. Normalized eigenvector is nothing but an eigenvector having unit length.
It can be found by simply dividing each component of the vector by the length of the vector. By doing so, the vector is converted into the vector of length one.

The formula for finding length of vector:

X=[x1x2..xn]

L=x12+x22+…+xn2

For example: Given eigenvector is

[1−5−1]

Here,

L=12+(−5)2+12

L = 3√3

Its normalized form is represented by:

[133−533−133]

Eigenvector Decomposition

The decomposition of a square matrix A into eigenvalues and eigenvectors is known as eigen decomposition. The decomposition of any square matrix into eigenvalues and eigenvectors is always possible as long as the matrix consisting of the eigenvectors of given matrix is square matrix, also explained in eigen decomposition theorem.

As we are well known to the fact that a matrix represents a system of linear equations. The matrix can be worked out in order to determine its eigenvalues as well as eigenvectors. The determination of eigenvectors can be done only after the computation of eigenvalues. The whole process of determining eigenvectors is known as eigenvalue decomposition. It is also termed as eigendecomposition.

Solved Examples On Eigenvector Decomposition

Example 1: Show the process of eigenvector decomposition of matrix

A=[1013]

Solution:

The 2 × 2 real matrix

A=[1013] may be decomposed into a diagonal matrix through multiplication of a non-

singular matrix

B=[abcd]∈R2×2.

Then

[abcd]−1[1013][abcd]=[x00y]

for some real diagonal matrix [x00y]

Multiplying both sides of the equation on the left by B, we get

[1013][abcd]=[abcd][x00y].

The above equation can be decomposed into two simultaneous equations.

Factoring out the eigenvalues x and y:

{[1013][ac]=x[ac][1013][bd]=y[bd]

Letting:

a→=[ac],b→=[bd],

this gives us two vector equations:

$\{Aa\to=xa\to Ab\to=yb\to$

And can be represented by a single vector equation involving two solutions as eigenvalues:

$Au=\lambda u$ where λ represents the two eigenvalues x and y, and u represents the vectors

a and b. Shifting λu to the left-hand side and factoring u out

$(A-\lambda I)u=0$

Since B is non-singular, it is essential that u is non-zero. Therefore,

$\det(A-\lambda I)=0$ Thus,

$(1-\lambda)(3-\lambda)=0$

giving us the solutions of the eigenvalues for the matrix A as $\lambda = 1$ or $\lambda = 3$, and the resulting diagonal matrix from the eigendecomposition of A is thus

[1003].

Putting the solutions back into the above simultaneous equations

$\{[1013][ac]=1[ac][1013][bd]=3[b$

d] Solving the equations, we

have $a=-2c and b=0,[c,d]\in R$.

Thus the matrix B required for the eigendecomposition of A is

$B=[-2c0cd],[c,d]\in R$,

that is:

$[-2c0cd]-1[1013][-2c0cd]=[1003],[c,d]\in R$

Related Links:

Eigenvectors of a Matrix

Eigenvalues and Eigenvectors Problems

Frequently Asked Questions

What do you mean by eigenvector decomposition?

Eigenvector decomposition is the method of decomposition of a square matrix A into eigenvalues and eigenvectors.

Give two applications of eigenvectors.

Eigenvectors are used in quantum mechanics. Eigenvector decomposition is used in order to solve linear equations of first order.

What is the eigenvalue of a singular matrix?

Singular matrix has a 0 eigenvalue.