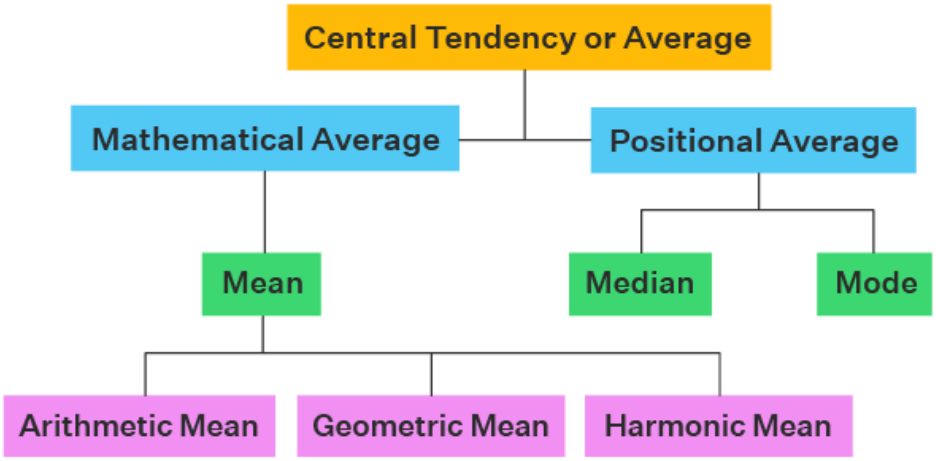
1(A) Explain briefly about mean, median and mode

Mean, median, and mode are the measures of central tendency, used to study the various characteristics of a given set of data. A **measure of central tendency** describes a set of data by identifying the central position in the data set as a single value. We can think of it as a tendency of data to cluster around a middle value. In statistics, the three most common measures of central tendencies are Mean, Median, and Mode. Choosing the best measure of central tendency depends on the type of data we have.



Mean

The **arithmetic mean** of a given data is the sum of all observations divided by the number of observations

For example :

Aa cricketer's scores in five ODI matches are as follows: 12, 34, 45, 50, 24. To find his average score in a match, we calculate the arithmetic mean of data using the mean

Formula:

Mean = Sum of all observations/Number of observations

Mean = (12 + 34 + 45 + 50 + 24)/5

Mean = 165/5 = 33

Mean is denoted by $\bar{x}$ (pronounced as x bar).

Frequency Distribution (Tabular) Form

When the data is present in tabular form, we use the following formula:

Mean, $\bar{x} = (x_1f_1 + x_2f_2 + ... + x_nf_n)/(f_1 + f_2 + ... + f_n)$

Consider the following example.

Example 1: Find the mean of the following distribution:

x	4	6	9	10	15
f	5	10	10	7	8

Solution:

Calculation table for arithmetic mean:

x_i	f_i	$\mathbf{x_if_i}$
4	5	20
6	10	60
9	10	90
10	7	70
15	8	120
	$\mathbf{\sum f_i = 40}$	$\mathbf{\sum x_i f_i = 360}$

Mean, $\bar{x} = (\sum x_i f_i)/(\sum f_i)$

$= 360/40$

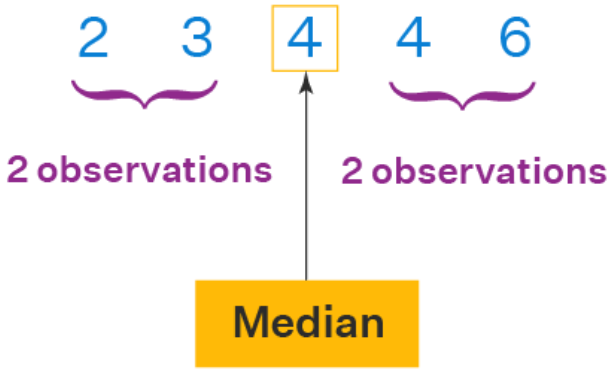
$= 9$

Thus, Mean = 9

Median

The value of the **middlemost observation**, obtained after arranging the data in ascending or descending order, is called the **median** of the data.

For example, consider the data: 4, 4, 6, 3, 2. Let's arrange this data in ascending order: 2, 3, 4, 4, 6. There are 5 observations. Thus, median = middle value i.e. 4.



Case 1: Ungrouped Data

Step 1: Arrange the data in ascending or descending order.
Step 2: Let the total number of observations be n.

To find the median, we need to consider if n is even or odd. If **n is odd**, then use the formula:

Median = $(n + 1)/2^{th}$ observation

Example 1: Let's consider the data: 56, 67, 54, 34, 78, 43, 23. What is the median?

Solution:

Arranging in ascending order, we get: 23, 34, 43, 54, 56, 67, 78. Here, n (number of observations) = 7

So, $(7 + 1)/2 = 4$

∴ Median = 4^{th} observation

Median = 54

If **n is even**, then use the formula:

Median = $[(n/2)^{th}$ obs.+ $((n/2) + 1)^{th}$ obs.$]/2$

Mode

The value which appears most often in the given data i.e. the observation with the highest frequency is called a **mode** of data.

Case 1: Ungrouped Data

For ungrouped data, we just need to identify the observation which occurs maximum times.

Mode = Observation with maximum frequency

For example in the data: 6, 8, 9, 3, 4, 6, 7, 6, 3, the value 6 appears the most number of times. Thus, mode = 6. An easy way to remember mode is: **M**ost **O**ften **D**ata **E**ntered.

Note: A data may have no mode, 1 mode, or more than 1 mode. Depending upon the number of modes the data has, it can be called unimodal, bimodal, trimodal, or multimodal.

1(B). Write a python program to find mean, median and mode of given data.

Ans:

Mean : The mean is the average of all numbers and is sometimes called the arithmetic mean. This code calculates Mean or Average of a list containing numbers:

```
# Python program to print
# mean of elements
# list of elements to calculate mean
n_num = [1, 2, 3, 4, 5]
n = len(n_num)
get_sum = sum(n_num)
mean = get_sum / n
print("Mean / Average is: " + str(mean))
```

Output:

```
Mean / Average is: 3.0
```

Median : The median is the middle number in a group of numbers. This code calculates Median of a list containing numbers:

```
# Python program to print
# median of elements
# list of elements to calculate median
n_num = [1, 2, 3, 4, 5]
n = len(n_num)
```

```
n_num.sort()

if n % 2 == 0:
median1 = n_num[n//2]
median2 = n_num[n//2 - 1]
median = (median1 + median2)/2
else:
median = n_num[n//2]
print("Median is: " + str(median))
```
Output:

```
Median is: 3
```

Mode : The mode is the number that occurs most often within a set of numbers. This code calculates Mode of a list containing numbers:

```
# Python program to print
# mode of elements

from collections import Counter
# list of elements to calculate mode
n_num = [1, 2, 3, 4, 5, 5]
n = len(n_num)
data = Counter(n_num)
get_mode = dict(data)
mode = [k for k, v in get_mode.items() if v == max(list(data.values()))]
if len(mode) == n:
get_mode = "No mode found"
else:
get_mode = "Mode is / are: " + ', '.join(map(str, mode))
print(get_mode)
```
Output:

```
Mode is / are: 5
```

2. Write a python program to find harmonic mean and geometric mean of given data.

Ans:

```
# Python3 code to demonstrate working of
# Geometric Mean of List
# using loop + formula
import math

# initialize list
test_list = [6, 7, 3, 9, 10, 15]

# printing original list
print("The original list is : " + str(test_list))

# Geometric Mean of List
# using loop + formula
temp = 1
for i in range(0, len(test_list)) :
   temp = temp * test_list[i]
temp2 = (float)(math.pow(temp, (1 / len(test_list))))
res = (float)(temp2)

# printing result
print("The geometric mean of list is : " + str(res))
```

Output :

The original list is : [6, 7, 3, 9, 10, 15]

The geometric mean of list is : 7.443617568993922

Using statistics.geometric_mean()

This task can also be performed using inbuilt function of geometric_mean(). This is new in Python versions >= 3.8.

```
# Python3 code to demonstrate working of
# Geometric Mean of List
# using statistics.geometric_mean()
```

```
import statistics

# initialize list
test_list = [6, 7, 3, 9, 10, 15]

# printing original list
print("The original list is : " + str(test_list))

# Geometric Mean of List
# using statistics.geometric_mean()
res = statistics.geometric_mean(test_list, 1)

# printing result
print("The geometric mean of list is : " + str(res))
```

Output :

The original list is : [6, 7, 3, 9, 10, 15]

The geometric mean of list is : 7.443617568993922

Harmonic Mean of List

Method #1 :

```
# Python3 code to demonstrate working of
# Harmonic Mean of List
# using loop + formula

# initialize list
test_list = [6, 7, 3, 9, 10, 15]

# printing original list
print("The original list is : " + str(test_list))

# Harmonic Mean of List
# using loop + formula
sum = 0
```

```
for ele in test_list:
    sum += 1 / ele
res = len(test_list)/sum

# printing result
print("The harmonic mean of list is : " + str(res))
```

Output :

The original list is : [6, 7, 3, 9, 10, 15]

The harmonic mean of list is : 6.517241379310345

Method #2 : Using statistics.harmonic_mean()

This task can also be performed using inbuilt function of harmonic_mean(). This is new in Python versions >= 3.8.

```
# Python3 code to demonstrate working of
# Harmonic Mean of List
# using statistics.harmonic_mean()
import statistics
test_list = [6, 7, 3, 9, 10, 15]
print("The original list is : " + str(test_list))
  res = statistics.harmonic_mean(test_list)
print("The harmomin mean of list is : " + str(res))
```

Output :

The original list is : [6, 7, 3, 9, 10, 15]

The harmonic mean of list is : 6.517241379310345

Learn any one method

3. What is mean deviation? Explain types with an example.

Mean deviation: Mean Deviation is also known as an average deviation; it can be computed using the Mean or Median of the data. Mean deviation is represented as the arithmetic deviation of a different item that follows the central tendency.

Formula:

As mentioned, the Mean Deviation can be calculated using Mean and Median.

- Mean Deviation using Mean: ∑ | X – M | / N
- Mean Deviation using Median: ∑ | X – X1 | / N

Example: the Mean Deviation of 3, 6, 6, 7, 8, 11, 15, 16

Step 1: Find the **mean**:

Mean = (*3 + 6 + 6 + 7 + 8 + 11 + 15 + 16/*)**8** = *72/***8** = 9

Step 2: Find the **distance** of each value from that mean:

Value	Distance from 9(Mean- x)
3	6
6	3
6	3
7	2
8	1
11	2
15	6
16	7

Step 3. Find the **mean of those distances**:

Mean Deviation = (*6 + 3 + 3 + 2 + 1 + 2 + 6 + 7)/***8** = *30/***8** = **3.75**

So, the **mean = 9**, and the **mean deviation = 3.75**

```
import math

data=[4,8,6,5,3,2,8,9,2,5]
n = len(data)
mean = sum(data) / n

deviations=0
for x in data:
    deviations = deviations + pow((x - mean),2)

variance = deviations/ n
print(variance)
std = math.sqrt(variance)
print (std)
```

4. (A) Explain about standard deviation and variance of measures of dispersion.

Standard Deviation: Standard Deviation can be represented as the square root of Variance.

To find the standard deviation of any data, you need to find the variance first. Standard Deviation is considered the best measure of dispersion.

Formula:

Standard Deviation = √σ

Example 1: Find the population standard deviation of the data set {1, 3, 6, 7, 12}.

Solution: Standard deviation is a measure of dispersion given by the formula $\sqrt{\frac{\sum_1^n (X-\overline{X})^2}{n}}$.

$n = 5$

$\overline{X} = (1 + 3 + 6 + 7 + 12) / 5 = 5.8$

$S.D = \sqrt{\frac{(1-5.8)^2+(3-5.8)^2+(6-5.8)^2+(7-5.8)^2+(12-5.8)^2}{5}} = 3.76$

Answer: Standard deviation = 3.76

Variance (σ2): In simple terms, the variance can be calculated by obtaining the sum of the squared distance of each term in the distribution from the Mean, and then dividing this by the total number of the terms in the distribution.

Formula:

$(\sigma 2) = \sum (X - \bar{x})^2 / N$

n = no. of terms

x_i = elements of data

$\bar{x}$ = mean

4.(B). Write a python to find standard deviation and variance of given data.

```
import math

# Finding the variance is essential before calculating the standard deviation

def varinc(val, ddof=0):

    n = len(val)

    m = sum(val) / n

    return sum((x - m) ** 2 for x in val) / (n - ddof)

# finding the standard deviation

def stddev(val):

    vari = varinc(val)

    stdev = math.sqrt(vari)

    return stdev

print(stddev([5, 9, 6, 2, 6, 3, 7, 4, 8, 6]))
```

5.(a) Explain briefly about hypothesis testing and random variable.

Hypothesis Testing

Hypothesis testing is an act in statistics whereby an analyst tests an assumption regarding a population parameter. The methodology employed by the analyst depends on the nature of the data used and the reason for the analysis.

Hypothesis testing is used to assess the plausibility of a hypothesis by using sample data. Such data may come from a larger population, or from a data-generating process. The word "population" will be used for both of these cases in the following descriptions.

- Hypothesis testing is used to assess the plausibility of a hypothesis by using sample data.
- The test provides evidence concerning the plausibility of the hypothesis, given the data.
- Statistical analysts test a hypothesis by measuring and examining a random sample of the population being analyzed.

4 Steps of Hypothesis Testing

All hypotheses are tested using a four-step process:

1. The first step is for the analyst to state the two hypotheses so that only one can be right.
2. The next step is to formulate an analysis plan, which outlines how the data will be evaluated.
3. The third step is to carry out the plan and physically analyze the sample data.
4. The fourth and final step is to analyze the results and either reject the null hypothesis, or state that the null hypothesis is plausible, given the data.

A statistical hypothesis is an assumption about a population which may or may not be true. Hypothesis testing is a set of formal procedures used by statisticians to either accept or reject statistical hypotheses.

Statistical hypotheses are of two types:

- **Null hypothesis,** H_0 - represents a hypothesis of chance basis.
- **Alternative hypothesis,** H_a - represents a hypothesis of observations which are influenced by some non-random cause.

Example

suppose we wanted to check whether a coin was fair and balanced. A null hypothesis might say, that half flips will be of head and half will of tails whereas alternative hypothesis might say that flips of head and tail may be very different.

H_0:P=0.5

Ha:P≠0.5

For example if we flipped the coin 50 times, in which 40 Heads and 10 Tails results. Using result, we need to reject the null hypothesis and would conclude, based on the evidence, that the coin was probably not fair and balanced.

Random Variable Definition

A random variable is a variable whose value is unknown or a function that assigns values to each of an experiment's outcomes

A random variable is a rule that assigns a numerical value to each outcome in a sample space. Random variables may be either discrete or continuous. A random variable is said to be discrete if it assumes only specified values in an interval. Otherwise, it is continuous. We generally denote the random variables with capital letters such as X and Y. When X takes values 1, 2, 3, …, it is said to have a discrete random variable.

As a function, a random variable is needed to be measured, which allows probabilities to be assigned to a set of potential values. It is obvious that the results depend on some physical variables which are not predictable. Say, when we toss a fair coin, the final result of happening to be heads or tails will depend on the possible physical conditions. We cannot predict which outcome will be noted. Though there are other probabilities like the coin could break or be lost, such consideration is avoided.

Types of Random Variable

As discussed in the introduction, there are two random variables, such as:

- Discrete Random Variable
- Continuous Random Variable

5.(B).Explain about basics of probability.

Ans :

Probability can be defined as the ratio of the number of favorable outcomes to the total number of outcomes of an event. For an experiment having 'n' number of outcomes, the number of favorable outcomes can be denoted by x. The formula to calculate the probability of an event is as follows.

Probability(Event) = Favorable Outcomes/Total Outcomes = x/n

The probability is classified into theoretical probability and experimental probability.

Terminology of Probability Theory

The following terms in probability help in a better understanding of the concepts of probability.

Experiment: A trial or an operation conducted to produce an outcome is called an experiment.

Sample Space: All the possible outcomes of an experiment together constitute a sample space. For example, the sample space of tossing a coin is head and tail.

Favorable Outcome: An event that has produced the desired result or expected event is called a favorable outcome. For example, when we roll two dice, the possible/favorable outcomes of getting the sum of numbers on the two dice as 4 are (1,3), (2,2), and (3,1).

Trial: A trial denotes doing a random experiment.

Random Experiment: An experiment that has a well-defined set of outcomes is called a random experiment. For example, when we toss a coin, we know that we would get ahead or tail, but we are not sure which one will appear.

Event: The total number of outcomes of a random experiment is called an event.

Probability Formula

The probability formula defines the likelihood of the happening of an event. It is the ratio of favorable outcomes to the total favorable outcomes. The probability formula can be expressed as,

$$P(A) = \frac{\text{Number of favorable outcomes to A}}{\text{Total number of possible outcomes}}$$

where,

- P(B) is the probability of an event 'B'.
- n(B) is the number of favorable outcomes of an event 'B'.
- n(S) is the total number of events occurring in a sample space.

6. Explain about the probability distributions: Bernoulli, Binomial, Poisson.

A probability distribution is a statistical function that describes all the possible values and probabilities for a random variable within a given range. This range will be bound by the minimum and maximum possible values, but where the possible value would be plotted on the probability distribution will be determined by a number of factors. The mean (average), standard deviation, skewness, and kurtosis of the distribution are among these factors.

Types of Probability Distribution

The probability distribution is divided into two parts:

1. Discrete Probability Distributions
2. Continuous Probability Distributions

Bernoulli Distribution

A **Bernoulli distribution** has only two possible outcomes, namely 1 (success) and 0 (failure), and a single trial. So the random variable X which has a Bernoulli distribution can take value 1 with the probability of success, say p, and the value 0 with the probability of failure, say q or 1-p.

Here, the occurrence of a head denotes success, and the occurrence of a tail denotes failure. Probability of getting a head = 0.5 = Probability of getting a tail since there are only two possible outcomes.

The probability mass function is given by**: $p^x(1-p)^{1-x}$**

Binomial Distribution

distribution where only two outcomes are possible, such as success or failure, gain or loss, win or lose and where the probability of success and failure is same for all the trials is called a Binomial Distribution.

The outcomes need not be equally likely. Remember the example of a fight between me and Undertaker? So, if the probability of success in an experiment is 0.2 then the probability of failure can be easily computed as q = 1 – 0.2 = 0.8.

Each trial is independent since the outcome of the previous toss doesn't determine or affect the outcome of the current toss. An experiment with only two possible outcomes repeated n number of times is called binomial. The parameters of a binomial distribution are n and p where n is the total number of trials and p is the probability of success in each trial.

On the basis of the above explanation, the properties of a Binomial Distribution are

1. Each trial is independent.

2. There are only two possible outcomes in a trial- either a success or a failure.
3. A total number of n identical trials are conducted.
4. The probability of success and failure is same for all trials. (Trials are identical.)

The mathematical representation of binomial distribution is given by:

$$P(x) = \frac{n!}{(n-x)!\,x!} p^x q^{n-x}$$

Poisson Distribution

A distribution is called **Poisson distribution** when the following assumptions are valid:

1. Any successful event should not influence the outcome of another successful event.
2. The probability of success over a short interval must equal the probability of success over a longer interval.
3. The probability of success in an interval approaches zero as the interval becomes smaller.

Now, if any distribution validates the above assumptions then it is a Poisson distribution. Some notations used in Poisson distribution are:

- λ is the rate at which an event occurs,
- t is the length of a time interval,
- And X is the number of events in that time interval.

Here, X is called a Poisson Random Variable and the probability distribution of X is called Poisson distribution.

Let μ denote the mean number of events in an interval of length t. Then, μ = λ*t.

The PMF of X following a Poisson distribution is given by:

$$P(X = x) = e^{-\mu} \frac{\mu^x}{x!} \qquad for\ x = 0, 1, 2, \ldots..$$

7.(A). What is gaussian distribution. Implement gaussian probability distribution in python.

Normal Distribution

This is the most commonly discussed distribution and most often found in the real world. Many continuous distributions often reach normal distribution given a large enough sample. This has two parameters namely mean and standard deviation.

This distribution has many interesting properties. The mean has the highest probability and all other values are distributed equally on either side of the mean in a symmetric fashion. The standard normal distribution is a special case where the mean is 0 and the standard deviation of 1.

$$f(\mathrm{x}) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

where μ is the mean of the random variable X and σ is the standard deviation.

Implementation of Normal Distribution

```
# Importing required libraries
import numpy as np
import matplotlib.pyplot as plt
 # Creating a series of data of in range of 1-50.
x = np.linspace(1,50,200)
 #Creating a Function.
def normal_dist(x , mean , sd):
    prob_density = (np.pi*sd) * np.exp(-0.5*((x-mean)/sd)**2)
    return prob_density
 #Calculate mean and Standard deviation.
mean = np.mean(x)
sd = np.std(x)
 #Apply function to the data.
pdf = normal_dist(x,mean,sd)
#Plotting the Results
plt.plot(x,pdf , color = 'red')
plt.xlabel('Data points')
plt.ylabel('Probability Density')
```

7. (B). What is exponential distribution? Implement exponential distribution in python.

Exponential Distribution

Recall the discrete probability distribution we have discussed in the Discrete Probability post. In the Poisson distribution, we took the example of calls received by the customer care center. In that example, we considered the average number of calls per hour. Now, in this distribution, the time between successive calls is explained.

The exponential distribution can be seen as an inverse of the Poisson distribution. The events in consideration are independent of each other.

The PDF is given by,

$$f(\mathrm{x}) = \lambda e^{-\lambda x}$$

where λ is the rate parameter. λ = 1/(average time between events).

```
# import packages
from scipy.stats import poisson
import numpy as np
import matplotlib.pyplot as plt

# random variable
x = np.arange(0, 15)

# poisson distribution
y = poisson.pmf(x, mu=5)

# plotting
plt.figure(figsize=(9,5))
plt.plot(x, y, marker='o', color='black')
plt.xlabel('Random Variable X', fontsize=14)
plt.ylabel('Probability', fontsize=14)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.show()
```

8. (A) what is data science, how does data science relate to other fields?

Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. Data science uses complex machine learning algorithms to build predictive models.

The data used for analysis can come from many different sources and presented in various formats.

The field of data science encompasses multiple subdisciplines such as data analytics, data mining, artificial intelligence, machine learning, and others.

Data Analytics
While data analysts are focused on extracting meaningful insights from various data sources, data scientists go beyond that to "forecast the future based on past patterns," according to SimpliLearn. "A data scientist creates questions, while a data analyst finds answers to the existing set of questions."

Artificial Intelligence
Commonly called AI, artificial intelligence, according to Techopedia, "aims to imbue software with the ability to analyze its environment using either predetermined rules and search algorithms or pattern recognizing machine learning models, and then make decisions based on those analyses. In this way, AI attempts to mimic biological intelligence to allow the software application or system to act with varying degrees of autonomy, thereby reducing manual human intervention for a wide range of functions."

Machine Learning
Machine learning algorithms use statistics to find patterns in massive amounts of data, according to MIT Technology Review. A subdiscipline of AI, "machine learning is the process that powers many of the services we use today — recommendation systems like those on Netflix, YouTube, and Spotify; search engines like Google and Baidu; social-media feeds like Facebook and Twitter; voice assistants like Siri and Alexa. The list goes on."

8. (B) What is Chi – square distribution. Implement Chi – square probability distribution in python.

Chi-square Distribution

This distribution is equal to the sum of squares of p normal random variables. p is the number of degrees of freedom. Like the t-distribution, as the degrees of freedom increase, the

distribution gradually approaches the normal distribution. Below is a chi-square distribution with three degrees of freedom.

$$f(x) = \frac{\left(x^{\frac{\mathrm{p}}{2}-1} e^{-\frac{x}{2}}\right)}{2^{\mathrm{p}/2}\, \Gamma\left(\frac{\mathrm{p}}{2}\right)}$$

where p is the degrees of freedom and Γ is the gamma function.

The chi-square value is calculated as follows:

$$\chi^2 = \sum \frac{(o_i - E_i)^2}{E_i}$$

where o is the observed value and E represents the expected value. This is used in hypothesis testing to draw inferences about the population variance of normal distributions.

Plot Multiple Chi-Square Distributions

The following code shows how to plot multiple Chi-square distribution curves with different degrees of freedom:

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import chi2

#x-axis ranges from 0 to 20 with .001 steps

x = np.arange(0, 20, 0.001)

#define multiple Chi-square distributions

plt.plot(x, chi2.pdf(x, df=4), label='df: 4')

plt.plot(x, chi2.pdf(x, df=8), label='df: 8')

plt.plot(x, chi2.pdf(x, df=12), label='df: 12')

#add legend to plot

plt.legend()
```

9. Define Eigen values &Eigen vectors ? Explain briefly about Sparse matrices.

Eigenvalue Definition

Eigenvalues are the special set of scalars associated with the system of linear equations. It is mostly used in matrix equations. 'Eigen' is a German word that means 'proper' or 'characteristic'. Therefore, the term eigenvalue can be termed as characteristic value, characteristic root, proper values or latent roots as well. In simple words, the eigenvalue is a scalar that is used to transform the eigenvector. The basic equation is

$$\mathbf{Ax = \lambda x}$$

The number or scalar value "**λ"** is an eigenvalue of A.

In Mathematics, an eigenvector corresponds to the real non zero eigenvalues which point in the direction stretched by the transformation whereas eigenvalue is considered as a factor by which it is stretched. In case, if the eigenvalue is negative, the direction of the transformation is negative.

For every real matrix, there is an eigenvalue. Sometimes it might be complex. The existence of the eigenvalue for the complex matrices is equal to the fundamental theorem of algebra.

What are EigenVectors?

Eigenvectors are the vectors (non-zero) that do not change the direction when any linear transformation is applied. It changes by only a scalar factor. In a brief, we can say, if A is a linear transformation from a vector space V and **x** is a vector in V, which is not a zero vector, then v is an eigenvector of A if A(X) is a scalar multiple of **x**.

An **Eigenspace** of vector **x** consists of a set of all eigenvectors with the equivalent eigenvalue collectively with the zero vector. Though, the zero vector is not an eigenvector.

Let us say A is an "n × n" matrix and λ is an eigenvalue of matrix A, then **x**, a non-zero vector, is called as eigenvector if it satisfies the given below expression;

A**x** = λ**x**

x is an eigenvector of A corresponding to eigenvalue, λ.

Note:

- There could be infinitely many Eigenvectors, corresponding to one eigenvalue.
- For distinct eigenvalues, the eigenvectors are linearly dependent.

Definition of Sparse Matrix

Sparse Matrix is a matrix that contains a few non-zero elements. Almost all the places are filled

with zero. Matrix of m*n dimension refers to a 2-D array with m number of rows and n number

of columns. And if the non-zero elements in the matrix are more than zero elements in the matrix then it is called a sparse matrix. And in case the size of the matrix is big, a lot of space is wasted to represent such a small number of non-zero elements. Similarly for scanning the same non-zero will take more time.

Syntax:

Thus to limit the processing time and space usage instead of storing a lesser number of non-zero elements in a matrix, we use the below 2 representation:

1. Array representation

The 2D array is converted to 1 D array with 3 columns representing:

Popular Course in this category

a. Row – Row index of non-zero element

b. Column – Column index of non-zero element

c. Value – value at the same row, column index in 2D matrix

2. LinkedList representation:

In linked list representation, each node has four fields as given below:

- Row: Row index of the non-zero elements in the matrix.
- Column: Column index of the non-zero elements in the matrix.
- Value: Value of the non zero elements at (row, column) position in the matrix
- Next node: Reference to the next node.

10. Explain in details about Sparse matrices. With example

A sparse matrix is a matrix in which most of the elements have zero value and thus efficient ways of storing such matrices are required. Sparse matrices are generally utilized in applied machine learning such as in data containing data-encodings that map categories to count and also in entire subfields of machine learning such as natural language processing (NLP).
Examples:

0 0 0 1 0

2 0 0 0 3

0 0 0 4 0

Above is sparse matrix with only 4 non-zero elements.

Representing a sparse matrix by a 2D array leads to wastage of lots of memory as zeroes in the matrix are of no use in most cases. So, instead of storing zeroes with non-zero elements, we only store non-zero elements. These efficient ways require only the non-zero values to be stored along with their index so that the original matrix can be retrieved when required. One such efficient way in Python is the use of a dictionary. Dictionary in Python stores data in key-value pairs like maps in Java. Dictionary stores data in an unordered manner.

Approach:

- First, we take a sparse matrix and create an empty dictionary.
- Then we iterate through all the elements of the matrix and check if they are zero or non-zero elements.
- The non-zero elements are added to the dictionary with their index as the key and their data as the value in the key-value pairs of the dictionary.
- Finally, we can print the dictionary giving each element with its index.

Implementation:

```
# creating sparse matrix

arr = [[0, 0, 0, 1, 0],

       [2, 0, 0, 0, 3],

       [0, 0, 0, 4, 0]]
```

```
# creating empty dictionary
dic = {}


# iterating through the matrix
for i in range(len(arr)):
        for j in range(len(arr[i])):
                if arr[i][j] != 0:


                        # adding non zero elements to
                        # the dictionary
                        dic[i, j] = arr[i][j]


print("Position of non-zero elements in the matrix:")
print(dic)
```

Output:

The following dictionary shows the positional of non-zero elements in the sparse matrix

{(0, 3): 1, (1, 0): 2, (1, 4): 3, (2, 3): 4}