**Question1 What is the difference between static and dynamic variables in Python?**

**Static Variables:**

Static variables, also known as class variables, have a fixed memory location throughout the execution of a program. They are declared within a class or a function and retain their values between function calls. Here are the defining characteristics of static variables:

Memory Allocation: Static variables are allocated memory when the program starts and retain their memory until the program terminates. This allocation occurs outside of any function or method.

Scope: Static variables have a scope that is limited to the block or function in which they are defined. They are not accessible outside their defining scope.

Lifetime: The lifetime of a static variable extends from the program’s start until its termination. Any value assigned to a static variable persists across multiple function calls.

Initialization: Static variables are initialized only once, at the beginning of the program execution. If no explicit value is assigned, they are automatically initialized to zero or a null value.

Data Sharing: Since static variables have a global nature within a class or function, they can be shared among different instances of the class or multiple function calls.

**Dynamic Variables:**

Dynamic variables, also referred to as instance variables, are allocated memory during runtime. Unlike static variables, dynamic variables have a memory location that changes as the program executes. Let’s explore the key features of dynamic variables:

Memory Allocation: Dynamic variables are allocated memory at runtime when they are declared or instantiated. The memory is released when the variable goes out of scope or is explicitly deallocated.

Scope: Dynamic variables can have a broader scope compared to static variables. They can be accessed and modified from various parts of the program, depending on their accessibility modifiers.

Lifetime: The lifetime of a dynamic variable depends on when it is created and when it is destroyed. It exists as long as its containing object or block is in scope.

Initialization: Dynamic variables can be initialized multiple times during runtime. They can be assigned values based on user input, calculations, or other dynamic factors.

Data Isolation: Dynamic variables are typically associated with specific instances of a class or objects. Each instance maintains its own copy of the dynamic variable, which can have different values.

**Question 2 Explain the purpose of "pop ()","popitem ()","clear ()" in a dictionary with suitable examples?**

Answer: pop (): It is used to delete an element from the dictionary. In this method we

Pass key as parameter and it returns the value along with removing the element from dictionary. For example: -

Let name is diction such that

Name = {1: “Amit”, 2: “Rohit”, 3: “Rohan”}

Name. pop(2) will remove element **2:Rohit** and return **Rohit.**

**Popitem**(): - This removes the last inserted key value pair from the dictionary. No parameter is required in this method. For example: -

Name.popitem() will remove **3: “Rohan” and returns the same.**

**Clear(): -** Roves all the items from dictionary. It doesn’t return anything.

**Question 3 What do you mean by FrozenSet? Explain it with suitable examples?**

**Answer: -** Frozenset is similar to set in python, except that frozen sets are immutable, which implies that once generated, elements from the frozenset cannot be added or removed. This function accepts any iterable object as input and transforms it into an immutable object. It is not assured that the order of the elements will be retained.

vowels = (‘a’, ‘e’, ‘i’, ‘o’, ‘u’)

fSet = frozenset(vowels)

fSet.add(‘v’) we can’t add, remove or update any element in frozen set and it is immutable and give error.

**Question 4 Differentiate between mutable and immutable data types in Python and give examples of mutable and immutable data types?**

Answer: Mutable objects are those that allow you to change their value or data in place without affecting the object’s identity. In contrast, immutable objects don’t allow this kind of operation. You’ll just have the option of creating new objects of the same type with different values.

| **Mutable Objects** | **Immutable Objects** |
| --- | --- |
| Lists | Numbers |
| Dictionaries | Strings |
| Sets | Tuples |

**Question5 What is \_\_init\_\_? Explain with an example?**

**Answer: -** \_**\_init\_\_ method** in Python is used to initialize objects of a class. It is also called a constructor.

**class** Trees:

    # init method or constructor

**def** \_\_init\_\_ (self, name):

        self.name **=** name

    # Sample Method

**def** names(self):

        print ('Hello, this is a ', self.name)

p **=** Person('Mango')

p.names()

Here we are initializing the name variable with the help of init while creating the object of the class tree.

**Question6: -What is docstring in Python? Explain with an example?**

**Answer: -** Python documentation string or commonly known as docstring, is a string literal, and it is used in the class, module, function, or method definition. Docstrings are accessible from the doc attribute (\_\_doc\_\_) for any of the Python objects and also with the built-in help () function. An object's docstring is defined by including a string constant as the first statement in the object's definition.

For Example: -

def square(a):

'''Returned argument a is squared.'''

return a\*\*a

print (square. \_\_doc\_\_)

output: - Returned argument a is squared.

**Question 7 What are unit tests in Python?**

**Answer: -** Unit tests are segments of code written to test other pieces of code, typically a single function or method, that we refer to as a unit. They are a very important part of the software development process, as they help to ensure that code works as intended and catch bugs early on.

**Question 8 What is break, continue and pass in Python?**

**Answer** The **break** statement in python is used to terminate the loop or statement in which it is present. After that, the control will pass to the statements that are present after the break statement, if available. If the break statement is present in the nested loop, then it terminates only those loops which contain the break statement.

**Continue** is also a loop control statement just like the break statement. continue statement is opposite to that of the break statement, instead of terminating the loop, it forces to execute the next iteration of the loop.

The **pass** statement in Python is used when a statement is required syntactically but you do not want any command or code to execute. It is like a null operation, as nothing will happen if it is executed.

**Question 9 What is the use of self in Python?**

**Answer: -** The self is used to represent the instance of the class. With this keyword, you can access the attributes and methods of the class in python. It binds the attributes with the given arguments. self is a parameter in function and the user can use a different parameter name in place of it.

**Question 10 What are global, protected and private attributes in Python**

**Answer: -** In Python, global, protected, and private attributes are used to control the visibility and accessibility of class attributes.

Global attributes:

Global Attributes are defined outside of a class and can be accessed from anywhere in the code, including inside classes.

Protected attributes:

Protected attributes are denoted with a single underscore (\_) before the attribute name and are intended to be accessed only within the class and its subclasses.

Private attributes:

Private attributes are denoted with double underscores (\_\_) before the attribute name and are intended to be accessed only within the class in which they are defined.

It is important to note that Python's private attributes are not truly private, as they can still be accessed from outside the class by name mangling. However, this is discouraged as it goes against the principle of encapsulation, which is a fundamental concept in object-oriented programming.

**Question 11 What are modules and packages in Python?**

**Answer: -**In Python, both modules and packages organize and structure the code but serve different purposes.

In simple terms, a module is a single file containing Python code, whereas a package is a collection of modules that are organized in a directory hierarchy.

The module is a simple Python file that contains collections of functions and global variables and with having a .py extension file. It is an executable file and to organize all the modules we have the concept called Package in Python.

**Question 12 What are lists and tuples? What is the key difference between the two?**

**Answer** **Lists** are mutable, meaning they can be changed after creation. You can add, remove, or modify elements in a list. Lists are defined using square brackets [ ] and elements are separated by commas.

**Tuples**, on the other hand, are immutable, meaning once they are created, their elements cannot be changed. Tuples are defined using parentheses ( ) and elements are separated by commas. For example:

The key difference between lists and tuples lies in their mutability: lists are mutable (can be changed), while tuples are immutable (cannot be changed).

Lists are typically used for collections of homogeneous items where flexibility (such as adding or removing elements) is needed. Tuples, being immutable, are used for fixed collections of heterogeneous data, such as coordinates or records, where ensuring data integrity is important.

**Question13 What is an Interpreted language & dynamically typed language? Write 5 differences between them?**

**Answer** An interpreted language executes source code directly without prior compilation, translating it into machine code or an intermediate form on-the-fly. Examples include Python and JavaScript. Conversely, a dynamically typed language determines variable types at runtime rather than compile time. Both Python and JavaScript serve as examples here as well.

Interpreted languages interpret code line-by-line from source, while dynamically typed languages handle variable types and type-checking during runtime. Performance-wise, interpreted languages tend to be slower due to runtime translation, whereas dynamically typed languages may incur runtime type errors but offer flexibility and expressiveness. In development, interpreted languages speed up prototyping and debugging with immediate feedback, while dynamically typed languages facilitate faster coding with implicit type declarations. Debugging for interpreted languages involves real-time code examination, while dynamically typed languages trace runtime errors like type mismatches.

**Question 14 What are Dict and List comprehensions**

Answer In both comprehensions, the expression defines how each element or key-value pair is computed, item iterates over the iterable, and condition (optional) filters elements based on a Boolean expression. Comprehensions are concise and efficient, reducing the need for explicit loops and making code more readable by focusing on the data transformation logic.

List comprehensions allow you to create lists based on existing iterables like lists, tuples, or strings. They follow the syntax [expression for item in iterable if condition]

Dictionary comprehensions extend this concept to create dictionaries using a similar syntax {key\_expression: value\_expression for item in iterable if condition}.

**Question 15 What are decorators in Python? Explain it with an example. Write down its use cases?**

**Answer** Decorators in Python are functions that modify the behavior of other functions or methods without changing their code explicitly. They are widely used to add functionality to existing functions dynamically.

# Decorator function

def my\_decorator(func):

def wrapper():

print("Something is happening before the function is called.")

func() # Call the original function

print("Something is happening after the function is called.")

return wrapper

# Function to be decorated

@my\_decorator

def say\_hello():

print("Hello!")

# Calling the decorated function

say\_hello()

Use cases of decorators:

1. Logging: Decorators can log function calls, parameters, and outputs for debugging or monitoring purposes.
2. Authentication/Authorization: Decorators can check if a user is authenticated before allowing access to certain functions.
3. Caching: Decorators can cache function results to improve performance by storing computed results for future use.
4. Timing: Decorators can measure and log the time taken by functions to execute.
5. Validation: Decorators can validate function inputs or outputs to ensure data integrity.

**Question 16 How is memory managed in Python?**

**Answer** Memory management in Python is handled by the Python memory manager, which is responsible for allocating and deallocating memory for objects in the Python program. Here are key aspects of how memory is managed in Python:

1. Automatic Memory Allocation: Python uses a private heap containing all Python objects and data structures. The management of this private heap is ensured internally by the Python memory manager. Developers do not need to explicitly allocate memory for new objects.
2. Reference Counting: Python uses a technique called reference counting to keep track of how many references point to an object. When an object's reference count drops to zero, meaning no variable or data structure refers to it anymore, the memory occupied by the object is automatically released.
3. Garbage Collection: In addition to reference counting, Python also employs a garbage collector to reclaim memory used by objects with cyclical references or objects that are no longer referenced correctly due to weak references. The garbage collector runs periodically to clean up such unused memory and free up resources.
4. Memory Pooling: Python's memory manager includes a mechanism for efficient memory allocation called memory pooling. This means that small objects of the same type are allocated in blocks to improve performance and reduce memory fragmentation.

**Question 17 What is lambda in Python? Why is it used?**

**Answer** In Python, a lambda function is a small anonymous function defined with the lambda keyword. Unlike regular functions defined using def, lambda functions can only contain a single expression. They are primarily used for creating simple, one-line functions without the need to define a full-fledged function using def.

Lambda functions are commonly used in situations where a function is needed temporarily for a short period and does not need a name beyond its immediate use.

Lambda functions are also useful in functional programming paradigms where functions are treated as first-class citizens, allowing for more concise and readable code when passing functions as arguments or returning them from other functions.

**Question 18 Explain split () and join () functions in Python?**

**Answer** split() is used to break a string into a list of substrings based on a delimiter, while join() is used to concatenate elements of an iterable into a single string using a specified separator. These functions are fundamental for manipulating strings in various ways, such as parsing text or formatting output**.**

**Question 19 What are iterators , iterable & generators in Python?**

**Answer** In Python, iterables are objects like lists or strings that can be iterated over using a loop or functions like `for` or `in`. Iterators are objects that facilitate this iteration by implementing methods like `\_\_iter\_\_()` and `\_\_next\_\_()`, allowing them to sequentially return elements from the iterable. They maintain state and remember where they are in the sequence.

Generators are a type of iterator defined using functions with `yield` statements instead of `return`. They generate values on-the-fly, one at a time, conserving memory by not storing the entire sequence in memory at once. Generators are useful for efficiently handling large datasets or creating sequences that would otherwise consume significant memory. They offer a concise and readable way to create iterators without explicitly implementing a class with iterator protocol methods.

**Question 20 What is the difference between xrange and range in Python?**

Answer In Python 2.x, `range()` returns a list of numbers, while `xrange()` returns an xrange object that generates numbers lazily, conserving memory. `range()` is suitable for small ranges but may be memory-intensive for large ones. In Python 3.x, `range()` behaves like Python 2.x's `xrange()`, returning an iterable sequence rather than a list, which is more memory efficient for large ranges.

**Question 21 Pillars of Oops?**

**Answer O**bject-oriented programming (OOP) in Python and other languages is built upon four key principles, often referred to as the "pillars" or "four pillars" of OOP:

1. Encapsulation: Encapsulation refers to the bundling of data (attributes) and methods (functions that operate on the data) into a single unit known as a class. It helps in hiding the internal state of an object from the outside world and only exposing the necessary functionalities. This improves security, reusability, and modularity of code.
2. Abstraction: Abstraction involves hiding complex implementation details and showing only the essential features of an object. It focuses on what an object does rather than how it does it. Abstraction allows programmers to manage complexity by providing simplified interfaces and reducing dependencies between parts of a system.
3. Inheritance: Inheritance is a mechanism where a new class (subclass or derived class) can inherit attributes and methods from an existing class (base class or superclass). It promotes code reusability and allows for hierarchical relationships between classes. Subclasses can override or extend the behaviour of their superclass.
4. Polymorphism: Polymorphism means the ability to take different forms. In OOP, polymorphism allows objects of different classes to be treated as objects of a common superclass. It enables flexibility and extensibility in the code by allowing functions to work with objects of multiple types and classes through method overriding and method overloading.

**Question 22 How will you check if a class is a child of another class?**

**Answer** We can check if a class is a subclass (child class) of another class using the issubclass() function or by using the \_\_subclasscheck\_\_() special method. Here’s how you can do it:

1. Using issubclass() function:

The issubclass() function checks if a class is a subclass of another class.

Syntax: issubclass(subclass, superclass)

Example class A:

pass

class B(A):

pass

print(issubclass(B, A)) # Output: True

**Question 23 Explain all types of inheritance with an example?**

**Answer** In object-oriented programming (OOP), inheritance is a mechanism where a new class (subclass or derived class) can inherit attributes and methods from an existing class (base class or superclass). There are several types of inheritance:

1. Single Inheritance:

A subclass inherits from only one superclass. This is the simplest form of inheritance.

1. Multiple Inheritance:

A subclass inherits from multiple super classes. It allows the subclass to combine features from different parent classes.

1. Multilevel Inheritance:

One class serves as a superclass for another class, which in turn serves as a superclass for another class. This forms a chain of inheritance.

1. Hierarchical Inheritance:

Multiple subclasses inherit from a single superclass. This allows different subclasses to share common characteristics defined in the superclass.

1. Hybrid Inheritance:

This is a combination of two or more types of inheritance. It can involve any combination of single, multiple, multilevel, or hierarchical inheritance.

**Question24 What is encapsulation? Explain it with an example?**

Answer Encapsulation in object-oriented programming (OOP) is the concept of bundling data (attributes) and methods (functions) that operate on the data into a single unit called a class. It enables data hiding by restricting access to certain components of the object, typically through the use of access specifiers (e.g., public, private, protected).

A real-life analogy of encapsulation can be seen in a bank's ATM machine. The ATM machine provides a simplified interface (the screen and keypad) for users to interact with the complex internal operations (handling transactions, accessing bank accounts, dispensing cash) that are hidden from the users.

**Question 25 What is polymorphism? Explain it with an example.**

**Answer** Polymorphism in object-oriented programming refers to the ability of different classes to be treated as instances of a common superclass. It allows objects to be processed uniformly, despite their individual differences in behavior. This is achieved through method overriding, where subclasses provide their own implementation of methods defined in their superclass.

For example, in a zoo simulation:

* Animals such as lions, elephants, and monkeys can all have a make\_sound() method.
* Each animal subclass (lion, elephant, monkey) implements make\_sound() differently to produce their respective sounds.
* Despite the differences in how make\_sound() is implemented, the zookeeper can call make\_sound() on each animal object without needing to know the specific subclass, allowing for a unified approach to interacting with all animals.

**Question 1. 2. Which of the following identifier names are invalid and why?**

**a) Serial\_no.**

**b) 1st\_Room**

**c) Hundred$**

**d) Total\_Marks**

**e) total-Marks**

**f) Total Marks**

**g) True**

**h) \_Percentag**

**Answer** b) 1st\_Room Invalid. Identifiers cannot begin with a number.

e) total-Marks Invalid. Hyphens (-) are not allowed in identifiers.

f) Total Marks Invalid. Spaces are not allowed in identifiers.

**20. What do you mean by Measure of Central Tendency and Measures of Dispersion. How it can be calculated.**

**Answer:** Measures of Central Tendency: Descriptive statistics include calculating the mean, median, and mode, which offer insights into the center of the data distribution**.**

The central tendency of data is the center of the distribution of data. It describes the location of data and concentrates on where the data is located. The three most widely used measures of the “center” of the data are Mean, Median, and Mode.

Mean

The “Mean” is the average of the data. The average can be identified by summing up all the numbers and then dividing them by the number of observations.

Mean = X1+ X2+ X3 +… +Xn / n

Median

It is the 50th percentile of the data. In other words, it is exactly the center point of the data. The median can be identified by ordering the data, splitting it into two equal parts, and then finding the number in the middle. It is the best way to find the center of the data.

Mode

The mode of the data is the most frequently occurring data or elements in a dataset. If an element occurs the highest number of times, it is the mode of that data. If no number in the data is repeated, then that data has no mode. There can be more than one mode in a dataset if two values have the same frequency, which is also the highest frequency.

Outliers don’t influence the data in this case. The mode can be calculated for both quantitative and qualitative data.

**Measures of Dispersion:** Variance, standard deviation, and range help us understand the spread or variability of the data.

The dispersion is the “spread of the data”. It measures how far the data is spread. In most of the dataset, the data values are closely located near the mean. The values are widely spread out of the mean on some other datasets. These dispersions of data can be measured by the Inter Quartile Range (IQR), range, standard deviation, and variance of the data.

**Inter Quartile Range (IQR):** -

Quatriles are the special percentile Q1, Q2, Q3 corresponding to 25th, 50th and 75th percentile respectively.

IQR = Q3-Q

It is the spread of middle half (50%) of data. For Quartiles divide the data into four equal parts. For percentile divide it into 100 equal parts.

**Standard Deviation: -**

The most common measure of spread is the standard deviation. The Standard deviation measures how far the data deviates from the mean value. The standard deviation is always positive or zero. It will be large when data points are spread out from mean.

Sample Standard Deviation, s = Square root of [Σ (x −x’ )2/ n-1] where x’ is average and n is no. of samples.

**Variance: -**

The variance is a measure of variability. It is the average squared deviation from the mean. The symbol σ2 represents the population variance, and the symbol for s2represents sample variance.

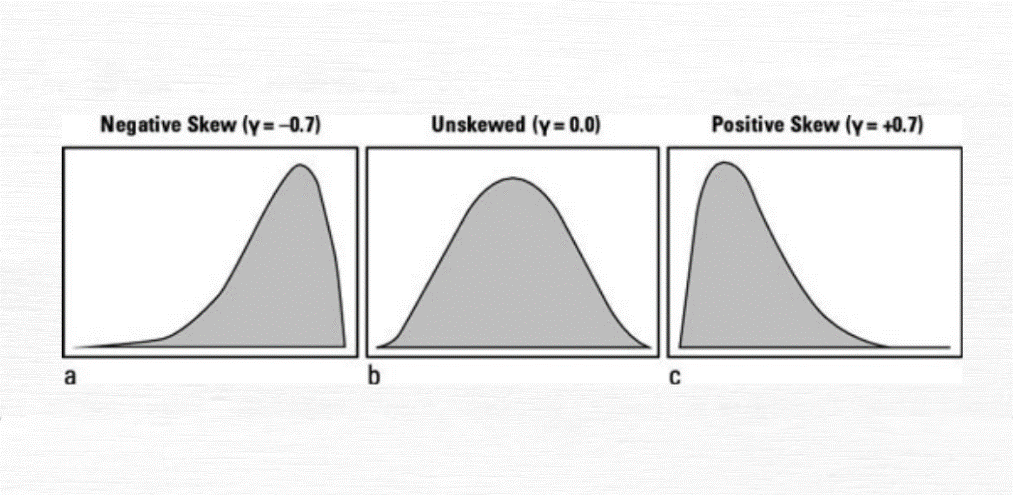
Population variance   σ2= [ Σ (x − μ)2 / N]  
Sample Variance s2 = [ Σ (x − x ¯ )2/ n-1]

**21. What do you mean by skewness. Explain its types. Use graph to show.**

**Skewness**

Skewness is the measure of the asymmetry of the distribution of data. The data is not symmetrical (i.e.) it is skewed towards one side. Skewness is classified into two types: positive skew and negative skew.

* **Positively skewed**: In a Positively skewed distribution, the data values are clustered around the left side of the distribution, and the right side is longer. The mean and median will be greater than the mode in the positive skew.
* **Negatively skewed**: In a Negatively skewed distribution, the data values are clustered around the right side of the distribution, and the left side is longer. The mean and median will be less than the mode.



**22. Explain PROBABILITY MASS FUNCTION (PMF) and PROBABILITY DENSITY FUNCTION (PDF). and what is the difference between them?**  
PMF also known as the discrete density function, is a density function for discrete data. This density function is used to calculate the probability of a certain value of a discrete random variable.

P(X=x) = f(x)

X = random variable

x = one possible value of variable X

The probability value is always a non-negative number. The sum of the probability of all possible values of X will equal 1.

A fair coin is tossed twice. Let us consider we want to identify the probability that the result has one head. When two coins are tossed, the resultant values can be {HH, HT, TH, TT}.

Let us consider X as the random variable with the count of heads resulted from a coin toss.

X can have {0,1,2}.

We need P(X=1) = P({HT,TH})= number of outcomes with {HT,TH}/total number of outcomes = 2.4 = 0.5

Also, the sum of probability of all values of X will equal 1:

P(X=0) + P(X=1) + P(X=2)

=cnt({TT})/ cnt({HH, HT, TH, TT}) + cnt({HT,TH})/cnt({HH, HT, TH, TT}) + cnt({HH})/cnt({HH, HT, TH, TT})

=1/4 + 2/4 + 1/4

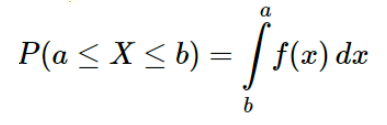
=1

ii) Probability Density Function(PDF):

For continuous variables, p(X=x) = 0, for all x∈R

P(weight = 76.34kg)=0, as it can be argued that weight 76.3!=76.34 and 76.341!=76.34

Hence it will be more appropriate to use P(weight between 76.3 to 76.4) for continuous variables. So instead of calculating the probability of the variable being a particular single value, the probability is calculated for the variable to be within a range of values.



The area between the limit values a and b in the function f(x) is calculated by using integration.

**23. What is correlation. Explain its type in details. What are the methods of determining correlation.**

**Answer**: - **Association of any two variables is known as Correlation**. It is the numerical measurement showing the degree of relation between two variables.

**Correlation:** It is a numerical measure of the direction and magnitude of the mutual relationship between the variables (X and Y).

1. **Positive correlation:** It is said to be positive when the values of the two variables move in the same direction so that an increase in one variable is followed by an increase in the other variable or a decrease in one variable is followed by a decrease in the other variable.

* Two variables X and Y are going in the same direction.
* If X rises, Y also rises, and vice-versa.
* **Examples of positive correlation are** (a) Age and Income, (b) Amount of rainfall, and the yield of the crop.

**2. Negative correlation:** It is said to be negative when the values of the two variables move in the opposite direction so that an increase in one variable is followed by a decrease in the other variable.

* Two variables X and Y are going in the opposite direction.
* If X rises, Y falls, and vice versa.
* **Examples of negative correlation are** (a) Height above sea level and temperature, (b) Sales of woollen clothes and temperature.

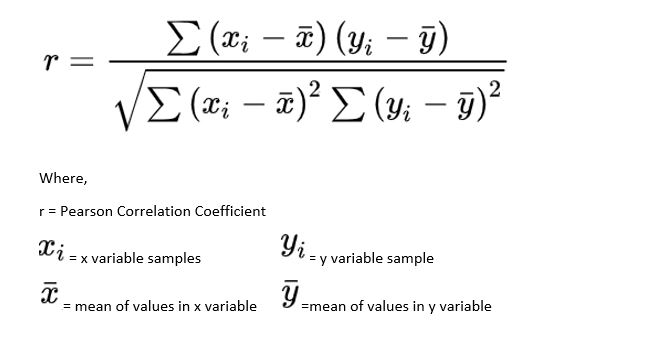
Types of Correlation Metrics

* Pearson Correlation
* Spearman’s Rank Correlation
* Kendall Rank Correlation
* Point Biserial Correlation

Pearson Correlation

Pearson correlation is a normalized measurement of the covariance. It also measures the linear relationship between two variables and fails to capture the non-linear relationship of two variables. Pearson correlation assumes that both variables are normally distributed. It can be used for nominal variables or continuous variables.

Pearson correlation coefficient between two variables X and Y can be calculated by the following formula:



Importing the necessary modules

from scipy.stats import pearsonr

import numpy as np

pearsonr(a,b)

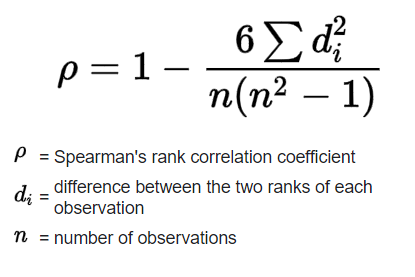
**Spearman’s Rank Correlation**

It is a non-parametric (no prior assumptions about distribution) measure

for calculating correlation coefficient that is used for ordinal variables or

continuous variables. Spearman’s rank correlation can capture both linear

or non-linear relationships.



import numpy as np

from scipy.stats import spearmanr

spearmanr(a,b)

**24. Calculate coefficient of correlation between the marks obtained by 10 students in Accountancy and statistics:  Use Karl Pearson’s Coefficient of Correlation Method to find it.**

**Ans: -**

r = ∑(x-x’) (y-y’)/ √(x-x’)2(y-y’)2

here x’ = 61 and y’ = 64

r = (45-61) (70-61) (65-61) (30-61) (90-61) (40-61) (50-61) (75-61) (85-61) (60-61) \* (35-64) (90-64) (70-64) (40-64) (95-64) (40-64) (60-64) (80-64) (80-64) (80-64) (50-64)

r = 0

Hence, it shows there is no correlation between the subject’s marks.

**25. Discuss the 4 differences between correlation and regression.**

**Definition: Correlation**is used to measure statistics that determine the

connection between two variables. **Regression**is used to represent the

connection between the independent and dependent variables.

**Usages: Correlation is used to**show the linear connection between two

variables. **Regression is used to**get the best data and to estimate a

single variable on the basis of other variables.

**Independent and dependent variables: In Correlation, t**here is no

difference between both variables. In **Regression**, both of the variables

are different from each other.

**Indicate:**The coefficient of **Correlation**signifies the extent to which the

two variables values that move together. **Regression** signifies the effect of

changes in the units that are known as a variable (X) on the estimated

variable (Y).

**Aim: Correlation helps t**o get the numeric values expressions’ relation

between variables. **Regression helps t**o determine the values of selected

variables on the basis of the fixed variables.

**Data representation: Correlation**represents a single point.

**Regression**can represent the data with a line.

**27. In a partially destroyed laboratory record of an analysis of correlation data, the following results only are legible: Variance of x = 9, Regression equations are: (i) 8x−10y = −66; (ii) 40x − 18y = 214. What are (a) the mean values of x and y, (b) the coefficient of correlation between x and y, (c) the σ of y**.

Answer: Given variance = 9

Std x = 3

10y = 8x+66…………. (1)

18y = 40x-214…………. (2)

From equation 1 and 2, x’ = 13, y’ = 17

**Mean of x = 13 and mean of y = 17**

Y = 8/10x +66/10

Byx = 4/5

X = 18/40y + 214/40

Bxy = 9/20

R= +-√9/20\*4/5

**R= 0.6**

Byx = r\*stdy/stdx

4/5 = 0.6\*stdy / 3

Stdy = 4

**28. What is Normal Distribution? What are the four Assumptions of Normal Distribution? Explain in detail**.

The Normal distribution is also known as **Gaussian** or **Gauss** distribution. Many groups follow this type of pattern.

There are several reasons why the normal distribution is crucial in statistics. Some of those are as follows:

**1.** The**statistical hypothesis test** assumes that the data follows a normal distribution.

**2.** Both linear and non-linear regression assumes that the**residual**follows the normal distribution.

**3.**Moreover, the central limit theorem states that as the **sample size** increases the distribution of the mean follows normal distribution irrespective of the distribution of the original variable

There are two main parameters of a normal distribution- the**mean**and**standard deviation**.

**1. Mean**

* Researchers used the mean or average value as a measure of central tendency. It can be used to describe the distribution of variables that are measured as ratios or intervals.
* The mean determines the location of the peak, and most of the data points are clustered around the mean in a normal distribution graph.
* If we change the value of the mean, then the curve of normal distribution moves either to the left or right along the X-axis.

**2. Standard Deviation**

* The standard deviation measures how the data points are dispersed relative to the mean.
* It determines how far the data points are away from the mean and represents the distance between the mean and the data points.
* The standard deviation defines the width of the graph. As a result, changing the value of standard deviation tightens or expands the width of the distribution along the x-axis.
* Usually, a smaller standard deviation wrt to the mean results in a steep curve while a larger standard deviation results in a flatter curve.
* **1.** One sample t-test: It’s assumed that the sample data is normally distributed.
* **2.** Two sample t-test: It’s assumed that both samples are normally distributed.
* **3.** ANOVA: It’s assumed that the residuals from the model are normally distributed.
* **4.** Linear regression: It’s assumed that the residuals from the model are normally distributed.

**29.Write all the characteristics or Properties of the Normal Distribution Curve.**

The total area under the bell curve is 1.

The bell curve is symmetric about its mean. i.e. area to the right side of the centre is exactly same as area to the left side of the bell curve.

The mean, median and mode are all equal.

It follows empirical rule. Empirical rule tells us about what percentage of data will fall within certain number of standard deviations from the mean.

About 68.2% of the data lies within 1 standard deviation.  
About 95.5% of the data lies within 2 standard deviations.  
About 99.7% of the data lies within 3 standard deviations

**30.Which of the following options are correct about Normal Distribution Curve.**

**(a) Within a range 0.6745 of σ on both sides the middle 50% of the observations occur i,e. mean ±0.6745σ covers 50% area 25% on each side.**

**(b) Mean ±1S.D. (i,e.µ ± 1σ) covers 68.268% area, 34.134 % area lies on either side of the mean.**

**(c) Mean ±2S.D. (i,e. µ ± 2σ) covers 95.45% area, 47.725% area lies on either side of the mean.**

**(d) Mean ±3 S.D. (i,e. µ ±3σ) covers 99.73% area, 49.856% area lies on the either side of the mean.  (e) Only 0.27% area is outside the range µ ±3σ.**

**Answer**

(a) Within a range 0.6745 of σ on both sides the middle 50% of the observations occur i,e. mean ±0.6745σ covers 50% area 25% on each side**.**

**31. The mean of a distribution is 60 with a standard deviation of 10. Assuming that the distribution is normal, what percentage of items be (i) between 60 and 72, (ii) between 50 and 60, (iii) beyond 72 and (iv) between 70 and 80?**

Answer: -

1. between 60 and 72

Z score = µ-x’/std

Z60 = 60-60/10 = 0 (corresponding z probability = 0.5)

Z72 = 72 – 60/ 10 = 1.2 (corresponding z probability = 0.8849)

Total area under the curve = 0.8849 -0.5 = 3849

**Hence 38.49% of items are between 60 and 72**

2. between 50 and 60

Z60 = 0 (corresponding z probability = 0.50)

Z50 = 50-60/10 = -1(corresponding z probability = 0.1587)

Total area under the curve = 0.5- 0.1587 = 0.3413

**Hence 34.13 % of items are between 50 and 60**

3. beyond 72

Z72 = 72 – 60/ 10 = 1.2 (corresponding z probability = 0.8849)

Area under the curve beyond 72 = 1-0.8849 = 0.1151

**Hence 11.51% of items are beyond 72.**

4.between 70 and 80

Z70 = 70-60/10 = 1 (corresponding z probability = 0.8413)

Z80 = 80 -60/10 = 2 (corresponding z probability = 0.9772)

Total area under the curve = = 0.9772 - 0.8413= 0.1359

**Hence 13.59% of items are between 70 and 80**

**32. 15000 students sat for an examination. The mean marks were 49 and the distribution**

**of marks had a standard deviation of 6. Assuming that the marks were normally distributed what proportion of students scored (a) more than 55 marks, (b) more than 70 marks.**

A) more than 55 marks

Z55 = 55 – 49/6 = 1 (Corresponding probability from z table = 0.8413)

Total area under the curve = 1- 0.8413 = 0.1587

Total students scored more than 55 = 1500\*0.1587

= 238

B) more than 70 marks.

Z70 = 70-49/6 = 21/6 = 3.5 (Corresponding probability from z table = 0.9998)

Total area under the curve = 1- 0.9998 = 0.0002 or 0.02%

Total students scored more than 70 = 1500\*0.0002

= 3

**33. If the height of 500 students is normally distributed with mean 65 inch and standard deviation 5 inch. How many students have height: a) greater than 70 inches. b) between 60 and 70 inches**.

a) greater than 70 inches

Z70 = 70-65/5 = 1 (Corresponding probability from z table = 0.8413)

Total area under the curve = 1- 0.8413 = 0.1587

Total students with height more than 70= 500\*0.1587 = 79

b) between 60 and 70 inches.

Z70 = 70-65/5 = 1 (Corresponding probability from z table = 0.8413)

Z60 = 60-65/5 = -1 (Corresponding probability from z table = 0.1587)

Total area under the curve = 0.8413 – 0.1587 = 0.6 826 or 68.26%

Total students with height between 60 and 70 = 500 \* 0.6826

= 341

**34. What is the statistical hypothesis? Explain the errors in hypothesis testing. b) Explain the Sample. What are Large Samples & Small Samples?**

**The hypothesis is a statement, assumption or claim about the value of the parameter (mean, variance, median etc.).**

A hypothesis is an educated guess about something in the world around you. It should be testable, either by experiment or observation.

The **null hypothesis** is the hypothesis to be tested for possible rejection under the assumption that it is true. It is denoted by H0.

The **alternative hypothesis** complements the Null hypothesis. It is the opposite of the null hypothesis such that both Alternate and null hypothesis together cover all the possible values of the population parameter. It is denoted by H1.

The critical region is that region in the sample space in which if the calculated value lies then we reject the null hypothesis. The critical region is a pre-defined area corresponding to a cut off value in the probability distribution curve. It is denoted by **α.**

The significance level, in the simplest of terms, is the threshold probability of incorrectly rejecting the null hypothesis when it is in fact true. This is also known as the type I error rate.

It is the probability of a type 1 error. It is also the size of the critical region.

Type I and type II error is one of the most important topics of hypothesis testing.

**A false positive (type I error)** — when you reject a true null hypothesis.

**A false negative (type II error)** — when you accept a false null hypothesis.

The probability of committing Type I error (False positive) is equal to the significance level or size of critical region α.

α= P [rejecting H0 when H0 is true]

The probability of committing Type II error (False negative) is equal to the beta β. It is called the ‘power of the test’.

β = P [not rejecting H0 when h1 is true]

The term sample refers to a smaller, manageable version of a larger group. It is a subset containing the characteristics of a larger population. Samples are used in statistical testing when population sizes are too large to include all possible members or observations. A sample should represent the population as a whole and not reflect any bias toward a specific attribute.

Large Samples:

* Large samples typically consist of a large number of observations, usually more than 30.
* With large samples, the Central Limit Theorem (CLT) applies, which states that the sampling distribution of the mean approaches a normal distribution regardless of the shape of the population distribution.
* The CLT allows for more accurate estimation of population parameters, such as the mean or proportion, using sample statistics.
* In large samples, the standard error of the estimate tends to be smaller, resulting in more precise and reliable estimates.
* Large samples are more likely to provide representative and unbiased estimates of the population parameters.
* Statistical tests and confidence intervals based on large samples tend to be more robust and accurate.

Small Samples:

* Small samples typically consist of a limited number of observations, usually less than 30.
* With small samples, the CLT may not apply, and the sampling distribution of the mean may not be normally distributed.
* In small samples, the standard error of the estimate tends to be larger, resulting in less precise and reliable estimates.
* Small samples are more susceptible to sampling variability and may not accurately represent the population.
* Statistical tests and confidence intervals based on small samples may be less reliable and may require additional assumptions or adjustments.
* Specialized statistical techniques, such as non-parametric tests, may be more appropriate for small samples.

**35. A random sample of size 25 from a population gives the sample standard derivation**

**to be 9.0. Test the hypothesis that the population standard derivation is 10.5.**

**Hint (Use chi-square distribution).**

**Answer:** Null Hypothesis (H0​): The population standard deviation is 10.5. (σ=10.5).

Alternative Hypothesis (H1​): The population standard deviation is not 10.5 (σ≠10.5).

The test statistic for the chi-square test for standard deviation is given by:

χ2=(n−1)s2/σ02 ​

n is the sample size

s is the sample standard deviation

σ0​ is the hypothesized population standard deviation

s2=9.02=81

σ0=10.52=110.25

χ2= (25−1) × 81/110.25 =24 × 81​/110.25 = 17.63

The degrees of freedom for the chi-square distribution in this test n-1 = 25-1 = 24

α=0.05

Lower critical value (for α/2=0.025) = 13.848

Upper critical value (for α/2=0.975(1- α/2) = 36.415

Since the test statistic (17.63) falls within the range of the critical values, we fail to reject the null hypothesis.

**37.100 students of a PW IOI obtained the following grades in Data Science paper : Grade :[A, B, C, D, E]  Total Frequency :[15, 17, 30, 22, 16, 100] Using the  χ 2 test , examine the hypothesis that the distribution of grades is uniform.**

Null Hypothesis (H0​): The grades are uniformly distributed.

Alternative Hypothesis (H1​): The grades are not uniformly distributed.

Expected Frequency for each grade= Total number of students /Number of grades

​=100/5​=20

The test statistic for the chi-square test is given by:

χ2=∑(Oi−Ei)2/Ei2

where:

* Oi is the observed frequency for grade = [15,17,30,22,16]
* Ei​ is the expected frequency for grade = [20,20,20,20,20]

χ2= (15−20)^2​/20+(17−20)^2/20​+ (30−20)^2/20​+(22−20) ^2/20​+ (16−20) ^2/20​

20

χ2 = 7.7

df = Number of categories−1=5−1=4

Using a chi-square distribution table or calculator for α=0.05 df=4:

Critical value≈9.488

If the test statistic is greater than the critical value, we reject the null hypothesis.

χ2=7.7and the critical value Is 9.488.

Since 7.7<9.4887.7 < 9.4887.7<9.488, we fail to reject the null hypothesis.

**38.Anova Test: To study the performance of two detergents and three different water temperatures the following whiteness readings were obtained with specially designed instrument**

|  |  |  |  |
| --- | --- | --- | --- |
| **Water temp** | **Detergent A** | **Detergent B** | **Detergent C** |
| **Cold water** | **57** | **55** | **67** |
| **Hot**  **water** | **49** | **42** | **68** |
| **Normal water** | **54** | **46** | **58** |

**Means:**

* **Mean for Detergent A: (57+49+54)/3=53.33**
* **Mean for Detergent B: (55+42+46)/3=47.67**
* **Mean for Detergent C: (67+68+58)/3=64.33**
* **Mean for Cold Water: (57+55+67)/3=59.67**
* **Mean for Hot Water: (49+42+68)/3=53.00**
* **Mean for Normal Water: (54+46+58)/3=52.67**
* **Overall Mean: (57+55+67+49+42+68+54+46+58)/9=55.33**

**Total Sum of Squares (SST): ∑(Xij−Xˉ)^2**

**Sum of Squares for Detergents (SSDetergent): 3×∑(Xˉi−Xˉ)^2**

**Sum of Squares for Water Temperatures (SSTemperature): 3×∑(Xˉj−Xˉ)^2**

**Error Sum of Squares (SSE): SST - SSDetergent – SSTemperature**

**Where Xij is the individual observation, Xˉi ​ is the mean for each detergent, Xˉj\​ is the mean for each temperature, and Xˉ is the overall mean.**

**SST=(57−55.33)2+(55−55.33)2+(67−55.33)2+(49−55.33)2+(42−55.33)2+(68−55.33)2+(54−55.33)2+(46−55.33)2+(58−55.33)2=424.67 SSDetergent=3×((53.33−55.33)2+(47.67−55.33)2+(64.33−55.33)2)=388.00**

**SSTemperature=3×((59.67−55.33)2+(53.00−55.33)2+(52.67−55.33)2)=36.67**

**SSE=SST−SSDetergent−SSTemperature=424.67−388.00−36.67=0**

**dfDetergent = Number of Detergents - 1 = 3 - 1 = 2**

**dfTemperature = Number of Temperatures - 1 = 3 - 1 = 2**

**dfError = (Number of Detergents - 1) \times (Number of Temperatures - 1) = 2 \times 2 = 4**

**dfTotal = Number of Observations - 1 = 9 - 1 = 8**

**Mean Squares:**

* **MSDetergent = SSDetergent / dfDetergent = 388.00 / 2 = 194.00**
* **MSTemperature = SSTemperature / dfTemperature = 36.67 / 2 = 18.33**
* **MSError = SSE / dfError = 0 / 4 = 0**

**F-Statistics:**

* **FDetergent = MSDetergent / MSError = 194.00 / 0 = ∞**
* **FTemperature = MSTemperature / MSError = 18.33 / 0 = ∞**

Given that the MSError is 0, the F-statistics for both detergents and temperature are infinite, which means that the variance due to detergents and temperatures is significantly different from the error variance. Therefore, we reject the null hypothesis that the mean whiteness readings are the same across different detergents and water temperatures.

This indicates that at least one detergent or temperature has a significantly different effect on the whiteness readings.