**Data Analysis (Analytics):**

Data Analysis or Data Analytics is studying, cleaning, modeling, and transforming data to find useful information, suggest conclusions, and support decision-making.

**Data Analysis Process**

Data Analysis is developed by the statistician John Tukey in the 1970s. It is a procedure for analyzing data, methods for interpreting the results of such systems, and modes of planning the group of data to make its analysis easier, more accurate, or more factual.

Therefore, data analysis is a process for getting large, unstructured data from different sources and converting it into information that is gone through the below process:

**Data Requirements Specification**

Data Collection

Data Processing

Data Cleaning

Data Analysis

Communication

**Need for Data Analysis**

Data analytics is significant for business optimization performance. An organization can also use data analytics to make better business decisions and support analyzing customer trends and fulfillment, which can lead to unknown and better products and services. Executing it into the business model indicates businesses can help reduce costs by recognizing more efficient modes of doing business.

**Applications of Data Analysis**

**Better decision-making:** The Key advantage of data analysis is better decision-making in the long term. Rather than depending only on knowledge, businesses are increasingly looking at data before deciding.

**Identification of potential risks:** Companies in today’s world succeed in high-risk conditions, but those environments require critical risk management processes, and extensive data has contributed to developing new risk management solutions. Data can enhance the effectiveness of actual simulations to predict future risks and create better planning.

**Increase the efficiency of work:** Data analysis allows you to analyze a large set of data and present it in a structured way to help reach your organization’s objectives. Possibilities and progress within the organization are reflected, and activities can increase work efficiency and productivity. It enables a culture of efficiency and collaboration by allowing managers to share detailed data with employees.

**Delivering relevant products:** Products are the oil for every organization, and often the most important asset of organizations. The role of the product management team is to determine trends that drive strategic creation, and activity plans for unique functions and services.

**Track customer behavioral changes**: Consumers have a lot to choose from in products available in the markets. Organizations have to pay attention to consumer demands and expectations, So to analyze the behavior of the customer data analysis is very important.

**Data Analysis Libraries**

* Pandas
* NumPy - Python Library
* Data Analysis with SciPy

**What is Pandas Libray in Python?**

Pandas is a powerful and versatile library that simplifies the tasks of data manipulation in Python. Pandas is well-suited for working with tabular data, such as spreadsheets or SQL tables.

The Pandas library is an essential tool for data analysts, scientists, and engineers working with structured data in Python.

**What is Python Pandas used for?**

The Pandas library is generally used for data science, but have you wondered why? This is because the Pandas library is used in conjunction with other libraries that are used for data science.

It is built on top of the NumPy library which means that a lot of the structures of NumPy are used or replicated in Pandas.

The data produced by Pandas is often used as input for plotting functions in Matplotlib, statistical analysis in SciPy, and machine learning algorithms in Scikit-learn.

You must be wondering, Why should you use the Pandas Library. Python’s Pandas library is the best tool to analyze, clean, and manipulate data.

Here is a list of things that we can do using Pandas.

Data set cleaning, merging, and joining.

Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data.

Columns can be inserted and deleted from DataFrame and higher-dimensional objects.

Powerful group by functionality for performing split-apply-combine operations on data sets.

Data Visualization.

**Getting Started with Pandas**

Let’s see how to start working with the Python Pandas library:

**Installing Pandas**

The first step in working with Pandas is to ensure whether it is installed in the system or not. If not, then we need to install it on our system using the pip command.

**Follow these steps to install Pandas:**

Step 1: Type ‘cmd’ in the search box and open it.

Step 2: Locate the folder using the cd command where the python-pip file has been installed.

Step 3: After locating it, type the command:

pip install pandas

**Importing Pandas**

After the Pandas have been installed in the system, you need to import the library. This module is generally imported as follows:

**import pandas as pd**

Note: Here, pd is referred to as an alias for the Pandas. However, it is not necessary to import the library using the alias, it just helps in writing less code every time a method or property is called.

**NumPy:**

NumPy is a general-purpose array-processing Python library which provides handy methods/functions for working n-dimensional arrays. NumPy is a short form for “Numerical Python“. It provides various computing tools such as comprehensive mathematical functions, and linear algebra routines.

NumPy provides both the flexibility of Python and the speed of well-optimized compiled C code.

Its easy-to-use syntax makes it highly accessible and productive for programmers from any background

**Why Numpy ?**

NumPy revolutionized the way we handle numerical data in Python. It is created to address the limitations of traditional Python lists when it comes to numerical computing. It is developed by Travis Olliphant in 2005.

NumPy provides a powerful array object that is both efficient and flexible. Its primary goal is to facilitate complex mathematical and scientific operations by introducing array-oriented computing capabilities. NumPy’s design allows for seamless integration with other scientific libraries, enabling faster execution of numerical tasks.

As a result, NumPy has become a cornerstone in the Python ecosystem, essential for data manipulation, machine learning, and scientific research.

**Hypothesis Testing:**

Hypothesis testing involves formulating assumptions about population parameters based on sample statistics and rigorously evaluating these assumptions against empirical evidence. This article sheds light on the significance of hypothesis testing and the critical steps involved in the process.

**What is Hypothesis Testing?**

Hypothesis testing is the act of testing whether a hypothesis or inference is true. When an alternate hypothesis is introduced, we test it against the null hypothesis to know which is correct. Let's use a plant experiment by a 12-year-old student to see how this works.

The hypothesis is that a plant will grow taller when given a certain type of fertilizer. The student takes two samples of the same plant, fertilizes one, and leaves the other unfertilized. He measures the plants' height every few days and records the results in a table.

After a week or two, he compares the final height of both plants to see which grew taller. If the plant given fertilizer grew taller, the hypothesis is established as fact. If not, the hypothesis is not supported. This simple experiment shows how to form a hypothesis, test it experimentally, and analyze the results.

In hypothesis testing, there are two types of error: Type I and Type II.

When we reject the null hypothesis in a case where it is correct, we've committed a Type I error. Type II errors occur when we fail to reject the null hypothesis when it is incorrect**.**

In our plant experiment above, if the student finds out that both plants' heights are the same at the end of the test period yet opines that fertilizer helps with plant growth, he has committed a Type I error.

However, if the fertilized plant comes out taller and the student records that both plants are the same or that the one without fertilizer grew taller, he has committed a Type II error because he has failed to reject the null hypothesis.

**What are the Steps in Hypothesis Testing?**

**The following steps explain how we can test a hypothesis:**

**Step #1 - Define the Null and Alternative Hypotheses**

Before making any test, we must first define what we are testing and what the default assumption is about the subject. In this article, we'll be testing if the average weight of 10-year-old children is more than 32kg.

Our null hypothesis is that 10 year old children weigh 32 kg on average. Our alternate hypothesis is that the average weight is more than 32kg. Ho denotes a null hypothesis, while H1 denotes an alternate hypothesis.

Ho = 32

H1 = 32

**Step #2 - Choose a Significance Level**

The significance level is a threshold for determining if the test is valid. It gives credibility to our hypothesis test to ensure we are not just luck-dependent but have enough evidence to support our claims. We usually set our significance level before conducting our tests. The criterion for determining our significance value is known as p-value.

A lower p-value means that there is stronger evidence against the null hypothesis, and therefore, a greater degree of significance. A p-value of 0.05 is widely accepted to be significant in most fields of science. P-values do not denote the probability of the outcome of the result, they just serve as a benchmark for determining whether our test result is due to chance. For our test, our p-value will be 0.05.

**Step #3 - Collect Data and Calculate a Test Statistic**

You can obtain your data from online data stores or conduct your research directly. Data can be scraped or researched online. The methodology might depend on the research you are trying to conduct.

We can calculate our test using any of the appropriate hypothesis tests. This can be a T-test, Z-test, Chi-squared, and so on. There are several hypothesis tests, each suiting different purposes and research questions. In this article, we'll use the T-test to run our hypothesis, but I'll explain the Z-test, and chi-squared too.

**T-test** is used for comparison of two sets of data when we don't know the population standard deviation. It's a parametric test, meaning it makes assumptions about the distribution of the data. These assumptions include that the data is normally distributed and that the variances of the two groups are equal. In a more simple and practical sense, imagine that we have test scores in a class for males and females, but we don't know how different or similar these scores are. We can use a t-test to see if there's a real difference.

**The Z-test** is used for comparison between two sets of data when the population standard deviation is known. It is also a parametric test, but it makes fewer assumptions about the distribution of data. The z-test assumes that the data is normally distributed, but it does not assume that the variances of the two groups are equal. In our class test example, with the t-test, we can say that if we already know how spread out the scores are in both groups, we can now use the z-test to see if there's a difference in the average scores.

**The Chi-squared** test is used to compare two or more categorical variables. The chi-squared test is a non-parametric test, meaning it does not make any assumptions about the distribution of data. It can be used to test a variety of hypotheses, including whether two or more groups have equal proportions.

**Step #4 - Decide on the Null Hypothesis Based on the Test Statistic and Significance Level**

After conducting our test and calculating the test statistic, we can compare its value to the predetermined significance level. If the test statistic falls beyond the significance level, we can decide to reject the null hypothesis, indicating that there is sufficient evidence to support our alternative hypothesis.

On the other contrary, if the test statistic does not exceed the significance level, we fail to reject the null hypothesis, signifying that we do not have enough statistical evidence to conclude in favor of the alternative hypothesis.

**Step #5 - Interpret the Results**

Depending on the decision made in the previous step, we can interpret the result in the context of our study and the practical implications. For our case study, we can interpret whether we have significant evidence to support our claim that the average weight of 10 year old children is more than 32kg or not.

For our test, we are generating random dummy data for the weight of the children. We'll use a t-test to evaluate whether our hypothesis is correct or not.

import numpy as np

import scipy.stats as stats

**# Create a dummy dataset of 10 year old children's weight**

**data = np.random.randint(20, 40, 10)**

**# Define the null hypothesis**

**H0 = "The average weight of 10 year old children is 32kg."**

**# Define the alternative hypothesis**

**H1 = "The average weight of 10 year old children is more than 32kg."**

**# Calculate the test statistic**

**t\_stat, p\_value = stats.ttest\_1samp(data, 32)**

**# Print the results**

**print("Test statistic:", t\_stat)**

**print("p-value:", p\_value)**

**# Conclusion**

**if p\_value < 0.05:**

**print("Reject the null hypothesis.")**

**else:**

**print("Fail to reject the null hypothesis.")**

**For a better understanding, let's look at what each block of code does.**

**import numpy as np**

**import scipy.stats as stats**

The first block is the import statement, where we import numpy and scipy.stats. Numpy is a Python library used for scientific computing. It has a large library of functions for working with arrays. Scipy is a library for mathematical functions. It has a stat module for performing statistical functions, and that's what we'll be using for our t-test.

**# Create a dummy dataset of 10 year old children's weight**

**data = np.random.randint(20, 40, 100)**

The weights of the children were generated at random since we aren't working with an actual dataset. The random module within the Numpy library provides a function for generating random numbers, which is randint.

The randint function takes three arguments. The first (20) is the lower bound of the random numbers to be generated. The second (40) is the upper bound, and the third (100) specifies the number of random integers to generate. That is, we are generating random weight values for 100 children. In real circumstances, these weight samples would have been obtained by taking the weight of the required number of children needed for the test**.**

**# Define the null hypothesis**

H0 = "The average weight of 10 year old children is 32kg."

**# Define the alternative hypothesis**

H1 = "The average weight of 10 year old children is more than 32kg."

Using the code above, we declared our null and alternate hypotheses stating the average weight of a 10-year-old in both cases.

**# Calculate the test statistic**

**t\_stat, p\_value = stats.ttest\_1samp(data, 32)**

t\_stat and p\_value are the variables in which we'll store the results of our functions. stats.ttest\_1samp is the function that calculates our test. It takes in two variables, the first is the data variable that stores the array of weights for children, and the second (32) is the value against which we'll test the mean of our array of weights or dataset in cases where we are using a real-world dataset.

**# Print the results**

**print("Test statistic:", t\_stat)**

**print("p-value:", p\_value)**

**The code above prints both values for t\_stats and p\_value.**

**# Conclusion**

**if p\_value < 0.05:**

**print("Reject the null hypothesis.")**

**else:**

**print("Fail to reject the null hypothesis.")**

Lastly, we evaluated our p\_value against our significance value, which is 0.05. If our p\_value is less than 0.05, we reject the null hypothesis. Otherwise, we fail to reject the null hypothesis. Below is the output of this program. Our null hypothesis was rejected.

**Test statistic: -5.114430435590074**

**p-value: 1.541000376540265e-06**

**Reject the null hypothesis.**