**AIM:**

To select the best sample and explain it using inferential Statistics.

**DESCRIPTION:**

**Sampling**

Sampling is the process of selecting a subset (sample) from a larger group (population) with the goal of making observations and drawing conclusions about the population.

**Purpose**: The main purpose of sampling is to gather information about a population in a cost-effective and efficient manner, without having to study the entire population.

**Inferential Statistics:**

Inferential statistics involve using sample data to make inferences or predictions about a population. It extends the findings from a sample to the entire population.

**Purpose**: The main purpose of inferential statistics is to draw conclusions, make predictions, or test hypotheses about a population based on a sample of data.

**Methods**:

**Hypothesis Testing:** Making decisions about population parameters based on sample data.

**Confidence Intervals:** Estimating the range within which a population parameter is likely to fall.

**CODE**:

**READ THE DATASET**

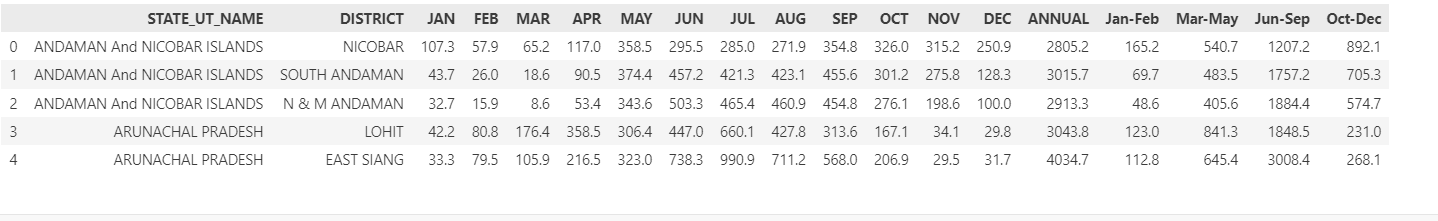
import pandas as pd

from scipy import stats

import numpy as np

df = pd.read\_csv('../RAIN DATASET/district wise rainfall normal.csv')

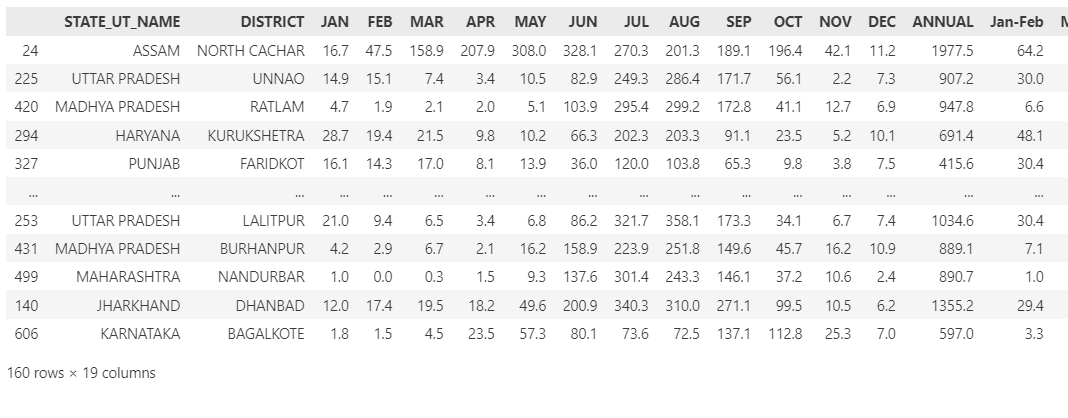
df.head()



**SAMPLING (RANDOM SAMPLE USING sample() method)**

samples=df.sample(frac=.25)

samples



We have selected ¼ th of the population as sample data.

Population- 641 rows

Sample- 160 rows

**DESCRIPTION STATISTICS ABOUT POPULATION**

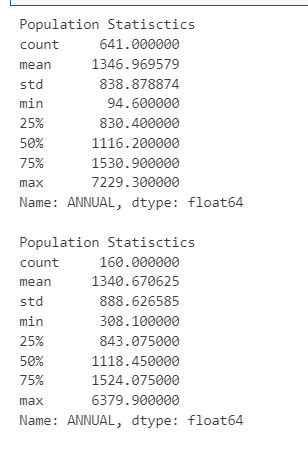
**COLUMN OF INTEREST = “ANNUAL”**

pop\_desc=df['ANNUAL'].describe()

sample\_desc=samples['ANNUAL'].describe()

print("Population Statisctics",pop\_desc,sep="\n",end="\n\n")

print("Population Statisctics",sample\_desc,sep="\n",end="\n")



**ANALYSING THE SAMPLE USING HYPOTHESIS TESTING (INFERENTIAL STASTICS)**

population\_mean=1346.97

sample\_annual=np.array(samples['ANNUAL'])

print(sample\_annual.mean())

t\_stat,p\_value=stats.ttest\_1samp(sample\_annual,population\_mean)

print('T-Statistic:', t\_stat)

print('P-value',p\_value)

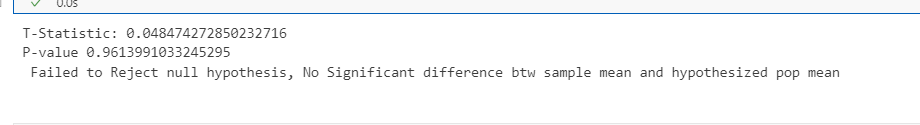
alpha=0.05

if p\_value<alpha:

    print("Reject null hypothesis, Significant difference btw sample mean and hypothesized pop mean")

else:

    print(" Failed to Reject null hypothesis, No Significant difference btw sample mean and hypothesized pop mean")



It is observed that we failed to failed to reject the null hypothesis and it means that there is not enough evidence in the sample data to reject the assumption stated in the null hypothesis. So we can use this sample data to make assumptions about population.

**FINDING CONFIDENCE INTERVAL**

# Calculate the 95% confidence interval for the population mean

confidence\_level = 0.95

n = len(sample\_annual)

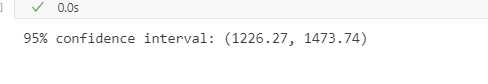
se = np.std(sample\_annual, ddof=1) / np.sqrt(n)

margin\_of\_error = stats.t.ppf(1 - (1 - confidence\_level) / 2, n-1) \* se

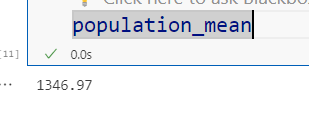
lower\_limit = np.mean(sample\_annual) - margin\_of\_error

upper\_limit = np.mean(sample\_annual) + margin\_of\_error

print(f"95% confidence interval: ({lower\_limit:.2f}, {upper\_limit:.2f})")



 The 95% confidence interval for the population mean is (1226.27, 1473.74). This means that we can be 95% confident that the true population mean falls within this range.



**PLOTING FREQUENCY DISTRIBUTION FOR POPULATION AND SAMPLE USINH HISTOGRAM**

import matplotlib.pyplot as plt

plt.figure(figsize=(10,8))

plt.hist(df['ANNUAL'],alpha=0.5)

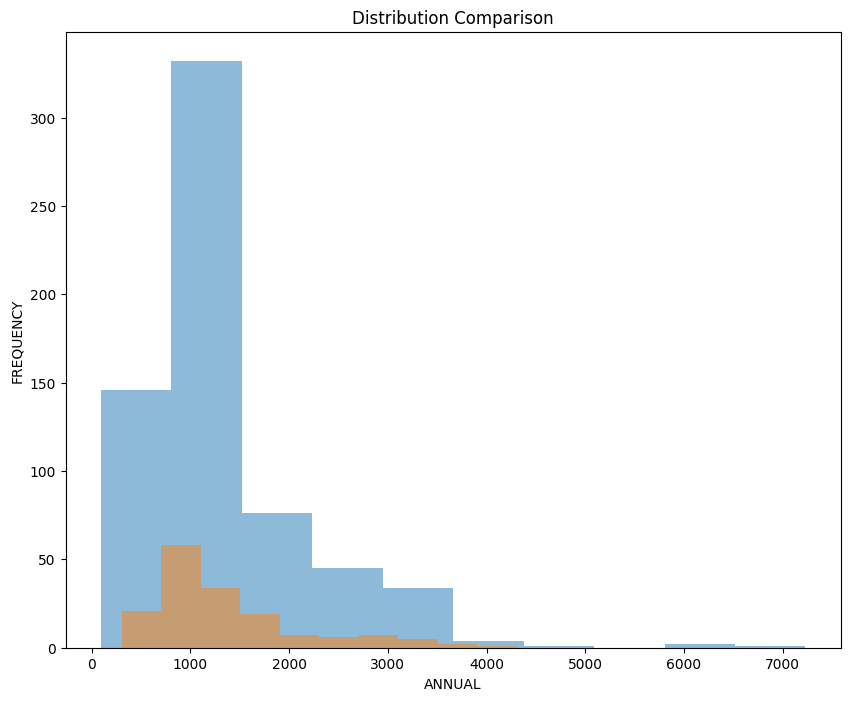
plt.hist(samples['ANNUAL'],alpha=0.5)

plt.title("Distribution Comparison")

plt.xlabel('ANNUAL')

plt.ylabel('FREQUENCY')

plt.show()



This histogram shows that the population sample and population and uniformally distributed.

**COEFFICIENT OF VARIATION FOR POPULATION**

p=np.array(df['ANNUAL'])

# Calculate the mean

mean = np.mean(p)

# Calculate the standard deviation

std\_dev = np.std(p)

# Calculate the coefficient of variation

cv = std\_dev / mean

print(f"Coefficient of Variation: {cv:.2f}")

****

**COEFFICIENT OF VARIATION FOR SAMPLE**

p=np.array(samples['ANNUAL'])

# Calculate the mean

mean = np.mean(p)

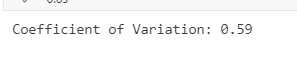
# Calculate the standard deviation

std\_dev = np.std(p)

# Calculate the coefficient of variation

cv = std\_dev / mean

print(f"Coefficient of Variation: {cv:.2f}")

****

It is observed that sample and population coefficient of variation are moreover same. We can infer that the sample is a true representation of the population