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:

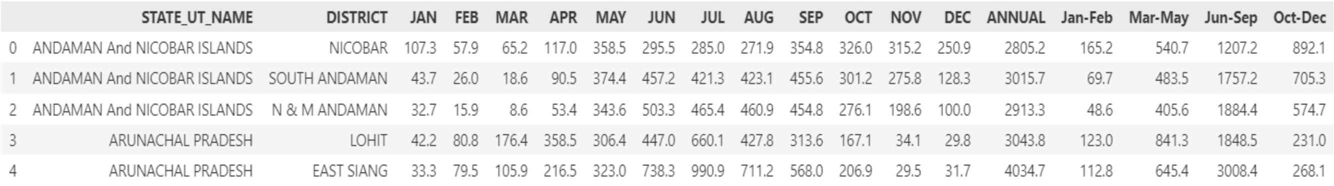
Hypothesis Testing: Making decisions about population parameters based on sample data.

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

CODE:

READ THE DATASET

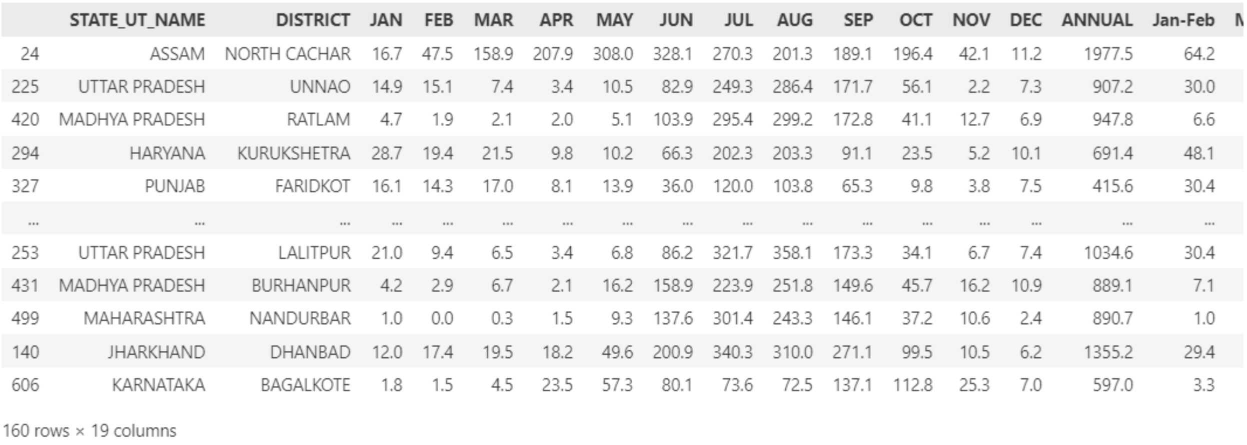
```
import pandas as pd
from scipy import stats
import numpy as np
df = pd.read_csv('../RAIN DATASET/district wise
rainfall normal.csv')


df.head()
```



	STATE_UT_NAME	DISTRICT	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	ANDAMAN And NICOBAR ISLANDS	NICOBAR	107.3	57.9	65.2	117.0	358.5	295.5	285.0	271.9	354.8	326.0	315.2	250.9	2805.2	165.2	540.7	1207.2	892.1
1	ANDAMAN And NICOBAR ISLANDS	SOUTH ANDAMAN	43.7	26.0	18.6	90.5	374.4	457.2	421.3	423.1	455.6	301.2	275.8	128.3	3015.7	69.7	483.5	1757.2	705.3
2	ANDAMAN And NICOBAR ISLANDS	N & M ANDAMAN	32.7	15.9	8.6	53.4	343.6	503.3	465.4	460.9	454.8	276.1	198.6	100.0	2913.3	48.6	405.6	1884.4	574.7
3	ARUNACHAL PRADESH	LOHIT	42.2	80.8	176.4	358.5	306.4	447.0	660.1	427.8	313.6	167.1	34.1	29.8	3043.8	123.0	841.3	1848.5	231.0
4	ARUNACHAL PRADESH	EAST SIANG	33.3	79.5	105.9	216.5	323.0	738.3	990.9	711.2	568.0	206.9	29.5	31.7	4034.7	112.8	645.4	3008.4	268.1

SAMPLING (RANDOM SAMPLE USING sample() method)

```
samples=df.sample(frac=.25)
samples
```



	STATE_UT_NAME	DISTRICT	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	M
24	ASSAM	NORTH CACHAR	16.7	47.5	158.9	207.9	308.0	328.1	270.3	201.3	189.1	196.4	42.1	11.2	1977.5	64.2	
225	UTTAR PRADESH	UNNAO	14.9	15.1	7.4	3.4	10.5	82.9	249.3	286.4	171.7	56.1	2.2	7.3	907.2	30.0	
420	MADHYA PRADESH	RATLAM	4.7	1.9	2.1	2.0	5.1	103.9	295.4	299.2	172.8	41.1	12.7	6.9	947.8	6.6	
294	HARYANA	KURUKSHETRA	28.7	19.4	21.5	9.8	10.2	66.3	202.3	203.3	91.1	23.5	5.2	10.1	691.4	48.1	
327	PUNJAB	FARIDKOT	16.1	14.3	17.0	8.1	13.9	36.0	120.0	103.8	65.3	9.8	3.8	7.5	415.6	30.4	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
253	UTTAR PRADESH	LALITPUR	21.0	9.4	6.5	3.4	6.8	86.2	321.7	358.1	173.3	34.1	6.7	7.4	1034.6	30.4	
431	MADHYA PRADESH	BURHANPUR	4.2	2.9	6.7	2.1	16.2	158.9	223.9	251.8	149.6	45.7	16.2	10.9	889.1	7.1	
499	MAHARASHTRA	NANDURBAR	1.0	0.0	0.3	1.5	9.3	137.6	301.4	243.3	146.1	37.2	10.6	2.4	890.7	1.0	
140	JHARKHAND	DHANBAD	12.0	17.4	19.5	18.2	49.6	200.9	340.3	310.0	271.1	99.5	10.5	6.2	1355.2	29.4	
606	KARNATAKA	BAGALKOTE	1.8	1.5	4.5	23.5	57.3	80.1	73.6	72.5	137.1	112.8	25.3	7.0	597.0	3.3	

160 rows × 19 columns

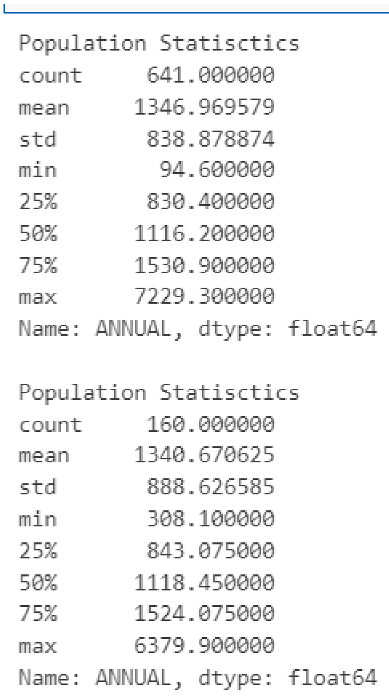
We have selected ¼ th of the population as sample data.

Population- 641 rows
Sample- 160 rows

DESCRIPTION STATISTICS ABOUT POPULATION

COLUMN OF INTEREST = "ANNUAL"

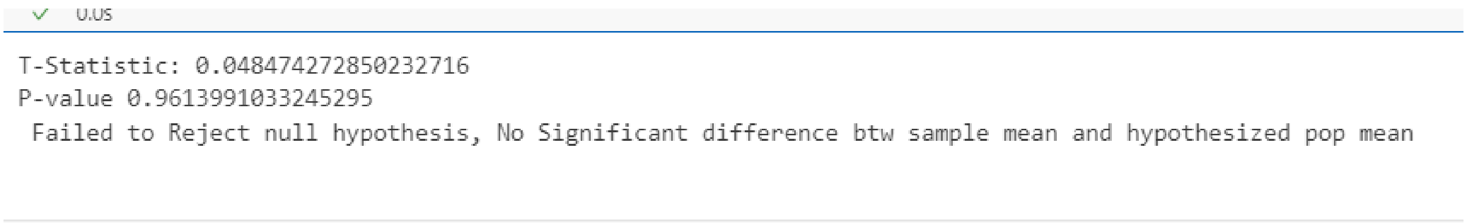
```
pop_desc=df['ANNUAL'].describe()
sample_desc=samples['ANNUAL'].describe()

print("Population Statisctics",pop_desc,sep="\n",end="\n\n")
print("Population Statisctics",sample_desc,sep="\n",end="\n")
```



```
Population Statisctics
count     641.000000
mean     1346.969579
std       838.878874
min        94.600000
25%       830.400000
50%      1116.200000
75%      1530.900000
max      7229.300000
Name: ANNUAL, dtype: float64

Population Statisctics
count     160.000000
mean     1340.670625
std       888.626585
min       308.100000
25%       843.075000
50%      1118.450000
75%      1524.075000
max      6379.900000
Name: ANNUAL, dtype: float64
```

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")
```

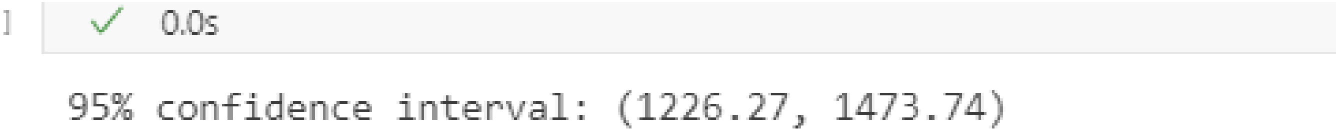


```
T-Statistic: 0.048474272850232716
P-value 0.9613991033245295
 Failed to Reject null hypothesis, No Significant difference btw sample mean and hypothesized pop mean
```

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.

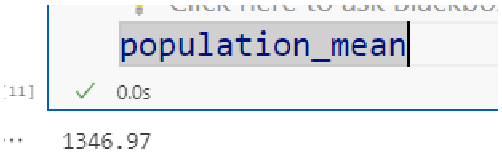
FINDING CONFIDENCE INTERVAL

```
# 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})")
```



```
✓ 0.0s

95% confidence interval: (1226.27, 1473.74)
```

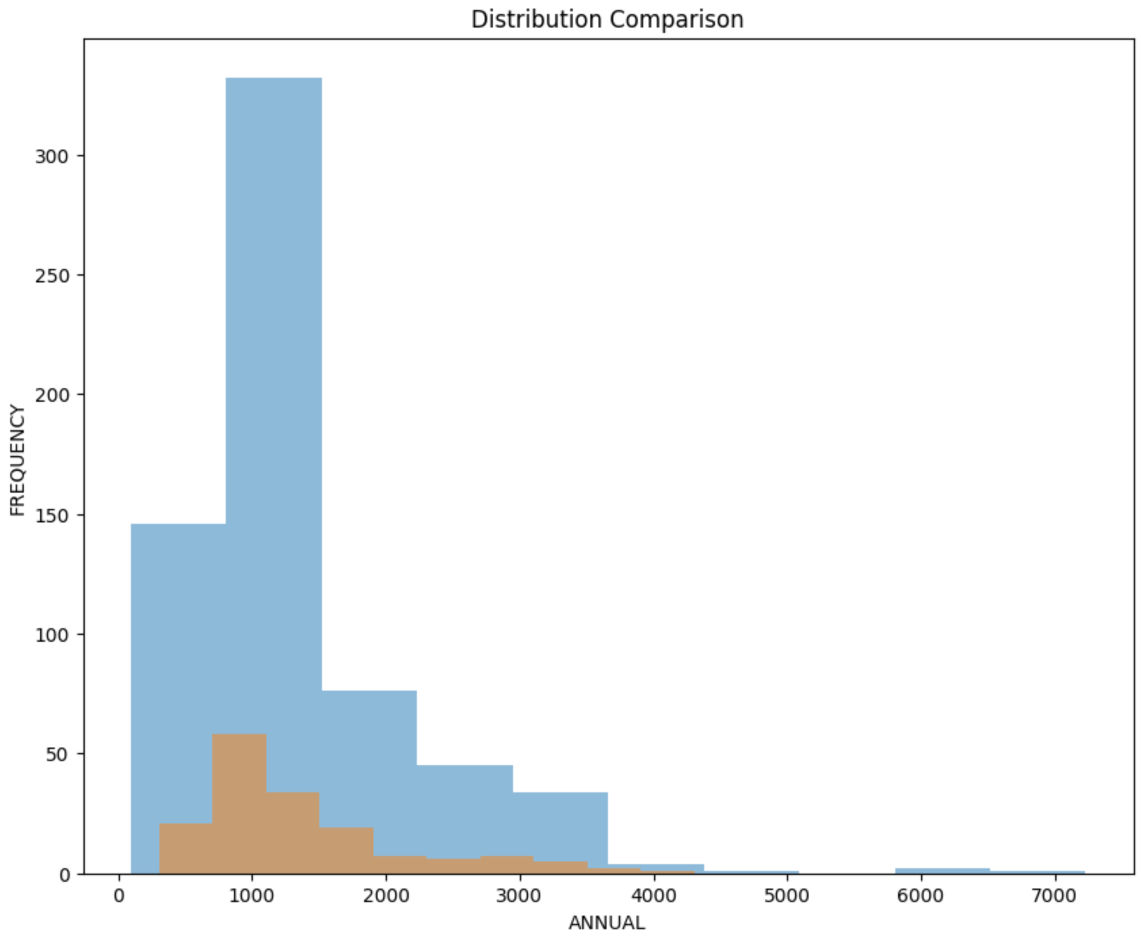
The 95% confidence interval for the population mean is (1226.27, 1473.74). This means that we can be 95% confident that the true population mean falls within this range.



```
population_mean
[11] ✓ 0.0s
··· 1346.97
```

PLOTING FREQUENCY DISTRIBUTION FOR POPULATION AND SAMPLE USINH HISTOGRAM

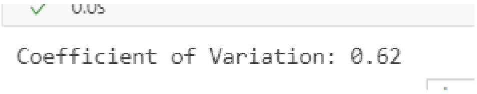
```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,8))
plt.hist(df['ANNUAL'],alpha=0.5)
plt.hist(samples['ANNUAL'],alpha=0.5)

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}")
```

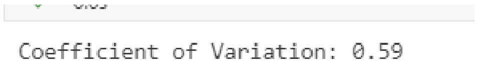


```
✓ 0.0s

Coefficient of Variation: 0.62
```

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}")
```



```
Coefficient of Variation: 0.59
```

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