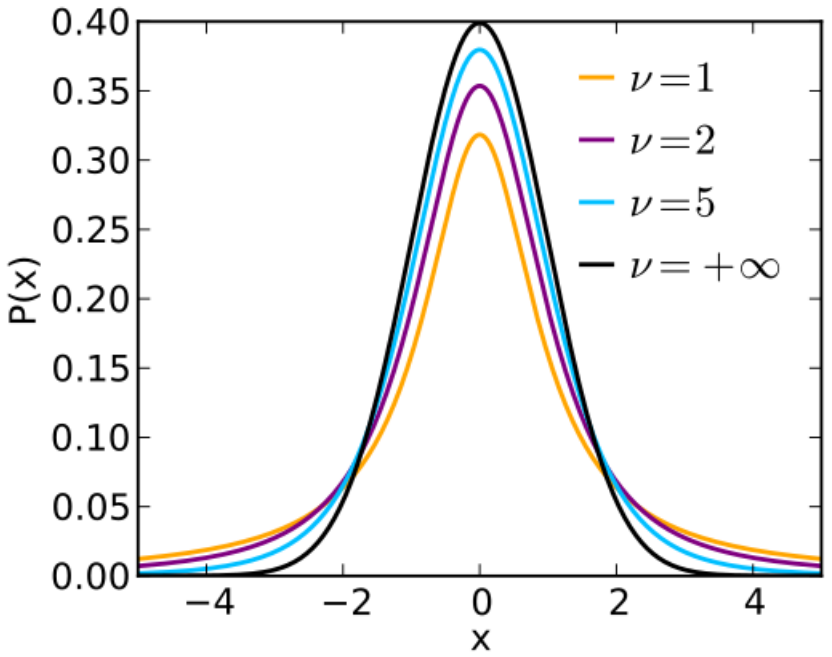
T-Test

T - test

- Test statistic follows a Student's t-distribution under the null hypothesis.
- Usage,
  - If the population variance is unknown and the sample size is not large $(n < 30)$

Student's t-distribution

- If we take a sample of *n* observations from a normal distribution, then the t-distribution with ν=n-1 degrees of freedom



Types of T-Test

- One-sample T-test
  - Whether the sample mean is statistically different from a known or hypothesised population mean.
- Two-sampled T-test
  - Compares the means of two independent groups to determine whether population means are significantly different

ONE-SAMPLE T-TEST

Consider the problem statement

- Is the Mean of the sample (1,2,3,4,5) equal to the population mean of 3.5 with confidence level of 95 percentage

Define Null and Alternate Hypothesis

Null Hypothesis

$$H_0 : \mu = 3.5$$

Alternative Hypothesis

$$H_1 : \mu \neq 3.5$$

Import necessary Libraries in Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

```
# Import t distribution
from scipy.stats import t
```

Calculate the sample statistics

```
sample=np.array([1,2,3,4,5])
print("Sample values",sample)
```

```
Sample values [1 2 3 4 5]
```

```
pop_mean=3.5
sample_mean=np.mean(sample)
sample_std=np.std(sample,ddof=1)
n=len(sample)
dof=n-1
cl=0.95
significant=0.05
print("Population Mean",pop_mean)
print("No of Samples",n)
print("Sample Mean",sample_mean)
print("Sample Standard Deviation",sample_std)
print("Degree of Freedome",dof)
print("Confidence level",cl)
print("Significant level",significant)
```

```
Population Mean 3.5
No of Samples 5
Sample Mean 3.0
Sample Standard Deviation 1.5811388300841898
Degree of Freedome 4
Confidence level 0.95
Significant level 0.05
```

T Statistics

```
tstatistics=(sample_mean-pop_mean)/(sample_std/np.sqrt(n))
print("T Statistics is",tstatistics)
```

```
T Statistics is -0.7071067811865475
```

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

$\bar{x}$ - Sample Mean

μ - Population Mean

s - Standard deviation of sample

n - Sample size

Critical Values

```
tcritical_l=t.ppf(q=significant/2,df=dof)
tcritical_u=-tcritical_l
print("Critical Values are ",tcritical_l,tcritical_u)
```

```
Critical Values are  -2.7764451051977996 2.7764451051977996
```

Decision Making - using Statistics and critical value

```
if tstatistics<tcritical_l or tstatistics>tcritical_u:
    print("Reject the Null hypothesis")
else:
    print("Fail to reject the Null hypothesis")
```

```
Fail to reject the Null hypothesis
```

Decision Making - using p-value

```
pvalue=2*t.cdf(tstatistics,df=dof)
print("pvalue",pvalue)
if pvalue<0.05:
    print("Reject the Null hypothesis")
else:
    print("Fail to reject the Null hypothesis")
```

```
pvalue 0.5185185185185183
Fail to reject the Null hypothesis
```

Standard Error

$$S.E. = \frac{s}{\sqrt{(n)}}$$

s - Sample SD

n - Number of samples

```
SE=sample_std/np.sqrt(n)
print("Standard Error",SE)
```

```
Standard Error 0.7071067811865476
```

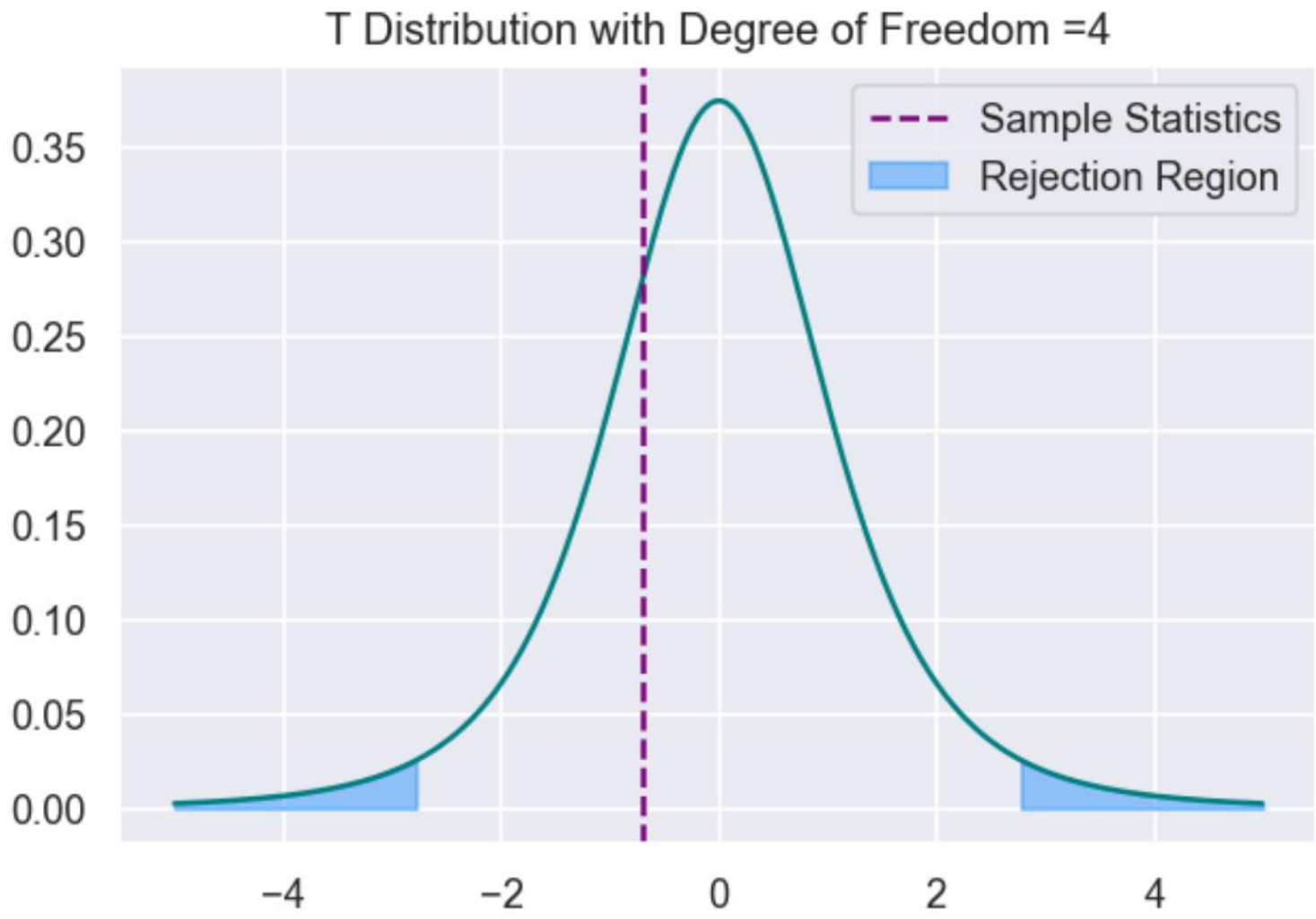
Confidence Interval

$$C.I. = \bar{x} \pm SE * Tcritical_{\frac{\alpha}{2}}$$

```
print("Confidence Interval",sample_mean+(SE)*np.array([tcritical_l,tcritical_u]))
```

```
Confidence Interval [1.03675684 4.96324316]
```

Plotting all the Findings



T-TEST USING INBUILT TTEST IN SCIPY

Single sample T Test using ttest_1samp from scipy

```
from scipy.stats import ttest_1samp
```

```
tstats,pvalue=ttest_1samp(sample, pop_mean)
print("Test Satistics",tstats)
print("pvalue",pvalue)
if pvalue<0.05:
    print("Reject the Null hypothesis")
else:
    print("Fail to reject the Null hypothesis")
```

```
Test Satistics -0.7071067811865475
pvalue 0.5185185185185183
Fail to reject the Null hypothesis
```

```
t.interval(0.95,dof,loc=sample_mean,scale=SE)
```

```
(1.036756838522439, 4.9632431614775605)
```

TWO-SAMPLE T-TEST

Null and Alternate Hypothesis

- Null Hypothesis
  - H0 : Mean of Sample1 = Mean of Sample 2
  - H0 : (Mean_Sample1 ~ Mean_Sample 2) =0
- Alternate Hypothesis
  - H1 : Mean_Sample 1 is different than Mean_Sample 2

Two Sample T-test

```
np.random.seed(123)
sample1=np.random.normal(4.7,1,20)
sample2=np.random.normal(5.3,1,20)
print("Mean of Sample1",np.mean(sample1))
print("Mean of Sample2",np.mean(sample2))
tstatistics, pvalue = ttest_ind(sample1,sample2)
print("\nT Statistics",tstatistics)
print("P value",pvalue)
if pvalue<0.05:
    print("\nReject the Null Hypothesis")
else:
    print("\nFails to reject the Null Hypothesis"
```

Result

Mean of Sample1 4.814417731955291
Mean of Sample2 5.018821601570846 4

T Statistics -0.5481788436227369
P value 0.5867752732889268

Fails to reject the Null Hypothesis