



→ V Adithya Sai

Import all Libraries and create t_score Function

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import tnorm
```

```
In [2]: def t_score(sample_size,sample_mean,population_mean,sample_std):
    numerator = sample_mean - population_mean
    denominator = sample_std / sample_size**0.5
    return numerator/denominator
```

Q-1: Pista House selling Hyderabadi Chicken Dum biryani claims that each parcel packet has 500 grams of biryani (also mentioned on the label of packet). You are sceptic of their claims and believe that on average each packet does not contain 500 grams of biryani.

How do you prove your claim?

Step – 1 :

Alternate hypothesis (bold claim) :

$H_1 = \text{Each packet does not contain 500 grams}$

Null hypothesis (status quo) :

$H_0 = \text{each packet contains 500grams}$

Step – 2 :

- *Collection of sample size : $n = 10$*

[490, 220, 370, 400, 495, 510, 398, 508, 476]

- *Compute sample mean :*

$$\bar{x} = ?$$

Step – 3 :

Compute test statistic (as population std. dev is not given) :

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

Step – 4 :

Decide α or significance level.

Step – 5.1 :

Calculate tailed t – test ?

Reject Null hypothesis or not ?

Step – 5.2 :

Calculate p – value ?

```
In [3]: sample = [490,220,370,400,495,510,398,508,476]
sample_mean = np.mean(sample)
sample_std = np.std(sample)
population_mean = 500
sample_size = 10
print(sample_mean)
print(sample_std)
```

429.6666666666667
89.4725283723073

```
In [4]: T_score = t_score(sample_size,sample_mean,population_mean,sample_std)
print(T_score)
```

-2.4858303751033297

```
In [5]: confidence_level = 0.95
df = 9 # df = n-1 = 10-1 = 9
alpha = 1 - confidence_level
t_critical = t.ppf(1-alpha/2,df) # 2 tailed test
print(t_critical)
```

2.2621571627409915

```
In [6]: #plotting the sample distribution with rejection region
x_min = 400
x_max = 600

#defining sample mean and sample std
mean = population_mean
std = sample_std/ sample_size**0.5

#plotting the graph and setting Limits
x = np.linspace(x_min,x_max,100)
y = norm.pdf(x, mean ,std)
plt.xlim(x_min,x_max)
plt.plot(x,y)

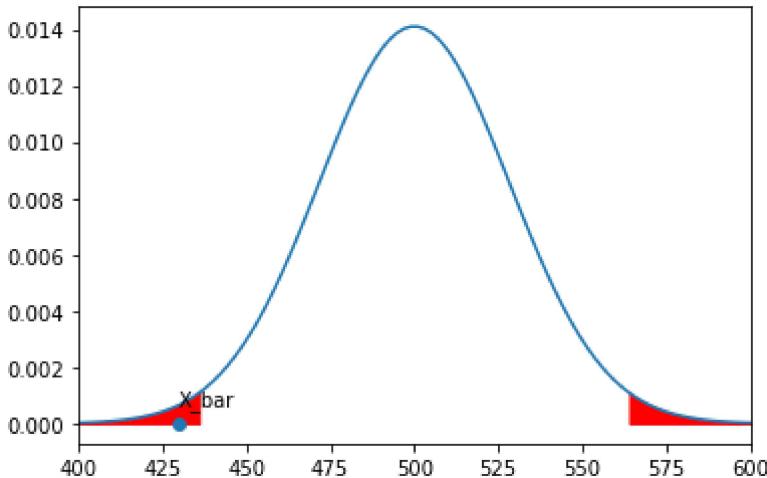
#Computing the Left and right critical values
t_critical_left = population_mean + (-t_critical*std)
t_critical_right = population_mean + (t_critical*std)

#Shading the Left rejection region
x1 = np.linspace(x_min,t_critical_left,100)
y1 = norm.pdf(x1,mean,std)
plt.fill_between(x1,y1,color='red')

#Shading the right rejection region
x2 = np.linspace(t_critical_right,x_max,100)
y2 = norm.pdf(x2,mean,std)
plt.fill_between(x2,y2,color='red')

#plotting the sample mean
plt.scatter(sample_mean,0)
plt.annotate('X_bar',(sample_mean,0.0007))
```

Out[6]: Text(429.6666666666667, 0.0007, 'X_bar')



```
In [7]: #conclusion using t-test
if(np.abs(T_score)>t_critical):
    print("Reject null hypothesis")
else:
    print("Failed to reject null hypothesis")
```

Reject null hypothesis

```
In [8]: #Conclusion using p-test:
p_value = 2 * (1.0 - norm.cdf(np.abs(T_score)))
print("P_value:",p_value)
if (p_value<alpha):
    print("Reject null hypothesis")
else:
    print("Failes to reject null hypothesis")
```

P_value: 0.012924955539059368
Reject null hypothesis

Q-2: You have developed a new Natural Language Processing Algorithms and done a user study. You claim that the average rating given by the users is greater than 4 on a scale of 1 to 5. How do you prove this to your client?

Step – 1 :

Alternate hypothesis (bold claim) :

$$H_1 = \text{All users rating are above 4}$$

Null hypothesis (status quo) :

$$H_0 = \text{Users ratings are not above 4}$$

Step – 2 :

- Collection of sample size : $n = 20$

[4, 3, 5, 4, 5, 5, 5, 5, 4, 4, 4, 4, 5, 5, 4, 4, 5, 4, 5, 4, 5]

- Compute sample mean :

$$\bar{x} = ?$$

Step – 3 :

Compute test statistic (as population std. dev is not given) :

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

Step – 4 :

Decide α or significance level.

Step – 5.1 :

Calculate tailed t – test ?

Reject Null hypothesis or not ?

Step – 5.2 :

Calculate p – value ?

```
In [9]: #Calculating sample mean, sample standard deviation
sample = [4,3,5,4,5,5,5,5,4,4,4,4,5,5,4,4,5,4,5,4,5]
sample_mean = np.mean(sample)
sample_std = np.std(sample)
population_mean = 4
sample_size = 20
print(sample_mean)
print(sample_std)
```

4.45

0.5894913061275798

```
In [10]: T_score = t_score(sample_size,sample_mean,population_mean,sample_std)
print(T_score)
```

3.413894588148968

```
In [11]: #Calculating t-critical
confidence_level = 0.95
df = 19
alpha = 1 - confidence_level
t_critical = t.ppf(1-alpha,df)
print(t_critical)
```

1.729132811521367

```
In [12]: #plotting the sampling distribution with rejection region
x_min = 2.5
x_max = 5.5

#defineing the sample mean and sample standard deviation
mean = population_mean
std = sample_std/(sample_size**0.5)

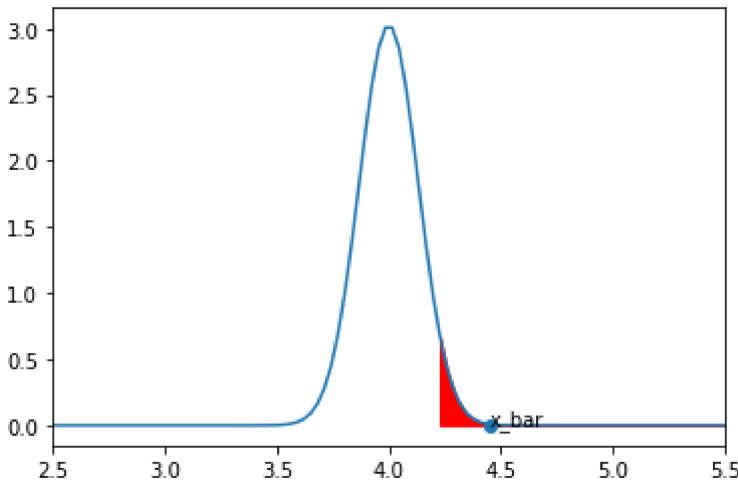
#plotting the graph and setting x_limit
x = np.linspace(x_min,x_max,100)
y = norm.pdf(x,mean,std)
plt.xlim(x_min,x_max)
plt.plot(x,y)

#Computing right critical t value
t_critical_right = population_mean + (t_critical * std)

#shading the right rejection area
x1 = np.linspace(t_critical_right,x_max,100)
y1 = norm.pdf(x1,mean,std)
plt.fill_between(x1,y1,color="red")
```

```
#plotting the sample_mean and concluding the results
plt.scatter(sample_mean,0)
plt.annotate("x_bar", (sample_mean,0.01))
```

Out[12]: Text(4.45, 0.01, 'x_bar')



```
In [13]: if (t_critical < T_score):
    print("Reject null hypothesis")
else:
    print("Failed to reject null hypothesis")
```

Reject null hypothesis

```
In [14]: #Conclusion using p-test
p_value = 1.0 - norm.cdf(np.abs(T_score))
print("P_value :",p_value)
if(p_value<alpha):
    print("Reject null hypothesis")
else:
    print("Failed to reject null hypothesis")
```

P_value : 0.00032020674155808226
Reject null hypothesis

Q-3: TATA has developed a better fuel management system for the SUV segment. They claim that with this system, on average the SUV's mileage is at least 15 km/litre?

Step – 1 :

Alternate hypothesis (bold claim) :

$$H_1 = \text{Average mileage is atleast } 15\text{km/litre}$$

Null hypothesis (status quo) :

$$H_0 = \text{Average mileage is not } 15\text{kmpf}$$

Step – 2 :

- Collection of sample size : $n = 10$

[13.78, 16.26, 15.30, 15.51, 14.83, 16.79, 15.98, 16.68, 12.33, 15.43]

- Compute sample mean :

$$\bar{x} = ?$$

Step – 3 :

Compute test statistic (as population std. dev is not given) :

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

Step – 4 :

Decide α or significance level.

Step – 5.1 :

Calculate tailed t – test ?

Reject Null hypothesis or not ?

Step – 5.2 :

Calculate p – value ?

```
In [15]: # calculate sample mean and sample standard deviation
sample = [13.78,16.26,15.30,15.51,14.83,16.79,15.98,16.68,12.33,15.43]
sample_mean = np.mean(sample)
sample_std = np.std(sample)
population_mean = 15
sample_size = 10
print(sample_mean)
print(sample_std)

15.289000000000001
1.3007878381965292
```

```
In [16]: T_score = t_score(sample_size,sample_mean,population_mean,sample_std)
print(T_score)

0.7025728692664717
```

```
In [17]: #Calculate t-critical
confidence_level = 0.95
alpha = 1 - confidence_level
df = 19
t_critical = t.ppf(1-alpha,df)
print(t_critical)

1.729132811521367
```

```
In [18]: #plotting sampling distribution with rejection region
x_min = 13
x_max = 17

#defineing the sample mean and sample std
mean = population_mean
std = sample_std / (sample_size**0.5)

#plotting the graph and setting x limits
x = np.linspace(x_min, x_max, 100)
y = norm.pdf(x, mean, std)
plt.xlim(x_min, x_max)
plt.plot(x, y)

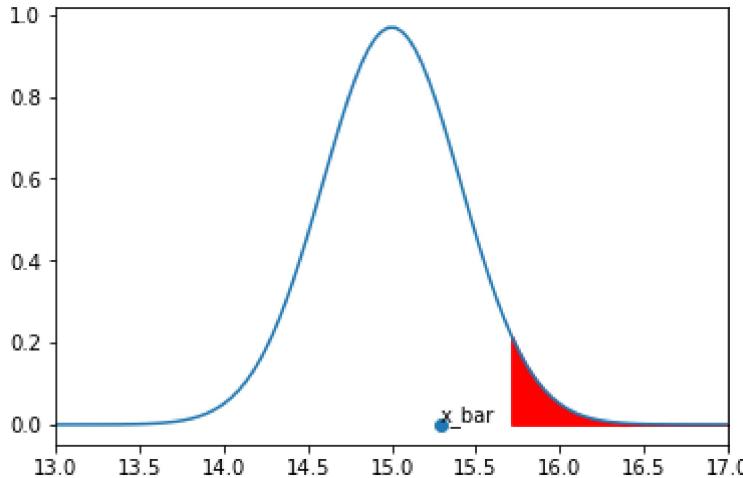
#defineing the t critical right value
t_critical_right = population_mean + (t_critical * std)

#shading the rejection region
```

```
x1 = np.linspace(t_critical_right,x_max,100)
y1 = norm.pdf(x1,mean, std)
plt.fill_between(x1,y1,color='red')

#plotting sample mean and concluding results
plt.scatter(sample_mean,0)
plt.annotate("x_bar", (sample_mean,0.01))
```

Out[18]: Text(15.289000000000001, 0.01, 'x_bar')



In [19]: `#Conclusion using t-test
if(T_score>t_critical):
 print("Reject null hypothesis")
else:
 print("Failed to reject null hypothesis")`

Failed to reject null hypothesis

In [20]: `#Conclusion using p-test
p_value = 1.0 - norm.cdf(np.abs(T_score))
print("P_value",p_value)
if (p_value<alpha):
 print("Reject null hypothesis")
else:
 print("Failed to reject null hypothesis")`

P_value 0.24116098758041715
Failed to reject null hypothesis

Q-4: You have developed a new Machine Learning Application and claim that on average it takes less than 100 ms to predict for any future datapoint. How do you convince your client about this claim?

Step – 1 :

Alternate hypothesis (bold claim) :

$$H1 = \text{Takes less than } 100\text{ms for prediction}$$

Null hypothesis (status quo) :

$$H0 = \text{Takes more than } 100\text{ms}$$

Step – 2 :

- Collection of sample size : $n = 20$

[80, 95, 94, 89, 101, 99, 96, 103, 87, 98, 99, 87, 106, 98, 98, 85, 97, 99, 102, 105]

- Compute sample mean :

$$\bar{x} = ?$$

Step - 3 :

Compute test statistic (as population std. dev is not given) :

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

Step - 4 :

Decide α or significance level.

Step - 5.1 :

Calculate tailed t - test ?

Reject Null hypothesis or not ?

Step - 5.2 :

Calculate p - value ?

```
In [21]: #Calculating sample mean and sample standard deviation
sample = [80,95,94,89,101,99,96,103,87,98,99,87,106,98,98,85,97,99,102,105]
sample_mean = np.mean(sample)
sample_std = np.std(sample)
population_mean = 100
sample_size = 20
print(sample_mean)
print(sample_std)
```

95.9

6.796322535018479

```
In [22]: #calculating t-score
T_score = t_score(sample_size,sample_mean,population_mean,sample_std)
print(T_score)
```

-2.697893945000713

```
In [23]: #calculating t-critical
confidence_level = 0.99
alpha = 1 - confidence_level
df = 19
t_critical = t.ppf(1-alpha,df)
print(t_critical)
```

2.539483190622288

```
In [24]: # plotting sample distribution with rejection region
x_min = 80
x_max = 120

#Calculating mean and standard deviation
mean = population_mean
std = sample_std / (sample_size**0.5)

#plotting the graph and setting xlims
x = np.linspace(x_min,x_max,100)
y = norm.pdf(x,mean,std)
plt.xlim(x_min,x_max)
```

```

plt.plot(x,y)

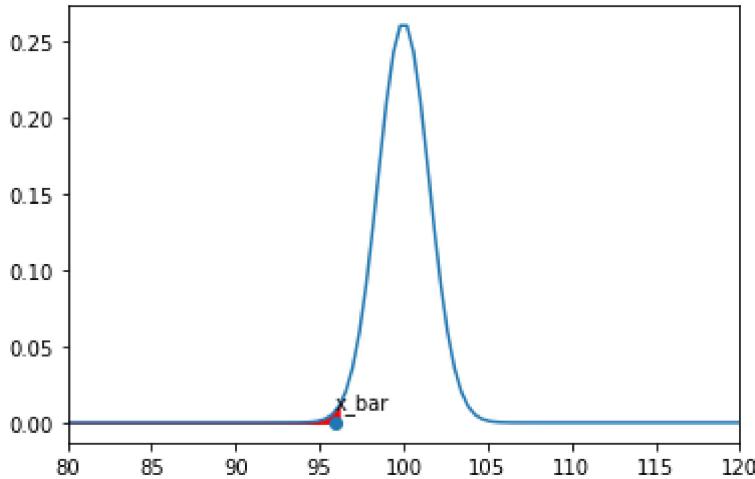
#defining t critical left value
t_critical_left = population_mean + (-t_critical * std)

#shading the rejection region
x1 = np.linspace(x_min,t_critical_left,100)
y1 = norm.pdf(x1,mean,std)
plt.fill_between(x1,y1,color='red')

#Plotting sample mean and conclusion
plt.scatter(sample_mean,0)
plt.annotate("x_bar", (sample_mean,0.01))

```

Out[24]: Text(95.9, 0.01, 'x_bar')



In [25]:

```

#conclusion using t_test
if (T_score<-t_critical):
    print("Reject null hypothesis")
else:
    print("Failed to reject null hypothesis")

```

Reject null hypothesis

In [26]:

```

#Conclusion using p- test
p_value = 1.0 - norm.cdf(np.abs(T_score))
print("P_value:",p_value)
if (p_value<alpha):
    print("Reject null hypothesis")
else:
    print("Failed to reject null hypothesis")

```

P_value: 0.0034889833663744874
Reject null hypothesis

In []:

In []: