```
import numpy as np
from scipy.special import gamma
def t_distribution_pdf(x, nu):
#given code by greg
    x = 3
    nu = 5
    coeff = gamma((nu + 1) / 2) / (np.sqrt(nu * np.pi) * gamma(nu / 2))
    density = coeff * (1 + x**2 / nu) ** (-0.5 * (nu + 1))
    return density

print(t_distribution_pdf(3,5)) # printing
```

```
0.017292578800222964
```

```
score = [92.64,79.00,84.79,97.41,93.68,65.23,84.50,73.49,73.97,79.11] # list with variables
mean=(sum(score))/len(score) # finding mean by getting sum of numbers and their total size using sum and len functions
def standdev(score): #calling function
    diff = 0 # define variable
    for i in score:  # for loop to loop through our list
        diff+=(i-mean)**2 # code to calculate st dev
        var=diff/(len(score)-1)
        stdev=(var)**(1/2)
    return stdev

print(mean) #printing results
print(standdev(score))
```

```
82.382
10.193467189005581
```

```
t0 = (mean-75)/((standdev(score))/(np.sqrt(len(score)))) # finding t0
print (t0)
```

```
2.290087686017293
```

```
prob = 0.95 # defining variables
nu=len(score)-1
x_start=0
x_end=20
num_points=10000

def t_distribution_pdf(x, nu):
#given code by greg
    coeff = gamma((nu + 1) / 2) / (np.sqrt(nu * np.pi) * gamma(nu / 2))
    density = coeff * (1 + x**2 / nu) ** (-0.5 * (nu + 1))
    return density

def find_t_star(prob, nu, x_start=0, x_end=20, num_points=10000): # greg code
# Define the x values
    x = np.linspace(x_start, x_end, num_points)
    # Apply the density function to the x values
    y = t_distribution_pdf(x, nu)
    # This next line is the integration (exercise: why does this work?)
    cdf = np.cumsum(y) * (x[1] - x[0])
    # Find the t-value where the cumulative probability reaches half of the
    target_half_prob = prob / 2
    index = np.where(cdf >= target_half_prob)[0][0]
    return x[index]
print(find_t_star(prob,nu,x_start,x_end,num_points)) # printing tstar
```

```
2.2522252225222523
```

```
if (find_t_star(prob,nu,x_start,x_end,num_points))<=t0<=(find_t_star(prob,nu,x_start,x_end,num_points)):
    print ("true")

else:
    print ("false")
```

```
false
```