problem 2 :

Q1:

```
In this problem we will use the T-test in order to find the          statistical results for a s
ample of population less than 30:
```

In [1]:
```
import numpy as np
import matplotlib.pyplot
import scipy.stats as stats
from scipy.stats import ttest_1samp
from scipy.stats import norm
from scipy.stats import t
```

In [25]:
```
#inputs

data = [3, -3, 3, 15, 15, -16, 14, 21, 30, -24, 32]

n0 = len(data) #sample size

df0 = n0 - 1

c_level0 = 0.9 # confidance interval level

alpha0 = (1-c_level0)/2
```

In [26]:
```
# statistical measurments

mean0 = np.mean(data)
print("Sample mean:",mean0)

std0 = np.std(data, ddof=1)
print("Sample Standard Deviation:",std0)

SE0 = std0 / (n0 ** (1/2))
print("Stander error:",SE0)

t0 = ttest_1samp(data, alpha0)[0]
#or
t0 = stats.t.ppf(1 - alpha0, df0)
print("t-value:",t0)

p0 = ttest_1samp(data, alpha0)[1]
#or
p0 = 2 * stats.t.cdf(-abs(t0), df0)
print("p-value: ",p0)

range0 = (mean0 - t0 * SE0, mean0 + t0 * SE0)
print("Interval Confidence Range: ",range0)
```

```
Sample mean: 8.181818181818182
Sample Standard Deviation: 17.70208000105175
Stander error: 5.337377942940253
t-value: 1.8124611228107335
p-value:  0.10000000000015384
Interval Confidence Range:  (-1.4919718375085527, 17.855608201144918)
```

Q2:

```
We will carry out the same statistical calculations for a 95%        confidence interval.
```

In [27]:
```
#inputs

c_level1 = 0.95 # confidance interval level

alpha1 = (1-c_level1)/2
```

In [28]:
```
# statistical measurments

mean1 = np.mean(data)
print("Sample mean:",mean1)

std1 = np.std(data, ddof=1)
print("Sample Standard Deviation:",std1)

SE1 = std1 / (n0 ** (1/2))
print("Stander error:",SE1)

t1 = ttest_1samp(data, alpha1)[0]
#or
t1 = stats.t.ppf(1 - alpha1, df0)
print("t-value:",t1)

p1 = ttest_1samp(data, alpha1)[1]
#or
p1 = 2 * stats.t.cdf(-abs(t1), df0)
print("p-value:",p1)

range1 = (mean1 - t1 * SE1, mean1 + t1 * SE1)
print("Interval Confidence Range: ",range1)
```

```
Sample mean: 8.181818181818182
Sample Standard Deviation: 17.70208000105175
Stander error: 5.337377942940253
t-value: 2.2281388519649385
p-value: 0.050000000000180862
Interval Confidence Range:  (-3.7106009804676994, 20.074237344104063)
```

As we can see the statistical results that don't depend on the confidence level haven't changed. However, the p-value and t-value have increased,

```
where for 90% level of confidence:
t-value = 1.812
p-value = 0.1
for 95% level of confidence:
t-value = 2.2281
p-value = 0.05
```

Q3:

```
For this part, we will use hypothesis testing (Z-score) for the given stander deviation (= 16.83
5) to carry out the needed calculations.
```

In [30]:
```
std2 = 16.836
print("Standard Deviation: ",std2)

SE2 = std2 / (n0**(1/2))
print("Standard Error: ",SE2)

z2 = stats.norm.ppf(1 - alpha1)
print("z-value: ",z2)

range2 = (mean0 - z2 * SE2, mean0 + z2 * SE2)
print("Interval Confidence Range: ",range2)
```

```
Standard Deviation:  16.836
Standard Error:  5.076244997311228
z-value:  1.959963984540054
Interval Confidence Range:  (-1.7674391896134498, 18.131075553249815)
```

Therefore, we result that all confidence level non-dependant results (ex. mean, standard deviation) will remain the same. On the other hand, the t-value, z-score, p-value, and range will differ depending on the confidence level.

Q4:

```
Finally, in this task, we will work on finding the optimized level of confidence that the team i
s expected to win on average.
```

In [32]:
```
t3 = mean0 / SE0 #t-value
p3 = 2 * stats.t.cdf(-abs(t3), df0)

c_level_op = (1 - p3)
print("The optimized level of confidance is {}". format(c_level_op))
print("i.e. Confidance Interval is {}%". format(int(c_level_op * 100)))
```

```
The optimized level of confidance is 0.8437015439608815
i.e. Confidance Interval is 84%
```

In [ ]: