ASSIGNMENT-02

STATISTICAL INFERENCE IN BIOENGINEERING

BM23MTECH11006
P TEJA VENKATA RAMANA KUMAR

Q1):-

```python
import numpy as np
import pandas as pd
from scipy import stats
```

```python
import numpy as np
import statsmodels.stats.api as sms
# Effect size
effect_size = 2
# Significance level (alpha)
alpha = 0.05
# Desired power
power = 0.95
```

a)

```python
# Calculate the minimum sample size
n=sms.tt_ind_solve_power(effect_size=effect_size, nobs1=None, alpha=alpha, power=power,
ratio=1.0, alternative='two-sided')
print("Minimum sample size:", n)
```
Minimum sample size: 7.60895791594358

b)

```python
alpha_new = 0.01
# Calculate the sample size for the new alpha
n_new = sms.tt_ind_solve_power(effect_size=effect_size, nobs1=None, alpha=alpha_new,
power=power, ratio=1.0, alternative='two-sided')
print("Sample size for alpha = 0.01:", n_new)
```
Sample size for alpha = 0.01: 10.710971202419103

C)

```python
power_new = 0.99
# Calculate the sample size for the new power
n_power = sms.tt_ind_solve_power(effect_size=effect_size, nobs1=None, alpha=alpha,
power=power_new, ratio=1.0, alternative='two-sided')
print("Sample size for power = 0.99:", n_power)
```
Sample size for power = 0.99: 10.267583224566895

d)

```python
n = 5
# Calculate the smallest detectable difference
detectable_difference = sms.tt_ind_solve_power(effect_size=None, nobs1=n, alpha=alpha,
power=power_new, ratio=1.0, alternative='two-sided')
print("Smallest difference, |µc – µd| =", detectable_difference*10)
```
Smallest difference, |µc – µd| = 31.22095933284507

e)

```python
desired_difference = 2
# Calculate the probability of detecting the desired difference
power_for_desired_difference = sms.tt_ind_solve_power(effect_size=desired_difference, nobs1=n,
alpha=alpha, power=None, ratio=1.0, alternative='two-sided')
print("Probabilityofdetectingadifference=", power_for_desired_difference)
```
Probability of detecting a difference = 0.790542363607261

Q2):-

a)

```python
desired_difference = 2
n_new=3
# Calculate the probability of detecting the desired difference
power_for_desired_difference = sms.tt_ind_solve_power(effect_size=desired_difference,
nobs1=n_new, alpha=alpha, power=None, ratio=1.0, alternative='two-sided')
print("n=3 then Power value =", power_for_desired_difference)
```
n=3 then Power value = 0.46264080923120443

b)

```python
without_delivery=np.array([138,155,121])
with_delivery=np.array([121,142,131])
# Set the significance level
alpha = 0.05
print(stats.ttest_ind(without_delivery, with_delivery, equal_var=True))
# Perform the two-sample t-test
t_stat, p_value = stats.ttest_ind(without_delivery, with_delivery)

# Check if the p-value is less than the significance level
if p_value < alpha:
    print(f"The P value = {p_value} < α = 0.05. So Null hypotheses rejected.\nThe delivery mechanism
works in this experiment.")
else:
    print(f"The P value = {p_value} > α = 0.05. So Null hypotheses can not be rejected.\nThe delivery
mechanism not works in this experiment.")
```
Ttest_indResult(statistic=0.5778319966569868, pvalue=0.5943464395237137)
The P value = 0.5943464395237137 > α = 0.05. **So Null hypotheses can not be rejected.**
The delivery mechanism not works in this experiment.

why ?

Q3):-

a)

```
#Calcium concentration in seawater μo
#The average value of the sample of calcium concentration in animals μa
#Null hypotheses H0: μo = μa
#Alternative hypotheses Ha: μo ≠ μa
```

(handwritten: 2)

b)

```
data=np.array([28, 27, 29, 29, 30, 30, 31, 30, 33, 27, 30, 32,31])
population_mean = 32
print(stats.ttest_1samp(data,population_mean))
t_stat, p_value = stats.ttest_1samp(data, population_mean)
if p_value < 0.05:
    print(f"The P value is ={p_value} < 0.05. So H0 can be rejected.")
else:
    print(f"The P value is ={p_value} > 0.05. So H0 can not be rejected.")
TtestResult(statistic=-4.501673049096729, pvalue=0.0007245394934862643, df=12)
```
The P value is =0.0007245394934862643 < 0.05. So H0 can be rejected.

(handwritten: ✓ 5)

c)

```
#Calcium concentration in seawater μo
# The average value of the sample of calcium concentration in animals μa
# Null hypotheses H0: μo > μa
# Alternative hypotheses Ha: μo < μa
```

(handwritten: ✗)

```
print(stats.ttest_1samp(data, population_mean, alternative='greater'))
t_stat, p_value = stats.ttest_1samp(data, population_mean, alternative='greater')
if p_value < 0.05:
    print(f"The P value is ={p_value} < 0.05. So H0 can be rejected. The species maintains a coelomic
calcium level greater than that of their environment.")
else:
    print(f"The P value is ={p_value} > 0.05. So H0 cannot be rejected. The species maintains a
coelomic calcium levelless than that of their environment")
TtestResult(statistic=-4.501673049096729, pvalue=0.9996377302532569, df=12)
```
The P value is =0.9996377302532569 > 0.05. So H0 cannot be rejected. The species maintains a coelomic calcium levelless than that of their environment

(handwritten: 2, ✗)

Q4):-

```
male=np.array([220.1,218.6,229.6,228.8,222.0,224.1,226.5])
female=np.array([223.4,221.5,230.2,224.3,223.8,230.8])
```

a)

The average value of the sample of Serum cholesterol concentrations in male μm

The average value of the sample of Serum cholesterol concentrations in female μf

Null hypotheses H0: μm = μf

Alternative hypotheses Ha: μm ≠ μf

b)

```
#Shapiro Test for checking Normality
print("shapiro male results:",stats.shapiro(male))
print("shapiro female results:",stats.shapiro(female))
_,p_value_male=stats.shapiro(male)
_,p_value_female=stats.shapiro(female)
if p_value_male > 0.05 and p_value_female > 0.05:
    print("Both male and female data appear to be normally distributed.")
else:
    print("At least one of the datasets does not appear to be normally distributed.")
print("By the Shpiro-Wilk normality test the P-value > Alpha. So Null hypotheses can not be rejected
and the data was normally distributed.")
```

shapiro male results: ShapiroResult(statistic=0.9439237117767334, pvalue=0.6742751598358154)
shapiro female results: ShapiroResult(statistic=0.841784656047821, pvalue=0.13486963510513306)
Both male and female data appear to be normally distributed.
**By the Shpiro-Wilk normality test the P-value > Alpha. So Null hypotheses can not be rejected and
the data was normally distributed.**

c)

This experiment has two independent data set groups: male & female
The data are normally distributed.
The aim was to compare the two distinct sample means.
Two sample t-test was suitable for the above conditions.

d)

```
print(stats.ttest_ind(male,female))
t_stat, p_value = stats.ttest_ind(male, female)
if p_value < 0.05:
    print("Reject the null hypothesis. There is evidence that male and female turtles have different
mean serum cholesterol concentrations.")
else:
    print(f"the P-value={p_value} > Alpha (0.05). There is no strong evidence to reject the null
hypotheses. So, the mean of both mele and female serum cholesterol concentrations are same.")
```
Ttest_indResult(statistic=-0.6268108404512706, pvalue=0.543572996867541)
**the P-value=0.543572996867541 > Alpha (0.05). There is no strong evidence to reject the null
hypotheses. So, the mean of both mele and female serum cholesterol concentrations are same.**

Q5):-

```
exam_after_coffee=np.array([72,64,81,90,73,54,87])
exam_after_tea=np.array([75,64,79,85,82,90,90])
```

a)

""" **The same group of students gave the two sets of exams in the two different situations. The two different situations involved consuming tea or coffee before the exam. The given data looks suitable for a paired sample t-test**."""

```python
# Shapiro Test for checking Normality
print(stats.shapiro(exam_after_coffee))
print(stats.shapiro(exam_after_tea))
_,p_value_coffee=stats.shapiro(exam_after_coffee)
_,p_value_tea=stats.shapiro(exam_after_tea)
if p_value_coffee > 0.05 and p_value_tea > 0.05:
    print("Both exam_after_coffee and exam after tea data appear to be normally distributed.")
else:
    print("At least one of the datasets does not appear to be normally distributed.")
print("By the Shpiro-Wilk normality test the P-value > Alpha. So Null hypotheses can not be rejected and the data was normally distributed.")
```

ShapiroResult(statistic=0.9621812105178833, pvalue=0.8372160792350769)
ShapiroResult(statistic=0.918850302696228, pvalue=0.46051329374313354)

Both exam_after_coffee and exam after tea data appear to be normally distributed.
By the Shpiro-Wilk normality test the P-value > Alpha. So Null hypotheses can not be rejected and the data was normally distributed.

```python
# Levene Test for checking Variance
stats.levene(exam_after_coffee,exam_after_tea)
```
LeveneResult(statistic=0.7067307692307687, pvalue=0.4169679986979009)

b)

"""**The paired observations should be independent of each other.**
• The variance of the pairs was approximately equal, and the data were considered normally distributed."""

c)

"""**The average value of the sample of exam after coffee μc**
The average value of the sample of exam after tea μt
Null hypotheses H0: (μc - μt) = 0
Alternative hypotheses Ha: (μm - μf) ≠ 0"""

```python
print(stats.ttest_rel(exam_after_coffee, exam_after_tea))
t_stat, p_value = stats.ttest_rel(exam_after_coffee, exam_after_tea)
if p_value < 0.05:
    print("Reject the null hypothesis. There is evidence that coffee and tea have different effects on exam performance.")
else:
    print(f"The P-value={p_value} > Alpha (0.05). Null hypotheses cannot be rejected. There is no significant difference between the means of the students' exams after the coffee and after the tea.")
```
TtestResult(statistic=-1.2025868531803559, pvalue=0.27443586794212094, df=6)

Q6):-

```python
import pandas as pd
from scipy import stats
BME=pd.DataFrame([78.49, 58.25, 76.33, 75.39, 73.00, 83.62, 67.81, 72.84, 82.02, 77.64, 71.44,
61.54, 66.20, 47.35, 82.45, 81.11,
72.84, 77.09, 69.41, 76.14, 72.05, 78.43, 59.73, 71.60, 82.22])
NonBME=pd.DataFrame([82.81, 78.36, 71.43, 55.71, 85.42, 26.17, 59.20, 62.38, 57.35, 70.45, 66.53,
74.57, 77.74, 73.85, 83.07,
83.66, 84.64, 85.65, 77.77, 69.47, 80.22, 74.72, 77.79])
```

a)

BME.describe()

count	25.000000
mean	72.599600
std	8.729939
min	47.350000
25%	69.410000
50%	73.000000
75%	78.430000
max	83.620000

BME.var()
76.211837

② *appropriate ?*

NonBME.describe()

count	23.000000
mean	72.128696
std	13.487674
min	26.170000
25%	68.000000
50%	74.720000
75%	81.515000
max	85.650000

NonBME.var()
181.917348

b)

```python
checking for which group is better performed
if int(BME.mean()>NonBME.mean()):
  print("BME performed Better")
else:
  print("Non BME group Performed Better")
```
BME performed Better

(handwritten: why mean? ②)

c)

```python
"""Mean meaks of BME Students µBME
 The mean of Non-BME Students µNon-BME
 Null hypotheses H0: µBME = µNon-BME
 Alternative hypotheses Ha: µBME ≠ µNon-BME"""
print(stats.ttest_ind(BME, NonBME))
t_stat, p_value = stats.ttest_ind(BME, NonBME)
t_stat,p_value
Ttest_indResult(statistic=array([0.14475831]), pvalue=array([0.8855341]))
(array([0.14475831]), array([0.8855341]))
```

(handwritten: why t-test? 10)

```python
if p_value < 0.05:
    print(" Conclusion: Reject the null hypothesis. Having a BME background may help in scoring higher marks.")
else:
    print(f"The P-value {p_value} > Alpha (0.05). The null hypotheses were true. The BME background did not provide any help in scoring higher marks in BM5033. ")
```
The P-value [0.8855341] > Alpha (0.05). The null hypotheses were true. The BME background did not provide any help in scoring higher marks in BM5033.