***Ans-***

1. Calculate the mean sales for each region.

```python

region\_a = [10, 15, 12, 8, 14]

region\_b = [18, 20, 16, 22, 25]

mean\_region\_a = sum(region\_a) / len(region\_a)

mean\_region\_b = sum(region\_b) / len(region\_b)

print("Mean sales for Region A:", mean\_region\_a)

print("Mean sales for Region B:", mean\_region\_b)

```

2. Calculate the mode of the survey responses.

```python

survey\_responses = [4, 5, 2, 3, 5, 4, 3, 2, 4, 5]

mode\_response = max(set(survey\_responses), key=survey\_responses.count)

print("Mode of the survey responses:", mode\_response)

```

3. Calculate the median salary for each department.

```python

department\_a = [5000, 6000, 5500, 7000]

department\_b = [4500, 5500, 5800, 6000, 5200]

sorted\_department\_a = sorted(department\_a)

sorted\_department\_b = sorted(department\_b)

median\_department\_a = sorted\_department\_a[len(sorted\_department\_a) // 2]

median\_department\_b = sorted\_department\_b[len(sorted\_department\_b) // 2]

print("Median salary for Department A:", median\_department\_a)

print("Median salary for Department B:", median\_department\_b)

```

4. Calculate the range of the stock prices.

```python

stock\_prices = [25.5, 24.8, 26.1, 25.3, 24.9]

price\_range = max(stock\_prices) - min(stock\_prices)

print("Range of the stock prices:", price\_range)

```

5. Perform a t-test to determine if there is a significant difference in the mean scores between the two groups.

```python

import scipy.stats as stats

group\_a = [85, 90, 92, 88, 91]

group\_b = [82, 88, 90, 86, 87]

t\_statistic, p\_value = stats.ttest\_ind(group\_a, group\_b)

alpha = 0.05

if p\_value < alpha:

print("There is a significant difference in the mean scores between the two groups.")

else:

print("There is no significant difference in the mean scores between the two groups.")

```

6. Calculate the correlation coefficient between advertising expenditure and sales.

```python

import numpy as np

advertising\_expenditure = [10, 15, 12, 8, 14]

sales = [25, 30, 28, 20, 26]

correlation\_coefficient = np.corrcoef(advertising\_expenditure, sales)[0, 1]

print("Correlation coefficient between advertising expenditure and sales:", correlation\_coefficient)

```

7. Calculate the standard deviation of the heights.

```python

heights = [160, 170, 165, 155, 175, 180, 170]

standard\_deviation = np.std(heights)

print("Standard deviation of the heights:", standard\_deviation)

```

8. Perform a linear regression analysis to predict job satisfaction based on employee tenure.

```python

import statsmodels.api as sm

employee\_tenure = [2, 3, 5, 4, 6, 2, 4]

job\_satisfaction = [7, 8, 6, 9, 5, 7, 6]

X = sm.add\_constant(employee\_tenure)

model = sm.OLS(job\_satisfaction, X)

results = model.fit()

intercept, slope = results.params

print("Intercept:", intercept)

print("Slope (coefficient for employee tenure):", slope)

```

9. Perform an analysis of variance (ANOVA) to determine if there is a significant difference in the mean recovery times between the two medications.

```python

medication\_a = [10, 12, 14, 11, 13]

medication\_b = [15, 17, 16, 14, 18]

f\_statistic, p\_value = stats.f\_oneway(medication\_a, medication\_b)

alpha = 0.05

if p\_value < alpha:

print("There is a significant difference in the mean recovery times between the two medications.")

else:

print("There is no significant difference in the mean recovery times between the two medications.")

```

10. Calculate the 75th percentile of the feedback ratings.

```python

feedback\_ratings = [8, 9, 7, 6, 8, 10, 9, 8, 7, 8]

percentile\_75 = np.percentile(feedback\_ratings, 75)

print("75th percentile of the feedback ratings:", percentile\_75)

```

11. Perform a hypothesis test to determine if the mean weight differs significantly from 10 grams.

```python

from scipy.stats import ttest\_1samp

weights = [10.2, 9.8, 10.0, 10.5, 10.3, 10.1]

t\_statistic, p\_value = ttest\_1samp(weights, 10)

alpha = 0.05

if p\_value < alpha:

print("The mean weight differs significantly from 10 grams.")

else:

print("The mean weight does not differ significantly from 10 grams.")

```

12. Perform a chi-square test to determine if there is a significant difference in the click-through rates between the two designs.

```python

from scipy.stats import chi2\_contingency

design\_a = [100, 120, 110, 90, 95]

design\_b = [80, 85, 90, 95, 100]

observed = np.array([design\_a, design\_b])

chi2, p\_value, \_, \_ = chi2\_contingency(observed)

alpha = 0.05

if p\_value < alpha:

print("There is a significant difference in the click-through rates between the two designs.")

else:

print("There is no significant difference in the click-through rates between the two designs.")

```

13. Calculate the 95% confidence interval for the population mean satisfaction score.

```python

satisfaction\_scores = [7, 9, 6, 8, 10, 7, 8, 9, 7, 8]

confidence\_interval = stats.t.interval(0.95, len(satisfaction\_scores) - 1, loc=np.mean(satisfaction\_scores), scale=stats.sem(satisfaction\_scores))

print("95% confidence interval for the population mean satisfaction score:", confidence\_interval)

```

14. Perform a simple linear regression to predict performance based on temperature.

```python

temperature = [20, 22, 23, 19, 21]

performance = [8, 7, 9, 6, 8]

X = sm.add\_constant(temperature)

model = sm.OLS(performance, X)

results = model.fit()

intercept, slope = results.params

print("Intercept:", intercept)

print("Slope (coefficient for temperature):", slope)

```

15. Perform a Mann-Whitney U test to determine if there is a significant difference in the median preferences between the two groups.

```python

from scipy.stats import mannwhitneyu

group\_a = [4, 3, 5, 2, 4]

group\_b = [3, 2, 4, 3, 3]

statistic, p\_value = mannwhitneyu(group\_a, group\_b)

alpha = 0.05

if p\_value < alpha:

print("There is a significant difference in the median preferences between the two groups.")

else:

print("There is no significant difference in the median preferences between the two groups.")

```

16. Calculate the interquartile range (IQR) of the ages.

```python

ages = [25, 30, 35, 40, 45, 50, 55, 60, 65, 70]

q1 = np.percentile(ages, 25)

q3 = np.percentile(ages, 75)

iqr = q3 - q1

print("Interquartile range (IQR) of the ages:", iqr)

```

17. Perform a Kruskal-Wallis test to determine if there is a significant difference in the median accuracy scores between the algorithms.

```python

algorithm\_a = [0.85, 0.80, 0.82, 0.87, 0.83]

algorithm\_b = [0.78, 0.82, 0.84, 0.80, 0.79]

algorithm\_c = [0.90, 0.88, 0.89, 0.86, 0.87]

statistic, p\_value = stats.kruskal(algorithm\_a, algorithm\_b, algorithm\_c)

alpha = 0.05

if p\_value < alpha:

print("There is a significant difference in the median accuracy scores between the algorithms.")

else:

print("There is no significant difference in the median accuracy scores between the algorithms.")

```

18. Perform a simple linear regression to predict sales based on price.

```python

price = [10, 15, 12, 8, 14]

sales = [100, 80, 90, 110, 95]

X = sm.add\_constant(price)

model = sm.OLS(sales, X)

results = model.fit()

intercept, slope = results.params

print("Intercept:", intercept)

print("Slope (coefficient for price):", slope)

```

19. Calculate the standard error of the mean satisfaction score.

```python

satisfaction\_scores = [7, 8, 9, 6, 8, 7, 9, 7, 8, 7]

standard\_error = stats.sem(satisfaction\_scores)

print("Standard error of the mean satisfaction score:", standard\_error)

```

20. Perform a multiple regression analysis to predict sales based on advertising expenditure.

```python

advertising\_expenditure = [10, 15, 12, 8, 14]

sales = [25, 30, 28, 20, 26]

X = sm.add\_constant(advertising\_expenditure)

model = sm.OLS(sales, X)

results = model.fit()

intercept, slope = results.params

print("Intercept:", intercept)

print("Slope (coefficient for advertising expenditure):", slope)

```