1. **What is the difference between descriptive statistics and inferential statistics? Explain with examples.**

Ans . Descriptive statistics consists of organising and summarising data for the entire population but in inferential statistics we take a sample and generalise for the entire population. Eg: Average weight of the class (descriptive) and average weight of the entire population(inferential).

2. **What is sampling in statistics? Explain the differences between random and stratified sampling.**

Ans . Sampling means to take a portion of entire population for observation and to generalise that observation for the entire population.

Random sampling means that every member of the population has an equal chance of being a part of the sample. Stratified Sampling involves dividing the population into distinct subgroups based on a shared characteristic (e.g., age, gender, income level), and then performing random sampling within each subgroup.

3. **Define mean, median, and mode. Explain why these measures of central tendency are important.**

Ans. Mean- average of all the data points

Median - after we arrange the data in ascending order the physical midpoint of the data is the median

Mode- The data that has the maximum frequency

Mean, median, and mode are essential tools in statistics because they summarize large datasets into a single representative value.

4. **Explain skewness and kurtosis. What does a positive skew imply about the data?**

Ans. skew- it represents the asymmetricity of the data .

Kurtosis- it represents the peakness of the data , whether the data has more or less peaked.

Positive skew implies that most of the data in the distribution lies on the right side.

5. **Implement a Python program to compute the mean, median, and mode of a given list of numbers.**
   numbers = [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]

Ans.

```
import numpy as np
number=[12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]
print(np.mean(number))
print(np.median(number))
```

✓ 0.0s

```
19.6
19.0
```

```
import statistics
statistics.mode(number)
```

✓ 0.0s

```
12
```

6. Compute the covariance and correlation coefficient between the following two datasets provided as lists in Python: list_x = [10, 20, 30, 40, 50] list_y = [15, 25, 35, 45, 60]

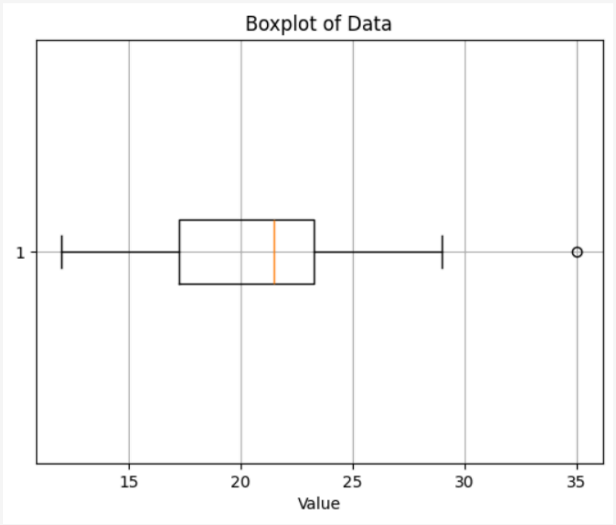
Ans.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Your data
list_x = [10, 20, 30, 40, 50]
list_y = [15, 25, 35, 45, 60]

# Create DataFrame
df = pd.DataFrame({'X': list_x, 'Y': list_y})

# Compute covariance and correlation
cov_xy = df['X'].cov(df['Y'])          # sample covariance
corr_xy = df['X'].corr(df['Y'])        # Pearson correlation

print(f"Covariance: {cov_xy}")
print(f"Correlation coefficient: {corr_xy:.6f}")
```

7. Write a Python script to draw a boxplot for the following numeric list and identify its outliers. Explain the result: data = [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35]

```python
import numpy as np
import matplotlib.pyplot as plt

data= [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35]
data_array=np.array(data)
q1=np.percentile(data,25)
q3=np.percentile(data,75)
iqr=q3-q1

lower_bound=q1-1.5*iqr
upper_bound=q3+1.5*iqr

outliers=[x for x in data if x<lower_bound or x>upper_bound]

print("outliers:",outliers)

plt.boxplot(data, vert=False)
plt.title('Boxplot of Data')
plt.xlabel('Value')
plt.grid(True)
plt.show()
```

```
outliers: [35]
```



8. : You are working as a data analyst in an e-commerce company. The marketing team wants to know if there is a relationship between advertising spend and daily sales. ● Explain how you would use covariance and correlation to explore this relationship. ● Write Python code to compute the correlation between the two lists: advertising_spend = [200, 250, 300, 400, 500] daily_sales = [2200, 2450, 2750, 3200, 4000]

Ans. Positive covariance says that with increase in advertising spend the daily sale has increased. Positive correlation near 1 will indicate that both the lists are highly correlated.

```python
import numpy as np
from scipy.stats import pearsonr

advertising_spend = [200, 250, 300, 400, 500]
daily_sales = [2200, 2450, 2750, 3200, 4000]

cov_var=np.cov(advertising_spend,daily_sales,bias=False)
corr=cov_var[0,1]

print(f"Covariance: {corr}")

corr_coeff, p_value = pearsonr(advertising_spend, daily_sales)
print(f"Pearson correlation coefficient: {corr_coeff:.4f}")
print(f"P-value for testing non-correlation: {p_value:.4g}")
```

✓ 0.0s

```
Covariance: 84875.0
Pearson correlation coefficient: 0.9936
P-value for testing non-correlation: 0.0006166
```

9. Your team has collected customer satisfaction survey data on a scale of 1-10 and wants to understand its distribution before launching a new product. ● Explain which summary statistics and visualizations (e.g. mean, standard deviation, histogram) you'd use. ● Write Python code to create a histogram using Matplotlib for the survey data: survey_scores = [7, 8, 5, 9, 6, 7, 8, 9, 10, 4, 7, 6, 9, 8, 7]

Ans I would use all of them.

```python
import matplotlib.pyplot as plt
import numpy as np
import statistics as stats

scores=[7, 8, 5, 9, 6, 7, 8, 9, 10, 4, 7, 6, 9, 8, 7]

mean=stats.mean(scores)
median=stats.median(scores)
mode=stats.mode(scores)
minimum,maximum=np.min(scores),np.max(scores)
print(f"mean: {mean}")
print(f"median: {median}")
print(f"mode: {mode}")
print(f"range: {minimum-maximum}")

plt.hist(scores, bins=range(4, 12), edgecolor='black', align='l
plt.title('Distribution of Customer Satisfaction Scores')
plt.xlabel('Survey Score')
plt.ylabel('Frequency')
```

```
mean: 7.333333333333333
median: 7
mode: 7
range: -6

Text(0, 0.5, 'Frequency')
```