Assignment Code: DS-AG-005

Statistics Basics| **Assignment**

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

Answer:

Descriptive statistics describe or summarize the characteristics of a dataset, such as its mean or frequency, without drawing conclusions about a larger population. For example, if you measure the heights of every student in a class and calculate the average, that's descriptive statistics.

Inferential statistics, on the other hand, uses data from a sample to make predictions or inferences about a larger population. An example would be taking a random sample of students from a university to estimate the average height of all students at that university.

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

Answer:

Sampling is the process of selecting a subset of a population to represent the entire group. This is often done to save time and resources when studying large populations.

- **Random sampling** is a method where every member of the population has an equal chance of being selected. This helps to reduce bias. For example, selecting 50 names from a list of 5,000 employees using a random number generator.
- **Stratified sampling** involves dividing the population into specific subgroups (strata) based on shared characteristics, such as age or gender. A random sample is then drawn from each stratum in proportion to its size in the population. This method ensures that the sample accurately reflects the demographic makeup of the population. For instance, if a company is 60% male and 40% female, a stratified sample of 100 people would include 60 men and 40 women.

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

Answer:

- **Mean:** The average of a dataset, calculated by summing all values and dividing by the number of values. It is sensitive to outliers.
- **Median:** The middle value of a dataset when it is ordered. It is not affected by outliers, making it a robust measure for skewed data.
- **Mode:** The value that appears most frequently in a dataset.

These measures of central tendency are important because they provide a single, representative value that summarizes the entire dataset. They help you quickly understand where the data is centered and how it is distributed.

Question 4: **E**xplain skewness and kurtosis. What does a positive skew imply about the data?

Answer:

- **Skewness** is a measure of the asymmetry of a distribution. A symmetrical distribution has zero skewness.
- **Kurtosis** measures the "tailedness" or "peakedness" of a distribution.

A **positive skew** indicates that the tail of the data distribution is longer on the right side. This means that the majority of data points are clustered to the left, and there are some high-value outliers pulling the mean to the right. In a positively skewed distribution, the mean is typically greater than the median.

Question 5: Implement a Python program to compute the mean, median, and mode of a given list of numbers.

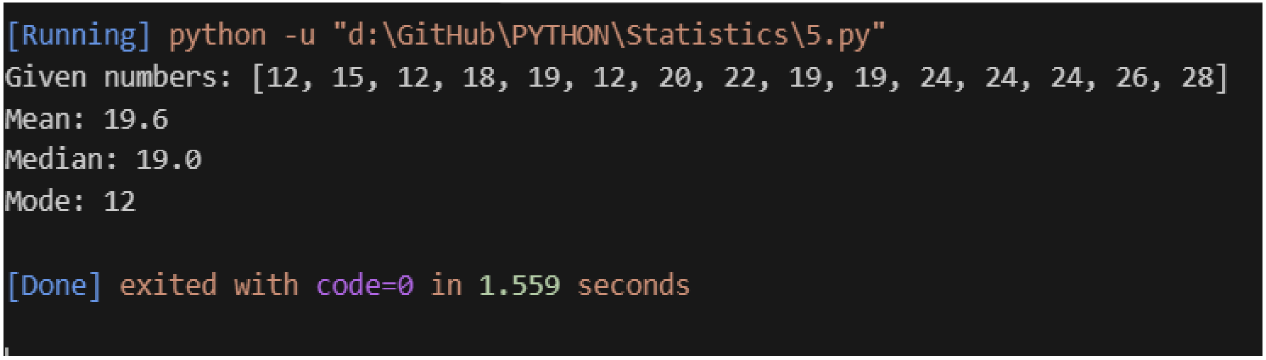
numbers = [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]

(Include your Python code and output in the code box below.)

Answer:

Paste your code and output inside the box below:

```
import numpy as np
from scipy import stats

numbers = [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]

# Compute Mean
mean_value = np.mean(numbers)

# Compute Median
median_value = np.median(numbers)

# Compute Mode
try:
   mode_result = stats.mode(numbers, keepdims=True)
   mode_value = mode_result.mode[0]
except TypeError:
   mode_result = stats.mode(numbers)
   mode_value = mode_result.mode[0]

print(f"Given numbers: {numbers}")
print(f"Mean: {mean_value}")
print(f"Median: {median_value}")
print(f"Mode: {mode_value}")
```



```
[Running] python -u "d:\GitHub\PYTHON\Statistics\5.py"
Given numbers: [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]
Mean: 19.6
Median: 19.0
Mode: 12

[Done] exited with code=0 in 1.559 seconds
```

Question 6: Compute the covariance and correlation coefficient between the following two datasets provided as lists in Python:

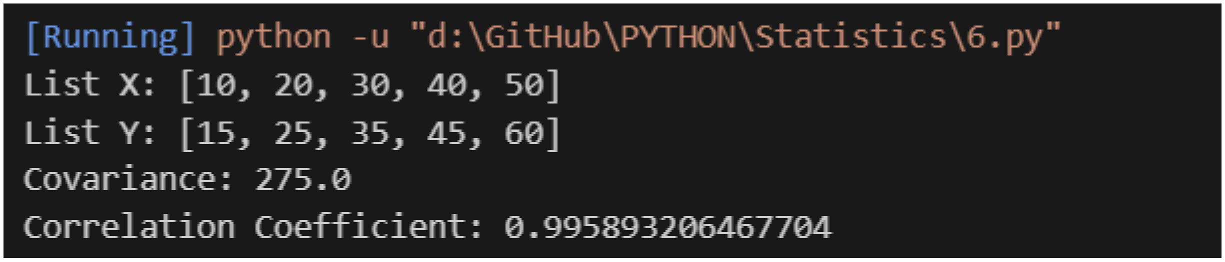
list_x = [10, 20, 30, 40, 50]
list_y = [15, 25, 35, 45, 60]

(*Include your Python code and output in the code box below.*)

Answer:

Paste your code and output inside the box below:

```
import numpy as np

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

# Compute Covariance
covariance = np.cov(list_x, list_y)[0, 1]

# Compute Correlation Coefficient
correlation_coefficient = np.corrcoef(list_x, list_y)[0, 1]

print(f"List X: {list_x}")
print(f"List Y: {list_y}")
print(f"Covariance: {covariance}")
print(f"Correlation Coefficient: {correlation_coefficient}")
```



```
[Running] python -u "d:\GitHub\PYTHON\Statistics\6.py"
List X: [10, 20, 30, 40, 50]
List Y: [15, 25, 35, 45, 60]
Covariance: 275.0
Correlation Coefficient: 0.995893206467704
```

Question 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]

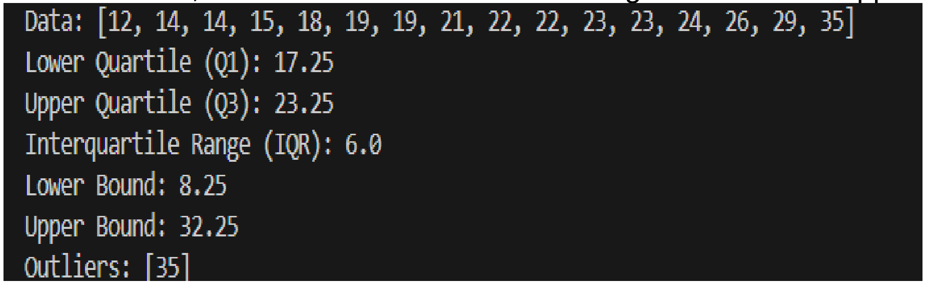
(*Include your Python code and output in the code box below.*)

Answer:

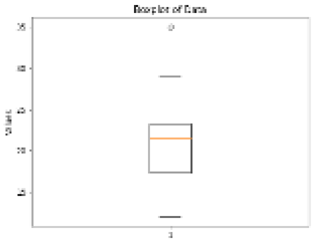
```
import matplotlib.pyplot as plt
import numpy as np

data = [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35]

plt.boxplot(data)
plt.title('Boxplot of Data')
plt.ylabel('Values')
plt.show()

# Identifying outliers using the IQR method
q1, q3 = np.percentile(data, [25, 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(f"Data: {data}")
print(f"Lower Quartile (Q1): {q1}")
print(f"Upper Quartile (Q3): {q3}")
print(f"Interquartile Range (IQR): {iqr}")
print(f"Lower Bound: {lower_bound}")
print(f"Upper Bound: {upper_bound}")
print(f"Outliers: {outliers}")
```

Explanation:

The boxplot visually represents the distribution of the data, with the box showing the interquartile range (IQR) from Q1 to Q3. The line inside the box is the median. The "whiskers" extend to the minimum and maximum values that are not outliers. The outlier is defined as a data point that falls below the lower bound ($Q1 - 1.5 * IQR$) or above the upper bound ($Q3 + 1.5 * IQR$). Based on the calculation, the value 35 is an outlier as it is greater than the upper bound of 29.25**.**



```
Data: [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35]
Lower Quartile (Q1): 17.25
Upper Quartile (Q3): 23.25
Interquartile Range (IQR): 6.0
Lower Bound: 8.25
Upper Bound: 32.25
Outliers: [35]
```



Question 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]

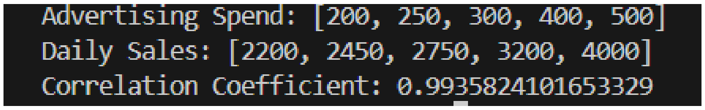
daily_sales = [2200, 2450, 2750, 3200, 4000]

(*Include your Python code and output in the code box below.*)

Answer:
To explore the relationship between advertising spend and daily sales, you would use covariance and correlation. Covariance shows the direction of the relationship (positive or negative), but its value is not standardized and can be difficult to interpret. The correlation coefficient, however, is a standardized measure between -1 and 1 that indicates both the strength and direction of the linear relationship. A value close to 1 suggests a strong positive relationship, meaning as advertising spend increases, daily sales also increase. This is the more useful metric for the marketing team as it provides a clear, interpretable value.

Paste your code and output inside the box below:

```
import numpy as np

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

# Compute the correlation coefficient
correlation = np.corrcoef(advertising_spend, daily_sales)[0, 1]

print(f"Advertising Spend: {advertising_spend}")
print(f"Daily Sales: {daily_sales}")
print(f"Correlation Coefficient: {correlation}")
```



```
Advertising Spend: [200, 250, 300, 400, 500]
Daily Sales: [2200, 2450, 2750, 3200, 4000]
Correlation Coefficient: 0.9935824101653329
```

Question 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]

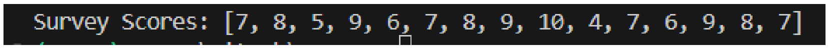
(*Include your Python code and output in the code box below.*)

Answer:
To understand the distribution of customer satisfaction survey data, you would use both summary statistics and visualizations.

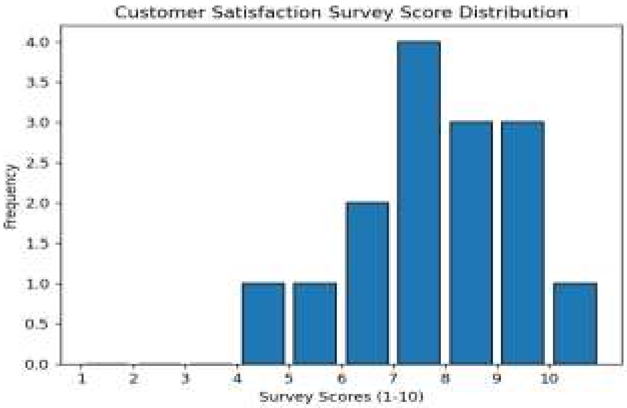
- Summary Statistics: You'd use the mean, median, and mode to find the typical satisfaction score and check for skewness. The standard deviation would be used to measure the spread of the data, indicating how much variation there is in the scores.
- Visualizations: A histogram would be the most effective visualization to show the distribution of scores. It would quickly reveal which scores are most frequent and whether the data is concentrated at the high end (indicating good satisfaction) or low end of the scale.

Paste your code and output inside the box below**:**

```
import matplotlib.pyplot as plt

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

plt.hist(survey_scores, bins=range(1, 12), edgecolor='black', rwidth=0.8)
plt.title('Customer Satisfaction Survey Score Distribution')
plt.xlabel('Survey Scores (1-10)')
plt.ylabel('Frequency')
plt.xticks(range(1, 11))
plt.show()

print(f"Survey Scores: {survey_scores}")
```



```
Survey Scores: [7, 8, 5, 9, 6, 7, 8, 9, 10, 4, 7, 6, 9, 8, 7]
```


Customer Satisfaction Survey Score Distribution
Frequency
Survey Scores (1-10)