Assignment Code: DS-AG-005

Statistics Basics| **Assignment**

Instructions: Carefully read each question. Use Google Docs, Microsoft Word, or a similar tool to create a document where you type out each question along with its answer. Save the document as a PDF, and then upload it to the LMS. Please do not zip or archive the files before uploading them. Each question carries 20 marks.

Total Marks: 200

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

Answer:

Descriptive statistics and **inferential statistics** are two key branches of statistics, differing in purpose and application:

- **Descriptive Statistics**: Summarizes and describes the main features of a dataset without drawing conclusions beyond it. It uses measures like **mean, median, mode, standard deviation, and visualizations** (histograms, boxplots) to present data.
    - *Example*: A teacher calculates the **average marks** of students in her class to describe their performance.

- **Inferential Statistics**: Goes beyond the given data to make **predictions, generalizations, or decisions** about a larger population, using sampling and probability theory. It involves **hypothesis testing, confidence intervals, and regression**.
    - *Example*: A researcher takes a **sample of 100 voters** to infer the **voting preference of the entire city**.

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

Answer:

Sampling in statistics is the process of selecting a **subset of individuals (sample)** from a larger **population** to draw conclusions or make inferences about the whole population. It is widely used because studying an entire population is often **impractical, costly, or time-consuming**.

Differences between Random Sampling and Stratified Sampling:

Aspect	Random Sampling	Stratified Sampling
Definition	Every individual in the population has an **equal chance** of being selected.	The population is divided into **strata (groups)** based on characteristics, and samples are taken from each group.
Selection Method	Purely by chance, using methods like lottery or random number generators.	Ensures representation from each subgroup, often proportionally.
Use Case	Best when the population is **homogeneous**.	Best when the population is **heterogeneous** with distinct subgroups.
Example	Randomly picking 100 students from a school of 1000.	Dividing the school into **grades (strata)** and then sampling students proportionally from each grade.

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

Answer:

Mean, Median, and Mode are the three main **measures of central tendency**, which describe the center or typical value of a dataset:

- **Mean**: The arithmetic average, calculated by summing all values and dividing by the number of observations.
  - *Example*: Mean of [2, 4, 6] = (2+4+6)/3 = 4.
- **Median**: The middle value when data is arranged in order. If the dataset has an even number of values, it is the average of the two middle numbers.
  - *Example*: Median of [2, 4, 6] = 4; Median of [1, 3, 5, 7] = (3+5)/2 = 4.
- **Mode**: The most frequently occurring value(s) in a dataset.
  - *Example*: Mode of [2, 4, 4, 6] = 4.

Importance:
These measures help **summarize large datasets with a single representative value**, making it easier to understand and compare distributions. They are widely used in **statistics, economics, research, and data analysis** to identify trends, typical behavior, or central patterns.

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

Answer:

Skewness and **kurtosis** are statistical measures that describe the **shape of a distribution**:

- **Skewness**: Measures the **asymmetry** of a distribution.
  - **Positive Skew (Right-skewed)**: Tail is longer on the right; most data values lie to the left of the mean. Example: income distribution (many low, few very high).
  - **Negative Skew (Left-skewed)**: Tail is longer on the left; most data values lie to the right of the mean.
- **Kurtosis**: Measures the **tailedness** or sharpness of a distribution compared to a normal distribution.
  - **High Kurtosis (Leptokurtic)**: Heavier tails and sharper peak (more outliers).
  - **Low Kurtosis (Platykurtic)**: Lighter tails and flatter peak (fewer outliers).
  - **Normal Kurtosis (Mesokurtic)**: Similar to a normal bell curve.

Positive skew implies the dataset has a **long right tail**, meaning a majority of observations are **below the mean**, with a few very large values pulling the mean to the right.

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:

Python Code:

```
import numpy as np
import pandas as pd

# Given list of numbers
numbers = [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]

# Using NumPy
mean_value = np.mean(numbers)
median_value = np.median(numbers)

# Using Pandas (for mode, since NumPy doesn't have a direct mode function)
mode_value = pd.Series(numbers).mode()[0]

# Print results
print("Mean:", mean_value)
print("Median:", median_value)
print("Mode:", mode_value)
```

Output:

```
Mean: 19.6
Median: 19.0
Mode: 12
```

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:

Python Code:

```
import numpy as np
import pandas as pd

cov_matrix = np.cov(list_x, list_y, bias=True)
cov_xy = cov_matrix[0, 1]

corr_xy = np.corrcoef(list_x, list_y)[0, 1]

print("Covariance:", cov_xy)
print("Correlation Coefficient:", corr_xy)
```

Output:

```
Covariance: 187.5
Correlation Coefficient: 0.9938586931957764
```

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

# --- Boxplot ---
plt.figure(figsize=(5,4))
plt.boxplot(data, vert=False, patch_artist=True, boxprops=dict(facecolor="lightblue"))
plt.title("Boxplot of Given Data")
plt.xlabel("Values")
plt.show()

# --- Outlier detection using IQR ---
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("Q1:", Q1)
print("Q3:", Q3)
print("IQR:", IQR)
print("Lower Bound:", lower_bound)
print("Upper Bound:", upper_bound)
print("Outliers:", outliers)
```

Output:

```
Q1: 18.25
Q3: 24.25
IQR: 6.0
Lower Bound: 9.25
Upper Bound: 33.25
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:

- **Covariance** shows the **direction** of the relationship between two variables (positive = move together, negative = move in opposite directions). But it doesn't standardize the values, so the magnitude is not easily interpretable.
- **Correlation** standardizes covariance by dividing by the standard deviations, giving a value between **-1 and +1**. This makes it easier to interpret:
  - **+1** → perfect positive relationship
  - **0** → no linear relationship
  - **-1** → perfect negative relationship

In this case, covariance will tell us if higher **advertising spend** tends to increase **sales**, while correlation will show **how strong** that relationship is.

Python Code:

```
import numpy as np
import pandas as pd

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

# --- NumPy ---
# Covariance matrix
cov_matrix = np.cov(advertising_spend, daily_sales, bias=True)
cov_xy = cov_matrix[0, 1]

# Correlation coefficient
corr_xy = np.corrcoef(advertising_spend, daily_sales)[0, 1]

print("NumPy Results:")
print("Covariance:", cov_xy)
print("Correlation:", corr_xy)
```

Output:

```
NumPy Results:
Covariance: 77500.0
Correlation: 0.9912858222289925
```

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 scores (1–10 scale), we should use:

- **Mean & Median** → measures of central tendency (average satisfaction).
- **Standard Deviation & Variance** → how spread out the scores are.
- **Minimum, Maximum, Range** → overall limits of satisfaction.
- **Histogram** → visualize the frequency of scores to see if data is skewed or centered.
- **Boxplot** (optional) → to check for outliers.

This combination gives both **numerical summaries** and **visual understanding** of customer satisfaction.

Python Code:

```
# Survey data
survey_scores = [7, 8, 5, 9, 6, 7, 8, 9, 10, 4, 7, 6, 9, 8, 7]

# Summary statistics
mean_score = np.mean(survey_scores)
std_score = np.std(survey_scores, ddof=1)   # sample std deviation
median_score = np.median(survey_scores)

print("Mean:", mean_score)
print("Median:", median_score)
print("Standard Deviation:", std_score)

# Histogram
plt.figure(figsize=(5,4))
plt.hist(survey_scores, bins=6, color="skyblue", edgecolor="black")
plt.title("Histogram of Customer Satisfaction Scores")
plt.xlabel("Score")
plt.ylabel("Frequency")
plt.legend()
plt.show()
```

Output:

```
Mean: 7.4
Median: 7.0
Standard Deviation: 1.55
```