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 :
The part of statistics with the description and summarization of data is called descriptive statistics.

Example:

A teacher calculates the average test score for their class to understand how the class performed on the exam. The mean score, median score, and the distribution of scores are descriptive statistics

Inferential statistics:
In contrast, use a sample to make generalizations, predictions, and conclusions about a larger population, employing methods such as hypothesis testing, confidence intervals, and regression analysis.

Example: Using the average test score from a sample of 100 students across the state, an inferential statistic could be used to estimate the average test score of all 8th-grade students in the entire state. This conclusion is made with a certain level of probability, acknowledging it's based on a sample.

- Hypothesis testing: Performing tests like a t-test to determine if there is a significant difference between two groups or if a hypothesis about a population is supported by the sample.
- Confidence Intervals: Estimating a population parameter (like the population mean) within a specific range, with a certain level of confidence.

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

Answer: Sampling is selecting a subset of individuals from a larger population to make inferences about the entire population, and the main difference is that random sampling selects members from the entire population with an equal chance of selection, while stratified sampling first divides the population into subgroups (strata) and then randomly selects members from each subgroup.

Simple Random Sampling

Method:

Every member of the population has an equal chance of being selected for the sample.

How it works:

A sample is drawn directly from the entire population without any prior divisions.

Best for:
Situations where the population is homogeneous or when a general, unbiased representation is sufficient.

Stratified Sampling

- **Method**: The population is first divided into distinct subgroups, or strata, based on shared characteristics (e.g., age, gender, location).
-
- **How it works**: A random sample is then taken from each of these subgroups.
-
- **Best for**: Populations with diverse characteristics where ensuring accurate representation from each important subgroup is crucial.

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

Answer: Often in statistics, we tend to represent a set of data by a representative value which would approximately define the entire collection. This representative value is called the measure of central tendency, and the name suggests that it is a value around which the data is centred. These central tendencies are mean, median and mode.

i) By comparing the mean, median, and mode, you can gain insight into the shape of the data's distribution. In a perfectly symmetrical distribution, all three measures are equal, but in a skewed distribution (where data is pulled to one side), they will differ.

ii) These measures provide a basis for comparing different datasets. For example, you can compare the average test scores (mean) of two different classes to see which class performed better.

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

Answer: Skewness is a key statistical measure that shows how data is spread out in a dataset. It tells us if the data points are skewed to the left (negative skew) or to the right (positive skew) in relation to the mean. It is important because it helps us to understand the shape of the data distribution which is important for accurate data analysis and helps in identifying outliers and finding the best statistical methods to use for analysis. In this article, we will see skewness, different types of skewness and its core concepts.

Kurtosis is a statistical measure used to describe a characteristic of a dataset. It generally takes the form of a bell when normally distributed data is plotted on a graph. This is called the bell curve. The plotted data that are farthest from the mean of the data usually form the tails on each side of the curve. Kurtosis indicates how much data resides in the tails.

Kurtosis and skewness are both statistical measures used to describe the shape of a probability distribution but they focus on different aspects. Kurtosis measures the tailedness of a distribution. Skewness measures the asymmetry of a distribution.

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]

Answer: import statistics

```
from collections import Counter

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

mean_value = sum(numbers) / len(numbers)

median_value = statistics.median(numbers)

counts = Counter(numbers)
max_count = max(counts.values())
mode_values = [key for key, value in counts.items() if value == max_count]

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

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]

Answer:

```
import numpy as np

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

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

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}")
```

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]

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]

# Create a boxplot
plt.figure(figsize=(8, 6))
plt.boxplot(data)
plt.title('Boxplot of Data')
plt.ylabel('Values')
plt.grid(True)
plt.show()

# Calculate quartiles and IQR to identify outliers
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(f"First Quartile (Q1): {Q1}")
print(f"Third Quartile (Q3): {Q3}")
print(f"Interquartile Range (IQR): {IQR}")
print(f"Lower Bound for Outliers: {lower_bound}")
print(f"Upper Bound for Outliers: {upper_bound}")
print(f"Identified Outliers: {outliers}")
```

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]

Answer: As a data analyst, I would use both covariance and the correlation coefficient to determine the nature and strength of the relationship between advertising spend and daily sales.

1. Covariance

What it measures: Covariance measures the *direction* of the linear relationship between two variables. It indicates whether

higher spending generally coincides with higher sales (a positive relationship) or if higher spending coincides with lower sales (a negative relationship).

Limitation: Covariance values are not standardized, meaning the magnitude of the number (e.g., 275 or 275,000) depends entirely on the units of measurement (dollars, rupees, etc.). This makes it impossible to compare the strength of the relationship with other pairs of variables or across different studies.

2. Correlation Coefficient (Pearson's r)

What it measures: The correlation coefficient standardizes the covariance measure, allowing us to quantify both the *direction* and the *strength* of the linear relationship. The result always falls within a fixed range of -1 to +1.

Conclusion:

By using both metrics, we first confirm the *direction* of the relationship (positive covariance), and then we confirm the *strength* of that relationship (high positive correlation). This provides clear, actionable insights for the marketing team.

Python code:

```
import matplotlib.pyplot as plt
```

```
import pandas as pd

import seaborn as sns


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


# Convert the list to a pandas Series for easier calculation

data_series = pd.Series(data)


# Calculate quartiles and IQR

Q1 = data_series.quantile(0.25)

Q3 = data_series.quantile(0.75)

IQR = Q3 - Q1


# Define the lower and upper bounds for outliers

lower_bound = Q1 - 1.5 * IQR

upper_bound = Q3 + 1.5 * IQR
```

```
# Identify outliers

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


# Plot the boxplot

plt.figure(figsize=(8, 6))

sns.boxplot(y=data)

plt.title('Boxplot of the Data')

plt.ylabel('Values')

plt.show()


print(f"Data: {data}")

print(f"First Quartile (Q1): {Q1}")

print(f"Third Quartile (Q3): {Q3}")

print(f"Interquartile Range (IQR): {IQR}")

print(f"Lower Bound (Q1 - 1.5*IQR): {lower_bound}")
```

```
print(f"Upper Bound (Q3 + 1.5*IQR): {upper_bound}")

print(f"Identified Outliers: {outliers}")
```

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]

Answer: To understand the distribution of customer satisfaction survey data effectively, I would use a combination of key summary statistics and visualization tools. These help in summarizing the central tendency, spread, and shape of the data.

Explanation of Summary Statistics and Visualizations

Type	Statistic / Visualization	Purpose

Central Tendency	Mean	Provides the average satisfaction score. Useful for a quick summary of typical feedback.
Central Tendency	Median	The middle value when the data is ordered. Comparing the mean and median helps identify if the data is skewed by extreme scores.
Variability	Standard Deviation	Measures the spread or dispersion of scores around the mean. A low standard deviation means most customers gave similar ratings; a high standard deviation indicates a wide variety of opinions.
Visualization	Histogram	A histogram is ideal for visualizing the *entire distribution*. It shows the frequency of each score (e.g., how many people rated a 7 versus a 9). This helps identify: the most common scores (mode), the overall shape of the distribution (skewness), and any unusual patterns or gaps in the data.

By using these tools, we can gain a comprehensive understanding of customer sentiment before proceeding with the new product launch.

Python Code:

```
import matplotlib.pyplot as plt
import pandas as pd

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

scores_series = pd.Series(survey_scores)
mean_score = scores_series.mean()
median_score = scores_series.median()
std_dev_score = scores_series.std()
mode_scores = scores_series.mode().tolist()

print(f"Mean Score: {mean_score:.2f}")
print(f"Median Score: {median_score}")
print(f"Standard Deviation: {std_dev_score:.2f}")
print(f"Mode(s): {mode_scores}\n")

plt.figure(figsize=(10, 6))

plt.hist(survey_scores, bins=range(4, 12), rwidth=0.8, align='left',
color='skyblue', edgecolor='black')
```

```
plt.title('Distribution of Customer Survey Scores (Scale 1-10)',
fontsize=16)
plt.xlabel('Survey Score', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

plt.xticks(range(4, 11))

plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```