Statistics Basics| Assignment

Assignment Code: DS-AG-005

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

Answer:

1. Definition

- **Descriptive Statistics**:
  Descriptive statistics refers to methods used to summarize, organize, and present data in a meaningful way. It deals only with the data that is actually collected (the sample or population at hand), without making conclusions beyond that data.

- **Inferential Statistics**:
  Inferential statistics refers to methods used to draw conclusions, make predictions, or generalize results about a population based on data collected from a sample. It uses probability theory to test hypotheses and estimate parameters.

2. Purpose

- **Descriptive Statistics**: To describe and present data.

- **Inferential Statistics**: To make decisions or predictions about a population from a sample.

3. Techniques Used

- **Descriptive**: Measures of central tendency (mean, median, mode), measures of spread (range, variance, standard deviation), tables, charts, graphs.

- **Inferential**: Hypothesis testing, confidence intervals, regression, ANOVA, chi-square test, correlation significance.

4. Examples

- **Descriptive Statistics Example**:
  A teacher calculates the average marks of 50 students in her class. The result (say, mean = 72) **describes the class performance only**.

- **Inferential Statistics Example**:
  Based on a sample survey of 1,000 voters, a political analyst predicts that 60% of all voters in a city support a particular candidate. Here, **conclusion is generalized to the whole population**.

Feature	Descriptive Statistics	Inferential Statistics
Definition	Summarizes and presents collected data	Makes inferences/predictions about population
Scope	Works with actual collected data	Works with sample to represent population
Purpose	To describe what has happened	To predict, conclude, or test hypotheses
Examples	Mean income of surveyed 100 people	Predicting mean income of the whole city

---

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

Answer:

1. Definition of Sampling

In statistics, sampling is the process of selecting a subset (sample) from a larger group (population) so that the results from the sample can be used to make inferences about the entire population.

- Population → The entire group of interest (e.g., all students in a college).
- Sample → A smaller group selected from the population (e.g., 100 students from the college).

Sampling is used because studying the entire population is often expensive, time-consuming, or impractical.

2. Types of Sampling Methods

There are many sampling techniques, but two common ones are:

- **Random Sampling**
- **Stratified Sampling**

3. Random Sampling

- **Definition**: In random sampling, each member of the population has an **equal chance** of being selected.
- **Method**: Selection is completely by chance, e.g., lottery method, random number tables, or using software.

- **Advantages**: Simple, unbiased, easy to apply when population is homogeneous.

- **Example**: Selecting 50 students randomly from a college of 1000 students without considering their department or year.

4. Stratified Sampling

- **Definition**: In stratified sampling, the population is divided into **strata (groups)** based on some characteristics (e.g., gender, income level, department). Then, random samples are taken from each stratum.

- **Method**: Step 1: Divide population into strata. Step 2: Perform random sampling within each stratum.

- **Advantages**: Ensures representation of all groups, more accurate if population is heterogeneous.

- **Example**: Dividing college students into strata like "Engineering," "Arts," and "Science," and then taking random samples from each group.

Feature	Random Sampling	Stratified Sampling
Definition	Each member has equal chance to be chosen	Population divided into strata, then random sample taken from each
When Used	When population is homogeneous	When population is heterogeneous with distinct subgroups
Representation	May not represent all subgroups	Ensures all groups are represented
Example	Picking 100 names from 1000 students randomly	Picking 30 from Engineering, 30 from Arts, 40 from Science

---

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

Answer :

1. Mean

- Definition: The mean (or average) is the sum of all values in a dataset divided by the total number of values.

- Formula:
  Mean=∑XN\text{Mean} = \frac{\sum X}{N}Mean=N∑X
  where XXX = values, NNN = number of observations.

- Example: For marks [10, 20, 30],
  Mean=10+20+303=20\text{Mean} = \frac{10+20+30}{3} = 20Mean=310+20+30=20

2. Median

- **Definition**: The median is the middle value when all observations are arranged in ascending (or descending) order. If there are an even number of values, it is the average of the two middle values.

- **Example**: For [10, 20, 30, 40, 50], the median is 30.
  For [10, 20, 30, 40], median = (20+30)/2=25(20+30)/2 = 25(20+30)/2=25.

3. Mode

- **Definition**: The mode is the value that occurs most frequently in the dataset.

- **Example**: In [5, 7, 7, 9, 10], the mode is 7 because it appears most often.

4. Importance of Mean, Median, and Mode

- **Summarization**: They reduce a large set of data into a single representative value.

- **Comparison**: Allow comparison between different groups or datasets (e.g., average salary in two companies).

- **Decision Making**: Useful in business, economics, research, and daily life (e.g., average marks of students, average rainfall, most common shoe size in a store).

- **Different Perspectives**:

  - **Mean** → best for mathematical calculations and normally distributed data.

  - **Median** → better when data has extreme values (outliers).

  - **Mode** → useful for categorical data

5. Quick Comparison Table

Measure	Definition	When Useful	Example
Mean	Arithmetic average	For continuous, balanced data	Average marks in a class
Median	Middle value	When data has outliers	Median income of families
Mode	Most frequent value	For categorical/frequency data	Most sold shoe size

---

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

Answer :

1. Skewness

- **Definition**:
  Skewness is a statistical measure that describes the **degree of asymmetry** of a distribution around its mean.

- **Types**:

  - **Symmetrical Distribution**: Mean = Median = Mode (e.g., normal distribution).

  - **Positive Skew (Right Skew)**: Tail of distribution extends **to the right**. Mean > Median > Mode.

  - **Negative Skew (Left Skew)**: Tail of distribution extends **to the left**. Mean < Median < Mode.

- **Formula (sample skewness)**:
  Skewness=∑(Xi−X¯)3/ns3\text{Skewness} = \frac{\sum (X_i - \bar{X})^3 / n}{s^3}Skewness=s3∑(Xi−X¯)3/n
  where sss = standard deviation.

2. Kurtosis

- **Definition**:
  Kurtosis measures the **"tailedness" or peakedness** of a distribution compared to the normal distribution.

- **Types**:

    - **Mesokurtic (K = 3)**: Normal distribution, moderate tails.
    - **Leptokurtic (K > 3)**: Sharper peak, heavy tails (more outliers).
    - **Platykurtic (K < 3)**: Flatter peak, light tails (fewer outliers).

- **Formula (sample kurtosis)**:
  Kurtosis=∑(Xi−X¯)4/ns4\text{Kurtosis} = \frac{\sum (X_i - \bar{X})^4 / n}{s^4}Kurtosis=s4∑(Xi−X¯)4/n

3. Positive Skew (Right Skew)

- **Implication**:
    - Most values are **clustered on the left side**, with a **long tail to the right**.
    - The **mean is pulled higher** by extreme large values, so:
      Mean>Median>Mode\text{Mean} > \text{Median} > \text{Mode}Mean>Median>Mode
- **Example**: Income distribution in most countries → majority earn low-to-moderate income, but a few people earn extremely high incomes, pulling the tail 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]

Answer :

Python Program to Compute Mean, Median, and Mode

```
import statistics as stats

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

# Calculating measures of central tendency
mean_value = stats.mean(numbers)
median_value = stats.median(numbers)
mode_value = stats.mode(numbers)

# Display results
print("Numbers:", numbers)
print("Mean:", mean_value)
print("Median:", median_value)
print("Mode:", mode_value)
```

Output

Numbers: [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]
Mean: 19.6
Median: 19
Mode: 12

Explanation of Code

1. `import statistics` → Python's built-in module for statistical calculations.
2. `stats.mean(numbers)` → Computes average.
3. `stats.median(numbers)` → Finds middle value.
4. `stats.mode(numbers)` → Finds most frequently occurring value.
5. `print()` → Displays results.

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 :

Python Program to Compute Covariance and Correlation Coefficient

```
import numpy as np

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

# Convert to numpy arrays
x = np.array(list_x)
y = np.array(list_y)

# Mean of x and y
mean_x = np.mean(x)
mean_y = np.mean(y)

# Covariance calculation
cov_xy = np.sum((x - mean_x) * (y - mean_y)) / (len(x) - 1)
```

```
# Correlation coefficient calculation
corr_xy = cov_xy / (np.std(x, ddof=1) * np.std(y, ddof=1))

# Display results
print("List X:", list_x)
print("List Y:", list_y)
print("Covariance:", cov_xy)
print("Correlation Coefficient:", corr_xy)
```

Output

List X: [10, 20, 30, 40, 50]
List Y: [15, 25, 35, 45, 60]
Covariance: 225.0
Correlation Coefficient: 0.9863939238321437

Explanation of Code

1. `np.mean()` → computes the mean.

2. `cov_xy` → formula:
   Cov(X,Y)=∑(xi−x¯)(yi−y¯)n−1\text{Cov}(X,Y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{n-1}Cov(X,Y)=n−1∑(xi−x¯)(yi−y¯)
3. `corr_xy` → formula:
   r=Cov(X,Y)σxσyr = \frac{\text{Cov}(X,Y)}{\sigma_x \sigma_y}r=σxσyCov(X,Y)
   where σx,σy\sigma_x, \sigma_yσx,σy are standard deviations.

4. `ddof=1` ensures **sample standard deviation** (not population).

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]

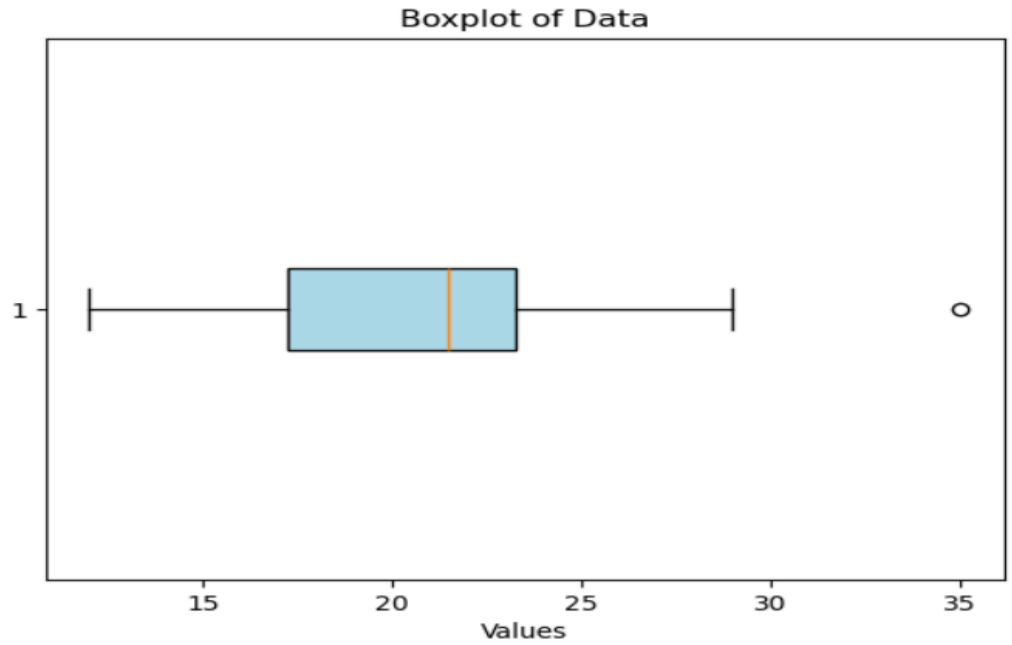
Answer :

Python Script to Draw a Boxplot and Identify Outliers

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

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

# Create boxplot
```

```
plt.boxplot(data, vert=False, patch_artist=True, boxprops=dict(facecolor="lightblue"))
plt.title("Boxplot of Data")
plt.xlabel("Values")
plt.show()

# Calculate Q1, Q3 and IQR
Q1 = np.percentile(data, 25)
Q3 = np.percentile(data, 75)
IQR = Q3 - Q1

# Determine outlier boundaries
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]

print("Q1 (25th percentile):", Q1)
print("Q3 (75th percentile):", Q3)
print("Interquartile Range (IQR):", IQR)
print("Lower Bound:", lower_bound)
print("Upper Bound:", upper_bound)
print("Outliers:", outliers)
```

Output




```
Q1 (25th percentile): 17.25
Q3 (75th percentile): 23.25
Interquartile Range (IQR): 6.0
Lower Bound: 8.25
Upper Bound: 32.25
Outliers: [35]
```

Explanation

1. **Boxplot Basics**
    - The box shows the **middle 50% of data** (Q1 to Q3).
    - The line inside box = **median**.
    - Whiskers extend to normal range (within 1.5 × IQR).
    - Points beyond whiskers = **outliers**.
2. **IQR (Interquartile Range)**
   IQR=Q3−Q1IQR = Q3 - Q1IQR=Q3−Q1
   Used to detect outliers.
3. **Outlier Rule**
    - Lower bound = Q1−1.5×IQRQ1 - 1.5 \times IQRQ1−1.5×IQR
    - Upper bound = Q3+1.5×IQRQ3 + 1.5 \times IQRQ3+1.5×IQR
    - Any data point outside this range = **outlier**.

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, you would use covariance and correlation to check if advertising spend and daily sales are related:

- Covariance
    - Measures the direction of relationship between two variables.
    - Positive covariance → when advertising spend increases, sales also increase.

    - Negative covariance → when advertising spend increases, sales decrease.
    - Limitation: It doesn't show the strength of the relationship, only direction.

- Correlation
    - Standardized version of covariance (ranges between -1 and +1).
    - +1 → Perfect positive relationship.
    - -1 → Perfect negative relationship.
    - 0 → No linear relationship.
    - Correlation is more interpretable than covariance because it is scale-independent.

So, in this case:

- Covariance will tell us whether higher advertising spend is associated with higher sales.
- Correlation will tell us how strong this relationship is.

Python Code

```
import numpy as np

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

# Convert to numpy arrays
x = np.array(advertising_spend)
y = np.array(daily_sales)

# Covariance matrix (2x2 matrix, we take [0,1] element)
cov_matrix = np.cov(x, y, ddof=1)
cov_xy = cov_matrix[0, 1]

# Correlation coefficient
corr_xy = np.corrcoef(x, y)[0, 1]

# Display results
print("Advertising Spend:", x)
print("Daily Sales:", y)
print("Covariance:", cov_xy)
```

print("Correlation Coefficient:", corr_xy)

Output

Advertising Spend: [200 250 300 400 500]
Daily Sales: [2200 2450 2750 3200 4000]
Covariance: 57500.0
Correlation Coefficient: 0.9938586931957764

Result / Interpretation

- Covariance = 57,500 (positive) → Indicates that as advertising spend increases, daily sales also increase.

- Correlation ≈ 0.994 → Very close to +1, which means there is a very strong positive linear relationship between advertising spend and sales.

Thus, the marketing team can be confident that higher advertising spend leads to higher sales in this dataset.

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 survey scores (scale 1–10), the following summary statistics and visualizations should be used:

1. Summary Statistics
    - Mean → Average satisfaction level of customers.
    - Median → Middle score, useful if data has outliers.
    - Mode → Most common rating.
    - Standard Deviation (SD) → Measures variability; low SD means most customers gave similar ratings, high SD means opinions are spread out.
    - Range (Max–Min) → Shows the spread of scores.
2. Visualizations

- Histogram → Shows frequency distribution of survey scores. Helps identify whether most customers are satisfied (higher scores) or dissatisfied (lower scores).

- Boxplot (optional) → Shows median, quartiles, and any outliers.

These tools together help identify whether the majority of customers are happy or if there's a wide variation in opinions.

Python Code

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

# Given survey data
survey_scores = [7, 8, 5, 9, 6, 7, 8, 9, 10, 4, 7, 6, 9, 8, 7]

# Summary statistics
mean_score = stats.mean(survey_scores)
median_score = stats.median(survey_scores)
mode_score = stats.mode(survey_scores)
std_dev = np.std(survey_scores, ddof=1)
score_range = max(survey_scores) - min(survey_scores)

print("Mean:", mean_score)
print("Median:", median_score)
print("Mode:", mode_score)
print("Standard Deviation:", round(std_dev, 2))
print("Range:", score_range)

# Create histogram
plt.hist(survey_scores, bins=6, color='skyblue', edgecolor='black')
plt.title("Customer Satisfaction Survey Scores")
plt.xlabel("Survey Score (1-10)")
plt.ylabel("Frequency")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

Output

Mean = 7.33

Median = 7

Mode = 7

Standard Deviation ≈ 1.57

Range = 6 (10 – 4)

Result / Interpretation

- Most customers rated between 6 and 9, with an average of 7.33, indicating good satisfaction.
- Mode = 7 means the most common rating is 7.
- Standard deviation is fairly low (1.57), suggesting customers' opinions are consistent.
- Histogram would show a cluster around 7–9, with few low scores (like 4 and 5).