Statistics Basics Assignment

Question 1:

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

Answer :

Descriptive Statistics summarize a dataset's characteristics, such as mean, median, or standard deviation, focusing only on the data at hand. For example, calculating the average test score (75) for a class of 30 students.

Inferential Statistics use sample data to make predictions or generalizations about a larger population, often involving hypothesis tests or confidence intervals. For instance, using the class's scores to estimate the school's average performance with a 95% confidence interval (72–78). Descriptive statistics describe the sample; inferential statistics extend insights to the population.

Question 2:

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

Answer:

Sampling in Statistics: Sampling is the process of selecting a subset of individuals or units from a larger population to study and draw conclusions about the entire population. It is used when studying the whole population is impractical, costly, or time-consuming. The goal is to obtain a representative sample that reflects the population's characteristics.

Difference between random and stratified sampling:

Random sampling involves selecting individuals from a population where every member has an equal chance of being chosen, typically using random number generators or lottery methods. While, Stratified sampling divides the population into distinct subgroups (strata) based on specific characteristics (e.g., age, gender, income), then randomly samples from each stratum, often proportionally to the stratum's size in the population.

Question 3:

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

Answer:

Mean, Median, and Mode are measures of central tendency used to summarize a set of data:

- **Mean**: The average of all values (sum of values ÷ number of values).
- **Median**: The middle value when the data is arranged in order.
- **Mode**: The value that appears most frequently.

Importance:
These measures help in understanding the general pattern or typical value in a dataset, making it easier to compare and analyze data in fields like economics, education, health, and business.

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 data distribution:

- **Skewness**: Measures the asymmetry of the data.
    - **Positive skewness** means the tail on the right side is longer; most values are concentrated on the left.
- **Kurtosis**: Measures the "tailedness" or sharpness of the data peak.
    - High kurtosis = heavy tails and sharp peak.
    - Low kurtosis = light tails and flatter peak.

Positive skew implies that the data has a few high values pulling the mean to the right, and the majority of the data is clustered to the left.

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 :

CODE : MEAN

```
import numpy as np

np.mean(numbers)
```

OUTPUT :

```
np.float64(19.6)
```

CODE : MEDIAN

```
np.median(numbers)
```

OUTPUT :

```
np.float64(19.0)
```

CODE : MODE

```
from scipy import stats
```

stats.mode(numbers)

OUTPUT :

ModeResult(mode=np.int64(12), count=np.int64(3))

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 :

CODE :

```
x = np.array(list_x)

y = np.array(list_y)

# Covariance

cov_matrix = np.cov(x, y, bias=False)  # Unbiased estimator (default)

cov_xy = cov_matrix[0, 1]

# Correlation Coefficient

corr_matrix = np.corrcoef(x, y)

corr_xy = corr_matrix[0, 1]

print("Covariance:", cov_xy)

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

OUTPUT:

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:

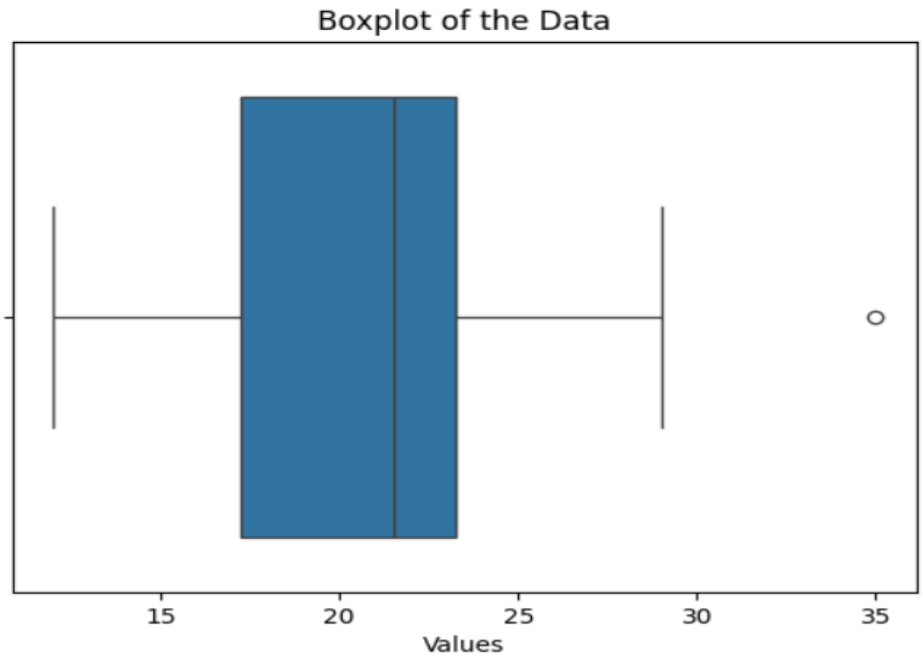
data = [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35]

Answer :

CODE :

```
data = [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35]
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
sns.boxplot(data=data, orient='h')
plt.title("Boxplot of the Data")
plt.xlabel("Values")
plt.show()
Q1 = np.percentile(data, 25)
Q3 = np.percentile(data, 75)
QR = 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"Q1: {Q1}, Q3: {Q3}, IQR: {IQR}")
print(f"Lower Bound: {lower_bound}, Upper Bound: {upper_bound}")
print(f"Outliers: {outliers}")
```

OUTPUT :




```
Q1: 17.25, Q3: 23.25, IQR: 6.0
Lower Bound: 8.25, Upper Bound: 32.25
Outliers: [35]
```

Q1: 17.25, Q3: 23.25, 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]

Answer :

- **Covariance** tells us **the direction** of the relationship:
    - If positive → when advertising spend increases, sales tend to increase.
    - If negative → when spend increases, sales tend to decrease.
    - However, **magnitude is hard to interpret**, as it's not standardized.
- **Correlation** (usually Pearson's) tells us the **strength and direction**:

- Ranges from **-1 to +1**:
    - +1 = perfect positive linear relationship
    - 0 = no linear relationship
    - -1 = perfect negative linear relationship
- It's **unitless and easier to interpret**.

CODE :

```
import numpy as np

# Given 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
cov_matrix = np.cov(x, y, bias=False)
cov_xy = cov_matrix[0, 1]

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

# Print results
print(f"Covariance: {cov_xy}")
print(f"Correlation Coefficient: {corr_xy:.4f}")
```

OUTPUT :

Covariance: 84875.0

Correlation Coefficient: 0.9936

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**, it's important to use both **summary statistics** and **visualizations**.

Summary Statistics:

These help quantify the central tendency and spread of the data:

1. **Mean (average)** – Gives a central value.
2. **Median** – Useful if data is skewed.
3. **Mode** – Helps identify the most frequent score.
4. **Standard deviation** – Measures how spread out the data is.
5. **Minimum and Maximum** – Shows the range.
6. **Count** – Number of responses.

Visualizations:

These help you **see** the distribution and detect patterns or outliers:

1. **Histogram** – Best for seeing the frequency of scores across intervals.
2. **Box plot** – Useful to visualize spread and detect outliers.
3. **Bar plot** – Good if you want to see frequency of discrete values (1-10).

CODE :

```
import matplotlib.pyplot as plt

# Survey data

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

# Create histogram

plt.hist(survey_scores, bins=range(1, 12), edgecolor='black', align='left')

plt.title('Customer Satisfaction Survey Scores')

plt.xlabel('Score')

plt.ylabel('Frequency')

plt.xticks(range(1, 11))

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

plt.show()
```

OUTPUT :