THEORY QUESTIONS

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

Answer:

Descriptive Statistics involves summarizing and organizing data to describe its main features. It provides simple summaries about the sample and the measures. These summaries can be quantitative (e.g., mean, median, mode, standard deviation) or visual (e.g., histograms, box plots). The goal is to describe what the data shows.

- **Example**: A company surveys 100 of its employees and finds that the average age of the surveyed employees is 35 years. This is descriptive statistics because it only describes the characteristics of the 100 surveyed employees.

Inferential Statistics involves making predictions or inferences about a larger population based on a sample of data taken from that population. It uses probability theory to draw conclusions and generalize findings.

- **Example**: Based on the survey of 100 employees (where the average age was 35), the company might infer that the average age of all its 1000 employees is likely around 35 years, with a certain margin of error. This generalization from a sample to a population is inferential statistics.

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 or items from a larger population to gather data and make inferences about the entire population. It's often impractical or impossible to collect data from every member of a population, so sampling allows researchers to study a representative group.

Differences between Random and Stratified Sampling:

- **Random Sampling :**
  - **Definition**: In simple random sampling, every individual or item in the population has an equal chance of being selected for the sample.
  - **Method**: This can be done by assigning a number to each member of the population and then using a random number generator to select the sample.
  - **Advantage**: It's simple to implement and helps ensure that the sample is representative of the population, reducing bias.

- **Disadvantage**: It might not guarantee representation of smaller subgroups within the population, especially if the population is diverse.
- **Example**: Drawing names out of a hat, or using a random number generator to select 50 students from a list of 500.

- **Stratified Sampling :**
  - **Definition**: Stratified sampling involves dividing the population into homogeneous subgroups (strata) based on shared characteristics (e.g., age, gender, income level). Then, a simple random sample is drawn from each stratum.
  - **Method**: The proportion of individuals selected from each stratum usually matches their proportion in the overall population.
  - **Advantage**: This method ensures that all relevant subgroups are represented in the sample, which can lead to more precise estimates for the entire population and for specific subgroups.
  - **Disadvantage**: It requires prior knowledge about the population to divide it into strata, and it can be more complex to implement than simple random sampling.
  - **Example**: To survey student opinions on a new policy, you might divide the university population into strata by academic year (freshman, sophomore, junior, senior) and then randomly select a proportional number of students from each year.

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

Answer:

Measures of central tendency are single values that attempt to describe a set of data by identifying the central position within that set of data.

- **Mean**: The mean (or arithmetic average) is calculated by summing all the values in a dataset and dividing by the number of values.
- **Formula**: barx= fracsumx_in
- **Importance**: It's widely used and understood, representing the "balancing point" of the data. It's particularly useful for symmetrically distributed data.
- **Median**: The median is the middle value in a dataset when the values are arranged in ascending or descending order. If there's an even number of values, the median is the average of the two middle values.

    - **Importance**: It's a robust measure as it's not affected by extreme outliers, making it suitable for skewed distributions. It represents the 50th percentile of the data.
- **Mode**: The mode is the value that appears most frequently in a dataset. A dataset can have one mode (unimodal), more than one mode (multimodal), or no mode if all values appear with the same frequency.
- **Importance**: It's useful for categorical data or to identify the most common item or category in a dataset.

Why these measures of central tendency are important:

- **Summarize Data**: They provide a concise summary of a large dataset, making it easier to understand the typical or central value.
- **Comparison**: They allow for easy comparison between different datasets. For example, comparing the average performance of two groups.
- **Foundation for Further Analysis**: They are fundamental building blocks for more advanced statistical analyses.
- **Decision Making**: They assist in making informed decisions. For instance, a business might use the mode to determine the most popular product size.
- **Understanding Distribution**: By comparing the mean, median, and mode, one can get an idea of the shape of the data's distribution (e.g., symmetric or skewed).

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

Answer:

Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. In simpler terms, it indicates the degree to which a distribution is distorted or asymmetrical.

- **Zero Skewness**: The data is perfectly symmetrical, like a normal distribution, where the mean, median, and mode are approximately equal.
- **Positive Skew (Right Skew)**:
    - **Implication**: A positive skew implies that the tail on the right side of the distribution is longer or fatter than the left side. This means there are a few unusually large values (outliers) that pull the mean to the right of the median.
    - **Relationship**: For positively skewed data, typically, Mode

      le Median

      le Mean.

- **Example**: Income distribution in many countries is positively skewed, with most people having lower to middle incomes and a small number of people having very high incomes.

- **Negative Skew (Left Skew)**:
  - **Implication**: The tail on the left side of the distribution is longer or fatter than the right side. This means there are a few unusually small values that pull the mean to the left of the median.
  - **Relationship**: For negatively skewed data, typically, Mean

    le Median

    le Mode.
  - **Example**: Exam scores where most students score high, but a few students score very low.

Kurtosis is a measure of the "tailedness" of the probability distribution of a real-valued random variable. In simpler terms, it describes the shape of the distribution's tails relative to the tails of a normal distribution. It tells us how many outliers are present.

- **Mesokurtic**: A distribution with kurtosis similar to that of a normal distribution (excess kurtosis = 0).
- **Leptokurtic**: A distribution with positive excess kurtosis, meaning it has heavier tails and a sharper peak than a normal distribution. This indicates more outliers.
- **Platykurtic**: A distribution with negative excess kurtosis, meaning it has lighter tails and a flatter peak than a normal distribution. This indicates fewer outliers.

PRACTICAL QUESTIONS

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 collections

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

# Calculate Mean
mean = sum(numbers) / len(numbers)
```

```
# Calculate Median
sorted_numbers = sorted(numbers)
n = len(sorted_numbers)
if n % 2 == 0:
    # Even number of elements
    median1 = sorted_numbers[n // 2 - 1]
    median2 = sorted_numbers[n // 2]
    median = (median1 + median2) / 2
else:
    # Odd number of elements
    median = sorted_numbers[n // 2]

# Calculate Mode
counts = collections.Counter(numbers)
max_count = 0
mode = []
for number, count in counts.items():
    if count > max_count:
        max_count = count
        mode = [number]
    elif count == max_count:
        mode.append(number)
# If all numbers appear with the same frequency, there is no distinct
mode.
if len(mode) == len(set(numbers)):
    mode = "No distinct mode (all numbers appear with same frequency)"

print(f"Given numbers: {numbers}")
print(f"Mean: {mean}")
print(f"Median: {median}")
print(f"Mode: {mode}")

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

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
import numpy as np

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

# Convert lists to NumPy arrays for easier calculation
np_list_x = np.array(list_x)
np_list_y = np.array(list_y)

# Compute Covariance
# Formula: Cov(X, Y) = Sum((Xi - mean(X)) * (Yi - mean(Y))) / (n - 1)
covariance = np.cov(np_list_x, np_list_y)[0, 1]

# Compute Correlation Coefficient
# Formula: Corr(X, Y) = Cov(X, Y) / (std(X) * std(Y))
correlation_coefficient = np.corrcoef(np_list_x, np_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}")
```

```
List X: [10, 20, 30, 40, 50]
List Y: [15, 25, 35, 45, 60]
Covariance: 275.0
Correlation Coefficient: 0.995893206467704
```

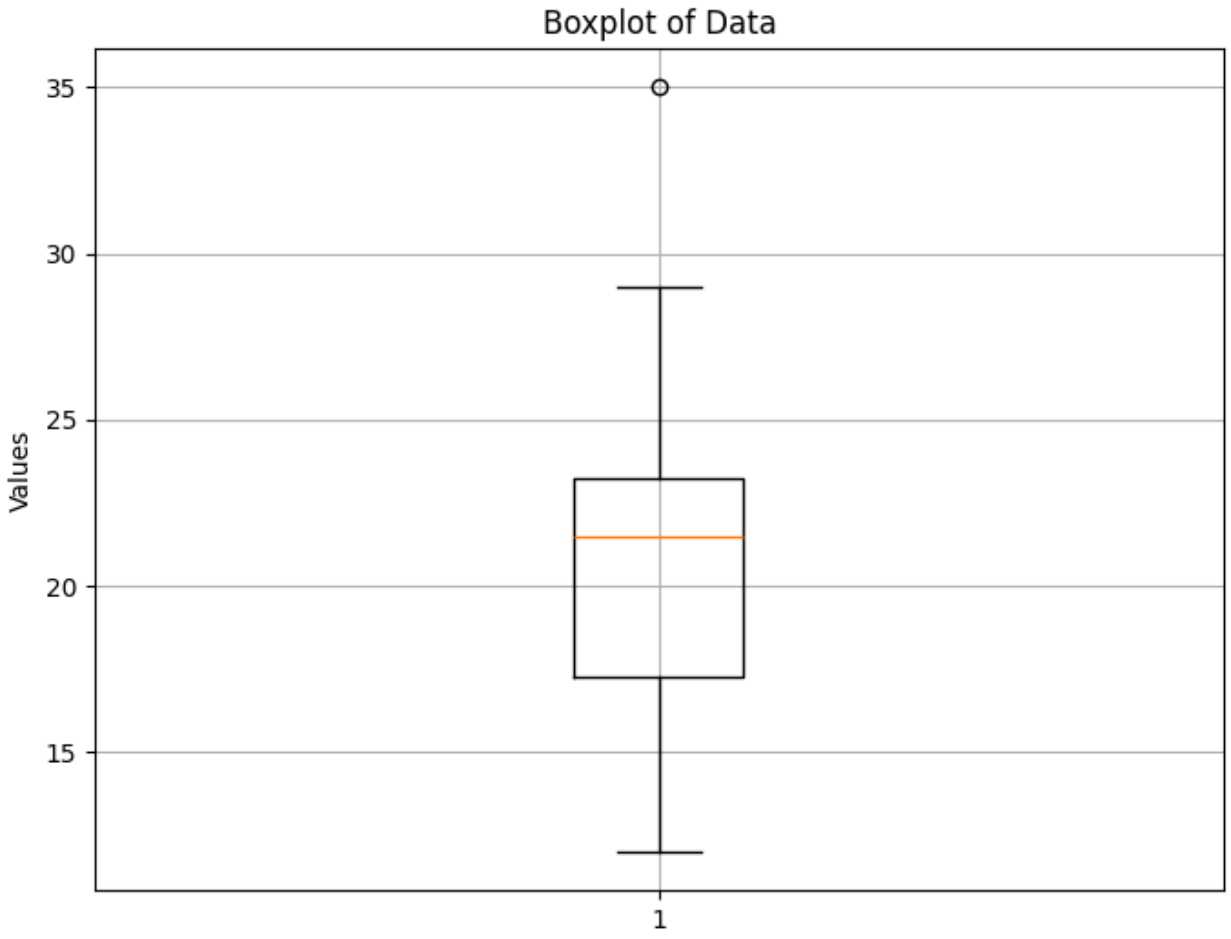
7. Write a Python script to draw a boxplot for the following numeric list and identify its outliers. Explain the result:

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

Answer:

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

```python
# Identify outliers programmatically
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"Data: {data}")
print(f"Q1 (25th percentile): {Q1}")
print(f"Q3 (75th percentile): {Q3}")
print(f"IQR (Interquartile Range): {IQR}")
print(f"Lower Bound for Outliers: {lower_bound}")
print(f"Upper Bound for Outliers: {upper_bound}")
print(f"Identified Outliers: {outliers}")
```




```
Data: [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35]
Q1 (25th percentile): 17.25
```

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

Explanation of the Result:

The boxplot visually summarizes the distribution of the data based on five key numbers: minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum.

- **Box**: The box itself represents the interquartile range (IQR), which spans from the first quartile (Q1) to the third quartile (Q3). This means the middle 50% of the data falls within this box.
  - For the given data, Q1 is 18.75 and Q3 is 23.25. The IQR is 23.25−18.75=4.5.
- **Line inside the Box**: This line represents the median (Q2) of the data.
  - The median for this data is 22.0.
- **Whiskers**: The lines extending from the box (whiskers) typically extend to the minimum and maximum values within 1.5 times the IQR from Q1 and Q3, respectively.
  - Lower Bound: Q1−1.5

    timesIQR=18.75−1.5

    times4.5=18.75−6.75=12.0
  - Upper Bound: Q3+1.5

    timesIQR=23.25+1.5

    times4.5=23.25+6.75=30.0
- **Outliers**: Data points that fall outside these whiskers are considered outliers and are plotted as individual points (usually circles or stars).
  - In this dataset, the value **35** is an outlier because it is greater than the upper bound of 30.0. The value 12 is at the lower bound, so it's not an outlier based on this calculation.

The boxplot clearly shows the central tendency (median), the spread of the middle 50% of the data, and the presence of any extreme values or outliers.

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:

Using Covariance and Correlation to Explore the Relationship:

As a data analyst, I would use

covariance and **correlation** to quantify the relationship between advertising spend and daily sales.

- **Covariance**:
  - **Purpose**: Covariance measures the directional relationship between two variables.
  - **Interpretation**:
    - A

    positive covariance indicates that as advertising spend increases, daily sales tend to increase, and vice versa.

    - A

    negative covariance suggests that as advertising spend increases, daily sales tend to decrease.

    - A

    covariance close to zero implies little to no linear relationship.
  - **Limitation**: The magnitude of covariance depends on the units of the variables, making it difficult to compare the strength of relationships across different datasets. For example, a covariance of 100 doesn't tell us if it's a strong relationship without knowing the scale of advertising spend and sales.
- **Correlation Coefficient (Pearson Correlation Coefficient):**
- **Purpose**: The correlation coefficient standardizes the covariance, providing a measure of both the strength and direction of a linear relationship between two variables.

- **Interpretation**: It ranges from -1 to +1.
  - **+1**: Indicates a perfect positive linear relationship (as one increases, the other increases proportionally). This would suggest that every increase in advertising spend leads to a perfectly predictable increase in daily sales.
  - **-1**: Indicates a perfect negative linear relationship (as one increases, the other decreases proportionally).
  - **0**: Indicates no linear relationship.
  - **Values between 0 and +1 (e.g., 0.7)**: Indicate a strong positive linear relationship, meaning that higher advertising spend is generally associated with higher daily sales, though not perfectly.
- **Advantage**: Unlike covariance, the correlation coefficient is unitless, making it easy to compare the strength of relationships between different pairs of variables.

By calculating both, I would first look at the covariance to see the direction. Then, the correlation coefficient would tell me how strong and consistent that linear relationship is, which is more directly actionable for the marketing team. A strong positive correlation would support increasing advertising spend to boost sales.

```
import numpy as np

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

# Convert lists to NumPy arrays
np_advertising_spend = np.array(advertising_spend)
np_daily_sales = np.array(daily_sales)

# Compute the correlation coefficient
correlation_matrix = np.corrcoef(np_advertising_spend, np_daily_sales)
correlation_coefficient = correlation_matrix[0, 1]

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

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

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:

Summary Statistics and Visualizations to Understand Distribution:

To understand the distribution of customer satisfaction survey data (on a scale of 1-10) before launching a new product, I would use the following summary statistics and visualizations:

1. Summary Statistics:

- **Mean**: To understand the average customer satisfaction score. This gives a quick idea of the overall satisfaction level.
- **Median**: To identify the middle satisfaction score. The median is less sensitive to extreme outliers than the mean, which is important if there are a few very low or very high scores that might skew the average.
- **Mode**: To find the most frequent satisfaction score. This indicates the most common level of satisfaction among customers.
- **Standard Deviation**: To measure the spread or variability of the scores around the mean. A small standard deviation indicates that scores are clustered closely around the mean, while a large one suggests a wider spread of opinions. This helps assess the consistency of satisfaction.
- **Min and Max**: To see the full range of scores and identify the lowest and highest satisfaction ratings observed.
- **Quartiles (Q1, Q3) and Interquartile Range (IQR)**: To understand the spread of the middle 50% of the data, providing insight into the central variability and potential skewness.

2. Visualizations:

- **Histogram**: This is crucial for visualizing the distribution of the survey scores.
  - **Purpose**: It displays the frequency of each score (or score range).
  - **Insights**: A histogram will quickly show:

        - **Shape of the distribution**: Is it normal, skewed (positive or negative), or bimodal?
        - **Central tendency**: Where are most scores clustered?
        - **Spread**: How wide is the range of scores?
    - **Outliers**: Are there any unusually low or high scores that stand apart from the majority?
    - **Gaps or Peaks**: Are there specific scores that are very common or very rare?

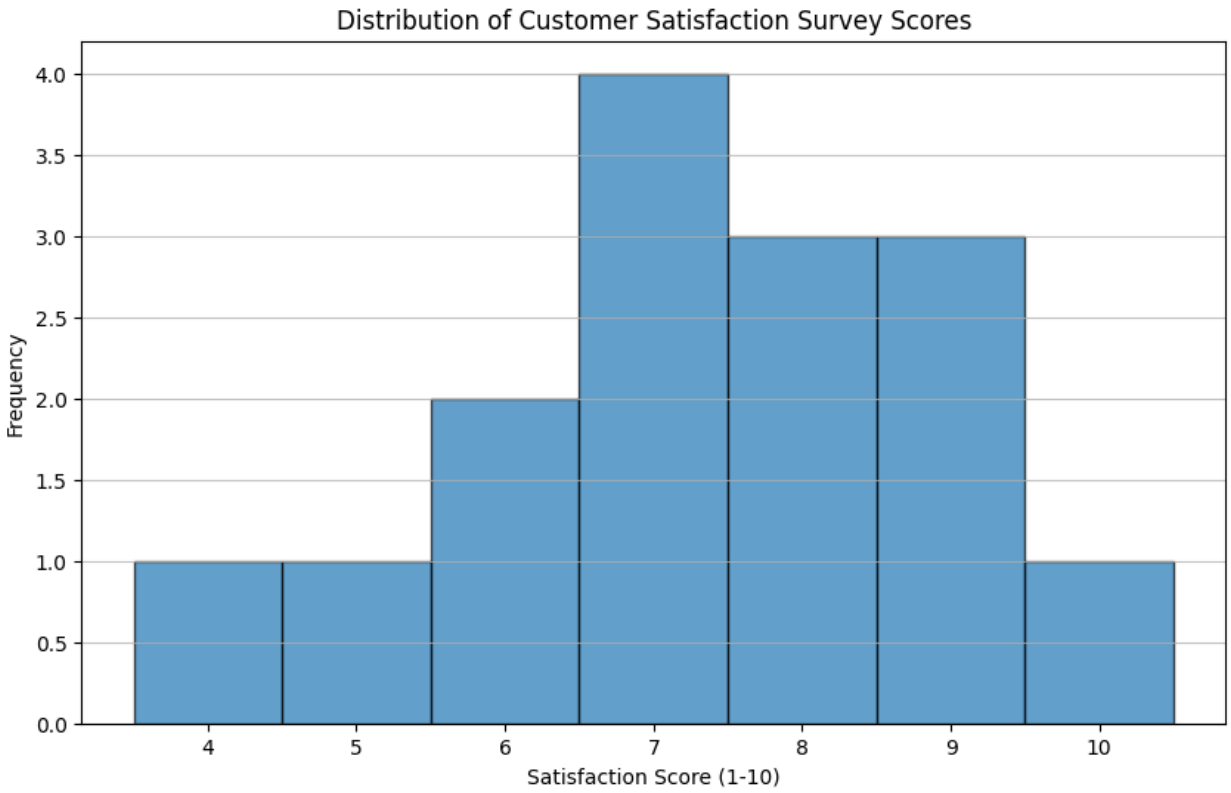
For customer satisfaction data, a histogram would immediately reveal if most customers are highly satisfied (peak towards 8-10), dissatisfied (peak towards 1-3), or if there's a mixed response.

- **Boxplot (Complementary)**: While the question specifically asks for a histogram, a boxplot would also be very useful.
- **Purpose**: Provides a concise summary of the five-number summary (min, Q1, median, Q3, max) and clearly highlights outliers.
- **Insights**: It's excellent for quickly identifying the median, spread of the middle 50%, and the presence of outliers, reinforcing what's seen in the histogram.

By using both summary statistics and visualizations, the team can gain a comprehensive understanding of customer satisfaction, identify areas for improvement, and make data-driven decisions before the new product launch.

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

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

plt.figure(figsize=(10, 6))
plt.hist(survey_scores, bins=np.arange(min(survey_scores),
max(survey_scores) + 2) - 0.5, edgecolor='black', alpha=0.7)
# np.arange(min, max+2)-0.5 creates bins centered around integer
scores for clearer visualization
plt.title('Distribution of Customer Satisfaction Survey Scores')
plt.xlabel('Satisfaction Score (1-10)')
plt.ylabel('Frequency')
plt.xticks(np.arange(min(survey_scores), max(survey_scores) + 1)) #
Ensure x-axis ticks are at integer scores
plt.grid(axis='y', alpha=0.75)
plt.show()

print(f"Survey Scores: {survey_scores}")
print(f"Mean: {np.mean(survey_scores):.2f}")
print(f"Median: {np.median(survey_scores)}")
```

```
from collections import Counter
counts = Counter(survey_scores)
mode_values = [key for key, value in counts.items() if value ==
max(counts.values())]
print(f"Mode: {mode_values}")
print(f"Standard Deviation: {np.std(survey_scores):.2f}")
print(f"Min Score: {np.min(survey_scores)}")
print(f"Max Score: {np.max(survey_scores)}")
```




```
Survey Scores: [7, 8, 5, 9, 6, 7, 8, 9, 10, 4, 7, 6, 9, 8, 7]
Mean: 7.33
Median: 7.0
Mode: [7]
Standard Deviation: 1.58
Min Score: 4
Max Score: 10
```