1. What are the three measures of central tendency? When should each be used?

The three measures of central tendency are:

- **Mean (Arithmetic Average)**: Sum of all values divided by the number of values. Best used when data is **normally distributed** and free of extreme outliers.
- **Median (Middle Value)**: The middle value when data is sorted. Best used when data is **skewed or has outliers**.
- **Mode (Most Frequent Value)**: The most frequently occurring value. Best used for **categorical data** or when identifying the most common occurrence.

Example: For the dataset: **[10, 20, 30, 40, 50]**

- **Mean** = (10+20+30+40+50)/5 = **30**
- **Median** = Middle value = **30**
- **Mode** = No repeating values → No mode

For the dataset: **[10, 20, 20, 40, 50]**

- **Mean** = (10+20+20+40+50)/5 = **28**
- **Median** = Middle value = **20**
- **Mode** = Most frequent value = **20**

---

2. How do mean, median, and mode differ when the data is skewed?

- **Right (Positive) Skewed Distribution**:
  Mean > Median > Mode
  Example: Income distribution (few high earners pull the mean upwards).
- **Left (Negative) Skewed Distribution**:
  Mode > Median > Mean
  Example: Scores in a difficult test (many low scores pull the mean down).
- **Symmetric Distribution**:
  Mean = Median = Mode

Example Data:

- Right Skewed: **[1, 2, 3, 4, 100]** → Mean = 22, Median = 3, Mode = None
- Left Skewed: **[1, 1, 2, 3, 4]** → Mean = 2.2, Median = 2, Mode = 1

---

3. What are the different measures of dispersion, and why are they important?

Measures of dispersion show how spread out the data is:

- **Range**: Difference between the highest and lowest values.
- **Variance**: Average squared difference from the mean.
- **Standard Deviation**: Square root of variance, showing spread in original units.
- **Interquartile Range (IQR)**: Range between Q3 (75th percentile) and Q1 (25th percentile), resistant to outliers.

Dispersion helps in understanding variability and risk (e.g., stock market volatility).

---

4. How is standard deviation different from variance?

- **Variance (σ^2)**: Measures average squared deviations from the mean.
- **Standard Deviation (σ)**: Square root of variance, giving a measure in original units.

Formula:

$$\sigma^2 = \frac{\sum(x_i - \bar{x})^2}{n}$$

$$\sigma = \sqrt{\sigma^2}$$

Example: Data: **[2, 4, 6, 8, 10]**
Mean: **6**
Variance: **$[(2-6)^2 + (4-6)^2 + (6-6)^2 + (8-6)^2 + (10-6)^2] / 5 = 8$**
Standard Deviation: **$\sqrt{8} \approx 2.83$**

---

5. What does a high standard deviation indicate in a dataset?

A **high standard deviation** means the data points are **widely spread** from the mean.

Example:

- **Low SD**: Test scores clustered around the mean (e.g., 78, 80, 82, 79, 81).
- **High SD**: Test scores vary significantly (e.g., 50, 90, 30, 100, 75).

---

6. How do outliers affect the mean and median?

- **Mean**: Strongly influenced by outliers.
- **Median**: Less affected by outliers.

Example: Data: **[10, 12, 14, 16, 100]**

- Mean: **30.4** (highly influenced)
- Median: **14** (remains stable)

---

7. What is the interquartile range (IQR), and how is it useful in detecting outliers?

IQR = **Q3 - Q1** (middle 50% range).
Outliers are **below Q1 - 1.`5×IQR or above Q3 + 1.5×IQR**.

Example: Data: **[1, 2, 3, 4, 5, 6, 7, 8, 100]**
Q1 = **2.5**, Q3 = **7.5**, IQR = **5**
Outlier threshold: **Below -5 or above 15**
Outlier = **100**

Code:

```
import numpy as np

data = [1, 2, 3, 4, 5, 6, 7, 8, 100]
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("Outliers:", outliers)
```

---

8. Can the standard deviation ever be negative? Why or why not?

No, standard deviation cannot be negative because it is derived from squared differences, which are always non-negative.

---

9. What is the five-number summary, and how is it used in boxplots?

The **five-number summary** consists of:

- Minimum
- Q1 (25th percentile)
- Median (50th percentile)
- Q3 (75th percentile)

- Maximum

Used in **boxplots** to visualize data spread and detect outliers.

Code:

```
import matplotlib.pyplot as plt

data = [1, 2, 3, 4, 5, 6, 7, 8, 100]
plt.boxplot(data)
plt.show()
```

---

10. You are analyzing the monthly income of employees in a company. The mean salary is $75,000, but the median salary is $50,000. What does this tell you about the salary distribution?

Analysis:

- **Mean ($75,000) > Median ($50,000) → Right-skewed distribution**.
- Some **high salaries (outliers)** are pulling the mean upwards.

Example Calculation:

Consider salaries: **[30k, 40k, 50k, 55k, 60k, 500k]**

- Mean: **(30+40+50+55+60+500)/6 = 122.5k**
- Median: **(50+55)/2 = 52.5k**

Clearly, **a few high earners (like 500k) distort the mean**, making it much higher than the median.

Implication:

- Most employees earn **below** the mean.
- The company has **income inequality**.

Code:

```
import numpy as np

salaries = [30000, 40000, 50000, 55000, 60000, 500000]
mean_salary = np.mean(salaries)
median_salary = np.median(salaries)

print("Mean Salary:", mean_salary)
print("Median Salary:", median_salary)
```

Conclusion:

- When **Mean > Median**, the data is **right-skewed**.
- **Boxplots and histograms** help visualize salary distribution.
- **Outliers distort the mean, making the median a better measure for skewed data.**

11. What are some key steps in Exploratory Data Analysis (EDA)?

EDA is the process of analyzing datasets to summarize their main characteristics using **statistical methods and visualization**.

Key Steps:

1. **Understand the Data**
   - Load the dataset, check data types, structure, and missing values.
2. **Descriptive Statistics**
   - Calculate **mean, median, mode, standard deviation, variance, IQR**.
3. **Handle Missing Values**
   - Identify missing values and decide on **removal or imputation**.
4. **Detect and Handle Outliers**
   - Use **boxplots, IQR method, or Z-scores**.
5. **Data Visualization**
   - Histograms, boxplots, scatter plots, and correlation heatmaps.
6. **Feature Engineering**
   - Creating new variables or transforming data (e.g., **log transformation**).
7. **Check for Multicollinearity**
   - Use **correlation matrix and VIF (Variance Inflation Factor)**.
8. **Check Distribution**
   - Use **histograms, QQ plots, and normality tests**.

Code:

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

# Load dataset
df = pd.read_csv("data.csv")

# Summary statistics
print(df.describe())

# Check missing values
print(df.isnull().sum())

# Visualize distributions
df.hist(figsize=(10,6))
```

```
plt.show()
```

---

12. What are the differences between univariate, bivariate, and multivariate analysis?

Type of Analysis	Definition	Example
Univariate	Examines one variable at a time	Histogram of employee salaries
Bivariate	Analyzes the relationship between two variables	Scatter plot of advertising budget vs. sales
Multivariate	Examines more than two variables simultaneously	Regression model predicting sales using advertising, price, and region

Example Visualization:

```
# Univariate: Histogram of sales
df['sales'].hist()

# Bivariate: Scatter plot of advertising vs sales
sns.scatterplot(x=df['advertising_budget'], y=df['sales'])

# Multivariate: Pair plot for multiple variables
sns.pairplot(df)
plt.show()
```

---

13. How do you handle missing data during EDA?

Methods to Handle Missing Data:

1. **Check Missing Values**

   ```
   df.isnull().sum()
   ```

2. **Drop Missing Values**

   ```
   df.dropna(inplace=True)  # Removes rows with missing values
   ```

3. **Imputation Methods:**
    - **Mean/Median Imputation** (For numerical values)

```
df['age'].fillna(df['age'].median(), inplace=True)
```

    - **Mode Imputation** (For categorical values)

```
df['gender'].fillna(df['gender'].mode()[0], inplace=True)
```

4. **Predict Missing Values using Machine Learning**

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
df[['age']] = imputer.fit_transform(df[['age']])
```

---

14. When should you use a histogram vs. a box plot?

Plot Type	When to Use
Histogram	To show the **distribution** of a single variable (e.g., Exam Scores, Salary distribution)
Box Plot	To visualize **spread, outliers, and quartiles** of a variable

Example:

```
# Histogram of salary

df['salary'].hist()

# Boxplot of salary
sns.boxplot(x=df['salary'])
plt.show()
```

---

15. What insights can a scatter plot provide?

- **Identifies relationships between two variables.**
- **Detects patterns, clusters, or outliers.**
- **Shows correlation (positive, negative, or no correlation).**

Example:

```
sns.scatterplot(x=df['ad_budget'], y=df['sales'])
plt.show()
```

- **Positive correlation**: Sales increase with advertising.
- **Negative correlation**: Higher prices reduce sales.
- **No correlation**: Random pattern.

---

16. How do you detect and treat outliers in a dataset?

Detection Methods:

1. **Boxplot** (Values outside whiskers are outliers)

```
sns.boxplot(x=df['salary'])
```

2. **IQR Method** (Finds extreme values)

```
Q1 = df['salary'].quantile(0.25)
Q3 = df['salary'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = df[(df['salary'] < lower_bound) | (df['salary'] >
upper_bound)]
```

Treatment Methods:

1. **Remove Outliers**

```
df = df[(df['salary'] >= lower_bound) & (df['salary'] <= upper_bound)]
```

2. **Transform Data (Log, Square Root)**

```
df['salary'] = np.log(df['salary'])
```

---

17. Explain how correlation and causation are different.

- **Correlation**: When two variables **move together** (e.g., ice cream sales and temperature).
- **Causation**: When one variable **directly influences** another.

Example:

- **Correlation but NOT causation**: Number of fire trucks and fire damage (larger fires need more trucks).
- **Causation**: Smoking causes lung cancer.

---

18. What is multicollinearity? How can you detect it?

Multicollinearity occurs when **independent variables are highly correlated**, making it hard to determine their individual effects.

Detection Methods:

1. **Correlation Matrix**

```
sns.heatmap(df.corr(), annot=True)
```

2. **Variance Inflation Factor (VIF)**

```
from statsmodels.stats.outliers_influence import
variance_inflation_factor
X = df[['var1', 'var2', 'var3']]
vif = [variance_inflation_factor(X.values, i) for i in
range(X.shape[1])]
print(vif)
```

VIF > 5 indicates multicollinearity.

Solution:

- Remove one of the highly correlated variables.
- Use **Principal Component Analysis (PCA)**.

---

19. How would you check for normality in a dataset?

Methods:

1. **Histogram**

```
df['sales'].hist()
```

2. **QQ Plot**

```
import scipy.stats as stats
import matplotlib.pyplot as plt
stats.probplot(df['sales'], dist="norm", plot=plt)
plt.show()
```

3. **Shapiro-Wilk Test ($p < 0.05 \rightarrow$ Not Normal)**

```
from scipy.stats import shapiro
stat, p = shapiro(df['sales'])
print('p-value:', p)
```

---

20. You have a dataset with sales data from different regions. How would you explore trends in sales performance?

Approach:

1. **Group Sales by Region**

```
df.groupby('region')['sales'].sum().plot(kind='bar')
```

2. **Time-Series Analysis**

```
df['date'] = pd.to_datetime(df['date'])
df.set_index('date')['sales'].plot()
```

3. **Compare Seasonal Trends**

```
df['month'] = df['date'].dt.month
sns.lineplot(x='month', y='sales', hue='region', data=df)
```

4. **Correlation with Advertising & Price**

```
sns.scatterplot(x=df['ad_budget'], y=df['sales'], hue=df['region'])
```

Probability Theory

21. What is the probability of getting at least one head when flipping two fair coins?

Possible outcomes: **HH, HT, TH, TT**
Total outcomes: **4**
Favorable outcomes (at least one H): **HH, HT, TH** (**3 out of 4**)

$$P(\text{at least 1 head}) = 1 - P(\text{no heads}) = 1 - \frac{1}{4} = \frac{3}{4} = 0.75$$

---

22. What are mutually exclusive and independent events?

- **Mutually Exclusive**: Events **cannot** occur together (e.g., rolling a die and getting a 2 **or** a 5).
- **Independent**: One event does **not affect** the other (e.g., flipping two coins).

Example:

- **Mutually Exclusive**: Drawing an Ace or a King from a single card draw.
- **Independent**: Rolling a die and flipping a coin (one does not affect the other).

---

23. What is the probability of drawing an Ace from a standard deck of 52 cards?

Total Aces = **4**, Total Cards = **52**

$$P(Ace) = \frac{4}{52} = \frac{1}{13} \approx 0.0769$$

---

24. Explain Bayes' Theorem with an example.

Bayes' Theorem:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Example:
A medical test for a disease is **99% accurate** (true positive) but has a **1% false positive rate**.
The disease affects **1 in 1000** people.

Given a **positive test result**, what is the probability that a person actually has the disease?

$$P(Disease|Positive) = \frac{P(Positive|Disease) \cdot P(Disease)}{P(Positive)}$$

$$= \frac{0.99 \times 0.001}{(0.99 \times 0.001) + (0.01 \times 0.999)}$$

$$\approx 0.090$$

Only **9%** of those who test positive actually have the disease.

Code:

```
P_D = 0.001
P_Pos_D = 0.99
P_Pos_NoD = 0.01
P_Pos = (P_Pos_D * P_D) + (P_Pos_NoD * (1 - P_D))

P_D_Pos = (P_Pos_D * P_D) / P_Pos
print("Probability of having disease given a positive test:", P_D_Pos)
```

---

25. What is the difference between discrete and continuous probability distributions?

Type	Definition	Example
Discrete	Finite set of values	Number of customer purchases
Continuous	Infinite possible values	Heights of people

---

26. A factory produces 80% of its products in Plant A and 20% in Plant B. The defect rates are 5% and 10%, respectively. If a product is defective, what is the probability it came from Plant A?

Using **Bayes' Theorem**:

$$P(A|Defective) = \frac{P(Defective|A) \cdot P(A)}{P(Defective)}$$

$$= \frac{(0.05 \times 0.80)}{(0.05 \times 0.80) + (0.10 \times 0.20)}$$

$$= \frac{0.04}{0.06} = 0.6667$$

66.67% probability the defective product is from Plant A.

---

27. How do you define the expected value of a probability distribution?

The **expected value (E[X])** is the weighted average of possible values:

$$E(X) = \sum x_i P(x_i)$$

Example: A die roll (1-6) has **equal probabilities**.

$$E(X) = (1 \times \frac{1}{6}) + (2 \times \frac{1}{6}) + ... + (6 \times \frac{1}{6}) = 3.5$$

Code:

```
import numpy as np

outcomes = np.array([1, 2, 3, 4, 5, 6])
probabilities = np.full(6, 1/6)
expected_value = np.sum(outcomes * probabilities)
print("Expected value of a die roll:", expected_value)
```

---

28. If two dice are rolled, what is the probability that their sum is at least 10?

Favorable outcomes: **(4,6), (5,5), (5,6), (6,4), (6,5), (6,6)**
Total possible outcomes: **6 × 6 = 36**

$$P(Sum \geq 10) = \frac{6}{36} = \frac{1}{6} = 0.1667$$

---

29. What does the Law of Large Numbers state?

As sample size **increases**, the sample mean **approaches** the population mean.

Example:

- **Small samples**: Flipping a coin **10 times** might not give **50% heads**.
- **Large samples**: Flipping **100,000 times** approaches **50% heads**.

Simulation:

```
import numpy as np

flips = np.random.choice([0,1], size=100000, p=[0.5, 0.5])
print("Proportion of heads:", np.mean(flips))
```

---

30. A bag contains **5 red balls** and **3 blue balls**. If you randomly pick **one ball**, what is the probability that it is **red**?

Step 1: Find Total Outcomes

Total balls in the bag = **5 (red) + 3 (blue) = 8**

Step 2: Find Favorable Outcomes

Favorable outcome (picking a red ball) = **5**

Step 3: Use Probability Formula

$$P(\text{Red}) = \frac{\text{Number of Red Balls}}{\text{Total Number of Balls}}$$

$$P(\text{Red}) = \frac{5}{8}$$

P(Red)=0.625 or 62.5%

Probability Distributions

31. What are the key properties of a normal distribution?

- **Bell-shaped and symmetric**
- **Mean = Median = Mode**
- Defined by **mean (μ)** and **standard deviation (σ)**

Example:

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

x = np.linspace(-3, 3, 100)
y = stats.norm.pdf(x, 0, 1)
plt.plot(x, y)
plt.show()
```

---

32. Explain the 68-95-99.7 rule in a normal distribution.

- **68%** of values fall within **1 standard deviation** ($\mu \pm \sigma$)
- **95%** within **2 standard deviations** ($\mu \pm 2\sigma$)
- **99.7%** within **3 standard deviations** ($\mu \pm 3\sigma$)

---

33. How is the Poisson distribution different from the Binomial distribution?

- **Binomial**: Discrete, for **fixed** trials (e.g., coin flips).
- **Poisson**: Discrete, for **rate of occurrences over time/space** (e.g., call arrivals).

---

34. When would you use an exponential distribution?

For modeling **time until an event occurs**, e.g., time between customer arrivals.

Example:

```
from scipy.stats import expon
import matplotlib.pyplot as plt

x = np.linspace(0, 5, 100)
y = expon.pdf(x, scale=1)
plt.plot(x, y)
plt.show()
```

35. A call center receives 10 calls per hour on average. What is the probability that exactly 5 calls arrive in the next hour (Poisson distribution)?

The **Poisson distribution** models the probability of a given number of events occurring in a **fixed interval of time**.

Formula:

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$$

Where:

- λ= 10 (average rate = 10 calls per hour)
- k= 5 (exactly 5 calls)
- e is Euler's number(≈2.718)

- **Calculation:**

$$P(5) = \frac{e^{-10}10^5}{5!}$$

Code:

```
from scipy.stats import poisson

lambda_value = 10  # Average number of calls per hour
k = 5  # Number of calls we want the probability for

probability = poisson.pmf(k, lambda_value)
print(f"Probability of exactly 5 calls: {probability:.4f}")
```

Answer:

P(5) ~0.0378 or 3.78%

---

36. If a coin is flipped 100 times, what is the probability of getting exactly 50 heads (Binomial distribution)?

The **Binomial distribution** models the number of successes in a fixed number of independent trials.

Formula:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Where:

- n= 100 (number of trials)
- k=50 (desired number of heads)
- p=0.5 (probability of getting heads per flip)

Code:

```
from scipy.stats import binom

n = 100  # Number of trials (flips)
p = 0.5  # Probability of heads
k = 50  # Desired number of heads

probability = binom.pmf(k, n, p)
print(f"Probability of getting exactly 50 heads: {probability:.4f}")
```

Answer:

P(50) ~ 0.0796 or 7.96%

---

37. What is the Central Limit Theorem, and why is it important?

The **Central Limit Theorem (CLT)** states that:

- The **sampling distribution of the sample mean** will be **approximately normal**, regardless of the population distribution, if the sample size is **large enough** (n>30).
- The **mean** of the sampling distribution is equal to the **population mean**.
- The **standard deviation** of the sampling distribution (Standard Error) is:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

Why is it important?

- Allows us to **use normal distribution** for inferential statistics (confidence intervals, hypothesis testing).
- Justifies the **use of t-tests and z-tests** even for non-normally distributed data.

Example Simulation (Code):

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

# Generate a non-normal population (exponential distribution)
population = np.random.exponential(scale=2, size=10000)

# Take multiple sample means (size=30)
sample_means = [np.mean(np.random.choice(population, 30)) for _ in
range(1000)]

# Plot the distribution of sample means
plt.hist(sample_means, bins=30, density=True)
plt.title("Sampling Distribution (Central Limit Theorem)")
plt.show()
```

- **Even if the population is skewed, the sample means form a normal distribution!**

---

38. Can a normal distribution be skewed? Why or why not?

No, a **normal distribution cannot be skewed** because:

- It is **symmetric around the mean**.
- Mean = Median = Mode.
- Skewness = **0**.

However, **real-world data** often deviates from normality due to skewness.

Example of Skewed vs. Normal Data (Code):

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

# Normal distribution
normal_data = np.random.normal(loc=50, scale=10, size=1000)

# Skewed distribution (exponential)
skewed_data = np.random.exponential(scale=10, size=1000)

# Plot both distributions
sns.histplot(normal_data, bins=30, kde=True, color='blue', label="Normal")
sns.histplot(skewed_data, bins=30, kde=True, color='red', label="Skewed")
plt.legend()
plt.title("Normal vs. Skewed Distributions")
plt.show()
```

- **Blue**: Normal Distribution (Symmetric)
- **Red**: Skewed Distribution (Right-Skewed)

---

39. If the standard deviation of a dataset is increased, how does it affect the shape of a normal distribution?

- **Increasing Standard Deviation (σ) → Flattens & Widens** the normal curve.
- **Decreasing Standard Deviation (σ) → Narrows & Sharpens** the normal curve.

Example Visualization (Code):

```
import scipy.stats as stats

x = np.linspace(-10, 10, 1000)
y1 = stats.norm.pdf(x, loc=0, scale=1)  # Small SD
y2 = stats.norm.pdf(x, loc=0, scale=3)  # Large SD

plt.plot(x, y1, label="σ = 1")
plt.plot(x, y2, label="σ = 3")
plt.legend()
plt.title("Effect of Standard Deviation on Normal Distribution")
plt.show()
```

Key Insight:

- **Smaller σ**: Tall, narrow curve.
- **Larger σ**: Short, wide curve.

---

40. You are running a system that gets failures at an average rate of 3 per day. What is the probability that no failures occur in the next day?

This follows a **Poisson distribution** with **λ = 3** and **k = 0**.

Formula:

$$P(X = k) = \frac{e^{-\lambda}\lambda^{k}}{k!}$$

Where:

- λ=3 (average failures per day)
- k=0 (no failures)

$$P(0) = \frac{e^{-3}3^{0}}{0!} = e^{-3}$$

Code:

```
from scipy.stats import poisson

lambda_value = 3  # Average failures per day
k = 0  # No failures

probability = poisson.pmf(k, lambda_value)
print(f"Probability of no failures: {probability:.4f}")
```

Answer:

P(0) ~ 0.0498 or 4.98%

Inferential Statistics

41. What is the difference between descriptive and inferential statistics?

Type	Definition	Example
Descriptive Statistics	Summarizes and describes data	Mean, median, mode, variance, standard deviation
Inferential Statistics	Makes predictions or generalizations about a population based on sample data	Hypothesis testing, confidence intervals

Example:

- **Descriptive**: "The average salary of 100 employees is **$50,000**."
- **Inferential**: "Based on this sample, we estimate the average salary in the company is between **$48,000 and $52,000**."

---

42. What are Type I and Type II errors? Can you give a real-life example?

Error Type	Definition	Example
Type I Error (False Positive)	Rejecting a true null hypothesis	A pregnancy test incorrectly shows a man is pregnant
Type II Error (False Negative)	Failing to reject a false null hypothesis	A pregnancy test fails to detect pregnancy in a pregnant woman

Example in Hypothesis Testing:

- H_0: The new drug does not work.
- **Type I Error**: We reject H_0, even though the drug actually **does not work**.
- **Type II Error**: We fail to reject H_0, even though the drug **actually works**.

---

43. How do you interpret a p-value in hypothesis testing?

- **p-value** represents the probability of observing the data **assuming the null hypothesis (H_0) is true**.
- If **$p \leq 0.05$**, reject H_0 (**strong evidence against H_0**).
- If **$p > 0.05$**, fail to reject H_0 (**not enough evidence**).

Example ():

```
from scipy.stats import ttest_1samp
data = [45, 50, 55, 52, 49, 48, 51]
```

```
t_stat, p_value = ttest_1samp(data, 50)
print("p-value:", p_value)
```

Interpretation:

- If **p < 0.05**, we reject the null hypothesis.

---

44. What is a confidence interval, and how does it help in decision-making?

A **confidence interval (CI)** is a range of values that estimates the population parameter.

Formula:

$$CI = \bar{x} \pm Z_{\alpha/2} \times \frac{\sigma}{\sqrt{n}}$$

Example:
If a **95% CI for average salary = [48,000, 52,000]**, we are **95% confident** the true mean salary is within this range.

Code:

```
import scipy.stats as stats
import numpy as np

data = [45, 50, 55, 52, 49, 48, 51]
mean = np.mean(data)
std_error = stats.sem(data)  # Standard error
confidence_interval = stats.t.interval(0.95, len(data)-1, loc=mean,
scale=std_error)
print("95% Confidence Interval:", confidence_interval)
```

---

45. You conduct an A/B test with a 95% confidence interval. What does it mean if zero is included in the interval?

- If **zero is included** in the confidence interval, it means there is **no significant difference** between groups A and B.
- If **zero is NOT included**, there **is a significant difference**.

Example:

- CI = **[-2, 5] → No significant difference** (zero included).
- CI = **[1, 6] → Significant difference** (zero NOT included).

---

46. What are the assumptions of a t-test, and when should you use it?

T-test assumptions:

1. Data is **normally distributed**.
2. Samples are **independent**.
3. Variances are **equal (for independent t-tests)**.

Types of T-tests:

T-test Type	Use Case
One-sample t-test	Compare sample mean to a known population mean
Independent (two-sample) t-test	Compare means of two independent groups
Paired t-test	Compare means of two related groups (e.g., before/after test scores)

Code (One-Sample T-Test):

```
from scipy.stats import ttest_1samp
sample_data = [50, 55, 60, 53, 58, 52, 57]
t_stat, p_value = ttest_1samp(sample_data, 55)
print("p-value:", p_value)
```

---

47. How does an ANOVA test differ from a t-test?

Test	Use Case
T-test	Compares means of **two** groups
ANOVA (Analysis of Variance)	Compares means of **three or more** groups

Code (One-Way ANOVA):

```
from scipy.stats import f_oneway
```

```
group1 = [50, 55, 60, 53, 58, 52, 57]
group2 = [62, 65, 68, 61, 64, 67, 63]
group3 = [70, 75, 80, 73, 78, 72, 77]

f_stat, p_value = f_oneway(group1, group2, group3)
print("p-value:", p_value)
```

If **p < 0.05**, at least one group mean is significantly different.

---

48. A researcher wants to test if a new drug improves recovery time compared to a placebo. What hypothesis test should they use?

- **Two-sample t-test (Independent t-test)** because it compares the mean recovery times between **two groups** (new drug vs. placebo).

Code:

```
from scipy.stats import ttest_ind

drug_group = [10, 12, 14, 13, 11, 15, 14]
placebo_group = [16, 18, 20, 19, 17, 21, 20]

t_stat, p_value = ttest_ind(drug_group, placebo_group)
print("p-value:", p_value)
```

If **p < 0.05**, the drug **significantly reduces** recovery time.

---

49. If the p-value is 0.002, should we reject or fail to reject the null hypothesis at a 0.05 significance level?

- Since **p-value (0.002) < significance level (0.05)**, we **reject the null hypothesis**.
- This means the result is **statistically significant**.

---

50. You want to compare the effectiveness of three different marketing strategies. Which statistical test would you use, and why?

- **ANOVA (Analysis of Variance)** because it compares the means of **three or more groups**.

Code:

```
strategy_A = [120, 130, 125, 140, 135, 128]
strategy_B = [110, 115, 108, 120, 118, 119]
strategy_C = [150, 155, 160, 158, 152, 157]

f_stat, p_value = f_oneway(strategy_A, strategy_B, strategy_C)
print("p-value:", p_value)
```

If **p < 0.05**, at least one marketing strategy performs significantly differently.