

CS23334-FUNDAMENTALS OF DATA SCIENCE

DEVA

DHARSHINI P(240701107)

MATHPLOTLIB-DATA VISUALIZATION

1.a)

Aim:

To analyze the trend of Data Science job postings over the last decade.

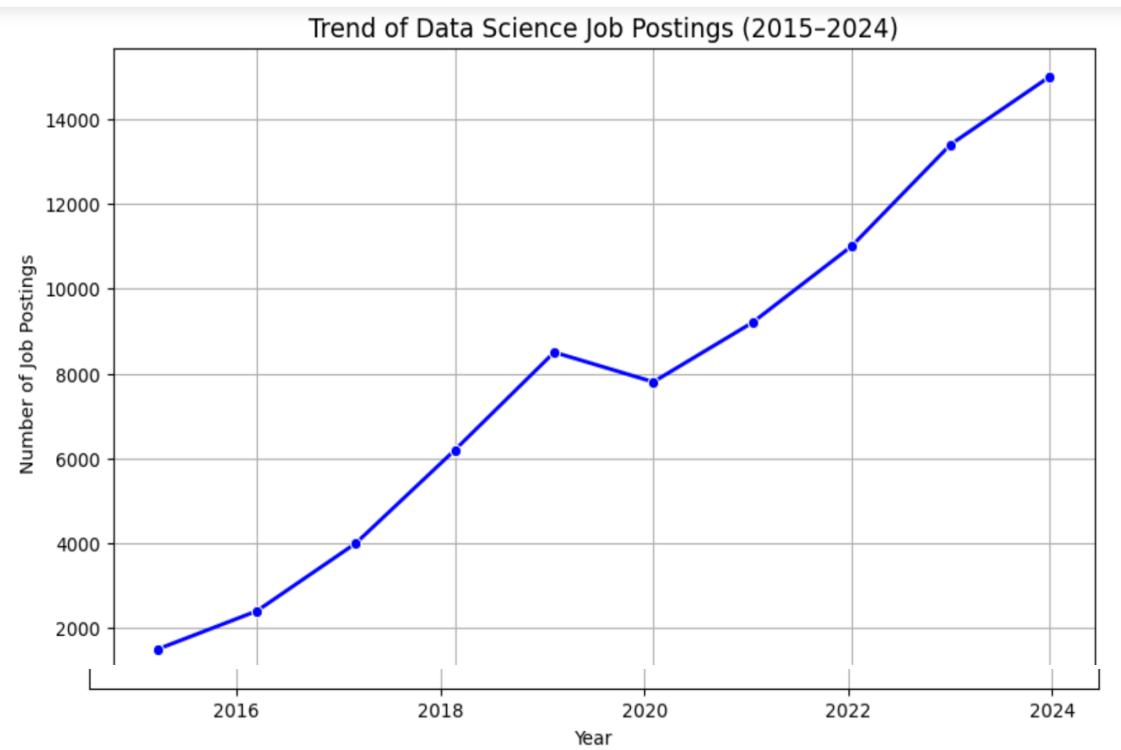
Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = {
    'Year': [2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024],
    'Job_Postings': [1500, 2400, 4000, 6200, 8500, 7800, 9200, 11000, 13400, 15000]
}
df = pd.DataFrame(data)
print("Data Science Job Postings over the last decade:")
print(df)
```

Data Science Job Postings over the last decade:

	Year	Job_Postings
0	2015	1500
1	2016	2400
2	2017	4000
3	2018	6200
4	2019	8500
5	2020	7800
6	2021	9200
7	2022	11000
8	2023	13400
9	2024	15000

```
plt.figure(figsize=(10,6))
sns.lineplot(x='Year', y='Job_Postings', data=df, marker='o', linewidth=2, color='b')
plt.title('Trend of Data Science Job Postings (2015-2024)', fontsize=14)
plt.xlabel('Year')
plt.ylabel('Number of Job Postings')
plt.grid(True)
plt.show()
```



Result:

The visualization clearly shows a significant upward trend in Data Science job opportunities over the last decade.

This indicates the growing demand for data professionals in industries globally.

1.b) MATHPLOTLIB-DATA VISUALIZATION

Aim:

To analyze and visualize the distribution of various data science roles such as Data Analyst, Data Engineer, Data Scientist, etc., from a dataset.

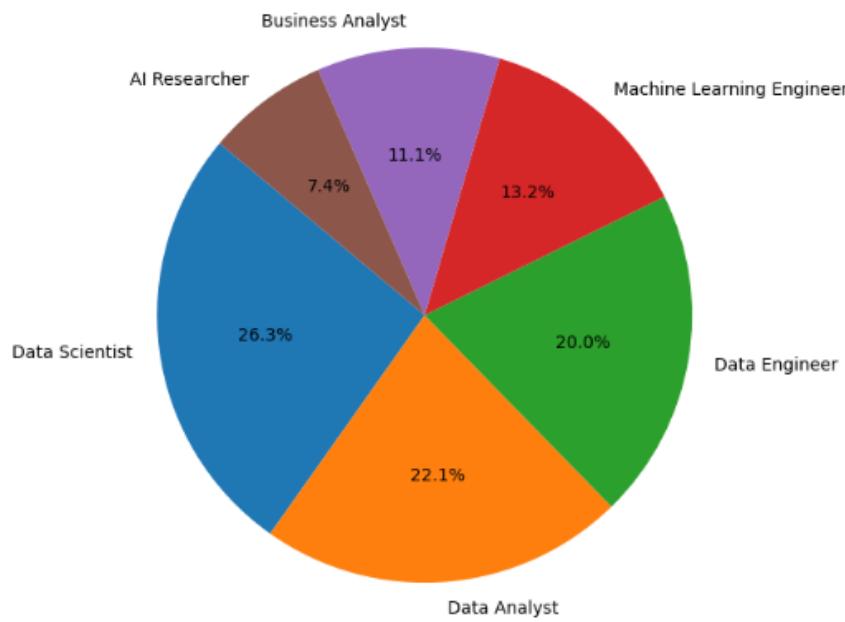
Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = {
    'Role': ['Data Scientist', 'Data Analyst', 'Data Engineer',
             'Machine Learning Engineer', 'Business Analyst', 'AI Researcher'],
    'Job_Postings': [5000, 4200, 3800, 2500, 2100, 1400]
}
df = pd.DataFrame(data)
print("Dataset showing job postings by role:")
print(df)
```

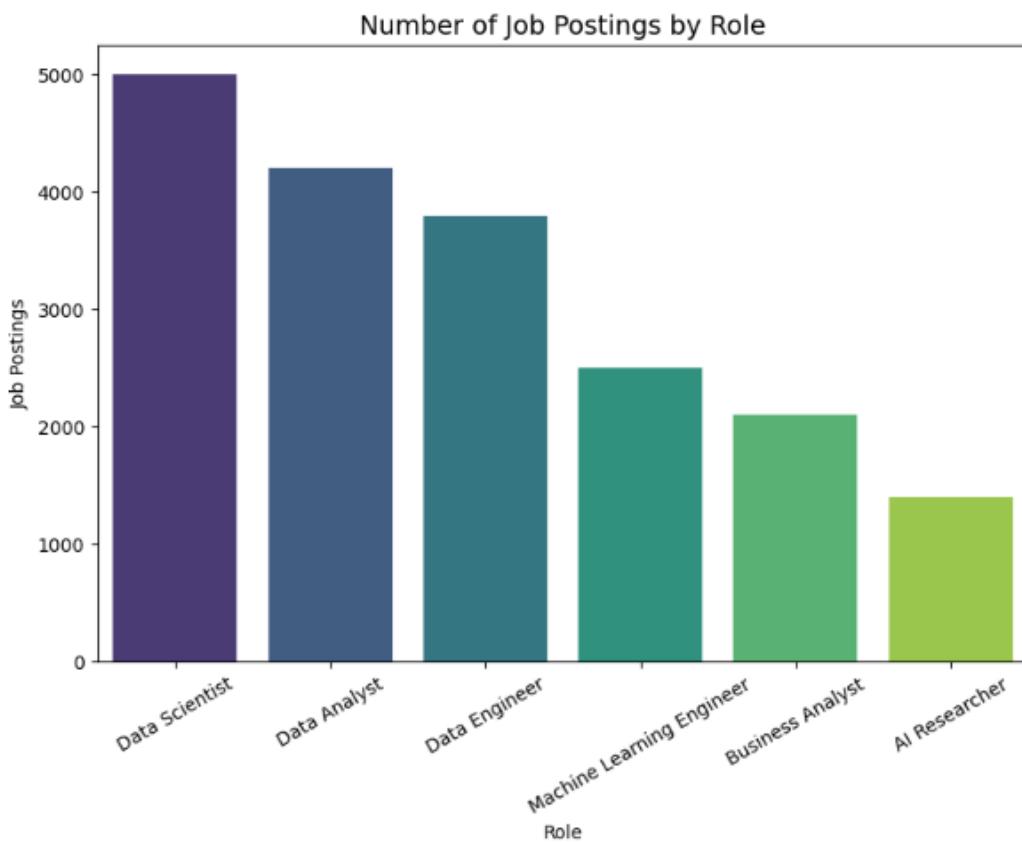
```
Dataset showing job postings by role:
      Role  Job_Postings
0   Data Scientist      5000
1     Data Analyst      4200
2     Data Engineer      3800
3 Machine Learning Eng  2500
4   Business Analyst      2100
5       AI Researcher      1400
```

```
plt.figure(figsize=(7,7))
plt.pie(df['Job_Postings'], labels=df['Role'], autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Data Science Roles', fontsize=14)
plt.show()
```

Distribution of Data Science Roles



```
| plt.figure(figsize=(9,6))
| sns.barplot(x='Role', y='Job_Postings', data=df, palette='viridis')
| plt.title('Number of Job Postings by Role', fontsize=14)
| plt.xlabel('Role')
| plt.ylabel('Job Postings')
| plt.xticks(rotation=30)
| plt.show()
```



Result:

From the visualizations:

- Data Scientist and Data Analyst roles have the highest job postings.
- AI Researcher has the least, indicating it's more specialized.

1.c) DISPLAY STRUCTURED , SEMI-STRUCTURED AND UNSTRUCTURED DATA

Aim:

To conduct an experiment to differentiate Structured, Unstructured, and Semi-structured data using small datasets and explain their characteristics.

Code:

```
import pandas as pd
structured_data = {
    'Employee_ID': [101, 102, 103],
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Department': ['Data Science', 'Analytics', 'AI'],
    'Salary': [70000, 65000, 80000]
}
structured_df = pd.DataFrame(structured_data)
print("Structured Data (Tabular Format):")
print(structured_df)
print("\nCharacteristics: Organized in rows & columns, easy to query using SQL/Pandas.\n")

Structured Data (Tabular Format):
   Employee_ID      Name  Department  Salary
0           101     Alice  Data Science  70000
1           102      Bob    Analytics  65000
2           103  Charlie        AI     80000

Characteristics: Organized in rows & columns, easy to query using SQL/Pandas.
```

```
unstructured_data = """
Employee Feedback:
Alice: "I love working with data models."
Bob: "The analytics project is challenging but fun!"
Charlie: "AI research needs more computing power."
"""

print("Unstructured Data (Text Format):")
print(unstructured_data)
print("Characteristics: Free-form text, no fixed structure, hard to store in databases.\n")

Unstructured Data (Text Format):

Employee Feedback:
Alice: "I love working with data models."
Bob: "The analytics project is challenging but fun!"
Charlie: "AI research needs more computing power."

Characteristics: Free-form text, no fixed structure, hard to store in databases.
```

```
semi_structured_data = [
    {"Employee_ID": 101, "Name": "Alice", "Skills": ["Python", "ML"]},
    {"Employee_ID": 102, "Name": "Bob", "Skills": ["SQL", "Excel"]},
    {"Employee_ID": 103, "Name": "Charlie", "Skills": ["AI", "Deep Learning"]}
]
print("Semi-Structured Data (JSON Format):")
print(semi_structured_data)
print("Characteristics: Uses tags/keys for structure, flexible, easily parsed by JSON tools.\n")

Semi-Structured Data (JSON Format):
[{"Employee_ID": 101, "Name": "Alice", "Skills": ["Python", "ML"]}, {"Employee_ID": 102, "Name": "Bob", "Skills": ["SQL", "Excel"]}, {"Employee_ID": 103, "Name": "Charlie", "Skills": ["AI", "Deep Learning"]}]
Characteristics: Uses tags/keys for structure, flexible, easily parsed by JSON tools.
```

Result:

The experiment successfully differentiates the three types of data formats:

- Structured: Tabular, fixed schema.
- Unstructured: Raw data without defined format.
- Semi-structured: Partially organized with flexible schema (e.g., JSON).

1.d) CRYPTOGRAPHIC PROGRAM

Aim:

To conduct an experiment to encrypt and decrypt a given piece of sensitive data using the Python cryptography library.

Code:

```
from cryptography.fernet import Fernet
key = Fernet.generate_key()
print("Generated Key:", key)
cipher = Fernet(key)
message = "MyPassword123@Bank"
print("\nOriginal Message:", message)
encrypted_message = cipher.encrypt(message.encode())
print("\nEncrypted Message:", encrypted_message)
decrypted_message = cipher.decrypt(encrypted_message).decode()
print("\nDecrypted Message:", decrypted_message)
```

Generated Key: b'tfjAHmd7EE8BX3bQ9_j01TjsXJzFqJiFp1RUFj2EtKU='

Original Message: MyPassword123@Bank

Encrypted Message: b'gAAAAABpCX1A4H6X7IHXXG_YTAhFIUN6ukLc1kzAdNGFYMWY0iNSPOB3lQ1bgqpmXrHYkTxBUW4PSifyabZTiU5I1_JC4rz4ydnaQ_CMIsfSJI2yiS_guo='

Decrypted Message: MyPassword123@Bank

Result:

The given sensitive data was successfully encrypted into unreadable text and then decrypted back to its original form using the generated key.