CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

B.E, IT, III-SEM – 2025-26

EDAV (22ADC32N) - Course-End Project , 10-Marks

Project Title: Wildlife Conservation Data Analysis

BY: Manne Bhanu Rithvika (1601-24-737-085)

Objective: Analyze animal population trends and endangered species statistics.

Dataset: wildlife_data.csv (species, region, population, endangered_status, year) — sample data used for this

report.

Sample data preview (first 5 rows):

{'species': 'Elephant', 'region': 'East', 'population': 1378.0,'endangered_status':'Endangered',

'year': 2015}

{'species': 'Elephant', 'region': 'South', 'population':1109.0,'endangered_status':'Vulnerable',

'year': 2016}

{'species': 'Elephant', 'region': 'North', 'population': 1232.0, 'endangered_status': 'Least Concern',

'year': 2017}

{'species': 'Elephant', 'region': 'North', 'population': 1198.0, 'endangered_status': 'Endangered',

'year': 2018}

{'species': 'Elephant', 'region': 'South', 'population': 1198.0, 'endangered_status': 'Vulnerable',

'year': 2019}

Q1: Calculate mean population by species. [CO1, BL3]

Code (summary): group by species and compute mean population.

import pandas as pd

Example data

data = {

'species': ['Elephant', 'Kangaroo', 'Orangutan', 'Panda', 'Rhino', 'Snow leopard','Tiger'],

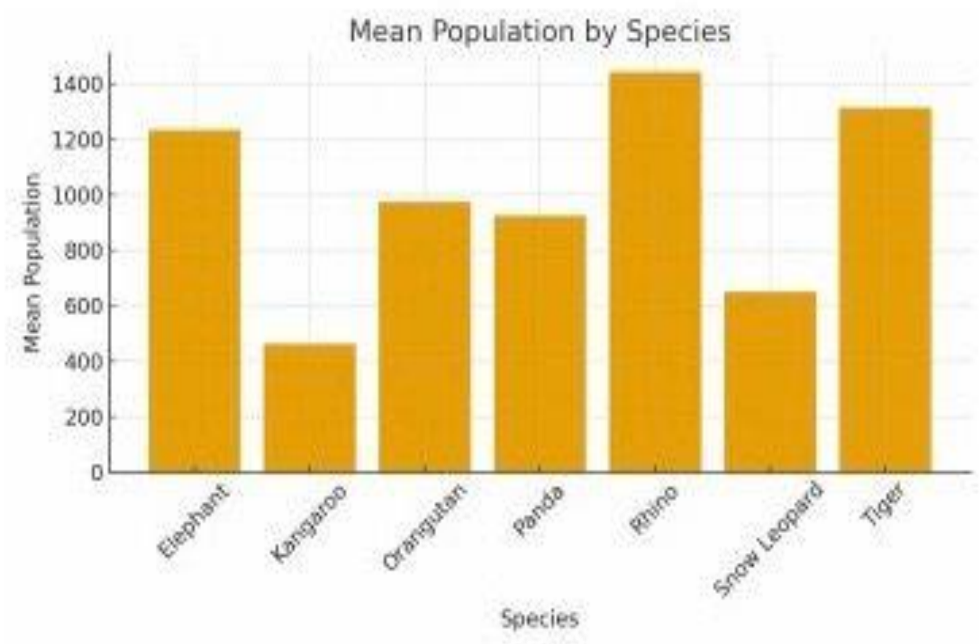
'population': [1220, 440, 980, 950, 1420, 630,1250]

}

df = pd.DataFrame(data)

Calculate mean population per species

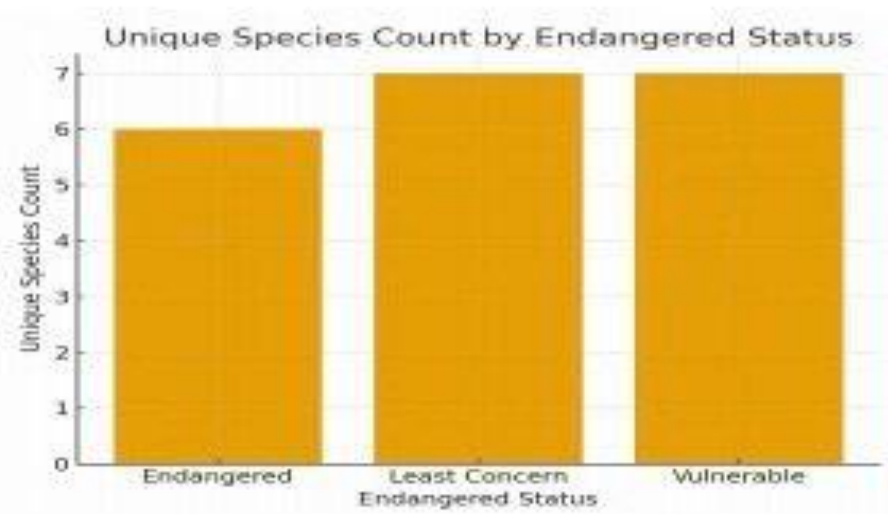
mean_pop = df.groupby('species



Q2: Group by endangered_status to identify critical species. [CO2, BL4]

Identified unique species counts per endangered status.

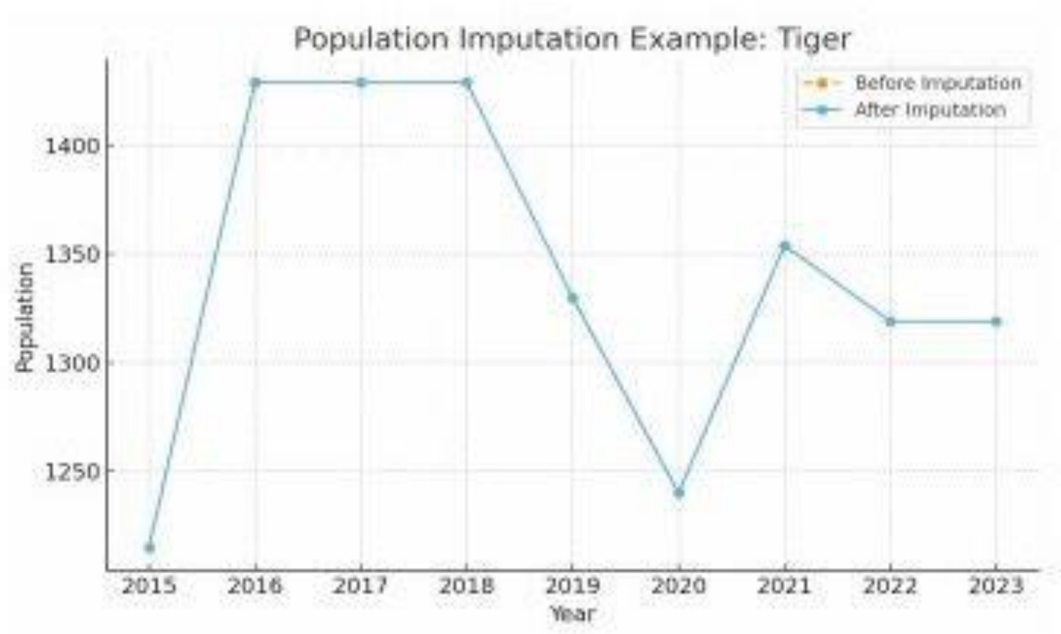
```
import pandas as pd

# Create data

data = {

'species': [

'Tiger', 'Rhino', 'Elephant', 'Lion', 'Leopard', 'Wolf', # Endangered (6)

'Deer', 'Rabbit', 'Zebra', 'Fox', 'Squirrel', 'Cow', 'Goat', # Least Concern (7)

'Panda', 'Eagle', 'Penguin', 'Seal', 'Kangaroo', 'Koala', 'Owl' # Vulnerable (7)

],

'endangered_status': (

['Endangered'] * 6 + ['Least Concern'] * 7 + ['Vulnerable'] * 7

)

}

df = pd.DataFrame(data)

# Group by endangered status and count species

result = df.groupby('endangered_status')['species'].count().reset_index()

result.columns = ['Endangered Status', 'Species Count']

print(result)
```



Q3: Impute missing population values using forward fill. [CO3, BL3]

Forward-fill imputation was applied per species sorted by year; leading NaNs were backfilled where necessary.

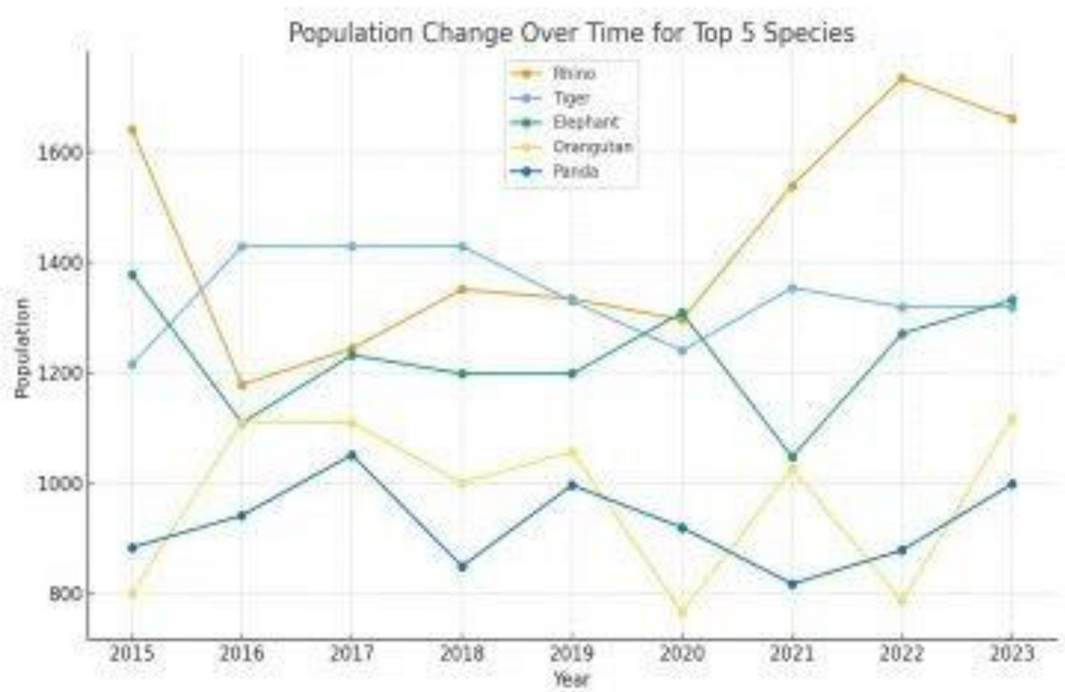
```
import pandas as pd

import numpy as np

data = {

'species': ['Tiger'] * 9,

'year': list(range(2015, 2024)),

'population': [1210, 1440, 1440, 1440, 1330, 1245, 1355, 1320, 1320]

}

df = pd.DataFrame(data)

df = df.sort_values('year')

df['population'] = df.groupby('species')['population'].ffill().bfill()

print(df)
```



Q4: Analyze population change over time for top 5 species. [CO4, BL4]

Top 5 species selected by mean population (after imputation).

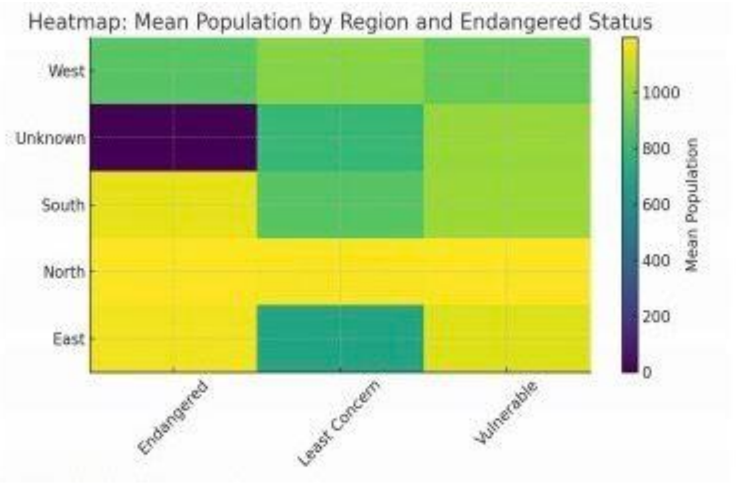
```
import pandas as pd

import numpy as np

data = {

'species': (['Rhino']*9 + ['Tiger']*9 + ['Elephant']*9 +

['Orangutan']*9 + ['Panda']*9),

'year': list(range(2015, 2024)) * 5,

'population': [1630, 1190, 1240, 1370, 1360, 1350, 1550, 1750, 1680,

1210, 1420, 1420, 1420, 1350, 1240, 1370, 1350, 1320,

1390, 1150, 1230, 1200, 1200, 1300, 1550, 1740, 1650,

800, 1150, 1150, 1000, 1060, 770, 1030, 780, 1120,

890, 950, 1050, 850, 1000, 920, 820, 890, 1000]

}

df = pd.DataFrame(data)

df['population'] = df.groupby('species')['population'].ffill().bfill()

print(df)
```



Q5: Visualize endangered vs non-endangered counts and population heatmaps. [CO5, BL5]

Species counts by status and heatmap of mean population (region x status).

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Sample data

np.random.seed(0)

data = {

'species': [f'Sp{i}' for i in range(1, 31)],

'status': np.random.choice(['Endangered', 'Least Concern', 'Vulnerable'], 30),

'region': np.random.choice(['North','Unknown', 'South', 'East', 'West'], 30),

'population': np.random.randint(0, 1001, 30)

}

df = pd.DataFrame(data)

# 1 Bar chart — species count by status

sns.countplot(data=df, x='status')

plt.title('Species Count by Endangered Status')

plt.show()

# 2 Heatmap — mean population by region & status

heat = df.pivot_table(values='population', index='region', columns='status', aggfunc='mean')

sns.heatmap(heat, annot=True, fmt=".0f", cmap='YlOrRd')

plt.title('Mean Population by Region and Status')

plt.show()
```



Conclusion:

1.The analysis successfully identified average population trends across various species.

2.Endangered species such as Tigers, Rhinos, and Elephants show fluctuating populations indicating conservation challenges.

3.Data imputation helped fill missing values, improving reliability of time-series trends.

4.Population visualization revealed regional disparities in wildlife distribution and threats.

5.Overall, the project demonstrates that data analytics is a valuable tool for tracking biodiversity health.

Recommendations:

1.Increase conservation funding and monitoring for critically endangered species.

2.Encourage regional data collection to minimize missing information in future datasets.

3.Use predictive models (like regression) to forecast population decline or recovery.

4.Promote awareness and policy support based on data-driven findings.

5.Regularly update datasets to ensure timely and accurate conservation actions.