assignment-3-3

November 6, 2024

Assignment no - 3

Name: Aarya Trifale, Roll no: 26, Section: F

Problem statement:

A. Measure central tendency and variance. Data analysis in terms of Normal Distribution, Binomial Distribution.

B. To measure statistical relationships in data using Correlation and Linear regression.

C. To perform data visualization of various datasets in terms of charts and plots.

[1]:
```
import pandas as pd
 import numpy as np

file = pd.read_excel("City_Environment_Data.xlsx")
file
```

[1]:

	Air_Quality_Index	Green_Cover_Percentage	Population_Density	\
0	172	19.182039	8629	
1	47	71.219337	5377	
2	117	78.519330	8070	
3	192	39.968722	215	
4	323	16.897670	5819	
..	…	…	…	
295	111	64.257037	7862	
296	488	81.250959	3408	
297	258	15.290090	265	
298	27	25.531383	1515	
299	467	24.402407	1921	

	Annual_Rainfall	Waste_Collection_Efficiency	Public_Transport_Usage	\
0	897.559548	76.004071	52	
1	599.224193	51.533052	56	
2	598.407301	61.220681	79	
3	1784.414269	97.683785	38	
4	743.180391	79.115987	53	
..	…	…	…	
295	515.095543	64.713308	60	

```
296      1213.739294                    71.767673                    48
297      1563.155586                    89.772826                    78
298       565.963148                    83.875418                    49
299      1819.282225                    96.893219                    23

     Noise_Pollution_Level  Energy_Consumption_Per_Capita  Average_Temperature
0                76.187341                    1900.006379            32.190733
1                76.135166                    4349.305504            33.313057
2                81.394048                    2307.768999            15.384641
3                73.219156                    1419.336829            26.397443
4                88.740655                    1334.122354            20.893005
..                     …                            …                     …
295              54.788086                    1054.621626            15.492291
296              61.369457                    3448.723492            27.395442
297              32.666603                    3894.492378            20.424331
298              38.750470                    2155.627085            16.976919
299              66.011065                    4894.566073            23.058169

[300 rows x 9 columns]
```

Part-A

Central Tendency

[2]:
```
mean = file.mean()
print(f"mean is {mean}")
```

```
mean is Air_Quality_Index                241.353333
Green_Cover_Percentage            48.169519
Population_Density              5278.163333
Annual_Rainfall                 1250.378033
Waste_Collection_Efficiency       75.457111
Public_Transport_Usage            44.966667
Noise_Pollution_Level             60.626692
Energy_Consumption_Per_Capita   2966.631600
Average_Temperature               25.096592
dtype: float64
```

[3]:
```
mode = file.mode()
print(f"mode is {mode}")
```

```
mode is      Air_Quality_Index  Green_Cover_Percentage  Population_Density  \
0                488.0                5.004705              3063.0
1                  NaN                5.344089              3352.0
2                  NaN                5.785406              5377.0
3                  NaN                5.823495              9214.0
4                  NaN                5.830147                 NaN
..                 …                     …                   …
```

```
295                NaN               88.518872                NaN
296                NaN               88.668002                NaN
297                NaN               88.727744                NaN
298                NaN               89.353694                NaN
299                NaN               89.356509                NaN

     Annual_Rainfall  Waste_Collection_Efficiency  Public_Transport_Usage  \
0         500.818947                    50.033211                    27.0
1         502.075025                    50.103232                     NaN
2         505.790527                    50.232742                     NaN
3         507.729446                    50.265502                     NaN
4         515.095543                    50.567756                     NaN
..               …                            …                       …
295      1963.825803                    99.394458                     NaN
296      1970.728495                    99.633350                     NaN
297      1981.023136                    99.790765                     NaN
298      1987.835494                    99.803564                     NaN
299      1999.712867                    99.899700                     NaN

     Noise_Pollution_Level  Energy_Consumption_Per_Capita  Average_Temperature
0                30.022041                    1000.294798            15.018066
1                30.222586                    1006.378282            15.109896
2                30.543600                    1019.149541            15.110209
3                30.798967                    1024.813659            15.211715
4                30.944294                    1029.124056            15.313522
..                     …                            …                     …
295              88.975589                    4932.810953            34.619588
296              89.345285                    4934.221673            34.809066
297              89.345306                    4949.739959            34.810784
298              89.439102                    4958.100332            34.898038
299              89.911595                    4992.795811            34.922009

[300 rows x 9 columns]
```

[4]:
```
median = file.median()
print(f"median is {median}")
```

```
median is Air_Quality_Index                239.500000
Green_Cover_Percentage              47.072714
Population_Density                5428.000000
Annual_Rainfall                   1224.483333
Waste_Collection_Efficiency         75.439118
Public_Transport_Usage              44.500000
Noise_Pollution_Level               60.568205
Energy_Consumption_Per_Capita     2996.119174
Average_Temperature                 25.051193
dtype: float64
```

Standard deviation and variance

[5]:
```
variance = file.var()
print(f"variance is {variance}")
```

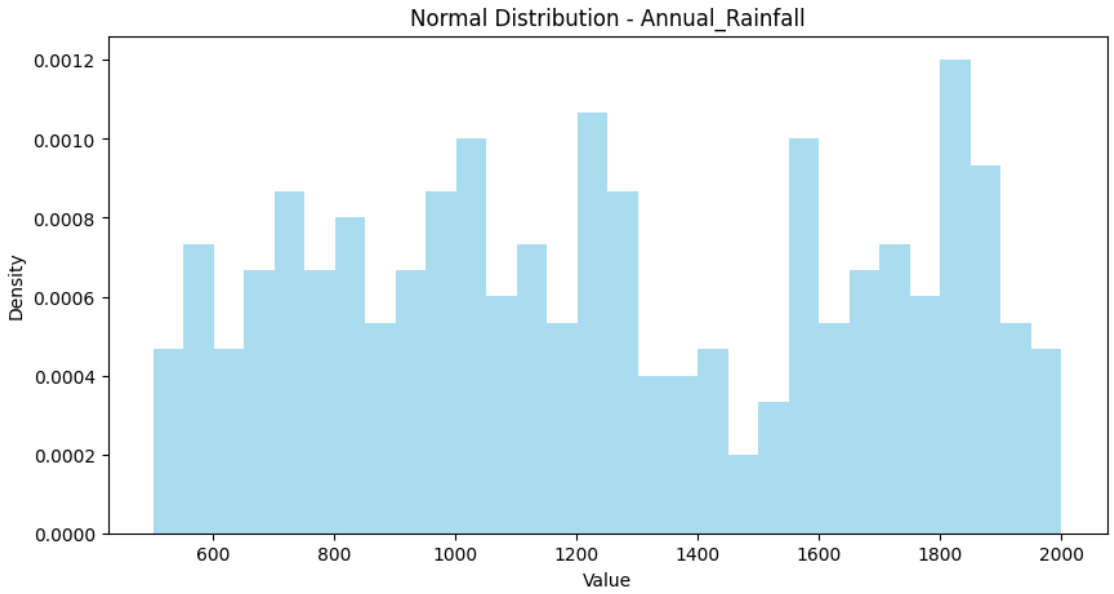
```
variance is Air_Quality_Index                2.032185e+04
Green_Cover_Percentage           5.895826e+02
Population_Density               8.019069e+06
Annual_Rainfall                  1.886441e+05
Waste_Collection_Efficiency      2.318504e+02
Public_Transport_Usage           4.182464e+02
Noise_Pollution_Level            3.150559e+02
Energy_Consumption_Per_Capita    1.349351e+06
Average_Temperature              3.404017e+01
dtype: float64
```

[6]:
```
std_dev = file.std()
print(f"standard dev is {std_dev}")
```

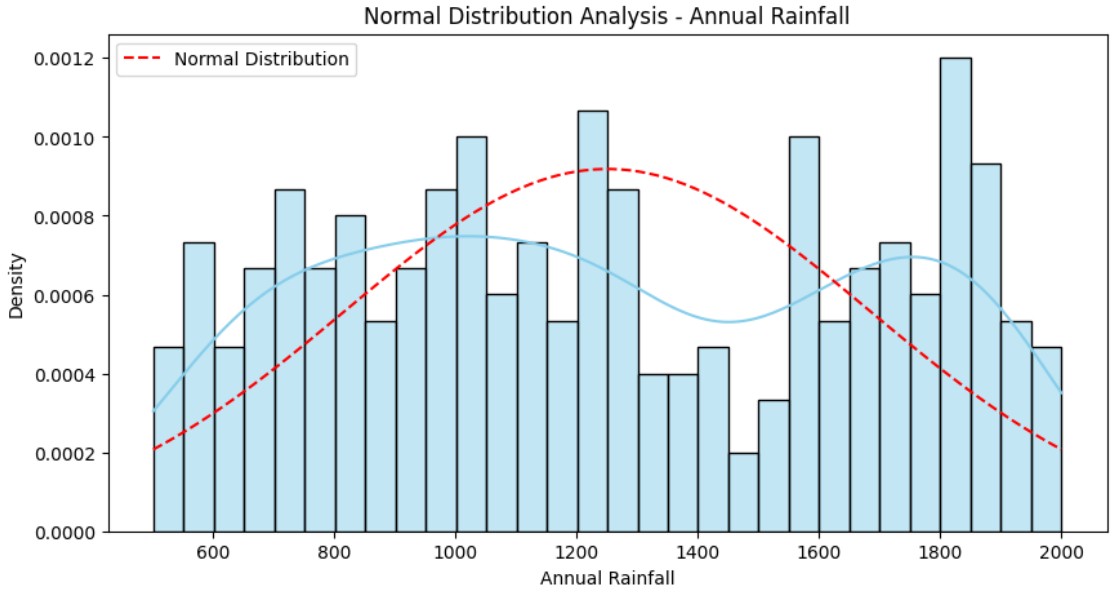
```
standard dev is Air_Quality_Index                 142.554719
Green_Cover_Percentage             24.281323
Population_Density               2831.796020
Annual_Rainfall                   434.331747
Waste_Collection_Efficiency        15.226635
Public_Transport_Usage             20.451073
Noise_Pollution_Level              17.749814
Energy_Consumption_Per_Capita    1161.615572
Average_Temperature                 5.834395
dtype: float64
```

Distribution Analysis

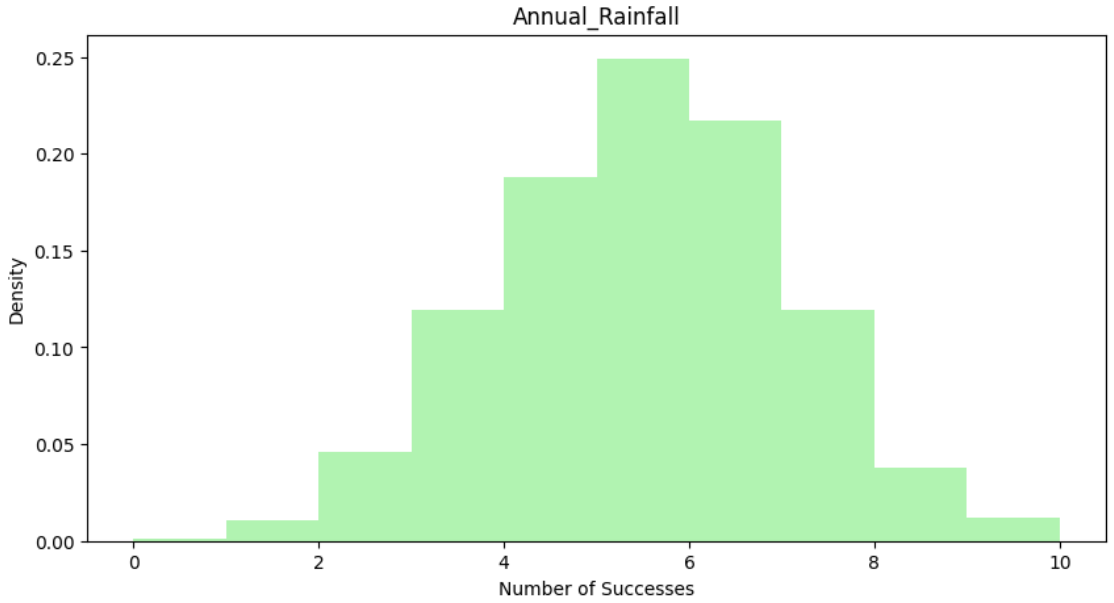
[7]:
```
import matplotlib.pyplot as plt
 # Normal Distribution Plot
plt.figure(figsize=(10, 5))
plt.hist(file['Annual_Rainfall'], bins=30, color='skyblue', alpha=0.7,␣
 ↪density=True)
plt.title('Normal Distribution - Annual_Rainfall')
plt.xlabel('Value')
plt.ylabel('Density')
plt.show()
```



[8]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm

# Mean and Standard Deviation for Annual Rainfall
mean = file['Annual_Rainfall'].mean()
std = file['Annual_Rainfall'].std()

# Plotting the Histogram with KDE and Overlaying a Normal Curve
plt.figure(figsize=(10, 5))
sns.histplot(file['Annual_Rainfall'], bins=30, color='skyblue', kde=True,␣
 ↪stat='density')

# Overlay theoretical normal distribution
x = np.linspace(file['Annual_Rainfall'].min(), file['Annual_Rainfall'].max(),␣
 ↪100)
plt.plot(x, norm.pdf(x, mean, std), color='red', linestyle='--', label='Normal␣
 ↪Distribution')

# Titles and Labels
plt.title('Normal Distribution Analysis - Annual Rainfall')
plt.xlabel('Annual Rainfall')
plt.ylabel('Density')
plt.legend()
plt.show()
```



[9]:
```
# Binomial Distribution Example
n, p = 10, 0.5  # Example parameters for binomial distribution
binom_data = np.random.binomial(n, p, 1000)
plt.figure(figsize=(10, 5))
plt.hist(binom_data, bins=10, color='lightgreen', alpha=0.7, density=True)
plt.title('Annual_Rainfall')
plt.xlabel('Number of Successes')
plt.ylabel('Density')
plt.show()
```



Part B:

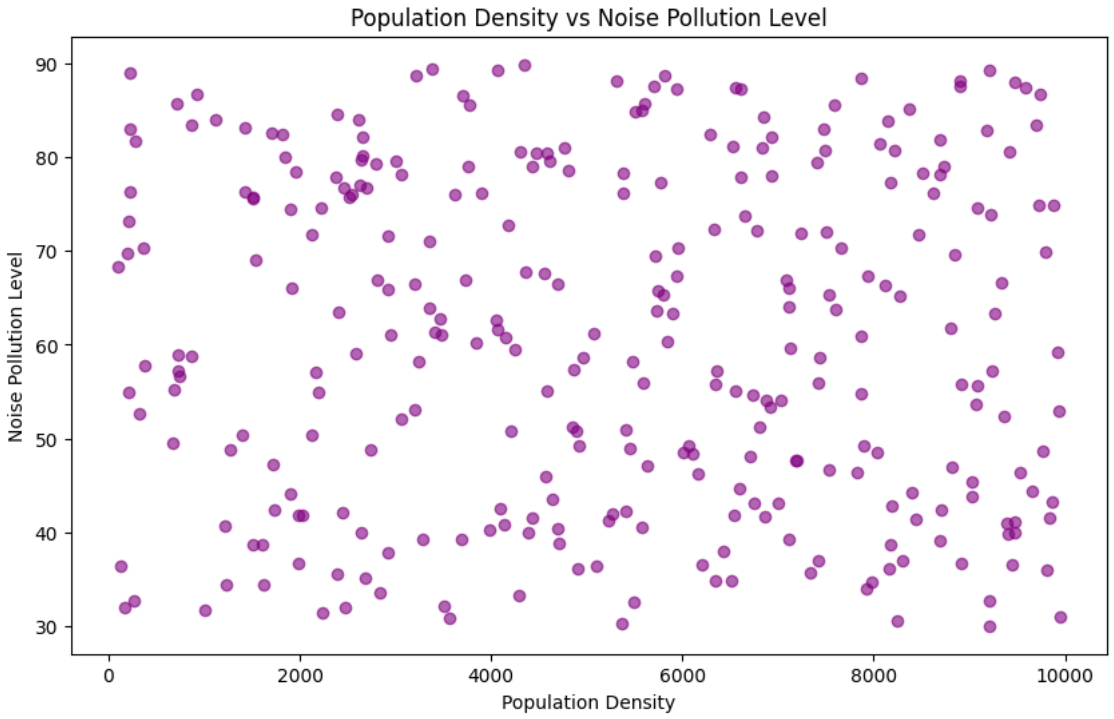
Statistical Relationships

[10]:
```
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Select two columns for linear regression analysis
X = file[['Population_Density']].values  # Independent variable
y = file['Air_Quality_Index'].values     # Dependent variable

# Initialize and fit the linear regression model
model = LinearRegression()
model.fit(X, y)

# Predictions
y_pred = model.predict(X)

# Plotting the linear regression
plt.figure(figsize=(10, 6))
plt.scatter(X, y, color='blue', alpha=0.5, label="Actual Data")
plt.plot(X, y_pred, color='red', label="Regression Line")
plt.title("Linear Regression - Population Density vs Air Quality Index")
plt.xlabel("Population Density")
plt.ylabel("Air Quality Index")
plt.legend()
plt.show()
```

Part - C

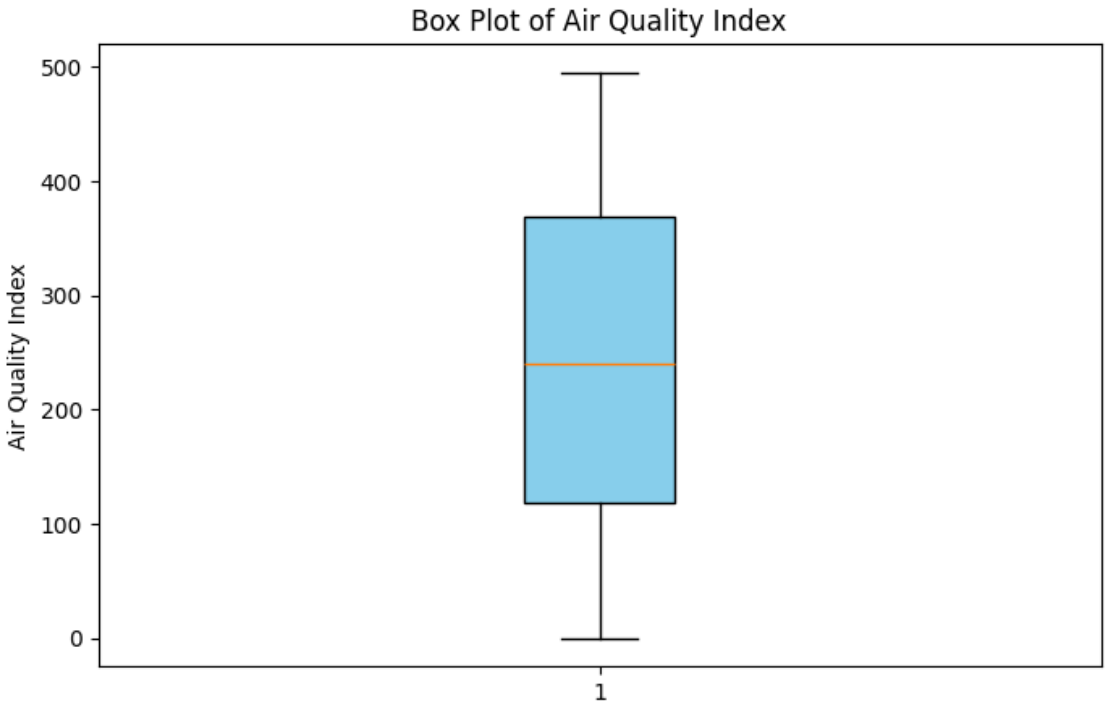
Scatter Plot (e.g., Population Density vs Noise Pollution Level)

[11]:
```
import matplotlib.pyplot as plt

# Scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(file['Population_Density'], file['Noise_Pollution_Level'],␣
 ↪color='purple', alpha=0.6)
plt.title("Population Density vs Noise Pollution Level")
plt.xlabel("Population Density")
plt.ylabel("Noise Pollution Level")
plt.show()
```



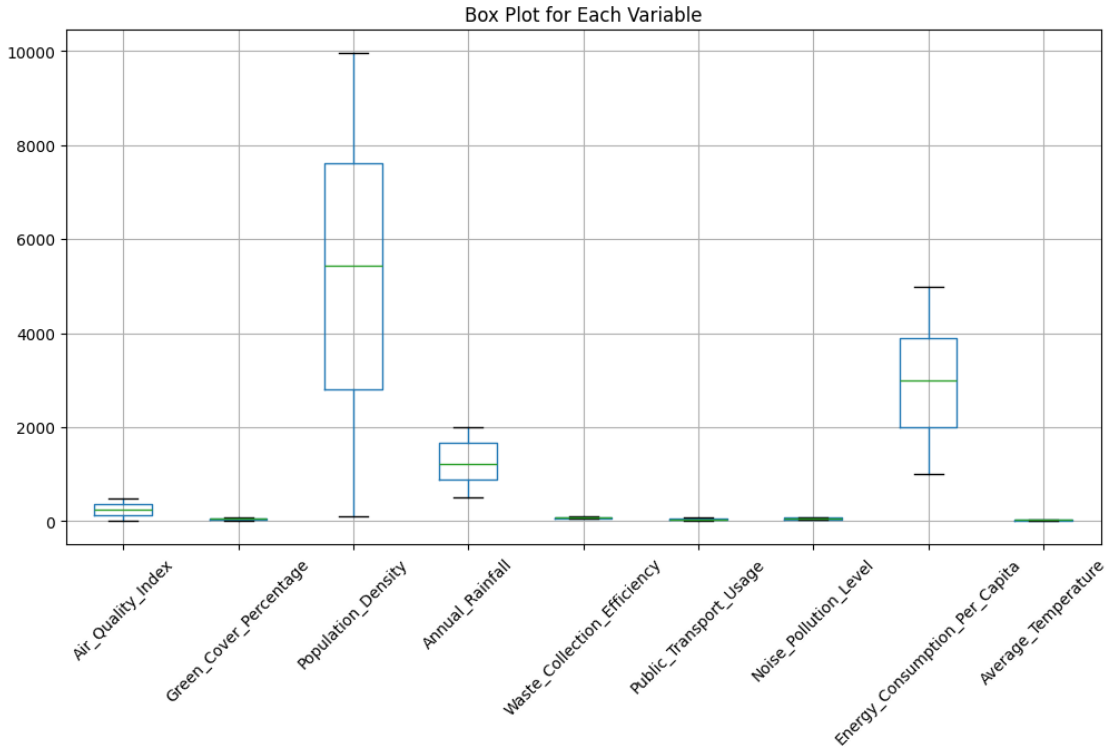
Box Plot (e.g., Air Quality Index)

[12]:
```
# Box plot
plt.figure(figsize=(8, 5))
plt.boxplot(file['Air_Quality_Index'], patch_artist=True,␣
 ↪boxprops=dict(facecolor="skyblue"))
plt.title("Box Plot of Air Quality Index")
plt.ylabel("Air Quality Index")
plt.show()
```



Bar Chart (e.g., Average Temperature by City or Categories)

[20]:
```
# Box plot for each variable
file.boxplot(figsize=(12, 6), rot=45)
plt.title("Box Plot for Each Variable")
plt.show()
```


Box Plot for Each Variable
10000
8000
6000
4000
2000
0
Air_Quality_Index
Green_Cover_Percentage
Population_Density
Annual_Rainfall
Waste_Collection_Efficiency
Public_Transport_Usage
Noise_Pollution_Level
Energy_Consumption_Per_Capita
Average_Temperature

[ ]: