Day_14_Task

Assignment:

1.ecommerce sales analysis:

an online store in flipkart wants to understand daily sales variation

take 30 days of the sales

tasks:

calculate mean,median,mode

calculate std and var

calculate correlation

```
# E-commerce Sales Analysis (30 Days)
import pandas as pd
import numpy as np
# 30 days Flipkart daily sales data (in Rs)
sales_data = [12000, 15000, 13000, 17000, 16000,
              18000, 20000, 21000, 19000, 22000,
              23000, 25000, 24000, 26000, 27000,
              30000, 28000, 29000, 31000, 32000,
              33000, 34000, 35000, 36000, 37000,
              38000, 39000, 40000, 42000, 41000]
# Create DataFrame
df = pd.DataFrame({
    "Day": range(1, 31),
    "Sales": sales_data
})
# Display data
print("30 Days Sales Data:\n")
```

```
print(df)
# Mean, Median, Mode
mean_value = df["Sales"].mean()
median_value = df["Sales"].median()
mode_value = df["Sales"].mode()
# Standard Deviation and Variance
std_value = df["Sales"].std()
var_value = df["Sales"].var()
# Correlation (Day vs Sales)
correlation_value = df["Day"].corr(df["Sales"])
# Print Results
print("\nStatistical Analysis Results:\n")
print("Mean Sales:", mean_value)
print("Median Sales:", median_value)
if len(mode_value) > 1:
    print("Mode: No unique mode")
else:
    print("Mode:", mode_value[0])
print("Standard Deviation:", std_value)
print("Variance:", var_value)
print("Correlation between Day and Sales:", correlation_value)
```

```
TERMINAL

PS C:\Users\mouni\Downloads> python sales_analysis.py
>>
30 Days Sales Data:

    Day  Sales
0     1  12000
1     2  15000
2     3  13000
3     4  17000
4     5  16000
5     6  18000
6     7  20000
7     8  21000
8     9  19000
9    10  22000
10   11  23000
11   12  25000
12   13  24000
13   14  26000
14   15  27000
15   16  30000
16   17  28000
17   18  29000
18   19  31000
19   20  32000
20   21  33000
21   22  34000
22   23  35000
23   24  36000
24   25  37000
25   26  38000
26   27  39000
27   28  40000
28   29  42000
29   30  41000
```

```
Statistical Analysis Results:

Mean Sales: 27433.333333333332
Median Sales: 27500.0
Mode: No unique mode
Standard Deviation: 8916.019420057954
Variance: 79495402.29885058
Correlation between Day and Sales: 0.9948382391045318
PS C:\Users\mouni\Downloads>
```

2.salary distribution:

we need to check salary related insights

tasks:

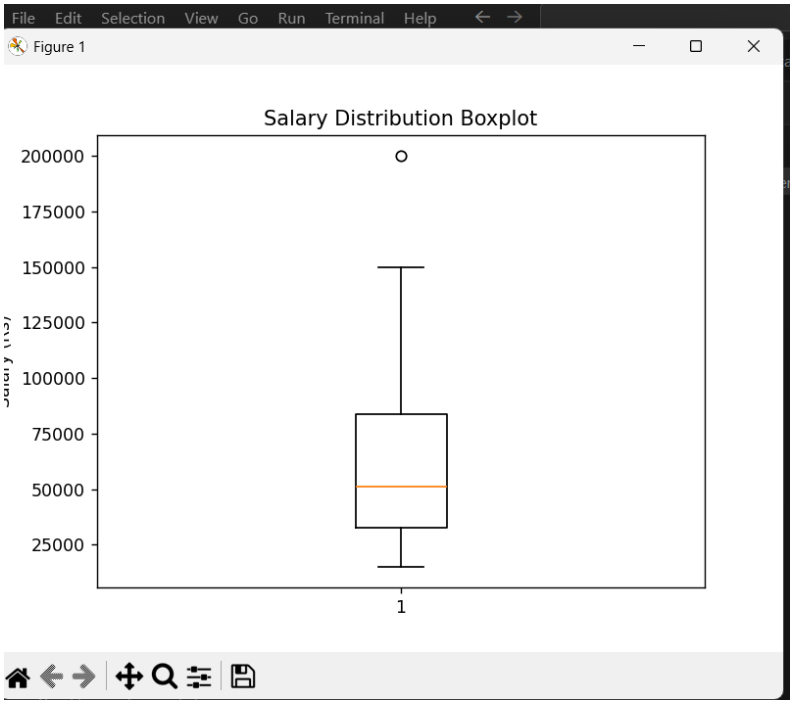
calculate quartiles

plot boxplot to check the outliers

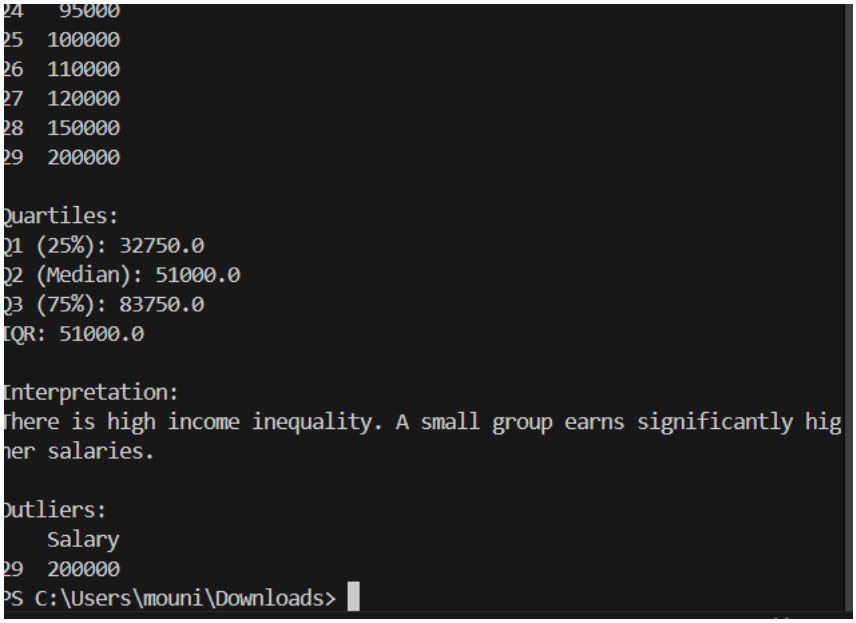
interpret on income on the inequality

```
# Salary Distribution Analysis
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Step 1: Create Salary Dataset (in Rs per month)
salary_data = [15000, 18000, 20000, 22000, 25000,
               27000, 30000, 32000, 35000, 38000,
               40000, 42000, 45000, 48000, 50000,
               52000, 55000, 60000, 65000, 70000,
               75000, 80000, 85000, 90000, 95000,
               100000, 110000, 120000, 150000, 200000]   # High values create inequality
# Create DataFrame
df = pd.DataFrame({"Salary": salary_data})

print("Salary Dataset:\n")
print(df)
# Step 2: Calculate Quartiles
Q1 = df["Salary"].quantile(0.25)
Q2 = df["Salary"].quantile(0.50)  # Median
Q3 = df["Salary"].quantile(0.75)
IQR = Q3 - Q1
```

```
print("\nQuartiles:")
print("Q1 (25%):", Q1)
print("Q2 (Median):", Q2)
print("Q3 (75%):", Q3)
print("IQR:", IQR)
# Step 3: Boxplot to detect outliers
plt.boxplot(df["Salary"])
plt.title("Salary Distribution Boxplot")
plt.ylabel("Salary (Rs)")
plt.show()
# Step 4: Interpretation (Income Inequality)
print("\nInterpretation:")
if (Q3 - Q1) > 50000:
    print("There is high income inequality. A small group earns significantly higher salaries.")
else:
    print("Income distribution is relatively balanced.")
# Detect Outliers using IQR rule
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = df[(df["Salary"] < lower_bound) | (df["Salary"] > upper_bound)]
print("\nOutliers:")
print(outliers)
```




```
TERMINAL

PS C:\Users\mouni\Downloads> python salary_analysis.py
>>
Salary Dataset:

    Salary
0    15000
1    18000
2    20000
3    22000
4    25000
5    27000
6    30000
7    32000
8    35000
9    38000
10   40000
11   42000
12   45000
13   48000
14   50000
15   52000
16   55000
17   60000
18   65000
19   70000
20   75000
21   80000
22   85000
23   90000
24   95000
25  100000
26  110000
27  120000
28  150000
29  200000
```



3.hospital waiting time

A hospital recorded patient waiting time

tasks:

find the variance and range

identify skewness

detect the outlier if any by using IQR

Hospital Waiting Time Analysis (in minutes)

import pandas as pd

import numpy as np

Step 1: Create dataset (waiting time in minutes for 30 patients)

waiting_time = [5, 7, 8, 10, 12,

15, 18, 20, 22, 25,

28, 30, 32, 35, 38,

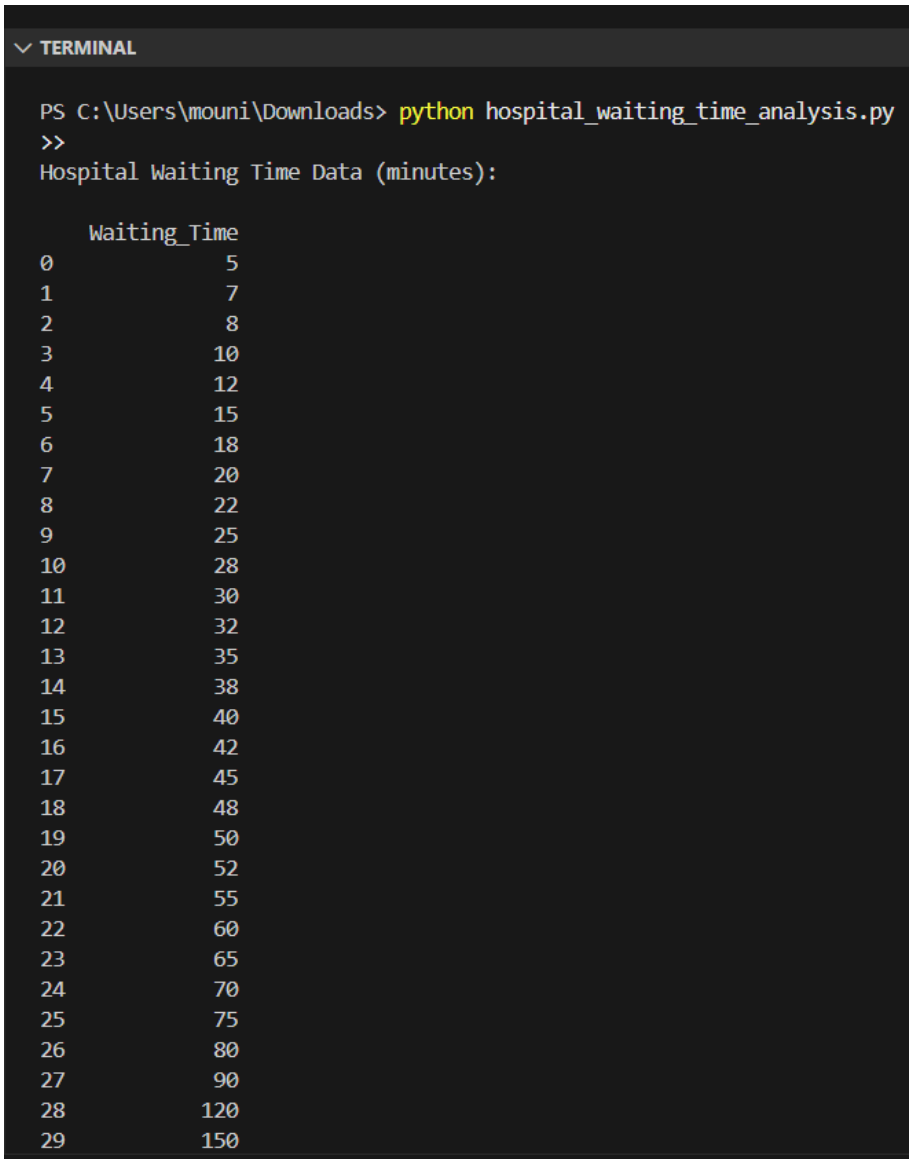
40, 42, 45, 48, 50,

52, 55, 60, 65, 70,

75, 80, 90, 120, 150] # High values create skewness & outliers

```
# Create DataFrame
df = pd.DataFrame({"Waiting_Time": waiting_time})
print("Hospital Waiting Time Data (minutes):\n")
print(df)
# Step 2: Variance
variance_value = df["Waiting_Time"].var()
#Step 3: Range
range_value = df["Waiting_Time"].max() - df["Waiting_Time"].min()
print("\nVariance:", variance_value)
print("Range:", range_value)
# Step 4: Skewness
skewness_value = df["Waiting_Time"].skew()
print("Skewness:", skewness_value)
if skewness_value > 0:
    print("Distribution is Positively Skewed (Right skewed)")
elif skewness_value < 0:
    print("Distribution is Negatively Skewed (Left skewed)")
else:
print("Distribution is Symmetrical")
# Step 5: Outlier Detection using IQR
Q1 = df["Waiting_Time"].quantile(0.25)
Q3 = df["Waiting_Time"].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = df[(df["Waiting_Time"] < lower_bound) | (df["Waiting_Time"] >
upper_bound)]
```

```
print("\nQ1:", Q1)
print("Q3:", Q3)
print("IQR:", IQR)
print("Lower Bound:", lower_bound)
print("Upper Bound:", upper_bound)
print("\nOutliers:")
print(outliers)
```



```
Variance: 1138.2310344827586
Range: 145
Skewness: 1.3816752534832375
Distribution is Positively Skewed (Right skewed)

Q1: 20.5
Q3: 58.75
IQR: 38.25
Lower Bound: -36.875
Upper Bound: 116.125

Outliers:
    Waiting_Time
28           120
29           150
PS C:\Users\mouni\Downloads>
```