**Batch: A3** **Experiment Number:4**

**Roll Number:16010423099** **Name: Suryanshu Banerjee**

**Aim of the Experiment:** To perform exploratory data analysis using python NUMPY

**Program/ Steps:**

```
import pandas as pd
import numpy as np

df = pd.read_csv('C:/Users/SuryanshuBanerjee/myFiles/college/steam.csv')

print("Dataset Info:")
print(df.info())
print("\nBasic Statistics:")
print(df.describe())
print("\nMissing Values:")
print(df.isnull().sum())

df['publishers'] = df['publishers'].fillna("Unknown")
df['developers'] = df['developers'].fillna("Unknown")
print("\nMissing Values After Cleaning:")
print(df.isnull().sum())

print("Summary of 'price' column:")
print(df['price'].describe())

print("\nSummary of 'copiesSold' column:")
print(df['copiesSold'].describe())
```

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

df = pd.read_csv('C:/Users/SuryanshuBanerjee/myFiles/college/steam.csv')

plt.figure(figsize=(10, 6))
df['price'].plot(kind='density', color='blue', linewidth=2)
plt.title("Density Plot of Game Prices")
plt.xlabel("Price")
plt.show()

plt.figure(figsize=(10, 6))
df['copiesSold'].plot(kind='hist', bins=30, color='orange', edgecolor='black')
plt.title("Frequency Plot of Copies Sold")
plt.xlabel("Copies Sold")
plt.show()

print("Measures of Central Tendency:")
print("Price Mean:", df['price'].mean())
print("Price Median:", df['price'].median())
print("Copies Sold Mean:", df['copiesSold'].mean())
print("Copies Sold Median:", df['copiesSold'].median())

plt.figure(figsize=(8, 6))
```

```
plt.boxplot(df['price'].dropna(), vert=False, patch_artist=True,
boxprops=dict(facecolor='skyblue'))

plt.title("Boxplot of Game Prices")

plt.xlabel("Price")

plt.show()


plt.figure(figsize=(10, 6))

plt.scatter(df['price'], df['copiesSold'], color='green', marker='o', s=10)

plt.title("Scatter Plot of Price vs Copies Sold")

plt.xlabel("Price")

plt.ylabel("Copies Sold")

plt.legend(["Price vs Copies Sold"], loc="upper right")

plt.show()


df['releaseDate'] = pd.to_datetime(df['releaseDate'], errors='coerce')

df_sorted = df.sort_values('releaseDate')

plt.figure(figsize=(12, 6))

plt.plot(df_sorted['releaseDate'], df_sorted['price'], linestyle='--', color='purple')

plt.title("Line Plot of Release Date vs Price")

plt.xlabel("Release Date")

plt.ylabel("Price")

plt.legend(["Release Date vs Price"], loc="upper right")

plt.show()


plt.figure(figsize=(10, 6))
```

```
plt.bar(df['name'][:10], df['price'][:10], color='c', label='Price')

plt.bar(df['name'][:10], df['copiesSold'][:10], bottom=df['price'][:10], color='orange',
label='Copies Sold')

plt.title("Bar Plot Comparison of Price and Copies Sold for Top 10 Games")

plt.xlabel("Game Name")

plt.ylabel("Values")

plt.legend(loc="upper left")

plt.xticks(rotation=45, ha="right")

plt.tight_layout()

plt.show()


plt.figure(figsize=(10, 6))

plt.plot(df_sorted['releaseDate'], df_sorted['copiesSold'], linestyle='-', color='teal', marker='*',
markersize=5)

plt.title("Styled Line Plot of Release Date vs Copies Sold")

plt.xlabel("Release Date")

plt.ylabel("Copies Sold")

plt.legend(["Styled Line Plot"], loc="upper right")

plt.show()
```

**Output/Result:**

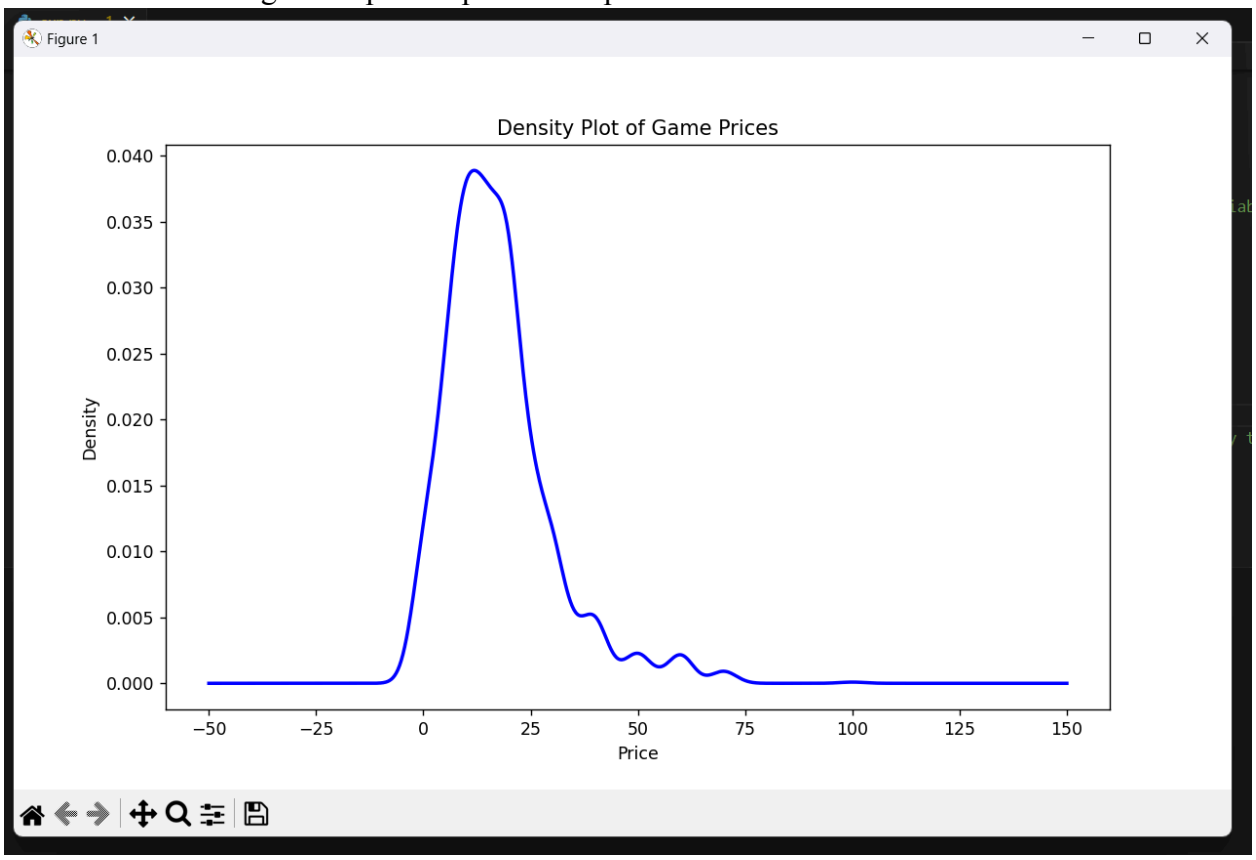
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   name            1500 non-null   object
 1   releaseDate     1500 non-null   object
 2   copiesSold      1500 non-null   int64
 3   price           1500 non-null   float64
 4   revenue         1500 non-null   float64
 5   avgPlaytime     1500 non-null   float64
 6   reviewScore     1500 non-null   int64
 7   publisherClass  1500 non-null   object
 8   publishers      1499 non-null   object
 9   developers      1498 non-null   object
 10  steamId         1500 non-null   int64
dtypes: float64(3), int64(3), object(5)
memory usage: 129.0+ KB
None

Basic Statistics:
        copiesSold        price       revenue  avgPlaytime  reviewScore       steamId
count  1.500000e+03  1500.000000  1.500000e+03  1500.000000  1500.000000  1.500000e+03
mean   1.414826e+05    17.519513  2.632382e+06    12.562704    76.201333  2.183788e+06
std    1.132757e+06    12.646612  2.781024e+07    21.542173    24.319438  6.067725e+05
min    5.930000e+02     0.000000  2.067400e+04     0.000000     0.000000  2.488000e+04
25%    4.918750e+03     9.990000  4.550425e+04     3.564848    72.000000  1.792795e+06
50%    1.192850e+04    14.990000  1.090530e+05     6.762776    83.000000  2.321985e+06
75%    3.786975e+04    19.990000  4.551568e+05    13.104473    92.000000  2.693228e+06
max    3.073915e+07    99.990000  8.377934e+08   296.332852   100.000000  3.107330e+06
```

```
Missing Values:
name             0
releaseDate      0
copiesSold       0
price            0
revenue          0
avgPlaytime      0
reviewScore      0
publisherClass   0
publishers       1
developers       2
steamId          0
dtype: int64

Missing Values After Cleaning:
name             0
releaseDate      0
copiesSold       0
price            0
revenue          0
avgPlaytime      0
reviewScore      0
publisherClass   0
publishers       0
developers       0
steamId          0
dtype: int64
```

```
Summary of 'price' column:
count    1500.000000
mean       17.519513
std        12.646612
min         0.000000
25%         9.990000
50%        14.990000
75%        19.990000
max        99.990000
Name: price, dtype: float64

Summary of 'copiesSold' column:
count    1.500000e+03
mean     1.414826e+05
std      1.132757e+06
min      5.930000e+02
25%      4.918750e+03
50%      1.192850e+04
75%      3.786975e+04
max      3.073915e+07
Name: copiesSold, dtype: float64
```

5 observations:

1. Dataset Composition: The dataset has 1,500 rows with both numeric and non-numeric columns, providing a broad set of variables, including game details like price, copies sold, and developer information.
2. Missing Data: Several columns, notably publishers and developers, had missing values, which have been addressed by filling them with "Unknown" to maintain data consistency.
3. Price Distribution: The price column shows a significant variation, with some games priced at zero (likely free games) and others priced significantly higher. The average price can give insight into general pricing trends.
4. Popularity Measure: The copiesSold column also shows a wide range, from very few copies to highly popular games with large sales numbers, indicating a mix of niche and popular games in the dataset.
5. Data Structure: After minimal EDA, the dataset is confirmed as suitable for further detailed analysis, including visualizations and correlation analysis, particularly for understanding the impact of price on copies sold or reviews.



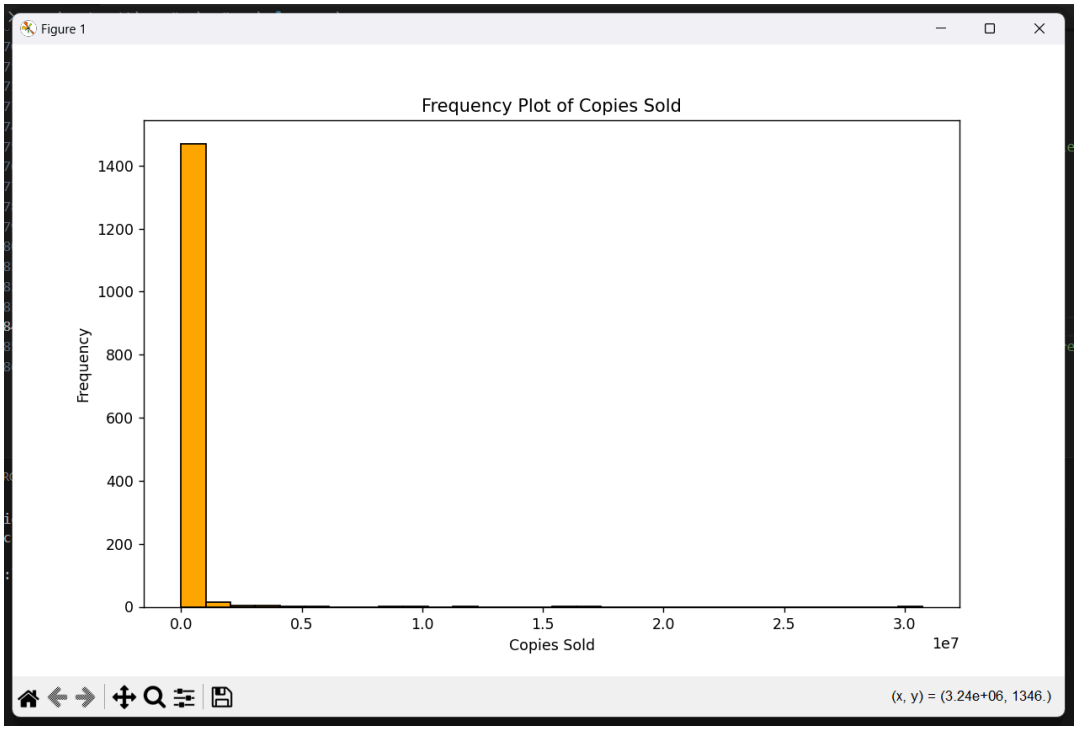


Figure 1
Frequency Plot of Copies Sold
Frequency
Copies Sold
1e7
(x, y) = (3.24e+06, 1346.)

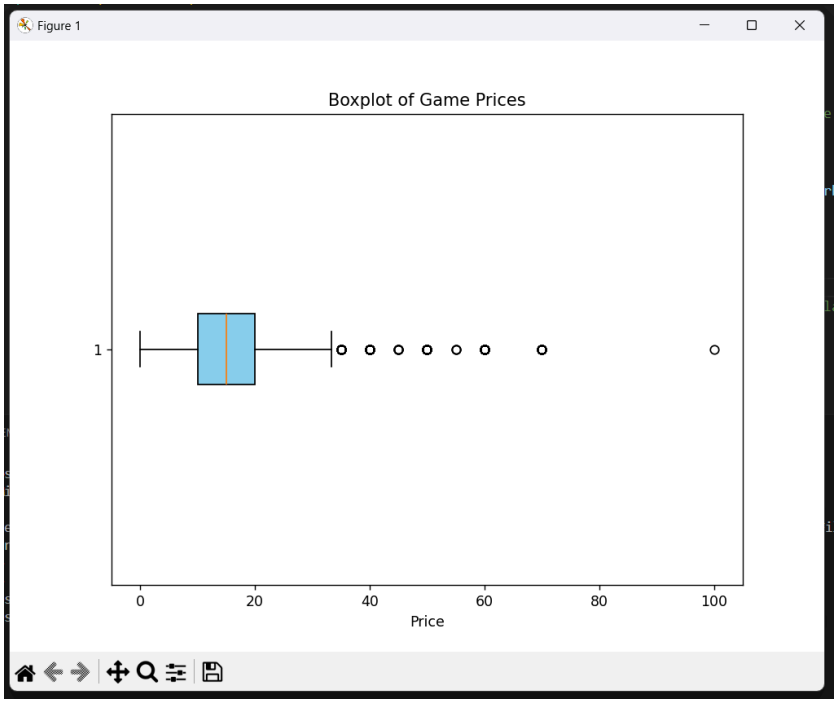


Figure 1
Boxplot of Game Prices
Price

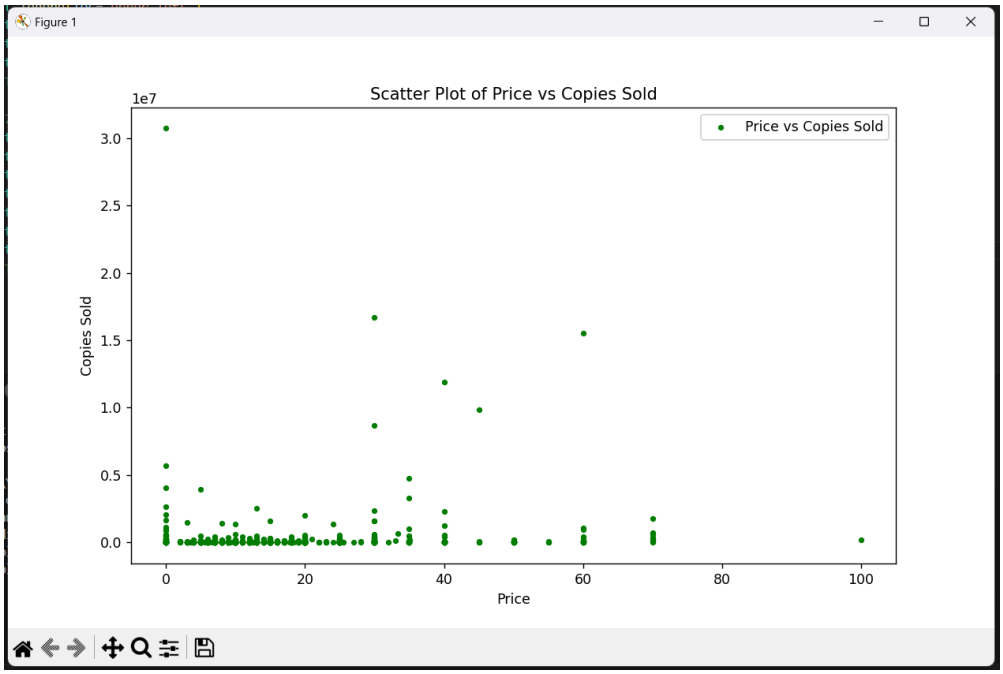


Figure 1
Scatter Plot of Price vs Copies Sold
1e7
Price vs Copies Sold
Copies Sold
Price
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100

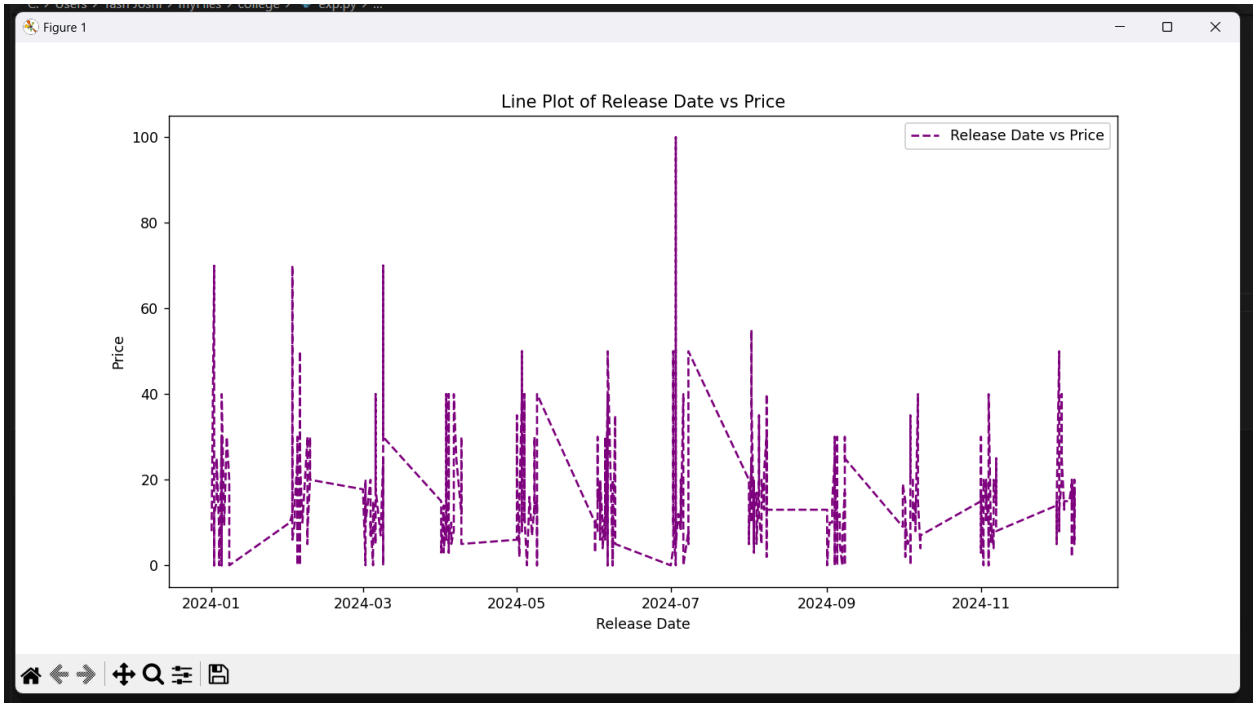


Figure 1
Line Plot of Release Date vs Price
Release Date vs Price
Price
Release Date
100
80
60
40
20
0
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11

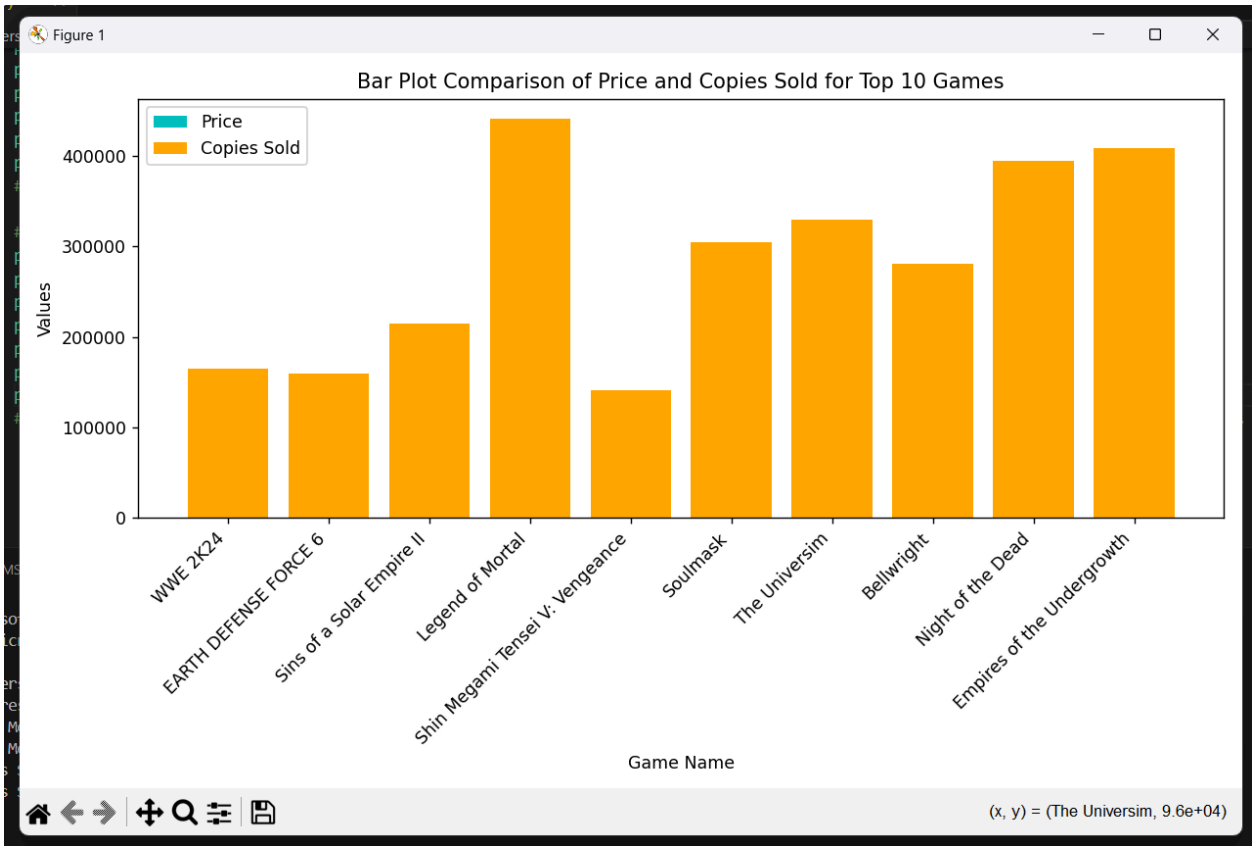


Figure 1
Bar Plot Comparison of Price and Copies Sold for Top 10 Games
Price
Copies Sold
Values
0
100000
200000
300000
400000
WWE 2K24
EARTH DEFENSE FORCE 6
Sins of a Solar Empire II
Legend of Mortal
Shin Megami Tensei V: Vengeance
Soulmask
The Universim
Bellwright
Night of the Dead
Empires of the Undergrowth
Game Name
(x, y) = (The Universim, 9.6e+04)

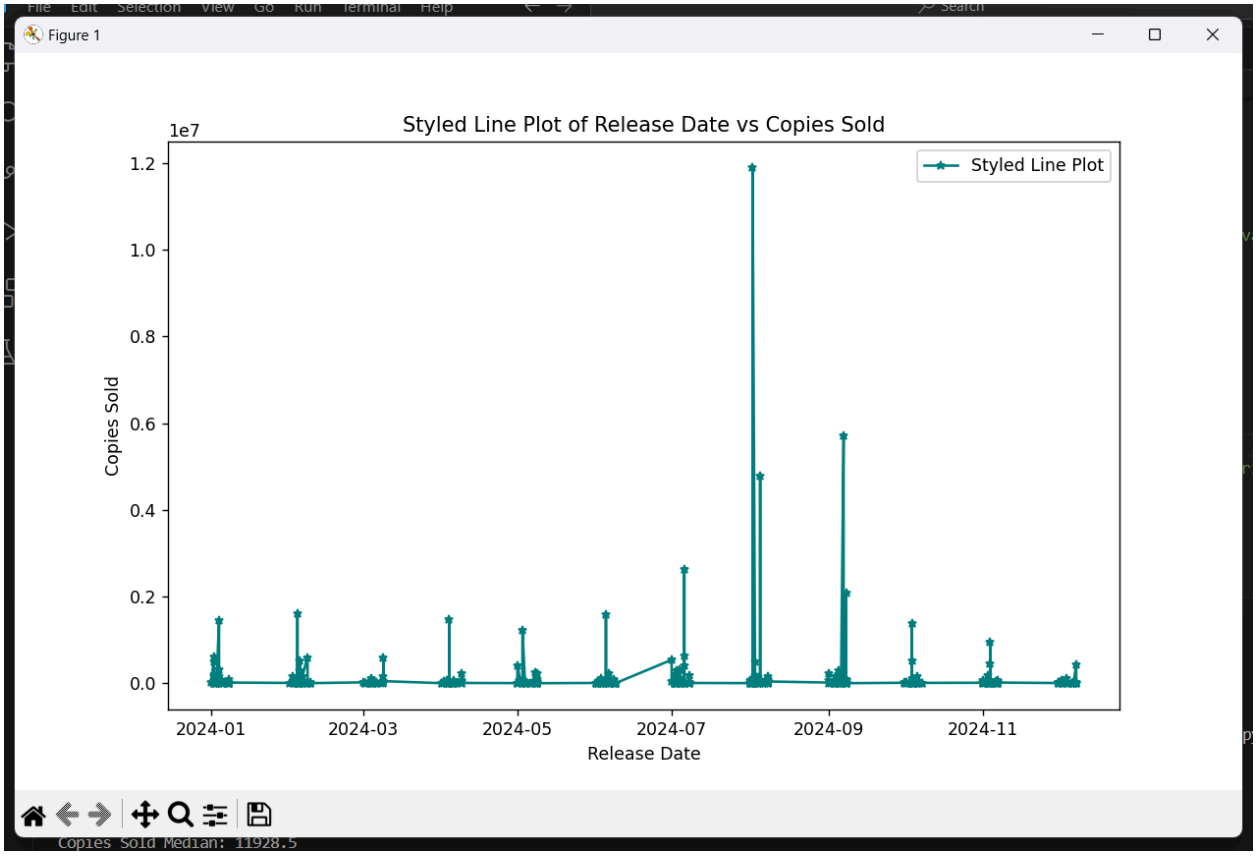


Figure 1
Styled Line Plot of Release Date vs Copies Sold
1e7
Styled Line Plot
Copies Sold
0.0
0.2
0.4
0.6
0.8
1.0
1.2
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
Release Date
Copies Sold Median: 11928.5

**Post Lab Question-Answers:**

**None.**

**Outcomes:**

**CO3:** Inculcate the knowledge of python libraries like NumPy, pandas, Matplotlib for scientific-computing and data visualization.

**Conclusion (based on the Results and outcomes achieved):**

Successfully applied python libraries on a dataset and executed the program.

**References:**

**Books/ Journals/ Websites referred:**

1. Reema Thareja, *Python Programming: Using Problem Solving Approach*, Oxford University Press, First Edition 2017, India
2. Sheetal Taneja and Naveen Kumar, *Python Programming: A modular Approach*, Pearson India, Second Edition 2018,India