

#Python Matplotlib Exercise (100 points)

This Matplotlib exercise project helps Python developers learn and practice data visualization using Matplotlib by solving multiple questions and problems.

In this exercise, we are using Pandas and Matplotlib to visualize Technological Products Data.

1. Use the following csv file for this exercise.
2. Read this file using Pandas or NumPy or using in-built matplotlib function.
3. Analyze the data and generate the appropriate plot including its properties:
 - a. Read each device's date of release and sales. Show it using a line plot.
 - b. Get the total sales and show line plot with the following style properties:
 - Line Style dotted and Line-color should be red
 - Show legend at the lower right location.
 - X label name = Year
 - Y label name = Sales
 - Add a circle marker.
 - Line marker color as read
 - Line width should be 3
 - c. Read all product sales data and show it using a multiline plot. Label them.
 - d. Read each brand sales data and show it using the bar chart. Label them.
 - e. Read each device and show it using the histogram to see the most common Country of Origin. Label them.
 - f. Create a pie chart that shows each brand Country of Origin. Label them.
1. Save your file into Matplotlib-< StudentName >.pdf

```
!pip install matplotlib
!pip install numpy
!pip install pandas
```

```
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
```

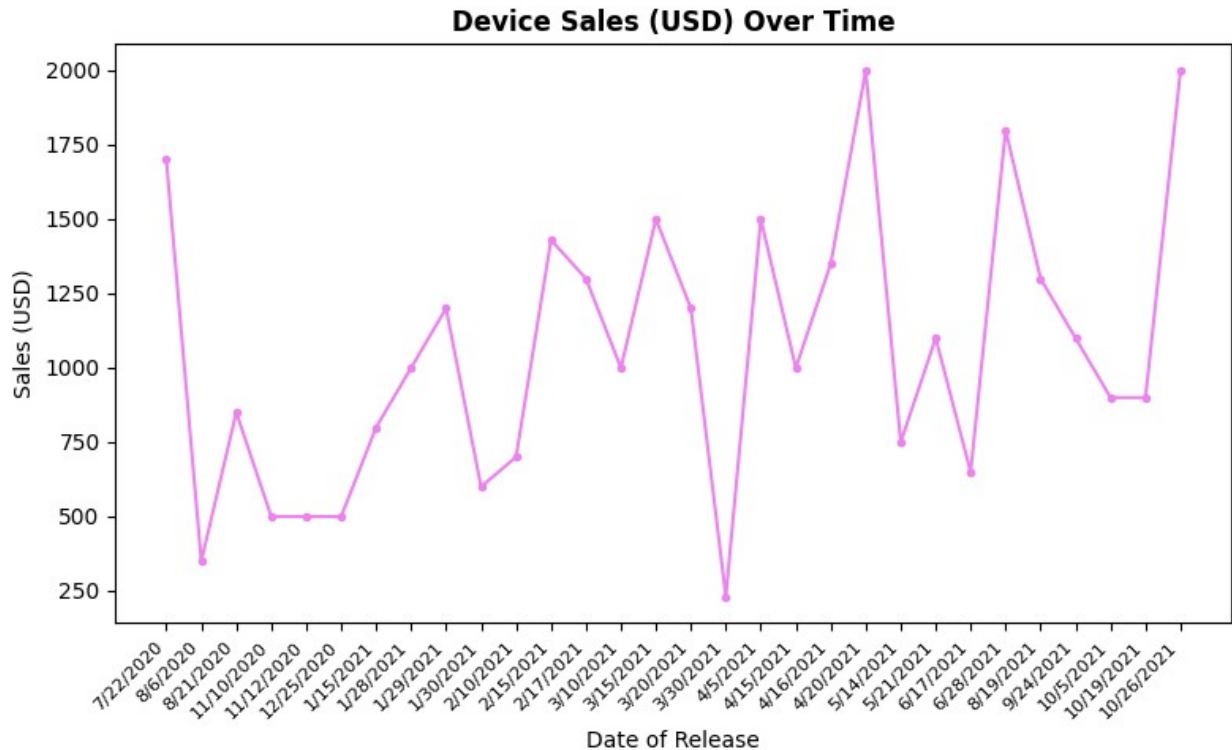
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: numpy>=1.20 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib) (1.16.0)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (1.26.4)
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.22.4 in
/usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas) (1.16.0)

```
import pandas as pd
```

```
from google.colab import drive  
drive.mount('/content/drive')  
file_path = '/content/drive/MyDrive/Technological-Products-Sample-  
Data.xlsx - Tech Products (Sort).csv'  
df = pd.read_csv(file_path, encoding='latin-1')  
df
```

Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).

```
{"summary":{"name": "df",  
"rows": 30,  
"fields": [  
  "column": "Brand",  
  "properties": {  
    "dtype": "category",  
    "num_unique_values": 10,  
    "samples": [  
      "HP",  
      "Sony",  
      "Acer"  
    ],  
    "semantic_type": ""  
  }  
]}
```

#b. Get the total sales and show line plot with the following style properties.

- Line Style dotted and Line-color should be red
- Show legend at the lower right location.
- X label name = Year
- Y label name = Sales
- Add a circle marker.
- Line marker color as red
- Line width should be 3

```
prices_by_year = df.groupby('Date of Release')['Sales (USD)'].sum()

plt.figure(figsize=(10, 6))
plt.plot(prices_by_year.index, prices_by_year.values, linestyle=':',
color='red', marker='o', mfc='red', linewidth=3, label='Total Sales (USD)')

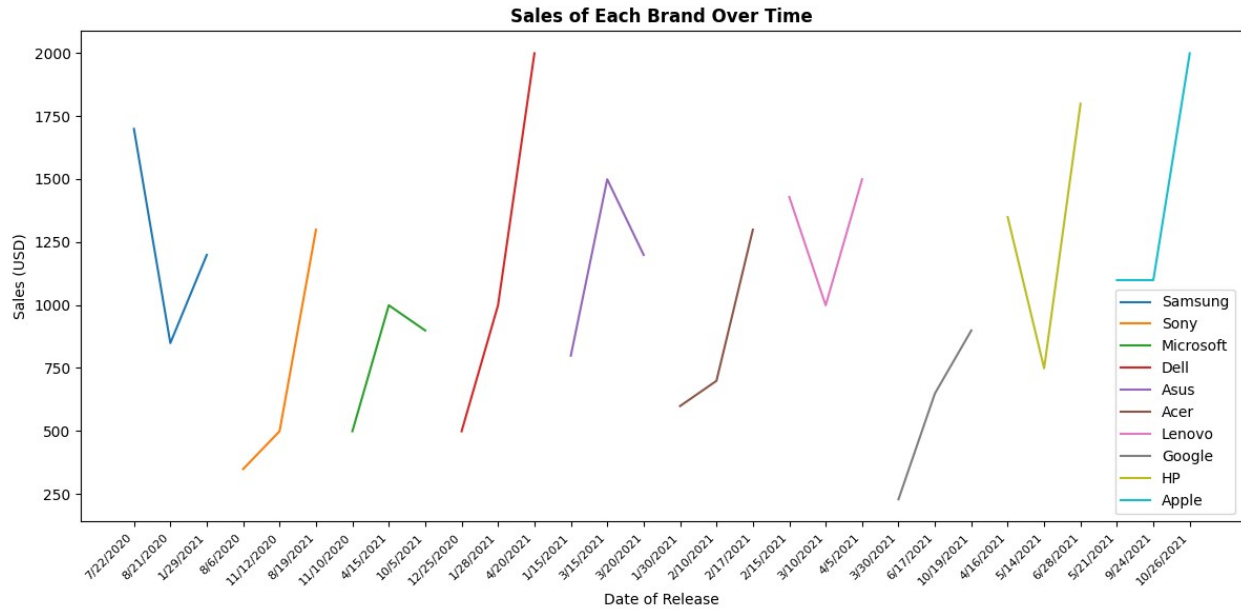
plt.xlabel('Year')
plt.ylabel('Sales')
plt.title('Prices Over Time', fontweight='bold')
plt.legend(loc='lower right')
plt.grid(axis='y', linestyle='-', linewidth=0.5)
plt.xticks(rotation=45, ha='right', fontsize=8)
plt.show()
```



#c. Read all product sales data and show it using a multiline plot. Label them.

```
plt.figure(figsize=(12, 6))
for brand in df['Brand'].unique():
    brand_data = df[df['Brand'] == brand]
    plt.plot(brand_data['Date of Release'], brand_data['Sales (USD)'],
    label=brand)

plt.xlabel('Date of Release')
plt.ylabel('Sales (USD)')
plt.title('Sales of Each Brand Over Time', fontweight='bold')
plt.legend()
plt.xticks(rotation=45, ha='right', fontsize=8)
plt.tight_layout()
plt.show()
```

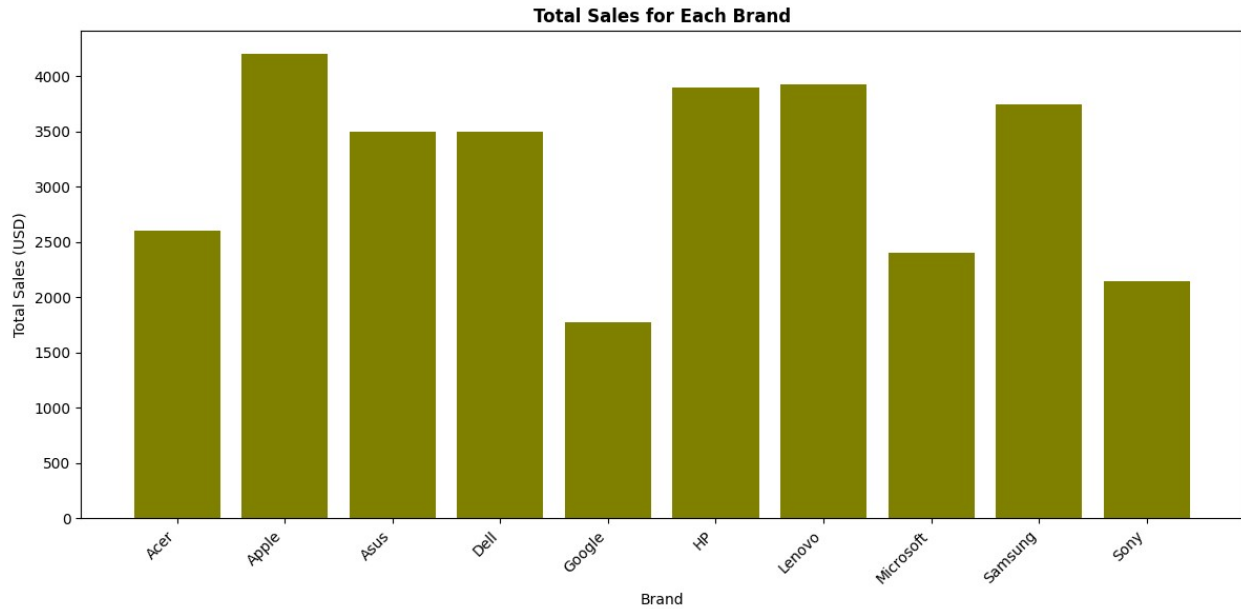


#d. Read each brand sales data and show it using the bar chart. Label them.

```
import matplotlib.pyplot as plt

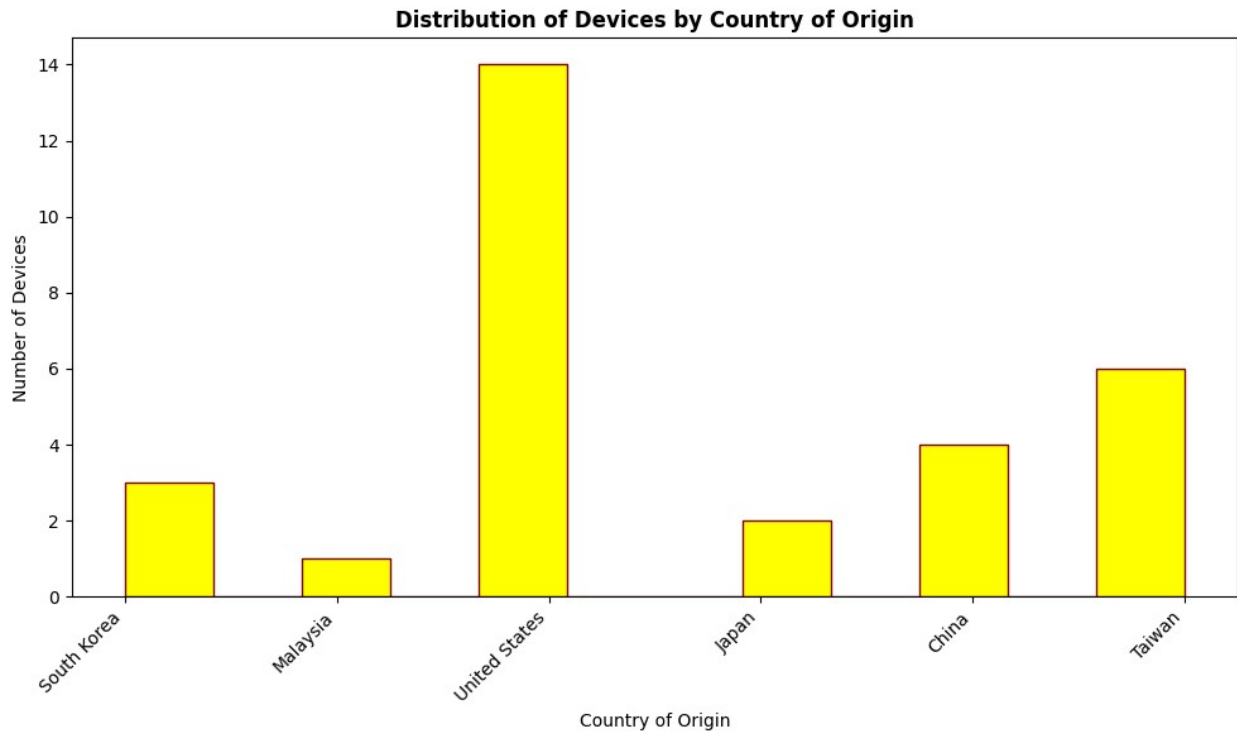
plt.figure(figsize=(12, 6))
brand_sales = df.groupby('Brand')['Sales (USD)'].sum()

plt.bar(brand_sales.index, brand_sales.values, color='olive')
plt.xlabel('Brand')
plt.ylabel('Total Sales (USD)')
plt.title('Total Sales for Each Brand', fontweight='bold')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



#e. Read each device and show it using the histogram to see the most common Country of Origin. Label them.

```
plt.figure(figsize=(10, 6))
plt.hist(df['Country of Origin'], bins=12, color='yellow',
edgecolor='maroon')
plt.xlabel('Country of Origin')
plt.ylabel('Number of Devices')
plt.title('Distribution of Devices by Country of Origin',
fontweight='bold')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



#f. Create a pie chart that shows each brand Country of Origin. Label them.

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo']
plt.figure(figsize=(12, 8))
country_origin_counts = df.groupby('Country of Origin')
['Brand'].count()
plt.pie(country_origin_counts, labels=country_origin_counts.index,
autopct='%1.1f%%', startangle=90, colors=colors)
plt.title('Distribution of Brands by Country of Origin',
fontweight='bold')
plt.axis('equal') #Equal aspect ratio to ensure pie is drawn as
circle
plt.legend(loc='lower right')
plt.show()
```


Distribution of Brands by Country of Origin

