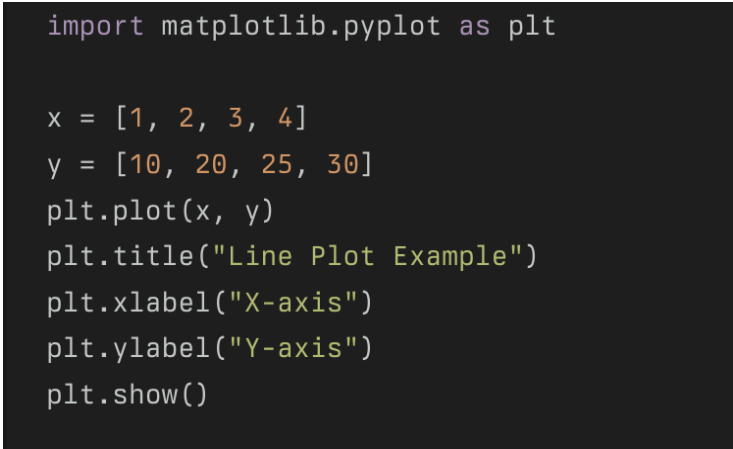
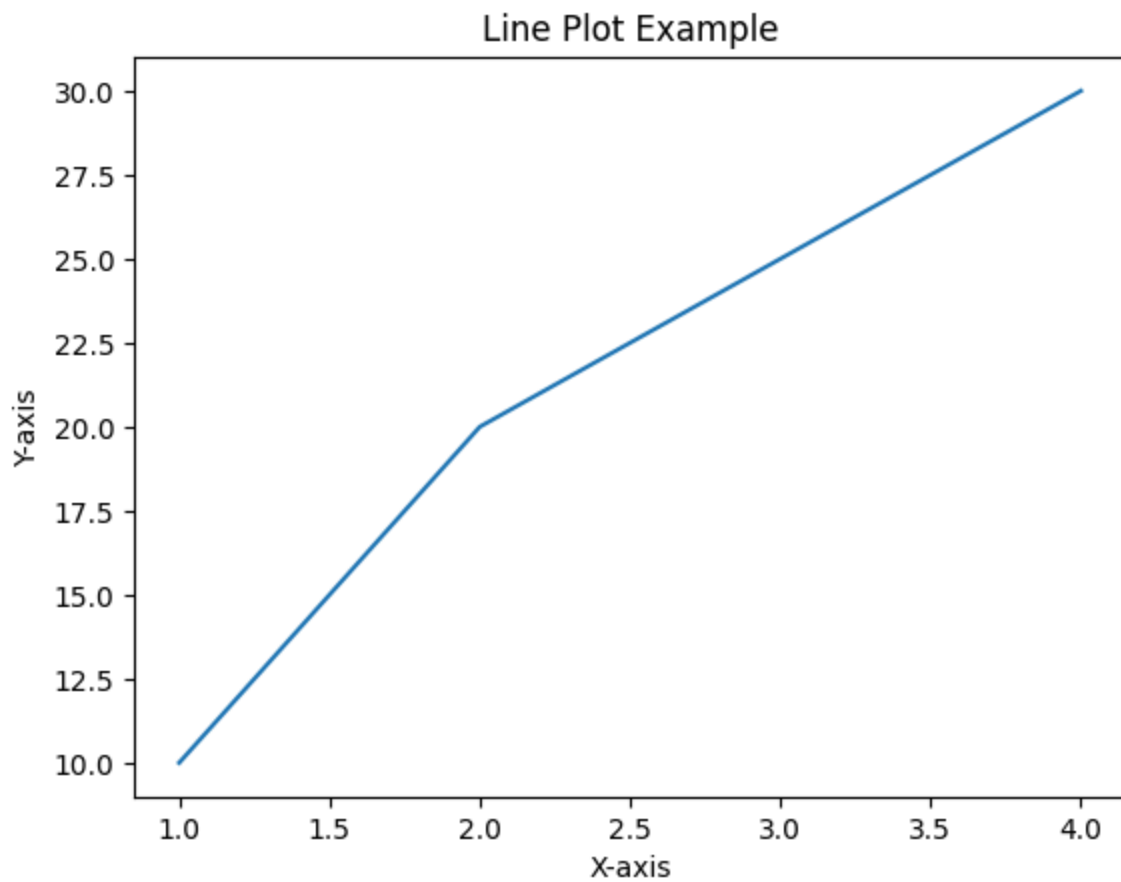


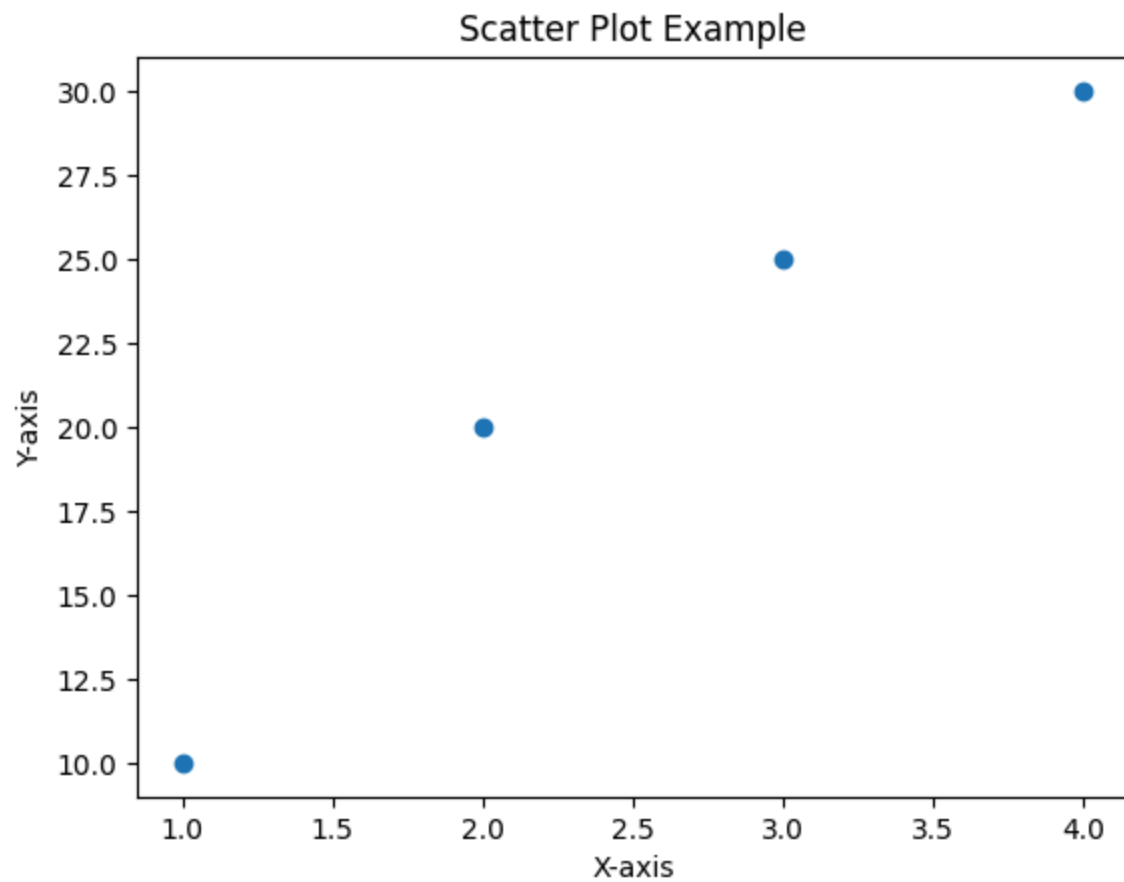
Introduction to Matplotlib	
<b>Course Code:</b> CPE 031	<b>Program:</b> Computer Engineering
<b>Course Title:</b> Visualization and Data Analysis	<b>Date Performed:</b> 10/22/24
<b>Section:</b> CPE21S4	<b>Date Submitted:</b> 10/22/24
<b>Name:</b> Magistrado, Aira Pauleen M.	<b>Instructor:</b> Maria Rizette Sayo
<b>Intended Learning Outcomes (ILO):</b>  By the end of this laboratory session, learners will be able to: <ol style="list-style-type: none"> <li>1. Utilize Matplotlib's pyplot interface to create a variety of visualizations, including line plots, scatter plots, histograms, and box plots, demonstrating an understanding of the library's syntax and functionality.</li> <li>2. Customize visual elements such as titles, labels, and legends to enhance the clarity and aesthetics of their plots, applying best practices in data visualization.</li> <li>3. Analyze and interpret visual data representations to extract meaningful insights, effectively communicating findings through well-structured graphical presentations.</li> </ol>	
<b>Part 1:</b> Perform the following codes, and understand the difference between line plot, scatter plot, histogram, bar chart, box plot, and pie chart using matplotlib's pyplot sub-module. <b>(Provide a screenshot of your output.)</b> <ol style="list-style-type: none"> <li>1. <b>Line Plot</b>  <pre>import matplotlib.pyplot as plt  x = [1, 2, 3, 4] y = [10, 20, 25, 30] plt.plot(x, y) plt.title("Line Plot Example") plt.xlabel("X-axis") plt.ylabel("Y-axis") plt.show()</pre> </li> </ol>	



## 2. Scatter Plot

```
import matplotlib.pyplot as plt

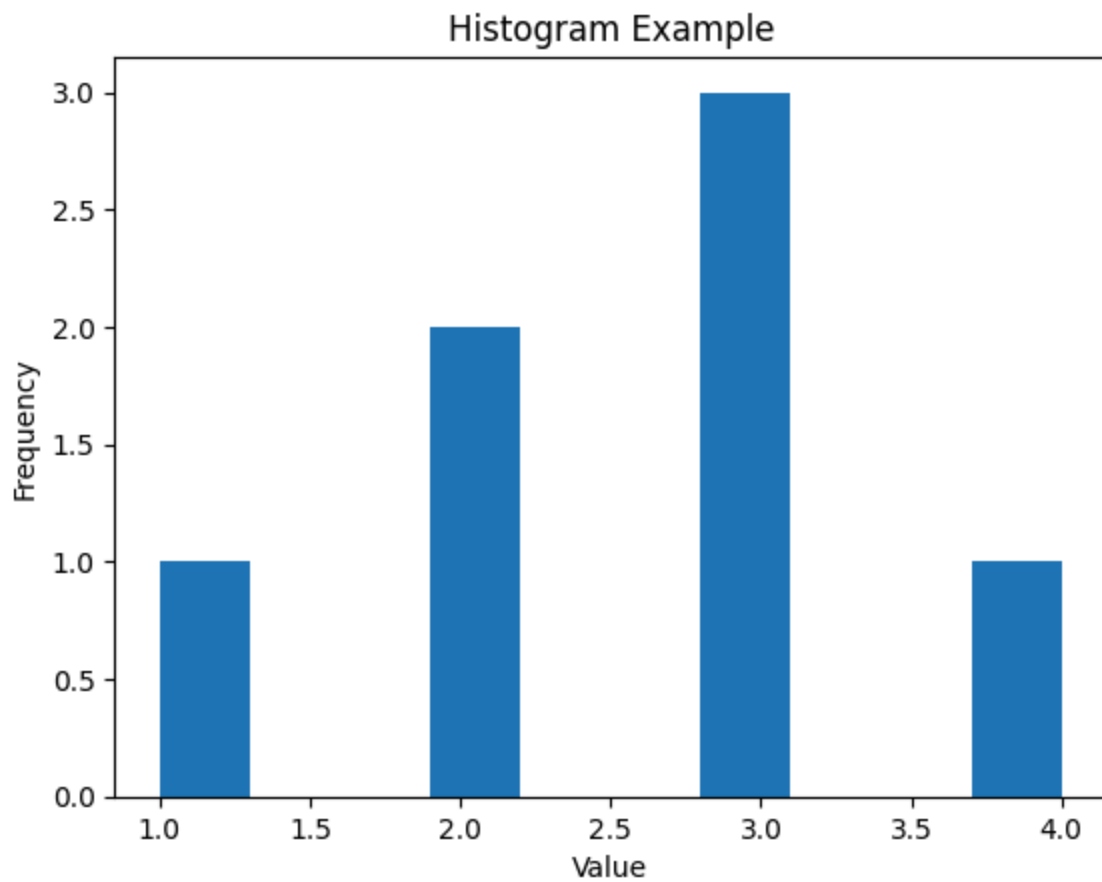
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
plt.scatter(x, y)
plt.title("Scatter Plot Example")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.show()
```



### 3. Histogram

```
import matplotlib.pyplot as plt

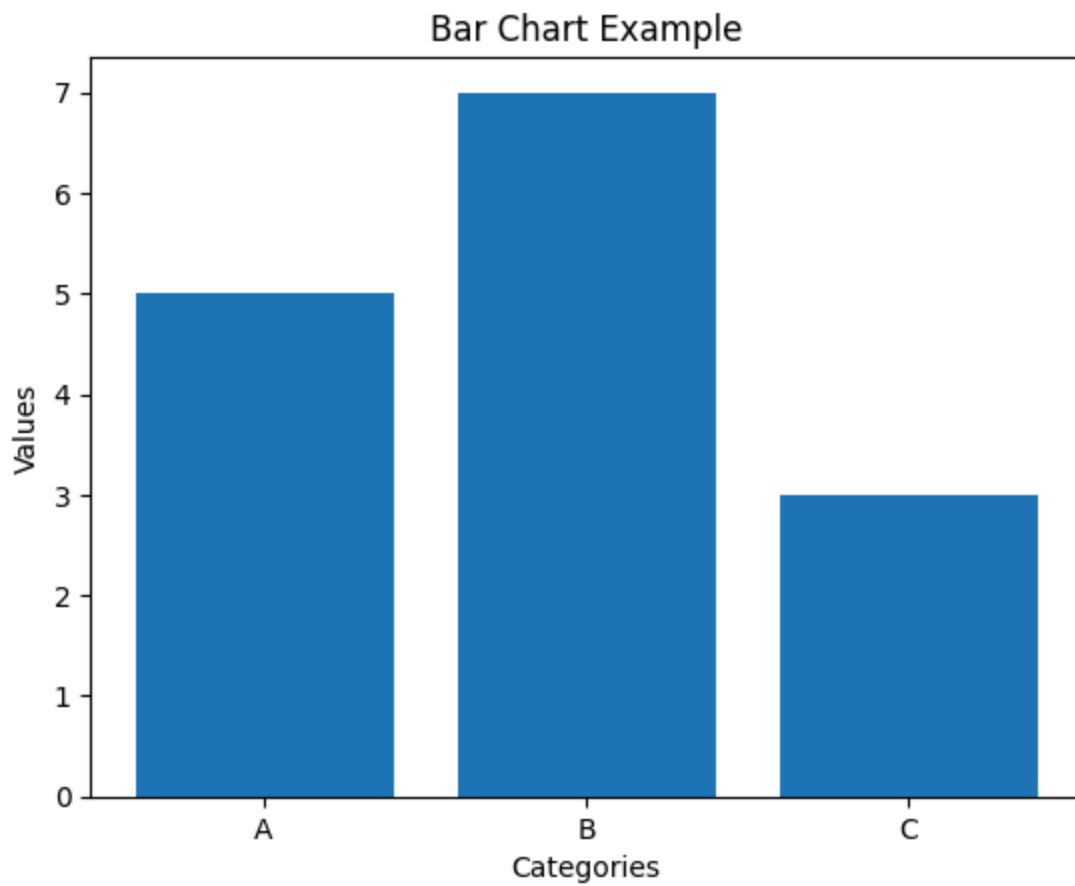
data = [1, 2, 2, 3, 3, 3, 4]
plt.hist(data)
plt.title("Histogram Example")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```



#### 4. Bar Chart

```
import matplotlib.pyplot as plt

categories = ['A', 'B', 'C']
values = [5, 7, 3]
plt.bar(categories, values)
plt.title("Bar Chart Example")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()
```



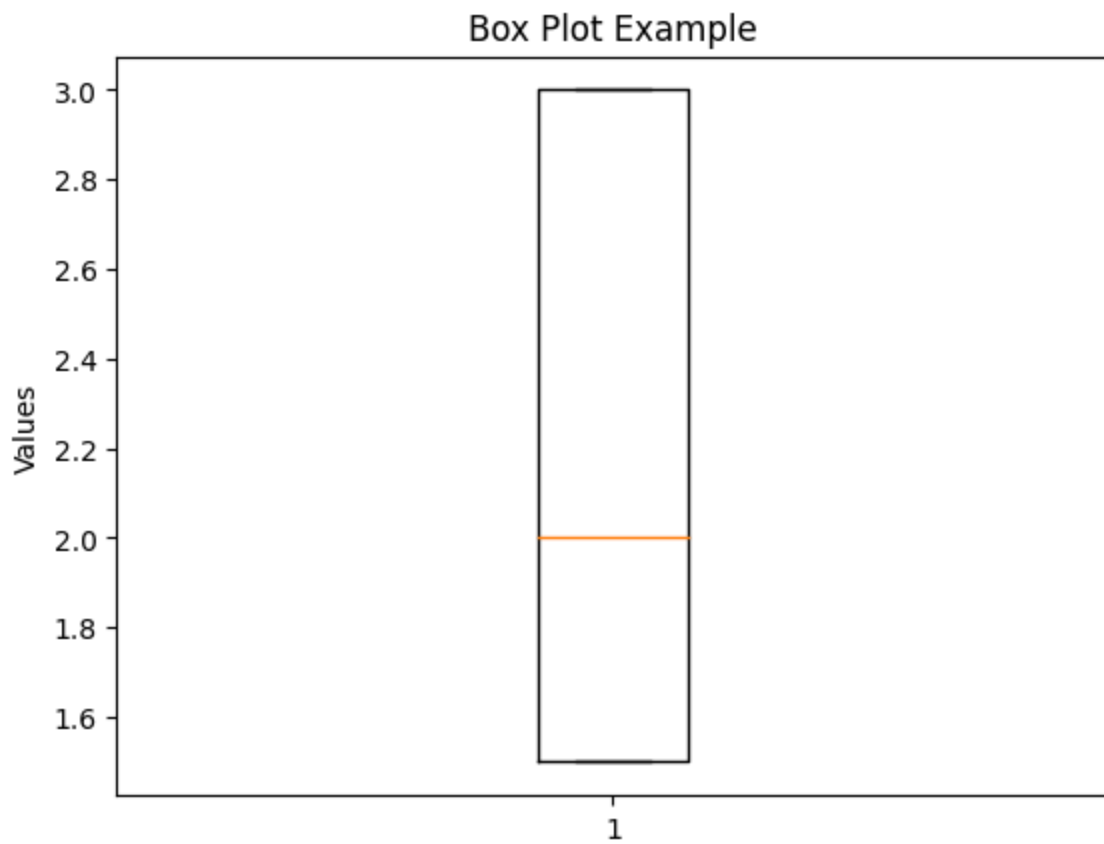
## 5. Box plot

```
import matplotlib.pyplot as plt

data = [[1.5]*10 + [2]*10 + [3]*10]

plt.boxplot(data)

plt.title("Box Plot Example")
plt.ylabel("Values")
plt.show()
```



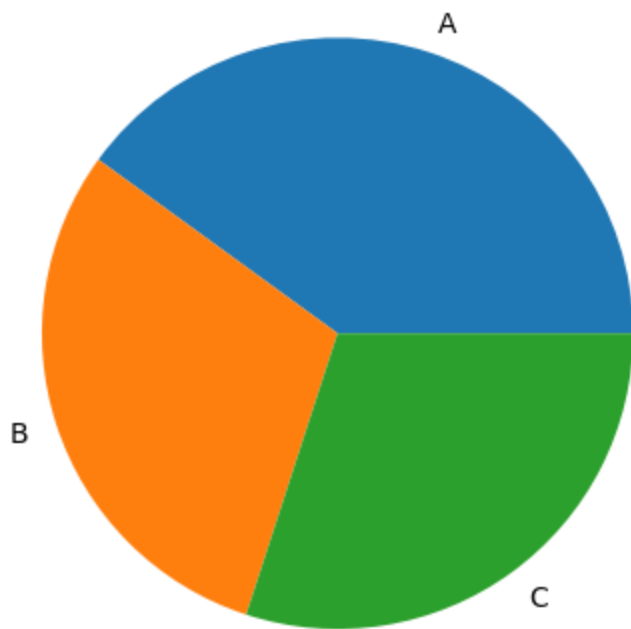
#### 6. Pie chart

```
import matplotlib.pyplot as plt

labels = ['A', 'B', 'C']
sizes = [40, 30, 30]

plt.pie(sizes, labels=labels)
plt.title("Pie Chart Example")
plt.show()
```

Pie Chart Example



## Part 2: Refer to the instructions below.

1. **Find a dataset for this activity:** Please visit Kaggle and look for a new dataset that would allow you to perform visualization and analysis using matplotlib.
2. **Creating a dataframe from your CSV file:** Once you have successfully loaded your dataset, you need to create a dataframe from your uploaded CSV file

```
[2] !pip install pandas

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)

[3] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[5] import pandas as pd

path="/content/drive/MyDrive/Dataset/top50.csv"
df=pd.read_csv(path)
df.head(51)
```

Unnamed: 0	Track Name	Artist Name	Genre	Beats.Per.Minute	Energy	Danceability	Loudness..db..	Liveness	Valence..	Length..	Acousticness..	Speechiness..	Popularity	
0	1	Seventi	Shawn Mendes	canadian pop	117	55	76	-6	8	75	191	4	3	79
1	2	China	Anuel AA	reggaeton flow	105	81	79	-4	8	61	302	8	9	92
2	3	boyfriend (with Social House)	Ariana Grande	dance pop	190	80	40	-4	16	70	186	12	46	85
3	4	Beautiful People (feat. Khalid)	Ed Sheeran	pop	93	65	64	-8	8	55	198	12	19	86
4	5	Goodbyes (Feat. Young Thug)	Post Malone	dhw rap	150	65	58	-4	11	18	175	45	7	94
5	6	I Don't Care (with Justin Bieber)	Ed Sheeran	pop	102	68	80	-5	9	84	220	9	4	84
6	7	Ransom	Lil Tecca	trap music	180	64	75	-6	7	23	131	2	29	92
7	8	How Do You Sleep?	Sam Smith	pop	111	68	48	-5	8	35	202	15	9	90
8	9	Old Town Road - Remix	Lil Nas X	country rap	136	62	88	-6	11	64	157	5	10	87
9	10	bad guy	Billie Eilish	electropop	135	43	70	-11	10	56	194	33	38	95

## 3. Import the matplotlib.pyplot

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

4. Based on your chosen dataset, you will develop three questions that you will answer using pyplot visualizations. This means that you will need to produce at least three pyplot visualizations. You are also required to make certain customizations on your data vizes.
5. Provide observations for each of your data viz, then **produce one insight not longer than five sentences given your three observations**. Your output shall follow this outline:
  - a. Introduction (Describe your dataset)
  - b. Questions
  - c. Visualization and Observation
  - d. Insight
6. Your grade will depend on the quality of the question, difficulty/complexity of the visualization, and value-add of the insight that you will generate.



## I. Introduction

The dataset that I chose contains the top 50 Spotify songs in 2019. It includes data on each Tracks name, artist, genre, beats per minute, energy level, danceability, loudness, liveliness, valence, length, acousticness, speechiness, and popularity.

## II. Questions

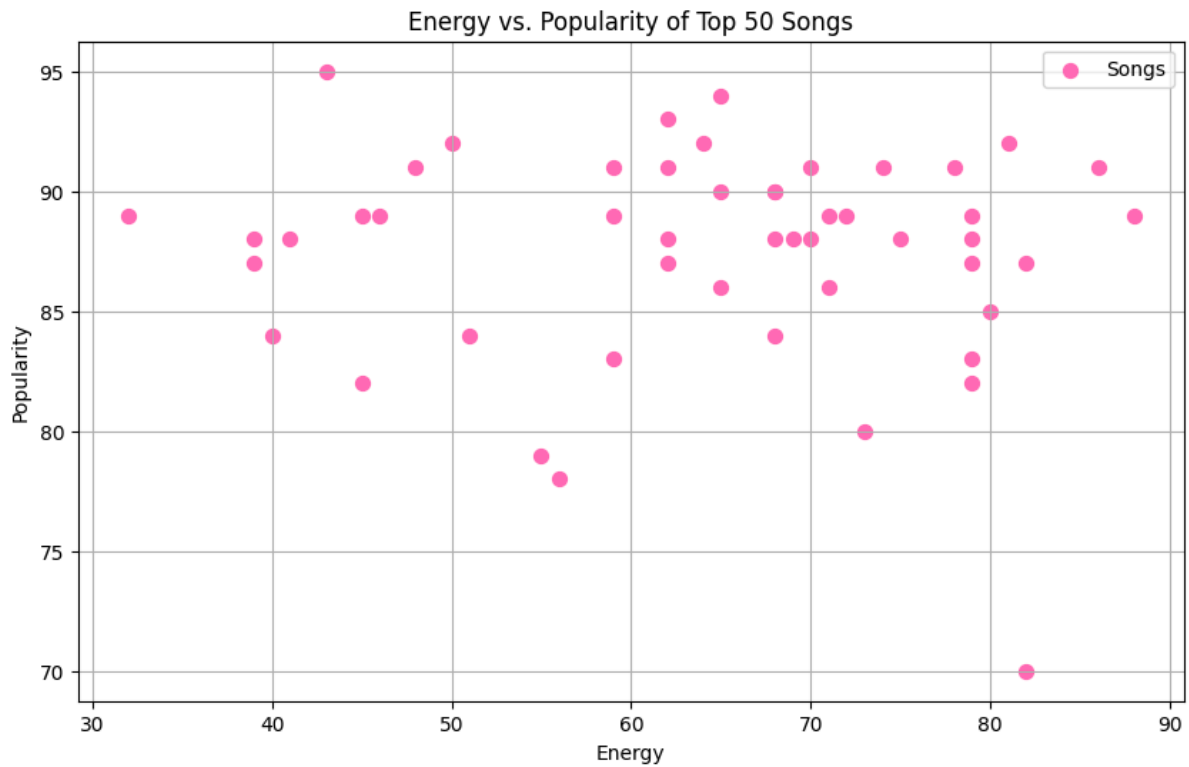
- 1) What is the relationship between a song's energy level and its popularity?
- 2) Which genres are most represented in the top 50 songs?
- 3) Do shorter songs tend to be more danceable than longer songs?

## III. Visualizations and Observations

1)

Python

```
plt.figure(figsize=(10, 6))
print(df.columns)
plt.scatter(df['Energy'], df['Popularity'], s=50, c='hotpink', label='Songs')
plt.legend()
plt.xlabel('Energy')
plt.ylabel('Popularity')
plt.title('Energy vs. Popularity of Top 50 Songs')
plt.grid()
plt.show()
```

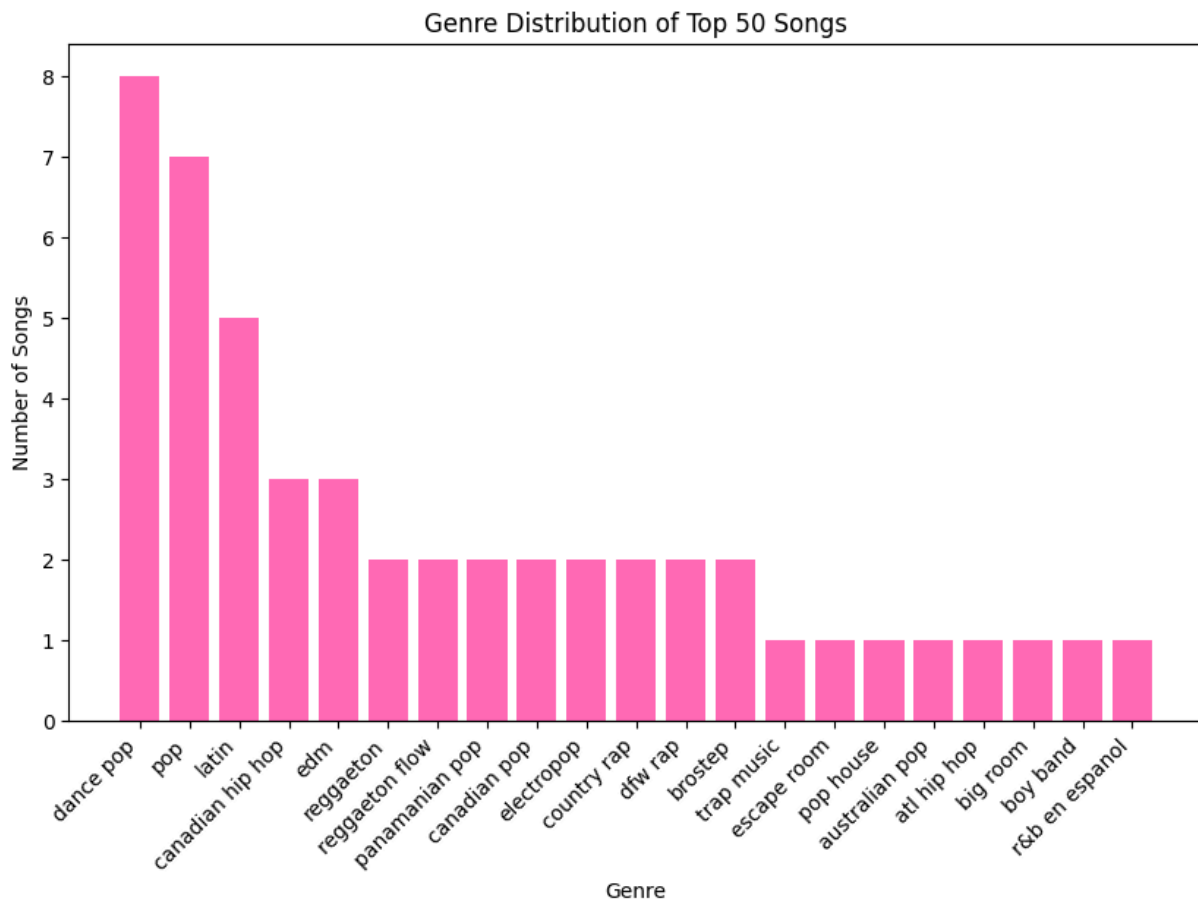


**Observation:** The scatter plot shows a slight positive correlation between the song's energy level and popularity. It can be concluded that songs with higher energy levels are more popular but the relationship between them isn't very strong.

2)

Python

```
genre_counts = df['Genre'].value_counts()
plt.figure(figsize=(10, 6))
plt.bar(genre_counts.index, genre_counts.values, color='hotpink')
plt.xlabel('Genre')
plt.ylabel('Number of Songs')
plt.title('Genre Distribution of Top 50 Songs')
plt.xticks(rotation=45, ha='right')
plt.show()
```



**Observation:** The bar chart shows that dance pop, pop, and latin hip hop are the most common genres in the top 50 lists. It can be concluded that these genres are the most popular back in 2019.

3)

Python

```
plt.figure(figsize=(10, 6))
plt.scatter(df['Danceability'], df['Length.'], s=50, c='hotpink',
label='Songs') /
plt.legend()
plt.xlabel('Danceability')
plt.ylabel('Length (seconds)')
plt.title('Danceability vs. Length of Top 50 Songs')
plt.grid()
plt.show()
```

