Lab1

September 2, 2023

[2]:
```
import pandas as pd
import numpy as np

# Create a Pandas DataFrame from a NumPy array
data = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
df = pd.DataFrame(data, columns=['A', 'B', 'C'])

# Accessing and manipulating data with Pandas
print("Original DataFrame:")
print(df)

# You can perform operations on DataFrame columns using NumPy functions
df['D'] = np.sqrt(df['A'])  # Adding a new column 'D' with square root of 'A'

# Using NumPy to perform element-wise operations
df['E'] = np.where(df['B'] > 4, 'High', 'Low')  # Adding a new column based on a␣
↪condition

# Filtering data using NumPy
filtered_data = df[df['C'] > 5]  # Select rows where 'C' is greater than 5

print("\nModified DataFrame:")
print(df)

print("\nFiltered DataFrame:")
print(filtered_data)
```

```
Original DataFrame:
   A  B  C
0  1  2  3
1  4  5  6
2  7  8  9

Modified DataFrame:
   A  B  C         D     E
0  1  2  3  1.000000   Low
1  4  5  6  2.000000  High
2  7  8  9  2.645751  High
```

```
Filtered DataFrame:
   A  B  C         D     E
1  4  5  6  2.000000  High
2  7  8  9  2.645751  High
```

[3]:
```python
# Create an empty graph (dictionary)
graph = {}

# Add nodes to the graph
nodes = [1, 2, 3]
for node in nodes:
    graph[node] = []

# Add edges to the graph
edges = [(1, 2), (2, 3), (1, 3)]  # Example edges between nodes
for edge in edges:
    node1, node2 = edge
    graph[node1].append(node2)
    graph[node2].append(node1)

# Print the graph
for node, neighbors in graph.items():
    print(f"Node {node} is connected to nodes: {neighbors}")
```

```
Node 1 is connected to nodes: [2, 3]
Node 2 is connected to nodes: [1, 3]
Node 3 is connected to nodes: [2, 1]
```

[2]:
```python
import numpy as np
a=print("enter number of inputs")
# b=int(input())
ls=[1, 2, 3, 4, -5, -6]
# for i in range(b):
#   ls.append(int(input()))
print(ls)

# print(ls[::-1])
ls=np.array(ls[::-1],dtype="f")
print(ls)
```

```
enter number of inputs
[1, 2, 3, 4, -5, -6]
[-6. -5.  4.  3.  2.  1.]
```

[3]:
```python
ls=[1,2,3,4,5,6,7,8,9]
ls=np.array(ls)
ls=np.reshape(ls,(3,3))
```

```
print(ls)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

[4]:
```
sha=list(map(int,input().split()))
zero=np.zeros((sha[0],sha[1]))
print(zero)
```

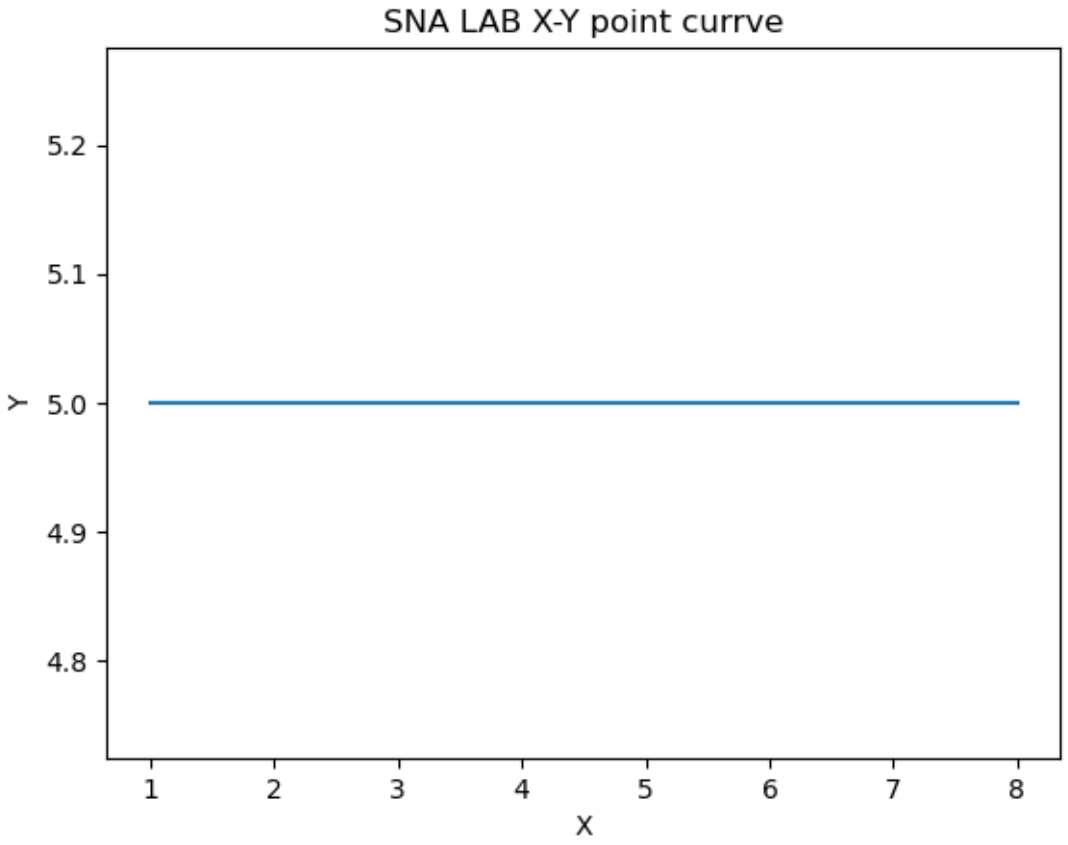
```
 1 2 3 4 5 6
[[0. 0.]]
```

[16]:
```
a=[[2,3,4],[5,6,7],[8,9,10]]
b=[[1,2,3],[4,5,6],[7,8,9]]
a=np.array(a)
b=np.array(b)

print(a+b)
print(a-b)
print(a*b)
print(a/b)
print(a%b)
print(a**b)
```

```
[[ 3  5  7]
 [ 9 11 13]
 [15 17 19]]
[[1 1 1]
 [1 1 1]
 [1 1 1]]
[[ 2  6 12]
 [20 30 42]
 [56 72 90]]
[[2.         1.5        1.33333333]
 [1.25       1.2        1.16666667]
 [1.14285714 1.125      1.11111111]]
[[0 1 1]
 [1 1 1]
 [1 1 1]]
[[         2          9         64]
 [       625       7776     117649]
 [   2097152   43046721 1000000000]]
```

[17]:
```
import numpy as np
import matplotlib.pyplot as plt

# Dataset
```

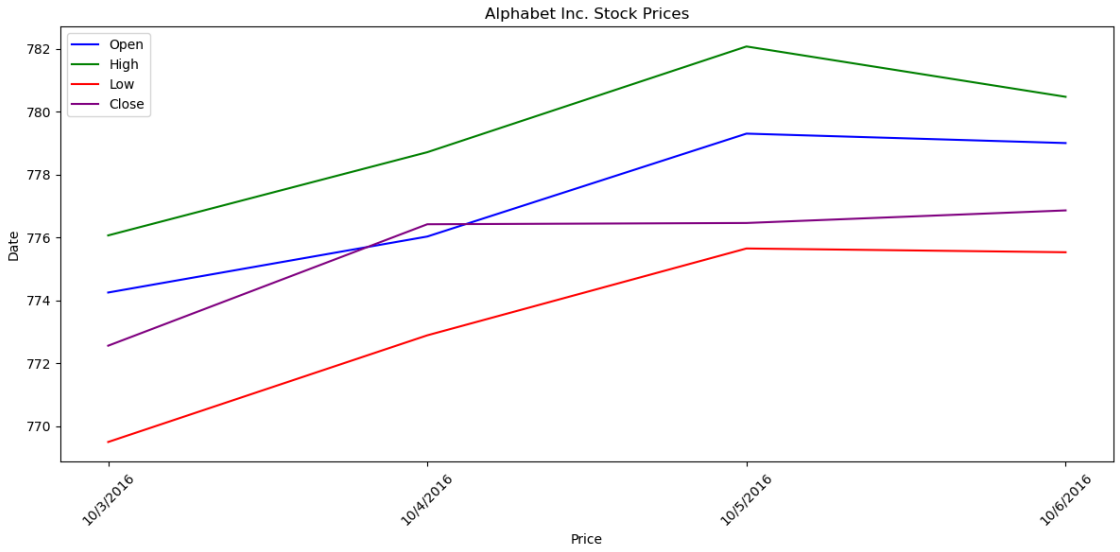
```
x = np.array([ 1,  2,  3,  4,  5,  6,  7, 8 ])
y = np.array([ 5, 5, 5, 5, 5, 5, 5, 5 ])

# Plotting the Graph
plt.plot(x, y)
plt.title("SNA LAB X-Y point currve")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```



[30]:
```
import matplotlib.pyplot as plt
import pandas as pd

# Load the CSV file into a Pandas DataFrame
data = pd.read_csv('/Users/hemant/Downloads/fdata - fdata.csv')

print(data)

# Extract data for plotting
date = data['Date']
```

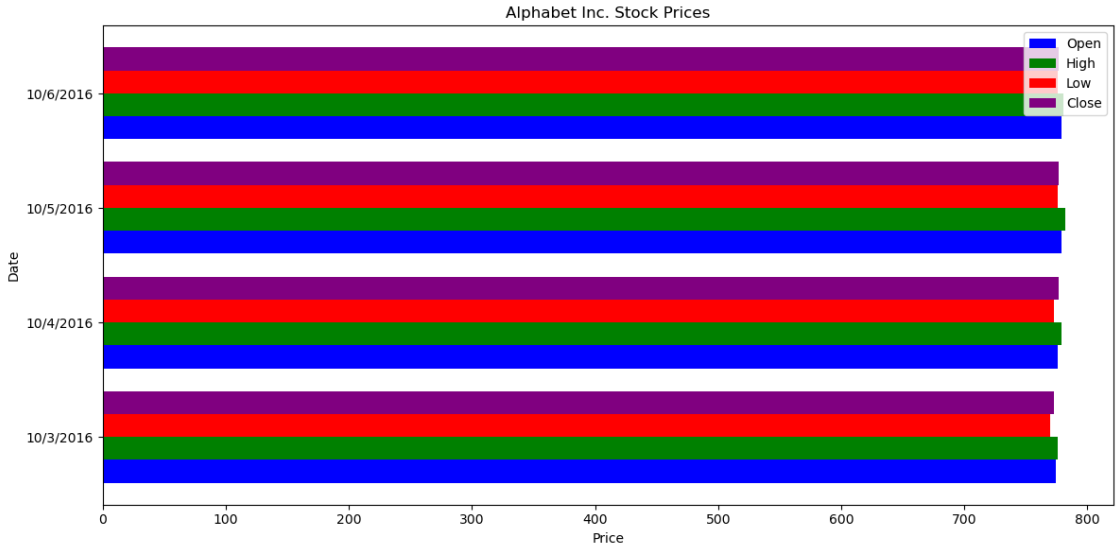
```
open_prices = data['Open']
high_prices = data['High']
low_prices = data['Low']
close_prices = data['Close']

# Create a line chart with different colors for open, high, low, and close
plt.figure(figsize=(12, 6))
plt.plot(date, open_prices, color='blue', label='Open')
plt.plot(date, high_prices, color='green', label='High')
plt.plot(date, low_prices, color='red', label='Low')
plt.plot(date, close_prices, color='purple', label='Close')

# Set labels and title
plt.ylabel('Date')
plt.xlabel('Price')
plt.title('Alphabet Inc. Stock Prices')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Add a legend
plt.legend()

# Show the plot
plt.tight_layout()
plt.show()
```

	Date	Open	High	Low	Close
0	10/3/2016	774.25	776.065002	769.50	772.559998
1	10/4/2016	776.03	778.710000	772.89	776.420000
2	10/5/2016	779.30	782.070000	775.65	776.460000
3	10/6/2016	779.00	780.470000	775.53	776.859950



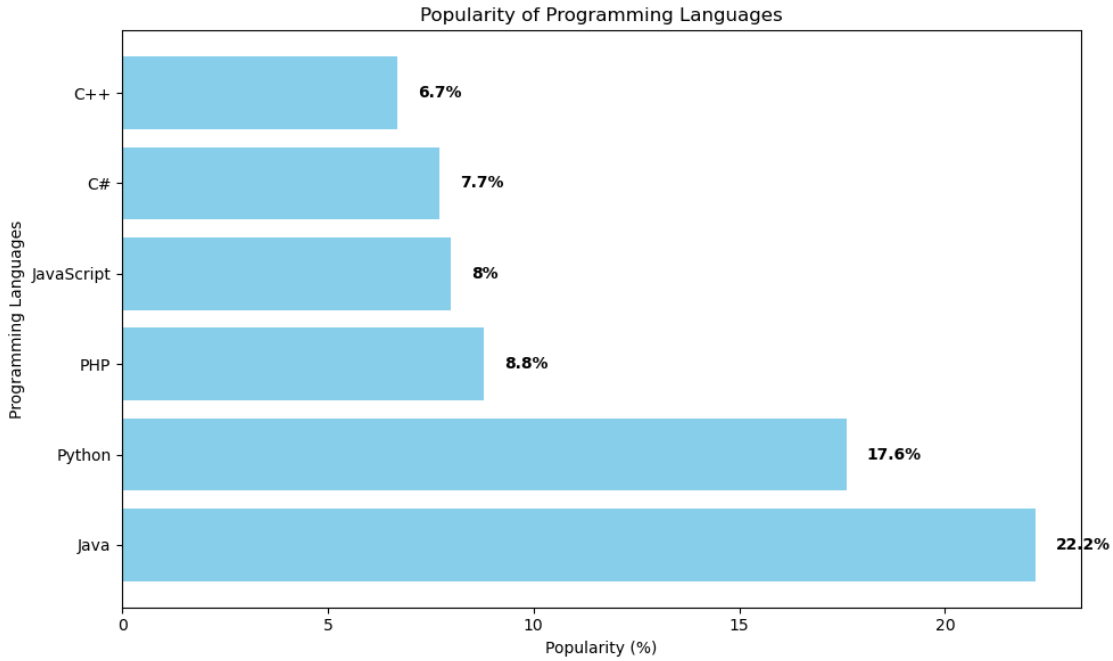
[31]:
```
import matplotlib.pyplot as plt
import pandas as pd

# Load the CSV file into a Pandas DataFrame
data = pd.read_csv('/Users/hemant/Downloads/fdata - fdata.csv')

# Extract data for plotting
date = data['Date']
open_prices = data['Open']
high_prices = data['High']
low_prices = data['Low']
close_prices = data['Close']

# Create a bar chart with different colors for open, high, low, and close
plt.figure(figsize=(12, 6))
bar_width = 0.2
index = range(len(date))

plt.barh(index, open_prices, bar_width, color='blue', label='Open')
plt.barh([i + bar_width for i in index], high_prices, bar_width, color='green',␣
 ↪label='High')
plt.barh([i + 2 * bar_width for i in index], low_prices, bar_width, color='red',␣
 ↪label='Low')
plt.barh([i + 3 * bar_width for i in index], close_prices, bar_width,␣
 ↪color='purple', label='Close')

# Set labels and title
plt.ylabel('Date')
```

```
plt.xlabel('Price')
plt.title('Alphabet Inc. Stock Prices')

# Add a legend
plt.legend()

# Show the plot
plt.yticks([i + 1.5 * bar_width for i in index], date)
plt.tight_layout()
plt.show()
```



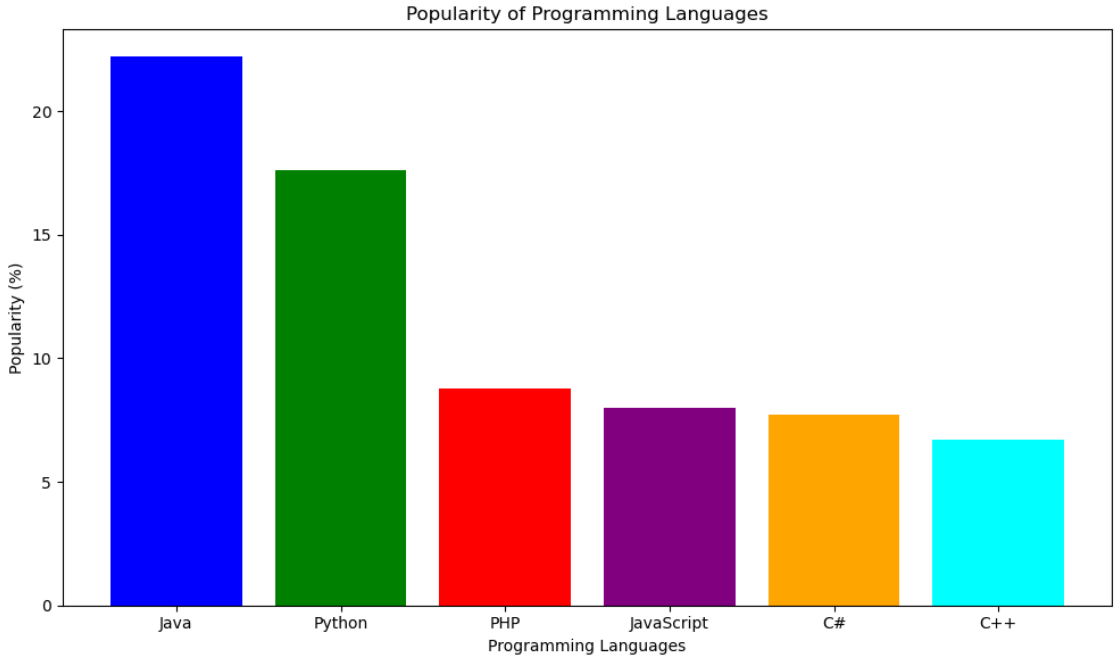
[32]:
```
import matplotlib.pyplot as plt

# Sample data
languages = ["Java", "Python", "PHP", "JavaScript", "C#", "C++"]
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

# Create a horizontal bar chart
plt.figure(figsize=(10, 6))
plt.barh(languages, popularity, color='skyblue')

# Set labels and title
plt.xlabel('Popularity (%)')
plt.ylabel('Programming Languages')
plt.title('Popularity of Programming Languages')

# Add values next to the bars
for i, val in enumerate(popularity):
```

```
    plt.text(val + 0.5, i, f'{val}%', va='center', color='black',␣
 ↪fontweight='bold')

# Show the plot
plt.tight_layout()
plt.show()
```



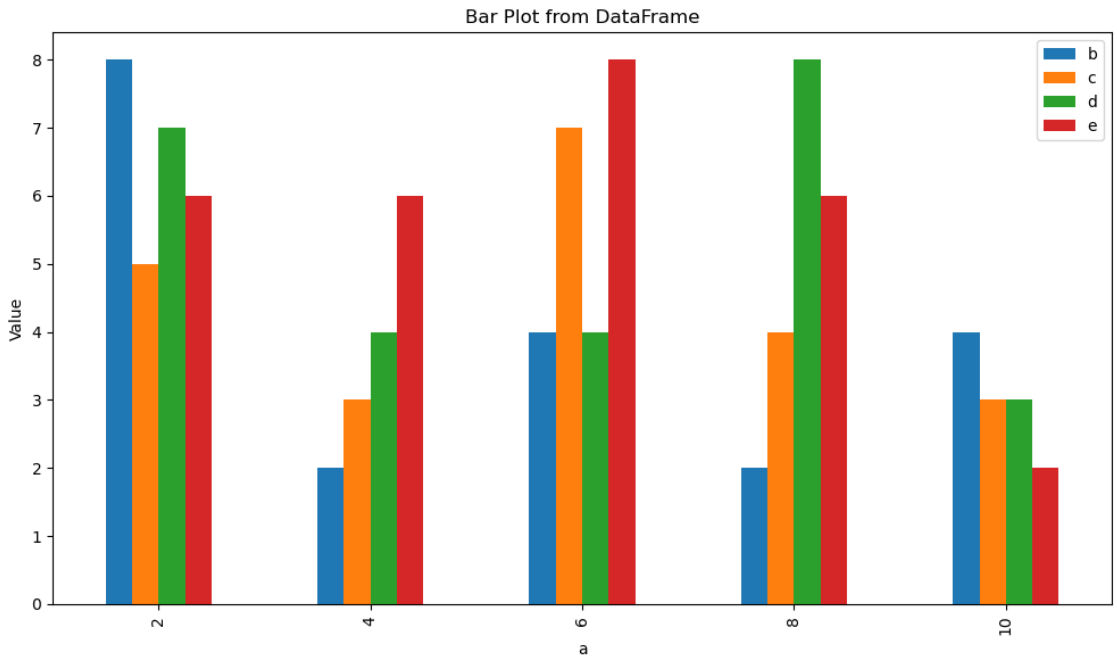
[33]:
```
import matplotlib.pyplot as plt

# Sample data
languages = ["Java", "Python", "PHP", "JavaScript", "C#", "C++"]
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

# Define colors for each bar
colors = ['blue', 'green', 'red', 'purple', 'orange', 'cyan']

# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(languages, popularity, color=colors)

# Set labels and title
plt.xlabel('Programming Languages')
plt.ylabel('Popularity (%)')
plt.title('Popularity of Programming Languages')
```

```
# Show the plot
plt.tight_layout()
plt.show()
```



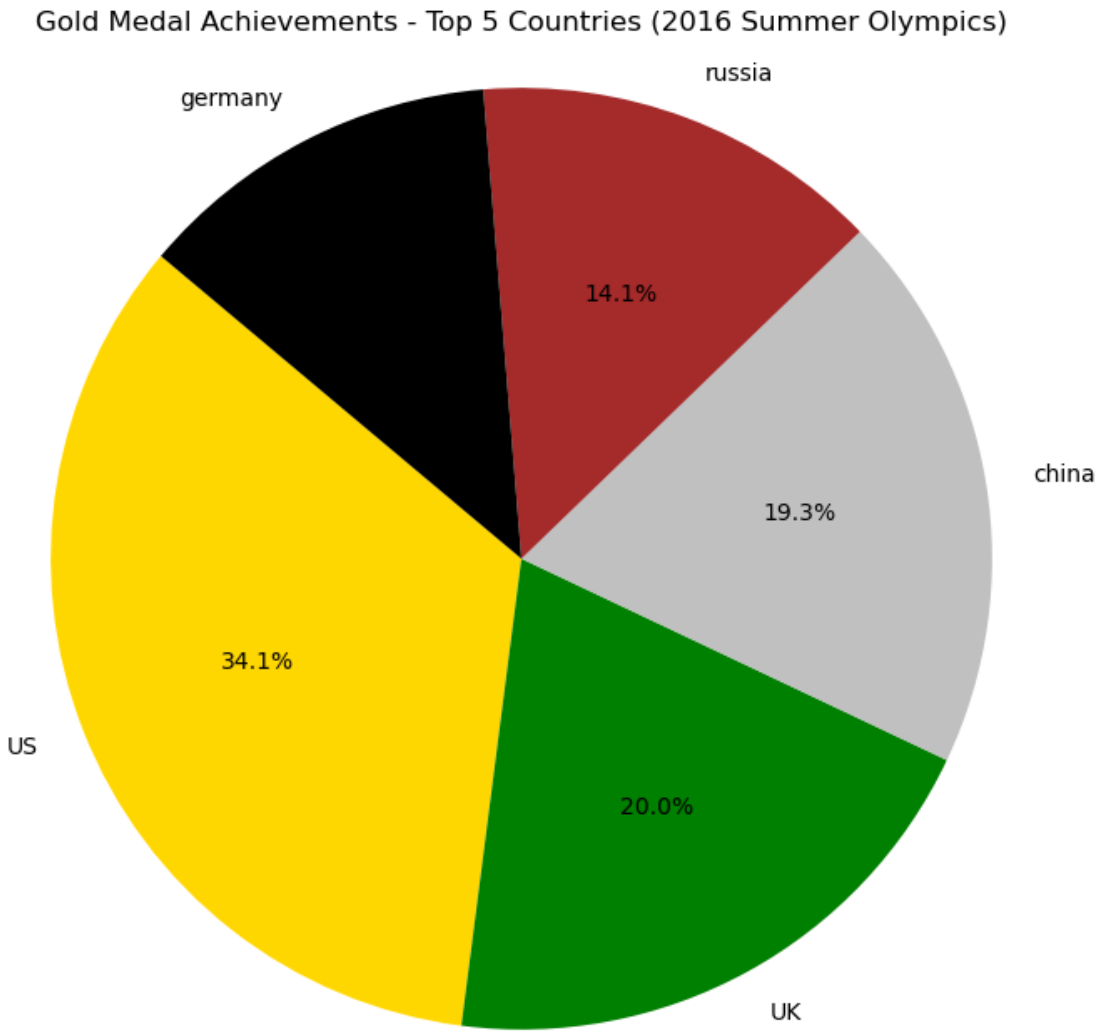
[34]:
```
import matplotlib.pyplot as plt
import pandas as pd

# Sample DataFrame
data = {'a': [2, 4, 6, 8, 10],
        'b': [8, 2, 4, 2, 4],
        'c': [5, 3, 7, 4, 3],
        'd': [7, 4, 4, 8, 3],
        'e': [6, 6, 8, 6, 2]}

df = pd.DataFrame(data)

# Set the index column (optional, use if you want 'a' as the x-axis labels)
df.set_index('a', inplace=True)

# Create a bar plot
df.plot(kind='bar', figsize=(10, 6))

# Set labels and title
plt.xlabel('a')
plt.ylabel('Value')
```

```
plt.title('Bar Plot from DataFrame')

# Show the plot
plt.tight_layout()
plt.show()
```



[40]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Read data from the CSV file
df = pd.read_csv('/Users/hemant/Downloads/medal - Sheet1.csv')

# Sort the DataFrame by gold medals in descending order and select the top 5␣
 ↪countries
df.sort_values(by='Medal', ascending=False, inplace=True)
top_countries = df.head(5)

# Extract data for the pie chart
countries = top_countries['Country']
gold_medals = top_countries['Medal']

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(gold_medals, labels=countries, autopct='%1.1f%%', startangle=140,␣
 ↪colors=['gold', 'green', 'silver', 'brown','black'])
```

```
# Set title
plt.title('Gold Medal Achievements - Top 5 Countries (2016 Summer Olympics)')

# Show the plot
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



[ ]: