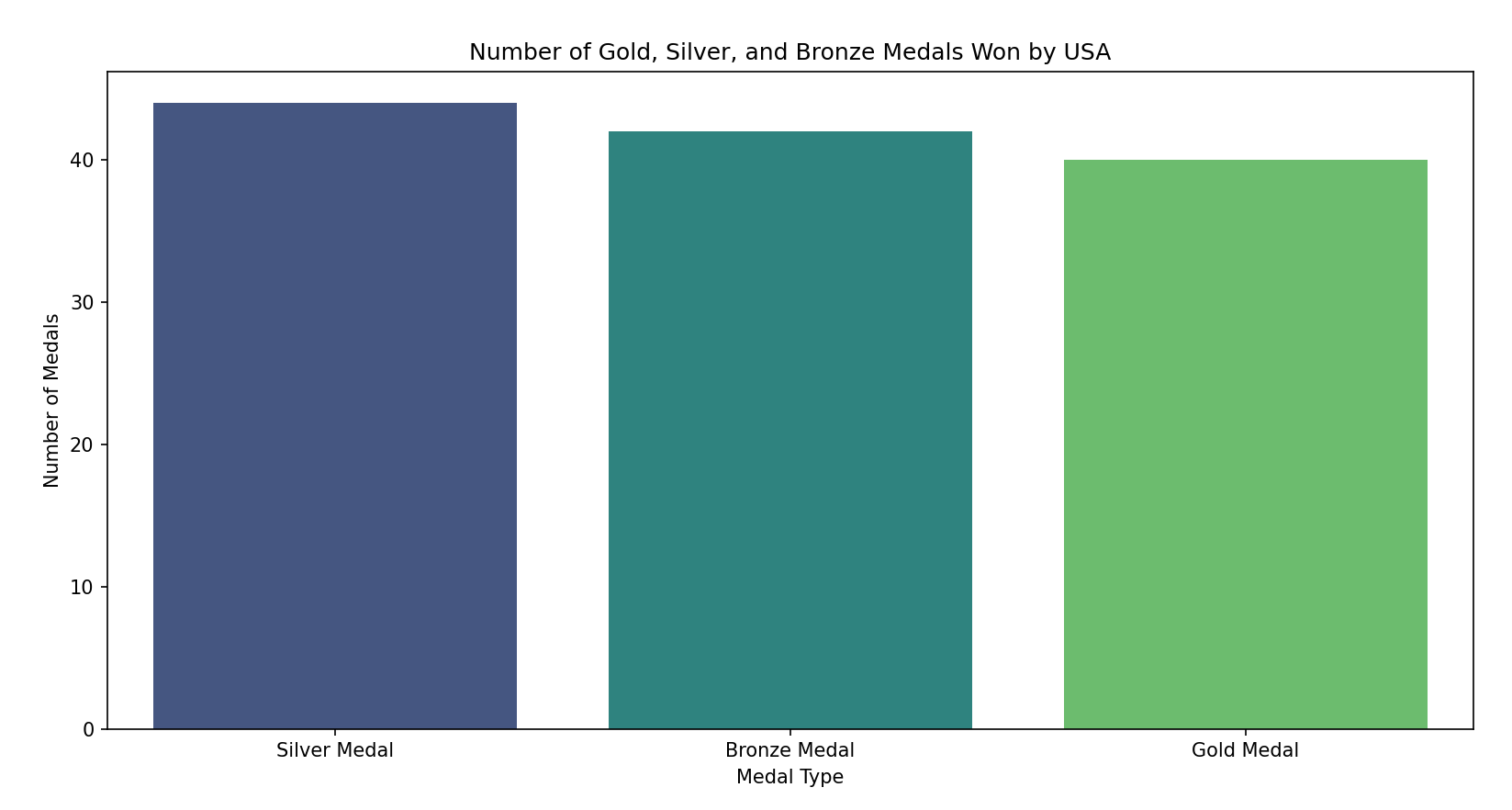
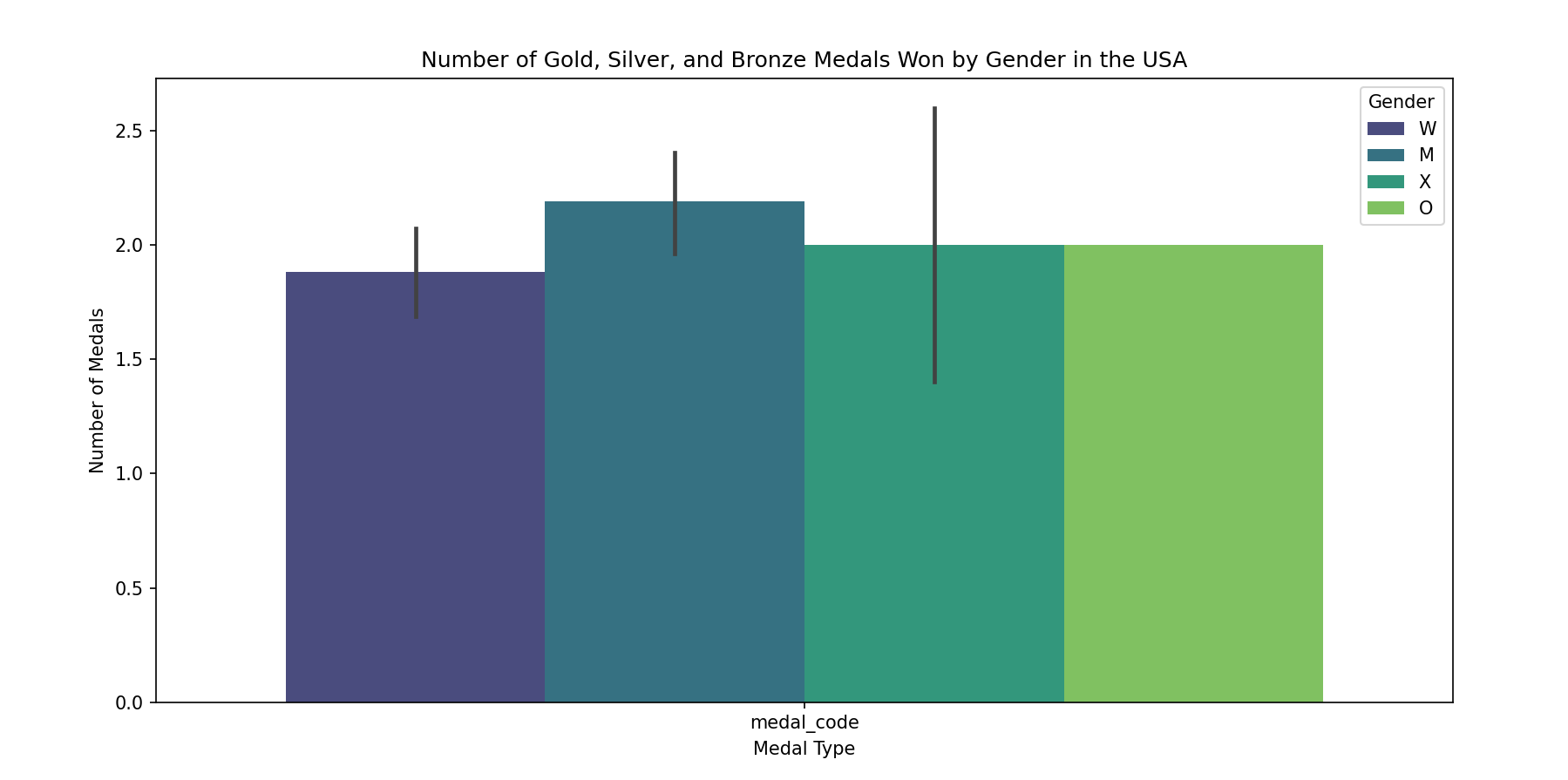
<https://www.kaggle.com/datasets/piterfm/paris-2024-olympic-summer-games?resource=download>

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# Load the CSV file into a DataFrame  
csv\_file\_path = "medals.csv" # Update this path if necessary  
df = pd.read\_csv(csv\_file\_path)  
  
# Filter for USA and count medals  
usa\_medals\_df = df[df['country\_code'] == 'USA']  
medal\_counts = usa\_medals\_df['medal\_type'].value\_counts()  
  
# Convert to a DataFrame for plotting  
medal\_counts\_df = pd.DataFrame(medal\_counts).reset\_index()  
medal\_counts\_df.columns = ['medal\_type', 'medal\_code']  
  
  
  
# Plotting  
plt.figure(figsize=(8, 6))  
sns.barplot(data=medal\_counts\_df, x='medal\_type', y='medal\_code', palette='viridis')  
plt.xlabel('Medal Type')  
plt.ylabel('Number of Medals')  
plt.title('Number of Gold, Silver, and Bronze Medals Won by USA')  
plt.show()



Calculate games, Now calculate gender

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# Load the CSV file into a DataFrame  
csv\_file\_path = "medals.csv" # Update this path if necessary  
df = pd.read\_csv(csv\_file\_path)  
  
# Filter for USA  
usa\_medals\_df = df[df['country\_code'] == 'USA']  
  
# Reshape the DataFrame to have one column for medal type, gender, and count  
melted\_data = usa\_medals\_df.melt(id\_vars=['gender'],  
 value\_vars=['medal\_code'],  
 var\_name='Medal',  
 value\_name='Count')  
  
# Plotting  
plt.figure(figsize=(10, 6))  
sns.barplot(data=melted\_data, x='Medal', y='Count', hue='gender', palette='viridis')  
  
plt.xlabel('Medal Type')  
plt.ylabel('Number of Medals')  
plt.title('Number of Gold, Silver, and Bronze Medals Won by Gender in the USA')  
plt.legend(title='Gender')  
plt.show()



import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load CSV data into a DataFrame  
df = pd.read\_csv('medals.csv')  
  
# Display the first few rows of the DataFrame (optional, for verification)  
print(df.head())  
  
# Count the number of each type of medal per country  
result\_counts = df.groupby(['country\_code', 'medal\_code']).size().unstack(fill\_value=0)  
  
# Calculate the total number of medals per country  
result\_counts['Total'] = result\_counts.sum(axis=1)  
  
# Filter countries with more than 50 total medals  
filtered\_result\_counts = result\_counts[result\_counts['Total'] > 50]  
  
# Drop the 'Total' column for plotting  
filtered\_result\_counts = filtered\_result\_counts.drop(columns='Total')  
  
# Get the top 20 countries with the most medals  
top\_20\_countries = filtered\_result\_counts.sum(axis=1).nlargest(20).index  
top\_20\_result\_counts = filtered\_result\_counts.loc[top\_20\_countries]  
  
# Count the number of males and females per country  
gender\_counts = df.groupby(['country\_code', 'gender']).size().unstack(fill\_value=0)  
  
# Filter the gender counts to include only 'M' and 'W', and match the top 20 countries  
top\_20\_gender\_counts = gender\_counts.loc[top\_20\_countries]  
top\_20\_gender\_counts = top\_20\_gender\_counts[['M', 'W']] # Only include 'M' and 'W'  
  
# Display the filtered\_result\_counts and gender\_counts for verification  
print("\nTop 20 Medal Counts:")  
print(top\_20\_result\_counts)  
  
print("\nTop 20 Gender Counts per Country (only M and W):")  
print(top\_20\_gender\_counts)  
  
# Plotting  
fig, ax = plt.subplots(figsize=(14, 8))  
  
# Plot medal counts per country  
top\_20\_result\_counts.plot(kind='bar', stacked=True, ax=ax, colormap='viridis')  
  
# Customize the legend  
ax.legend(['1.0 Gold', '2.0 Silver', '3.0 Bronze'], title='Medals')  
  
# Add labels and title  
plt.xlabel('Country Code')  
plt.ylabel('Number of Medals')  
plt.title('Top 20 Medal Counts for Countries with More Than 50 Total Medals')  
plt.grid(axis='y', linestyle='--', alpha=0.7)  
  
# Set the x-ticks to be country codes  
ax.set\_xticks(range(len(top\_20\_result\_counts.index)))  
ax.set\_xticklabels(top\_20\_result\_counts.index, rotation=45, ha='right')  
  
# Add secondary axis for gender distribution  
ax2 = ax.twinx() # Create a second y-axis sharing the same x-axis  
top\_20\_gender\_counts.plot(kind='bar', stacked=True, ax=ax2, alpha=0.5, colormap='plasma', position=1)  
  
# Customize the secondary y-axis  
ax2.set\_ylabel('Number of Athletes')  
ax2.set\_title('Gender Distribution per Country')  
ax2.legend(['Male', 'Female'], title='Gender', loc='upper left')  
  
plt.tight\_layout()  
plt.show()