Homework 4 - IST 462

Student name:

```
#importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
```

```
# Please make sure that the olympics1992_2008 file has been uploaded to the same folder where you have this
#       notebook file in your google drive and connect your drive

from google.colab import drive
drive.mount('/content/drive')
```

```
    Mounted at /content/drive
```

```
odata = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/data/HW4_olympics1992_2008.csv',skiprows=4)
```

```
# Start exploratory data analysis
odata.info()
```

```
    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 9619 entries, 0 to 9618
    Data columns (total 10 columns):
     #   Column        Non-Null Count  Dtype 
    ---  ------        --------------  ----- 
     0   City          9619 non-null   object
     1   Edition       9619 non-null   int64 
     2   Sport         9619 non-null   object
     3   Discipline    9619 non-null   object
     4   Athlete       9619 non-null   object
     5   NOC           9619 non-null   object
     6   Gender        9619 non-null   object
     7   Event         9619 non-null   object
     8   Event_gender  9619 non-null   object
     9   Medal         9619 non-null   object
    dtypes: int64(1), object(9)
    memory usage: 751.6+ KB
```

```
odata.head()
```

	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
0	Barcelona	1992	Aquatics	Diving	XIONG, Ni	CHN	Men	10m platform	M	Bronze

```
# Add cells with any additional exploratory data analysis commands/functions that you think are necessary. This will
# not be graded but will help you in solving this homework's tasks
# Hint.. get the unique entries for columns of interest

unique_cities = odata['City'].unique()
unique_cities
unique_sport = odata['Sport'].unique()
unique_sport
```

```
array(['Aquatics', 'Archery', 'Athletics', 'Badminton', 'Baseball',
       'Basketball', 'Boxing', 'Canoe / Kayak', 'Cycling', 'Equestrian',
       'Fencing', 'Football', 'Gymnastics', 'Handball', 'Hockey', 'Judo',
       'Modern Pentathlon', 'Rowing', 'Sailing', 'Shooting',
       'Table Tennis', 'Tennis', 'Volleyball', 'Weightlifting',
       'Wrestling', 'Softball', 'Taekwondo', 'Triathlon'], dtype=object)
```

Solve the following tasks. You can add as many additional cells as you need to solve each one of them.

Task #1 (30 points)

a) List the 5 countries that accumulated the most medals across all the olympic game editions covered in the dataset

b) List the 5 countries that accumulated the most GOLD medals across all the olympic game editions covered in the dataset

```
medal_counts = odata.groupby('NOC')['Medal'].count()
#grouping the dataframe by country and then counting the medals each country got. putting this into medal_counts

# Sort the cities by the number of medals in descending order
sorted_medal_counts = medal_counts.sort_values(ascending=False)
#then sorting the medal_counts list into desending order with the country with the most metals at the top


top_countries = sorted_medal_counts.head(5)
#pulling the top 5 countries and creating a list of top countries


print("Top 5 countries with the most medals:")
print(top_countries)
```

```
Top 5 countries with the most medals:
NOC
USA    1311
GER     691
AUS     678
RUS     638
```

```
CHN     550
Name: Medal, dtype: int64
```

```
gold_odata = odata[odata['Medal'] == 'Gold']
#filtering the data to only pull where the value of medal is gold, including the information of the other columns in this dataframe (gold_odata)

# Group the filtered DataFrame by 'City' and count the number of GOLD medals
gold_medal_counts = gold_odata.groupby('NOC')['Medal'].count()
#then grouping by country and then medal (which is only gold) within the gold_odata data frame. Then storing this as a list as gold_medal_counts

sorted_gold_medal_counts = gold_medal_counts.sort_values(ascending=False)
#sorting the gold_medal_counts list as ascending. so having the greatest value at the top

# Get the top 5 cities with the most GOLD medals
top_gold_countries = sorted_gold_medal_counts.head(5)
#then pulling the top 5 countries with the most gold medals by pulling the head. saving this as top_gold_countries list


print("Top 5 countries with the most GOLD medals:")
print(top_gold_countries)
```

```
Top 5 countries with the most GOLD medals:
NOC
USA    620
GER    237
CHN    202
RUS    192
AUS    186
Name: Medal, dtype: int64
```

Task #2 (15 points)

List the number of Gold, Silver and Bronze medals obtained by Women and Men across all the olympic game editions covered in the dataset

```
medal_counts = odata.groupby(['Gender', 'Medal'])['Medal'].count().unstack()
#grouping by gender and medal. Having medal be counted for each gender and medal
#unstack allows the data to be shown in columns (medal) and gender (rows) in the panda series

print("Medal counts obtained by Women and Men:")
print(medal_counts)
#printing the panda series

print("-" * 40)
print("Medal Count for each Gender- Specifying each medal")
for gender in ['Women', 'Men']:
    for medal in medal_counts.columns:
        count = medal_counts.at[gender, medal]
        print(f"{gender}: {medal} - Count: {count}")
```

```
Medal counts obtained by Women and Men:
Medal   Bronze  Gold  Silver
Gender
Men       1918  1807    1797
Women     1386  1357    1354
----------------------------------------
```

```
Medal Count for each Gender- Specifying each medal
Women: Bronze - Count: 1386
Women: Gold - Count: 1357
Women: Silver - Count: 1354
Men: Bronze - Count: 1918
Men: Gold - Count: 1807
Men: Silver - Count: 1797
```

Task #3 (15 points)

List the names of the 5 male athletes and 5 female athletes that obtained the most medals across all the olympic game editions covered in the dataset

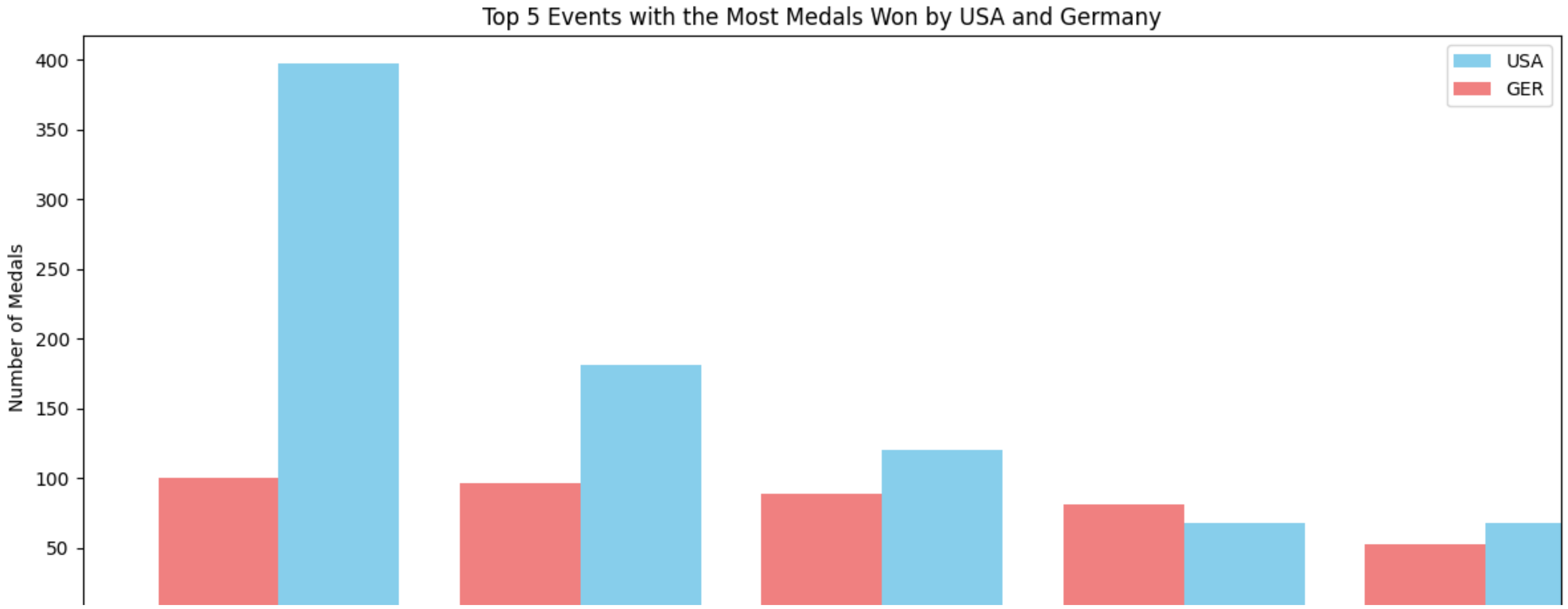
```
medal_counts = odata.groupby(['Athlete', 'Gender'])['Medal'].count().reset_index()
#grouping the odata dataframe by athlete and gender, and counting the number of medals for each category.


sorted_medal_counts = medal_counts.sort_values(by='Medal', ascending=False)
#then sorting the values by medal. so this datafram has each athlete gender and medal, with the athlete with the most medals at the top


male_athletes = sorted_medal_counts[sorted_medal_counts['Gender'] == 'Men'].head(5)
female_athletes = sorted_medal_counts[sorted_medal_counts['Gender'] == 'Women'].head(5)
#then filtering the sorted_medal_counts dataframe to pull the top 5 where gender is men and top 5 where gender is women to create 2 separate dataframes w both genders.


print("Top 5 male athletes with the most medals:")
print(male_athletes[['Athlete', 'Medal']].to_string(index=False))

print("-" * 40)

print("\nTop 5 female athletes with the most medals:")
print(female_athletes[['Athlete', 'Medal']].to_string(index=False))
#printing it and adding .to_string(index_False) to remove the indexes
```

```
Top 5 male athletes with the most medals:
        Athlete  Medal
PHELPS, Michael     16
  NEMOV, Alexei     12
 HALL, Gary Jr.     10
SCHERBO, Vitaly     10
    THORPE, Ian      9
----------------------------------------

Top 5 female athletes with the most medals:
               Athlete  Medal
       THOMPSON, Jenny     12
     COUGHLIN, Natalie     11
VAN ALMSICK, Franziska     10
          TORRES, Dara      9
    TRILLINI, Giovanna      8
```

Task #4 (40 points)

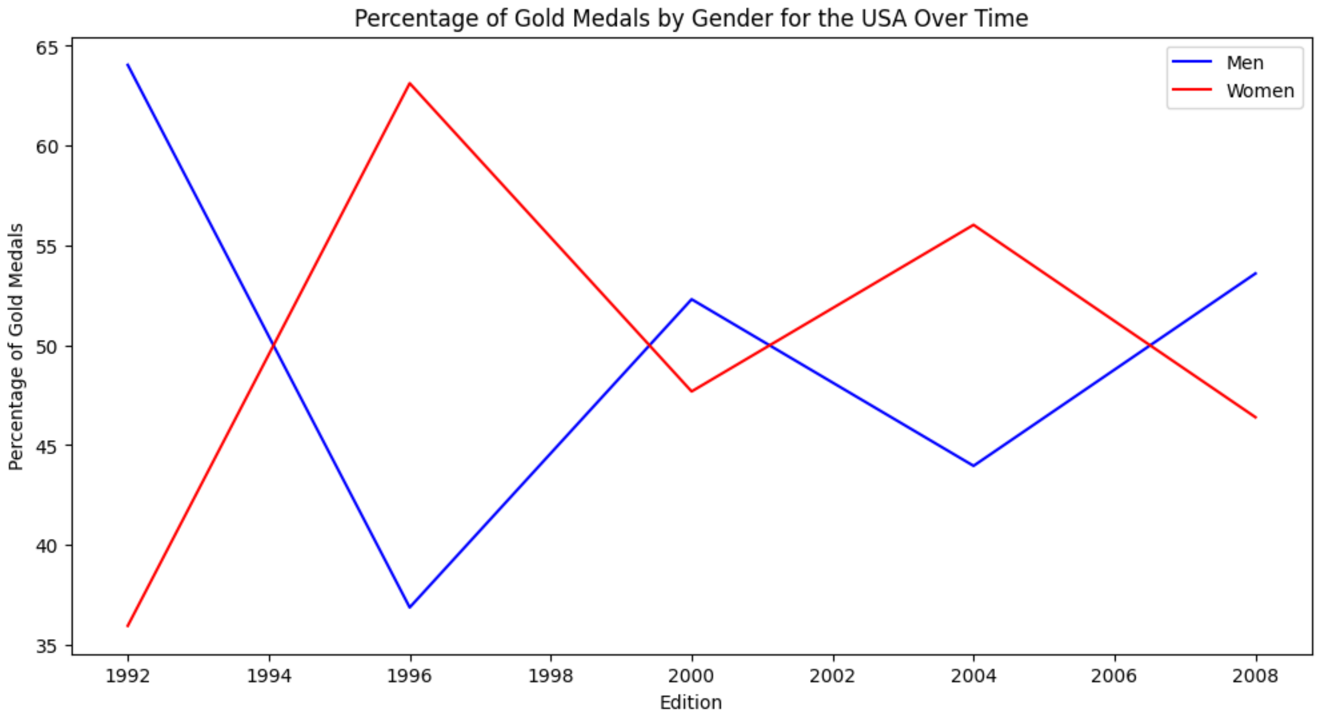
Provide two additional analysis results that you can derive from the dataset (they must be different than those obtained in tasks 1 to 3). The results can include graphs (but it is not required). Describe the results obtained in the cell provided for that purpose

```
import matplotlib.pyplot as plt



usa_germany_odata = odata[odata['NOC'].isin(['USA', 'GER'])]
#pulling the data from the countries that are either usa or germany (in the NOC column)
#then putting this into a dataframe called usa_germany_odata

medal_counts = usa_germany_odata.groupby(['Sport', 'NOC'])['Medal'].count().unstack(fill_value=0)
#then grouping by sport and country and counting the number of metals for each of the two countries.
#unstacking so it creates a cleaner data frame and storing it as metal_counts

top_5_usa = medal_counts['USA'].nlargest(5)
top_5_germany = medal_counts['GER'].nlargest(5)
#grabbing the top 5 medal counts for both germany and the usa


plt.figure(figsize=(12, 6))
#creating a bar graph to visualize this
#setting the width to 12 inches and height to 6 inches

top_5_usa.plot(kind='bar', position=0, width=0.4, label='USA', color='skyblue')
#pulling the top 5 sport for the usa, creating the position as 0, which means that the plot occurs on the 0 position of the x axis.
#creating the bar widths, labeling it and creating a color
top_5_germany.plot(kind='bar', position=1, width=0.4, label='GER', color='lightcoral')
#pulling top 5 of germany (relevant to the top 5 pulled from the us), setting position to 1, which means this bar occurs to the left of the us
#setting the width to the same as the us bar, labeling it, and creating a color for it

plt.xlabel('Sport')
#labeling the x axis
plt.ylabel('Number of Medals')
#labeling the y axis
plt.title('Top 5 Events with the Most Medals Won by USA and Germany')
#creating a title
plt.xticks(rotation=45)
#rotating the x axis so its more readable
plt.legend()
#creating a legand showing which color is usa and which is germant

plt.tight_layout()
#this helps to improve the layout of the plot
plt.show()
#displaying the plot
```



```
usa_odata = odata[odata['NOC'] == 'USA']
#only include the data for the us by filtering the odata frame and looking to pull where NOC == USA.
#storing this as another dataframe named usa_odata


gold_medal_counts = usa_odata[usa_odata['Medal'] == 'Gold'].groupby(['Edition', 'Gender'])['Medal'].count().unstack(fill_value=0)
#grouping the usa data frame by year, gender. Only pulling where the medal is equal to gold and then counting where the medal is equal to gold
#setting the fill value to 0 incase there is a year where no men or women won a gold medal
#storing this as a dataframe


gold_medal_counts['Total'] = gold_medal_counts['Men'] + gold_medal_counts['Women']
gold_medal_counts['Men_Percentage'] = (gold_medal_counts['Men'] / gold_medal_counts['Total']) * 100
gold_medal_counts['Women_Percentage'] = (gold_medal_counts['Women'] / gold_medal_counts['Total']) * 100
#calculating the percentages of gold medals by men and women


plt.figure(figsize=(12, 6))
#creating a plot w the dimensions 12*6

plt.plot(gold_medal_counts.index, gold_medal_counts['Men_Percentage'], label='Men', color='blue')
#creating a line with the gold medal count percentages for men.
#The gold_medal_counts_index allows us to define the x axis as the index of this dataframe which is years
#and the gold_medal_counts['Men Percentage'] allows us to define the y axis as the percentage of gold medals
#then we label this line as men and set the color to blue
plt.plot(gold_medal_counts.index, gold_medal_counts['Women_Percentage'], label='Women', color='red')
#this adds a line for the medal count percentage for women
#uses the same index in gold_medal_counts to keep the x axis
#then pulls the women_percentage to define the y axis while simultaneously creating a line for women, labeling it, and setting the color to red

plt.xlabel('Edition')
#labeling the x axis as edition
plt.ylabel('Percentage of Gold Medals')
plt.ylabel('Percentage of Gold Medals')
#labeling the y axis as percentage of gold medals
plt.title('Percentage of Gold Medals by Gender for the USA Over Time')
#creating a title
plt.legend()
#adding a legand to show which line is men and women

plt.show()
#display the plot
```

```
<Figure size 1000x800 with 0 Axes>
```



RESULTS DESCRIPTION

```
#The first graph shows the number of medals won by germany and the us for the top 5 sports
#I wanted to look into this, as the us and germany are the top two countries for obtaining the most medals (as found in task #1)
#I wanted to identify which sports were contributing the most greately to the USA having the most medals and look to see if there is any closeness within a sport
#Through this, I found that the US seems to pull a lot more medals in the top sports
#(which is also evident through task #1 as the USA has won 1311 medals compared to germany only winning 691)
#looking at the graph, it shows that the US has a lot of medals won through aquatics (with around 400 compared to Germany's 100).
#In order for Germany to be more competitive in obtaining more medals then the US,
#the country would likely invest in its aquatics program to take these victorys away from the US
#Germany also has slightly more medals within hockey, so the US could capitalize on this and focus on their hockey program to be able to beat out Germany within this sport


#For the second plot, I wanted to compare the medals won overtime by both men and women within the USA
#There is a large separating within the pay and advertisement men and women sports team achieve
```

```
#a lot of people deem this to men's teams being more successful then womens, so I wanted to see if this was true within the olympics
#the results were the complete opposite, women have won more medals then men in all of the years recorded (except 3)
#what this shows is that the pay gap within the sports industry between men and women is obsurd as women frequently bring home more olympic medals

#this study could be expanded in comparing the men and women's obtainment of medals within the whole world
#although, I found this particularly interesting with the common debates over pay especially regarding the men's national soccer team and women's national socer team
```