Homework 06

⚠️Before you start ⚠️

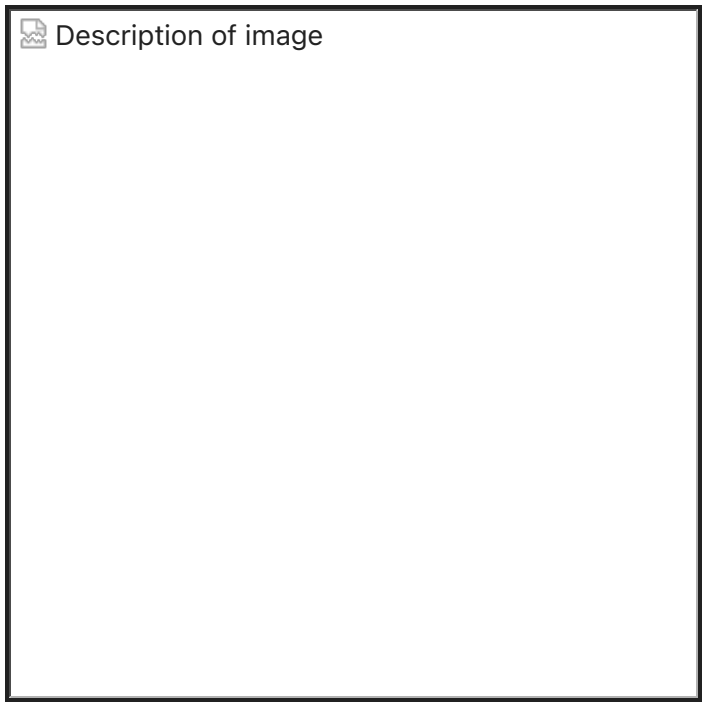
Duplicate this Jupyter Notebook in your `week-07` folder (right-click -> Duplicate) and then add your last name to the beginning of it (ie. `blevins-hw-06.ipynb` - otherwise you risk having all your work overwritten when you try to sync your GitHub repository with your instructor's repository.

Student Name: Your name here

Overview

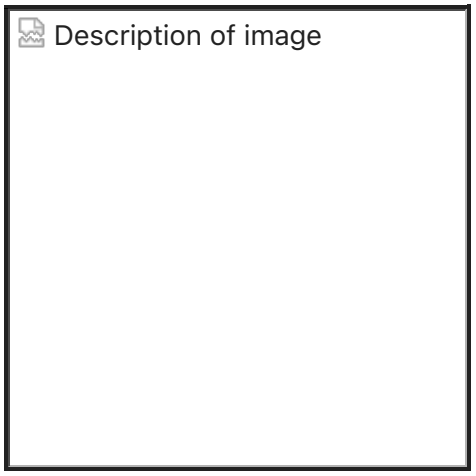
This homework assignment will help you learn how to use the Pandas library to explore tabular data by applying some of the concepts and lessons from Melanie Walsh, Pandas I to a new dataset.

This week we're going to use a spreadsheet of historical data transcribed by CU Denver history major Ryan Hanlon as part of his final project for the course Introduction to Digital Studies in Spring 2021. The data consists of a passenger list from a steamship that arrived in Boston on April 9, 1884 carrying several hundred immigrants.



The Steamship Grecian

The nine-page passenger list from the *Steamship Grecian* was submitted to authorities at the Port of Boston. This document was later scanned by the Church of Latter Day Saints and made available through its FamilySearch online archive.



Passenger list from the Steamship Grecian

Ryan then transcribed the data in Spring 2021 into a spreadsheet formatted as a CSV (comma separated value) file contained in this folder: `boston-passenger-list-1884.csv` .

For this homework, you will be following many of Melanie Walsh's steps in Pandas I and then adapting them to fit this new dataset.

---

1. Import the Pandas library (use the alias `pd`) and read in the CSV file, storing the contents of the file as a dataframe named `passengers_df`. Add a second line of code to display the contents of the dataframe (truncated).

In [10]:
```
import pandas as pd

passengers_df = pd.read_csv('boston-passenger-list-1884.csv')
```

2. Display the **first 6 rows** of the dataframe.

In [12]:
```
passengers_df.head(6)
```

Out[12]:

	first_name	last_name	date	age	native_country	destination_city	destination_sta
0	Jno	McNab	09 Apr 1884	21	Scotland	Suelpla	Cana
1	Thos	Campbell	09 Apr 1884	24	Scotland	Suelpla	Cana
2	Jas	Mitchell	09 Apr 1884	23	Scotland	Detroit	
3	Don	Cumming	09 Apr 1884	24	Scotland	Detroit	
4	Jno	McKinlay	09 Apr 1884	24	Scotland	Winnipeg	Cana
5	John	Wilson	09 Apr 1884	28	Scotland	Boston	N

3. Show a **random sample of 10 rows** from your dataframe.

In [14]:
```
passengers_df.sample(10)
```

Out[14]:

	first_name	last_name	date	age	native_country	destination_city	destination_
480	Mary	Mullin	09 Apr 1884	25	Ireland	Pittsburg	
89	Jno	Canny	09 Apr 1884	21	Ireland	Boston	
173	Jno	McDermott	09 Apr 1884	19	Ireland	Dobbs Ferry	
231	Pat	Gibbons	09 Apr 1884	17	Ireland	Philadelphia	
417	Jas	Madden	09 Apr 1884	42	Ireland	San Fransisco	
87	Jas	Dolan	09 Apr 1884	22	Ireland	Boston	
97	Rodger	McLaughlin	09 Apr 1884	21	Ireland	Boston	
26	Cath	McBride	09 Apr 1884	11	Scotland	Chicago	
169	Bgt	Halligan	09 Apr 1884	17	Ireland	Brooklyn	
71	Jas	O'Donnell	09 Apr 1884	19	Ireland	Boston	

4. What are **two historical questions** about this list of passengers that you might be able to answer using Pandas?

1.mayhaps the average age of the passenger 2. the total share of people on the ship that were headed for Chicago.

Analyzing the Data

5. Calculate "summary statistics" for the passenger data.

In [19]:
```
#Your Code Here
```

```
passengers_df.describe()

# most freq lastname

passengers_df['last_name'].value_counts()


#occupations (to display one over other comment out either code line)
passengers_df['occupation'].value_counts()

#oldest passenger
max(passengers_df['age'])
```

Out[19]: 64

6. Looking at the summary statistics, answer the following questions:

- What is the most frequently occuring last name?
- How often does the most frequently occuring last name appear?
- How many different *kinds* of occupations are listed in the data?
- How old is the oldest passenger?

Your answers here:

- What is the most frequently occuring last name?
- How often does the most frequently occuring last name appear?

  Doherty is the most frequent last name, appearing 12 times.

- How many different *kinds* of occupations are listed in the data? **9 different occupations appear in the data**
- How old is the oldest passenger? **64 years old**

7. Write code to answer: what was the **median** age of the passengers?

In [23]:
```
grecian_age_df = passengers_df[['age']]

grecian_age_df.describe()

#the 50% shouldddd be the same as the median, so the median age of the passe
```

Out[23]:

	age
count	515.000000
mean	21.440777
std	13.115392
min	0.000000
25%	11.000000
50%	20.000000
75%	28.000000
max	64.000000

8. What were the **ten most frequent cities** that passengers were traveling to and how many of them were going to each of these cities?

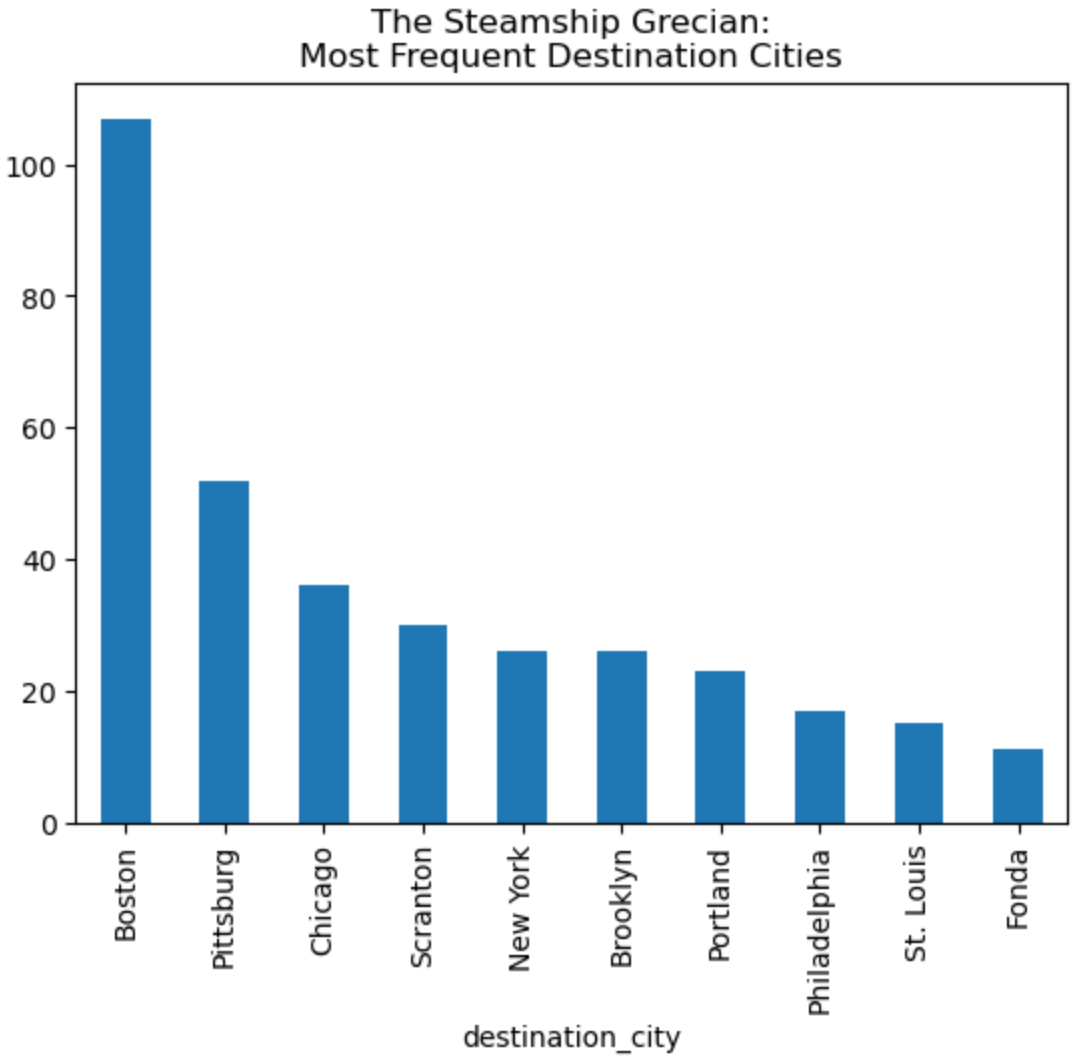
In [25]:
```
passengers_df['destination_city'].value_counts()[:10]
```

Out[25]:
```
destination_city
Boston          107
Pittsburg        52
Chicago          36
Scranton         30
New York         26
Brooklyn         26
Portland         23
Philadelphia     17
St. Louis        15
Fonda            11
Name: count, dtype: int64
```

9. Follow Walsh's example and adapt her code to make a bar chart of the **top ten most frequent destination cities** based on **how many passengers** were going to each of them.

In [27]:
```
passengers_df['destination_city'].value_counts()[:10].plot(kind='bar', title
```

Out[27]:
```
<Axes: title={'center': 'The Steamship Grecian:\nMost Frequent Destination
Cities'}, xlabel='destination_city'>
```



10. Where were passengers coming from? Print out **the most frequent countries** they were immigrating from and how many passengers were coming from each country. Hint: use `value_counts()` and index.

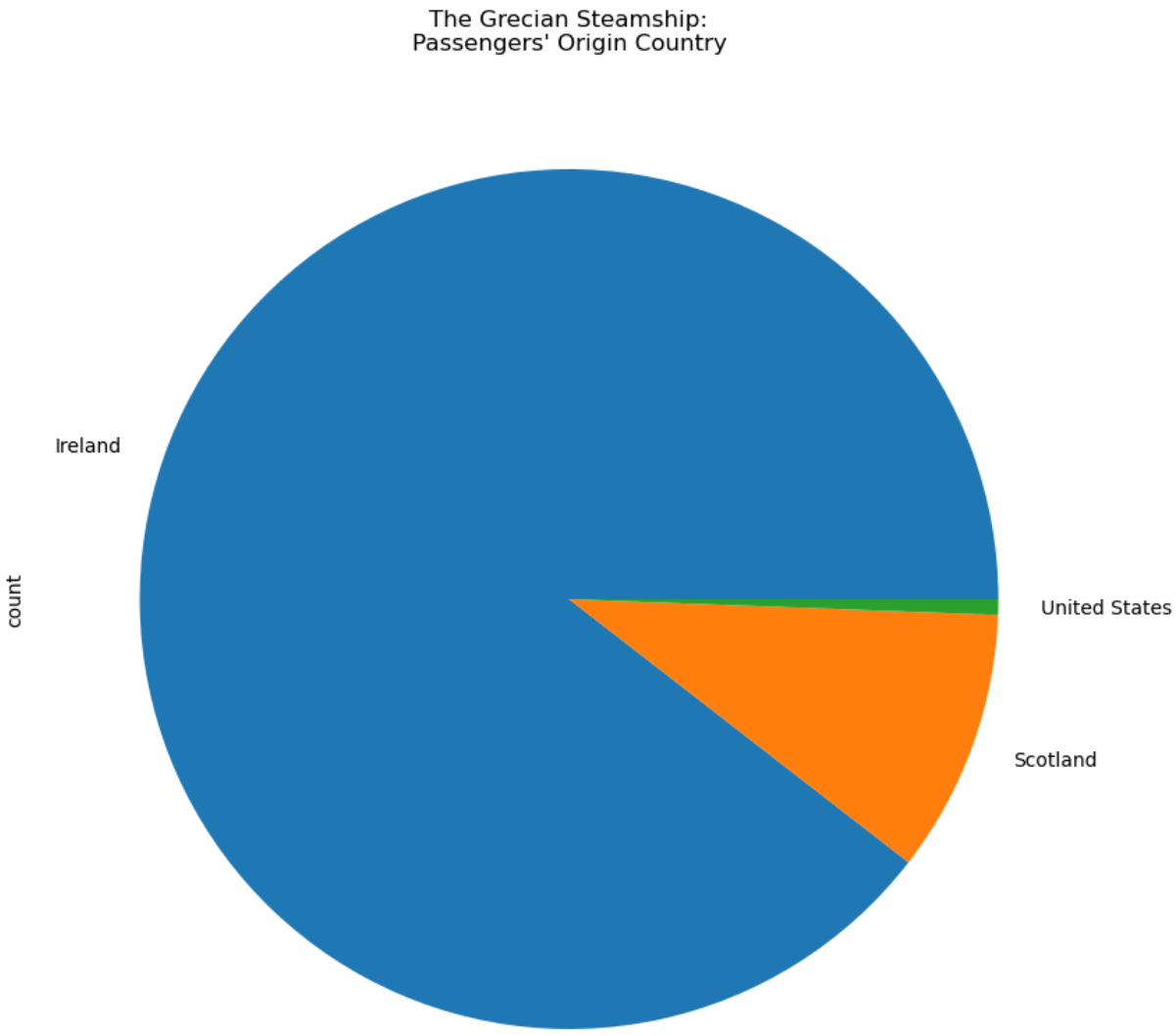
In [29]:
```
passengers_df['native_country'].value_counts()
```

Out[29]:
```
native_country
Ireland          461
Scotland          51
United States      3
Name: count, dtype: int64
```

11. Make a pie chart showing **how many passengers were coming from each country**. Adapt Walsh's example.

In [31]:
```
passengers_df['native_country'].value_counts().plot(kind = 'pie', figsize=(1
```

Out[31]:
```
<Axes: title={'center': "The Grecian Steamship:\nPassengers' Origin Countr
y"}, ylabel='count'>
```



12. Create a new variable called `children_filter` and assign it a True/False statement to that variable that specifies passengers who were **children**. Then use this new `children_filter` to create a new dataframe called `children_df` that **only contains passengers who were children**. Display a sample of **five random rows** from this new dataframe. Hint: look under the `occupation` column in your dataframe. Hint: Walsh example.

In [33]:
```
children_filter = passengers_df['occupation'] == 'Child'
children_df = passengers_df[children_filter]
children_df.sample(5)
```

Out[33]:

	first_name	last_name	date	age	native_country	destination_city	destination_
481	Mary	Mullin	09 Apr 1884	3	Ireland	Pittsburg	
128	Wm	Routledge	09 Apr 1884	8	Ireland	Elmira	
253	Aud	Nolan	09 Apr 1884	9	Ireland	Painesville	
29	Maggie	McBride	09 Apr 1884	4	Scotland	Chicago	
240	Homina	Cafferty	09 Apr 1884	11	Ireland	Chicago	

13. Create a **new CSV file** named `passenger-list-children.csv` that only contains records for passengers who were children. Hint: you'll be printing the contents of `children_df` to a CSV file using `to_csv()` method. Walsh example. To check to make sure you successfully created the file, add a line of code that reads in the newly created CSV file using `pd.read_csv()` .

In [35]:
```
children_df.to_csv("grecian_ship_children.csv", encoding='utf-8', index=Fals

test_df = pd.read_csv("grecian_ship_children.csv")

test_df[test_df['age'] >= 11]
```

Out[35]:

	first_name	last_name	date	age	native_country	destination_city	destination_s
0	Alex	McBride	09 Apr 1884	12	Scotland	Chicago	
1	Cath	McBride	09 Apr 1884	11	Scotland	Chicago	
7	Mary	Watson	09 Apr 1884	11	Scotland	Auburn	
14	James	Doherty	09 Apr 1884	11	Ireland	Boston	
16	Rachel	Routledge	09 Apr 1884	11	Ireland	Elmira	
20	Ellen	Odonnell	09 Apr 1884	11	Ireland	Scranton	
29	Thos	Gorman	09 Apr 1884	11	Ireland	New York	
33	Mary	Lofus	09 Apr 1884	11	Ireland	Oneida	
35	Pat	Halligan	09 Apr 1884	11	Ireland	Brooklyn	
37	Agnes	McGiven	09 Apr 1884	11	Ireland	Mottville	
48	Homina	Cafferty	09 Apr 1884	11	Ireland	Chicago	
75	Mary	Divine	09 Apr 1884	11	Ireland	Fonda	
87	Ann	Brennan	09 Apr 1884	11	Ireland	New York	
90	Mich	Reanny	09 Apr 1884	11	Ireland	Scranton	
97	Mary	Saul	09 Apr 1884	11	Ireland	Indianapolis	

	first_name	last_name	date	age	native_country	destination_city	destination_
104	Pat	Quinn	09 Apr 1884	11	Ireland	Boston	
105	Jas	Conway	09 Apr 1884	11	Ireland	New York	
110	Colman	Mulkern	09 Apr 1884	11	Ireland	Pittsburg	
113	Mary	Winkle	09 Apr 1884	11	Ireland	Portland	
116	Mary	Kerrigan	09 Apr 1884	11	Ireland	Winchester	
118	Barb	Adley	09 Apr 1884	11	Ireland	Portland	
121	Colman	Flaherty	09 Apr 1884	11	Ireland	Pittsburg	
126	Barb	Deran	09 Apr 1884	11	Ireland	Pittsburg	

Bonus Questions

What was the cut-off age for classifying a passenger as a child? Ie. What was **the oldest a passenger could be to still be considered a child**? Write code that prints out the answer to this question.

In [38]:
```
children_df[children_df['age']== max(children_df['age'])]
```

Out[38]:

	first_name	last_name	date	age	native_country	destination_city	destination_st
25	Alex	McBride	09 Apr 1884	12	Scotland	Chicago	

Age Comparison: Calculate and write print() statements that show:

- The average age of passengers from **Ireland**
- The average age of passengers from **Scotland**.
- The difference in years between these average

In [40]:
```
#Your Code Here

from statistics import mean

ireland_df= passengers_df[passengers_df['native_country'] == 'Ireland']

ir_age= mean(ireland_df['age'])
ir_age = round(ir_age,)

scotland_df = passengers_df[passengers_df['native_country']== 'Scotland']

sc_age= mean(scotland_df['age'])
sc_age= round(sc_age,)

print(f"the average age of passengers from ireland is: {ir_age}")
print(f"the average age of passengers from scotland is: {sc_age}")
print(f"difference in years between the two is {(sc_age-ir_age)} years")
```

```
the average age of passengers from ireland is: 21
the average age of passengers from scotland is: 25
difference in years between the two is 4 years
```

Save a Filtered Dataset: Create a new CSV file that contains data for: only adult passengers (**age 18 and over**) who were heading to **Boston**.

In [42]:
```
##boston18_filter=

boston_18plus_df= passengers_df.query('age>=18 and destination_city == "Bost

boston_18plus_df.to_csv("Boston_18_over_list.csv",encoding = 'utf-8',index=F
```

In [ ]:
```

```