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: Kylie Miller

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 [9]:
```
import pandas as pd
passengers_df = pd.read_csv('boston-passenger-list-1884.csv', delimiter=",")
passengers_df
```

Out[9]:

	first_name	last_name	date	age	native_country	destination_city	destination_
0	Jno	McNab	09 Apr 1884	21	Scotland	Suelpla	Ca
1	Thos	Campbell	09 Apr 1884	24	Scotland	Suelpla	Ca
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	Ca
...	...	...	...	...	...	...	
510	Mich	Mulkern	09 Apr 1884	15	Ireland	Boston	
511	Math	Griffins	09 Apr 1884	2	Ireland	Pittsburg	
512	T	McDermott	09 Apr 1884	35	United States	Boston	
513	Wm	Hewitt	09 Apr 1884	25	United States	Boston	
514	F	Doherty	09 Apr 1884	27	United States	Boston	

515 rows × 9 columns

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

In [11]:
```
passengers_df[:6]
```

Out[11]:

	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	

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

In [13]:
```
passengers_df.sample(10)
```

Out[13]:

	first_name	last_name	date	age	native_country	destination_city	destination_
77	Neal	Doherty	09 Apr 1884	2	Ireland	Boston	
280	Mary	Egan	09 Apr 1884	40	Ireland	St. Louis	
40	Wm	Martin	09 Apr 1884	26	Scotland	Brookville	
384	Peter	Saul	09 Apr 1884	7	Ireland	Indianapolis	
232	Rich	Gethins	09 Apr 1884	30	Ireland	Philadelphia	
30	Robt	McBride	09 Apr 1884	6	Scotland	Chicago	
184	Ann	Kearnes	09 Apr 1884	52	Ireland	San Fransisco	
94	Mary	Hanna	09 Apr 1884	36	Ireland	Boston	
392	Terence	Marren	09 Apr 1884	42	Ireland	Boston	
419	Judy	Hession	09 Apr 1884	48	Ireland	Boston	

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

Your answer here.

1. What was the most common country of origin of immigrants on the Steamship Grecian on April 9, 1884?
2. What was the most common occupation of immigrants on the Steamship Grecian on April 9, 1884?

Analyzing the Data

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

In [18]:
```
passengers_df.describe(include='all')
```

Out[18]:

	first_name	last_name	date	age	native_country	destination_city	de
count	512	515	515	515.000000	515	515	
unique	102	170	1	NaN	3	56	
top	Mary	Doherty	09 Apr 1884	NaN	Ireland	Boston	
freq	59	12	515	NaN	461	107	
mean	NaN	NaN	NaN	21.440777	NaN	NaN	
std	NaN	NaN	NaN	13.115392	NaN	NaN	
min	NaN	NaN	NaN	0.000000	NaN	NaN	
25%	NaN	NaN	NaN	11.000000	NaN	NaN	
50%	NaN	NaN	NaN	20.000000	NaN	NaN	
75%	NaN	NaN	NaN	28.000000	NaN	NaN	
max	NaN	NaN	NaN	64.000000	NaN	NaN	

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? ***Doherty***
- How often does the most frequently occuring last name appear? ***12 times***
- How many different *kinds* of occupations are listed in the data? ***9 unique occupations***
- How old is the oldest passenger? ***64 years old***

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

In [22]:
```
print(f"The median age of passengers was {passengers_df['age'].median()}.")
```

```
The median age of passengers was 20.0.
```

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?

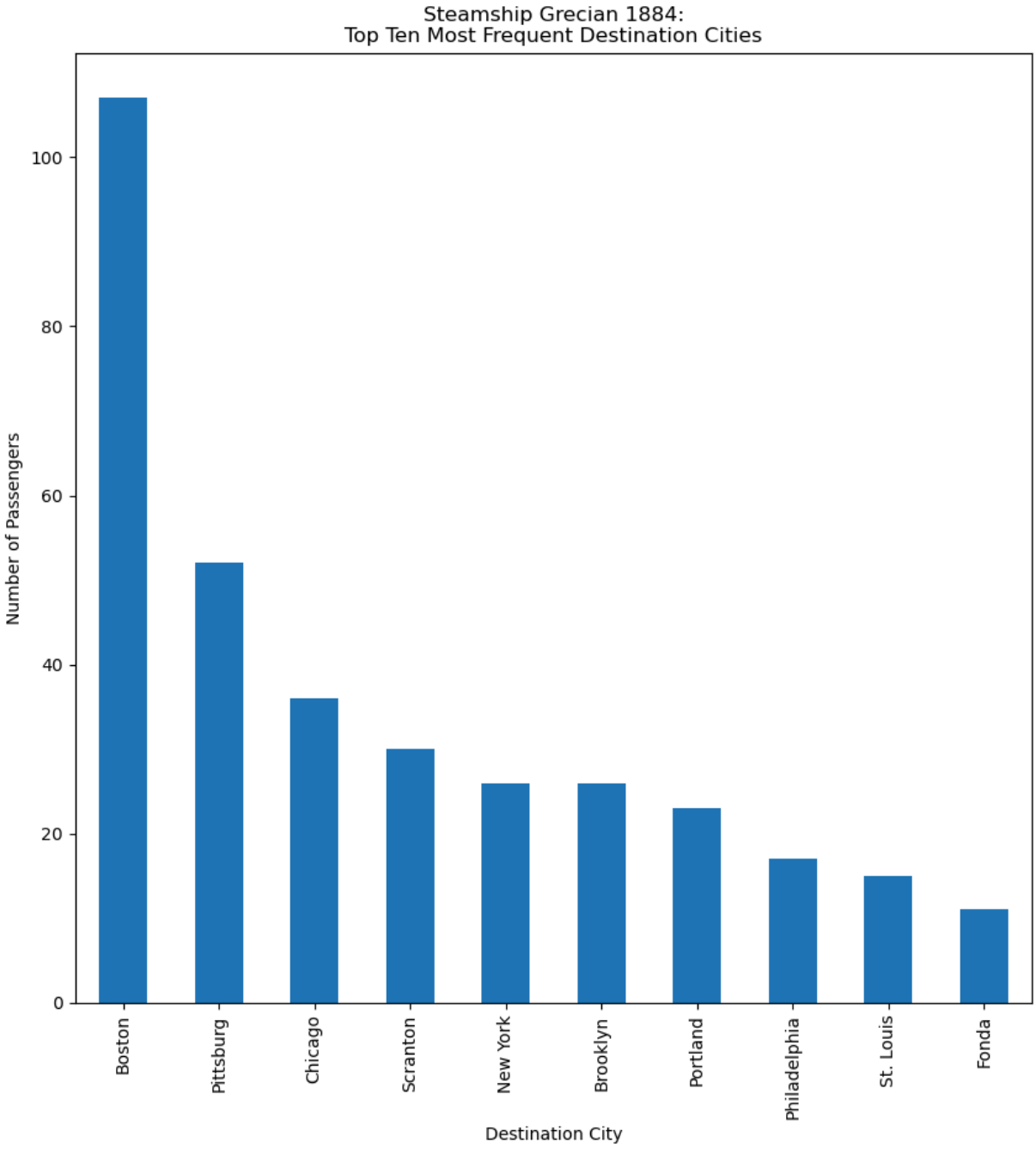
In [24]:
```
passengers_df['destination_city'].value_counts()[:10]
```

Out[24]:
```
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 [26]:
```
passengers_df['destination_city'].value_counts()[:10].plot(kind='bar', figsi
```

Out[26]:
```
<Axes: title={'center': 'Steamship Grecian 1884:\nTop Ten Most Frequent Des
tination Cities'}, xlabel='Destination City', ylabel='Number of Passenger
s'>
```



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.

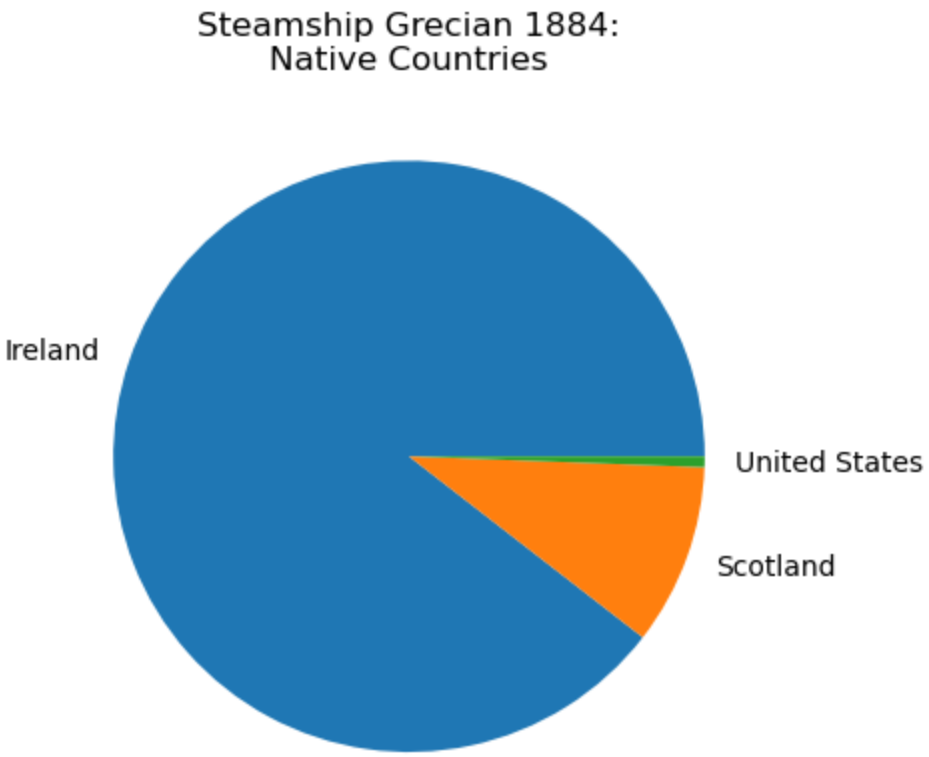
In [28]:
```
passengers_df['native_country'].value_counts()
```

Out[28]:
```
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 [30]:
```
passengers_df['native_country'].value_counts().plot(kind='pie', ylabel='', t
```

Out[30]:
```
<Axes: title={'center': 'Steamship Grecian 1884:\nNative Countries'}>
```



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 [32]:
```
children_filter = passengers_df['occupation'] == 'Child'
children_df = passengers_df[children_filter]
children_df.sample(5)
```

Out[32]:

	first_name	last_name	date	age	native_country	destination_city	destination_
186	Bgt	Kearnes	09 Apr 1884	1	Ireland	San Fransisco	
475	Barb	Adley	09 Apr 1884	11	Ireland	Portland	
134	Edw	Odonnell	09 Apr 1884	9	Ireland	Scranton	
75	Cath	Doherty	09 Apr 1884	6	Ireland	Boston	
26	Cath	McBride	09 Apr 1884	11	Scotland	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 [34]:
```
children_df.to_csv("passenger-list-children.csv", encoding='utf-8', index=Fa
pd.read_csv("passenger-list-children.csv")
```

Out[34]:

	first_name	last_name	date	age	native_country	destination_city	destination_
0	Alex	McBride	09 Apr 1884	12	Scotland	Chicago	
1	Cath	McBride	09 Apr 1884	11	Scotland	Chicago	
2	Wm	McBride	09 Apr 1884	9	Scotland	Chicago	
3	Agnes	McBride	09 Apr 1884	7	Scotland	Chicago	
4	Maggie	McBride	09 Apr 1884	4	Scotland	Chicago	
...	...	...	...	...	...	...	
127	Ellen	Deran	09 Apr 1884	8	Ireland	Pittsburg	
128	Pat	Kyne	09 Apr 1884	9	Ireland	Portland	
129	John	Kyne	09 Apr 1884	3	Ireland	Portland	
130	Mich	Kyne	09 Apr 1884	0	Ireland	Portland	
131	Math	Griffins	09 Apr 1884	2	Ireland	Pittsburg	

132 rows × 9 columns

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 [37]:
```
children_df['age'].max()
```

Out[37]: 12

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 [39]:
```
ireland_filter = passengers_df['native_country'] == 'Ireland'
ireland_passengers = passengers_df[ireland_filter]
averageage_ireland = ireland_passengers['age'].mean()
print(f"The average age of passengers from Ireland was {averageage_ireland}"

scotland_filter = passengers_df['native_country'] == 'Scotland'
scotland_passengers = passengers_df[scotland_filter]
averageage_scotland = scotland_passengers['age'].mean()
print(f"The average age of passengers from Scotland was {averageage_scotlanc

if averageage_ireland > averageage_scotland:
    print(f"On average, Irish passengers were {averageage_ireland-averageage
else:
    print(f"On average, Scottish passengers were {averageage_scotland-averag
```

```
The average age of passengers from Ireland was 20.98698481561822
The average age of passengers from Scotland was 25.098039215686274
On average, Scottish passengers were 4.111054400068053 years older than Scot
tish passengers.
```

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 [41]:
```
boston_filter = passengers_df['destination_city'] == 'Boston'
boston_passengers_df = passengers_df[boston_filter]
over18_filter = boston_passengers_df['age'] >= 18
bostonpassengers_over18_df = boston_passengers_df[over18_filter]
bostonpassengers_over18_df.to_csv("passenger-list-bostonover18.csv", encodir
pd.read_csv("passenger-list-bostonover18.csv")
```

Out[41]:

	first_name	last_name	date	age	native_country	destination_city	destination_s
0	John	Wilson	09 Apr 1884	28	Scotland	Boston	
1	Rob	Watson	09 Apr 1884	20	Scotland	Boston	
2	Thos.B	Watson	09 Apr 1884	23	Scotland	Boston	
3	NaN	Roberts	09 Apr 1884	40	Scotland	Boston	
4	Robt	Chalmers	09 Apr 1884	23	Scotland	Boston	
...	...	...	...	...	...	...	
91	Bgt	Adley	09 Apr 1884	20	Ireland	Boston	
92	Ned	Flaherty	09 Apr 1884	22	Ireland	Boston	
93	T	McDermott	09 Apr 1884	35	United States	Boston	
94	Wm	Hewitt	09 Apr 1884	25	United States	Boston	
95	F	Doherty	09 Apr 1884	27	United States	Boston	

96 rows × 9 columns

In [ ]: