Homework 07

⚠️ Before you start ⚠️

Duplicate this Jupyter Notebook in your `week-08` folder (right-click -> Duplicate) and then add your last name to the beginning of it (ie. `blevins-hw-07.ipynb` - otherwise you risk having all your work overwritten when you try to sync your GitHub repository with your instructor's repository.

Student Name: *Jessie Aldridge*

We're going to be practing using the Pandas library to explore another dataset: a famouse collection of information about some passengers on board the *Titanic*. To find out more information about this dataset look at the data dictionary on this page: https://www.kaggle.com/c/titanic/data#:~:text=should%20look%20like.-,data%20dictionary,Variable

Import the pandas library.

In [8]:
```
#Your Code Here
import pandas as pd
```

Read in the CSV file.

In [10]:
```
#Your Code Here
titanic_df = pd.read_csv('titanic.csv')
```

Display the first 12 rows of your dataset.

In [12]:
```
#Your Code Here
titanic_df.head(12)
```

Out[12]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171

1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783

What are the different data types contained in each column?

In [14]:
```
#Your Code Here
titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In your own words, what is the difference in the data types for `Survived` vs. `Age` columns?

Survived column is an int64 data type, which represents an integer, which is a whole number, whereas age is represented by a float data type which represents a float, which can have decimals.

Use the `.isna()` or `.notna()` methods in conjunction with a filter to only select rows from your dataframe consisting of passengers for which we have information about the cabin they were in.

In [17]:
```
#Your Code Here
cabin_info_known = titanic_df['Cabin'].notna()
titanic_df[cabin_info_known].sample(20)
```

Out[17]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
55	56	1	1	Woolner, Mr. Hugh	male	NaN	0	0	19947
641	642	1	1	Sagesser, Mlle. Emma	female	24.0	0	0	PC 17477

194	195	1	1	Brown, Mrs. James Joseph (Margaret Tobin)	female	44.0	0	0	PC 17610
339	340	0	1	Blackwell, Mr. Stephen Weart	male	45.0	0	0	113784
205	206	0	3	Strom, Miss. Telma Matilda	female	2.0	0	1	347054
291	292	1	1	Bishop, Mrs. Dickinson H (Helen Walton)	female	19.0	1	0	11967
257	258	1	1	Cherry, Miss. Gladys	female	30.0	0	0	110152
492	493	0	1	Molson, Mr. Harry Markland	male	55.0	0	0	113787
310	311	1	1	Hays, Miss. Margaret Bechstein	female	24.0	0	0	11767
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463
209	210	1	1	Blank, Mr. Henry	male	40.0	0	0	112277
839	840	1	1	Marechal, Mr. Pierre	male	NaN	0	0	11774
66	67	1	2	Nye, Mrs. (Elizabeth Ramell)	female	29.0	0	0	C.A. 29395
248	249	1	1	Beckwith, Mr. Richard Leonard	male	37.0	1	1	11751

763	764	1	1	Carter, Mrs. William Ernest (Lucile Polk)	female	36.0	1	2	113760
298	299	1	1	Saalfeld, Mr. Adolphe	male	NaN	0	0	19988
572	573	1	1	Flynn, Mr. John Irwin ("Irving")	male	36.0	0	0	PC 17474
268	269	1	1	Graham, Mrs. William Thompson (Edith Junkins)	female	58.0	0	1	PC 17582
183	184	1	2	Becker, Master. Richard F	male	1.0	2	1	230136
504	505	1	1	Maioni, Miss. Roberta	female	16.0	0	0	110152

What percentage of rows (passengers) in the dataset have information about their cabin number?

In [19]:
```
#Your Code Here
cabin_info_known.value_counts(normalize=True)
```

Out[19]:
```
Cabin
False    0.771044
True     0.228956
Name: proportion, dtype: float64
```

Some of our columns are hard to read. **Rename the following columns:**

- The `SibSp` column contains information about whether the passenger had family on board (siblings or spouses). **Rename the column `siblings_spouses`.**
- The `Pclass` column stands for the ticket class (1st, 2nd, or 3rd). **Rename the column `ticket_class`.**

Hint: remember to change it permanently rather than temporarily.

In [21]:
```
#Your Code Here
titanic_df = titanic_df.rename(columns={'SibSp': 'siblings_spouses'})
titanic_df = titanic_df.rename(columns={'Pclass': 'ticket_class'})
titanic_df = titanic_df.rename(columns={'PassengerId': 'pass_id'})
titanic_df.sample(1)
```

Out[21]:

	pass_id	Survived	ticket_class	Name	Sex	Age	siblings_spouses	Parch
333	334	0	3	Vander Planke, Mr. Leo Edmondus	male	16.0	2	0

Which passengers bought the nine most expensive tickets?

In [23]:
```
#Your Code Here
titanic_df.sort_values(by="Fare", ascending=False).head(9)
```

Out[23]:

	pass_id	Survived	ticket_class	Name	Sex	Age	siblings_spouses	Parch
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2
341	342	1	1	Fortune, Miss. Alice Elizabeth	female	24.0	3	2
438	439	0	1	Fortune, Mr. Mark	male	64.0	1	4
311	312	1	1	Ryerson, Miss. Emily Borie	female	18.0	2	2
742	743	1	1	Ryerson, Miss. Susan Parker "Suzette"	female	21.0	2	2

What was the median age of passengers on the Titanic?

In [25]:
```
#Your Code Here
titanic_df["Age"].median()
```

Out[25]: 28.0

Who was the oldest passenger on the Titanic in our dataset?

In [27]:
```
#Your Code Here
titanic_df["Age"].max()
```

Out[27]:
```
80.0
```

In [28]:
```
titanic_df.sort_values(by="Age", ascending=False).head(1)
```

Out[28]:

	pass_id	Survived	ticket_class	Name	Sex	Age	siblings_spouses	Parch
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0

Use the `groupby` function to count how many passengers bought each class of ticket.

In [30]:
```
#Your Code Here
ticket_class = titanic_df.groupby("ticket_class")["pass_id"].count()
ticket_class
```

Out[30]:
```
ticket_class
1    216
2    184
3    491
Name: pass_id, dtype: int64
```

Use the `groupby` function to group passengers into different classes of ticket and then calculate the median age of passengers within each ticket class.

In [32]:
```
#Your Code Here
age_per_class = titanic_df.groupby("ticket_class")["Age"].median()
age_per_class
```

Out[32]:
```
ticket_class
1    37.0
2    29.0
3    24.0
Name: Age, dtype: float64
```

Use the `groupby` function to group passengers into different classes of ticket and then calculate the median ticket fare within each ticket class.

In [34]:
```
#Your Code Here
```

```
fare_per_class = titanic_df.groupby("ticket_class")["Fare"].median()
fare_per_class
```

Out[34]:
```
ticket_class
1    60.2875
2    14.2500
3     8.0500
Name: Fare, dtype: float64
```

Bonus Questions

Bonus: Make the Survived column more legible. Write a function and apply it to the dataframe that changes the 0 and 1 values to "Died" and "Lived." Then display the first 10 rows to see if it worked.

Note: when changing the values in columns, you might make mistakes. That's okay! You can always reload the dataframe from the original file to start over. When trying to answer this questions, each time you run it I'm going to have you start with the "original" dataframe so that you don't have to go back to the beginning of the notebook and run all the cells again.

In [37]:
```
titanic_df = pd.read_csv('titanic.csv')
```

In [38]:
```
# Your Code Here
def cleanup_aisle_dead(status):
    if status == 0:
        return "Died"
    elif status == 1:
        return "Lived"
```

In [39]:
```
titanic_df['Survived'] = titanic_df["Survived"].apply(cleanup_aisle_dead)
titanic_df.head(10)
```

Out[39]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	Died	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	
1	2	Lived	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	7
2	3	Lived	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	
3	4	Lived	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	5
4	5	Died	3	Allen, Mr. William Henry	male	35.0	0	0	373450	
5	6	Died	3	Moran, Mr. James	male	NaN	0	0	330877	
6	7	Died	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	5
7	8	Died	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	2
8	9	Lived	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	
9	10	Lived	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	3

Bonus: What percentage of people survived the Titanic?

In [41]:
```
#Your Code Here
survival_percent = titanic_df["Survived"].value_counts(normalize=True)
survival_percent
```

Out[41]:
```
Survived
Died     0.616162
Lived    0.383838
Name: proportion, dtype: float64
```

Bonus: Make a pie chart visualizing the proportion of people who survived the Titanic. Hint: use the total number of rows in the dataframe to calculate the percentage.

In [43]:
```
#Your Code Here

survival_percent.plot(kind="pie", title="Survival Rate of Passengers aboard
```

Out[43]:
```
<Axes: title={'center': 'Survival Rate of Passengers aboard the Titanic'},
ylabel='proportion'>
```