Pandas Review Homework

Import pandas

In [1]:
```
import pandas as pd
```

1. Make a data frame from a Python dictionary.

Create a Python dictionary containing

- the names of four of your friends (real or imaginary)
- their ages
- the year they started college
- their majors

In [2]:
```
friends = {
    "Ava Martinez": {"age": 21, "year_started_college": 2023, "major": "Biol
    "Liam Chen": {"age": 22, "year_started_college": 2022, "major": "Compute
    "Noah Johnson": {"age": 20, "year_started_college": 2024, "major": "Mech
    "Sofia Patel": {"age": 23, "year_started_college": 2021, "major": "Psych
```

Make a pandas data frame from your dictionary.

In [13]:
```
df = pd.DataFrame.from_dict(friends, orient="index")
df.reset_index(inplace=True)
df.rename(columns={"index": "name"}, inplace=True)
```

Show your new data frame.

In [14]:
```
df
```

Out[14]:

	name	age	year_started_college	major
0	Ava Martinez	21	2023	Biology
1	Liam Chen	22	2022	Computer Science
2	Noah Johnson	20	2024	Mechanical Engineering
3	Sofia Patel	23	2021	Psychology

Fetch the ages of all your friends.

In [15]:
```
df['age']
```

Out[15]:
```
0    21
1    22
2    20
3    23
Name: age, dtype: int64
```

Fetch the name of your fourth friend.

In [16]:
```
df['name'][3]
```

Out[16]:
```
'Sofia Patel'
```

Fetch the age of your third friend.

In [17]:
```
df['age'][2]
```

Out[17]:
```
20
```

Compute and show the average age of your friends.

In [21]:
```
df["age"].mean()
```

Out[21]:
```
21.5
```

2. Find a table of data on Wikipedia and import it.

Go to Widepedia and find a table of data. It can be anything you want.

In the cell below, import the data and display it (first and last five rows).

In [25]:
```
wiki = pd.read_clipboard()
print(wiki.head(), wiki.tail())
```

```
   Census    Pop.  Note      %±
0    1850     629   NaN       –
1    1860   3,494   NaN  455.5%
2    1870   4,428   NaN   26.7%
3    1880  11,013   NaN  148.7%
4    1890  14,575   NaN   32.3%     Census     Pop.  Note     %±
13    1980  345,890   NaN  36.4%
14    1990  465,622   NaN  34.6%
15    2000  656,562   NaN  41.0%
16    2010  790,390   NaN  20.4%
17    2020  961,855   NaN  21.7%
```

3. Load the RMS titanic data and export a subset of columns

Load the titanic data, make a new `DataFrame` of the fare paid and the survival columns, and export it as a `.csv` file.

In [27]:
```
In [2]: titanic = pd.read_csv("data/titanic.csv")

fare_survival = titanic[["Fare", "Survived"]]
fare_survival.to_csv("fare_survival.csv", index=False)
```

Import your new `.csv` file into a new `DataFrame` and show it (first and last five rows).

In [28]:
```
fare_survival_new = pd.read_csv("fare_survival.csv")
fare_survival_new.head()
```

Out[28]:

	Fare	Survived
0	7.2500	0
1	71.2833	1
2	7.9250	1
3	53.1000	1
4	8.0500	0

4. Fetch specific rows of data of the titanic data

Fetch all the second class passengers of the titanic data and put them in a new `DataFrame` and show it.

In [29]:
```
second_class = titanic[titanic["Pclass"] == 2]

second_class
```

Out[29]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Tick
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	2377
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	0	0	2487
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	2443
20	21	0	2	Fynney, Mr. Joseph J	male	35.0	0	0	2398
21	22	1	2	Beesley, Mr. Lawrence	male	34.0	0	0	2486
...	...	...	...	...	...	...	...	...	
866	867	1	2	Duran y More, Miss. Asuncion	female	27.0	1	0	SC/PAR 21
874	875	1	2	Abelson, Mrs. Samuel (Hannah Wizosky)	female	28.0	1	0	P/PP 33
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)	female	25.0	0	1	2304
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOT 340
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	2115

184 rows × 12 columns

Fetch all the first and third class passengers, put them in a new `DataFrame`, and show it.

In [31]:
```
first_third = titanic[titanic["Pclass"].isin([1, 3])]

first_third
```

Out[31]:

	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
...	...	...	...	...	...	...	...	...	...
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376

707 rows × 12 columns

5. Plot some Titanic data

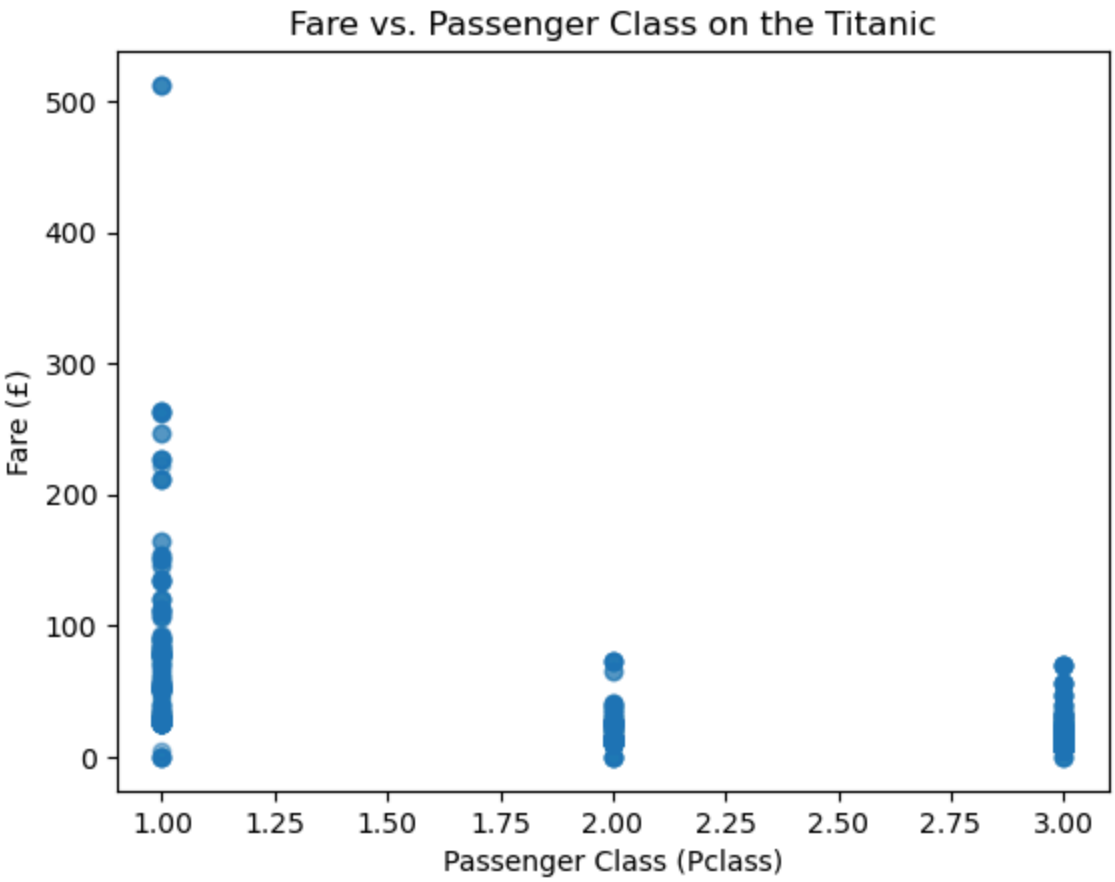
First, import `matplotlib`

In [30]:
```
import matplotlib.pyplot as plt
```

5.a - Scatter plot

Make a scatter plot of fare vs. ticket class (`Pclass`) – seems like these should be perfectly related, but...

In [32]:
```
plt.scatter(titanic["Pclass"], titanic["Fare"], alpha=0.5)
plt.xlabel("Passenger Class (Pclass)")
plt.ylabel("Fare (£)")
plt.title("Fare vs. Passenger Class on the Titanic")
plt.show()
```



5.b - Distribution plot (challenging!)

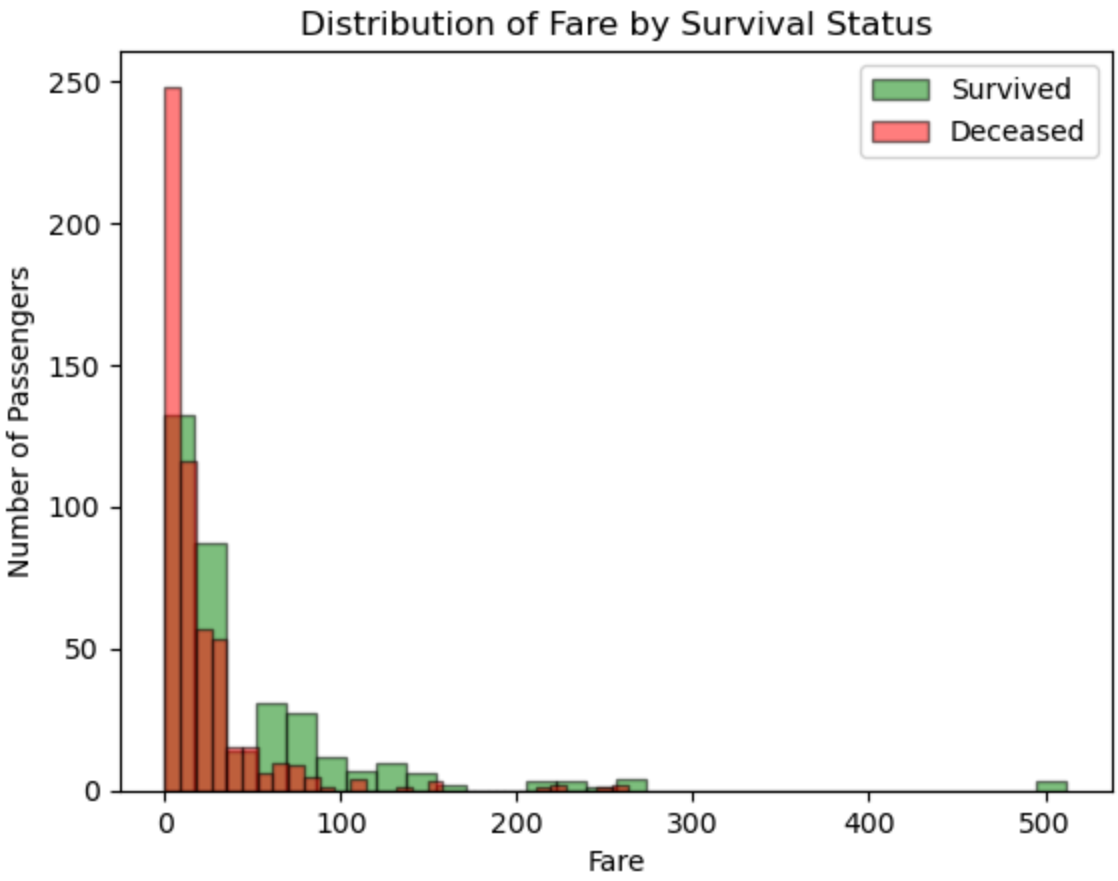
Plot the distributions of fare paid for survivors and deceased in a way that makes for a good visual comparison.

In [35]:
```
fares_survived = titanic.loc[titanic["Survived"] == 1, "Fare"]
fares_deceased = titanic.loc[titanic["Survived"] == 0, "Fare"]

# Plot overlapping histograms
```

```
plt.hist(fares_survived, bins=30, alpha=0.5, label="Survived", color="g", ec
plt.hist(fares_deceased, bins=30, alpha=0.5, label="Deceased", color="r", ec

plt.xlabel("Fare")
plt.ylabel("Number of Passengers")
plt.title("Distribution of Fare by Survival Status")
plt.legend()
plt.show()
```



6. Calculate new columns

6.a - Compute total number of relatives

Create a new column in your titanic `DataFrame` quantifying the total number of relatives on board (siblings + parents – the number of siblings are in `SibSp` and the number of parents are in `Parch`).

In [37]:
```
titanic["RelativesOnBoard"] = titanic["SibSp"] + titanic["Parch"]
titanic[["SibSp", "Parch", "RelativesOnBoard"]].head()
```

Out[37]:

	SibSp	Parch	RelativesOnBoard
0	1	0	1
1	1	0	1
2	0	0	0
3	1	0	1
4	0	0	0

6.b - Did a person have any relatives on board?

Add another column – a Boolean column – indicating whether each person had any relatives on board.

In [38]:
```
titanic["HasRelatives"] = titanic["RelativesOnBoard"] > 0
titanic[["RelativesOnBoard", "HasRelatives"]].head()
```

Out[38]:

	RelativesOnBoard	HasRelatives
0	1	True
1	1	True
2	0	False
3	1	True
4	0	False

7. Computing descriptive statistics

7.a - Compute a mean for a column

Compute the proportion of survivors of the RMS Titanic. **Hint**: the coding of `Survival` as 0 or 1 really works to our advantage here: the proportion of survivors in any group is easily computed using a common statistical function. The 7.a section header should also give you a big clue!

In [39]:
```
titanic["Survived"].mean()
```

Out[39]:
```
0.3838383838383838
```

7.a - Compute a mean for a subset of data

Compute the proportion of survivors for the females on the RMS Titanic (you can do this in one go, or two steps, using an intermediate object containing just the female data).

In [41]:
```
titanic.loc[titanic["Sex"] == "female", "Survived"].mean()
```

Out[41]:
```
0.7420382165605095
```

7.b - Compute statistics by group

Compute the proportion of female vs. male survivors of the RMS Titanic.

In [42]:
```
titanic.groupby("Sex")["Survived"].mean()
```

Out[42]:
```
Sex
female    0.742038
male      0.188908
Name: Survived, dtype: float64
```

Now compute the proportion of female vs. male survivors of the RMS Titanic, *along with the standard error of the mean*. The **bold** type should give you a hint about the name of the method to compute the standard error. To do this, you'll need to combine the `groupby()` and `agg()` methods!

In [43]:
```
titanic.groupby("Sex")["Survived"].agg(["mean", "sem"])
```

Out[43]:

Sex	mean	sem
female	0.742038	0.02473
male	0.188908	0.01631

What does this tell you about gender roles when the RMS Titanic was sunk?

Women were deemed more important to save.

Compute the proportion of survivors by cabin class and their standard error.

In [44]:
```
titanic.groupby("Pclass")["Survived"].agg(["mean", "sem"])
```

Out[44]:

Pclass	mean	sem
1	0.629630	0.032934
2	0.472826	0.036906
3	0.242363	0.019358

What does this tell you about socio-economic status when the RMS Titanic was sunk?

Those who paid more were more likely to survive.