Warm Up Exercises:

1. Sorting and Subsetting:

Complete all following Task: • Dataset for the Task: "titanic.csv" Following task is common for all the problem:

1. Load the provided dataset and import in pandas DataFrame.
2. Check info of the DataFrame and identify following:

Problem 1 - Sorting:

1. Create a DataFrame called fare that contains only the Fare column of the Titanic dataset. Print the head of the result.
2. Create a DataFrame called class age that contains only the Pclass and Age columns of the Titanic dataset, in that order. Print the head of the result.
3. Create a DataFrame called survived gender that contains the Survived and Sex columns of the Titanic dataset, in that order.Print the head of the result.

```
import pandas as pd
titanic_df = pd.read_csv("/content/drive/MyDrive/Concept of AI/Titanic-Dataset.csv")

print("Dataset Information:")
print(titanic_df.info())

print("\nPreview of the dataset:")
print(titanic_df.head())

fare = titanic_df[['Fare']]
print("\nDataFrame with only the 'Fare' column:")
print(fare.head())

class_age = titanic_df[['Pclass', 'Age']]
print("\nDataFrame with 'Pclass' and 'Age' columns:")
print(class_age.head())

survived_gender = titanic_df[['Survived', 'Sex']]
print("\nDataFrame with 'Survived' and 'Sex' columns:")
print(survived_gender.head())
```

```
  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
None

Preview of the dataset:
   PassengerId Survived Pclass \
0 1 0 3
1 2 1 1
2 3 1 3
3 4 1 1
4 5 0 3

                                                            Name Sex Age SibSp \
0 Braund, Mr. Owen Harris male 22.0 1 1 Cumings, Mrs. John Bradley (Florence
Briggs Th... female 38.0 1 2 Heikkinen, Miss. Laina female 26.0 0 3
Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 1 4 Allen, Mr.
William Henry male 35.0 0

   Parch Ticket Fare Cabin Embarked
0 0 A/5 21171 7.2500 NaN S
1 0 PC 17599 71.2833 C85 C
2 0 STON/O2. 3101282 7.9250 NaN S
3 0 113803 53.1000 C123 S
4 0 373450 8.0500 NaN S

DataFrame with only the 'Fare' column:
      Fare
0 7.2500
1 71.2833
2 7.9250
3 53.1000
4 8.0500
   Survived Sex
0 0 male
1 1 female
2 1 female
3 1 female
4 0 male
```

Problem - 2 - Subsetting:

Complete all the following Task: Subsetting Rows:

1. Filter the Titanic dataset for cases where the passenger’s fare is greater than 100, assigning

it to fare gt 100.View the printed result.

2. Filter the Titanic dataset for cases where the passenger’s class (Pclass) is 1, assigning it to first class.View the printed result.
3. Filter the Titanic dataset for cases where the passenger’s age is less than 18 and the passenger is female (Sex is "female"), assigning it to female under 18. View the printed result.

Subsetting Rows by Categorical variables:

1. Filter the Titanic dataset for passengers whose Embarked port is either "C" (Cherbourg) or "S" (Southampton), assigning the result to embarked c or s. View the printed result.
2. Filter the Titanic dataset for passengers whose Pclass is in the list [1, 2] (indicating first or second class), assigning the result to first second class.View the printed result.

```
fare_gt_100 = titanic_df[titanic_df['Fare'] > 100]
print("\nPassengers with fare > 100:")
print(fare_gt_100.head())
first_class = titanic_df[titanic_df['Pclass'] == 1]
print("\nPassengers in first class (Pclass = 1):")
print(first_class.head())
female_under_18 = titanic_df[(titanic_df['Age'] < 18) & (titanic_df['Sex'] == "female")]
print("\nFemale passengers under 18:")
print(female_under_18.head())
embarked_c_or_s = titanic_df[titanic_df['Embarked'].isin(['C', 'S'])]
print("\nPassengers who embarked at 'C' or 'S':")
print(embarked_c_or_s.head())
first_second_class = titanic_df[titanic_df['Pclass'].isin([1, 2])]
print("\nPassengers in first or second class (Pclass in [1, 2]):")
print(first_second_class.head())
```

```
Passengers with fare > 100:
     PassengerId Survived Pclass \
27 28 0 1
31 32 1 1
88 89 1 1
118 119 0 1
195 196 1 1

                                                          Name Sex Age SibSp \
27  Fortune,  Mr.  Charles  Alexander  male  19.0  3  31  Spencer,  Mrs.  William
Augustus (Marie Eugenie) female NaN 1 88 Fortune, Miss. Mabel Helen female
23.0  3  118  Baxter,  Mr.  Quigg  Edmond  male  24.0  0  195  Lurette,  Miss.  Elise
female 58.0 0

     Parch Ticket Fare Cabin Embarked
```

```
27 2 19950 263.0000 C23 C25 C27 S
31 0 PC 17569 146.5208 B78 C
88 2 19950 263.0000 C23 C25 C27 S
118 1 PC 17558 247.5208 B58 B60 C
195 0 PC 17569 146.5208 B80 C

Passengers in first class (Pclass = 1):
    PassengerId Survived Pclass \
1 2 1 1
3 4 1 1
6 7 0 1
11 12 1 1
23 24 1 1

                                                              Name Sex Age SibSp \
1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 1 3 Futrelle,
Mrs. Jacques Heath (Lily May Peel) female 35.0 1 6 McCarthy, Mr. Timothy J
male 54.0 0 11 Bonnell, Miss. Elizabeth female 58.0 0 23 Sloper, Mr. William
Thompson male 28.0 0

    Parch Ticket Fare Cabin Embarked
1 0 PC 17599 71.2833 C85 C
3 0 113803 53.1000 C123 S
6 0 17463 51.8625 E46 S
11 0 113783 26.5500 C103 S
23 0 113788 35.5000 A6 S

Female passengers under 18:
     PassengerId Survived Pclass Name \ 9 10 1 2 Nasser, Mrs. Nicholas (Adele
    Achem) 10 11 1 3 Sandstrom, Miss. Marguerite Rut 14 15 0 3 Vestrom, Miss.
       Hulda Amanda Adolfina 22 23 1 3 McGowan, Miss. Anna "Annie" 24 25 0 3
                                                Palsson, Miss. Torborg Danira

       Sex Age SibSp Parch Ticket Fare Cabin Embarked
9 female 14.0 1 0 237736 30.0708 NaN C 10 female 4.0 1 1 PP 9549
16.7000 G6 S 14 female 14.0 0 0 350406 7.8542 NaN S 22 female 15
0 0 0 330923 8 0292 NaN Q
```

Exploratory Data Analysis Practice Exercise - 1.

(Warning: Handle missing values in the Age column by filling them with the median age of the dataset before performing the division.)

Answer the following questions from Dataset:

Which passenger had the highest fare paid relative to their age?

To answer the question perform following operations:

1. Add a column to the Titanic dataset, fare per year, containing the fare divided by the age of the

passenger(i.e., Fare/Age).

2. Subset rows where fare per year is higher than 5, assigning this to high fare age. 3. Sort high fare age by descending fare per year, assigning this to high fare age srt. 4. Select only the Name and fare per year columns of high fare age srt and save the result as result.
5. Look at the result.

```
import pandas as pd

titanic_df = pd.read_csv("/content/drive/MyDrive/Concept of AI/Titanic-Dataset.csv")
median_age = titanic_df['Age'].median()
titanic_df['Age'] = titanic_df['Age'].fillna(median_age)
titanic_df['fare_per_year'] = titanic_df['Fare'] / titanic_df['Age']
high_fare_age = titanic_df[titanic_df['fare_per_year'] > 5]
high_fare_age_srt = high_fare_age.sort_values(by='fare_per_year', ascending=False)
result = high_fare_age_srt[['Name', 'fare_per_year']]
print("\nPassengers with the highest fare per year:")
print(result.head())
```

```
Passengers with the highest fare per year:
                                   Name fare_per_year
305 Allison, Master. Hudson Trevor 164.728261
297 Allison, Miss. Helen Loraine 75.775000
386 Goodwin, Master. Sidney Leonard 46.900000
164 Panula, Master. Eino Viljami 39.687500
183 Becker, Master. Richard F 39.000000
```

Which adult male passenger (age ≥ 18 and Sex is 'male') paid the highest □fare relative to their class?

To answer the question perform following operations:

1. Add a column to the Titanic dataset, fare per class, containing the fare divided by the passenger class i.e. Fare / Pclass.
2. Subset rows where the passenger is male (Sex is ”male”) and an adult (Age is greater than or equal to 18), assigning this to adult males.
3. Sort adult males by descending fare per class, assigning this to adult males srt. 4. Select only the Name, Age, and fare per class columns of adult males sr and save the result as result.
5. Look at the result.

```
import pandas as pd
titanic_df = pd.read_csv("/content/drive/MyDrive/Concept of AI/Titanic-Dataset.csv")
titanic_df['fare_per_class'] = titanic_df['Fare'] / titanic_df['Pclass'] adult_males =
titanic_df[(titanic_df['Sex'] == 'male') & (titanic_df['Age'] >= 18)] adult_males_srt =
adult_males.sort_values(by='fare_per_class', ascending=False) result =
adult_males_srt[['Name', 'Age', 'fare_per_class']]
print("\nAdult males with the highest fare per class:")
print(result.head())
```

```
Adult males with the highest fare per class:
                                         Name Age fare_per_class
737 Lesurer, Mr. Gustave J 35.0 512.3292
679 Cardeza, Mr. Thomas Drake Martinez 36.0 512.3292
27 Fortune, Mr. Charles Alexander 19.0 263.0000
438 Fortune, Mr. Mark 64.0 263.0000
118 Baxter, Mr. Quigg Edmond 24.0 247.5208
```

Exploratory Data Analysis with Group-by Method Practice

□ Exercise:

Based on the dataset Answer the following question:

□ What percent of the total fare revenue came from each passenger class?

To answer the question perform following operation:

1. Calculate the total Fare paid across all passengers in the Titanic dataset.
2. Subset for passengers in first class (Pclass is 1) and calculate their total fare. 3. Do the same for second class (Pclass is 2) and third class (Pclass is 3).
4. Combine the fare totals from first, second, and third classes into a list.
5. Divide the totals for each class by the overall total fare to get the proportion of fare revenue by class.

```
import pandas as pd
titanic_df = pd.read_csv("/content/drive/MyDrive/Concept of AI/Titanic-Dataset.csv")

total_fare = titanic_df['Fare'].sum()
print(f"Total fare across all passengers: {total_fare}")
first_class_fare = titanic_df[titanic_df['Pclass'] == 1]['Fare'].sum()

second_class_fare = titanic_df[titanic_df['Pclass'] == 2]['Fare'].sum()
```

```
third_class_fare = titanic_df[titanic_df['Pclass'] == 3]['Fare'].sum()
fare_totals_by_class = [first_class_fare, second_class_fare, third_class_fare]
print("\nFare totals by class:")
print(f"First class: {first_class_fare}, Second class: {second_class_fare}, Third class:

proportion_of_fare_by_class = [fare / total_fare for fare in fare_totals_by_class]


percentage_of_fare_by_class = [prop * 100 for prop in proportion_of_fare_by_class]
print("\nPercentage of total fare revenue by passenger class:")
print(f"First class: {percentage_of_fare_by_class[0]:.2f}%")
print(f"Second class: {percentage_of_fare_by_class[1]:.2f}%")
print(f"Third class: {percentage_of_fare_by_class[2]:.2f}%")
```

```
Total fare across all passengers: 28693.9493

Fare totals by class:
First class: 18177.4125, Second class: 3801.8417, Third class: 6714.6951

Percentage of total fare revenue by passenger class:
First class: 63.35%
Second class: 13.25%
Third class: 23.40%
```

What percent of the total number of passengers on the Titanic belonged to each age group

(e.g., child, adult, senior)? To answer the question perform following operation:

1. Create a new column, age group, that categorizes passengers into "child" (age < 18), "adult" (age 18{64), and "senior" (age 65 and above).
2. Calculate the total number of passengers on the Titanic.
3. Count the number of passengers in each age group.
4. Divide the count of each age group by the total number of passengers to get the proportion of passengers in each age group.
5. Display the proportion as a percentage.

```
def categorize_age(age):
    if age < 18:
        return 'child'
    elif age < 65:
        return 'adult'
    else:
        return 'senior'

titanic_df['age_group'] = titanic_df['Age'].apply(categorize_age)
```

```
total_passengers = len(titanic_df)
print(f"Total number of passengers on the Titanic: {total_passengers}")

age_group_counts = titanic_df['age_group'].value_counts()

age_group_proportion = (age_group_counts / total_passengers) * 100
print("\nPercentage of passengers in each age group:")
print(age_group_proportion)
```

```
     Total number of passengers on the Titanic: 891

     Percentage of passengers in each age group:
     age_group
     adult 66.217733
     senior 21.099888
     child 12.682379
     Name: count, dtype: float64
```