```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

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.

```
import pandas as pd
titanic = pd.read_csv('/content/drive/MyDrive/NMC-DATASETS/Titanic-Dataset.csv')
# Create a DataFrame with only the "Fare" column
fare = titanic[['Fare']]
print(fare.head())
```

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

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.

```
class_age = titanic[['Pclass', 'Age']]
print(class_age.head())
```

```
   Pclass   Age
0       3  22.0
1       1  38.0
2       3  26.0
3       1  35.0
4       3  35.0
```

3. Create a DataFrame called survived gender that contains the Survived and Sex lumns of the Titanic dataset, in that order.Print the head of the result.

```
Survived_Sex = titanic[['Survived', 'Sex']]
print(Survived_Sex.head())
```

```
   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.

```
fare_gt_100 = titanic[titanic['Fare'] > 100]
print(fare_gt_100)
```

```
     PassengerId  Survived  Pclass  \
27            28         0       1
31            32         1       1
88            89         1       1
118          119         0       1
195          196         1       1
215          216         1       1
258          259         1       1
268          269         1       1
269          270         1       1
297          298         0       1
299          300         1       1
```

```
305          306         1       1
306          307         1       1
307          308         1       1
311          312         1       1
318          319         1       1
319          320         1       1
325          326         1       1
332          333         0       1
334          335         1       1
337          338         1       1
341          342         1       1
373          374         0       1
377          378         0       1
380          381         1       1
390          391         1       1
393          394         1       1
435          436         1       1
438          439         0       1
498          499         0       1
505          506         0       1
527          528         0       1
537          538         1       1
544          545         0       1
550          551         1       1
557          558         0       1
581          582         1       1
609          610         1       1
659          660         0       1
660          661         1       1
679          680         1       1
689          690         1       1
698          699         0       1
700          701         1       1
708          709         1       1
716          717         1       1
730          731         1       1
737          738         1       1
742          743         1       1
763          764         1       1
779          780         1       1
802          803         1       1
856          857         1       1

                                                 Name     Sex    Age  SibSp  \
27                      Fortune, Mr. Charles Alexander    male  19.00      3
31      Spencer, Mrs. William Augustus (Marie Eugenie)  female    NaN      1
```

2. Filter the Titanic dataset for cases where the passenger’s class (Pclass) is 1, assigning it to first class.View the printed result.

```
First_Class = titanic[titanic['Pclass'] == 1]
print(fare_gt_100)
```

```
390      2    113760  120.0000          B96 B98        S
393      0     35273  113.2750              D36        C
435      2    113760  120.0000          B96 B98        S
438      4     19950  263.0000      C23 C25 C27        S
498      2    113781  151.5500          C22 C26        S
505      0  PC 17758  108.9000              C65        C
527      0  PC 17483  221.7792              C95        S
537      0  PC 17761  106.4250              NaN        C
544      0  PC 17761  106.4250              C86        C
550      2     17421  110.8833              C70        C
557      0  PC 17757  227.5250              NaN        C
581      1     17421  110.8833              C68        C
609      0  PC 17582  153.4625             C125        S
659      2     35273  113.2750              D48        C
660      0  PC 17611  133.6500              NaN        S
679      1  PC 17755  512.3292      B51 B53 B55        C
689      1     24160  211.3375               B5        S
698      1     17421  110.8833              C68        C
700      0  PC 17757  227.5250          C62 C64        C
708      0    113781  151.5500              NaN        S
716      0  PC 17757  227.5250              C45        C
730      0     24160  211.3375               B5        S
737      0  PC 17755  512.3292             B101        C
742      2  PC 17608  262.3750  B57 B59 B63 B66        C
763      2    113760  120.0000          B96 B98        S
779      1     24160  211.3375               B3        S
802      2    113760  120.0000          B96 B98        S
856      1     36928  164.8667              NaN        S
```

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.

```
Female_Under_18 = titanic[(titanic['Age'] < 18) & (titanic['Sex'] == 'female')]
print(Female_Under_18)
```

```
    720      1         248727   33.0000      NaN        S
    750      1          29103   23.0000      NaN        S
    777      0         364516   12.4750      NaN        S
    780      0           2687    7.2292      NaN        C
    781      0          17474   57.0000      B20        S
    813      2         347082   31.2750      NaN        S
    830      0           2659   14.4542      NaN        C
    852      1           2678   15.2458      NaN        C
    853      1       PC 17592   39.4000      D28        S
    875      0           2667    7.2250      NaN        C
```

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.

```
embarked_c_or_s = titanic[(titanic['Embarked'] == 'C') | (titanic['Embarked'] == 'S')]
print(embarked_c_or_s)
```

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
884          885         0       3
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1

                                                  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
..                                                 ...     ...   ...    ...
884                             Sutehall, Mr. Henry Jr    male  25.0      0
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      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
..     ...               ...      ...   ...      ...
884      0   SOTON/OQ 392076   7.0500   NaN        S
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C

[812 rows x 12 columns]
```

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.

```
first_second_class = titanic[titanic['Pclass'].isin([1, 2])]
print(first_second_class)
```

```
     PassengerId  Survived  Pclass  \
1              2         1       1
3              4         1       1
6              7         0       1
9             10         1       2
11            12         1       1
..           ...       ...     ...
880          881         1       2
883          884         0       2
886          887         0       2
887          888         1       1
```

```
889         890         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
9                  Nasser, Mrs. Nicholas (Adele Achem)  female  14.0      1
11                            Bonnell, Miss. Elizabeth  female  58.0      0
..                                                 ...     ...   ...    ...
880       Shelley, Mrs. William (Imanita Parrish Hall)  female  25.0      0
883                       Banfield, Mr. Frederick James    male  28.0      0
886                              Montvila, Rev. Juozas    male  27.0      0
887                        Graham, Miss. Margaret Edith  female  19.0      0
889                              Behr, Mr. Karl Howell    male  26.0      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
9        0            237736  30.0708   NaN        C
11       0            113783  26.5500  C103        S
..     ...               ...      ...   ...      ...
880      1            230433  26.0000   NaN        S
883      0  C.A./SOTON 34068  10.5000   NaN        S
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
889      0            111369  30.0000  C148        C

[400 rows x 12 columns]
```

3.2 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.

```
titanic['Age'].fillna(titanic['Age'].median(), inplace=True)

titanic['fare_per_year'] = titanic['Fare'] / titanic['Age']

high_fare_age = titanic[titanic['fare_per_year'] > 5]

high_fare_age_srt = high_fare_age.sort_values('fare_per_year', ascending=False)

result = high_fare_age_srt[['Name', 'fare_per_year']]

print(result)
```

```
                                                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
..                                               ...            ...
348          Coutts, Master. William Loch "William"       5.300000
31   Spencer, Mrs. William Augustus (Marie Eugenie)       5.232886
205                       Strom, Miss. Telma Matilda       5.231250
813               Andersson, Miss. Ebba Iris Alfrida       5.212500
480                   Goodwin, Master. Harold Victor       5.211111

[71 rows x 2 columns]
<ipython-input-12-f6e4e46ebb8f>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignm
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting valu

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].me


  titanic['Age'].fillna(titanic['Age'].median(), inplace=True)
```

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.

```
titanic['fare_per_class'] = titanic['Fare'] / titanic['Pclass']

adult_males = titanic[(titanic['Sex'] == 'male') & (titanic['Age'] >= 18)]

adult_males_srt = adult_males.sort_values('fare_per_class', ascending=False)

result = adult_males_srt[['Name', 'Age', 'fare_per_class']]

print(result)
```

```
                                    Name   Age  fare_per_class
737             Lesurer, Mr. Gustave J  35.0        512.3292
679  Cardeza, Mr. Thomas Drake Martinez  36.0        512.3292
438                  Fortune, Mr. Mark  64.0        263.0000
27       Fortune, Mr. Charles Alexander  19.0        263.0000
118            Baxter, Mr. Quigg Edmond  24.0        247.5208
..                                   ...   ...             ...
806             Andrews, Mr. Thomas Jr  39.0          0.0000
481    Frost, Mr. Anthony Wood "Archie"  28.0          0.0000
413      Cunningham, Mr. Alfred Fleming  28.0          0.0000
466               Campbell, Mr. William  28.0          0.0000
271        Tornquist, Mr. William Henry  25.0          0.0000

[519 rows x 3 columns]
```

3.3 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.

```
total_fare = titanic['Fare'].sum()

fare_first_class = titanic[titanic['Pclass'] == 1]['Fare'].sum()
fare_second_class = titanic[titanic['Pclass'] == 2]['Fare'].sum()
fare_third_class = titanic[titanic['Pclass'] == 3]['Fare'].sum()

fare_totals = [fare_first_class, fare_second_class, fare_third_class]

fare_proportions = [fare / total_fare * 100 for fare in fare_totals]

print("Percentage of Total Fare Revenue by Class:")
print(f"First Class: {fare_proportions[0]:.2f}%")
print(f"Second Class: {fare_proportions[1]:.2f}%")
print(f"Third Class: {fare_proportions[2]:.2f}%")
```

```
Percentage of Total Fare Revenue by Class:
First Class: 63.35%
Second Class: 13.25%
Third Class: 23.40%
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

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 age_group(age):
    if pd.isna(age):
        return "unknown"
    elif age < 18:
        return "child"
    elif age < 65:
        return "adult"
    else:
        return "senior"

titanic['AgeGroup'] = titanic['Age'].apply(age_group)

total_passengers = len(titanic)

age_group_counts = titanic['AgeGroup'].value_counts()

age_group_proportions = (age_group_counts / total_passengers) * 100

print("\nPercentage of Passengers by Age Group:")
for group, proportion in age_group_proportions.items():
    print(f"{group.capitalize()}: {proportion:.2f}%")
```

```
Percentage of Passengers by Age Group:
Adult: 66.22%
Unknown: 19.87%
Child: 12.68%
Senior: 1.23%
```