5CS037 - Concepts and Technologies of AI.
Worksheet-2: Exploratory Data Analysis with Pandas -Part-1.

Prepared By: Siman Giri {Module Leader - 5CS037}

November 5, 2024

To - Do - Task

Please Complete all the problem listed below.

0.1 Warming Up Exercises - Basic Inspection and Exploration:

Problem 1 - Data Read, Write and Inspect:

Complete all following Task:

- Dataset for the Task: "bank.csv"

1. Load the provided dataset and import in pandas DataFrame.

2. Check info of the DataFrame and identify following:

   (a) columns with dtypes=object

```
#Check info of the DataFrame and identify following:
df_info = df.info()
#(a) columns with dtypes=object
object_col = df.select_dtypes(include=['object']).columns
print("Column with datatypes objects",object_col)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        45211 non-null  int64
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64
 10  month      45211 non-null  object
 11  duration   45211 non-null  int64
 12  campaign   45211 non-null  int64
 13  pdays      45211 non-null  int64
 14  previous   45211 non-null  int64
 15  poutcome   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
Column with datatypes objects Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact'
       'month', 'poutcome', 'y'],
      dtype='object')
```

(b) unique values of those columns.

```
#unique values of those columns.
unique_values = {col: df[col].unique() for col in object_col}
print(unique_values)
```

```
{'job': array(['management', 'technician', 'entrepreneur', 'blue-collar',
       'unknown', 'retired', 'admin.', 'services', 'self-employed',
       'unemployed', 'housemaid', 'student'], dtype=object), 'marital': array(['married', 'single', 'divorced'], dtype=object), 'educ
       'mar', 'apr', 'sep'], dtype=object), 'poutcome': array(['unknown', 'failure', 'other', 'success'], dtype=object), 'y': array([
```

(c) check for the total number of null values in each column.

```
#check for the total number of null values in each column.
null_values = df.isnull().sum()
print(null_values)
```

```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

3. Drop all the columns with dtypes object and store in new DataFrame, also write the DataFrame in ".csv" with name "banknumericdata.csv"

```
#Drop all the columns with dtypes object and store in new DataFrame, also write the DataFrame in".csv" with name "banknumericdata.csv"
df_numeric = df.select_dtypes(exclude=['object'])
df_numeric.to_csv('banknumericdata.csv',index=False)
```

4. Read "banknumericdata.csv" and Find the summary statistics.

Problem 2 - Data Imputations:

Complete all the following Task:

- Dataset for the Task: "medical_student.csv"

1. Load the provided dataset and import in pandas DataFrame.

Problem 2 - Data Imputations:

Dataset for the Task: "medical_student.csv"

```
#Load the provided dataset and import in pandas DataFrame
import pandas as pd
medical_student_df = pd.read_csv('/content/drive/MyDrive/Concept of AI -- week 2/Copy of medical_students_dataset.csv
print(medical_student_df.head())
```

```
   Student ID   Age  Gender      Height     Weight Blood Type        BMI  \
0         1.0  18.0  Female  161.777924  72.354947          O  27.645835
1         2.0   NaN    Male  152.069157  47.630941          B        NaN
2         3.0  32.0  Female  182.537664  55.741083          A  16.729017
3         NaN  30.0    Male  182.112867  63.332207          B  19.096042
4         5.0  23.0  Female         NaN  46.234173          O        NaN

   Temperature  Heart Rate  Blood Pressure  Cholesterol Diabetes Smoking
0          NaN        95.0           109.0        203.0       No     NaN
1    98.714977        93.0           104.0        163.0       No      No
2    98.260293        76.0           130.0        216.0      Yes      No
3    98.839605        99.0           112.0        141.0       No     Yes
4    98.480008        95.0             NaN        231.0       No      No
```

2. Check info of the DataFrame and identify column with missing (null) values.

```
#Check info of the DataFrame and identify column with missing (null) values
print(medical_student_df.info())

print(medical_student_df.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 13 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Student ID      180000 non-null  float64
 1   Age             180000 non-null  float64
 2   Gender          180000 non-null  object
 3   Height          180000 non-null  float64
 4   Weight          180000 non-null  float64
 5   Blood Type      180000 non-null  object
 6   BMI             180000 non-null  float64
 7   Temperature     180000 non-null  float64
 8   Heart Rate      180000 non-null  float64
 9   Blood Pressure  180000 non-null  float64
 10  Cholesterol     180000 non-null  float64
 11  Diabetes        180000 non-null  object
 12  Smoking         180000 non-null  object
dtypes: float64(9), object(4)
memory usage: 19.8+ MB
None
Student ID        20000
Age               20000
Gender            20000
Height            20000
Weight            20000
Blood Type        20000
```

3. For the column with missing values fill the values using various techniques we discussed above. Try to explain why did you select the particular methods for particular column.

```
#For the column with missing values fill the values using various techniques we discussed above. Try to explain why did you select the parti
# Example imputation
for column in medical_student_df.columns:
    if medical_student_df[column].isnull().sum() > 0:
        if medical_student_df[column].dtype == "float64" or medical_student_df[column].dtype == "int64":
            # Numerical column: fill missing values with the median (preferred for skewed distributions)
            medical_student_df[column].fillna(medical_student_df[column].median(), inplace=True)
        else:
            # Categorical column: fill missing values with mode
            medical_student_df[column].fillna(medical_student_df[column].mode()[0], inplace=True)
```

```
<ipython-input-12-23348de8b48e>:7: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values a

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


  medical_student_df[column].fillna(medical_student_df[column].median(), inplace=True)
<ipython-input-12-23348de8b48e>:10: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values a

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


  medical_student_df[column].fillna(medical_student_df[column].mode()[0], inplace=True)
```

4. Check for any duplicate values present in Dataset and do necessary to manage the duplicate items. {Hint: dataset.duplicated.sum()}

```
[13] #Check for any duplicate values present in Dataset and do necessary to manage the duplicate items.{Hint: dataset.duplicated.sum()}

     # Count duplicate rows
     num_duplicates = medical_student_df.duplicated().sum()
     print(f"Number of duplicate rows: {num_duplicates}")

     # Drop duplicate rows if any
     if num_duplicates > 0:
         medical_student_df = medical_student_df.drop_duplicates()
         print("Duplicates removed.")
```

```
Number of duplicate rows: 12879
Duplicates removed.
```

0.2 Exercises - Data Cleaning and Transformations with ”Titanic Dataset”:

Dataset Used: "titanic.csv"

Problem - 1:

Create a DataFrame that is subsetted for the columns ’Name’, ’Pclass’, ’Sex’, ’Age’, ’Fare’, and ’Survived’. Retain only those rows where ’Pclass’ is equal to 1, representing first-class passengers. What is the mean, median, maximum value, and minimum value of the ’Fare’ column?

Worksheet - 2: Exploratory Data Analysis with Pandas - Part - 1.

Siman Giri

Exercises - Data Cleaning and Transformations with "Titanic Dataset":

```
import pandas as pd
titanic_df = pd.read_csv('/content/drive/MyDrive/Concept of AI -- week 2/Copy of Titanic-Dataset.csv')
#Create a DataFrame that is subsetted for the columns 'Name', 'Pclass', 'Sex', 'Age', 'Fare', and 'Survived'.
df_subset = titanic_df[['Name','Pclass','Sex','Age','Fare','Survived']]
#Retain only those rows where 'Pclass' is equal to 1, representing first-class passengers.
first_class_df = df_subset[df_subset['Pclass'] == 1]
print(first_class_df)
#What is the mean,median, maximum value, and minimum value of the 'Fare' column?
fare_mean = first_class_df['Fare'].mean()
fare_median = first_class_df['Fare'].median()
fare_max = first_class_df['Fare'].max()
fare_min = first_class_df['Fare'].min()
print("Mean:",fare_mean)
print("Median:",fare_median)
print("Max:",fare_max)
print("Min:",fare_min)
```

```
                                                  Name  Pclass     Sex   Age  \
1    Cumings, Mrs. John Bradley (Florence Briggs Th...       1  female  38.0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)       1  female  35.0
6                              McCarthy, Mr. Timothy J       1    male  54.0
11                            Bonnell, Miss. Elizabeth       1  female  58.0
23                         Sloper, Mr. William Thompson       1    male  28.0
..                                                  ...     ...     ...   ...
871   Beckwith, Mrs. Richard Leonard (Sallie Monypeny)       1  female  47.0
872                            Carlsson, Mr. Frans Olof       1    male  33.0
879      Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)       1  female  56.0
887                         Graham, Miss. Margaret Edith       1  female  19.0
889                               Behr, Mr. Karl Howell       1    male  26.0

        Fare  Survived
1    71.2833         1
3    53.1000         1
6    51.8625         0
11   26.5500         1
23   35.5000         1
..       ...       ...
871  52.5542         1
872   5.0000         0
879  83.1583         1
887  30.0000         1
889  30.0000         1

[216 rows x 6 columns]
Mean: 84.1546875
Median: 60.287499999999994
Max: 512.3292
Min: 0.0
```

Problem - 2:

How many null values are contained in the 'Age' column in your subsetted DataFrame? Once you've found this out, drop them from your DataFrame.

```
#How many null values are contained in the 'Age' column in your subsetted DataFrame? Once you've foundthis out, drop them from your DataFram
null_age_count = first_class_df['Age'].isnull().sum()
print("Null values in Age column:",null_age_count)
first_class_df_cleaned = first_class_df.dropna(subset=['Age'])
print(first_class_df)
```

```
Null values in Age column: 30
                                                Name  Pclass     Sex   Age  \
1    Cumings, Mrs. John Bradley (Florence Briggs Th...       1  female  38.0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)       1  female  35.0
6                              McCarthy, Mr. Timothy J       1    male  54.0
11                            Bonnell, Miss. Elizabeth       1  female  58.0
23                         Sloper, Mr. William Thompson       1    male  28.0
..                                                  ...     ...     ...   ...
871   Beckwith, Mrs. Richard Leonard (Sallie Monypeny)       1  female  47.0
872                            Carlsson, Mr. Frans Olof       1    male  33.0
879      Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)       1  female  56.0
887                         Graham, Miss. Margaret Edith       1  female  19.0
889                                Behr, Mr. Karl Howell       1    male  26.0

        Fare  Survived
1    71.2833         1
3    53.1000         1
6    51.8625         0
11   26.5500         1
23   35.5000         1
..       ...       ...
871  52.5542         1
872   5.0000         0
879  83.1583         1
887  30.0000         1
889  30.0000         1

[216 rows x 6 columns]
```

Problem - 3:

The 'Embarked' column in the Titanic dataset contains categorical data representing the ports of embarkation:

- 'C' for Cherbourg
- 'Q' for Queenstown
- 'S' for

Southampton Task:

1. Use one-hot encoding to convert the 'Embarked' column into separate binary columns ('Embarked C', 'Embarked Q', 'Embarked S').

2. Add these new columns to the original DataFrame.

3. Drop the original 'Embarked' column.

4. Print the first few rows of the modified DataFrame to verify the changes.

```
#Use one-hot encoding to convert the 'Embarked' column into separate binary columns ('Embarked C','Embarked Q', 'Embar
titanic_df = pd.read_csv('/content/drive/MyDrive/Concept of AI -- week 2/Copy of Titanic-Dataset.csv')
embarked_dummies = pd.get_dummies(titanic_df['Embarked'], prefix='Embarked')
titanic_df = pd.concat([titanic_df,embarked_dummies], axis=1)
titanic_df = titanic_df.drop('Embarked', axis=1)
print(titanic_df.head())
```

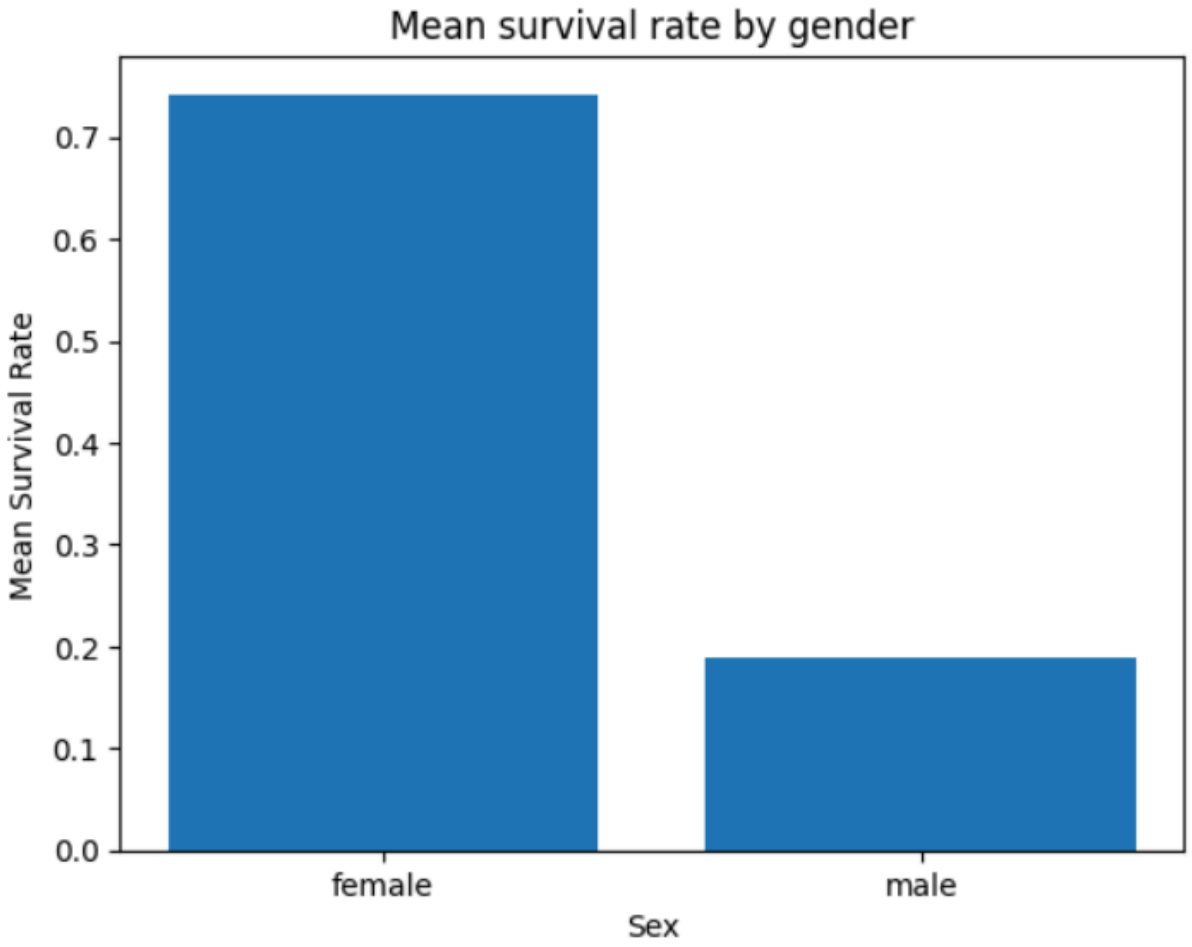
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                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_C  Embarked_Q  Embarked_S
0      0         A/5 21171   7.2500   NaN       False       False        True
1      0          PC 17599  71.2833   C85        True       False       False
2      0  STON/O2. 3101282   7.9250   NaN       False       False        True
3      0            113803  53.1000  C123       False       False        True
4      0            373450   8.0500   NaN       False       False        True
```

Problem - 4:

Compare the mean survival rates ('Survived') for the different groups in the 'Sex' column. Draw a visual- ization to show how the survival distributions vary by gender.

```
#Compare the mean survival rates ('Survived') for the different groups in the 'Sex' column. Draw a visual-ization to show how the survival dis
import pandas as pd
import matplotlib.pyplot as plt

titanic_df = pd.read_csv('/content/drive/MyDrive/Concept of AI -- week 2/Copy of Titanic-Dataset.csv')

mean_survival_rate = titanic_df.groupby('Sex')['Survived'].mean()
print("Mean survival rate by gender:")
print(mean_survival_rate)

plt.bar(mean_survival_rate.index, mean_survival_rate.values)
plt.xlabel('Sex')
plt.ylabel('Mean Survival Rate')
plt.title('Mean survival rate by gender')
plt.show()
```

```
Mean survival rate by gender:
Sex
female    0.742038
male      0.188908
Name: Survived, dtype: float64
```



Problem - 5:

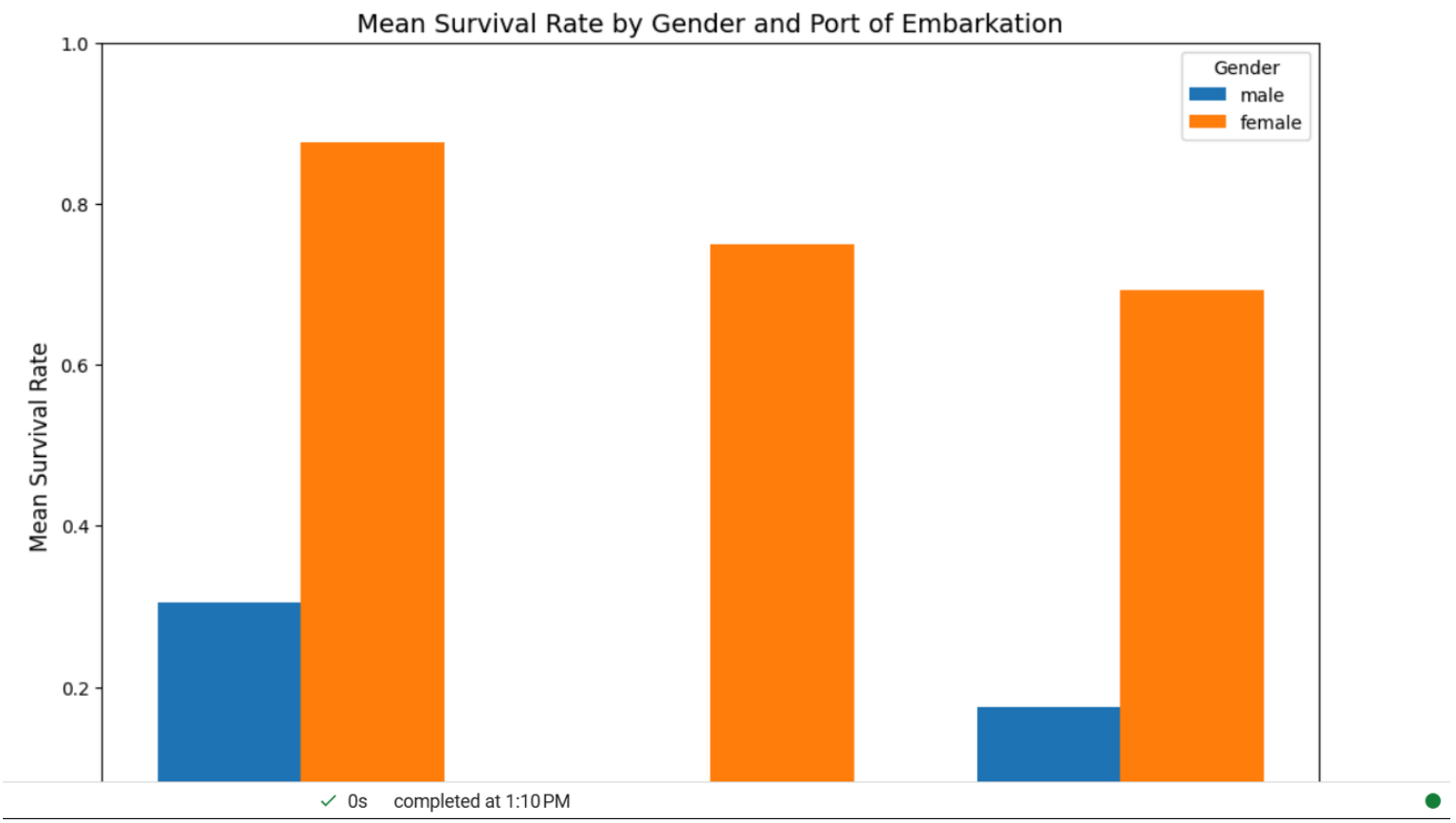
Draw a visualization that breaks your visualization from Exercise 3 down by the port of embarkation ('Embarked'). In this instance, compare the ports 'C' (Cherbourg), 'Q' (Queenstown), and 'S' (Southampton).

Worksheet - 2: Exploratory Data Analysis with Pandas - Part - 1.

Siman Giri

```
#Draw a visualization that breaks your visualization from Exercise 3 down by the port of embarkation (’Em-barked’). In this instance, compare t
import pandas as pd
import matplotlib.pyplot as plt

titanic_df = pd.read_csv('/content/drive/MyDrive/Concept of AI -- week 2/Copy of Titanic-Dataset.csv')

titanic_df['Embarked'] = titanic_df['Embarked'].fillna('S')

mean_survival_rate = titanic_df.groupby(['Embarked', 'Sex'])['Survived'].mean()


ports = ['C', 'Q', 'S']
genders = ['male', 'female']
x = range(len(ports))
width = 0.35


fig, ax = plt.subplots(figsize=(12, 8))

for i, gender in enumerate(genders):
    ax.bar(
        [pos + i * width for pos in x],
        [mean_survival_rate.get((port, gender), 0) for port in ports],
        width=width,
        label=gender
    )
```

```
# Customize plot
ax.set_title("Mean Survival Rate by Gender and Port of Embarkation", fontsize=14)
ax.set_xlabel("Port of Embarkation", fontsize=12)
ax.set_ylabel("Mean Survival Rate", fontsize=12)
ax.set_xticks([pos + width / 2 for pos in x])
ax.set_xticklabels(ports)
ax.legend(title="Gender")
plt.ylim(0, 1)  # Survival rates range between 0 and 1
plt.show()
```



Problem - 6{Optional}:

Show how the survival rates ('Survived') vary by age group and passenger class ('Pclass'). Break up the 'Age' column into five quantiles in your DataFrame, and then compare the means of 'Survived' by class and age group. Draw a visualization using a any plotting library to represent this graphically.

———————— The - End ————————-