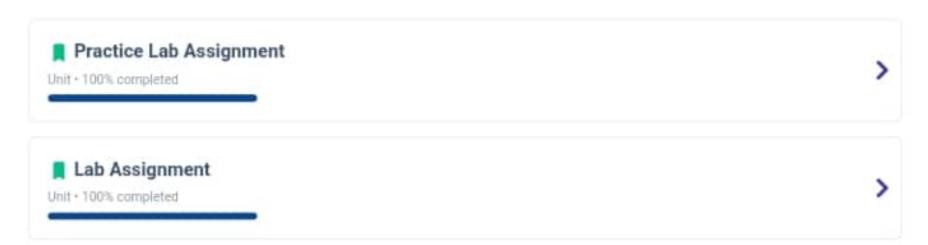


# Practical 4

# About this unit

Practical 4





Write a Python program that takes a list of numbers from the user, creates a Pandas series from it, and then calculates the mean of even and odd numbers separately using the groupby and mean() operations.

4.1.1. Pandas - series creation and manipul... (155) A L Z 2 -

# Input Format:

 The user should enter a list of numbers separated by space when prompted.

### **Output Format:**

- · The program should display the mean of even and odd numbers separately.
- Each mean value should be displayed with a label indicating whether it corresponds to even or odd numbers.



Sample Test Cases



4.1.2. Dictionary to dataframe



A dictionary of lists has been provided to you in the editor. Create a DataFrame from the dictionary of lists and perform the listed operations, then display the DataFrame before and after each manipulation.

#### Create the DataFrame:

· Convert the dictionary to a Pandas DataFrame.

#### Add a new row:

- Take inputs from the user for the new row data (name, age).
- Add the new row to the DataFrame.
- Display the DataFrame after adding the new row.

### Modify a row:

- Modify a specific row by changing the age. Take the row index and new age value from the user.
- . Display the DataFrame after modifying the row.

#### Delete a row:

- Take the row index to be deleted from the user.
- · Remove the specified row.
- . Display the DataFrame after deleting the row.

### Add a new column:

- Add a column Gender with values taken from the user.
- Display the DataFrame after adding the new column.

### Modify a column:

- Convert names to uppercase.
- Display the DataFrame after modifying the column.

### Delete a column:

- · Remove the Age column.
- . Display the DataFrame after deleting the column.

Explorer dataframe... Debugger import pandas as pd 2 -3 # Provided dictionary of lists 4 v data = { 'Name': ['Alice', 'Bob', 5 'Charlie'], 6 'Age': [25, 30, 35], 7 8 9 # Convert the dictionary to a DataFrame 10 df = pd.DataFrame(data) 11 12 # Display the original DataFrame 13 print("Original DataFrame:") 14 print(df) 15 16 # Adding a new row new\_name = input("New name: ") 17 18 new\_age = int(input("New age: ")) 19 df.loc[len(df)] = [new\_name, new\_age] 20 21 # Display the DataFrame after adding a new row print("After adding a row:\n",df) 22 23 # Modifying a row 24 mod\_index = int(input("Index of row 25 to modify: ")) 26 new\_age = int(input("New age: ")) 27 v if 0 <= mod\_index < len(df):</pre> 28 df.at[mod\_index, 'Age'] = new\_age 29 v else: 30 print("Invalid index!:") 31 # Display the DataFrame after 32 modifying a row print("After modifying a row:") 33 34 print(df) 35 36 # Deleting a row 37 del\_index = int(input("Index of row to delete: ")) v if 0 <= del\_index < len(df):</pre> 38 39 → df = df.drop(del\_index).reset\_index(drop=T rue) else: 40 41 print("Invalid index!") 42 43 # Display the DataFrame after deleting a row 44 print("After deleting a row:") print(df) 45 46 47 # Adding a new column genders = input("Enter genders 48 separated by space: ").split() 49 df['Gender'] = genders 50 51 # Display the DataFrame after adding a new column print("After adding a new column:") 52 print(df) 53 54 55 # Modifying a column 56 df['Name'] = df['Name'].str.upper() 57 58 # Display the DataFrame after modifying a column print("After modifying a column:") 59 print(df) 60

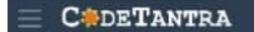
# Dalating a calumn

61

63

**X**.

=



4.1.2. Dictionary to dataframe

A dictionary of lists has been provided to you in the editor. Create a DataFrame from the dictionary of lists and perform the listed operations, then display the DataFrame before and after each manipulation.

ORD ALEBO

### Create the DataFrame:

Convert the dictionary to a Pandas DataFrame.

#### Add a new row:

- Take inputs from the user for the new row data (name, age).
- Add the new row to the DataFrame.
- Display the DataFrame after adding the new row.

### Modify a row:

- Modify a specific row by changing the age. Take the row index and new age value from the user.
- Display the DataFrame after modifying the row.

#### Delete a row:

- Take the row index to be deleted from the user.
- · Remove the specified row.
- Display the DataFrame after deleting the row.

### Add a new column:

- Add a column Gender with values taken from the user.
- Display the DataFrame after adding the new column.

### Modify a column:

- Convert names to uppercase.
- Display the DataFrame after modifying the column.

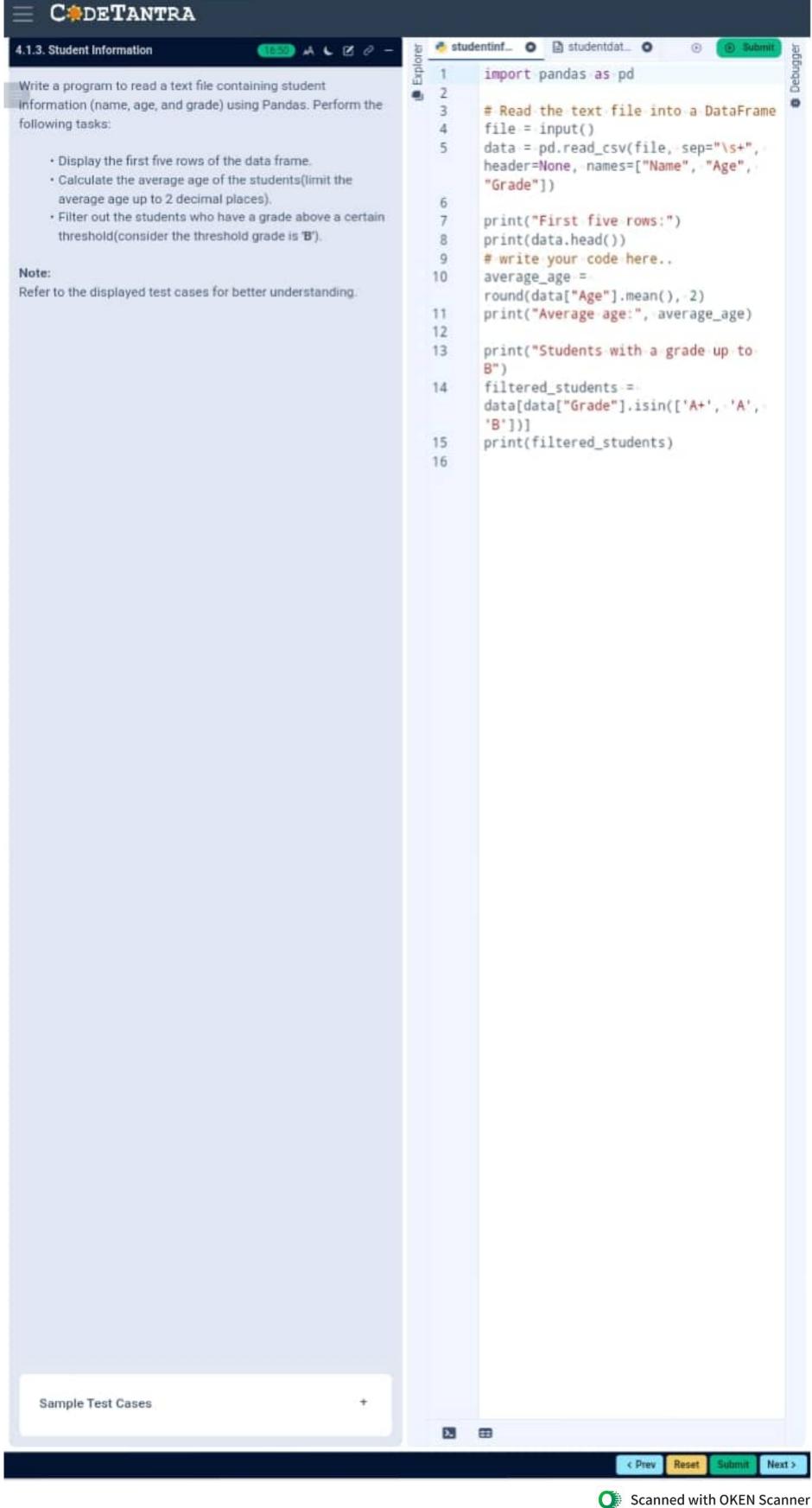
### Delete a column:

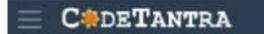
- · Remove the Age column.
- Display the DataFrame after deleting the column.

Explorer dataframe... 8 -9 # Convert the dictionary to a DataFrame 10 df = pd.DataFrame(data) 11 12 # Display the original DataFrame 13 print("Original DataFrame:") 14 print(df) 15 16 # Adding a new row 17 new\_name = input("New name: ") new\_age = int(input("New age:-")) 18 19 df.loc[len(df)] = [new\_name, new\_age] 20 21 # Display the DataFrame after adding a new row 22 print("After adding a row:\n",df) 23 24 # Modifying a row 25 mod\_index = int(input("Index of row to modify: ")) 26 new\_age = int(input("New age: ")) if 0 <= mod\_index < len(df):</pre> 27 28 df.at[mod\_index, 'Age'] = new\_age v else: 29 print("Invalid index!:") 30 31 32 # Display the DataFrame after modifying a row print("After modifying a row:") 33 34 print(df) 35 36 # Deleting a row del\_index = int(input("Index of row 37 to delete: ")) v if 0 <= del\_index < len(df):</pre> 38 df = 39 df.drop(del\_index).reset\_index(drop=T rue) else: 40 41 print("Invalid index!") 42 43 # Display the DataFrame after deleting a row 44 print("After deleting a row:") 45 print(df) 46 47 # Adding a new column genders = input("Enter genders 48 separated by space: ").split() 49 df['Gender'] = genders 50 51 # Display the DataFrame after adding a new column print("After adding a new column:") 52 53 print(df) 54 55 # Modifying a column df['Name'] = df['Name'].str.upper() 56 57 58 # Display the DataFrame aftermodifying a column print("After modifying a column:") 59 60 print(df) 61 # Deleting a column 62 df.drop('Age', axis=1, inplace=True) 63 64 # Display the DataFrame after 65 deleting a column print("After deleting a column:") 66 67 print(df) 68

**X**.

=





each month.

4.2.1. Month with the Highest Total Sales

Write a Python program that takes the file name of a CSV file as

input, reads the data, and performs the following operations: . The CSV file contains the columns: Date, Product,

- Quantity, Price, and City. . Group the data by Month and calculate the total sales for
- Find the month with the highest total sales and display it.
- · Also, display the total sales for the best month.

### Sample Data:

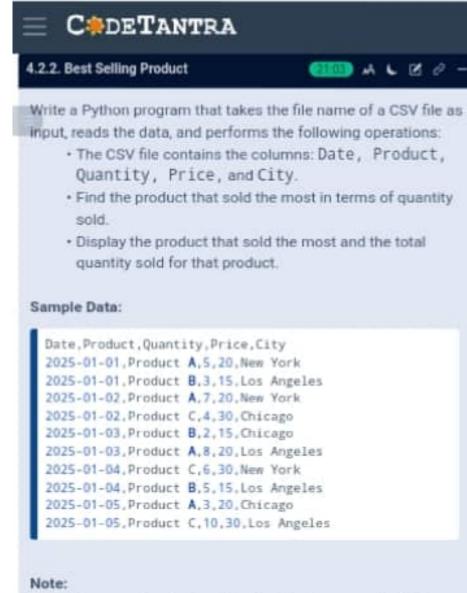
```
Date, Product, Quantity, Price, City
2025-01-01, Product A, S, 20, New York
2025-01-01, Product B, 3, 15, Los Angeles
2025-01-02, Product A, 7, 20, New York
2025-01-02, Product C, 4, 30, Chicago
2025-01-03, Product B, 2, 15, Chicago
2025-01-03, Product A, 8, 20, Los Angeles
2025-01-04, Product C, 6, 30, New York
2025-01-04, Product B, 5, 15, Los Angeles
2025-01-05, Product A, 3, 20, Chicago
2025-01-05, Product C, 10, 30, Los Angeles
```

### Note:

The data cannot be displayed in the file. You can refer to the sample data provided for insights.

Explorer monthForS... O is sales\_data... O import pandas as pd 2 # Prompt the user for the file name 3 file name = input() 4 5 6 # Load the data 7 df = pd.read\_csv(file\_name) 8 df['Date'] = 9 pd.to\_datetime(df['Date']) df['Month'] = 10 df['Date'].dt.to\_period('M').astype(s df['Total'] = df['Quantity'] \* 11 df['Price'] 12 monthly\_sales = df.groupby('Month') 13 ['Total'].sum() 14 # Find the month with the highest 15 total sales best\_month = monthly\_sales.idxmax() 16 highest\_sales = monthly\_sales.max() 17 18 19 print(f"Best month: {best\_month}") print(f"Total sales: 20 \${highest\_sales:.2f}") 21

A ===



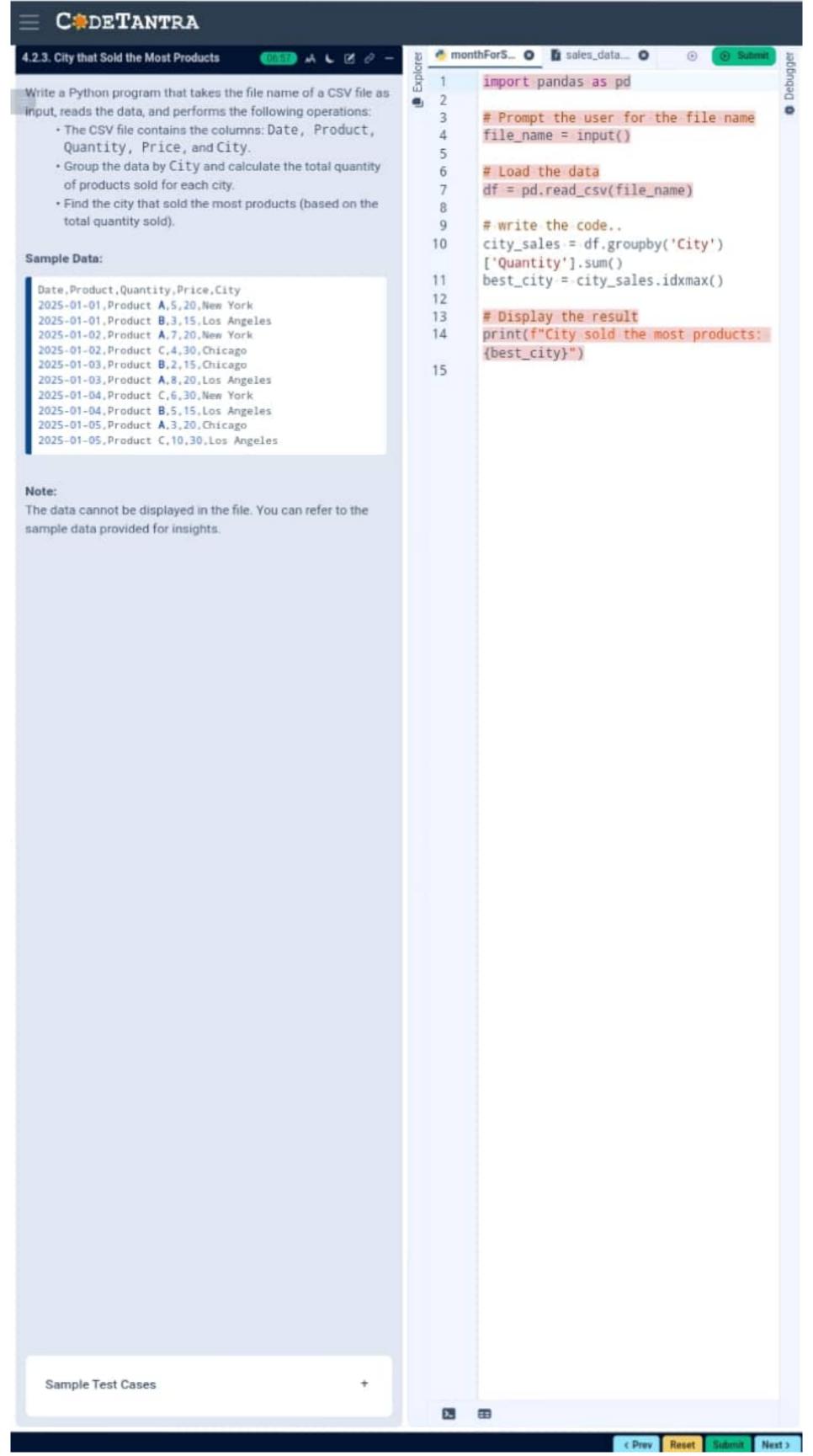
# Explorer monthForS... O is sales\_data... O import pandas as pd 2 # Prompt the user for the file name 3 4 file\_name = input() 5 6 # Load the data df = pd.read\_csv(file\_name) 7 8 product\_sales = df.groupby('Product') ['Quantity'].sum() # Find the product with the highest 9 total quantity sold best\_product = product\_sales.idxmax() 10 highest\_quantity = 11 product\_sales.max() 12 13 # Display the result print(f"Best selling product: 14 {best\_product}") print(f"Total quantity sold: 15 {highest\_quantity}") 16

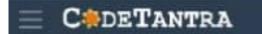
The data cannot be displayed in the file. You can refer to the sample data provided for insights.

Sample Test Cases

A ===







4.2.4. Most Frequently Sold Product Pairs



Write a Python program that takes the file name of a CSV file as input, reads the data, and performs the following operations:

- . The CSV file contains the following columns: Date, Product, Quantity, Price, and City.
- . For each date, find all pairs of products that were sold together (i.e., two products sold on the same date).
- Output the product pair/s that was sold most frequently.

### Sample Data:

```
Date, Product, Quantity, Price, City
2025-01-01, Product A, 5, 20, New York
2025-01-01, Product B, 3, 15, Los Angeles
2025-01-02, Product A, 7, 20, New York
2025-01-02, Product C,4,30, Chicago
2025-01-03, Product B, 2, 15, Chicago
2025-01-03, Product A, 8, 20, Los Angeles
2025-01-04, Product C, 6, 30, New York
2025-01-04, Product B, 5, 15, Los Angeles
2025-01-05, Product A, 3, 20, Chicago
2025-01-05, Product C, 10, 30, Los Angeles
```

#### Explanation:

### Transactions:

- 2025-01-01: Product A, Product B
- · 2025-01-02: Product A, Product C
- 2025-01-03: Product B, Product A
- . 2025-01-04: Product C. Product B
- 2025-01-05: Product A. Product C.

Now, let's count how often the pairs of products appear together:

- . Product A and Product B: Appear in transactions on 2025-01-01 and 2025-01-03.
- · Product A and Product C: Appear in transactions on 2025-01-02 and 2025-01-05.
- · Product B and Product C: Appears in transactions on 2025-01-04.

Most Frequent Product Combinations:

- Product A and Product B (2 times)
- · Product A and Product C (2 times)

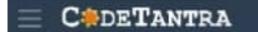
# Note:

The data cannot be displayed in the file. You can refer to the sample data provided for insights.

Explorer frequently... O is sales\_data... O import pandas as pd 2 from itertools import combinations 3 from collections import Counter 4 5 # Prompt user to input the file name 6 file\_name = input() 7 8 # Read data from the specified CSV file df = pd.read csv(file name) 9 10 # write the code 11 date\_products = {} 12 13 v for date, group in 14 df.groupby('Date'): products = 15 group['Product'].unique() if len(products) > 1: 16 date\_products[date] = 17 products 18 19 pair\_counter = Counter() 20 for products in 21 date\_products.values(): 22 pairs = combinations(sorted(products), 2) pair\_counter.update(pairs) 23 24 25 v if pair\_counter: 26 max\_count = max(pair\_counter.values()) 27 28 ofor pair, count in pair\_counter.items(): 29 if count == max count: # # print(f\*{pair[0]} and 30 {pair[1]}: {count} times") 31 else: print("No product pairs found.") 32 33 # Output the most frequent product 34 pairs

**X**.

===



4.2.5. Titanic Dataset Analysis and Data Cle... (IIIII) 🗚 📞 🗷 🔗



titanicData...

You are provided with the Titanic dataset containing Information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset. For each question, perform necessary data cleaning, transformations, and calculations as required.

- Display the first 5 rows of the dataset.
- Display the last 5 rows of the dataset.
- 3. Get the shape of the dataset (number of rows and columns).
- Get a summary of the dataset (using .info()).
- 5. Get basic statistics (mean, standard deviation, etc.) of the dataset using .describe().
- 6. Check for missing values and display the count of missing values for each column.
- 7. Fill missing values in the 'Age' column with the median age
- 8. Fill missing values in the 'Embarked' column with the most frequent value (mode).
- Drop the 'Cabin' column due to many missing values.
- Create a new column, 'FamilySize' by adding the 'SibSp' and 'Parch' columns.

The Titanic dataset contains columns as shown below.

	P N a a m s e s	Sex	A g e	S I b S p	b a b	T i c k e t	F a r e	C a b i n	E m b a r k e d
--	-----------------	-----	-------------	-----------	-------	-------------	---------	-----------	--------------------------------

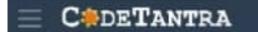
# Sample Data:

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ti 1,0,3,"Braund, Mr. Owen Harris", male, 22,1,0,A/5 21171,7 2,1,1, "Cunings, Mrs. John Bradley (Florence Briggs Thay 3,1,3, "Heikkinen, Miss. Laina", female, 26,0,0,5TON/02, 3 4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",fe 5,0,3, "Allen, Mr, William Henry", male, 35,0,0,373450,8.0 6,0,3, "Moran, Mr. James", male, .0,0,330877,8.4583, .Q 7,0,1, "McCarthy, Mr. Timothy J", male, 54,0,0,17463,51.86 8,0,3, "Palsson, Master. Gosta Leonard", male, 2,3,1,34990 9,1,3, "Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg 10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)", female, 14,

Note: Refer to the visible test case for better reference.

```
Explorer
         import pandas as pd
   2
         import numpy as np
   3
   4
         # Load the Titanic dataset
         data = pd.read_csv('Titanic-
   5
         Dataset.csv')
   6
         # 1. Display the first 5 rows of the
   7
         dataset
   8
   9
         print(data.head())
  10
         # 2. Display the last 5 rows of the
  11
         dataset
  12
  13
         print(data.tail())
  14
         # 3. Get the shape of the dataset
  15
  16
  17
         print(data.shape)
  18
  19
         # 4. Get a summary of the dataset
         (info)
  20
         print(data.info())
  21
  22
         # 5. Get basic statistics of the
  23
         dataset
  24
  25
         print(data.describe())
  26
         # 6. Check for missing values
  27
  28
  29
         print(data.isnull().sum())
  30
  31
         # 7. Fill missing values in the
         'Age' column with the median age
         median_age = data['Age'].median()
  32
  33
         data['Age'].fillna(median_age,
         inplace=True)
  34
  35
         # 8. Fill missing values in the
         'Embarked' column with the mode
  36
         mode_embarked =
         data['Embarked'].mode()[0]
         data['Embarked'].fillna(mode embarked
  37
         , inplace=True)
  38
         # 9. Drop the 'Cabin' column due to
  39
         many missing values
  40
         data.drop('Cabin', axis=1,
         inplace=True)
  41
  42
         # 10. Create a new column
         'FamilySize' by adding 'SibSp' and
         'Parch'
         data['FamilySize'] = data['SibSp'] +-
  43
         data['Parch']
  44
```

A ===



You are provided with the Titanic dataset containing Information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset.

4.2.6. Titanic Dataset Analysis and Data Cle... (1211) 🗚 🕻 🗷 🗷

- 1. Create a new column 'IsAlone' which is 1 if the passenger is alone (FamilySize = 0), otherwise 0.
- Convert the 'Sex' column to numeric values (male: 0, female: 1).
- 3. One-hot encode the 'Embarked' column, dropping the first category.
- Get the mean age of passengers.
- Get the median fare of passengers.
- Get the number of passengers by class.
- Get the number of passengers by gender.
- Get the number of passengers by survival status.
- Calculate the survival rate of passengers.
- Calculate the survival rate by gender.

The Titanic dataset contains columns as shown below,

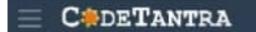
Passenger I d	Survived	P c l a s	N a m e	83 e x	A g e	5   b S p	P a r c h	T i c k e t	F a r e	C a b i n	E m b a r k e d
---------------	----------	-----------	------------------	--------	-------	-----------	-----------	-------------	------------------	-----------	-----------------------

### Sample Data:

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ti 1,0,3, "Braund, Mr. Owen Harris", male, 22,1,0,A/5 21171,7 2,1,1, "Cumings, Mrs. John Bradley (Florence Briggs Thay 3,1,3,"Heikkinen, Miss. Laina", female, 26,0,0,5TON/02. 3 4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)", fee 5,0,3, "Allen, Mr. William Henry", male, 35,0,0,373450,8.0 6,0,3,"Moran, Mr. James", male,,0,0,330877,8.4583,,Q 7,0,1, "McCarthy, Mr. Timothy J", male, 54,0,0,17463,51.86 8,0,3,"Palsson, Master. Gosta Leonard", male,2,3,1,34990 9,1,3, "Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg 10,1,2, "Nasser, Mrs. Nicholas (Adele Achem)", female, 14,

Note: Refer to the visible test case for better reference.

```
Explorer
   titanicData...
         import pandas as pd
   2
         import numpy as np
   3
   4
         # Load the Titanic dataset
         data = pd.read_csv('Titanic-
   5
         Dataset.csv')
   6
         data['FamilySize'] = data['SibSp'] +
         data['Parch']
   8
         data['Alone'] =
         np.where(data['FamilySize'] == 0, 1,
   9
  10
         # 3. Convert 'Sex' to numeric (male:
         0, female: 1)
         data['Sex'] =
  11
         data['Sex'].map({'male': 0,
         'female': 1})
  12
         # 4. One-hot encode the 'Embarked'
  13
         column, dropping the first category
         data = pd.get_dummies(data, columns=
  14
         ['Embarked'], drop_first=True)
  15
  16
         # 6. Get the mean age of passengers
  17
         mean_age = data['Age'].mean()
         print(mean age)
  18
  19
         # 7. Get the median fare of
  20
         passengers
         median fare = data['Fare'].median()
  21
         print(median_fare)
  22
  23
         # 8. Get the number of passengers by
  24
         class
         passengers_by_class =
  25
         data['Pclass'].value_counts()
         print(passengers_by_class)
  26
  27
         # 9. Get the number of passengers by
  28
         gender
  29
         passengers_by_gender =
         data['Sex'].value_counts().sort_index
         print(passengers_by_gender)
  30
  31
         # 10. Get the number of passengers
  32
         by survival status
         passengers_by_survival =
  33
         data['Survived'].value_counts().sort_
         index()
  34
         print(passengers_by_survival)
  35
  36
         # 11. Calculate the survival rate
  37
         survival rate =
         data['Survived'].mean()
         print(survival_rate)
  38
  39
  40
         # 12. Calculate the survival rate by
         gender
         survival_rate_by_gender =
  41
         data.groupby('Sex')
         ['Survived'].mean()
         print(survival_rate_by_gender)
  42
  43
  44
  45
  46
  47
  48
    X.
       =
```



4.2.7. Titanic Dataset Analysis and Data Cle... (IIIII) 🗚 📞 🗹 🖉



You are provided with the Titanic dataset containing Information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset.

- Calculate the survival rate by class.
- Calculate the survival rate by embarkation location. (Embarked\_S).
- Calculate the survival rate by family size (FamilySize).
- Calculate the survival rate by being alone (IsAlone).
- Get the average fare by passenger class (Pclass).
- Get the average age by passenger class (Pclass).
- Get the average age by survival status (Survived).
- Get the average fare by survival status (Survived).
- Get the number of survivors by class (Pclass).
- Get the number of non-survivors by class (Pclass).

The Titanic dataset contains columns as shown below,

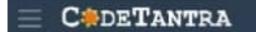
Passenger I d	ass	N a m e	S e x	A g e	SIBSP	P a r c h	T i c k e t	F a r e	C a b i n	E m b a r k e d
---------------	-----	------------------	-------	-------	-------	-----------------------	-------------	------------------	-----------	-----------------------------------

### Sample Data:

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ti 1,0,3,"Braund, Mr. Owen Harris", male, 22,1,0,A/5 21171,7 2,1,1, "Cumings, Mrs. John Bradley (Florence Briggs Thay 3,1,3,"Heikkinen, Miss. Laina", female, 26,0,0,5TON/02. 3 4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)", fee 5,0,3,"Allen, Mr. William Henry", male, 35,0,0,373450,8.0 6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q 7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.86 8,0,3, "Palsson, Master. Gosta Leonard", male,2,3,1,34990 9,1,3, "Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg 10,1,2, "Nasser, Mrs. Nicholas (Adele Achem)", female, 14,

Note: Refer to the visible test case for better reference.

```
Explorer
   titanicData...
         import pandas as pd
   2
         import numpy as np
   3
   4
         # Load the Titanic dataset
         data = pd.read_csv('Titanic-
   5
         Dataset.csv')
   6
         data['FamilySize'] = data['SibSp'] +
         data['Parch']
         data['IsAlone'] =
   7
         np.where(data['FamilySize'] > 0, 0,
         data = pd.get_dummies(data, columns=
   8
         ['Embarked'], drop_first=True)
   9
         print(data.groupby('Pclass')
  10
         ['Survived'].mean())
  11
         #2. Calculate the survival rate by
  12
         embarked location (Embarked_S)
         print(data.groupby('Embarked_S')
  13
         ['Survived'].mean())
  14
         #3. Calculate the survival rate by
  15
         family size
         print(data.groupby('FamilySize')
  16
         ['Survived'].mean())
  17
         #4. Calculate the survival rate by
  18
         being alone
         print(data.groupby('IsAlone')
  19
         ['Survived'].mean())
  20
  21
         #5. Get the average fare by class
         print(data.groupby('Pclass')
  22
         ['Fare'].mean())
  23
  24
         #6. Get the average age by class
  25
         print(data.groupby('Pclass')
         ['Age'].mean())
  26
  27
         #7. Get the average age by survival
         print(data.groupby('Survived')
  28
         ['Age'].mean())
  29
  30
         #8. Get the average fare by survival
         print(data.groupby('Survived')
  31
         ['Fare'].mean())
  32
  33
         #9. Get the number of survivors by
         class (sort by values descending)
         print(data[data['Survived'] == 1]
  34
         ['Pclass'].value_counts())
  35
         #10. Get the number of non-survivors
  36
         by class (sort by values descending)
         print(data[data['Survived'] == 0]
  37
         ['Pclass'].value_counts())
  38
  39
  40
    A ===
```



You are provided with the Titanic dataset containing Information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset.

4.2.8. Titanic Dataset Analysis and Data Cle... (EDD) 🗚 📞 🗷 🔗

Explorer

2

3

4

5

6

7

8

9

10

titanicData...

import pandas as pd

# Load the Titanic dataset

survivors\_by\_gender =

['Sex'].value\_counts()

data[data['Survived'] == 1]

print(survivors by gender)

data = pd.read csv('Titanic-

['Embarked'], drop\_first=True)

data = pd.get\_dummies(data, columns=

0

import numpy as np

Dataset.csv')

- Get the number of survivors by gender (Sex).
- Get the number of non-survivors by gender (Sex).
- 3. Get the number of survivors by embarkation location (Embarked\_S).
- 4. Get the number of non-survivors by embarkation location (Embarked\_S).
- Calculate the percentage of children (Age < 18) who</li>
- Calculate the percentage of adults (Age >= 18) who survived.
- 7. Get the median age of survivors.
- 8. Get the median age of non-survivors.
- 9. Get the median fare of survivors.
- Get the median fare of non-survivors.

The Titanic dataset contains columns as shown below,

Passur Name Name x	S P a r c p h	T C a b i n	E m b a r k e d
--------------------	---------------	-------------	--------------------------------

# Sample Data:

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ti 1,0,3, "Braund, Mr. Owen Harris", male, 22,1,0,A/5 21171,7 2,1,1, "Cumings, Mrs. John Bradley (Florence Briggs Thay 3,1,3,"Heikkinen, Miss. Laina", female, 26,0,0,5TON/02. 3 4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",fe 5,0,3, "Allen, Mr. William Henry", male, 35,0,0,373450,8.0 6,0,3,"Moran, Mr. James", male, ,0,0,330877,8.4583, ,Q 7,0,1, "McCarthy, Mr. Timothy J", male, 54,0,0,17463,51.86 8,0,3, "Palsson, Master, Gosta Leonard", male,2,3,1,34990 9,1.3. "Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg 10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)", female, 14,

Note: Refer to the visible test case for better reference.

Sample Test Cases

11 # 2. Get the number of non-survivors-12 by-gender non\_survivors\_by\_gender = 13 data[data['Survived'] == 0] ['Sex'].value\_counts() 14 print(non\_survivors\_by\_gender) 15 # 3. Get the number of survivors by 16 embarked location (Embarked\_S) survivors\_by\_embarked\_s = 17 data[data['Survived'] == 1] ['Embarked\_S'].value\_counts() print(survivors\_by\_embarked\_s) 18 19 # 4. Get the number of non-survivors 20 by embarked location (Embarked\_S) non survivors by embarked s = 21 data[data['Survived'] == 0] ['Embarked\_S'].value\_counts() print(non\_survivors\_by\_embarked\_s) 22 23 24 #5. Percentage of children (Age < 18) who survived children=data [data['Age'] < 18] 25 children\_survival\_rate= 26 children['Survived'].mean() print(children\_survival\_rate) 27 28 29 #6. Percentage of adults (Age->--18) who survived adults= data[data['Age'] >=18] 30 adults\_survival\_rate= 31 adults['Survived'].mean() print(adults survival rate) 32 33 #7. Median age of survivors 34 median\_age\_survivors = 35 data[data['Survived'] == 1] ['Age'].median() print(median\_age\_survivors) 36 37 38 #8. Median age of non-survivors 39 median\_age\_non\_survivors = data[data['Survived'] == 0] ['Age'].median() print(median\_age\_non\_survivors) 40 41 #9. Median fare of survivors 42 median\_fare\_survivors = 43 data[data['Survived'] == 1] ['Fare'].median() print(median\_fare\_survivors) 44 45 # 10. Median fare of non-survivors 46 47 median\_fare\_non\_survivors = data[data['Survived'] == 0] ['Fare'].median() print(median\_fare\_non\_survivors) 48 Att

**X**.

■