

Sample Test Cases

15 16

Activate Windows

Go to Settings to activate Wind

< Prev Reset

Submit

Next >

Debugger





C∯DETANTRA # Home

· 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.

Sample Test Cases

Explorer Debugger datafram... Submit 1 import pandas as pd 2 # Provided dictionary of lists 3 v data = { 'Name': ['Alice', 'Bob', 'Charlie'], 5 6 'Age': [25, 30, 35], 8 # Convert the dictionary to a DataFrame 9 df = pd.DataFrame(data) 10 11 # Display the original DataFrame 12 print("Original DataFrame:") 13 14 print(df) 15 # Adding a new row 16 new name=input("New name: ") 17 new age=int(input("New age: ")) 18 new row={'Name': new name,'Age':new age} 19 df=pd.concat([df,pd.DataFrame([new row])],ignore index=True) 20 # Display the DataFrame after adding a new row 21 Activate Windows > Terminal ⊞ Test cases

< Prev

Reset

Submit

it N





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:

4.1.2. Dictionary to 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.

Sample Test Cases

+

22 23 datafram... Submit Debugger print("After adding a row:\n",df) 23 24 # Modifying a row modify index=int(input("Index of row to modify: ")) 25 new age mod=int(input("New age: ")) 26 df.loc[modify index, "Age"]=new age mod 27 # Display the DataFrame after modifying a row 28 print("After modifying a row:") 29 print(df) 30 31 32 # Deleting a row delete index=int(input("Index of row to delete: ")) 33 df=df.drop(delete index).reset index(drop=True) 34 35 # Display the DataFrame after deleting a row print("After deleting a row:") 36 print(df) 37 38 39 # Adding a new column 40 gender input=input("Enter genders separated by space: ") 41 genders=gender input.split() df["Gender"]=genders 42 #\_Dicolou the Determonent of the Activate Windows >\_ Terminal ⊞ Test cases

< Prev

Reset



### 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:

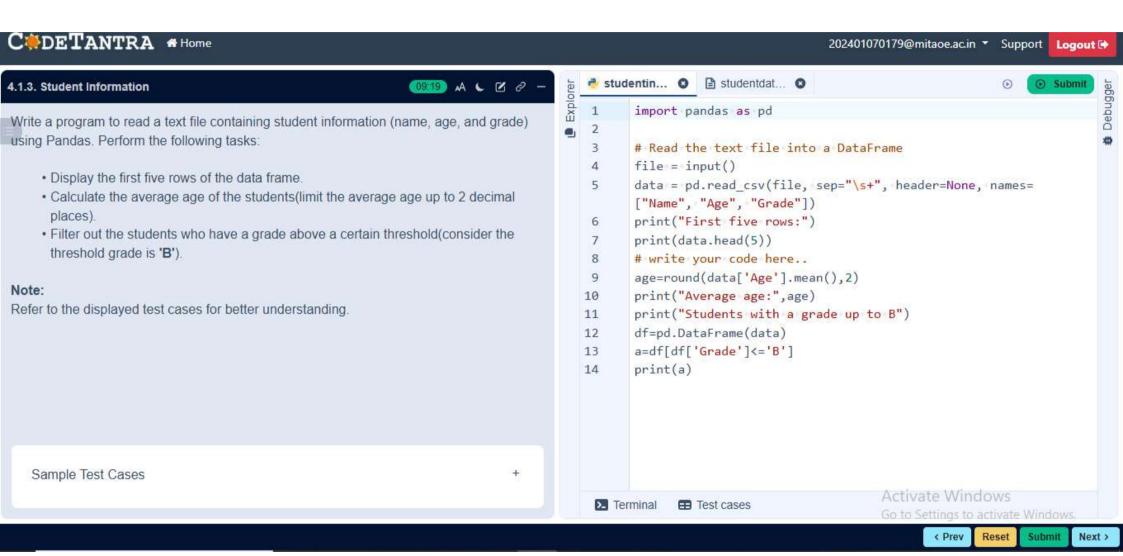
- · Modify a specific row by changing the age. Take the row index and new age value from the user.
- · Display the DataFrame after modifying the row.

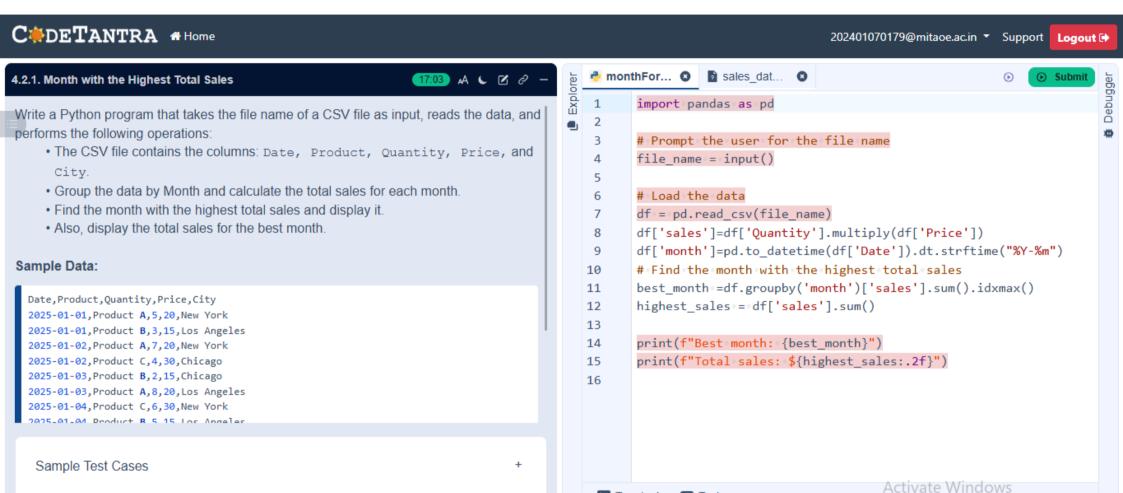
Sample Test Cases



< Prev

Reset



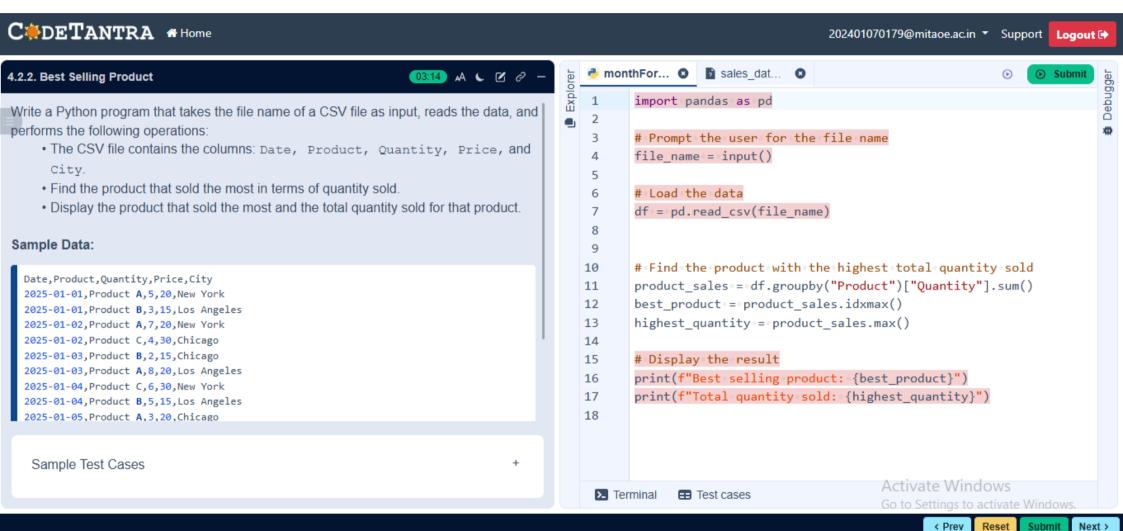


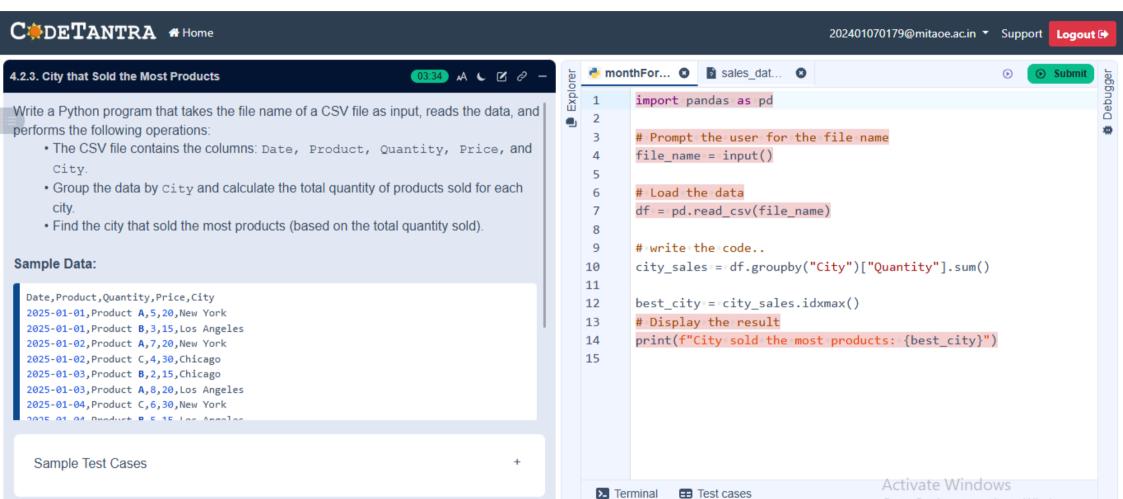
> Terminal

⊞ Test cases

< Prev

Reset





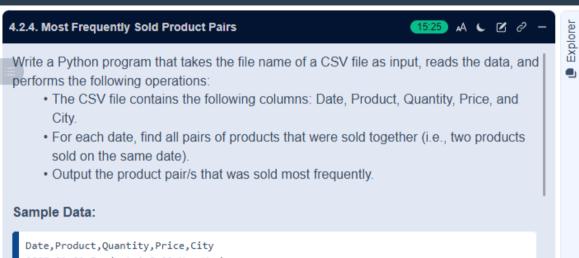
Reset

Submit

< Prev

Next >

Submit



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 Sample Test Cases

Debugger 1 import pandas as pd from itertools import combinations 2 from collections import Counter 3 4 # Prompt user to input the file name 5 6 file name = input() 7 8 # Read data from the specified CSV file 9 df = pd.read csv(file name) 10 # write the code 11 12 date products == {} 13 for date, group in df.groupby('Date'): 14 >|products==group['Product'].unique() 15 v → if len(products) > 1: 16 17 → date products[date] = products 18 19 pair counter = Counter() 20 , for products in date products.values(): Activate Windows >\_ Terminal Test cases

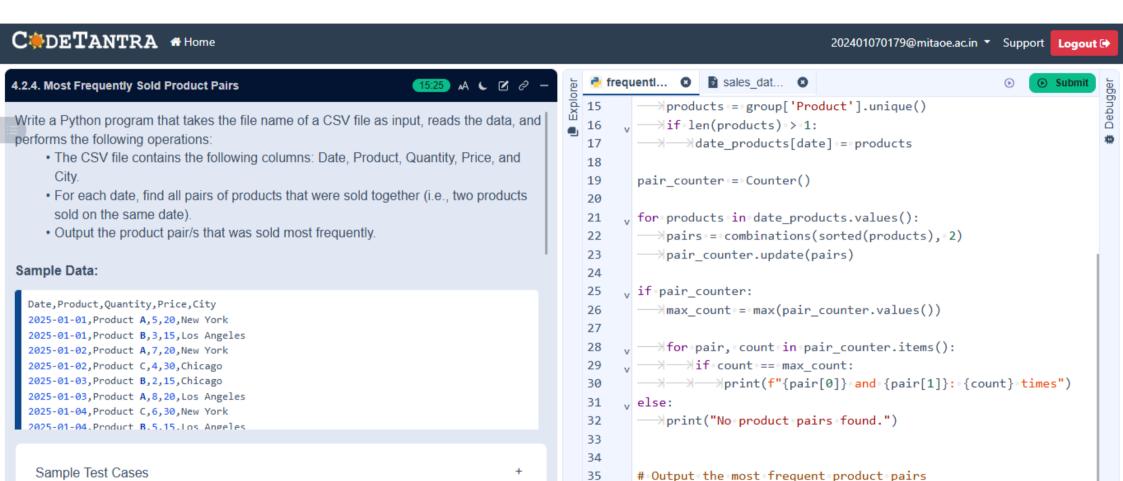
🔁 frequenti... 🔞 🛮 a sales dat... 🔞

< Prev

Reset

Submit

Next >



> Terminal

< Prev Reset

Activate Windows

Submit

Next >



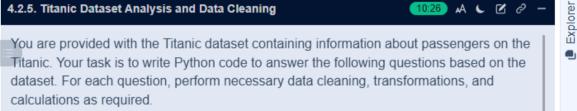


Submit

Debugger



4.2.5. Titanic Dataset Analysis and Data Cleaning



- 1. Display the first 5 rows of the dataset.
- 2. Display the last 5 rows of the dataset.
- 3. Get the shape of the dataset (number of rows and columns).
- 4. 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).
- 9. Drop the 'Cabin' column due to many missing values.
- 10. Create a new column, 'FamilySize' by adding the 'SibSp' and 'Parch' columns.

The Titanic dataset contains columns as shown below.

Sample Test Cases

import numpy as np 2 3 4 # Load the Titanic dataset data = pd.read csv('Titanic-Dataset.csv') 5 6 7 # 1. Display the first 5 rows of the dataset print(data.head()) 8 9 10 # 2. Display the last 5 rows of the dataset 11 print(data.tail()) 12 13 # 3. Get the shape of the dataset 14 print(data.shape) 15 16 # 4. Get a summary of the dataset (info) 17 print(data.info()) 18 19 # 5. Get basic statistics of the dataset

print(data.describe())

⊞ Test cases

import pandas as pd

titanicDat...

1

20

21

> Terminal

10:26 A C Z 2 -

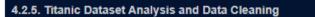
< Prev

Activate Windows

Reset

Debugger





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.

- 1. Display the first 5 rows of the dataset.
- 2. Display the last 5 rows of the dataset.
- 3. Get the shape of the dataset (number of rows and columns).
- 4. 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).
- 9. Drop the 'Cabin' column due to many missing values.
- 10. Create a new column, 'FamilySize' by adding the 'SibSp' and 'Parch' columns.

The Titanic dataset contains columns as shown below.

Sample Test Cases

10:26 A C Z 2 -

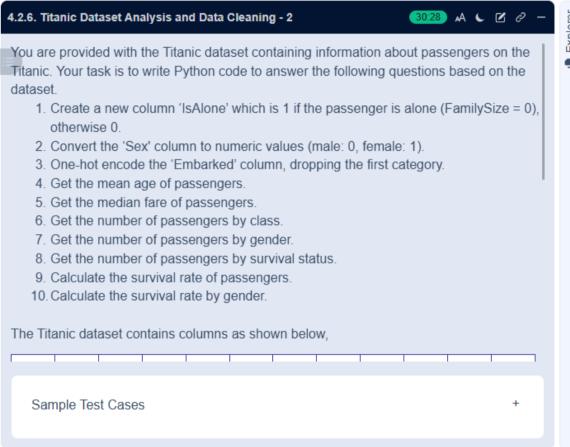
```
Explorer 18
   titanicDat...
                                                                   Submit
  19
          # 5. Get basic statistics of the dataset
   20
          print(data.describe())
  21
   22
          # 6. Check for missing values
          print(data.isnull().sum())
  23
   24
  25
          # 7. Fill missing values in the 'Age' column with the median
          age
          median age = data['Age'].median()
  26
  27
          data['Age'].fillna(median age, inplace=True)
          # 8. Fill missing values in the 'Embarked' column with the mode
   28
          mode embarked = data['Embarked'].mode()[0]
   29
  30
          data['Embarked'].fillna(mode embarked, inplace=True)
          # 9. Drop the 'Cabin' column due to many missing values
  31
  32
          data.drop('Cabin', axis=1,inplace=True)
  33
   34
          # 10. Create a new column 'FamilySize' by adding 'SibSp' and
          'Parch'
          data['FamilySize'] = data['SibSp'] + data['Parch']
  35
   36
                                             Activate Windows
    > Terminal
                Test cases
```







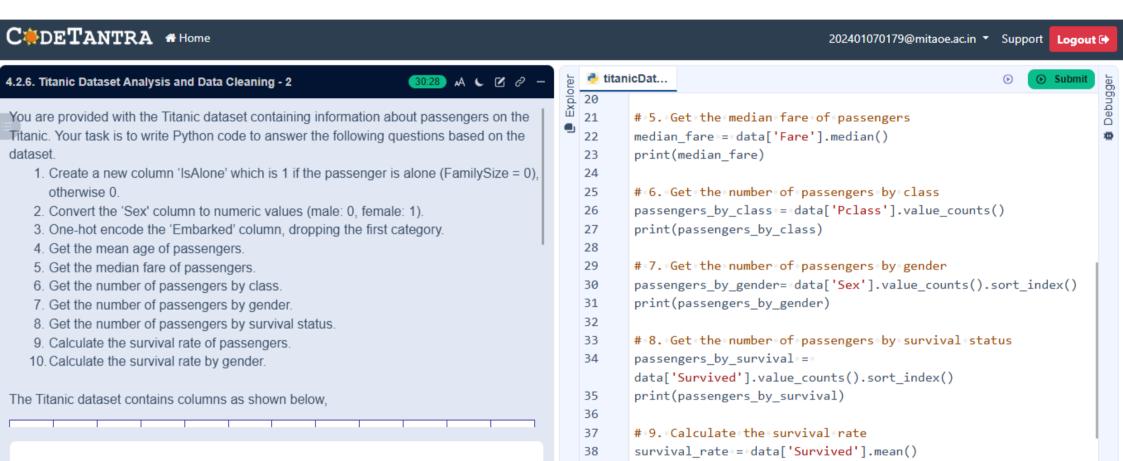




```
titanicDat...
                                                                    Submit
                                                                             Debugger
Explorer
         import pandas as pd
   1
   2
         import numpy as np
   3
   4
         # Load the Titanic dataset
   5
         data = pd.read csv('Titanic-Dataset.csv')
         data['FamilySize'] = data['SibSp'] + data['Parch']
   6
   7
   8
         # 1. Create a new column 'IsAlone' (1 if alone, 0 otherwise)
         data['IsAlone'] = np.where(data['FamilySize'] == 0, 1, 0)
   9
  10
         # 2. Convert 'Sex' to numeric (male: 0, female: 1)
  11
         data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
  12
  13
  14
         # 3. One-hot encode the 'Embarked' column
         data = pd.get dummies(data, columns=['Embarked'],
  15
         drop first=True)
  16
         # 4. Get the mean age of passengers
  17
         mean age = data['Age'].mean()
  18
         print(mean age)
  19
  20
                                             Activate Windows
    > Terminal

    ⊞ Test cases
```

< Prev Reset



39

> Terminal

Sample Test Cases

print(survival rate)

Test cases

< Prev Reset

Activate Windows

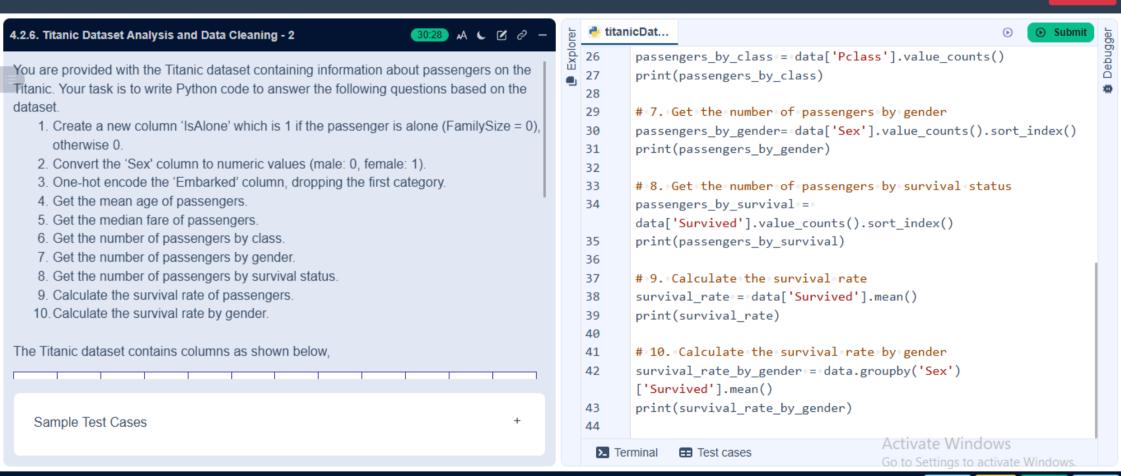
Submit

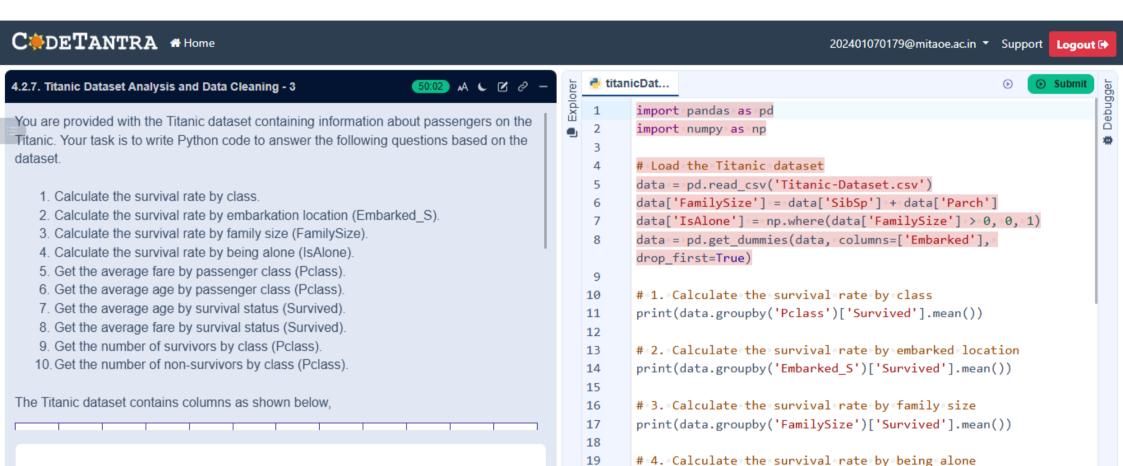
Next

< Prev

Reset







20

>\_ Terminal

⊞ Test cases

print(data.groupby('IsAlone')['Survived'].mean())

Activate Windows

< Prev

Reset

Submit

Sample Test Cases

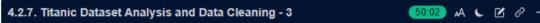
< Prev

Reset

Submit

Debugger





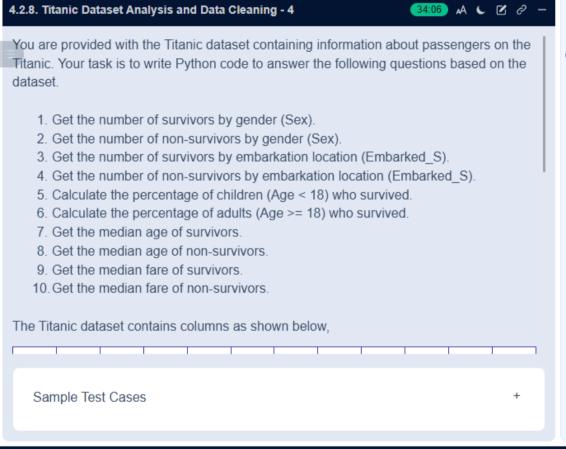
You are provided with the Titanic dataset containing information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset.

- 1. Calculate the survival rate by class.
- 2. Calculate the survival rate by embarkation location (Embarked S).
- 3. Calculate the survival rate by family size (FamilySize).
- 4. Calculate the survival rate by being alone (IsAlone).
- 5. Get the average fare by passenger class (Pclass).
- 6. Get the average age by passenger class (Pclass).
- 7. Get the average age by survival status (Survived).
- 8. Get the average fare by survival status (Survived).
- 9. Get the number of survivors by class (Pclass).
- 10. Get the number of non-survivors by class (Pclass).

The Titanic dataset contains columns as shown below,

Sample Test Cases

Explorer 18 titanicDat... Submit 19 # 4. Calculate the survival rate by being alone print(data.groupby('IsAlone')['Survived'].mean()) 20 21 22 # 5. Get the average fare by class print(data.groupby('Pclass')['Fare'].mean()) 23 24 25 # 6. Get the average age by class print(data.groupby('Pclass')['Age'].mean()) 26 27 # 7. Get the average age by survival status 28 29 print(data.groupby('Survived')['Age'].mean()) 30 31 # 8. Get the average fare by survival status print(data.groupby('Survived')['Fare'].mean()) 32 33 34 # 9. Get the number of survivors by class print(data[data['Survived'] == 1]['Pclass'].value counts()) 35 36 # 10. Get the number of non-survivors by class 37 print(data[data['Survived'] == 0]['Pclass'].value\_counts()) 38 Activate Windows >\_ Terminal 



```
Explorer
   ditanicDat...
                                                                    Submit
                                                                             Debugger
   1
          import pandas as pd
   2
          import numpy as np
   3
   4
          # Load the Titanic dataset
          data = pd.read csv('Titanic-Dataset.csv')
   5
         data = pd.get dummies(data, columns=['Embarked'],
   6
          drop first=True)
   7
   8
         survivors by gender = data[data['Survived'] == 1]
   9
          ['Sex'].value counts()
  10
         print(survivors by gender)
  11
          # 2. Get the number of non-survivors by gender
  12
         non_survivors_by_gender = data[data['Survived'] == 0]
  13
          ['Sex'].value counts()
         print(non survivors by gender)
  14
  15
  16
          # 3. Get the number of survivors by embarked location
          survivors by embarked s==data[data['Survived']===1]
  17
          ['Embarked S'].value counts()
                                             Activate Windows
    >_ Terminal

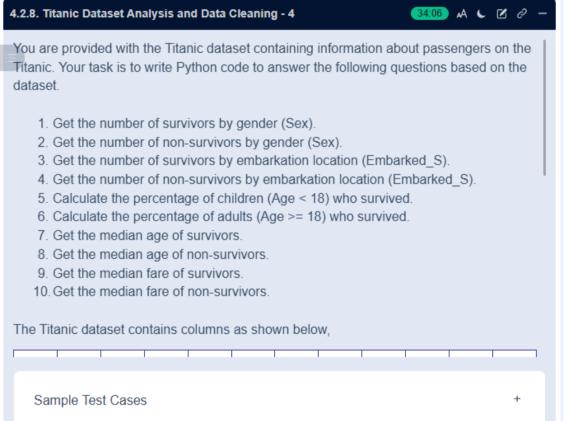
    ⊞ Test cases

                                                            Reset
                                                     < Prev
                                                                   Submit
```

Reset

Submit

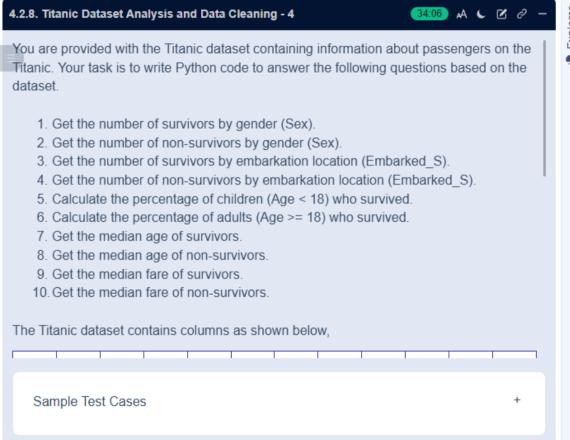
< Prev

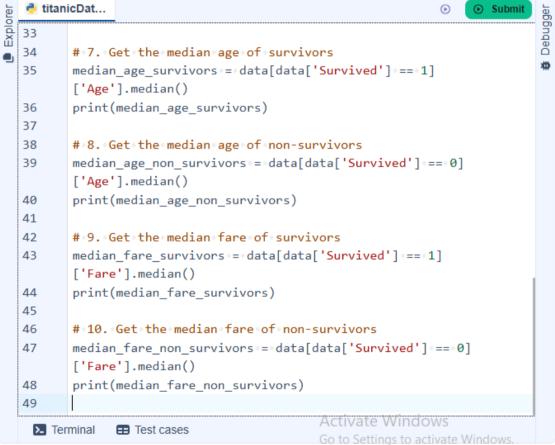


```
Explorer
   ditanicDat...
                                                                             Debugger
                                                                    Submit
          ['Embarked S'].value counts()
  18
          print(survivors by embarked s)
  19
          # 4. Get the number of non-survivors by embarked location
  20
         non survivors by embarked s= data[data['Survived'] == 0]
  21
          ['Embarked S'].value counts()
         print(non survivors by embarked s)
  22
  23
  24
          # 5. Calculate the percentage of children (Age < 18) who
          survived
         children = data[data['Age'] < 18]
  25
          children survival rate = children['Survived'].mean()
  26
         print(children survival rate)
  27
  28
          # 6. Calculate the percentage of adults (Age >= 18) who
  29
          survived
          adults = data[data['Age'] >= 18]
  30
          adults_survival_rate = adults['Survived'].mean()
  31
         print(adults survival rate)
  32
  33
          # 7. Get the median age of survivors
  34
                                             Activate Windows
    >_ Terminal

    ⊞ Test cases
```







< Prev

Reset