> Terminal

Test cases

< Prev Reset Submit Next >

4880 A C B 8 -4.1.2. Dictionary to dataframe

adatafram...

10

11 12

13

14

15 16

17

18

19

20

21

22 23 24

25

Terminal

data = {

print(df)

import pandas as pd

Provided dictionary of lists

'Age': [25, 30, 35],

Display the original DataFrame

print("Original DataFrame:")

new name=input("New name: ")

new age=int(input("New age: "))

print("After adding a row:\n",df)

new row={'Name': new name, 'Age':new age}

Display the DataFrame after adding a new row

modify index=int(input("Index of row to modify: "))

df=pd.concat([df,pd.DataFrame([new row])],ignore index=True)

df = pd.DataFrame(data)

Adding a new row

Modifying a row

Test cases

'Name': ['Alice', 'Bob', 'Charlie'],

Convert the dictionary to a 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.
- Add a new column:

- Sample Test Cases

- · Remove the specified row. · Display the DataFrame after deleting the row.





















< Prev Reset Submit

Next >



> Terminal



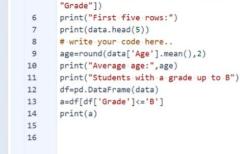
Write a program to read a text file containing student information (name, age, and grade) using Pandas. Perform the following tasks:

- · Display the first five rows of the data frame.
- · Calculate the average age of the students(limit the average age up to 2 decimal places).
- · Filter out the students who have a grade above a certain threshold(consider the threshold grade is 'B').

Note:

Sample Test Cases

Refer to the displayed test cases for better understanding.



Explor

studentin... O studentdat... O

file = input()

import pandas as pd

Read the text file into a DataFrame

data = pd.read csv(file, sep="\s+", header=None, names=["Name", "Age",

4.2.1. Month with the Highest Total Sales MER A L M 2 -

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 each month.
- · Find the month with the highest total sales and display it.
- . Also, display the total sales for the best month.

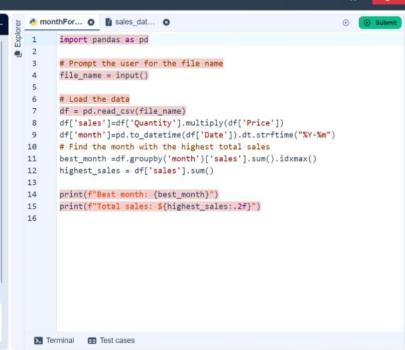
Sample Data:

```
Date Product Quantity Price City
2025-01-01.Product A.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
```

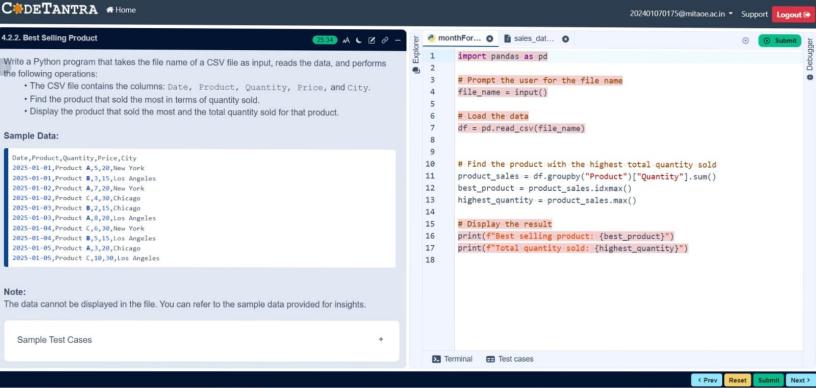
2025-01-04, Product 8,5,15, Los Angeles 2025-01-05, Product A, 3, 20, Chicago 2025-01-05, Product C, 10, 30, Los Angeles

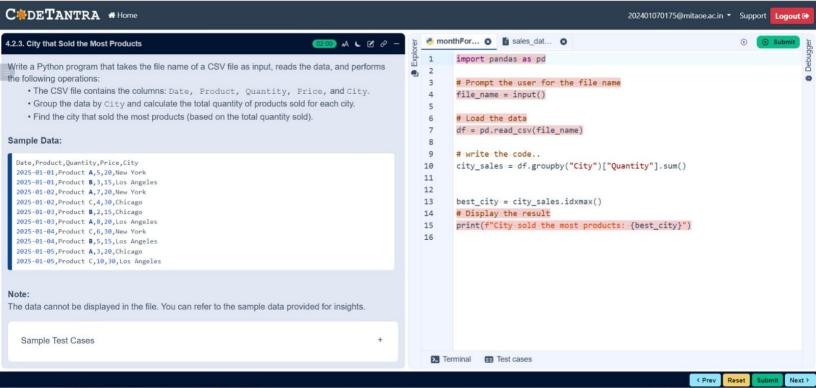
Note:

Sample Test Cases



< Prev





< Prev

*

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:

```
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
```

Date, Product, Quantity, Price, City

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:

Sample Test Cases

23 24 25

> Terminal

10

11

12

13

14

15

16

17

18

19 20

21

22

frequentl... O sales_dat... O

import pandas as pd

file name = input()

write the code

date products = {}

from itertools import combinations

Prompt user to input the file name

Read data from the specified CSV file

v for date, group in df.groupby('Date'):

date products[date] = products

for products in date products.values():

pair counter.update(pairs)

pairs = combinations(sorted(products), 2)

products = group['Product'].unique()

from collections import Counter

df = pd.read csv(file name)

if len(products) > 1:

pair counter = Counter()

Test cases



4.2.4. Most Frequently Sold Product Pairs

CODETANTRA # Home



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 B, 5, 15, Los Angeles

2025-01-04, Product C, 6, 30, New York

2025-01-05, Product A, 3, 20, Chicago 2025-01-05, Product C, 10, 30, Los Angeles

Explanation:

Transactions:

33 Sample Test Cases

```
frequentl... O sales dat... O
Explorer
  10
  11
        # write the code
  13
        date products = {}
  14
  15
       v for date, group in df.groupby('Date'):
         products = group['Product'].unique()
  16
       if len(products) > 1:
  17
  18
         date products[date] = products
  19
  20
        pair counter = Counter()
  21
  22
       for products in date products.values():
         pairs = combinations(sorted(products), 2)
  24
  25
  26
       , if pair counter:
  27
         max count = max(pair counter.values())
  28
       v for pair, count in pair_counter.items():
       if count == max count:
  30
        print(f"{pair[0]} and {pair[1]}: {count} times")
  31
```

32

Output the most frequent product pairs > Terminal

, else:

pair counter.update(pairs)

print("No product pairs found.")

⊞ Test cases

10

11

12

13

14

15

20 21

22 23 24

25

4.2.5. Titanic Dataset Analysis and Data Cleaning



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.
- 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().
- 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, Pas

n erl	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed	
----------	------------------	------------	----------	-----	-----	-----------	-----------	------------	----------	-----------	------------------	--

Sample Test Cases

titanicDat...

import pandas as pd

import numpy as np

Load the Titanic dataset data = pd.read_csv('Titanic-Dataset.csv')

print(data.head())

2. Display the last 5 rows of the dataset print(data.tail())

3. Get the shape of the dataset

print(data.shape)

16 print(data.info()) 17 # 5. Get basic statistics of the dataset 18 19 print(data.describe())

4. Get a summary of the dataset (info)

6. Check for missing values print(data.isnull().sum())

7. Fill missing values in the 'Age' column with the median age median age = data['Age'].median()

Test cases > Terminal

4.2.5. Titanic Dataset Analysis and Data Cleaning



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

- 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()).

question, perform necessary data cleaning, transformations, and calculations as required.

- 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.

Sur Em Pdla Na Sib Par Tick Far Cab Sex Age bark vive Sp ch SS me et e in ed

17 18 # 5. Get basic statistics of the dataset 19 print(data.describe()) 20 21

titanicDat...

15

16

23

24

25

26

27

28

29

30

31 32

37 38

39

6. Check for missing values print(data.isnull().sum())

print(data.info())

7. Fill missing values in the 'Age' column with the median age

median_age = data['Age'].median() data['Age'].fillna(median age, inplace=True)

8. Fill missing values in the 'Embarked' column with the mode mode embarked = data['Embarked'].mode()[0]

4. Get a summary of the dataset (info)

data['Embarked'].fillna(mode_embarked, inplace=True) # 9. Drop the 'Cabin' column due to many missing values

33 data.drop('Cabin', axis=1, inplace=True) 34 35 # 10. Create a new column 'FamilySize' by adding 'SibSp' and 'Parch' 36 data['FamilySize'] = data['SibSp'] + data['Parch']

> Terminal Test cases

ger

Sample Test Cases

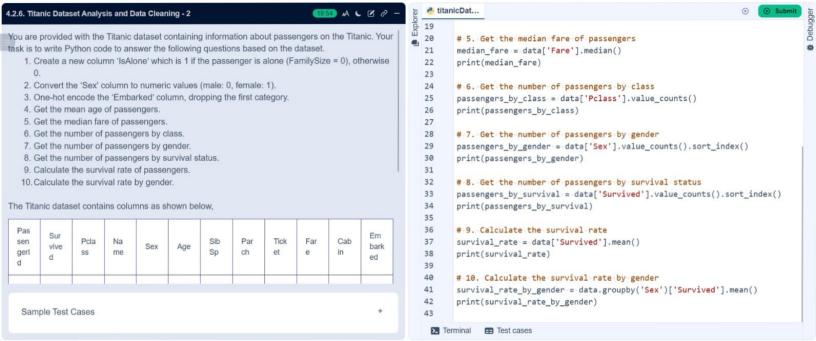
Pas sen

Logout [

< Prev

Pas Sur Em Pcla Na Sib Tick Far Cab Par Sex Age vive bark me Sp ch et e in ed Sample Test Cases

atitanicDat... import pandas as pd import numpy as np # Load the Titanic dataset data = pd.read csv('Titanic-Dataset.csv') data['FamilySize'] = data['SibSp'] + data['Parch'] data['IsAlone'] = np.where(data['FamilySize'] -- 0, 1, 0) 10 # 2. Convert 'Sex' to numeric (male: 0, female: 1) 11 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1}) 12 13 # 3. One-hot encode the 'Embarked' column 14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True) 15 # 4. Get the mean age of passengers 16 17 mean age = data['Age'].mean() 18 print(mean age) 19 20 # 5. Get the median fare of passengers 21 median fare = data['Fare'].median() 22 print(median fare) 23 24 # 6. Get the number of passengers by class passengers by class = data['Pclass'].value counts() > Terminal



C#DETANTRA # Home

4.2.7. Titanic Dataset Analysis and Data Cleaning - 3 0818 A L Z 2 -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).

9. Get the number of survivors by class (Pclass).

7. Get the average age by survival status (Survived).

8. Get the average fare by survival status (Survived). 10. Get the number of non-survivors by class (Pclass). The Titanic dataset contains columns as shown below. Pas Sur Em Pcla Sib Par Tick Far Cab sen Na vive Sex Age bark gerl me Sp ch in ed Sample Test Cases

titanicDat... 13 print(data.groupby('Embarked S')['Survived'].mean()) 14 15 # 3. Calculate the survival rate by family size print(data.groupby('FamilySize')['Survived'].mean()) 16 17 18 # 4. Calculate the survival rate by being alone print(data.groupby('IsAlone')['Survived'].mean()) 19 20 21 # 5. Get the average fare by class 22 print(data.groupby('Pclass')['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 status 28 print(data.groupby('Survived')['Age'].mean()) 29 30 # 8. Get the average fare by survival status 31 print(data.groupby('Survived')['Fare'].mean()) 32 33 # 9. Get the number of survivors by class 34 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()) > Terminal ☐ Test cases

4.2.8. Titanic Dataset Analysis and Data Cleaning - 4 ALBO-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. Get the number of survivors by gender (Sex). 2. 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). 5. Calculate the percentage of children (Age < 18) who survived. 6. 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. 10. Get the median fare of non-survivors.

Pas Sur Em Pcla Na Sib Par Tick Far Cab Sex Age vive bark in gerl me ch et e d ed

The Titanic dataset contains columns as shown below. Sample Test Cases

atitanicDat... import pandas as pd import numpy as np # Load the Titanic dataset data = pd.read csv('Titanic-Dataset.csv') data = pd.get dummies(data, columns=['Embarked'], drop first=True) 9 survivors by gender = data[data['Survived'] == 1]['Sex'].value counts() 10 print(survivors by gender) 11 12 # 2. Get the number of non-survivors by gender 13 non survivors by gender = data[data['Survived'] == 0] ['Sex'].value_counts() 14 print(non survivors by gender) 15 16 # 3. Get the number of survivors by embarked location survivors_by_embarked_s = data[data['Survived'] == 1] 17 ['Embarked_S'].value_counts() 18 print(survivors by embarked s) 19 20 # 4. Get the number of non-survivors by embarked location 21 non survivors by embarked s = data[data['Survived'] == 0] ['Embarked S'].value counts()

print(non_survivors_by_embarked_s)

Test cases

> Terminal

4.2.8. Titanic Dataset Analysis and Data Cleaning - 4



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. Get the number of survivors by gender (Sex).
- 2. 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 survived.
- 6. 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
- 10 Get the median fare of non-survivors

The Titanic dataset contains columns as shown below.

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed
-------------------------	------------------	------------	----------	-----	-----	-----------	-----------	------------	----------	-----------	------------------

Sample Test Cases

```
titanicDat...
```

24

5. Calculate the percentage of children (Age < 18) who survived

children = data[data['Age'] < 18]

adults = data[data['Age'] >= 18]

7. Get the median age of survivors

8. Get the median age of non-survivors

9. Get the median fare of survivors

10. Get the median fare of non-survivors

print(children survival rate)

print(adults survival rate)

print(median age survivors)

print(median age non survivors)

print(median fare survivors)

⊞ Test cases

print(median fare non survivors)

children survival rate = children['Survived'].mean()

adults survival rate = adults['Survived'].mean()

6. Calculate the percentage of adults (Age >= 18) who survived

median age survivors = data[data['Survived'] == 1]['Age'].median()

median age non survivors = data[data['Survived'] == 0]['Age'].median()

median fare survivors = data[data['Survived'] == 1]['Fare'].median()

median fare non survivors = data[data['Survived'] == 0]['Fare'].median()

26

27

28

29

30

31

32

33

34

35

36

37 38

39

40

41 42

43

44 45 46

47

48

> Terminal