

4.1.1. Pandas - series creation and manipulation

33:02

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.

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

+

seriesMa...

Submit

```
1 import pandas as pd
2
3 # Take inputs from the user to create a list of numbers
4 numbers = list(map(int, input().split()))
5
6 # Create a Pandas series from the list of numbers
7 data_list=numbers
8 series_from_list=pd.Series(data_list)
9 # Grouping by even and odd numbers and calculating the mean
10 grouped =series_from_list.groupby(series_from_list%2==
11 0).mean()
12
13 # Display the mean of even and odd numbers with labels
14 grouped.index = ['Even' if is_even else 'Odd' for is_even in
15 grouped.index]
16 print("Mean of even and odd numbers:")
17 print(grouped)
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

4.1.2. Dictionary to dataframe

15:18



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.

Sample Test Cases



datafram...

[Submit](#)

```
1 import pandas as pd
2
3 # Provided dictionary of lists
4 data = {
5     'Name': ['Alice', 'Bob', '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
17 new_name=input("New name: ")
18 new_age=int(input("New age: "))
19 new_row={'Name': new_name, 'Age': new_age}
20 df=pd.concat([df,pd.DataFrame([new_row]),ignore_index=True)
21 # Display the DataFrame after adding a new row
```

Terminal

Test cases

Activate Windows

Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

4.1.2. Dictionary to dataframe

15:18

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.

Sample Test Cases

+

datafram...

Submit

```
22 print("After adding a row:\n",df)
23
24 # Modifying a row
25 modify_index=int(input("Index of row to modify: "))
26 new_age_mod=int(input("New age: "))
27 df.loc[modify_index,"Age"]=new_age_mod
28 # Display the DataFrame after modifying a row
29 print("After modifying a row:")
30 print(df)
31
32 # Deleting a row
33 delete_index=int(input("Index of row to delete: "))
34 df=df.drop(delete_index).reset_index(drop=True)
35 # Display the DataFrame after deleting a row
36 print("After deleting a row:")
37 print(df)
38
39 # Adding a new column
40 gender_input=input("Enter genders separated by space: ")
41 genders=gender_input.split()
42 df["Gender"]=genders
43 # Display the DataFrame after adding a new column
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

4.1.2. Dictionary to dataframe

15:18



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.

Sample Test Cases



Explorer

datafram...



Submit

Debugger

```
38
39 # Adding a new column
40 gender_input=input("Enter genders separated by space: ")
41 genders=gender_input.split()
42 df["Gender"]=genders
43 # Display the DataFrame after adding a new column
44 print("After adding a new column:")
45 print(df)
46
47 # Modifying a column
48 df["Name"]=df["Name"].str.upper()
49 # Display the DataFrame after modifying a column
50 print("After modifying a column:")
51 print(df)
52
53 # Deleting a column
54 df=df.drop(columns=['Age'])
55 # Display the DataFrame after deleting a column
56 print("After deleting a column:")
57 print(df)
58
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

4.1.3. Student Information

09:19

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:

Refer to the displayed test cases for better understanding.

Sample Test Cases

+

Explorer

studentin... studentdat...

Submit

```
1 import pandas as pd
2
3 # Read the text file into a DataFrame
4 file = input()
5 data = pd.read_csv(file, sep="\s+", header=None, names=
6 ["Name", "Age", "Grade"])
7 print("First five rows:")
8 print(data.head(5))
9 # write your code here..
10 age=round(data['Age'].mean(),2)
11 print("Average age:",age)
12 print("Students with a grade up to B")
13 df=pd.DataFrame(data)
14 a=df[df['Grade']<='B']
15 print(a)
```

Debugger

Terminal Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev Reset Submit Next >

4.2.1. Month with the Highest Total Sales

17:03

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 B	5	15	Los Angeles

Sample Test Cases

monthFor... sales_dat...

Submit

```
1 import pandas as pd
2
3 # Prompt the user for the file name
4 file_name = input()
5
6 # Load the data
7 df = pd.read_csv(file_name)
8 df['sales'] = df['Quantity'].multiply(df['Price'])
9 df['month'] = pd.to_datetime(df['Date']).dt.strftime("%Y-%m")
10 # Find the month with the highest total sales
11 best_month = df.groupby('month')['sales'].sum().idxmax()
12 highest_sales = df['sales'].sum()
13
14 print(f"Best month: {best_month}")
15 print(f"Total sales: ${highest_sales:.2f}")
16
```

Terminal Test cases

Activate Windows Go to Settings to activate Windows.

4.2.2. Best Selling Product

03:14

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.
- Find the product that sold the most in terms of quantity sold.
- Display the product that sold the most and the total quantity sold for that product.

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
```

Sample Test Cases

+

monthFor...

sales_dat...

Submit

```
1 import pandas as pd
2
3 # Prompt the user for the file name
4 file_name = input()
5
6 # Load the data
7 df = pd.read_csv(file_name)
8
9
10 # Find the product with the highest total quantity sold
11 product_sales = df.groupby("Product")["Quantity"].sum()
12 best_product = product_sales.idxmax()
13 highest_quantity = product_sales.max()
14
15 # Display the result
16 print(f"Best selling product: {best_product}")
17 print(f"Total quantity sold: {highest_quantity}")
18
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

4.2.3. City that Sold the Most Products

03:34

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 city and calculate the total quantity of products sold for each city.
- Find the city that sold the most products (based on the total quantity sold).

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

Sample Test Cases

monthFor... sales_dat...

Submit

```

1 import pandas as pd
2
3 # Prompt the user for the file name
4 file_name = input()
5
6 # Load the data
7 df = pd.read_csv(file_name)
8
9 # write the code..
10 city_sales = df.groupby("City")["Quantity"].sum()
11
12 best_city = city_sales.idxmax()
13 # Display the result
14 print(f"City sold the most products: {best_city}")
15

```

Terminal Test cases

Activate Windows
Go to Settings to activate Windows.

4.2.4. Most Frequently Sold Product Pairs

15:25



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
```

Sample Test Cases



```
frequentl... sales_dat... Submit
1 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
9 df = pd.read_csv(file_name)
10
11 # write the code
12 date_products = {}
13
14 for date, group in df.groupby('Date'):
15     products = group['Product'].unique()
16     if len(products) > 1:
17         date_products[date] = products
18
19 pair_counter = Counter()
20
21 for products in date_products.values():
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

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

Sample Test Cases

```
15 products = group['Product'].unique()
16 if len(products) > 1:
17     date_products[date] = products
18
19 pair_counter = Counter()
20
21 for products in date_products.values():
22     pairs = combinations(sorted(products), 2)
23     pair_counter.update(pairs)
24
25 if pair_counter:
26     max_count = max(pair_counter.values())
27
28 for pair, count in pair_counter.items():
29     if count == max_count:
30         print(f"{pair[0]} and {pair[1]}: {count} times")
31 else:
32     print("No product pairs found.")
33
34
35 # Output the most frequent product pairs
```

4.2.5. Titanic Dataset Analysis and Data Cleaning

10:26

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

+

Explorer

titanicDat...

⌂

Submit

Debugger

```
1 import pandas as pd
2 import numpy as np
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # 1. Display the first 5 rows of the dataset
8 print(data.head())
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
20 print(data.describe())
21
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

4.2.5. Titanic Dataset Analysis and Data Cleaning

10:26

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

+

titanicDat...

Submit

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

[Terminal](#) [Test cases](#)

Activate Windows
Go to Settings to activate Windows.

[< Prev](#) [Reset](#) [Submit](#) [Next >](#)

30:28

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

The Titanic dataset contains columns as shown below.

[illegible]

+

Submit

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

Terminal Test cases

Activate Windows
Go to Settings to activate Windows.

4.2.6. Titanic Dataset Analysis and Data Cleaning - 2

30:28

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

The Titanic dataset contains columns as shown below,

--	--	--	--	--	--	--	--	--	--	--

Sample Test Cases

+

titanicDat...

Submit

```
20
21 # 5. Get the median fare of passengers
22 median_fare = data['Fare'].median()
23 print(median_fare)
24
25 # 6. Get the number of passengers by class
26 passengers_by_class = data['Pclass'].value_counts()
27 print(passengers_by_class)
28
29 # 7. Get the number of passengers by gender
30 passengers_by_gender = data['Sex'].value_counts().sort_index()
31 print(passengers_by_gender)
32
33 # 8. Get the number of passengers by survival status
34 passengers_by_survival =
35 data['Survived'].value_counts().sort_index()
36 print(passengers_by_survival)
37
38 # 9. Calculate the survival rate
39 survival_rate = data['Survived'].mean()
40 print(survival_rate)
```

Terminal Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev Reset Submit Next >

30:28

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

The Titanic dataset contains columns as shown below.

[illegible]

+

 Submit

```

26 passengers_by_class = data['Pclass'].value_counts()
27 print(passengers_by_class)
28
29 # 7. Get the number of passengers by gender
30 passengers_by_gender = data['Sex'].value_counts().sort_index()
31 print(passengers_by_gender)
32
33 # 8. Get the number of passengers by survival status
34 passengers_by_survival =
35 data['Survived'].value_counts().sort_index()
36 print(passengers_by_survival)
37
38 # 9. Calculate the survival rate
39 survival_rate = data['Survived'].mean()
40 print(survival_rate)
41
42 # 10. Calculate the survival rate by gender
43 survival_rate_by_gender = data.groupby('Sex')
44 ['Survived'].mean()
45 print(survival_rate_by_gender)

```

Terminal Test cases

Activate Windows
Go to Settings to activate Windows.

50:02

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).

Sample Test Cases +

```

1 import pandas as pd
2 import numpy as np
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6 data['FamilySize'] = data['SibSp'] + data['Parch']
7 data['IsAlone'] = np.where(data['FamilySize'] > 0, 0, 1)
8 data = pd.get_dummies(data, columns=['Embarked'],
9 drop_first=True)
10
11 # 1. Calculate the survival rate by class
12 print(data.groupby('Pclass')['Survived'].mean())
13
14 # 2. Calculate the survival rate by embarked location
15 print(data.groupby('Embarked_S')['Survived'].mean())
16
17 # 3. Calculate the survival rate by family size
18 print(data.groupby('FamilySize')['Survived'].mean())
19
20 # 4. Calculate the survival rate by being alone
21 print(data.groupby('IsAlone')['Survived'].mean())

```

50:02

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,

[illegible]

+

Submit

```

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

```

Terminal Test cases

Activate Windows
Go to Settings to activate Windows.

4.2.8. Titanic Dataset Analysis and Data Cleaning - 4 34:06

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.

The Titanic dataset contains columns as shown below,

Survived	PassengerId	Survived	PassengerId	Survived	PassengerId	Survived	PassengerId	Survived	PassengerId

Sample Test Cases
+

titanicDat... Submit

```

1 import pandas as pd
2 import numpy as np
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6 data = pd.get_dummies(data, columns=['Embarked'],
7 drop_first=True)
8
9 survivors_by_gender = data[data['Survived'] == 1]
10 ['Sex'].value_counts()
11 print(survivors_by_gender)
12
13 # 2. Get the number of non-survivors by gender
14 non_survivors_by_gender = data[data['Survived'] == 0]
15 ['Sex'].value_counts()
16 print(non_survivors_by_gender)
17
18 # 3. Get the number of survivors by embarked location
19 survivors_by_embarked_s = data[data['Survived'] == 1]
20 ['Embarked_S'].value_counts()
    
```

Terminal
Test cases

Activate Windows
Go to Settings to activate Windows.

34:06

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.

The Titanic dataset contains columns as shown below.

[illegible]

+

Submit

```

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]
22   ['Embarked_S'].value_counts()
23 print(non_survivors_by_embarked_s)
24
25 # 5. Calculate the percentage of children (Age < 18) who
26   survived
27 children = data[data['Age'] < 18]
28 children_survival_rate = children['Survived'].mean()
29 print(children_survival_rate)
30
31 # 6. Calculate the percentage of adults (Age >= 18) who
32   survived
33 adults = data[data['Age'] >= 18]
34 adults_survival_rate = adults['Survived'].mean()
35 print(adults_survival_rate)
36
37 # 7. Get the median age of survivors

```

Terminal Test cases

Activate Windows

Go to Settings to activate Windows.

4.2.8. Titanic Dataset Analysis and Data Cleaning - 4

34:06

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.

The Titanic dataset contains columns as shown below,

Sample Test Cases

+

titanicDat...

Submit

```
33
34 # 7. Get the median age of survivors
35 median_age_survivors = data[data['Survived'] == 1]
36   ['Age'].median()
37   print(median_age_survivors)
38
39 # 8. Get the median age of non-survivors
40 median_age_non_survivors = data[data['Survived'] == 0]
41   ['Age'].median()
42   print(median_age_non_survivors)
43
44 # 9. Get the median fare of survivors
45 median_fare_survivors = data[data['Survived'] == 1]
46   ['Fare'].median()
47   print(median_fare_survivors)
48
49 # 10. Get the median fare of non-survivors
50 median_fare_non_survivors = data[data['Survived'] == 0]
51   ['Fare'].median()
52   print(median_fare_non_survivors)
```

[Terminal](#) [Test cases](#)

Activate Windows
Go to Settings to activate Windows.

[< Prev](#) [Reset](#) [Submit](#) [Next >](#)