

Practical 4

About this unit

Practical 4

Practice Lab Assignment

Unit • 100% completed



Lab Assignment

Unit • 100% completed



 CODETANTRA

4.1.1. Pandas - series creation and manipul...

17:20 A ☾ ☽

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.

import pandas as pd

```
1 import pandas as pd
2
3 # Take inputs from the user to
4 # create a list of numbers
5 numbers = list(map(int,
6     input().split()))
7
8 # Create a Pandas series from the
9 # list of numbers
10 series = pd.Series(numbers)
11 # Grouping by even and odd numbers
12 # and calculating the mean
13 grouped = series.groupby(series % 2
14 == 0).mean()
15
16 # Display the mean of even and odd
17 # numbers with labels
18 grouped.index = ['Even' if is_even
19 else 'Odd' for is_even in
20 grouped.index]
21
22 print("Mean of even and odd
23 numbers:")
24 print(grouped)
```

Sample Test Cases

< Prev Reset Submit Next >

CODETANTRA

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.

```

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
10 # DataFrame
11 df = pd.DataFrame(data)
12
13 # Display the original DataFrame
14 print("Original DataFrame:")
15 print(df)
16
17 # Adding a new row
18 new_name = input("New name: ")
19 new_age = int(input("New age: "))
20 df.loc[len(df)] = [new_name, new_age]
21
22 # Display the DataFrame after adding
23 # a new row
24 print("After adding a row:\n", df)
25
26 # Modifying a row
27 mod_index = int(input("Index of row
28 to modify: "))
29 new_age = int(input("New age: "))
30 if 0 <= mod_index < len(df):
31     df.at[mod_index, 'Age'] = new_age
32 else:
33     print("Invalid index!")
34
35 # Display the DataFrame after
36 # modifying a row
37 print("After modifying a row:")
38 print(df)
39
40 # Deleting a row
41 del_index = int(input("Index of row
42 to delete: "))
43 if 0 <= del_index < len(df):
44     df = df.drop(del_index).reset_index(drop=True)
45 else:
46     print("Invalid index!")
47
48 # Display the DataFrame after
49 # deleting a row
50 print("After deleting a row:")
51 print(df)
52
53 # Adding a new column
54 genders = input("Enter genders
55 separated by space: ").split()
56 df['Gender'] = genders
57
58 # Display the DataFrame after
59 # adding a new column
60 print("After adding a new column:")
61 print(df)
62
63 # Modifying a column
64 df['Name'] = df['Name'].str.upper()
65
66 # Display the DataFrame after
67 # modifying a column
68 print("After modifying a column:")
69 print(df)
70
71 # Deleting a column
72 df.drop('Age', axis=1, inplace=True)
73
74 # Display the DataFrame after
75 # deleting a column
76 print(df)
    
```

Sample Test Cases +

< Prev Reset Submit Next >

 CodeTANTRA

4.1.3. Student Information

24.42 A ⚡ -

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.

```
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=["Name", "Age", "Grade"])
6 print("First five rows:")
7 print(data.head())
8 average_age =
9 round(data["Age"].mean(), 2)
10 print("Average age:", average_age)
11 print("Students with a grade up to B")
12 filtered_students =
13 data[data["Grade"].isin(['A+', 'A', 'B'])]
14 print(filtered_students)
```

Sample Test Cases +

< Prev Reset Submit Next >

Lab Assignment

About this unit

Lab Assignment

`</> Month with the Highest Total Sales`

Question



`</> Best Selling Product`

Question



`</> City that Sold the Most Products`

Question



`</> Most Frequently Sold Product Pairs`

Question



`</> Titanic Dataset Analysis and Data Cleaning`

Question



`</> Titanic Dataset Analysis and Data Cleaning - 2`

Question



`</> Titanic Dataset Analysis and Data Cleaning - 3`

Question



`</> Titanic Dataset Analysis and Data Cleaning - 4`

Question



CODETANTRA

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 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
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.

```
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 df['Date'] =
10 pd.to_datetime(df['Date'])
11 df['Month'] =
12 df['Date'].dt.to_period('M').astype(s
13 tr)
14 df['Total'] = df['Quantity'] *
15 df['Price']
16
17 monthly_sales = df.groupby('Month')
18 ['Total'].sum()
19
20 # Find the month with the highest
21 total sales
22 best_month = monthly_sales.idxmax()
23 highest_sales = monthly_sales.max()
24
25 print(f"Best month: {best_month}")
26 print(f"Total sales:
27 ${highest_sales:.2f}")
```

Sample Test Cases +

< Prev Reset Submit Next >

CODETANTRA

4.2.2. Best Selling Product

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
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.

Sample Test Cases +

```
import pandas as pd

# Prompt the user for the file name
file_name = input()

# Load the data
df = pd.read_csv(file_name)
product_sales = df.groupby('Product')[['Quantity']].sum()

# Find the product with the highest total quantity sold
best_product = product_sales.idxmax()
highest_quantity = product_sales.max()

# Display the result
print(f"Best selling product: {best_product}")
print(f"Total quantity sold: {highest_quantity}")
```

< Prev Reset Submit Next >

CODETANTRA

4.2.3. City that Sold the Most Products 02:34 A ⚡ -

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
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.

Sample Test Cases +

```
import pandas as pd

# Prompt the user for the file name
file_name = input()

# Load the data
df = pd.read_csv(file_name)

# write the code..
city_sales = df.groupby('City')[['Quantity']].sum()
best_city = city_sales.idxmax()

# Display the result
print(f"City sold the most products: {best_city}")
```

< Prev Reset Submit Next >

CODETANTRA

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.

Sample Test Cases +

```
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
9 file
10 df = pd.read_csv(file_name)
11
12 # write the code
13 date_products = {}
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():
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
# Output the most frequent product pairs
```

CODETANTRA

4.2.5. Titanic Dataset Analysis and Data Cle... 41:49 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. 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,

P	a	S	p	N	s	A	S	P	T	F	C	E
s	s	u	c	a	e	g	b	a	i	a	a	m
e	r	v	i	a	x	e	S	r	c	r	b	b
I	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,STON/O2. 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 (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 +

```

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

< Prev Reset Submit Next >

CODETANTRA

4.2.6. Titanic Dataset Analysis and Data Cle... 34/12 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.

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,

P	a	s	S	u	P	c	N	a	m	S	A	i	b	P	a	T	i	c	k	e	F	a	C	b	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,STON/O2. 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 +

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

< Prev Reset Submit Next >

☰ CODETANTRA

4.2.7. Titanic Dataset Analysis and Data Cle... 27:34 AA ⚡ ↻ ⌂ ⌂

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 Data:

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

```
titanicData...  
Submit  
Debugger  
Explorer  
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'], drop_first=True)  
9  
10 print(data.groupby('Pclass')  
['Survived'].mean())  
11  
12 # 2. Calculate the survival rate by  
embarked location  
13 print(data.groupby('Embarked_S')  
['Survived'].mean())  
14  
15 # 3. Calculate the survival rate by  
family size  
16 print(data.groupby('FamilySize')  
['Survived'].mean())  
17  
18 # 4. Calculate the survival rate by  
being alone  
19 print(data.groupby('IsAlone')  
['Survived'].mean())  
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())
```

Sample Test Cases

+

[◀ Prev](#) [Reset](#) [Submit](#) [Next ▶](#)



CODETANTRA

4.2.8. Titanic Dataset Analysis and Data Cle... 29/12 A C E -

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,

P	a	s	S	u	P	N	S	A	S	P	T	F	C	E
a	s	s	e	r	c	a	e	g	i	a	i	a	b	m
s	e	v	i	s	I	m	x	e	b	r	c	r	a	b
P	a	s	S	u	P	N	S	A	S	P	T	F	C	E
a	s	s	e	r	I	m	x	e	b	r	i	a	b	m
s	e	v	i	s	a	m	e	g	a	r	c	r	a	b
e	r	e	s	s	s	a	m	e	b	r	k	e	b	a
r	l	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,STON/O2. 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.

```
titanicData...
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)

survivors_by_gender =-
data[data['Survived'] == 1]-
['Sex'].value_counts()
print(survivors_by_gender)

# 2. Get the number of non-survivors
by gender
non_survivors_by_gender =-
data[data['Survived'] == 0]-
['Sex'].value_counts()
print(non_survivors_by_gender)

# 3. Get the number of survivors by
embarked location
survivors_by_embarked_s =-
data[data['Survived'] == 1]-
['Embarked_S'].value_counts()
print(survivors_by_embarked_s)

# 4. Get the number of non-survivors
by embarked location
non_survivors_by_embarked_s =-
data[data['Survived'] == 0]-
['Embarked_S'].value_counts()
print(non_survivors_by_embarked_s)

# 5. Calculate the percentage of
children (Age < 18) who survived
children=data[data['Age'] < 18]
children_survival_rate=-
children['Survived'].mean()
print(children_survival_rate)

# 6. Calculate the percentage of
adults (Age >= 18) who survived
adults= data[data['Age'] >=18]
adults_survival_rate=-
adults['Survived'].mean()
print(adults_survival_rate)

# 7. Get the median age of survivors
median_age_survivors =-
data[data['Survived'] == 1]-
['Age'].median()
print(median_age_survivors)

# 8. Get the median age of non-
survivors
median_age_non_survivors =-
data[data['Survived'] == 0]-
['Age'].median()
print(median_age_non_survivors)

# 9. Get the median fare of survivors
median_fare_survivors =-
data[data['Survived'] == 1]-
['Fare'].median()
print(median_fare_survivors)

# 10. Get the median fare of non-
survivors
median_fare_non_survivors =-
data[data['Survived'] == 0]-
['Fare'].median()
print(median_fare_non_survivors)
```

Sample Test Cases +