

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=5773e362f19c5320ca6bd5#contents/5773e455f19c5320ca6bd19/6773e4e4f19c5320ca6bdce/57860994c50fe56b8dcf8747

CODETANTRA

Home

202401050058@mitaoe.ac.inSupportLogout

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.

Sample Test Cases

monthFor...

sales_dat...

Submit

1import pandas as pd

2

3# Prompt the user for the file name

4file_name = input()

5

6# Load the data

7df = pd.read_csv(file_name)

8df['Date'] = pd.to_datetime(df['Date'])

9

10df['Month'] = df['Date'].dt.to_period('M')

11

12df['Total_Sales'] = df['Quantity'] * df['Price']

13

14monthly_sales = df.groupby('Month')['Total_Sales'].sum()

15

16# Find the month with the highest total sales

17best_month = monthly_sales.idxmax()

18highest_sales = monthly_sales.max()

19

20print(f"Best month: {best_month}")

21print(f"Total sales: \${highest_sales:.2f}")

22

TerminalTest cases

PrevResetSubmitNext

BA FINANCE

+126%

Search

13:41

07-05-2025

New repository

Course

mitaoe.codetantra.com/secure/course.jsp?moduleId=5773e3f2f19c5320ca6bd5#contents/5773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bd2/667cf60f2366ba61ef5a2ce7

CODETANTRA

Home

202401050058@mitaoe.ac.in

Support

Logout

4.1.3. Student Information

100.0%

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

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=["Name", "Age", "Grade"])

6

print("First five rows:")

7

print(data.head(5))

8

write your code here..

9

age=round(data['Age'].mean(),2)

10

print("Average age:",age)

11

print("Students with a grade up to B")

12

df=pd.DataFrame(data)

13

a=df[df['Grade']<='B']

14

print(a)

15

Terminal

Test cases

Prev

Reset

Submit

Next

BA FINANCE

+1.28%

Search

13:41

07-05-2025

New repository

Course

mitaoe.codetantra.com/secure/course.jsp?moduleId=5773e3f2f19c5320ca6bd5#contents/5773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bd2/667a7e04282204636497a3c

mitaoe.codetantra.com

202401050058@mitaoe.ac.in

Support

Logout

4.1.1. Pandas - series creation and manipulation

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...

```
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 series = pd.Series(numbers)
8 # Grouping by even and odd numbers and calculating the mean
9 grouped = series.groupby(series % 2==0).mean()
10
11 # Display the mean of even and odd numbers with labels
12 grouped.index = ['Even' if is_even else 'Odd' for is_even in grouped.index]
13 print("Mean of even and odd numbers:")
14 print(grouped)
15
```

Terminal

Test cases

BAI FINANCE

+1.28%

Search

11:41

07-05-2023

New repository

Course

mitaoe.codetanttra.com/course.jsp?moduleId=6773e382f19c5320ca6bd5#contents/6773e455f19c5320ca6bd19/6773e4d5f19c5320ca6bd2/667aa3078220463649bb62

CODETANTRA

20048109058@mitaoe.com

Support

Logout

4.1.2. Dictionary to dataframe

100%

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.

Sample Test Cases

dataframe...

```
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 = df.append(new_row, ignore_index=True)
21 # Display the DataFrame after adding a new row
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 = int(input("New age: "))
27 df.loc[modify_index, 'age'] = new_age
28 # Display the DataFrame after modifying a row
29 print("After modifying a row:\n", df)
30
31 # Deleting a row
32 delete_index = int(input("Index of row to delete: "))
33 df = df.drop(delete_index, axis=0)
34 # Display the DataFrame after deleting a row
35 print("After deleting a row:\n", df)
36
37 # Adding a new column
38 gender_input = input("Enter genders separated by space: ")
39 genders = gender_input.split()
40 df['gender'] = genders
41 # Display the DataFrame after adding a new column
42 print("After adding a new column:\n", df)
43
44 # Modifying a column
45 df['name'] = df['name'].str.upper()
46 # Display the DataFrame after modifying a column
47 print("After modifying a column:\n", df)
48
49 # Deleting a column
50 df = df.drop(columns=['age'])
51 # Display the DataFrame after deleting a column
52 print("After deleting a column:\n", df)
```

Terminal Test cases

BAFINANCE

+1.26%

Search

13:41

07-05-2023

mitaoe.codetantra.com/course.jsp?ucId=6773e3f1f9c5320ca6bd5#contents/6773e45f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e3b9303025661d3ae574f

CodeTANTRA

202401090058@mitaoe.ac.in · Support

Logout

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.

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,

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,0,5,23714,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,3,0,17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2, 3101202,7.925,,S
4,1,1,"Vuttrille, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113683,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.85,,S
6,0,3,"Moran, Mr. James",male,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,34,0,0,17463,51.0625,E06,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,3,349000,21.075,,S
9,1,3,"Jonnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,31.1333,,S
10,1,2,"Mason, Mrs. Nicholas (Adele Aiche)",female,14,1,0,237736,39.6708,,C

Sample Test Cases

titanicDat...

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

import pandas as pd
import numpy as np

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

1. Display the first 5 rows of the dataset
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)

4. Get a summary of the dataset (info)
print(data.info())

5. Get basic statistics of the dataset
print(data.describe())

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()
data['Age'].fillna(median_age, inplace=True)
8. Fill missing values in the 'Embarked' column with the mode
mode_embarked = data['Embarked'].mode()[0]
data['Embarked'].fillna(mode_embarked, inplace=True)
9. Drop the 'Cabin' column due to many missing values
data.drop('Cabin', axis=1, inplace=True)

10. Create a new column 'FamilySize' by adding 'SibSp' and 'Parch'
data['FamilySize'] = data['SibSp'] + data['Parch']

Terminal

Test cases

BAFINANCE

+126%

Search

13:42

07-05-2023

4.2.7. Titanic Dataset Analysis and Data Cleaning - 3

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]

Sample Data:

[illegible]

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

Sample Test Cases

```

1  # Titanic Data Set
2  import pandas as pd
3  import numpy as np
4
5  # Load the Titanic dataset
6  data = pd.read_csv('Titanic-Dataset.csv')
7  data['FamilySize'] = data['SibSp'] + 4 * data['Parch']
8  data['IsAlone'] = np.where(data['FamilySize'] > 0, 0, 1)
9  data = pd.get_dummies(data, columns=['embarked'], 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_5')['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())
22
23 # 5. Get the average fare by class
24 print(data.groupby('Pclass')['Fare'].mean())
25
26 # 6. Get the average age by class
27 print(data.groupby('Pclass')['Age'].mean())
28
29 # 7. Get the average age by survival status
30 print(data.groupby('Survived')['Age'].mean())
31
32 # 8. Get the average fare by survival status
33 print(data.groupby('Survived')['Fare'].mean())
34
35 # 9. Get the number of survivors by class
36 print([data[data['Survived'] == 1]['Pclass'].value_counts()])
37
38 # 10. Get the number of non-survivors by class
39 print([data[data['Survived'] == 0]['Pclass'].value_counts()])

```


Course

mitaoe.codedtantra.com/secure/course.jsp?eucId=6773e32f1f9c5320ca6bd5#contents/6773e455f1f9c5320ca6bd19/6773e4c4f1f9c5320ca6bdce/67c378a17028471c94dad0a6

CODETANTRA

2024/10/07 09:07:58 mlase.x.m Support Logout

4.2.B. 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:

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:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

```
passengerid,survived,pclass,name,sex,age,sibsp,parch,ticket,fare,cabin,embarked
1,0,1,"Braund, Mr. Owen Harris",male,22,1,0,0,1137,7.25,S
2,1,1,"Cummings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,0,17900,71.2833,C85,C
3,1,1,"Heikkinen, Miss. Laina",female,26,0,0,0,1796/32, 2101292,7.925,S
4,1,1,"Hartvig, Mrs. Jacques Heath (Lily May Peel)",female,29,1,0,1,12887,11.1,C123,S
5,0,1,"Kallio, Mr. William Henry",male,35,0,0,0,17143,8.45,S
6,0,1,"Korari, Mr. Joseph",male,30,0,0,0,17143,8.45,S
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,0,17463,51.6625,C44,S
8,0,1,"Malison, Master. Gustav Leonard",male,2,1,1,1,142005,12.875,S
9,1,1,"Mason, Mrs. Dorcas W (Elizabeth Vilhelmina Berg)",female,47,0,1,1,147142,31.555,S
10,1,1,"Messer, Mrs. Mircelot (Adele Achon)",female,18,1,0,1,17735,30.078,C
```

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

Sample Test Cases

titanicDat...

```
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'], drop_first=True)
16
17 # 4. Get the mean age of passengers
18 mean_age = data['Age'].mean()
19 print(mean_age)
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 = data['Survived'].value_counts().sort_index()
35 print(passengers_by_survival)
36
37 # 9. Calculate the survival rate
38 survival_rate = data['Survived'].mean()
39 print(survival_rate)
40
41 # 10. Calculate the survival rate by gender
42 survival_rate_by_gender = data.groupby('Sex')['Survived'].mean()
43 print(survival_rate_by_gender)
44
```

Terminal Test Cases

BAFINANCE

+1.26%

Search

13:42

07-05-2025

New repository

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=5773e3f2f19c5320ca6bd19/6773e4e4f19c5320ca6bdce/67861370e50te56b8dcf9620

CODETANTRA

Home

202401090058@mitaoe.ac.inSupportLogout

4.2.2. Best Selling Product

95%

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

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

BAFINANCE

+126%

Search

ENG

11:42

07-05-2025

New repository

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=5773e3f2f19c5320ca6bd5#contents/5773e455f19c5320ca6bdce/57861553e50fe56b8dcf9a6e

CODETANTRAHome

202401090058@mitaoe.ac.inSupportLogout

4.2.3. City that Sold the Most Products

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

monthFor...

sales_dat...

Submit

1import pandas as pd

2

3# Prompt the user for the file name

4file_name = input()

5

6# Load the data

7df = pd.read_csv(file_name)

8

9# write the code..

10city_sales = df.groupby("city")["Quantity"].sum()

11

12# Find the city with the highest total quantity sold

13best_city = city_sales.idxmax()

14# Display the result

15print(f"City sold the most products: {best_city}")

16

TerminalTest cases

PrevResetSubmitNext

BA FINANCE

+126%

Search

11:42

07-05-2025

New repositoryCourse

mitaoe.codetantra.com/secure/course.jsp?euid=5773e362f19c5320ca6bd5#contents/5773e455f19c5320ca6bd19/6773e4e4f19c5320ca6bdce/5788ee09f79b6ee9e3ee1

CODETANTRAHome

202401050058@mitaoe.ac.inSupportLogout

4.2.4. Most Frequently Sold Product Pairs12/53

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.

Sample Test Cases

frequentli...sales_dat...

```
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 grouped=df.groupby('Date')['Product'].apply(list)
13 pair_counter=Counter()
14 # Output the most frequent product pairs
15 for products in grouped:
16     products=sorted(products)
17     pairs=combinations(products,2)
18     pair_counter.update(pairs)
19 max_count=max(pair_counter.values()) if pair_counter else 0
20 most_frequent_pairs=[pair for pair,count in pair_counter.items() if count==max_count]
21 most_frequent_pairs.sort()
22 for pair in most_frequent_pairs:
23     print(f"{pair[0]} and {pair[1]}: {max_count} times")
```

TerminalTest cases

< PrevResetSubmitNext >

BA FINANCE+126%

Search

ENGin

13:4207-05-2023

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=5773e363f19c5320ca6bd5#contents/5773e455f19c5320ca6bd19/6773e4c4f19c5320ca6bdce/67e367fc7028471d94db0238

CODETANTRA

2024/10/07 09:58:01 mlase.x.m Support Logout

4.2.8. 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:

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 < 16) who survived.

6. Calculate the percentage of adults (Age >= 16) 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.

Passenger Id	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,4,"Brande, Mr. Owen Harris", male, 22, 1, 0, 61, 7.25, 5
2,1,1,"Carnegie, Mrs. John Bradley (Florence Briggs Thayer)", female, 38, 1, 0, 91, 53.1, C85, 5
3,1,3,"Hicksdon, Miss. Laura", female, 26, 0, 0, 51, 9.5, 2101262, 7.925, 5
4,1,1,"Hart, Mrs. Jacques Heath (Lily May Peel)", female, 25, 1, 0, 31, 5.42, 2101263, 5.12, C223, 5
5,0,3,"Allen, Mr. William Henry", male, 35, 0, 0, 37, 8.66, 5
6,0,3,"Haran, Mr. Zena", male, 4, 0, 100277, 8.451, 2
7,0,1,"McCarthy, Mr. Timothy J", male, 54, 0, 0, 51, 51.86, 644, 5
8,0,3,"Palsson, Per", male, 26, 0, 0, 34, 16.59, 42, 475, 5
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)", female, 37, 0, 2, 34, 53.1, 5
10,1,1,"Nasser, Mrs. Nicholas (Adele Achem)", female, 54, 1, 0, 51, 51.86, 644, 5

Note: Refer to the visible test case for better reference

Sample Test Cases

titanicDat..

1import pandas as pd

2import numpy as np

3

4# Load the Titanic dataset

5data = pd.read_csv('Titanic-Dataset.csv')

6data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

7

8

9# 1. Get the number of survivors by gender

10survivors_by_gender = data[data['Survived'] == 1]['Sex'].value_counts()

11print(survivors_by_gender)

12# 2. Get the number of non-survivors by gender

13non_survivors_by_gender = data[data['Survived'] == 0]['Sex'].value_counts()

14print(non_survivors_by_gender)

15

16# 3. Get the number of survivors by embarked location

17survivors_by_embarked_s = data[data['Survived'] == 1]['Embarked_S'].value_counts()

18print(survivors_by_embarked_s)

19

20# 4. Get the number of non-survivors by embarked location

21non_survivors_by_embarked_s = data[data['Survived'] == 0]['Embarked_S'].value_counts()

22print(non_survivors_by_embarked_s)

23# 5. Calculate the percentage of children (Age < 16) who survived

24children = data[data['Age'] < 16]

25children_survival_rate = children['Survived'].mean()

26print(children_survival_rate)

27# 6. Calculate the percentage of adults (Age >= 16) who survived

28adults = data[data['Age'] >= 16]

29adults_survival_rate = adults['Survived'].mean()

30print(adults_survival_rate)

31# 7. Get the median age of survivors

32median_age_survivors = data[data['Survived'] == 1]['Age'].median()

33print(median_age_survivors)

34# 8. Get the median age of non-survivors

35median_age_non_survivors = data[data['Survived'] == 0]['Age'].median()

36print(median_age_non_survivors)

37# 9. Get the median fare of survivors

38median_fare_survivors = data[data['Survived'] == 1]['Fare'].median()

39print(median_fare_survivors)

40# 10. Get the median fare of non-survivors

41median_fare_non_survivors = data[data['Survived'] == 0]['Fare'].median()

42print(median_fare_non_survivors)

Terminal

Test Cases

BA FINANCE

+126%

Search

13:42

07-05-2025