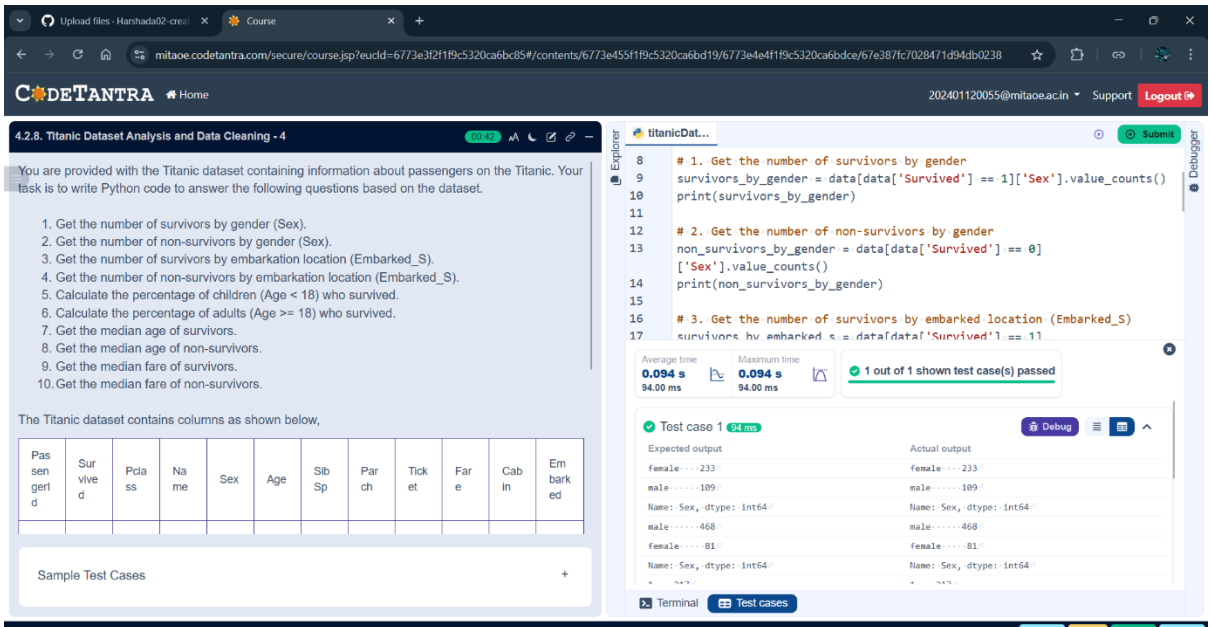
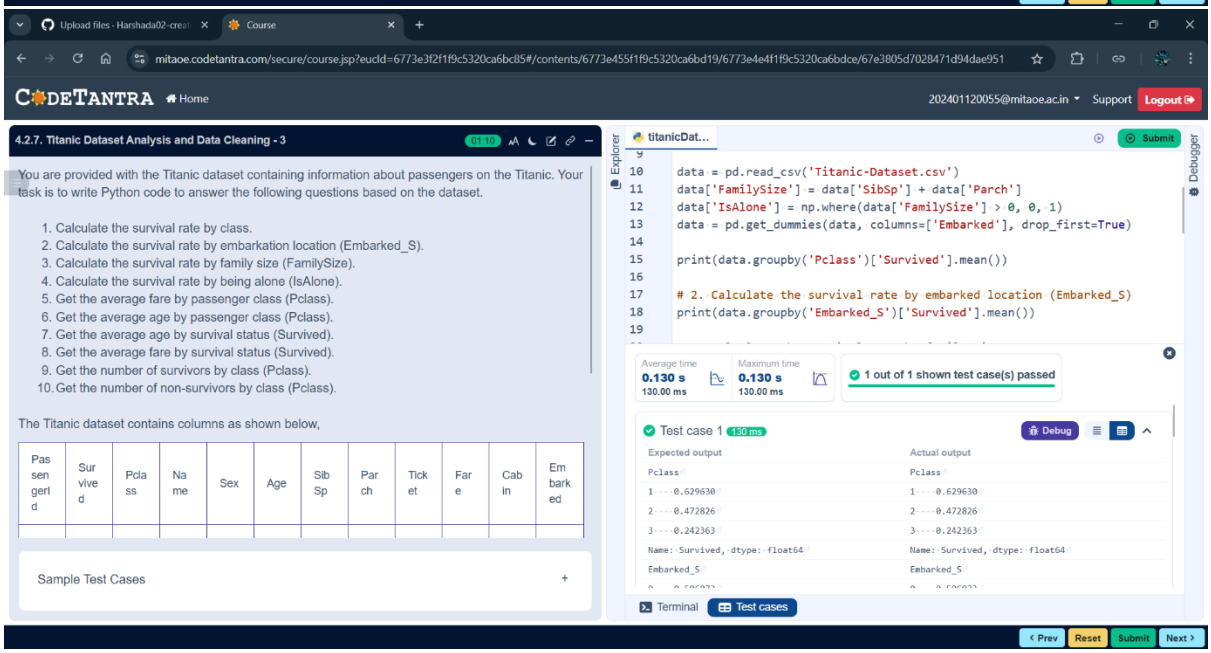
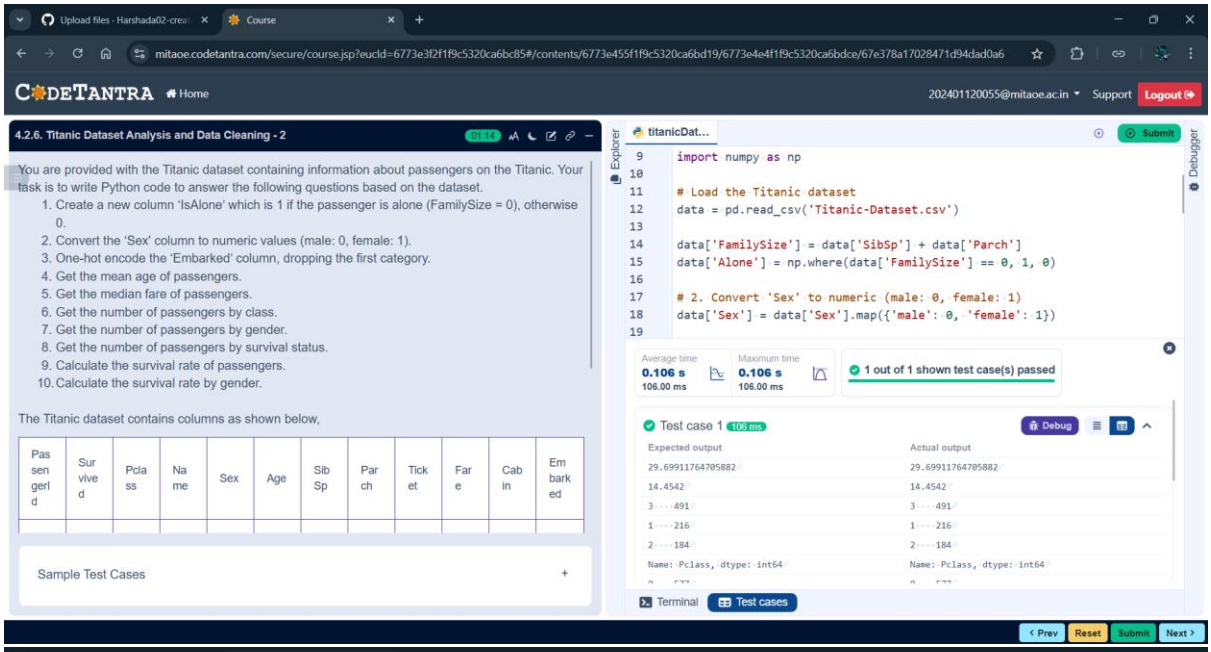
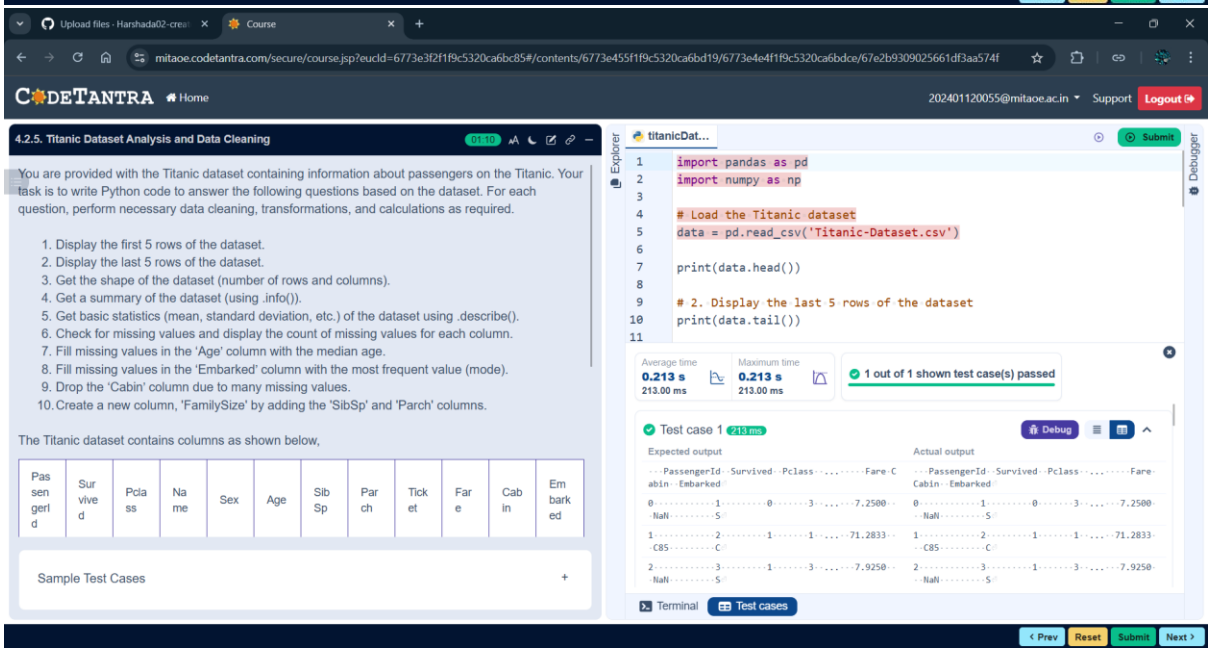
PRACTICAL 4

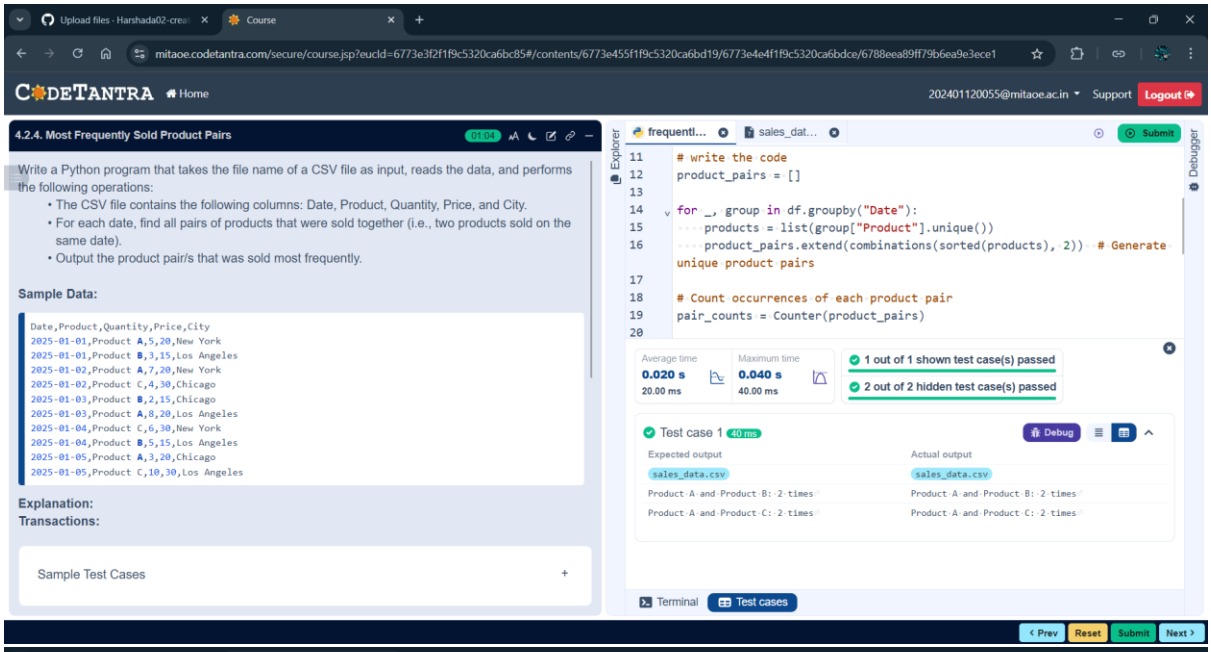
ALL PRACTICALS:

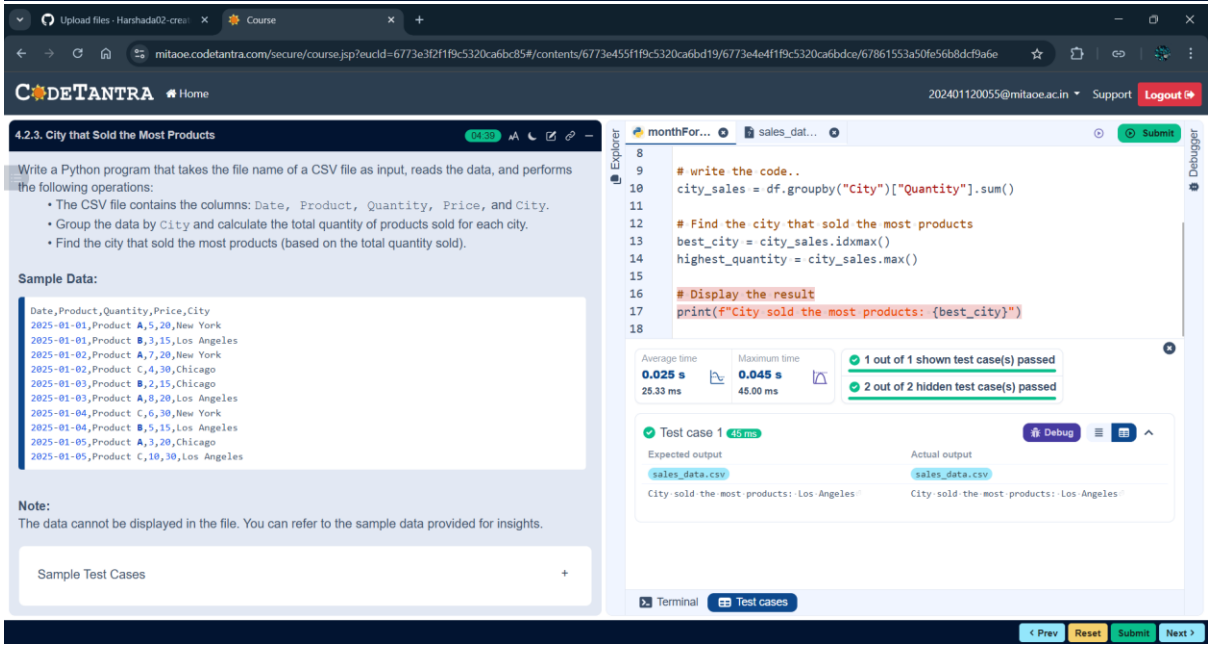


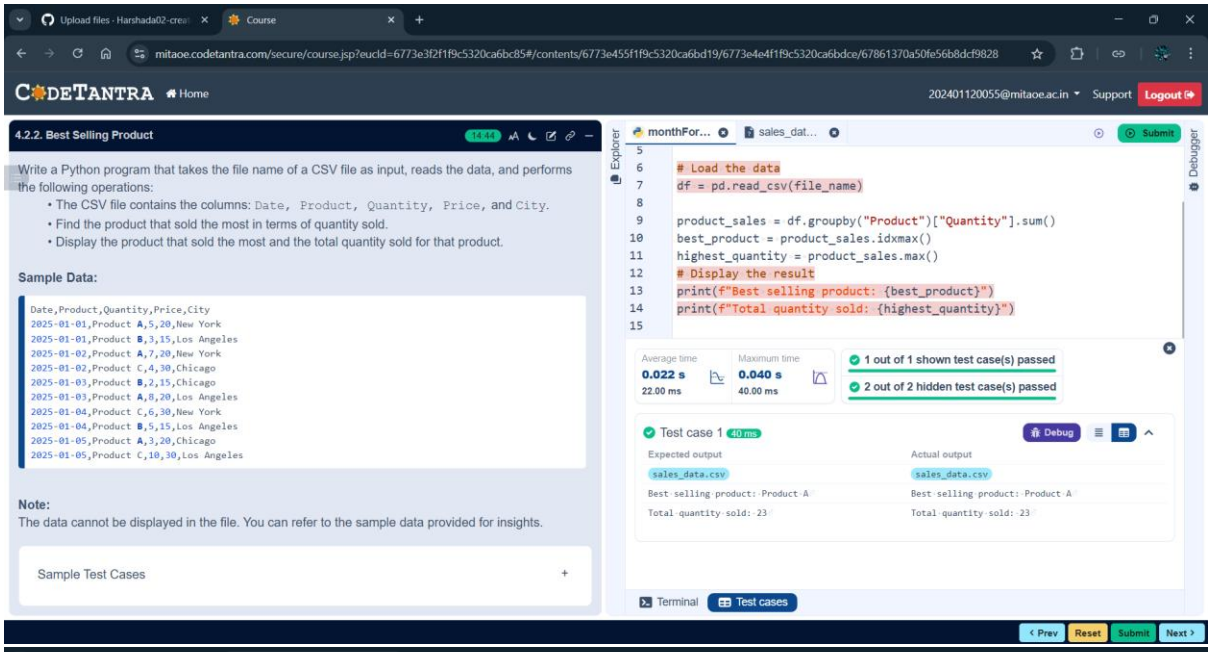


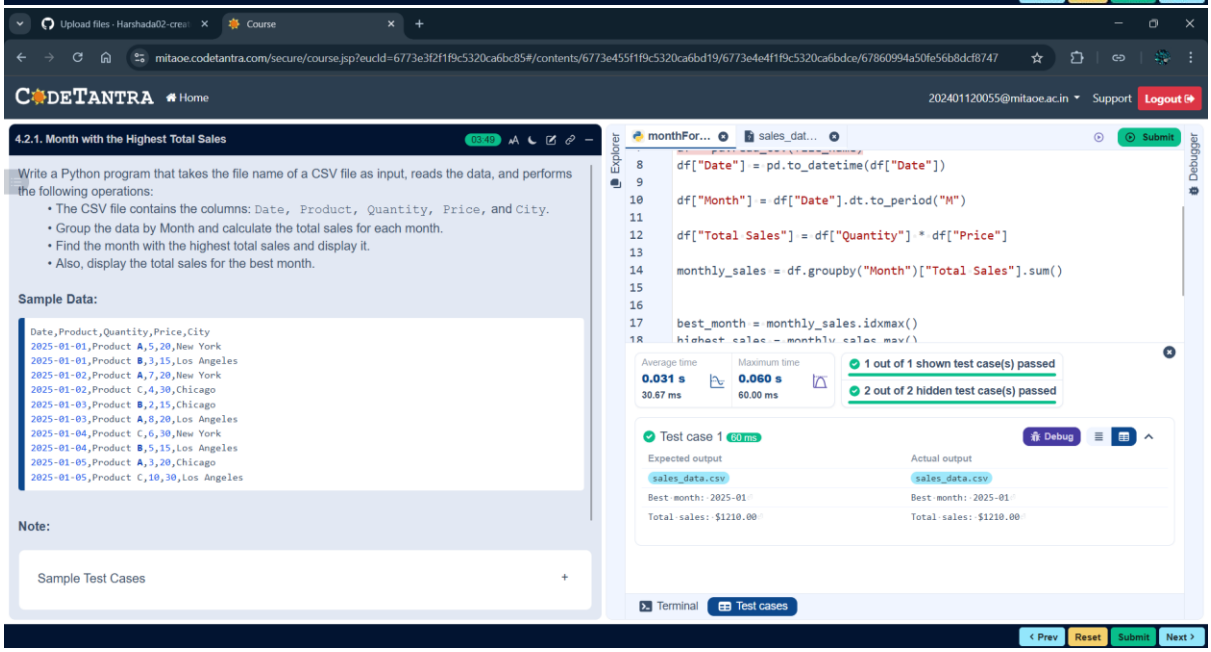

Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e378a17028471d94dad0a6
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
4.2.6. Titanic Dataset Analysis and Data Cleaning - 2
01:14
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 Test Cases
Explorer
titanicDat...
Submit
Debugger
import numpy as np
# Load the Titanic dataset
data = pd.read_csv('Titanic-Dataset.csv')
data['FamilySize'] = data['SibSp'] + data['Parch']
data['Alone'] = np.where(data['FamilySize'] == 0, 1, 0)
2. Convert 'Sex' to numeric (male: 0, female: 1)
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
Average time
0.106 s
106.00 ms
Maximum time
0.106 s
106.00 ms
1 out of 1 shown test case(s) passed
Test case 1
106 ms
Debug
Expected output
29.69911764705882
14.4542
3 491
1 216
2 184
Name: Pclass, dtype: int64
Actual output
29.69911764705882
14.4542
3 491
1 216
2 184
Name: Pclass, dtype: int64
Terminal
Test cases
Prev
Reset
Submit
Next

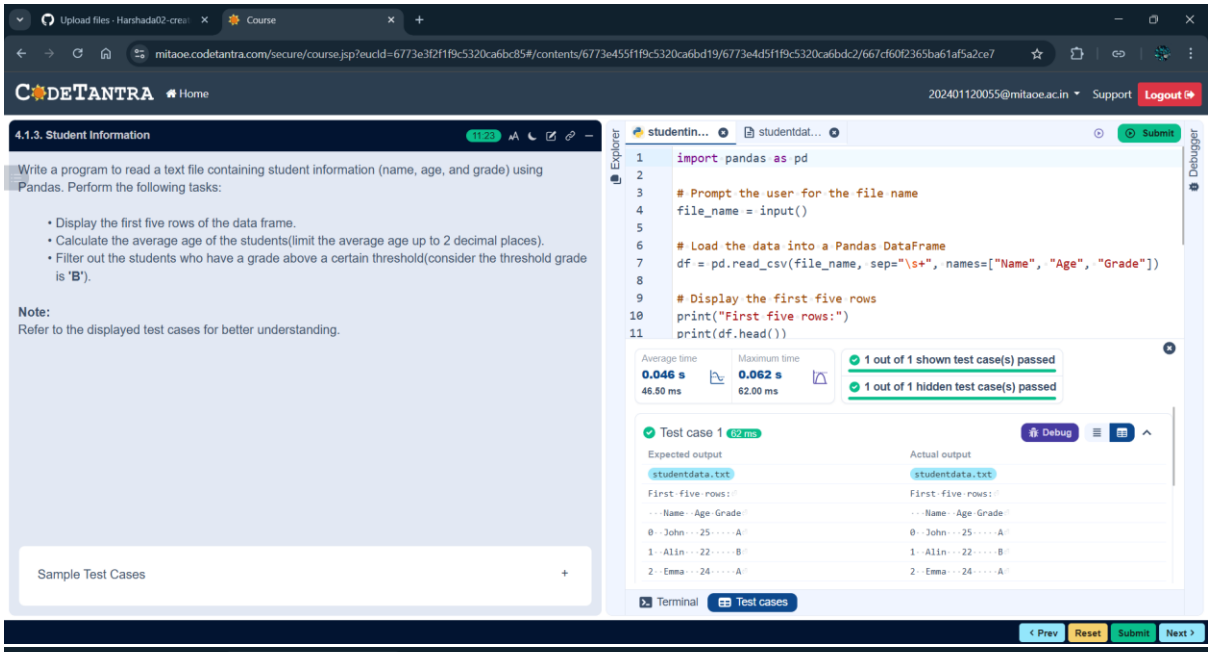

Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e2b9309025661df3aa574f
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
4.2.5. Titanic Dataset Analysis and Data Cleaning
01:10
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 Test Cases
Explorer
titanicDat...
Submit
Debugger
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())
Average time
0.213 s
213.00 ms
Maximum time
0.213 s
213.00 ms
1 out of 1 shown test case(s) passed
Test case 1
213 ms
Debug
Expected output
PassengerId Survived Pclass ... Fare Cabin Embarked
0 1 0 3 ... 7.2500 NaN S
1 2 1 1 ... 71.2833 C85 C
2 3 1 3 ... 7.9250 NaN S
Actual output
PassengerId Survived Pclass ... Fare Cabin Embarked
0 1 0 3 ... 7.2500 NaN S
1 2 1 1 ... 71.2833 C85 C
2 3 1 3 ... 7.9250 NaN S
Terminal
Test cases
Prev
Reset
Submit
Next

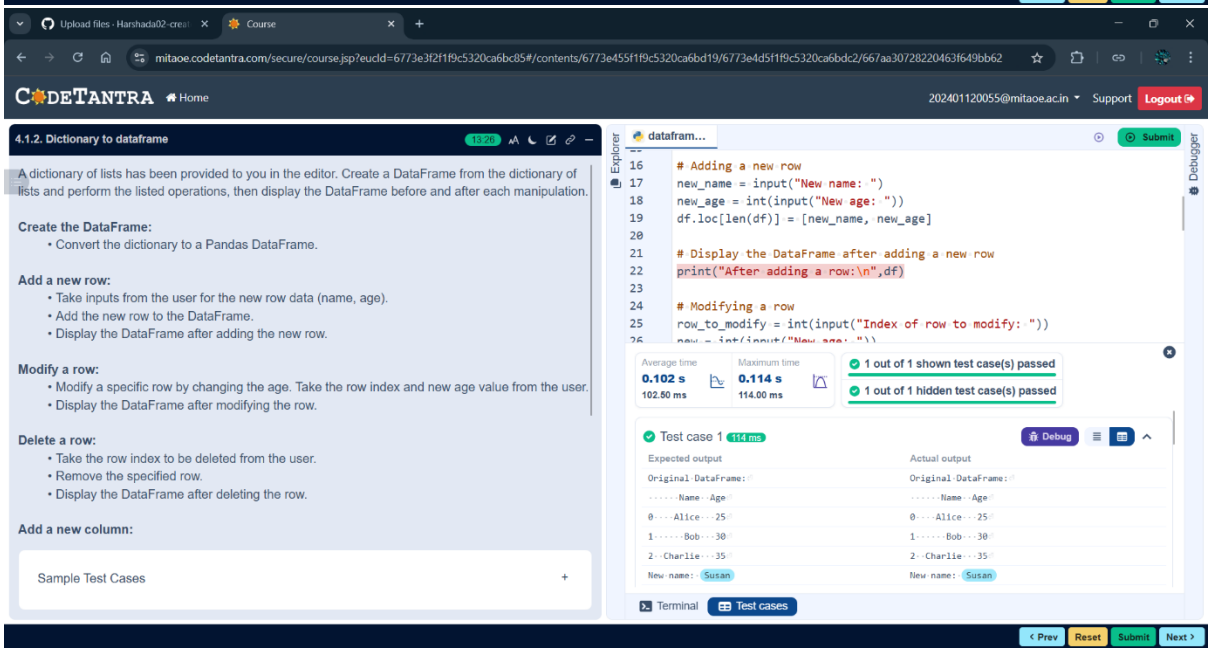

Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/6788eea89ff79b6ea9e3ece1
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
4.2.4. Most Frequently Sold Product Pairs
01:04
Write 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:
Sample Test Cases
Explorer
frequentl...
sales_dat...
Submit
Debugger
# write the code
product_pairs = []
for _, group in df.groupby("Date"):
products = list(group["Product"].unique())
product_pairs.extend(combinations(sorted(products), 2)) # Generate unique product pairs
# Count occurrences of each product pair
pair_counts = Counter(product_pairs)
Average time
0.020 s
20.00 ms
Maximum time
0.040 s
40.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
40 ms
Debug
Expected output
sales_data.csv
Product A and Product B: 2 times
Product A and Product C: 2 times
Actual output
sales_data.csv
Product A and Product B: 2 times
Product A and Product C: 2 times
Terminal
Test cases
Prev
Reset
Submit
Next

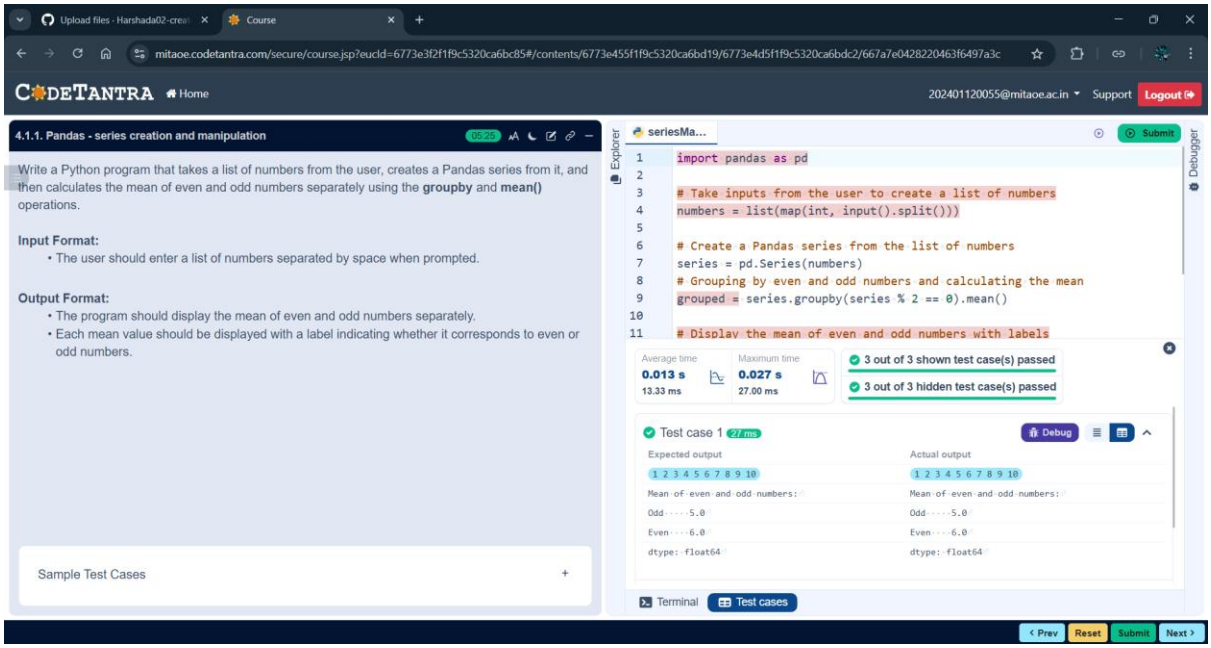

Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67861553a50fe56b8dcf9a6e
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
4.2.3. City that Sold the Most Products
04:39
Write 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
Explorer
monthFor...
sales_dat...
Submit
Debugger
# write the code..
city_sales = df.groupby("City")["Quantity"].sum()
# Find the city that sold the most products
best_city = city_sales.idxmax()
highest_quantity = city_sales.max()
# Display the result
print(f"City sold the most products: {best_city}")
Average time
0.025 s
25.33 ms
Maximum time
0.045 s
45.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
45 ms
Debug
Expected output
sales_data.csv
City sold the most products: Los Angeles
Actual output
sales_data.csv
City sold the most products: Los Angeles
Terminal
Test cases
Prev
Reset
Submit
Next


Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67861370a50fe56b8dcf9828
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
4.2.2. Best Selling Product
14:44
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
Explorer
monthFor...
sales_dat...
Submit
Debugger
# Load the data
df = pd.read_csv(file_name)
product_sales = df.groupby("Product")["Quantity"].sum()
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}")
Average time
0.022 s
22.00 ms
Maximum time
0.040 s
40.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
40 ms
Debug
Expected output
sales_data.csv
Best selling product: Product A
Total quantity sold: 23
Actual output
sales_data.csv
Best selling product: Product A
Total quantity sold: 23
Terminal
Test cases
Prev
Reset
Submit
Next


Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67860994a50fe56b8dcf8747
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
4.2.1. Month with the Highest Total Sales
03:49
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:
Sample Test Cases
Explorer
monthFor...
sales_dat...
Submit
Debugger
df["Date"] = pd.to_datetime(df["Date"])
df["Month"] = df["Date"].dt.to_period("M")
df["Total Sales"] = df["Quantity"] * df["Price"]
monthly_sales = df.groupby("Month")["Total Sales"].sum()
best_month = monthly_sales.idxmax()
Average time
0.031 s
30.67 ms
Maximum time
0.060 s
60.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
60 ms
Debug
Expected output
sales_data.csv
Best month: 2025-01
Total sales: $1210.00
Actual output
sales_data.csv
Best month: 2025-01
Total sales: $1210.00
Terminal
Test cases
Prev
Reset
Submit
Next


Upload files · Harshada02-creat
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667cf60f2365ba61af5a2ce7
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
4.1.3. Student Information
11:23
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
Explorer
Debugger
import pandas as pd
# Prompt the user for the file name
file_name = input()
# Load the data into a Pandas DataFrame
df = pd.read_csv(file_name, sep="\s+", names=["Name", "Age", "Grade"])
# Display the first five rows
print("First five rows:")
print(df.head())
Average time
0.046 s
46.50 ms
Maximum time
0.062 s
62.00 ms
1 out of 1 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1
62 ms
Debug
Expected output
studentdata.txt
First five rows:
Name Age Grade
0 John 25 A
1 Alin 22 B
2 Emma 24 A
Actual output
studentdata.txt
First five rows:
Name Age Grade
0 John 25 A
1 Alin 22 B
2 Emma 24 A
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›


Upload files · Harshada02-creat
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667aa30728220463f649bb62
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
4.1.2. Dictionary to dataframe
13:26
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:
Sample Test Cases
datafram...
Submit
Explorer
Debugger
# Adding a new row
new_name = input("New name: ")
new_age = int(input("New age: "))
df.loc[len(df)] = [new_name, new_age]
# Display the DataFrame after adding a new row
print("After adding a row:\n",df)
# Modifying a row
row_to_modify = int(input("Index of row to modify: "))
Average time
0.102 s
102.50 ms
Maximum time
0.114 s
114.00 ms
1 out of 1 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1
114 ms
Debug
Expected output
Original DataFrame:
Name Age
0 Alice 25
1 Bob 30
2 Charlie 35
New name: Susan
Actual output
Original DataFrame:
Name Age
0 Alice 25
1 Bob 30
2 Charlie 35
New name: Susan
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›


Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667a7e0428220463f6497a3c
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
4.1.1. Pandas - series creation and manipulation
05:25
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
Explorer
seriesMa...
Submit
Debugger
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
Average time
0.013 s
13.33 ms
Maximum time
0.027 s
27.00 ms
3 out of 3 shown test case(s) passed
3 out of 3 hidden test case(s) passed
Test case 1 27 ms
Debug
Expected output
1 2 3 4 5 6 7 8 9 10
Mean of even and odd numbers:
Odd 5.0
Even 6.0
dtype: float64
Actual output
1 2 3 4 5 6 7 8 9 10
Mean of even and odd numbers:
Odd 5.0
Even 6.0
dtype: float64
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›