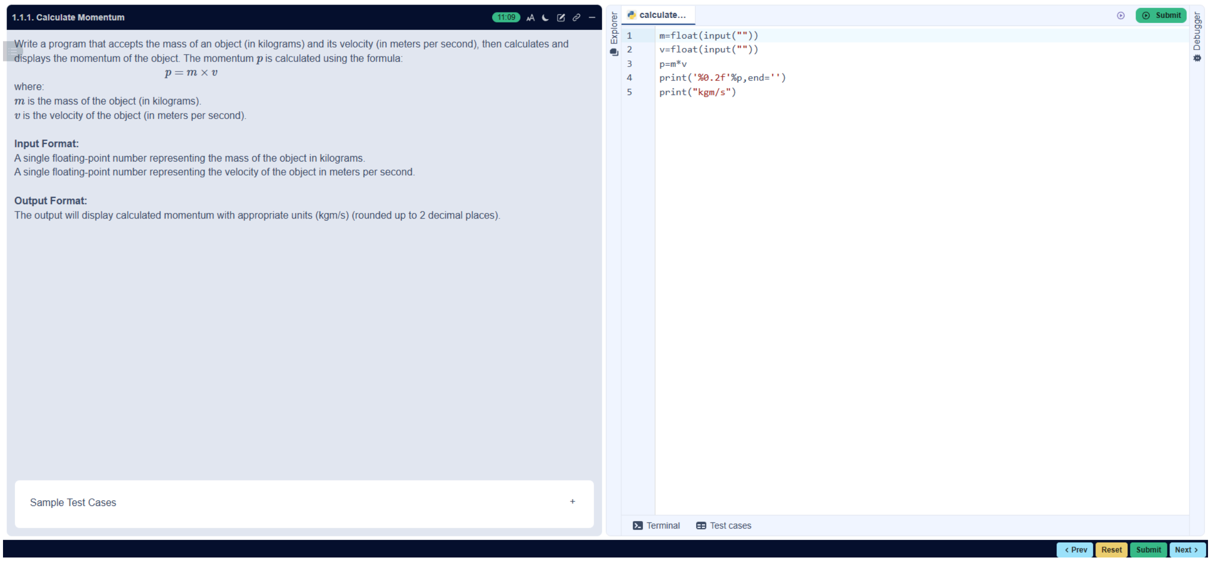
PYTHON PROGRAMS!

NAME : Shravani More

DIV: ME

ROLL NO: ME20

PRN NO: 202401090051



Write a program that accepts the mass of an object (in kilograms) and its velocity (in meters per second), then calculates and displays the momentum of the object. The momentum p is calculated using the formula:

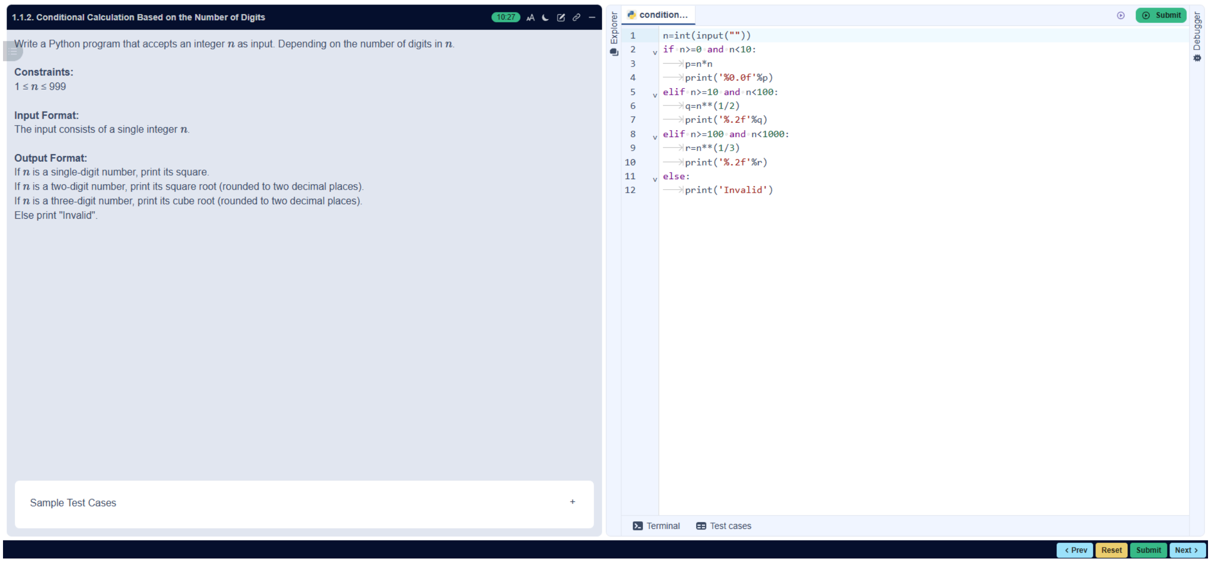
$$p = m \times v$$

where:
m is the mass of the object (in kilograms).
v is the velocity of the object (in meters per second).

Input Format:
A single floating-point number representing the mass of the object in kilograms.
A single floating-point number representing the velocity of the object in meters per second.

Output Format:
The output will display calculated momentum with appropriate units (kgm/s) (rounded up to 2 decimal places).

```python
m=float(input(""))
v=float(input(""))
p=m*v
print('%0.2f'%p,end='')
print("kgm/s")
```



1.1.2. Conditional Calculation Based on the Number of Digits

Write a Python program that accepts an integer n as input. Depending on the number of digits in n.

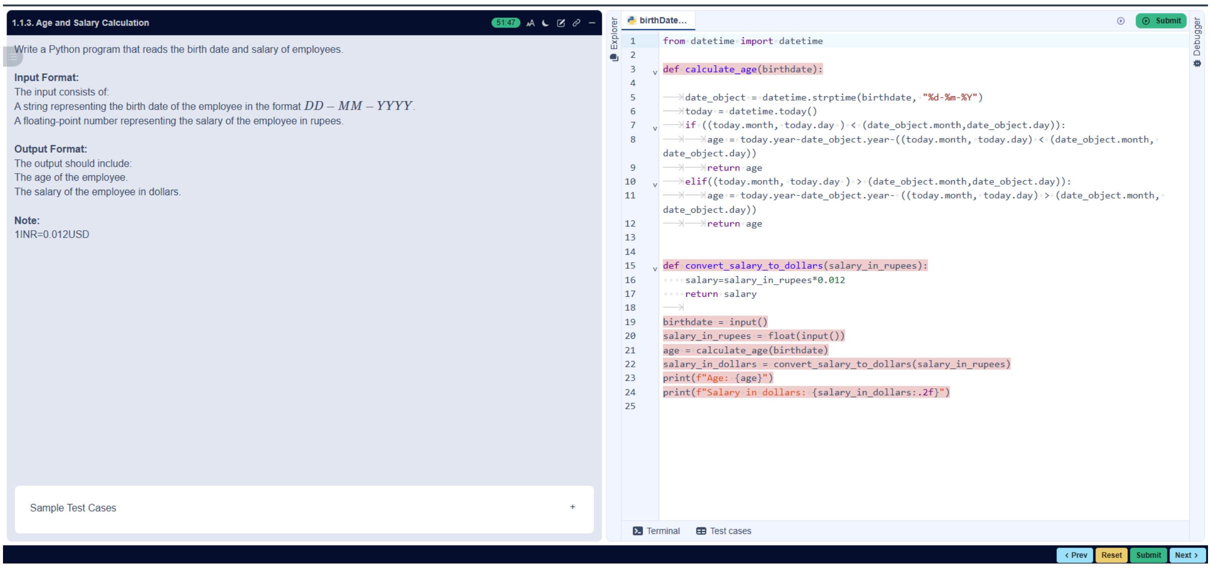
Constraints:
$1 \le n \le 999$

Input Format:
The input consists of a single integer n.

Output Format:
If n is a single-digit number, print its square.
If n is a two-digit number, print its square root (rounded to two decimal places).
If n is a three-digit number, print its cube root (rounded to two decimal places).
Else print "Invalid".

Sample Test Cases +

```python
n=int(input(""))
if n>=0 and n<10:
    p=n*n
    print('%0.0f'%p)
elif n>=10 and n<100:
    q=n**(1/2)
    print('%.2f'%q)
elif n>=100 and n<1000:
    r=n**(1/3)
    print('%.2f'%r)
else:
    print('Invalid')
```



1.1.3. Age and Salary Calculation

Write a Python program that reads the birth date and salary of employees.

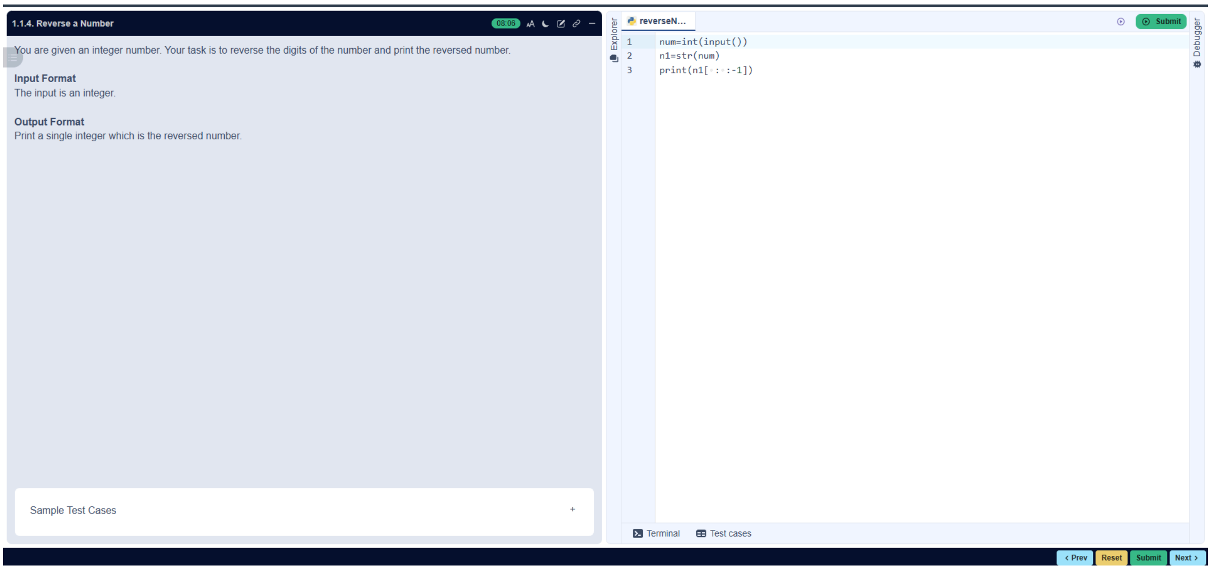
Input Format:
The input consists of:
A string representing the birth date of the employee in the format $DD-MM-YYYY$.
A floating-point number representing the salary of the employee in rupees.

Output Format:
The output should include:
The age of the employee.
The salary of the employee in dollars.

Note:
1INR=0.012USD

Sample Test Cases +

```python
from datetime import datetime

def calculate_age(birthdate):

    date_object = datetime.strptime(birthdate, "%d-%m-%Y")
    today = datetime.today()
    if ((today.month, today.day ) < (date_object.month,date_object.day)):
        age = today.year-date_object.year-((today.month, today.day) < (date_object.month, date_object.day))
        return age
    elif((today.month, today.day ) > (date_object.month,date_object.day)):
        age = today.year-date_object.year- ((today.month, today.day) > (date_object.month, date_object.day))
        return age


def convert_salary_to_dollars(salary_in_rupees):
    salary=salary_in_rupees*0.012
    return salary
    
birthdate = input()
salary_in_rupees = float(input())
age = calculate_age(birthdate)
salary_in_dollars = convert_salary_to_dollars(salary_in_rupees)
print(f"Age: {age}")
print(f"Salary in dollars: {salary_in_dollars:.2f}")
```



1.1.4. Reverse a Number

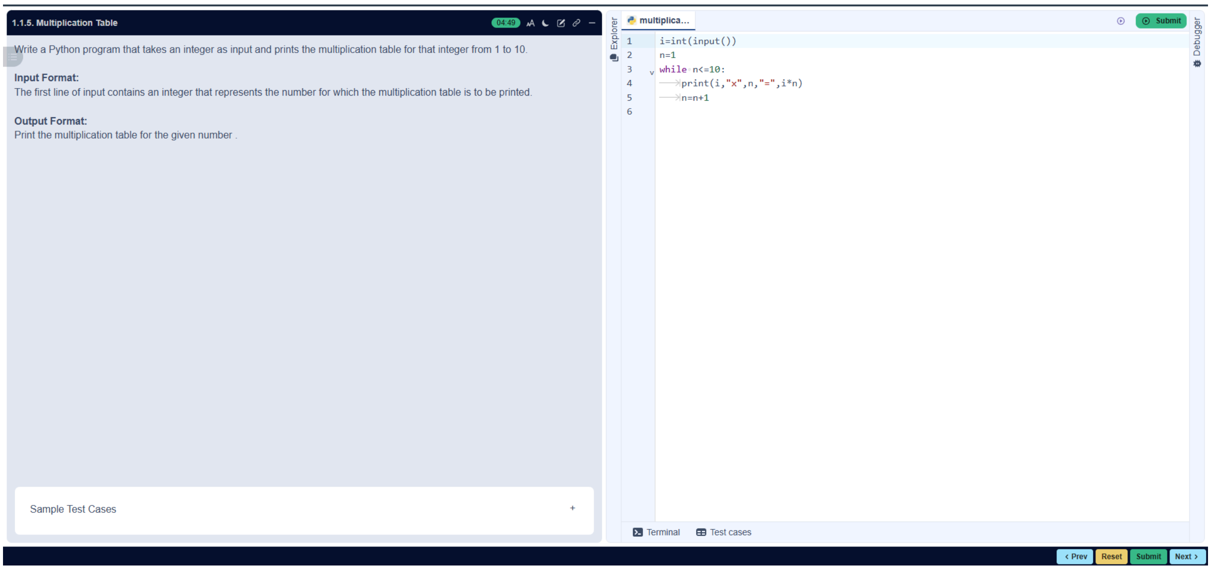
You are given an integer number. Your task is to reverse the digits of the number and print the reversed number.

Input Format
The input is an integer.

Output Format
Print a single integer which is the reversed number.

Sample Test Cases +

```python
num=int(input())
n1=str(num)
print(n1[ : :-1])
```



1.1.5. Multiplication Table

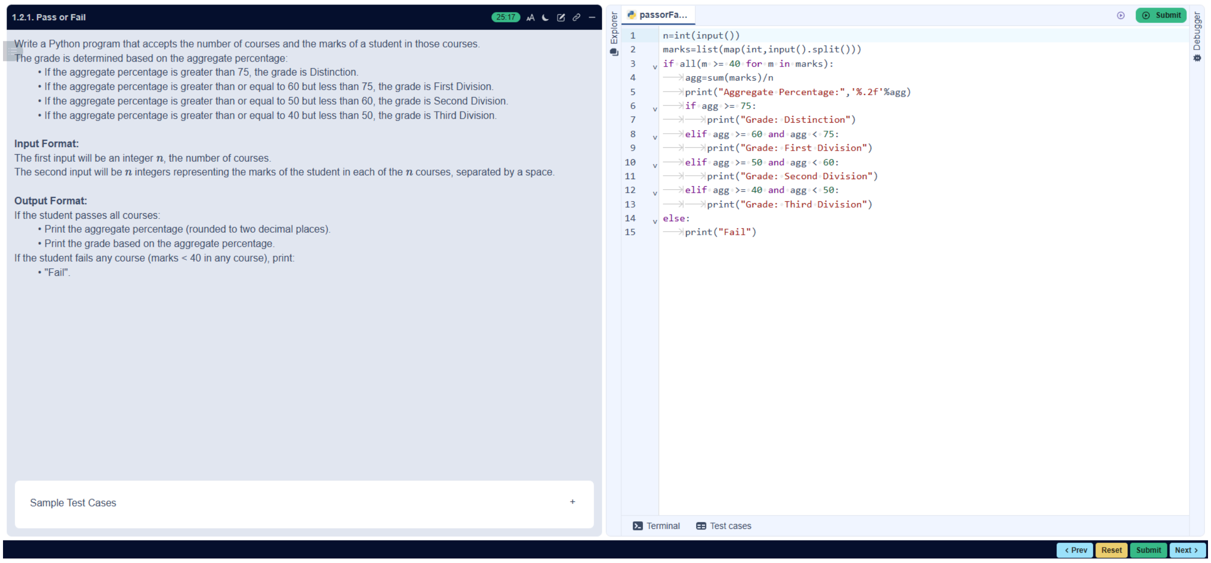
Write a Python program that takes an integer as input and prints the multiplication table for that integer from 1 to 10.

Input Format:
The first line of input contains an integer that represents the number for which the multiplication table is to be printed.

Output Format:
Print the multiplication table for the given number .

multiplica...

```python
i=int(input())
n=1
while n<=10:
	print(i,"x",n,"=",i*n)
	n=n+1

```



1.2.1. Pass or Fail

Write a Python program that accepts the number of courses and the marks of a student in those courses.
The grade is determined based on the aggregate percentage:

- If the aggregate percentage is greater than 75, the grade is Distinction.
- If the aggregate percentage is greater than or equal to 60 but less than 75, the grade is First Division.
- If the aggregate percentage is greater than or equal to 50 but less than 60, the grade is Second Division.
- If the aggregate percentage is greater than or equal to 40 but less than 50, the grade is Third Division.

Input Format:
The first input will be an integer n, the number of courses.
The second input will be n integers representing the marks of the student in each of the n courses, separated by a space.

Output Format:
If the student passes all courses:

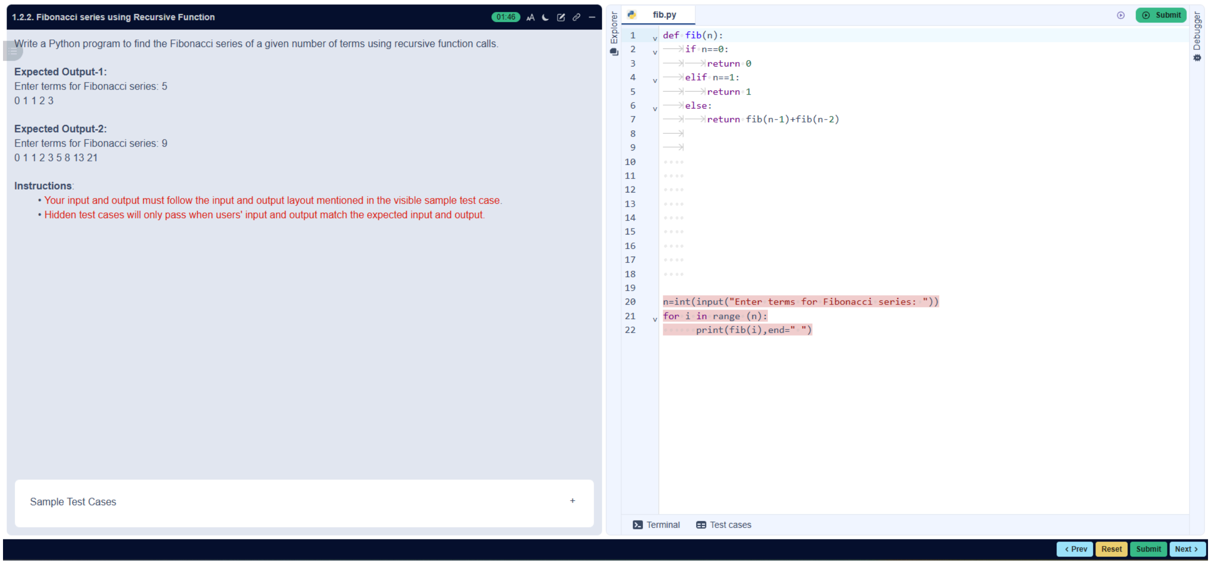
- Print the aggregate percentage (rounded to two decimal places).
- Print the grade based on the aggregate percentage.

If the student fails any course (marks < 40 in any course), print:

- "Fail".

passorFa...

```python
n=int(input())
marks=list(map(int,input().split()))
if all(m >= 40 for m in marks):
	agg=sum(marks)/n
	print("Aggregate Percentage:",'%.2f'%agg)
	if agg >= 75:
		print("Grade: Distinction")
	elif agg >= 60 and agg < 75:
		print("Grade: First Division")
	elif agg >= 50 and agg < 60:
		print("Grade: Second Division")
	elif agg >= 40 and agg < 50:
		print("Grade: Third Division")
else:
	print("Fail")
```



1.2.2. Fibonacci series using Recursive Function

Write a Python program to find the Fibonacci series of a given number of terms using recursive function calls.

Expected Output-1:
Enter terms for Fibonacci series: 5
0 1 1 2 3

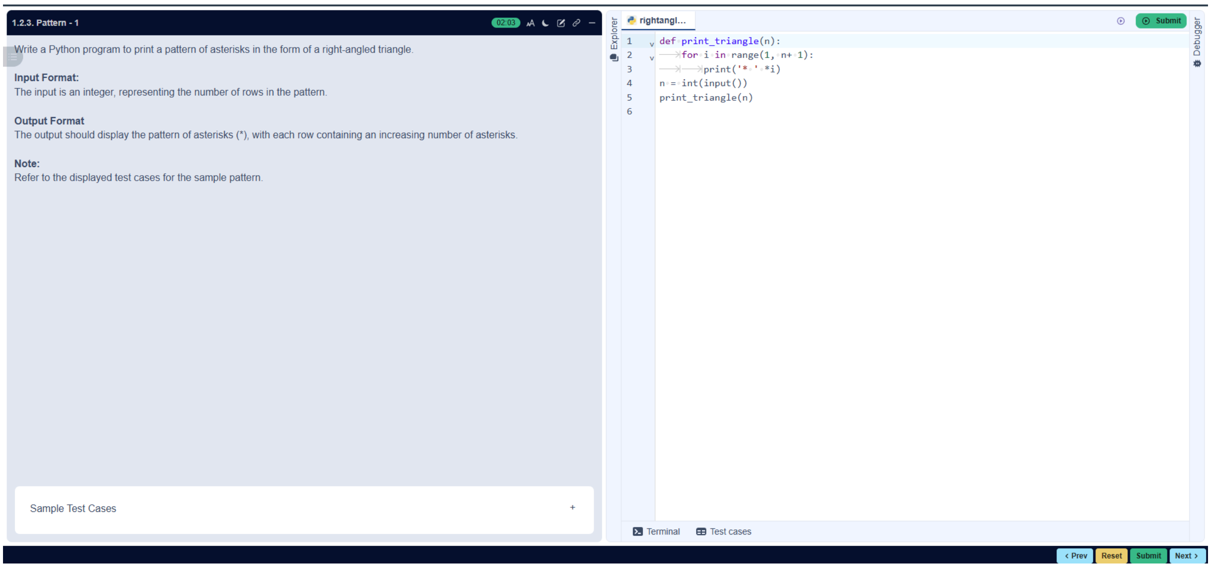
Expected Output-2:
Enter terms for Fibonacci series: 9
0 1 1 2 3 5 8 13 21

Instructions:

- Your input and output must follow the input and output layout mentioned in the visible sample test case.
- Hidden test cases will only pass when users' input and output match the expected input and output.

fib.py

```python
def fib(n):
	if n==0:
		return 0
	elif n==1:
		return 1
	else:
		return fib(n-1)+fib(n-2)
	
	











n=int(input("Enter terms for Fibonacci series: "))
for i in range (n):
      print(fib(i),end=" ")
```



Write a Python program to print a pattern of asterisks in the form of a right-angled triangle.

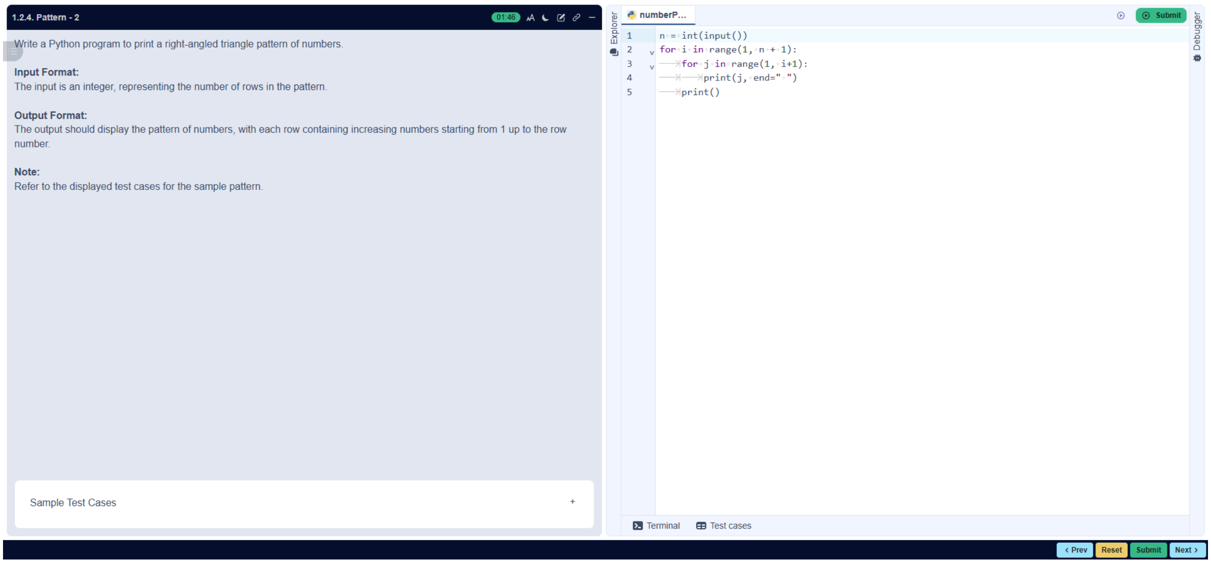
Input Format:
The input is an integer, representing the number of rows in the pattern.

Output Format
The output should display the pattern of asterisks (*), with each row containing an increasing number of asterisks.

Note:
Refer to the displayed test cases for the sample pattern.

Sample Test Cases

```
def print_triangle(n):
    for i in range(1, n+ 1):
        print('* '*i)
n = int(input())
print_triangle(n)
```



Write a Python program to print a right-angled triangle pattern of numbers.

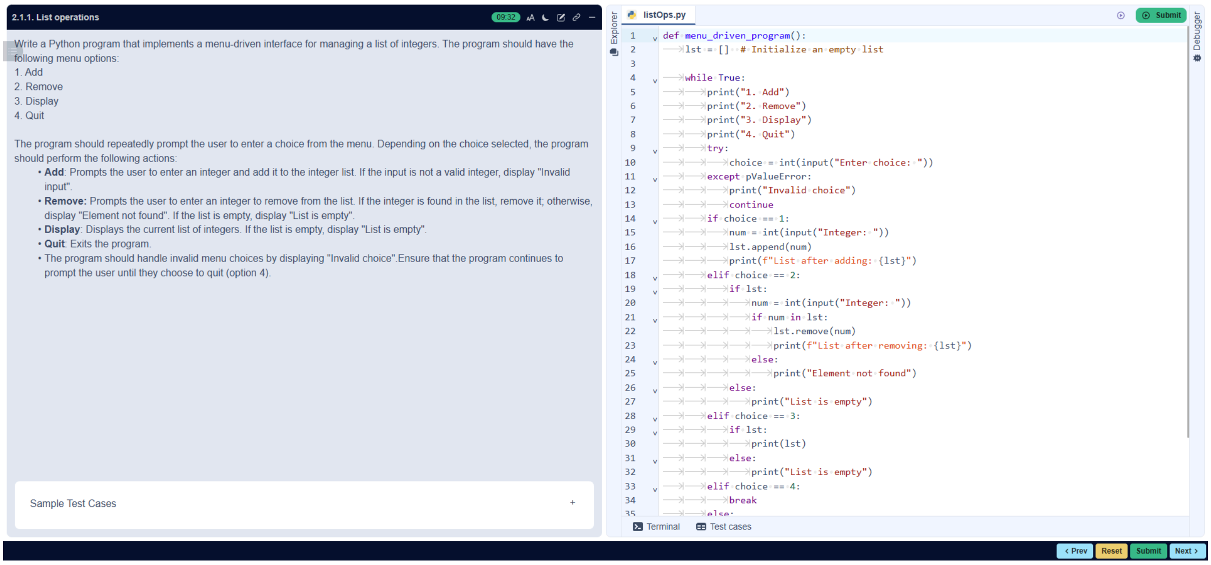
Input Format:
The input is an integer, representing the number of rows in the pattern.

Output Format:
The output should display the pattern of numbers, with each row containing increasing numbers starting from 1 up to the row number.

Note:
Refer to the displayed test cases for the sample pattern.

Sample Test Cases

```
n = int(input())
for i in range(1, n + 1):
    for j in range(1, i+1):
        print(j, end=" ")
    print()
```



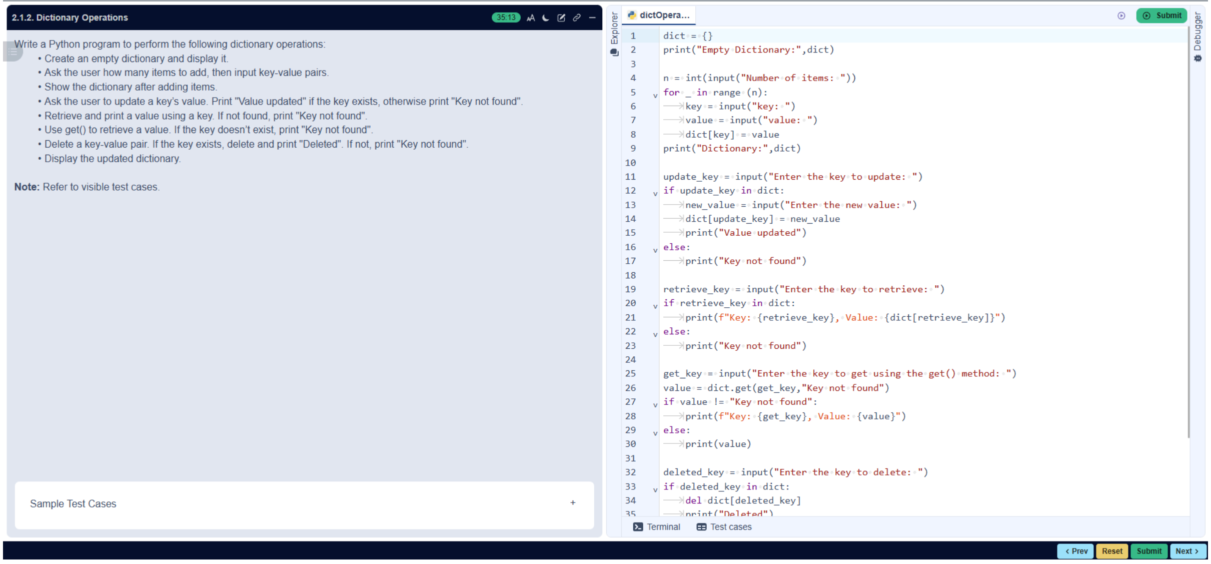
Write a Python program that implements a menu-driven interface for managing a list of integers. The program should have the following menu options:
1. Add
2. Remove
3. Display
4. Quit

The program should repeatedly prompt the user to enter a choice from the menu. Depending on the choice selected, the program should perform the following actions:

- **Add**: Prompts the user to enter an integer and add it to the integer list. If the input is not a valid integer, display "Invalid input".
- **Remove:** Prompts the user to enter an integer to remove from the list. If the integer is found in the list, remove it; otherwise, display "Element not found". If the list is empty, display "List is empty".
- **Display**: Displays the current list of integers. If the list is empty, display "List is empty".
- **Quit**: Exits the program.
- The program should handle invalid menu choices by displaying "Invalid choice".Ensure that the program continues to prompt the user until they choose to quit (option 4).

Sample Test Cases

```
def menu_driven_program():
    lst = []  # Initialize an empty list

    while True:
        print("1. Add")
        print("2. Remove")
        print("3. Display")
        print("4. Quit")
        try:
            choice = int(input("Enter choice: "))
        except pValueError:
            print("Invalid choice")
            continue
        if choice == 1:
            num = int(input("Integer: "))
            lst.append(num)
            print(f"List after adding: {lst}")
        elif choice == 2:
            if lst:
                num = int(input("Integer: "))
                if num in lst:
                    lst.remove(num)
                    print(f"List after removing: {lst}")
                else:
                    print("Element not found")
            else:
                print("List is empty")
        elif choice == 3:
            if lst:
                print(lst)
            else:
                print("List is empty")
        elif choice == 4:
            break
        else:
```



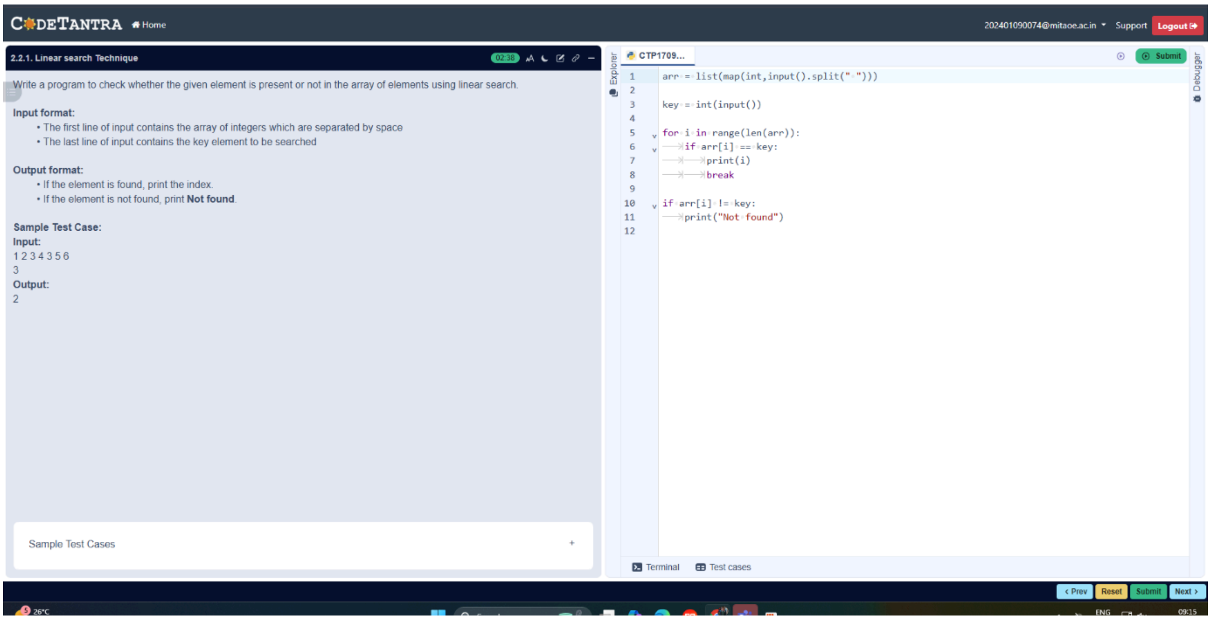
2.1.2. Dictionary Operations

Write a Python program to perform the following dictionary operations:

- Create an empty dictionary and display it.
- Ask the user how many items to add, then input key-value pairs.
- Show the dictionary after adding items.
- Ask the user to update a key's value. Print "Value updated" if the key exists, otherwise print "Key not found".
- Retrieve and print a value using a key. If not found, print "Key not found".
- Use get() to retrieve a value. If the key doesn't exist, print "Key not found".
- Delete a key-value pair. If the key exists, delete and print "Deleted". If not, print "Key not found".
- Display the updated dictionary.

Note: Refer to visible test cases.

```python
dict = {}
print("Empty Dictionary:",dict)

n = int(input("Number of items: "))
for _ in range (n):
    key = input("key: ")
    value = input("value: ")
    dict[key] = value
print("Dictionary:",dict)

update_key = input("Enter the key to update: ")
if update_key in dict:
    new_value = input("Enter the new value: ")
    dict[update_key] = new_value
    print("Value updated")
else:
    print("Key not found")

retrieve_key = input("Enter the key to retrieve: ")
if retrieve_key in dict:
    print(f"Key: {retrieve_key}, Value: {dict[retrieve_key]}")
else:
    print("Key not found")

get_key = input("Enter the key to get using the get() method: ")
value = dict.get(get_key,"Key not found")
if value != "Key not found":
    print(f"Key: {get_key}, Value: {value}")
else:
    print(value)

deleted_key = input("Enter the key to delete: ")
if deleted_key in dict:
    del dict[deleted_key]
    print("Deleted")
```



2.2.1. Linear search Technique

Write a program to check whether the given element is present or not in the array of elements using linear search.

Input format:

- The first line of input contains the array of integers which are separated by space
- The last line of input contains the key element to be searched

Output format:

- If the element is found, print the index.
- If the element is not found, print **Not found**.

Sample Test Case:

Input:

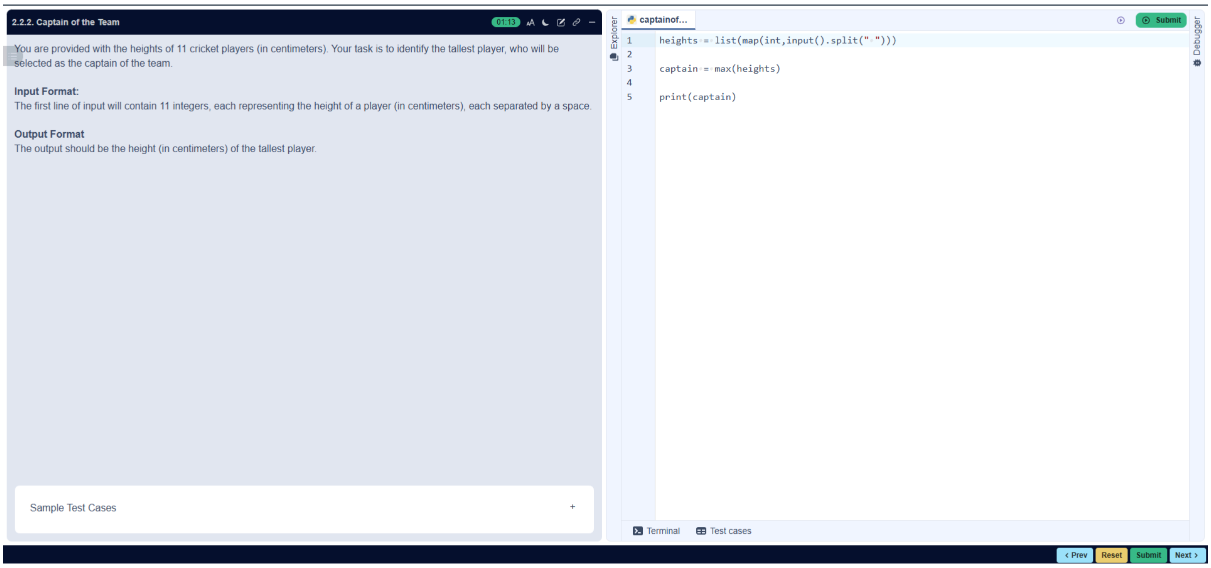
1 2 3 4 3 5 6

3

Output:

2

```python
arr = list(map(int,input().split(" ")))

key = int(input())

for i in range(len(arr)):
    if arr[i] == key:
        print(i)
        break

if arr[i] != key:
    print("Not found")

```



2.2.2. Captain of the Team

You are provided with the heights of 11 cricket players (in centimeters). Your task is to identify the tallest player, who will be selected as the captain of the team.

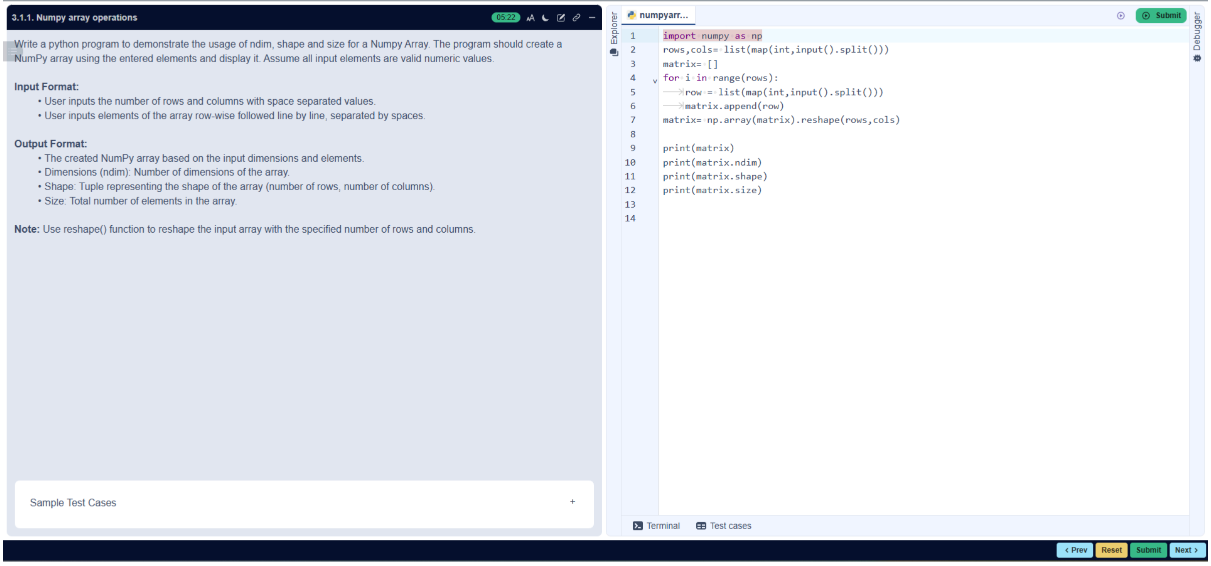
Input Format:

The first line of input will contain 11 integers, each representing the height of a player (in centimeters), each separated by a space.

Output Format

The output should be the height (in centimeters) of the tallest player.

```python
heights = list(map(int,input().split(" ")))

captain = max(heights)

print(captain)
```



3.1.1. Numpy array operations

Write a python program to demonstrate the usage of ndim, shape and size for a Numpy Array. The program should create a NumPy array using the entered elements and display it. Assume all input elements are valid numeric values.

Input Format:

- User inputs the number of rows and columns with space separated values.
- User inputs elements of the array row-wise followed line by line, separated by spaces.

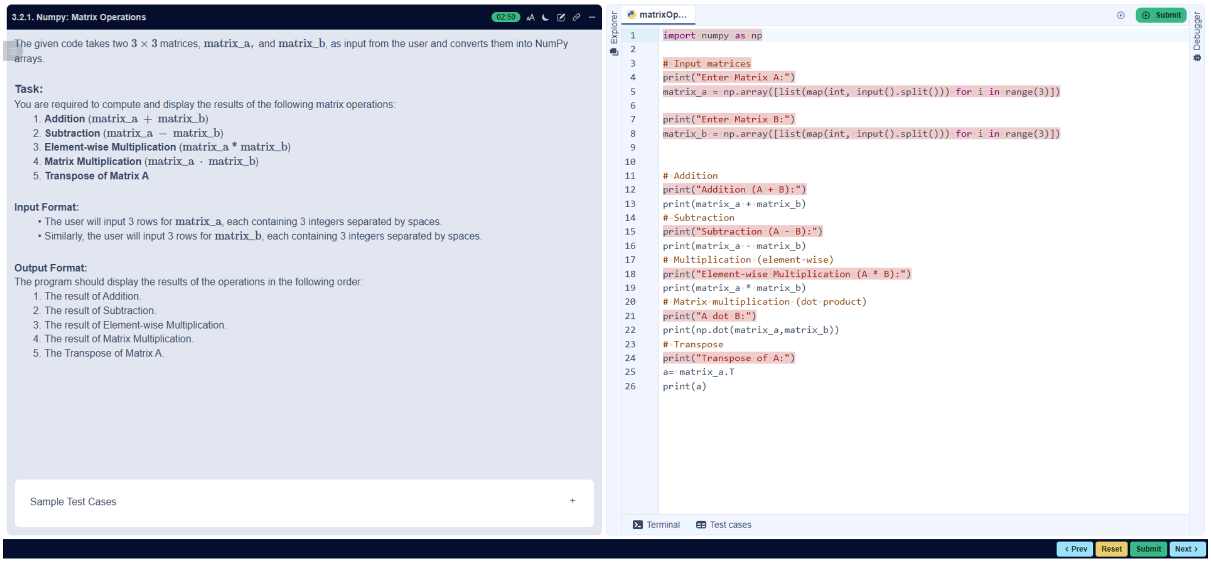
Output Format:

- The created NumPy array based on the input dimensions and elements.
- Dimensions (ndim): Number of dimensions of the array.
- Shape: Tuple representing the shape of the array (number of rows, number of columns).
- Size: Total number of elements in the array.

Note: Use reshape() function to reshape the input array with the specified number of rows and columns.

Sample Test Cases +

```python
import numpy as np
rows,cols= list(map(int,input().split()))
matrix= []
for i in range(rows):
    row = list(map(int,input().split()))
    matrix.append(row)
matrix= np.array(matrix).reshape(rows,cols)

print(matrix)
print(matrix.ndim)
print(matrix.shape)
print(matrix.size)
```



3.2.1. Numpy: Matrix Operations

The given code takes two 3×3 matrices, $\mathrm{matrix\_a},$ and $\mathrm{matrix\_b}$, as input from the user and converts them into NumPy arrays.

Task:

You are required to compute and display the results of the following matrix operations:

1. **Addition** $(\mathrm{matrix\_a} + \mathrm{matrix\_b})$
2. **Subtraction** $(\mathrm{matrix\_a} - \mathrm{matrix\_b})$
3. **Element-wise Multiplication** $(\mathrm{matrix\_a} * \mathrm{matrix\_b})$
4. **Matrix Multiplication** $(\mathrm{matrix\_a} \cdot \mathrm{matrix\_b})$
5. **Transpose of Matrix A**

Input Format:

- The user will input 3 rows for $\mathrm{matrix\_a}$, each containing 3 integers separated by spaces.
- Similarly, the user will input 3 rows for $\mathrm{matrix\_b}$, each containing 3 integers separated by spaces.

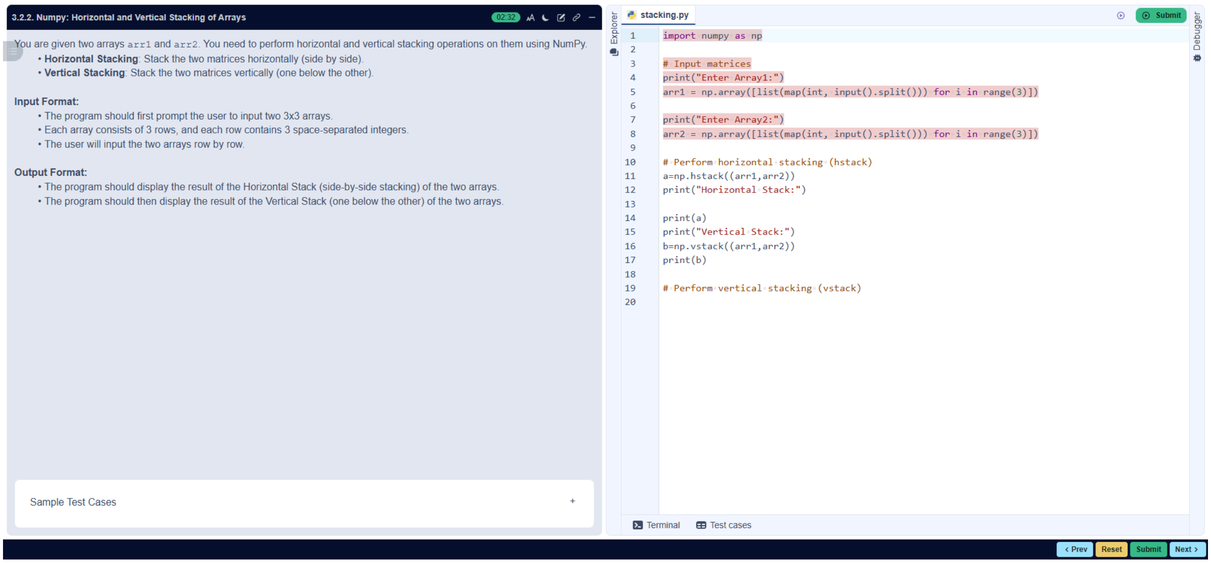
Output Format:

The program should display the results of the operations in the following order:

1. The result of Addition.
2. The result of Subtraction.
3. The result of Element-wise Multiplication.
4. The result of Matrix Multiplication.
5. The Transpose of Matrix A.

Sample Test Cases +

```python
import numpy as np

# Input matrices
print("Enter Matrix A:")
matrix_a = np.array([list(map(int, input().split())) for i in range(3)])

print("Enter Matrix B:")
matrix_b = np.array([list(map(int, input().split())) for i in range(3)])


# Addition
print("Addition (A + B):")
print(matrix_a + matrix_b)
# Subtraction
print("Subtraction (A - B):")
print(matrix_a - matrix_b)
# Multiplication (element-wise)
print("Element-wise Multiplication (A * B):")
print(matrix_a * matrix_b)
# Matrix multiplication (dot product)
print("A dot B:")
print(np.dot(matrix_a,matrix_b))
# Transpose
print("Transpose of A:")
a= matrix_a.T
print(a)
```



3.2.2. Numpy: Horizontal and Vertical Stacking of Arrays

You are given two arrays `arr1` and `arr2`. You need to perform horizontal and vertical stacking operations on them using NumPy.

- **Horizontal Stacking**: Stack the two matrices horizontally (side by side).
- **Vertical Stacking**: Stack the two matrices vertically (one below the other).

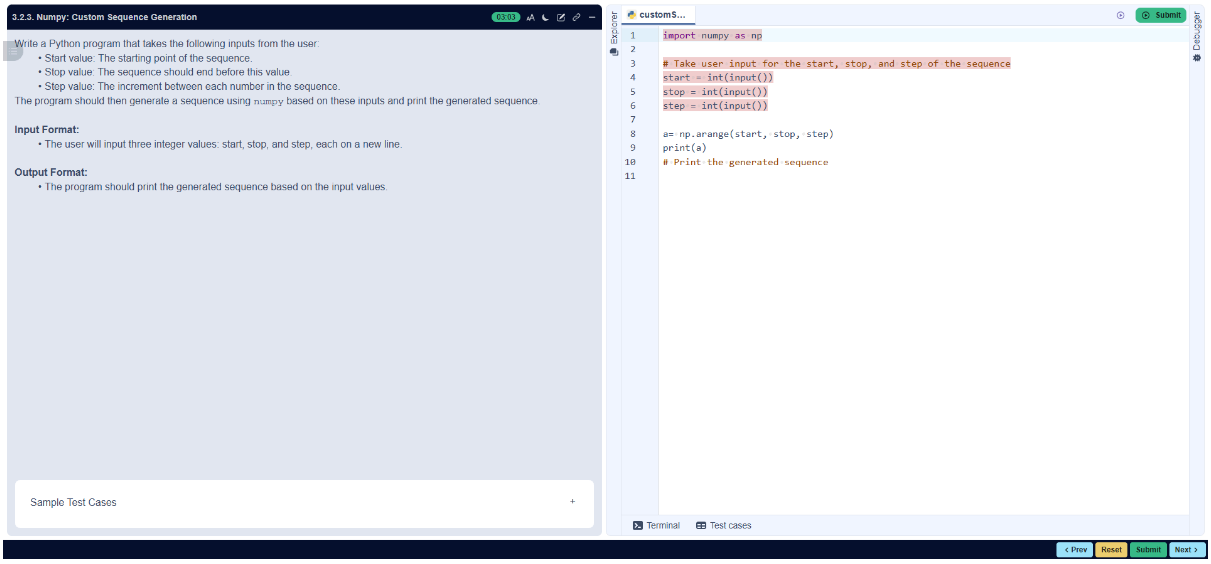
Input Format:

- The program should first prompt the user to input two 3x3 arrays.
- Each array consists of 3 rows, and each row contains 3 space-separated integers.
- The user will input the two arrays row by row.

Output Format:

- The program should display the result of the Horizontal Stack (side-by-side stacking) of the two arrays.
- The program should then display the result of the Vertical Stack (one below the other) of the two arrays.

```python
import numpy as np

# Input matrices
print("Enter Array1:")
arr1 = np.array([list(map(int, input().split())) for i in range(3)])

print("Enter Array2:")
arr2 = np.array([list(map(int, input().split())) for i in range(3)])

# Perform horizontal stacking (hstack)
a=np.hstack((arr1,arr2))
print("Horizontal Stack:")

print(a)
print("Vertical Stack:")
b=np.vstack((arr1,arr2))
print(b)

# Perform vertical stacking (vstack)

```



3.2.3. Numpy: Custom Sequence Generation

Write a Python program that takes the following inputs from the user:

- Start value: The starting point of the sequence.
- Stop value: The sequence should end before this value.
- Step value: The increment between each number in the sequence.

The program should then generate a sequence using `numpy` based on these inputs and print the generated sequence.

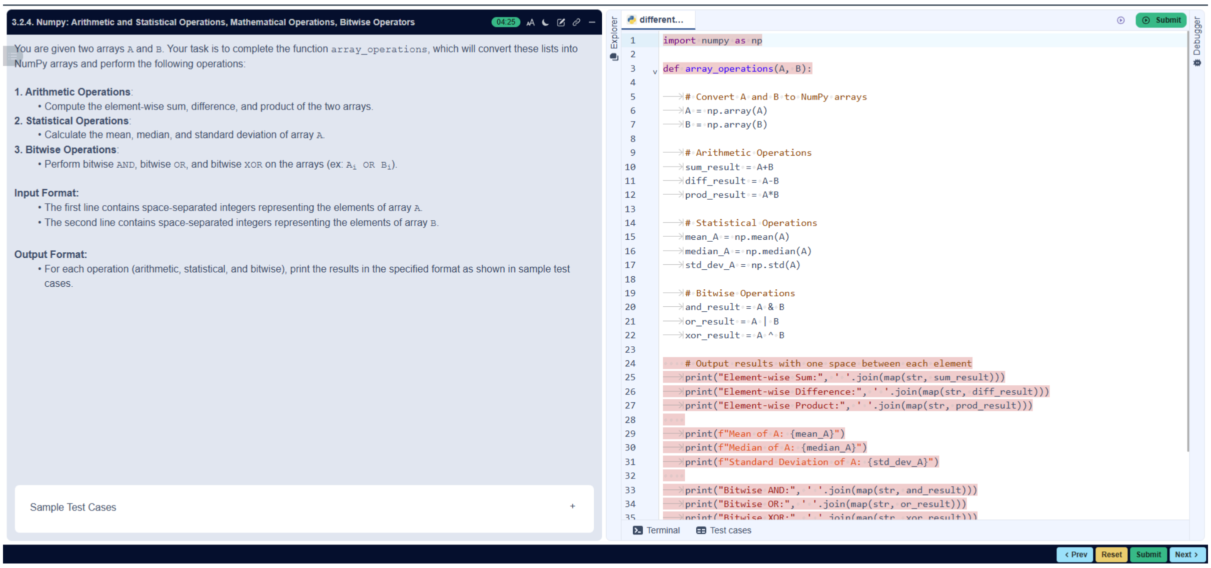
Input Format:

- The user will input three integer values: start, stop, and step, each on a new line.

Output Format:

- The program should print the generated sequence based on the input values.

```python
import numpy as np

# Take user input for the start, stop, and step of the sequence
start = int(input())
stop = int(input())
step = int(input())

a= np.arange(start, stop, step)
print(a)
# Print the generated sequence

```



3.2.4. Numpy: Arithmetic and Statistical Operations, Mathematical Operations, Bitwise Operators

You are given two arrays `A` and `B`. Your task is to complete the function `array_operations`, which will convert these lists into NumPy arrays and perform the following operations:

1. Arithmetic Operations:

- Compute the element-wise sum, difference, and product of the two arrays.

2. Statistical Operations:

- Calculate the mean, median, and standard deviation of array `A`.

3. Bitwise Operations:

- Perform bitwise `AND`, bitwise `OR`, and bitwise `XOR` on the arrays (ex: A_i `OR` B_i).

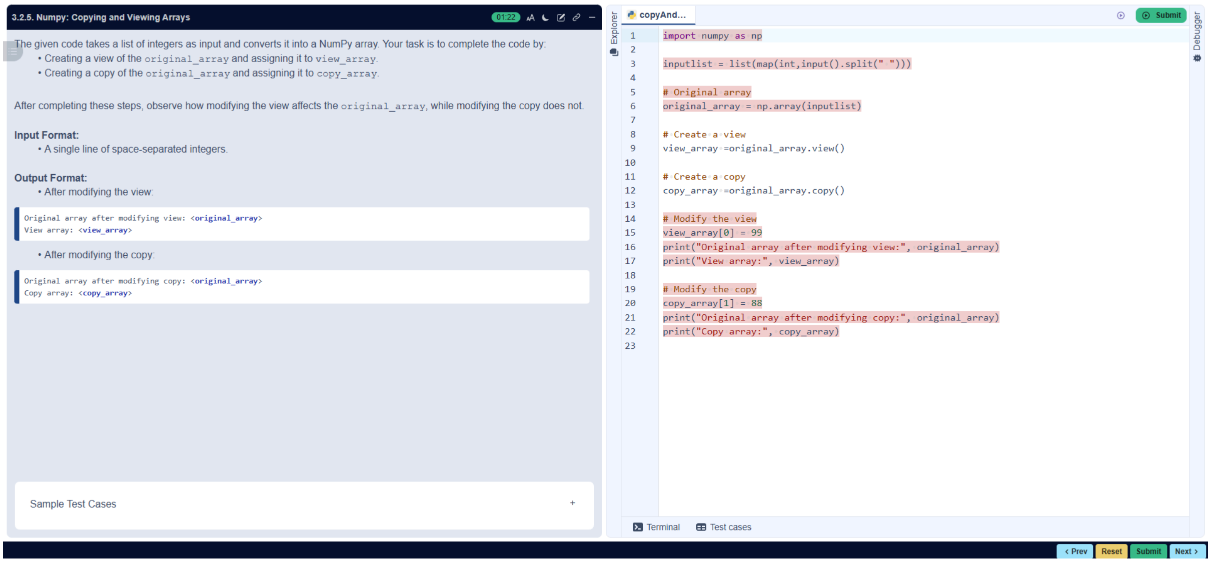
Input Format:

- The first line contains space-separated integers representing the elements of array `A`.
- The second line contains space-separated integers representing the elements of array `B`.

Output Format:

- For each operation (arithmetic, statistical, and bitwise), print the results in the specified format as shown in sample test cases.

```python
import numpy as np

def array_operations(A, B):

    # Convert A and B to NumPy arrays
    A = np.array(A)
    B = np.array(B)

    # Arithmetic Operations
    sum_result = A+B
    diff_result = A-B
    prod_result = A*B

    # Statistical Operations
    mean_A = np.mean(A)
    median_A = np.median(A)
    std_dev_A = np.std(A)

    # Bitwise Operations
    and_result = A & B
    or_result = A | B
    xor_result = A ^ B

    # Output results with one space between each element
    print("Element-wise Sum:", ' '.join(map(str, sum_result)))
    print("Element-wise Difference:", ' '.join(map(str, diff_result)))
    print("Element-wise Product:", ' '.join(map(str, prod_result)))
    
    print(f"Mean of A: {mean_A}")
    print(f"Median of A: {median_A}")
    print(f"Standard Deviation of A: {std_dev_A}")
    
    print("Bitwise AND:", ' '.join(map(str, and_result)))
    print("Bitwise OR:", ' '.join(map(str, or_result)))
    print("Bitwise XOR:", ' '.join(map(str, xor_result)))
```



3.2.5. Numpy: Copying and Viewing Arrays

The given code takes a list of integers as input and converts it into a NumPy array. Your task is to complete the code by:

- Creating a view of the `original_array` and assigning it to `view_array`.
- Creating a copy of the `original_array` and assigning it to `copy_array`.

After completing these steps, observe how modifying the view affects the `original_array`, while modifying the copy does not.

Input Format:

- A single line of space-separated integers.

Output Format:

- After modifying the view:

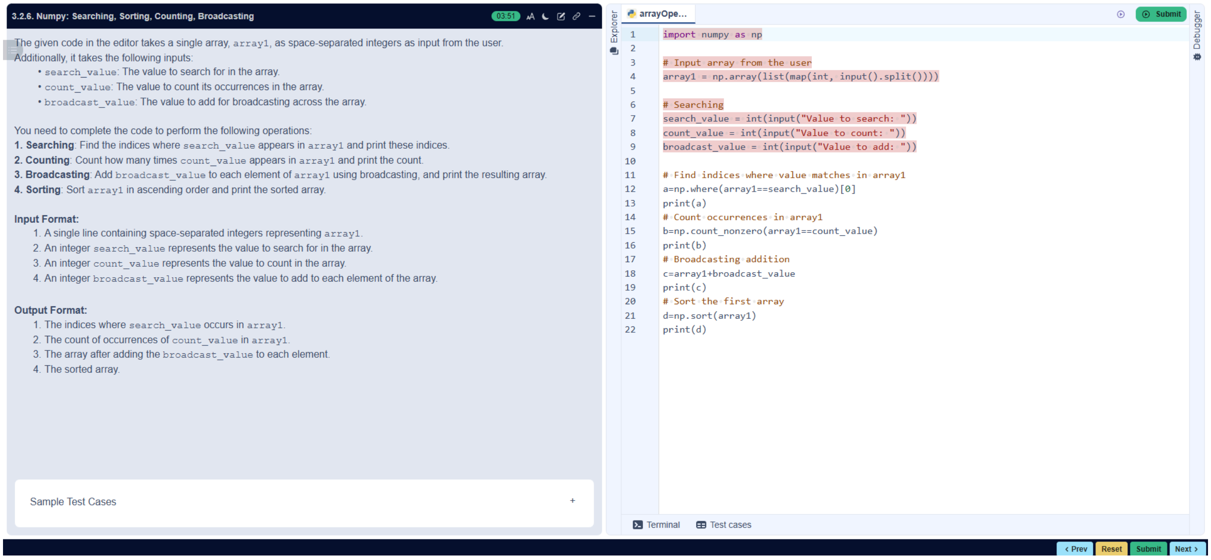
```
Original array after modifying view: <original_array>
View array: <view_array>
```

- After modifying the copy:

```
Original array after modifying copy: <original_array>
Copy array: <copy_array>
```

Sample Test Cases +

```python
import numpy as np

inputlist = list(map(int,input().split(" ")))

# Original array
original_array = np.array(inputlist)

# Create a view
view_array =original_array.view()

# Create a copy
copy_array =original_array.copy()

# Modify the view
view_array[0] = 99
print("Original array after modifying view:", original_array)
print("View array:", view_array)

# Modify the copy
copy_array[1] = 88
print("Original array after modifying copy:", original_array)
print("Copy array:", copy_array)

```



3.2.6. Numpy: Searching, Sorting, Counting, Broadcasting

The given code in the editor takes a single array, `array1`, as space-separated integers as input from the user. Additionally, it takes the following inputs:

- `search_value`: The value to search for in the array.
- `count_value`: The value to count its occurrences in the array.
- `broadcast_value`: The value to add for broadcasting across the array.

You need to complete the code to perform the following operations:

1. **Searching**: Find the indices where `search_value` appears in `array1` and print these indices.
2. **Counting**: Count how many times `count_value` appears in `array1` and print the count.
3. **Broadcasting**: Add `broadcast_value` to each element of `array1` using broadcasting, and print the resulting array.
4. **Sorting**: Sort `array1` in ascending order and print the sorted array.

Input Format:

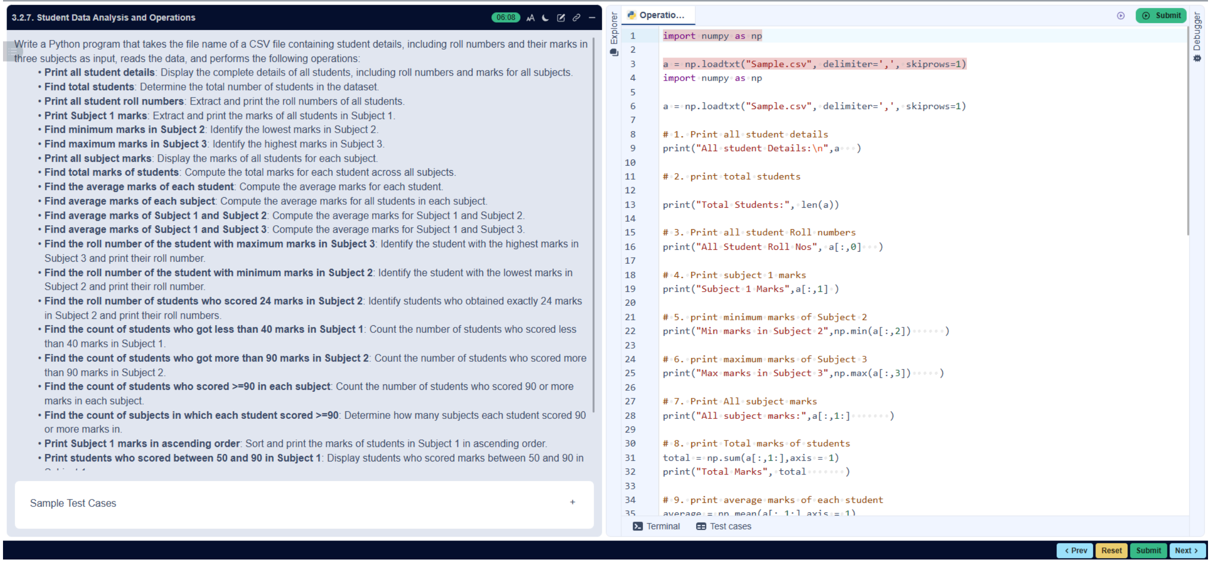
1. A single line containing space-separated integers representing `array1`.
2. An integer `search_value` represents the value to search for in the array.
3. An integer `count_value` represents the value to count in the array.
4. An integer `broadcast_value` represents the value to add to each element of the array.

Output Format:

1. The indices where `search_value` occurs in `array1`.
2. The count of occurrences of `count_value` in `array1`.
3. The array after adding the `broadcast_value` to each element.
4. The sorted array.

Sample Test Cases +

```python
import numpy as np

# Input array from the user
array1 = np.array(list(map(int, input().split())))

# Searching
search_value = int(input("Value to search: "))
count_value = int(input("Value to count: "))
broadcast_value = int(input("Value to add: "))

# Find indices where value matches in array1
a=np.where(array1==search_value)[0]
print(a)
# Count occurrences in array1
b=np.count_nonzero(array1==count_value)
print(b)
# Broadcasting addition
c=array1+broadcast_value
print(c)
# Sort the first array
d=np.sort(array1)
print(d)
```



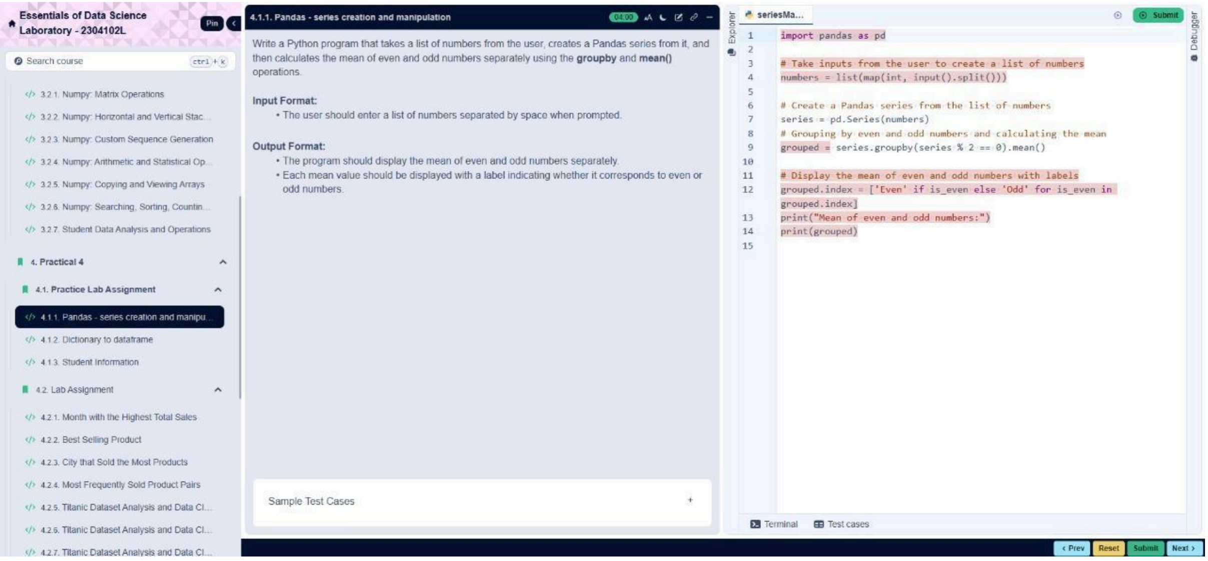
3.2.7. Student Data Analysis and Operations

Write a Python program that takes the file name of a CSV file containing student details, including roll numbers and their marks in three subjects as input, reads the data, and performs the following operations:

- **Print all student details**: Display the complete details of all students, including roll numbers and marks for all subjects.
- **Find total students**: Determine the total number of students in the dataset.
- **Print all student roll numbers**: Extract and print the roll numbers of all students.
- **Print Subject 1 marks**: Extract and print the marks of all students in Subject 1.
- **Find minimum marks in Subject 2**: Identify the lowest marks in Subject 2.
- **Find maximum marks in Subject 3**: Identify the highest marks in Subject 3.
- **Print all subject marks**: Display the marks of all students for each subject.
- **Find total marks of students**: Compute the total marks for each student across all subjects.
- **Find the average marks of each student**: Compute the average marks for each student.
- **Find average marks of each subject**: Compute the average marks for all students in each subject.
- **Find average marks of Subject 1 and Subject 2**: Compute the average marks for Subject 1 and Subject 2.
- **Find average marks of Subject 1 and Subject 3**: Compute the average marks for Subject 1 and Subject 3.
- **Find the roll number of the student with maximum marks in Subject 3**: Identify the student with the highest marks in Subject 3 and print their roll number.
- **Find the roll number of the student with minimum marks in Subject 2**: Identify the student with the lowest marks in Subject 2 and print their roll number.
- **Find the roll number of students who scored 24 marks in Subject 2**: Identify students who obtained exactly 24 marks in Subject 2 and print their roll numbers.
- **Find the count of students who got less than 40 marks in Subject 1**: Count the number of students who scored less than 40 marks in Subject 1.
- **Find the count of students who got more than 90 marks in Subject 2**: Count the number of students who scored more than 90 marks in Subject 2.
- **Find the count of students who scored >=90 in each subject**: Count the number of students who scored 90 or more marks in each subject.
- **Find the count of subjects in which each student scored >=90**: Determine how many subjects each student scored 90 or more marks in.
- **Print Subject 1 marks in ascending order**: Sort and print the marks of students in Subject 1 in ascending order.
- **Print students who scored between 50 and 90 in Subject 1**: Display students who scored marks between 50 and 90 in Subject 1.

Sample Test Cases +

```python
import numpy as np

a = np.loadtxt("Sample.csv", delimiter=',', skiprows=1)
import numpy as np

a = np.loadtxt("Sample.csv", delimiter=',', skiprows=1)

# 1. Print all student details
print("All student Details:\n",a   )

# 2. print total students

print("Total Students:", len(a))

# 3. Print all student Roll numbers
print("All Student Roll Nos", a[:,0]   )

# 4. Print subject 1 marks
print("Subject 1 Marks",a[:,1] )

# 5. print minimum marks of Subject 2
print("Min marks in Subject 2",np.min(a[:,2])      )

# 6. print maximum marks of Subject 3
print("Max marks in Subject 3",np.max(a[:,3])     )

# 7. Print All subject marks
print("All subject marks:",a[:,1:]       )

# 8. print Total marks of students
total = np.sum(a[:,1:],axis = 1)
print("Total Marks", total       )

# 9. print average marks of each student
average = np.mean(a[:,1:],axis = 1)
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
3.2.1. Numpy: Matrix Operations
3.2.2. Numpy: Horizontal and Vertical Stac...
3.2.3. Numpy: Custom Sequence Generation
3.2.4. Numpy: Arithmetic and Statistical Op...
3.2.5. Numpy: Copying and Viewing Arrays
3.2.6. Numpy: Searching, Sorting, Countin...
3.2.7. Student Data Analysis and Operations
4. Practical 4
4.1. Practice Lab Assignment
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
04:00
seriesMa...
Submit
Explorer
Debugger
Sample Test Cases
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

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.

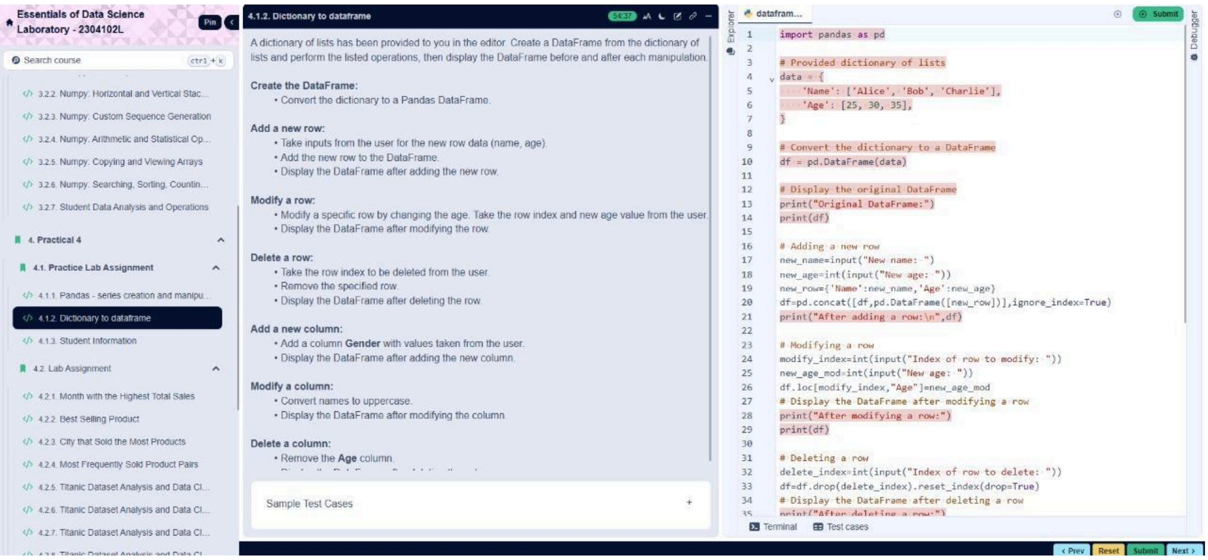
Input Format:

- 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.

```python
import pandas as pd

# Take inputs from the user to create a list of numbers
numbers = list(map(int, input().split()))

# Create a Pandas series from the list of numbers
series = pd.Series(numbers)
# Grouping by even and odd numbers and calculating the mean
grouped = series.groupby(series % 2 == 0).mean()

# Display the mean of even and odd numbers with labels
grouped.index = ['Even' if is_even else 'Odd' for is_even in grouped.index]
print("Mean of even and odd numbers:")
print(grouped)

```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
3.2.2. Numpy: Horizontal and Vertical Stac...
3.2.3. Numpy: Custom Sequence Generation
3.2.4. Numpy: Arithmetic and Statistical Op...
3.2.5. Numpy: Copying and Viewing Arrays
3.2.6. Numpy: Searching, Sorting, Countin...
3.2.7. Student Data Analysis and Operations
4. Practical 4
4.1. Practice Lab Assignment
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
54:37
datafram...
Submit
Explorer
Debugger
Sample Test Cases
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

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.

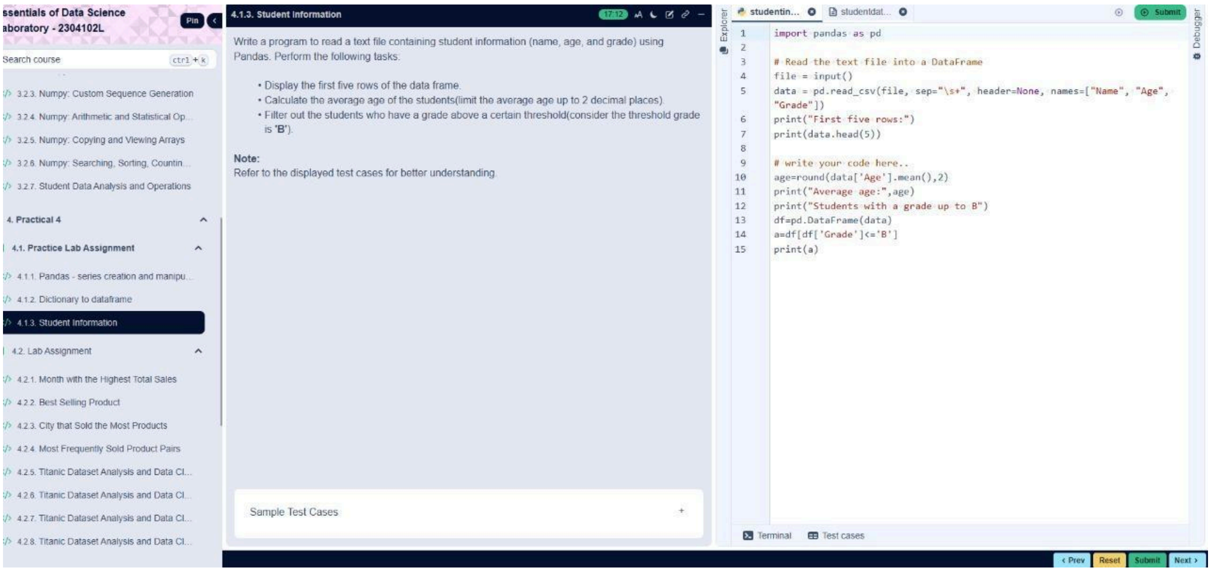
Modify a column:

- Convert names to uppercase.
- Display the DataFrame after modifying the column.

Delete a column:

- Remove the **Age** column.

```python
import pandas as pd

# Provided dictionary of lists
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
}

# Convert the dictionary to a DataFrame
df = pd.DataFrame(data)

# Display the original DataFrame
print("Original DataFrame:")
print(df)

# Adding a new row
new_name=input("New name: ")
new_age=int(input("New age: "))
new_row={'Name':new_name,'Age':new_age}
df=pd.concat([df,pd.DataFrame([new_row])],ignore_index=True)
print("After adding a row:\n",df)

# Modifying a row
modify_index=int(input("Index of row to modify: "))
new_age_mod=int(input("New age: "))
df.loc[modify_index,"Age"]=new_age_mod
# Display the DataFrame after modifying a row
print("After modifying a row:")
print(df)

# Deleting a row
delete_index=int(input("Index of row to delete: "))
df=df.drop(delete_index).reset_index(drop=True)
# Display the DataFrame after deleting a row
print("After deleting a row:")
```


ssentials of Data Science aboratory - 2304102L
Pin
Search course
ctrl + k
3.2.3. Numpy: Custom Sequence Generation
3.2.4. Numpy: Arithmetic and Statistical Op...
3.2.5. Numpy: Copying and Viewing Arrays
3.2.6. Numpy: Searching, Sorting, Countin...
3.2.7. Student Data Analysis and Operations
4. Practical 4
4.1. Practice Lab Assignment
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
17:12
studentin...
studentdat...
Submit
Explorer
Debugger
Sample Test Cases
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

4.1.3. Student Information

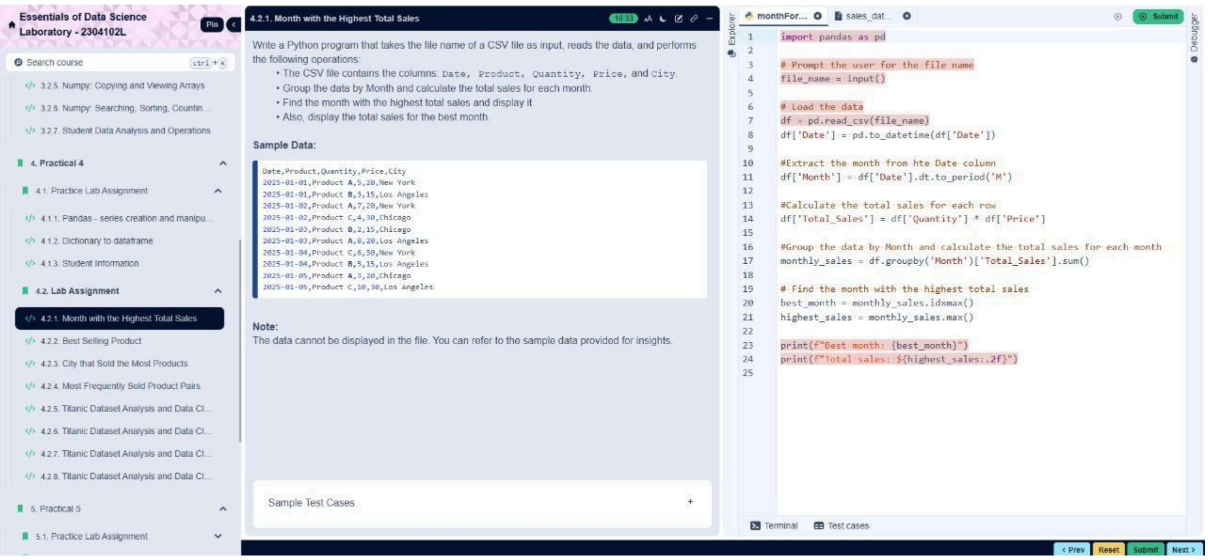
Write a program to read a text file containing student information (name, age, and grade) using Pandas. Perform the following tasks:

- 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.

```python
import pandas as pd

# Read the text file into a DataFrame
file = input()
data = pd.read_csv(file, sep="\s+", header=None, names=["Name", "Age", "Grade"])
print("First five rows:")
print(data.head(5))

# write your code here..
age=round(data['Age'].mean(),2)
print("Average age:",age)
print("Students with a grade up to B")
df=pd.DataFrame(data)
a=df[df['Grade']<='B']
print(a)
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
3.2.5. Numpy: Copying and Viewing Arrays
3.2.6. Numpy: Searching, Sorting, Countin...
3.2.7. Student Data Analysis and Operations
4. Practical 4
4.1. Practice Lab Assignment
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
10:33
monthFor...
sales_dat...
Submit
Explorer
Debugger
Sample Test Cases
Terminal
Test cases
< Prev
Reset
Submit
Next >

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:

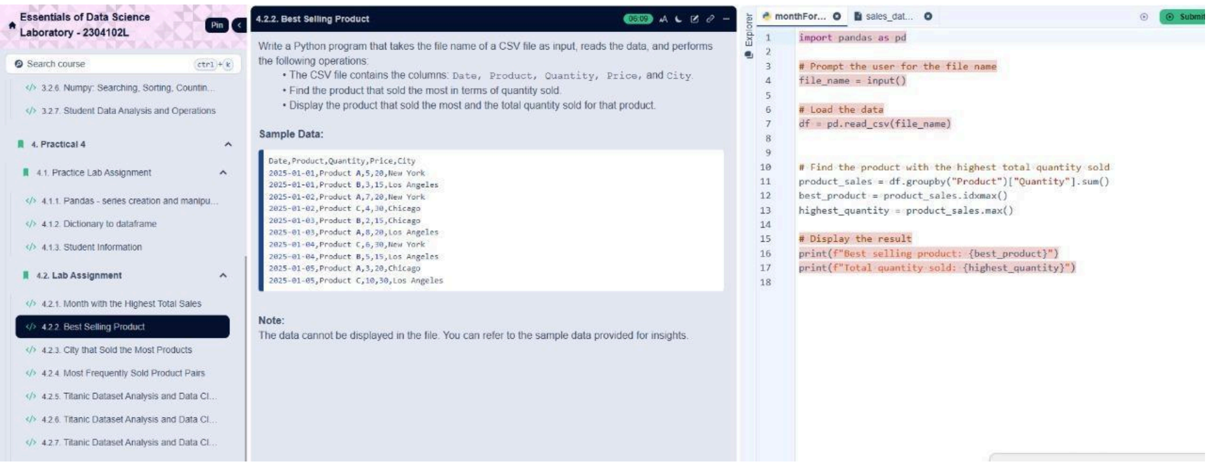
- The CSV file contains the columns: `Date, Product, Quantity, Price,` and `City`.
- Group the data by Month and calculate the total sales for each month.
- Find the month with the highest total sales and display it.
- Also, display the total sales for the best month.

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.

```python
import pandas as pd

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

# Load the data
df = pd.read_csv(file_name)
df['Date'] = pd.to_datetime(df['Date'])

#Extract the month from hte Date column
df['Month'] = df['Date'].dt.to_period('M')

#Calculate the total sales for each row
df['Total_Sales'] = df['Quantity'] * df['Price']

#Group the data by Month and calculate the total sales for each month
monthly_sales = df.groupby('Month')['Total_Sales'].sum()

# Find the month with the highest total sales
best_month = monthly_sales.idxmax()
highest_sales = monthly_sales.max()

print(f"Best month: {best_month}")
print(f"Total sales: ${highest_sales:.2f}")
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
3.2.6. Numpy: Searching, Sorting, Countin...
3.2.7. Student Data Analysis and Operations
4. Practical 4
4.1. Practice Lab Assignment
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
06:09
monthFor...
sales_dat...
Submi
Explorer

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:

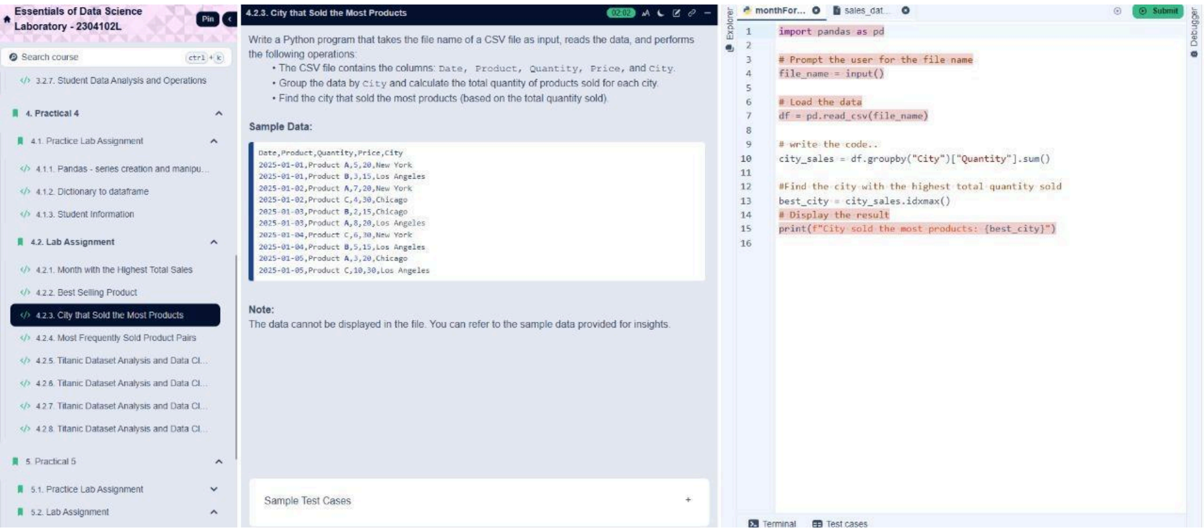
- The CSV file contains the columns: `Date, Product, Quantity, Price,` and `City`.
- Find the product that sold the most in terms of quantity sold.
- Display the product that sold the most and the total quantity sold for that product.

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.

```python
import pandas as pd

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

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


# Find the product with the highest total quantity sold
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}")
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
3.2.7. Student Data Analysis and Operations
4. Practical 4
4.1. Practice Lab Assignment
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.2. Lab Assignment
02:02
monthFor...
sales_dat...
Submit
Explorer
Debugger
Sample Test Cases
Terminal
Test cases

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:

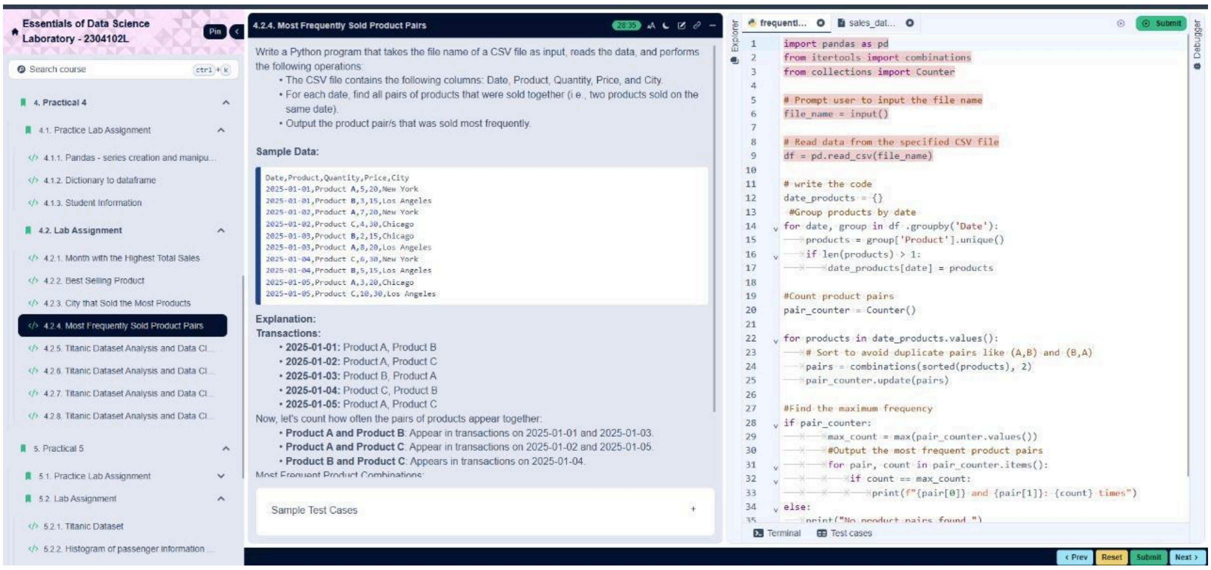
- The CSV file contains the columns: `Date, Product, Quantity, Price,` and `City`.
- Group the data by `City` and calculate the total quantity of products sold for each city.
- Find the city that sold the most products (based on the total quantity sold).

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.

```python
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()

#Find the city with the highest total quantity sold
best_city = city_sales.idxmax()
# Display the result
print(f"City sold the most products: {best_city}")
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4. Practical 4
4.1. Practice Lab Assignment
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
28:35
Explorer
frequentl...
sales_dat...
Submit
Debugger
Terminal
Test cases
‹ 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
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

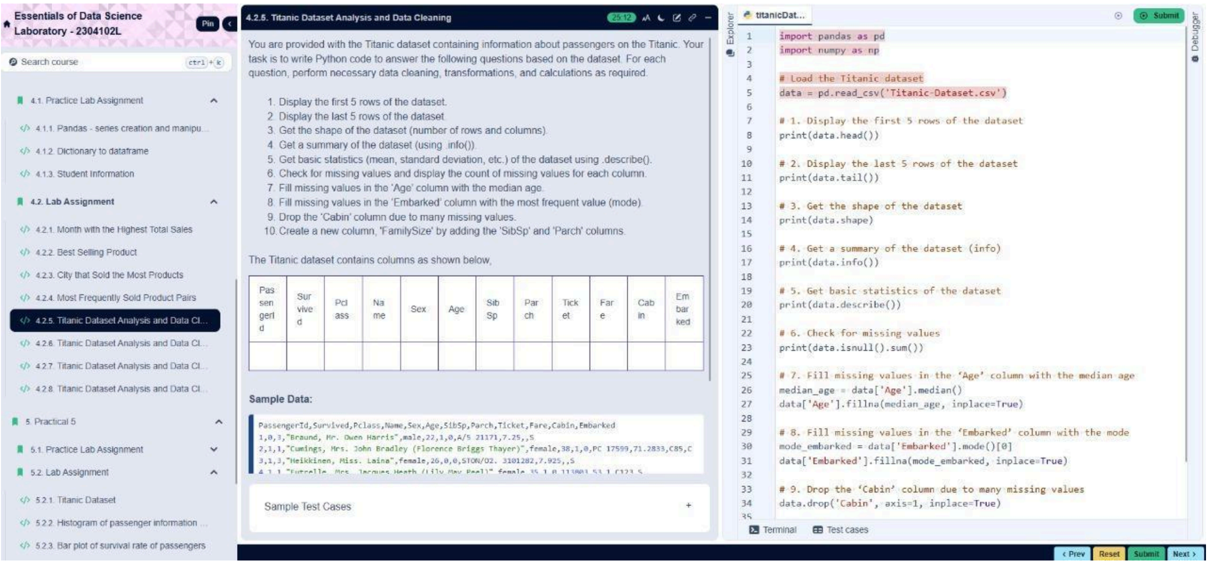
Now, let's count how often the pairs of products appear together:

- **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:

Sample Test Cases +

```python
import pandas as pd
from itertools import combinations
from collections import Counter

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

# Read data from the specified CSV file
df = pd.read_csv(file_name)

# write the code
date_products = {}
 #Group products by date
for date, group in df .groupby('Date'):
    products = group['Product'].unique()
    if len(products) > 1:
        date_products[date] = products

#Count product pairs
pair_counter = Counter()

for products in date_products.values():
    # Sort to avoid duplicate pairs like (A,B) and (B,A)
    pairs = combinations(sorted(products), 2)
    pair_counter.update(pairs)

#Find the maximum frequency
if pair_counter:
        max_count = max(pair_counter.values())
        #Output the most frequent product pairs
        for pair, count in pair_counter.items():
            if count == max_count:
                print(f"{pair[0]} and {pair[1]}: {count} times")
else:
    print("No product pairs found.")
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.1. Practice Lab Assignment
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
25:12
Explorer
titanicDat...
Submit
Debugger
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

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,

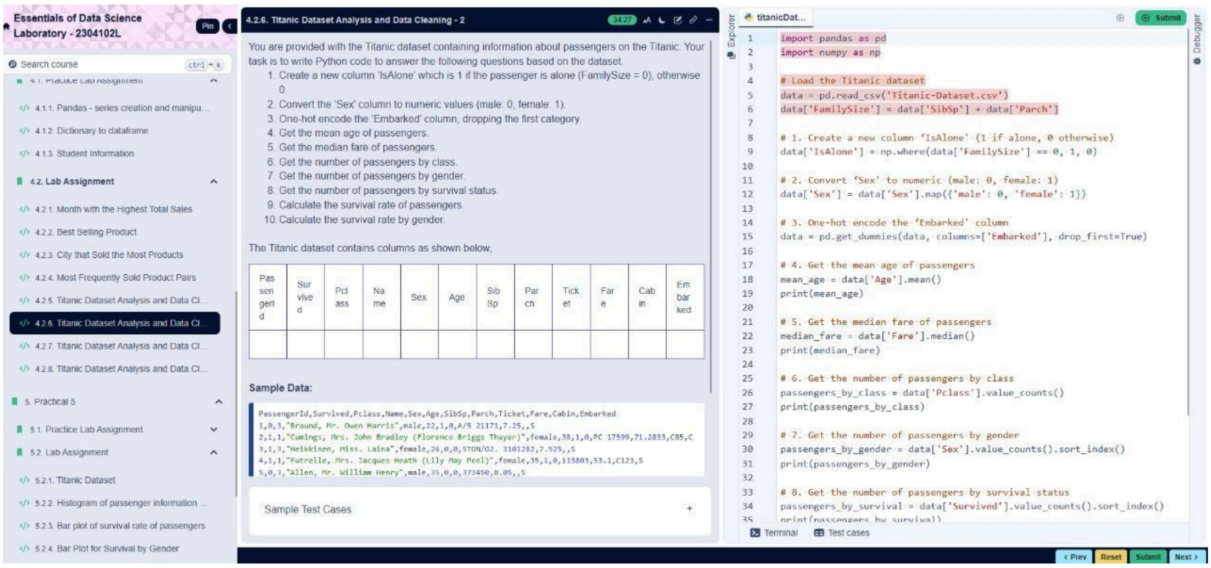
PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

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,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
```

Sample Test Cases +

```python
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)
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.1. Practice Lab Assignment
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
34:27
Explorer
titanicDat...
Submit
Debugger
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

4.2.6. Titanic Dataset Analysis and Data Cleaning - 2

You are provided with the Titanic dataset containing information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset.

1. 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,

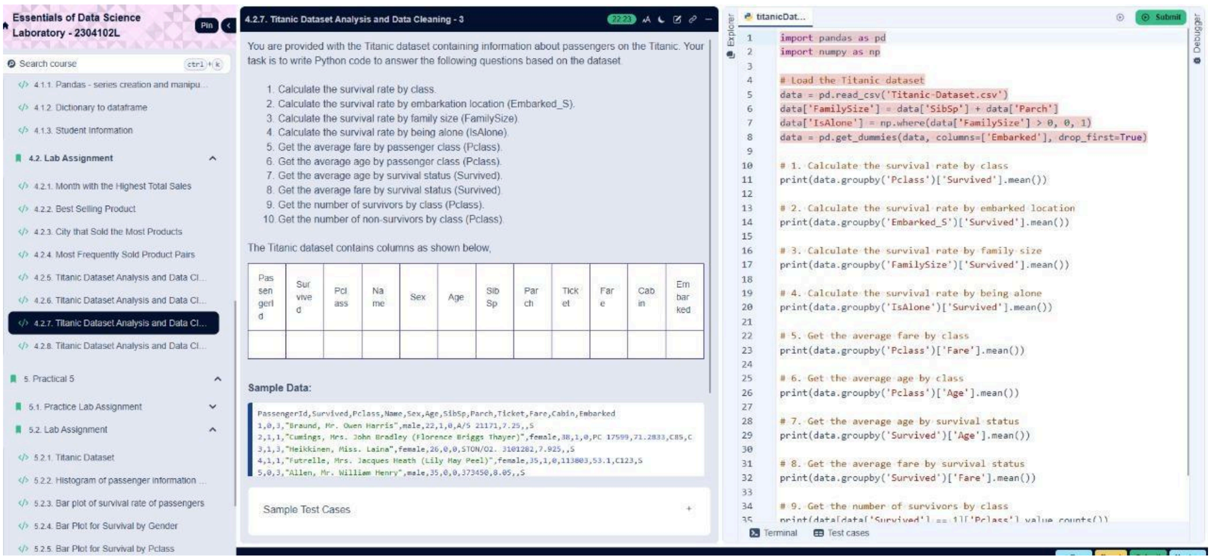
PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
```

Sample Test Cases +

```python
import pandas as pd
import numpy as np

# Load the Titanic dataset
data = pd.read_csv('Titanic-Dataset.csv')
data['FamilySize'] = data['SibSp'] + data['Parch']

# 1. Create a new column ‘IsAlone’ (1 if alone, 0 otherwise)
data['IsAlone'] = 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})

# 3. One-hot encode the ‘Embarked’ column
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# 4. Get the mean age of passengers
mean_age = data['Age'].mean()
print(mean_age)

# 5. Get the median fare of passengers
median_fare = data['Fare'].median()
print(median_fare)

# 6. Get the number of passengers by class
passengers_by_class = data['Pclass'].value_counts()
print(passengers_by_class)

# 7. Get the number of passengers by gender
passengers_by_gender = data['Sex'].value_counts().sort_index()
print(passengers_by_gender)

# 8. Get the number of passengers by survival status
passengers_by_survival = data['Survived'].value_counts().sort_index()
print(passengers_by_survival)
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
4.1.1. Pandas - series creation and manipu...
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
22:23
Explorer
titanicDat...
Submit
Debugger
Terminal
Test cases

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,

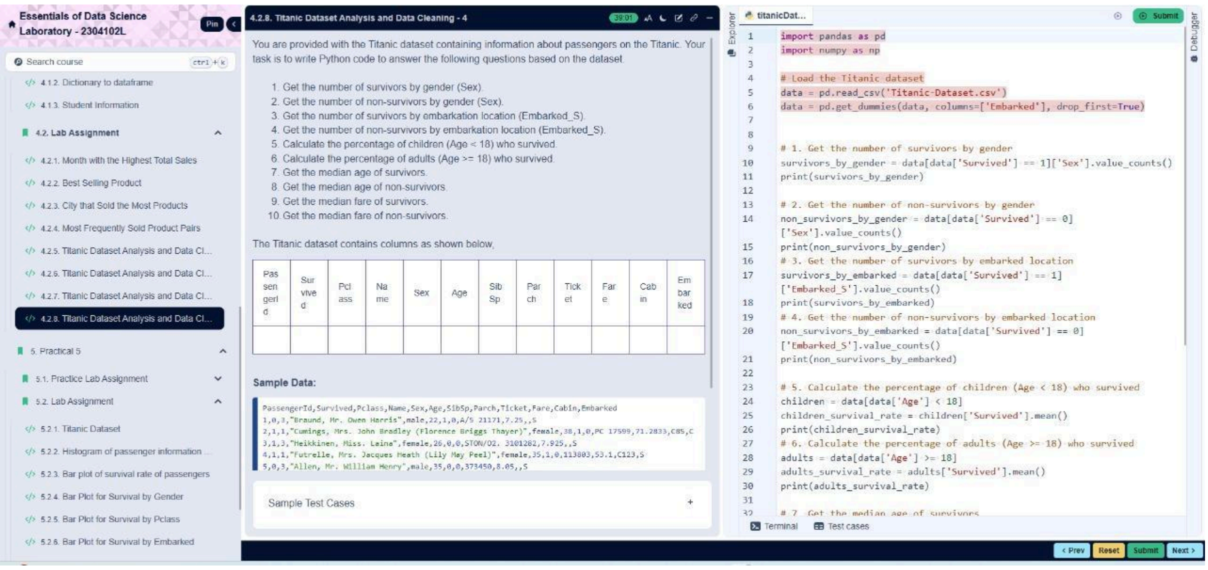
PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

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,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
```

Sample Test Cases +

```python
import pandas as pd
import numpy as np

# Load the Titanic dataset
data = pd.read_csv('Titanic-Dataset.csv')
data['FamilySize'] = data['SibSp'] + data['Parch']
data['IsAlone'] = np.where(data['FamilySize'] > 0, 0, 1)
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# 1. Calculate the survival rate by class
print(data.groupby('Pclass')['Survived'].mean())

# 2. Calculate the survival rate by embarked location
print(data.groupby('Embarked_S')['Survived'].mean())

# 3. Calculate the survival rate by family size
print(data.groupby('FamilySize')['Survived'].mean())

# 4. Calculate the survival rate by being alone
print(data.groupby('IsAlone')['Survived'].mean())

# 5. Get the average fare by class
print(data.groupby('Pclass')['Fare'].mean())

# 6. Get the average age by class
print(data.groupby('Pclass')['Age'].mean())

# 7. Get the average age by survival status
print(data.groupby('Survived')['Age'].mean())

# 8. Get the average fare by survival status
print(data.groupby('Survived')['Fare'].mean())

# 9. Get the number of survivors by class
print(data[data['Survived'] == 1]['Pclass'].value_counts())
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
4.1.2. Dictionary to dataframe
4.1.3. Student Information
4.2. Lab Assignment
4.2.1. Month with the Highest Total Sales
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
39:01
Explorer
titanicDat...
Submit
Debugger
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

4.2.8. Titanic Dataset Analysis and Data Cleaning - 4

You are provided with the Titanic dataset containing information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset.

1. Get the number of survivors by gender (Sex).
2. Get the number of non-survivors by gender (Sex).
3. Get the number of survivors by embarkation location (Embarked_S).
4. Get the number of non-survivors by embarkation location (Embarked_S).
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,

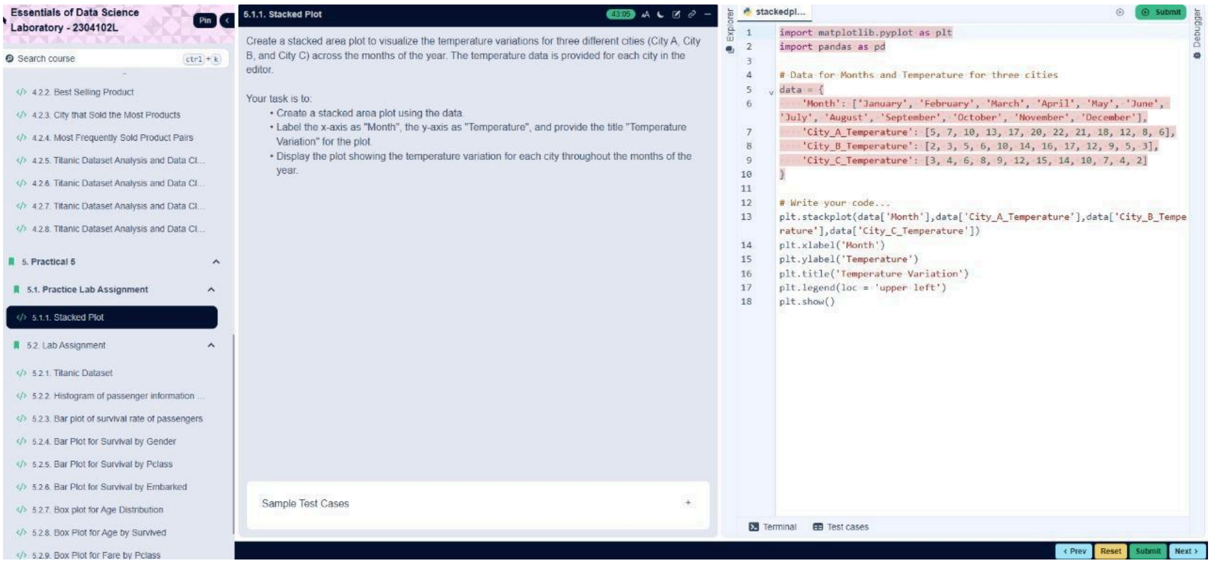
PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

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,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
```

Sample Test Cases +

```python
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)


# 1. Get the number of survivors by gender
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 = data[data['Survived'] == 1]
['Embarked_S'].value_counts()
print(survivors_by_embarked)
# 4. Get the number of non-survivors by embarked location
non_survivors_by_embarked = data[data['Survived'] == 0]
['Embarked_S'].value_counts()
print(non_survivors_by_embarked)

# 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
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
4.2.2. Best Selling Product
4.2.3. City that Sold the Most Products
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box Plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
43:05
Explorer
stackedpl...
Submit
Debugger
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

5.1.1. Stacked Plot

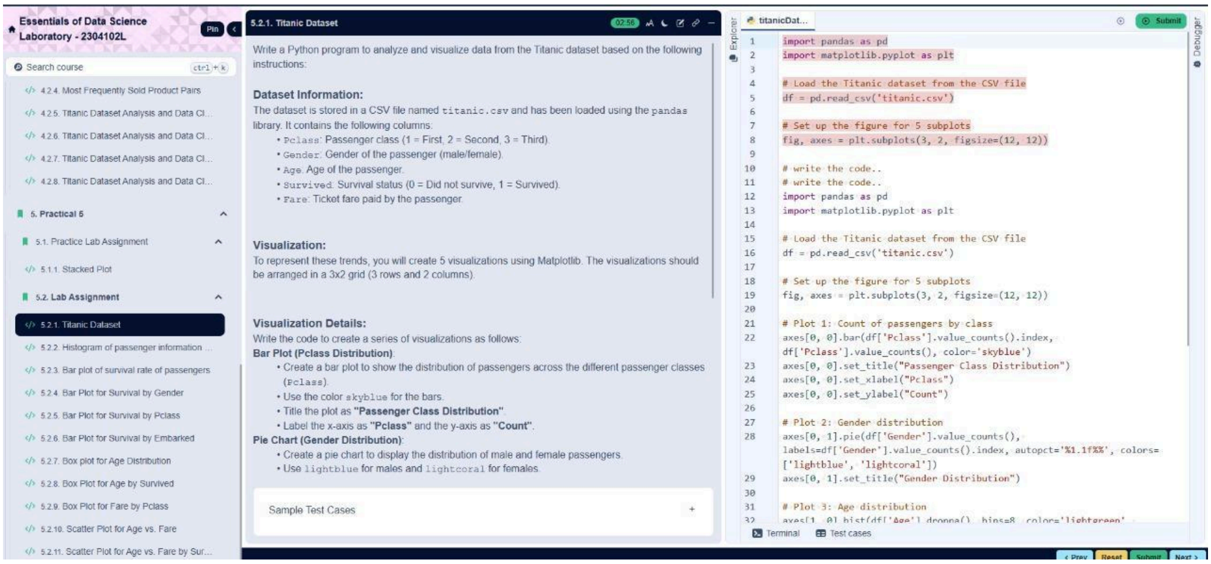
Create a stacked area plot to visualize the temperature variations for three different cities (City A, City B, and City C) across the months of the year. The temperature data is provided for each city in the editor.

Your task is to:

- Create a stacked area plot using the data.
- Label the x-axis as "Month", the y-axis as "Temperature", and provide the title "Temperature Variation" for the plot.
- Display the plot showing the temperature variation for each city throughout the months of the year.

Sample Test Cases +

```python
import matplotlib.pyplot as plt
import pandas as pd

# Data for Months and Temperature for three cities
data = {
    'Month': ['January', 'February', 'March', 'April', 'May', 'June',
'July', 'August', 'September', 'October', 'November', 'December'],
    'City_A_Temperature': [5, 7, 10, 13, 17, 20, 22, 21, 18, 12, 8, 6],
    'City_B_Temperature': [2, 3, 5, 6, 10, 14, 16, 17, 12, 9, 5, 3],
    'City_C_Temperature': [3, 4, 6, 8, 9, 12, 15, 14, 10, 7, 4, 2]
}

# Write your code...
plt.stackplot(data['Month'],data['City_A_Temperature'],data['City_B_Tempe
rature'],data['City_C_Temperature'])
plt.xlabel('Month')
plt.ylabel('Temperature')
plt.title('Temperature Variation')
plt.legend(loc = 'upper left')
plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.2.4. Most Frequently Sold Product Pairs
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
02:56
Explorer
titanicDat...
Submit
Debugger
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

5.2.1. Titanic Dataset

Write a Python program to analyze and visualize data from the Titanic dataset based on the following instructions:

Dataset Information:

The dataset is stored in a CSV file named `titanic.csv` and has been loaded using the `pandas` library. It contains the following columns:

- `Pclass`: Passenger class (1 = First, 2 = Second, 3 = Third).
- `Gender`: Gender of the passenger (male/female).
- `Age`: Age of the passenger.
- `Survived`: Survival status (0 = Did not survive, 1 = Survived).
- `Fare`: Ticket fare paid by the passenger.

Visualization:

To represent these trends, you will create 5 visualizations using Matplotlib. The visualizations should be arranged in a 3x2 grid (3 rows and 2 columns).

Visualization Details:

Write the code to create a series of visualizations as follows:

Bar Plot (Pclass Distribution):

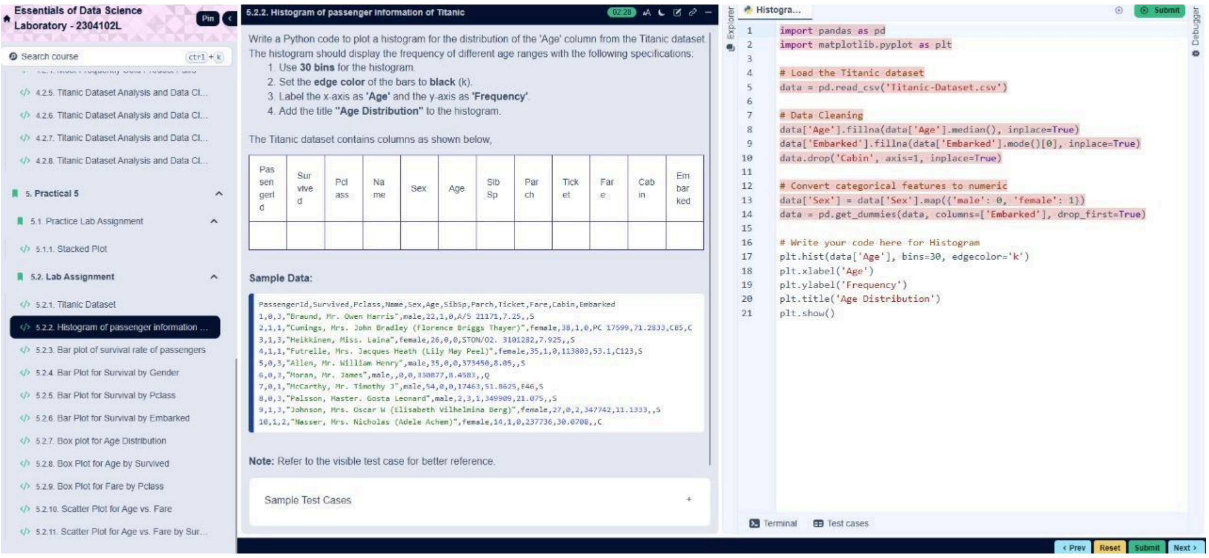
- Create a bar plot to show the distribution of passengers across the different passenger classes (`Pclass`).
- Use the color `skyblue` for the bars.
- Title the plot as **"Passenger Class Distribution"**.
- Label the x-axis as **"Pclass"** and the y-axis as **"Count"**.

Pie Chart (Gender Distribution):

- Create a pie chart to display the distribution of male and female passengers.
- Use `lightblue` for males and `lightcoral` for females.

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the Titanic dataset from the CSV file
df = pd.read_csv('titanic.csv')

# Set up the figure for 5 subplots
fig, axes = plt.subplots(3, 2, figsize=(12, 12))

# write the code..
# write the code..
import pandas as pd
import matplotlib.pyplot as plt

# Load the Titanic dataset from the CSV file
df = pd.read_csv('titanic.csv')

# Set up the figure for 5 subplots
fig, axes = plt.subplots(3, 2, figsize=(12, 12))

# Plot 1: Count of passengers by class
axes[0, 0].bar(df['Pclass'].value_counts().index,
df['Pclass'].value_counts(), color='skyblue')
axes[0, 0].set_title("Passenger Class Distribution")
axes[0, 0].set_xlabel("Pclass")
axes[0, 0].set_ylabel("Count")

# Plot 2: Gender distribution
axes[0, 1].pie(df['Gender'].value_counts(),
labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
['lightblue', 'lightcoral'])
axes[0, 1].set_title("Gender Distribution")

# Plot 3: Age distribution
axes[1, 0].hist(df['Age'].dropna(), bins=8, color='lightgreen',
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
02:28
Explorer
Histogra...
Submit
Debugger
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

5.2.2. Histogram of passenger information of Titanic

Write a Python code to plot a histogram for the distribution of the 'Age' column from the Titanic dataset. The histogram should display the frequency of different age ranges with the following specifications:

1. Use **30 bins** for the histogram.
2. Set the **edge color** of the bars to **black** (k).
3. Label the x-axis as **'Age'** and the y-axis as **'Frequency'**.
4. Add the title **"Age Distribution"** to the histogram.

The Titanic dataset contains columns as shown below,

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

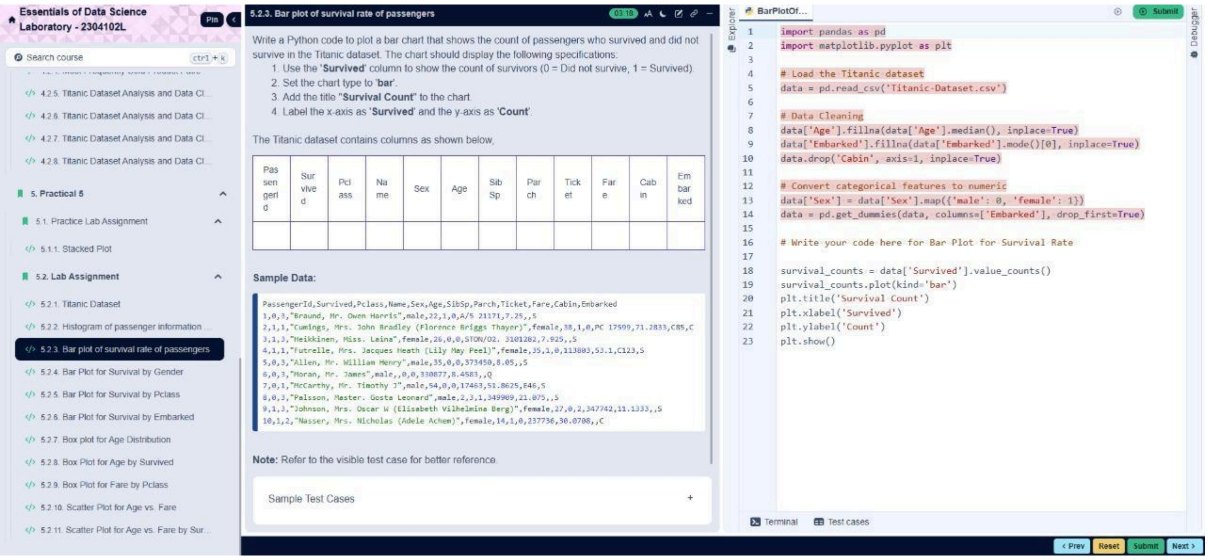
Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
```

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

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Histogram
plt.hist(data['Age'], bins=30, edgecolor='k')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Age Distribution')
plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
03:18
Explorer
BarPlotOf...
Submit
Debugger
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

5.2.3. Bar plot of survival rate of passengers

Write a Python code to plot a bar chart that shows the count of passengers who survived and did not survive in the Titanic dataset. The chart should display the following specifications:

1. Use the '**Survived**' column to show the count of survivors (0 = Did not survive, 1 = Survived).
2. Set the chart type to '**bar**'.
3. Add the title "**Survival Count**" to the chart.
4. Label the x-axis as '**Survived**' and the y-axis as '**Count**'.

The Titanic dataset contains columns as shown below,

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

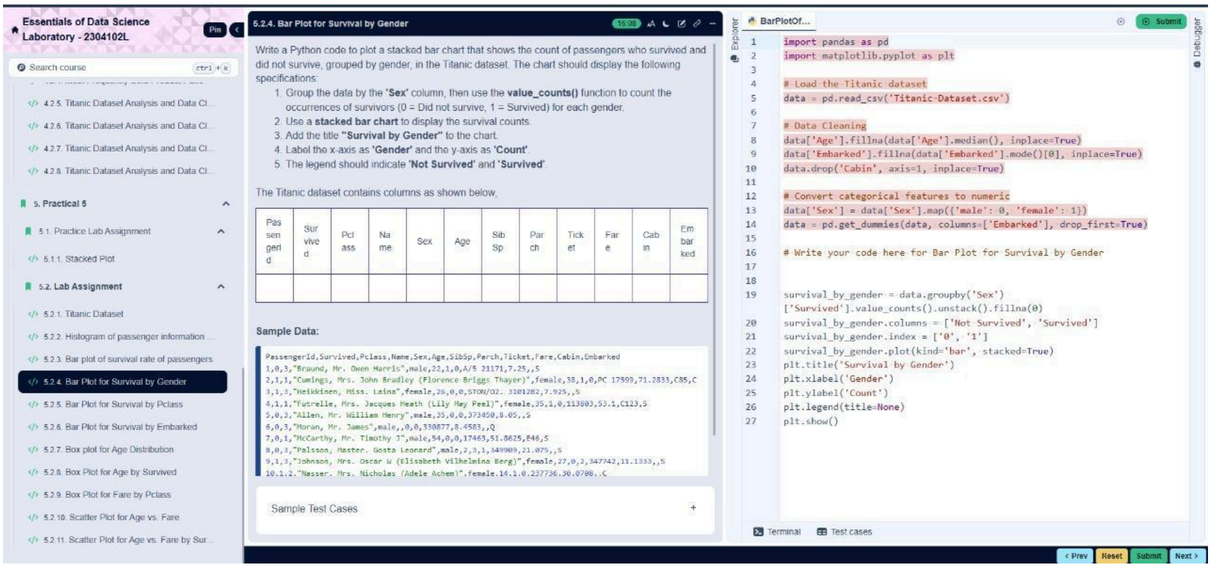
Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
```

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

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Bar Plot for Survival Rate

survival_counts = data['Survived'].value_counts()
survival_counts.plot(kind='bar')
plt.title('Survival Count')
plt.xlabel('Survived')
plt.ylabel('Count')
plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
16:08
Explorer
BarPlotOf...
Submit
Debugger
Terminal
Test cases
< Prev
Reset
Submit
Next >

5.2.4. Bar Plot for Survival by Gender

Write a Python code to plot a stacked bar chart that shows the count of passengers who survived and did not survive, grouped by gender, in the Titanic dataset. The chart should display the following specifications:

1. Group the data by the **'Sex'** column, then use the **value_counts()** function to count the occurrences of survivors (0 = Did not survive, 1 = Survived) for each gender.
2. Use a **stacked bar chart** to display the survival counts.
3. Add the title **"Survival by Gender"** to the chart.
4. Label the x-axis as **'Gender'** and the y-axis as **'Count'**.
5. The legend should indicate **'Not Survived'** and **'Survived'**.

The Titanic dataset contains columns as shown below,

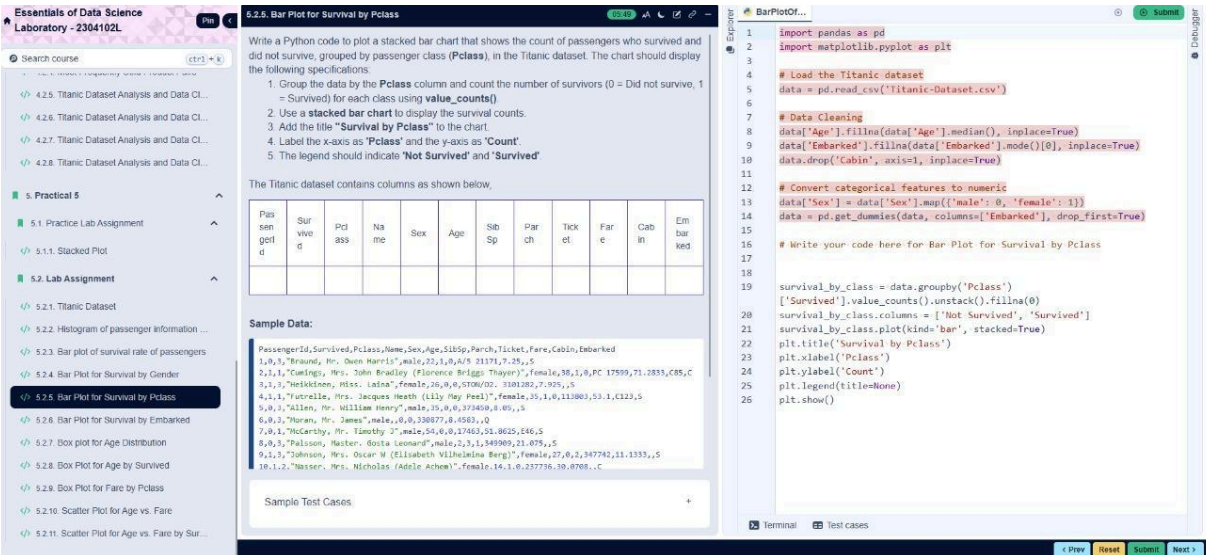
Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

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,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
```

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Bar Plot for Survival by Gender


survival_by_gender = data.groupby('Sex')['Survived'].value_counts().unstack().fillna(0)
survival_by_gender.columns = ['Not Survived', 'Survived']
survival_by_gender.index = ['0', '1']
survival_by_gender.plot(kind='bar', stacked=True)
plt.title('Survival by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend(title=None)
plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
05:49
Explorer
BarPlotOf...
Submit
Debugger
Terminal
Test cases
< Prev
Reset
Submit
Next >

5.2.5. Bar Plot for Survival by Pclass

Write a Python code to plot a stacked bar chart that shows the count of passengers who survived and did not survive, grouped by passenger class (**Pclass**), in the Titanic dataset. The chart should display the following specifications:

1. Group the data by the **Pclass** column and count the number of survivors (0 = Did not survive, 1 = Survived) for each class using **value_counts()**.
2. Use a **stacked bar chart** to display the survival counts.
3. Add the title **"Survival by Pclass"** to the chart.
4. Label the x-axis as **'Pclass'** and the y-axis as **'Count'**.
5. The legend should indicate **'Not Survived'** and **'Survived'**.

The Titanic dataset contains columns as shown below,

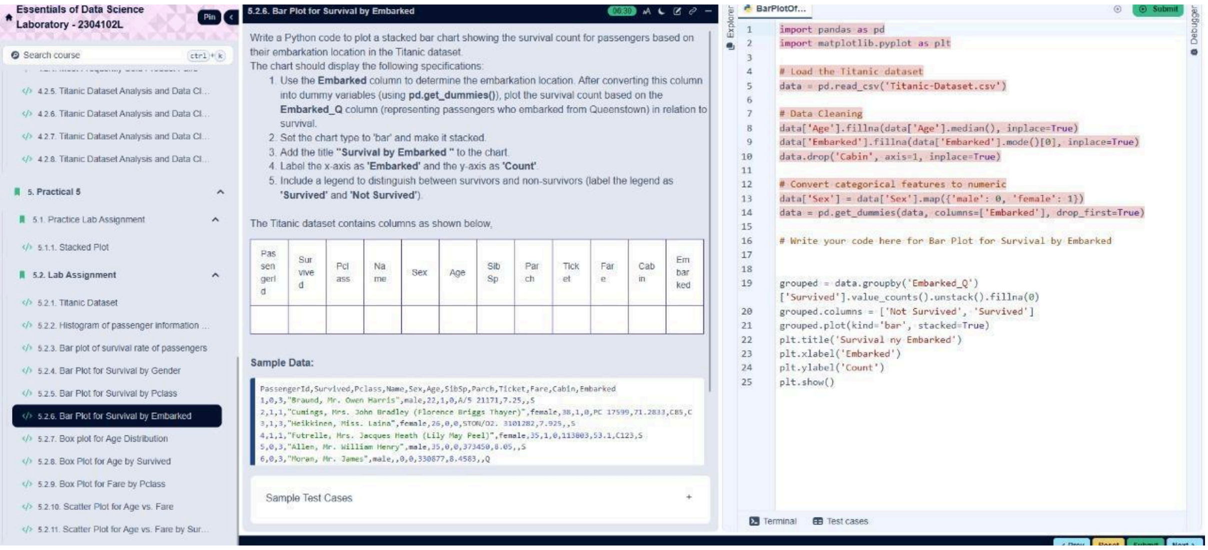
Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

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,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
```

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Bar Plot for Survival by Pclass


survival_by_class = data.groupby('Pclass')['Survived'].value_counts().unstack().fillna(0)
survival_by_class.columns = ['Not Survived', 'Survived']
survival_by_class.plot(kind='bar', stacked=True)
plt.title('Survival by Pclass')
plt.xlabel('Pclass')
plt.ylabel('Count')
plt.legend(title=None)
plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
06:30
Explorer
BarPlotOf...
Submit
Debugger
Terminal
Test cases

5.2.6. Bar Plot for Survival by Embarked

Write a Python code to plot a stacked bar chart showing the survival count for passengers based on their embarkation location in the Titanic dataset.
The chart should display the following specifications:

1. Use the **Embarked** column to determine the embarkation location. After converting this column into dummy variables (using **pd.get_dummies()**), plot the survival count based on the **Embarked_Q** column (representing passengers who embarked from Queenstown) in relation to survival.
2. Set the chart type to 'bar' and make it stacked.
3. Add the title **"Survival by Embarked "** to the chart.
4. Label the x-axis as **'Embarked'** and the y-axis as **'Count'**.
5. Include a legend to distinguish between survivors and non-survivors (label the legend as **'Survived'** and **'Not Survived'**).

The Titanic dataset contains columns as shown below,

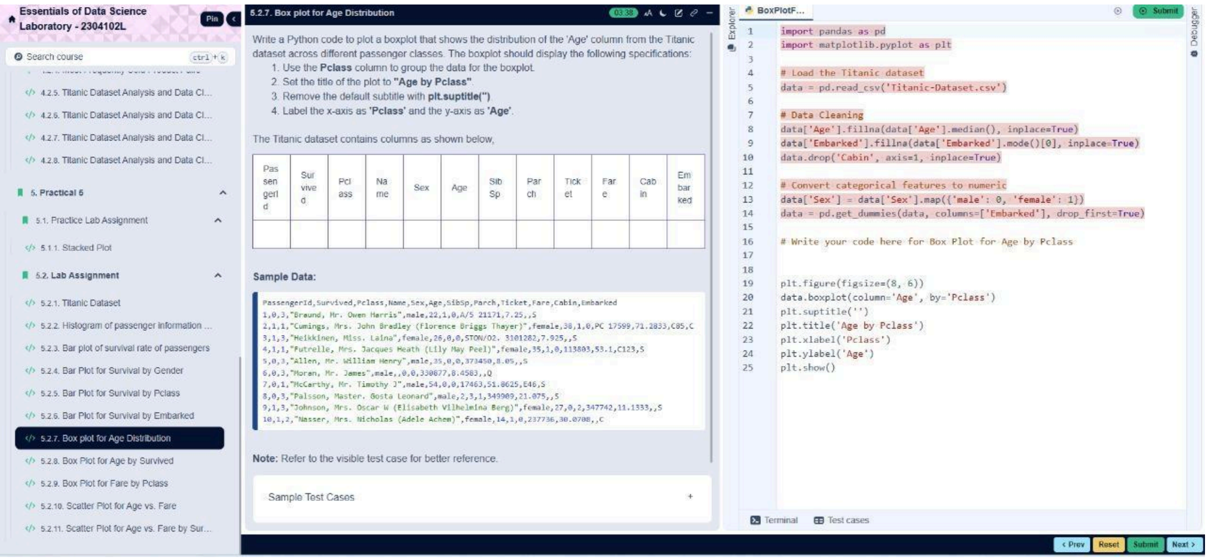
Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

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,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
```

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Bar Plot for Survival by Embarked


grouped = data.groupby('Embarked_Q')['Survived'].value_counts().unstack().fillna(0)
grouped.columns = ['Not Survived', 'Survived']
grouped.plot(kind='bar', stacked=True)
plt.title('Survival ny Embarked')
plt.xlabel('Embarked')
plt.ylabel('Count')
plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
03:38
Explorer
BoxPlotF...
Submit
Debugger
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

5.2.7. Box plot for Age Distribution

Write a Python code to plot a boxplot that shows the distribution of the 'Age' column from the Titanic dataset across different passenger classes. The boxplot should display the following specifications:

1. Use the **Pclass** column to group the data for the boxplot.
2. Set the title of the plot to **"Age by Pclass"**.
3. Remove the default subtitle with **plt.suptitle('')**.
4. Label the x-axis as **'Pclass'** and the y-axis as **'Age'**.

The Titanic dataset contains columns as shown below,

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

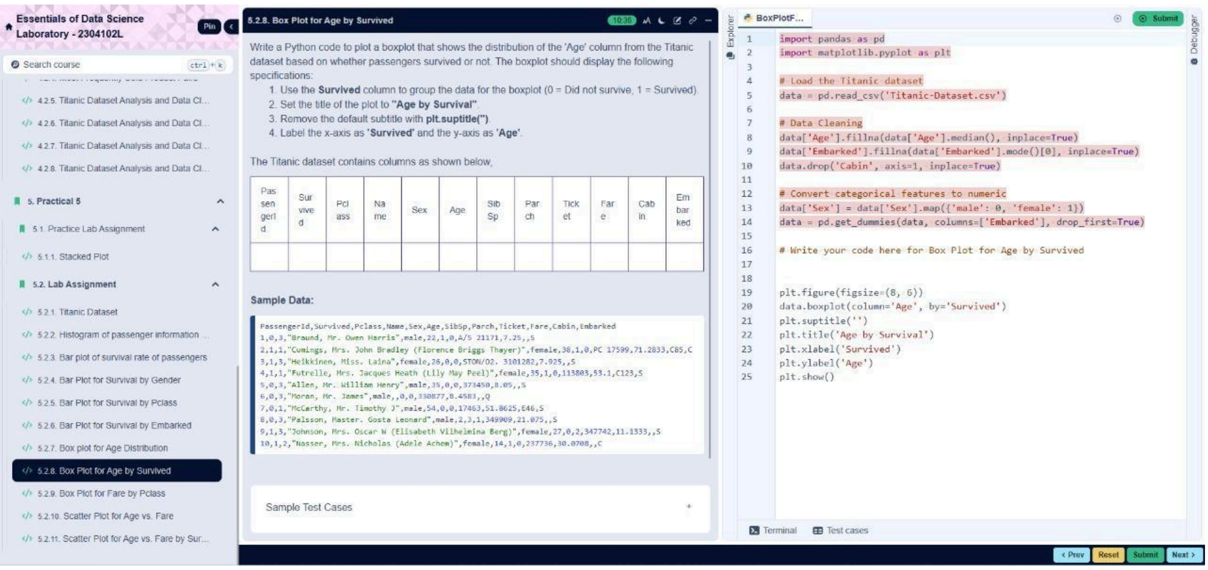
Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
```

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

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Box Plot for Age by Pclass


plt.figure(figsize=(8, 6))
data.boxplot(column='Age', by='Pclass')
plt.suptitle('')
plt.title('Age by Pclass')
plt.xlabel('Pclass')
plt.ylabel('Age')
plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
10:36
Explorer
BoxPlotF...
Submit
Debugger
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

5.2.8. Box Plot for Age by Survived

Write a Python code to plot a boxplot that shows the distribution of the 'Age' column from the Titanic dataset based on whether passengers survived or not. The boxplot should display the following specifications:

1. Use the **Survived** column to group the data for the boxplot (0 = Did not survive, 1 = Survived).
2. Set the title of the plot to **"Age by Survival"**.
3. Remove the default subtitle with **plt.suptitle('')**.
4. Label the x-axis as **'Survived'** and the y-axis as **'Age'**.

The Titanic dataset contains columns as shown below,

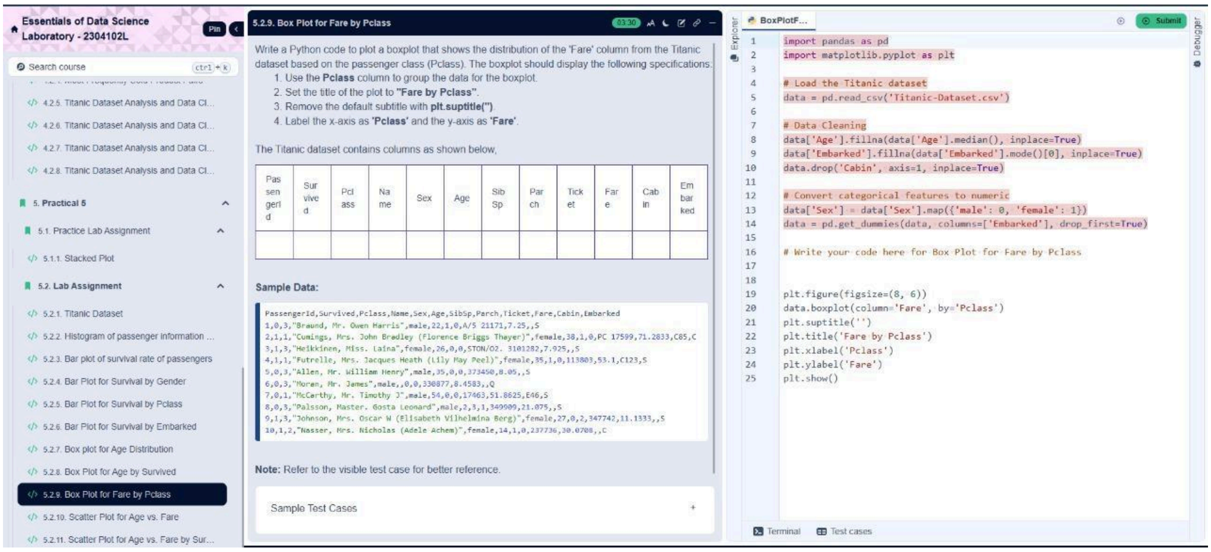
PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

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,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
```

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Box Plot for Age by Survived


plt.figure(figsize=(8, 6))
data.boxplot(column='Age', by='Survived')
plt.suptitle('')
plt.title('Age by Survival')
plt.xlabel('Survived')
plt.ylabel('Age')
plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course
ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
03:30
Explorer
BoxPlotF...
Submit
Debugger
Terminal
Test cases

5.2.9. Box Plot for Fare by Pclass

Write a Python code to plot a boxplot that shows the distribution of the 'Fare' column from the Titanic dataset based on the passenger class (Pclass). The boxplot should display the following specifications:

1. Use the **Pclass** column to group the data for the boxplot.
2. Set the title of the plot to **"Fare by Pclass"**.
3. Remove the default subtitle with **plt.suptitle('')**.
4. Label the x-axis as **'Pclass'** and the y-axis as **'Fare'**.

The Titanic dataset contains columns as shown below,

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

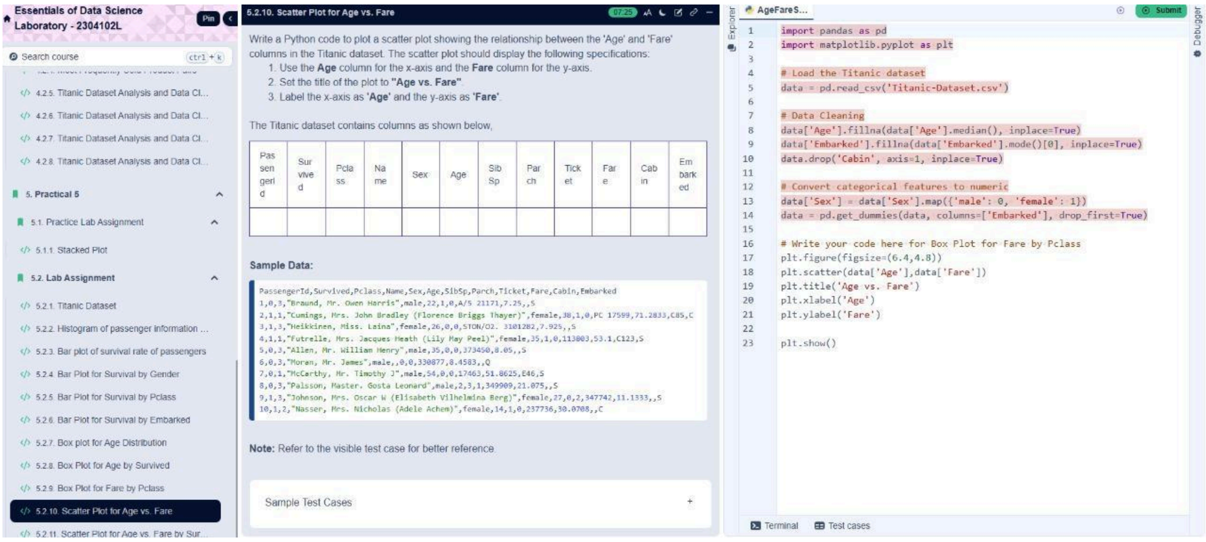
Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
```

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

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Box Plot for Fare by Pclass


plt.figure(figsize=(8, 6))
data.boxplot(column='Fare', by='Pclass')
plt.suptitle('')
plt.title('Fare by Pclass')
plt.xlabel('Pclass')
plt.ylabel('Fare')
plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
07:25
Explorer
AgeFareS...
Submit
Debugger
Terminal
Test cases

5.2.10. Scatter Plot for Age vs. Fare

Write a Python code to plot a scatter plot showing the relationship between the 'Age' and 'Fare' columns in the Titanic dataset. The scatter plot should display the following specifications:

1. Use the **Age** column for the x-axis and the **Fare** column for the y-axis.
2. Set the title of the plot to **"Age vs. Fare"**.
3. Label the x-axis as **'Age'** and the y-axis as **'Fare'**.

The Titanic dataset contains columns as shown below,

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

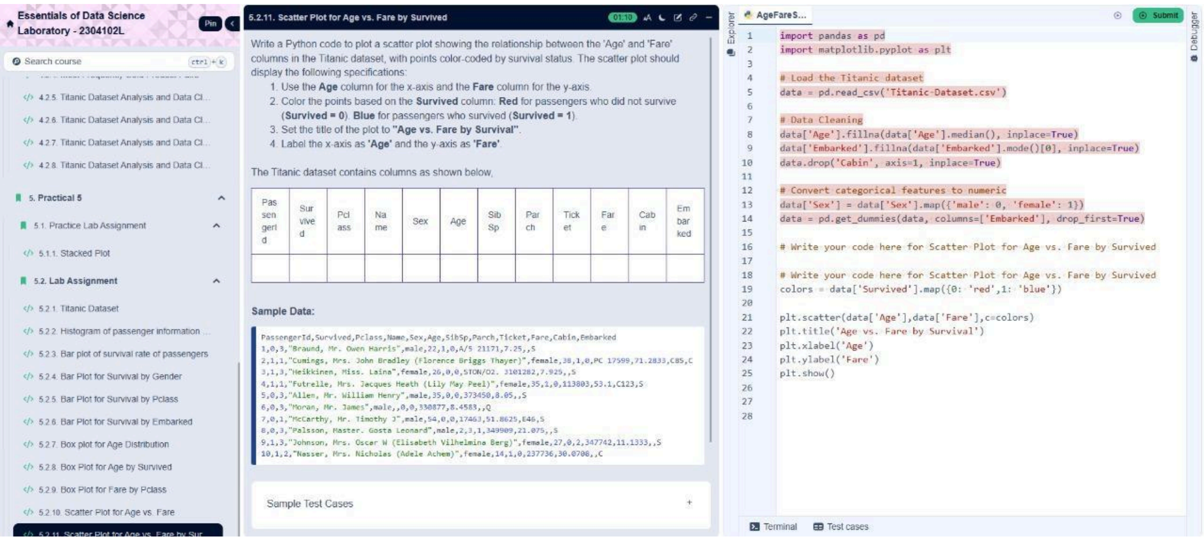
Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
```

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

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Box Plot for Fare by Pclass
plt.figure(figsize=(6.4,4.8))
plt.scatter(data['Age'],data['Fare'])
plt.title('Age vs. Fare')
plt.xlabel('Age')
plt.ylabel('Fare')

plt.show()
```


Essentials of Data Science Laboratory - 2304102L
Pin
Search course ctrl + k
4.2.5. Titanic Dataset Analysis and Data Cl...
4.2.6. Titanic Dataset Analysis and Data Cl...
4.2.7. Titanic Dataset Analysis and Data Cl...
4.2.8. Titanic Dataset Analysis and Data Cl...
5. Practical 5
5.1. Practice Lab Assignment
5.1.1. Stacked Plot
5.2. Lab Assignment
5.2.1. Titanic Dataset
5.2.2. Histogram of passenger information ...
5.2.3. Bar plot of survival rate of passengers
5.2.4. Bar Plot for Survival by Gender
5.2.5. Bar Plot for Survival by Pclass
5.2.6. Bar Plot for Survival by Embarked
5.2.7. Box plot for Age Distribution
5.2.8. Box Plot for Age by Survived
5.2.9. Box Plot for Fare by Pclass
5.2.10. Scatter Plot for Age vs. Fare
5.2.11. Scatter Plot for Age vs. Fare by Sur...
01:10
Explorer
AgeFareS...
Submit
Debugger
Terminal
Test cases

5.2.11. Scatter Plot for Age vs. Fare by Survived

Write a Python code to plot a scatter plot showing the relationship between the 'Age' and 'Fare' columns in the Titanic dataset, with points color-coded by survival status. The scatter plot should display the following specifications:

1. Use the **Age** column for the x-axis and the **Fare** column for the y-axis.
2. Color the points based on the **Survived** column: **Red** for passengers who did not survive (**Survived = 0**). **Blue** for passengers who survived (**Survived = 1**).
3. Set the title of the plot to **"Age vs. Fare by Survival"**.
4. Label the x-axis as **'Age'** and the y-axis as **'Fare'**.

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,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
```

Sample Test Cases +

```python
import pandas as pd
import matplotlib.pyplot as plt

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

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Scatter Plot for Age vs. Fare by Survived

# Write your code here for Scatter Plot for Age vs. Fare by Survived
colors = data['Survived'].map({0: 'red',1: 'blue'})

plt.scatter(data['Age'],data['Fare'],c=colors)
plt.title('Age vs. Fare by Survival')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```