

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e447f19c5320ca6bc84/6773e494f19c5320ca6bd76/6773e9c2f4ab787eca9d09e4 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

1.1.1. Calculate Momentum 15:20 Submit Debugger

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).

Sample Test Cases +

calculate... Explorer

```
1 m=float(input())
2 v=float(input())
3 p=m*v
4 print("%0.2f"%p,end=' ')
5 print("kgm/s")
```

Terminal Test cases < Prev Reset Submit Next >

22:29:32 07-05-2025 ENG IN

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773eb5df4ab787eca9d0bf7 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

1.1.2. Conditional Calculation Based on the Number of Digits

12:09 A ☰

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

Constraints:
 $1 \leq n \leq 999$

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 +

condition...

```
1 n=int(input())
2 v if n>=0 and n<=10:
3   --->p=n*n
4   --->print('%0.0f'%p)
5 v elif n>=10 and n<=99:
6   --->q=n**0.5
7   --->print('%0.2f'%q)
8 v elif n>=100 and n<=999:
9   --->r=n**(1/3)
10  --->print('%2f'%r)
11 v else:
12   --->print("Invalid")
```

Terminal Test cases < Prev Reset Submit Next >

22:29:14 07-05-2025 ENG IN

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bc84/6773e494f1f9c5320ca6bd76/6773ef42f4ab787eca9d11f7 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

1.1.3. Age and Salary Calculation 45:54 Submit Debugger

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 +

birthDate...
18 print(f"Age: {age}")
19 print(f"Salary in dollars: {salary_in_dollars:.2f}")
20 from datetime import datetime
21 v def calculate_age(birthdate):
22 # Write your code here
23 birth_day, birth_month, birth_year = map(int, birthdate.split("-"))
24 current_date = datetime.now()
25 age = current_date.year - birth_year
26 if (current_date.month, current_date.day) < (birth_month, birth_day):
27 age -= 1
28 return age
29 v def convert_salary_to_dollars(salary_in_rupees):
30 # write your code here
31 return salary_in_rupees * 0.012
32
33
34 birthdate = input()
35 salary_in_rupees = float(input())
36 age = calculate_age(birthdate)
37 salary_in_dollars = convert_salary_to_dollars(salary_in_rupees)
38 print(f"Age: {age}")
39 print(f"Salary in dollars: {salary_in_dollars:.2f}")
40
41
42

Terminal Test cases

< Prev Reset Submit Next >

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e447f19c5320ca6bc85#contents/6773e447f19c5320ca6bc85#contents/6773e494f19c5320ca6bd76/6774c276f4ab787eca9d88e3 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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 +

reverseN...
1 num=int(input())
2 n1=str(num)
3 print(n1[::-1])

Terminal Test cases < Prev Reset Submit Next >

The screenshot displays a comprehensive development environment. At the top, a browser window shows the CodeTantra website with a user profile and navigation links. Below the browser is the main application interface, divided into several panes: a code editor containing Python code to reverse a number, a terminal window, and a test cases section. A sidebar on the left provides input and output format specifications. The bottom of the screen features a Windows taskbar with various pinned icons and system status indicators.

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e447f19c5320ca6bcce4/6773e494f19c5320ca6bd76/678dd2f0fa4f5e20f6cdc1a Verify it's you

202401120031@mitaoe.ac.in Support Logout

CODETANTRA Home

Essentials of Data Science
Laboratory - 2304102L

Search course

1. Practical 1

- 1.1. Practice Lab Assignment
 - 1.1.1. Calculate Momentum
 - 1.1.2. Conditional Calculation Based on th...
 - 1.1.3. Age and Salary Calculation
 - 1.1.4. Reverse a Number
- 1.1.5. Multiplication Table

2. Lab Assignment

- 1.2.1. Pass or Fail
- 1.2.2. Fibonacci series using Recursive Fu...
- 1.2.3. Pattern - 1
- 1.2.4. Pattern - 2

2. Practical 2

- 2.1. Practice Lab Assignment

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.

Sample Test Cases

Explorer

```
i=multiplica...
1 i=int(input())
2 n=1
3 v while n<=10:
4     print(i,"x",n,"=",i*n)
5     n=n+1
```

Terminal Test cases

< Prev Reset Submit Next >

22:28:00 07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bd76/6773eb5df4ab787eca9d0bf7 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

1.1.2. Conditional Calculation Based on the Number of Digits

12:09

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

Constraints:
 $1 \leq n \leq 999$

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 +

condition...
Submit Debugger

```
1 n=int(input())
2 v if n>=0 and n<=10:
3     p=n*n
4     print('%0.0f'%p)
5 v elif n>=10 and n<=99:
6     q=n**0.5
7     print('%0.2f'%q)
8 v elif n>=100 and n<=999:
9     r=n**(1/3)
10    print('%2f'%r)
11 v else:
12     print("Invalid")
```

Terminal Test cases < Prev Reset Submit Next >

22:29:14 07-05-2025

Essentials of Data Science
Laboratory - 2304102L

Pin

Search course ctrl + k

1. Practical 1

1.1. Practice Lab Assignment

1.2. Lab Assignment

1.2.1. Pass or Fail

1.2.2. Fibonacci series using Recursive Function

1.2.3. Pattern - 1

1.2.4. Pattern - 2

2. Practical 2

2.1. Practice Lab Assignment

2.1.1. List operations

2.1.2. Dictionary Operations

2.2. Lab Assignment

3. Practical 3

1.2.2. Fibonacci series using Recursive Function

01:47 A ⚡ ⚡ -

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

```
1 v def fib(n):
2 v   --> if n<=1:
3 v     -->--> return n
4 v   --> else:
5 v     -->--> return fib(n-1) + fib(n-2)
6 n=int(input("Enter terms for Fibonacci series: "))
7 v for i in range (n):
8   -->--> print(fib(i),end=" ")
```

Submit

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >

22:26:58
07-05-2025

ENG IN

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bc84/6773e49af1f9c5320ca6bd79/6774c521f4ab787eca9d8a75 Verify it's you

202401120031@mitaoe.ac.in Support Logout

1.2.3. Pattern - 1

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 +

rightangl...
rows = int(input(""))
i=0
for i in range(i,rows):
 i += 1
 print("* "*i)
 i+=1

Terminal Test cases < Prev Reset Submit Next >

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bcce4/6773e49af1f9c5320ca6bd79/6774cc86f4ab787eca9d8f99 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

1.2.4. Pattern - 2 02:59

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 +

Explorer numberP...
1 n= int(input())
2 v for i in range(1,n+1):
3 v ---->for j in range(1,i+1):
4 v ----->print(j , end=" ")
5 v ----->print()

Terminal Test cases < Prev Reset Submit Next >

6 Search File Firefox Edge Excel S Google Chrome Spotify WhatsApp Microsoft Edge 22:25:55 ENG IN 07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#contents/6773e44bf19c5320ca6bc8e/6773e49ef19c5320ca6bd7d/667bb7f01060f3d690f3fb3 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

2.1.1. List operations

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 +

listOps.py

```
1 # write the code..
2 menu = ["Add", "Remove", "Display", "Quit"]
3 list = []
4 while True:
5     for i in range(len(menu)):
6         print(f"{i+1}. {menu[i]}")
7     choice = int(input("Enter choice: "))
8     if choice == 1:
9         try:
10             a = int(input("Integer: "))
11             list.append(a)
12             print("List after adding:", list)
13         except int:
14             print("Invalid input")
15         elif choice == 2:
16             if list == []:
17                 print("List is empty")
18             else:
19                 a = int(input("Integer: "))
20                 if a in list:
21                     list.remove(a)
22                     print("List after removing:", list)
23                 else:
24                     print("Element not found")
25         elif choice == 3:
```

< Prev Reset Submit Next >

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bc8e/6773e49ef1f9c5320ca6bd7d/677bb1ca63fd602f55ad634f Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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.

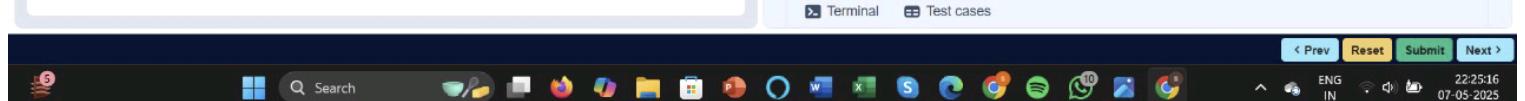
Note: Refer to visible test cases.

Sample Test Cases +

dictOpera...

```
88
89     d={}
90     print("Empty Dictionary: {}")
91     n=int(input("Number of items: "))
92     for i in range(n):
93         key=input("key: ")
94         value=input("value: ")
95         d[key]=value
96     print("Dictionary:",d)
97     update_key=input("Enter the key to update: ")
98     if update_key in d:
99         d[update_key] = input("Enter the new value: ")
100        print("Value updated")
101    else:
102        print("key not found")
103    retrieve_key=input("Enter the key to retrieve: ")
104    if retrieve_key in d:
105        print(f"Key: {retrieve_key}, Value: {d[retrieve_key]}")
106    else:
107        print("Key not found")
108    get_key=input("Enter the key to get using the get() method: ")
109    if get_key in d:
110        print(f"Key: {get_key}, Value: {d[get_key]}")
111    else:
112        print("Key not found")
```

< Prev Reset Submit Next >



Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e4b0f1f9c5320ca6bd85/64ac0cb9b4547106899a4e1b Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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

Sample Test Cases +

CTP1709...
1 v def linear_search(arr, key):
2 v for index in range(len(arr)):
3 v if arr[index] == key:
4 v return index
5 v return "Not found"
6
7 arr = list(map(int, input().split()))
8
9 key = int(input())
10
11 result = linear_search(arr, key)
12 print(result)#

Terminal Test cases < Prev Reset Submit Next >

22:23:28 07-05-2025 ENG IN

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e44bf19c5320ca6bc8/6773e4b0f19c5320ca6bd85/6774d0abf4ab787eca9d9230

Verify it's you

Logout

CODETANTRA Home

202401120031@mitaoe.ac.in Support Logout

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.

Sample Test Cases

```
heights = list(map(int, input().split()))
tallest_player = max(heights)
print(tallest_player)
```

Terminal Test cases

< Prev Reset Submit Next >

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4cf1f9c5320ca6bdb2/6684d40e5d95376955a73c1d Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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 +

numpyarr...

```
import numpy as np
def main():
    rows, cols = map(int,input().split())
    elements=[]
    for _ in range(rows):
        elements.extend(map(int,input().split()))
    array=np.array(elements)
    array=array.reshape(rows,cols)
    print(array)
    print(array.ndim)
    print(array.shape)
    print(array.size)
if __name__ == "__main__":
    main()
```

Terminal Test cases < Prev Reset Submit Next >

22:22:51 07-05-2025 ENG IN

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774e3fff4ab787eca9da092 Verify it's you

202401120031@mitaoe.ac.in Support Logout

3.2.1. Numpy: Matrix Operations

The given code takes two 3×3 matrices, `matrix_a`, and `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 (`matrix_a + matrix_b`)
2. Subtraction (`matrix_a - matrix_b`)
3. Element-wise Multiplication (`matrix_a * matrix_b`)
4. Matrix Multiplication (`matrix_a . matrix_b`)
5. Transpose of Matrix A

Input Format:
The user will input 3 rows for `matrix_a`, each containing 3 integers separated by spaces.
Similarly, the user will input 3 rows for `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.

Sample Test Cases +

```
matrixOp...
1 import numpy as np
2
3 # Input matrices
4 print("Enter Matrix A:")
5 matrix_a = np.array([list(map(int, input().split())) for i in range(3)])
6
7 print("Enter Matrix B:")
8 matrix_b = np.array([list(map(int, input().split())) for i in range(3)])
9
10
11 # Addition
12 addition_result = matrix_a + matrix_b
13 print("Addition (A + B):")
14 print(addition_result)
15 # Subtraction
16 sub_result = matrix_a - matrix_b
17 print("Subtraction (A - B):")
18 print(sub_result)
19 # Multiplication (element-wise)
20 multi_result = matrix_a * matrix_b
21 print("Element-wise Multiplication (A * B):")
22 print(multi_result)
23 # Matrix multiplication (dot product)
24 matrix_multi_result = np.dot(matrix_a, matrix_b)
25 print("A dot B:")
```

< Prev Reset Submit Next >

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ee19f4ab787eca9da7fe Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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.

stacking.py

```
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)
```

Sample Test Cases +

Terminal Test cases < Prev Reset Submit Next >

22:22:09 07-05-2025 ENG IN

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ee9df4ab787eca9da869 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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.

Sample Test Cases +

customS...
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

Terminal Test cases < Prev Reset Submit Next >

5 Search File Firefox Edge Excel S Google Chrome Spotify WhatsApp Microsoft Word 22:21:44 ENG IN 07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ef2cf4ab787eca9da8c0 Verify it's you

Logout

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

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: $A_i \text{ OR } B_i$).

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.

Sample Test Cases

```
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)))
```

Submit Debugger

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774efaa4ab787eca9da91e Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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 +

copyAnd... Submit

```
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)
```

Terminal Test cases Prev Reset Submit Next >

22:21:03 07-05-2025 ENG IN

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774f064f4ab787eca9da9b0 Verify it's you

202401120031@mitaoe.ac.in Support Logout

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:

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

Input Format:

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

Sample Test Cases +

arrayOpe...

```
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)
```

Terminal Test cases < Prev Reset Submit Next >

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.

Sample Test Cases

```
Operations
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
r,c=a.shape
print("Total Students:", r)
# 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:])
```

Terminal Test cases

< Prev Reset Submit Next >

22:20:13
07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667a7e0428220463f6497a3c Verify it's you

202401120031@mitaoe.ac.in Support Logout

4.1.1. Pandas - series creation and manipulation

16:53

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.

seriesMa...

```
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)
```

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >

ENG IN 22:18:20 07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f19c5320ca6bdc2/667aa30728220463f649bb62 Verify it's you

202401120065@mitaoe.ac.in Support Logout

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:

Sample Test Cases +

dataframe...

```
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: "))
df.loc[len(df)] = (new_name, new_age)
# Display the DataFrame after adding a new row
print("After adding a row:\n", df)

# Modifying a row
row_index = int(input("Index of row to modify: "))
new_age = int(input("New age: "))
```

< Prev Reset Submit Next >

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667cf60f2365ba61af5a2ce7 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

4.1.3. Student Information

25:47

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

- Display the first five rows of the data frame.
- Calculate the average age of the students (limit the average age up to 2 decimal places).
- Filter out the students who have a grade above a certain threshold (consider the threshold grade is 'B').

Note:
Refer to the displayed test cases for better understanding.

Sample Test Cases +

studentin... studentdat... Submit Debugger

```
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)
```

Terminal Test cases < Prev Reset Submit Next >

5 Search File Firefox Microsoft Word Microsoft Excel Microsoft Powerpoint Microsoft OneNote Microsoft Edge Spotify Google Chrome WhatsApp Google Sheets 22:16:46 ENG IN 07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e455f19c5320ca6bd19/6773e4e4f19c5320ca6bdce/67860994a50fe56b8dcf8747 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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:

Sample Test Cases +

monthFor... sales_dat... 10:50 Submit Debugger

```
1 """import pandas as pd
2
3 # Prompt the user for the file name
4 file_name = input()
5
6 # Load the data
7 df = pd.read_csv(file_name)
8
9
10 # Find the month with the highest total sales
11 best_month =
12 highest_sales =
13
14 print(f"Best month: {best_month}")
15 print(f"Total sales: ${highest_sales:.2f}")
16 """
17 import pandas as pd
18
19 # Prompt the user for the file name
20 file_name = input()
21
22 # Load the data
23 df = pd.read_csv(file_name)
24 df["Date"] = pd.to_datetime(df["Date"])
25 df["Month"] = df["Date"].dt.strftime("%Y-%m")
```

< Prev Reset Submit Next > ENG IN 22:16:20 07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4f1f9c5320ca6bdce/67861370a50fe56b8dcf9828 Verify it's you

202401120031@mitaoe.ac.in Support Logout

4.2.2. Best Selling Product

04:18

Write a Python program that takes the file name of a CSV file as input, reads the data, and performs the following operations:

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

Sample Data:

```
Date,Product,Quantity,Price,City  
2025-01-01,Product A,5,20,New York  
2025-01-01,Product B,3,15,Los Angeles  
2025-01-02,Product A,7,20,New York  
2025-01-02,Product C,4,30,Chicago  
2025-01-03,Product B,2,15,Chicago  
2025-01-03,Product A,8,20,Los Angeles  
2025-01-04,Product C,6,30,New York  
2025-01-04,Product B,5,15,Los Angeles  
2025-01-05,Product A,3,20,Chicago  
2025-01-05,Product C,10,30,Los Angeles
```

Note:
The data cannot be displayed in the file. You can refer to the sample data provided for insights.

Sample Test Cases +

monthFor... sales_dat... 0 Submit Debugger

```
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}")
```

< Prev Reset Submit Next >

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e455f19c5320ca6bd19/6773e4e4f19c5320ca6bdce/67861553a50fe56b8dcf9a6e Verify it's you

202401120031@mitaoe.ac.in Support Logout

4.2.3. City that Sold the Most Products

Write a Python program that takes the file name of a CSV file as input, reads the data, and performs the following operations:

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

Sample Data:

```
Date,Product,Quantity,Price,City
2025-01-01,Product A,5,20,New York
2025-01-01,Product B,3,15,Los Angeles
2025-01-02,Product A,7,20,New York
2025-01-02,Product C,4,30,Chicago
2025-01-03,Product B,2,15,Chicago
2025-01-03,Product A,8,20,Los Angeles
2025-01-04,Product C,6,30,New York
2025-01-04,Product B,5,15,Los Angeles
2025-01-05,Product A,3,20,Chicago
2025-01-05,Product C,10,30,Los Angeles
```

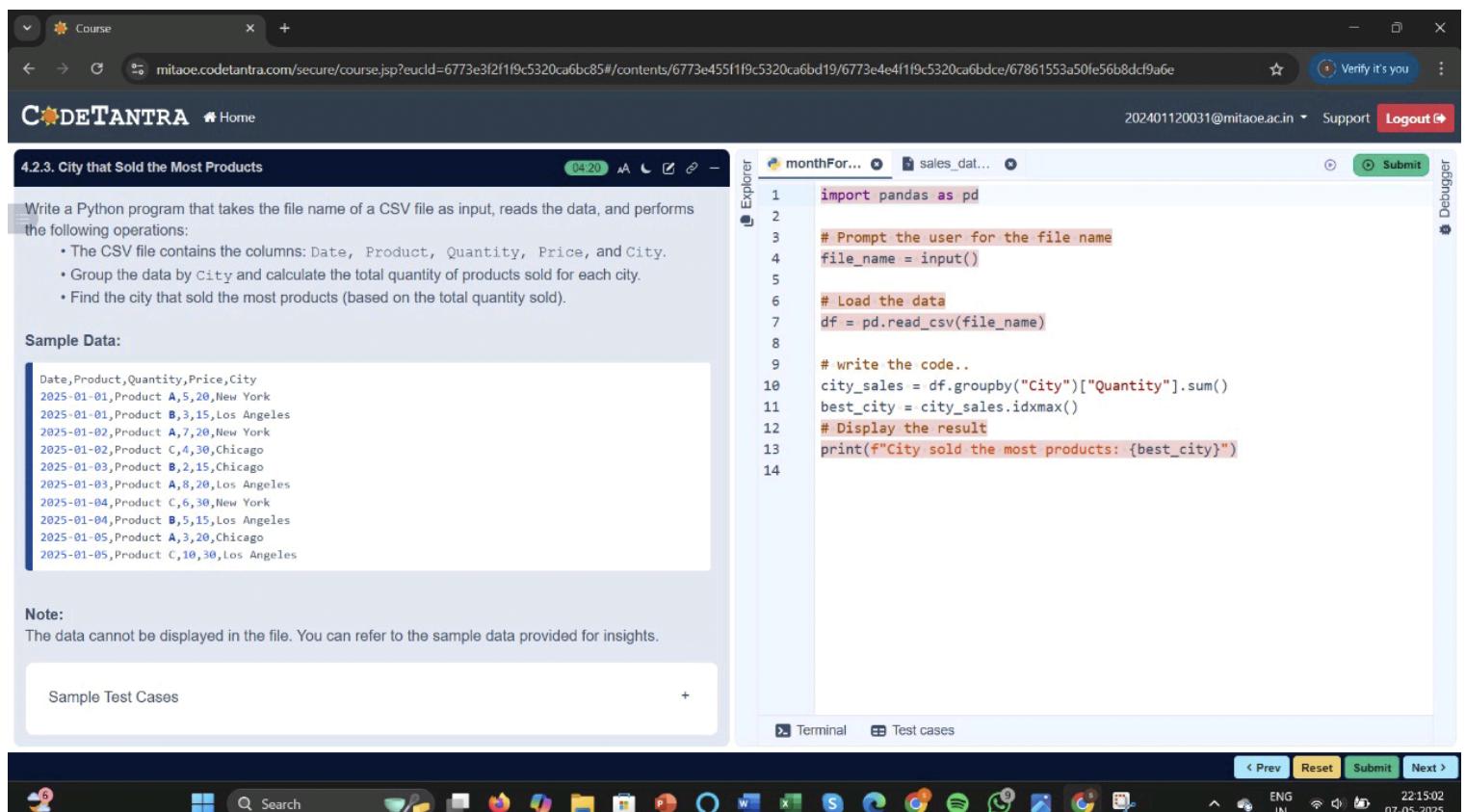
Note:
The data cannot be displayed in the file. You can refer to the sample data provided for insights.

Sample Test Cases +

monthFor... sales_dat... 0 Submit Debugger

```
import pandas as pd
# Prompt the user for the file name
file_name = input()
# Load the data
df = pd.read_csv(file_name)
# write the code..
city_sales = df.groupby("City")["Quantity"].sum()
best_city = city_sales.idxmax()
# Display the result
print(f"City sold the most products: {best_city}")
```

< Prev Reset Submit Next >



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:

Sample Test Cases

```
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
grouped = df.groupby("Date")["Product"].apply(list)

product_pair = []
for products in grouped:
    if len(products) > 1:
        product_pair.extend(combinations(sorted(products), 2))

pair_counts = Counter(product_pair)

if pair_counts:
    max_count = max(pair_counts.values())
    most_common_pairs = [pair for pair, count in pair_counts.items() if
```

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed
-------------------------	------------------	------------	----------	-----	-----	-----------	-----------	------------	----------	-----------	------------------

Sample Test Cases

```
titanicDat...  
1 import pandas as pd  
2 import numpy as np  
3  
4 # Load the Titanic dataset  
5 data = pd.read_csv('Titanic-Dataset.csv')  
6  
7 # 1. Display the first 5 rows of the dataset  
8 print(data.head())  
9  
10 # 2. Display the last 5 rows of the dataset  
11 print(data.tail())  
12  
13 # 3. Get the shape of the dataset  
14 print(data.shape)  
15  
16 # 4. Get a summary of the dataset (info)  
17 print(data.info())  
18  
19 # 5. Get basic statistics of the dataset  
20 print(data.describe())  
21  
22 # 6. Check for missing values  
23 print(data.isnull().sum())  
24  
25 # 7. Fill missing values in the 'Age' column with the median age
```

Terminal Test cases

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Fare	Cab in	Em bark ed

Sample Test Cases

```
titanicDat...  
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'], prefix='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)
```

Terminal Test cases

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4e9f19c5320ca6bdd2/675a8d2c09f9fa022faa9dc0 Verify it's you

CODETANTRA Home 202401120065@mitaoe.ac.in Support Logout

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.

```
stackedplot.py
1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 # Data for Months and Temperature for three cities
5 v data = {
6     'Month': ['January', 'February', 'March', 'April', 'May', 'June',
7               'July', 'August', 'September', 'October', 'November', 'December'],
8     'City_A_Temperature': [5, 7, 10, 13, 17, 20, 22, 21, 18, 12, 8, 6],
9     'City_B_Temperature': [2, 3, 5, 6, 10, 14, 16, 17, 12, 9, 5, 3],
10    'City_C_Temperature': [3, 4, 6, 8, 9, 12, 15, 14, 10, 7, 4, 2]
11 }
12
13 # Write your code...
14 df = pd.DataFrame(data)
15
16 plt.figure()
17 plt.stackplot(df['Month'],df['City_A_Temperature'],df['City_B_Temperature'],
18                df['City_C_Temperature'],labels = ['City A', 'City B', 'City C'],
19                alpha = 1)
20
21 plt.xlabel("Month")
22 plt.ylabel("Temperature")
23 plt.title("Temperature Variation")
24 plt.show()
```

Sample Test Cases +

< Prev Reset Submit Next >

ENG IN 22:10:21 07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f19c5320ca6bdd5/67e28fb27c0b912d860e7db4 Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

Sample Data:

Sample Test Cases +

```

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')

```

< Prev Reset Submit Next >

ENG IN 22:02:04 07-05-2025

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,

Pas sen gerl d	Sur vive d	Pcia ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

Sample Test Cases

```
Explorer titanicDat...
1 import pandas as pd
2 import numpy as np
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
7
8
9 # 1. Get the number of survivors by gender
10 print(data[data['Survived']==1]['Sex'].value_counts())
11
12 # 2. Get the number of non-survivors by gender
13 print(data[data['Survived']==0]['Sex'].value_counts())
14
15 # 3. Get the number of survivors by embarked location
16 print(data[data['Survived']==1]
17 ['Embarked_S'].value_counts())
18
19 # 4. Get the number of non-survivors by embarked location
20 print(data[data['Survived']==0]
21 ['Embarked_S'].value_counts())
22
23 # 5. Calculate the percentage of children (Age < 18) who survived
24 print(data[data['Age']<18]['Survived'].mean())
25
```

Terminal Test cases

Submit



< Prev Reset Submit Next >

ENG IN

22:11:42

07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e291be7c0b912d860e876f Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

Sample Data:

Sample Test Cases +

```

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')

```

< Prev Reset Submit Next >

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e28d6e7c0b912d860e75ad Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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.

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

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
```

Sample Test Cases

Code Editor

```
BarPlotOf...
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()
```

Terminal Test cases

Prev Reset Submit Next < Prev Reset Submit Next > ENG IN 22:02:32 07-05-2025

4.2.7. Titanic Dataset Analysis and Data Cleaning - 3

10:12 A ⚡ ⚡

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

1. 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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Fare	Cab in	Em bark ed

Sample Test Cases

titanicDat...

```
1 import pandas as pd
2 import numpy as np
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6 data['FamilySize'] = data['SibSp'] + data['Parch']
7 data['IsAlone'] = np.where(data['FamilySize'] > 0, 0, 1)
8 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
9
10 # 1. Calculate the survival rate by class
11 print(data.groupby('Pclass')['Survived'].mean())
12
13 # 2. Calculate the survival rate by embarked location
14 print(data.groupby('Embarked_S')['Survived'].mean())
15
16 # 3. Calculate the survival rate by family size
17 print(data.groupby('FamilySize')['Survived'].mean())
18
19 # 4. Calculate the survival rate by being alone
20 print(data.groupby('IsAlone')['Survived'].mean())
21
22 # 5. Get the average fare by class
23 print(data.groupby('Pclass')['Fare'].mean())
24
25 # 6. Get the average age by class
```

Terminal Test cases

Submit

Debugger

< Prev Reset Next >

22:12:20
ENG IN
07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/677cbdd52372a2277b32ab1c Verify it's you

202401120065@mitaoe.ac.in Support Logout

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:

Sample Test Cases +

```
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..
count=df['Pclass'].value_counts().sort_index()
count_gen=df['Gender'].value_counts()
count_s=df['Survived'].value_counts().sort_index()

axes[0,0].bar(count.index,count.values,color=['skyblue'])
axes[0,0].set_title('Passenger Class Distribution')
axes[0,0].set_xlabel('Pclass')
axes[0,0].set_ylabel('Count')

axes[0,1].pie(count_gen.values,labels=['male','female'],colors=['lightblue','lightcoral'],autopct='%.1f%%')
axes[0,1].set_title('Gender Distribution')

axes[1,0].hist(df['Age'],color=['lightgreen'],edgecolor='black',bins=8)
axes[1,0].set_title('Age Distribution')
```

< Prev Reset Submit Next >

Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2ff9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e28a397c0b912d860e689c

Verify it's you

Logout

CODETANTRA Home

202401120031@mitaoe.ac.in Support Logout

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

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
```

Sample Test Cases

Histogram...

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Histogram
17 plt.hist(data['Age'], bins=30, edgecolor='k')
18 plt.xlabel('Age')
19 plt.ylabel('Frequency')
20 plt.title('Age Distribution')
21 plt.show()

```

Terminal Test cases

< Prev Reset Submit Next >

ENG IN 22:03:02 07-05-2025

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e45af19c5320ca6bd1d/6773e4f1f19c5320ca6bdd5/67e29db57c0b912d860eb45 Verify it's you

CODE TANTRA Home 202401120031@mitaoe.ac.in Support Logout

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,311317,7.25,S
```

Sample Test Cases +

BoxPlotF...

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Box Plot for Age by Survived
17
18
19 plt.figure(figsize=(8, 6))
20 data.boxplot(column='Age', by='Survived')
21 plt.suptitle('')
22 plt.title('Age by Survival')
23 plt.xlabel('Survived')
24 plt.ylabel('Age')
25 plt.show()

```

Terminal Test cases < Prev Reset Submit Next >

21:57:15 07-05-2025 ENG IN

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Fare	Cab in	Em bark ed

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
```

Sample Test Cases

AgeFareS...

```
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()
plt.scatter(data['Age'],data['Fare'])
plt.title('Age vs. Fare')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```

Terminal Test cases

< Prev Reset Submit Next >



Search



ENG IN 21:56:10 07-05-2025

5.2.11. Scatter Plot for Age vs. Fare by Survived

20:06

Submit

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

Sample Data:

Sample Test Cases +

AgeFareS...

Explorer

```
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
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()
```

Terminal Test cases

Submit

Debugger

< Prev Reset Submit

21:53:09
07-05-2025



Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e29

CODETANTRA Home

5.2.6. Bar Plot for Survival by Embarked

17:20

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed
-------------------------	------------------	------------	----------	-----	-----	-----------	-----------	------------	----------	-----------	------------------

```

import pandas as pd
import matplotlib.pyplot as plt

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

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

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

# Write your code here for the following steps
grouped = data.groupby(['Embarked', 'Survived']).value_counts()

```

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e297bf7c0b912d860e9c81 Verify it's you

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

Sample Test Cases +

```

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 by Embarked')
plt.xlabel('Embarked')
plt.ylabel('Count')

```

< Prev Reset Submit Next >

Course mitaoe.codetantra.com/secure/course.jsp?eucid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e2a1727c0b912d860eb39b Verify it's you

CODETANTRA Home 202401120031@mitaoe.ac.in Support Logout

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

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
```

Sample Test Cases +

BoxPlotF...

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Box Plot for Fare by Pclass
17
18
19 plt.figure(figsize=(8, 6))
20 data.boxplot(column='Fare', by='Pclass')
21 plt.suptitle('')
22 plt.title('Fare by Pclass')
23 plt.xlabel('Pclass')
24 plt.ylabel('Fare')
25 plt.show()

```

Terminal Test cases < Prev Reset Submit Next >

Search

Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f19c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f19c5320ca6bdd5/67e2992e7c0b912d860ea071 Verify it's you

CODE TANTRA Home 202401120031@mitaoe.ac.in Support Logout

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,

Pas sen gerl d	Sur vive d	Pcla ss	Na me	Sex	Age	Sib Sp	Par ch	Tick et	Far e	Cab in	Em bark ed

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
```

Sample Test Cases +

BoxPlotF...

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Box Plot for Age by Pclass
17
18
19 plt.figure(figsize=(8, 6))
20 data.boxplot(column='Age', by='Pclass')
21 plt.suptitle('')
22 plt.title('Age by Pclass')
23 plt.xlabel('Pclass')
24 plt.ylabel('Age')
25 plt.show()

```

< Prev Reset Submit Next >