

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773e9c2f4ab787eca9d09e4

CODETANTRA Home

202401090058@mitaoe.ac.in Support Logout

1.1.1. Calculate Momentum

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 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...

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

Terminal Test cases

Thunderstorm w... in effect ENG IN 13:37 07-05-2025 < Prev Reset Submit Next >

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6774c276f4ab787eca9d88e3

CODETANTRA Home 202401090058@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 >

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/678dd2f0fa4f5e20f6fdct1

CODETANTRA Home 202401090058@mitaoe.ac.in Support Logout

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 multiplica...

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

Terminal Test cases < Prev Reset Submit Next >

Thunderstorm w... in effect ENG IN 13:38 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6773f211f4ab787eca9d15d0

CODETANTRA Home 202401090058@mitaoe.ac.in Support Logout

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".

passorFail.py

```

1 n = int(input())
2 marks = list(map(int,input().split(" ")))
3 if all (mark>=40 for mark in marks):
4     per = sum(marks)/n
5     #print(f"Aggregate Percentage: {per:.2f}")
6     if per>=75:
7         print("Grade: Distinction")
8     elif 60<=per<75:
9         print("Grade: First Division")
10    elif 50<=per<60:
11        print ("Grade: Second Division")
12    elif 40<=per<50:
13        print ("Grade: Third Division")
14    elif any (mark<40 for mark in marks):
15        print("Fail")
16

```

Sample Test Cases +

Terminal Test cases

Thunderstorm w... in effect ENG IN 13:38 07-05-2025 < Prev Reset Submit Next >

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6774cc86f4ab787eca9d8f99

CODETANTRA Home 202401090058@mitaoe.ac.in Support Logout

1.2.4. Pattern - 2

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

```
1 n=int(input())
2 i=1
3 v while (i<=n):
4     j=1
5     v while (j<=i):
6         print(j,end=" ")
7         j=j+1
8     print()
9     i=i+1
```

Terminal Test cases < Prev Reset Submit Next >

BAJFINANCE +126% Search ENG IN 13:38 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e44bf1f9c5320ca6bc8/6773e49ef1f9c5320ca6bd7d/667bbd7f01060f3d690f3f83

CODETANTRA Home 202401090058@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).

```
listOps.py
1  def menu_driven_program():
2      lst = [] # Initialize an empty list
3
4      while True:
5          print("1. Add")
6          print("2. Remove")
7          print("3. Display")
8          print("4. Quit")
9
10         try:
11             choice = int(input("Enter choice: "))
12         except ValueError:
13             print("Invalid choice")
14             continue
15
16         if choice == 1:
17             num = int(input("Integer: "))
18             lst.append(num)
19             print(f"List after adding: {lst}")
20
21         elif choice == 2:
22             if lst:
23                 num = int(input("Integer: "))
24                 if num in lst:
25                     lst.remove(num)
26                     print(f"List after removing: {lst}")
27                 else:
28                     print("Element not found")
29
30             elif choice == 3:
31                 if lst:
32                     print(lst)
33                     print("List is empty")
34             elif choice == 4:
35                 break
36             else:
37                 print("Invalid choice")
38
39 # Run the menu-driven program
40 menu_driven_program()
```

Sample Test Cases

BAJFINANCE +126% Search ENG IN 13:39 07-05-2025

The screenshot shows a browser window with the URL <https://mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bc8/6773e49ef1f9c5320ca6bd7d/677bb1ca63fd602f55ad634f>. The page title is "Course". The main content area is titled "2.1.2. Dictionary Operations". It contains a list of requirements for a Python program and a sample test case. On the right, there is an "Editor" tab showing a Python script named "dictOpera...". The script performs various operations on a dictionary, including creation, addition, update, retrieval, get(), and deletion. The code uses raw input for all user interactions. The "Test cases" section at the bottom contains a single test case: "Key: 1, Value: 100". The "Submit" button is visible at the top right.

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

```
my_dict = {}
print("Empty Dictionary:", my_dict)
n = int(input("Number of items: "))
size = n
for _ in range(n):
    key = input("key: ")
    value = input("value: ")
    my_dict[key] = value
print("Dictionary:", my_dict)
key_to_update = input("Enter the key to update: ")
if key_to_update in my_dict:
    new_value = input("Enter the new value: ")
    my_dict[key_to_update] = new_value
    print("Value updated")
else:
    print("Key not found")
key_to_access = input("Enter the key to retrieve: ")
if key_to_access in my_dict:
    print(f"Key: {key_to_access}, Value: {my_dict[key_to_access]}")
else:
    print("Key not found")
key_to_get = input("Enter the key to get using the get() method: ")
value = my_dict.get(key_to_get,"Key not found")
if value != "Key not found":
    print(f"Key: {key_to_get}, Value: {value}")
else:
    print(value)
key_to_delete = input("Enter the key to delete: ")
if key_to_delete in my_dict:
    del my_dict[key_to_delete]
    print("Deleted")
else:
    print("Key not found")
print("Updated Dictionary:", my_dict)
```

< Prev Reset Submit Next >

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2ff9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bc8/6773e4b0ff9c5320ca6bd85/6774d0abf4ab787eca9d9230

CODETANTRA Home 202401090058@mitaoe.ac.in Support Logout

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.

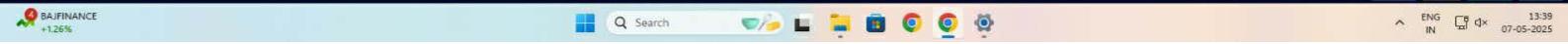
Input Format:
The first line of input will contain 11 integers, each representing the height of a player (in centimeters), each separated by a space.

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

Sample Test Cases +

Explorer captainof... 1 heights = list(map(int,input().split(" "))) 2 3 captain = max(heights) 4 5 print(captain) 6

Terminal Test cases < Prev Reset Submit Next >



New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ee9df4ab787eca9da869

202401090058@mitaoe.ac.in Support Logout

3.2.3. Numpy: Custom Sequence Generation

00:44 A C E

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.

customS...

```
1 import numpy as np
2
3 # Take user input for the start, stop, and step of the sequence
4 start = int(input())
5 stop = int(input())
6 step = int(input())
7
8 # Generate the sequence using np.arange()
9 a = np.arange(start, stop, step)
10 # Print the generated sequence
11 print(a)
```

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >



New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bcd2/667cf60f2365ba61af5a2ce7

CODETANTRA Home

202401090058@mitaoe.ac.in Support Logout

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.

Exorer studentin... studentdat...

```
1 import pandas as pd
2
3 # Read the text file into a DataFrame
4 file = input()
5 data = pd.read_csv(file, sep="\s+", header=None, names=["Name", "Age", "Grade"])
6 print("First five rows:")
7 print(data.head(5))
8 # write your code here..
9 age=round(data['Age'].mean(),2)
10 print("Average age:",age)
11 print("Students with a grade up to B")
12 df=pd.DataFrame(data)
13 a=df[df['Grade']<='B']
14 print(a)
15
```

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >

BAJFINANCE +1.26% Search ENG IN 13:41 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2ff9c5320ca6bc85#contents/6773e455ff9c5320ca6bd19/6773e4e4ff9c5320ca6bdce/6708eeaa89ff79b6ea9e3ece1

CODETANTRA Home

202401090058@mitaoe.ac.in Support Logout

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.

Sample Test Cases +

File Explorer frequent... sales_dat...

```

1 import pandas as pd
2 from itertools import combinations
3 from collections import Counter
4
5 # Prompt user to input the file name
6 file_name = input()
7
8 # Read data from the specified csv file
9 df = pd.read_csv(file_name)
10
11 # write the code
12 grouped=df.groupby('Date')[['Product']].apply(list)
13 pair_counter=Counter()
14 # Output the most frequent product pairs
15 for products in grouped:
16     products=sorted(products)
17     pairs=combinations(products,2)
18     pair_counter.update(pairs)
19 max_count=max(pair_counter.values()) if pair_counter else 0
20 most_frequent_pairs=[pair for pair,count in pair_counter.items() if count==max_count]
21 most_frequent_pairs.sort()
22 for pair in most_frequent_pairs:
23     print(f'{pair[0]} and {pair[1]}: {max_count} times')

```

Terminal Test cases < Prev Reset Submit Next >

BAJFINNCE +126% ENG IN 13:42 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e378a17028471d94dad0a6

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.

- Create a new column 'IsAlone' which is 1 if the passenger is alone (FamilySize = 0), otherwise 0.
- Convert the 'Sex' column to numeric values (male: 0, female: 1).
- One-hot encode the 'Embarked' column, dropping the first category.
- Get the mean age of passengers.
- Get the median fare of passengers.
- Get the number of passengers by class.
- Get the number of passengers by gender.
- Get the number of passengers by survival status.
- Calculate the survival rate of passengers.
- 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,22,1,0,313785,7.25,0,C
2,1,1,"Allen, Mr. William Henry",male,35,1,0,313478,72.3833,1,1,C
3,1,3,"Haesemann, Miss. Laina",female,28,0,0,313502,7.925,0,C
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,35,0,0,313459,8.05,S
6,0,3,"Moran, Mr. James",male,30,0,0,313877,0.5833,0
7,0,1,"McCarthy, Mr. Timothy J.",male,54,0,0,17481,51.8625,E46,S
8,0,1,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.3583,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achen)",female,24,1,0,237736,30.0708,C

```

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

Sample Test Cases

```

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)

# 9. Calculate the survival rate
survival_rate = data['Survived'].mean()
print(survival_rate)

# 10. Calculate the survival rate by gender
survival_rate_by_gender = data.groupby('Sex')['Survived'].mean()
print(survival_rate_by_gender)

```

Terminal Test cases

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.

- Get the number of survivors by gender (Sex).
- Get the number of non-survivors by gender (Sex).
- Get the number of survivors by embarkation location (Embarked_S).
- Get the number of non-survivors by embarkation location (Embarked_S).
- Calculate the percentage of children (Age < 18) who survived.
- Calculate the percentage of adults (Age >= 18) who survived.
- Get the median age of survivors.
- Get the median age of non-survivors.
- Get the median fare of survivors.
- 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,22,1,0,313359,7.25,5
2,1,1,"Cumings, Mrs. John Bradley (Florence Brigitte Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,313470,113.2282,75.925,5
4,1,3,"Leino, Mr. Sven Gustav Rosander",male,35,0,0,313471,113.2282,75.925,5
5,0,3,"Allen, Mr. William Henry",male,35,0,0,313459,8.65,5
6,0,3,"Moran, Mr. James",male,30,0,0,313087,41.5882,0,Q
7,0,1,"McCarthy, Mr. Timothy J.",male,54,0,0,17485,51.8625,646,5
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,5
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1533,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achen)",female,14,1,0,237736,30.0708,,C

```

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

Sample Test Cases

```

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

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

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

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

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

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

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

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

```

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e28d6e70b912d860e75ad

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,
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

Exorer BarPlotOf...

```

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 Bar Plot for Survival Rate
17 survival_counts=data['Survived'].value_counts()
18 survival_counts.plot(kind="bar")
19 plt.title("Survival Count")
20 plt.xlabel("Survived")
21 plt.ylabel("Count")
22 plt.show()
23

```

Terminal Test cases < Prev Reset Submit Next >

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e29db57c0b912d860eab45

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,22,1,0,3/1171.725,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,
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,35,0,0,373456,8.05,,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

BoxPlotForSurvived.py

```

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()

```

Terminal Test cases

< Prev Reset Submit Next >

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e2a5387c0b912d860eba29

CODE TANTRA Home

202401090058@mitaoe.ac.in Support Logout

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,22,1,0,21171,7.25,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,S
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,27,0,2,347742,11.333,S
```

Sample Test Cases

Exorer

```
AgeFareS...
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 Scatter Plot for Age vs. Fare by Survived
17 colors=data["Survived"].map({0: "#FF0000",1: "#0000FF"})
18 plt.scatter(data["Age"],data["Fare"],color=colors)
19 plt.title("Age vs. Fare by Survival")
20 plt.xlabel("Age")
21 plt.ylabel("Fare")
22 plt.show()
23
24
```

Terminal Test cases

BAJFINANCE +126% ENG IN 13:43 07-05-2025 < Prev Reset Submit

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e2a2977c0b912d860eb621

CODETANTRA Home 202401090058@mitaoe.ac.in Support Logout

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,22,1,0,0/21171,7.25,,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 Heatt (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373458,8.05,,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,2,3,1,349090,21.075,,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,14,1,0,237736,30.0788,,C
```

Sample Test Cases

Exorer AgeFareS... 04:49 A ⚡ Submit Debugger

```

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 plt.figure(figsize=(6.4,4.8))
18 plt.scatter(data['Age'],data['Fare'])
19 plt.title("Age vs. Fare")
20 plt.xlabel('Age')
21 plt.ylabel('Fare')
22 plt.show()
23

```

< Prev Reset Submit Next >

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e2a1727c0b912d860eb39b

CODETANTRA Home 202401090058@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,

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,22,1,0,32,1,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,
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,35,0,0,373456,8.05,,S
6,0,3,"Moran, Mr. James",male,0,0,330877,8.4563,,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 +

BoxPlotF... Exorer 02:50

```

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 plt.figure(figsize=(8,6))
18 data.boxplot(column='Fare', by="Pclass")
19 plt.suptitle('')
20 plt.title("Fare by Pclass")
21 plt.xlabel("Pclass")
22 plt.ylabel("Fare")
23 plt.show()

```

Terminal Test cases < Prev Reset Submit Next >

New repository Course 03:25

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e2992e70b912d860ea071

CODETANTRA Home 202401090058@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,

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,22,1,0,3/1171.725,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,
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,35,0,0,373456,8.05,,S
6,0,3,"Moran, Mr. James",male,0,0,330877,8.4563,,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 +

BoxPlotF... 03:25

```

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 plt.figure(figsize=(8,6))
18 data.boxplot(column='Age', by="Pclass")
19 plt.suptitle('')
20 plt.title("Age by Pclass")
21 plt.xlabel('Pclass')
22 plt.ylabel('Age')
23 plt.show()
    
```

Terminal Test cases < Prev Reset Submit Next >

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45aff1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e297bf7c0b912d860e9c81

CODETANTRA Home 202401090058@mitaoe.ac.in Support Logout

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,

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

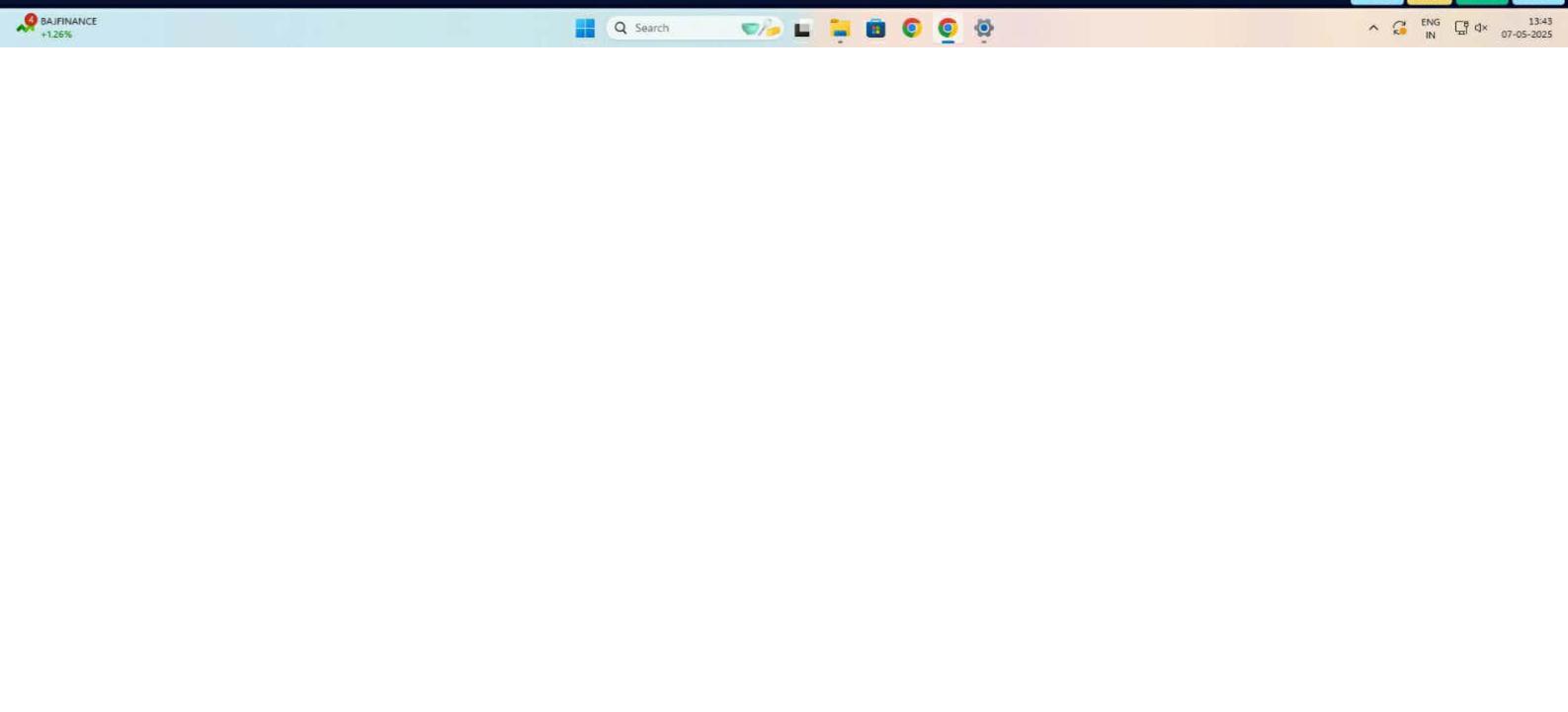
Exorer BarPlotOf...

```

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 Bar Plot for Survival by Embarked
17 grouped=data.groupby('Embarked_Q')['Survived'].value_counts().unstack().fillna(0)
18 grouped.columns=['Not Survived','Survived']
19 grouped.plot(kind='bar',stacked=True)
20 plt.title("Survival by Embarked")
21 plt.xlabel("Embarked")
22 plt.ylabel("Count")
23 plt.legend(title=None)
24 plt.show()

```

Terminal Test cases < Prev Reset Submit Next >



New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45aff1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e291be7c0b912d860e876f

CODETANTRA Home 202401090058@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:

- 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()`.
- Use a **stacked bar chart** to display the survival counts.
- Add the title "**Survival by Pclass**" to the chart.
- Label the x-axis as '**Pclass**' and the y-axis as '**Count**'.
- The legend should indicate '**Not Survived**' and '**Survived**'.

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,"Helikkinen, 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. Gustaf Larsson",male,2,0,1,340000,31.275,C
```

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")
plt.legend(title=None)
plt.show()
```

Terminal Test cases < Prev Reset Submit Next >

BAJFINANCE +126% ENG IN 13:43 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e28fb27c0b912d860e7db4

CODETANTRA Home 202401090058@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:

- 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.
- Use a **stacked bar chart** to display the survival counts.
- Add the title "**Survival by Gender**" to the chart.
- Label the x-axis as '**Gender**' and the y-axis as '**Count**'.
- The legend should indicate '**Not Survived**' and '**Survived**'.

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,"Helikkinen, 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. Gustaf Larsson",male,2,0,1,340000,31.075,C
```

Sample Test Cases

Exorer BarPlotOf...

```

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 Bar Plot for Survival by Gender
17 survival_by_gender = data.groupby("Sex")['Survived'].value_counts().unstack().fillna(0)
18 survival_by_gender.columns = ['Not Survived', 'Survived']
19 survival_by_gender.index = [0, 1]
20 survival_by_gender.plot(kind="bar", stacked=True)
21 plt.title("Survival by Gender")
22 plt.xlabel("Gender")
23 plt.ylabel("Count")
24 plt.legend(title=None)
25 plt.show()

```

Terminal Test cases < Prev Reset Submit Next >

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f9c5320ca6bdd5/67e28a397c0b912d860e689c

CODETANTRA # Home 202401 110% Reset 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,

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,22,1,0,3/12345,21.075,,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 +

Exorer Histogram... 1 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()

Terminal Test cases < Prev Reset Submit Next >

BAJFINANCE +126% ENG IN 13:43 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45aff1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/677cbdd52372a2277b32ab1c

CODETANTRA Home 202401090058@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:

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.
 - Include percentages on the slices (use autopct="%1.1f%%").
 - Title the plot as "Gender Distribution".

Histogram (Age Distribution):
Create a histogram to visualize the distribution of passengers' ages.
 - Use lightgreen for the bars with black edges (edgecolor = 'black').
 - Set the number of bins to 8 for the histogram.
 - Title the plot as "Age Distribution".
 - Label the x-axis as "Age" and the y-axis as "Frequency".

Bar Plot (Survival Count):
Create a bar plot to show the count of passengers who survived and those who did not, based on the Survived column.
 - Use the colors lightblue for survivors (1) and lightcoral for non-survivors (0).
 - Title the plot as "Survival Count".
 - Label the x-axis as "Survived (0 = No, 1 = Yes)" and the y-axis as "Count".

Scatter Plot (Fare vs Age):

```

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...
pclass_counts=df['Pclass'].value_counts()
axes[0,0].bar(x=pclass_counts.index,height=pclass_counts.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(df['Gender'].value_counts(),labels=df['Gender'].value_counts().index,autopct="%1.1f%%",colors=['lightblue','lightcoral'])
axes[0,1].set_title('Gender Distribution')

axes[1,0].hist(df['Age'].dropna(),bins=8,color="lightgreen",edgecolor="black")
axes[1,0].set_title("Age Distribution")
axes[1,0].set_xlabel("Age")
axes[1,0].set_ylabel("Frequency")

axes[1,1].bar(df["Survived"].value_counts().index,df["Survived"].value_counts(),color=["lightblue","lightcoral"])
axes[1,1].set_title("Survival Count")
axes[1,1].set_xlabel("Survived(No=0,Yes=1)")
axes[1,1].set_ylabel("Count")

axes[2,0].scatter(df['Age'],df['Fare'],color='orange')
axes[2,0].set_title('Fare vs Age')
axes[2,0].set_xlabel("Age")
axes[2,0].set_ylabel("Fare")

plt.tight_layout()
plt.show()

```

Sample Test Cases

BAJFINANCE +126% Search ENG IN 13:42 07-05-2025

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e45aff1f9c5320ca6bd1d/6773e4e9ff1f9c5320ca6bdd2/675a8d2c09f9fa022faa9dc0

CODETANTRA Home

202401090058@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.

stackedpl...

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 # Data for Months and Temperature for three cities
5 data = {
6     'Month': ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October',
7               '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, 13, 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 plt.stackplot(df['Month'],df['City_A_Temperature'],df['City_B_Temperature'],df['City_C_Temperature'])
16 plt.title("Temperature Variation")
17 plt.xlabel("Month")
18 plt.ylabel("Temperature")
19 plt.close("2")
20 plt.show()
```

Sample Test Cases

Terminal Test cases

< Prev Reset Submit Next >

BAJFINANCE +12.6%

Search

ENG IN 13:42 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e3805d7028471d94dae951

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,22,1,0,0,313382,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Brigitte Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,0,313400,7.925,,S
4,1,3,"Leikin, Mr. Karl Georg",male,35,0,0,0,313400,7.925,,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,0,313459,8.05,,S
6,0,3,"Moran, Mr. James",male,30,0,0,0,313877,41.5833,Q
7,0,1,"McCarthy, Mr. Timothy J.",male,54,0,0,0,17485,51.8625,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,27,0,2,347742,11.1533,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achen)",female,14,1,0,237736,30.0708,,C

```

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

Sample Test Cases

```

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())

# 10. Get the number of non-survivors by class
print(data[data['Survived'] == 0]['Pclass'].value_counts())

```

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e2b9309025661df3aa574f

CodeTANTRA Home 202401090058@mitaoe.ac.in Support Logout

4.2.5. Titanic Dataset Analysis and Data Cleaning

You are provided with the Titanic dataset containing information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset. For each question, perform necessary data cleaning, transformations, and calculations as required.

- Display the first 5 rows of the dataset.
- Display the last 5 rows of the dataset.
- Get the shape of the dataset (number of rows and columns).
- Get a summary of the dataset (using .info()).
- Get basic statistics (mean, standard deviation, etc.) of the dataset using .describe().
- Check for missing values and display the count of missing values for each column.
- Fill missing values in the 'Age' column with the median age.
- Fill missing values in the 'Embarked' column with the most frequent value (mode).
- Drop the 'Cabin' column due to many missing values.
- 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,22,1,0,21171,7.25,,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,"McCarthy, Mr. Timothy J",male,35,0,0,330877,8.4583,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gusta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmine Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achen)",female,14,1,0,237736,30.0708,,C
```

Sample Test Cases

Code Editor

```
titanicData...
1 import pandas as pd
2 import numpy as np
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
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
26 median_age = data['Age'].median()
27 data['Age'].fillna(median_age, inplace=True)
28 # 8. Fill missing values in the 'Embarked' column with the mode
29 mode_embarked = data['Embarked'].mode()[0]
30 data['Embarked'].fillna(mode_embarked, inplace=True)
31 # 9. Drop the 'Cabin' column due to many missing values
32 data.drop('Cabin', axis=1, inplace=True)
33
34 # 10. Create a new column 'FamilySize' by adding 'SibSp' and 'Parch'
35 data['FamilySize'] = data['SibSp'] + data['Parch']
36
```

Terminal Test cases

BAJFINANCE +126% ENG IN 13:42 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67061553a50fe56b8dcf9a6e

202401090058@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 +

File Explorer monthFor... sales_dat... 03:29 A ⚡

```
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}")
```

Terminal Test cases < Prev Reset Submit Next >

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67061370a50fe56b8dcf9828

202401090058@mitaoe.ac.in Support Logout

CODETANTRA Home

4.2.2. Best Selling Product

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

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

Sample Data:

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

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

Sample Test Cases +

Excoer monthFor... sales_dat...
1 import pandas as pd
2
3 # Prompt the user for the file name
4 file_name = input()
5
6 # Load the data
7 df = pd.read_csv(file_name)
8
9
10 # Find the product with the highest total quantity sold
11 product_sales = df.groupby("Product")["Quantity"].sum()
12 best_product = product_sales.idxmax()
13 highest_quantity = product_sales.max()
14
15 # Display the result
16 print(f"Best selling product: {best_product}")
17 print(f"Total quantity sold: {highest_quantity}")
18

Terminal Test cases < Prev Reset Submit Next >

BAJFINANCE +126% Search ENG IN 13:42 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67860994a50fe56b8dcf8747

202401090058@mitaoe.ac.in Support Logout

CODETANTRA Home

4.2.1. Month with the Highest Total Sales

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

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

Sample Data:

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

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

Sample Test Cases +

Exorer monthFor... sales_dat... 08:52

```
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 df['Date'] = pd.to_datetime(df['Date'])
9
10 df['Month'] = df['Date'].dt.to_period('M')
11
12 df['Total_Sales'] = df['Quantity'] * df['Price']
13
14 monthly_sales = df.groupby('Month')['Total_Sales'].sum()
15
16 # Find the month with the highest total sales
17 best_month = monthly_sales.idxmax()
18 highest_sales = monthly_sales.max()
19
20 print(f"Best month: {best_month}")
21 print(f"Total sales: ${highest_sales:.2f}")
```

Terminal Test cases < Prev Reset Submit Next >

BAJFINNCE +126% Search ENG IN 13:41 07-05-2025

New repository Course

[mitaoe.codetantra.com](#) / secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bd2/667aa30728220463f649bb62

202401090058@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:

- Add a column Gender with values taken from the user.
- Display the DataFrame after adding the new column.

Modify a column:

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

Delete a column:

- Remove the Age column.
- Display the DataFrame after deleting the column.

Sample Test Cases

```

1 import pandas as pd
2
3 # Provided dictionary of lists
4 data = {
5     "Name": ["Alice", "Bob", "Charlie"],
6     "Age": [25, 30, 35],
7 }
8
9 # Convert the dictionary to a DataFrame
10 df = pd.DataFrame(data)
11
12 # Display the original DataFrame
13 print("Original DataFrame:")
14 print(df)
15
16 # Adding a new row
17 new_name=input("New name: ")
18 new_age=int(input("New age: "))
19 new_row={"Name": new_name,"Age":new_age}
20 df=pd.concat([df,pd.DataFrame([new_row])],ignore_index=True)
21 # Display the DataFrame after adding a new row
22 print("After adding a row:")
23 print(df)
24
25 # Modifying a row
26 modify_index=int(input("Index of row to modify: "))
27 new_age_mod=int(input("New age: "))
28 df.loc[modify_index,"Age"]=new_age_mod
29 # Display the DataFrame after modifying a row
30 print("After modifying a row:")
31 print(df)
32
33 # Deleting a row
34 delete_index=int(input("Index of row to delete: "))
35 df=df.drop(delete_index).reset_index(drop=True)
36 # Display the DataFrame after deleting a row
37 print("After deleting a row:")
38 print(df)
39
40 # Adding a new column
41 gender_input=input("Enter genders separated by space: ")
42 genders=gender_input.split()
43 df["Gender"]=genders
44 # Display the DataFrame after adding a new column
45 print("After adding a new column:")
46 print(df)
47
48 # Modifying a column
49 df["Name"]=[(name).str.upper() for name in df["Name"]]
50 # Display the DataFrame after modifying a column
51 print("After modifying a column:")
52 print(df)
53
54 # Deleting a column
55 df=df.drop(columns=["Age"])
56 # Display the DataFrame after deleting a column
57 print("After deleting a column:")
58 print(df)

```

Terminal Test cases

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bcd2/667a7e0428220463f6497a3c

CODETANTRA Home

202401090058@mitaoe.ac.in Support Logout

4.1.1. Pandas - series creation and manipulation

Write a Python program that takes a list of numbers from the user, creates a Pandas series from it, and then calculates the mean of even and odd numbers separately using the `groupby` and `mean()` operations.

Input Format:

- The user should enter a list of numbers separated by space when prompted.

Output Format:

- The program should display the mean of even and odd numbers separately.
- Each mean value should be displayed with a label indicating whether it corresponds to even or odd numbers.

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

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >

BAJFINANCE +1.26% ENG IN 13:41 07-05-2025

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774f064f4ab707eca9da9b0

20240109005 125% Reset 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.

Output Format:

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 >

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774efaa4ab787eca9da91e

CODETANTRA Home 202401090058@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.
- After modifying the copy.

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

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

copyAnd... `import numpy as np`
1 `inputlist = list(map(int,input().split(".")))`
2
3
4 `# Original array`
5 `original_array = np.array(inputlist)`
6
7 `# Create a view`
8 `view_array = original_array.view()`
9
10 `# Create a copy`
11 `copy_array = original_array.copy()`
12
13 `# Modify the view`
14 `view_array[0] = 99`
15 `print("Original array after modifying view:", original_array)`
16 `print("View array:", view_array)`
17
18 `# Modify the copy`
19 `copy_array[1] = 88`
20 `print("Original array after modifying copy:", original_array)`
21 `print("Copy array:", copy_array)`
22
23

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ef2cf4ab787eca9da8c0

202401090058@mitaoe.ac.in Support Logout

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_1 OR B_1).

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

different...
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)))
A = list(map(int, input().split())) # Elements of array A
B = list(map(int, input().split())) # Elements of array B
array_operations(A, B)

< Prev Reset Submit Next >

BAJFINANCE +126% ENG IN 13:40 07-05-2025

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e451ff9c5320ca6bd17/6773e4d1ff9c5320ca6bdb8/6774ee19f4ab787eca9da7fe

20240109005 125% - + Reset Logout

CODETANTRA Home

3.2.2. Numpy: Horizontal and Vertical Stacking of Arrays 02:28 A ⚡ -

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

```
1 import numpy as np
2
3 # Input matrices
4 print("Enter Array1:")
5 arr1 = np.array([list(map(int, input().split())) for i in range(3)])
6
7 print("Enter Array2:")
8 arr2 = np.array([list(map(int, input().split())) for i in range(3)])
9
10 # Perform horizontal stacking (hstack)
11 a=np.hstack((arr1,arr2))
12 print("Horizontal Stack:")
13 print(a)
14 print("Vertical Stack:")
15 b=np.vstack((arr1,arr2))
16 print(b)
```

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >

BAJFINNCE +1.26% ENG IN 13:40 07-05-2025

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774e3ff4ab787eca9da092

202401090058@mitaoe.ac.in Support Logout

CODETANTRA Home

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.
3. The result of Element-wise Multiplication.
4. The result of Matrix Multiplication.
5. The Transpose of Matrix A.

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 print("Addition (A + B):")
13 print(matrix_a + matrix_b)
14 # Subtraction
15 print("Subtraction (A - B):")
16 print(matrix_a - matrix_b)
17 # Multiplication (element-wise)
18 print("Element-wise Multiplication (A * B):")
19 print(matrix_a * matrix_b)
20 # Matrix multiplication (dot product)
21 print("A dot B:")
22 print(np.dot(matrix_a, matrix_b))
23 # Transpose
24 print("Transpose of A:")
25 a= matrix_a.T
26 print(a)
```

Sample Test Cases

Terminal Test cases

< Prev Reset Submit Next >

BAJFINANCE +126%

Search

ENG IN 13:40 07-05-2025

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e451f1f9c5320ca6bd17/6773e4ccf1f9c5320ca6bdb2/6684d40e5d95376955a73ctd

202401090058@mitaoe.ac.in Support Logout

CodeTANTRA Home

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

Explorer numpyarr...

```
1 import numpy as np
2 rows,cols= list(map(int,input().split()))
3 matrix= []
4 v for i in range(rows):
5     row = list(map(int,input().split()))
6     matrix.append(row)
7 matrix= np.array(matrix).reshape(rows,cols)
8 print(matrix)
9 print(matrix.ndim)
10 print(matrix.shape)
11 print(matrix.size)
```

Terminal Test cases

< Prev Reset Submit Next >

BAJFINANCE +126% Search ENG IN 13:40 07-05-2025

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e44bf1f9c5320ca6bc8/6773e4b0ff9c5320ca6bd85/64ac0cb9b4547106899a4e1b

CODETANTRA Home 202401090058@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 arr = list(map(int,input().split()))
2
3 key = int(input())
4
5 for i in range(len(arr)):
6     if arr[i] == key:
7         print(i)
8         break
9
10 if arr[i] != key:
11     print("Not found")
```

Terminal Test cases < Prev Reset Submit Next >

BAJFINANCE +126% ENG IN 13:39 07-05-2025

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773eb5df4ab787eca9d0bf7

CODETANTRA Home 202401090058@mitaoe.ac.in Support Logout

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:
 $1 \leq n \leq 999$

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".

condition...

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

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >



New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6774c521f4ab787eca9d8a75

CODETANTRA Home 202401090058@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... Explorer

```
1 a=int(input())
2 v for i in range (1,a+1):
3     print('*'*i)
```

Terminal Test cases < Prev Reset Submit Next >

BAJFINANCE +126% Search ENG IN 13:38 07-05-2025

New repository Course

mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/65f7d776bfe2dc3ed510ca60

CODETANTRA Home

202401090058@mitaoe.ac.in Support Logout

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

```
1 v def fib(i):
2 v     if(i==0):
3 v         return 0
4 v     elif(i==1):
5 v         return 1
6 v     else:
7 v         return fib(i-1)+fib(i-2)
8 v
9 v
10 v
11 v
12 v
13 v
14 v
15 v
16 v
17 v
18 n=int(input("Enter terms for Fibonacci series: "))
19 for i in range (n):
20     print(fib(i),end=" ")
```

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >

BAJFINANCE +126% Search ENG IN 13:38 07-05-2025

New repository Course mitaoe.codetantra.com/secure/course.jsp?euclid=6773e3f2f1f9c5320ca6bc85#contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773ef42f4ab787eca9d11f7

CODETANTRA Home 202401090058@mitaoe.ac.in Support Logout

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

Explorer birthDate...

```
1 # import the required package
2 from datetime import datetime,date
3 v def calculate_age(birthdate):
4     # Write your code here
5     day,month,year=map(int,birthdate.split("-"))
6     birth_date=date(year,month,day)
7     to=date.today()
8     age=to.year-birth_date.year
9     if (birth_date.month,birth_date.day)>(to.month,to.day):
10        age=age-1
11    return age
12 v def convert_salary_to_dollars(salary_in_rupees):
13     # write your code here
14     return salary_in_rupees*0.012
15 birthdate = input()
16 salary_in_rupees = float(input())
17 age = calculate_age(birthdate)
18 salary_in_dollars = convert_salary_to_dollars(salary_in_rupees)
19 print(f"Age: {age}")
20 print(f"Salary in dollars: {salary_in_dollars:.2f}")
```

Terminal Test cases < Prev Reset Submit Next >