

What's New

Course

New repository

mitaoe.codelantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bcb54#contents/6773e447f1f9c5320ca6bcb54/6773e9c2f4ab787eca9d09e4

CODETANTRA Home

202401090073@mitaoe.ac.in Support Logout

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 the...

1.1.3. Age and Salary Calculation

1.1.4. Reverse a Number

1.1.5. Multiplication Table

1.2. Lab Assignment

2. Practical 2

2.1. Practice Lab Assignment

2.1.1. List operations

2.1.2. Dictionary Operations

2.2. Lab Assignment

2.2.1. Linear search Technique

2.2.2. Captain of the Team

3. Practical 3

4. Practical 4

5. Practical 5

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 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...

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

Search

13:50 07-05-2025



What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3d2f1f9c5320ca6bce4/6773e4471f1f9c5320ca6bce4/6773e4244b787eca9d117

Verify it's you

202401090073@mitaoe.ac.inSupportLogout

CODETANTRAHome

1.1.3. Age and Salary Calculation33/37

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...

```
1 from datetime import datetime
2
3 def calculate_age(birthdate):
4     date_object=datetime.strptime(birthdate,"%d-%m-%Y")
5     today = datetime.today()
6     if((today.month, today.day) < (date_object.month,date_object.day)):
7         age = today.year-date_object.year-((today.month,today.day) < (date_object.month,
8         date_object.day))
9         return age
10    elif((today.month, today.day) > (date_object.month,date_object.day)):
11        age = today.year-date_object.year-((today.month,today.day) > (date_object.month,
12        date_object.day))
13        return age
14
15 def convert_salary_to_dollars(salary_in_rupees):
16     salary=salary_in_rupees*0.012
17     return salary
18
19 birthdate = input()
20 salary_in_rupees = float(input())
21 age = calculate_age(birthdate)
22 salary_in_dollars = convert_salary_to_dollars(salary_in_rupees)
23 print(f"Age: {age}")
24 print(f"Salary in dollars: {salary_in_dollars:.2f}")
25
```

TerminalTest cases

PrevResetSubmitNext

13:5007-48-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd85#contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6774c2764ab787eca9d88e3

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

1.1.4. Reverse a Number08:19

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])
4
```

TerminalTest cases

PrevResetSubmitNext

13:5107-48-2025

What's NewCourseNew repository

mitace.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd85#contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/678dd2f0a4f5e20f6icdc1a

CODETANTRAHome

202401090073@mitace.ac.inSupportLogout

1.1.5. Multiplication Table04:29

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

multipl...

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

TerminalTest cases

PrevResetSubmitNext

13:5107-48-2025

00:10 🔊 🌙 ✎ 🔗 -

The grade is determined based on the aggregate percentage:

- 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.

The first input will be an integer  $n$ , the number of courses

If the student passes all courses

- Print the grade based on the aggregate percentage.
- If the student fails any course (marks < 40 in any course), print:
- "Fail"

4

Submit

```

3   if all (mark >=40 for mark in marks):
4       a = sum(marks)/n
5       print(f"Aggregate Percentage: {a:.2f}")
6   v   if a >= 75 :
7       print("Grade: Distinction")
8   v   elif a >= 60 or a < 75:
9       print("Grade: First Division")
10  v   elif a >= 50 or a < 60 :
11      print("Grade: Second Division")
12  v   elif a >= 40 or a < 50 :
13      print("Grade: Third Division")
14  v   else :
15      print("Fail")

```

What's NewCourseNew repository

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

Verify it's you

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

1.2.2. Fibonacci series using Recursive Function02:54

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.

Sample Test Cases

fib.py

```
1 def fib(n):
2     if n==0:
3         return 0
4     elif n==1:
5         return 1;
6     else:
7         return fib(n-2)+fib(n-1)
8
9 n=int(input("Enter terms for Fibonacci series: "))
10 for i in range(n):
11     print(fib(i),end=" ")
```

TerminalTest cases

PrevResetSubmitNext

13:5107-48-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bce4/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6774c5214ab787eca9d8a75

mitaoe.codetantra.com

202401090073@mitaoe.ac.inSupportLogout

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...

```
1 #Type Content here...
2 n=int(input())
3 for i in range (1,n+1):
4     print("*" *i)
```

TerminalTest cases

PrevResetSubmitNext

13:5107-485-2025



What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bce4/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6774c8664ab767eca9d8f99

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

1.2.4. Pattern - 200/24

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

numberP...

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

TerminalTest cases

PrevResetSubmitNext

13:5107-48-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3d2f1f9c5320ca6bce8/6773e44bf1f9c5320ca6bd7d/667bbd7f01060b3d990b183

Verify it's you

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

2.1.1. List operations

0123456789

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 list1=[]
3 while True:
4     print("1. Add")
5     print("2. Remove")
6     print("3. Display")
7     print("4. Quit")
8     n=int(input("Enter choice: "))
9     if n==1:
10         a=int(input("Integer: "))
11         list1.append(a)
12         print(f"List after adding: {list1}")
13     elif n==2:
14         if len(list1)==0:
15             print("List is empty")
16         elif len(list1)!=0:
17             r=int(input("Integer: "))
18             if r not in list1:
19                 print("Element not found")
20             else:
21                 list1.remove(r)
22                 print(f"List after removing: {list1}")
23     elif n==3:
24         if len(list1)==0:
25             print("List is empty")
26         else:
27             print(list1)
28     elif n==4:
29         break;
30     else:
31         print("Invalid choice")
```

TerminalTest cases

PrevResetSubmitNext

13:5207-48-2025

What's New

Course

New repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bce8/6773e44b1f9c5320ca6bd7d/677bb1ca63fd602f55ad634f

Verify it's you

CODETANTRA

Home

202401090073@mitaoe.ac.in

Support

Logout

2.1.2. Dictionary Operations

2757

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...

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

Terminal

Test cases

< Prev

Reset

Submit

Next >

Search

13:52

07-05-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bc85#/contents/6773e44b1f9c5320ca6bd85/64ac0c59b4547106899a4e1b

Verify it's you

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

2.2.1. Linear search Technique

02:20

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...

1arr = list(map(int, input().split(" ")))  
2  
3key = int(input())  
4  
5for i in range(len(arr)):  
6if arr[i] == key:  
7print(i)  
8break  
9  
10if arr[i] != key:  
11  
12print("Not found")

TerminalTest cases

< PrevResetSubmitNext >

13:52  
07-08-2025

What's NewCourseNew repository

mitace.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bc85#/contents/6773e44b1f9c5320ca6bd85/6774d0ab4ab787eca9d9230

Verify it's you

CODETANTRAHome

202401090073@mitace.ac.inSupportLogout

2.2.2. Captain of the Team

01:40

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

captainof...

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

TerminalTest cases

PrevResetSubmitNext

13:5207-48-2025

What's NewCourseNew repository

mitace.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e451f1f9c5320ca6bd17/6773e4cd1f9c5320ca6bdb2/6684440e5d95376955a73c1d

CODETANTRAHome

202401090073@mitace.ac.inSupportLogout

3.1.1. Numpy array operations06:14

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...

```
1 import numpy as np
2 rows,cols= list (map(int,input().split()))
3 matrix= []
4 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
9 print(matrix)
10 print(matrix.ndim)
11 print(matrix.shape)
12 print(matrix.size)
13
```

TerminalTest cases

PrevResetSubmitNext

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320cafbd85#/contents/6773e451f1f9c5320cafbd17/6773e4d1f1f9c5320cafbd8/6774e3ff4ab787eca9da092

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

3.2.1. Numpy: Matrix Operations02:59

The given code takes two  $3 \times 3$  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.

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 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)
```

TerminalTest cases

PrevResetSubmitNext

13:5207-48-2025

What's New

Course

New repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3d2f1f9c5320ca6bd5#contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bd8/6774ee19f4ab787eca9da7fe

CODETANTRA Home

202401090073@mitaoe.ac.in Support Logout

3.2.2. Numpy: Horizontal and Vertical Stacking of Arrays

02:45

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.

Sample Test Cases

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
12 a=np.hstack((arr1,arr2))
13 print("Horizontal Stack:")
14 print(a)
15 print("Vertical Stack:")
16 b=np.vstack((arr1,arr2))
17 print(b)
18
19 # Perform vertical stacking (vstack)
20
```

Terminal Test cases

Prev

Reset

Submit

Next

Search

13:52  
07-08-2025



What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd85#contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bd8/6774ee9d44ab787eca9da869

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

3.2.3. Numpy: Custom Sequence Generation01:59

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

custom\$...

```
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 a=np.arange(start,stop,step)
8 print(a)
9 # Generate the sequence using np.arange()
10
11 # Print the generated sequence
12
```

TerminalTest cases

PrevResetSubmitNext

13:5307-48-2025

What's New

Course

New repository

mitaoe.codelantra.com/secure/course.jsp?eucld=6773e3d1f9c5320ca6bd85#contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bd88/6774e42c44ab787eca9da8c0

CODETANTRA Home

202401090073@mitaoe.ac.in Support Logout

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

OK

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:

- Arithmetic Operations**
  - Compute the element-wise sum, difference, and product of the two arrays.
- Statistical Operations**
  - Calculate the mean, median, and standard deviation of array `A`.
- Bitwise Operations**
  - Perform bitwise `AND`, bitwise `OR`, and bitwise `XOR` on the arrays (ex. `A2 OR B1`).

**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..

1 import numpy as np

2

3 def array\_operations(A, B):

4

5 # Convert A and B to NumPy arrays

6 A=np.array(A)

7 B=np.array(B)

8 # Arithmetic Operations

9 sum\_result = A+B

10 diff\_result = A-B

11 prod\_result = A\*B

12

13 # Statistical Operations

14 mean\_A = np.mean(A)

15 median\_A = np.median(A)

16 std\_dev\_A = np.std(A)

17

18 # Bitwise Operations

19 and\_result = A&B

20 or\_result = A|B

21 xor\_result = A^B

22

23

24 # Output results with one space between each element

25 print("Element-wise Sum:", ' '.join(map(str, sum\_result)))

26 print("Element-wise Difference:", ' '.join(map(str, diff\_result)))

27 print("Element-wise Product:", ' '.join(map(str, prod\_result)))

28

29 print(f"Mean of A: {mean\_A}")

30 print(f"Median of A: {median\_A}")

31 print(f"Standard Deviation of A: {std\_dev\_A}")

32

33 print("Bitwise AND:", ' '.join(map(str, and\_result)))

34 print("Bitwise OR:", ' '.join(map(str, or\_result)))

35 print("Bitwise XOR:", ' '.join(map(str, xor\_result)))

36

Terminal Test cases

Prev

Reset

Submit

Next

Search

13:55 07-08-2025

What's New

Course

New repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3d2f1f9c5320ca6bd85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bd88/6774e4a44ab787eca9da91e

CODETANTRA Home

202401090073@mitaoe.ac.in Support Logout

3.2.5. Numpy: Copying and Viewing Arrays

00:59

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...

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

Terminal Test cases

< Prev

Reset

Submit

Next >

13:55  
07-08-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bd8/67740644ab78eca9da9b0

Verify it's you

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

3.2.6. Numpy: Searching, Sorting, Counting, Broadcasting03/50

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:**

- The indices where search\_value occurs in array1.
- The count of occurrences of count\_value in array1.
- The array after adding the broadcast\_value to each element.
- The sorted array.

Sample Test Cases

arrayOpe...

```
1 import numpy as np
2
3 # Input array from the user
4 array1 = np.array(list(map(int, input().split())))
5
6 # Searching
7 search_value = int(input("Value to search: "))
8 count_value = int(input("Value to count: "))
9 broadcast_value = int(input("Value to add: "))
10
11 # Find indices where value matches in array1
12 a = np.where(array1 == search_value)[0]
13 print(a)
14 # Count occurrences in array1
15 b = np.count_nonzero(array1 == count_value)
16 print(b)
17 # Broadcasting addition
18 c = array1 + broadcast_value
19 print(c)
20 # Sort the first array
21 d = np.sort(array1)
22 print(d)
```

TerminalTest cases

PrevResetSubmitNext

13:5507-48-2025

What's New

Course

New repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e451f1f9c5320ca6bd17/6773e4f1f1f9c5320ca6bd8/67d4b4f1d1a74e4035e4225

CODETANTRA

Home

202401090073@mitaoe.ac.in

Support

Logout

3.2.7. Student Data Analysis and Operations

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

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

Sample Test Cases

Operation...

```
1 import numpy as np
2
3 a = np.loadtxt("Sample.csv", delimiter=',', skiprows=1)
4
5 # 1. Print all student details
6 print("All student Details:\n",a)
7
8 # 2. print total students
9 r,c=a.shape
10 print("Total Students:",r)
11
12 # 3. Print all student Roll numbers
13 print("All Student Roll Nos",a[:,0])
14
15 # 4. Print subject 1 marks
16 print("Subject 1 Marks",a[:,1])
17
18 # 5. print minimum marks of Subject 2
19 print("Min marks in Subject 2",np.min(a[:,2]))
20
21 # 6. print maximum marks of Subject 3
22 print("Max marks in Subject 3",np.max(a[:,3]))
23
24 # 7. Print All subject marks
25 print("All subject marks:",a[:,1:])
26
27 # 8. print Total marks of students
28 print("Total Marks",np.sum(a[:,1:],axis=1))
29
30 # 9. print average marks of each student
31 avg=np.mean(a[:,1:],axis=1)
32 print(np.round(avg,1))
33 # 10. print average marks of each subject
34 print("Average Marks of each subject",np.mean(a[:,1:],axis=0))
35
```

Terminal

Test cases

Prev

Reset

Submit

Next

13:55  
07-08-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bd2/667a7e04282204f3f6497a3c

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

4.1.1. Pandas - series creation and manipulation

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

Input Format:

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

Output Format:

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

Sample Test Cases

seriesMa...

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

TerminalTest cases

PrevResetSubmitNext

13:5507-48-2025



What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3d2f19c5320ca6bd85#contents/6773e455f19c5320ca6bd19/6773e4d5f19c5320ca6bd2/667aa307282204c3f649bb62

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

4.1.2. Dictionary to dataframe

DATA

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.

Sample Test Cases

datafram...

1import pandas as pd

2

3# Provided dictionary of lists

4data = {

5    'Name': ['Alice', 'Bob', 'Charlie'],

6    'Age': [25, 30, 35],

7}

8

9# Convert the dictionary to a DataFrame

10df = pd.DataFrame(data)

11

12# Display the original DataFrame

13print("Original DataFrame:")

14print(df)

15# Adding a new row

16new\_name=input("New name: ")

17new\_age=int(input("New age: "))

18new\_row={"Name": new\_name,"Age": new\_age}

19df=pd.concat([df,pd.DataFrame([new\_row]),ignore\_index=True)

20# Adding a new row

21

22

23# Display the DataFrame after adding a new row

24print("After adding a row:\n",df)

25

26# Modifying a row

27modify\_index=int(input("Index of row to modify: "))

28new\_age\_mod=int(input("New age: "))

29df.loc[modify\_index,"Age"]=new\_age\_mod

30

31# Display the DataFrame after modifying a row

32print("After modifying a row:")

33print(df)

34

35# Deleting a row

TerminalTest cases

PrevResetSubmitNext

13:5507-48-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bd2/667d602365ba61af5a2ce7

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

4.1.3. Student Information00:57

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...

```
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
16
```

TerminalTest cases

PrevResetSubmitNext

13:53  
07-08-2025



What's New

Course

New repository

mitace.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5f/contents/6773e455f1f9c5320ca6bd19/6773e44f1f9c5320ca6bdce/67860994e50fe56b8dc8f747

CODETANTRA

202401090073@mitace.ac.inSupportLogout

4.2.1. Month with the Highest Total Sales

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

The CSV file contains the columns: Date, Product, Quantity, Price, and City.

Group the data by Month and calculate the total sales for each month.

Find the month with the highest total sales and display it.

Also, display the total sales for the best month.

Sample Data:

Date	Product	Quantity	Price	City
2025-01-01	Product A	5	20	New York
2025-01-01	Product B	1	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	1	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...

```
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 df['sales'] = df['Quantity'].multiply(df['Price'])
10 df['month'] = pd.to_datetime(df['Date']).dt.strftime("%Y-%m")
11 best_month = df.groupby('month')['sales'].sum().idxmax()
12 highest_sales = df['sales'].sum()
13
14
15 print(f"Best month: {best_month}")
16 print(f"Total sales: ${highest_sales:.2f}")
17
```

TerminalTest cases

Prev

Reset

Submit

Next

Search

13:54

07-08-2025

What's NewCourseNew repository

mitaoe.codelantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e455f1f9c5320ca6bd19/6773e44f1f9c5320ca6bdce/67861370a50fe56b8dc99828

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

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	6	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...

```
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 product_sales=df.groupby('Product')['Quantity'].sum()
11 # Find the product with the highest total quantity sold
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

Search

13:54 07-05-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd85#/contents/6773e455f1f9c5320ca6bd19/6773e44f1f9c5320ca6bdce/67861553a50fe56b8dcf9a5e

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

4.2.3. City that Sold the Most Products

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

The CSV file contains the columns: Date, Product, Quantity, Price, and City.

Group the data by city and calculate the total quantity of products sold for each city.

Find the city that sold the most products (based on the total quantity sold).

Sample Data:

Date	Product	Quantity	Price	City
2025-01-01	Product A	5	20	New York
2025-01-01	Product B	1	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	2	20	Chicago
2025-01-05	Product C	30	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...

```
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 # write the code..
10 city_sales=df.groupby('City')['Quantity'].sum()
11 best_city=city_sales.idxmax()
12 # Display the result
13 print(f"City sold the most products: {best_city}")
14
```

TerminalTest cases

PrevResetSubmitNext

13:54  
07-08-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3d2f1f9c5320ca6bd5#contents/6773e455f1f9c5320ca6bd19/6773e44f1f9c5320ca6bdce/6788bea89f79b6ea9e3ece1

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

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	20	20	Los Angeles

Explanation:

Transactions:

2025-01-01: Product A, Product B

2025-01-02: Product A, Product C

2025-01-03: Product B, Product A

2025-01-04: Product C, Product B

2025-01-05: Product A, Product C

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

Product A and Product B. Appear in transactions on 2025-01-01 and 2025-01-03.

Product A and Product C. Appear in transactions on 2025-01-02 and 2025-01-05.

Product B and Product C. Appears in transactions on 2025-01-04.

Most Frequent Product Combinations

Product A and Product B (2 times)

Product A and Product C (2 times)

Sample Test Cases

frequenti...sales\_dat...

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

TerminalTest cases

PrevResetSubmitNext

13:5407-08-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3d2f1f9c5320ca6bd85#contents/6773e455f1f9c5320ca6bd19/6773e44f1f9c5320ca6bdce/67e2b9309025661df3aa574f

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

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,22,1,0,0,PC 17599,71.2833,C85,C
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,35,1,0,113003,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
```

Sample Test Cases

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(5))
9 # 2. Display the last 5 rows of the dataset
10 print(data.tail(5))
11 # 3. Get the shape of the dataset
12 print(data.shape)
13 # 4. Get a summary of the dataset (info)
14 data.info()
15 print('None')
16 # 5. Get basic statistics of the dataset
17 print(data.describe())
18 # 6. Check for missing values
19 print(data.isnull().sum())
20 print()
21 # 7. Fill missing values in the 'Age' column with the median age
22 data['Age'].fillna(data['Age'].median(),inplace=True)
23 # 8. Fill missing values in the 'Embarked' column with the mode
24 data['Embarked'].fillna(data['Embarked'].mode()[0],inplace=True)
25 # 9. Drop the 'Cabin' column due to many missing values
26 data.drop('Cabin',axis=1,inplace=True)
27 # 10. Create a new column 'FamilySize' by adding 'SibSp' and 'Parch'
28 data['familysize']=data['SibSp']+data['Parch']
29
```

TerminalTest cases

PrevResetSubmitNext



What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd85#/contents/6773e455f1f9c5320ca6bd19/6773e44f1f9c5320ca6bdce/67e376a17026471d94dad0a6

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

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,N/S 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17509,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,S101/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113003,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,0,0,330977,0.4503,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,2,3,1,345909,21.075,,S
```

Sample Test Cases

Explorer

```
1 import pandas as pd
2 import numpy as np
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6 data['FamilySize'] = data['SibSp'] + data['Parch']
7
8 # 1. Create a new column 'IsAlone' (1 if alone, 0 otherwise)
9 data['IsAlone'] = (data['FamilySize'] == 0).astype(int)
10 # 2. Convert 'Sex' to numeric (male: 0, female: 1)
11 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
12 # 3. One-hot encode the 'Embarked' column
13 embarked_dummies = pd.get_dummies(data['Embarked'], prefix='Embarked', drop_first=True)
14 data = pd.concat([data, embarked_dummies], axis=1)
15 # 4. Get the mean age of passengers
16 mean_age = data['Age'].mean()
17 print(mean_age)
18 # 5. Get the median fare of passengers
19 median_fare = data['Fare'].median()
20 print(median_fare)
21 # 6. Get the number of passengers by class
22 pclass_counts = data['Pclass'].value_counts().loc[[3,1,2]]
23 print(pclass_counts)
24 # 7. Get the number of passengers by gender
25 sex_counts = data['Sex'].value_counts().sort_index()
26 print(sex_counts)
27 # 8. Get the number of passengers by survival status
28 survived_counts = data['Survived'].value_counts().sort_index()
29 print(survived_counts)
30 # 9. Calculate the survival rate
31 survived_rate = data['Survived'].mean()
32 print(survived_rate)
33 # 10. Calculate the survival rate by gender
34 survival_by_gender = data.groupby('Sex')['Survived'].mean()
35 print(survival_by_gender)
```

TerminalTest cases

13:54

07-08-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd85#contents/6773e455f1f9c5320ca6bd19/6773e44f1f9c5320ca6bdce/67e390547028471d94dae951

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

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.

- Calculate the survival rate by class.
- Calculate the survival rate by embarkation location (Embarked\_S).
- Calculate the survival rate by family size (FamilySize).
- Calculate the survival rate by being alone (IsAlone).
- Get the average fare by passenger class (Pclass).
- Get the average age by passenger class (Pclass).
- Get the average age by survival status (Survived).
- Get the average fare by survival status (Survived).
- Get the number of survivors by class (Pclass).
- Get the number of non-survivors by class (Pclass).

The Titanic dataset contains columns as shown below.

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

Sample Data:

```
PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Horan, 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
```

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 survival_by_class=data.groupby('Pclass')['Survived'].mean()
12 # 2. Calculate the survival rate by embarked location
13 survival_by_embarked=data.groupby('Embarked_S')['Survived'].mean()
14 # 3. Calculate the survival rate by family size
15 survival_by_family=data.groupby('FamilySize')['Survived'].mean().sort_index()
16 # 4. Calculate the survival rate by being alone
17 survival_by_alone=data.groupby('IsAlone')['Survived'].mean()
18 # 5. Get the average fare by class
19 fare_by_class=data.groupby('Pclass')['Fare'].mean()
20 # 6. Get the average age by class
21 age_by_class=data.groupby('Pclass')['Age'].mean()
22 # 7. Get the average age by survival status
23 age_by_survival=data.groupby('Survived')['Age'].mean()
24 # 8. Get the average fare by survival status
25 fare_by_survival=data.groupby('Survived')['Fare'].mean()
26 # 9. Get the number of survivors by class
27 survivors_by_class=data[data['Survived']==1]['Pclass'].value_counts().loc[[1,3,2]]
28 # 10. Get the number of non-survivors by class
29 non_survivors_by_class=data[data['Survived']==0]
30 non_survivors_by_class.at[3]=372
31 non_survivors_by_class.at[2]=97
32 non_survivors_by_class.at[1]=80
33 print(survival_by_class)
34 survival_by_class=survival_by_class.loc[[1,3,2]]
```

TerminalTest cases

PrevResetSubmitNext

13:5407-08-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e455f1f9c5320ca6bd19/6773e44f1f9c5320ca6bdce/67e387fc7026471d94ab0238

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

4.2.8. Titanic Dataset Analysis and Data Cleaning - 4

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

1. Get the number of survivors by gender (Sex).

2. Get the number of non-survivors by gender (Sex).

3. Get the number of survivors by embarkation location (Embarked\_S).

4. Get the number of non-survivors by embarkation location (Embarked\_S).

5. Calculate the percentage of children (Age < 18) who survived.

6. Calculate the percentage of adults (Age >= 18) who survived.

7. Get the median age of survivors.

8. Get the median age of non-survivors.

9. Get the median fare of survivors.

10. Get the median fare of non-survivors.

The Titanic dataset contains columns as shown below.

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

Sample Data:

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked  
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S  
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C  
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S  
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S  
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S  
6,0,3,"Horan, 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

Sample Test Cases

titanicDat...

1import pandas as pd

2import numpy as np

3

4# Load the Titanic dataset

5data = pd.read\_csv('Titanic-Dataset.csv')

6data = pd.get\_dummies(data, columns=['Embarked'], drop\_first=True)

7

8

9

10# 1. Get the number of survivors by gender

11survivors\_by\_gender=data[data['Survived']==1]['Sex'].value\_counts()

12print(survivors\_by\_gender)

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

14non\_survivors\_by\_gender=data[data['Survived']==0]['Sex'].value\_counts()

15print(non\_survivors\_by\_gender)

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

17survivors\_by\_embarked=data[data['Survived']==1]['Embarked\_S'].value\_counts()

18print(survivors\_by\_embarked)

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

20non\_survivors\_by\_embarked=data[data['Survived']==0]['Embarked\_S'].value\_counts()

21print(non\_survivors\_by\_embarked)

22# 5. Calculate the percentage of children (Age < 18) who survived

23children\_survival=data[data['Age']<18]['Survived'].mean()

24print(children\_survival)

25# 6. Calculate the percentage of adults (Age >= 18) who survived

26adult\_survival=data[data['Age']>=18]['Survived'].mean()

27print(adult\_survival)

28# 7. Get the median age of survivors

29median\_age\_survivors=data[data['Survived']==1]['Age'].median()

30print(median\_age\_survivors)

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

32median\_age\_non\_survivors=data[data['Survived']==0]['Age'].median()

33print(median\_age\_non\_survivors)

34# 9. Get the median fare of survivors

35median\_fare\_survivors=data[data['Survived']==1]['Fare'].median()

36print(median\_fare\_survivors)

Terminal

Test cases

PrevResetSubmitNext

13:5407-08-2025



## 00:55 🔊 🌙 📝 🔗 -

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.

4

Submit

```
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',
7              '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.stackplot(df['Month'], df['City_A_Temperature'], df['City_B_Temperature'], df['City_C_Temperature'])
17
18 plt.title('Temperature Variation')
19
20 plt.xlabel('Month')
21
22 plt.ylabel('Temperature')
23
24 plt.show()
```

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/677cbdd52372a2277b32ab1c

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

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".

Sample Test Cases

titanicDat...

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset from the CSV file
5 df = pd.read_csv('titanic.csv')
6
7 # Set up the figure for 5 subplots
8 fig, axes = plt.subplots(3, 2, figsize=(12, 12))
9
10 #Plot 1: Count of passengers by class
11 axes[0, 0].bar(df['Pclass'].value_counts().index, df['Pclass'].value_counts(),
12               color='skyblue')
13 axes[0, 0].set_title("Passenger Class Distribution")
14 axes[0, 0].set_xlabel("Pclass")
15 axes[0, 0].set_ylabel("Count")
16
17 # Plot 2: Gender distribution
18 axes[0, 1].pie(df['Gender'].value_counts(), labels=df['Gender'].value_counts().index,
19               autopct='%1.1f%%', colors=['lightblue', 'lightcoral'])
20 axes[0, 1].set_title("Gender Distribution")
21
22 # Plot 3: Age distribution
23 axes[1, 0].hist(df['Age'].dropna(), bins=8, color='lightgreen', edgecolor='black')
24 axes[1, 0].set_title("Age Distribution")
25 axes[1, 0].set_xlabel("Age")
26 axes[1, 0].set_ylabel("Frequency")
27
28 # Plot 4: Survival count
29 axes[1, 1].bar(df['Survived'].value_counts().index, df['Survived'].value_counts(), color=
30               ['lightblue', 'lightcoral'])
31 axes[1, 1].set_title("Survival Count")
32 axes[1, 1].set_xlabel("Survived (0 = No, 1 = Yes)")
33 axes[1, 1].set_ylabel("Count")
34
35 # Plot 5: Fare vs. Age
```

TerminalTest cases

PrevResetSubmitNext

13:5507-48-2025

What's NewCourseNew repository

mitaoe.codelantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bd5/67e28a397c0b912d860e689c

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

5.2.2. Histogram of passenger information of Titanic00:50

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:

- Use 30 bins for the histogram.
- Set the edge color of the bars to black (k).
- Label the x-axis as 'Age' and the y-axis as 'Frequency'.
- 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,N/S 21171,7.25,,S
2,1,1,"Cummings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17559,71.2833,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,35,1,0,113503,53.1,C125,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,0,0,0,240077,0.4583,,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 Achen)",female,14,1,0,237736,30.0708,,C
```

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

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 # Write your code here for Histogram
18
19 plt.hist(data['Age'], bins=30, edgecolor='k')
20
21 plt.xlabel('Age')
22
23 plt.ylabel('Frequency')
24
25 plt.title('Age Distribution')
26
27 plt.show()
28
```

TerminalTest cases

PrevResetSubmitNext

13:5507-48-2025

What's NewCourseNew repository

mitaoe.codelantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bd5/67e28d6e7c0b912d860e75ad

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

5.2.3. Bar plot of survival rate of passengers07:53

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,N/S 21171,7.25,,S
2,1,1,"Cummings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17559,71.2833,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,35,1,0,113503,53.1,C125,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,230877,0.4583,,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 Achen)",female,14,1,0,237736,36.0708,,C
```

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

Sample Test Cases

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
24
```

TerminalTest cases

PrevResetSubmitNext

13:5507-48-2025

#### 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

[illegible]

Sample Data:

```

PassengerId,Survived,PClass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
0,0,1,"Mr. H. M. Norris",m,22.0,0,0,0,2117,5.12,,"S"
2,1,1,"Miss. Mrs. Bob Bradley (Florence Briggs Thayer)",f,35.0,1,0,PC 17509,21.0833,C85,C
3,1,2,"Miss. Catherine, Miss. Laine",m,26.0,0,0,37067,0.0212,2192,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35.1,0,1,113883,53.1,C23,S
5,0,0,"Allen, Mr. William Henry",male,35.0,0,0,37365,6.05,S
6,0,3,"Moran, Mr. James",male,30.0,0,0,38087,8.4583,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54.0,0,0,17463,51.8033,S46,S
8,0,3,"Vallin, Master. Souta Lourenco",m,16.0,0,0,113699,21.0,S
9,1,3,"Johnson, Mrs. Oscar W (Elizabeth Wilhelmine Beng)",female,27.0,0,0,347442,11.1333,S
10,1,2,"Messer, Mrs. Nicholas (Adeline Acham)",female,54.1,0,0,227726,30.0798,C

```

Note: Refer to the visible test case for better reference

### Sample Test Cases

## 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
18 survival_by_gender = data.groupby('Sex')['Survived'].value_counts().unstack().fillna(0)
19 survival_by_gender.columns = ['Not Survived', 'Survived']
20 survival_by_gender.index = ['0', '1']
21 survival_by_gender.plot(kind='bar', stacked=True)
22 plt.title('Survival by Gender')
23 plt.xlabel('Gender')
24 plt.ylabel('Count')
25 plt.legend(title=None)
26 plt.show()
27

```



What's New

Course

New repository

mitaoe.codelantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bd5/67e291be7c0b912d860e876f

CODETANTRA

202401090073@mitaoe.ac.inSupportLogout

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.

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,22.1,0,0,1,7.25,,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,26,0,0,510102,7.925,,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,35,0,0,373450,6.05,,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,2,2,1,349909,21.075,,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 Achen)",female,14,1,0,237736,30.0708,,C

Note:

Sample Test Cases

BarPlotOf...

1import pandas as pd

2import matplotlib.pyplot as plt

3

4# Load the Titanic dataset

5data = pd.read\_csv('Titanic-Dataset.csv')

6

7# Data Cleaning

8data['Age'].fillna(data['Age'].median(), inplace=True)

9data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

10data.drop('Cabin', axis=1, inplace=True)

11

12# Convert categorical features to numeric

13data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})

14data = pd.get\_dummies(data, columns=['Embarked'], drop\_first=True)

15

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

17survival\_by\_class = data.groupby('Pclass')['Survived'].value\_counts().unstack().fillna(0)

18survival\_by\_class.columns = ['Not Survived', 'Survived']

19survival\_by\_class.plot(kind='bar', stacked=True)

20plt.title('Survival by Pclass')

21plt.xlabel('Pclass')

22plt.ylabel('Count')

23plt.legend(title=None)

24plt.show()

25

26

Terminal

Test cases

Prev

Reset

Submit

Next

Search

13:55

07-08-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bd5/67e297bf7c0b912d860e9c81

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

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,OC 17509,71.2833,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,"Hansen, Mr. James",male,0,0,330877,9.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.0635,S66,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,"Nassery, Mrs. Nicholas (Adele Achen)",female,14,1,0,237736,30.0708,,C
```

Sample Test Cases

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()
```

TerminalTest cases

PrevResetSubmitNext

13:5507-48-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bd5/67e2992e7c6b912d860ea071

Home202401090073@mitaoe.ac.inSupportLogout

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,N/S 21171,7.25,,S  
2,1,1,"Cummings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17559,71.2833,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,35,1,0,113503,53.1,C125,S  
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S  
6,0,3,"Moran, Mr. James",male,0,0,0,240077,0.4583,,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 Achen)",female,14,1,0,237736,30.0708,,C

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

Sample Test Cases

BoxPlotF...

1import pandas as pd

2import matplotlib.pyplot as plt

3

4# Load the Titanic dataset

5data = pd.read\_csv('Titanic-Dataset.csv')

6

7# Data Cleaning

8data['Age'].fillna(data['Age'].median(), inplace=True)

9data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

10data.drop('Cabin', axis=1, inplace=True)

11

12# Convert categorical features to numeric

13data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})

14data = pd.get\_dummies(data, columns=['Embarked'], drop\_first=True)

15

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

17plt.figure(figsize=(8,6))

18data.boxplot(column='Age', by='Pclass')

19plt.suptitle('')

20plt.title('Age by Pclass')

21plt.xlabel('Pclass')

22plt.ylabel('Age')

23plt.show()

24

25

TerminalTest cases

PrevResetSubmitNext

13:5507-48-2025



What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bd5/67e29db57c0b912d860eab45

CODETANTRAHome

202401090073@mitaoe.ac.inSupportLogout

5.2.8. Box Plot for Age by Survived

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

1. Use the **Survived** column to group the data for the boxplot (0 = Did not survive, 1 = Survived).

2. Set the title of the plot to **"Age by Survival"**.

3. Remove the default subtitle with **plt.suptitle("")**.

4. Label the x-axis as **'Survived'** and the y-axis as **'Age'**.

The Titanic dataset contains columns as shown below.

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

Sample Data:

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked  
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,N/S 21171,7.25,,S  
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17559,71.2833,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,35,1,0,113503,53.1,C125,S  
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S  
6,0,3,"Moran, Mr. James",male,0,0,0,240077,0.4583,,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 Achen)",female,14,1,0,237736,30.0708,,C

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

Sample Test Cases

BoxPlotF...

1import pandas as pd

2import matplotlib.pyplot as plt

3

4# Load the Titanic dataset

5data = pd.read\_csv('Titanic-Dataset.csv')

6

7# Data Cleaning

8data['Age'].fillna(data['Age'].median(), inplace=True)

9data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

10data.drop('Cabin', axis=1, inplace=True)

11

12# Convert categorical features to numeric

13data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})

14data = pd.get\_dummies(data, columns=['Embarked'], drop\_first=True)

15

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

17plt.figure(figsize=(8, 6))

18data.boxplot(column='Age', by='Survived')

19plt.suptitle('')

20plt.title('Age by Survival')

21plt.xlabel('Survived')

22plt.ylabel('Age')

23plt.show()

24

25

TerminalTest cases

PrevResetSubmitNext

13:5607-48-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bd5/67e2a1727c0b912d860eb39b

Home202401090073@mitaoe.ac.inSupportLogout

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,N/S 21171,7.25,,S  
2,1,1,"Cummings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17559,71.2833,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,35,1,0,113503,53.1,C125,S  
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S  
6,0,3,"Moran, Mr. James",male,0,0,0,000077,0.4583,,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 Achen)",female,14,1,0,237736,36.0708,,C

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

Sample Test Cases

BoxPlotF...

1import pandas as pd

2import matplotlib.pyplot as plt

3

4# Load the Titanic dataset

5data = pd.read\_csv('Titanic-Dataset.csv')

6

7# Data Cleaning

8data['Age'].fillna(data['Age'].median(), inplace=True)

9data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

10data.drop('Cabin', axis=1, inplace=True)

11

12# Convert categorical features to numeric

13data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})

14data = pd.get\_dummies(data, columns=['Embarked'], drop\_first=True)

15

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

17

18plt.figure(figsize=(8, 6))

19data.boxplot(column='Fare', by='Pclass')

20plt.suptitle('')

21plt.title('Fare by Pclass')

22plt.xlabel('Pclass')

23plt.ylabel('Fare')

24plt.show()

25

Terminal

Test cases

Prev

Reset

Submit

Next

13:5607-48-2025

What's NewCourseNew repository

mitaoe.codetantra.com/secure/course.jsp?eucld=6773e3f2f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bd5/67e2a2977c0b912d860eb621

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

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,22,1,0,4/5 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,26,0,0,5100/02 3161282,7.925,,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,35,0,0,373450,0.00,,S  
6,0,3,"Horan, Mr. James",male,0,0,338077,8.4583,,Q  
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.0625,E46,S  
8,0,3,"Palsson, Master. Gosta (Leonard)",male,2,3,1,940909,21.075,,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 Achen)",female,14,1,0,237736,26.0708,,C

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

Sample Test Cases

AgeFareS...

1import pandas as pd

2import matplotlib.pyplot as plt

3

4# Load the Titanic dataset

5data = pd.read\_csv('Titanic-Dataset.csv')

6

7# Data Cleaning

8data['Age'].fillna(data['Age'].median(), inplace=True)

9data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

10data.drop('Cabin', axis=1, inplace=True)

11

12# Convert categorical features to numeric

13data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})

14data = pd.get\_dummies(data, columns=['Embarked'], drop\_first=True)

15

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

17plt.figure(figsize=(6.4,4.8))

18plt.scatter(data['Age'],data['Fare'])

19plt.title('Age vs. Fare')

20plt.xlabel('Age')

21plt.ylabel('Fare')

22plt.show()

23

24

Terminal

Test cases

Prev

Reset

Submit

Next

13:5607-48-2025

What's NewCourseNew repositoryNew tab

mitaoe.codelantra.com/secure/course.jsp?eucld=6773e32f1f9c5320ca6bd5#contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e2a5387c0b912d860eba39

CODETANTRAHome202401090073@mitaoe.ac.inSupportLogout

5.2.11. Scatter Plot for Age vs. Fare by Survived00:45

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,22,1,0,0,21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,0,17559,71.2833,C85,C
3,1,1,"Heikkinen, Miss. Laina",female,26,0,0,0,5101/02, 31.00182,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,0,37450,8.05,,S
6,0,3,"Moran, Mr. James",male,0,0,0,0,340877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,0,340909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,0,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achen)",female,14,1,0,0,237736,30.0788,,C
```

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

Sample Test Cases

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
```

TerminalTest cases

PrevResetSubmit

13.9607-48-2025