

5.1.1. Stacked Plot

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

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

Sample Test Cases

stackedpl... Submit

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

Terminal Test cases

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.

Sample Test Cases

+

titanicDat... Submit

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 # write the code.
11 import pandas as pd
12 import matplotlib.pyplot as plt
13
14 df = pd.read_csv('titanic.csv')
15 fig, axes = plt.subplots(3, 2, figsize=(12,12))
16
17 #Plot 1: Count of passengers by class
18
19 axes[0, 0].bar(df['Pclass'].value_counts().index,
20 df['Pclass'].value_counts(), color='skyblue')
21
22 axes[0, 0].set_title("Passenger Class Distribution")
23 axes[0, 0].set_xlabel("Pclass")
24 axes[0, 0].set_ylabel("Count")
25 #Plot 2: Gender distribution
26
27 axes[0, 1].pie(df['Gender'].value_counts(),
28 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
29 ['lightblue', 'lightcoral'])
30
31 #Plot 3: Age distribution
32
33 axes[1, 0].hist(df['Age'], bins=10, color='lightcoral', edgecolor='black')
34 axes[1, 0].set_title("Age Distribution")
35 axes[1, 0].set_xlabel("Age")
36 axes[1, 0].set_ylabel("Frequency")
37
38 #Plot 4: Survival status distribution
39
40 axes[1, 1].pie(df['Survived'].value_counts(),
41 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
42 ['lightcoral', 'lightblue'])
43
44 #Plot 5: Fare distribution
45
46 axes[2, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
47 axes[2, 0].set_title("Fare Distribution")
48 axes[2, 0].set_xlabel("Fare")
49 axes[2, 0].set_ylabel("Frequency")
50
51 #Plot 6: Gender distribution
52
53 axes[2, 1].pie(df['Gender'].value_counts(),
54 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
55 ['lightblue', 'lightcoral'])
56
57 #Plot 7: Survival status distribution
58
59 axes[3, 0].pie(df['Survived'].value_counts(),
60 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
61 ['lightcoral', 'lightblue'])
62
63 #Plot 8: Fare distribution
64
65 axes[3, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
66 axes[3, 1].set_title("Fare Distribution")
67 axes[3, 1].set_xlabel("Fare")
68 axes[3, 1].set_ylabel("Frequency")
69
70 #Plot 9: Gender distribution
71
72 axes[4, 0].pie(df['Gender'].value_counts(),
73 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
74 ['lightblue', 'lightcoral'])
75
76 #Plot 10: Survival status distribution
77
78 axes[4, 1].pie(df['Survived'].value_counts(),
79 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
80 ['lightcoral', 'lightblue'])
81
82 #Plot 11: Fare distribution
83
84 axes[5, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
85 axes[5, 0].set_title("Fare Distribution")
86 axes[5, 0].set_xlabel("Fare")
87 axes[5, 0].set_ylabel("Frequency")
88
89 #Plot 12: Gender distribution
90
91 axes[5, 1].pie(df['Gender'].value_counts(),
92 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
93 ['lightblue', 'lightcoral'])
94
95 #Plot 13: Survival status distribution
96
97 axes[6, 0].pie(df['Survived'].value_counts(),
98 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
99 ['lightcoral', 'lightblue'])
100
101 #Plot 14: Fare distribution
102
103 axes[6, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
104 axes[6, 1].set_title("Fare Distribution")
105 axes[6, 1].set_xlabel("Fare")
106 axes[6, 1].set_ylabel("Frequency")
107
108 #Plot 15: Gender distribution
109
110 axes[7, 0].pie(df['Gender'].value_counts(),
111 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
112 ['lightblue', 'lightcoral'])
113
114 #Plot 16: Survival status distribution
115
116 axes[7, 1].pie(df['Survived'].value_counts(),
117 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
118 ['lightcoral', 'lightblue'])
119
120 #Plot 17: Fare distribution
121
122 axes[8, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
123 axes[8, 0].set_title("Fare Distribution")
124 axes[8, 0].set_xlabel("Fare")
125 axes[8, 0].set_ylabel("Frequency")
126
127 #Plot 18: Gender distribution
128
129 axes[8, 1].pie(df['Gender'].value_counts(),
130 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
131 ['lightblue', 'lightcoral'])
132
133 #Plot 19: Survival status distribution
134
135 axes[9, 0].pie(df['Survived'].value_counts(),
136 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
137 ['lightcoral', 'lightblue'])
138
139 #Plot 20: Fare distribution
140
141 axes[9, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
142 axes[9, 1].set_title("Fare Distribution")
143 axes[9, 1].set_xlabel("Fare")
144 axes[9, 1].set_ylabel("Frequency")
145
146 #Plot 21: Gender distribution
147
148 axes[10, 0].pie(df['Gender'].value_counts(),
149 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
150 ['lightblue', 'lightcoral'])
151
152 #Plot 22: Survival status distribution
153
154 axes[10, 1].pie(df['Survived'].value_counts(),
155 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
156 ['lightcoral', 'lightblue'])
157
158 #Plot 23: Fare distribution
159
160 axes[11, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
161 axes[11, 0].set_title("Fare Distribution")
162 axes[11, 0].set_xlabel("Fare")
163 axes[11, 0].set_ylabel("Frequency")
164
165 #Plot 24: Gender distribution
166
167 axes[11, 1].pie(df['Gender'].value_counts(),
168 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
169 ['lightblue', 'lightcoral'])
170
171 #Plot 25: Survival status distribution
172
173 axes[12, 0].pie(df['Survived'].value_counts(),
174 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
175 ['lightcoral', 'lightblue'])
176
177 #Plot 26: Fare distribution
178
179 axes[12, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
180 axes[12, 1].set_title("Fare Distribution")
181 axes[12, 1].set_xlabel("Fare")
182 axes[12, 1].set_ylabel("Frequency")
183
184 #Plot 27: Gender distribution
185
186 axes[13, 0].pie(df['Gender'].value_counts(),
187 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
188 ['lightblue', 'lightcoral'])
189
190 #Plot 28: Survival status distribution
191
192 axes[13, 1].pie(df['Survived'].value_counts(),
193 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
194 ['lightcoral', 'lightblue'])
195
196 #Plot 29: Fare distribution
197
198 axes[14, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
199 axes[14, 0].set_title("Fare Distribution")
200 axes[14, 0].set_xlabel("Fare")
201 axes[14, 0].set_ylabel("Frequency")
202
203 #Plot 30: Gender distribution
204
205 axes[14, 1].pie(df['Gender'].value_counts(),
206 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
207 ['lightblue', 'lightcoral'])
208
209 #Plot 31: Survival status distribution
210
211 axes[15, 0].pie(df['Survived'].value_counts(),
212 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
213 ['lightcoral', 'lightblue'])
214
215 #Plot 32: Fare distribution
216
217 axes[15, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
218 axes[15, 1].set_title("Fare Distribution")
219 axes[15, 1].set_xlabel("Fare")
220 axes[15, 1].set_ylabel("Frequency")
221
222 #Plot 33: Gender distribution
223
224 axes[16, 0].pie(df['Gender'].value_counts(),
225 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
226 ['lightblue', 'lightcoral'])
227
228 #Plot 34: Survival status distribution
229
230 axes[16, 1].pie(df['Survived'].value_counts(),
231 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
232 ['lightcoral', 'lightblue'])
233
234 #Plot 35: Fare distribution
235
236 axes[17, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
237 axes[17, 0].set_title("Fare Distribution")
238 axes[17, 0].set_xlabel("Fare")
239 axes[17, 0].set_ylabel("Frequency")
240
241 #Plot 36: Gender distribution
242
243 axes[17, 1].pie(df['Gender'].value_counts(),
244 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
245 ['lightblue', 'lightcoral'])
246
247 #Plot 37: Survival status distribution
248
249 axes[18, 0].pie(df['Survived'].value_counts(),
250 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
251 ['lightcoral', 'lightblue'])
252
253 #Plot 38: Fare distribution
254
255 axes[18, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
256 axes[18, 1].set_title("Fare Distribution")
257 axes[18, 1].set_xlabel("Fare")
258 axes[18, 1].set_ylabel("Frequency")
259
260 #Plot 39: Gender distribution
261
262 axes[19, 0].pie(df['Gender'].value_counts(),
263 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
264 ['lightblue', 'lightcoral'])
265
266 #Plot 40: Survival status distribution
267
268 axes[19, 1].pie(df['Survived'].value_counts(),
269 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
270 ['lightcoral', 'lightblue'])
271
272 #Plot 41: Fare distribution
273
274 axes[20, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
275 axes[20, 0].set_title("Fare Distribution")
276 axes[20, 0].set_xlabel("Fare")
277 axes[20, 0].set_ylabel("Frequency")
278
279 #Plot 42: Gender distribution
280
281 axes[20, 1].pie(df['Gender'].value_counts(),
282 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
283 ['lightblue', 'lightcoral'])
284
285 #Plot 43: Survival status distribution
286
287 axes[21, 0].pie(df['Survived'].value_counts(),
288 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
289 ['lightcoral', 'lightblue'])
290
291 #Plot 44: Fare distribution
292
293 axes[21, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
294 axes[21, 1].set_title("Fare Distribution")
295 axes[21, 1].set_xlabel("Fare")
296 axes[21, 1].set_ylabel("Frequency")
297
298 #Plot 45: Gender distribution
299
300 axes[22, 0].pie(df['Gender'].value_counts(),
301 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
302 ['lightblue', 'lightcoral'])
303
304 #Plot 46: Survival status distribution
305
306 axes[22, 1].pie(df['Survived'].value_counts(),
307 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
308 ['lightcoral', 'lightblue'])
309
310 #Plot 47: Fare distribution
311
312 axes[23, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
313 axes[23, 0].set_title("Fare Distribution")
314 axes[23, 0].set_xlabel("Fare")
315 axes[23, 0].set_ylabel("Frequency")
316
317 #Plot 48: Gender distribution
318
319 axes[23, 1].pie(df['Gender'].value_counts(),
320 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
321 ['lightblue', 'lightcoral'])
322
323 #Plot 49: Survival status distribution
324
325 axes[24, 0].pie(df['Survived'].value_counts(),
326 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
327 ['lightcoral', 'lightblue'])
328
329 #Plot 50: Fare distribution
330
331 axes[24, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
332 axes[24, 1].set_title("Fare Distribution")
333 axes[24, 1].set_xlabel("Fare")
334 axes[24, 1].set_ylabel("Frequency")
335
336 #Plot 51: Gender distribution
337
338 axes[25, 0].pie(df['Gender'].value_counts(),
339 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
340 ['lightblue', 'lightcoral'])
341
342 #Plot 52: Survival status distribution
343
344 axes[25, 1].pie(df['Survived'].value_counts(),
345 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
346 ['lightcoral', 'lightblue'])
347
348 #Plot 53: Fare distribution
349
350 axes[26, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
351 axes[26, 0].set_title("Fare Distribution")
352 axes[26, 0].set_xlabel("Fare")
353 axes[26, 0].set_ylabel("Frequency")
354
355 #Plot 54: Gender distribution
356
357 axes[26, 1].pie(df['Gender'].value_counts(),
358 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
359 ['lightblue', 'lightcoral'])
360
361 #Plot 55: Survival status distribution
362
363 axes[27, 0].pie(df['Survived'].value_counts(),
364 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
365 ['lightcoral', 'lightblue'])
366
367 #Plot 56: Fare distribution
368
369 axes[27, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
370 axes[27, 1].set_title("Fare Distribution")
371 axes[27, 1].set_xlabel("Fare")
372 axes[27, 1].set_ylabel("Frequency")
373
374 #Plot 57: Gender distribution
375
376 axes[28, 0].pie(df['Gender'].value_counts(),
377 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
378 ['lightblue', 'lightcoral'])
379
380 #Plot 58: Survival status distribution
381
382 axes[28, 1].pie(df['Survived'].value_counts(),
383 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
384 ['lightcoral', 'lightblue'])
385
386 #Plot 59: Fare distribution
387
388 axes[29, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
389 axes[29, 0].set_title("Fare Distribution")
390 axes[29, 0].set_xlabel("Fare")
391 axes[29, 0].set_ylabel("Frequency")
392
393 #Plot 60: Gender distribution
394
395 axes[29, 1].pie(df['Gender'].value_counts(),
396 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
397 ['lightblue', 'lightcoral'])
398
399 #Plot 61: Survival status distribution
400
401 axes[30, 0].pie(df['Survived'].value_counts(),
402 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
403 ['lightcoral', 'lightblue'])
404
405 #Plot 62: Fare distribution
406
407 axes[30, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
408 axes[30, 1].set_title("Fare Distribution")
409 axes[30, 1].set_xlabel("Fare")
410 axes[30, 1].set_ylabel("Frequency")
411
412 #Plot 63: Gender distribution
413
414 axes[31, 0].pie(df['Gender'].value_counts(),
415 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
416 ['lightblue', 'lightcoral'])
417
418 #Plot 64: Survival status distribution
419
420 axes[31, 1].pie(df['Survived'].value_counts(),
421 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
422 ['lightcoral', 'lightblue'])
423
424 #Plot 65: Fare distribution
425
426 axes[32, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
427 axes[32, 0].set_title("Fare Distribution")
428 axes[32, 0].set_xlabel("Fare")
429 axes[32, 0].set_ylabel("Frequency")
430
431 #Plot 66: Gender distribution
432
433 axes[32, 1].pie(df['Gender'].value_counts(),
434 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
435 ['lightblue', 'lightcoral'])
436
437 #Plot 67: Survival status distribution
438
439 axes[33, 0].pie(df['Survived'].value_counts(),
440 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
441 ['lightcoral', 'lightblue'])
442
443 #Plot 68: Fare distribution
444
445 axes[33, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
446 axes[33, 1].set_title("Fare Distribution")
447 axes[33, 1].set_xlabel("Fare")
448 axes[33, 1].set_ylabel("Frequency")
449
450 #Plot 69: Gender distribution
451
452 axes[34, 0].pie(df['Gender'].value_counts(),
453 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
454 ['lightblue', 'lightcoral'])
455
456 #Plot 70: Survival status distribution
457
458 axes[34, 1].pie(df['Survived'].value_counts(),
459 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
460 ['lightcoral', 'lightblue'])
461
462 #Plot 71: Fare distribution
463
464 axes[35, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
465 axes[35, 0].set_title("Fare Distribution")
466 axes[35, 0].set_xlabel("Fare")
467 axes[35, 0].set_ylabel("Frequency")
468
469 #Plot 72: Gender distribution
470
471 axes[35, 1].pie(df['Gender'].value_counts(),
472 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
473 ['lightblue', 'lightcoral'])
474
475 #Plot 73: Survival status distribution
476
477 axes[36, 0].pie(df['Survived'].value_counts(),
478 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
479 ['lightcoral', 'lightblue'])
480
481 #Plot 74: Fare distribution
482
483 axes[36, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
484 axes[36, 1].set_title("Fare Distribution")
485 axes[36, 1].set_xlabel("Fare")
486 axes[36, 1].set_ylabel("Frequency")
487
488 #Plot 75: Gender distribution
489
490 axes[37, 0].pie(df['Gender'].value_counts(),
491 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
492 ['lightblue', 'lightcoral'])
493
494 #Plot 76: Survival status distribution
495
496 axes[37, 1].pie(df['Survived'].value_counts(),
497 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
498 ['lightcoral', 'lightblue'])
499
500 #Plot 77: Fare distribution
501
502 axes[38, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
503 axes[38, 0].set_title("Fare Distribution")
504 axes[38, 0].set_xlabel("Fare")
505 axes[38, 0].set_ylabel("Frequency")
506
507 #Plot 78: Gender distribution
508
509 axes[38, 1].pie(df['Gender'].value_counts(),
510 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
511 ['lightblue', 'lightcoral'])
512
513 #Plot 79: Survival status distribution
514
515 axes[39, 0].pie(df['Survived'].value_counts(),
516 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
517 ['lightcoral', 'lightblue'])
518
519 #Plot 80: Fare distribution
520
521 axes[39, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
522 axes[39, 1].set_title("Fare Distribution")
523 axes[39, 1].set_xlabel("Fare")
524 axes[39, 1].set_ylabel("Frequency")
525
526 #Plot 81: Gender distribution
527
528 axes[40, 0].pie(df['Gender'].value_counts(),
529 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
530 ['lightblue', 'lightcoral'])
531
532 #Plot 82: Survival status distribution
533
534 axes[40, 1].pie(df['Survived'].value_counts(),
535 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
536 ['lightcoral', 'lightblue'])
537
538 #Plot 83: Fare distribution
539
540 axes[41, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
541 axes[41, 0].set_title("Fare Distribution")
542 axes[41, 0].set_xlabel("Fare")
543 axes[41, 0].set_ylabel("Frequency")
544
545 #Plot 84: Gender distribution
546
547 axes[41, 1].pie(df['Gender'].value_counts(),
548 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
549 ['lightblue', 'lightcoral'])
550
551 #Plot 85: Survival status distribution
552
553 axes[42, 0].pie(df['Survived'].value_counts(),
554 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
555 ['lightcoral', 'lightblue'])
556
557 #Plot 86: Fare distribution
558
559 axes[42, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
560 axes[42, 1].set_title("Fare Distribution")
561 axes[42, 1].set_xlabel("Fare")
562 axes[42, 1].set_ylabel("Frequency")
563
564 #Plot 87: Gender distribution
565
566 axes[43, 0].pie(df['Gender'].value_counts(),
567 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
568 ['lightblue', 'lightcoral'])
569
570 #Plot 88: Survival status distribution
571
572 axes[43, 1].pie(df['Survived'].value_counts(),
573 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
574 ['lightcoral', 'lightblue'])
575
576 #Plot 89: Fare distribution
577
578 axes[44, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
579 axes[44, 0].set_title("Fare Distribution")
580 axes[44, 0].set_xlabel("Fare")
581 axes[44, 0].set_ylabel("Frequency")
582
583 #Plot 90: Gender distribution
584
585 axes[44, 1].pie(df['Gender'].value_counts(),
586 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
587 ['lightblue', 'lightcoral'])
588
589 #Plot 91: Survival status distribution
590
591 axes[45, 0].pie(df['Survived'].value_counts(),
592 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
593 ['lightcoral', 'lightblue'])
594
595 #Plot 92: Fare distribution
596
597 axes[45, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
598 axes[45, 1].set_title("Fare Distribution")
599 axes[45, 1].set_xlabel("Fare")
600 axes[45, 1].set_ylabel("Frequency")
601
602 #Plot 93: Gender distribution
603
604 axes[46, 0].pie(df['Gender'].value_counts(),
605 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
606 ['lightblue', 'lightcoral'])
607
608 #Plot 94: Survival status distribution
609
610 axes[46, 1].pie(df['Survived'].value_counts(),
611 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
612 ['lightcoral', 'lightblue'])
613
614 #Plot 95: Fare distribution
615
616 axes[47, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
617 axes[47, 0].set_title("Fare Distribution")
618 axes[47, 0].set_xlabel("Fare")
619 axes[47, 0].set_ylabel("Frequency")
620
621 #Plot 96: Gender distribution
622
623 axes[47, 1].pie(df['Gender'].value_counts(),
624 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
625 ['lightblue', 'lightcoral'])
626
627 #Plot 97: Survival status distribution
628
629 axes[48, 0].pie(df['Survived'].value_counts(),
630 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
631 ['lightcoral', 'lightblue'])
632
633 #Plot 98: Fare distribution
634
635 axes[48, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
636 axes[48, 1].set_title("Fare Distribution")
637 axes[48, 1].set_xlabel("Fare")
638 axes[48, 1].set_ylabel("Frequency")
639
640 #Plot 99: Gender distribution
641
642 axes[49, 0].pie(df['Gender'].value_counts(),
643 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
644 ['lightblue', 'lightcoral'])
645
646 #Plot 100: Survival status distribution
647
648 axes[49, 1].pie(df['Survived'].value_counts(),
649 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
650 ['lightcoral', 'lightblue'])
651
652 #Plot 101: Fare distribution
653
654 axes[50, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
655 axes[50, 0].set_title("Fare Distribution")
656 axes[50, 0].set_xlabel("Fare")
657 axes[50, 0].set_ylabel("Frequency")
658
659 #Plot 102: Gender distribution
660
661 axes[50, 1].pie(df['Gender'].value_counts(),
662 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
663 ['lightblue', 'lightcoral'])
664
665 #Plot 103: Survival status distribution
666
667 axes[51, 0].pie(df['Survived'].value_counts(),
668 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
669 ['lightcoral', 'lightblue'])
670
671 #Plot 104: Fare distribution
672
673 axes[51, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
674 axes[51, 1].set_title("Fare Distribution")
675 axes[51, 1].set_xlabel("Fare")
676 axes[51, 1].set_ylabel("Frequency")
677
678 #Plot 105: Gender distribution
679
680 axes[52, 0].pie(df['Gender'].value_counts(),
681 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
682 ['lightblue', 'lightcoral'])
683
684 #Plot 106: Survival status distribution
685
686 axes[52, 1].pie(df['Survived'].value_counts(),
687 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
688 ['lightcoral', 'lightblue'])
689
690 #Plot 107: Fare distribution
691
692 axes[53, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
693 axes[53, 0].set_title("Fare Distribution")
694 axes[53, 0].set_xlabel("Fare")
695 axes[53, 0].set_ylabel("Frequency")
696
697 #Plot 108: Gender distribution
698
699 axes[53, 1].pie(df['Gender'].value_counts(),
700 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
701 ['lightblue', 'lightcoral'])
702
703 #Plot 109: Survival status distribution
704
705 axes[54, 0].pie(df['Survived'].value_counts(),
706 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
707 ['lightcoral', 'lightblue'])
708
709 #Plot 110: Fare distribution
710
711 axes[54, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
712 axes[54, 1].set_title("Fare Distribution")
713 axes[54, 1].set_xlabel("Fare")
714 axes[54, 1].set_ylabel("Frequency")
715
716 #Plot 111: Gender distribution
717
718 axes[55, 0].pie(df['Gender'].value_counts(),
719 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
720 ['lightblue', 'lightcoral'])
721
722 #Plot 112: Survival status distribution
723
724 axes[55, 1].pie(df['Survived'].value_counts(),
725 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
726 ['lightcoral', 'lightblue'])
727
728 #Plot 113: Fare distribution
729
730 axes[56, 0].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
731 axes[56, 0].set_title("Fare Distribution")
732 axes[56, 0].set_xlabel("Fare")
733 axes[56, 0].set_ylabel("Frequency")
734
735 #Plot 114: Gender distribution
736
737 axes[56, 1].pie(df['Gender'].value_counts(),
738 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
739 ['lightblue', 'lightcoral'])
740
741 #Plot 115: Survival status distribution
742
743 axes[57, 0].pie(df['Survived'].value_counts(),
744 labels=df['Survived'].value_counts().index, autopct='%1.1f%%', colors=
745 ['lightcoral', 'lightblue'])
746
747 #Plot 116: Fare distribution
748
749 axes[57, 1].hist(df['Fare'], bins=10, color='lightblue', edgecolor='black')
750 axes[57, 1].set_title("Fare Distribution")
751 axes[57, 1].set_xlabel("Fare")
752 axes[57, 1].set_ylabel("Frequency")
753
754 #Plot 117: Gender distribution
755
756 axes[58, 0].pie(df['Gender'].value_counts(),
757 labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=
758 ['lightblue', 'lightcoral'])
759
760 #Plot 118: Survival status distribution
761
762 axes[58, 1].pie(df['Survived'].value_counts(),
763

5.2.2. Histogram of passenger information of Titanic

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

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

The Titanic dataset contains columns as shown below,

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

Sample Data:

```
PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
```

Sample Test Cases +

Histogra... Submit

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

Terminal Test cases

5.2.3. Bar plot of survival rate of passengers

10:25

Write a Python code to plot a bar chart that shows the count of passengers who survived and did not survive in the Titanic dataset. The chart should display the following specifications:
1. Use the 'Survived' column to show the count of survivors (0 = Did not survive, 1 = Survived).
2. Set the chart type to 'bar'.
3. Add the title "Survival Count" to the chart.
4. Label the x-axis as 'Survived' and the y-axis as 'Count'.

The Titanic dataset contains columns as shown below,

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

Sample Data:

PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S

Sample Test Cases

BarPlotOf...

Submit

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()

Terminal

Test cases

< Prev

Reset

Submit

Next >

5.2.4. Bar Plot for Survival by Gender

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

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

The Titanic dataset contains columns as shown below,

| 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
```

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 survival_by_gender = data.groupby('Sex')
18 ['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
```


5.2.5. Bar Plot for Survival by Pclass

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

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

The Titanic dataset contains columns as shown below,

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

Sample Data:

```
PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
```

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 Pclass
17 survival_by_class = data.groupby('Pclass')
18 ['Survived'].value_counts().unstack().fillna(0)
19 survival_by_class.columns = ['Not Survived', 'Survived']
20
21 survival_by_class.plot(kind='bar', stacked=True)
22 plt.title('Survival by Pclass')
23 plt.xlabel('Pclass')
24 plt.ylabel('Count')
25 plt.legend(title=None)
26 plt.show()
27
28
```

Terminal

Test cases

5.2.6. Bar Plot for Survival by Embarked

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

The chart should display the following specifications:

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

The Titanic dataset contains columns as shown below,

| PassengerId | Survived | Class | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|-------------|----------|-------|------|-----|-----|-------|-------|--------|------|-------|----------|
| | | | | | | | | | | | |

Sample Data:

PassengerId, Survived, Class, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked

Sample Test Cases

```
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 survival_by_class = data.groupby('Embarked_Q')
18 ['Survived'].value_counts().unstack().fillna(0)
19 survival_by_class.columns = ['Not Survived', 'Survived']
20
21 survival_by_class.plot(kind='bar', stacked=True)
22 plt.title('Survival by Embarked')
23 plt.xlabel('Embarked')
24 plt.ylabel('Count')
25 plt.legend (title=None)
26 plt.show()
27
28
29
```

5.2.7. Box plot for Age Distribution

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

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

The Titanic dataset contains columns as shown below,

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

Sample Data:

```
PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
```

Sample Test Cases

BoxPlotF...

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Box Plot for Age by Pclass
17 plt.figure(figsize=(8,6))
18 data.boxplot(column='Age', by='Pclass')
19 plt.suptitle('')
20 plt.title('Age by Pclass')
21 plt.xlabel('Pclass')
22 plt.ylabel('Count')
23
24 plt.show()
25
26
```

Terminal Test cases

5.2.8. Box Plot for Age by Survived

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

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

The Titanic dataset contains columns as shown below,

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

Sample Data:

```
PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,0
```

Sample Test Cases



BoxPlotF...

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Box Plot for Age by Survived
17 plt.figure(figsize=(8,6))
18 data.boxplot(column='Age', by='Survived')
19 plt.suptitle('')
20 plt.title('Age by Survival')
21 plt.xlabel('Survived')
22 plt.ylabel('Age')
23
24 plt.show()
25
26
27
28
```

5.2.8. Box Plot for Age by Survived

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

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

The Titanic dataset contains columns as shown below,

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

Sample Data:

```
PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
```

Sample Test Cases

BoxPlotF... Submit

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Box Plot for Age by Survived
17 plt.figure(figsize=(8,6))
18 data.boxplot(column='Age', by='Survived')
19 plt.suptitle('')
20 plt.title('Age by Survival')
21 plt.xlabel('Survived')
22 plt.ylabel('Age')
23
24 plt.show()
25
26
27
28
```

Terminal Test cases

5.2.9. Box Plot for Fare by Pclass

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

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

The Titanic dataset contains columns as shown below,

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

Sample Data:

```
PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,4/5 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,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
```

Sample Test Cases

BoxPlotF...

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Box Plot for Fare by Pclass
17 plt.figure(figsize=(8,6))
18 data.boxplot(column='Fare', by='Pclass')
19 plt.suptitle('')
20 plt.title('Fare by Pclass')
21 plt.xlabel('Pclass')
22 plt.ylabel('Fare')
23
24 plt.show()
25
26
27
28
```

5.2.10. Scatter Plot for Age vs. Fare

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

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

The Titanic dataset contains columns as shown below,

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

Sample Data:

```
PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.0625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
```

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 Box Plot for Fare by Pclass

17

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

18

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

19

20

plt.title('Age vs. Fare')

21

plt.xlabel('Age')

22

plt.ylabel('Fare')

23

24

plt.show()

25

26

Terminal

Test cases

5.2.11. Scatter Plot for Age vs. Fare by Survived

04:45

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:

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

The Titanic dataset contains columns as shown below,

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

Sample Data:

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

Sample Test Cases

+

AgeFareS...

Submit

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 Scatter Plot for Age vs. Fare by Survived

17colors = data['Survived'].map({0: 'red', 1: 'blue'})

18

19plt.scatter(data['Age'], data['Fare'], c=colors)

20plt.title('Age vs. Fare by Survival')

21plt.xlabel('Age')

22plt.ylabel('Fare')

23plt.show()

24

25

26

27

28

29

Terminal

Test cases

Debugger