

Python Programming

Lab:- 24(Pandas Series)

Student Id:- AF0417098

Student Name:- Ankush

Panda is a powerful Python library used for data manipulation and analysis. It provides two main data structures:

- **Series:** A one-dimensional labeled array (similar to a list or column in a table).
- **DataFrame:** A two-dimensional labeled data structure, similar to a table or Excel spreadsheet.

Here are some key feature and components of the Pandas library:-

DataFrame: The DataFrame is one of the central data structures in Pandas. It is a two-dimensional table with rows and columns, similar to a spreadsheet or a SQL table. Each column in a DataFrame can have a different data type, making it suitable for heterogeneous and structured data.

Series: A Series is a one-dimensional array-like object in Pandas. It can be thought of as a single column of data within a DataFrame, with an associated index. Series are used for representing and working with one-dimensional data.

Data Input and Output: Pandas provides functions to read data from various file formats, such as CSV, Excel, SQL databases, and more. It also allows you to write data to these formats.

Data Cleaning: Pandas includes functions for handling missing data, removing duplicates, and transforming data to make it suitable for analysis.

Data Transformation: You can perform various data transformation operations, such as filtering, merging, and reshaping data. This is essential for data preprocessing and analysis.

Grouping and Aggregation: Pandas allows you to group data by one or more columns and perform aggregation operations on the groups. This is valuable for summarizing and analyzing data.

Indexing and Selection: Pandas provides powerful indexing capabilities, including label-based and integer-based indexing. This allows you to select and access specific parts of the data easily.

Assignment Questions:-

Ques1:- We have to create a Pandas series [10,20,30,40,50] without index .

Introduction to Pandas series:

A pandas Series is a one-dimensional labeled array in Python, capable of holding any data type (integers, strings, floats, etc.). It is similar to a list or array, but with an index that labels each data point. This index allows for easier data manipulation and retrieval. Series are highly efficient and are commonly used for handling structured data in a flexible, accessible way.

Program:-

```
0024.py > ...
1  import pandas as pd
2  data = [10, 20, 30, 40, 50] # This is the list [10, 20, 30, 40, 50] that you're passing to the pd.Series() function.
3  # Each value in the list will become an element in the Series.
4
5  series = pd.Series(data)
6  # This is a function from the pandas library that creates a one-dimensional labeled array, called a Series.
7  # A Series is similar to a list or an array, but with additional features like an index, making it more powerful for handling data.
8
9  print(series)
10 |
```

Output:-

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS PORTS

Users\Raj Kumar\Desktop\python programming\lab24.py"
0 10
1 20
2 30
3 40
4 50
dtype: int64
PS C:\Users\Raj Kumar\Desktop\python programming>
PS C:\Users\Raj Kumar\Desktop\python programming>
PS C:\Users\Raj Kumar\Desktop\python programming>
```

Ques 2.What is Create Pandas and also give one DataFrame example .

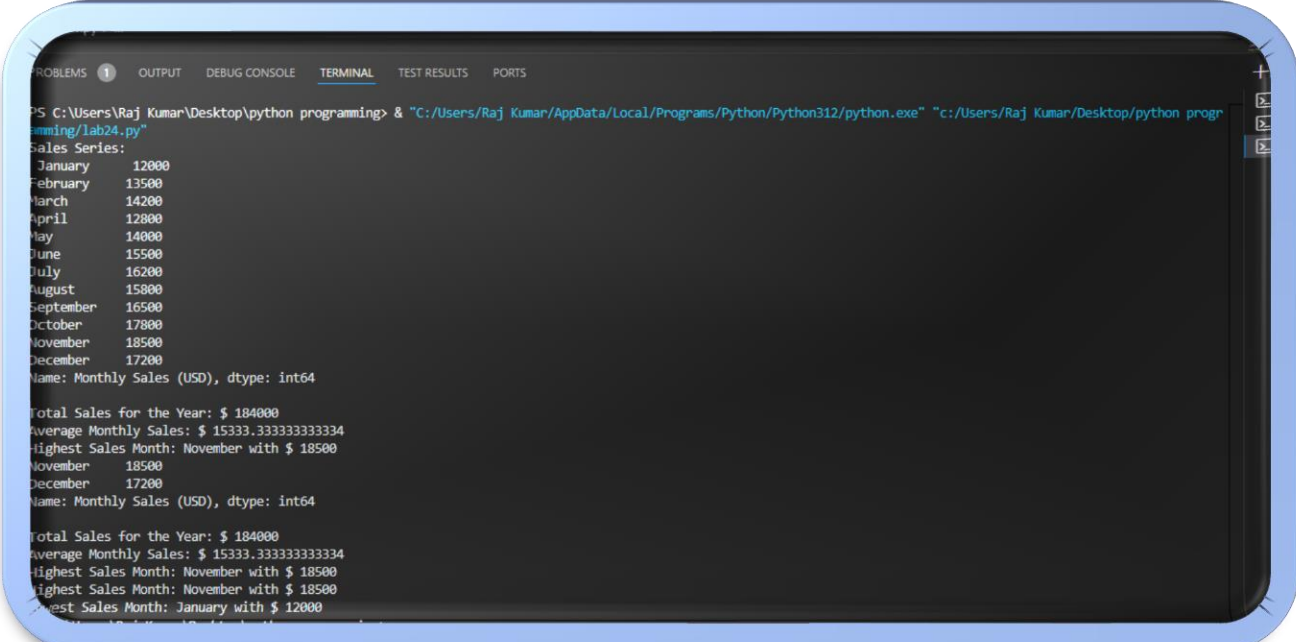
How to create a Pandas data frame:

To create a pandas DataFrame, you can pass data in various forms such as dictionaries, lists, or arrays to the `pd.DataFrame()` function. A DataFrame is essentially a two-dimensional, labeled data structure, similar to a table in Excel, where each column can be of different data types. It is widely used in data analysis for handling structured data.

Program:-

```
lab24.py > ...
10
11 import pandas as pd
12
13 # Months in a year
14 months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
15           'September', 'October', 'November', 'December']
16
17 # Monthly sales data for a product (example data)
18 sales_data = [12000, 13500, 14200, 12800, 14000, 15500, 16200, 15800, 16500, 17800, 18500, 17200]
19
20 # Create a Pandas Series with months as the index
21 sales_series = pd.Series(sales_data, index=months, name='Monthly Sales (USD)')
22
23 # Display the Series
24 print("Sales Series:\n", sales_series)
25
26 # Additional Analysis
27 print("\nTotal Sales for the Year: $", sales_series.sum())
28 print("Average Monthly Sales: $", sales_series.mean())
29 print("Highest Sales Month:", sales_series.idxmax(), "with $", sales_series.max())
30 print("Lowest Sales Month:", sales_series.idxmin(), "with $", sales_series.min())
31
32 # Plotting the sales data (if matplotlib is installed)
33 # Uncomment the next three lines if you want to see a plot
34 # import matplotlib.pyplot as plt
35 # sales_series.plot(kind='bar', title='Monthly Sales (USD)')
36 # plt.show()
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
```

Output:-



```
PS C:\Users\Raj Kumar\Desktop\python programming> & "C:/Users/Raj Kumar/AppData/Local/Programs/Python/Python312/python.exe" "c:/Users/Raj Kumar/Desktop/python programming/lab24.py"
Sales Series:
January      12000
February     13500
March        14200
April        12800
May          14000
June         15500
July         16200
August       15800
September    16500
October      17800
November     18500
December     17200
Name: Monthly Sales (USD), dtype: int64

Total Sales for the Year: $ 184000
Average Monthly Sales: $ 15333.333333333334
Highest Sales Month: November with $ 18500
November      18500
December      17200
Name: Monthly Sales (USD), dtype: int64

Total Sales for the Year: $ 184000
Average Monthly Sales: $ 15333.333333333334
Highest Sales Month: November with $ 18500
Highest Sales Month: November with $ 18500
Lowest Sales Month: January with $ 12000
```

Ques 3. Monthly Sales Data Imagine you are a sales manager for a retail company, and you want to analyze the monthly sales performance of a particular product in a given year. You have recorded the monthly sales figures for that product, and you want to represent this data using a Pandas Series.

Program:-

```

lab24.py > _
38 import pandas as pd
39 import matplotlib.pyplot as plt
40
41 # List of months in the year
42 months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
43           'September', 'October', 'November', 'December']
44
45 # Recorded monthly sales data (example figures in USD)
46 sales_data = [12000, 13500, 14200, 12800, 14000, 15500, 16200, 15800, 16500, 17800, 18500, 17200]
47
48 # Create a Pandas Series with sales data, using months as the index
49 sales_series = pd.Series(sales_data, index=months, name='Monthly Sales (USD)')
50
51 # Identify the month with the highest and lowest sales
52 highest_sales_month = sales_series.idxmax()
53 lowest_sales_month = sales_series.idxmin()
54
55 # Plotting the sales data using matplotlib's bar chart
56 plt.figure(figsize=(12, 6))
57
58 # Define bar colors: highlight the highest and lowest sales months
59 bar_colors = ['lightcoral' if month == highest_sales_month else
60               'lightblue' if month == lowest_sales_month else 'skyblue' for month in months]
61
62 # Create bar chart with months on x-axis and sales data on y-axis
63 bars = plt.bar(months, sales_data, color=bar_colors)
64
65 # Adding chart title and labels
66 plt.title('Monthly Sales Performance', fontsize=16)
67 plt.xlabel('Month', fontsize=12)
68 plt.ylabel('Sales (USD)', fontsize=12)
69
70 # Rotate x-axis labels for better readability
71 plt.xticks(rotation=45)
72
73 # Add data labels (sales values) on top of each bar
74 for bar in bars:
75     height = bar.get_height()
76     plt.text(bar.get_x() + bar.get_width()/2., height - 1000, f'{int(height)}',
77              ha='center', va='bottom', color='black', fontsize=10, fontweight='bold')
78
79 # Highlight the highest and lowest sales months in the legend
80 plt.legend(['Highest Sales Month', 'Lowest Sales Month', 'Other Months'], loc='upper left')
81
82 # Display the chart
83 plt.tight_layout()
84 plt.show()

```

```

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

```

Output:-

```
Monthly Sales Data:
January      12000
February     13500
March        14200
April        12800
May          14000
June         15500
July         16200
August       15800
September    16500
October      17800
November     18500
December     17200
Name: Monthly Sales (USD), dtype: int64
July         16200
August       15800
September    16500
October      17800
July         16200
August       15800
September    16500
October      17800
November     18500
December     17200
Name: Monthly Sales (USD), dtype: int64

Total Sales for the Year: $ 184000
Average Monthly Sales: $ 15333.333333333334
Highest Sales Month: November with $18500
Lowest Sales Month: January with $12000
```

Matplotlib chart:-

