

Python 3.11.7 | packaged by Anaconda, Inc. | (main, Dec 15 2023, 18:05:47) [MSC v.1916 64 bit (AMD64)]

Type "copyright", "credits" or "license" for more information.

IPython 8.20.0 -- An enhanced Interactive Python.

```
In [1]: import pandas as pd
...: import numpy as np
```

```
In [2]: # Destinating Pieces files - merge these two datasets
```

```
In [3]: DestinatingPieces1 = pd.read_csv("C:/Users/andie/OneDrive/Documents/Capstone/
Mail_v2/Piece Output v2/Destinating Pieces pt.1 v2.csv")
```

```
In [4]: DestinatingPieces2 = pd.read_csv("C:/Users/andie/OneDrive/Documents/Capstone/
Mail_v2/Piece Output v2/Destinating Pieces pt.2 v2.csv")
```

```
In [5]: #----- Filter out data by ACTUAL_DLVRY_DATE that isn't between 01/08/24 to
01/21/24
```

```
In [6]: dp_january = DestinatingPieces1[(DestinatingPieces1['ACTUAL_DLVRY_DATE'] >
"2024-01-08") & (DestinatingPieces1['ACTUAL_DLVRY_DATE'] < "2024-01-21")]
```

```
In [7]: dp2_january = DestinatingPieces2[(DestinatingPieces2['ACTUAL_DLVRY_DATE'] >
"2024-01-08") & (DestinatingPieces2['ACTUAL_DLVRY_DATE'] < "2024-01-21")]
```

```
In [8]: # Drop null and duplicate entries
```

```
In [9]: noNull_dp1 = dp_january.dropna(how='any',axis=0)
```

```
In [10]: clean_dp1 = noNull_dp1.drop_duplicates()
```

```
In [11]: noNull_dp2 = dp2_january.dropna(how='any',axis=0)
```

```
In [12]: clean_dp2 = noNull_dp2.drop_duplicates()
```

```
In [13]: # Add columns that determine if the mail was On Time Exactly, Early, or Late
```

```
In [14]: clean_dp1['OnTimeExactly'] = np.where(clean_dp1['ACTUAL_DLVRY_DATE'] ==
clean_dp1['EXPECTED_DELIVERY_DATE'], True, False)
```

```
In [15]: clean_dp1['Early'] = np.where(clean_dp1['ACTUAL_DLVRY_DATE'] <
clean_dp1['EXPECTED_DELIVERY_DATE'], True, False)
```

```
In [16]: clean_dp1['Late'] = np.where(clean_dp1['ACTUAL_DLVRY_DATE'] >
clean_dp1['EXPECTED_DELIVERY_DATE'], True, False)
```

```
In [17]: clean_dp2['OnTimeExactly'] = np.where(clean_dp2['ACTUAL_DLVRY_DATE'] ==
clean_dp2['EXPECTED_DELIVERY_DATE'], True, False)
```

```
In [18]: clean_dp2['Early'] = np.where(clean_dp2['ACTUAL_DLVRY_DATE'] <
clean_dp2['EXPECTED_DELIVERY_DATE'], True, False)
```

```
In [19]: clean_dp2['Late'] = np.where(clean_dp2['ACTUAL_DLVRY_DATE'] >
clean_dp2['EXPECTED_DELIVERY_DATE'], True, False)
```

```
In [20]: # Combine the datasets with only data between Jan 8th and Jan 21st of 2024
```

```
In [21]: destinatingPieces_stormPeriod = pd.concat([clean_dp1, clean_dp2])
```

```

In [22]: # Compare the delivery status of the mail (Late vs. OnTime / Early)

In [23]: destinating_late = destinatingPieces_stormPeriod['Late'].values.sum()

In [24]: destinating_early = destinatingPieces_stormPeriod['Early'].values.sum()

In [25]: destinating_ontime = destinatingPieces_stormPeriod['OnTimeExactly'].values.sum()

In [26]: print("Late mail: "+ str(destinating_late) + ", Early Mail: "+
str(destinating_early)+", On Time Mail: "+ str(destinating_ontime))
Late mail: 6653098, Early Mail: 21326645, On Time Mail: 11776028

In [27]: # ratio of late vs rest

In [28]: destinating_late_ratio = destinating_late / (destinating_early +
destinating_ontime)

In [29]: print("Ratio of Late Mail vs. Early or On Time mail: " + str("%f" %
destinating_late_ratio))
Ratio of Late Mail vs. Early or On Time mail: 0.200984

In [30]: # distribution of mail shape by lateness

In [31]: destinatingPieces_stormPeriod['MAIL_SHAPE'].value_counts()
Out[31]:
MAIL_SHAPE
Letter      34267206
Flat        4569252
Card         919313
Name: count, dtype: int64

In [32]: dp_grouped = destinatingPieces_stormPeriod.groupby(by=["MAIL_SHAPE",
"Late"]).size() # True = Late, False = either Early or On Time Exactly

In [33]: print(dp_grouped)
MAIL_SHAPE  Late
Card        False      583117
             True       336196
Flat        False     3889537
             True       679715
Letter      False     28630019
             True       5637187
dtype: int64

In [34]: # Mail Class

In [35]: destinatingPieces_stormPeriod['MAIL_CLASS'].value_counts()
Out[35]:
MAIL_CLASS
USPS Marketing Mail      19182120
First Class Presort      13932727
Single Piece First Class  5604764
Periodicals              1036160
Name: count, dtype: int64

In [36]: dp_class = destinatingPieces_stormPeriod.groupby(by=["MAIL_CLASS", "Late"]).size()
# True = Late, False = either Early or On Time Exactly

```

```

In [37]: print(dp_class)
MAIL_CLASS      Late
First Class Presort  False    10171160
                  True      3761567
Periodicals      False     886349
                  True      149811
Single Piece First Class  False    4269783
                  True      1334981
USPS Marketing Mail    False    17775381
                  True      1406739
dtype: int64

In [38]: #EXPECTED_DESTINATION_FACILITY

In [39]: destinatingPieces_stormPeriod['EXPECTED_DESTINATION_FACILITY'].value_counts()
Out[39]:
EXPECTED_DESTINATION_FACILITY
NASHVILLE - 1441275      22270962
MEMPHIS - 1441274        14205357
MUSIC CITY ANNEX - 1532174  3279452
Name: count, dtype: int64

In [40]: dp_destFacility =
destinatingPieces_stormPeriod.groupby(by=["EXPECTED_DESTINATION_FACILITY", "Late"]).size() #
True = Late, False = either Early or On Time Exactly

In [41]: print(dp_destFacility)
EXPECTED_DESTINATION_FACILITY  Late
MEMPHIS - 1441274              False    10479767
                              True      3725590
MUSIC CITY ANNEX - 1532174      False    2852712
                              True       426740
NASHVILLE - 1441275            False    19770194
                              True      2500768
dtype: int64

In [42]: # Convert delivery date columns to the 'date' data type

In [43]: destinatingPieces_stormPeriod['ACTUAL_DLVRY_DATE'] =
pd.to_datetime(destinatingPieces_stormPeriod['ACTUAL_DLVRY_DATE'])

In [44]: print(destinatingPieces_stormPeriod['ACTUAL_DLVRY_DATE'].head())
2    2024-01-13
3    2024-01-11
5    2024-01-09
6    2024-01-10
10   2024-01-13
Name: ACTUAL_DLVRY_DATE, dtype: datetime64[ns]

In [45]: destinatingPieces_stormPeriod['EXPECTED_DELIVERY_DATE'] =
pd.to_datetime(destinatingPieces_stormPeriod['EXPECTED_DELIVERY_DATE'])

In [46]: # Difference between EXPECTED and ACTUAL delivery dates - Positive values indicate
LATE deliveries

In [47]: destinatingPieces_stormPeriod['Difference'] =
(destinatingPieces_stormPeriod['ACTUAL_DLVRY_DATE'] -
destinatingPieces_stormPeriod['EXPECTED_DELIVERY_DATE']).dt.days

```

```

In [48]: # Average days late a piece of mail arrives

In [49]: late_deliveries =
destinatingPieces_stormPeriod.loc[destinatingPieces_stormPeriod.Late]

In [50]: latemean = late_deliveries['Difference'].mean()

In [51]: print("The mean for mail delivered late is: " + str("%f" % latemean) + " days after
expected delivery date")
The mean for mail delivered late is: 2.486570 days after expected delivery date

In [52]: # Differences and their Count by Mail Class

In [53]: late_by_class = late_deliveries.groupby(by=["MAIL_CLASS", "Difference"]).size() #
True = Late, False = either Early or On Time Exactly

In [54]: pd.set_option('display.max_rows', 250)

In [55]: print(late_by_class)
MAIL_CLASS      Difference
First Class Presort  1          2053351
                   2          616154
                   3          544643
                   4          292230
                   5          100907
                   6           28657
                   7           24800
                   8           17205
                   9           10115
                  10           12429
                  11            7947
                  12           2675
                  13           2705
                  14           9097
                  15           2763
                  16           2084
                  17           3915
                  18           2231
                  19           1496
                  20           1661
                  21           3478
                  22           2001
                  23           1019
                  24           1876
                  25           1723
                  26           1747
                  27            873
                  28           1408
                  29           2018
                  30            915
                  31           1240
                  32           1836
                  33           1158
                  34            478
                  35            803
                  36           1163
                  37            160
                  38            298
                  39            158

```

	40	95
	41	34
	42	16
	43	4
	48	1
Periodicals	1	74896
	2	20512
	3	15732
	4	6076
	5	2624
	6	3227
	7	3551
	8	5600
	9	2156
	10	2736
	11	2260
	12	551
	13	1076
	14	2054
	15	926
	16	382
	17	895
	18	487
	19	149
	20	524
	21	571
	22	315
	23	251
	24	346
	25	188
	26	108
	27	313
	28	346
	29	327
	30	108
	31	208
	32	85
	33	40
	34	97
	35	42
	36	24
	37	6
	38	9
	39	5
	40	3
	42	1
	43	2
	44	1
	46	1
Single Piece First Class	1	610829
	2	211603
	3	193460
	4	112285
	5	43494
	6	23535
	7	28238
	8	25118
	9	10500
	10	12456

11	11382
12	5376
13	3170
14	6610
15	3163
16	2030
17	3146
18	2895
19	1769
20	1930
21	3132
22	2225
23	1473
24	1846
25	2144
26	1874
27	1302
28	1152
29	1570
30	1332
31	1254
32	850
33	540
34	276
35	194
36	227
37	97
38	79
39	78
40	64
41	24
42	21
43	21
44	8
45	10
46	11
47	12
48	8
49	6
50	10
51	1
52	5
53	8
54	8
55	5
56	7
57	5
58	7
59	7
60	4
61	8
62	4
63	1
64	8
65	1
66	1
67	1
69	1
70	2

	71	1
	72	1
	73	1
	76	1
	77	2
	81	1
	82	2
	83	2
	85	3
	88	5
	90	1
	91	2
	92	1
	93	1
	95	3
	96	2
	97	2
	98	2
	99	2
	100	1
	101	2
	104	2
	105	2
	106	1
	107	1
	108	3
	111	1
	112	1
	113	2
	117	1
	119	2
	120	1
	121	2
	125	1
	126	1
	127	1
	129	2
	136	2
	137	1
	141	2
	146	1
USPS Marketing Mail	1	859399
	2	229665
	3	95377
	4	58444
	5	28999
	6	32714
	7	31523
	8	19182
	9	10079
	10	8127
	11	5719
	12	2495
	13	5704
	14	3926
	15	1787
	16	1512
	17	1428
	18	950

19	498
20	720
21	1649
22	746
23	514
24	748
25	603
26	443
27	513
28	470
29	550
30	305
31	890
32	174
33	272
34	161
35	168
36	152
37	59
38	14
39	5
40	1
41	1
42	1
52	1
57	1
59	47
60	3

dtype: int64

In [56]: # Differences and their Count by Mail Shape

In [57]: late\_by\_shape = late\_deliveries.groupby(by=["MAIL\_SHAPE", "Difference"]).size() #  
True = Late, False = either Early or On Time Exactly

In [58]: print(late\_by\_shape)

MAIL_SHAPE	Difference	
Card	1	177511
	2	48643
	3	52607
	4	31674
	5	6968
	6	5547
	7	2074
	8	2179
	9	2053
	10	1088
	11	901
	12	269
	13	173
	14	1098
	15	240
	16	245
	17	244
	18	195
	19	354
	20	132
	21	385
	22	195



	23	74
	24	154
	25	126
	26	242
	27	97
	28	108
	29	114
	30	57
	31	106
	32	94
	33	42
	34	18
	35	46
	36	95
	37	10
	38	23
	39	9
	40	3
	41	1
	42	2
Flat	1	327335
	2	125657
	3	81343
	4	48012
	5	15583
	6	12127
	7	12862
	8	13120
	9	5636
	10	6414
	11	4410
	12	2282
	13	3297
	14	4392
	15	2174
	16	1222
	17	1680
	18	1395
	19	524
	20	890
	21	1529
	22	814
	23	568
	24	678
	25	542
	26	526
	27	652
	28	688
	29	811
	30	351
	31	474
	32	290
	33	245
	34	234
	35	257
	36	244
	37	98
	38	65
	39	56

40	27
41	16
42	13
43	11
44	2
45	4
46	4
47	7
48	7
49	4
50	6
51	1
52	4
53	5
54	6
55	3
56	6
57	5
58	6
59	6
60	4
61	7
62	2
63	1
64	7
65	1
66	1
67	1
69	1
70	2
71	1
72	1
73	1
76	1
77	2
81	1
82	2
83	2
85	3
88	5
90	1
91	2
92	1
93	1
95	3
96	2
97	2
98	2
99	2
100	1
101	2
104	2
105	2
106	1
107	1
108	3
111	1
112	1
113	2

	117	1
	119	2
	120	1
	121	2
	125	1
	126	1
	127	1
	129	2
	136	2
	137	1
	141	2
	146	1
Letter	1	3093629
	2	903634
	3	715262
	4	389349
	5	153473
	6	70459
	7	73176
	8	51806
	9	25161
	10	28246
	11	21997
	12	8546
	13	9185
	14	16197
	15	6225
	16	4541
	17	7460
	18	4973
	19	3034
	20	3813
	21	6916
	22	4278
	23	2615
	24	3984
	25	3990
	26	3404
	27	2252
	28	2580
	29	3540
	30	2252
	31	3012
	32	2561
	33	1723
	34	760
	35	904
	36	1227
	37	214
	38	312
	39	181
	40	133
	41	42
	42	24
	43	16
	44	7
	45	6
	46	8
	47	5

48	2
49	2
50	4
52	2
53	3
54	2
55	2
56	1
57	1
58	1
59	48
60	3
61	1
62	2
64	1

dtype: int64

In [59]: