Pandas

In [107…
```
import pandas as pd
```

In [108…
```
df_infy = pd.read_csv("INFY.csv")
```

In [109…
```
df_infy.head()
```

Out[109]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-09-17	22.950001	22.990000	22.719999	22.850000	22.401333	8070400
1	2021-09-20	22.750000	22.950001	22.570000	22.760000	22.313101	7148800
2	2021-09-21	23.129999	23.230000	23.100000	23.120001	22.666033	4785000
3	2021-09-22	23.120001	23.379999	23.070000	23.270000	22.813086	6613800
4	2021-09-23	23.680000	23.790001	23.600000	23.730000	23.264053	5323800

In [110…
```
df_infy.sort_values(by = "Volume",ascending = False)
```

Out[110]:

	Date	Open	High	Low	Close	Adj Close	Volume
251	2022-09-16	17.200001	17.459999	17.090000	17.379999	17.379999	42686600
126	2022-03-18	24.389999	25.040001	24.209999	25.040001	24.763243	29035600
154	2022-04-28	20.410000	20.660000	20.350000	20.500000	20.273422	24229100
20	2021-10-15	22.900000	23.400000	22.760000	23.379999	22.920927	22295600
250	2022-09-15	17.700001	17.809999	17.510000	17.790001	17.790001	21950100
...	...	...	...	...	...	...	...
232	2022-08-19	20.040001	20.040001	19.750000	19.780001	19.780001	2868500
132	2022-03-28	24.730000	24.770000	24.520000	24.719999	24.446779	2834400
68	2021-12-23	24.690001	24.799999	24.500000	24.730000	24.456669	2488600
71	2021-12-29	25.190001	25.379999	25.139999	25.379999	25.099483	2330400
72	2021-12-30	25.540001	25.600000	25.389999	25.410000	25.129152	2293900

252 rows × 7 columns

In [111…
```
df_dict = {'player_id': {0: 401, 1: 401, 2:  401, 3: 401, 4: 401, 5: 401, 6: 401, 7: 401, 8: 401, 9: 401, 10: 402, 1
```

In [112…
```
df = pd.DataFrame(df_dict)
df
```

Out[112]:

	player_id	match_date	match_result
0	401	2021-05-04 00:00:00	W
1	401	2021-05-09 00:00:00	L
2	401	2021-05-16 00:00:00	L
3	401	2021-05-18 00:00:00	W
4	401	2021-05-22 00:00:00	L
5	401	2021-06-15 00:00:00	L
6	401	2021-06-16 00:00:00	W
7	401	2021-06-18 00:00:00	W
8	401	2021-07-06 00:00:00	L
9	401	2021-07-13 00:00:00	L
10	402	2021-05-14 00:00:00	L
11	402	2021-05-23 00:00:00	L
12	402	2021-05-24 00:00:00	W
13	402	2021-06-01 00:00:00	W
14	402	2021-06-02 00:00:00	W
15	402	2021-07-01 00:00:00	W
16	402	2021-07-11 00:00:00	W
17	402	2021-07-20 00:00:00	L
18	402	2021-07-26 00:00:00	L
19	402	2021-07-30 00:00:00	L
20	403	2021-05-03 00:00:00	L
21	403	2021-05-11 00:00:00	W
22	403	2021-05-12 00:00:00	W
23	403	2021-05-13 00:00:00	W

	player id	match_date	match_result
24	403	2021-05-20 00:00:00	W
25	403	2021-05-25 00:00:00	W
26	403	2021-07-06 00:00:00	L
27	403	2021-07-15 00:00:00	L
28	403	2021-07-22 00:00:00	W
29	403	2021-07-23 00:00:00	W
30	404	2021-05-10 00:00:00	W
31	404	2021-05-16 00:00:00	W
32	404	2021-05-20 00:00:00	W
33	404	2021-05-22 00:00:00	W
34	404	2021-05-28 00:00:00	L
35	404	2021-06-06 00:00:00	L
36	404	2021-06-14 00:00:00	W
37	404	2021-07-25 00:00:00	W
38	404	2021-07-26 00:00:00	L
39	405	2021-05-07 00:00:00	L
40	405	2021-05-25 00:00:00	L
41	405	2021-06-06 00:00:00	L
42	405	2021-06-07 00:00:00	L
43	405	2021-06-14 00:00:00	L
44	405	2021-07-01 00:00:00	L
45	405	2021-07-02 00:00:00	L
46	405	2021-07-14 00:00:00	W
47	405	2021-07-16 00:00:00	L

	player_id	match_date	match_result
48	405	2021-07-30 00:00:00	L

In [113…
```
# with open("csv/sample.json","w") as file :
#     json.dump(df_dict,file) #javascript
```

In [114…
```
df[df["player_id"] == 401]
```

Out[114]:

	player_id	match_date	match_result
0	401	2021-05-04 00:00:00	W
1	401	2021-05-09 00:00:00	L
2	401	2021-05-16 00:00:00	L
3	401	2021-05-18 00:00:00	W
4	401	2021-05-22 00:00:00	L
5	401	2021-06-15 00:00:00	L
6	401	2021-06-16 00:00:00	W
7	401	2021-06-18 00:00:00	W
8	401	2021-07-06 00:00:00	L
9	401	2021-07-13 00:00:00	L

In [115…
```
df["player_id"] == 401
```

Out[115]:

0	True
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True
9	True
10	False
11	False
12	False
13	False
14	False
15	False
16	False
17	False
18	False
19	False
20	False
21	False
22	False
23	False
24	False
25	False
26	False
27	False
28	False
29	False
30	False
31	False
32	False
33	False
34	False
35	False
36	False
37	False
38	False
39	False
40	False
41	False
42	False
43	False

```
44    False
45    False
46    False
47    False
48    False
Name: player_id, dtype: bool
```

In [ ]:
```
# player id should be 403 and give me wins vs losses
```

In [179…
```
df[df["player_id"] == 403].groupby("match_result").count()
```

Out[179]:

match_result	player_id	match_date
L	3	3
W	7	7

In [180…
```
df.groupby(["player_id","match_result"]).count()
```

Out[180]:

player_id	match_result	match_date
401	L	6
	W	4
402	L	5
	W	5
403	L	3
	W	7
404	L	3
	W	6
405	L	9
	W	1

In [181…
```
df.groupby(["match_result","player_id"]).count()
```

Out[181]:

		match_date
match_result	**player_id**	
L	**401**	6
	402	5
	403	3
	404	3
	405	9
W	**401**	4
	402	5
	403	7
	404	6
	405	1

accessing elements in dataframe

In [183…
```
df.head(10)
```

Out[183]:

	player_id	match_date	match_result
0	401	2021-05-04 00:00:00	W
1	401	2021-05-09 00:00:00	L
2	401	2021-05-16 00:00:00	L
3	401	2021-05-18 00:00:00	W
4	401	2021-05-22 00:00:00	L
5	401	2021-06-15 00:00:00	L
6	401	2021-06-16 00:00:00	W
7	401	2021-06-18 00:00:00	W
8	401	2021-07-06 00:00:00	L
9	401	2021-07-13 00:00:00	L

In [184…

```
df[["player_id","match_result"]]
```

Out[184]:

	player_id	match_result
0	401	W
1	401	L
2	401	L
3	401	W
4	401	L
5	401	L
6	401	W
7	401	W
8	401	L
9	401	L
10	402	L
11	402	L
12	402	W
13	402	W
14	402	W
15	402	W
16	402	W
17	402	L
18	402	L
19	402	L
20	403	L
21	403	W
22	403	W
23	403	W

	player_id	match_result
24	403	W
25	403	W
26	403	L
27	403	L
28	403	W
29	403	W
30	404	W
31	404	W
32	404	W
33	404	W
34	404	L
35	404	L
36	404	W
37	404	W
38	404	L
39	405	L
40	405	L
41	405	L
42	405	L
43	405	L
44	405	L
45	405	L
46	405	W
47	405	L

	player_id	match_result
48	405	L

In [185…

```
df.iloc[:,0]
```

Out[185]:

0	401
1	401
2	401
3	401
4	401
5	401
6	401
7	401
8	401
9	401
10	402
11	402
12	402
13	402
14	402
15	402
16	402
17	402
18	402
19	402
20	403
21	403
22	403
23	403
24	403
25	403
26	403
27	403
28	403
29	403
30	404
31	404
32	404
33	404
34	404
35	404
36	404
37	404
38	404
39	405
40	405
41	405
42	405
43	405

```
44    405
45    405
46    405
47    405
48    405
Name: player_id, dtype: int64
```

In [187…

```
df.iloc[:,[0,2]]
```

Out[187]:

	player_id	match_result
0	401	W
1	401	L
2	401	L
3	401	W
4	401	L
5	401	L
6	401	W
7	401	W
8	401	L
9	401	L
10	402	L
11	402	L
12	402	W
13	402	W
14	402	W
15	402	W
16	402	W
17	402	L
18	402	L
19	402	L
20	403	L
21	403	W
22	403	W
23	403	W

	player_id	match_result
24	403	W
25	403	W
26	403	L
27	403	L
28	403	W
29	403	W
30	404	W
31	404	W
32	404	W
33	404	W
34	404	L
35	404	L
36	404	W
37	404	W
38	404	L
39	405	L
40	405	L
41	405	L
42	405	L
43	405	L
44	405	L
45	405	L
46	405	W
47	405	L

	player_id	match_result
48	405	L

In [188…

```
df.iloc[:,1:]
```

Out[188]:

	match_date	match_result
0	2021-05-04 00:00:00	W
1	2021-05-09 00:00:00	L
2	2021-05-16 00:00:00	L
3	2021-05-18 00:00:00	W
4	2021-05-22 00:00:00	L
5	2021-06-15 00:00:00	L
6	2021-06-16 00:00:00	W
7	2021-06-18 00:00:00	W
8	2021-07-06 00:00:00	L
9	2021-07-13 00:00:00	L
10	2021-05-14 00:00:00	L
11	2021-05-23 00:00:00	L
12	2021-05-24 00:00:00	W
13	2021-06-01 00:00:00	W
14	2021-06-02 00:00:00	W
15	2021-07-01 00:00:00	W
16	2021-07-11 00:00:00	W
17	2021-07-20 00:00:00	L
18	2021-07-26 00:00:00	L
19	2021-07-30 00:00:00	L
20	2021-05-03 00:00:00	L
21	2021-05-11 00:00:00	W
22	2021-05-12 00:00:00	W
23	2021-05-13 00:00:00	W

	match_date	match_result
24	2021-05-20 00:00:00	W
25	2021-05-25 00:00:00	W
26	2021-07-06 00:00:00	L
27	2021-07-15 00:00:00	L
28	2021-07-22 00:00:00	W
29	2021-07-23 00:00:00	W
30	2021-05-10 00:00:00	W
31	2021-05-16 00:00:00	W
32	2021-05-20 00:00:00	W
33	2021-05-22 00:00:00	W
34	2021-05-28 00:00:00	L
35	2021-06-06 00:00:00	L
36	2021-06-14 00:00:00	W
37	2021-07-25 00:00:00	W
38	2021-07-26 00:00:00	L
39	2021-05-07 00:00:00	L
40	2021-05-25 00:00:00	L
41	2021-06-06 00:00:00	L
42	2021-06-07 00:00:00	L
43	2021-06-14 00:00:00	L
44	2021-07-01 00:00:00	L
45	2021-07-02 00:00:00	L
46	2021-07-14 00:00:00	W
47	2021-07-16 00:00:00	L

	match_date	match_result
48	2021-07-30 00:00:00	L

In [190…

```
df.iloc[5:,1:]
```

Out[190]:

	match_date	match_result
5	2021-06-15 00:00:00	L
6	2021-06-16 00:00:00	W
7	2021-06-18 00:00:00	W
8	2021-07-06 00:00:00	L
9	2021-07-13 00:00:00	L
10	2021-05-14 00:00:00	L
11	2021-05-23 00:00:00	L
12	2021-05-24 00:00:00	W
13	2021-06-01 00:00:00	W
14	2021-06-02 00:00:00	W
15	2021-07-01 00:00:00	W
16	2021-07-11 00:00:00	W
17	2021-07-20 00:00:00	L
18	2021-07-26 00:00:00	L
19	2021-07-30 00:00:00	L
20	2021-05-03 00:00:00	L
21	2021-05-11 00:00:00	W
22	2021-05-12 00:00:00	W
23	2021-05-13 00:00:00	W
24	2021-05-20 00:00:00	W
25	2021-05-25 00:00:00	W
26	2021-07-06 00:00:00	L
27	2021-07-15 00:00:00	L
28	2021-07-22 00:00:00	W

	match_date	match_result
29	2021-07-23 00:00:00	W
30	2021-05-10 00:00:00	W
31	2021-05-16 00:00:00	W
32	2021-05-20 00:00:00	W
33	2021-05-22 00:00:00	W
34	2021-05-28 00:00:00	L
35	2021-06-06 00:00:00	L
36	2021-06-14 00:00:00	W
37	2021-07-25 00:00:00	W
38	2021-07-26 00:00:00	L
39	2021-05-07 00:00:00	L
40	2021-05-25 00:00:00	L
41	2021-06-06 00:00:00	L
42	2021-06-07 00:00:00	L
43	2021-06-14 00:00:00	L
44	2021-07-01 00:00:00	L
45	2021-07-02 00:00:00	L
46	2021-07-14 00:00:00	W
47	2021-07-16 00:00:00	L
48	2021-07-30 00:00:00	L

In [191…

```
df.iloc[40:47,1:]
```

Out[191]:

	match_date	match_result
40	2021-05-25 00:00:00	L
41	2021-06-06 00:00:00	L
42	2021-06-07 00:00:00	L
43	2021-06-14 00:00:00	L
44	2021-07-01 00:00:00	L
45	2021-07-02 00:00:00	L
46	2021-07-14 00:00:00	W

In [ ]:

In [ ]:

In [ ]:

In [116…

```
df_infy
```

Out[116]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-09-17	22.950001	22.990000	22.719999	22.850000	22.401333	8070400
1	2021-09-20	22.750000	22.950001	22.570000	22.760000	22.313101	7148800
2	2021-09-21	23.129999	23.230000	23.100000	23.120001	22.666033	4785000
3	2021-09-22	23.120001	23.379999	23.070000	23.270000	22.813086	6613800
4	2021-09-23	23.680000	23.790001	23.600000	23.730000	23.264053	5323800
...	...	...	...	...	...	...	...
247	2022-09-12	19.230000	19.410000	19.190001	19.240000	19.240000	3607400
248	2022-09-13	18.760000	18.900000	18.490000	18.559999	18.559999	15227800
249	2022-09-14	18.370001	18.440001	17.969999	18.080000	18.080000	16757300
250	2022-09-15	17.700001	17.809999	17.510000	17.790001	17.790001	21950100
251	2022-09-16	17.200001	17.459999	17.090000	17.379999	17.379999	42686600

252 rows × 7 columns

In [117…
```
df_infy["Volume"]
```

Out[117]:
```
0       8070400
1       7148800
2       4785000
3       6613800
4       5323800
         ...
247     3607400
248    15227800
249    16757300
250    21950100
251    42686600
Name: Volume, Length: 252, dtype: int64
```

In [118…
```
df_infy["Volume"].sum()
```

Out[118]:
```
2381188600
```

how many unique player id are there

In [119…
```
df["player_id"].unique()
```

Out[119]:
```
array([401, 402, 403, 404, 405], dtype=int64)
```

In [120…
```
df
```

Out[120]:

	player_id	match_date	match_result
0	401	2021-05-04 00:00:00	W
1	401	2021-05-09 00:00:00	L
2	401	2021-05-16 00:00:00	L
3	401	2021-05-18 00:00:00	W
4	401	2021-05-22 00:00:00	L
5	401	2021-06-15 00:00:00	L
6	401	2021-06-16 00:00:00	W
7	401	2021-06-18 00:00:00	W
8	401	2021-07-06 00:00:00	L
9	401	2021-07-13 00:00:00	L
10	402	2021-05-14 00:00:00	L
11	402	2021-05-23 00:00:00	L
12	402	2021-05-24 00:00:00	W
13	402	2021-06-01 00:00:00	W
14	402	2021-06-02 00:00:00	W
15	402	2021-07-01 00:00:00	W
16	402	2021-07-11 00:00:00	W
17	402	2021-07-20 00:00:00	L
18	402	2021-07-26 00:00:00	L
19	402	2021-07-30 00:00:00	L
20	403	2021-05-03 00:00:00	L
21	403	2021-05-11 00:00:00	W
22	403	2021-05-12 00:00:00	W
23	403	2021-05-13 00:00:00	W

	player id	match_date	match_result
24	403	2021-05-20 00:00:00	W
25	403	2021-05-25 00:00:00	W
26	403	2021-07-06 00:00:00	L
27	403	2021-07-15 00:00:00	L
28	403	2021-07-22 00:00:00	W
29	403	2021-07-23 00:00:00	W
30	404	2021-05-10 00:00:00	W
31	404	2021-05-16 00:00:00	W
32	404	2021-05-20 00:00:00	W
33	404	2021-05-22 00:00:00	W
34	404	2021-05-28 00:00:00	L
35	404	2021-06-06 00:00:00	L
36	404	2021-06-14 00:00:00	W
37	404	2021-07-25 00:00:00	W
38	404	2021-07-26 00:00:00	L
39	405	2021-05-07 00:00:00	L
40	405	2021-05-25 00:00:00	L
41	405	2021-06-06 00:00:00	L
42	405	2021-06-07 00:00:00	L
43	405	2021-06-14 00:00:00	L
44	405	2021-07-01 00:00:00	L
45	405	2021-07-02 00:00:00	L
46	405	2021-07-14 00:00:00	W
47	405	2021-07-16 00:00:00	L

	player_id	match_date	match_result
48	405	2021-07-30 00:00:00	L

In [121…

```
df = pd.DataFrame(df_dict)
df
```

Out[121]:

	player_id	match_date	match_result
0	401	2021-05-04 00:00:00	W
1	401	2021-05-09 00:00:00	L
2	401	2021-05-16 00:00:00	L
3	401	2021-05-18 00:00:00	W
4	401	2021-05-22 00:00:00	L
5	401	2021-06-15 00:00:00	L
6	401	2021-06-16 00:00:00	W
7	401	2021-06-18 00:00:00	W
8	401	2021-07-06 00:00:00	L
9	401	2021-07-13 00:00:00	L
10	402	2021-05-14 00:00:00	L
11	402	2021-05-23 00:00:00	L
12	402	2021-05-24 00:00:00	W
13	402	2021-06-01 00:00:00	W
14	402	2021-06-02 00:00:00	W
15	402	2021-07-01 00:00:00	W
16	402	2021-07-11 00:00:00	W
17	402	2021-07-20 00:00:00	L
18	402	2021-07-26 00:00:00	L
19	402	2021-07-30 00:00:00	L
20	403	2021-05-03 00:00:00	L
21	403	2021-05-11 00:00:00	W
22	403	2021-05-12 00:00:00	W
23	403	2021-05-13 00:00:00	W

	player id	match_date	match_result
24	403	2021-05-20 00:00:00	W
25	403	2021-05-25 00:00:00	W
26	403	2021-07-06 00:00:00	L
27	403	2021-07-15 00:00:00	L
28	403	2021-07-22 00:00:00	W
29	403	2021-07-23 00:00:00	W
30	404	2021-05-10 00:00:00	W
31	404	2021-05-16 00:00:00	W
32	404	2021-05-20 00:00:00	W
33	404	2021-05-22 00:00:00	W
34	404	2021-05-28 00:00:00	L
35	404	2021-06-06 00:00:00	L
36	404	2021-06-14 00:00:00	W
37	404	2021-07-25 00:00:00	W
38	404	2021-07-26 00:00:00	L
39	405	2021-05-07 00:00:00	L
40	405	2021-05-25 00:00:00	L
41	405	2021-06-06 00:00:00	L
42	405	2021-06-07 00:00:00	L
43	405	2021-06-14 00:00:00	L
44	405	2021-07-01 00:00:00	L
45	405	2021-07-02 00:00:00	L
46	405	2021-07-14 00:00:00	W
47	405	2021-07-16 00:00:00	L

	player_id	match_date	match_result
48	405	2021-07-30 00:00:00	L

In [122…
```
df["player_id"].unique() #
```

Out[122]:
```
array([401, 402, 403, 404, 405], dtype=int64)
```

In [123…
```
df["player_id"].nunique() # count of unique columns
```

Out[123]:
```
5
```

In [124…
```
spotify_dict = {'id': {0: 303651, 1: 85559, 2: 1046089, 3: 350824, 4: 776822, 5: 462814, 6: 33445, 7: 727284, 8: 1046
```

In [125…
```
df_spotify = pd.DataFrame(spotify_dict)
```

In [126…
```
df_spotify
```

Out[126]:

	id	position	trackname	artist	streams	url	date	region
0	303651	52	Heart Won't Forget	Matoma	28047	https://open.spotify.com/track/2of2DM5LqTh7ohm...	2017-02-04 00:00:00	no
1	85559	160	Someone In The Crowd - From "La La Land" Sound...	Emma Stone	17134	https://open.spotify.com/track/7xE4vKvjqUTtHyJ...	2017-02-26 00:00:00	fr
2	1046089	175	The Greatest	Sia	10060	https://open.spotify.com/track/7xHWNBFm6ObGEQP...	2017-03-06 00:00:00	cl
3	350824	25	Unforgettable	French Montana	46603	https://open.spotify.com/track/3B54sVLJ402zGa6...	2017-10-01 00:00:00	no
4	776822	1	Bad and Boujee (feat. Lil Uzi Vert)	Migos	1823391	https://open.spotify.com/track/4Km5HrUvYTaSUfi...	2017-01-27 00:00:00	us
...	...	...	...	...	...	...	...	...
95	792423	2	DNA.	Kendrick Lamar	3013496	https://open.spotify.com/track/6HZILIRieu8S0iq...	2017-04-15 00:00:00	us
96	792223	2	DNA.	Kendrick Lamar	3643231	https://open.spotify.com/track/6HZILIRieu8S0iq...	2017-04-14 00:00:00	us
97	793422	1	HUMBLE.	Kendrick Lamar	3144482	https://open.spotify.com/track/7KXjTSCq5nL1LoY...	2017-04-20 00:00:00	us
98	793622	1	HUMBLE.	Kendrick Lamar	3172718	https://open.spotify.com/track/7KXjTSCq5nL1LoY...	2017-04-21 00:00:00	us
99	793022	1	HUMBLE.	Kendrick Lamar	3394456	https://open.spotify.com/track/7KXjTSCq5nL1LoY...	2017-04-18 00:00:00	us

100 rows × 8 columns

In [127…

```
df_spotify["streams"]
```

Out[127]:
```
0       28047
1       17134
2       10060
3       46603
4     1823391
       ...
95    3013496
96    3643231
97    3144482
98    3172718
99    3394456
Name: streams, Length: 100, dtype: int64
```

In [128…
```
df_spotify["streams"].sum()
```

Out[128]:
```
39673624
```

In [129…
```
df_spotify["streams"].sum()/100
```

Out[129]:
```
396736.24
```

In [130…
```
host_dict = {'host_id': {0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9, 10: 10, 11: 11, 12: 0, 13: 0, 14
```

In [131…
```
df_host = pd.DataFrame(host_dict)
df_host
```

Out[131]:

	host_id	nationality	gender	age
0	0	USA	M	28
1	1	USA	F	29
2	2	China	F	31
3	3	China	M	24
4	4	Mali	M	30
...	...	...	...	...
171	7	Luxembourg	F	25
172	6	Luxembourg	M	25
173	7	Luxembourg	F	25
174	6	Luxembourg	M	25
175	7	Luxembourg	F	25

176 rows × 4 columns

In [132…
```
appart_dict = {'host_id': {0: 0, 1: 0, 2: 0, 3: 1, 4: 1, 5: 2, 6: 3, 7: 3, 8: 4, 9: 5, 10: 5, 11: 6, 12: 7, 13: 8, 14
```

In [133…
```
df_apprt = pd.DataFrame(appart_dict)
df_apprt
```

Out[133]:

	host_id	apartment_id	apartment_type	n_beds	n_bedrooms	country	city
0	0	A1	Room	1	1	USA	New York
1	0	A2	Room	1	1	USA	New Jersey
2	0	A3	Room	1	1	USA	New Jersey
3	1	A4	Apartment	2	1	USA	Houston
4	1	A5	Apartment	2	1	USA	Las Vegas
5	2	A6	Yurt	3	1	Mongolia	-
6	3	A7	Penthouse	3	3	China	Tianjin
7	3	A8	Penthouse	5	5	China	Beijing
8	4	A9	Apartment	2	1	Mali	Bamako
9	5	A10	Room	3	1	Mali	Segou
10	5	A11	Room	2	1	Mali	Segou
11	6	A12	Penthouse	6	6	Luxembourg	Luxembourg
12	7	A13	Room	4	1	Luxembourg	Luxembourg
13	8	A14	Apartment	2	1	Australia	Perth
14	9	A15	Apartment	2	1	Australia	Perth
15	9	A16	Apartment	2	1	Australia	Perth
16	10	A17	Room	4	1	Brazil	Rio De Janeiro
17	10	A18	Room	4	1	Argentina	Mendoza
18	10	A19	Room	4	2	Uruguay	Mercedes
19	10	A20	Room	4	2	Brazil	Brasilia
20	11	A21	Apartment	2	2	Mexico	Mexico City

In [134…
```
df_host_apprt = pd.merge(df_host,df_apprt, on = "host_id")
```

In [135…
```
df_host_apprt
```

Out[135]:

	host_id	nationality	gender	age	apartment_id	apartment_type	n_beds	n_bedrooms	country	city
0	0	USA	M	28	A1	Room	1	1	USA	New York
1	0	USA	M	28	A2	Room	1	1	USA	New Jersey
2	0	USA	M	28	A3	Room	1	1	USA	New Jersey
3	0	USA	M	28	A1	Room	1	1	USA	New York
4	0	USA	M	28	A2	Room	1	1	USA	New Jersey
...	...	...	...	...	...	...	...	...	...	...
347	10	Brazil	M	39	A20	Room	4	2	Brazil	Brasilia
348	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City
349	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City
350	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City
351	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City

352 rows × 10 columns

In [136…

```
df_host_apprt[df_host_apprt["host_id"]==0]
```

Out[136]:

	host_id	nationality	gender	age	apartment_id	apartment_type	n_beds	n_bedrooms	country	city
0	0	USA	M	28	A1	Room	1	1	USA	New York
1	0	USA	M	28	A2	Room	1	1	USA	New Jersey
2	0	USA	M	28	A3	Room	1	1	USA	New Jersey
3	0	USA	M	28	A1	Room	1	1	USA	New York
4	0	USA	M	28	A2	Room	1	1	USA	New Jersey
...	...	...	...	...	...	...	...	...	...	...
211	0	USA	M	28	A2	Room	1	1	USA	New Jersey
212	0	USA	M	28	A3	Room	1	1	USA	New Jersey
213	0	USA	M	28	A1	Room	1	1	USA	New York
214	0	USA	M	28	A2	Room	1	1	USA	New Jersey
215	0	USA	M	28	A3	Room	1	1	USA	New Jersey

216 rows × 10 columns

In [137…

```
df_host_apprt[df_host_apprt["nationality"] != df_host_apprt["country"]]
```

Out[137]:

	host_id	nationality	gender	age	apartment_id	apartment_type	n_beds	n_bedrooms	country	city
232	2	China	F	31	A6	Yurt	3	1	Mongolia	-
233	2	China	F	31	A6	Yurt	3	1	Mongolia	-
234	2	China	F	31	A6	Yurt	3	1	Mongolia	-
235	2	China	F	31	A6	Yurt	3	1	Mongolia	-
333	10	Brazil	M	39	A18	Room	4	1	Argentina	Mendoza
334	10	Brazil	M	39	A19	Room	4	2	Uruguay	Mercedes
337	10	Brazil	M	39	A18	Room	4	1	Argentina	Mendoza
338	10	Brazil	M	39	A19	Room	4	2	Uruguay	Mercedes
341	10	Brazil	M	39	A18	Room	4	1	Argentina	Mendoza
342	10	Brazil	M	39	A19	Room	4	2	Uruguay	Mercedes
345	10	Brazil	M	39	A18	Room	4	1	Argentina	Mendoza
346	10	Brazil	M	39	A19	Room	4	2	Uruguay	Mercedes
348	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City
349	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City
350	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City
351	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City

In [138…

```
df_host_apprt[df_host_apprt["nationality"] == df_host_apprt["country"]]
```

Out[138]:

	host_id	nationality	gender	age	apartment_id	apartment_type	n_beds	n_bedrooms	country	city
0	0	USA	M	28	A1	Room	1	1	USA	New York
1	0	USA	M	28	A2	Room	1	1	USA	New Jersey
2	0	USA	M	28	A3	Room	1	1	USA	New Jersey
3	0	USA	M	28	A1	Room	1	1	USA	New York
4	0	USA	M	28	A2	Room	1	1	USA	New Jersey
...	...	...	...	...	...	...	...	...	...	...
339	10	Brazil	M	39	A20	Room	4	2	Brazil	Brasilia
340	10	Brazil	M	39	A17	Room	4	1	Brazil	Rio De Janeiro
343	10	Brazil	M	39	A20	Room	4	2	Brazil	Brasilia
344	10	Brazil	M	39	A17	Room	4	1	Brazil	Rio De Janeiro
347	10	Brazil	M	39	A20	Room	4	2	Brazil	Brasilia

336 rows × 10 columns

In [139…
```
df_host_apprt[df_host_apprt["nationality"] != df_host_apprt["country"]]["host_id"].nunique()
```

Out[139]: 3

Joins are very important : https://www.w3schools.com/sql/sql_join.asp

rename

In [140…
```
df_host_apprt.rename(columns = {"host_id" : "renamed_host_id"},inplace = True )
```

In [141…
```
df_host_apprt
```

Out[141]:

	renamed_host_id	nationality	gender	age	apartment_id	apartment_type	n_beds	n_bedrooms	country	city
0	0	USA	M	28	A1	Room	1	1	USA	New York
1	0	USA	M	28	A2	Room	1	1	USA	New Jersey
2	0	USA	M	28	A3	Room	1	1	USA	New Jersey
3	0	USA	M	28	A1	Room	1	1	USA	New York
4	0	USA	M	28	A2	Room	1	1	USA	New Jersey
...	...	...	...	...	...	...	...	...	...	...
347	10	Brazil	M	39	A20	Room	4	2	Brazil	Brasilia
348	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City
349	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City
350	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City
351	11	Brazil	F	42	A21	Apartment	2	2	Mexico	Mexico City

352 rows × 10 columns

In [142…
```
df_host_apprt.columns
```

Out[142]:
```
Index(['renamed_host_id', 'nationality', 'gender', 'age', 'apartment_id',
       'apartment_type', 'n_beds', 'n_bedrooms', 'country', 'city'],
      dtype='object')
```

Date wide data is called timeseries data we can use this to predict something

19th sept 2022

In [143…
```
df_infy
```

Out[143]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-09-17	22.950001	22.990000	22.719999	22.850000	22.401333	8070400
1	2021-09-20	22.750000	22.950001	22.570000	22.760000	22.313101	7148800
2	2021-09-21	23.129999	23.230000	23.100000	23.120001	22.666033	4785000
3	2021-09-22	23.120001	23.379999	23.070000	23.270000	22.813086	6613800
4	2021-09-23	23.680000	23.790001	23.600000	23.730000	23.264053	5323800
...	...	...	...	...	...	...	...
247	2022-09-12	19.230000	19.410000	19.190001	19.240000	19.240000	3607400
248	2022-09-13	18.760000	18.900000	18.490000	18.559999	18.559999	15227800
249	2022-09-14	18.370001	18.440001	17.969999	18.080000	18.080000	16757300
250	2022-09-15	17.700001	17.809999	17.510000	17.790001	17.790001	21950100
251	2022-09-16	17.200001	17.459999	17.090000	17.379999	17.379999	42686600

252 rows × 7 columns

In [144…
```
mean_open = df_infy["Open"].mean()
```

In [145…
```
df_infy["far_apart_open"] = df_infy["Open"]**2 - mean_open**2
```

In [146…
```
df_infy
```

Out[146]:

	Date	Open	High	Low	Close	Adj Close	Volume	far_apart_open
0	2021-09-17	22.950001	22.990000	22.719999	22.850000	22.401333	8070400	59.995105
1	2021-09-20	22.750000	22.950001	22.570000	22.760000	22.313101	7148800	50.855059
2	2021-09-21	23.129999	23.230000	23.100000	23.120001	22.666033	4785000	68.289413
3	2021-09-22	23.120001	23.379999	23.070000	23.270000	22.813086	6613800	67.827006
4	2021-09-23	23.680000	23.790001	23.600000	23.730000	23.264053	5323800	94.034959
...	...	...	...	...	...	...	...	...
247	2022-09-12	19.230000	19.410000	19.190001	19.240000	19.240000	3607400	-96.914541
248	2022-09-13	18.760000	18.900000	18.490000	18.559999	18.559999	15227800	-114.769841
249	2022-09-14	18.370001	18.440001	17.969999	18.080000	18.080000	16757300	-129.250504
250	2022-09-15	17.700001	17.809999	17.510000	17.790001	17.790001	21950100	-153.417405
251	2022-09-16	17.200001	17.459999	17.090000	17.379999	17.379999	42686600	-170.867406

252 rows × 8 columns

In [147…
```
summation_of_far_apart_open = df_infy["far_apart_open"].sum()
```

In [148…
```
summation_of_far_apart_open
```

Out[148]:
```
1367.4751837390095
```

In [149…
```
var = summation_of_far_apart_open/25
var
```

Out[149]:
```
54.69900734956038
```

In [150…
```
type(df_infy.shape)
```

Out[150]:
```
tuple
```

In [151…
```
df_infy.head()
```

Out[151]:

	Date	Open	High	Low	Close	Adj Close	Volume	far_apart_open
0	2021-09-17	22.950001	22.990000	22.719999	22.850000	22.401333	8070400	59.995105
1	2021-09-20	22.750000	22.950001	22.570000	22.760000	22.313101	7148800	50.855059
2	2021-09-21	23.129999	23.230000	23.100000	23.120001	22.666033	4785000	68.289413
3	2021-09-22	23.120001	23.379999	23.070000	23.270000	22.813086	6613800	67.827006
4	2021-09-23	23.680000	23.790001	23.600000	23.730000	23.264053	5323800	94.034959

we want make a dict with date and far apart open

In [152…
```
def convert_to_dict(row):
    return{row["Date"]:row["far_apart_open"]}
```

In [153…
```
df_infy["date_spread_dict"] = df_infy.apply(convert_to_dict, axis = 1)
```

In [154…
```
df_infy.drop(["Date","far_apart_open"],axis = 1 , inplace = True)
```

In [155…
```
df_infy
```

Out[155]:

	Open	High	Low	Close	Adj Close	Volume	date_spread_dict
0	22.950001	22.990000	22.719999	22.850000	22.401333	8070400	{'2021-09-17': 59.995105167695044}
1	22.750000	22.950001	22.570000	22.760000	22.313101	7148800	{'2021-09-20': 50.855059267694}
2	23.129999	23.230000	23.100000	23.120001	22.666033	4785000	{'2021-09-21': 68.28941300769503}
3	23.120001	23.379999	23.070000	23.270000	22.813086	6613800	{'2021-09-22': 67.82700550769488}
4	23.680000	23.790001	23.600000	23.730000	23.264053	5323800	{'2021-09-23': 94.03495926769398}
...	...	...	...	...	...	...	...
247	19.230000	19.410000	19.190001	19.240000	19.240000	3607400	{'2022-09-12': -96.91454073230597}
248	18.760000	18.900000	18.490000	18.559999	18.559999	15227800	{'2022-09-13': -114.76984073230597}
249	18.370001	18.440001	17.969999	18.080000	18.080000	16757300	{'2022-09-14': -129.25050399230503}
250	17.700001	17.809999	17.510000	17.790001	17.790001	21950100	{'2022-09-15': -153.417405332305}
251	17.200001	17.459999	17.090000	17.379999	17.379999	42686600	{'2022-09-16': -170.867406332305}

252 rows × 7 columns

how to separate the above column date_spread_dict --> apply

In [156…

```
def extract_col(row):
    key = list(row["date_spread_dict"].keys())[0]
    value = list(row["date_spread_dict"].values())[0]
    row["date"] = key
    row["far_apart_open"] = value
    return row
```

In [157…

```
df_infy.apply(extract_col,axis = 1)
```

Out[157]:

	Open	High	Low	Close	Adj Close	Volume	date_spread_dict	date	far_apart_open
0	22.950001	22.990000	22.719999	22.850000	22.401333	8070400	{'2021-09-17': 59.995105167695044}	2021-09-17	59.995105
1	22.750000	22.950001	22.570000	22.760000	22.313101	7148800	{'2021-09-20': 50.855059267694}	2021-09-20	50.855059
2	23.129999	23.230000	23.100000	23.120001	22.666033	4785000	{'2021-09-21': 68.28941300769503}	2021-09-21	68.289413
3	23.120001	23.379999	23.070000	23.270000	22.813086	6613800	{'2021-09-22': 67.82700550769488}	2021-09-22	67.827006
4	23.680000	23.790001	23.600000	23.730000	23.264053	5323800	{'2021-09-23': 94.03495926769398}	2021-09-23	94.034959
...	...	...	...	...	...	...	...	...	...
247	19.230000	19.410000	19.190001	19.240000	19.240000	3607400	{'2022-09-12': -96.91454073230597}	2022-09-12	-96.914541
248	18.760000	18.900000	18.490000	18.559999	18.559999	15227800	{'2022-09-13': -114.76984073230597}	2022-09-13	-114.769841
249	18.370001	18.440001	17.969999	18.080000	18.080000	16757300	{'2022-09-14': -129.25050399230503}	2022-09-14	-129.250504
250	17.700001	17.809999	17.510000	17.790001	17.790001	21950100	{'2022-09-15': -153.417405332305}	2022-09-15	-153.417405
251	17.200001	17.459999	17.090000	17.379999	17.379999	42686600	{'2022-09-16': -170.867406332305}	2022-09-16	-170.867406

252 rows × 9 columns

TQDM--> check

In [158…
```
!pip install tqdm
```

```
Requirement already satisfied: tqdm in c:\users\mause\appdata\local\programs\python\python310\lib\site-packages (4.64.1)
Requirement already satisfied: colorama in c:\users\mause\appdata\local\programs\python\python310\lib\site-packages (from tqdm) (0.4.5)
WARNING: You are using pip version 22.0.4; however, version 22.2.2 is available.
You should consider upgrading via the 'C:\Users\mause\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.
```

In [159…
```
from tqdm import tqdm
```

In [160…
```
tqdm.pandas()
```

In [161…
```
df_infy = df_infy.progress_apply(extract_col,axis = 1)
```

```
100%|██████████████████████████████████████████████████████████████████████| 252/252 [00:00<00:00, 792.52it
/s]
```

In [162…
```
df_infy.tail(10)
```

Out[162]:

	Open	High	Low	Close	Adj Close	Volume	date_spread_dict	date	far_apart_open
242	18.299999	18.440001	18.000000	18.090000	18.090000	5248800	{'2022-09-02': -131.81747733230503}	2022-09-02	-131.817477
243	18.139999	18.219999	17.930000	17.980000	17.980000	7158700	{'2022-09-06': -137.647877012305}	2022-09-06	-137.647877
244	18.139999	18.450001	18.110001	18.430000	18.430000	4769300	{'2022-09-07': -137.647877012305}	2022-09-07	-137.647877
245	18.350000	18.530001	18.280001	18.530001	18.530001	3936900	{'2022-09-08': -129.98494073230597}	2022-09-08	-129.984941
246	18.770000	19.100000	18.750000	19.059999	19.059999	5513600	{'2022-09-09': -114.39454073230604}	2022-09-09	-114.394541
247	19.230000	19.410000	19.190001	19.240000	19.240000	3607400	{'2022-09-12': -96.91454073230597}	2022-09-12	-96.914541
248	18.760000	18.900000	18.490000	18.559999	18.559999	15227800	{'2022-09-13': -114.76984073230597}	2022-09-13	-114.769841
249	18.370001	18.440001	17.969999	18.080000	18.080000	16757300	{'2022-09-14': -129.25050399230503}	2022-09-14	-129.250504
250	17.700001	17.809999	17.510000	17.790001	17.790001	21950100	{'2022-09-15': -153.417405332305}	2022-09-15	-153.417405
251	17.200001	17.459999	17.090000	17.379999	17.379999	42686600	{'2022-09-16': -170.867406332305}	2022-09-16	-170.867406

In [163…
```
df_infy_aj_close = df_infy[["Adj Close","date"]]
```

In [164…
```
df_infy_aj_close.dtypes
```

Out[164]:
```
Adj Close    float64
date          object
dtype: object
```

In [165…
```
date = "2021-02-21"
```

In [166…
```
date.split("-")
```

Out[166]:
```
['2021', '02', '21']
```

In [167…
```
date.split("-")[1]
```

Out[167]:
```
'02'
```

In [168…
```
df_infy_aj_close.date.str.split("-")
```

Out[168]:
```
0      [2021, 09, 17]
1      [2021, 09, 20]
2      [2021, 09, 21]
3      [2021, 09, 22]
4      [2021, 09, 23]
            ...
247    [2022, 09, 12]
248    [2022, 09, 13]
249    [2022, 09, 14]
250    [2022, 09, 15]
251    [2022, 09, 16]
Name: date, Length: 252, dtype: object
```

what is stringmethod?

In [169…
```
df_infy_aj_close["month"] = df_infy_aj_close.date.str.split("-").str[1]
```

```
C:\Users\mause\AppData\Local\Temp\ipykernel_9784\660655713.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_infy_aj_close["month"] = df_infy_aj_close.date.str.split("-").str[1]
```

In [170…
```
df_infy_aj_close
```

Out[170]:

	Adj Close	date	month
0	22.401333	2021-09-17	09
1	22.313101	2021-09-20	09
2	22.666033	2021-09-21	09
3	22.813086	2021-09-22	09
4	23.264053	2021-09-23	09
...	...	...	...
247	19.240000	2022-09-12	09
248	18.559999	2022-09-13	09
249	18.080000	2022-09-14	09
250	17.790001	2022-09-15	09
251	17.379999	2022-09-16	09

252 rows × 3 columns

In [171…

```
df_infy_aj_close.groupby("month")["Adj Close"].mean() # this is a series
```

Out[171]:

```
month
01    23.894947
02    22.542275
03    23.791064
04    21.559053
05    19.106501
06    18.640476
07    18.665000
08    19.644348
09    20.294808
10    22.369523
11    22.804656
12    23.589092
Name: Adj Close, dtype: float64
```

In [172…

```
df_infy_aj_close.groupby("month")["Adj Close"].mean().reset_index() # this is a dataframe
```

Out[172]:

	month	Adj Close
0	01	23.894947
1	02	22.542275
2	03	23.791064
3	04	21.559053
4	05	19.106501
5	06	18.640476
6	07	18.665000
7	08	19.644348
8	09	20.294808
9	10	22.369523
10	11	22.804656
11	12	23.589092

to get sort by month and year

In [173…
```
df_infy_aj_close["year"] = df_infy_aj_close.date.str.split("-").str[0]
```

```
C:\Users\mause\AppData\Local\Temp\ipykernel_9784\2795584799.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  df_infy_aj_close["year"] = df_infy_aj_close.date.str.split("-").str[0]
```

In [174…
```
df_infy_aj_close
```

Out[174]:

	Adj Close	date	month	year
0	22.401333	2021-09-17	09	2021
1	22.313101	2021-09-20	09	2021
2	22.666033	2021-09-21	09	2021
3	22.813086	2021-09-22	09	2021
4	23.264053	2021-09-23	09	2021
...	...	...	...	...
247	19.240000	2022-09-12	09	2022
248	18.559999	2022-09-13	09	2022
249	18.080000	2022-09-14	09	2022
250	17.790001	2022-09-15	09	2022
251	17.379999	2022-09-16	09	2022

252 rows × 4 columns

In [175…

```
def extract_month_year(row):
    row["year"] = row["date"].split("-")[0]
    row["month"] = row["date"].split("-")[1]
    return row
```

In [ ]:

In [ ]:

In [ ]: