```
1
2 """https://www1.nseindia.com/products/content/equities/indices/historical_index_data.h
```

```
'https://www1.nseindia.com/products/content/equities/indices/historical_index_data.h
tm'
```

```
1 import numpy as np
2 import pandas as pd
```

```
1 n50=pd.read_csv("nifty50_2021.csv")
2 nn50=pd.read_csv("niftynext50_2021.csv")
```

```
1 n50
```

	Date	Open	High	Low	Close	Shares Traded	Turnover (Rs. Cr)
0	01-Jan-2021	13996.10	14049.85	13991.35	14018.50	258090905	15873.75
1	04-Jan-2021	14104.35	14147.95	13953.75	14132.90	494999295	28705.09
2	05-Jan-2021	14075.15	14215.60	14048.15	14199.50	492475349	30872.87
3	06-Jan-2021	14240.95	14244.15	14039.90	14146.25	632323316	34615.55
4	07-Jan-2021	14253.75	14256.25	14123.10	14137.35	559173512	33446.47
...	...	...	...	...	...	...	...
243	27-Dec-2021	16937.75	17112.05	16833.20	17086.25	144777457	12567.03
244	28-Dec-2021	17177.60	17250.25	17161.15	17233.25	176026100	14553.76

```
1 nn50
```

	Date	Open	High	Low	Close	Shares Traded	Turnover (Rs. Cr)
0	01-Jan-2021	32608.95	32807.65	32554.35	32765.95	354161209	5571.06
1	04-Jan-2021	32998.95	33347.85	32805.30	33281.65	395945593	8039.85

```
n50=n50.loc[:,'Date':'Close']
n50
```

	Date	Open	High	Low	Close
0	01-Jan-2021	13996.10	14049.85	13991.35	14018.50
1	04-Jan-2021	14104.35	14147.95	13953.75	14132.90
2	05-Jan-2021	14075.15	14215.60	14048.15	14199.50
3	06-Jan-2021	14240.95	14244.15	14039.90	14146.25
4	07-Jan-2021	14253.75	14256.25	14123.10	14137.35
...	...	...	...	...	...
243	27-Dec-2021	16937.75	17112.05	16833.20	17086.25
244	28-Dec-2021	17177.60	17250.25	17161.15	17233.25
245	29-Dec-2021	17220.10	17285.95	17176.65	17213.60
246	30-Dec-2021	17201.45	17264.05	17146.35	17203.95
247	31-Dec-2021	17244.50	17400.80	17238.50	17354.05

248 rows × 5 columns

```
nn50=nn50.loc[:,'Date':'Close']
nn50
```

	Date	Open	High	Low	Close
0	01-Jan-2021	32608.95	32807.65	32554.35	32765.95
1	04-Jan-2021	32998.95	33347.85	32805.30	33281.65

```
1 nse=pd.concat([n50,nn50.loc[:,'Open':'Close']],axis='columns')
2 nse
```

	Date	Open	High	Low	Close	Open	High	Low	Clos
0	01-Jan-2021	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.35	32765.9
1	04-Jan-2021	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.30	33281.6
2	05-Jan-2021	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	32986.15	33818.7
3	06-Jan-2021	14240.95	14244.15	14039.90	14146.25	33951.20	34068.80	33376.25	33755.5
4	07-Jan-2021	14253.75	14256.25	14123.10	14137.35	34074.50	34138.75	33826.30	33889.3
...	...	...	...	...	...	...	...	...	

```
1 nse.columns=['date','50open','50high','50low','50close','n50open','n50high','n50low','n
```

```
1 nse
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50clo
0	01-Jan-2021	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.35	32765.

```
1 nse.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   date      248 non-null    object 
 1   50open    248 non-null    float64
 2   50high    248 non-null    float64
 3   50low     248 non-null    float64
 4   50close   248 non-null    float64
 5   n50open   248 non-null    float64
 6   n50high   248 non-null    float64
 7   n50low    248 non-null    float64
 8   n50close  248 non-null    float64
dtypes: float64(8), object(1)
memory usage: 17.6+ KB
```

```
1 nse.describe()
```

	50open	50high	50low	50close	n50open	n50h
count	248.000000	248.000000	248.000000	248.000000	248.000000	248.000
mean	16042.683871	16118.128226	15929.441331	16026.760685	38661.587903	38851.836
std	1289.929909	1284.668542	1291.407114	1285.160097	3624.433021	3618.420
min	13758.600000	13898.250000	13596.750000	13634.600000	31992.750000	32711.100
25%	14873.062500	14951.912500	14740.725000	14872.187500	34822.987500	35093.512
50%	15793.700000	15836.400000	15716.400000	15765.300000	39005.050000	39119.475
75%	17290.612500	17378.600000	17197.200000	17323.925000	42347.112500	42588.050
max	18602.350000	18604.450000	18445.300000	18477.050000	45216.550000	45290.800

```
1 """
2 1 In 2019 how many days was nifty 50 volatile ( high > 105% of low)
3 2 In 2019 how many days was nifty Next 50 volatile ( high > 105% of low)
4 3 In 2019 how many days belonged to 4 classes (nifty 50 volatile , nifty 50  non volati
5 4 Compute the mean, median and std var of closing value for each weekday in nifty 50 fo
6 5 Compute the mean, median and std var of closing value for each month in nifty 50 for 
7 6 On days where nifty50 closed higher than the open, what was the mean of ( close – ope
8 7 In 2019, how many days had the days high lower then the pervious days low in nifty50 
9 8 In 2019, how many days did days close exceed the 30day moving average in niftyNext50(
10 """
```

```
'\n1 In 2019 how many days was nifty 50 volatile ( high > 105% of low)\n2 In 2019 ho
w many days was nifty Next 50 volatile ( high > 105% of low)\n3 In 2019 how many day
s belonged to 4 classes (nifty 50 volatile , nifty 50  non volatile , Next 50 volati
le & Next 50 non volatile )\n4 Compute the mean, median and std var of closing value
```

1 In 2021 how many days was nifty 50 volatile (high > 105% of low)

```
nse[nse['50high'] > (105/100)*nse['50low']]
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close

in no day , nifty 50 high was 5% more than its low

```
#but there is one day high  4% more than low
```

```
nse[nse['50high']>(104/100)*nse['50low']]
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close
	01-								

2 In 2021 how many days was nifty Next 50 volatile (high > 105% of low)

```
nse[nse['n50high']>(105/100)*nse['n50low']]
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close

in no day nifty next 50 high was 5% more than its low

```
# but 3% is there
nse[nse['n50high']>(103/100)*nse['n50low']]
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50clo
16	25-Jan-2021	14477.80	14491.10	14218.60	14238.90	33618.05	33680.30	32636.75	33080.
19	29-Jan-2021	13946.60	13966.85	13596.75	13634.60	32784.40	32805.55	31638.15	31743.
20	01-Feb-2021	13758.60	14336.35	13661.75	14281.20	31992.75	32747.95	31647.65	32661.
	19								

3 In 2021 how many days belonged to 4 classes (nifty 50 volatile , nifty 50 non volatile , Next 50 volatile & Next 50 non volatile)

```
def comp(a,b) :
    if a > (105/100)* b :
        return ('Volatile')
    else :
        return('Non Volatile')
```

nifty 50 volatility

```
nse.apply(lambda x: comp(x['50high'],x['50low']),axis=1).describe()
```

```
count              248
unique               1
top       Non Volatile
freq               248
dtype: object
```

nifty next 50 volatility

```
nse.apply(lambda x: comp(x['n50high'],x['n50low']),axis=1).describe()
```

```
count              248
unique               1
top       Non Volatile
freq               248
dtype: object
```

4 Compute the mean, median and std var of closing value for each weekday in nifty 50 for 2021

```
nse['date2'] = pd.to_datetime(nse['date'])
nse.head()
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close
0	01-Jan-2021	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.35	32765.95
1	04-Jan-2021	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.30	33281.65
	05-								

```
nse['weekday'] = nse['date2'].dt.day_name()
nse.head()
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close
0	01-Jan-2021	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.35	32765.95
1	04-Jan-2021	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.30	33281.65
	05-								

```
1 nse.groupby('weekday')['50close'].agg(['mean','median','std'])
```

weekday	mean	median	std
Friday	15858.511458	15706.00	1261.716366
Monday	16002.517647	15811.85	1311.478749
Thursday	16091.745918	15778.45	1320.728316
Tuesday	16067.203922	15772.75	1282.041696
Wednesday	16109.729592	15767.55	1284.598589

```
1 nse
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50clo
0	01-Jan-2021	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.35	32765.
1	04-Jan-2021	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.30	33281.
2	05-Jan-2021	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	32986.15	33818.
3	06-Jan-2021	14240.95	14244.15	14039.90	14146.25	33951.20	34068.80	33376.25	33755.
4	07-Jan-2021	14253.75	14256.25	14123.10	14137.35	34074.50	34138.75	33826.30	33889.
...	...	...	...	...	...	...	...	...	

This is not actual , check for every week avg from Mon-Fri

```
nse['date3'] = pd.to_datetime(nse['date'], errors ='coerce')
nse.head()
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close
0	01-Jan-2021	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.35	32765.95
1	04-Jan-2021	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.30	33281.65
	05-								

```
nse['weekNumber'] = nse['date3'].dt.week
nse.head()
```

```
st-packages/ipykernel_launcher.py:1: FutureWarning: Series.dt.weekofyear and Series.dt
ing an IPython kernel.
```

	50low	50close	n50open	n50high	n50low	n50close	date2	weekday	DD	Mon
	13991.35	14018.50	32608.95	32807.65	32554.35	32765.95	2021-01-01	Friday	01	Jan
	13953.75	14132.90	32998.95	33347.85	32805.30	33281.65	2021-01-04	Monday	04	Jan
	14048.15	14199.50	33219.80	33895.90	32986.15	33818.75	2021-	Tuesday	05	Jan

```
nse.groupby('weekNumber')['50close'].agg(['mean','median','std'])
```

weekNumber	mean	median	std
1	14192.6500	14146.250	90.456060
2	14528.4700	14563.450	66.981962
3	14481.8800	14521.150	151.777932
4	13914.6375	13892.525	255.465325
5	14707.7800	14789.950	261.948663
6	15133.6400	15115.800	32.015278
7	15187.5500	15208.900	141.008293
8	14798.4000	14707.800	233.901301
9	14989.0200	14938.100	182.669061
10	15065.0875	15064.675	93.395953
11	14772.6200	14744.000	152.633092
12	14586.5500	14549.400	194.069363
13	14801.0500	14845.100	96.211265
14	14769.8000	14819.050	102.889206
15	14503.7250	14543.125	136.975548
16	14350.8375	14350.400	45.413294
17	14705.7200	14653.050	171.810676
18	14659.2900	14634.150	122.465849
19	14791.8500	14773.625	126.778659
20	15028.5500	15030.150	116.169849
21	15296.2200	15301.450	98.248280
22	15618.8900	15582.800	56.588044
23	15732.8400	15740.100	59.913504
24	15764.6800	15767.550	79.297948
25	15771.4000	15772.750	63.302133
26	15737.3700	15722.200	49.689028
27	15789.9900	15818.250	78.578873
28	15841.3000	15853.950	95.815754
29	15766.1500	15788.225	99.319157
30	15764.3600	15763.050	42.285997

\5 Compute the mean, median and std var of closing value for each month in nifty 50 for 2021

```
nse.head(1)
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close
	01-								

```
# write function to extract, year,  month 3 letters , date and using split function ,
# put it in specific columns # #0"
# then group by month , get mean median
#make it an array , get max of date of month value and find stamdard deviation
def sepdate(a):
    return(a.split('-'))[0]
nse['DD']=nse['date'].apply(sepdate)
nse.head(1)
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close
	01-								

```
def sepMon(a):
    return a.split('-')[1]
nse['Mon']=nse['date'].apply(sepMon)
nse.head()
```

```
1 nse.groupby('Mon')['50close'].mean()
```

```
Mon
Apr    14613.852632
Aug    16470.459524
Dec    17174.269565
Feb    14956.842500
Jan    14284.602500
Jul    15783.097619
Jun    15733.677273
Mar    14835.100000
May    14983.525000
Nov    17718.892500
Oct    18020.220000
Sep    17508.614286
Name: 50close, dtype: float64
```

```
1 nse.groupby('Mon')['50close'].median()
```

```
Mon
Apr    14631.100
Aug    16496.450
Dec    17203.950
Feb    15039.675
Jan    14314.275
Jul    15778.450
Jun    15743.300
Mar    14845.100
May    14932.750
Nov    17881.275
Oct    18053.425
Sep    17519.450
Name: 50close, dtype: float64
```

```
1 nse.groupby('Mon')['50close'].var()
```

```
Mon
Apr     43120.639020
Aug     77135.590405
Dec     52393.485850
Feb     75528.329283
Jan     78481.125914
Jul      6629.181369
Jun      6205.754697
Mar     54143.446250
May     91134.691974
Nov    137105.475862
Oct     75847.472211
Sep     45212.250286
Name: 50close, dtype: float64
```

```
1 nse.groupby('Mon')['50close'].std() # make month to MM Format for easy sorting in mont
```

```
Mon
Apr    207.655096
```

```
Aug    277.732948
Dec    228.896234
Feb    274.824179
Jan    280.144830
Jul     81.419785
Jun     78.776613
Mar    232.687443
May    301.885230
Nov    370.277566
Oct    275.404198
Sep    212.631725
Name: 50close, dtype: float64
```

6 On days where nifty50 closed higher than the open, what was the mean of (close – open) for niftyNext 50.

```
nse.head(0)
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close	date2	we

```
fclmop = nse[nse['50close'] > nse['50open'] ]
fclmop
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50c
0	01-Jan-2021	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.35	327
1	04-Jan-2021	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.30	332
2	05-Jan-2021	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	32986.15	338
5	08-Jan-2021	14258.40	14367.30	14221.65	14347.25	34124.10	34364.25	34075.20	343
6	11-Jan-2021	14474.05	14498.20	14383.10	14484.75	34566.95	34570.65	34143.60	344
...	...	...	...	...	...	...	...	...	

```
(fclmop['n50close']-fclmop['n50open']).mean()
```

```
145.24596774193512
```

7 In 2021, how many days had the days high lower then the pervious days low in nifty50

```
1 nse[1:][np.array(nse[1:]['50close'])>np.array(nse[:-1]['50close'])].describe()
```

	50open	50high	50low	50close	n50open	n5
count	137.000000	137.000000	137.000000	137.000000	137.000000	137.0
mean	15988.902555	16091.368613	15922.941241	16048.394161	38564.562774	38789.6
std	1242.564950	1232.185718	1243.718713	1231.785240	3522.803061	3507.4
min	13758.600000	14147.950000	13661.750000	14132.900000	31992.750000	32747.9
25%	14816.850000	14959.100000	14760.800000	14895.650000	34825.000000	35114.5
50%	15794.000000	15835.550000	15749.800000	15812.350000	39001.550000	39120.1
75%	17104.400000	17250.250000	17064.250000	17233.250000	41923.400000	42258.3
max	18500.100000	18543.150000	18445.300000	18477.050000	45216.550000	45290.8

8 In 2021, how many days did days close exceed the 30day moving average in niftyNext50(Excluding first month"""

```
1 nse[nse['Mon']=='Jan'].count()
```

```
date        20
50open      20
50high      20
50low       20
50close     20
n50open     20
n50high     20
n50low      20
n50close    20
date2       20
weekday     20
DD          20
Mon         20
dtype: int64
```

```
1 nse['30mov']=round(nse['50close'].rolling(30).mean())
2 nse
```