```
"""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'
```

```
import numpy as np
import pandas as pd
```

```
  n50=pd.read_csv("nifty50_2021.csv")
  nn50=pd.read_csv("niftynext50_2021.csv")
  #can use ,sep='\t' for tsv  or sep = '|'
```

```
n50.head()
```

	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

```
nn50.head()
```

	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
2	05-Jan-2021	33219.80	33895.90	32986.15	33818.75	341021632	9986.69

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

	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

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

	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
2	05-Jan-2021	33219.80	33895.90	32986.15	33818.75
3	06-Jan-2021	33951.20	34068.80	33376.25	33755.50
4	07-Jan-2021	34074.50	34138.75	33826.30	33889.35

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

	Date	Open	High	Low	Close	Open	High	Low	Close
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.shape
```

```
(248, 9)
```

```
nse.shape[0]
```

```
248
```

```
nse.shape[1]
```

```
9
```

```
nse.index
```

```
RangeIndex(start=0, stop=248, step=1)
```

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

```
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

```
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
```

```
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

```
nse.describe(include='all')
```

	date	50open	50high	50low	50close	n50open	
count	248	248.000000	248.000000	248.000000	248.000000	248.000000	
unique	248	NaN	NaN	NaN	NaN	NaN	
top	01-Jan-2021	NaN	NaN	NaN	NaN	NaN	
freq	1	NaN	NaN	NaN	NaN	NaN	
mean	NaN	16042.683871	16118.128226	15929.441331	16026.760685	38661.587903	38
std	NaN	1289.929909	1284.668542	1291.407114	1285.160097	3624.433021	3

```
#describe can be used for columns also
nse.day.describe()
```

```
count         219
unique          5
top       Tuesday
freq           46
Name: day, dtype: object
```

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

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 nse: comp(nse['50high'],nse['50low']),axis=1)
```

```
0      Non Volatile
1      Non Volatile
2      Non Volatile
3      Non Volatile
4      Non Volatile
           ...
243    Non Volatile
```

```
244    Non Volatile
245    Non Volatile
246    Non Volatile
247    Non Volatile
Length: 248, dtype: object
```

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

```
0      Non Volatile
1      Non Volatile
2      Non Volatile
3      Non Volatile
4      Non Volatile
           ...
243    Non Volatile
244    Non Volatile
245    Non Volatile
246    Non Volatile
247    Non Volatile
Length: 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.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['date'] = pd.to_datetime(nse['date'])
nse.head()
```

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

```
nse['day'] = nse['date'].dt.day_name()
nse.head()
```

	date	50open	50high	50low	50close	n50open	n50high	n50low	n50close
0	2021-01-01	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.35	32765.95
1	2021-01-04	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.30	33281.65
2	2021-01-05	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	32986.15	33818.75

```
#to rearrange columns
nse.columns
```

```
Index(['date', '50open', '50high', '50low', '50close', 'n50open', 'n50high',
       'n50low', 'n50close', 'day'],
      dtype='object')
```

```
nse = nse[['date','day','50open', '50high', '50low', '50close', 'n50open', 'n50high','
nse.head()
```

	date	day	50open	50high	50low	50close	n50open	n50high	n50l
0	2021-01-01	Friday	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.
1	2021-01-04	Monday	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.
2	2021-01-05	Tuesday	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	32986.

```
nse.groupby('day')['50close'].mean() #mean median count can also used
```

```
day
Friday       15858.511458
Monday       16002.517647
Thursday     16091.745918
Tuesday      16067.203922
Wednesday    16109.729592
Name: 50close, dtype: float64
```

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

day	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

to get for every week

```
nse['week#'] = nse['date'].dt.week
nse.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Series
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarnir
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/us
  """Entry point for launching an IPython kernel.
```

	date	day	50open	50high	50low	50close	n50open	n50high	n50l
0	2021-01-01	Friday	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.
1	2021-01-04	Monday	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.
2	2021-01-05	Tuesday	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	32986.
3	2021-	Wednesday	14240.95	14244.15	14039.00	14146.35	33951.20	34068.80	33276.

```
nse.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   date      248 non-null    datetime64[ns]
 1   day       248 non-null    object
 2   50open    248 non-null    float64
 3   50high    248 non-null    float64
 4   50low     248 non-null    float64
 5   50close   248 non-null    float64
 6   n50open   248 non-null    float64
```

```
 7   n50high   248 non-null    float64
 8   n50low    248 non-null    float64
 9   n50close  248 non-null    float64
 10  week#     248 non-null    int64
dtypes: datetime64[ns](1), float64(8), int64(1), object(1)
memory usage: 21.4+ KB
```

```
# 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):
    a=str(a)
    return(a.split('-'))[2][:2]
nse['DD']=nse['date'].apply(sepdate)
nse.head()
```

	date	day	50open	50high	50low	50close	n50open	n50high	n50l
0	2021-01-01	Friday	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.
1	2021-01-04	Monday	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.
2	2021-01-05	Tuesday	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	32986.

```
def sepmonth(a):
    a=str(a)
    return(a[5:7])
nse['MM']=nse['date'].apply(sepmonth)
nse.head()
```

	date	day	50open	50high	50low	50close	n50open	n50high	n50l
0	2021-01-01	Friday	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.
1	2021-01-04	Monday	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.
2	2021-01-05	Tuesday	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	32986.

```
nse.groupby('MM')['50close'].var()   #.std()
```

```
MM
01     78481.125914
02     75528.329283
03     54143.446250
04     43120.639020
05     91134.691974
06      6205.754697
```

```
07      6629.181369
08     77135.590405
09     45212.250286
10     75847.472211
11    137105.475862
12     52393.485850
Name: 50close, dtype: float64
```

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

	date	day	50open	50high	50low	50close	n50open	n50high	n50l
0	2021-01-01	Friday	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	32554.
1	2021-01-04	Monday	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	32805.
2	2021-01-05	Tuesday	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	32986.
5	2021-01-08	Friday	14258.40	14367.30	14221.65	14347.25	34124.10	34364.25	34075.
6	2021-01-11	Monday	14474.05	14498.20	14383.10	14484.75	34566.95	34570.65	34143.
...	...	...	...	...	...	...	...	...	
241	2021-12-23	Thursday	17066.80	17118.65	17015.55	17072.60	41787.30	41989.05	41762.
243	2021-12-27	Monday	16937.75	17112.05	16833.20	17086.25	41459.65	41666.25	41134.

```
nse[ nse['50close'] > nse['50open'] ].median()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: DataFr
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Droppi
  """Entry point for launching an IPython kernel.
50open      15736.950
50high      15819.075
50low       15690.400
50close     15784.450
n50open     38751.250
n50high     38965.000
n50low      38634.350
n50close    38824.750
week#          25.000
DD             13.000
MM              6.000
dtype: float64
```

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

	date	day	50open	50high	50low	50close	n50open	n50high	n5
1	2021-01-04	Monday	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	328(
2	2021-01-05	Tuesday	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	329
5	2021-01-08	Friday	14258.40	14367.30	14221.65	14347.25	34124.10	34364.25	3407
6	2021-01-11	Monday	14474.05	14498.20	14383.10	14484.75	34566.95	34570.65	3414
7	2021-01-12	Tuesday	14473.80	14590.65	14432.85	14563.45	34462.85	34563.45	3429
...	...	...	...	...	...	...	...	...	
240	2021-12-22	Wednesday	16865.55	16971.00	16819.50	16955.45	41187.50	41568.15	4112
241	2021-12-23	Thursday	17066.80	17118.65	17015.55	17072.60	41787.30	41989.05	4176

```
nse[nse['MM']=='01'].count()
```

```
date        20
day         20
50open      20
50high      20
50low       20
50close     20
n50open     20
n50high     20
n50low      20
n50close    20
week#       20
DD          20
MM          20
dtype: int64
```

moving average

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

	date	day	50open	50high	50low	50close	n50open	n50high	n5
0	2021-01-01	Friday	13996.10	14049.85	13991.35	14018.50	32608.95	32807.65	3255
1	2021-01-04	Monday	14104.35	14147.95	13953.75	14132.90	32998.95	33347.85	3280
2	2021-01-05	Tuesday	14075.15	14215.60	14048.15	14199.50	33219.80	33895.90	3298
3	2021-01-06	Wednesday	14240.95	14244.15	14039.90	14146.25	33951.20	34068.80	3337

```
nse.sort_index(ascending=0)
```

	date	day	50open	50high	50low	50close	n50open	n50high	n5
247	2021-12-31	Friday	17244.50	17400.80	17238.50	17354.05	41641.05	42258.30	4163
246	2021-12-30	Thursday	17201.45	17264.05	17146.35	17203.95	41849.80	41867.55	4150
245	2021-12-29	Wednesday	17220.10	17285.95	17176.65	17213.60	42002.05	42117.10	4183
244	2021-12-28	Tuesday	17177.60	17250.25	17161.15	17233.25	41833.20	42022.25	4179
243	2021-12-27	Monday	16937.75	17112.05	16833.20	17086.25	41459.65	41666.25	4113
...	...	...	...	...	...	...	...	...	
4	2021-01-07	Thursday	14253.75	14256.25	14123.10	14137.35	34074.50	34138.75	3382
3	2021-01-06	Wednesday	14240.95	14244.15	14039.90	14146.25	33951.20	34068.80	3337

```
nse.sort_values(by='day',ascending=0)
# if only one column , apart from index , just use .sort_values(ascending=0)
```

	date	day	50open	50high	50low	50close	n50open	n50high	n5
183	2021-09-29	Wednesday	17657.95	17781.75	17608.15	17711.30	42426.45	42664.80	4232
37	2021-02-24	Wednesday	14729.15	15008.80	14723.05	14982.00	34825.00	34850.50	3439

```
nse=nse.dropna()
```

```
nse['50close'].sum()
```

```
3554900.8000000003
```

```
nse['50close'].median()
```

```
15860.35
```

```
nse['day'].dtype
```

```
dtype('O')
```

```
nse['date'].dtype
```

```
dtype('<M8[ns]')
```

```
conv = lambda x : x/1000
conv
```

```
<function __main__.<lambda>>
```

```
nse['trial']=nse['50close'].apply(conv)
nse.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarnir
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/us
  """Entry point for launching an IPython kernel.
```

	date	day	50open	50high	50low	50close	n50open	n50high	n50
29	2021-02-12	Friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448
30	2021-02-15	Monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	3479(
31	2021-02-16	Tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	3484

```
nse['50close']/nse['50open']
```

```
29     0.998492
30     1.002908
31     0.996227
32     0.995353
33     0.992142
         ...
243    1.008767
244    1.003240
245    0.999623
246    1.000145
247    1.006353
Length: 219, dtype: float64
```

```
nse['day'].unique()
```

```
array(['Friday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday'],
      dtype=object)
```

```
nse['day'].nunique()
```

```
5
```

```
nse['day'].unique().max()
```

```
'Wednesday'
```

```
nse['day'].value_counts()
```

```
Tuesday      46
Monday       45
Wednesday    43
Thursday     43
Friday       42
Name: day, dtype: int64
```

```
nse['day'].value_counts().count()
```

```
5
```

```
  #how dataset indexed
  nse.index
```

```
Int64Index([ 29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
            ...
            238, 239, 240, 241, 242, 243, 244, 245, 246, 247],
           dtype='int64', length=219)
```

```
#Change index
nse.head()
```

	date	day	50open	50high	50low	50close	n50open	n50high	n50
29	2021-02-12	Friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448
30	2021-02-15	Monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	34790
31	2021-02-16	Tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	34849

```
nse.index=nse['date']
nse.head()
```

	date	day	50open	50high	50low	50close	n50open	n50high	n
date									
2021-02-12	2021-02-12	Friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34
2021-02-15	2021-02-15	Monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	34
2021-02-16	2021-02-16	Tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	34

```
del nse['date']
```

```
nse.head()
```

	day	50open	50high	50low	50close	n50open	n50high	n50low
date								
2021-02-12	Friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448.35
2021-02-15	Monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	34790.45
2021-02-16	Tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	34849.15

```
nse.head(25)
```

date	day	50open	50high	50low	50close	n50open	n50high	n50low
2021-02-12	Friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448.35
2021-02-15	Monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	34790.45
2021-02-16	Tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	34849.15
2021-02-17	Wednesday	15279.90	15314.30	15170.75	15208.90	35179.95	35326.10	35039.10
2021-02-18	Thursday	15238.70	15250.75	15078.05	15118.95	35364.95	35553.80	35248.30
2021-02-19	Friday	15074.80	15144.05	14898.20	14981.75	35406.95	35478.70	34558.75
2021-02-22	Monday	14999.05	15010.10	14635.05	14675.70	34938.25	35024.55	34192.15
2021-02-23	Tuesday	14782.25	14854.50	14651.85	14707.80	34520.95	34716.90	34311.70
2021-02-24	Wednesday	14729.15	15008.80	14723.05	14982.00	34825.00	34850.50	34396.60
2021-02-25	Thursday	15079.85	15176.50	15065.35	15097.35	34994.75	35180.15	34933.30
2021-02-26	Friday	14888.60	14919.45	14467.75	14529.15	34624.45	35009.25	34041.75
2021-03-01	Monday	14702.50	14806.80	14638.55	14761.55	34491.40	34655.05	34326.60
2021-03-02	Tuesday	14865.30	14959.10	14760.80	14919.10	34787.55	35233.00	34708.25

```
nse.tail()
```

date	day	50open	50high	50low	50close	n50open	n50high	n50low
2021-12-27	Monday	16937.75	17112.05	16833.20	17086.25	41459.65	41666.25	41134.65
2021-12-28	Tuesday	17177.60	17250.25	17161.15	17233.25	41833.20	42022.25	41794.85
2021-12-29	Wednesday	17220.10	17285.95	17176.65	17213.60	42002.05	42117.10	41836.00

```
#other way to access column
```

```
nse.trial
```

```
date
2021-02-12    15.16330
2021-02-15    15.31470
2021-02-16    15.31345
2021-02-17    15.20890
2021-02-18    15.11895
                 ...
2021-12-27    17.08625
2021-12-28    17.23325
2021-12-29    17.21360
2021-12-30    17.20395
2021-12-31    17.35405
Name: trial, Length: 219, dtype: float64
```

```
nse['50close'].mean()
```

```
16232.423744292246
```

```
nse.groupby('day')['50close'].max()
```

```
day
Friday       18114.90
Monday       18477.05
Thursday     18338.55
Tuesday      18418.75
Wednesday    18266.60
Name: 50close, dtype: float64
```

```
nse.groupby('day')['50close'].max().sort_index()
```

```
day
Friday       18114.90
Monday       18477.05
Thursday     18338.55
Tuesday      18418.75
Wednesday    18266.60
Name: 50close, dtype: float64
```

```
nse.groupby('day')['50close'].max().sort_values(ascending=False)
```

```
day
Monday       18477.05
Tuesday      18418.75
Thursday     18338.55
Wednesday    18266.60
Friday       18114.90
Name: 50close, dtype: float64
```

```
pd.get_dummies(nse.day)
```

date	Friday	Monday	Thursday	Tuesday	Wednesday
2021-02-12	1	0	0	0	0
2021-02-15	0	1	0	0	0
2021-02-16	0	0	0	1	0
2021-02-17	0	0	0	0	1
2021-02-18	0	0	1	0	0
...	...	...	...	...	...
2021-12-27	0	1	0	0	0
2021-12-28	0	0	0	1	0
2021-12-29	0	0	0	0	1
2021-12-30	0	0	1	0	0

```
nsec=nse.copy() #shallow copy , change wont affect
nsec.head()
```

date	day	50open	50high	50low	50close	n50open	n50high	n50low
2021-02-12	Friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448.35
2021-02-15	Monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	34790.45
2021-02-16	Tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	34849.15

```
nsec=pd.concat([nsec,pd.get_dummies(nsec.day)],axis='columns')
nsec.head()
```

	day	50open	50high	50low	50close	n50open	n50high	n50low
date								
2021-02-12	Friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448.35

```
nse.columns
```

```
Index(['day', '50open', '50high', '50low', '50close', 'n50open', 'n50high',
       'n50low', 'n50close', 'week#', 'DD', 'MM', '30mov', 'trial'],
      dtype='object')
```

```
#rename columns
nse.columns=['day', 'new50open', '50high', '50low', '50close', 'n50open', 'n50high',
       'n50low', 'n50close', 'week#', 'DD', 'MM', '30mov', 'trial']
```

```
nse.head()
```

	day	new50open	50high	50low	50close	n50open	n50high	n50low
date								
2021-02-12	Friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448.35
2021-02-15	Monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	34790.45
2021-02-16	Tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	34849.15

```
nse.loc[:,'day':'50low']
```

	day	new50open	50high	50low
date				

```
nse.iloc[:,1:4]
```

	new50open	50high	50low
date			
2021-02-12	15186.20	15243.50	15081.00
2021-02-15	15270.30	15340.15	15243.40
2021-02-16	15371.45	15431.75	15242.20
2021-02-17	15279.90	15314.30	15170.75
2021-02-18	15238.70	15250.75	15078.05
...	...	...	...
2021-12-27	16937.75	17112.05	16833.20
2021-12-28	17177.60	17250.25	17161.15
2021-12-29	17220.10	17285.95	17176.65
2021-12-30	17201.45	17264.05	17146.35
2021-12-31	17244.50	17400.80	17238.50

219 rows × 3 columns

```
nse.head()
```

	day	new50open	50high	50low	50close	n50open	n50high	n50low
date								
2021-02-12	Friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448.35
2021-02-15	Monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	34790.45
2021-02-16	Tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	34849.15

```
lowercase = lambda x : x.lower()
lowercase
```

```
<function __main__.<lambda>>
```

```
nse['day']=nse['day'].apply(lowercase)
nse.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarnir
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/us
  """Entry point for launching an IPython kernel.
```

	day	new50open	50high	50low	50close	n50open	n50high	n50low
date								
2021-02-12	friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448.35
2021-02-15	monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	34790.45
2021-02-16	tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	34849.15

```
def week_short(a) :
    if a=='monday':
        return 'Mon'
    elif a=='tuesday':
        return 'Tue'
    elif a=='wednesday':
        return 'Wed'
    elif a=='thursday':
        return 'Thu'
    elif a=='friday':
        return 'Fri'
```

```
nse['week_shor']=nse['day'].apply(week_short)
nse.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarnir
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/us
  """Entry point for launching an IPython kernel.
```

	day	new50open	50high	50low	50close	n50open	n50high	n50low
date								
2021-02-12	friday	15186.20	15243.50	15081.00	15163.30	34653.70	34748.60	34448.35
2021-02-15	monday	15270.30	15340.15	15243.40	15314.70	34816.95	35114.55	34790.45
2021-02-16	tuesday	15371.45	15431.75	15242.20	15313.45	35109.10	35243.05	34849.15

```
#multiply every number in dataset by 10
def mul10(a) :
```

```
  if type(a)==float or type(a)==int  :
      return 10*a
  else :
      return a
```

```
nsec.applymap(mul10)
```

date	day	50open	50high	50low	50close	n50open	n50high	n50low
2021-02-12	Friday	151862.0	152435.0	150810.0	151633.0	346537.0	347486.0	344483.5
2021-02-15	Monday	152703.0	153401.5	152434.0	153147.0	348169.5	351145.5	347904.5
2021-02-16	Tuesday	153714.5	154317.5	152422.0	153134.5	351091.0	352430.5	348491.5
2021-02-17	Wednesday	152799.0	153143.0	151707.5	152089.0	351799.5	353261.0	350391.0
2021-02-18	Thursday	152387.0	152507.5	150780.5	151189.5	353649.5	355538.0	352483.0
...	...	...	...	...	...	...	...	...
2021-12-27	Monday	169377.5	171120.5	168332.0	170862.5	414596.5	416662.5	411346.5
2021-12-28	Tuesday	171776.0	172502.5	171611.5	172332.5	418332.0	420222.5	417948.5
2021-12-29	Wednesday	172201.0	172859.5	171766.5	172136.0	420020.5	421171.0	418360.0
2021-12-30	Thursday	172014.5	172640.5	171463.5	172039.5	418498.0	418675.5	415075.0
2021-12-31	Friday	172445.0	174008.0	172385.0	173540.5	416410.5	422583.0	416388.0

219 rows × 19 columns

```
nse[nse==nse.max()]
```

date	day	new50open	50high	50low	50close	n50open	n50high	n50low	n50clo
2021-02-12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2021-02-15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2021-02-16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2021-02-17	wednesday	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2021-02-18	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

```
round(23.056768)
```

```
23
```

12-27									
2021-12-28	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N