Import python libraries

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
```

Read the excel-file (NIFTY 200 market data - "MW-NIFTY-200-11-Mar-2021.csv"

In [3]:

```
df=pd.read_csv('MW-NIFTY-200-11-Mar-2021.csv')
```

Read top 10 records

In [3]:

```
df.head(10)
```

Out[3]:

	SYMBOL \n	OPEN \n	HIGH \n	LOW \n	PREV. CLOSE \n	LTP \n	CHNG \n	%CHNG \n	VOLUME \n(shares)	VA
0	NIFTY 200	7920.60	7930.90	7869.65	7865.90	7909.55	43.65	0.55	1238073055	4.190310e
1	LTI	3955.00	4150.00	3936.05	3896.50	4142.00	245.50	6.30	1096653	4.488425e
2	EDELWEISS	80.00	85.40	77.00	78.75	83.65	4.90	6.22	11933408	9.817615e
3	HUDCO	48.45	51.90	47.80	48.10	51.00	2.90	6.03	7522890	3.809591e
4	COFORGE	2620.00	2757.00	2620.00	2582.80	2733.00	150.20	5.82	1193338	3.248612e
5	MINDTREE	1829.00	1902.00	1811.55	1796.35	1893.00	96.65	5.38	3329555	6.207889e
6	SRTRANSFIN	1291.15	1329.80	1289.95	1270.25	1320.60	50.35	3.96	3041247	3.998388e
7	JSWENERGY	84.75	89.00	84.50	83.70	86.80	3.10	3.70	11468549	9.985666e
8	BIOCON	392.00	404.45	388.70	389.00	402.80	13.80	3.55	7942179	3.161861e
9	MPHASIS	1610.95	1675.00	1598.00	1592.90	1647.55	54.65	3.43	599154	9.858061e

Read last 10 records

In [88]:

```
df.tail(10)
```

Out[88]:

	SYMBOL \n	OPEN \n	HIGH \n	LOW \n	PREV. CLOSE \n	LTP \n	CHNG \n	%CHNG \n	VOLUME \n(shares)	V
191	IOC	101.9	101.90	98.70	100.55	99.00	-1.55	-1.54	21160710	2.10697
192	ONGC	116.9	117.00	113.60	116.75	114.60	-2.15	-1.84	23841269	2.72863
193	PETRONET	252.0	253.30	246.00	251.00	246.35	-4.65	-1.85	2541063	6.31301
194	CONCOR	593.8	593.80	567.60	583.85	572.60	-11.25	-1.93	2596632	1.49433

	SYMBOL \n	OPEN \n	HIGH \n	LOW \n	PREV. CLOSE \n	LTP \n	CHNG \n	%CHNG \n	VOLUME \n(shares)	V
195	NATCOPHARM	850.1	855.00	820.60	845.05	825.30	-19.75	-2.34	336704	2.79878
196	GSPL	280.3	281.55	269.80	277.60	271.10	-6.50	-2.34	754885	2.07163
197	SBICARD	1061.8	1065.40	1022.10	1055.25	1028.00	-27.25	-2.58	1409421	1.46106
198	DHANI	305.6	305.60	288.05	303.20	293.80	-9.40	-3.10	2448102	7.21553
199	ADANITRANS	740.0	765.00	730.55	757.65	733.00	-24.65	-3.25	662901	4.89148
200	SBILIFE	973.7	974.00	936.00	971.30	937.50	-33.80	-3.48	2362998	2.24728

Columns present in the dataset

In [91]:
```
df.columns
```

Out[91]:
```
Index(['SYMBOL \n', 'OPEN \n', 'HIGH \n', 'LOW \n', 'PREV. CLOSE \n', 'LTP \n',
       'CHNG \n', '%CHNG \n', 'VOLUME \n(shares)', 'VALUE ', '52W H \n',
       '52W L \n', '365 D % CHNG \n 25-Feb-2020',
       '30 D % CHNG \n 25-Jan-2021'],
      dtype='object')
```

Checking for null values

In [32]:
```
df.isna().sum()
```

Out[32]:
```
SYMBOL \n                      0
OPEN \n                        0
HIGH \n                        0
LOW \n                         0
PREV. CLOSE \n                 0
LTP \n                         0
CHNG \n                        0
%CHNG \n                       0
VOLUME \n(shares)              0
VALUE                          0
52W H \n                       0
52W L \n                       0
365 D % CHNG \n 25-Feb-2020    0
30 D % CHNG \n 25-Jan-2021     0
dtype: int64
```

Statistical Analysis

In [47]:
```
# Equities in NIFTY 200
```

In [4]:
```
market_data=df.iloc[1:,]
```

Change the column names

In [5]:
```
market_data.columns=['SYMBOL', 'OPEN', 'HIGH', 'LOW', 'PREV. CLOSE', 'Last Traded Pr
       'CHNG', '%CHNG', 'Number of shares', 'Total_value ', '52W H',
       '52W L', '365 D % CHNG',
       '30 D % CHNG']
```

In [6]:
```
market_data
```

Out[6]:

	SYMBOL	OPEN	HIGH	LOW	PREV. CLOSE	Last Traded Price	CHNG	%CHNG	Number of shares	Total_
1	LTI	3955.00	4150.00	3936.05	3896.50	4142.00	245.50	6.30	1096653	4.488425
2	EDELWEISS	80.00	85.40	77.00	78.75	83.65	4.90	6.22	11933408	9.817615
3	HUDCO	48.45	51.90	47.80	48.10	51.00	2.90	6.03	7522890	3.809591
4	COFORGE	2620.00	2757.00	2620.00	2582.80	2733.00	150.20	5.82	1193338	3.248612
5	MINDTREE	1829.00	1902.00	1811.55	1796.35	1893.00	96.65	5.38	3329555	6.207889
...	...	...	...	...	...	...	...	...	...	
196	GSPL	280.30	281.55	269.80	277.60	271.10	-6.50	-2.34	754885	2.071631
197	SBICARD	1061.80	1065.40	1022.10	1055.25	1028.00	-27.25	-2.58	1409421	1.461062
198	DHANI	305.60	305.60	288.05	303.20	293.80	-9.40	-3.10	2448102	7.215536
199	ADANITRANS	740.00	765.00	730.55	757.65	733.00	-24.65	-3.25	662901	4.891480
200	SBILIFE	973.70	974.00	936.00	971.30	937.50	-33.80	-3.48	2362998	2.247282

200 rows × 14 columns

Query - Number of Listed Companies in NIFTY 200

In [85]:

```
print("Number of Company listed in NIFTY 200 : ",market_data['SYMBOL'].count())
```

```
Number of Company listed in NIFTY 200 :  200
```

In [54]:

```
share_price=market_data[["SYMBOL",'Last Traded Price']]
```

Query - Share with highest price

In [57]:

```
print(share_price.max())
```

```
SYMBOL                ZEEL
Last Traded Price    88800
dtype: object
```

Query - share with lowest price

In [58]:

```
print(share_price.min())
```

```
SYMBOL               AARTIIND
Last Traded Price        10.3
dtype: object
```

Query - Average share price for the shares listed in NIFTY 200

In [70]:

```
share_price_list=pd.Series(share_price['Last Traded Price'])
print('Average Share Price : ',share_price_list.mean())
```

```
Average Share Price :  2198.90525
```

Query - Total number of shares traded in that day

In [72]:

```
print("Total number of Shares Traded : ",market_data['Number of shares'].sum())
```

```
Total number of Shares Traded :  1238073055
```

Query : Average change in price of shares

```
In [75]: print('Average Change in share price is {} %'.format(market_data['%CHNG'].mean()))
```

```
Average Change in share price is 0.7003999999999998 %
```

Query - standard deviation of %change in share price

```
In [77]: print("Standard deviation of % change in share price : ",market_data['%CHNG'].std())
```

```
Standard deviation of % change in share price :  1.5590987088441821
```

Query - difference between 52 weeks higher and lower price of shares

```
In [81]: market_data['difference']=abs(market_data['52W H']-market_data['52W L'])
         market_data[['SYMBOL','difference']]
```

```
<ipython-input-81-e458995b4c18>: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/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  market_data['difference']=abs(market_data['52W H']-market_data['52W L'])
```

Out[81]:

	SYMBOL	difference
1	LTI	3273.00
2	EDELWEISS	56.75
3	HUDCO	36.90
4	COFORGE	2174.20
5	MINDTREE	1210.00
...	...	...
196	GSPL	165.30
197	SBICARD	644.00
198	DHANI	332.45
199	ADANITRANS	707.45
200	SBILIFE	464.35

200 rows × 2 columns

Query - sum of difference of 52 weeks higher and lower

```
In [83]: market_data['difference'].sum()
```

```
Out[83]: 255605.09
```

Query - sum of values of total shares traded in that day

```
In [103… market_data['Total_value '].sum()
```

```
Out[103… 419030903773.69
```

Query - varience in the share price

```
In [30]: market_data['Last Traded Price'].var()
```

```
Out[30]: 50780313.16794967
```

Query - compute the summary of data set

In [104…

```
market_data.describe()
```

Out[104…

	OPEN	HIGH	LOW	PREV. CLOSE	Last Traded Price	CHNG	%CHN
count	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.0000
mean	2202.143500	2220.229250	2177.579250	2187.655000	2198.905250	11.272750	0.7004
std	7177.212314	7190.285479	7081.753328	7126.568469	7126.030674	55.675893	1.5590
min	10.200000	10.400000	10.100000	10.100000	10.300000	-267.700000	-3.4800
25%	253.462500	255.250000	248.325000	251.825000	252.837500	-0.900000	-0.3000
50%	730.475000	743.300000	721.150000	727.600000	728.900000	2.225000	0.5800
75%	1711.750000	1719.275000	1689.862500	1700.537500	1696.250000	13.112500	1.5600
max	89702.750000	89703.000000	88379.450000	89067.700000	88800.000000	558.250000	6.3000

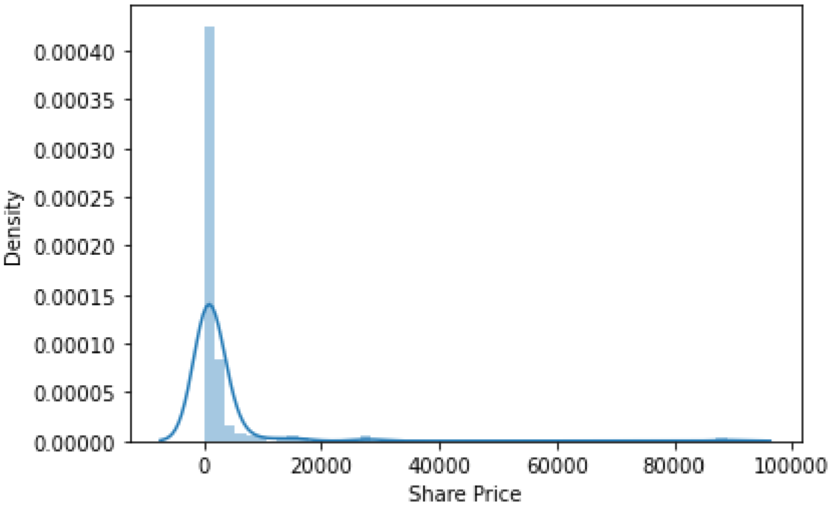
Graphical Representation

Distribution plot of Share Price

In [33]:

```
sb.distplot(market_data['Last Traded Price'])
plt.xlabel('Share Price')
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Pl
ease adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

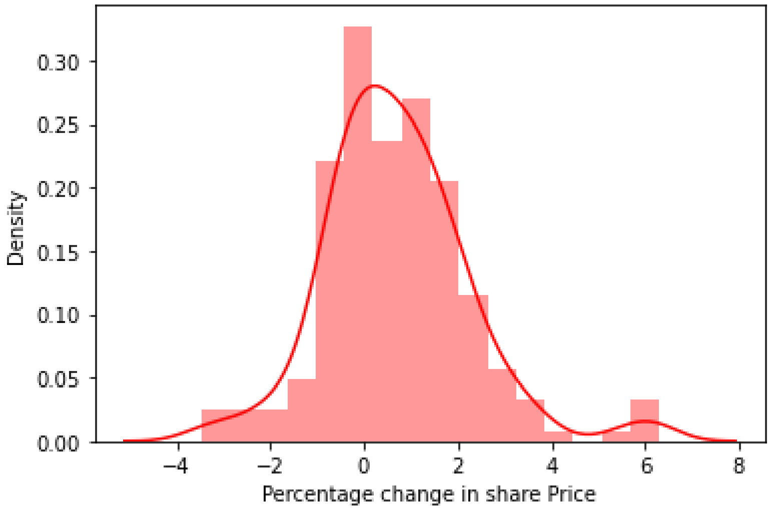


Histogram of percentage change in share price

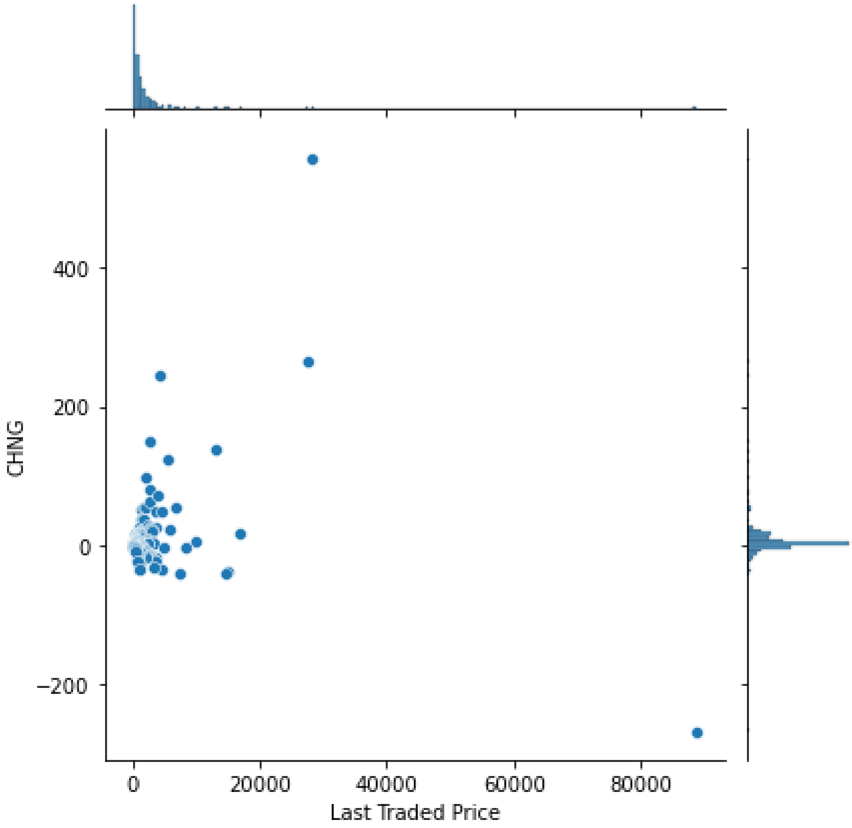
In [34]:

```
sb.distplot(market_data['%CHNG'],color='r')
plt.xlabel('Percentage change in share Price')
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Pl
ease adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

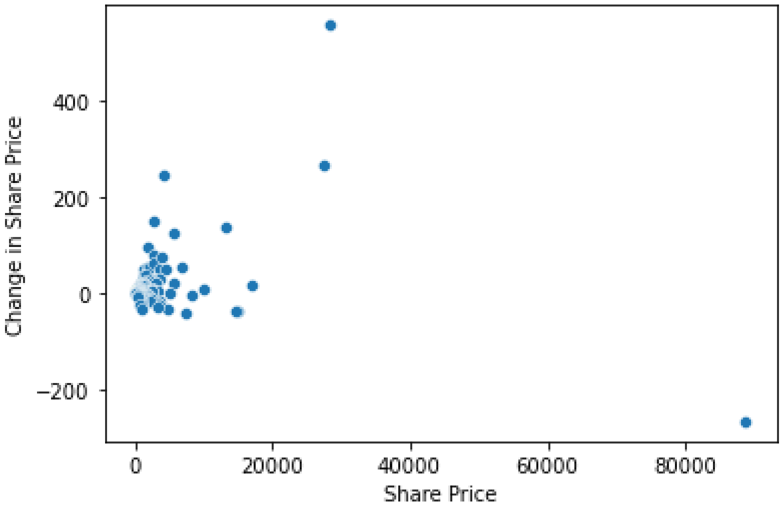


In [18]:
```
sb.jointplot(x=market_data['Last Traded Price'],y=market_data['CHNG'])
plt.show()
```



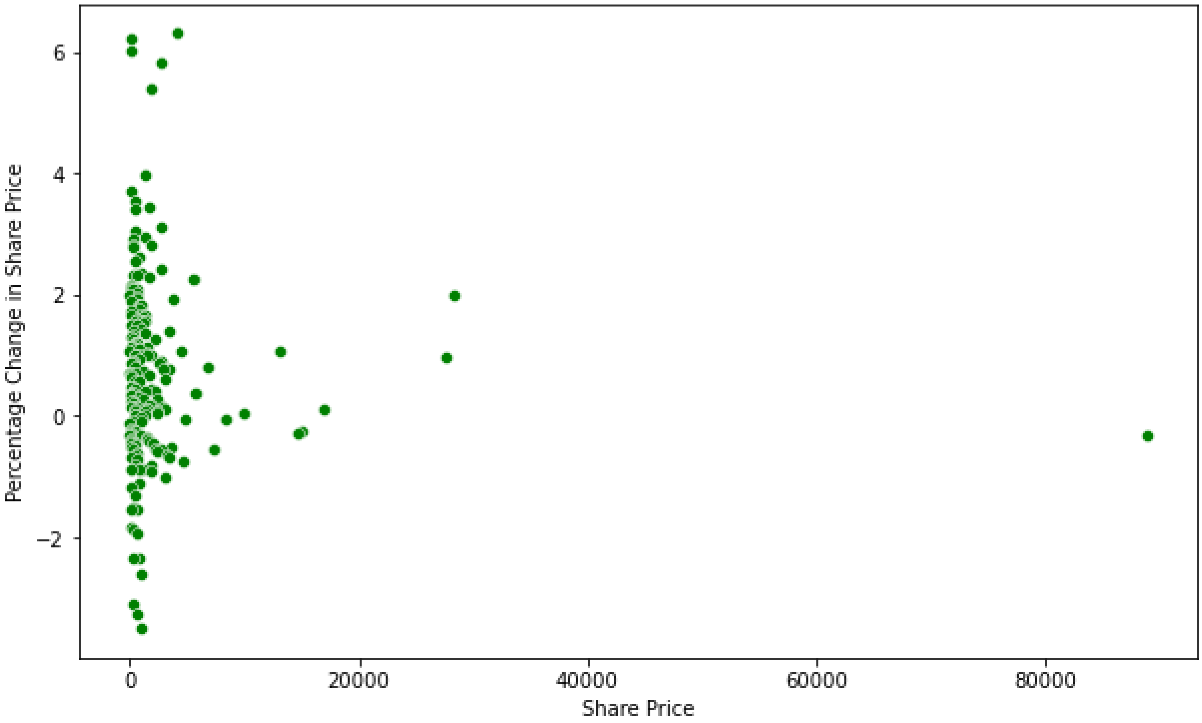
Plot of Share Price v/s Change in Share Price

In [35]:
```
sb.scatterplot(x=market_data['Last Traded Price'],y=market_data['CHNG'])
plt.xlabel('Share Price')
plt.ylabel('Change in Share Price')
plt.show()
```



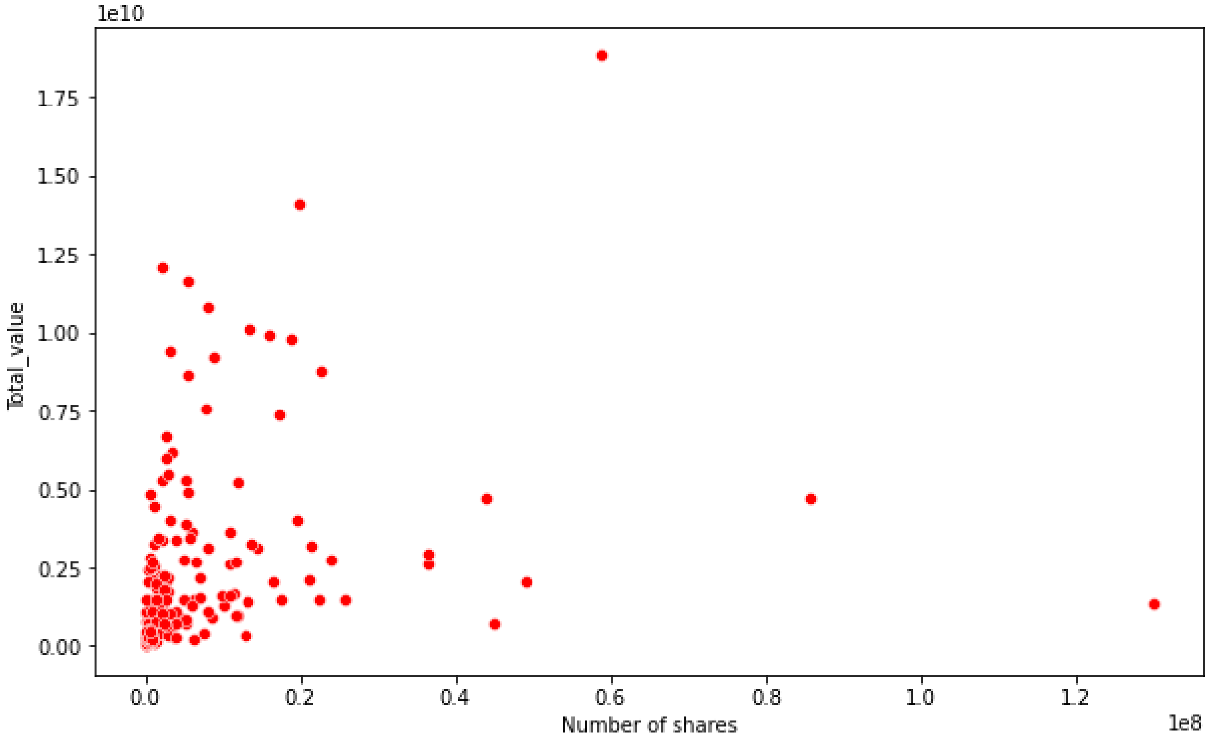
Plot of Share Price v/s Percentage Change in Share Price

In [37]:
```
plt.figure(figsize=(10,6))
sb.scatterplot(x=market_data['Last Traded Price'],y=market_data['%CHNG'],color='gree
plt.xlabel('Share Price')
plt.ylabel('Percentage Change in Share Price')
plt.show()
```



Plot of Total Share value v/s Number of Shares

In [41]:
```
plt.figure(figsize=(10,6))
sb.scatterplot(x=market_data['Number of shares'],y=market_data['Total_value '],color
plt.show()
```


1e10
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Total_value
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Number of shares
1e8

```
In [ ]:
```