In [1]:

```
import math
```

In [2]:

```
import cufflinks as cf
```

In [3]:

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

In [4]:

```
from plotly import __version__
```

In [5]:

```
from plotly import offline
```

In [6]:

```
from plotly.offline import download_plotlyjs,init_notebook_mode,plot,iplot
```

In [7]:

```
cf.go_offline()
```

In [8]:

```
%matplotlib inline
```

In [9]:

```
df = pd.read_csv('TATASTEEL.csv')
```

In [10]:

```
df
```

Out[10]:

	Date	Open	High	Low	Close
0	07-04-2017	460.10	467.90	459.75	462.40
1	10-04-2017	467.10	474.00	448.00	449.90
2	11-04-2017	448.00	455.50	442.05	450.75
3	12-04-2017	450.75	455.80	448.25	454.40
4	13-04-2017	460.00	460.70	449.10	450.85
...	...	...	...	...	...
996	16-04-2021	935.00	956.00	930.05	940.75
997	19-04-2021	948.30	983.00	944.30	977.75
998	20-04-2021	985.00	986.00	962.00	971.40
999	22-04-2021	983.00	1036.95	983.00	1031.35
1000	23-04-2021	1024.00	1052.60	1011.10	1034.00

1001 rows × 5 columns

In [18]:

```
index=0
for num in df['Low']:
    df['low']=math.floor(num)
df
```

Out[18]:

	Date	Open	High	Low	Close	open	close	high	low
0	07-04-2017	460.10	467.90	459.75	462.40	460	462	467	1011
1	10-04-2017	467.10	474.00	448.00	449.90	467	449	474	1011
2	11-04-2017	448.00	455.50	442.05	450.75	448	450	455	1011
3	12-04-2017	450.75	455.80	448.25	454.40	450	454	455	1011
4	13-04-2017	460.00	460.70	449.10	450.85	460	450	460	1011
...	...	...	...	...	...	...	...	...	...
996	16-04-2021	935.00	956.00	930.05	940.75	935	940	956	1011
997	19-04-2021	948.30	983.00	944.30	977.75	948	977	983	1011
998	20-04-2021	985.00	986.00	962.00	971.40	985	971	986	1011
999	22-04-2021	983.00	1036.95	983.00	1031.35	983	1031	1036	1011
1000	23-04-2021	1024.00	1052.60	1011.10	1034.00	1024	1034	1052	1011

1001 rows × 9 columns

In [19]:

```
index=0
for num in df['Low']:
    df['low'][index] = math.floor(num)
    index=index+1
df
```

Out[19]:

	Date	Open	High	Low	Close	open	close	high	low
0	07-04-2017	460.10	467.90	459.75	462.40	460	462	467	459
1	10-04-2017	467.10	474.00	448.00	449.90	467	449	474	448
2	11-04-2017	448.00	455.50	442.05	450.75	448	450	455	442
3	12-04-2017	450.75	455.80	448.25	454.40	450	454	455	448
4	13-04-2017	460.00	460.70	449.10	450.85	460	450	460	449
...	...	...	...	...	...	...	...	...	...
996	16-04-2021	935.00	956.00	930.05	940.75	935	940	956	930
997	19-04-2021	948.30	983.00	944.30	977.75	948	977	983	944
998	20-04-2021	985.00	986.00	962.00	971.40	985	971	986	962
999	22-04-2021	983.00	1036.95	983.00	1031.35	983	1031	1036	983
1000	23-04-2021	1024.00	1052.60	1011.10	1034.00	1024	1034	1052	1011

1001 rows × 9 columns

In [24]:

```
df
```

Out[24]:

	Date	open	close	high	low
0	07-04-2017	460	462	467	459
1	10-04-2017	467	449	474	448
2	11-04-2017	448	450	455	442
3	12-04-2017	450	454	455	448
4	13-04-2017	460	450	460	449
...	...	...	...	...	...
996	16-04-2021	935	940	956	930
997	19-04-2021	948	977	983	944
998	20-04-2021	985	971	986	962
999	22-04-2021	983	1031	1036	983
1000	23-04-2021	1024	1034	1052	1011

1001 rows × 5 columns

In [23]:

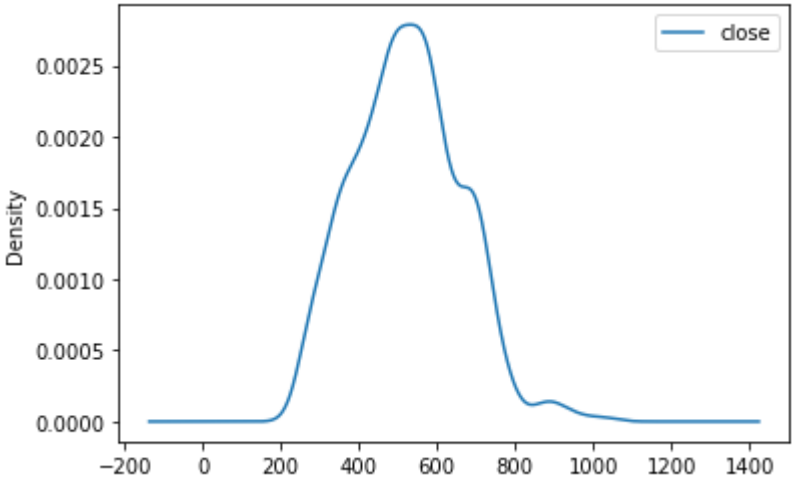
```
# df.drop('Close',axis=1,inplace=True)
```

In [25]:

```
df.plot(x='Date',y='close',kind='kde')
```

Out[25]:

<AxesSubplot:ylabel='Density'>

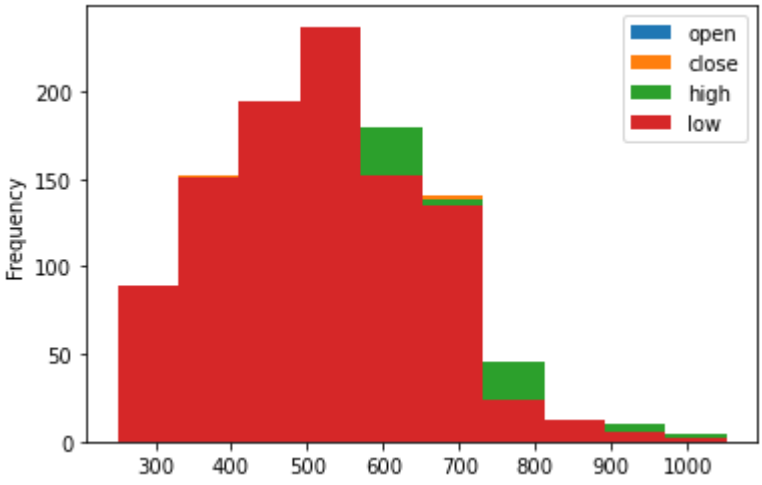


In [26]:

```
df.plot(kind='hist')
```

Out[26]:

<AxesSubplot:ylabel='Frequency'>

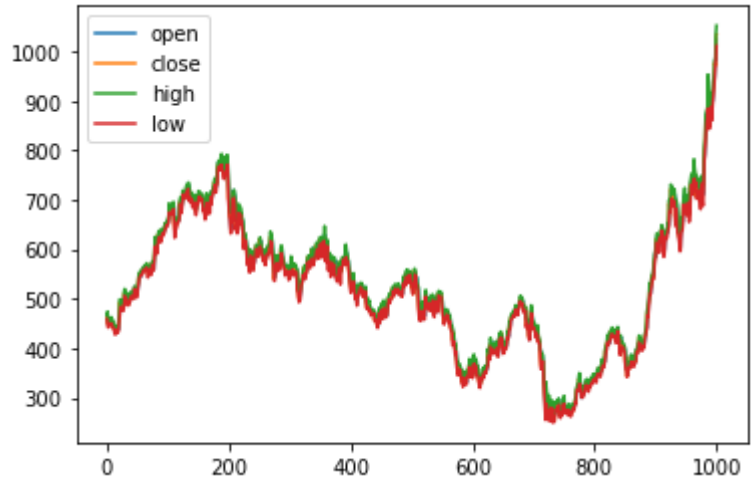


In [29]:

```
df.plot(kind='line')
```

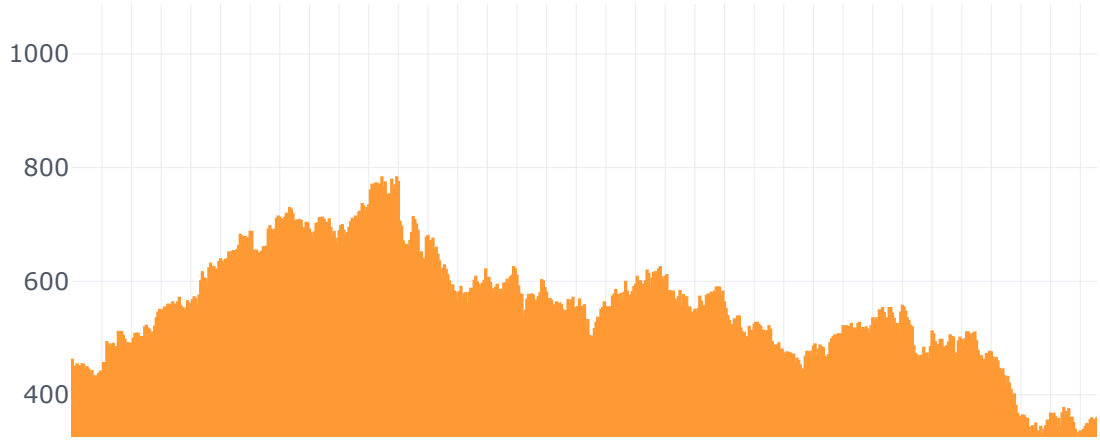
Out[29]:

<AxesSubplot:>



In [38]:

```
df.iplot(x='Date',y='close',kind='bar',bins=None)
```



In [54]:

```
df.iplot(x='Date',y='close',kind='candle')  # candle graph
```

In [85]:

```
mean = df['close'].mean()
mean
df['c-m'] = mean
```

In [86]:

```
index=0
for num in df['close']:
    df['c-m'][index] = num - mean
    index = index + 1
df
```

Out[86]:

	Date	open	close	high	low	m-c	c-m
0	07-04-2017	460	462	467	459	-61.985015	-61.985015
1	10-04-2017	467	449	474	448	-74.985015	-74.985015
2	11-04-2017	448	450	455	442	-73.985015	-73.985015
3	12-04-2017	450	454	455	448	-69.985015	-69.985015
4	13-04-2017	460	450	460	449	-73.985015	-73.985015
...	...	...	...	...	...	...	...
996	16-04-2021	935	940	956	930	416.014985	416.014985
997	19-04-2021	948	977	983	944	453.014985	453.014985
998	20-04-2021	985	971	986	962	447.014985	447.014985
999	22-04-2021	983	1031	1036	983	507.014985	507.014985
1000	23-04-2021	1024	1034	1052	1011	510.014985	510.014985

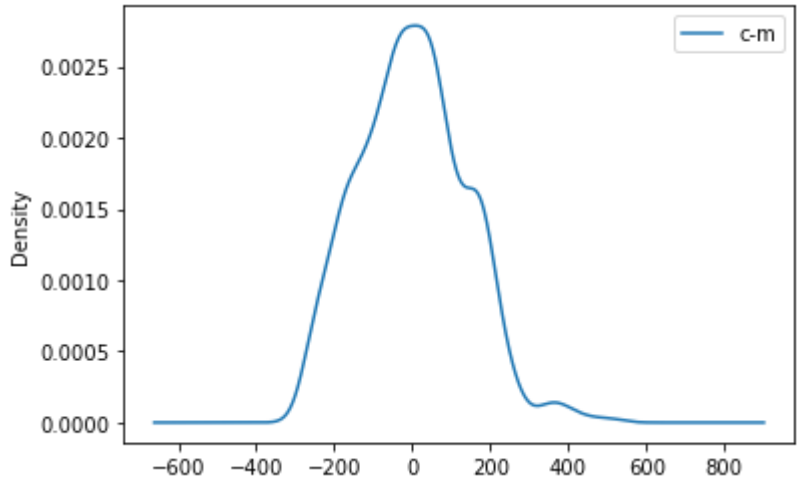
1001 rows × 7 columns

In [87]:

```
df.plot(x='Date',y='c-m',kind='kde')
```

Out[87]:

```
<AxesSubplot:ylabel='Density'>
```

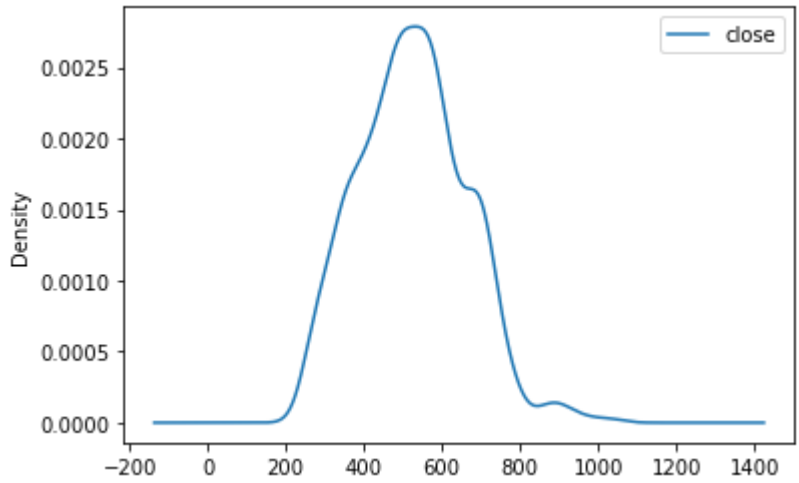


In [88]:

```
df.plot(x='Date',y='close',kind='kde') # both upper one and this one same graph
```

Out[88]:

```
<AxesSubplot:ylabel='Density'>
```



In [97]:

```
# df.drop('m-c',axis=1,inplace=True)
df
df['mean'] = mean
df
```

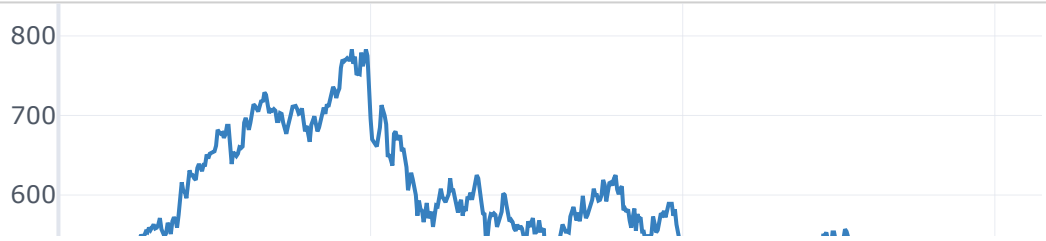
Out[97]:

	Date	open	close	high	low	c-m	mean
0	07-04-2017	460	462	467	459	-61.985015	523.985015
1	10-04-2017	467	449	474	448	-74.985015	523.985015
2	11-04-2017	448	450	455	442	-73.985015	523.985015
3	12-04-2017	450	454	455	448	-69.985015	523.985015
4	13-04-2017	460	450	460	449	-73.985015	523.985015
...	...	...	...	...	...	...	...
996	16-04-2021	935	940	956	930	416.014985	523.985015
997	19-04-2021	948	977	983	944	453.014985	523.985015
998	20-04-2021	985	971	986	962	447.014985	523.985015
999	22-04-2021	983	1031	1036	983	507.014985	523.985015
1000	23-04-2021	1024	1034	1052	1011	510.014985	523.985015

1001 rows × 7 columns

In [98]:

```
df[['mean','close']].iplot(kind='scatter',width = 2)
```



In [ ]:

```

```

In [116]:

```
index=0
for num in df['open']:
    df['open'][index] = int(num)
    index=index+1
# df
type(df['high'][2])
```

Out[116]:

```
numpy.int64
```

```
D:\Persnol_Documents\TOOLS\temp\lib\site-packages\cufflinks\plotlytools.py i
n <lambda>(x)
    846                                 if kind in ('spread','ratio'):
    847                                         if kind=='spread':
--> 848                                                 trace=self.a
pply(lambda x:x[0]-x[1],axis=1)
    849                                                 positive=tra
ce.apply(lambda x:x if x>=0 else pd.np.nan)
    850                                                 negative=tra
ce.apply(lambda x:x if x<0 else pd.np.nan)

TypeError: unsupported operand type(s) for -: 'str' and 'int'
```

In [129]:

```
df2 = pd.read_csv('TATASTEEL.csv')  # float data
```

In [130]:

```
df2
```

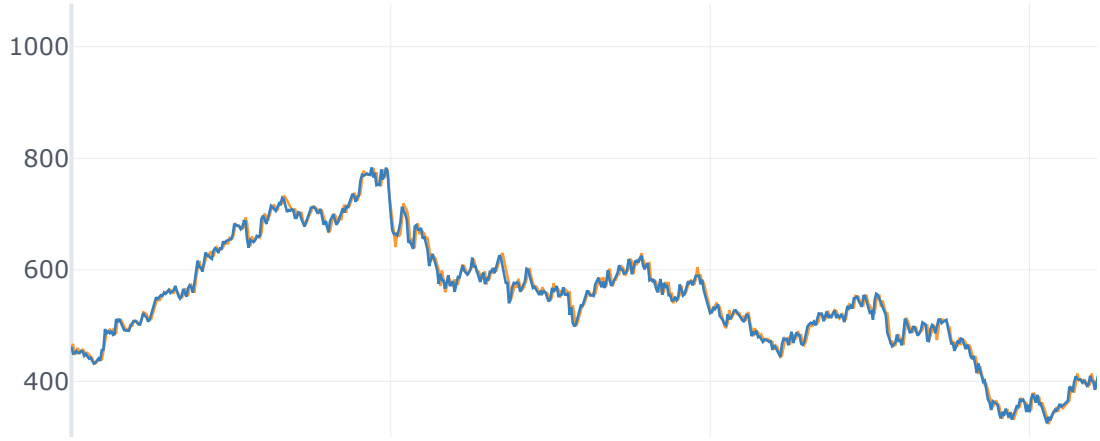
Out[130]:

	Date	Open	High	Low	Close
0	07-04-2017	460.10	467.90	459.75	462.40
1	10-04-2017	467.10	474.00	448.00	449.90
2	11-04-2017	448.00	455.50	442.05	450.75
3	12-04-2017	450.75	455.80	448.25	454.40
4	13-04-2017	460.00	460.70	449.10	450.85
...	...	...	...	...	...
996	16-04-2021	935.00	956.00	930.05	940.75
997	19-04-2021	948.30	983.00	944.30	977.75
998	20-04-2021	985.00	986.00	962.00	971.40
999	22-04-2021	983.00	1036.95	983.00	1031.35
1000	23-04-2021	1024.00	1052.60	1011.10	1034.00

1001 rows × 5 columns

In [135]:

```
df2[['Open','Close']].iplot(kind='spread')
```



In [ ]:

```

```