Effect of COVID-19 on Financial Market Analysis of listed Companies.

Prepared By : Jay Patel

1. Reliance India GoldBees MF
2. LIC Housing Finance Limited (LICHSGFIN.NS)
3. LIC Housing Finance Limited (LICHSGFIN.BO)
4. Reliance Industries Limited (RELIANCE.NS)
5. Reliance Industries Limited (RELIANCE.BO)

Notice : In this data project I'll only focus on exploratory data analysis of stock prices. Keep in mind, this project is just meant to practice visualization and python pandas skills, it is not meant to be a robust financial analysis or be taken as financial advice.

In [82]:
```
from pandas_datareader import data, wb
from pandas.util.testing import assert_frame_equal
import pandas as pd
import numpy as np
import datetime
import seaborn as sns
%matplotlib inline
```

In [6]:
```
start = datetime.datetime(2020,2,15)
end = datetime.datetime(2020,4,15)
```

In [15]:
```
# LIC India GoldBees MF
NIP_GoldBees = data.DataReader("GOLDBEES.NS", 'yahoo', start, end)

# LIC Housing Finance Limited (LICHSGFIN.NS)
LIC_Housing = data.DataReader("LICHSGFIN.NS", 'yahoo', start, end)

# LIC Housing Finance Limited (LICHSGFIN.BO)
LIC_Housing_BM = data.DataReader("LICHSGFIN.BO", 'yahoo', start, end)

# Reliance Industries Limited (RELIANCE.NS)
REL_IN = data.DataReader("RELIANCE.NS", 'yahoo', start, end)

# Reliance Industries Limited (RELIANCE.BO)
REL_BM = data.DataReader("RELIANCE.BO", 'yahoo', start, end)
```

In [16]:
```
NIP_GoldBees.head()
```

Out[16]:

Date	High	Low	Open	Close	Volume	Adj Close
2020-02-17	37.299999	35.869999	36.250000	35.930000	1117666.0	35.930000
2020-02-18	36.180000	36.049999	36.139999	36.139999	864267.0	36.139999
2020-02-19	36.900002	36.200001	36.900002	36.709999	1416101.0	36.709999
2020-02-20	37.000000	36.650002	36.750000	36.820000	1950317.0	36.820000
2020-02-24	39.000000	37.799999	37.810001	38.580002	11623173.0	38.580002

In [18]:
```
LIC_Housing.head()
```

Out[18]:

Date	High	Low	Open	Close	Volume	Adj Close
2020-02-17	399.000000	351.100006	399.000000	379.850006	36528007	379.850006
2020-02-18	382.000000	354.299988	380.000000	363.950012	18284085	363.950012
2020-02-19	374.399994	361.250000	366.950012	366.200012	6880554	366.200012
2020-02-20	375.649994	362.250000	362.250000	364.600006	6059345	364.600006
2020-02-24	361.700012	346.399994	357.950012	348.299988	6748965	348.299988

In [19]:
```
LIC_Housing_BM.head()
```

Out[19]:

Date	High	Low	Open	Close	Volume	Adj Close
2020-02-17	401.000000	361.299988	401.000000	379.850006	1583548	379.850006
2020-02-18	382.450012	354.500000	382.000000	363.649994	752613	363.649994
2020-02-19	373.000000	361.500000	365.600006	366.049988	245465	366.049988
2020-02-20	375.500000	362.350006	364.700012	364.549988	269927	364.549988
2020-02-24	362.000000	346.899994	362.000000	348.549988	225241	348.549988

In [20]:
```
REL_IN.head()
```

Out[20]:

Date	High	Low	Open	Close	Volume	Adj Close
2020-02-17	1506.150024	1474.300049	1489.000000	1478.250000	6011340	1478.250000
2020-02-18	1475.900024	1457.400024	1475.900024	1467.400024	5086964	1467.400024
2020-02-19	1506.449951	1475.099976	1479.349976	1503.800049	6438918	1503.800049
2020-02-20	1508.000000	1483.800049	1497.000000	1485.949951	7722577	1485.949951
2020-02-24	1477.000000	1439.550049	1469.750000	1444.949951	9769743	1444.949951

In [21]:
```
REL_BM.head()
```

Out[21]:

Date	High	Low	Open	Close	Volume	Adj Close
2020-02-17	1506.000000	1474.650024	1490.000000	1478.400024	517807	1478.400024
2020-02-18	1479.000000	1458.000000	1479.000000	1466.099976	370782	1466.099976
2020-02-19	1506.500000	1475.550049	1475.550049	1504.199951	291146	1504.199951
2020-02-20	1507.949951	1483.449951	1498.699951	1485.500000	1511993	1485.500000
2020-02-24	1477.000000	1439.599976	1477.000000	1444.849976	400408	1444.849976

Effect of COVID-19 on Financial Market Analysis of listed Companies

1. Reliance India GoldBees MF
2. LIC Housing Finance Limited (LICHSGFIN.NS)
3. LIC Housing Finance Limited (LICHSGFIN.BO)
4. Reliance Industries Limited (RELIANCE.NS)
5. Reliance Industries Limited (RELIANCE.BO)

In [22]:
```
tickers = ['NIP_GoldBees','LIC_Housing','LIC_Housing_BM','REL_IN','REL_BM']
```

In [23]:
```
bank_stocks = pd.concat([NIP_GoldBees,LIC_Housing,LIC_Housing_BM,REL_IN,REL_BM],axis=1,keys=tickers)
```

In [24]:
```
bank_stocks.head()
```

Out[24]:

	NIP_GoldBees						LIC_Housing				...	REL_IN				R
	High	Low	Open	Close	Volume	Adj Close	High	Low	Open	Close	...	Open	Close	Volume	Adj Close	H
Date																
2020-02-17	37.299999	35.869999	36.250000	35.930000	1117666.0	35.930000	399.000000	351.100006	399.000000	379.850006	...	1489.000000	1478.250000	6011340	1478.250000	1
2020-02-18	36.180000	36.049999	36.139999	36.139999	864267.0	36.139999	382.000000	354.299988	380.000000	363.950012	...	1475.900024	1467.400024	5086964	1467.400024	1
2020-02-19	36.900002	36.200001	36.900002	36.709999	1416101.0	36.709999	374.399994	361.250000	366.950012	366.200012	...	1479.349976	1503.800049	6438918	1503.800049	1
2020-02-20	37.000000	36.650002	36.750000	36.820000	1950317.0	36.820000	375.649994	362.250000	362.250000	364.600006	...	1497.000000	1485.949951	7722577	1485.949951	1
2020-02-24	39.000000	37.799999	37.810001	38.580002	11623173.0	38.580002	361.700012	346.399994	357.950012	348.299988	...	1469.750000	1444.949951	9769743	1444.949951	1

5 rows × 30 columns

In [25]:
```
bank_stocks.columns.names = ['Bank Ticker','Stock Info']
```

In [26]:
```
bank_stocks.head()
```

Out[26]:

Bank Ticker	NIP_GoldBees						LIC_Housing				...	REL_IN				R
Stock Info	High	Low	Open	Close	Volume	Adj Close	High	Low	Open	Close	...	Open	Close	Volume	Adj Close	H
Date																
2020-02-17	37.299999	35.869999	36.250000	35.930000	1117666.0	35.930000	399.000000	351.100006	399.000000	379.850006	...	1489.000000	1478.250000	6011340	1478.250000	1
2020-02-18	36.180000	36.049999	36.139999	36.139999	864267.0	36.139999	382.000000	354.299988	380.000000	363.950012	...	1475.900024	1467.400024	5086964	1467.400024	1
2020-02-19	36.900002	36.200001	36.900002	36.709999	1416101.0	36.709999	374.399994	361.250000	366.950012	366.200012	...	1479.349976	1503.800049	6438918	1503.800049	1
2020-02-20	37.000000	36.650002	36.750000	36.820000	1950317.0	36.820000	375.649994	362.250000	362.250000	364.600006	...	1497.000000	1485.949951	7722577	1485.949951	1
2020-02-24	39.000000	37.799999	37.810001	38.580002	11623173.0	38.580002	361.700012	346.399994	357.950012	348.299988	...	1469.750000	1444.949951	9769743	1444.949951	1

5 rows × 30 columns

EDA

Let's explore the data a bit!

What is the max Close price for each bank's stock throughout the time period?

In [27]:
```
for tick in tickers:
    print(tick, bank_stocks[tick]['Close'].max())
```

```
NIP_GoldBees 43.2599983215332
LIC_Housing 379.8500061035156
LIC_Housing_BM 379.8500061035156
REL_IN 1503.800048828125
REL_BM 1504.199951171875
```

In [28]:
```
bank_stocks.xs(key='Close',axis=1,level='Stock Info')
```

Out[28]:

Bank Ticker	NIP_GoldBees	LIC_Housing	LIC_Housing_BM	REL_IN	REL_BM
Date					
2020-02-17	35.930000	379.850006	379.850006	1478.250000	1478.400024
2020-02-18	36.139999	363.950012	363.649994	1467.400024	1466.099976
2020-02-19	36.709999	366.200012	366.049988	1503.800049	1504.199951
2020-02-20	36.820000	364.600006	364.549988	1485.949951	1485.500000
2020-02-24	38.580002	348.299988	348.549988	1444.949951	1444.849976
2020-02-25	37.439999	350.700012	350.850006	1416.400024	1416.300049
2020-02-26	37.520000	348.149994	347.899994	1392.000000	1391.400024
2020-02-27	37.380001	336.350006	336.399994	1386.250000	1385.800049
2020-02-28	37.369999	320.250000	320.549988	1328.650024	1328.650024
2020-03-02	37.029999	322.450012	321.299988	1316.150024	1314.849976
2020-03-03	37.119999	330.000000	330.299988	1342.849976	1343.650024
2020-03-04	38.099998	320.899994	320.799988	1339.699951	1339.150024
2020-03-05	38.040001	320.399994	320.399994	1311.150024	1311.500000
2020-03-06	39.049999	302.299988	302.299988	1271.000000	1270.050049
2020-03-09	38.459999	284.399994	302.299988	1114.150024	1270.050049
2020-03-11	38.180000	288.549988	288.200012	1153.550049	1153.250000
2020-03-12	37.959999	255.350006	255.600006	1063.000000	1061.599976
2020-03-13	36.619999	280.250000	280.250000	1105.300049	1106.900024
2020-03-16	35.669998	255.449997	255.550003	1015.700012	1015.250000
2020-03-17	34.830002	251.199997	250.750000	1008.000000	1008.900024
2020-03-18	35.439999	226.800003	226.899994	968.500000	968.849976
2020-03-19	35.209999	223.550003	223.699997	917.700012	917.099976
2020-03-20	36.110001	225.050003	223.699997	1017.950012	1020.200012
2020-03-23	35.020000	192.449997	191.600006	884.049988	883.849976
2020-03-24	36.389999	192.300003	192.500000	943.400024	943.099976
2020-03-25	36.990002	210.250000	210.100006	1082.250000	1081.250000
2020-03-26	36.720001	228.600006	228.600006	1066.199951	1074.750000
2020-03-27	38.169998	237.250000	237.550003	1065.599976	1065.500000
2020-03-30	39.740002	228.899994	228.550003	1030.449951	1032.349976
2020-03-31	38.209999	235.300003	235.199997	1113.750000	1112.449951
2020-04-01	38.240002	233.000000	232.899994	1080.449951	1080.650024
2020-04-03	38.790001	218.649994	218.850006	1077.449951	1078.199951
2020-04-07	NaN	216.000000	215.899994	1206.099976	1206.400024
2020-04-08	NaN	223.750000	223.699997	1192.150024	1192.150024
2020-04-09	NaN	255.350006	255.600006	1219.949951	1219.199951
2020-04-13	NaN	244.000000	243.899994	1189.150024	1189.250000
2020-04-15	NaN	248.149994	248.350006	1149.849976	1150.050049
2020-04-16	43.259998	257.299988	257.299988	1168.050049	1168.699951

In [29]:
```
bank_stocks.xs(key='Close',axis=1,level='Stock Info').max()
```

Out[29]:
```
Bank Ticker
NIP_GoldBees        43.259998
LIC_Housing        379.850006
LIC_Housing_BM     379.850006
REL_IN            1503.800049
REL_BM            1504.199951
dtype: float64
```

Create a new empty DataFrame called returns. This dataframe will contain the returns for each bank's stock. returns are typically defined by:**

$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}} = \frac{p_t}{p_{t-1}} - 1 \tag{1}$$

In [30]:
```
returns = pd.DataFrame()
```

In [31]:
```
for tick in tickers:
    returns[tick+ ' Return'] = bank_stocks[tick]['Close'].pct_change()
```

Return of the each Bank

In [32]:
```
returns.head()
```

Out[32]:

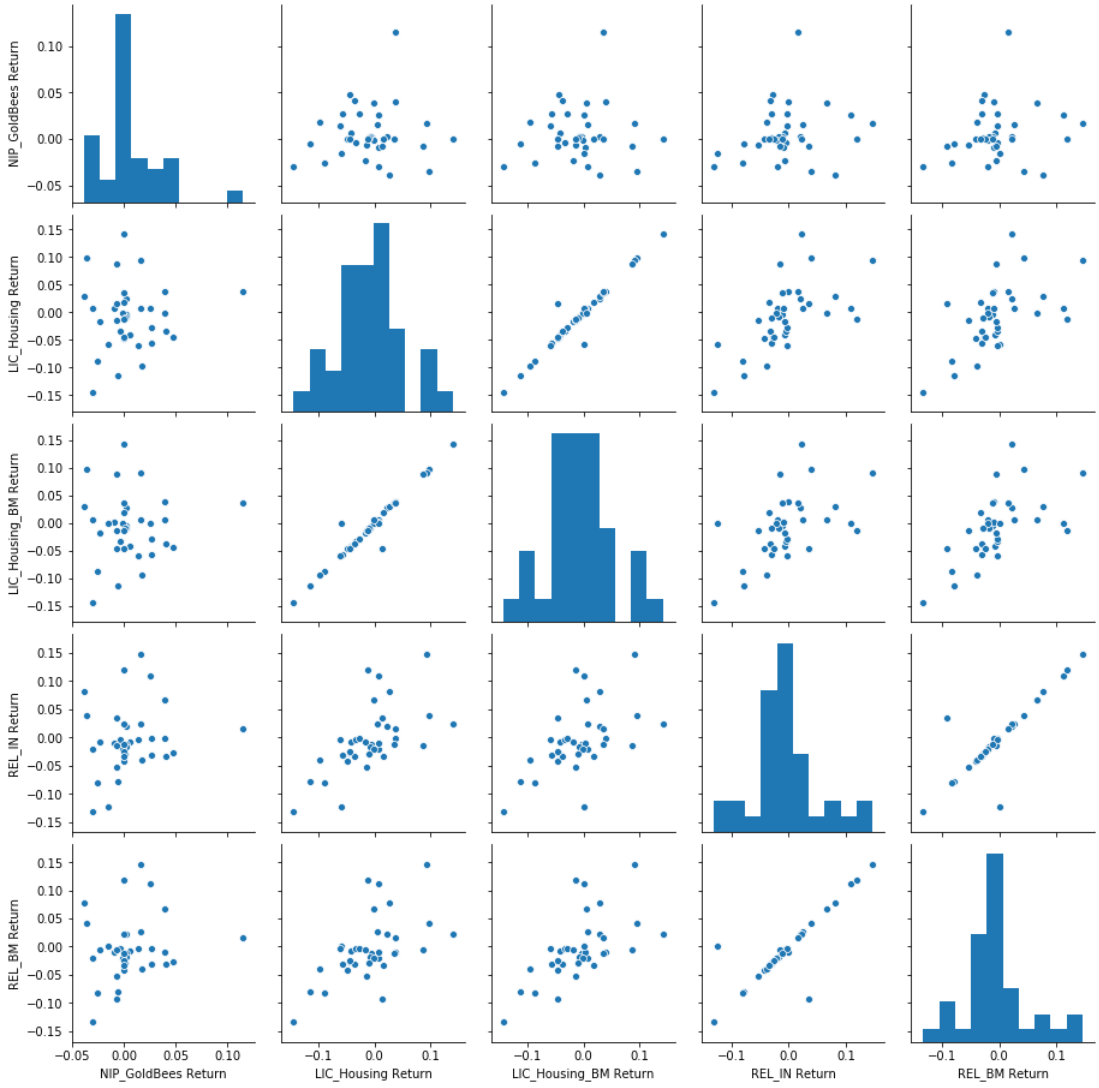
Date	NIP_GoldBees Return	LIC_Housing Return	LIC_Housing_BM Return	REL_IN Return	REL_BM Return
2020-02-17	NaN	NaN	NaN	NaN	NaN
2020-02-18	0.005845	-0.041859	-0.042648	-0.007340	-0.008320
2020-02-19	0.015772	0.006182	0.006600	0.024806	0.025987
2020-02-20	0.002996	-0.004369	-0.004098	-0.011870	-0.012432
2020-02-24	0.047800	-0.044707	-0.043890	-0.027592	-0.027365

In [33]:
```
import seaborn as sns
```

Let's find out all correlated data.

In [34]:
```
sns.pairplot(returns[1:])
```

Out[34]:
```
<seaborn.axisgrid.PairGrid at 0x24f3ec1e388>
```



Let's find out on what dates each bank stock had the best and worst single day returns. You should notice that 4 of the banks share the same day for the worst drop, did anything significant happen that day?

In [35]:
```
returns.min() # This are % values
```

Out[35]:
```
NIP_GoldBees Return     -0.038500
LIC_Housing Return      -0.144857
LIC_Housing_BM Return   -0.143496
REL_IN Return           -0.131539
REL_BM Return           -0.133650
dtype: float64
```

In [36]:
```
returns.idxmin() # This is the date on which all 4 banks has low data values
```

Out[36]:
```
NIP_GoldBees Return     2020-03-31
LIC_Housing Return      2020-03-23
LIC_Housing_BM Return   2020-03-23
REL_IN Return           2020-03-23
REL_BM Return           2020-03-23
dtype: datetime64[ns]
```

Quick Update : Here is what happened on March 23, 2020.

ANS. U.S. stock futures tumbled in early Monday trading as the death toll from the Coronavirus continued to rise globally.

Read more at: https://www.bloombergquint.com/markets/all-you-need-to-know-going-into-trade-on-march-23 (https://www.bloombergquint.com/markets/all-you-need-to-know-going-into-trade-on-march-23)

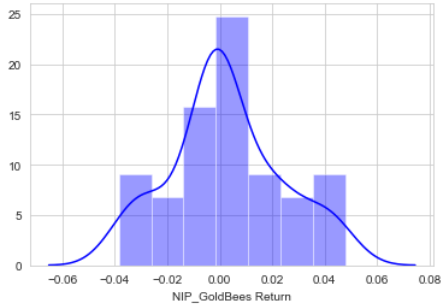
Take a look at the standard deviation of the returns, which stock would you classify as the riskiest over the entire time period? Which would you classify as the riskiest for the year 2020?

In [37]:
```
returns['2020-02-15':'2020-04-15'].std()
```

Out[37]:
```
NIP_GoldBees Return     0.021554
LIC_Housing Return      0.058751
LIC_Housing_BM Return   0.058198
REL_IN Return           0.058760
REL_BM Return           0.057025
dtype: float64
```

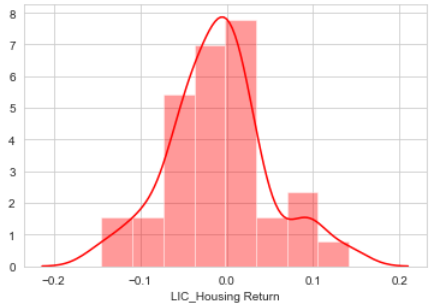
In [52]:
```
# 'NIP_GoldBees','LIC_Housing','LIC_Housing_BM','REL_IN','REL_BM'
sns.distplot(returns['2020-02-15':'2020-04-15']['NIP_GoldBees Return'],color='blue')
```

Out[52]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24f46efb748>
```



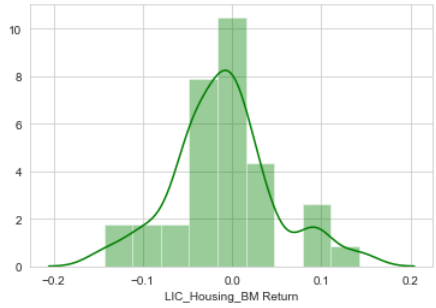
In [49]:
```
sns.distplot(returns['2020-02-15':'2020-04-15']['LIC_Housing Return'],color='red')
```

Out[49]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24f46eb9fc8>
```



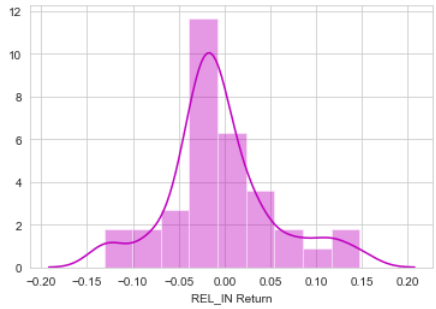
In [50]:
```
sns.distplot(returns['2020-02-15':'2020-04-15']['LIC_Housing_BM Return'],color='green')
```

Out[50]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24f45c30608>
```



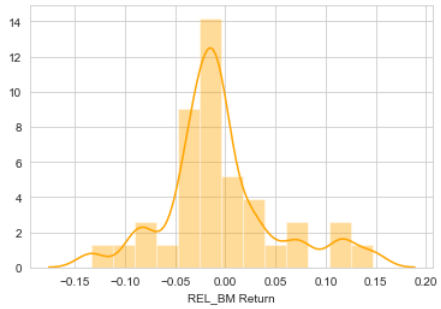
In [54]:
```
sns.distplot(returns['2020-02-15':'2020-04-15']['REL_IN Return'],color='m')
```

Out[54]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24f46da08c8>
```



In [55]:
```
sns.distplot(returns['2020-02-15':'2020-04-15']['REL_BM Return'],color='orange')
```

Out[55]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24f46e2a208>
```



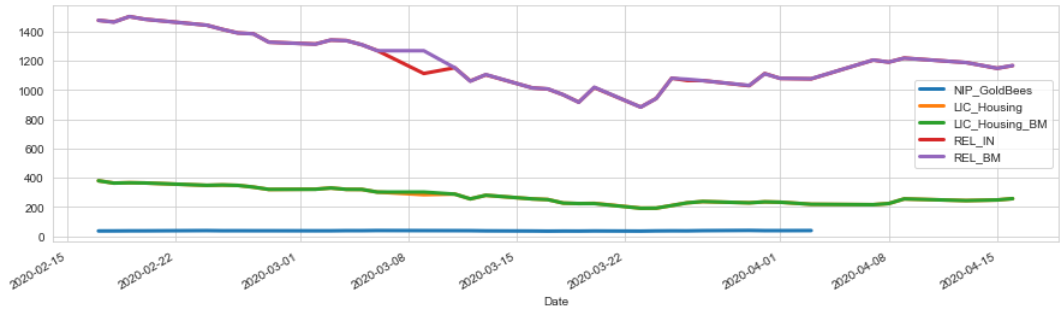
In [44]:
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
%matplotlib inline

# Optional Plotly Method Imports
import plotly
import cufflinks as cf
cf.go_offline()
```

Create a line plot showing Close price for each bank for the entire index of time.

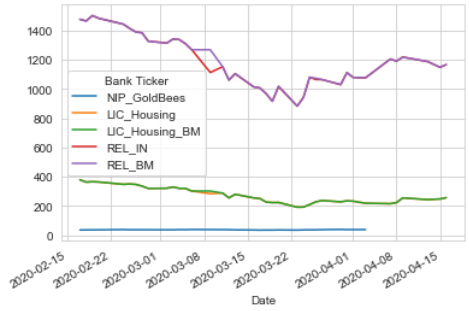
In [46]:
```
for tick in tickers:
    bank_stocks[tick]['Close'].plot(label=tick,figsize=(15,4),lw=3)
plt.legend()
```

Out[46]: <matplotlib.legend.Legend at 0x24f449c6b88>

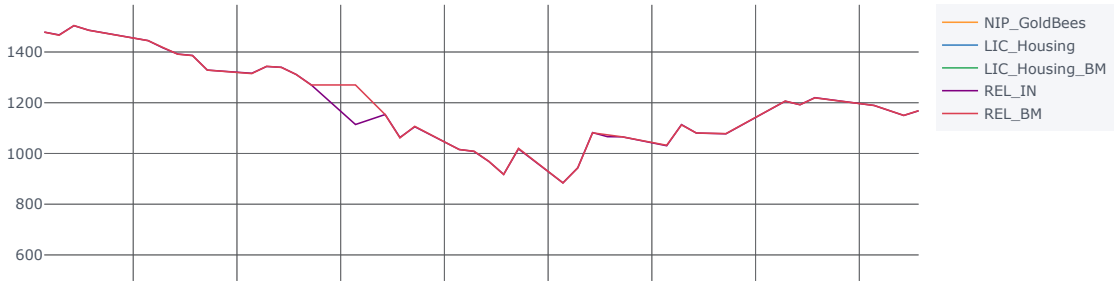


In [47]:
```
bank_stocks.xs('Close',axis=1,level='Stock Info').plot()
```

Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x24f44a1b708>



In [48]:
```
bank_stocks.xs('Close',axis=1,level='Stock Info').iplot()
```



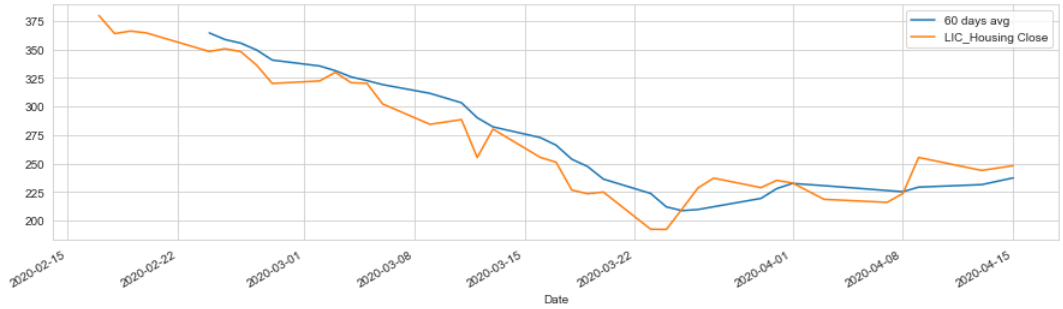
Moving Averages

Let's analyze the moving averages for these stocks in the year 2020.

Plot the rolling 60 day average against the Close Price for LIC's stock for the year 2020

In [64]:
```
# 'NIP_GoldBees','LIC_Housing','LIC_Housing_BM','REL_IN','REL_BM'
plt.figure(figsize=(15,4))
LIC_Housing['Close']['2020-02-15':'2020-04-15'].rolling(window=5).mean().plot(label = '60 days avg')
LIC_Housing['Close']['2020-02-15':'2020-04-15'].plot(label = 'LIC_Housing Close')
plt.legend()
```

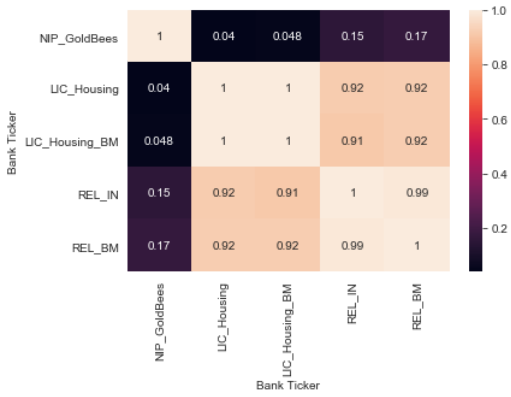
Out[64]: <matplotlib.legend.Legend at 0x24f4869e908>



Let's create Heat maps.

In [66]:
```
sns.heatmap(bank_stocks.xs(key='Close',axis=1,level='Stock Info').corr(),annot=True)
```

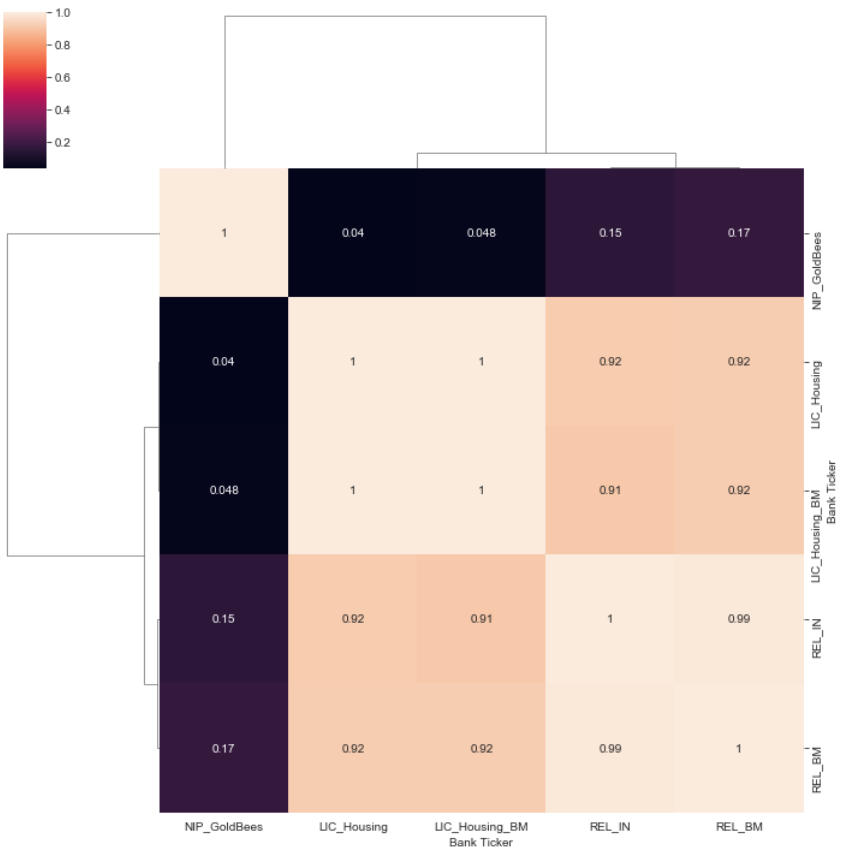
Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x24f487f36c8>



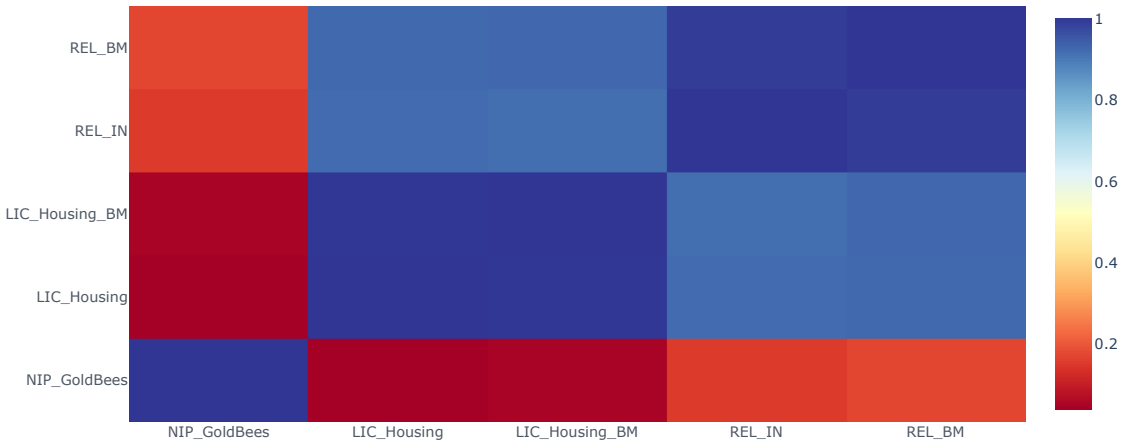
Let's create Cluster Map.

In [67]:
```
sns.clustermap(bank_stocks.xs(key='Close',axis=1,level='Stock Info').corr(),annot=True)
```

Out[67]:
```
<seaborn.matrix.ClusterGrid at 0x24f4887a448>
```



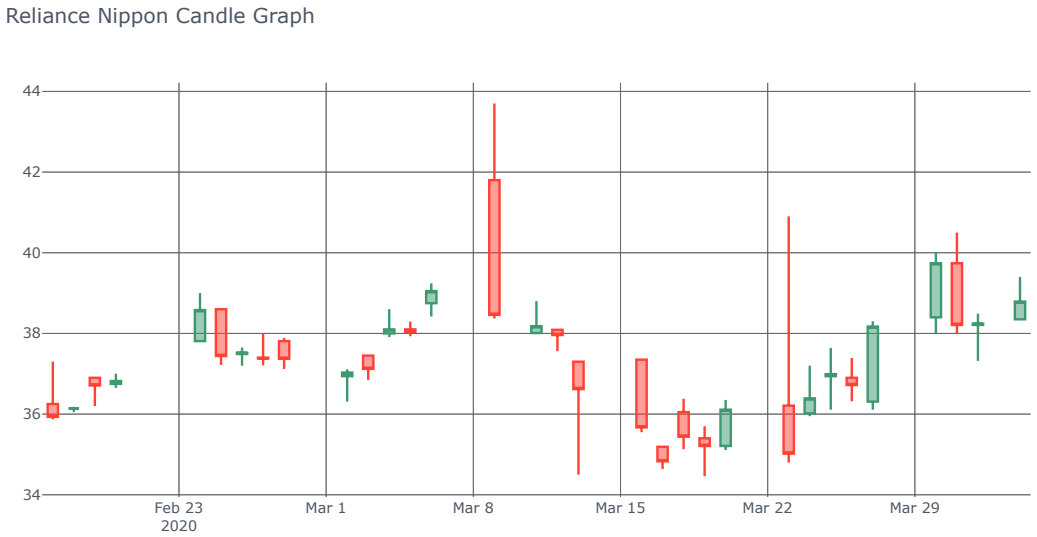
In [68]:
```
close_corr = bank_stocks.xs(key='Close',axis=1,level='Stock Info').corr()
close_corr.iplot(kind='heatmap',colorscale='rdylbu')
```



Export to plot.ly »

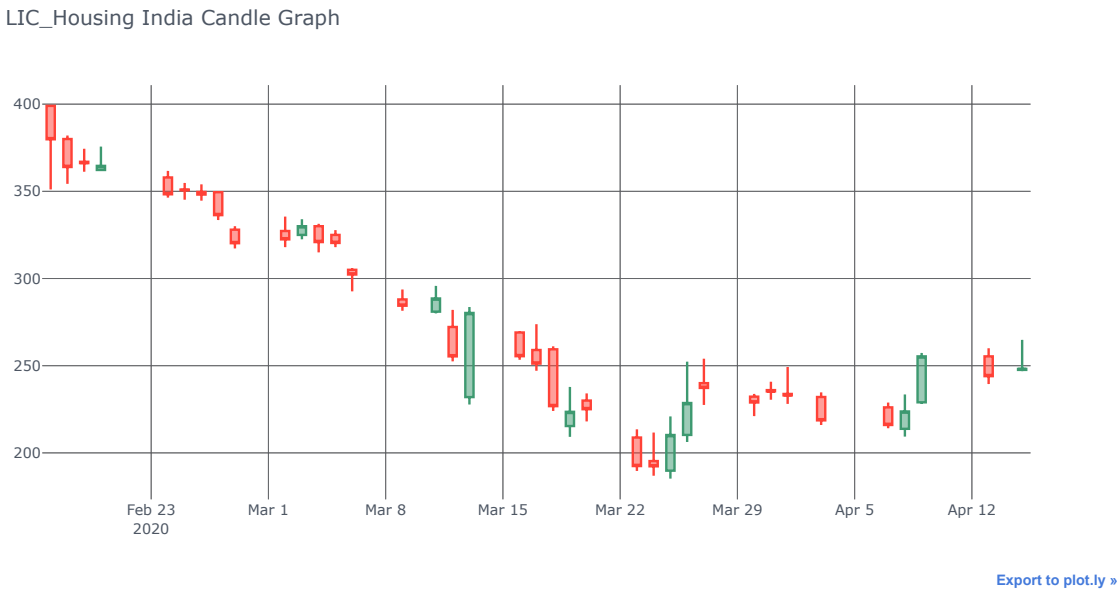
Let's do something interesting with Candle plot.

In [72]:
```
# 'NIP_GoldBees','LIC_Housing','LIC_Housing_BM','REL_IN','REL_BM'
NIP_GoldBees[['Open', 'High', 'Low', 'Close']]['2020-02-15':'2020-04-15'].iplot(kind='candle',title='Reliance Nippon Candle Graph')
```



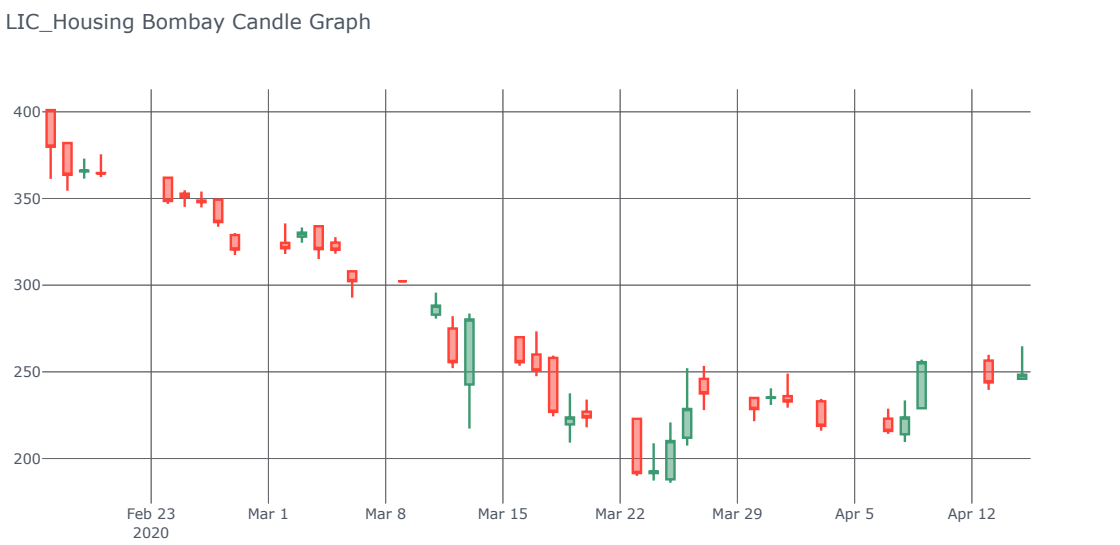
Export to plot.ly »

In [73]:
```
LIC_Housing[['Open', 'High', 'Low', 'Close']]['2020-02-15':'2020-04-15'].iplot(kind='candle',title='LIC_Housing India Candle Graph')
```



Export to plot.ly »

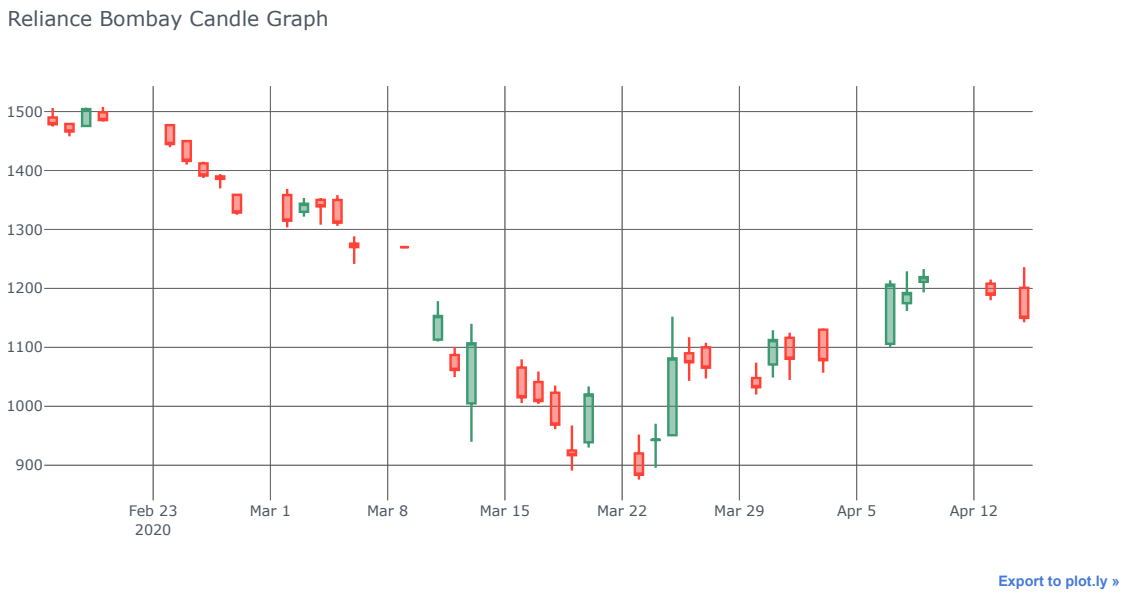
In [74]:
```
LIC_Housing_BM[['Open', 'High', 'Low', 'Close']]['2020-02-15':'2020-04-15'].iplot(kind='candle',title='LIC_Housing Bombay Candle Graph')
```



Export to plot.ly »

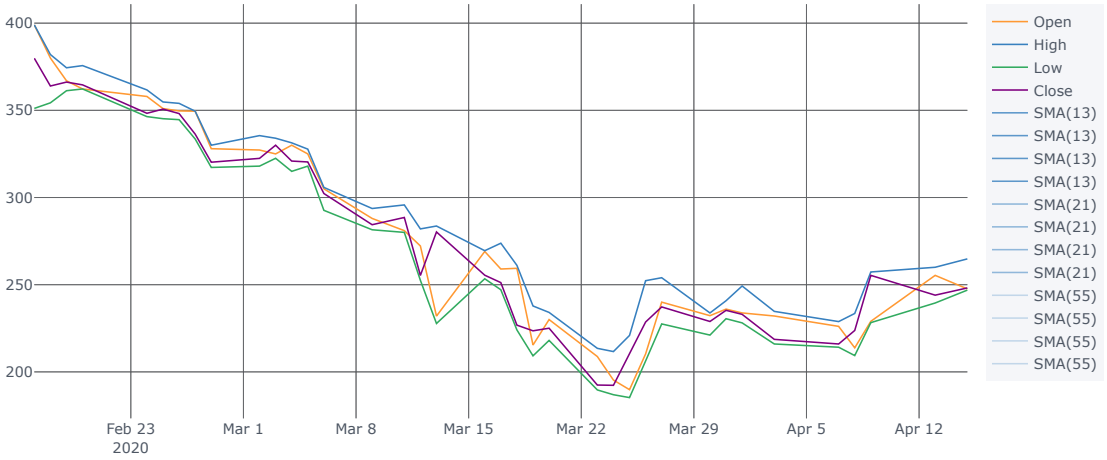
In [75]:
```
REL_IN[['Open', 'High', 'Low', 'Close']]['2020-02-15':'2020-04-15'].iplot(kind='candle',title='Reliance India Candle Graph')
```

In [76]:
```
REL_BM[['Open', 'High', 'Low', 'Close']]['2020-02-15':'2020-04-15'].iplot(kind='candle',title='Reliance Bombay Candle Graph')
```



Let's do some technical analysis on them by create a Simple Moving Averages plot of LIC Housing for the year 2020.

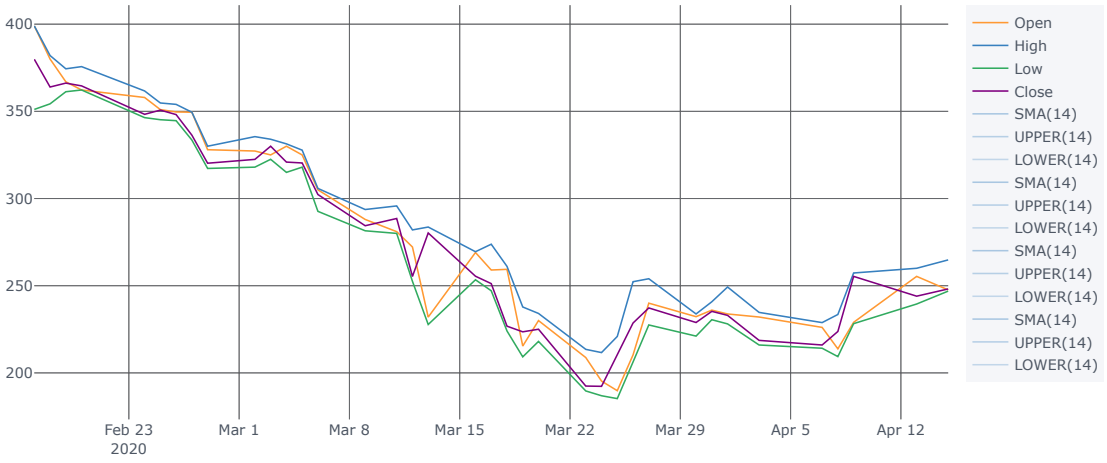
In [80]:
```
LIC_Housing[['Open', 'High', 'Low', 'Close']]['2020-02-15':'2020-04-15'].ta_plot(study='sma',periods=[13,21,55],title="Simple Moving Averages")
```



Export to plot.ly »

Let's do some more technical analysis on them by create a Bollinger Band Plot of LIC Housing for the year 2020.

In [81]:
```
LIC_Housing[['Open', 'High', 'Low', 'Close']]['2020-02-15':'2020-04-15'].ta_plot(study='boll')
```



Export to plot.ly »

Notice : In this data project I'll only focus on exploratory data analysis of stock prices. Keep in mind, this project is just meant to practice visualization and python pandas skills, it is not meant to be a robust financial analysis or be taken as financial advice.