Context

- Google Stock Price

Content of Dataset

- Historical stock prices (last 5 years) for all companies currently found on the S&P 500 index.
- All the files have the following columns:
  - Date - in format: yy-mm-dd
  - Open - price of the stock at market open (this is NYSE data so all in USD)
  - High - Highest price reached in the day
  - Low Close - Lowest price reached in the day
  - Volume - Number of shares traded
  - Name - the stock's ticker name

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

import plotly.graph_objs as go
import plotly.offline as offline

import warnings
warnings.filterwarnings('ignore')
```

```
Using TensorFlow backend.
```

```
data = pd.read_csv("/content/cs-1.csv")
data.head()
```

	date	open	high	low	close	volume	Name
0	2013-02-08	15.07	15.12	14.63	14.75	8407500	AAL
1	2013-02-11	14.89	15.01	14.26	14.46	8882000	AAL
2	2013-02-12	14.45	14.51	14.10	14.27	8126000	AAL
3	2013-02-13	14.30	14.94	14.25	14.66	10259500	AAL
4	2013-02-14	14.94	14.96	13.16	13.99	31879900	AAL

```
dataGOOG = data.loc[data['Name']  == 'GOOG']
dataGOOG
```

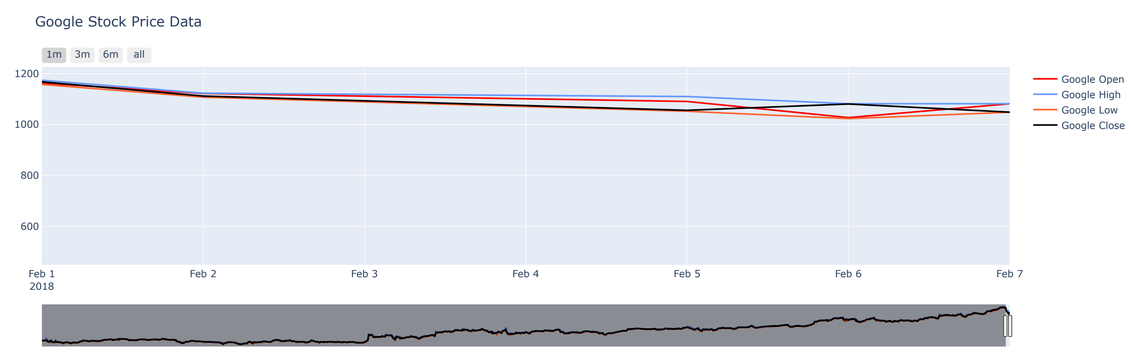
	date	open	high	low	close	volume	Name
251567	2014-03-27	568.000	568.00	552.92	558.46	13052	GOOG
251568	2014-03-28	561.200	566.43	558.67	559.99	41003	GOOG
251569	2014-03-31	566.890	567.00	556.93	556.97	10772	GOOG
251570	2014-04-01	558.710	568.45	558.71	567.16	7932	GOOG
251571	2014-04-02	565.106	604.83	562.19	567.00	146697	GOOG
...	...	...	...	...	...	...	...
252537	2018-02-01	1162.610	1174.00	1157.52	1167.70	2412114	GOOG
252538	2018-02-02	1122.000	1123.07	1107.28	1111.90	4857943	GOOG
252539	2018-02-05	1090.600	1110.00	1052.03	1055.80	3798301	GOOG
252540	2018-02-06	1027.180	1081.71	1023.14	1080.60	3447956	GOOG
252541	2018-02-07	1081.540	1081.78	1048.26	1048.58	2369232	GOOG

975 rows × 7 columns

```
print(dataGOOG.shape)
```

```
(975, 7)
```

```
trace_high = go.Scatter(x=dataGOOG.date,
                        y=dataGOOG.high,

                        name = "Google High",

                        line = dict(color = '#6699FF')
                       )
trace_low = go.Scatter( x=dataGOOG.date,
                        y=dataGOOG.low,

                        name = "Google Low",

                        line = dict(color = '#FF6633')
                       )
trace_open = go.Scatter( x=dataGOOG.date,
                        y=dataGOOG.open,

                        name = "Google Open",

                        line = dict(color = 'red')
                       )
trace_close = go.Scatter( x=dataGOOG.date,
                        y=dataGOOG.close,

                        name = "Google Close",

                        line = dict(color = 'black')
                       )
data_figure = [trace_open,trace_high, trace_low,trace_close]
layout = dict(

    title = 'Google Stock Price Data ',

    xaxis = dict(rangeselector = dict(buttons = list([dict(count = 1,
                                                           label = '1m',
                                                           step = 'month',
                                                           stepmode = 'todate',
                                                          visible = True),

                                                      dict(count = 3,
                                                           label = '3m',
                                                           step = 'month',
                                                           stepmode = 'backward',
                                                          visible = True),

                                                      dict(count = 6,
                                                           label = '6m',
                                                           step = 'month',
                                                           stepmode = 'backward',
                                                          visible = True),

                                                      dict(step = 'all')])
                                      ),

                rangeslider=dict(visible = True),
                type='date'
    )
)
fig = dict(data=data_figure,
           layout=layout)

offline.iplot(fig)
```



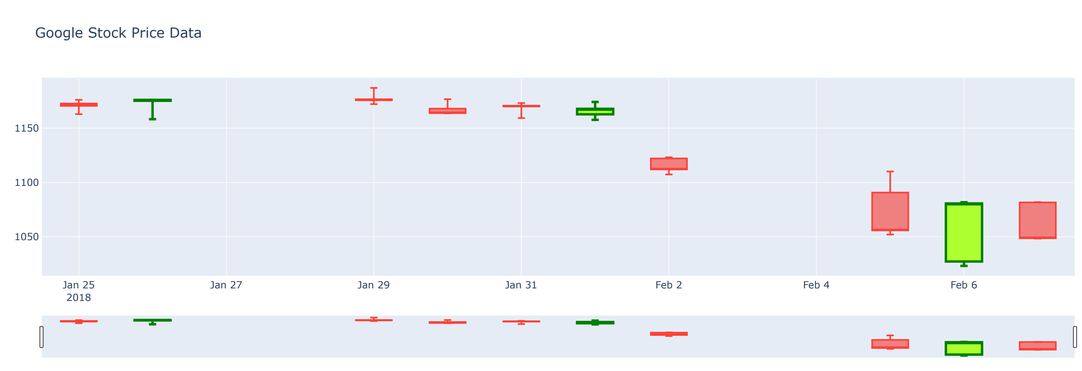
Plot the candlesticks

The plot has the following components:

- A bounding box whose y values represent the range between the stock's open and close prices
- A green box represents a higher close value than open (i.e. stock price went up that day)
- The box is red when the stock closed lower
- The vertical lines (bars above and below the box) show the range of intra-day high and low prices
- The vertical lines are capped at the top by horizontal lines called whiserks. By default, the width of the whisker is 0 which is why we don't see them

Notice that a slider appears by default along the X axis

```
data_temp = dataGOOG.iloc[965:975,:]
trace = go.Candlestick(x = data_temp.date,
                       open = data_temp.open,
                       high = data_temp.high,
                       low = data_temp.low,
                       close = data_temp.close,
                      increasing = dict(fillcolor = 'greenyellow',
                                        line = dict(color = 'green',
                                                    width = 3
                                                   )),
                       decreasing = dict(fillcolor = 'lightcoral'),
                       whiskerwidth = 0.2)
data_figure_2 = [trace]
layout = dict(title = 'Google Stock Price Data ')
fig = dict(data=data_figure_2,
           layout=layout)
offline.iplot(fig)
```



Open Price Animation

```
data_temp = dataGOOG.iloc[875:975,:]
data_open = list(data_temp['open'])
dateList = list(data_temp['date'])
xList = []
yList = []
framesList = []
for i in range(len(dateList)):

    xList.append(dateList[i])
    yList.append(data_open[i])

    framesList.append(dict(data = [dict(x = xList.copy(), y = yList.copy())]))
#
playButton = dict(label = 'Play',
                  method= 'animate',
                  args= [None,
                         dict(fromcurrent = True,
                              transition = dict(duration = 200),
                              frame = dict(duration = 100)
                             )
                        ]
                 )
#
pauseButton = dict(label = 'Pause',
                   method= 'animate',
                   args= [[None], dict(mode = 'immediate')]
                  )
#
layout = go.Layout(xaxis = dict(range = [dateList[0], dateList[-1]]),
                   yaxis = dict(range = [0, 1 + max(data_open)]),

                   updatemenus = [dict(type = 'buttons',
                                       buttons = [playButton, pauseButton]
                                       )
                                  ]
                   )
#
fig = dict(data=[{}],
           layout=layout,
           frames = framesList)

offline.iplot(fig)
```