Candlestick chart and Circle weighted graphs of stocks

In [1]:
```
# install package from jupyter notebook
# import sys
# !{sys.executable} -m pip install pandoc
```

In [2]:
```
import bokeh.plotting as bk
import pandas as pd
import numpy as np
import requests
from bokeh.models import Label, HoverTool, BoxZoomTool, PanTool, ZoomInTool, ZoomOutTool, ResetTool
from bokeh.io import output_notebook, show
from math import pi
from bokeh.plotting import figure, output_file #, show
# try color map feature
# from bokeh.palettes import Spectral6
# from bokeh.transform import linear_cmap
from bokeh.transform import transform
from bokeh.models import BasicTicker, ColorBar, ColumnDataSource, LinearColorMapper, PrintfTickFormatter

output_notebook()

API_URL = 'https://api.iextrading.com/1.0'
```

BokehJS 1.0.2 successfully loaded.

In [ ]:

In [3]:
```
stock_name = 'MSFT'
res = requests.get(f'{API_URL}/stock/{stock_name}/chart/5y')
df = pd.DataFrame(res.json())
df["date"] = pd.to_datetime(df["date"])
```

In [4]:
```
df.head()
```

Out[4]:

	change	changeOverTime	changePercent	close	date	high	label	low	open	unadjustedVolume	volume	vwap
0	0.487714	0.000000	1.517	32.6324	2013-12-20	32.7477	Dec 20, 13	32.0871	32.1004	62650324	62650324	32.4432
1	-0.159617	-0.004891	-0.489	32.4728	2013-12-23	32.7122	Dec 23, 13	32.4107	32.6413	25128740	25128740	32.4989
2	0.407907	0.007609	1.256	32.8807	2013-12-24	32.9605	Dec 24, 13	32.4905	32.5615	14242997	14242997	32.8248
3	0.319227	0.017391	0.971	33.1999	2013-12-26	33.2443	Dec 26, 13	32.9605	32.9871	17614984	17614984	33.1763
4	-0.133008	0.013315	-0.401	33.0669	2013-12-27	33.3596	Dec 27, 13	32.9605	33.3241	14563533	14563533	33.1304

In [5]:
```
seqs = np.arange(df.shape[0])
df['seqs'] = pd.Series(seqs)
df['changePercent'] = df['changePercent'].apply(lambda x: str(x)+'%')
df['mid'] = df.apply(lambda x: (x['open'] + x['close'])/2, axis = 1)
seqs
```

Out[5]:
```
array([   0,    1,    2, ..., 1255, 1256, 1257])
```

In [6]:
```
df.sample(5)
```

Out[6]:

	change	changeOverTime	changePercent	close	date	high	label	low	open	unadjustedVolume	volume	vwap	seqs	mid
291	-0.027525	0.223076	-0.069%	39.9119	2015-02-19	39.9361	Feb 19, 15	39.4990	39.6183	27603420	27603420	39.7690	291	39.765
640	0.873217	0.521200	1.791%	49.6404	2016-07-08	49.6973	Jul 8, 16	48.9285	49.0994	28391026	28391026	49.5381	640	49.369
1236	1.010000	2.318481	0.941%	108.2900	2018-11-16	108.8800	Nov 16, 18	106.8000	107.0800	33502121	33502121	107.9999	1236	107.685
1101	-0.404916	1.899633	-0.426%	94.6220	2018-05-08	94.9677	May 8, 18	93.8844	94.6572	23484589	23484589	94.4336	1101	94.639
503	0.655088	0.572419	1.293%	51.3118	2015-12-21	51.7984	Dec 21, 15	50.7467	51.3586	37246325	37246325	51.1669	503	51.335

In [7]:
```
df['height'] = df.apply(
    lambda x: x['close'] - x['open'] if x['close'] != x['open'] else 0.01,
    axis = 1)
df.head()
```

Out[7]:

	change	changeOverTime	changePercent	close	date	high	label	low	open	unadjustedVolume	volume	vwap	seqs	mid	heigh
0	0.487714	0.000000	1.517%	32.6324	2013-12-20	32.7477	Dec 20, 13	32.0871	32.1004	62650324	62650324	32.4432	0	32.36640	0.532
1	-0.159617	-0.004891	-0.489%	32.4728	2013-12-23	32.7122	Dec 23, 13	32.4107	32.6413	25128740	25128740	32.4989	1	32.55705	-0.168
2	0.407907	0.007609	1.256%	32.8807	2013-12-24	32.9605	Dec 24, 13	32.4905	32.5615	14242997	14242997	32.8248	2	32.72110	0.319
3	0.319227	0.017391	0.971%	33.1999	2013-12-26	33.2443	Dec 26, 13	32.9605	32.9871	17614984	17614984	33.1763	3	33.09350	0.212
4	-0.133008	0.013315	-0.401%	33.0669	2013-12-27	33.3596	Dec 27, 13	32.9605	33.3241	14563533	14563533	33.1304	4	33.19550	-0.257

In [8]:
```
inc = df.close > df.open
dec = df.close < df.open
w = .3
dec
```

```
1231     True
1232     True
1233     True
1234     True
1235    False
1236    False
1237     True
1238     True
1239     True
```

```
1240    False
1241    False
1242    False
1243    False
1244     True
1245    False
1246     True
1247     True
1248    False
1249     True
1250    False
```

In [9]:
```
sourceInc = bk.ColumnDataSource(df.loc[inc])
sourceDec = bk.ColumnDataSource(df.loc[dec])
```

In [ ]:
```
```

In [10]:
```
hover = HoverTool(
    tooltips=[
        ('Date', '@date'),
        ('Low', '@low'),
        ('High', '@high'),
        ('Open', '@open'),
        ('Close', '@close'),
        ('Percent', '@changePercent'),
    ]
)
```

In [11]:
```
TOOLS = [hover, BoxZoomTool(), PanTool(), ZoomInTool(), ZoomOutTool(), ResetTool()]
```

In [12]:
```
# set hight and width
p = bk.figure(plot_width = 1000, plot_height = 800, title = stock_name, tools=TOOLS, toolbar_location = 'above')
p.xaxis.major_label_orientation = np.pi/4

# set gird line width
p.grid.grid_line_alpha = w
descriptor = Label(x=70, y=70, text=f"5-year stock chart of {stock_name}")
p.add_layout(descriptor)
```

In [13]:
```
p.segment(df.seqs[inc],df.high[inc], df.seqs[inc], df.low[inc], color = 'green')

p.segment(df.seqs[dec],df.high[dec], df.seqs[dec], df.low[dec], color = 'red')
```

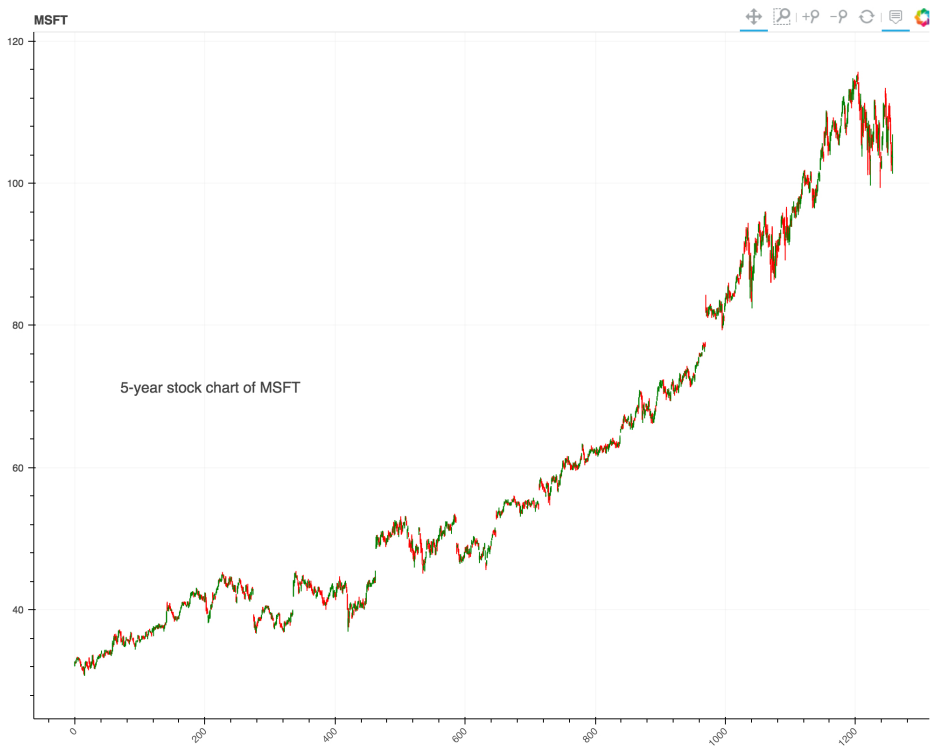
Out[13]: **GlyphRenderer**(id = '1049', ...)

In [14]:
```
p.rect(x='seqs', y='mid', width=w, height='height', fill_color='red', line_color='red', source=sourceDec)
p.rect(x='seqs', y='mid', width=w, height='height', fill_color='green', line_color='green', source=sourceInc)
```

Out[14]: **GlyphRenderer**(id = '1061', ...)

In [15]:
```
# p.segment?
```

In [16]:
```
bk.show(p)
```



In [17]:
```
# p.rect?
test = pd.DataFrame(np.random.rand(10,3))
test.columns = ['one','two','three']
test
```

Out[17]:

	one	two	three
0	0.327406	0.072095	0.485443
1	0.265271	0.905849	0.561913
2	0.774828	0.155890	0.450712
3	0.508982	0.204831	0.588177
4	0.191482	0.297027	0.596556
5	0.196370	0.797781	0.889716
6	0.564616	0.697607	0.842580

7	0.311767	0.820862	0.950460
8	0.797652	0.535998	0.976326
9	0.418908	0.790733	0.626770

Figure 2. Cooperate with volume and color map

The idea here is, the value of a company doesn't sole depend on stock price.
It will be useful to show how popular a company is by showing the volume of that company's stock at any given one day.

In [18]:
```
print(df.info())
df.head(2)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1258 entries, 0 to 1257
Data columns (total 15 columns):
change               1258 non-null float64
changeOverTime       1258 non-null float64
changePercent        1258 non-null object
close                1258 non-null float64
date                 1258 non-null datetime64[ns]
high                 1258 non-null float64
label                1258 non-null object
low                  1258 non-null float64
open                 1258 non-null float64
unadjustedVolume     1258 non-null int64
volume               1258 non-null int64
vwap                 1258 non-null float64
seqs                 1258 non-null int64
mid                  1258 non-null float64
height               1258 non-null float64
dtypes: datetime64[ns](1), float64(9), int64(3), object(2)
memory usage: 147.5+ KB
None
```

Out[18]:

	change	changeOverTime	changePercent	close	date	high	label	low	open	unadjustedVolume	volume	vwap	seqs	mid	height
0	0.487714	0.000000	1.517%	32.6324	2013-12-20	32.7477	Dec 20, 13	32.0871	32.1004	62650324	62650324	32.4432	0	32.36640	0.5320
1	-0.159617	-0.004891	-0.489%	32.4728	2013-12-23	32.7122	Dec 23, 13	32.4107	32.6413	25128740	25128740	32.4989	1	32.55705	-0.1685

In [19]:
```
df['date'] = pd.to_datetime(df['date'])
df['marketcap'] = df['volume'] * df['vwap']
```

In [20]:
```
df.head(2)
```

Out[20]:

	change	changeOverTime	changePercent	close	date	high	label	low	open	unadjustedVolume	volume	vwap	seqs	mid	height
0	0.487714	0.000000	1.517%	32.6324	2013-12-20	32.7477	Dec 20, 13	32.0871	32.1004	62650324	62650324	32.4432	0	32.36640	0.5320
1	-0.159617	-0.004891	-0.489%	32.4728	2013-12-23	32.7122	Dec 23, 13	32.4107	32.6413	25128740	25128740	32.4989	1	32.55705	-0.1685

In [ ]:

In [ ]:

In [ ]:

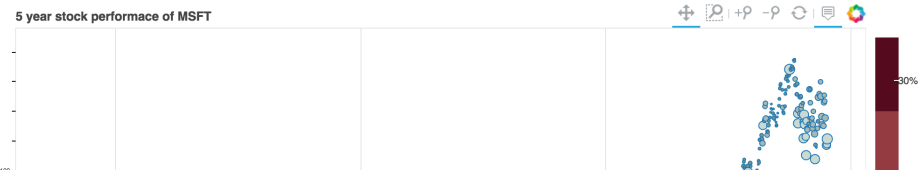
In [21]:
```
new_df = df.copy()
new_df = new_df[['date','vwap']]
new_df['adjVolume'] = 5*df['volume']//df['volume'].mean()
new_df.head()
```

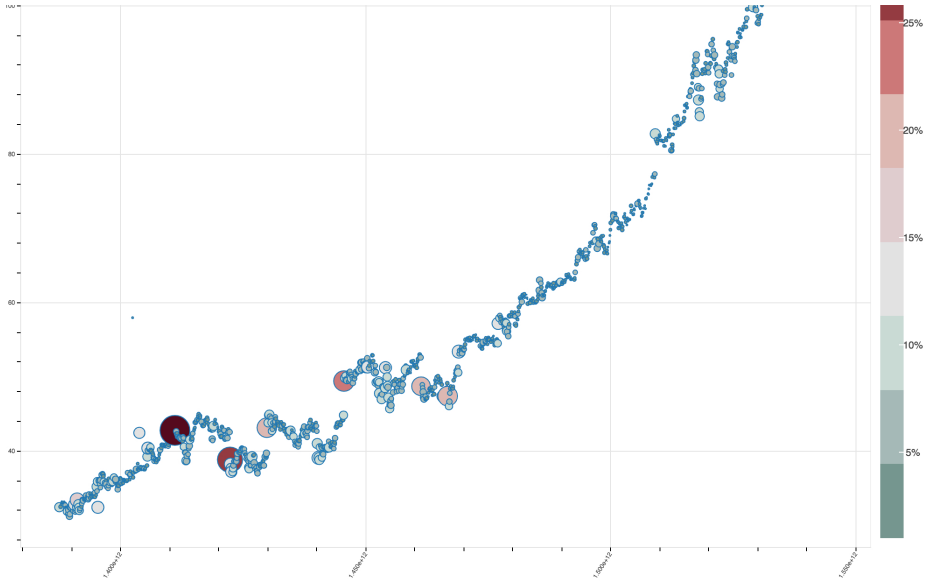
Out[21]:

	date	vwap	adjVolume
0	2013-12-20	32.4432	10.0
1	2013-12-23	32.4989	4.0
2	2013-12-24	32.8248	2.0
3	2013-12-26	33.1763	2.0
4	2013-12-27	33.1304	2.0

In [ ]:

In [22]:
```
# works
output_file('MSFT_Volumn_test.html')

source = ColumnDataSource(new_df)
colors = ["#75968f", "#a5bab7", "#c9d9d3", "#e2e2e2", "#dfccce", "#ddb7b1", "#cc7878", "#933b41", "#550b1d"]
mapper = LinearColorMapper(palette=colors, low=new_df.adjVolume.min(), high=new_df.adjVolume.max())


p = figure(plot_width=1000, plot_height=800, title= f'5 year stock performace of {stock_name}',
           tools=TOOLS, toolbar_location = 'above')

# p.circle(x=new_df["date"], y=new_df["vwap"], source=source, size=new_df['adjVolume'], fill_color=transform('adjVolume
p.circle(x="date", y="vwap", source=source, size='adjVolume', fill_color=transform('adjVolume', mapper))
color_bar = ColorBar(color_mapper=mapper, location=(0, 0),
                     ticker=BasicTicker(desired_num_ticks=len(colors)),
                     formatter=PrintfTickFormatter(format="%d%%"))

p.add_layout(color_bar, 'right')

p.axis.axis_line_color = None
p.axis.major_tick_line_color = None
p.axis.major_label_text_font_size = "5pt"
p.axis.major_label_standoff = 0
p.xaxis.major_label_orientation = 1.0

show(p)
```





This circle weighted graph clearly shows us straightforwardly when are the big days when many transaction happened.

```
In [ ]:
```