

Here we will be using bokeh tool in analysing time-series plots. Bokeh is a data visualization tool. It offers concise , human readable syntax with good aesthetics and pleasant data representation. Matplotlib creates static graphics that are useful for quick and simple visualizations, or for creating publication quality images. Bokeh creates visualizations for display on the web (whether locally or embedded in a webpage) and most importantly, the visualizations are meant to be highly interactive. Matplotlib does not offer either of these features.

In [6]:

```
import pandas as pd
import numpy as np
from pandas import datetime
import bokeh
from bokeh.plotting import figure , output_file , show
from bokeh.io import show , output_notebook
from bokeh.palettes import Blues9
from bokeh.palettes import RdBu3
from bokeh.models import ColumnDataSource, CategoricalColorMapper, ContinuousColorMapper
from bokeh.palettes import Spectral11
```

In [8]:

```
stock_nifty = pd.read_csv("nifty_it_index.csv")
stock_tcs = pd.read_csv("tcs_stock.csv")
stock_infy = pd.read_csv("infy_stock.csv")
```

In [9]:

```
stock_nifty.head(10)
```

Out[9]:

	Date	Open	High	Low	Close	Volume	Turnover
0	2015-01-01	11214.80	11235.75	11166.35	11215.70	4246150	3.575100e+09
1	2015-01-02	11214.65	11399.10	11214.65	11372.10	10004862	9.645600e+09
2	2015-01-05	11369.35	11433.75	11186.95	11248.55	8858018	1.059000e+10
3	2015-01-06	11186.10	11186.10	10909.00	10959.90	12515739	1.364500e+10
4	2015-01-07	11013.20	11042.35	10889.55	10916.00	10976356	1.203440e+10
5	2015-01-08	11031.15	11058.15	10915.05	11018.15	12975117	1.485630e+10
6	2015-01-09	11058.05	11484.90	10932.20	11399.65	24812224	3.536420e+10
7	2015-01-12	11456.00	11565.85	11378.80	11543.65	16505074	1.712310e+10
8	2015-01-13	11545.25	11546.60	11437.95	11502.80	12511358	1.417510e+10
9	2015-01-14	11561.95	11631.55	11521.00	11614.30	12544558	1.438530e+10

In [10]:

```
stock_tcs.head(10)
```

Out[10]:

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP
0	2015-01-01	TCS	EQ	2558.25	2567.0	2567.00	2541.00	2550.00	2545.55	2548.51
1	2015-01-02	TCS	EQ	2545.55	2551.0	2590.95	2550.60	2588.40	2579.45	2568.19
2	2015-01-05	TCS	EQ	2579.45	2581.0	2599.90	2524.65	2538.10	2540.25	2563.94
3	2015-01-06	TCS	EQ	2540.25	2529.1	2529.10	2440.00	2450.05	2446.60	2466.90
4	2015-01-07	TCS	EQ	2446.60	2470.0	2479.15	2407.45	2426.90	2417.70	2433.96
5	2015-01-08	TCS	EQ	2417.70	2442.4	2449.00	2420.55	2446.00	2443.80	2434.81
6	2015-01-09	TCS	EQ	2443.80	2455.0	2519.90	2450.00	2510.00	2512.30	2490.01
7	2015-01-12	TCS	EQ	2512.30	2517.0	2528.00	2480.25	2527.95	2509.70	2497.82
8	2015-01-13	TCS	EQ	2509.70	2520.0	2530.40	2480.10	2498.00	2497.90	2509.40
9	2015-01-14	TCS	EQ	2497.90	2516.0	2531.80	2500.50	2525.05	2521.95	2517.54



In [11]:

```
stock_infy.head(10)
```

Out[11]:

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWA
0	2015-01-01	INFY	EQ	1972.55	1968.95	1982.00	1956.90	1971.00	1974.40	1971.3
1	2015-01-02	INFY	EQ	1974.40	1972.00	2019.05	1972.00	2017.95	2013.20	2003.2
2	2015-01-05	INFY	EQ	2013.20	2009.90	2030.00	1977.50	1996.00	1995.90	2004.5
3	2015-01-06	INFY	EQ	1995.90	1980.00	1985.00	1934.10	1965.10	1954.20	1954.8
4	2015-01-07	INFY	EQ	1954.20	1965.00	1974.75	1950.00	1966.05	1963.55	1962.5
5	2015-01-08	INFY	EQ	1963.55	1985.60	1997.00	1950.00	1979.25	1973.45	1972.7
6	2015-01-09	INFY	EQ	1973.45	1980.10	2109.00	1913.05	2075.30	2074.45	2037.6
7	2015-01-12	INFY	EQ	2074.45	2092.00	2119.20	2075.00	2112.95	2115.95	2099.4
8	2015-01-13	INFY	EQ	2115.95	2107.80	2107.80	2075.00	2092.00	2088.90	2089.4
9	2015-01-14	INFY	EQ	2088.90	2098.50	2133.00	2092.60	2129.00	2128.65	2110.8

In [66]:

```
## Plotting of above data frame using Bokeh
stock_nifty_sorted = stock_nifty.sort_index(axis = 0 , ascending = True)
nifty_open = list(stock_nifty_sorted['Open'])
nifty_close = list(stock_nifty_sorted['Close'])

date_list = list(stock_nifty_sorted.index)

date_time =[ date_list for d in date_list]
```

In [73]:

```
p = figure(x_axis_type = 'datetime' , plot_width = 800 , plot_height = 500 , title = 'Nifty Plot' , tools = "" , toolbar_location = None)

p.circle(date_time , nifty_open , legend = 'Open Price' , size = 6 , color = 'red' , alpha = 0.5)

p.line(date_time , nifty_open , legend = 'Open Price' , color = 'blue' , alpha= 0.5)

p.line(date_time , nifty_close , legend = 'Close Price' , color = 'blue' , alpha = 0.5)

p.circle(date_time , nifty_close , legend = 'Close Price' , size = 6 , color = 'red' , alpha = 0.5)

show(p)

#output_file("Time_Series_Nifty.html")
```

In [86]:

```
data = dict(stock_infy=stock_infy['Close'], Date=stock_infy.index)

p = figure(plot_width=800, plot_height=250, title = 'Time Series for Financial Data', x_axis_type="datetime")
p.line(stock_infy.index, stock_infy['Close'], color='blue', alpha=0.5)

show(p)
```

In [83]:

```
data = dict(stock_tcs = stock_tcs['Close'] , Date = stock_tcs.index)

p = figure(plot_width = 800 , plot_height = 250 , title = 'Time Series Visualization for TCS' , x_axis_type = "datetime")
p.line(stock_tcs.index , stock_tcs['Close'] , color = 'blue' , alpha = 0.5)

show(p)
```

In [81]:

```
data = dict(stock_nifty = stock_nifty['Close'] , Date = stock_nifty.index)

p = figure(plot_width = 800 , plot_height = 250 , title = 'Time Series Visualization for Nifty' , x_axis_type = "datetime")
p.line(stock_nifty.index , stock_nifty['Close'] , color = 'blue' , alpha = 0.5)

show(p)
```