Initialising dataset and import library

In [1]:

```
#import required packages
from datetime import date,timedelta
from nsepy import get_history
import numpy as np
import pandas as pd
from pandas import datetime
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
# collecting data of 1 year starting from 2015-01-01 from nsepy

number_of_days = 365
strt = date(2015,1,1)
ends = strt+timedelta(days=number_of_days)
SBIN = get_history(symbol='sbin',
                   start= strt,
                   end=ends)
TCS = get_history(symbol='TCS',
                   start= strt,
                   end=ends)
INFY = get_history(symbol='INFY',
                   start= strt,
                   end=ends)
```

In [3]:

```
SBIN.head()
```

Out[3]:

Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Tu
2015-01-01	SBIN	EQ	311.85	312.45	315.00	310.70	314.0	314.00	313.67	6138488	1.92548
2015-01-02	SBIN	EQ	314.00	314.35	318.30	314.35	315.6	315.25	316.80	9935094	3.14738
2015-01-05	SBIN	EQ	315.25	316.25	316.80	312.10	312.8	312.75	313.84	9136716	2.86743
2015-01-06	SBIN	EQ	312.75	310.00	311.10	298.70	299.9	299.90	305.14	15329257	4.67760
2015-01-07	SBIN	EQ	299.90	300.00	302.55	295.15	301.4	300.15	299.95	15046745	4.51324

In [4]:

```
TCS.head()
```

Out[4]:

Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume
2015-01-01	TCS	EQ	2558.25	2567.0	2567.00	2541.00	2550.00	2545.55	2548.51	183415
2015-01-02	TCS	EQ	2545.55	2551.0	2590.95	2550.60	2588.40	2579.45	2568.19	462870
2015-01-05	TCS	EQ	2579.45	2581.0	2599.90	2524.65	2538.10	2540.25	2563.94	877121
2015-01-06	TCS	EQ	2540.25	2529.1	2529.10	2440.00	2450.05	2446.60	2466.90	1211892
2015-01-07	TCS	EQ	2446.60	2470.0	2479.15	2407.45	2426.90	2417.70	2433.96	1318166

In [5]:

```
INFY.head()
```

Out[5]:

Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume
2015-01-01	INFY	EQ	1972.55	1968.95	1982.00	1956.9	1971.00	1974.40	1971.34	500691
2015-01-02	INFY	EQ	1974.40	1972.00	2019.05	1972.0	2017.95	2013.20	2003.25	1694580
2015-01-05	INFY	EQ	2013.20	2009.90	2030.00	1977.5	1996.00	1995.90	2004.59	2484256
2015-01-06	INFY	EQ	1995.90	1980.00	1985.00	1934.1	1965.10	1954.20	1954.82	2416829
2015-01-07	INFY	EQ	1954.20	1965.00	1974.75	1950.0	1966.05	1963.55	1962.59	1812479

In [6]:

```
# resetting index of each data set
SBIN = SBIN.reset_index()
TCS = TCS.reset_index()
INFY = INFY.reset_index()

# typecasting date

SBIN["Date"] = pd.to_datetime(SBIN["Date"])
TCS["Date"] = pd.to_datetime(TCS["Date"])
INFY["Date"] = pd.to_datetime(INFY["Date"])

# naming each dataset

SBIN.name = 'SBIN'
TCS.name = 'TCS'
INFY.name = 'INFY'

# an array for future use
```

In [7]:

```
def assign_index(stock):
    stock.index = stock['Date']
    return stock
```

In [8]:

```
# assigning index

SBIN = assign_index(SBIN)
TCS = assign_index(TCS)
INFY = assign_index(INFY)
```

In [9]:

```
stocks = [SBIN,TCS,INFY]
```

In [50]:

```
#importing plot library

import matplotlib.pyplot as plt
%matplotlib inline
# Control the default size of figures in this Jupyter notebook
%pylab inline
pylab.rcParams['figure.figsize'] = (20, 12)
```

```
Populating the interactive namespace from numpy and matplotlib
```

Part -1

(Target variable will be the Close(closing price of share))

1. Create 4,16,....,52 week moving average(closing price) for each stock and index. This should happen through a function.)

In [51]:

```
#Moving average implementation

def moving_average(values,size):
    weights = np.repeat(1.0, size)/size
    smas = np.convolve(values,weights,'valid')
    print(type(smas))
    return smas
```

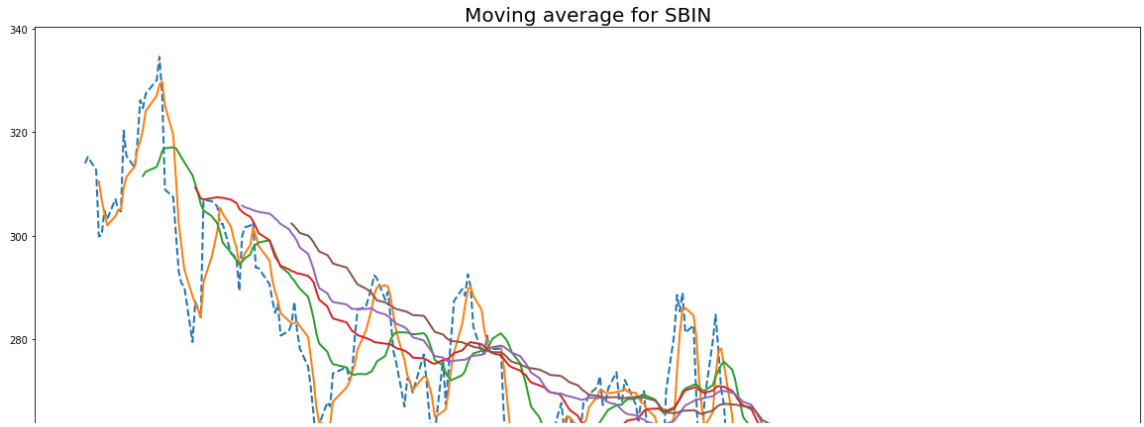
In [52]:

```
# function to calculate moving average of a stock and make a line chart of it.

def moving_average_PLOT(stock):
    # weeks size
    size_arr = [4,16,28,40,52]

    moving_avg = {}

    # Line chart for each graph
    plt.title("Moving average for "+stock.name,fontsize=20)
    # Original closing price as -- line
    plt.plot(stock["Date"],stock["Close"],label="Closing pricing",linestyle='--',li

    for i in range(len(size_arr)):
        # dummy size array 'a' to resize the frame with original size
        a = [None for i in range(size_arr[i]-1)]
        a = np.array(a)
        # merging both the array dummy and moving_average
        moving_avg[size_arr[i]] = np.hstack([a,moving_average(stock["Close"],size_a

        stock[str(size_arr[i])+"_moving_avg"] = moving_avg[size_arr[i]]

        name = "Moving average for "+str(size_arr[i])+" weeks"
        print(name + "is as follow :")
        print(stock[str(size_arr[i])+"_moving_avg"])
        plt.plot(stock["Date"],moving_avg[size_arr[i]],label = name,linewidth=2)

    plt.legend(title = "Legends",loc = 3,prop={'size': 12})
    plt.show()
```

In [53]:

```
moving_average_PLOT(SBIN)
```

```
2015-12-24    240.721
2015-12-28    240.441
2015-12-29    240.127
2015-12-30    239.781
2015-12-31      239.4
2016-01-01    238.993
Name: 52_moving_avg, Length: 249, dtype: object
```



In [54]:

```
moving_average_PLOT(TCS)
```

```
2015-12-08    2331.99
2015-12-09    2336.21
2015-12-10    2350.25
2015-12-11     2366.8
2015-12-14    2379.16
2015-12-15    2381.21
2015-12-16    2387.17
2015-12-17    2400.61
2015-12-18     2410.2
2015-12-21    2427.84
2015-12-22       2427
2015-12-23    2423.51
2015-12-24    2427.88
2015-12-28    2431.95
2015-12-29    2444.64
2015-12-30    2442.76
2015-12-31    2443.41
2016-01-01    2431.84
Name: 4_moving_avg, Length: 249, dtype: object
```

In [55]:

```
moving_average_PLOT(INFY)
```

```
<class 'numpy.ndarray'>
Moving average for 4 weeksis as follow :
Date
2015-01-01       None
2015-01-02       None
2015-01-05       None
2015-01-06    1984.42
2015-01-07    1981.71
2015-01-08    1971.78
2015-01-09    1991.41
2015-01-12    2031.85
2015-01-13    2063.19
2015-01-14    2101.99
2015-01-15    2116.94
2015-01-16    2117.53
2015-01-19    2120.93
2015-01-20    2119.81
2015-01-21    2128.72
2015-01-22    2148.26
2015-01-23     2176.4
```

2.Create rolling window of size 10 on each stock/index. Handle unequal time series due to stock market holidays. You should look to increase your rolling window size to 75 and see how the data looks like. Remember they will create stress on your laptop RAM load.

I have taken some more rolling size so as to make it more understanding

rolling_size = ["10","25","50","75"]

In [56]:

```
# rolling window function using inbuilt function

def rolling_window(stock):
    plt.title("Moving average with inbuilt function on" + stock.name,fontsize = 15)
    plt.plot(stock["Close"],label="Original closing Price")
    rolling_size = ["10","25","50","75"]
    for i in range(len(rolling_size)):
        temp_name = str(rolling_size[i])+" rolling window"
        stock[temp_name] = np.round(stock["Close"].rolling(window = int(rolling_siz
        text = "Rolling window of size : "+rolling_size[i]
        print(text)
        print(stock[temp_name])
        plt.plot(stock[temp_name],label=text)
    plt.legend(title = "Legends",loc = 3,prop={'size': 12})
    plt.show()
```

In [57]:

```
rolling_window(SBIN)
# starting values are NaN as rolling window is calculating 'valid' average.
```

```
Rolling window of size : 10
Date
2015-01-01       NaN
2015-01-02       NaN
2015-01-05       NaN
2015-01-06       NaN
2015-01-07       NaN
2015-01-08       NaN
2015-01-09       NaN
2015-01-12       NaN
2015-01-13       NaN
2015-01-14    306.70
2015-01-15    307.33
2015-01-16    307.35
2015-01-19    307.39
2015-01-20    309.21
2015-01-21    311.82
2015-01-22    313.80
2015-01-23    316.22
2015-01-27    318.52
```

In [58]:

```
rolling_window(TCS)
```

```
Rolling window of size : 10
Date
2015-01-01        NaN
2015-01-02        NaN
2015-01-05        NaN
2015-01-06        NaN
2015-01-07        NaN
2015-01-08        NaN
2015-01-09        NaN
2015-01-12        NaN
2015-01-13        NaN
2015-01-14    2501.52
2015-01-15    2500.88
2015-01-16    2496.14
2015-01-19    2493.22
2015-01-20    2498.61
2015-01-21    2508.22
2015-01-22    2515.19
2015-01-23    2514.32
2015-01-27    2513.55
```

In [59]:

```
rolling_window(INFY)
```

```
Rolling window of size : 10
Date
2015-01-01        NaN
2015-01-02        NaN
2015-01-05        NaN
2015-01-06        NaN
2015-01-07        NaN
2015-01-08        NaN
2015-01-09        NaN
2015-01-12        NaN
2015-01-13        NaN
2015-01-14    2028.27
2015-01-15    2044.25
2015-01-16    2054.76
2015-01-19    2065.42
2015-01-20    2082.42
2015-01-21    2103.06
2015-01-22    2125.36
2015-01-23    2139.42
2015-01-27    2141.47
```

3.1 Volume shocks

0/1 boolean time series for shock

In [60]:

```
# making a extra column as we need to compare with previous day's volume
SBIN["prev_day"] = SBIN.Volume.shift(1)
TCS["prev_day"] = TCS.Volume.shift(1)
INFY["prev_day"] = INFY.Volume.shift(1)
```

In [61]:

```
# Calculating volume shock
SBIN["Volume_Shock"] = ((((abs(SBIN["prev_day"]-SBIN["Volume"]))/SBIN["Volume"])*10
print(SBIN["Volume_Shock"])
```

```
Date
2015-01-01    0
2015-01-02    1
2015-01-05    0
2015-01-06    1
2015-01-07    0
2015-01-08    1
2015-01-09    1
2015-01-12    1
2015-01-13    1
2015-01-14    0
2015-01-15    1
2015-01-16    1
2015-01-19    1
2015-01-20    1
2015-01-21    1
2015-01-22    1
2015-01-23    1
2015-01-27    1
2015-01-28    1
2015-01-29    1
2015-01-30    1
2015-02-02    1
2015-02-03    1
2015-02-04    1
2015-02-05    1
2015-02-06    0
2015-02-09    0
2015-02-10    1
2015-02-11    1
2015-02-12    1
              ..
2015-11-19    1
2015-11-20    1
2015-11-23    1
2015-11-24    1
2015-11-26    1
2015-11-27    1
2015-11-30    1
2015-12-01    1
2015-12-02    1
2015-12-03    1
2015-12-04    1
2015-12-07    1
2015-12-08    0
2015-12-09    1
2015-12-10    1
2015-12-11    0
2015-12-14    0
2015-12-15    1
2015-12-16    1
2015-12-17    1
2015-12-18    1
2015-12-21    1
2015-12-22    0
```

```
2015-12-23    1
2015-12-24    0
2015-12-28    1
2015-12-29    1
2015-12-30    1
2015-12-31    0
2016-01-01    1
Name: Volume_Shock, Length: 249, dtype: int64
```

In [62]:

```
TCS["Volume_Shock"] = ((((abs(TCS["prev_day"]-TCS["Volume"]))/TCS["Volume"])*100)>1
print(TCS["Volume_Shock"])
```

```
Date
2015-01-01    0
2015-01-02    1
2015-01-05    1
2015-01-06    1
2015-01-07    0
2015-01-08    1
2015-01-09    1
2015-01-12    1
2015-01-13    0
2015-01-14    1
2015-01-15    1
2015-01-16    0
2015-01-19    1
2015-01-20    1
2015-01-21    1
2015-01-22    0
2015-01-23    1
2015-01-27    1
2015-01-28    1
2015-01-29    1
2015-01-30    1
2015-02-02    1
2015-02-03    1
2015-02-04    0
2015-02-05    1
2015-02-06    1
2015-02-09    0
2015-02-10    1
2015-02-11    1
2015-02-12    0
              ..
2015-11-19    1
2015-11-20    0
2015-11-23    1
2015-11-24    0
2015-11-26    1
2015-11-27    1
2015-11-30    1
2015-12-01    1
2015-12-02    0
2015-12-03    1
2015-12-04    1
2015-12-07    0
2015-12-08    1
2015-12-09    1
2015-12-10    1
2015-12-11    0
2015-12-14    1
2015-12-15    1
2015-12-16    1
2015-12-17    1
2015-12-18    1
2015-12-21    1
2015-12-22    0
2015-12-23    1
```

```
2015-12-24    0
2015-12-28    1
2015-12-29    1
2015-12-30    0
2015-12-31    1
2016-01-01    1
Name: Volume_Shock, Length: 249, dtype: int64
```

In [63]:

```
INFY["Volume_Shock"] = ((((abs(INFY["prev_day"]-INFY["Volume"]))/INFY["Volume"])*10
print(INFY["Volume_Shock"])
```

```
Date
2015-01-01    0
2015-01-02    1
2015-01-05    1
2015-01-06    0
2015-01-07    1
2015-01-08    1
2015-01-09    1
2015-01-12    1
2015-01-13    1
2015-01-14    1
2015-01-15    1
2015-01-16    1
2015-01-19    1
2015-01-20    1
2015-01-21    1
2015-01-22    1
2015-01-23    0
2015-01-27    1
2015-01-28    0
2015-01-29    1
2015-01-30    1
2015-02-02    1
2015-02-03    1
2015-02-04    0
2015-02-05    1
2015-02-06    1
2015-02-09    1
2015-02-10    1
2015-02-11    1
2015-02-12    0
              ..
2015-11-19    1
2015-11-20    1
2015-11-23    0
2015-11-24    1
2015-11-26    1
2015-11-27    1
2015-11-30    1
2015-12-01    1
2015-12-02    0
2015-12-03    1
2015-12-04    1
2015-12-07    1
2015-12-08    1
2015-12-09    1
2015-12-10    1
2015-12-11    0
2015-12-14    1
2015-12-15    1
2015-12-16    0
2015-12-17    1
2015-12-18    1
2015-12-21    1
2015-12-22    0
2015-12-23    1
```

```
2015-12-24    1
2015-12-28    1
2015-12-29    1
2015-12-30    1
2015-12-31    1
2016-01-01    1
Name: Volume_Shock, Length: 249, dtype: int64
```

0/1 dummy-coded time series for direction of shock

In [64]:

```
# Calculating direction volume shock for each share
def direction_shock(stock_name):
    if(stock_name["Volume_Shock"]==1):
        if(stock_name["Volume"]-stock_name["prev_day"]>0):
            return 1
        else:
            return 0
    else:
        return "NaN"
```

In [65]:

```
# putting NaN where volume shock is 0
SBIN["dir_shock"] = 'NaN'
SBIN["dir_shock"] = SBIN.apply(direction_shock,axis=1)
print(SBIN["dir_shock"])
```

```
Date
2015-01-01    NaN
2015-01-02      1
2015-01-05    NaN
2015-01-06      1
2015-01-07    NaN
2015-01-08      0
2015-01-09      1
2015-01-12      0
2015-01-13      1
2015-01-14    NaN
2015-01-15      1
2015-01-16      0
2015-01-19      0
2015-01-20      1
2015-01-21      1
2015-01-22      0
2015-01-23      1
2015-01-27      0
2015-01-28      1
2015-01-29      0
2015-01-30      1
2015-02-02      0
2015-02-03      1
2015-02-04      0
2015-02-05      0
2015-02-06    NaN
2015-02-09    NaN
2015-02-10      1
2015-02-11      0
2015-02-12      1
              ...
2015-11-19      0
2015-11-20      1
2015-11-23      0
2015-11-24      0
2015-11-26      1
2015-11-27      1
2015-11-30      0
2015-12-01      0
2015-12-02      1
2015-12-03      0
2015-12-04      1
2015-12-07      0
2015-12-08    NaN
2015-12-09      0
2015-12-10      1
2015-12-11    NaN
2015-12-14    NaN
2015-12-15      0
2015-12-16      1
2015-12-17      1
2015-12-18      1
2015-12-21      0
```

```
2015-12-22    NaN
2015-12-23      0
2015-12-24    NaN
2015-12-28      1
2015-12-29      0
2015-12-30      1
2015-12-31    NaN
2016-01-01      0
Name: dir_shock, Length: 249, dtype: object
```

In [66]:

```
# putting NaN where volume shock is 0
TCS["dir_shock"] = 'NaN'
TCS["dir_shock"] = TCS.apply(direction_shock,axis=1)
print(TCS["dir_shock"])
```

```
Date
2015-01-01    NaN
2015-01-02      1
2015-01-05      1
2015-01-06      1
2015-01-07    NaN
2015-01-08      0
2015-01-09      1
2015-01-12      0
2015-01-13    NaN
2015-01-14      1
2015-01-15      1
2015-01-16    NaN
2015-01-19      0
2015-01-20      1
2015-01-21      1
2015-01-22    NaN
2015-01-23      1
2015-01-27      0
2015-01-28      1
2015-01-29      1
2015-01-30      1
2015-02-02      0
2015-02-03      0
2015-02-04    NaN
2015-02-05      1
2015-02-06      0
2015-02-09    NaN
2015-02-10      1
2015-02-11      0
2015-02-12    NaN
              ...
2015-11-19      1
2015-11-20    NaN
2015-11-23      0
2015-11-24    NaN
2015-11-26      1
2015-11-27      0
2015-11-30      1
2015-12-01      0
2015-12-02    NaN
2015-12-03      1
2015-12-04      0
2015-12-07    NaN
2015-12-08      0
2015-12-09      1
2015-12-10      0
2015-12-11    NaN
2015-12-14      1
2015-12-15      0
2015-12-16      1
2015-12-17      1
2015-12-18      1
2015-12-21      0
```

```
2015-12-22    NaN
2015-12-23      0
2015-12-24    NaN
2015-12-28      1
2015-12-29      0
2015-12-30    NaN
2015-12-31      0
2016-01-01      0
Name: dir_shock, Length: 249, dtype: object
```

In [67]:

```
# putting NaN where volume shock is 0
INFY["dir_shock"] = 'NaN'
INFY["dir_shock"] = INFY.apply(direction_shock,axis=1)
print(INFY["dir_shock"])
```

```
Date
2015-01-01    NaN
2015-01-02      1
2015-01-05      1
2015-01-06    NaN
2015-01-07      0
2015-01-08      1
2015-01-09      1
2015-01-12      0
2015-01-13      0
2015-01-14      1
2015-01-15      0
2015-01-16      0
2015-01-19      0
2015-01-20      1
2015-01-21      1
2015-01-22      1
2015-01-23    NaN
2015-01-27      1
2015-01-28    NaN
2015-01-29      1
2015-01-30      0
2015-02-02      0
2015-02-03      0
2015-02-04    NaN
2015-02-05      1
2015-02-06      0
2015-02-09      0
2015-02-10      1
2015-02-11      0
2015-02-12    NaN
              ...
2015-11-19      0
2015-11-20      0
2015-11-23    NaN
2015-11-24      0
2015-11-26      1
2015-11-27      0
2015-11-30      1
2015-12-01      0
2015-12-02    NaN
2015-12-03      0
2015-12-04      1
2015-12-07      0
2015-12-08      1
2015-12-09      1
2015-12-10      0
2015-12-11    NaN
2015-12-14      0
2015-12-15      1
2015-12-16    NaN
2015-12-17      0
2015-12-18      1
2015-12-21      0
```

```
2015-12-22    NaN
2015-12-23      0
2015-12-24      0
2015-12-28      1
2015-12-29      0
2015-12-30      1
2015-12-31      1
2016-01-01      0
Name: dir_shock, Length: 249, dtype: object
```

3.1 Price shocks and Price black swan(both are same)

0/1 boolean time series for shock

In [68]:

```
#extra column for previous day closing price
SBIN["prev_day_close"] = SBIN.Close.shift(-1)
TCS["prev_day_close"] = TCS.Close.shift(-1)
INFY["prev_day_close"] = INFY.Close.shift(-1)
```

In [69]:

```
SBIN["Close_price_shock"] = ((((abs(SBIN["prev_day_close"]-SBIN["Close"]))/SBIN["Cl
print(SBIN["Close_price_shock"])
```

```
Date
2015-01-01    0
2015-01-02    0
2015-01-05    1
2015-01-06    0
2015-01-07    0
2015-01-08    0
2015-01-09    0
2015-01-12    0
2015-01-13    0
2015-01-14    1
2015-01-15    0
2015-01-16    0
2015-01-19    0
2015-01-20    1
2015-01-21    0
2015-01-22    0
2015-01-23    0
2015-01-27    0
2015-01-28    1
2015-01-29    1
2015-01-30    0
2015-02-02    1
2015-02-03    1
2015-02-04    0
2015-02-05    0
2015-02-06    1
2015-02-09    1
2015-02-10    0
2015-02-11    0
2015-02-12    1
             ..
2015-11-19    0
2015-11-20    0
2015-11-23    0
2015-11-24    0
2015-11-26    1
2015-11-27    0
2015-11-30    0
2015-12-01    1
2015-12-02    0
2015-12-03    0
2015-12-04    0
2015-12-07    0
2015-12-08    0
2015-12-09    0
2015-12-10    1
2015-12-11    0
2015-12-14    0
2015-12-15    0
2015-12-16    0
2015-12-17    0
2015-12-18    0
2015-12-21    0
2015-12-22    0
2015-12-23    0
```

```
2015-12-24    0
2015-12-28    0
2015-12-29    0
2015-12-30    0
2015-12-31    0
2016-01-01    0
Name: Close_price_shock, Length: 249, dtype: int64
```

In [70]:

```
TCS["Close_price_shock"] = ((((abs(TCS["prev_day_close"]-TCS["Close"]))/TCS["Close"
print(TCS["Close_price_shock"])
```

```
Date
2015-01-01    0
2015-01-02    0
2015-01-05    1
2015-01-06    0
2015-01-07    0
2015-01-08    1
2015-01-09    0
2015-01-12    0
2015-01-13    0
2015-01-14    0
2015-01-15    0
2015-01-16    0
2015-01-19    0
2015-01-20    0
2015-01-21    0
2015-01-22    0
2015-01-23    0
2015-01-27    0
2015-01-28    0
2015-01-29    1
2015-01-30    0
2015-02-02    0
2015-02-03    0
2015-02-04    0
2015-02-05    0
2015-02-06    1
2015-02-09    1
2015-02-10    0
2015-02-11    0
2015-02-12    1
             ..
2015-11-19    0
2015-11-20    0
2015-11-23    0
2015-11-24    0
2015-11-26    0
2015-11-27    0
2015-11-30    0
2015-12-01    0
2015-12-02    0
2015-12-03    0
2015-12-04    0
2015-12-07    0
2015-12-08    0
2015-12-09    0
2015-12-10    0
2015-12-11    0
2015-12-14    0
2015-12-15    0
2015-12-16    0
2015-12-17    0
2015-12-18    0
2015-12-21    0
2015-12-22    0
2015-12-23    0
```

```
2015-12-24    0
2015-12-28    0
2015-12-29    0
2015-12-30    0
2015-12-31    0
2016-01-01    0
Name: Close_price_shock, Length: 249, dtype: int64
```

In [71]:

```
INFY["Close_price_shock"] = ((((abs(INFY["prev_day_close"]-INFY["Close"]))/INFY["Cl
print(INFY["Close_price_shock"])
```

```
Date
2015-01-01    0
2015-01-02    0
2015-01-05    1
2015-01-06    0
2015-01-07    0
2015-01-08    1
2015-01-09    1
2015-01-12    0
2015-01-13    0
2015-01-14    0
2015-01-15    0
2015-01-16    0
2015-01-19    0
2015-01-20    1
2015-01-21    0
2015-01-22    0
2015-01-23    1
2015-01-27    0
2015-01-28    0
2015-01-29    0
2015-01-30    0
2015-02-02    0
2015-02-03    0
2015-02-04    1
2015-02-05    0
2015-02-06    0
2015-02-09    0
2015-02-10    0
2015-02-11    0
2015-02-12    0
             ..
2015-11-19    0
2015-11-20    0
2015-11-23    0
2015-11-24    0
2015-11-26    0
2015-11-27    1
2015-11-30    0
2015-12-01    0
2015-12-02    0
2015-12-03    0
2015-12-04    0
2015-12-07    0
2015-12-08    0
2015-12-09    0
2015-12-10    0
2015-12-11    0
2015-12-14    0
2015-12-15    0
2015-12-16    0
2015-12-17    1
2015-12-18    0
2015-12-21    0
2015-12-22    0
2015-12-23    0
```

```
2015-12-24    0
2015-12-28    0
2015-12-29    0
2015-12-30    0
2015-12-31    0
2016-01-01    0
Name: Close_price_shock, Length: 249, dtype: int64
```

0/1 dummy-coded time series for direction of shock of closing price

In [72]:

```
def direction_close_shock(stock_name):
    if(stock_name["Close_price_shock"]==1):
        if(stock_name["Close"]-stock_name["prev_day_close"]>0):
            return 1
        else:
            return 0
    else:
        return "Nan"
```

In [73]:

```
SBIN["dir_shock_price"] = 'Nan'
SBIN["dir_shock_price"] = SBIN.apply(direction_close_shock,axis=1)
print(SBIN["dir_shock_price"])
```

```
Date
2015-01-01    Nan
2015-01-02    Nan
2015-01-05      1
2015-01-06    Nan
2015-01-07    Nan
2015-01-08    Nan
2015-01-09    Nan
2015-01-12    Nan
2015-01-13    Nan
2015-01-14      0
2015-01-15    Nan
2015-01-16    Nan
2015-01-19    Nan
2015-01-20      0
2015-01-21    Nan
2015-01-22    Nan
2015-01-23    Nan
2015-01-27    Nan
2015-01-28      1
2015-01-29      1
2015-01-30    Nan
2015-02-02      1
2015-02-03      1
2015-02-04    Nan
2015-02-05    Nan
2015-02-06      1
2015-02-09      0
2015-02-10    Nan
2015-02-11    Nan
2015-02-12      0
              ...
2015-11-19    Nan
2015-11-20    Nan
2015-11-23    Nan
2015-11-24    Nan
2015-11-26      0
2015-11-27    Nan
2015-11-30    Nan
2015-12-01      1
2015-12-02    Nan
2015-12-03    Nan
2015-12-04    Nan
2015-12-07    Nan
2015-12-08    Nan
2015-12-09    Nan
2015-12-10      1
2015-12-11    Nan
2015-12-14    Nan
2015-12-15    Nan
2015-12-16    Nan
2015-12-17    Nan
2015-12-18    Nan
2015-12-21    Nan
2015-12-22    Nan
```

```
2015-12-23    Nan
2015-12-24    Nan
2015-12-28    Nan
2015-12-29    Nan
2015-12-30    Nan
2015-12-31    Nan
2016-01-01    Nan
Name: dir_shock_price, Length: 249, dtype: object
```

In [74]:

```
TCS["dir_shock_price"] = 'Nan'
TCS["dir_shock_price"] = TCS.apply(direction_close_shock,axis=1)
print(TCS["dir_shock_price"])
```

```
Date
2015-01-01    Nan
2015-01-02    Nan
2015-01-05      1
2015-01-06    Nan
2015-01-07    Nan
2015-01-08      0
2015-01-09    Nan
2015-01-12    Nan
2015-01-13    Nan
2015-01-14    Nan
2015-01-15    Nan
2015-01-16    Nan
2015-01-19    Nan
2015-01-20    Nan
2015-01-21    Nan
2015-01-22    Nan
2015-01-23    Nan
2015-01-27    Nan
2015-01-28    Nan
2015-01-29      1
2015-01-30    Nan
2015-02-02    Nan
2015-02-03    Nan
2015-02-04    Nan
2015-02-05    Nan
2015-02-06      1
2015-02-09      1
2015-02-10    Nan
2015-02-11    Nan
2015-02-12      0
              ...
2015-11-19    Nan
2015-11-20    Nan
2015-11-23    Nan
2015-11-24    Nan
2015-11-26    Nan
2015-11-27    Nan
2015-11-30    Nan
2015-12-01    Nan
2015-12-02    Nan
2015-12-03    Nan
2015-12-04    Nan
2015-12-07    Nan
2015-12-08    Nan
2015-12-09    Nan
2015-12-10    Nan
2015-12-11    Nan
2015-12-14    Nan
2015-12-15    Nan
2015-12-16    Nan
2015-12-17    Nan
2015-12-18    Nan
2015-12-21    Nan
2015-12-22    Nan
```

```
2015-12-23    Nan
2015-12-24    Nan
2015-12-28    Nan
2015-12-29    Nan
2015-12-30    Nan
2015-12-31    Nan
2016-01-01    Nan
Name: dir_shock_price, Length: 249, dtype: object
```

In [75]:

```
INFY["dir_shock_price"] = 'Nan'
INFY["dir_shock_price"] = INFY.apply(direction_close_shock,axis=1)
print(INFY["dir_shock_price"])
```

```
Date
2015-01-01    Nan
2015-01-02    Nan
2015-01-05      1
2015-01-06    Nan
2015-01-07    Nan
2015-01-08      0
2015-01-09      0
2015-01-12    Nan
2015-01-13    Nan
2015-01-14    Nan
2015-01-15    Nan
2015-01-16    Nan
2015-01-19    Nan
2015-01-20      0
2015-01-21    Nan
2015-01-22    Nan
2015-01-23      1
2015-01-27    Nan
2015-01-28    Nan
2015-01-29    Nan
2015-01-30    Nan
2015-02-02    Nan
2015-02-03    Nan
2015-02-04      0
2015-02-05    Nan
2015-02-06    Nan
2015-02-09    Nan
2015-02-10    Nan
2015-02-11    Nan
2015-02-12    Nan
              ...
2015-11-19    Nan
2015-11-20    Nan
2015-11-23    Nan
2015-11-24    Nan
2015-11-26    Nan
2015-11-27      0
2015-11-30    Nan
2015-12-01    Nan
2015-12-02    Nan
2015-12-03    Nan
2015-12-04    Nan
2015-12-07    Nan
2015-12-08    Nan
2015-12-09    Nan
2015-12-10    Nan
2015-12-11    Nan
2015-12-14    Nan
2015-12-15    Nan
2015-12-16    Nan
2015-12-17      1
2015-12-18    Nan
2015-12-21    Nan
2015-12-22    Nan
```

```
2015-12-23    Nan
2015-12-24    Nan
2015-12-28    Nan
2015-12-29    Nan
2015-12-30    Nan
2015-12-31    Nan
2016-01-01    Nan
Name: dir_shock_price, Length: 249, dtype: object
```

Pricing shock without volume shock

In [76]:

```
SBIN["notVolShock"]  = (~(SBIN["Volume_Shock"].astype(bool))).astype(int)
SBIN["Pshock_w/o_volShock"] =  (SBIN["notVolShock"] & SBIN["dir_shock_price"]).asty
print(SBIN["Pshock_w/o_volShock"])
```

```
Date
2015-01-01    1
2015-01-02    0
2015-01-05    1
2015-01-06    0
2015-01-07    1
2015-01-08    0
2015-01-09    0
2015-01-12    0
2015-01-13    0
2015-01-14    0
2015-01-15    0
2015-01-16    0
2015-01-19    0
2015-01-20    0
2015-01-21    0
2015-01-22    0
2015-01-23    0
2015-01-27    0
2015-01-28    0
2015-01-29    0
2015-01-30    0
2015-02-02    0
2015-02-03    0
2015-02-04    0
2015-02-05    0
2015-02-06    1
2015-02-09    0
2015-02-10    0
2015-02-11    0
2015-02-12    0
             ..
2015-11-19    0
2015-11-20    0
2015-11-23    0
2015-11-24    0
2015-11-26    0
2015-11-27    0
2015-11-30    0
2015-12-01    0
2015-12-02    0
2015-12-03    0
2015-12-04    0
2015-12-07    0
2015-12-08    1
2015-12-09    0
2015-12-10    0
2015-12-11    1
2015-12-14    1
2015-12-15    0
2015-12-16    0
2015-12-17    0
2015-12-18    0
2015-12-21    0
2015-12-22    1
```

```
2015-12-23    0
2015-12-24    1
2015-12-28    0
2015-12-29    0
2015-12-30    0
2015-12-31    1
2016-01-01    0
Name: Pshock_w/o_volShock, Length: 249, dtype: int64
```

In [77]:

```
TCS["notVolShock"]  = (~(TCS["Volume_Shock"].astype(bool))).astype(int)
TCS["Pshock_w/o_volShock"] =  (TCS["notVolShock"] & TCS["dir_shock_price"]).astype(
print(TCS["Pshock_w/o_volShock"])
```

```
Date
2015-01-01    1
2015-01-02    0
2015-01-05    0
2015-01-06    0
2015-01-07    1
2015-01-08    0
2015-01-09    0
2015-01-12    0
2015-01-13    1
2015-01-14    0
2015-01-15    0
2015-01-16    1
2015-01-19    0
2015-01-20    0
2015-01-21    0
2015-01-22    1
2015-01-23    0
2015-01-27    0
2015-01-28    0
2015-01-29    0
2015-01-30    0
2015-02-02    0
2015-02-03    0
2015-02-04    1
2015-02-05    0
2015-02-06    0
2015-02-09    1
2015-02-10    0
2015-02-11    0
2015-02-12    0
             ..
2015-11-19    0
2015-11-20    1
2015-11-23    0
2015-11-24    1
2015-11-26    0
2015-11-27    0
2015-11-30    0
2015-12-01    0
2015-12-02    1
2015-12-03    0
2015-12-04    0
2015-12-07    1
2015-12-08    0
2015-12-09    0
2015-12-10    0
2015-12-11    1
2015-12-14    0
2015-12-15    0
2015-12-16    0
2015-12-17    0
2015-12-18    0
2015-12-21    0
2015-12-22    1
```

```
2015-12-23    0
2015-12-24    1
2015-12-28    0
2015-12-29    0
2015-12-30    1
2015-12-31    0
2016-01-01    0
Name: Pshock_w/o_volShock, Length: 249, dtype: int64
```

In [78]:

```
INFY["notVolShock"]  = (~(INFY["Volume_Shock"].astype(bool))).astype(int)
INFY["Pshock_w/o_volShock"] =  (INFY["notVolShock"] & INFY["dir_shock_price"]).asty
print(INFY["Pshock_w/o_volShock"])
```

```
Date
2015-01-01    1
2015-01-02    0
2015-01-05    0
2015-01-06    1
2015-01-07    0
2015-01-08    0
2015-01-09    0
2015-01-12    0
2015-01-13    0
2015-01-14    0
2015-01-15    0
2015-01-16    0
2015-01-19    0
2015-01-20    0
2015-01-21    0
2015-01-22    0
2015-01-23    1
2015-01-27    0
2015-01-28    1
2015-01-29    0
2015-01-30    0
2015-02-02    0
2015-02-03    0
2015-02-04    0
2015-02-05    0
2015-02-06    0
2015-02-09    0
2015-02-10    0
2015-02-11    0
2015-02-12    1
              ..
2015-11-19    0
2015-11-20    0
2015-11-23    1
2015-11-24    0
2015-11-26    0
2015-11-27    0
2015-11-30    0
2015-12-01    0
2015-12-02    1
2015-12-03    0
2015-12-04    0
2015-12-07    0
2015-12-08    0
2015-12-09    0
2015-12-10    0
2015-12-11    1
2015-12-14    0
2015-12-15    0
2015-12-16    1
2015-12-17    0
2015-12-18    0
2015-12-21    0
2015-12-22    1
```

```
2015-12-23    0
2015-12-24    0
2015-12-28    0
2015-12-29    0
2015-12-30    0
2015-12-31    0
2016-01-01    0
Name: Pshock_w/o_volShock, Length: 249, dtype: int64
```

Part 2 (data visualization): ¶

In [79]:

```
# Importing plotting libraries
from bokeh.plotting import figure, show, output_file, output_notebook
from bokeh.palettes import Spectral11, colorblind, Inferno, BuGn, brewer,GnBu,Blues
from bokeh.models import HoverTool, value, LabelSet, Legend, ColumnDataSource,Linea
```

In [80]:

```
output_notebook()
```

(https://bokeh.pydata.org)

BokehJS 0.12.13 successfully loaded.

In [81]:

```
def bokeh_visuals(stock):
    fig = figure(x_axis_type="datetime")
    fig.line(stock.index, stock['Close'], color='blue', alpha=0.5)

    # fig.line(sbin.index[2:10],sbin['Close'],color='red',alpha=0.5)
    # flag = False
    # last_i = 0
    # segments = []
    # for i in range(len(sbin["Volume_Shock"])):
    #     if(sbin["Volume_Shock"][i] and flag):
    # #         fig.line(sbin.index[last_i:i], sbin['Close'], color='red', alpha=0.
    #         segments.append((last_i,i))
    #         flag = False
    #     elif(sbin["Volume_Shock"][i]):
    #         last_i = i
    #         flag = True
    # fig.segment(x0=sbin["Close"],x1=sbin["Close"],y0=segments[0],y1=segments[1])
    fig.circle(stock.index, stock.Close*stock["Pshock_w/o_volShock"], size=4, legen
    show(fig)
```

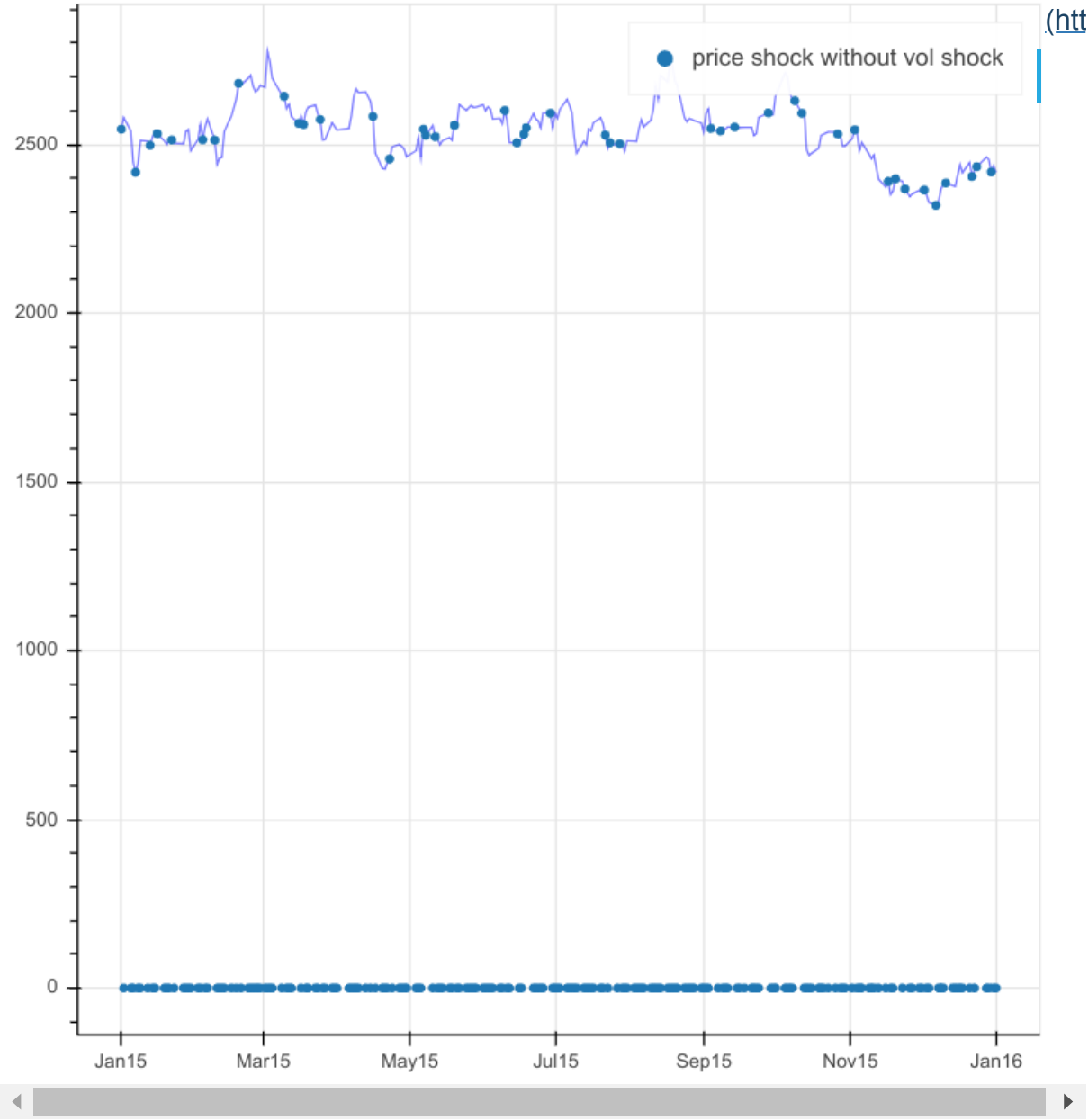
In [82]:

```
bokeh_visuals(SBIN)
```

(htt

In [83]:

```
bokeh_visuals(TCS)
```



(htt

In [84]:

```
bokeh_visuals(INFY)
```

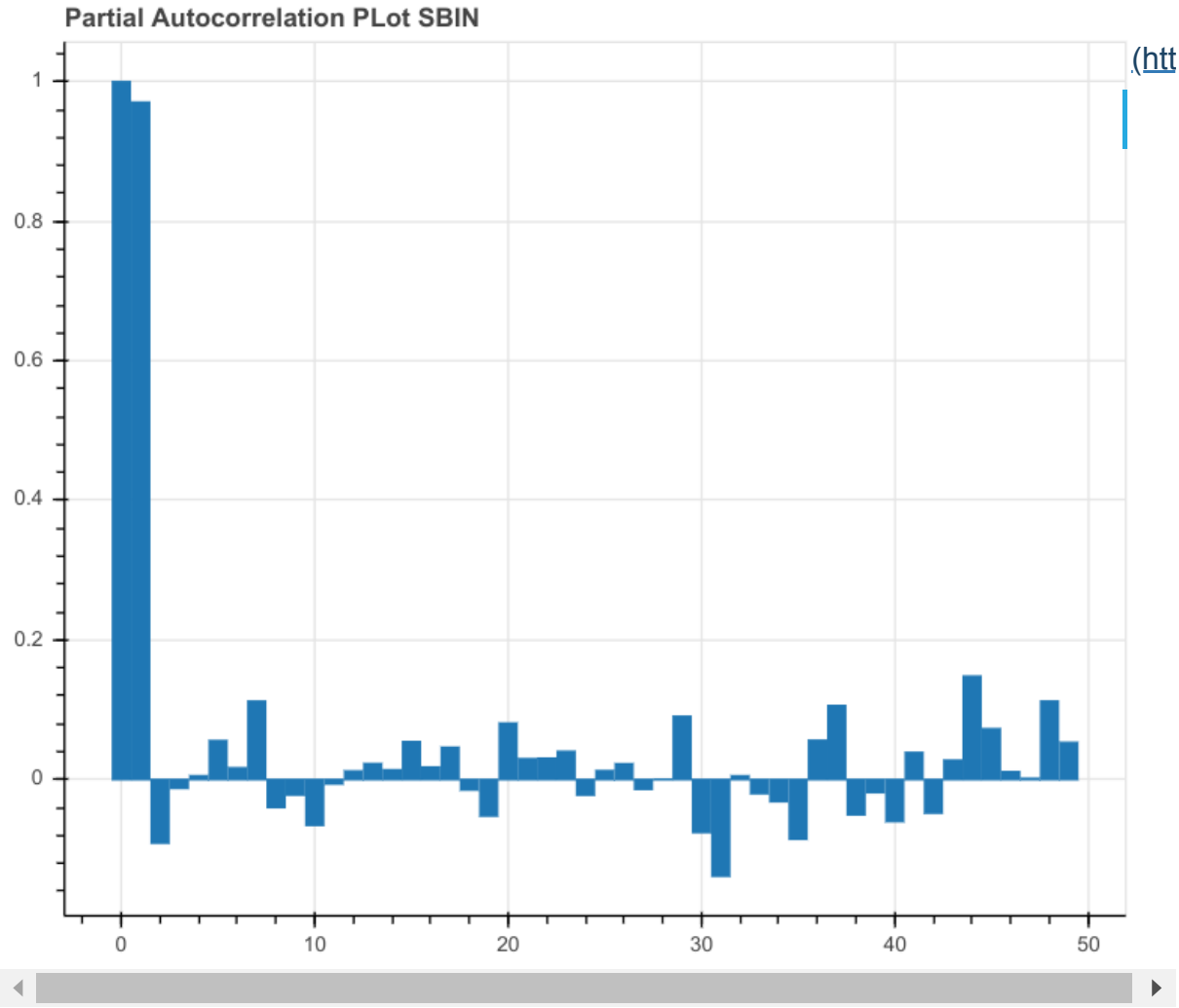
In [85]:

```
from statsmodels.tsa.stattools import acf, pacf

def draw_pacf(stock):

    lags = 50

    x = list(range(lags))

    p = figure(plot_height=500, title="Partial Autocorrelation PLot " +stock.name)

    partial_autocorr = pacf(stock["Close"], nlags=lags-1)
    p.vbar(x=x, top=partial_autocorr, width=0.9)
    show(p)
```

Kindly run it on your system as bokeg figure are not visible on github.

In [87]:

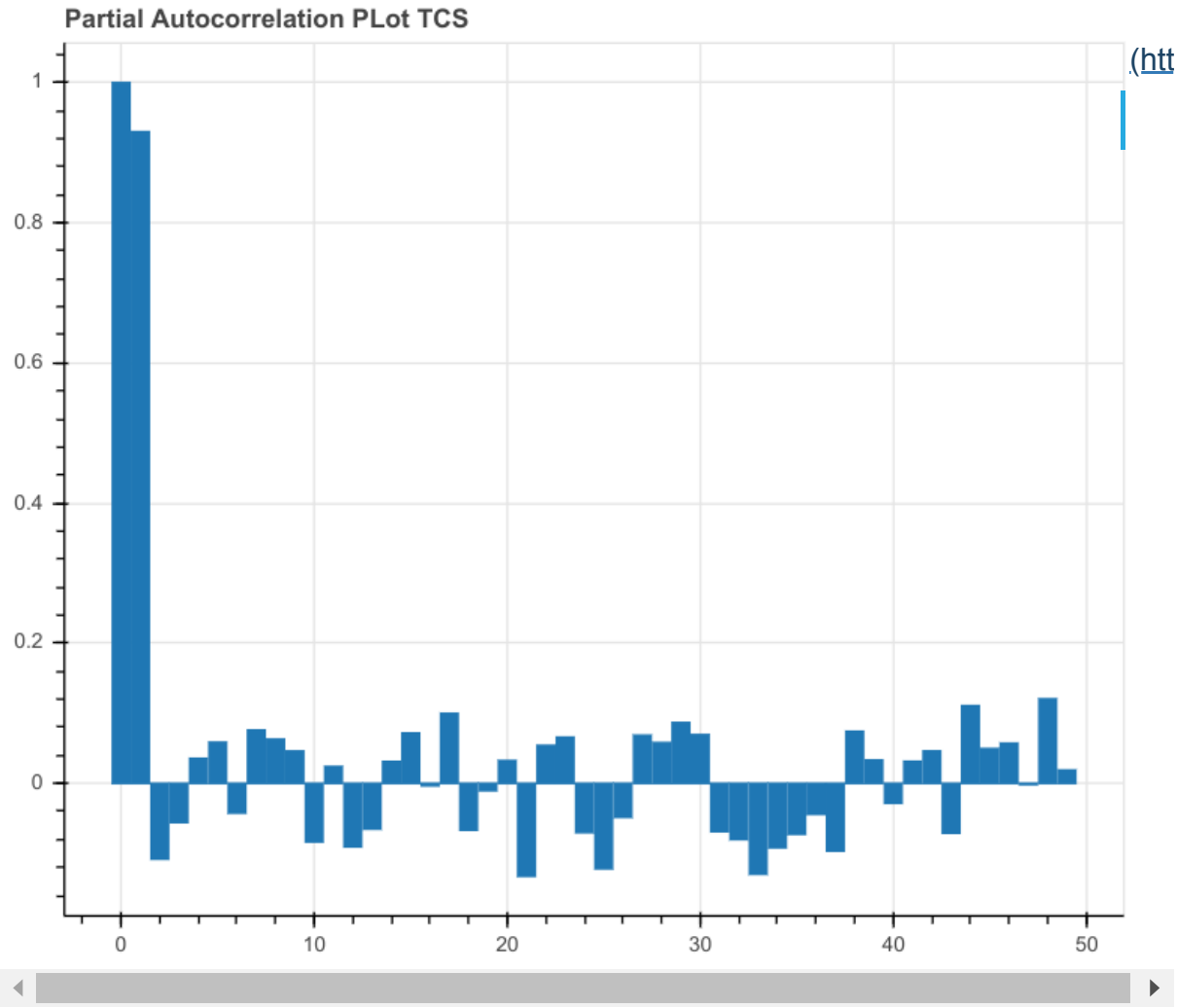
```
draw_pacf(SBIN)
```



(htt

In [88]:

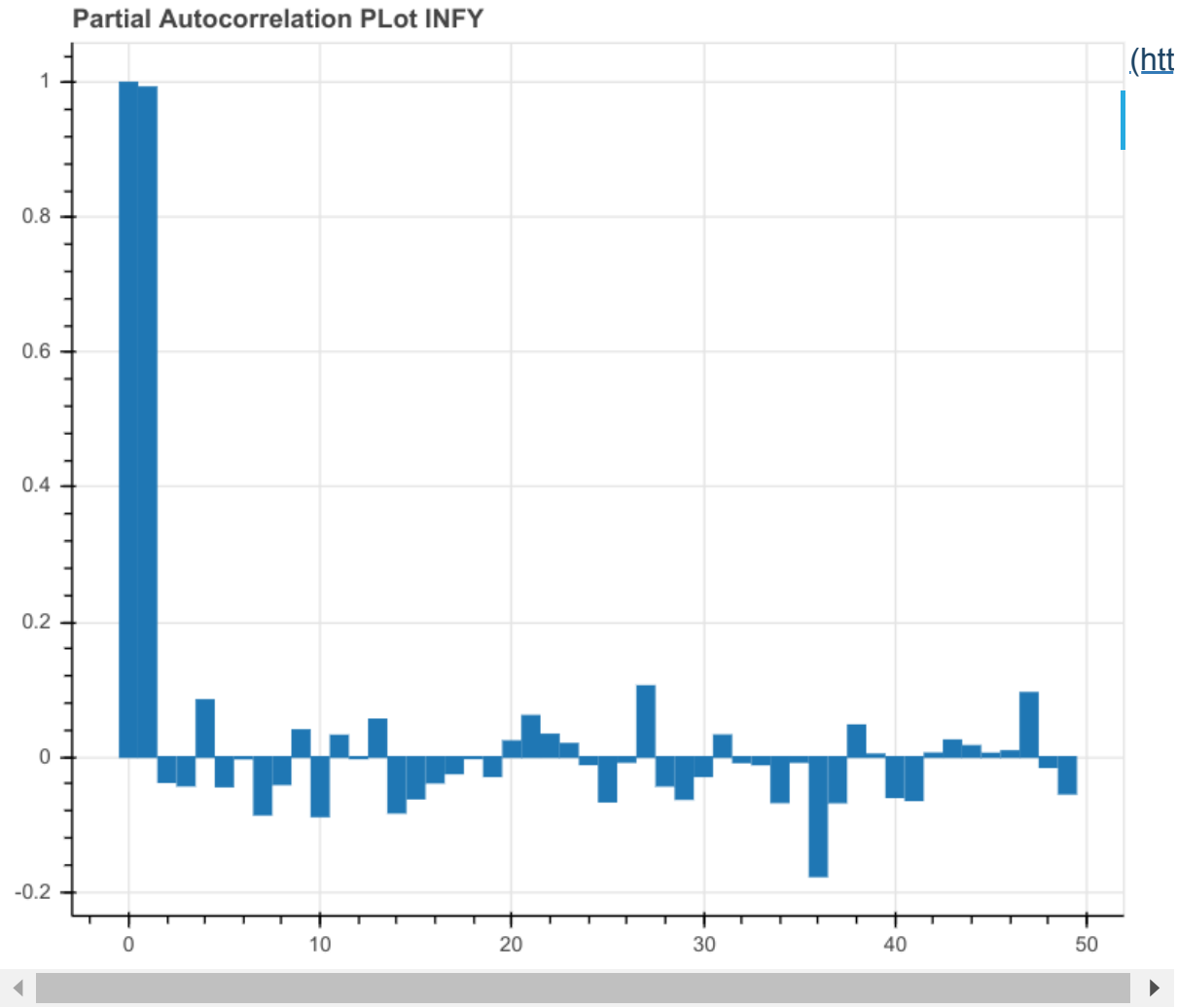
```
draw_pacf(TCS)
```



(htt

In [89]:

```
draw_pacf(INFY)
```



(htt