

# Reference Solutions to Assignment 1

## Question 1.

1(a)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
%run getstock.py
```

```
aapl=getMinuteStockPrices("AAPL")
aapl.head()
```

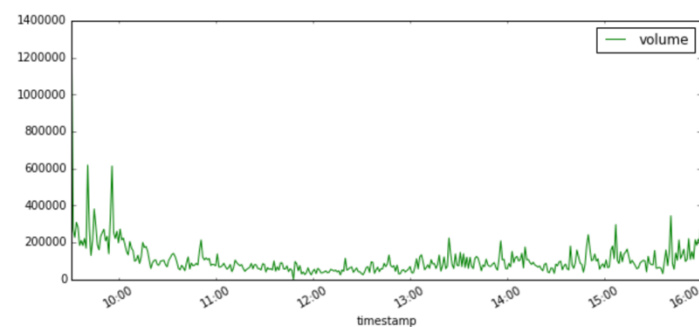
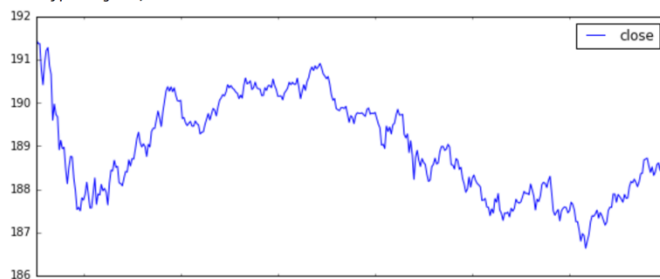
	open	high	low	close	volume
timestamp					
2019-03-25 09:31:00	191.54	191.7150	191.19	191.4600	1398191
2019-03-25 09:32:00	191.51	191.9592	191.32	191.3718	271269
2019-03-25 09:33:00	191.39	191.3900	191.05	191.3700	227858
2019-03-25 09:34:00	191.35	191.3900	190.66	190.7600	308765
2019-03-25 09:35:00	190.70	190.7300	190.34	190.4200	282864

1(b)

```
aapl0325=aapl['2019-03-25 09:31:00':'2019-03-25 16:00:00']
```

```
from datetime import datetime
aapl=aapl.sort_values('timestamp',ascending=True)
aapl0325[['close','volume']].plot(subplots=True, figsize=(10,10))
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x10a197320>,
      <matplotlib.axes._subplots.AxesSubplot object at 0x10ed29c18>],
      dtype=object)
```



1(c)

```
aapl0325.describe().round(4)
```

	open	high	low	close	volume
count	390.0000	390.0000	390.0000	390.0000	3.900000e+02
mean	188.9622	189.0805	188.8354	188.9536	1.067969e+05
std	1.1239	1.1118	1.1210	1.1172	9.715661e+04
min	186.6100	186.8800	186.6000	186.6350	8.570000e+02
25%	187.9418	188.1266	187.8300	187.9350	5.969175e+04
50%	188.8754	188.9650	188.7150	188.8600	8.272700e+04
75%	189.8875	189.9925	189.7875	189.8875	1.221585e+05
max	191.5400	191.9592	191.3200	191.4600	1.398191e+06

```
total_vol=sum(aapl0325['volume'])
total_vol
```

41650777

```
def vwap(stock):
    return ((stock.close*stock.volume).sum()/stock.volume.sum())
vwap(aapl0325).round(4)
```

188.8675

## Question 2.

Add the following functions in orderbook.py.

```
def random_price(side, bbo):
    if side == 's':
        price = bbo[1][0] + round(np.random.random(), 2)
    else:
        price = bbo[0][0] + round(np.random.random(), 2)
    return price

def new_BBO(n):
    bbo_df = pd.DataFrame(columns=['bid', 'offer'], index=list(range(n)))
    dummy_book()
    for i in range(n):
        bbo = get_BBO()
        side = np.random.choice(['s', 'b'])
        price = random_price(side, bbo)
        qty = int(np.random.randint(0, 2000))
        new_order(side, price, qty)
        new_bbo = get_BBO()
        bbo_df.bid[i] = new_bbo[0][0]
        bbo_df.offer[i] = new_bbo[1][0]
    return bbo_df

def plot_bbo(bbo_df):
    plt.figure(figsize=(14, 9))
    plt.plot(bbo_df.bid, "r-")
    plt.plot(bbo_df.offer, "b-")
    plt.xlabel("order")
    plt.ylabel("price")
    plt.legend(loc="upper right")
    plt.show()

if __name__ == '__main__':
    #dummy_book()
    iteration_times = 350 # for loop
    bbo_df = new_BBO(iteration_times)
    plot_bbo(bbo_df)
```

Then run the main function as follows:

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
```

```
%run orderbook.py
```

