In [1]:

```
import torch as torch
```

Question 1:

The DockerFile from the assignment (page 2) was used. Modification was made because macos was used. Instead of 'pwd', $(pwd) was used Python Version:

In [2]:

```
import sys
print(str(sys.version_info[0]) + '.' + str(sys.version_info[1]))
```

```
3.6
```

Pytorch Version:

In [3]:

```
print(torch.__version__)
```

```
0.3.0.post4
```

Question 2

In [4]:

```
import pandas as pd
```

In [5]:

```
#wget was used to get the file
#load data into pandas
data = pd.read_csv('btc.csv');
```

In [6]:

```
fullframe = data[['Timestamp','Close']]
```

In [7]:

```
# rollingMean = pd.rolling_mean(fullframe['Close'],10)
# rollingVar = pd.rolling_var(fullframe['Close'],10)
rollingMean = fullframe['Close'].rolling(window=10,center=False).mean()
rollingVar = fullframe['Close'].rolling(window=10,center=False).var()
```

In [8]:

```
fullframe.loc[:,'Rolling Mean'] = rollingMean
fullframe.loc[:,'Rolling Var'] = rollingVar
```

```
/opt/conda/lib/python3.6/site-packages/pandas/core/indexing.py:357:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/indexing.html#indexing-view-versus-copy
  self.obj[key] = _infer_fill_value(value)
/opt/conda/lib/python3.6/site-packages/pandas/core/indexing.py:537:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/indexing.html#indexing-view-versus-copy
  self.obj[item] = s
```

In [9]:

```
fullframe
```

Out[9]:

	Timestamp	Close	Rolling Mean	Rolling Var
0	2017-11-25 00:00:00	8717.99	NaN	NaN
1	2017-11-26 00:00:00	9271.06	NaN	NaN
2	2017-11-27 00:00:00	9708.07	NaN	NaN
3	2017-11-28 00:00:00	9868.82	NaN	NaN
4	2017-11-29 00:00:00	9824.68	NaN	NaN
5	2017-11-30 00:00:00	9947.67	NaN	NaN
6	2017-12-01 00:00:00	10840.45	NaN	NaN
7	2017-12-02 00:00:00	10872.00	NaN	NaN
8	2017-12-03 00:00:00	11250.00	NaN	NaN
9	2017-12-04 00:00:00	11613.07	10191.381	8.417531e+05
10	2017-12-05 00:00:00	11677.00	10487.282	7.484874e+05
11	2017-12-06 00:00:00	13623.50	10922.526	1.466520e+06
12	2017-12-07 00:00:00	16599.99	11611.718	4.356391e+06
13	2017-12-08 00:00:00	15800.00	12204.836	5.577071e+06

13	2017-12-08 00:00:00	15800.00	12204.836	5.377671e+06
14	2017-12-09 00:00:00	14607.49	12683.117	5.334858e+06
15	2017-12-10 00:00:00	14691.00	13157.450	4.701414e+06
16	2017-12-11 00:00:00	16470.00	13720.405	4.972004e+06
17	2017-12-12 00:00:00	16650.01	14298.206	4.653186e+06
18	2017-12-13 00:00:00	16250.00	14798.206	3.766290e+06
19	2017-12-14 00:00:00	16404.99	15277.398	2.670781e+06
20	2017-12-15 00:00:00	17471.50	15856.848	1.392291e+06
21	2017-12-16 00:00:00	19187.78	16413.276	1.726863e+06
22	2017-12-17 00:00:00	18953.00	16648.577	2.378160e+06
23	2017-12-18 00:00:00	18940.57	16962.634	2.772252e+06
24	2017-12-19 00:00:00	17700.00	17271.885	2.110101e+06
25	2017-12-20 00:00:00	16466.98	17449.483	1.406934e+06
26	2017-12-21 00:00:00	15600.01	17362.484	1.671987e+06
27	2017-12-22 00:00:00	14009.79	17098.462	2.787082e+06
28	2017-12-23 00:00:00	14619.00	16935.362	3.360619e+06
29	2017-12-24 00:00:00	14157.87	16710.650	4.130420e+06
30	2017-12-25 00:00:00	13911.28	16354.628	4.795983e+06
31	2017-12-26 00:00:00	15764.44	16012.294	3.812610e+06
32	2017-12-27 00:00:00	15364.93	15653.487	2.755266e+06
33	2017-12-28 00:00:00	14470.07	15206.437	1.488269e+06
34	2017-12-29 00:00:00	14340.00	14870.437	7.553684e+05
35	2017-12-30 00:00:00	12640.00	14487.739	8.621820e+05
36	2017-12-31 00:00:00	13880.00	14315.738	7.328883e+05
37	2018-01-01 00:00:00	13443.41	14259.100	8.034742e+05
38	2018-01-02 00:00:00	14678.94	14265.094	8.086273e+05
39	2018-01-03 00:00:00	15155.62	14364.869	8.844039e+05
40	2018-01-04 00:00:00	15143.67	14488.108	9.120605e+05
41	2018-01-05 00:00:00	16928.00	14604.464	1.377467e+06
42	2018-01-06 00:00:00	17149.67	14782.938	1.997605e+06
43	2018-01-07 00:00:00	16124.02	14948.333	2.156167e+06
44	2018-01-08 00:00:00	14999.99	15014.332	2.110505e+06

45	2018-01-09 00:00:00	14403.51	15190.683	1.491022e+06
46	2018-01-10 00:00:00	14890.02	15291.685	1.298855e+06
47	2018-01-11 00:00:00	13243.83	15271.727	1.384811e+06
48	2018-01-12 00:00:00	13781.41	15181.974	1.583599e+06
49	2018-01-13 00:00:00	14197.78	15086.190	1.680954e+06
50	2018-01-14 00:00:00	13647.99	14936.622	1.885555e+06
51	2018-01-15 00:00:00	13607.04	14604.526	1.518813e+06
52	2018-01-16 00:00:00	11386.34	14028.193	1.580743e+06
53	2018-01-17 00:00:00	11191.35	13534.926	1.716528e+06
54	2018-01-18 00:00:00	11247.57	13159.684	1.902919e+06
55	2018-01-19 00:00:00	11552.00	12874.533	1.927856e+06
56	2018-01-20 00:00:00	12775.99	12663.130	1.427924e+06
57	2018-01-21 00:00:00	11558.87	12494.634	1.494398e+06
58	2018-01-22 00:00:00	10808.99	12197.392	1.527962e+06
59	2018-01-23 00:00:00	10620.56	11839.670	1.217428e+06

Question 3

In [10]:

```
A = torch.Tensor([[[7],[3]],[[11],[3.5]]])
B = torch.Tensor([[[7,3]],[[4.5,4.5]]])
Answer = torch.matmul(A,B)
```

In [11]:

```
Answer
```

Out[11]:

```
(0 ,.,.) = 
  49.0000  21.0000
  21.0000   9.0000

(1 ,.,.) = 
  49.5000  49.5000
  15.7500  15.7500
[torch.FloatTensor of size 2x2x2]
```