Assignment 0
CS5304 - Environment Setup
Weisi Zhang (Iris) - wz337

1. Docker Exercise.

This exercise will walk you through installing
Docker, running a container, and starting a jupyter notebook.
After building and running docker image, provide the following information.
• The pytorch version (this should be 0.3.0).
• The python version (this should be 3.x, such as 3.6.2). • (optional) Indicate whether you can run pytorch on GPU. This will require nvidia-docker and the correct pytorch installation. Note: This is only possible if you have access to a GPU (many computers don't). -- NO GPU

In [111]:
```
!hostname
!whoami
```

```
c70bad507ad7
nyc
```

In [112]:
```
import torch
print(torch.__version__)
```

```
0.3.0.post4
```

In [3]:
```
import sys
print (sys.version)
```

```
3.6.2 |Anaconda, Inc.| (default, Sep 22 2017, 02:03:08)
[GCC 7.2.0]
```

2. Statistics Exercise.

We've collected 60 days worth of bitcoin prices.
Download the file and calculate the running mean and variance of the closing price for each day (starting on the 10th day) using a window size of 10.
The data can be downloaded with the following command:
$ wget http://bit.ly/cs5304-assignment0-btc (http://bit.ly/cs5304-assignment0-btc) -O btc.csv

In [5]:
```
import pandas as pd
```

In [126]:
```
data = pd.read_csv("btc.csv")
data.head()
```

Out[126]:

	Timestamp	Open	High	Low	Close	Volume (BTC)	Volume (Currency)	Weighted Price
0	2017-11-25 00:00:00	8199.83	8737.0	8114.78	8717.99	11611.67	9.762892e+07	8407.83
1	2017-11-26 00:00:00	8718.00	9366.6	8538.20	9271.06	12021.22	1.085258e+08	9027.86
2	2017-11-27 00:00:00	9278.99	9721.7	9267.00	9708.07	13272.45	1.264581e+08	9527.86
3	2017-11-28 00:00:00	9708.06	9968.0	9582.25	9868.82	11214.93	1.104968e+08	9852.65
4	2017-11-29 00:00:00	9877.63	11395.0	9250.00	9824.68	33432.34	3.469496e+08	10377.66

In [128]:
```
#Running Mean of the Closing Price for Each Day (starting on the 10th da
y) of Window Size 10
rolling_mean_closing = data[['Close']].rolling(window=10,center=False).m
ean()
rolling_mean_closing
```

Out[128]:

	Close
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
5	NaN
6	NaN
7	NaN
8	NaN
9	10191.381
10	10487.282
11	10922.526
12	11611.718
13	12204.836
14	12683.117
15	13157.450
16	13720.405
17	14298.206
18	14798.206
19	15277.398
20	15856.848
21	16413.276
22	16648.577
23	16962.634
24	17271.885
25	17449.483
26	17362.484
27	17098.462
28	16935.362
29	16710.650
30	16354.628
31	16012.294

	Close
32	15653.487
33	15206.437
34	14870.437
35	14487.739
36	14315.738
37	14259.100
38	14265.094
39	14364.869
40	14488.108
41	14604.464
42	14782.938
43	14948.333
44	15014.332
45	15190.683
46	15291.685
47	15271.727
48	15181.974
49	15086.190
50	14936.622
51	14604.526
52	14028.193
53	13534.926
54	13159.684
55	12874.533
56	12663.130
57	12494.634
58	12197.392
59	11839.670

In [130]:
```
#Running Variance of the Closing Price for Each Day (starting on the 10t
h day) of Window Size 10
rolling_var_closing = data[['Close']].rolling(window=10, center=False).v
ar(ddof=1)
rolling_var_closing
```

Out[130]:

	Close
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
5	NaN
6	NaN
7	NaN
8	NaN
9	8.417531e+05
10	7.484874e+05
11	1.466520e+06
12	4.356391e+06
13	5.577071e+06
14	5.334858e+06
15	4.701414e+06
16	4.972004e+06
17	4.653186e+06
18	3.766290e+06
19	2.670781e+06
20	1.392291e+06
21	1.726863e+06
22	2.378160e+06
23	2.772252e+06
24	2.110101e+06
25	1.406934e+06
26	1.671987e+06
27	2.787082e+06
28	3.360619e+06
29	4.130420e+06
30	4.795983e+06
31	3.812610e+06

	Close
32	2.755266e+06
33	1.488269e+06
34	7.553684e+05
35	8.621820e+05
36	7.328883e+05
37	8.034742e+05
38	8.086273e+05
39	8.844039e+05
40	9.120605e+05
41	1.377467e+06
42	1.997605e+06
43	2.156167e+06
44	2.110505e+06
45	1.491022e+06
46	1.298855e+06
47	1.384811e+06
48	1.583599e+06
49	1.680954e+06
50	1.885555e+06
51	1.518813e+06
52	1.580743e+06
53	1.716528e+06
54	1.902919e+06
55	1.927856e+06
56	1.427924e+06
57	1.494398e+06
58	1.527962e+06
59	1.217428e+06

3. Tensor Exercise.

A is a 2 x 2 x 1 tensor, B is a 2 x 1 x 2 tensor.
Perform a “batched matrix multiplication” between A and B using torch.matmul.

In [92]:
```
import torch
```

In [123]:
```
t1 = torch.FloatTensor([[[7],[3]],[[11],[3.5]]])
t2 = torch.FloatTensor([[[7, 9]],[[4.5, 4.5]]])
```

In [124]:
```
t1
```

Out[124]:
```
(0 ,.,.) = 
   7.0000
   3.0000

(1 ,.,.) = 
  11.0000
   3.5000
[torch.FloatTensor of size 2x2x1]
```

In [125]:
```
t2
```

Out[125]:
```
(0 ,.,.) = 
  7.0000  9.0000

(1 ,.,.) = 
  4.5000  4.5000
[torch.FloatTensor of size 2x1x2]
```

In [107]:
```
torch.matmul(torch.FloatTensor(a),torch.FloatTensor(b))
```

Out[107]:
```
(0 ,.,.) = 
  49.0000  63.0000
  21.0000  27.0000

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