Lab Number : **07**
Due Date : **Oct 28, 2020**
Student Details :

- Name : **AB Satyaprakash**
- Roll Number : **180123062**
- Department : **Mathematics and Computing**

Formulae used:
Apart from the ones needed for estimation of μ and σ, we have used the following formulae/algorithms in the code

1. $S(t) = S(0) * e^{((\mu - 1/2\sigma^2)t + \sigma W(t))}$
2. $W(t) = (\sqrt{t})Z, \ where\ Z \sim N(0,1)$
3. Box-Muller algorithm to generate N(0,1)
4. %error = (|actual-expected|/(actual))*100

Question 1 :

Explanation:

First of all we used **pandas** (a python library) to read the csv file, i.e. the file having the daily adjusted closing stock prices for SBI for **July 1, 2020 to Sept 30, 2020**.

Then we moved on to the specific column of the data and stored it as a list. This list is then used to generate the **Ui**s given by natural logarithm of ratio of the current and previous day stock prices (more like previous entry, since days were missing in the csv accounting for non-working days).

Using the formulae given in the question pdf, corresponding to the gBm model, we obtained the **estimated** values of μ and σ.

Now after this is done, all we need is to apply formula (1) (given above) and for that we need to estimate the **standard BM random variables**. Since we know it comes from the standard normal distribution, we used **Box-Muller Algorithm** and then applied formula (2), to obtain W(t). Now we can directly use formula (1) and we use S(0) as the stock price on the last day we have, i.e. 30th of Sept 2020.

Now we obtain S(t) after using all the obtained parameters and also putting t as the number of entries from Sept 30, 2020 till Oct 7th, 14th and 21st respectively.

This will give us the 1000 values of S(t) corresponding to the 1000 normal rvs we generated. Taking their mean, gives us the **expected stock prices** on these three days!

Output:

*Please make sure that the file '**SBIN.NS.csv**' is in the same directory as well for pandas to do its work.*
Also, the following command will install ***pandas*** *on the linux system. One can use pip if one doesn't use python3.*

```
pip3 install pandas
```

The following output is obtained in the terminal after running the program **180123062_ABSatyaprakash.py.**

```
lord:Lab 7 imperial_lord$ python3 180123062_ABSatyaprakash.py
Estimated value of µ (mean) = 0.0002981060700200034
Estimated value of σ (standard deviation) = 0.02228172705870555



Expected Stock price S(t) on 7th October 2020 = 185.50657619339663
Expected Stock price S(t) on 14th October 2020 = 186.5843857397671
Expected Stock price S(t) on 21st October 2020 = 186.34031006072485
```

The expected values can vary in one's case since N(0,1) is being generated randomly, but they'll roughly be the same.

Question 2 :

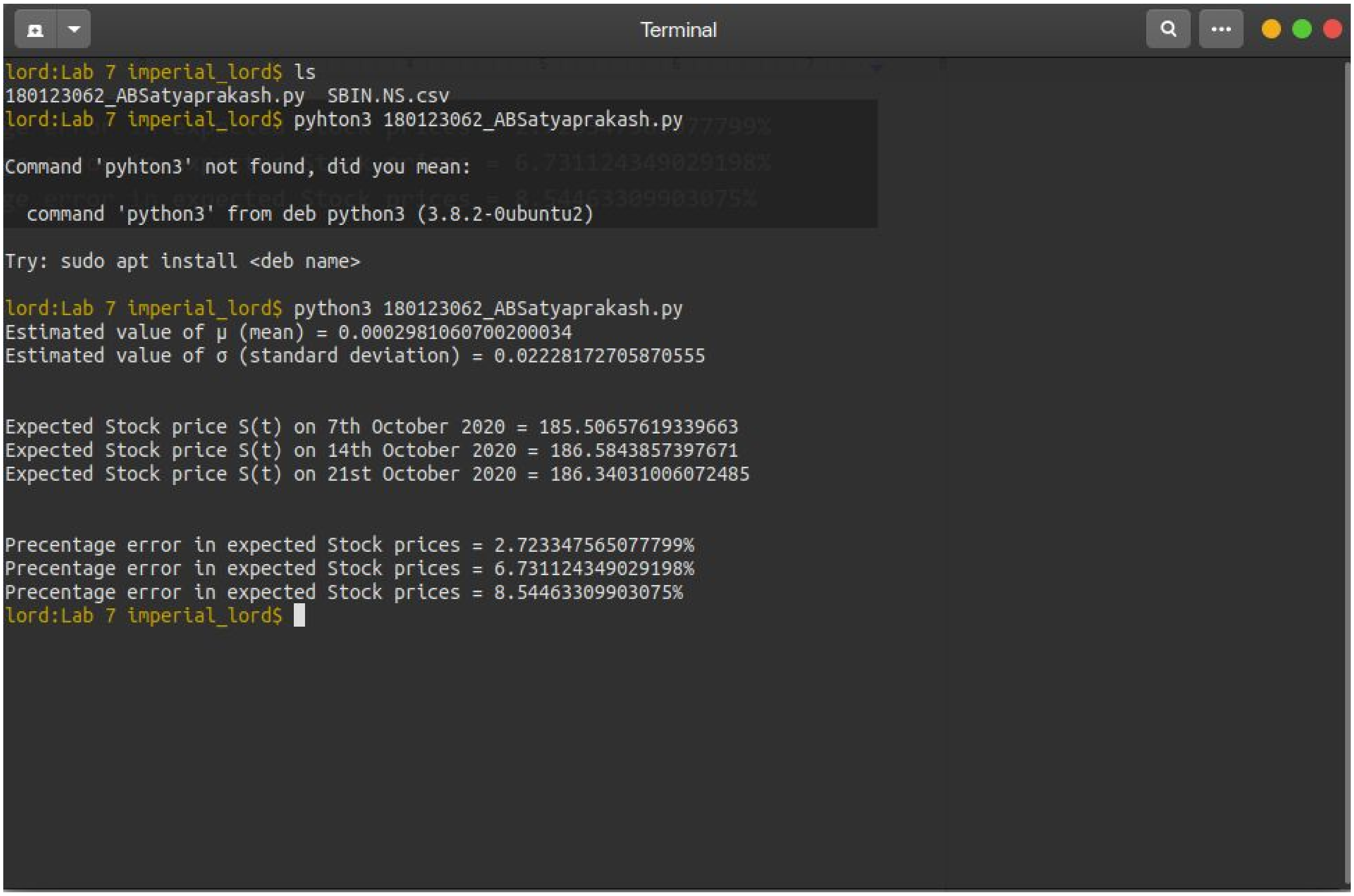
Explanation:

This part of the question just needs to find the percentage error. We peek into the data set at https://finance.yahoo.com/quote/SBIN.NS/history/ and find the closing prices on Oct 7, 14 and 21. Then we use formula 4 given above and obtain the percentage error in our estimated stock prices.

Output:

After we run the code the following output is obtained. Note that the both outputs are obtained on running the same code once.

```
Percentage error in expected Stock prices = 2.723347565077799%
Precentage error in expected Stock prices = 6.731124349029198%
Percentage error in expected Stock prices = 8.54463309903075%
```



```
lord:Lab 7 imperial_lord$ ls
180123062_ABSatyaprakash.py  SBIN.NS.csv
lord:Lab 7 imperial_lord$ pyhton3 180123062_ABSatyaprakash.py

Command 'pyhton3' not found, did you mean:

  command 'python3' from deb python3 (3.8.2-0ubuntu2)

Try: sudo apt install <deb name>

lord:Lab 7 imperial_lord$ python3 180123062_ABSatyaprakash.py
Estimated value of μ (mean) = 0.0002981060700200034
Estimated value of σ (standard deviation) = 0.02228172705870555


Expected Stock price S(t) on 7th October 2020 = 185.50657619339663
Expected Stock price S(t) on 14th October 2020 = 186.5843857397671
Expected Stock price S(t) on 21st October 2020 = 186.34031006072485


Precentage error in expected Stock prices = 2.723347565077799%
Precentage error in expected Stock prices = 6.731124349029198%
Precentage error in expected Stock prices = 8.54463309903075%
lord:Lab 7 imperial_lord$ 
```

This is how the output typically looks on a terminal!