Lab Number : 05
Due Date : Oct 7, 2020
Student Details :

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

Question 1 :

(a) The idea here is to use both the algorithms (**Box-Muller** and **Marsaglia and Bray**) and generate the bivariate Normal distribution **(Z1,Z2)** and then use Z1 and Z2 to generate the **N(0,1) distribution** by choosing 50% of values from each.

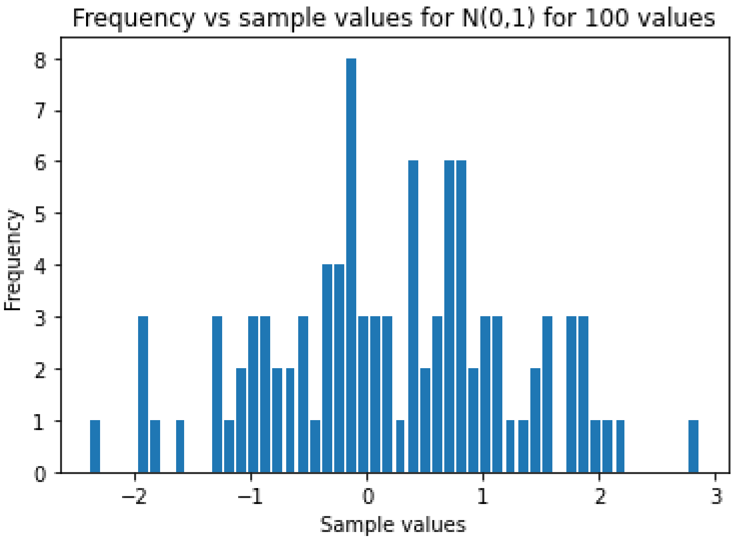
In the code this is acheived by the functions **find_values_N01()** which fill 2 lists **n01_100[]** and **n01_10000[]** corresponding to **100** and **10000** values from **N(0,1)** and taking Z1 and Z2 alternatively. After all this is done, I used the **statistics** library to find mean and variance. The results are as follows (also find on **terminal** on running the programs):

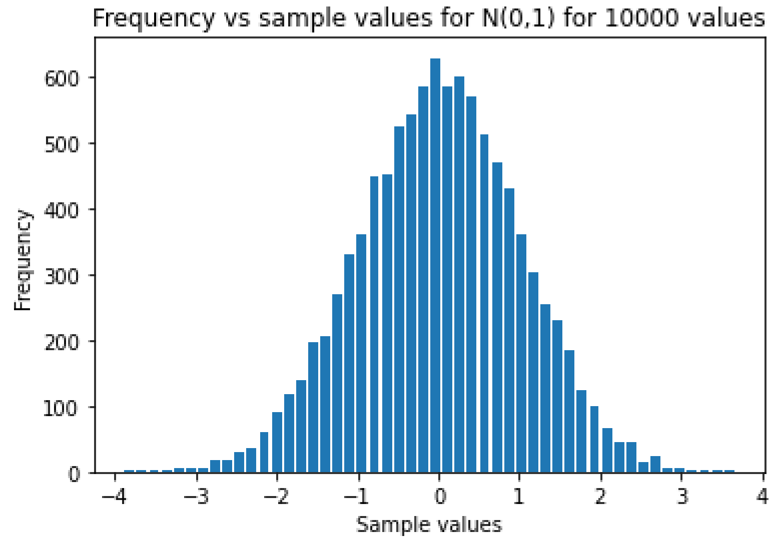
```
Box-Muller method
For 100 values
Sample mean= 0.2163719060464981
Sample variance= 1.1062330392466029

For 10000 values
Sample mean= 0.012424633921115728
Sample variance= 1.0070454929189074
```

```
Marsaglia and Bray method
For 100 values
Sample mean= -0.017555850491766476
Sample variance= 1.0526134252124868

For 10000 values
Sample mean= 0.013439732479124031
Sample variance= 0.9969904962676316
```

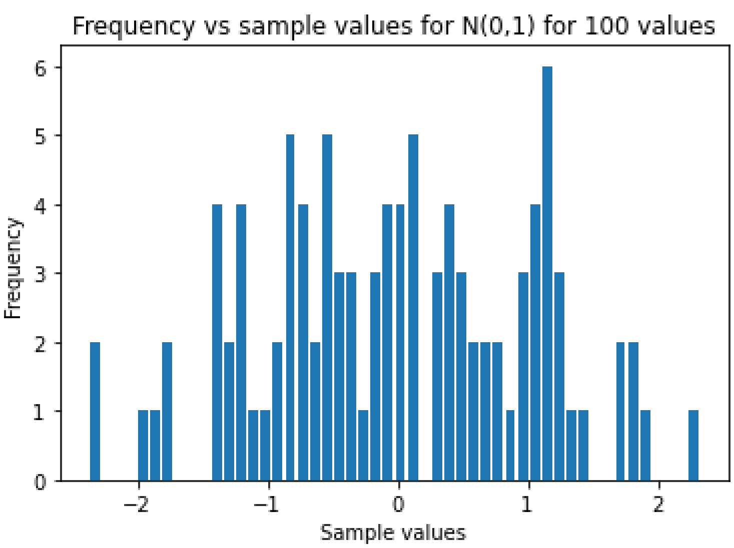
(b) The graphs are plot for **100** and **10000** values in both cases and the following histograms are obtained

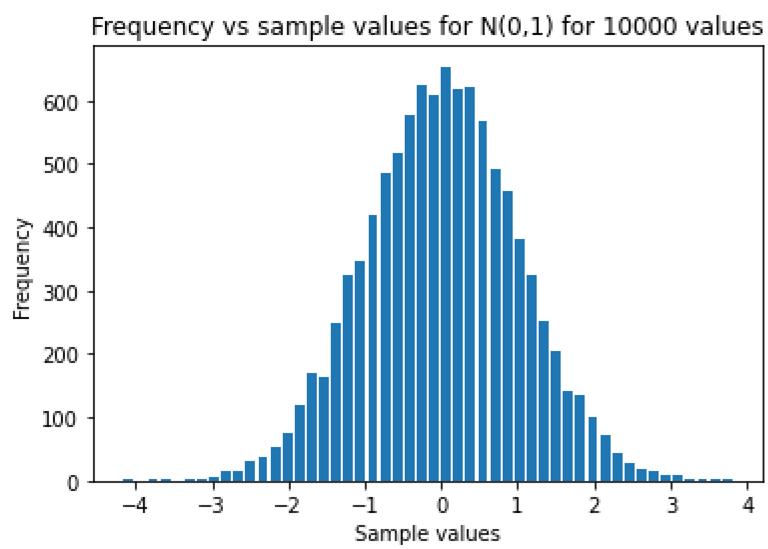
Box-Muller :





Marsaglia and Bray :





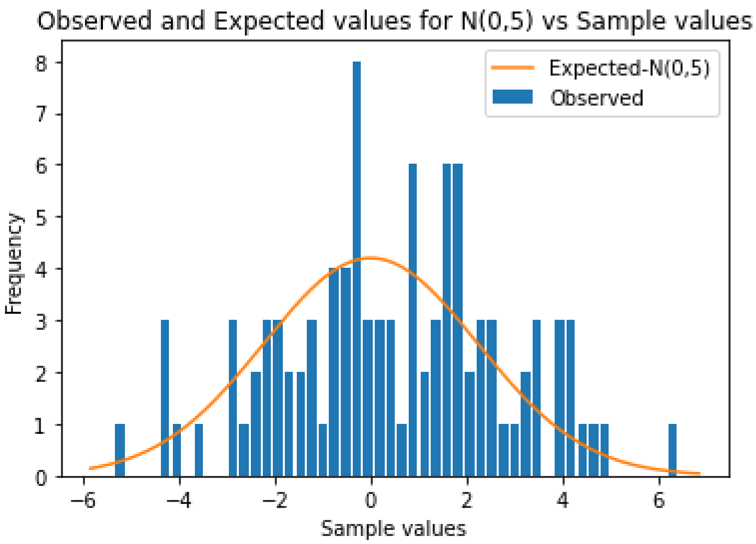
(c) To obtain the values of

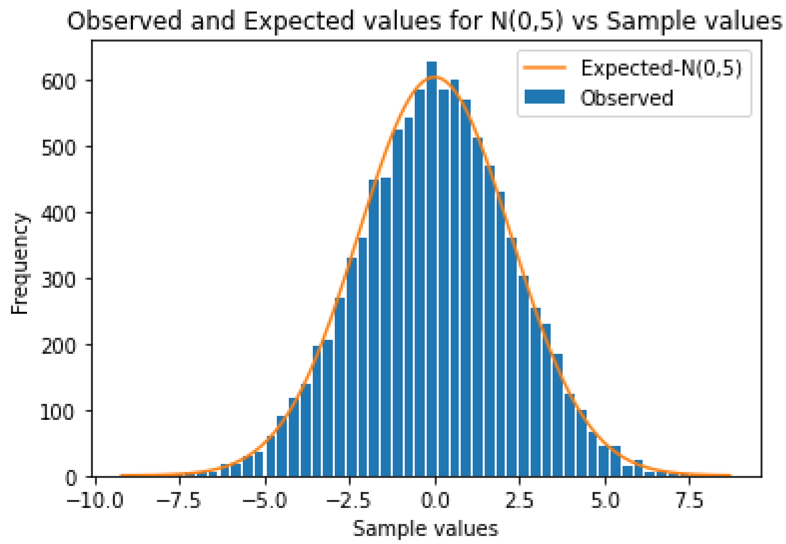
- **N(0,5)** - multiply them by $\sqrt{5}$
- **N(5,5)** - multiply by $\sqrt{5}$ and add 5

In this way we get the 100 and 10000 values. After this we plot them vs their frequencies. Additionally to get the PDF we make use of the bin width and number of observations to get the proper scaling factor. This done we plot both of them in the same graph.

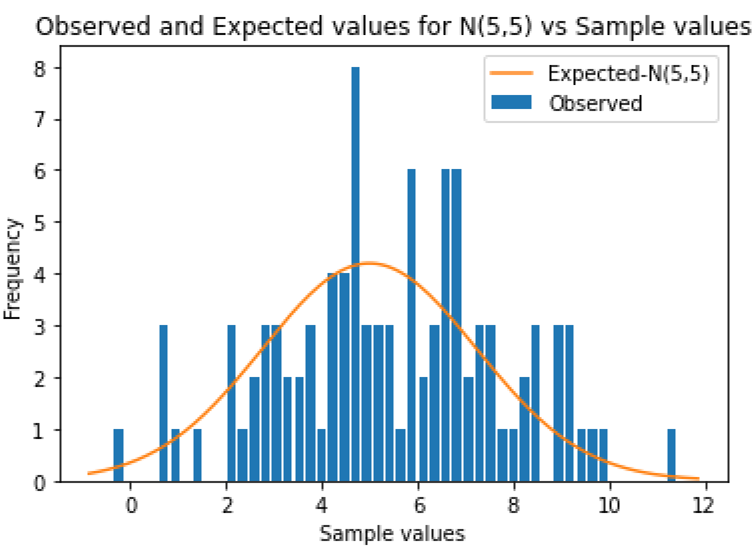
We observe that the expected value matches (roughly) the obtained values for all the cases, i.e. the graph traces out the tops of the histogram.

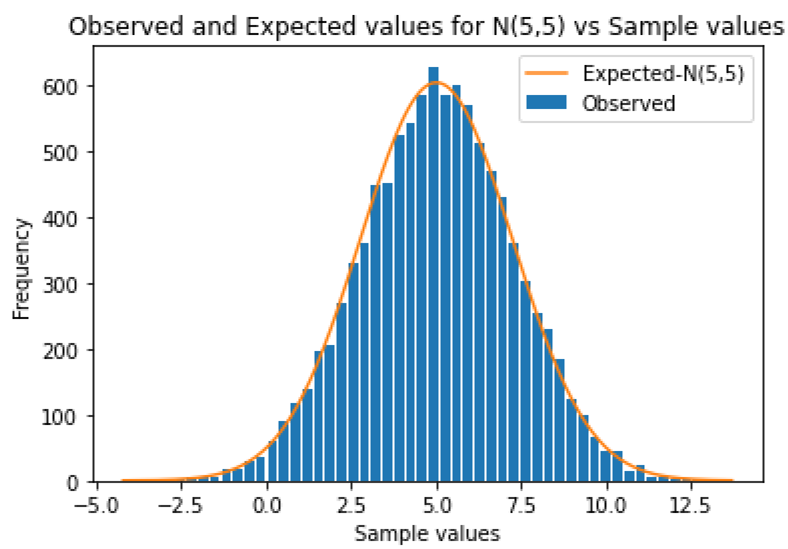
Box-Muller :
N(0,5) distribution-





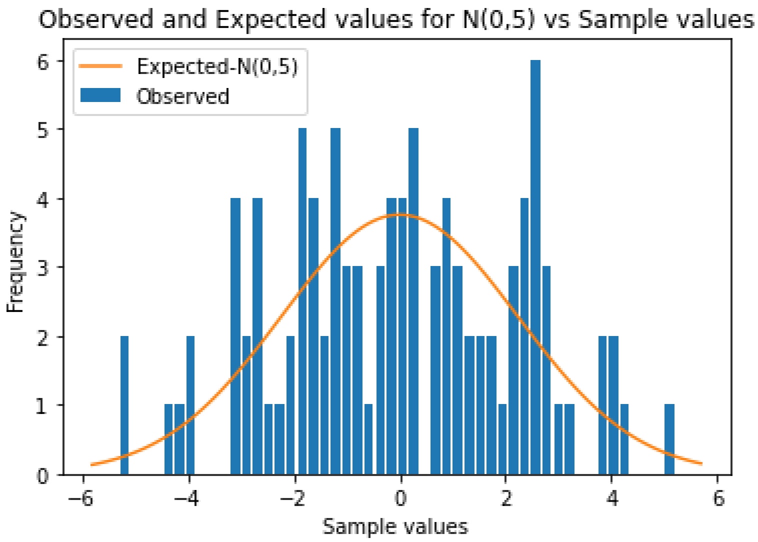
N(5,5) distribution-

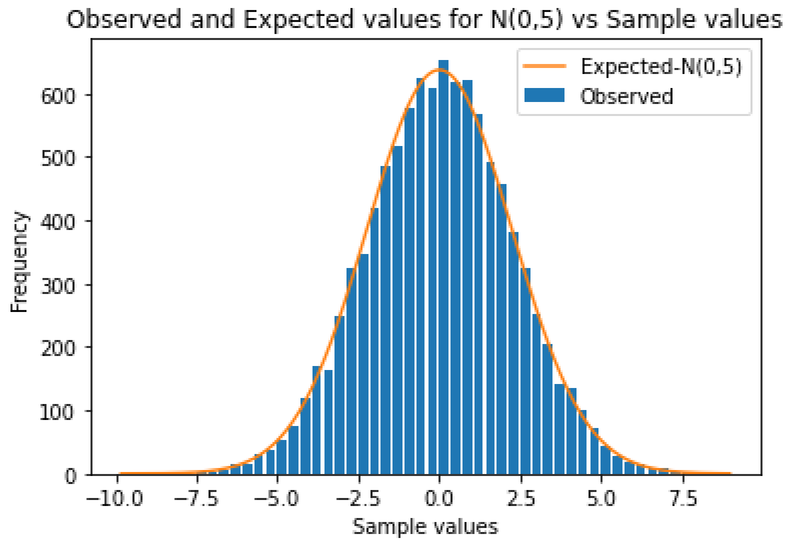




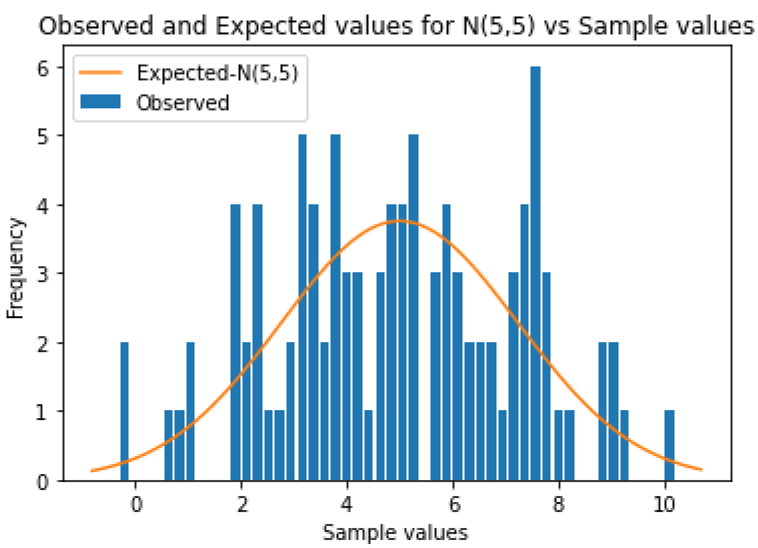
Marsaglia and Bray :

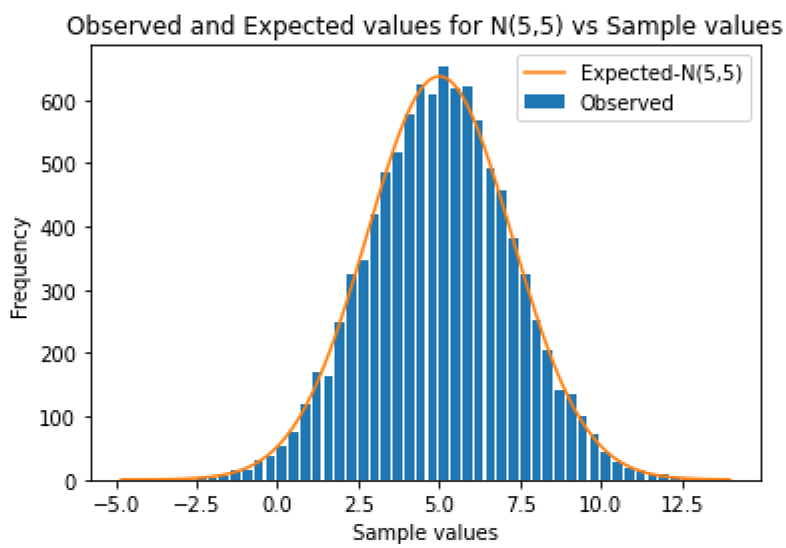
N(0,5) distribution-





N(5,5) distribution-





Question 2 :

The computation time is found out in both cases by using the time library and finding the **start** and **end** time in both cases (commented clearly in the code). We observe the following output :

```
Runtime of the program is 1.585282564163208 # for Box-Muller
Runtime of the program is 1.3668818473815918 # for Marsaglia and Bray
```

As is clearly seen, **Marsaglia and Bray** method takes a lot less time than Box-Muller and is therefore **faster**. This is because it saves computation time by avoiding the calculation of **cos** and **sin** values.

Question 3 :
As can be seen in the code's output we have found out the proportion of rejected values by **Marsaglia and Bray** method. The output (printed on terminal) is as follows:

```
For 100 values the rejected proportion is  0.24242424242424243
For 10000 values the rejected proportion is  0.2143923324691649
The value of 1-pi/4 is  0.21460183660255172
```

As is seen, the value of **rejected proportion approaches** $1-\pi/4$ **.**