Report

Question 1

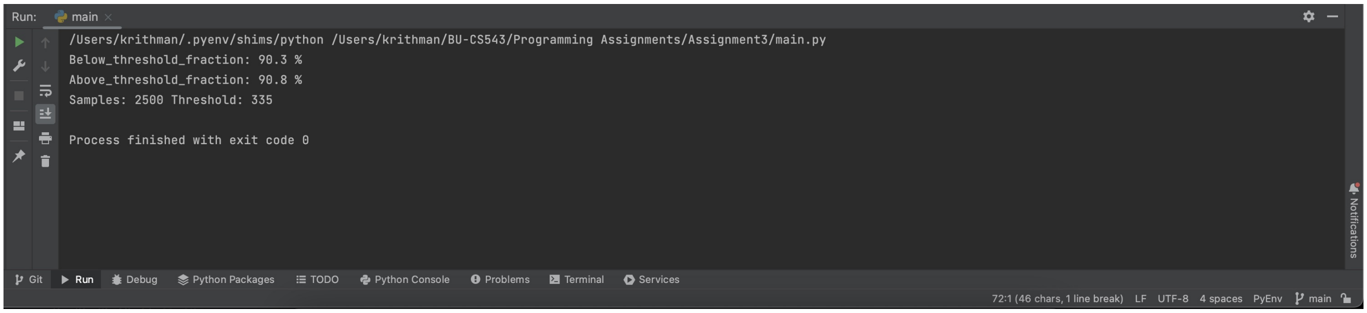
For the assignment purposes, I implemented a ϵ-far distribution from uniform in total variation distance named D_{far} as follows. Since n is even, half of elements will be chosen with probability $\frac{1-2*\epsilon}{n}$ and the rest of them with probability $\frac{1+2*\epsilon}{n}$. As a result, the total variation distance d_{TV} will be the following:

$$
\begin{aligned}
d_{TV}(U, D_{far}) &= \frac{1}{2} \cdot |\sum_{i=1}^{\frac{n}{2}} (\frac{1}{n} - \frac{1-2\cdot\epsilon}{n}) - \sum_{i=\frac{n}{2}}^{n} (\frac{1}{n} - \frac{1+2\cdot\epsilon}{n})| \Longrightarrow \\
&= \frac{1}{2} \cdot \sum_{i=1}^{\frac{n}{2}} |\frac{-2\cdot\epsilon}{n}| - \sum_{i=\frac{n}{2}}^{n} |\frac{2\cdot\epsilon}{n}| \Longrightarrow \\
&= \frac{1}{2} \cdot (\frac{2\cdot\epsilon}{n} \cdot \frac{n}{2} + \frac{\cdot\epsilon}{n} \cdot \frac{n}{2}) \Longrightarrow \\
&= \frac{1}{2} \cdot 2 \cdot \epsilon = \epsilon
\end{aligned}
$$

Proving that the D_{far} distribution is ϵ-far and given the two facts that first the collision probability of of any distribution that is ϵ-far from any uniform distribution U_n is greater than $\frac{1}{n} + \frac{4\cdot\epsilon^2}{n}$ and second that the uniform distribution minimizes the expected number of collisions then it is easily observable that the ϵ-far distribution minimizes the expected number of collisions since its collision probability is almost the same as the uniform one.

Question 2

For this question, I report the result of my uniformity test with parameters $n = 10000, \delta = 0.1 \& \epsilon = 0.2$. Towards finding the s and t parameters that satisfy the argument that the threshold t so that at least a 1δ fraction of collision numbers you observed for U are below the threshold and at least a 1δ fraction of collision numbers you observed for D_{far} are above the threshold, I observed the plot results until the requested $1 - \delta$ probability is guaranteed both for uniform and ϵ-far distribution. Figure 2 below presents the discrete collision distributions of U and D_{far} as well as the threshold t that guarantees the previous described argument as Figure 1 presents. In conclusion, the output parameters are $s = 2500$ and $t = 335$.



```
/Users/krithman/.pyenv/shims/python /Users/krithman/BU-CS543/Programming Assignments/Assignment3/main.py
Below_threshold_fraction: 90.3 %
Above_threshold_fraction: 90.8 %
Samples: 2500 Threshold: 335

Process finished with exit code 0
```

Figure 1: Uniformity test results between U and D_{far} Collision Distribution with $n = 10000$, $\epsilon = 0.2$ and $\delta = 0.1$. The output parameters are $s = 2500$ and $t = 335$.

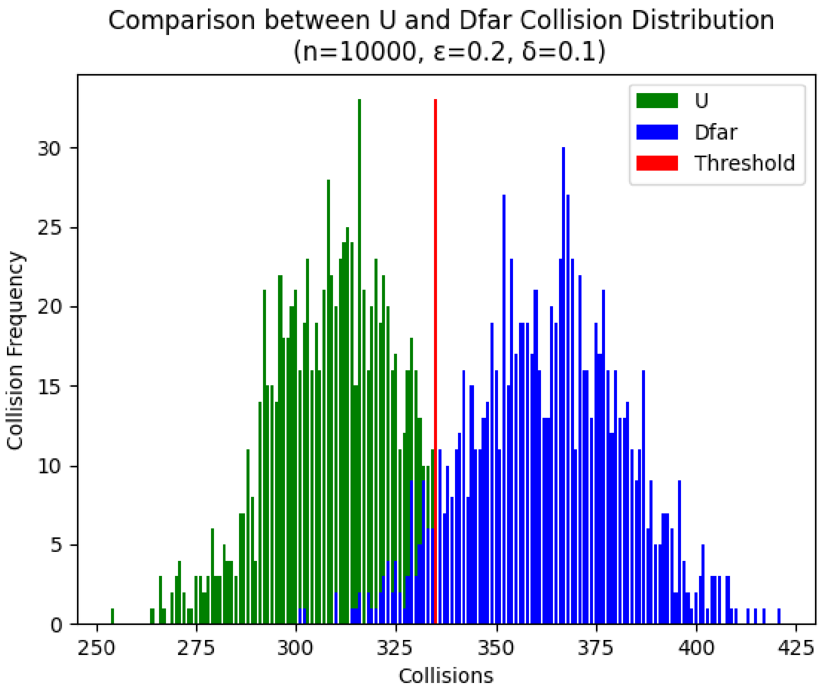


Figure 2: Comparison between U and D_{far} Collision Distribution with n = 10000, ϵ = 0.2 and δ = 0.1

Question 4

a)

I run my experiments with applying the uniformity test of question 2 to a real numeric dataset. This dataset describes the exchange rates between currencies and includes columns related to date, open value, high value, low value and close value. The link to this dataset is given here. The initial number of rows is 1048576 but I utilized only 10000 and worked with numeric data of column 'open'. In particular, I extract the second decimal and third decimal number from the dataset. I argue that the choice of those numerical data follow a uniform distribution as discussed on the first homework of the course with exchange currencies. Moreover, the way that the second and third decimal numbers are not dependent from previous dates so I state that those numbers are uniformly random extracted.

b)

Before applying the uniformity testing for single digits (aka the second decimal digits I extracted from sub question i)), I utilized the uniformity tester from question 2 with parameters $n = 10, \epsilon = 0.2$ & $\delta = 0.1$. My end goal is to find the appropriate threshold t and number of samples s such that later I will test the samples from the real exchange dataset to this uniformity tester with the above parameters. If the fraction of collisions for my real exchange dataset exceed the $1 - \delta$ of total collisions then the distribution of dataset samples is not uniform otherwise it is. Figure 3 presents the threshold between the uniform and ϵ-far distribution as well as states the number of samples we need to extract when the parameters are $n = 10, \epsilon = 0.2$ & $\delta = 0.1$(We pick $n = 10$ since we will sample single digits from real dataset). After extracting the appropriate s, t

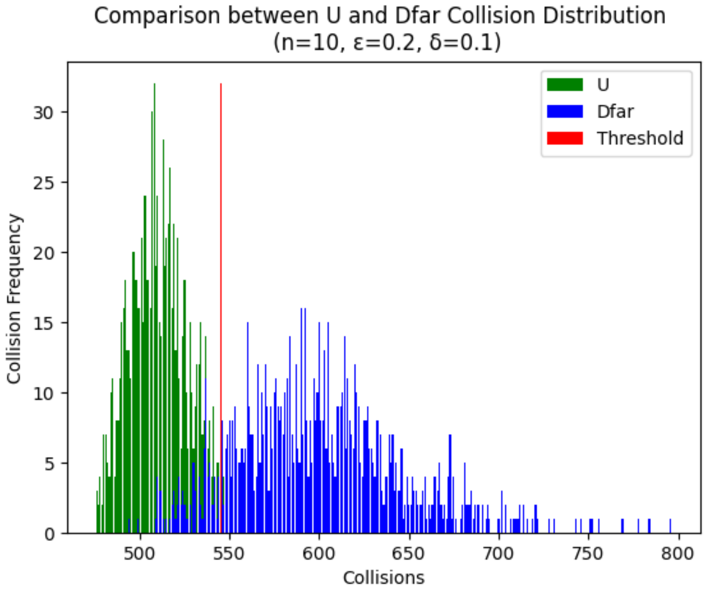


Figure 3: Comparison between U and D_{far} Collision Distribution with $n = 10$, $\epsilon = 0.2$ and $\delta = 0.1$. Output $s = 102$ and threshold $t = 545$.

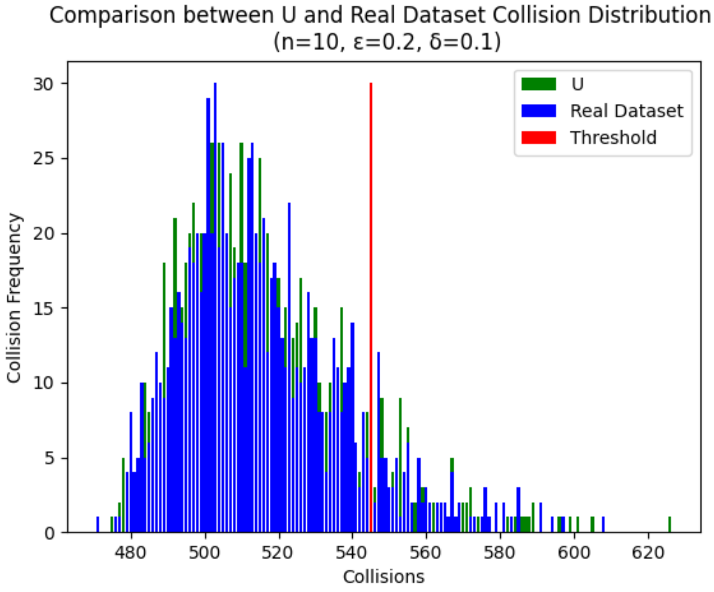
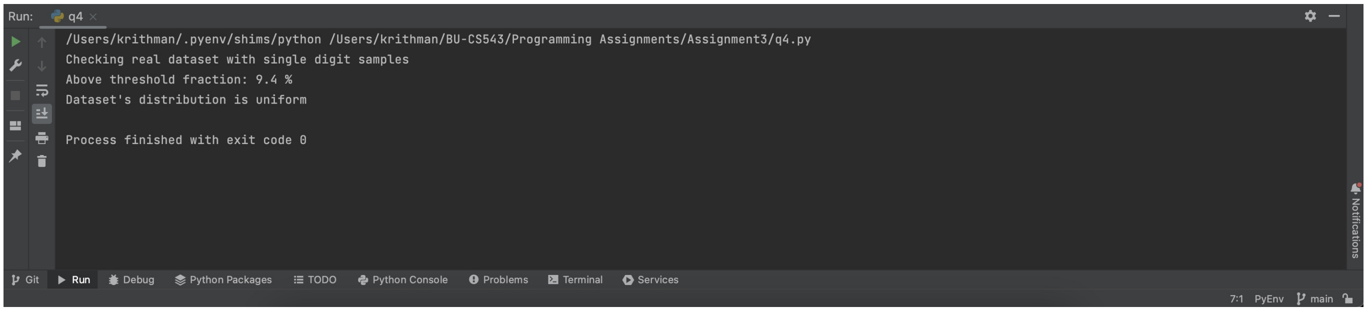


Figure 4: Comparison between U and single digit dataset Collision Distribution with $n = 10, \epsilon = 0.2, \delta = 0.1, s = 102$ and $t = 545$.

output variables for $n = 10, \epsilon = 0.2$ & $\delta = 0.1$ then I plug the real dataset samples. The output is given in the figure 4. As it is easily observable most of real dataset collisions fall under the threshold so the single digits



```
/Users/krithman/.pyenv/shims/python /Users/krithman/BU-CS543/Programming Assignments/Assignment3/q4.py
Checking real dataset with single digit samples
Above threshold fraction: 9.4 %
Dataset's distribution is uniform

Process finished with exit code 0
```

Figure 5: Fraction of single digit dataset collisions exceeds the threshold $t = 545$.

sequence of samples indicates that the dataset's distribution is uniform. Figure 5 shows that only 9.4% of the dataset collisions are above the threshold while we need at least 90% to declare it as non uniform.

c)

For pair of digits, we repeat the previous procedure but this time we take as sample input the pairs of the second and third decimal numbers of each open exchange value from the dataset. As before, I present the uniformity tester results from question 2 with parameters $n = 100, \epsilon = 0.2$ & $\delta = 0.1$. The output s, t variables are $s = 260$ and $t = 360$. Figure 6 presents the frequencies of collisions between uniform and ϵ-far distributions with the above parameters. We chose $n = 100$ since we are working with a pair of digits.

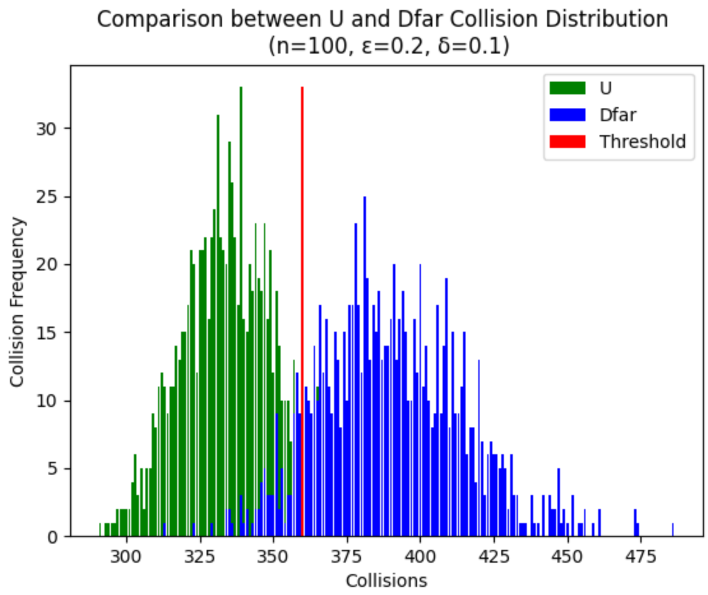


Figure 6: Comparison between U and D_{far} Collision Distribution with $n = 100$, $\epsilon = 0.2$ and $\delta = 0.1$. Output $s = 260$ and threshold $t = 360$.

After extracting the appropriate s, t output variables for $n = 100, \epsilon = 0.2$ & $\delta = 0.1$ then I plug the real dataset samples. The output is given in the figure 7. As it is easily observable most of real dataset collisions fall under the threshold so the pair digits sequence of samples indicates that the dataset's distribution is uniform. Figure 8 shows that only 10.9% of the dataset collisions are above the threshold while we need at least 90% to declare it as non uniform.

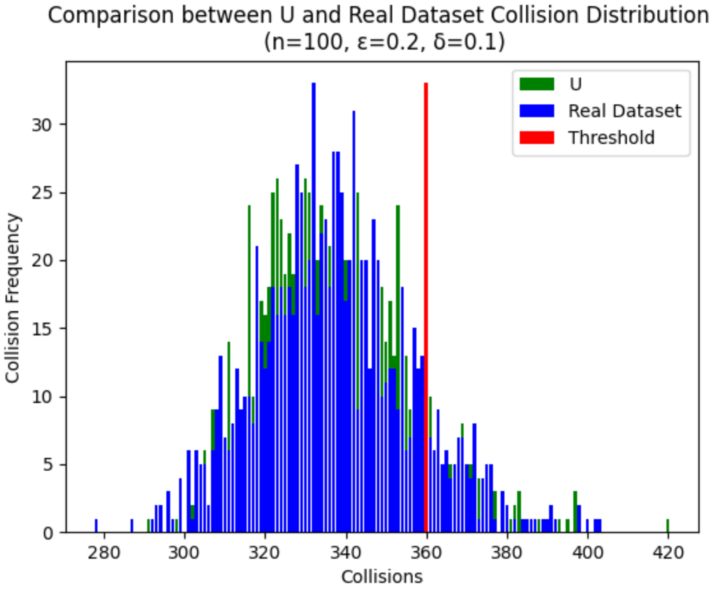
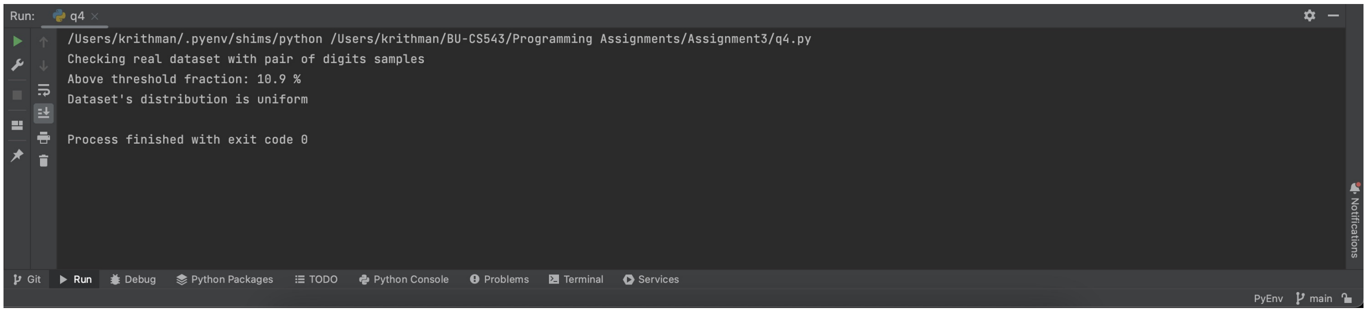


Figure 7: Comparison between U and pair digits dataset Collision Distribution with $n = 100, \epsilon = 0.2, \delta = 0.1, s = 260$ and $t = 360$.



```
/Users/krithman/.pyenv/shims/python /Users/krithman/BU-CS543/Programming Assignments/Assignment3/q4.py
Checking real dataset with pair of digits samples
Above threshold fraction: 10.9 %
Dataset's distribution is uniform

Process finished with exit code 0
```

Figure 8: Fraction of pair digits dataset collisions exceeds the threshold $t = 360$.

Question 5

Before answering this question, I followed the hint and created a sequence of 10000 samples, each of which is built after summing all q randomly created digits and in succession we output the result of the sum modulo 10. The sequence of 10000 elements are stored into txt files, each one having a different q as input parameter. In particular, I created txt files with $q = 10, 100, 1000, 10000$ respectively. Moreover, for the experiments that I run, I utilized the uniformity tester of question 2 with specific n, ϵ & δ for single digits, pairs and triples as required from sub questions a) and b). The sub figures below represent the s, t for single digits, pair and triples digits as well as with their input parameters n, ϵ & δ.

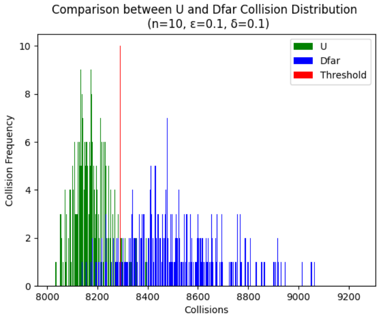


Figure 9: Comparison between U and D_{far} Collision Distribution with $n = 10$, $\epsilon = 0.1$ and $\delta = 0.1$. Output $s = 405$ and threshold $t = 8291$.

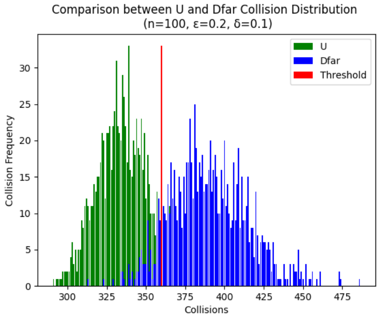


Figure 10: Comparison between U and D_{far} Collision Distribution with $n = 100$, $\epsilon = 0.2$ and $\delta = 0.1$. Output $s = 260$ and threshold $t = 360$.

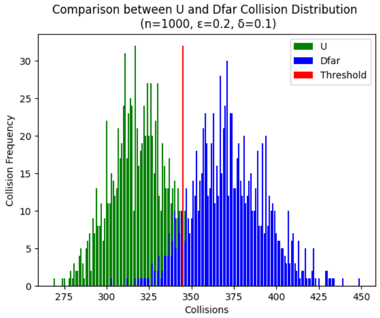


Figure 11: Comparison between U and D_{far} Collision Distribution with $n = 1000$, $\epsilon = 0.2$ and $\delta = 0.1$. Output $s = 800$ and threshold $t = 345$.

a)

The figures 12 below present the results after running the uniformity tester with $q = 1, 10, 100, 1000$ and 10000 respectively. All the above choices gave me uniform distributions as presented at the figures. The lowest $q = 1$ for which you can pass the uniformity test designed for small $\epsilon = 0.1$ and $\delta = 0.1$.

b)

The figures 13, 14 below present the results after running the uniformity tester with $q = 1, 10, 100, 1000$ and 10000 for pairs and triples respectively. All the above choices gave me uniform distributions as presented at the figures. Compared to the single digit case, both pass the uniformity tester with same q. I suspect that the reason behind that is that the random digit generation from the numpy.random.randint function produces uniform integers and that is the reason behind the uniform results I got at the end.

Code Reproducibility

The code was written and tested on Ubuntu 20.04 operating system, with Python 3.9 installed. The only python packages required to be installed to run the code are the following: *numpy, pandas, matplotlib*. To install them under pip you need to follow the command:

- pip install numpy pandas matplotlib

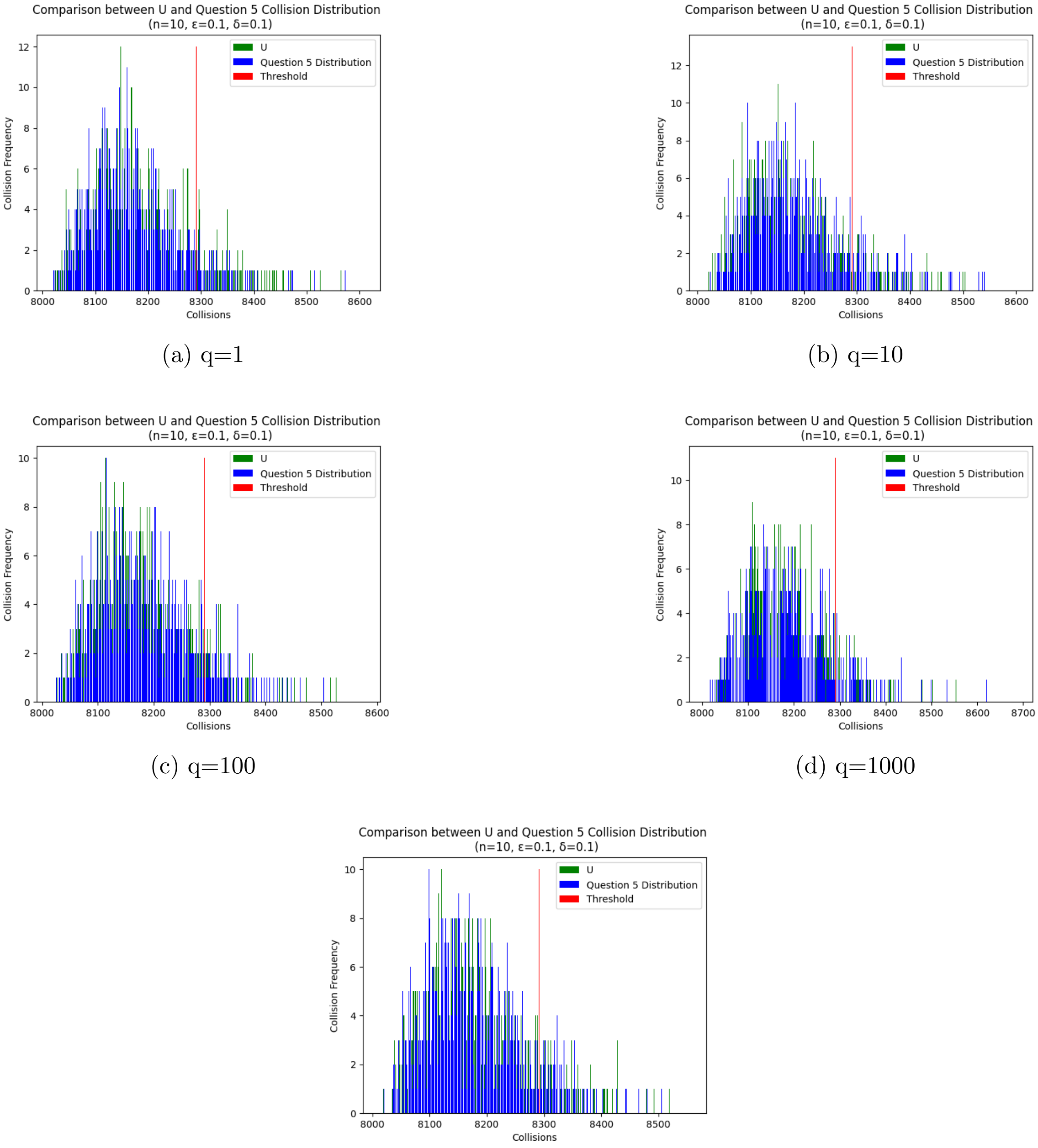
After unzipping the .zip folder and accessing it, the following lines provide you a description of how to run the subquestions of the exercise.

For question 2), please run the *q2* script as follows: **python q2.py**

For question 4), please run the *q4* script as follows: **python q4.py**

For question 5), please run the *q5* script as follows: **python q5.py -q X**. X could be any integer number. Here is an example of running question 5: *python q5.py -q 100.*

All the plots will be generated under the plots folder.



(a) q=1

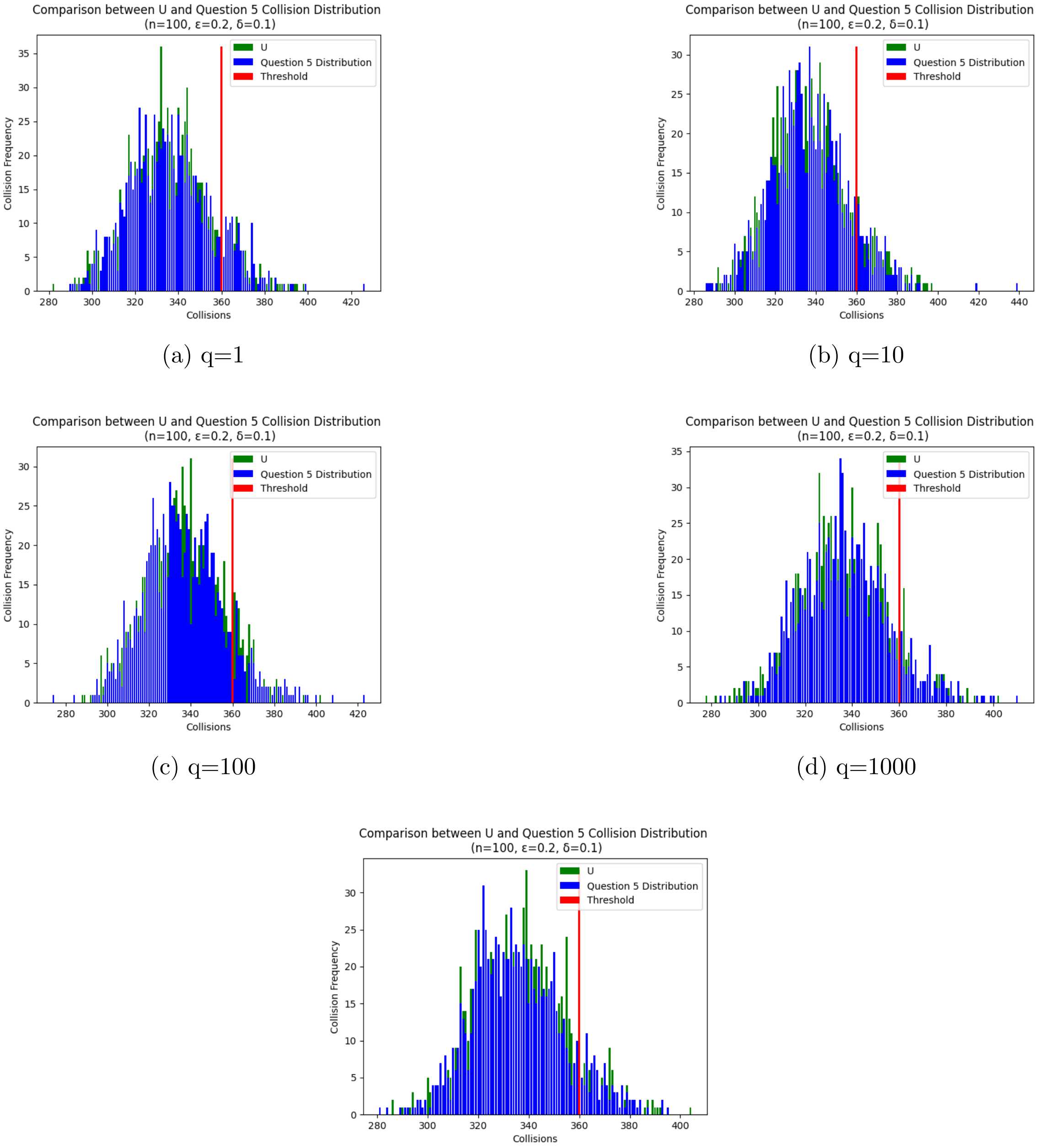
(b) q=10

(c) q=100

(d) q=1000

(e) q=10000

Figure 12: Results Question 5a with parameter $q = 1, 10, 100, 1000, 10000$ respectively. All of them output uniform distribution as a final result.



(a) q=1

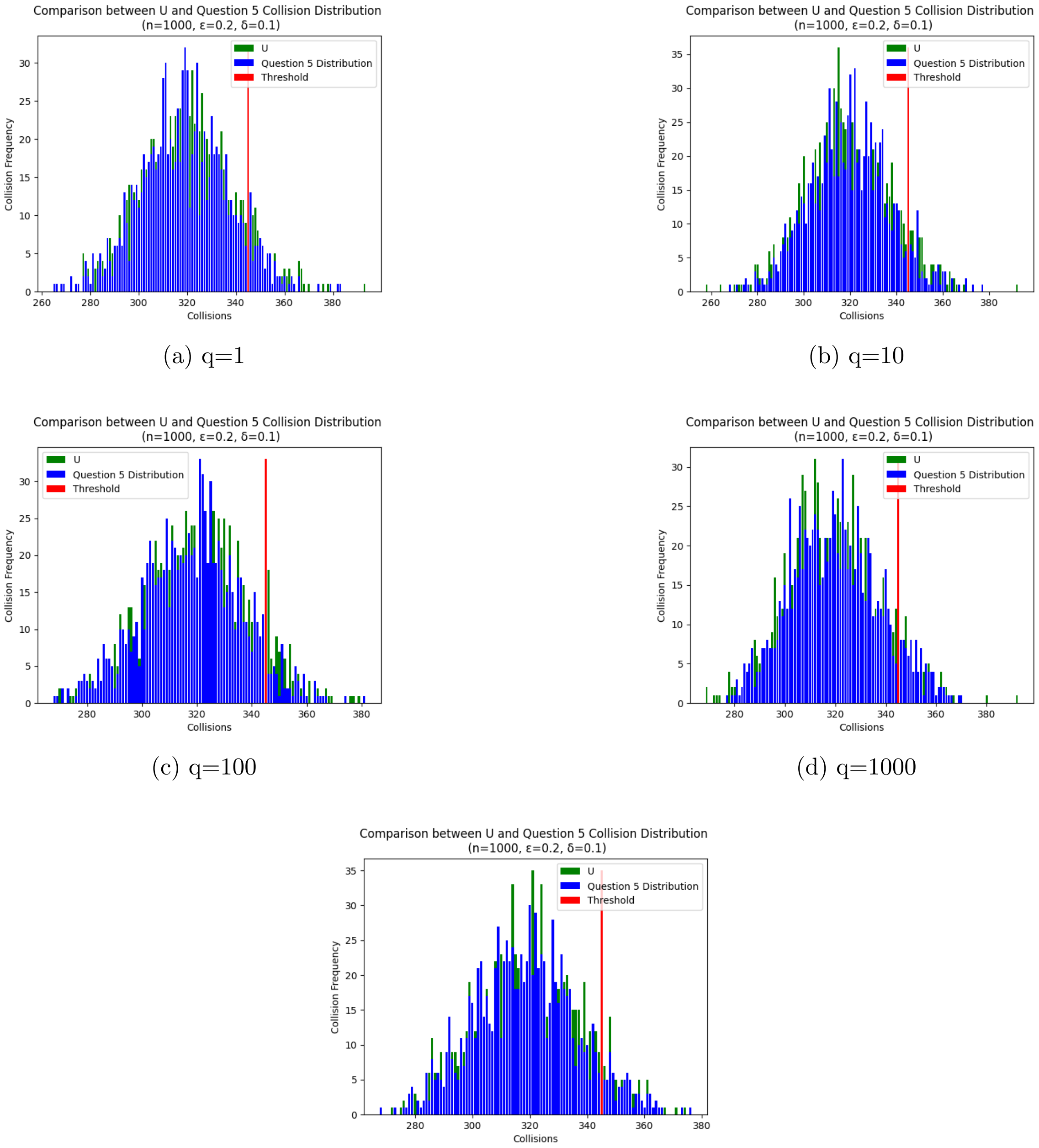
(b) q=10

(c) q=100

(d) q=1000

(e) q=10000

Figure 13: Results Question 5b pair digits with parameter $q = 1, 10, 100, 1000, 10000$ respectively. All of them output uniform distribution as a final result.



(a) q=1

(b) q=10

(c) q=100

(d) q=1000

(e) q=10000

Figure 14: Results Question 5b triple digits with parameter $q = 1, 10, 100, 1000, 10000$ respectively. All of them output uniform distribution as a final result.