How to execute the code:

1. To get the frequent patterns along with their support count in a file import our package and call the method **'storePatternsInFile'** using the following command:

   ```
   import Apriori as Myap
   Myap.storePatternsInFile(r"filepath or filename", listofminimumsupport)
   ```

   output is displayed as follows:

   - Total frequent itemsets are: number of frequent itemsets
   - All the Frequent itemsets will be stored in a file named as "outputminimumsupport"

   For example:
   If we execute the following command:

   ```
   import Apriori as Myap
   Myap.storePatternsInFile(r" transactional_T10I4D100K.csv", [0.01])
   ```

   output is displayed as follows:

   - Total frequent itemsets are: 385
   - All the Frequent itemsets will be stored in a file named as "output1000.0"

2. To get the frequent patterns along with their support count in a DataFrame import our package and call the function **'getPatternsInDf'** using the following command:

   ```
   import Apriori as Myap
   variable = Myap.getPatternsInDf(r"filepath or filename", listofminimumsupport)
   ```

   output is displayed as follows:

   - Total frequent itemsets are: number of frequent itemsets
   - All the Frequent itemsets will be stored in a Data Frame and returned to the called function.

   For example:
   If we execute the following command:

   ```
   import Apriori as Myap
   DF = Myap. getPatternsInDf(r" transactional_T10I4D100K.csv", [0.01])
   ```

output is displayed as follows:

- Total frequent itemsets are: 385
- All the Frequent itemsets will be stored in the variable '**DF**'

3. If we want to know the memory consumed by the apriori algorithm created by us then use the method **'getMemory'** along with any one of the above two methods (,i.e storePatternsInFile or getPatternsInDf):

```
import Apriori as Myap
Myap.storePatternsInFile(r"filepath or filename", listofminimumsupport)
Myap.getMemory()
```

output is displayed as follows:

- Total frequent itemsets are: number of frequent itemsets
- Memory details: minimum support : memory consumed by the program in MB
- All the Frequent itemsets will be stored in a file named as "outputminimumsupport"

For example:
If we execute the following command:

```
import Apriori as Myap
Myap.storePatternsInFile(r" transactional_T10I4D100K.csv", [0.01])
Myap.getMemory()
```

output is displayed as follows:

- Total frequent itemsets are: 385
- Memory details:
  1000.0 : 226.33203125
- All the Frequent itemsets will be stored in a file named as "output1000.0"

Note: In the above example, you may also use getPatternsInDf method in the place of storePatternsInFile

4. If we want to know the execution time taken by the program then use the method **'getRunTime'** along with any one of the above two methods (,i.e storePatternsInFile or getPatternsInDf):

```
import Apriori as Myap
Myap.storePatternsInFile(r"filepath or filename", listofminimumsupport)
Myap.getRunTime()
```

output is displayed as follows:

- Total frequent itemsets are: number of frequent itemsets
- Execution Time: minimum support : total time taken to execute our code in seconds
- All the Frequent itemsets will be stored in a file named as "outputminimumsupport"

For example:
If we execute the following command:

```
import Apriori as Myap
Myap.storePatternsInFile(r" transactional_T10I4D100K.csv", [0.01])
Myap.getRunTime()
```

output is displayed as follows:

- Total frequent itemsets are: 385
- Execution Time:
  1000.0 : 22.613293170928955
- All the Frequent itemsets will be stored in a file named as "output1000.0"

Note: In the above example, you may also use getPatternsInDf method in the place of storePatternsInFile

5. If we want to know memory details and execution time taken by the program then use both methods at a time:

```
import Apriori as Myap
Myap.storePatternsInFile(r"filepath or filename", listofminimumsupport)
Myap.getMemory()
Myap.getRunTime()
```

output is displayed as follows:

- Total frequent itemsets are: number of frequent itemsets
- Memory details: minimum support : memory consumed by the program in MB
- Execution Time: minimum support : total time taken to execute our code in seconds
- All the Frequent itemsets will be stored in a file named as "outputminimumsupport"

For example:
If we execute the following command:

```
import Apriori as Myap
Myap.storePatternsInFile(r" transactional_T10I4D100K.csv", [0.01])
Myap.getMemory()
Myap.getRunTime()
```

output is displayed as follows:

- Total frequent itemsets are: 385
- Memory details:
  1000.0 : 226.33203125
- Execution Time:
  1000.0 : 22.613293170928955
- All the Frequent itemsets will be stored in a file named as "output1000.0"

Note: In the above example, you may also use getPatternsInDf method in the place of storePatternsInFile

6. If we want to check the statistics of the input file or database then call the method **'getStatsInFile'** using the following command:

```
import Apriori as Myap
Myap.storePatternsInFile(r"filepath or filename", listofminimumsupport)
Myap.getStatsInFile()
```

output is displayed as follows:

- Total frequent itemsets are: number of frequent itemsets
- All the Frequent itemsets will be stored in a file named as "outputminimumsupport"
- Complete statistics of the input file or input database such as Total number of transactions, items, minimum length of the transactions, maximum length of the transactions, and average length of the transactions will be stored in a file named as "Statistics"

For example:
If we execute the following command:

```
import Apriori as Myap
Myap.storePatternsInFile(r" transactional_T10I4D100K.csv", [0.01])
Myap.getStatsInFile()
```

output is displayed as follows:

- Total frequent itemsets are: 385
- All the Frequent itemsets will be stored in a file named as "output1000.0"
- All the statistics of the input file or input database is stored in a file named as "Statistics"