Description

1. Implementation of the algorithm
   I implement the SON algorithm using A-Priori in order to process each chunk. In each chunk, I count the number of baskets of this chunk, and then get the ratio p = chunk size/ total size, then use the new support = p*s. Then I use the monotonicity of the itemsets to get the all the frequent candidates and pass them to the next count step. In the second step, map process will count support for the candidates in each chunk, and then in the reduce process will add the support for each candidate, and decide to keep this candidate based on s. After filtering, we will have hall the frequent itemsets.

2. Command lines
   Version: Scala: 2.11 Spark: 2.2.1, Under the folder "spark-2.2.1-bin-hadoop2.7", there should be all the 4 data csv files, and the Shiwei_Huang_SON.jar file. The commands for each problem are as follows:

   Problem 1:

```
 bin/spark-submit --driver-memory 2g --class Son Shiwei_Huang_Son.jar 1
small2.csv 3
 bin/spark-submit --driver-memory 2g --class Son Shiwei_Huang_Son.jar 2
small2.csv 5
```

   Problem 2:

```
 bin/spark-submit --driver-memory 2g --class Son Shiwei_Huang_Son.jar 1
beauty.csv 50
 bin/spark-submit --driver-memory 2g --class Son Shiwei_Huang_Son.jar 2
beauty.csv 40
 bin/spark-submit --driver-memory 2g --class Son Shiwei_Huang_Son.jar 1 books.csv
1200
 bin/spark-submit --driver-memory 2g --class Son Shiwei_Huang_Son.jar 2 books.csv
1500
```

3. Problem 2 Execution Table

File Name	Case Number	Support	Runtime(sec)
beauty.csv	1	50	98
beauty.csv	2	40	74
books.csv	1	1200	241
books.csv	2	1500	40