Fukushima_Congestion_experiments-Copy3

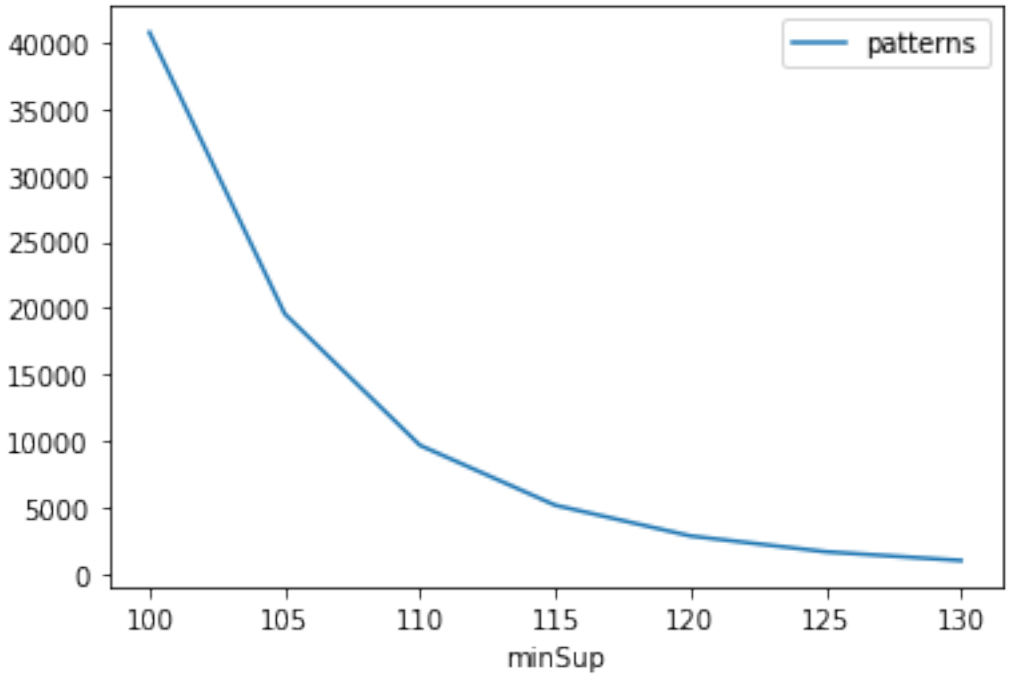
June 3, 2023

[1]:
```
import FFI_newVersion_Congestion  as alg
import pandas as pd
inputFile = 'updated_congestion_5.txt'

minimumSupportCountList = [100, 105, 110, 115, 120, 125, 130]  #Users can also␣
 ↪specify this constraint between 0 to 1.
seperator = '\t'
result = pd.DataFrame(columns=['algorithm', 'minSup', 'patterns', 'runtime',␣
 ↪'memory'])
#initialize a data frame to store the results of FFIMiner algorithm

algorithm = 'FFI'  #specify the algorithm name
for minSupCount in minimumSupportCountList:
    obj = alg.FFIMiner(iFile=inputFile, minSup=minSupCount, sep=seperator)
    obj.startMine()
    obj.save('congestion_patterns.txt')
    #store the results in the data frame
    result.loc[result.shape[0]] = [algorithm, minSupCount, len(obj.
 ↪getPatterns()), obj.getRuntime(), obj.getMemoryRSS()]

print(result)
```

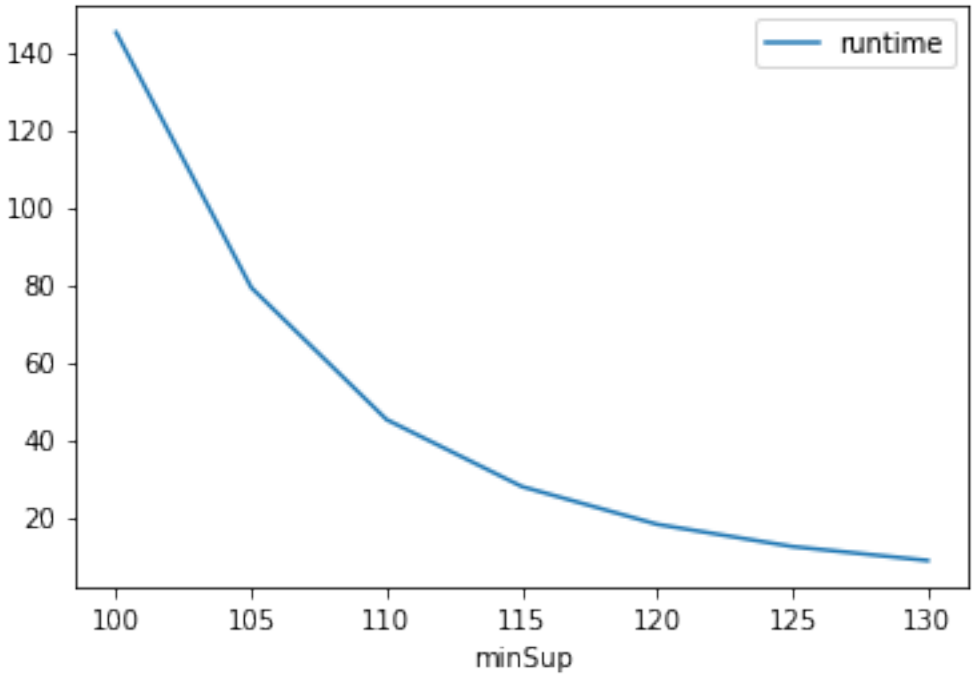
```
1461
11019
1461
11019
1461
11019
1461
11019
1461
11019
1461
11019
1461
11019
1461
11019
  algorithm  minSup  patterns     runtime     memory
0       FFI     100     40764  145.049374  231223296
```

1	FFI	105	19580	79.128715	223105024
2	FFI	110	9680	45.062253	219099136
3	FFI	115	5170	27.780133	216236032
4	FFI	120	2855	18.029947	215154688
5	FFI	125	1666	12.292263	214167552
6	FFI	130	1007	8.683164	213069824

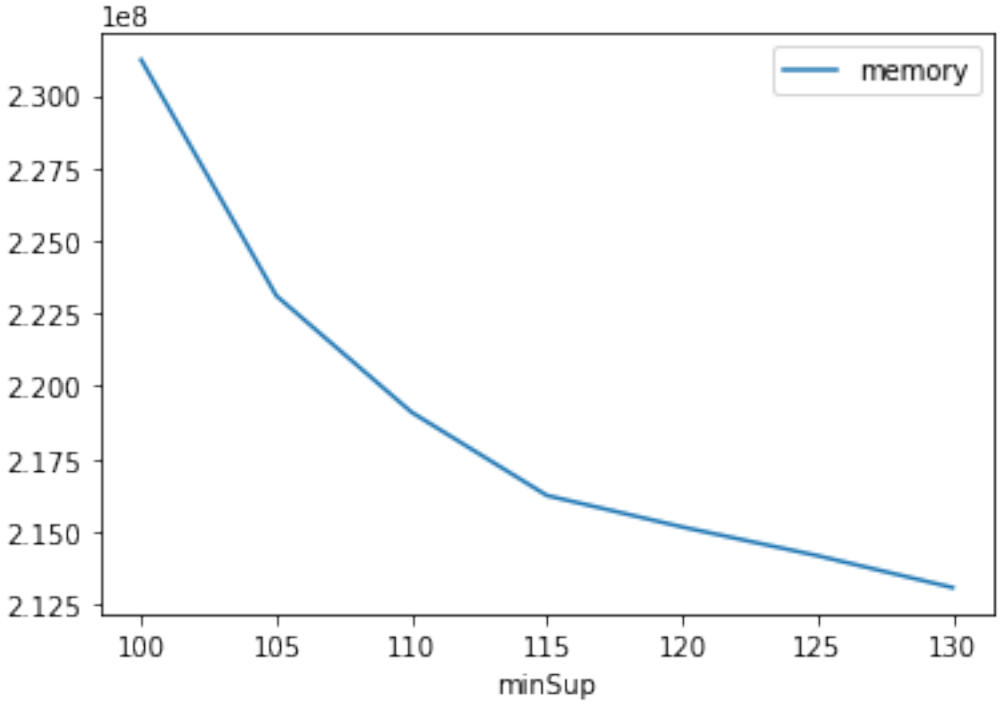
[2]:
```
from PAMI.extras.graph import plotLineGraphsFromDataFrame as plt
ab = plt.plotGraphsFromDataFrame(result)
ab.plotGraphsFromDataFrame() #drawPlots()
```




```
Graph for No Of Patterns is successfully generated!
```




```
Graph for Runtime taken is successfully generated!
```



```
Graph for memory consumption is successfully generated!
```

[3]:
```
from PAMI.extras.graph import generateLatexFileFromDataFrame as gdf
gdf.generateLatexCode(result)
```

```
Latex files generated successfully
```

[ ]:
```

```