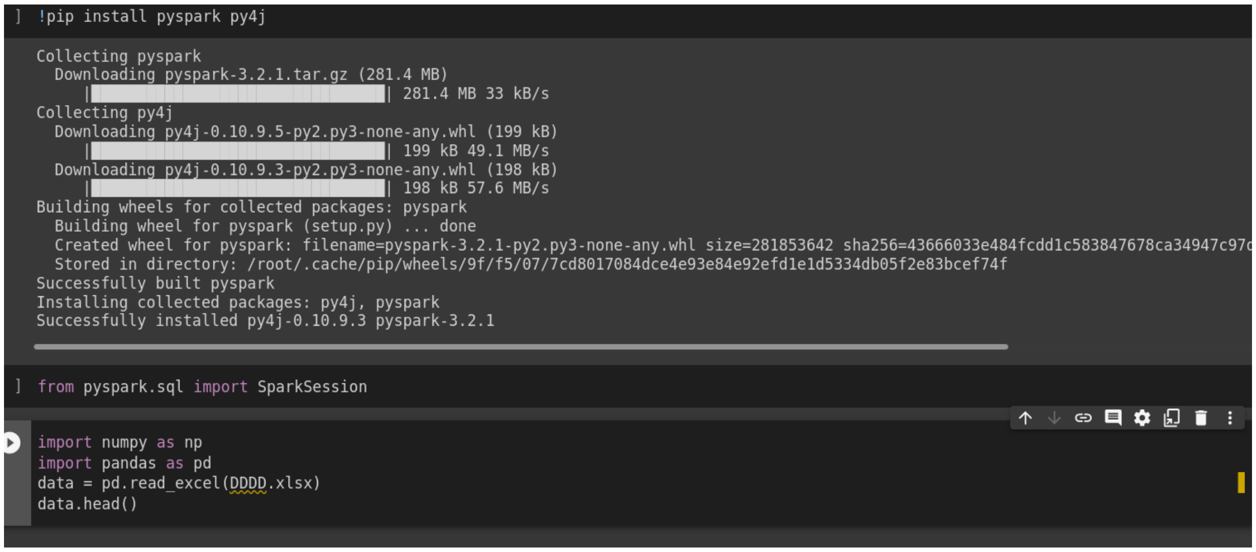
Paleti Krishnasai
CED18I039

ASBD Problem Set 8

5. Generate FIMs using FPgrowth algorithm on pyspark setup using a benchmark dataset available on the FIMI website.



```
] !pip install pyspark py4j

  Collecting pyspark
    Downloading pyspark-3.2.1.tar.gz (281.4 MB)
       |████████████████████████████████| 281.4 MB 33 kB/s
  Collecting py4j
    Downloading py4j-0.10.9.5-py2.py3-none-any.whl (199 kB)
       |████████████████████████████████| 199 kB 49.1 MB/s
    Downloading py4j-0.10.9.3-py2.py3-none-any.whl (198 kB)
       |████████████████████████████████| 198 kB 57.6 MB/s
  Building wheels for collected packages: pyspark
    Building wheel for pyspark (setup.py) ... done
    Created wheel for pyspark: filename=pyspark-3.2.1-py2.py3-none-any.whl size=281853642 sha256=43666033e484fcdd1c583847678ca34947c97c
    Stored in directory: /root/.cache/pip/wheels/9f/f5/07/7cd8017084dce4e93e84e92efd1e1d5334db05f2e83bcef74f
  Successfully built pyspark
  Installing collected packages: py4j, pyspark
  Successfully installed py4j-0.10.9.3 pyspark-3.2.1

] from pyspark.sql import SparkSession

  import numpy as np
  import pandas as pd
  data = pd.read_excel(DDDD.xlsx)
  data.head()
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
spark = SparkSession
\.builder
\.appName('fpgrowth')
\.getOrCreate()spark
```

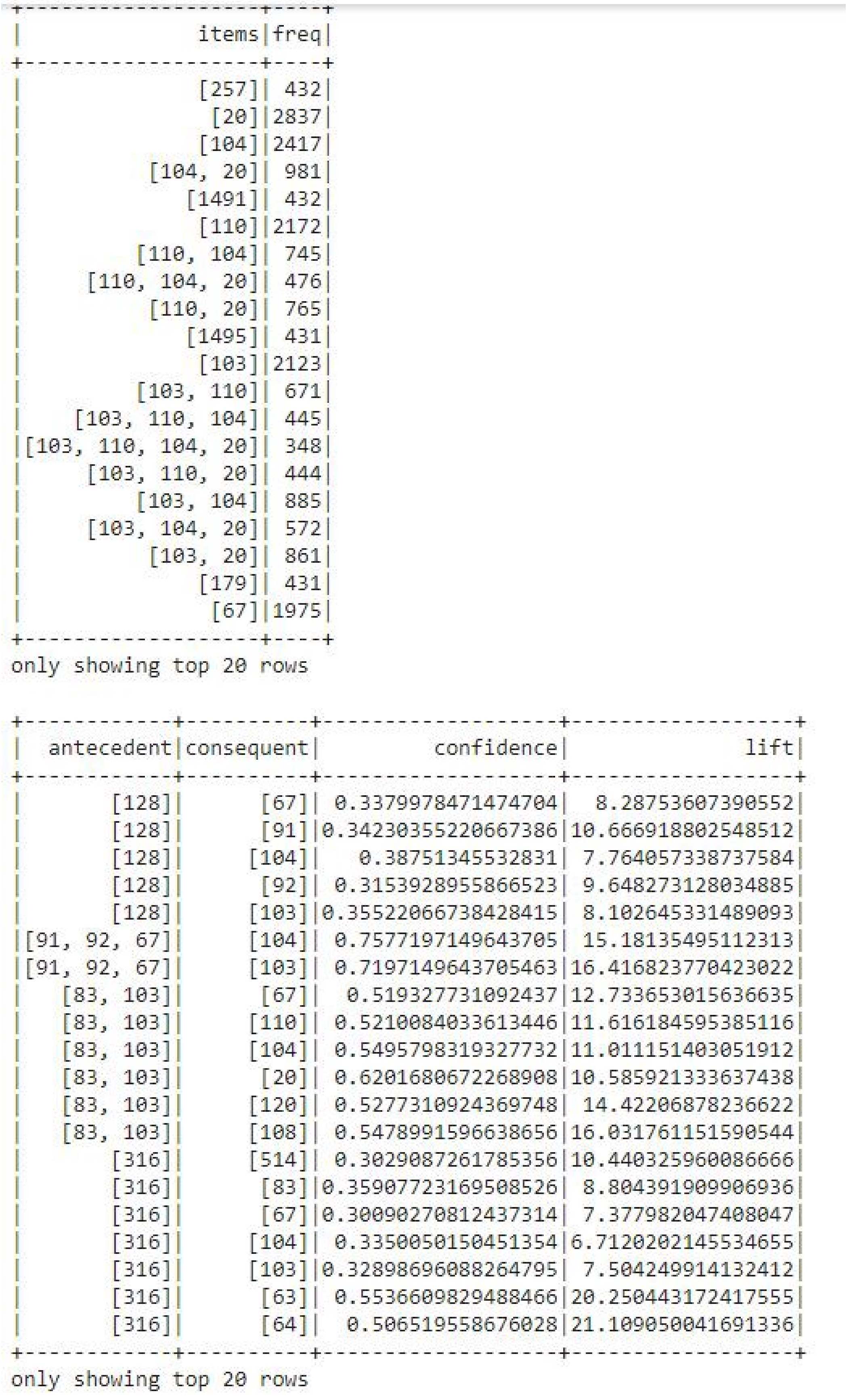
```
from pyspark.sql import functions as F
from pyspark.ml.fpm import FPGrowth
import pandas
sparkdata = spark.createDataFrame(data)
```

```
basketdata = sparkdata.dropDuplicates(['SalesTransactionID',
'SalesItem']).sort('SalesTransactionID')
```

Paleti Krishnasai
CED18I039

```
basketdata =
basketdata.groupBy("SalesTransactionID").agg(F.collect_list("SalesItem")).
sort('SalesTransactionID')
```

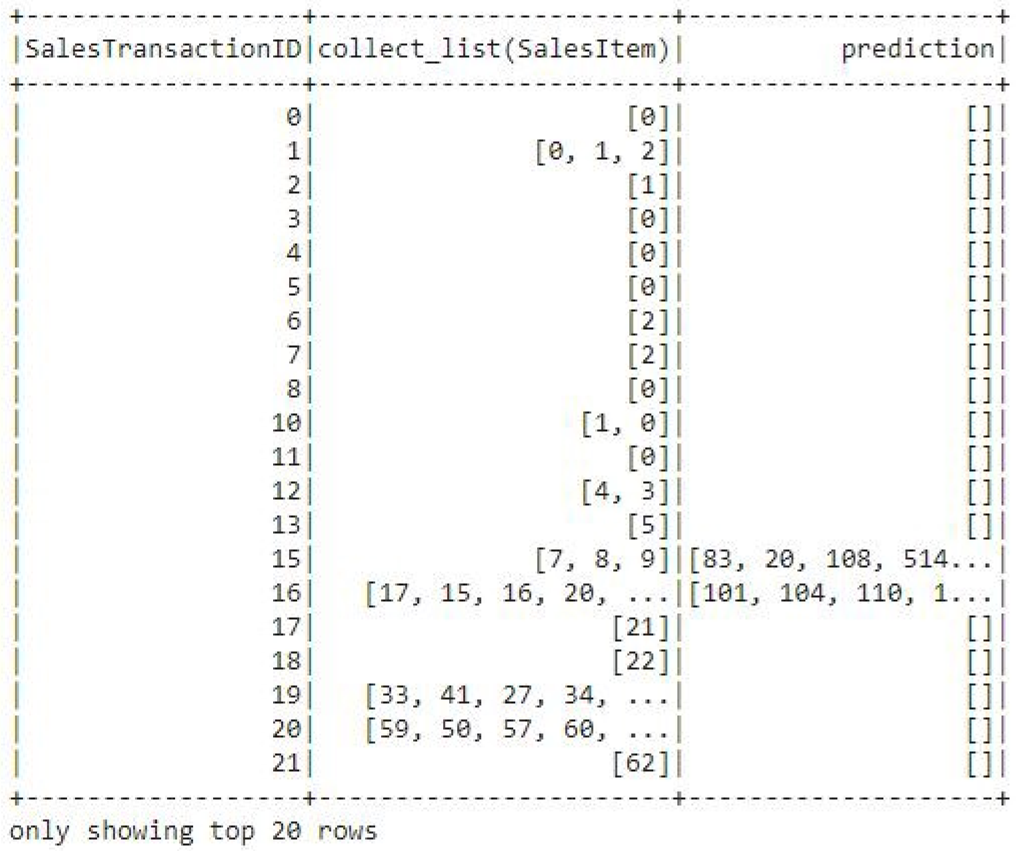
Paleti Krishnasai
CED18I039



```
+-------------------+----+
|              items|freq|
+-------------------+----+
|              [257]| 432|
|               [20]|2837|
|              [104]|2417|
|          [104, 20]| 981|
|             [1491]| 432|
|              [110]|2172|
|         [110, 104]| 745|
|     [110, 104, 20]| 476|
|          [110, 20]| 765|
|             [1495]| 431|
|              [103]|2123|
|         [103, 110]| 671|
|    [103, 110, 104]| 445|
|[103, 110, 104, 20]| 348|
|     [103, 110, 20]| 444|
|         [103, 104]| 885|
|     [103, 104, 20]| 572|
|          [103, 20]| 861|
|              [179]| 431|
|               [67]|1975|
+-------------------+----+
only showing top 20 rows
```

```
+------------+----------+------------------+------------------+
|  antecedent|consequent|        confidence|              lift|
+------------+----------+------------------+------------------+
|       [128]|      [67]| 0.337997847147470| 8.28753607390552|
|       [128]|      [91]|0.3423035522066738|10.66691880254851|
|       [128]|     [104]|   0.38751345532831| 7.764057338737584|
|       [128]|      [92]| 0.315392895586652| 9.648273128034885|
|       [128]|     [103]|0.3552206673842841| 8.102645331489093|
|[91, 92, 67]|     [104]| 0.757719714964370| 15.18135495112313|
|[91, 92, 67]|     [103]| 0.719714964370546|16.416823770423022|
|   [83, 103]|      [67]|  0.51932773109243|12.733653015636635|
|   [83, 103]|     [110]| 0.521008403361344|11.616184595385116|
|   [83, 103]|     [104]| 0.549579831932773|11.011151403051912|
|   [83, 103]|      [20]| 0.620168067226890|10.585921333637438|
|   [83, 103]|     [120]| 0.527731092436974| 14.42206878236622|
|   [83, 103]|     [108]| 0.547899159663865|16.031761151590544|
|       [316]|     [514]| 0.302908726178535|10.440325960086666|
|       [316]|      [83]|0.3590772316950852| 8.804391909906936|
|       [316]|      [67]|0.3009027081243731| 7.377982047408047|
|       [316]|     [104]| 0.335005015045135|6.7120202145534655|
|       [316]|     [103]|0.3289869608826479| 7.504249914132412|
|       [316]|      [63]| 0.553660982948846|20.250443172417555|
|       [316]|      [64]|  0.50651955867602|21.109050041691336|
+------------+----------+------------------+------------------+
only showing top 20 rows
```

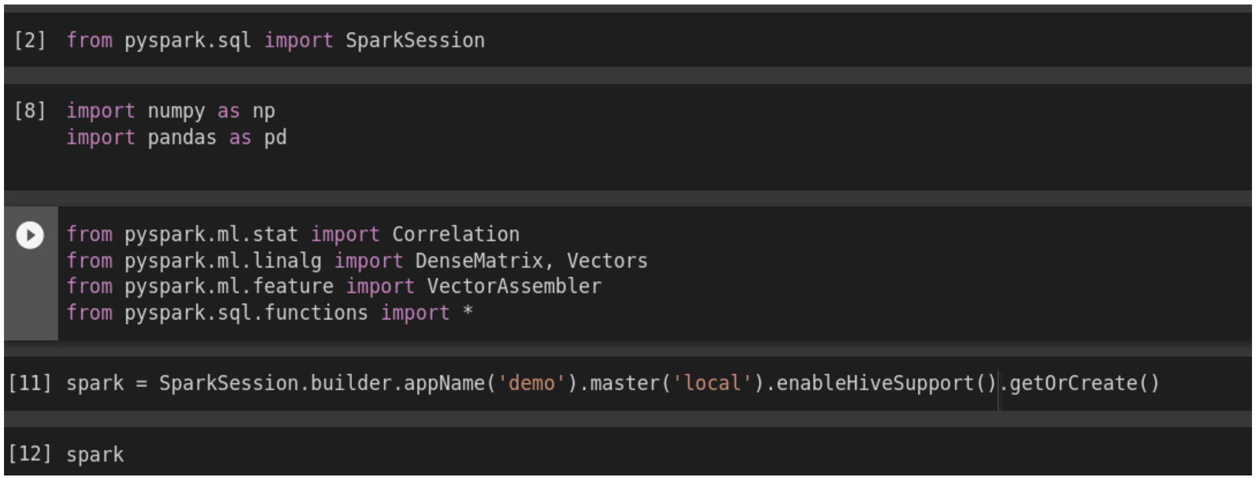
Paleti Krishnasai
CED18I039



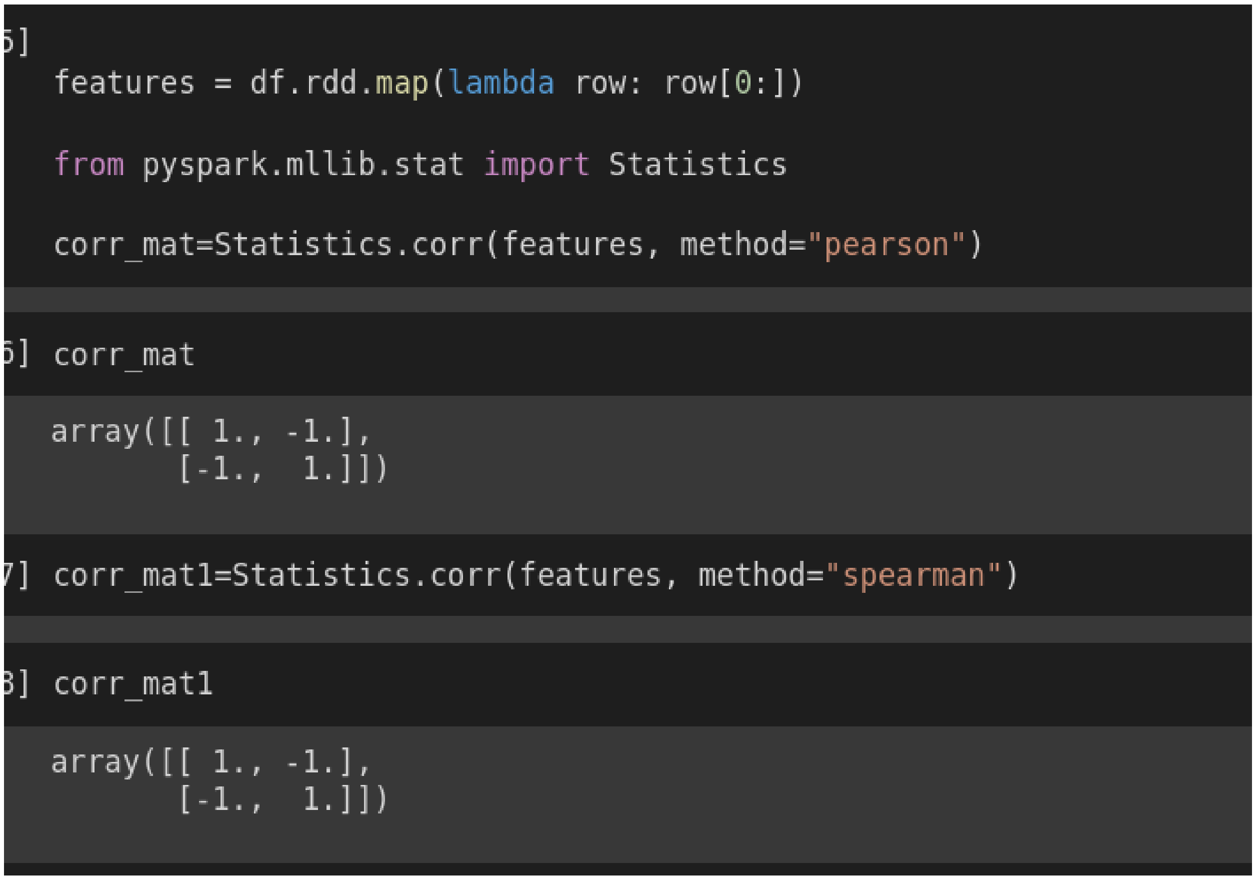
```
+------------------+---------------------+--------------------+
|SalesTransactionID|collect_list(SalesItem)|          prediction|
+------------------+---------------------+--------------------+
|                 0|                  [0]|                  []|
|                 1|            [0, 1, 2]|                  []|
|                 2|                  [1]|                  []|
|                 3|                  [0]|                  []|
|                 4|                  [0]|                  []|
|                 5|                  [0]|                  []|
|                 6|                  [2]|                  []|
|                 7|                  [2]|                  []|
|                 8|                  [0]|                  []|
|                10|               [1, 0]|                  []|
|                11|                  [0]|                  []|
|                12|               [4, 3]|                  []|
|                13|                  [5]|                  []|
|                15|            [7, 8, 9]|[83, 20, 108, 514...|
|                16| [17, 15, 16, 20, ...|[101, 104, 110, 1...|
|                17|                 [21]|                  []|
|                18|                 [22]|                  []|
|                19| [33, 41, 27, 34, ...|                  []|
|                20| [59, 50, 57, 60, ...|                  []|
|                21|                 [62]|                  []|
+------------------+---------------------+--------------------+
only showing top 20 rows
```

Paleti Krishnasai
CED18I039

3. Compute correlation between the given two series using Pearson’s and Spearman’s Method.
a. (Use the Spark MLlib libraries and helper functions available)
i. Series A: 35, 23, 47, 17, 10, 43, 9, 6, 28
ii. Series B: 30, 33, 45, 23, 8, 49, 12, 4, 31



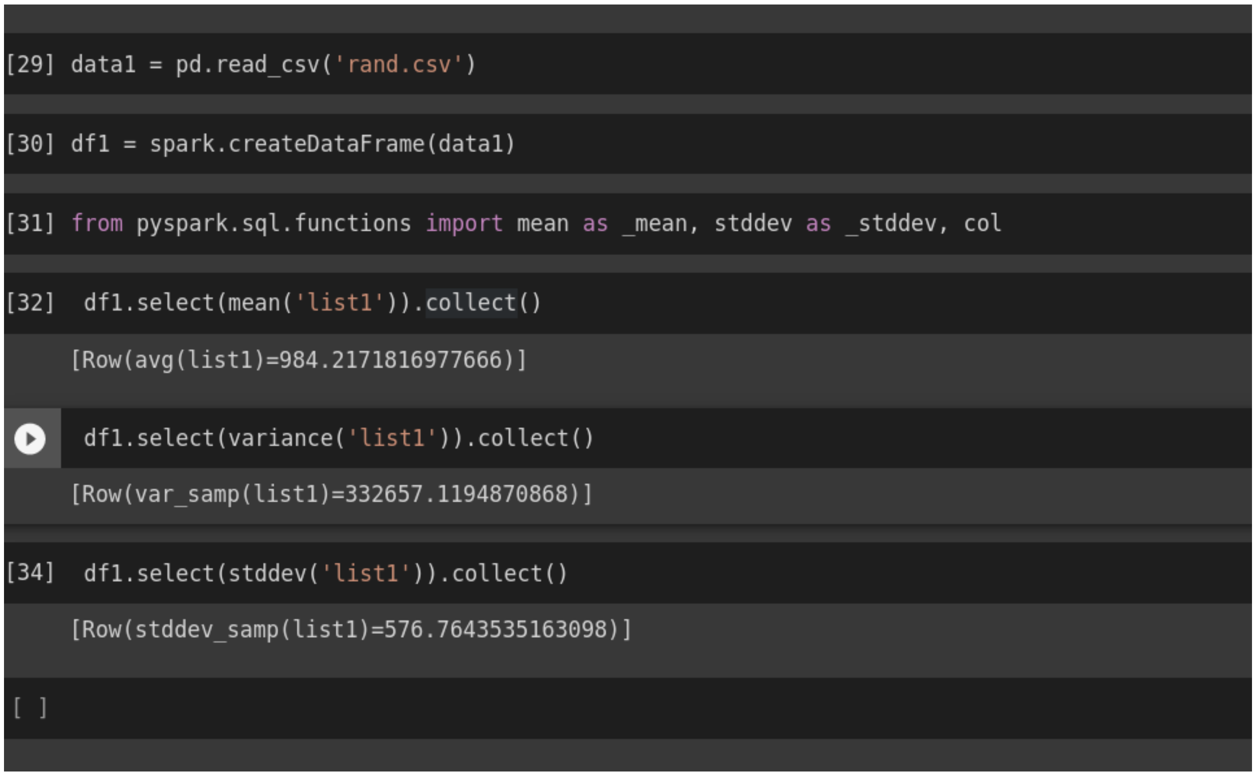
```
[2]  from pyspark.sql import SparkSession

[8]  import numpy as np
     import pandas as pd

     from pyspark.ml.stat import Correlation
     from pyspark.ml.linalg import DenseMatrix, Vectors
     from pyspark.ml.feature import VectorAssembler
     from pyspark.sql.functions import *

[11] spark = SparkSession.builder.appName('demo').master('local').enableHiveSupport().getOrCreate()

[12] spark
```



```
5]
   features = df.rdd.map(lambda row: row[0:])

   from pyspark.mllib.stat import Statistics

   corr_mat=Statistics.corr(features, method="pearson")

5] corr_mat

   array([[ 1., -1.],
          [-1.,  1.]])

7] corr_mat1=Statistics.corr(features, method="spearman")

3] corr_mat1

   array([[ 1., -1.],
          [-1.,  1.]])
```

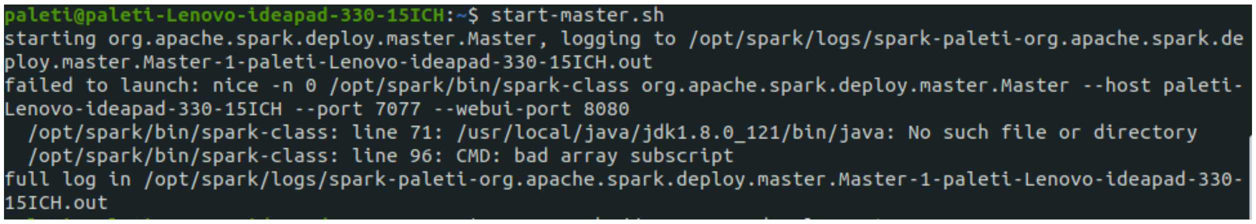
Paleti Krishnasai
CED18I039

2. Randomly populate 1000 numbers and calculate mean, variance, standard deviation for the generated data.



```
[29] data1 = pd.read_csv('rand.csv')

[30] df1 = spark.createDataFrame(data1)

[31] from pyspark.sql.functions import mean as _mean, stddev as _stddev, col

[32]  df1.select(mean('list1')).collect()

     [Row(avg(list1)=984.2171816977666)]

      df1.select(variance('list1')).collect()

     [Row(var_samp(list1)=332657.1194870868)]

[34]  df1.select(stddev('list1')).collect()

     [Row(stddev_samp(list1)=576.7643535163098)]

[ ]
```

(Question 1 and 4 are not working as pyspark master throws an error which i was unable to resolve)



```
paleti@paleti-Lenovo-ideapad-330-15ICH:~$ start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spark-paleti-org.apache.spark.de
ploy.master.Master-1-paleti-Lenovo-ideapad-330-15ICH.out
failed to launch: nice -n 0 /opt/spark/bin/spark-class org.apache.spark.deploy.master.Master --host paleti-
Lenovo-ideapad-330-15ICH --port 7077 --webui-port 8080
  /opt/spark/bin/spark-class: line 71: /usr/local/java/jdk1.8.0_121/bin/java: No such file or directory
  /opt/spark/bin/spark-class: line 96: CMD: bad array subscript
full log in /opt/spark/logs/spark-paleti-org.apache.spark.deploy.master.Master-1-paleti-Lenovo-ideapad-330-
15ICH.out
```