databricksExercise4

```
import pyspark
from pyspark.sql.types import StructType, StringType, IntegerType, DoubleType,
DateType, TimestampType
from pyspark.sql.functions import col, count, sum


schema = StructType()\
        .add("InvoiceNo", StringType(), True)\
        .add("StockCode", StringType(), True)\
        .add("Description", StringType(), True)\
        .add("Quantity", IntegerType(), True)\
        .add("InvoiceDate", DateType(), True)\
        .add("UnitPrice", DoubleType(), True)\
        .add("Customer", StringType(), True)\
        .add("Country", StringType(), True)


df = spark.read.format("csv")\
          .option("header", True)\
          .schema(schema)\
          .option("dateFormat", "MM/dd/yyyy HH:mm")\
          .load("/FileStore/tables/data.csv")


df.show(5)
```

```
+---------+---------+--------------------+--------+-----------+---------+-----
---+--------------+
|InvoiceNo|StockCode|         Description|Quantity|InvoiceDate|UnitPrice|Custo
mer|       Country|
+---------+---------+--------------------+--------+-----------+---------+-----
---+--------------+
|   536365|   85123A|WHITE HANGING HEA...|       6| 2010-12-01|     2.55|   17
850|United Kingdom|
|   536365|    71053| WHITE METAL LANTERN|       6| 2010-12-01|     3.39|   17
850|United Kingdom|
|   536365|   84406B|CREAM CUPID HEART...|       8| 2010-12-01|     2.75|   17
850|United Kingdom|
|   536365|   84029G|KNITTED UNION FLA...|       6| 2010-12-01|     3.39|   17
850|United Kingdom|
|   536365|   84029E|RED WOOLLY HOTTIE...|       6| 2010-12-01|     3.39|   17
850|United Kingdom|
+---------+---------+--------------------+--------+-----------+---------+-----
```

```
---+--------------+
only showing top 5 rows


df.printSchema()

root
 |-- InvoiceNo: string (nullable = true)
 |-- StockCode: string (nullable = true)
 |-- Description: string (nullable = true)
 |-- Quantity: integer (nullable = true)
 |-- InvoiceDate: date (nullable = true)
 |-- UnitPrice: double (nullable = true)
 |-- Customer: string (nullable = true)
 |-- Country: string (nullable = true)


df = df.filter(("Quantity > 0") or ("InvoiceNo not like 'C%'"))


df = df.filter("Customer is not NULL")


df = df.withColumn("Amount", col("Quantity") * col("UnitPrice"))


df.count()

Out[48]: 397924

df.show()
```

```
+---------+---------+--------------------+--------+-----------+---------+-----
---+--------------+------------------+
|InvoiceNo|StockCode|         Description|Quantity|InvoiceDate|UnitPrice|Custo
mer|       Country|            Amount|
+---------+---------+--------------------+--------+-----------+---------+-----
---+--------------+------------------+
|   536365|   85123A|WHITE HANGING HEA...|       6| 2010-12-01|     2.55|   17
850|United Kingdom|15.299999999999999|
|   536365|    71053| WHITE METAL LANTERN|       6| 2010-12-01|     3.39|   17
850|United Kingdom|             20.34|
|   536365|   84406B|CREAM CUPID HEART...|       8| 2010-12-01|     2.75|   17
850|United Kingdom|              22.0|
|   536365|   84029G|KNITTED UNION FLA...|       6| 2010-12-01|     3.39|   17
850|United Kingdom|             20.34|
```

```
|   536365|   84029E|RED WOOLLY HOTTIE...|       6| 2010-12-01|     3.39|   17
850|United Kingdom|             20.34|
|   536365|    22752|SET 7 BABUSHKA NE...|       2| 2010-12-01|     7.65|   17
850|United Kingdom|              15.3|
|   536365|    21730|GLASS STAR FROSTE...|       6| 2010-12-01|     4.25|   17
850|United Kingdom|              25.5|
```

```
from pyspark.sql import functions as F


grouped_df = df.groupBy("Customer",
F.month("InvoiceDate")).agg(F.sum(col("Amount")).alias("Amount"),\

F.countDistinct(col("InvoiceNo")).alias("InvoiceCount"), \

F.countDistinct(col("StockCode")).alias("ItemCount"))


grouped_df.show()
```

```
+--------+------------------+------------------+------------+---------+
|Customer|month(InvoiceDate)|            Amount|InvoiceCount|ItemCount|
+--------+------------------+------------------+------------+---------+
|   17596|                12|392.61999999999983|           1|       56|
|   13682|                 6|              59.5|           1|        3|
|   14800|                10|            392.24|           2|       36|
|   13755|                11|           1086.14|           5|      136|
|   15570|                 2|            431.75|           2|       32|
|   13881|                 5|1560.3400000000001|           1|       17|
|   18242|                 9|1538.4099999999996|           2|       52|
|   17193|                10|            935.16|           3|       17|
|   16161|                 1|200.25000000000003|           1|       29|
|   13451|                 3| 610.8800000000001|           1|      107|
|   15159|                 5|            810.95|           3|       27|
|   17758|                 7| 381.3699999999999|           1|       66|
|   12539|                11|           1050.66|           1|       43|
|   17692|                12|343.26000000000005|           1|       25|
|   17282|                 2| 741.8499999999999|           1|       78|
|   12472|                 6|1266.6599999999999|           1|       68|
|   18027|                 9|115.30999999999999|           1|       21|
|   12705|                 9| 842.6199999999999|           1|       38|
```

```
total_amount = grouped_df.agg(F.sum("Amount")).collect()[0][0]
```

```
total_invoice = grouped_df.agg(F.sum("InvoiceCount")).collect()[0][0]
total_item = grouped_df.agg(F.sum("ItemCount")).collect()[0][0]


grouped_df = grouped_df.withColumn("Adj_Amount", col("Amount") / total_amount)\
                       .withColumn("Adj_Invoice", col("InvoiceCount") /
total_invoice)\
                       .withColumn("Adj_Item", col("ItemCount") / total_item)


total_amount
```

```
Out[66]: 8911407.904000003
```

```
grouped_df.show()
```

```
+--------+------------------+------------------+------------+---------+-------
-------------+--------------------+--------------------+
|Customer|month(InvoiceDate)|            Amount|InvoiceCount|ItemCount|
Adj_Amount|         Adj_Invoice|            Adj_Item|
+--------+------------------+------------------+------------+---------+-------
-------------+--------------------+--------------------+
|   17596|                12|392.61999999999983|           1|       56|4.40581
3360016517E-5|5.394907207596029...|2.098934790593773...|
|   13682|                 6|              59.5|           1|        3|6.67683
4978375601E-6|5.394907207596029...|1.124429352103807...|
|   14800|                10|            392.24|           2|       36|4.40154
9162887472E-5|1.078981441519205...|1.349315222524568...|
|   13755|                11|           1086.14|           5|      136|1.21881
9755195441...|2.697453603798014...|5.097413062870593E-4|
|   15570|                 2|            431.75|           2|       32|4.84491
3448594396...|1.078981441519205...|1.199391308910727...|
|   13881|                 5|1560.3400000000001|           1|       17|1.75094
6670614888...|5.394907207596029...|6.371766328588241E-5|
|   18242|                 9|1538.4099999999996|           2|       52|1.72633
7764551732E-4|1.078981441519205...|1.949010876979932...|
|   17193|                10|            935.16|           3|       17|1.04939
```

```
grouped_df = grouped_df.withColumn("Score", (col("Adj_Amount") * 0.6) +
(col("Adj_Invoice") * 0.25) + (col("Adj_Item") * 0.15))


grouped_df.show()
```

```
+--------+------------------+------------------+------------+---------+-------
-------------+--------------------+--------------------+--------------------+
```

```
|Customer|month(InvoiceDate)|           Amount|InvoiceCount|ItemCount|
Adj_Amount|         Adj_Invoice|            Adj_Item|               Score|
+--------+------------------+-----------------+------------+---------+-------
------------+--------------------+--------------------+--------------------+
|   17596|                12|392.61999999999983|           1|       56|4.40581
3360016517E-5|5.394907207596029...|2.098934790593773...|7.140617003799577E-5|
|   13682|                 6|             59.5|           1|        3|6.67683
4978375601E-6|5.394907207596029...|1.124429352103807...|1.918001303417114...|
|   14800|                10|           392.24|           2|       36|4.40154
9162887472E-5|1.078981441519205...|1.349315222524568...|7.362355935317352E-5|
|   13755|                11|          1086.14|           5|      136|1.21881
9755195441...|2.697453603798014...|5.097413062870593E-4|2.170267213497357...|
|   15570|                 2|           431.75|           2|       32|4.84491
3448594396...|1.078981441519205...|1.199391308910727...|7.403488636320744E-5|
|   13881|                 5|1560.3400000000001|           1|       17|1.75094
6670614888...|5.394907207596029...|6.371766328588241E-5|1.281017177487657...|
|   18242|                 9|1538.4099999999996|           2|       52|1.72633
7764551732E-4|1.078981441519205...|1.949010876979932...|1.597899650657830...|
|   17193|                10|           935.16|           3|       17|1.04939
```

```
grouped_df.agg(F.min("Score")).show()
```

```
+--------------------+
|          min(Score)|
+--------------------+
|1.404948269504197...|
+--------------------+
```

```
grouped_df.agg(F.max("Score")).show()
```

```
+--------------------+
|          max(Score)|
+--------------------+
|0.020258281469089867|
+--------------------+
```

```
min_val = grouped_df.agg(F.min("Score")).collect()[0][0]
max_val = grouped_df.agg(F.max("Score")).collect()[0][0]
```

```
print(min_val)
print(max_val)
```

```
1.4049482695041978e-05
0.011357011345615509
```

```
grouped_df = grouped_df.withColumn("Norm_Score", ((col("Score") -
min_val)/(max_val - min_val)))\
              .orderBy(col("Norm_Score").desc())
```

```
grouped_df.show()
```

```
+--------+------------------+------------------+------------+---------+-------
-------------+--------------------+--------------------+--------------------+-
------------------+
|Customer|month(InvoiceDate)|            Amount|InvoiceCount|ItemCount|
Adj_Amount|         Adj_Invoice|            Adj_Item|               Score|
Norm_Score|
+--------+------------------+------------------+------------+---------+-------
-------------+--------------------+--------------------+--------------------+-
------------------+
|   16446|                12|          168469.6|           1|        1| 0.0189
0493643820078|5.394907207596029...|3.748097840346024...|0.011357011345615509|
1.0|
|   17450|                 9|          75412.64|          11|       52|0.00846
2483236363815|5.934397928355632E-4|1.949010876979932...|0.005255085053181879|
0.46205176688634564|
|   12346|                 1|           77183.6|           1|        1|0.00866
1212777091615|5.394907207596029...|3.748097840346024...|0.005210777148950011|
0.458145564541021|
|   18102|                10|52681.270000000004|          11|       55| 0.0059
1166632338234|5.934397928355632E-4|2.061453812190313...|0.003726281549421...|
0.32727184588896446|
```

```
grouped_df.agg(F.max("Norm_score")).show()
```

```
+---------------+
|max(Norm_score)|
+---------------+
|            1.0|
+---------------+
```

```
+---------------+
|min(Norm_score)|
+---------------+
|            0.0|
+---------------+
```