databricksExcercise8-Assessment1

```
import pyspark
from pyspark.sql.types import
StructType,StringType,IntegerType,DoubleType,FloatType,DateType

ex5_schema = StructType() \
        .add("InvoiceNo",StringType(),False) \
        .add("StockCode",StringType(),False) \
        .add("Description",StringType(),True) \
        .add("Quantity",IntegerType(),True) \
        .add("InvoiceDate",StringType(),True) \
        .add("UnitPrice",DoubleType(),True) \
        .add("CustomerID",IntegerType(),True) \
        .add("Country",StringType(),True)

kaggledata = spark.read.format("csv") \
          .option("header",True) \
          .schema(ex5_schema) \
          .load("FileStore/tables/data.csv")

kaggledata.printSchema()
kaggledata.show(5)
```

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

+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
|InvoiceNo|StockCode|         Description|Quantity|   InvoiceDate|UnitPrice|Cu
stomerID|       Country|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
|   536365|   85123A|WHITE HANGING HEA...|       6|12/1/2010 8:26|     2.55|
17850|United Kingdom|
|   536365|    71053| WHITE METAL LANTERN|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|
|   536365|   84406B|CREAM CUPID HEART...|       8|12/1/2010 8:26|     2.75|
17850|United Kingdom|
```

```
|   536365|   84029G|KNITTED UNION FLA...|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|
|   536365|   84029E|RED WOOLLY HOTTIE...|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
only showing top 5 rows
```

```
from pyspark.sql import *
from pyspark.sql.functions import col,round
from pyspark.sql import functions as F

Kaggle_min_quantity=kaggledata.groupBy(col('InvoiceNo')) \
                              .agg(F.min(col('Quantity')).alias('Minimum
quantity')) \
                              .filter(~col("InvoiceNo").startswith("%C")) \
                              .sort(col('InvoiceNo'))

display(Kaggle_min_quantity)
```

	InvoiceNo	Minimum quantity
1	536365	2
2	536366	6
3	536367	2
4	536368	3
5	536369	3
6	536370	3
7	536371	80
8	536372	6

Showing the first 1000 rows.

```
Kaggle_max_quantity=kaggledata.groupBy(col('InvoiceNo')) \
                              .agg(F.max(col('Quantity')).alias('Maximum
quantity')) \
                              .filter(~col("InvoiceNo").startswith("%C")) \
                              .sort(col('InvoiceNo'))

display(Kaggle_max_quantity)
```

1	536365	8
2	536366	6
3	536367	32
4	536368	6
5	536369	3
6	536370	48
7	536371	80
8	536372	6

Showing the first 1000 rows.

	InvoiceNo	Average quantity
1	536365	5.714285714285714
2	536366	6
3	536367	6.916666666666667
4	536368	3.75
5	536369	3
6	536370	22.45
7	536371	80
8	536372	6

Showing the first 1000 rows.