Christos Kallaras, p2822009
Department of Management Science and Technology
Master in Business Analytics
Athens University of Economics and Business

TASK 2

In this task I will create reports on the average and median departure delays of all the airports, and all the airways in the dataset.

Note 1: The code was written in Jupiter's Notebook and executed through there and the console. All figures are from the console. Configuring the max memory in 14g was necessary for the project to run successfully with Jupiter's Notebook.

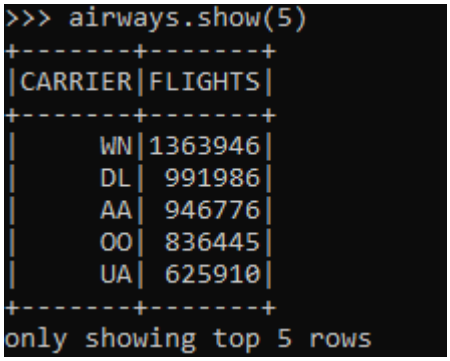
Note 2: In order for the code to run successfully with any given path a variable path was created in which you must declare the path of the file in your local PC.

DATA CLEANING

The dataset needed to be cleaned first. Specifically, I should not consider in my analysis any airport/airway belonging in the lowest 1% percentile, regarding the number of flights. I used SparkSQL again for this transformation. First, I created 2 dataframes one containing all the airports and the number of flights from that airport (let's call this dataframe **airports**), and the other containing the airways and the number of flights they participated in (lets call this dataframe airways).Those two dataframes can be seen in Figure 1 and Figure 2. The two dataframes were also transformed into tables and from there I extracted the 1% percentile for airports and airways (**airport_limit** and **airways_limit** respectively) using again SparkSQL. Then I created a query that joined the 2 tables, airports and airways, with the originally dataset with the condition that the number of flights is bigger or equal to the 1% percentile I extracted earlier (Figure 3).

```
>>> airports.show(5)
+------+-------+
|ORIGIN|FLIGHTS|
+------+-------+
|   ATL| 395009|
|   ORD| 339606|
|   DFW| 304344|
|   DEN| 252026|
|   CLT| 235496|
+------+-------+
only showing top 5 rows
```

Figure 1: airports dataframe



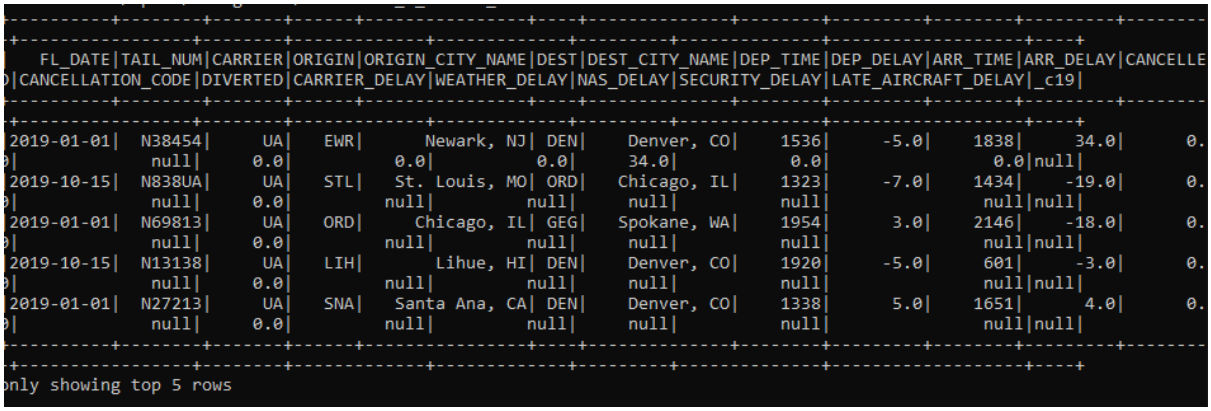
```
>>> airways.show(5)
+-------+-------+
|CARRIER|FLIGHTS|
+-------+-------+
|     WN|1363946|
|     DL| 991986|
|     AA| 946776|
|     OO| 836445|
|     UA| 625910|
+-------+-------+
only showing top 5 rows
```

Figure 2: airways dataframe

```
SELECT flights.* FROM flights,
(
select CARRIER,COUNT (*) as number_flights
from flights
group by CARRIER
) AS A ,
(
select ORIGIN,COUNT (*) as number_flights
from flights
group by ORIGIN
) AS B
WHERE flights.CARRIER = A.CARRIER
AND flights.ORIGIN = B.ORIGIN
AND A.number_flights > 625.9
AND B.number_flights > 122931.80000000002
```

Figure 3: Query for outliers

I runed that query and in that way I had created a dataframe with all the airport/airway with the number of flights greater than 1% percentile.The new dtaframe with no outliers can be seen in Figure 4.



```
+----------+--------+-------+------+---------------+----+-------------+--------+---------+--------+---------+--------
+-----------------+--------+------------+------------+---------+-------------+------------------+----+
   FL_DATE|TAIL_NUM|CARRIER|ORIGIN|ORIGIN_CITY_NAME|DEST|DEST_CITY_NAME|DEP_TIME|DEP_DELAY|ARR_TIME|ARR_DELAY|CANCELLE
D|CANCELLATION_CODE|DIVERTED|CARRIER_DELAY|WEATHER_DELAY|NAS_DELAY|SECURITY_DELAY|LATE_AIRCRAFT_DELAY|_c19|
+----------+--------+-------+------+---------------+----+-------------+--------+---------+--------+---------+--------
+-----------------+--------+------------+------------+---------+-------------+------------------+----+
|2019-01-01|  N38454|     UA|   EWR|      Newark, NJ| DEN|    Denver, CO|    1536|     -5.0|    1838|     34.0|      0.
0|             null|     0.0|         0.0|         0.0|     34.0|          0.0|               0.0|null|
|2019-10-15|  N838UA|     UA|   STL|   St. Louis, MO| ORD|   Chicago, IL|    1323|     -7.0|    1434|    -19.0|      0.
0|             null|     0.0|        null|        null|     null|         null|              null|null|
|2019-01-01|  N69813|     UA|   ORD|    Chicago, IL| GEG|   Spokane, WA|    1954|      3.0|    2146|    -18.0|      0.
0|             null|     0.0|        null|        null|     null|         null|              null|null|
|2019-10-15|  N13138|     UA|   LIH|       Lihue, HI| DEN|    Denver, CO|    1920|     -5.0|     601|     -3.0|      0.
0|             null|     0.0|        null|        null|     null|         null|              null|null|
|2019-01-01|  N27213|     UA|   SNA|   Santa Ana, CA| DEN|    Denver, CO|    1338|      5.0|    1651|      4.0|      0.
0|             null|     0.0|        null|        null|     null|         null|              null|null|
+----------+--------+-------+------+---------------+----+-------------+--------+---------+--------+---------+--------
+-----------------+--------+------------+------------+---------+-------------+------------------+----+
only showing top 5 rows
```

Figure 4:Data with no outliers

CREATE REPORTS

I now have to create 4 reports on the average and median departure delays of all the airports, and all the airways in the dataset. Using SparkSQL and function avg() I extracted the average for airports and airways and write the result in the files. For the median I used the function percentile_approx() that returns the approximate percentile value of the specified numeric column (dep delay) at the given percentage (0.5). This will give the approximate value for the median. I also created another way to calculate the median as can be seen in Figure 5 .But the results were quite similar so I kept the percentile_approx() function.

```
task2_ap_med = spark.sql("""
SELECT ORIGIN, avg(DEP_DELAY) as median
FROM ( SELECT ORIGIN, DEP_DELAY, row, (CASE WHEN column % 2 = 0 then (column DIV 2) ELSE (column DIV 2) + 1 end) as m1, (column DIV 2) + 1 as m2
        FROM (
            SELECT ORIGIN, DEP_DELAY, row_number() OVER (PARTITION BY ORIGIN ORDER BY DEP_DELAY ) as row, count(DEP_DELAY) OVER (PARTITION BY ORIGIN ) as column
            FROM data_without_outliers
        ) s
    ) r
WHERE row BETWEEN m1 and m2
GROUP BY ORIGIN
ORDER BY median  DESC
""")
task2_ap_med.show(5)
```

Figure 5: Median calculation

The structure of the files is as instriucted with no header containing the airport/airway and the average/median. Only top 100 rows were written.