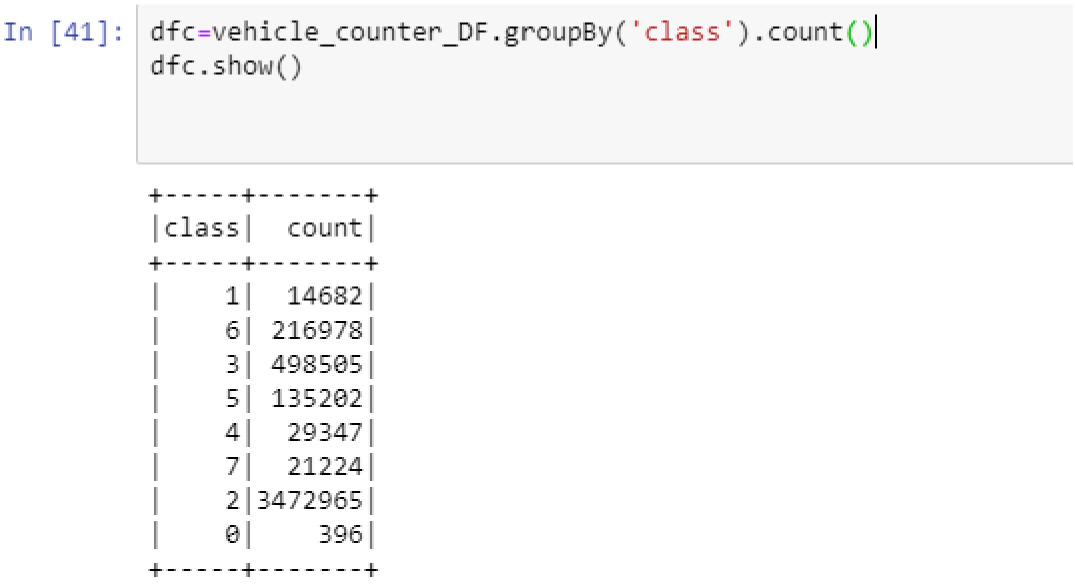
Big Data Management

Assignment 1

1. Show total number of counts by each vehicle class.

dfc=vehicle_counter_DF.groupBy('class').count()

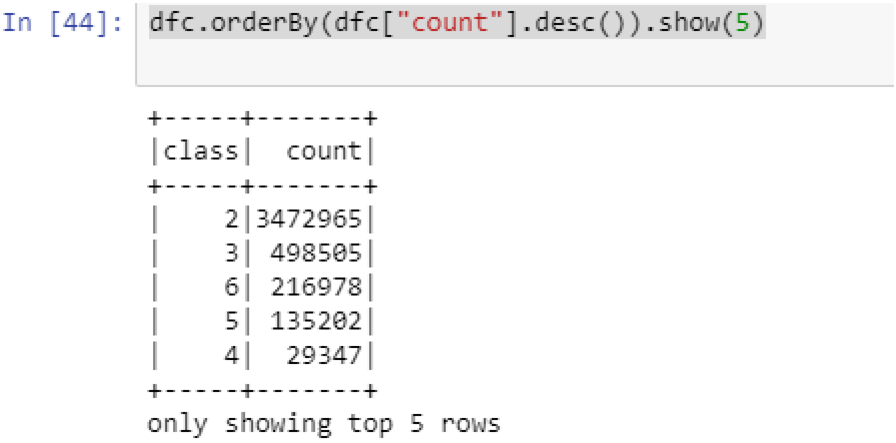
dfc.show()



```
In [41]: dfc=vehicle_counter_DF.groupBy('class').count()
         dfc.show()

         +-----+-------+
         |class|  count|
         +-----+-------+
         |    1|  14682|
         |    6| 216978|
         |    3| 498505|
         |    5| 135202|
         |    4|  29347|
         |    7|  21224|
         |    2|3472965|
         |    0|    396|
         +-----+-------+
```

2. Show top 5 highest number of counts by vehicle class.

dfc.orderBy(dfc["count"].desc()).show(5)

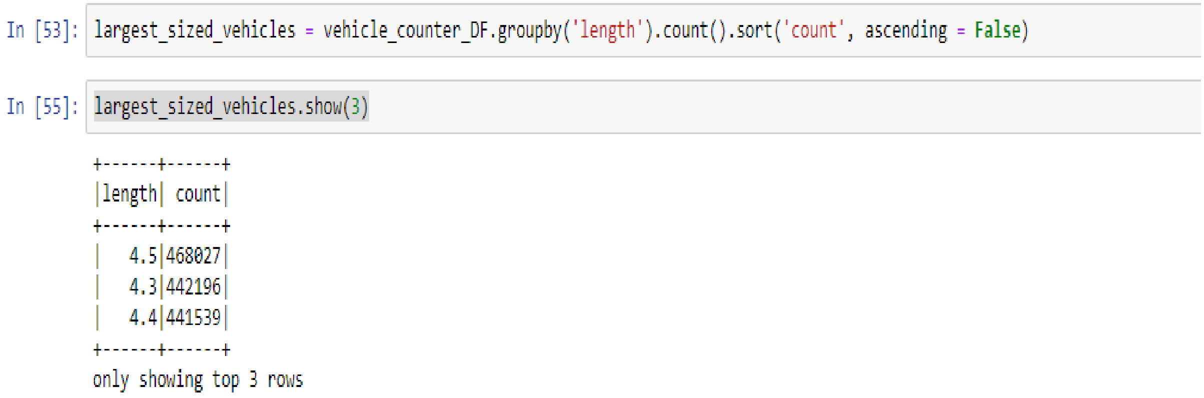


```
In [44]: dfc.orderBy(dfc["count"].desc()).show(5)

         +-----+-------+
         |class|  count|
         +-----+-------+
         |    2|3472965|
         |    3| 498505|
         |    6| 216978|
         |    5| 135202|
         |    4|  29347|
         +-----+-------+
         only showing top 5 rows
```

3. Show total number of counts by the largest sized (length) vehicle.

largest_sized_vehicles = vehicle_counter_DF.groupby('length').count().sort('count', ascending = False)

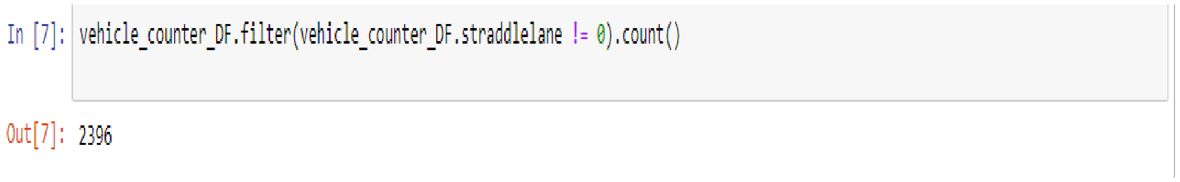
largest_sized_vehicles.show(3)



```
In [53]: largest_sized_vehicles = vehicle_counter_DF.groupby('length').count().sort('count', ascending = False)

In [55]: largest_sized_vehicles.show(3)

         +------+------+
         |length| count|
         +------+------+
         |   4.5|468027|
         |   4.3|442196|
         |   4.4|441539|
         +------+------+
         only showing top 3 rows
```

4. How many vehicles were counted on straddlelane?

vehicle_counter_DF.filter(vehicle_counter_DF.straddlelane != 0).count()

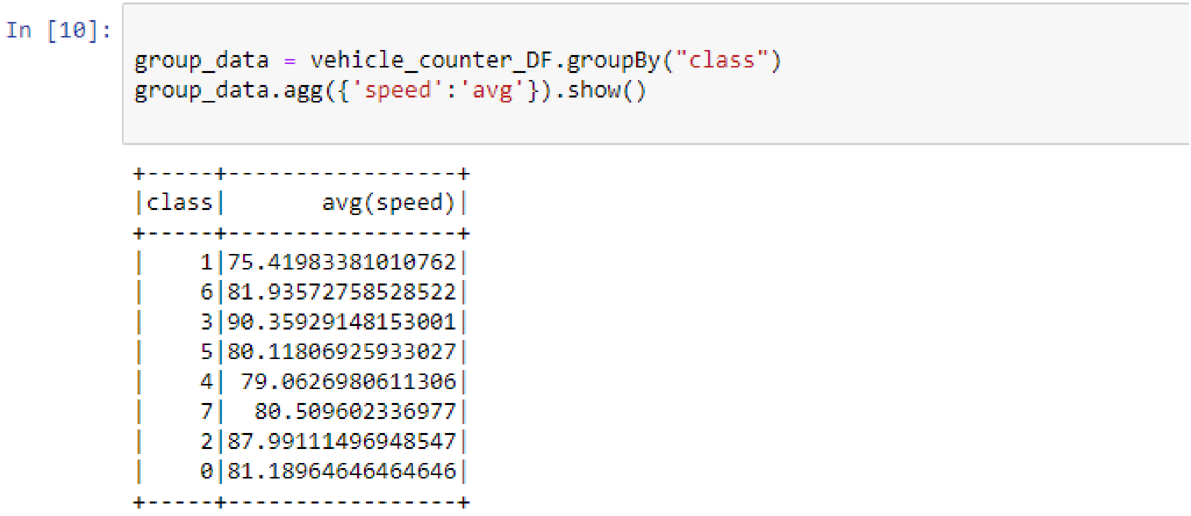


```
In [7]: vehicle_counter_DF.filter(vehicle_counter_DF.straddlelane != 0).count()

Out[7]: 2396
```

5. Compute average speed (for each counter) of vehicles with respect to their class.

group_data = vehicle_counter_DF.groupBy("class")

group_data.agg({'speed':'avg'}).show()



```
In [10]: group_data = vehicle_counter_DF.groupBy("class")
         group_data.agg({'speed':'avg'}).show()

         +-----+-----------------+
         |class|       avg(speed)|
         +-----+-----------------+
         |    1|75.41983381010762|
         |    6|81.93572758528522|
         |    3|90.35929148153001|
         |    5|80.11806925933027|
         |    4| 79.0626980611306|
         |    7|  80.509602336977|
         |    2|87.99111496948547|
         |    0|81.18964646464646|
         +-----+-----------------+
```

6. Combine the date and time fields (year, month, day, hour, minute, seconds, millisecond) into one variable (e.g. YYY-MM-DD-HH-MM-SSMMM) and call it timestamp.

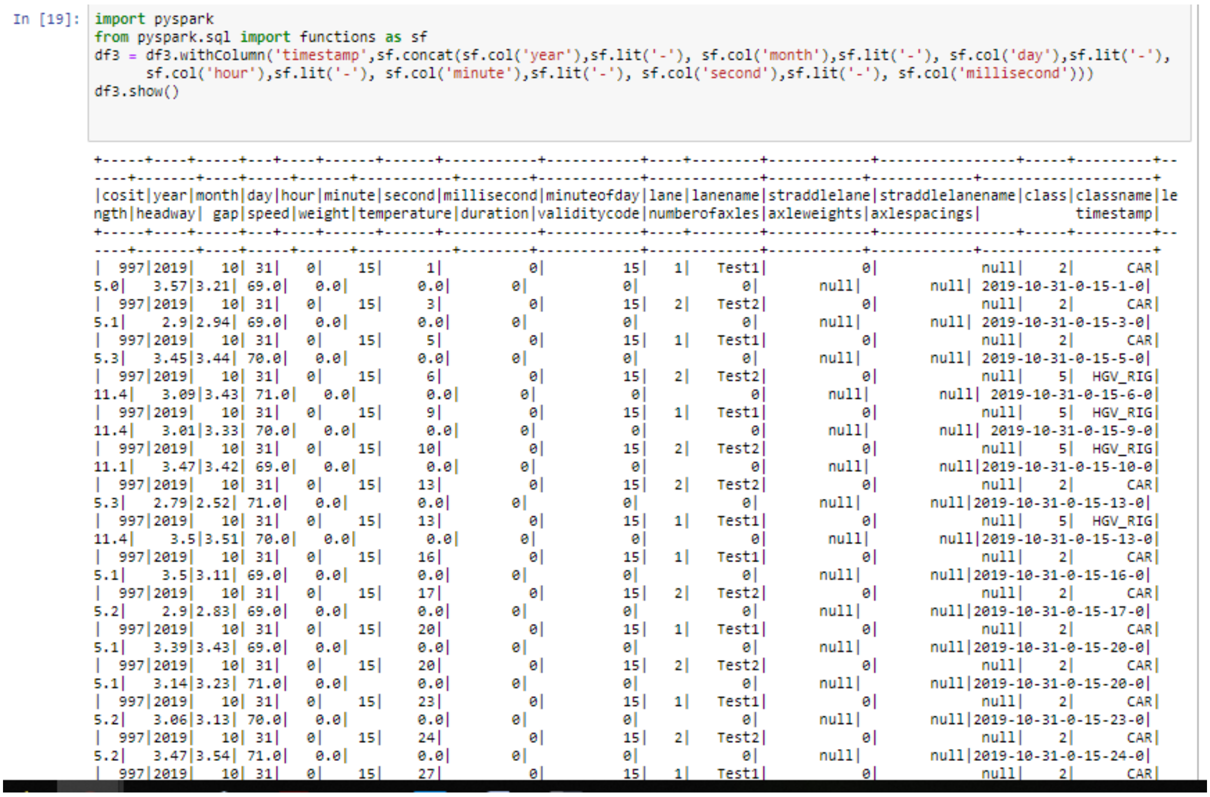
df3=vehicle_counter_DF

import pyspark

from pyspark.sql import functions as sf

df3 = df3.withColumn('timestamp',sf.concat(sf.col('year'),sf.lit('-'), sf.col('month'),sf.lit('-'), sf.col('day'),sf.lit('-'), sf.col('hour'),sf.lit('-'), sf.col('minute'),sf.lit('-'), sf.col('second'),sf.lit('-'), sf.col('millisecond')))

df3.show()

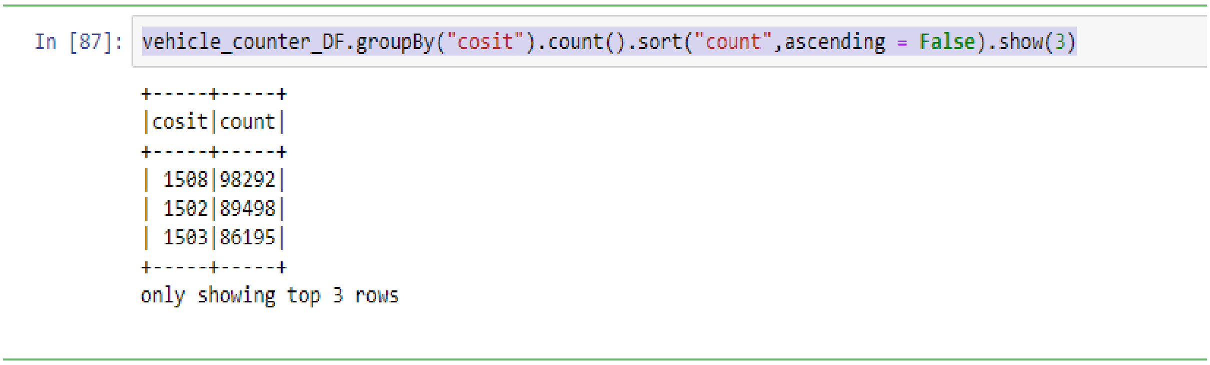


```
In [19]: import pyspark
         from pyspark.sql import functions as sf
         df3 = df3.withColumn('timestamp',sf.concat(sf.col('year'),sf.lit('-'), sf.col('month'),sf.lit('-'), sf.col('day'),sf.lit('-'),
               sf.col('hour'),sf.lit('-'), sf.col('minute'),sf.lit('-'), sf.col('second'),sf.lit('-'), sf.col('millisecond')))
         df3.show()
```

```
+-----+----+-----+---+----+------+------+-----------+-----------+----+--------+------------+----------------+-----+---------+--
----+-------+----+-----+------+-----------+--------+------------+-------------+-----------+------------+--------------------+
|cosit|year|month|day|hour|minute|second|millisecond|minuteofday|lane|lanename|straddlelane|straddlelanename|class|classname|le
ngth|headway| gap|speed|weight|temperature|duration|validitycode|numberofaxles|axleweights|axlespacings|           timestamp|
+-----+----+-----+---+----+------+------+-----------+-----------+----+--------+------------+----------------+-----+---------+--
----+-------+----+-----+------+-----------+--------+------------+-------------+-----------+------------+--------------------+
|  997|2019|   10| 31|   0|    15|     1|          0|         15|   1|   Test1|           0|            null|    2|      CAR|
5.0|   3.57|3.21| 69.0|   0.0|        0.0|       0|           0|            0|       null|        null| 2019-10-31-0-15-1-0|
|  997|2019|   10| 31|   0|    15|     3|          0|         15|   2|   Test2|           0|            null|    2|      CAR|
5.1|    2.9|2.94| 69.0|   0.0|        0.0|       0|           0|            0|       null|        null| 2019-10-31-0-15-3-0|
|  997|2019|   10| 31|   0|    15|     5|          0|         15|   1|   Test1|           0|            null|    2|      CAR|
5.3|   3.45|3.44| 70.0|   0.0|        0.0|       0|           0|            0|       null|        null| 2019-10-31-0-15-5-0|
|  997|2019|   10| 31|   0|    15|     6|          0|         15|   2|   Test2|           0|            null|    5|  HGV_RIG|
11.4|   3.09|3.43| 71.0|   0.0|        0.0|       0|           0|            0|       null|        null| 2019-10-31-0-15-6-0|
|  997|2019|   10| 31|   0|    15|     9|          0|         15|   1|   Test1|           0|            null|    5|  HGV_RIG|
11.4|   3.01|3.33| 70.0|   0.0|        0.0|       0|           0|            0|       null|        null| 2019-10-31-0-15-9-0|
|  997|2019|   10| 31|   0|    15|    10|          0|         15|   2|   Test2|           0|            null|    5|  HGV_RIG|
11.1|   3.47|3.42| 69.0|   0.0|        0.0|       0|           0|            0|       null|        null|2019-10-31-0-15-10-0|
|  997|2019|   10| 31|   0|    15|    13|          0|         15|   2|   Test2|           0|            null|    2|      CAR|
5.3|   2.79|2.52| 71.0|   0.0|        0.0|       0|           0|            0|       null|        null|2019-10-31-0-15-13-0|
|  997|2019|   10| 31|   0|    15|    13|          0|         15|   1|   Test1|           0|            null|    5|  HGV_RIG|
11.4|    3.5|3.51| 70.0|   0.0|        0.0|       0|           0|            0|       null|        null|2019-10-31-0-15-13-0|
|  997|2019|   10| 31|   0|    15|    16|          0|         15|   1|   Test1|           0|            null|    2|      CAR|
5.1|    3.5|3.11| 69.0|   0.0|        0.0|       0|           0|            0|       null|        null|2019-10-31-0-15-16-0|
|  997|2019|   10| 31|   0|    15|    17|          0|         15|   2|   Test2|           0|            null|    2|      CAR|
5.2|    2.9|2.83| 69.0|   0.0|        0.0|       0|           0|            0|       null|        null|2019-10-31-0-15-17-0|
|  997|2019|   10| 31|   0|    15|    20|          0|         15|   1|   Test1|           0|            null|    2|      CAR|
5.1|   3.39|3.43| 69.0|   0.0|        0.0|       0|           0|            0|       null|        null|2019-10-31-0-15-20-0|
|  997|2019|   10| 31|   0|    15|    20|          0|         15|   2|   Test2|           0|            null|    2|      CAR|
5.1|   3.14|3.23| 71.0|   0.0|        0.0|       0|           0|            0|       null|        null|2019-10-31-0-15-20-0|
|  997|2019|   10| 31|   0|    15|    23|          0|         15|   1|   Test1|           0|            null|    2|      CAR|
5.2|   3.06|3.13| 70.0|   0.0|        0.0|       0|           0|            0|       null|        null|2019-10-31-0-15-23-0|
|  997|2019|   10| 31|   0|    15|    24|          0|         15|   2|   Test2|           0|            null|    2|      CAR|
5.2|   3.47|3.54| 71.0|   0.0|        0.0|       0|           0|            0|       null|        null|2019-10-31-0-15-24-0|
|  997|2019|   10| 31|   0|    15|    27|          0|         15|   1|   Test1|           0|            null|    2|      CAR|
```

7. List the top 3 busiest roads in Ireland (sites).

vehicle_counter_DF.groupBy("cosit").count().sort("count",ascending = False).show(3)

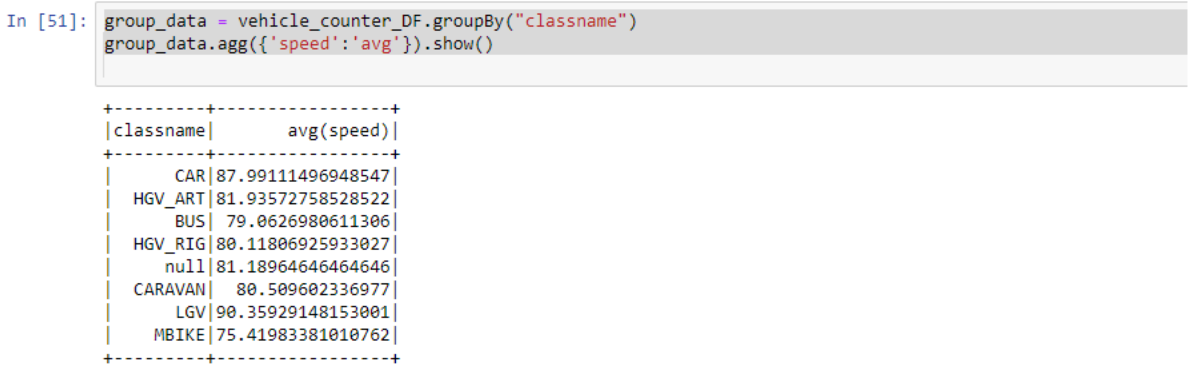


```
In [87]: vehicle_counter_DF.groupBy("cosit").count().sort("count",ascending = False).show(3)

+-----+-----+
|cosit|count|
+-----+-----+
| 1508|98292|
| 1502|89498|
| 1503|86195|
+-----+-----+
only showing top 3 rows
```

8. Your choice of question I - present any sensible statistic of the data.

group_data = vehicle_counter_DF.groupBy("classname")

group_data.agg({'speed':'avg'}).show()



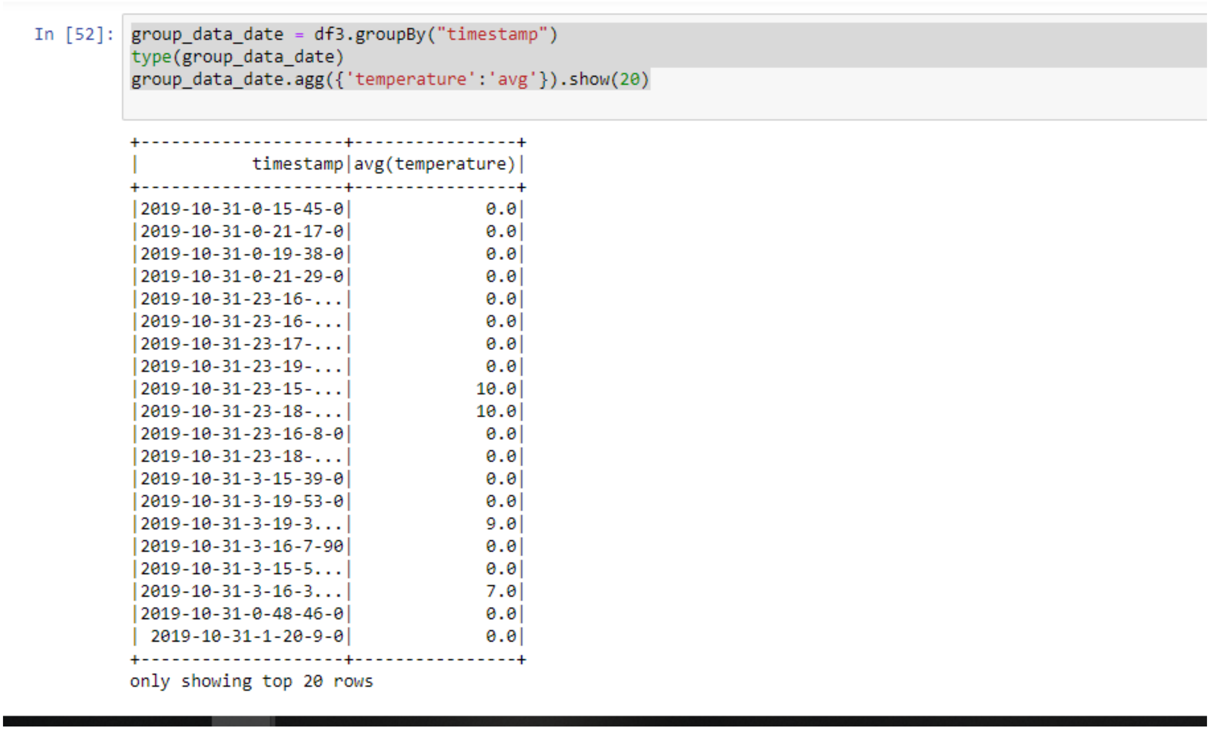
```
In [51]: group_data = vehicle_counter_DF.groupBy("classname")
         group_data.agg({'speed':'avg'}).show()

+---------+-----------------+
|classname|       avg(speed)|
+---------+-----------------+
|      CAR|87.99111496948547|
|  HGV_ART|81.93572758528522|
|      BUS| 79.0626980611306|
|  HGV_RIG|80.11806925933027|
|     null|81.18964646464646|
|  CARAVAN|  80.509602336977|
|      LGV|90.35929148153001|
|    MBIKE|75.41983381010762|
+---------+-----------------+
```

9. Your choice of question II - present any sensible statistic of the data.

group_data_date = df3.groupBy("timestamp")

type(group_data_date)

group_data_date.agg({'temperature':'avg'}).show(20)

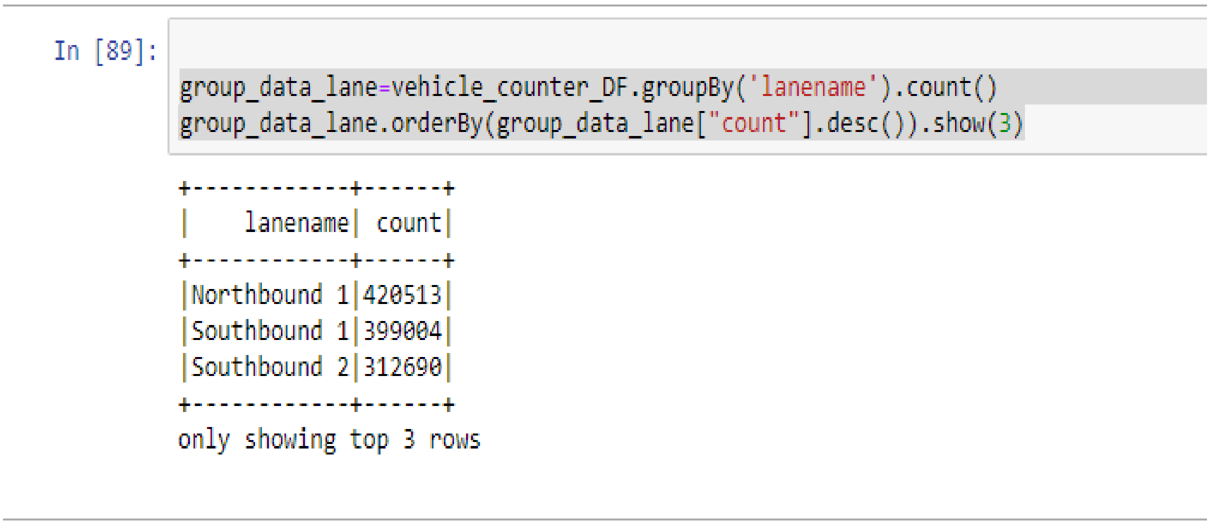


```
In [52]: group_data_date = df3.groupBy("timestamp")
         type(group_data_date)
         group_data_date.agg({'temperature':'avg'}).show(20)

+-------------------+----------------+
|          timestamp|avg(temperature)|
+-------------------+----------------+
|2019-10-31-0-15-45-0|            0.0|
|2019-10-31-0-21-17-0|            0.0|
|2019-10-31-0-19-38-0|            0.0|
|2019-10-31-0-21-29-0|            0.0|
|2019-10-31-23-16-...|            0.0|
|2019-10-31-23-16-...|            0.0|
|2019-10-31-23-17-...|            0.0|
|2019-10-31-23-19-...|            0.0|
|2019-10-31-23-15-...|           10.0|
|2019-10-31-23-18-...|           10.0|
|2019-10-31-23-16-8-0|            0.0|
|2019-10-31-23-18-...|            0.0|
|2019-10-31-3-15-39-0|            0.0|
|2019-10-31-3-19-53-0|            0.0|
|2019-10-31-3-19-3...|            9.0|
|2019-10-31-3-16-7-90|            0.0|
|2019-10-31-3-15-5...|            0.0|
|2019-10-31-3-16-3...|            7.0|
|2019-10-31-0-48-46-0|            0.0|
| 2019-10-31-1-20-9-0|            0.0|
+-------------------+----------------+
only showing top 20 rows
```

10. Your choice of question III - present any sensible statistic of the data.

group_data_lane=vehicle_counter_DF.groupBy('lanename').count()

group_data_lane.orderBy(group_data_lane["count"].desc()).show(3)



```
In [89]: group_data_lane=vehicle_counter_DF.groupBy('lanename').count()
         group_data_lane.orderBy(group_data_lane["count"].desc()).show(3)

+------------+------+
|    lanename| count|
+------------+------+
|Northbound 1|420513|
|Southbound 1|399004|
|Southbound 2|312690|
+------------+------+
only showing top 3 rows
```