NYPDCollisions

NYPD Collision Data Analysis

FINISHED

Data for this project is taken from NYC Open Data, https://data.cityofnewyork.us/public-safety/NYPD-Motor-Vehicle-Collisions website where data is provided by Police Department(NYPD).This project i decided to work with Spark. Visualization of the data is done with Zeppelin inbuilt features.

Data Loading

FINISHED

```
%pyspark

#Initialize SparkSession and SparkContext
from pyspark.sql import SparkSession
from pyspark import SparkConf
from pyspark import SparkContext


#Create a Spark Session
SpSession = SparkSession \
    .builder \
    .master("local[2]") \
    .appName("PG Spark App") \
    .config("spark.executor.memory", "1g") \
    .config("spark.sql.warehouse.dir", "file:///tmp/spark-warehouse")\
    .config("spark.cores.max","2") \
    .getOrCreate()


# read the nypd collision data which is a csv file into the datafarame
collisionDataDF = SpSession.read.csv("/Users/girishdurgaiah/spark/NYPDCollisionData.csv",he
```

FINISHED

```
%pyspark
# after the data is read into the dataframe make sure the data exists in dataframe
# show () will show top 20 rows but we can specify the number of rows we want to see

collisionDataDF.show(3)
```

```
+----------+----+--------+--------+---------+---------+--------------------+--------------
-----+------------------+--------------+-----------------------+---------------------
-+---------------------------+---------------------------+-----------------------+----
------------------+-------------------------+-----------------------+----------------
-----------+---------------------------+---------------------------+----------------
----------+---------------------------+---------+-------------------+----------------
--+------------------+------------------+------------------+
|      DATE|TIME| BOROUGH|ZIP CODE| LATITUDE|LONGITUDE|            LOCATION|      ON STREET
NAME|  CROSS STREET NAME|OFF STREET NAME|NUMBER OF PERSONS INJURED|NUMBER OF PERSONS KILLED
|NUMBER OF PEDESTRIANS INJURED|NUMBER OF PEDESTRIANS KILLED|NUMBER OF CYCLIST INJURED|NUMBE
R OF CYCLIST KILLED|NUMBER OF MOTORIST INJURED|NUMBER OF MOTORIST KILLED|CONTRIBUTING FACTO
R VEHICLE 1|CONTRIBUTING FACTOR VEHICLE 2|CONTRIBUTING FACTOR VEHICLE 3|CONTRIBUTING FACTOR
VEHICLE 4|CONTRIBUTING FACTOR VEHICLE 5|UNIQUE KEY|VEHICLE TYPE CODE 1| VEHICLE TYPE CODE 2
|VEHICLE TYPE CODE 3|VEHICLE TYPE CODE 4|VEHICLE TYPE CODE 5|
+----------+----+--------+--------+---------+---------+--------------------+--------------
-----+------------------+--------------+-----------------------+---------------------
-+---------------------------+---------------------------+-----------------------+----
------------------'-------------------------'-----------------------'----------------
```

```
%pyspark                                                                    FINISHED
# count the number of rows in he dataframe
# there are 1052351 rows in the dataframe
collisionDataDF.count()
```

```
1052351
```

```
%pyspark                                                                    FINISHED
#lets see the columns and their schema of the dataframe
# By printing schema we can see the column names and their type
collisionDataDF.printSchema()
```

```
root
 |-- DATE: string (nullable = true)
 |-- TIME: string (nullable = true)
 |-- BOROUGH: string (nullable = true)
 |-- ZIP CODE: string (nullable = true)
 |-- LATITUDE: string (nullable = true)
 |-- LONGITUDE: string (nullable = true)
 |-- LOCATION: string (nullable = true)
 |-- ON STREET NAME: string (nullable = true)
 |-- CROSS STREET NAME: string (nullable = true)
 |-- OFF STREET NAME: string (nullable = true)
 |-- NUMBER OF PERSONS INJURED: string (nullable = true)
 |-- NUMBER OF PERSONS KILLED: string (nullable = true)
 |-- NUMBER OF PEDESTRIANS INJURED: string (nullable = true)
 |-- NUMBER OF PEDESTRIANS KILLED: string (nullable = true)
 |-- NUMBER OF CYCLIST INJURED: string (nullable = true)
 |-- NUMBER OF CYCLIST KILLED: string (nullable = true)
 |-- NUMBER OF MOTORIST INJURED: string (nullable = true)
```

FINISHED

```
%pyspark
# rename the column names of dataframe
collisionDataDF = collisionDataDF.withColumnRenamed("NUMBER OF PERSONS KILLED", "personsKi
                    withColumnRenamed("NUMBER OF PERSONS INJURED", "personsInjured"). \
                    withColumnRenamed("NUMBER OF PEDESTRIANS INJURED", "pedestriansInjured
                    withColumnRenamed("NUMBER OF PEDESTRIANS KILLED", "pedestriansKilled")
                    withColumnRenamed("NUMBER OF CYCLIST INJURED", "cyclistInjured"). \
                    withColumnRenamed("NUMBER OF CYCLIST KILLED", "cyclistKilled"). \
                    withColumnRenamed("NUMBER OF MOTORIST INJURED", "motoristInjured"). \
                    withColumnRenamed("NUMBER OF MOTORIST KILLED", "motoristKilled"). \
                    withColumnRenamed("CONTRIBUTING FACTOR VEHICLE 1", "factorVehicle1"). \
                    withColumnRenamed("CONTRIBUTING FACTOR VEHICLE 2", "factorVehicle2"). \
                    withColumnRenamed("VEHICLE TYPE CODE 1", "vehicle1"). \
                    withColumnRenamed("VEHICLE TYPE CODE 2", "vehicle2"). \
                    withColumnRenamed("UNIQUE KEY", "key")
```

DATA ANALYSIS

FINISHED

FINISHED

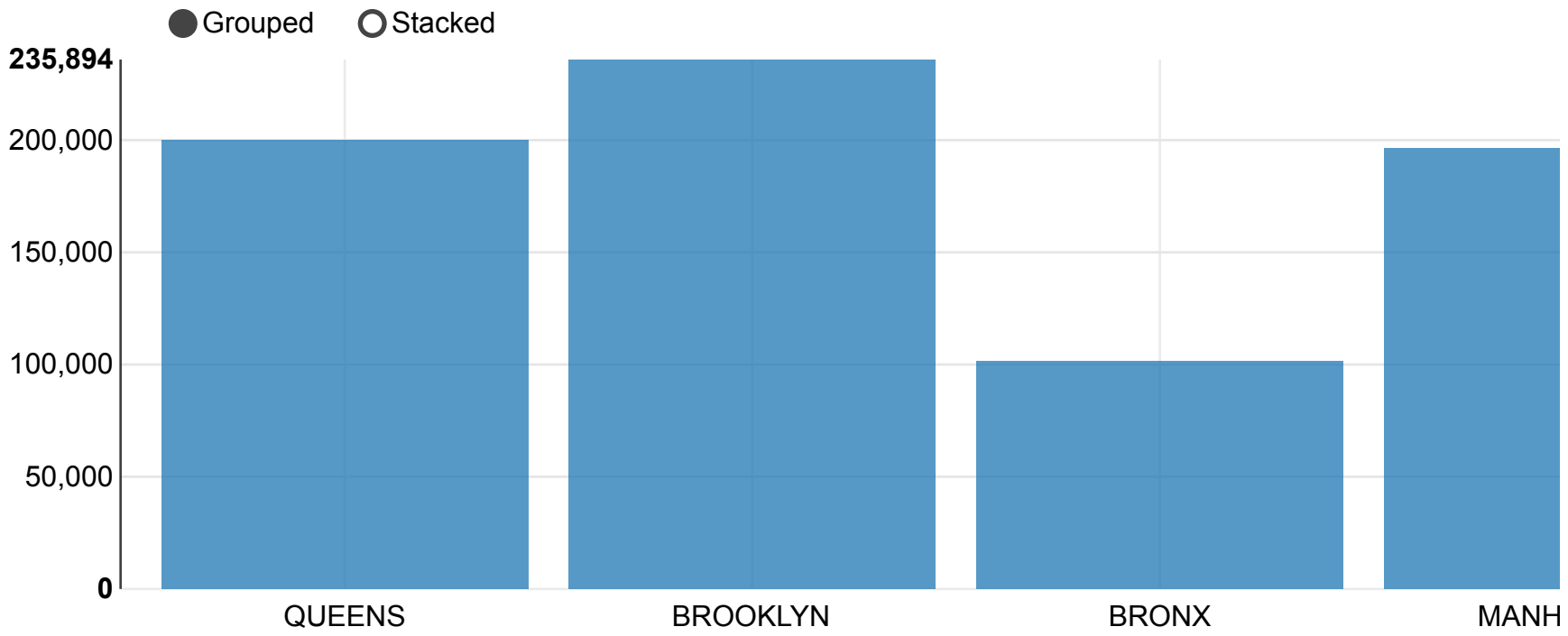
```
%pyspark

#Register a temp table called "collision" using collisionDataDF.
collisionDataDF.createOrReplaceTempView("collision")
```

NUMBER OF COLLISIONS IN EACH BOROUGH

FINISHED

FINISHED



Above table shows the total number of collisions for all 5 Boroughs.BROOKLYN tops the list with 235,894 collisions and STATEN ISLAND is the least with 35562 collisions.The reason may be the population. Staten Island is the least populated borough.

FINISHED

PEOPLE INJURED DURING COLLISION

FINISHED

```
%sql
select sum(CAST(personsInjured AS INT)), sum(CAST(pedestriansInjured AS INT)), sum(CAST(cy
    (motoristInjured AS INT)) from collision
```

sum(CAST(personsInjured AS INT))	sum(CAST(pedestriansInjured AS INT))	sum(CAST(cyclis
268335	55889	3312502

PEOPLE KILLED DURING COLLISION

FINISHED

```
%sql

select sum(CAST(personsKilled AS INT)), sum(CAST(pedestriansKilled AS INT)), sum(CAST(cycl
(motoristKilled AS INT)) from collision
```

FINISHED

sum(CAST(personsKilled AS INT))	sum(CAST(pedestriansKilled AS INT))	sum(CAST(cyclistKi
1259	706	79

From the above queries we see cyclist are most injured and least killed.where as pedestrians are least

FINISHED

injured and most killed.

```
%md
<strong>TYPE OF VEHICLE INVOLVING IN COLLISION</strong>
```

FINISHED

TYPE OF VEHICLE INVOLVING IN COLLISION

```
%sql
select  vehicle1, count(vehicle1) as count from collision where vehicle1 != 'null' group by
```

FINISHED

vehicle1
PASSENGER VEHICLE
SPORT UTILITY / STATION WAGON
TAXI
VAN
OTHER
UNKNOWN
PICK-UP TRUCK
SMALL COM VEH(4 TIRES)
LARGE COM VEH(6 OR MORE TIRES)

Passenger vehicle and Sport utility tops the list with most numbers where as scooter and pedicab are the least.

If the collision is between two vehicles vehicle type of second vehicle is also collected and shown in the table

```
%sql
select  vehicle2, count(vehicle2) as count from collision where vehicle2 != 'null' group by
```

FINISHED

vehicle2
PASSENGER VEHICLE
SPORT UTILITY / STATION WAGON
UNKNOWN
TAXI
OTHER
VAN
BICYCLE
PICK-UP TRUCK
SMALL COM VEH(4 TIRES)

FACTORS CONTRIBUTED IN COLLISION FINISHED

Now will see what are the main factors contributed for collision. First will see factors contributed by first vehicle then factors contributed by second vehicle if the collision is between two vehicles. Actual data shows factors for third and fourth vehicle also but here we are analyzing only two vehicles.

```
%sql                                                                                                FINISHED
select  factorVehicle1, count(factorVehicle1) as count from collision where factorVehicle1
    ORDER BY count DESC
```

factorVehicle1
Unspecified
Driver Inattention/Distraction
Fatigued/Drowsy
Failure to Yield Right-of-Way
Other Vehicular
Backing Unsafely
Turning Improperly
Lost Consciousness
Following Too Closely

```
%sql

select  factorVehicle2, count(factorVehicle2) as count from collision where factorVehicle2
    ORDER BY count DESC
```

FINISHED

factorVehicle2
Unspecified
Driver Inattention/Distraction
Other Vehicular
Fatigued/Drowsy
Failure to Yield Right-of-Way
Lost Consciousness
Turning Improperly
Backing Unsafely
Driver Inexperience

Factors contributing to most of the collisions are Unspecified.Next factor contributing to most collision is Driver inattention. Some of the other top contributing factors are fatigue, failure to yield and improper turning.

FINISHED

```
%md
<strong>TOTAL NUMBER OF COLLISIONS BY YEAR</strong>

We have the data from year 2012 to 2017. We will see how the number of collisions and the y
```

FINISHED

TOTAL NUMBER OF COLLISIONS BY YEAR

We have the data from year 2012 to 2017. We will see how the number of collisions and the year are related.

```
%sql
Select YEAR(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yyyy') AS TIMESTAMP)) AS year, count(YEAR(CAS
    TIMESTAMP))) as count from collision GROUP BY (YEAR(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/y
```

FINISHED

year	count
null	0
2012	100527
2013	203716
2014	205978
2015	217640
2016	227736
2017	96753

TOTAL NUMBER OF COLLISIONS BY MONTH

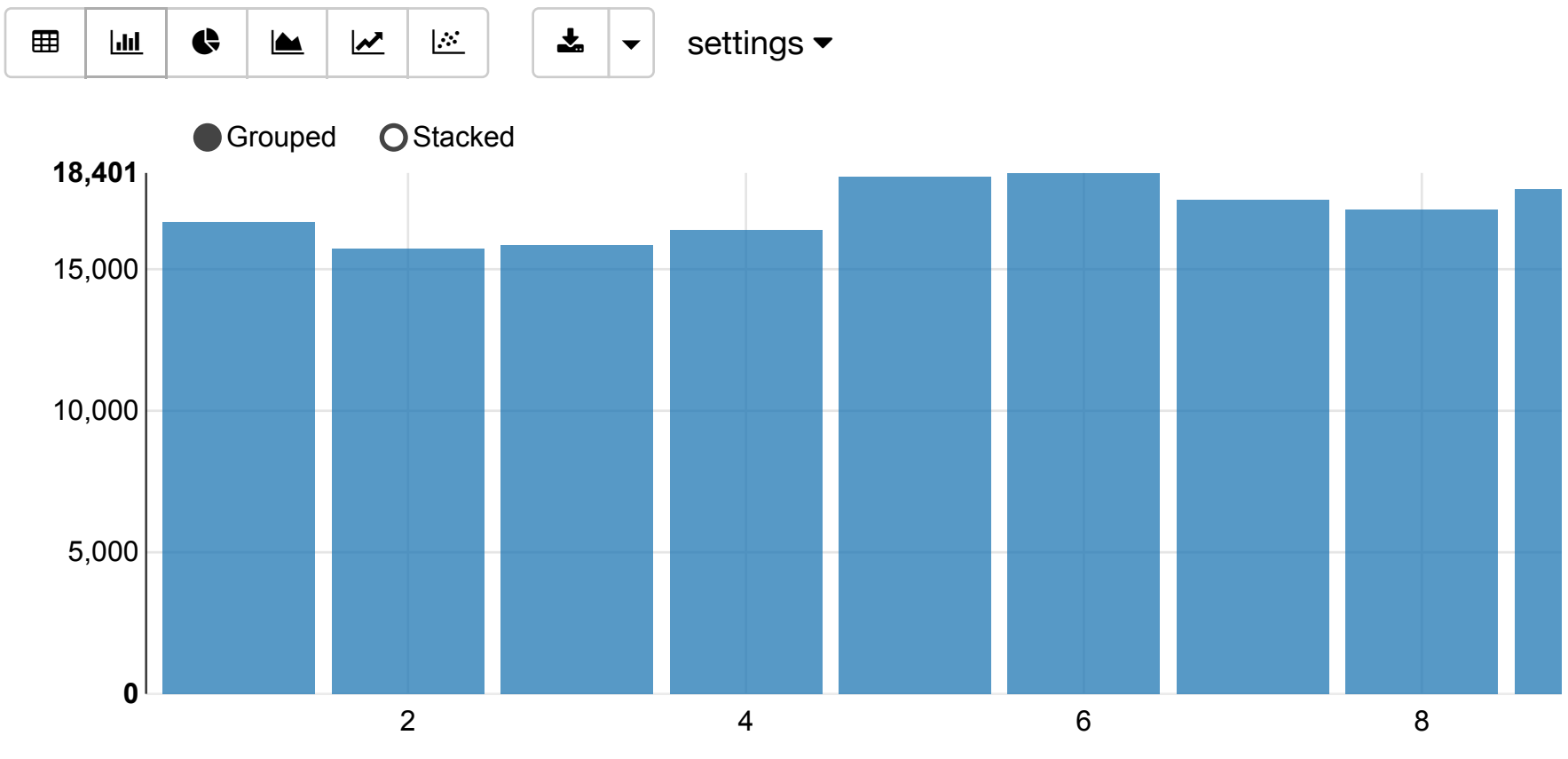
FINISHED

```
%sql
Select MONTH(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yyyy') AS TIMESTAMP)) AS month, count(MONTH(C
    AS TIMESTAMP))) as count from collision where (YEAR(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yy
    (MONTH(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yyyy') AS TIMESTAMP))) ORDER BY month ASC
```

FINISHED

month	count
1	15643
2	14396
3	16507
4	16438
5	18485
6	18204
7	17575
8	16754
9	16955

```
%sql                                                                                                                FINISHED
Select MONTH(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yyyy') AS TIMESTAMP)) AS month, count(MONTH(C
    AS TIMESTAMP))) as count from collision where (YEAR(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yy
    (MONTH(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yyyy') AS TIMESTAMP))) ORDER BY month ASC
```




```
%sql                                                                                                                FINISHED
Select MONTH(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yyyy') AS TIMESTAMP)) AS month, count(MONTH(C
    AS TIMESTAMP))) as count from collision where (YEAR(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yy
    (MONTH(CAST(UNIX_TIMESTAMP(Date, 'MM/dd/yyyy') AS TIMESTAMP))) ORDER BY month ASC
```

month	count
1	16122
2	15712
3	17948
4	16789
5	19272
6	18818
7	18764
8	18973
9	18510

```
%sql
```

READY