**Commands & Results:**

%run ./Includes/Project3-Setup

print("username: " + username)

print("userhome: " + userhome)

**Question 1 command:**

crimeDataLosAngelesDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Los-Angeles-2016.parquet")

crimeDataNewYorkDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-New-York-2016.parquet")

crimeDataPhiladelphiaDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Philadelphia-2016.parquet")

crimeDataDallasDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Dallas-2016.parquet")

crimeDataChicagoDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Chicago-2016.parquet")

crimeDataBostonDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Boston-2016.parquet")

crimeDataChicagoDF.registerTempTable("connections1")

crimeDataLosAngelesDF.registerTempTable("connections2")

crimeDataDallasDF.registerTempTable("connections3")

crimeDataNewYorkDF.registerTempTable("connections4")

crimeDataPhiladelphiaDF.registerTempTable("connections5")

crimeDataBostonDF.registerTempTable("connections6")

labels = sqlContext.sql("""

SELECT lower(city) as CITY, count(\*) AS CRIME\_COUNT

FROM connections3

GROUP BY lower(city) having lower(city) ="dallas"

union

SELECT 'BOSTON' as CITY, count(distinct(incident\_number,offense\_code)) AS CRIME\_COUNT

FROM connections6

union

SELECT 'NEWYORK' as CITY, count(\*) AS CRIME\_COUNT

FROM connections4

union

SELECT 'PHILADELPHIA' as CITY, count(\*) AS CRIME\_COUNT

FROM connections5

union

SELECT 'CHICAGO' as CITY, count(\*) AS CRIME\_COUNT

FROM connections1

union

SELECT 'LOS ANGELES' as CITY, count(\*) AS CRIME\_COUNT

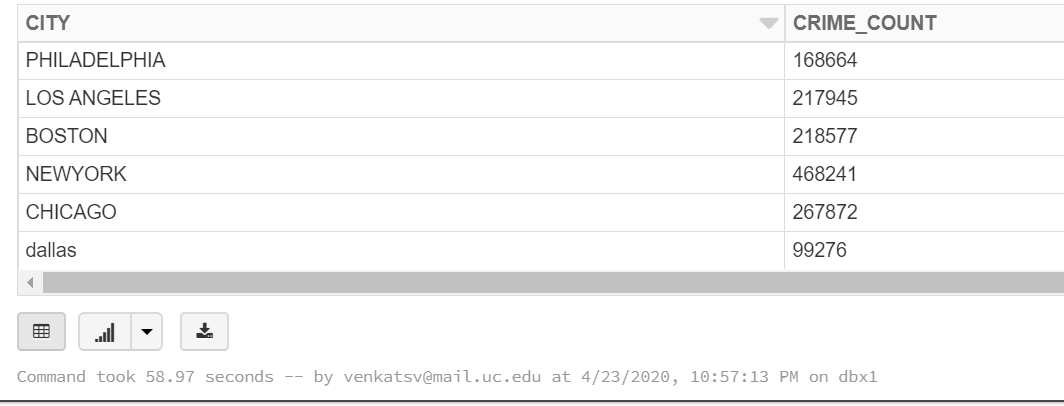
FROM connections2

""")

display(labels)

#display(crimeDataBostonDF)

**Results:**



**Question 2 command:**

crimeDataLosAngelesDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Los-Angeles-2016.parquet")

crimeDataNewYorkDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-New-York-2016.parquet")

crimeDataPhiladelphiaDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Philadelphia-2016.parquet")

crimeDataDallasDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Dallas-2016.parquet")

crimeDataChicagoDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Chicago-2016.parquet")

crimeDataBostonDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Boston-2016.parquet")

crimeDataChicagoDF.registerTempTable("connections1")

crimeDataLosAngelesDF.registerTempTable("connections2")

crimeDataDallasDF.registerTempTable("connections3")

crimeDataNewYorkDF.registerTempTable("connections4")

crimeDataPhiladelphiaDF.registerTempTable("connections5")

crimeDataBostonDF.registerTempTable("connections6")

#display(crimeDataDallasDF)

labels1 = sqlContext.sql("""

SELECT 'LOS ANGELES' as CITY, count(distinct(crimeCodeDescription)) as CRIMETYPE

FROM connections2

union

SELECT 'BOSTON' as CITY,count(distinct(OFFENSE\_CODE\_GROUP)) as CRIMETYPE

FROM connections6

union

SELECT 'CHICAGO' as CITY,count(distinct(primaryType)) as CRIMETYPE

FROM connections1

union

SELECT 'NEWYORK' as CITY,count(distinct(offenseDescription)) as CRIMETYPE

FROM connections4

union

SELECT 'PHILADELPHIA' as CITY,count(distinct(text\_general\_code)) as CRIMETYPE

FROM connections5

union

SELECT 'DALLAS' as CITY,count(distinct(typeOfIncident)) as CRIMETYPE

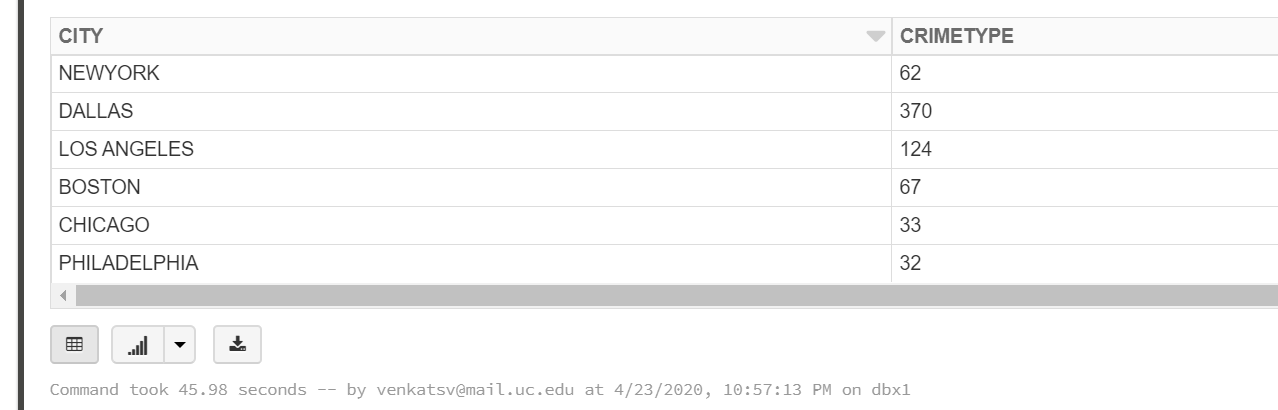
FROM connections3

where lower(city) ="dallas"

""")

display(labels1)

Results:



**Question 3** **command**:

robberyLosAngelesDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Los-Angeles-2016.parquet")

robberyPhiladelphiaDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Philadelphia-2016.parquet")

robberyDallasDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Dallas-2016.parquet")

robberyLosAngelesDF.registerTempTable("connections1")

robberyDallasDF.registerTempTable("connections2")

robberyPhiladelphiaDF.registerTempTable("connections3")

#display(crimeDataDallasDF)

labels2 = sqlContext.sql("""

SELECT 'PHILADELPHIA' as CITY, count(\*) as ROBBERYCOUNT

FROM connections3

WHERE lower(ucr\_general\_description) = "robbery"

union

SELECT 'LOS ANGELES' as CITY, count(\*) as ROBBERYCOUNT

FROM connections1

WHERE lower(crimeCodeDescription) = "robbery"

union

SELECT 'DALLAS' as CITY, count(\*) as ROBBERYCOUNT

FROM connections2

WHERE lower(typeOfIncident) LIKE 'robbery%' AND lower(city)= "dallas"

""")

display(labels2)

Results:



Question 4 command:

robberiesByMonthLosAngelesDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Los-Angeles-2016.parquet")

robberiesByMonthPhiladelphiaDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Philadelphia-2016.parquet")

robberiesByMonthDallasDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Dallas-2016.parquet")

robberiesByMonthLosAngelesDF.registerTempTable("connections1")

robberiesByMonthPhiladelphiaDF.registerTempTable("connections2")

robberiesByMonthDallasDF.registerTempTable("connections3")

labels3 = sqlContext.sql("""

(((SELECT 'PHILADELPHIA\_MIN' as CITY, (EXTRACT(MONTH FROM dispatch\_date\_time)) as MON, count(\*) as ROBBERIES

FROM connections2

WHERE lower(ucr\_general\_description) = "robbery"

GROUP BY (EXTRACT(MONTH FROM dispatch\_date\_time))

ORDER BY ROBBERIES LIMIT 1))

union

((SELECT 'PHILADELPHIA\_MAX' as CITY, (EXTRACT(MONTH FROM dispatch\_date\_time)) as MON, count(\*) as ROBBERIES

FROM connections2

WHERE lower(ucr\_general\_description) = "robbery"

GROUP BY (EXTRACT(MONTH FROM dispatch\_date\_time))

ORDER BY ROBBERIES DESC LIMIT 1)))

union

(((SELECT 'LOS ANGELES\_MIN' as CITY, (EXTRACT(MONTH FROM timeOccurred)) as MON, count(\*) as ROBBERIES

FROM connections1

WHERE lower(crimeCodeDescription) = "robbery"

GROUP BY (EXTRACT(MONTH FROM timeOccurred))

ORDER BY ROBBERIES LIMIT 1))

union

((SELECT 'LOS ANGELES\_MAX' as CITY, (EXTRACT(MONTH FROM timeOccurred)) as MON, count(\*) as ROBBERIES

FROM connections1

WHERE lower(crimeCodeDescription) = "robbery"

GROUP BY (EXTRACT(MONTH FROM timeOccurred))

ORDER BY ROBBERIES DESC LIMIT 1)))

union

(((SELECT 'DALLAS\_MIN' as CITY, (EXTRACT(MONTH FROM startingDateTime)) as MON, count(\*) as ROBBERIES

FROM connections3

WHERE lower(typeOfIncident) LIKE 'robbery%'

GROUP BY (EXTRACT(MONTH FROM startingDateTime))

ORDER BY ROBBERIES LIMIT 1))

union

((SELECT 'DALLAS\_MAX' as CITY, (EXTRACT(MONTH FROM startingDateTime)) as MON, count(\*) as ROBBERIES

FROM connections3

WHERE lower(typeOfIncident) LIKE 'robbery%'

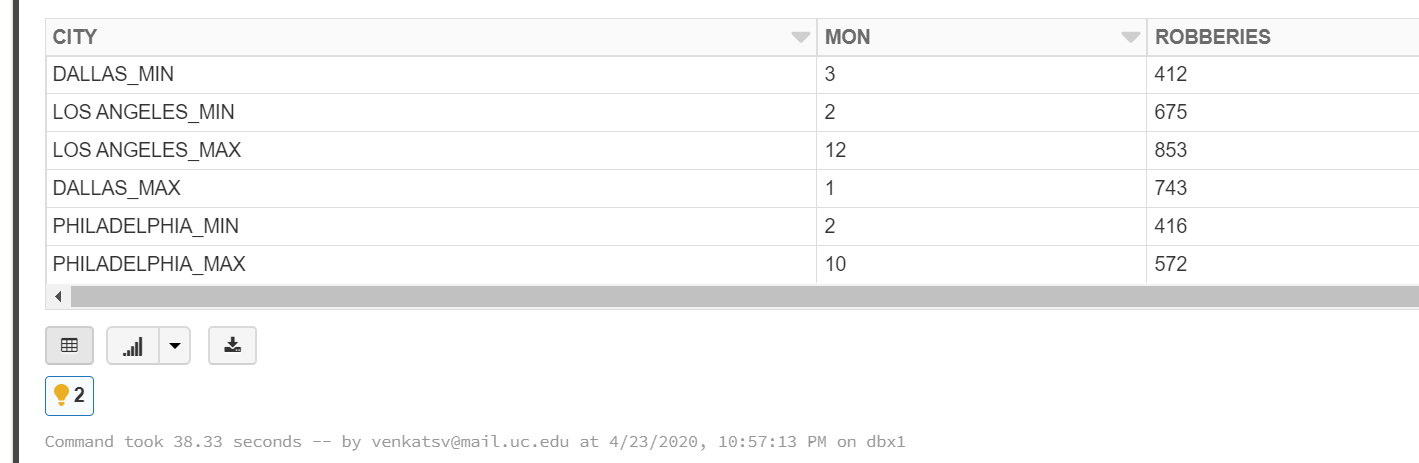
GROUP BY (EXTRACT(MONTH FROM startingDateTime))

ORDER BY ROBBERIES DESC LIMIT 1)))

""")

display(labels3)

Results:



Question 5 command:

robberiesByMonthLosAngelesDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Los-Angeles-2016.parquet")

robberiesByMonthPhiladelphiaDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Philadelphia-2016.parquet")

robberiesByMonthDallasDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Dallas-2016.parquet")

df1 = robberiesByMonthLosAngelesDF.selectExpr("timeOccurred as time", "crimeCodeDescription as crime")

df2 = robberiesByMonthPhiladelphiaDF.selectExpr("dispatch\_date\_time as time", "ucr\_general\_description as crime")

df3 = robberiesByMonthDallasDF.selectExpr("startingDateTime as time", "typeOfIncident as crime")

from pyspark.sql.functions import lit

df1 =df1.withColumn("CITY",lit("LOS ANGELES"))

df2 =df2.withColumn("CITY",lit("PHILADELPHIA"))

df3 =df3.withColumn("CITY",lit("DALLAS"))

combinedRobberiesByMonthDF=df1.union(df2).union(df3)

#display(combinedRobberiesByMonthDF)

#This now will display City, timestamp, robbery

combinedRobberiesByMonthDF.registerTempTable("connections1")

labels4 = sqlContext.sql("""

(((SELECT (EXTRACT(MONTH FROM time)) as MONTH, count(\*) as ROBBERIES

FROM connections1

WHERE (lower(crime)= "robbery" and lower(city)="los angeles") OR ((lower(crime)= "robbery" and lower(city)="philadelphia")) or (lower(crime) LIKE "robbery%" and lower(city)="dallas")

GROUP BY (EXTRACT(MONTH FROM time))

ORDER BY ROBBERIES LIMIT 1))

union

((SELECT (EXTRACT(MONTH FROM time)) as MONTH, count(\*) as ROBBERIES

FROM connections1

WHERE (lower(crime)= "robbery" and lower(city)="los angeles") OR ((lower(crime)= "robbery" and lower(city)="philadelphia")) or (lower(crime) LIKE "robbery%" and lower(city)="dallas")

GROUP BY (EXTRACT(MONTH FROM time))

ORDER BY ROBBERIES DESC LIMIT 1)))

""")

display(labels4)

Result:



Question 6 command:

robberiesByMonthLosAngelesDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Los-Angeles-2016.parquet")

robberiesByMonthPhiladelphiaDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Philadelphia-2016.parquet")

robberiesByMonthDallasDF = spark.read.parquet("dbfs:/mnt/training/crime-data-2016/Crime-Data-Dallas-2016.parquet")

df1 = robberiesByMonthLosAngelesDF.selectExpr("timeOccurred as time", "crimeCodeDescription as crime")

df2 = robberiesByMonthPhiladelphiaDF.selectExpr("dispatch\_date\_time as time", "ucr\_general\_description as crime")

df3 = robberiesByMonthDallasDF.selectExpr("startingDateTime as time", "typeOfIncident as crime")

from pyspark.sql.functions import lit

df1 =df1.withColumn("CITY",lit("LOS ANGELES"))

df2 =df2.withColumn("CITY",lit("PHILADELPHIA"))

df3 =df3.withColumn("CITY",lit("DALLAS"))

combinedRobberiesByMonthDF=df1.union(df2).union(df3)

#display(combinedRobberiesByMonthDF)

#This now will display City, timestamp, robbery

combinedRobberiesByMonthDF.registerTempTable("connections1")

labels5 = sqlContext.sql("""

SELECT \* FROM ((SELECT CITY, (EXTRACT(MONTH FROM time)) as MONTH, count(\*) as ROBBERIES

FROM connections1

WHERE lower(crime) = "robbery" AND lower(city) = "los angeles"

GROUP BY (EXTRACT(MONTH FROM time)),CITY

)

union

(SELECT CITY, (EXTRACT(MONTH FROM time)) as MONTH, count(\*) as ROBBERIES

FROM connections1

WHERE lower(crime) LIKE "robbery%" AND lower(city) = "dallas"

GROUP BY (EXTRACT(MONTH FROM time)),CITY

)

union

(SELECT CITY, (EXTRACT(MONTH FROM time)) as MONTH, count(\*) as ROBBERIES

FROM connections1

WHERE lower(crime) = "robbery" AND lower(city) = "philadelphia"

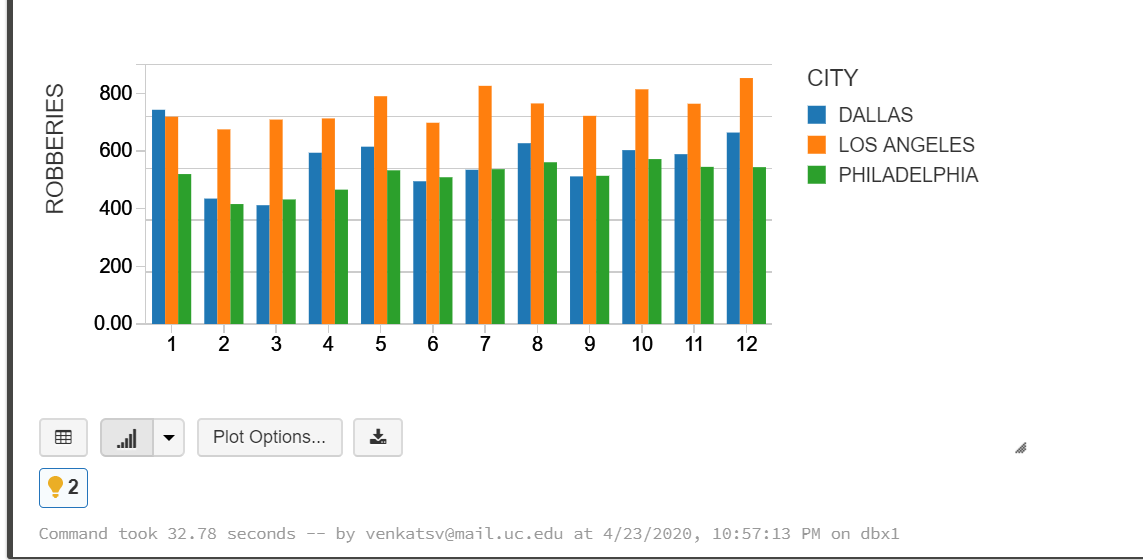
GROUP BY (EXTRACT(MONTH FROM time)),CITY

)) ORDER BY MONTH,CITY

""")

display(labels5)

Result 6:



Question 7 command:

cityDataDF = spark.read.parquet("dbfs:/mnt/training/City-Data.parquet")

cityDataDF.registerTempTable("connections1")

labels5.registerTempTable("connections2")

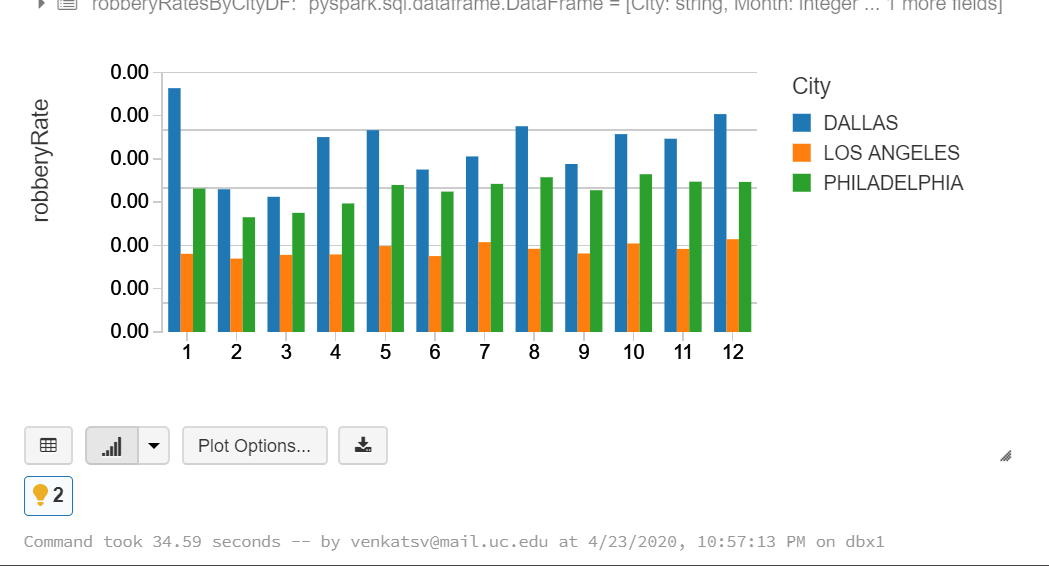
robberyRatesByCityDF = sqlContext.sql("""

select connections2.City, connections2.Month, (connections2.robberies/connections1.estPopulation2016) as robberyRate from connections2 join connections1 where lower(connections1.city) = lower(connections2.City)

""")

display(robberyRatesByCityDF)

Result :



Question 8 command:

crimeDataNewYorkDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-New-York-2016.parquet")

crimeDataBostonDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Boston-2016.parquet")

crimeDataNewYorkDF.registerTempTable("connections1")

crimeDataBostonDF.registerTempTable("connections2")

labels6 = sqlContext.sql("""

((SELECT 'New York' as CITY, (EXTRACT(MONTH FROM reportDate)) as MON, count(\*) as HOMICIDE

FROM connections1

WHERE lower(offenseDescription) LIKE "homicide%" or lower(offenseDescription) LIKE "murder%"

GROUP BY (EXTRACT(MONTH FROM reportDate))

)

union

((SELECT 'BOSTON' as CITY, MONTH as MON, count(\*) as HOMICIDE

FROM connections2

WHERE lower(OFFENSE\_CODE\_GROUP) = "homicide"

GROUP BY (MONTH))

))

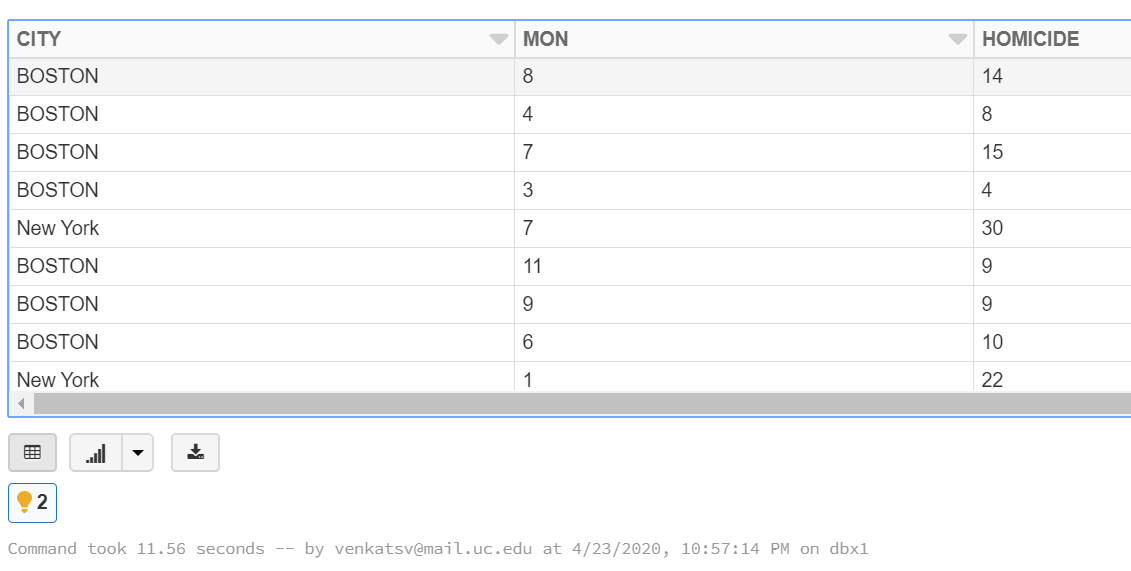
""")

display(labels6)

Result:

CITY MON HOMICIDE

|  |  |  |
| --- | --- | --- |
| BOSTON | 8 | 14 |
| BOSTON | 4 | 8 |
| BOSTON | 7 | 15 |
| BOSTON | 3 | 4 |
| New York | 7 | 30 |
| BOSTON | 11 | 9 |
| BOSTON | 9 | 9 |
| BOSTON | 6 | 10 |
| New York | 1 | 22 |
| New York | 10 | 20 |
| New York | 2 | 17 |
| New York | 11 | 21 |
| BOSTON | 10 | 9 |
| New York | 9 | 34 |
| New York | 4 | 28 |
| BOSTON | 12 | 10 |
| New York | 6 | 32 |
| BOSTON | 5 | 7 |
| New York | 12 | 22 |
| BOSTON | 1 | 7 |
| New York | 5 | 31 |
| BOSTON | 2 | 4 |
| New York | 3 | 25 |
| New York | 8 | 36 |



command:

labels6.registerTempTable("connections1")

labels7 = sqlContext.sql("""

((SELECT MON AS MONTH,HOMICIDE

FROM connections1

GROUP BY MONTH,HOMICIDE

ORDER BY HOMICIDE ASC LIMIT 1))

union

((SELECT MON AS MONTH, HOMICIDE

FROM connections1

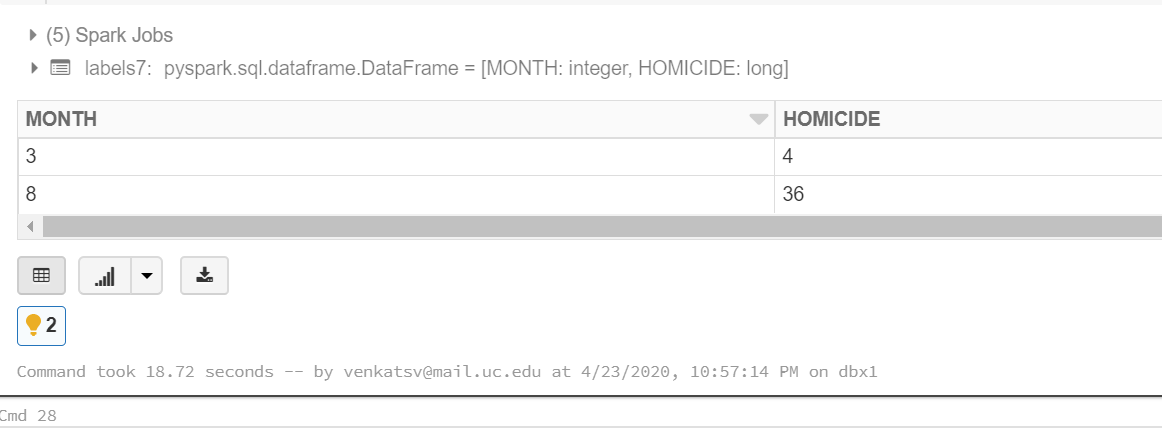
GROUP BY MONTH,HOMICIDE

ORDER BY HOMICIDE DESC LIMIT 1))

""")

display(labels7)

Result :



Question 9 command:

cityDataDF.registerTempTable("connections1")

labels6.registerTempTable("labels6")

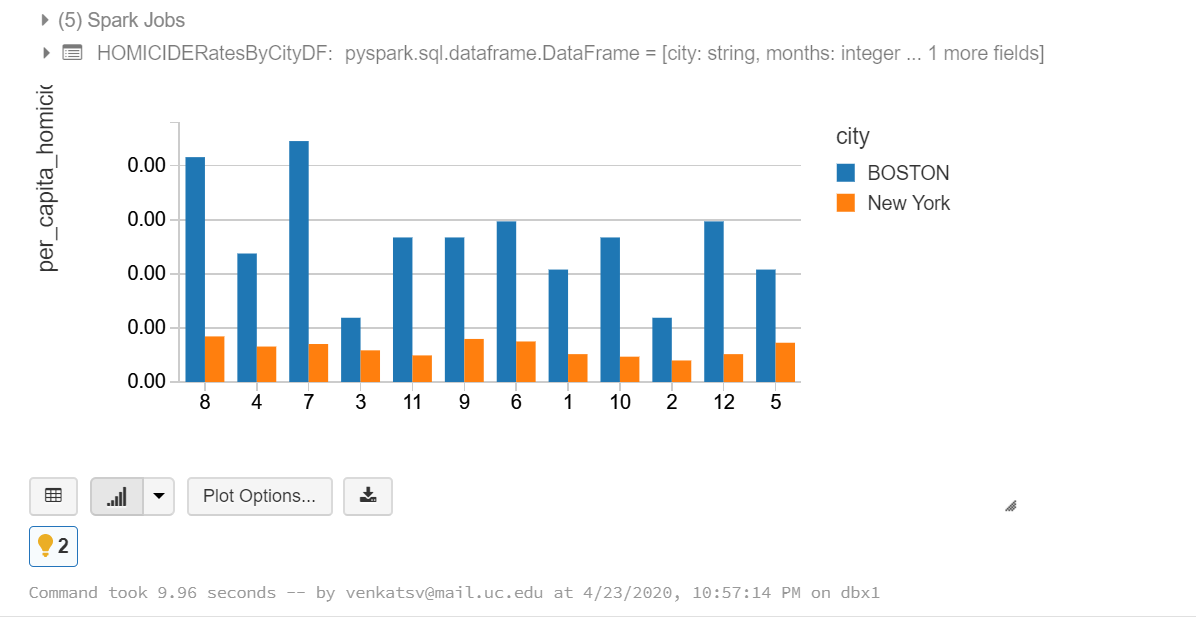
HOMICIDERatesByCityDF = sqlContext.sql("""

select labels6.city, labels6.mon as months, ((labels6.homicide/connections1.estPopulation2016)) as per\_capita\_homicide\_rates\_month from labels6 join connections1 where lower(connections1.city) = lower(labels6.city)

""")

display(HOMICIDERatesByCityDF)

Result:



Question 10 Result :

We can predict the monthly robbery count using Linear Regression algorithm. It is used to estimate real values based on continuous variables.

From month 7 to month 12 more crimes are occuring and month 12 is getting highest crime. This algorithm can tell us that the highest and lowest values per month based on the input from the year 2016.

The formula to calculate linear regression is:

y=θ1+θ2.x

where x is input training data, y is the output, θ1 is the intercept, θ2 is the coefficient of x

