Spark queries using RDD

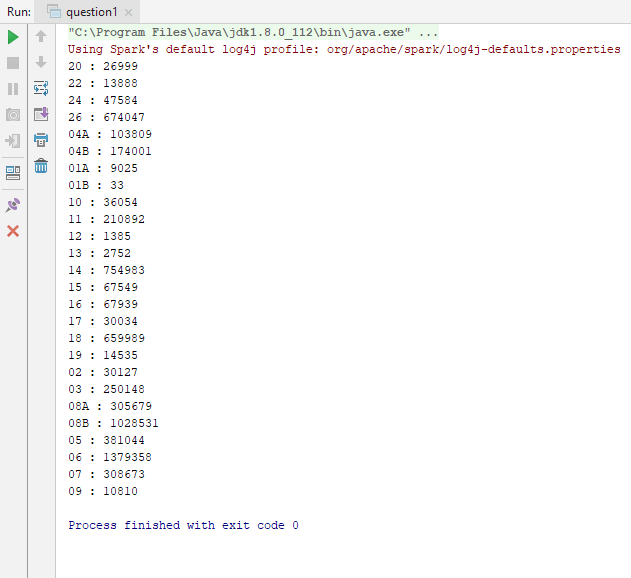
Question 1

Find number of crimes that happened under each FBI code.

Code:

**package** yorbit.rdd  
  
**import** com.sparkTutorial.commons.Utils  
**import** com.sparkTutorial.pairRdd.aggregation.reducebykey.housePrice.AvgCount  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** question1 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** conf = **new** SparkConf().setAppName(**"crime"**).setMaster(**"local[\*]"**)  
 **val** sc = **new** SparkContext(conf)  
  
 **val** data = sc.textFile(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
 **val** cleanData = data.mapPartitionsWithIndex { (idx, iter) => **if** (idx == 0) iter.drop(1) **else** iter }  
  
 **val** cleanDataRdd = cleanData.map((line: String) => (line.split(Utils.*COMMA\_DELIMITER*)(14), 1))  
  
 **val** fbiCodeCount = cleanDataRdd.reduceByKey((x, y) => x + y)  
 **for** ((code, total) <- fbiCodeCount.collect()) *println*(code + **" : "** + total)  
  
 }  
}

Output:



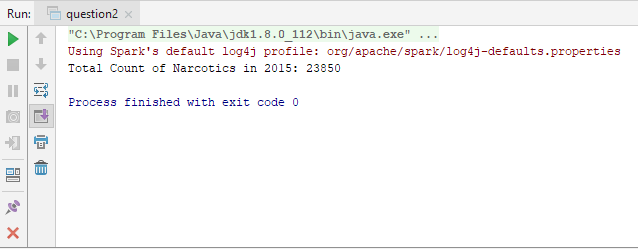
Question 2

Find number of ‘NARCOTICS’ cases filed in the year 2015.

Code :

**package** yorbit.rdd  
  
**import** com.sparkTutorial.commons.Utils  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** question2 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** conf = **new** SparkConf().setAppName(**"crime"**).setMaster(**"local[\*]"**)  
 **val** sc = **new** SparkContext(conf)  
  
 **val** data = sc.textFile(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
 **val** cleanData = data.mapPartitionsWithIndex { (idx, iter) => **if** (idx == 0) iter.drop(1) **else** iter }  
  
  
 **val** filteredData = cleanData.filter((line: String) =>  
 line.split(Utils.*COMMA\_DELIMITER*)(5) == **"NARCOTICS"** && line.split(Utils.*COMMA\_DELIMITER*)(17) == **"2015"**)  
  
 *println*(**"Total Count of Narcotics in 2015: "** + filteredData.collect().length)  
  
 }  
}

Output:



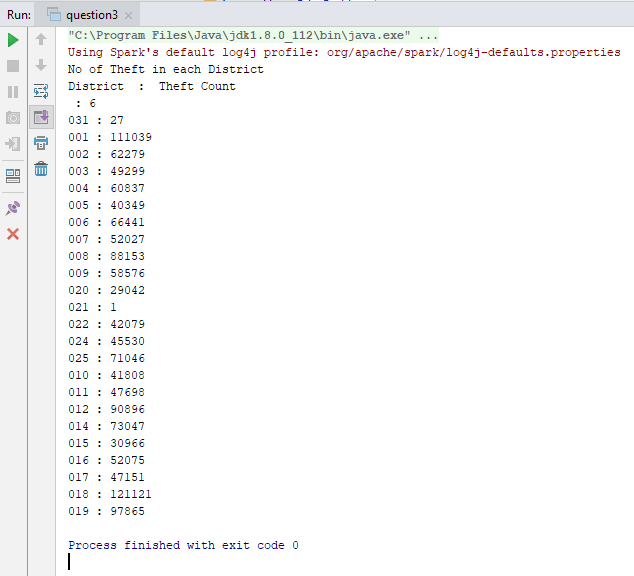
Question 3

Find the number of theft related arrests that happened in each district.

Code:

**package** yorbit.rdd  
  
**import** com.sparkTutorial.commons.Utils  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** question3 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** conf = **new** SparkConf().setAppName(**"crime"**).setMaster(**"local[\*]"**)  
 **val** sc = **new** SparkContext(conf)  
  
 **val** data = sc.textFile(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
 **val** cleanData = data.mapPartitionsWithIndex { (idx, iter) => **if** (idx == 0) iter.drop(1) **else** iter }  
  
 **val** theftData = cleanData.filter((line: String) =>  
 line.split(Utils.*COMMA\_DELIMITER*)(5) == **"THEFT"**).map((line: String) => (line.split(Utils.*COMMA\_DELIMITER*)(11), 1)).reduceByKey((x, y) => x + y)  
  
 *println*(**"No of Theft in each District"**)  
 *println*(**"District : Theft Count"**)  
 **val** theftCount = theftData.reduceByKey((x, y) => x + y)  
 **for** ((code, total) <- theftData.collect()) *println*(code + **" : "** + total)  
  
 }  
}

Output:



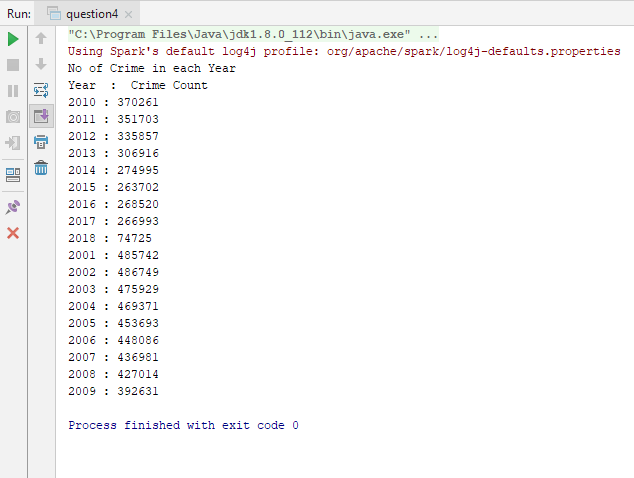
Question 4

Find number of crimes happened per year

Code:

**package** yorbit.rdd  
  
**import** com.sparkTutorial.commons.Utils  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** question4 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** conf = **new** SparkConf().setAppName(**"crime"**).setMaster(**"local[\*]"**)  
 **val** sc = **new** SparkContext(conf)  
  
 **val** data = sc.textFile(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
 **val** cleanData = data.mapPartitionsWithIndex { (idx, iter) => **if** (idx == 0) iter.drop(1) **else** iter }  
  
 **val** theftData = cleanData.map((line: String) => (line.split(Utils.*COMMA\_DELIMITER*)(17), 1)).reduceByKey((x, y) => x + y)  
  
 *println*(**"No of Crime in each Year"**)  
 *println*(**"Year : Crime Count"**)  
 **val** theftCount = theftData.reduceByKey((x, y) => x + y)  
 **for** ((code, total) <- theftData.collect()) *println*(code + **" : "** + total)  
  
 }  
}

Output:



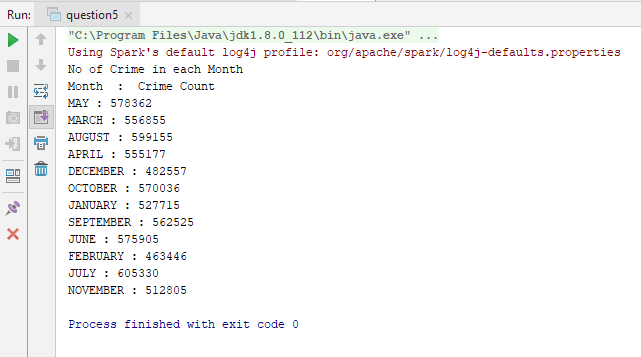
Question 5

Find number of crimes per month.

Code:

**package** yorbit.rdd  
  
**import** com.sparkTutorial.commons.Utils  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** question5 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** conf = **new** SparkConf().setAppName(**"crime"**).setMaster(**"local[\*]"**)  
 **val** sc = **new** SparkContext(conf)  
  
 **val** data = sc.textFile(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
 **val** cleanData = data.mapPartitionsWithIndex { (idx, iter) => **if** (idx == 0) iter.drop(1) **else** iter }  
  
 **val** theftData = cleanData.map((line: String) => {  
 **val** dateFormat = **"MM/dd/yyyy hh:mm:ss a"  
 val** dtf = java.time.format.DateTimeFormatter.*ofPattern*(dateFormat)  
 **val** d = java.time.LocalDate.*parse*(line.split(Utils.*COMMA\_DELIMITER*)(2), dtf)  
 **val** month = d.getMonth.toString  
 (month, 1)  
 }).reduceByKey((x, y) => x + y)  
  
 *println*(**"No of Crime in each Month"**)  
 *println*(**"Month : Crime Count"**)  
 **val** theftCount = theftData.reduceByKey((x, y) => x + y)  
 **for** ((code, total) <- theftData.collect()) *println*(code + **" : "** + total)  
  
 }  
}

Output:



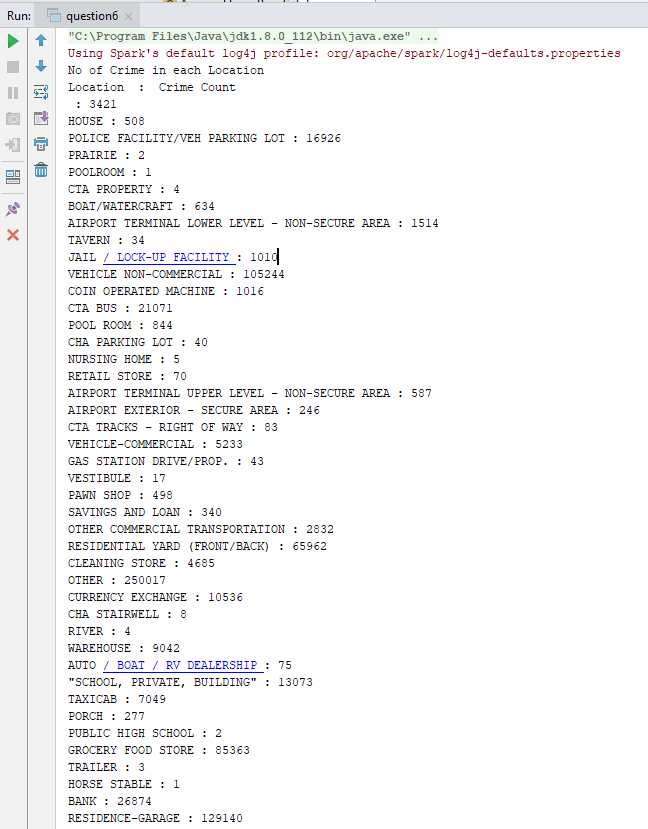
Question 6

Where do most crimes take pace?

Code:

**package** yorbit.rdd  
  
**import** com.sparkTutorial.commons.Utils  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** question6 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** conf = **new** SparkConf().setAppName(**"crime"**).setMaster(**"local[\*]"**)  
 **val** sc = **new** SparkContext(conf)  
  
 **val** data = sc.textFile(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
 **val** cleanData = data.mapPartitionsWithIndex { (idx, iter) => **if** (idx == 0) iter.drop(1) **else** iter }  
  
 **val** theftData = cleanData.map((line: String) => (line.split(Utils.*COMMA\_DELIMITER*)(7), 1)).reduceByKey((x, y) => x + y)  
  
 *println*(**"No of Crime in each Location"**)  
 *println*(**"Location : Crime Count"**)  
 **val** theftCount = theftData.reduceByKey((x, y) => x + y)  
 **for** ((code, total) <- theftData.collect()) *println*(code + **" : "** + total)  
 }  
}

Output:



Spark SQL queries

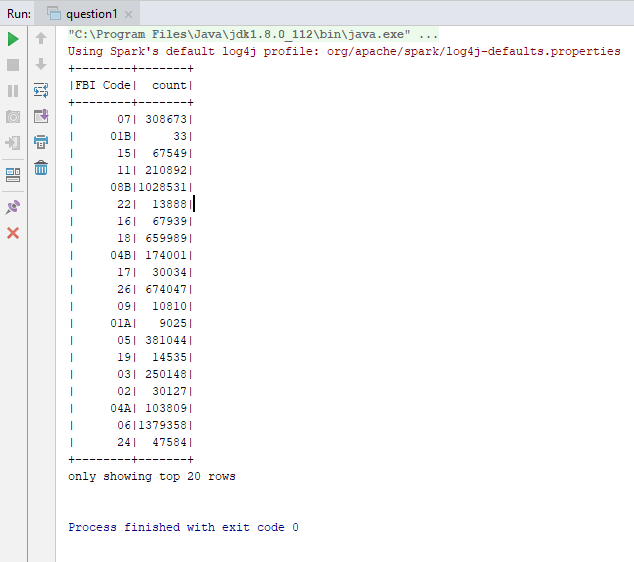
Question 1

Find number of crimes that happened under each FBI code.

Code:

**package** yorbit.sql  
  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.sql.SparkSession  
  
**object** question1 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** session = SparkSession.*builder*().appName(**"crime"**).master(**"local[\*]"**).getOrCreate()  
  
 **val** dataFrameReader = session.read  
  
 **val** responses = dataFrameReader  
 .option(**"header"**, **"true"**)  
 .option(**"inferSchema"**, value = **true**)  
 .csv(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
  
 **val** responseWithSelectedColumns = responses.select(**"FBI Code"**)  
  
 **val** groupedDataset = responseWithSelectedColumns.groupBy(**"FBI Code"**)  
 groupedDataset.count().show()  
 }  
}

Output:



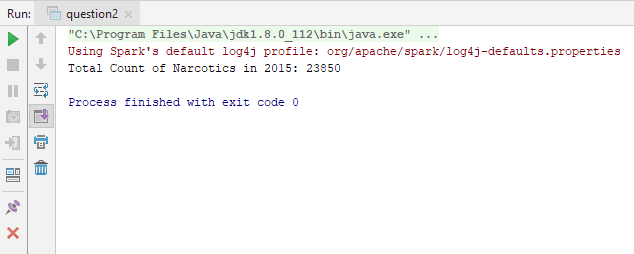
Question 2

Find number of ‘NARCOTICS’ cases filed in the year 2015.

Code :

**package** yorbit.sql  
  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.sql.SparkSession  
  
**object** question2 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** session = SparkSession.*builder*().appName(**"crime"**).master(**"local[\*]"**).getOrCreate()  
  
 **val** dataFrameReader = session.read  
  
 **val** responses = dataFrameReader  
 .option(**"header"**, **"true"**)  
 .option(**"inferSchema"**, value = **true**)  
 .csv(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
  
 **val** responseWithSelectedColumns = responses.select(**"Primary Type"**, **"Year"**)  
  
 **val** filterData=responseWithSelectedColumns.filter(responseWithSelectedColumns.col(**"Primary Type"**).===(**"NARCOTICS"**) && responseWithSelectedColumns.col(**"Year"**).===(**"2015"**))  
 *println*(**"Total Count of Narcotics in 2015: "** + filterData.count())  
  
 }  
}

Output:



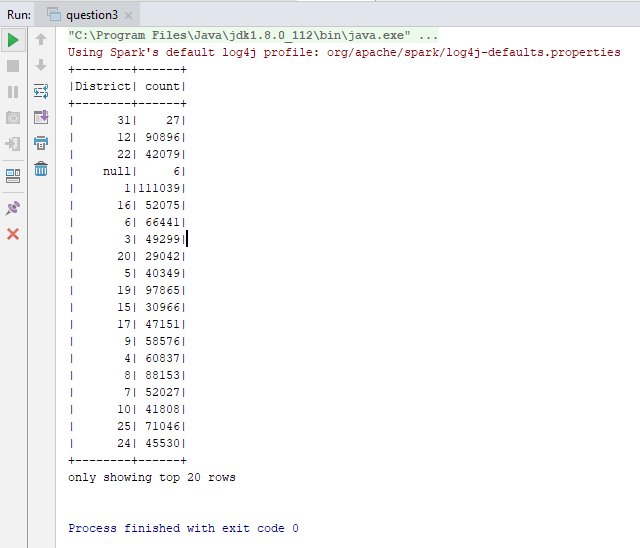
Question 3

Find the number of theft related arrests that happened in each district.

Code:

**package** yorbit.sql  
  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.sql.SparkSession  
  
**object** question3 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** session = SparkSession.*builder*().appName(**"crime"**).master(**"local[\*]"**).getOrCreate()  
  
 **val** dataFrameReader = session.read  
  
 **val** responses = dataFrameReader  
 .option(**"header"**, **"true"**)  
 .option(**"inferSchema"**, value = **true**)  
 .csv(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
  
 **val** responseWithSelectedColumns = responses.select(**"District"**, **"Primary Type"**)  
  
 **val** filterData = responseWithSelectedColumns.filter(responseWithSelectedColumns.col(**"Primary Type"**).===(**"THEFT"**))  
  
 **val** groupedDataset = filterData.groupBy(**"District"**)  
 groupedDataset.count().show()  
 }  
}

Output:



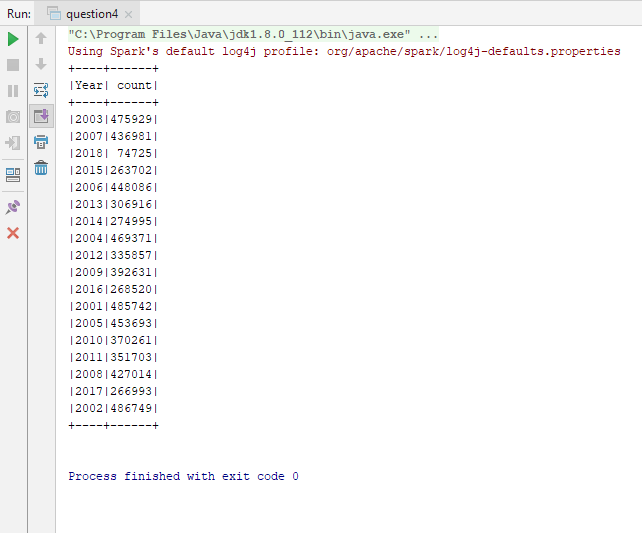
Question 4

Find number of crimes happened per year

Code:

**package** yorbit.sql  
  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.sql.SparkSession  
  
**object** question4 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** session = SparkSession.*builder*().appName(**"crime"**).master(**"local[\*]"**).getOrCreate()  
  
 **val** dataFrameReader = session.read  
  
 **val** responses = dataFrameReader  
 .option(**"header"**, **"true"**)  
 .option(**"inferSchema"**, value = **true**)  
 .csv(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
  
 **val** responseWithSelectedColumns = responses.select(**"Year"**)  
  
 **val** groupedDataset = responseWithSelectedColumns.groupBy(**"Year"**)  
 groupedDataset.count().show()  
  
  
 }  
}

Output:



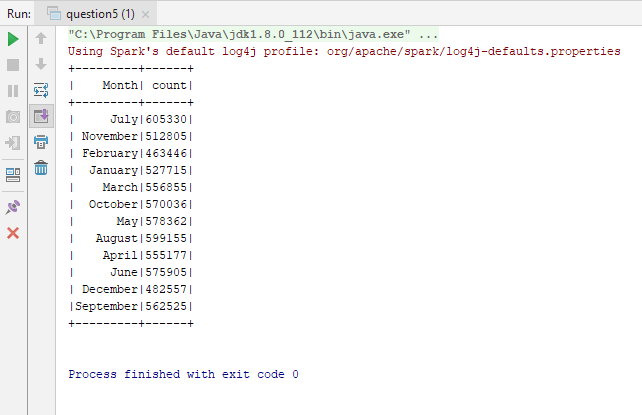
Question 5

Find number of crimes per month.

Code:

**package** yorbit.sql  
  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.sql.SparkSession  
**import** org.apache.spark.sql.functions.{*col*, from\_unixtime, unix\_timestamp}  
  
**object** question5 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** session = SparkSession.*builder*().appName(**"crime"**).master(**"local[\*]"**).getOrCreate()  
  
 **val** dataFrameReader = session.read  
  
 **val** responses = dataFrameReader  
 .option(**"header"**, **"true"**)  
 .option(**"inferSchema"**, value = **true**)  
 .csv(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
  
 **val** responseWithSelectedColumns = responses.select(**"Date"**)  
  
 **val** responseWithMonth = responseWithSelectedColumns.withColumn(**"Month"**, *from\_unixtime*(*unix\_timestamp*(*col*(**"Date"**), **"MM/dd/yyyy hh:mm:ss a"**), **"MMMMM"**))  
  
 **val** groupedDataset = responseWithMonth.groupBy(**"Month"**)  
 groupedDataset.count().show()  
  
 }  
}

Output:



Question 6

Where do most crimes take pace?

Code:

**package** yorbit.sql  
  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.sql.SparkSession  
  
**object** question6 {  
 **def** main(args: Array[String]) {  
  
 Logger.*getLogger*(**"org"**).setLevel(Level.*ERROR*)  
 **val** session = SparkSession.*builder*().appName(**"crime"**).master(**"local[\*]"**).getOrCreate()  
  
 **val** dataFrameReader = session.read  
  
 **val** responses = dataFrameReader  
 .option(**"header"**, **"true"**)  
 .option(**"inferSchema"**, value = **true**)  
 .csv(**"in/Crimes\_-\_2001\_to\_present.csv"**)  
  
 **val** responseWithSelectedColumns = responses.select(**"Description"**, **"Primary Type"**)  
  
 **val** groupedDataset = responseWithSelectedColumns.groupBy(**"Description"**)  
 groupedDataset.count().show(20, **false**)  
 }  
}

Output:

