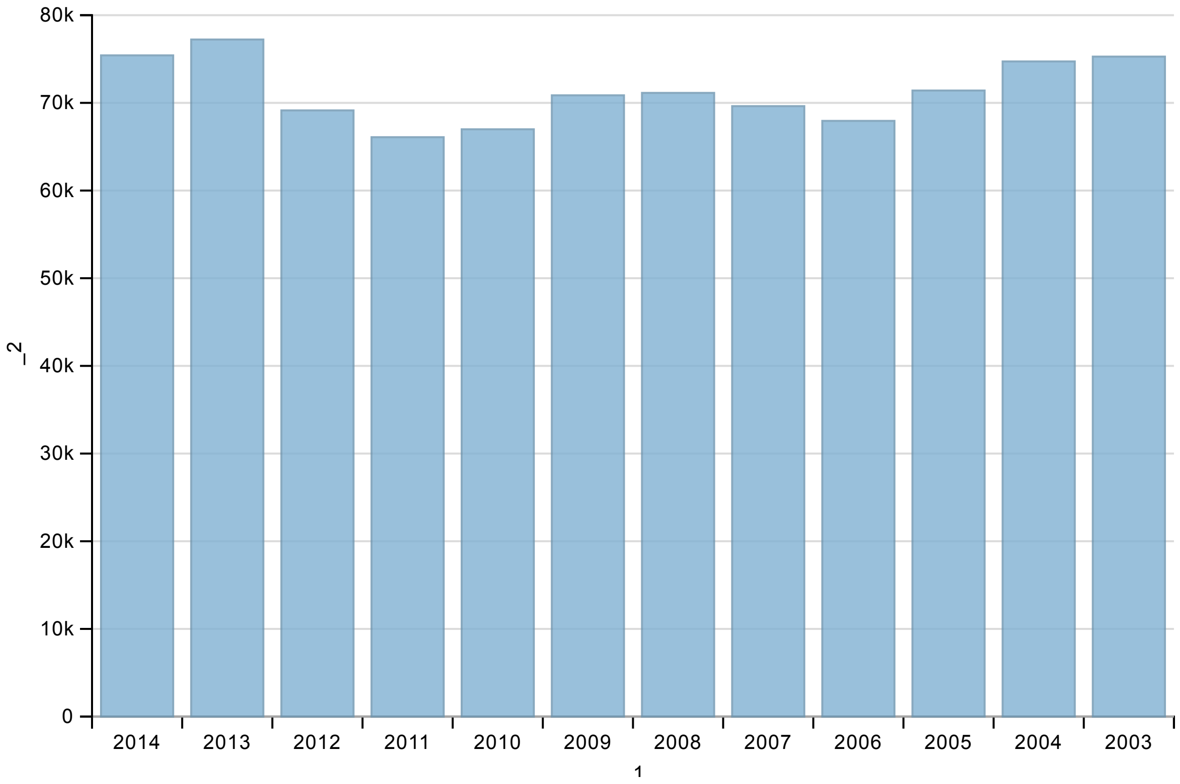
SFC

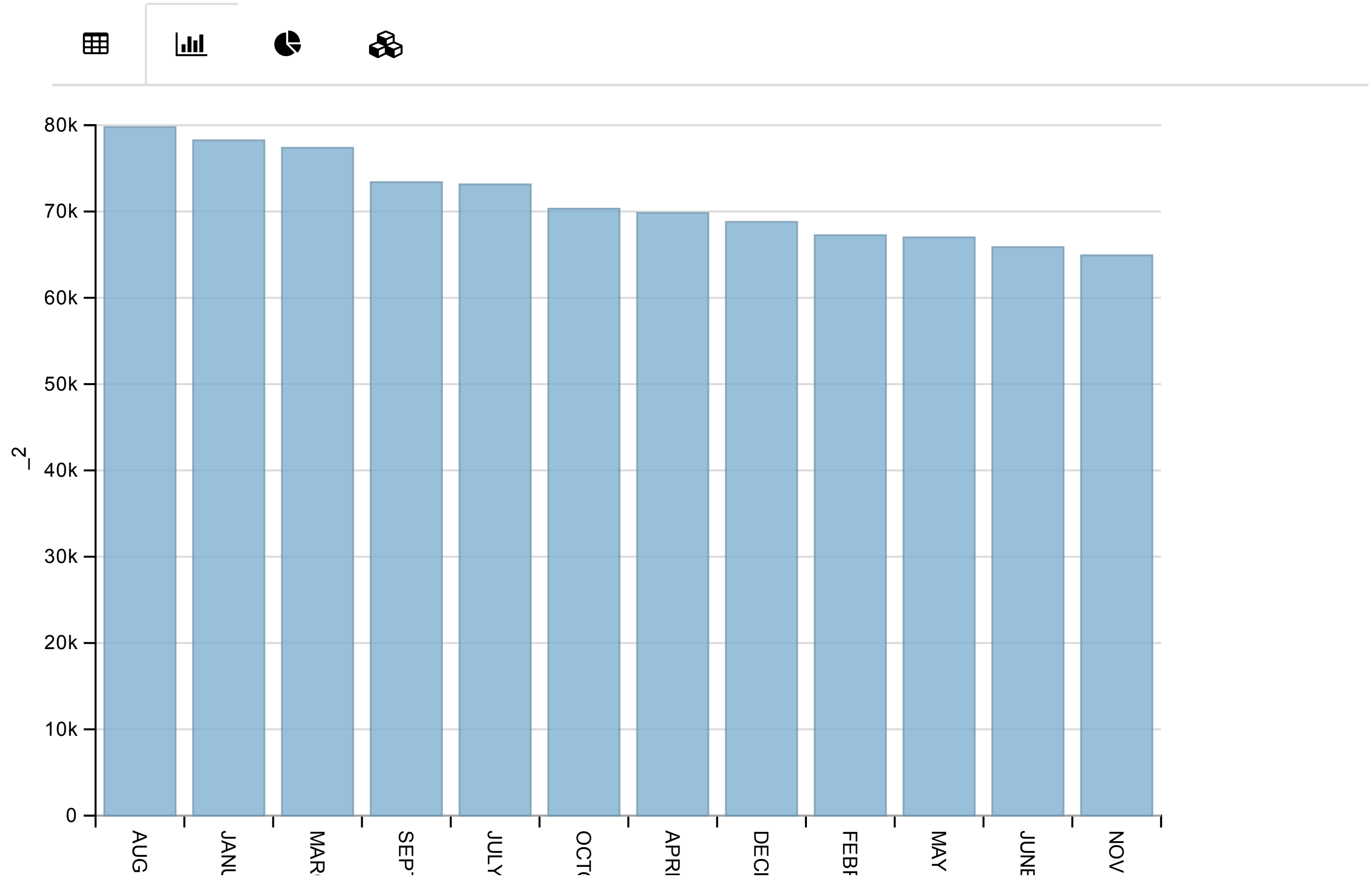
```
val rdd = sparkContext.textFile("sfc/test.csv")
//val descriptionIndex = rdd.first.split(",").zipWithIndex
val rows = rdd.mapPartitionsWithIndex{ (idx, iter) => if (idx == 0) iter.drop(1) else iter }.ma
//val samples3 = rows.takeSample(true, 3)
```

```
//rows.first.split(",").zipWithIndex
```

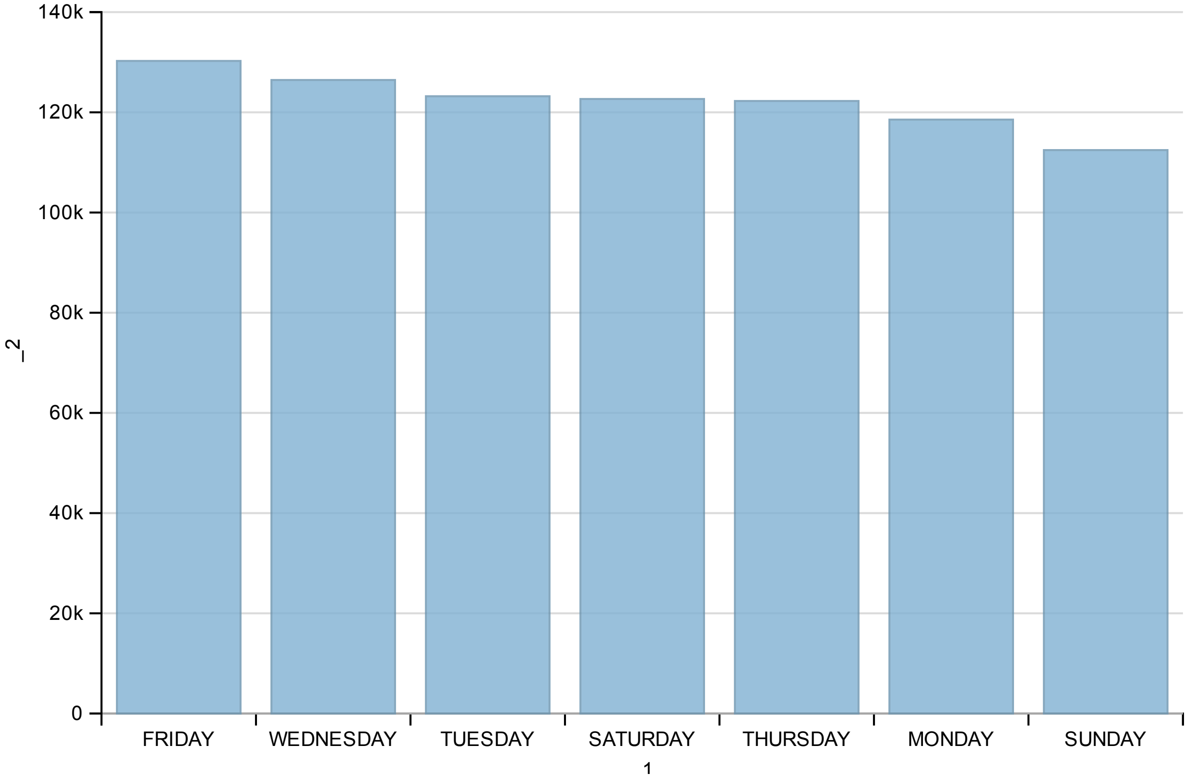
```
import java.time._
import java.time.format._

val splits = rows
      .map(x => (LocalDateTime.parse(x(1), DateTimeFormatter.ofPattern("yyyy-MM-dd kk:mm:ss")),
      .filter(_._1.getYear != 2015).cache

splits.groupBy(_._1.getYear).mapValues(_.size).sortBy(_._1, false).collect
//dates.groupBy(line => line.getYear).mapValues(x => x.size).map(x => x.swap).sortByKey().colle
```



```
splits.groupBy(_._1.getMonth).mapValues(_.size).sortBy(_._2, false).collect
```

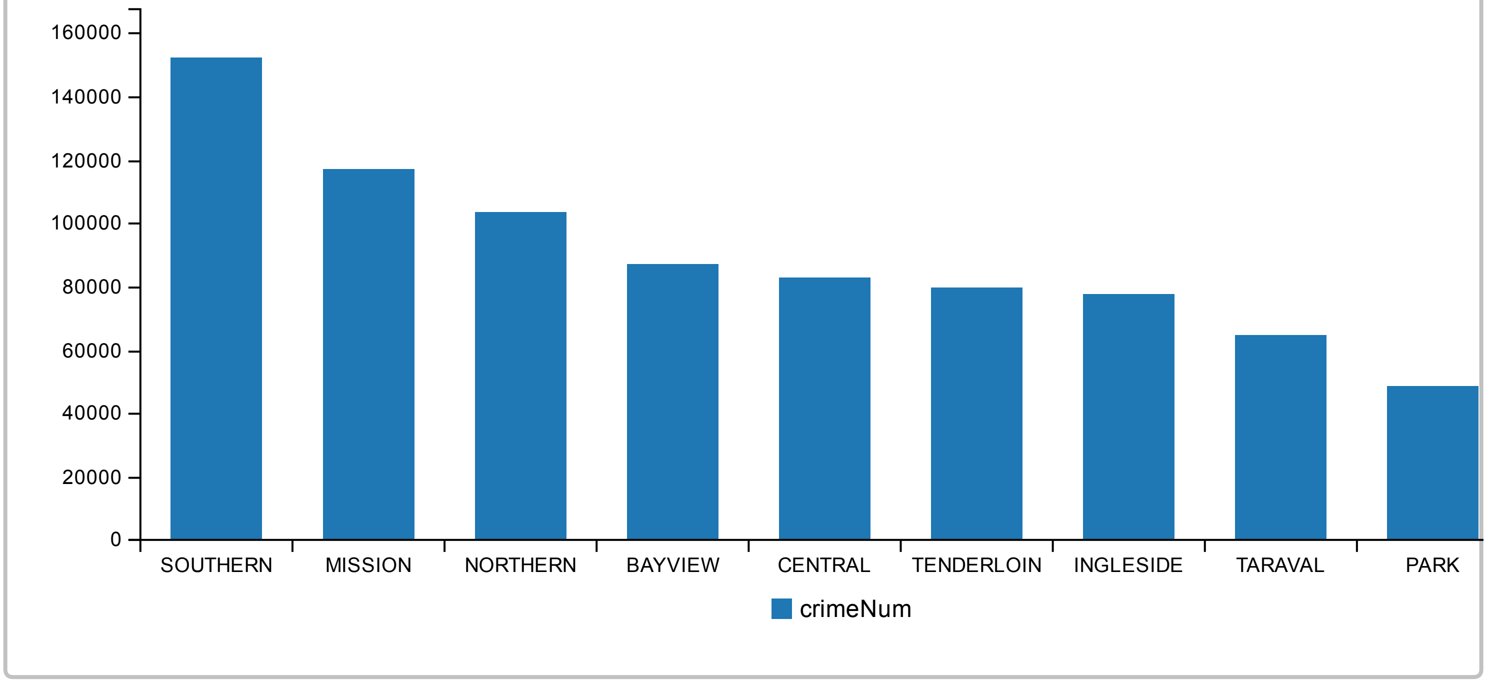


```
splits.groupBy(_._1.getDayOfWeek).mapValues(_.size).sortBy(_._2, false).collect
```



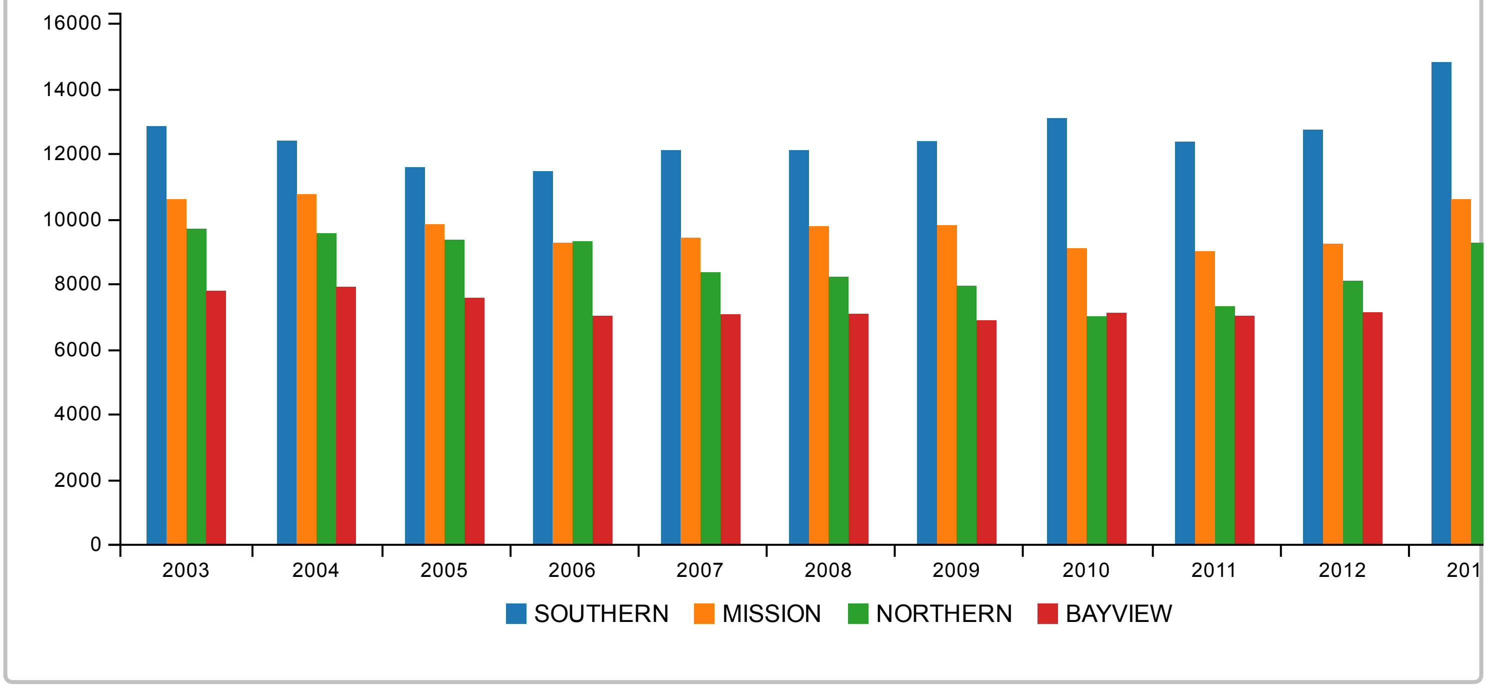
```
case class BlkCrime(blkName: String, crimeNum: Int)

val topCrAdd = splits.map(x => (x._2, 1)).reduceByKey(_+_).sortBy(_._2, false).cache
val top10CrimeBlk = topCrAdd.map{case (blkName, crimeNum) => BlkCrime(blkName, crimeNum)}.colle
```

```
CustomC3Chart(top10CrimeBlk,
             chartOptions = """
             { data: {x: 'blkName', type: 'bar'},
               axis: {x: {type: 'categorical'}}
             }
             """)
```

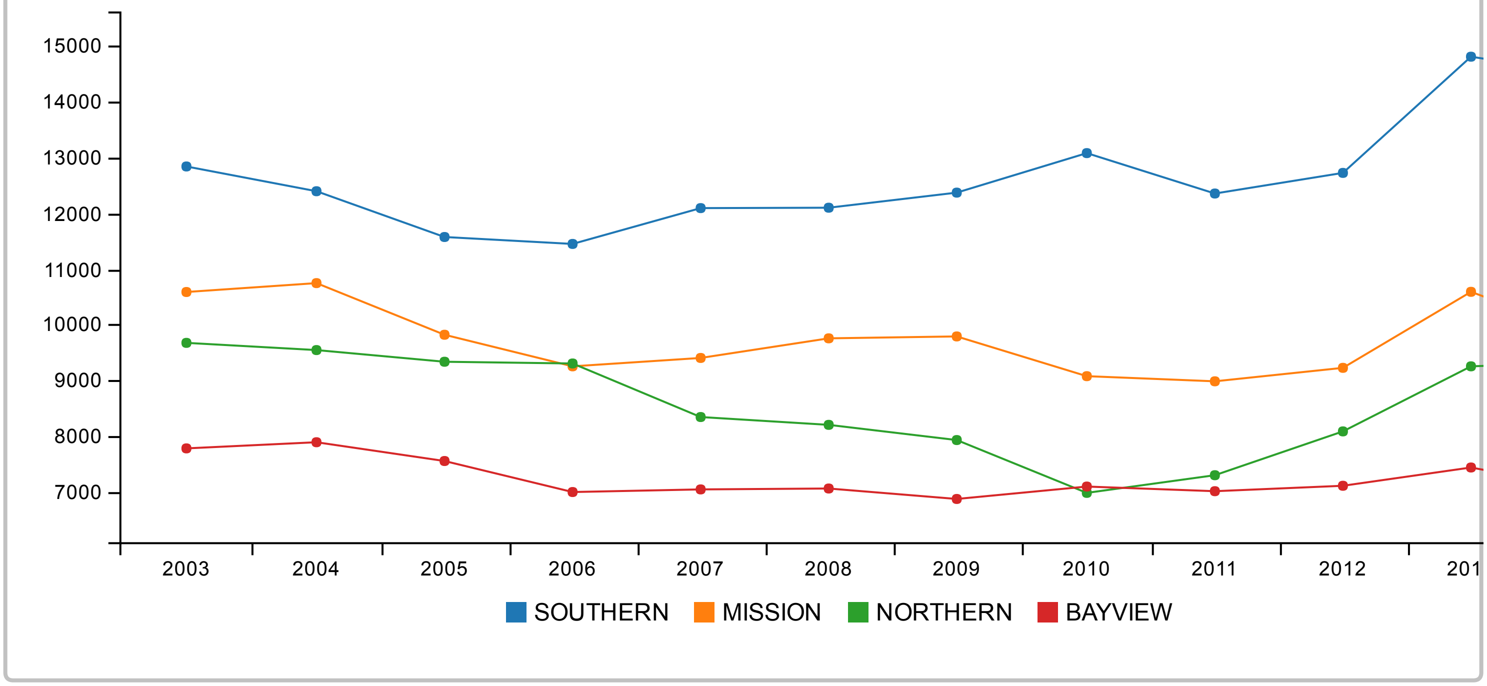


```
import java.time._
import java.time.format._

case class StaticRed(year: Int, SOUTHERN: Int, MISSION: Int, NORTHERN: Int, BAYVIEW: Int)

val top4AddNames = topCrAdd.map(_._1).collect().take(4).toList
val adds = splits.map(x => (x._2.trim, x._1.getYear))
//adds.map(x => (x, 1)).reduceByKey((x, y) => (x+y)).map(x => (x._1._1, x._1._2, x._2)).sortBy(

def matchAdd(input:String):Boolean = {
  for(add <- top4AddNames) {if(input.equals(add)) return true}
  false
}

val top4Add = adds.filter(x => matchAdd(x._1))
          .map(x => (x, 1)).reduceByKey((x, y) => (x+y)).map(x => (x._1._2, (x._1._1, x._2)))
          .groupByKey().mapValues(x => x.toMap)
          .map{case (year, map) => StaticRed(year, map(top4AddNames(0)),
                                            map(top4AddNames(1)),
                                            map(top4AddNames(2)),
                                            map(top4AddNames(3)))}
          .sortBy(_.year).collect
//.map(x => (x._1, x._2.toList(0)._3, x._2.toList(1)._3))
//.map{case (year, list) => StaticRed(year, list.toList(0)._3, list.toList(1)._3)}
//top4Add.take(12)
```

```
CustomC3Chart(top4Add,
             chartOptions = """
             { data: {x: 'year', type: 'bar'},
               axis: {x: {type: 'categorical'}}
             }
             """)
```



```
CustomC3Chart(top4Add,
             chartOptions = """
             { data: {x: 'year', type: 'line'},
               axis: {x: {type: 'categorical'}}
             }
             """)
```



Build: | **buildTime**-*Thu Mar 17 12:12:20 UTC 2016* | **formattedShaVersion**-*0.6.3-61efca56c2a45613ec609f350dc4911b166b3f28-SNAPSHOT* | **sbtVersion**-*0.13.8* | **scalaVersion**-*2.10.5* | **sparkNotebookVersion**-*0.6.3* | **hadoopVersion**-*2.7.2* | **jets3tVersion**-*0.7.1* | **jlineDef**-*(org.scala-lang,2.10.5)* | **sparkVersion**-*1.6.1* | **withHive**-*true* | **withParquet**-*true* |.

Spark Job Progress open SparkUI
(http://169.254.215.57:4040)

Terms defined

Name	Type

Chat Room

Choose an alias

Type your message...