University of California Riverside

CS 167 – Introduction to Big-Date
Spring 2022

Chicago Crimes

By Project Group 3
Alfredo Gonzalez
Nathan Lee
Nishitha Gouravelli

June 8, 2022

Task 1 – Data Preparation
By Nishitha Gouravelli
E-mail: ngour001@ucr.edu
UCR NetID: ngour001
Student ID: 862183759

Task 2 – Spatial Analysis
By Nathan Lee
E-mail: nlee096@ucr.edu
UCR NetID: nlee096
Student ID: 862179510

Task 3 – Temporal Analysis
By Alfredo Gonzalez
E-mail: agonz544@ucr.edu
UCR NetID: agonz544
Student ID: 862254840

Introduction

Our team was tasked with Project A – Chicago Crimes which required us to prepare data and create an output in Parquet format. We would then load the dataset aggregate, compute a few totals, and add geometry data to produce and draw a choropleth map. We would then load the dataset again, aggregate and calculate a few more totals and create a CSV file. The file would then be used to produce a bar graph of crimes given a date range. We mostly leveraged what we learned from Lab6 and Lab9 and used Scala, Beast, and SparkSQL for most of our code. Beast is useful in dealing with spatial data and shapefiles, while Scala and Spark SQL allows us to easily run queries that get the data we need to analyze.

Task 1 – Data Preparation

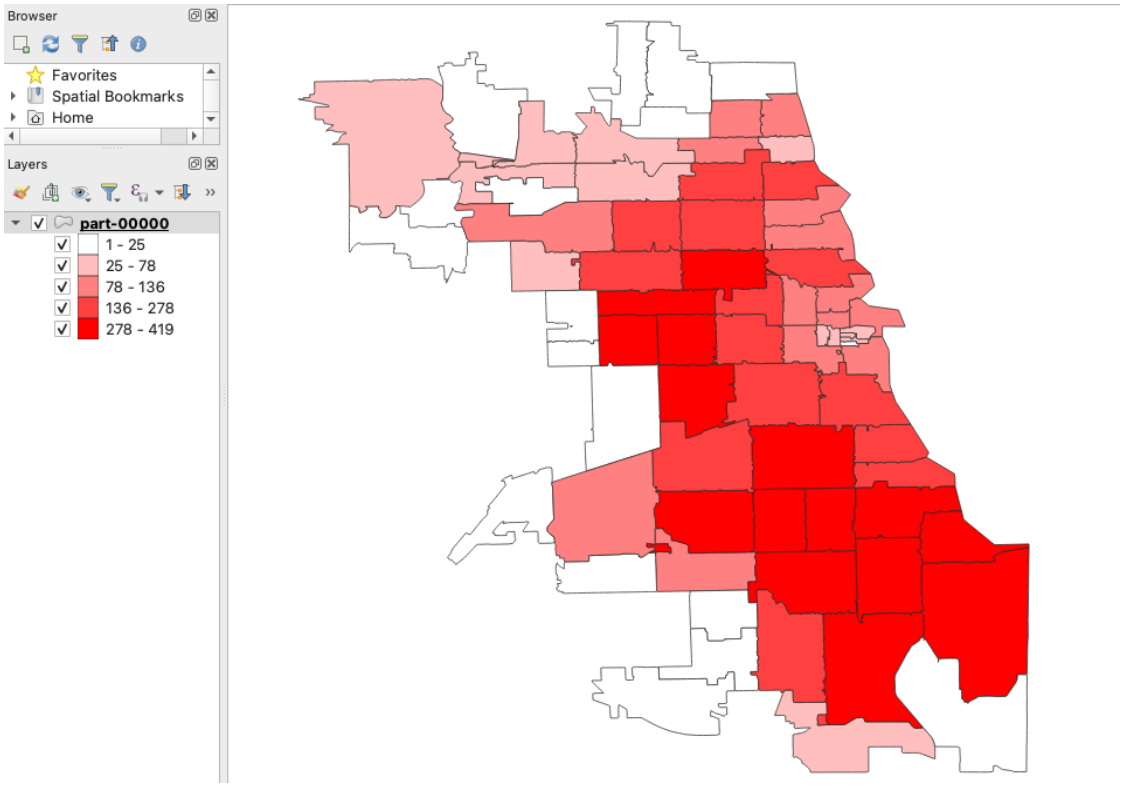
Project A task 1 deals with the data preparation part of the project. The two major steps in the preparation were to add a new attribute ZIPCode and convert the resulting file into a column-oriented Parquet format. We used a spatial join to link the ZIP codes to the crimes. The Parquet format is helpful because it is in column-format, which makes it useful for analytical queries. We also need to read specific columns from very large tables . This format also takes up a lot less disk space than the original CSV files as we can see from the table below, because of its column storage. It supports nesting and compression which makes parsing faster and efficient, especially with datasets that contain multiple thousands of records.

Dataset	**CSV size**	**Parquet size**
1,000	204,575 B	250,781 B
10,000	2,045,620 B	886,554 B
100,000	20,465,049 B	4,089,314 B

Task 2 – Spatial Analysis

Project A task 2 uses the dataset from task 1 to create a shapefile that visualizes the number of crimes in each given ZIP code. The code loads the dataset in Parquet format by inputting the file's name as a command line argument. Using a grouped-aggregate SQL query, to count the number of crimes for each ZIP Code. The ZIP Code Boundaries dataset is then loaded and joined with the previous dataset using an equi-join query on the attributes ZIPCode (from the 1st dataset) and ZCTA5CE10 (from the 2nd dataset). Using coalesce(1) and saveAsShapefile("ZIPCodeCrimeCount"), the result is stored as a Shapefile called ZIPCodeCrimeCount. This file is then imported to QGIS with graduated classification to plot the choropleth map. This task heavily references lab 9 and lab 6. It uses Beast to handle spatial data and shapefiles, as well as Scala to use Spark SQL in order to run SQL queries.

Below is the visualization of the result for the 10k file from Task A :



Task 3 – Temporal Analysis

Task three of Project A - Chicago Crimes consisted of a code that would assist in the temporal analysis of crime data to determine what type of criminal activity was prominent given a date range. The code would require you to pass two date parameters, start and end, to determine the date range of the crime data and produce a CSV. The CSV file would consist of PrimaryType (crime type) and Count (total count) fields. The PrimaryType field is grouped, and the Count field is the sum of those crimes within the given date range. I use two SQL functions, to_timestamp, to format the date and time field from the file, and to_date, to format the two date parameters; time is not being considered, so it was ignored. As to the code used, I settled on SparkRDD and SparkSQL and got most of my inspiration from Lab9 and Lab6. The processed data came from a parquet file, and I included all the necessary functions in one SQL statement that I then output to a single CSV file. I then imported the CSV data into an Excel file and produced the bar graph.

Below is the bar graph visualization from Excel of the results from the 10k file for Task 3: