Document Title:	Open Source Crime Prediction for the National Institute of Justice
Author(s):	Team: Ledray; Gregory Ledray
Document Number:	251180
Date Received:	October 2017
Award Number:	2016-NIJ-Challenge-0003

Open Source Crime Prediction for the National Institute of Justice

Gregory Ledray
g.ledray@gmail.com

Having access to open source software has been a major boon to the development of several software industries. Students using open source software that is adopted in industry can enter new jobs with weeks of experience, and the companies that support open source software can see a 200-percent return on their investment,[1] providing ample incentive to work out the details and support open source projects. As part of the National Institute of Justice's (NIJ) Real-Time Crime Forecasting Challenge, I used open source software to predict crimes in Portland, Oregon, over a three-month period. I will cover the geospatial software I used (OpenJump), the open source library I used for mapping the point data to the shapefile squares (shapelib-1.3.0), and the source code for the mapping. After successfully completing the mapping, I used Python to run a number of filters on the data and eliminated clear outliers by hand for the Burglary category and the sparser categories. Had more time been available, I had planned to improve this sorted data's prediction value by running the data through additional machine-learning algorithms made available through Conda.

OpenJump is an open source alternative to Esri's ArcGIS. I used this tool to take an existing shapefile containing geographic information on the city of Portland and generate a new shapefile with squares of a desired size. Unfortunately I do not have complete documentation for how I performed these steps, but I do have the following notes.

- Both of the operations I needed to use took several minutes, with a faster response time on the 600 x 600 foot squares. If you're not sure if the selected operation is the correct one, try it on very large polygons first, and then switch to the desired square/rectangle/other shape's size.
- In OpenJump there is no fishnet operator, so I am uncertain how I performed the initial polygon generation. I do know that the documentation at http://ojwiki.soldin.de/index.php?title=Tools was essential. In my notes, I mention that I may have used **Overlay t**o combine a map of Portland with a grid generated using **Create grid**. Alternatively, I may have also used **Polygonize** under **Convert** under **Edit Geometries**.
- When generating the shapefiles, I needed the fields:
    - FID (Object ID)

[1] https://opendri.org/leveraging-open-source-as-a-public-institution-new-analysis-reveals-significant-returns-on-investment-in-open-source-technologies/

    - Shape (Geometry)
    - Hotspot (Short Integer)
    - Area (Double)

- With the attributes:
    - Hotspot
    - Area

After creating the shapefiles with OpenJump, I needed a way to parse those shapefiles without using Esri's software. To do this, I found a C library called Shapelib that was capable of parsing shapefiles. I then wrote a C program to map the point data provided from the city of Portland to the geographic squares, and output that data to a .csv file where the line in the file was related to the square number. Documentation for Shapelib can be found here:
http://shapelib.maptools.org/

The most helpful information about the SHP API File can be found here:
http://shapelib.maptools.org/shp_api.html

My source code can be found here:
https://github.com/Gregory-Ledray/Portland_Data_Mining

This discussion includes almost all of my implementation documentation from my work on NIJ's Real-Time Crime Forecasting challenge. If something seems unclear, please contact me at g.ledray@gmail.com