Building of a Crime Mapper

A technical perspective

...

An interactive, visual website dedicated to interpreting Chicago crime data 2001-Present: http://24.38.216.122:4010

By Galil Gertner, Adnan Canovic,
Peizhen Chen & Ibrahim Akay

Requirements

1. Pinpoint accuracy crime data source
2. Interactive map with choropleth layering.
3. Backend to map crimes to zones
4. Front end to query user
5. Interface to pass query to backend and return processed data to from end.
6. Platform to run on a server

Tools Used

1. Chicago crime portal site
2. Google maps API
3. Python 3.5, mySQL 14.14, awk
4. HTML/JavaScript/Bootstrap
5. Flask, Ajax querying
6. Flask, uWSGI, gunicorn

Technical Challenges of Note

1. Mapping crimes in a choropleth over Chicago

2. Passing information between Python and JavaScript

Topics

Mapping

Galil: Creating a choropleth over Chicago

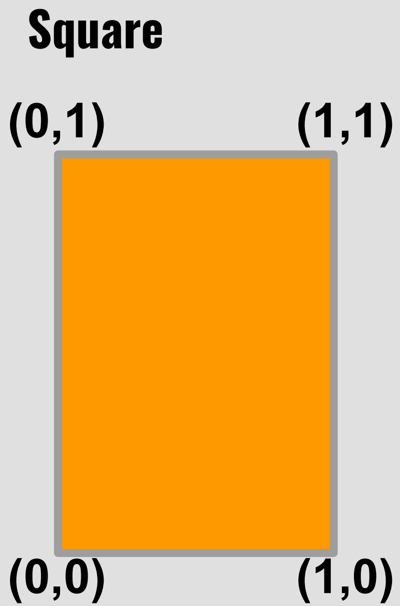
Adnan: Populating the choropleth with data

Query handling

Peizhen: Flask routing and function calling

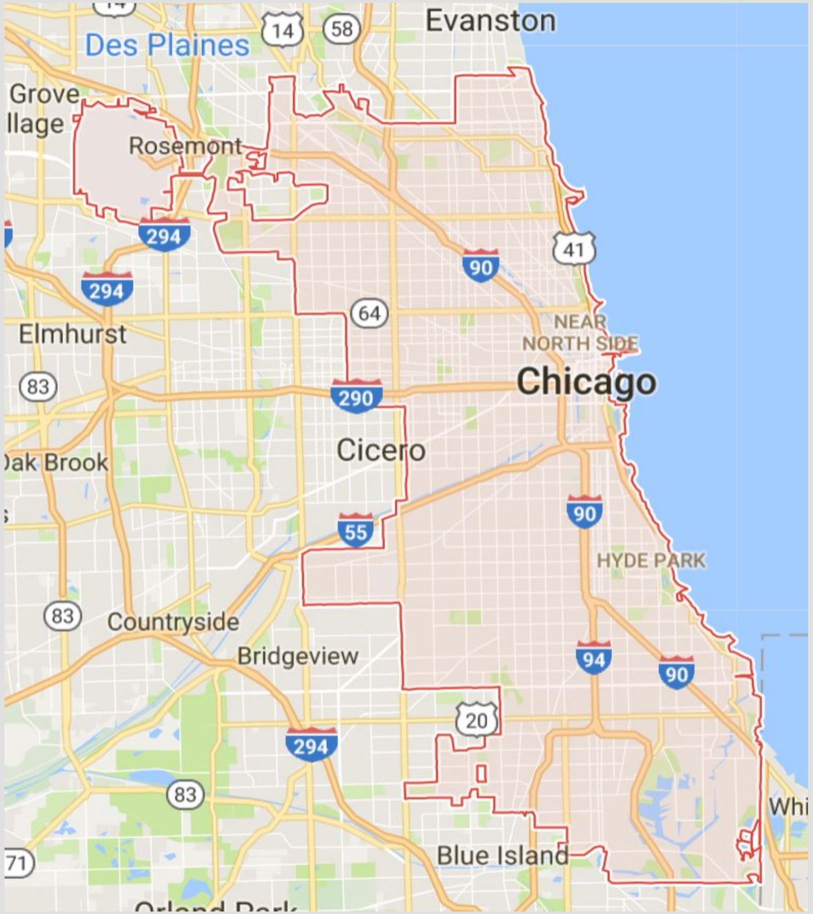
Ibrahim: Ajax query and response

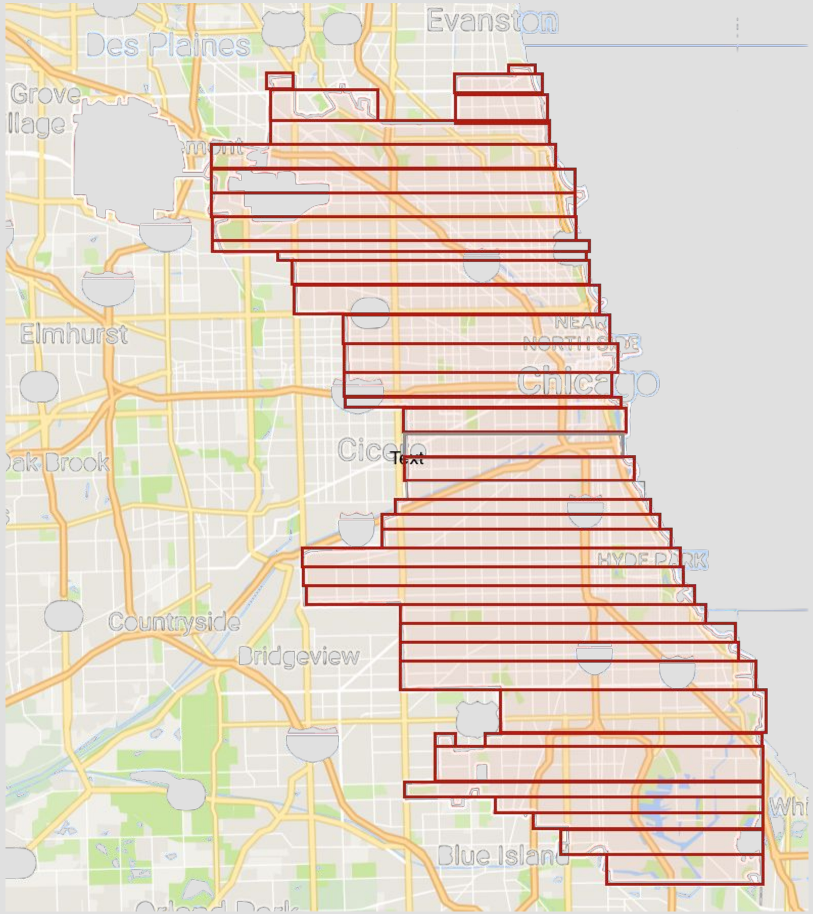
The geoJSON format



```
{
  "type":
"FeatureCollection",
  "features": [
        {
        "type": "Feature",
        "Id": “SQUARE",
        "properties": {
        },
        "geometry": {
        "type": "Polygon",
        "coordinates": [
        [
        [0, 0],
        [0, 1],
        [1, 1],
        [1, 0],
        [0, 0]
        ]
    ]
}
```

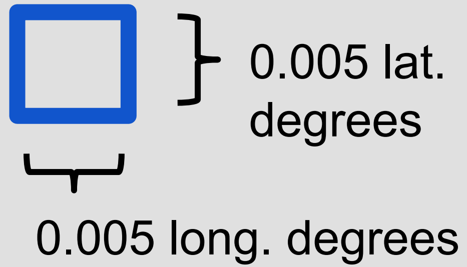
Creating projections onto Chicago.



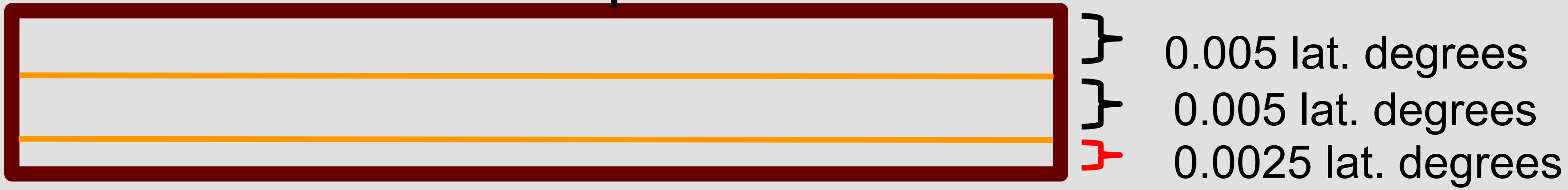


Our standard zone size:
0.005 degrees by 0.005 degrees

Area = 0.000025 degrees squared

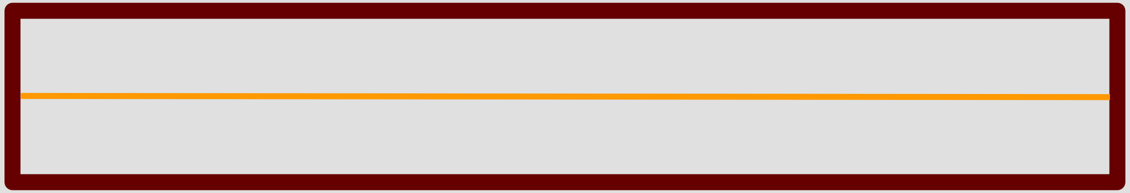


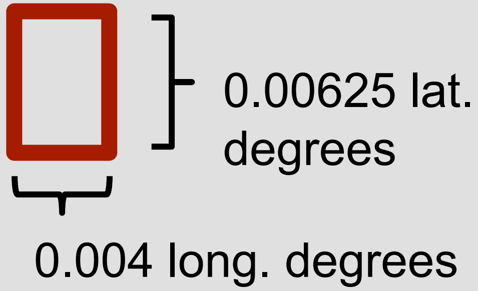
Band with two and half strips



New zone dimension with equal area:
0.00625 degrees by 0.004 degrees

Area = 0.000025 degrees squared





AWK and sed

Fixing the data for SQL

- Manipulate file contents on Unix command line
- Raw CSV is not clean
    - Dates are in an ambiguous MM-DD-YYYY format -- mySQL is not happy
    - A small percentage of locations are null
    - The CSV header would be input as a row
- AWK and sed can edit a file to fix these issues

Populating the mySQL Database

- Huge dataset
  - Over 6.3M rows of data with 22 columns
- Three table initial plan

- Data trimmed with awk down to 6 columns
  - Single table required
- SQL’s load data infile command can import a csv into the single table at high speed

Zoning Chicago Crime

Using the geoJSON
to group crimes into zones

- A python script generates SQL commands using geoJSON data
    - Rows from CSV are updated with new zone value directly from geoJSON
    - Corners of geoJSON coordinates define a simple WHERE clause for the update

- Our generated rectangular zones allow for a much faster and simpler query than using predefined Chicago areas

JavaScript/HTML front end

Getting the zone scores to Javascript

- We use Google Maps API to build the choropleth, we need to provide a score for each geoJSON
- How do we get the zone scores from python and mysql to the JavaScript google map api? Flask.

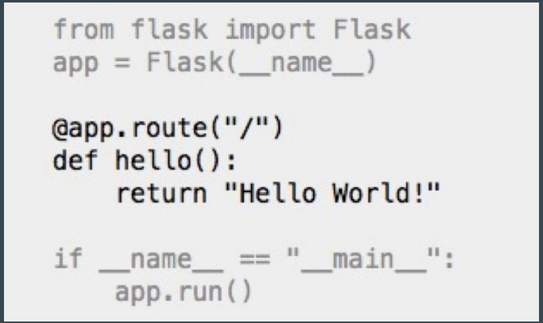
What is Flask?

- Flask is micro web framework
  - Written in Python
  - Does not require particular tools or libraries
  - However,
    - Werkzeug’s toolkit
    - Jinja2 template engine
- It is BD license

http://flask.pocoo.org

Format:

1. Import the flask class
2. Create instance of this class
3. Set URL for the function
4. Define a function
5. Repeat Step 3 & 4 for more functions
6. Run the app



```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

Flask Routing

Route(rule, option)

- Basic rule
  - app.route('/url')
- Variable rule
  - app.route('/user/<username>')

- Based on Werkzeug routing method
  - Beautiful and Unique URL
  - Top to bottom
- Used to bind a function to a URL

Function Calling

- Calling function as source using url_for()
  - {{ url_for('function_name") }}

- Calling function with parameter
  - in Flask,

    @app.route('/url', methods=['POST', 'GET'])

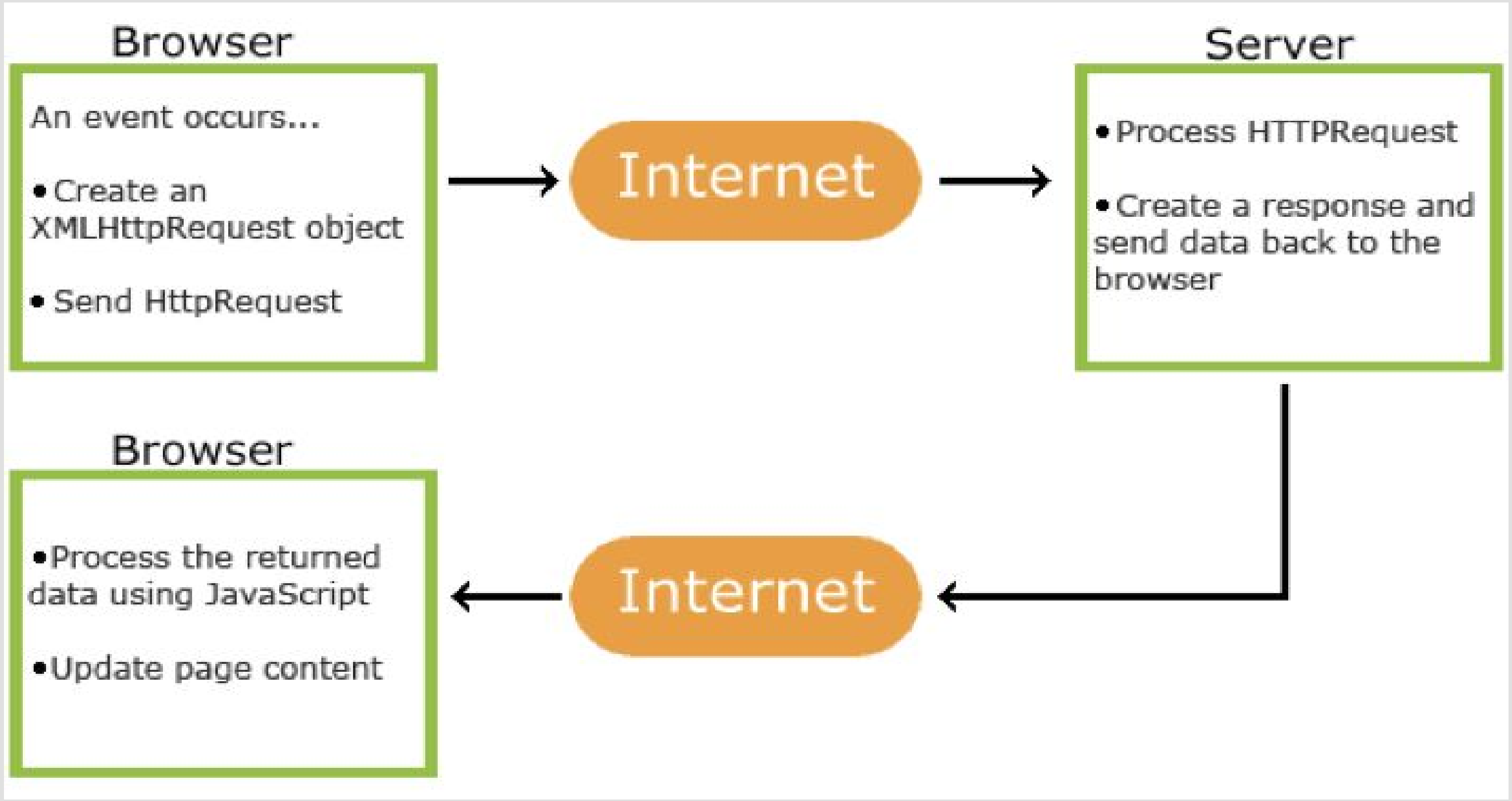
  - in HTML

    AJAX

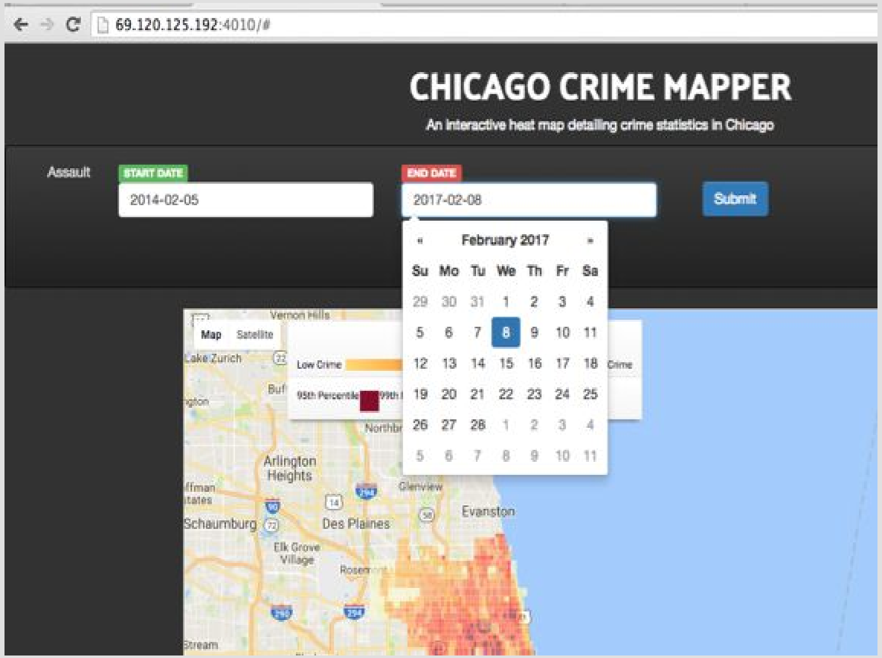
Asynchronous JavaScript And XML

- Uses:
  - Update a web page without reloading the page
  - Request, receive, and send data from a server
- Components:
  - Browser built-in XMLHttpRequest object
  - JavaScript and HTML DOM
- Data can be transported as XML, plain text, or JSON text.

AJAX


Browser
An event occurs...
• Create an XMLHttpRequest object
• Send HttpRequest
Internet
Server
• Process HTTPRequest
• Create a response and send data back to the browser
Internet
Browser
• Process the returned data using JavaScript
• Update page content

Crime Mapper & AJAX



- User selects crime type and date range on webpage
- JavaScript:
  - Data is collected by JavaScript function
  - AJAX POST request method is used
- Python:
  - Data is passed to Flask
  - On success, value is returned
- JavaScript:
  - Data returned is displayed on the map

Further Links

GitHub:
https://github.com/HunterCrimeMapper/ChicagoCrimeMapper

Webpage: http://24.38.216.122:4010

Questions?