DC Crime Analysis Dashboard

GW Data Analytics Bootcamp - Project 3

Reena Desai, Jimmy White, Mario Cosby, and Tim St. Onge

Issues

Crime data can be large, dense, and difficult to parse.

To gain insights into potential patterns of crime, we need tools that provide efficient methods for filtering and visualizing data.

Primary Objective

Create a dynamic, easy-to-use web application for quickly analyzing trends in DC crime data for over space and time

Workflow

- We used five years of DC crime data (2014-2018) to better understand crime trends
  - Data downloaded as CSVs from Open Data DC
- Data CSVs unified and loaded into a SQL database (via MySQLWorkbench)
- Supplemental SQL data tables (Ward and Crime Type listings) were also created to support crime filters in final application.
- We pushed SQL database to Amazon AWS for SQL Server so that final application would pull from cloud.

Workflow

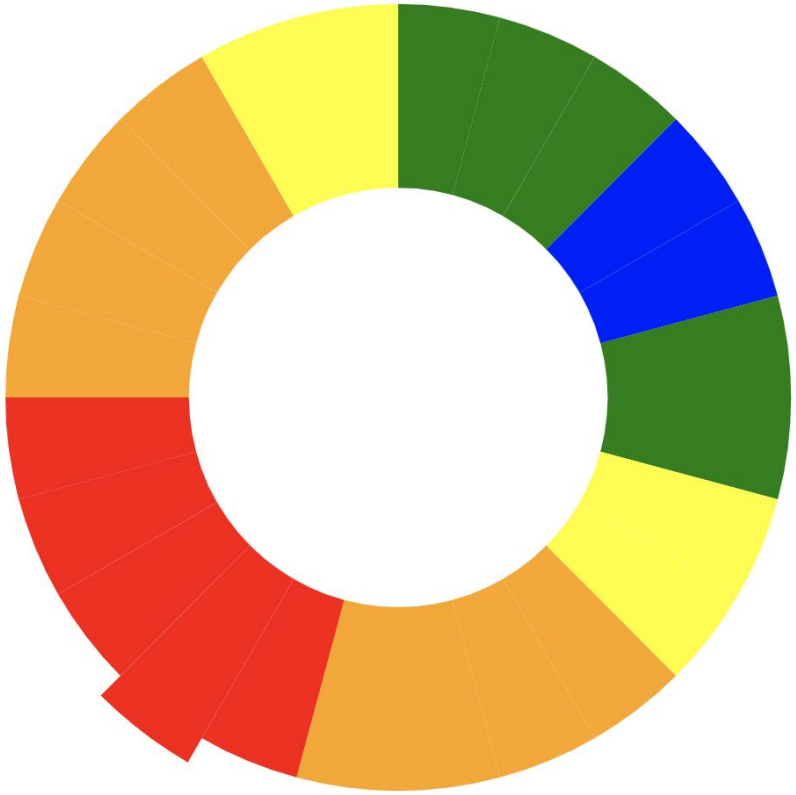
- We created our application file (app.py) with several endpoints that would receive data from AWS.
- Data streams from these endpoints would feed into three visualizations created in JS:
  - Map of crime incidents filtered by crime type and DC ward
  - Time wheel charting crime times over 24-hour cycle
  - Line graph charting crimes by month among each ward

```python

@app.route("/data")
def crime_data():
    """Return a list of sample names."""
    ward = request.args.get("WARD")
    offense = request.args.get("OFFENSE")
    # Use Pandas to perform the sql query
    if ward=="All" and offense=="All":
        query_all=f"SELECT CCN,CENSUS_TRACT,END_DATE,LATITUDE,LONGITUDE,METHOD,OFFENSE,PSA,REPORT_DAT,SH
    elif ward=="All":
        query_all=f"SELECT CCN,CENSUS_TRACT,END_DATE,LATITUDE,LONGITUDE,METHOD,OFFENSE,PSA,REPORT_DAT,SH
    elif offense=="All":
        query_all=f"SELECT CCN,CENSUS_TRACT,END_DATE,LATITUDE,LONGITUDE,METHOD,OFFENSE,PSA,REPORT_DAT,SH
    else:
        query_all = f"SELECT CCN,CENSUS_TRACT,END_DATE,LATITUDE,LONGITUDE,METHOD,OFFENSE,PSA,REPORT_DAT,

    remote_crime_data = pd.read_sql(query_all, conn)
    #print(remote_crime_data.to_dict(orient="records"))
    return(jsonify(remote_crime_data.to_dict(orient="records")))



@app.route("/ward_offense")
def offense_data():
    remote_offense_data=pd.read_sql("SELECT DISTINCT OFFENSE FROM crime_incidents_all",conn)
    offense_dict=remote_offense_data.to_dict(orient="records")
    remote_ward_data = pd.read_sql("SELECT * FROM dc_wards", conn)
    ward_dict=remote_ward_data.to_dict(orient="records")
    result_dict={"ward": ward_dict,"offense":offense_dict}
    return(jsonify(result_dict))
```

```python
@app.route("/charts_data")
def num_crimes():
    query_all=f"SELECT OFFENSE,END_DATE,WARD FROM crime_incidents_all LIMIT 6000"
    charts_crime_data = pd.read_sql(query_all, conn)
    print(charts_crime_data.to_dict(orient="records"))
    return(jsonify(charts_crime_data.to_dict(orient="records")))


@app.route("/ward_data")
def ward_data():
    """Return a list of sample names."""

    # Use Pandas to perform the sql query
    remote_ward_data = pd.read_sql("SELECT * FROM dc_wards", conn)
    #print(remote_crime_data.to_dict(orient="records"))
    return(jsonify(remote_ward_data.to_dict(orient="records")))

@app.route("/time_wheel")
def index2():

    return render_template("index2.html")

@app.route("/line_chart")
def index3():
    return render_template("index3.html")

#@app.route("/metadata/<sample>")
#def sample_metadata(sample):
#    """Return the MetaData for a given sample."""
#    sel = [
```

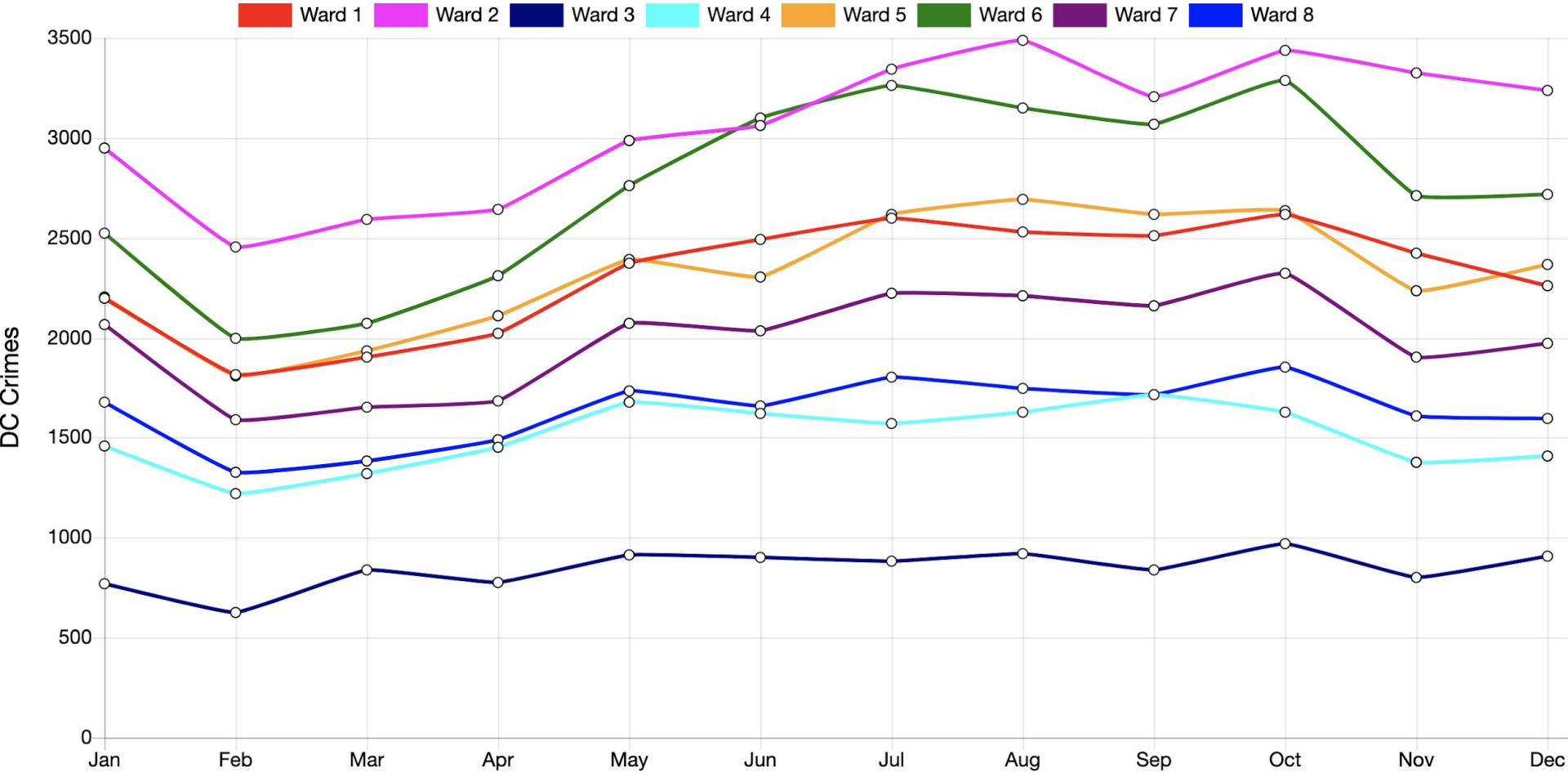
LET'S SEE THE THING!

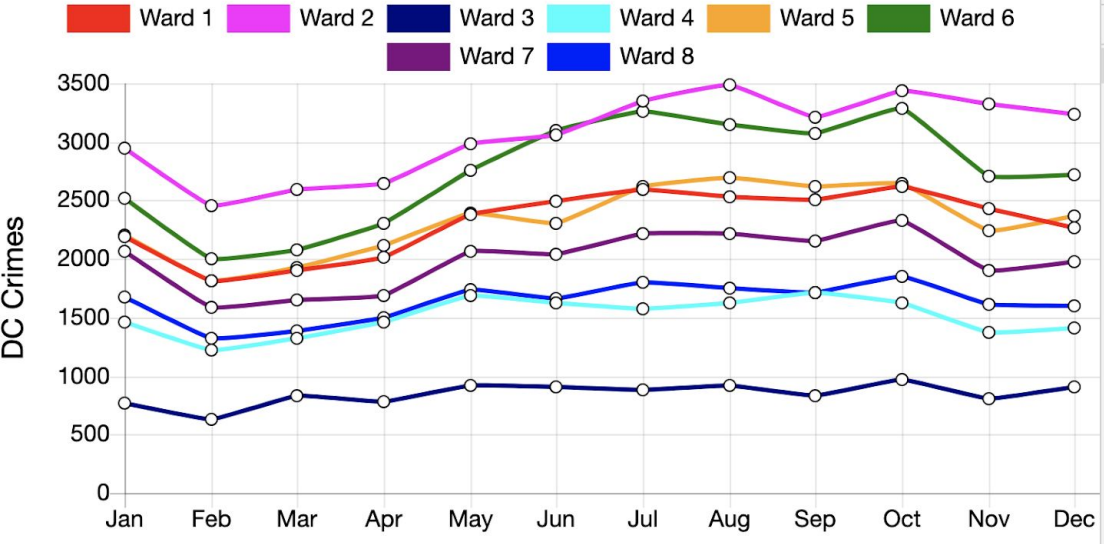
https://justice-league-dc.herokuapp.com/



15:00

High Threat


Ward 1
Ward 2
Ward 3
Ward 4
Ward 5
Ward 6
Ward 7
Ward 8
DC Crimes
3500
3000
2500
2000
1500
1000
500
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec



Elements Console Sources Network Performance

top Filter Default levels

15844 mes...
15842 user...
No errors
No warnings
15842 info
2 verbose

Hide network
Preserve log
Selected context only
Group similar
Log XMLHttpRequests
Eager evaluation
Autocomplete from history

15842 YES crimes.js:22

[Violation] Added non-passive event listener to a scroll-blocking 'touchstart' event. Consider marking event handler as 'passive' to make the page more responsive. See https://www.chromestatus.com/feature/5745543795965952 Chart.min.js:12

[Violation] Added non-passive event listener to a scroll-blocking 'touchmove' event. Consider marking event handler as 'passive' to make the page more responsive. See https://www.chromestatus.com/feature/5745543795965952 Chart.min.js:12

Console What's New ×