

Checkpoint 3

Team: the dapper squirrels

Proposed Visualizations

In this section, we verified our hypothesis by using SQL and D3 visualizations.
We proposed two visualization themes:

1. Highlighting the high and low socio-economy status communities with different colors and plot TRRs on them. Set up a time slider to see how it changes over time.
2. Using color code(heat map) of A&A (dara_officer assignment attendance) in different neighborhoods. Set up a time slider to see how it changes over time.

How to View?

The full report of the findings can be found here:

<https://observablehq.com/@liux2/findings>.

There are also two PDFs in the directory for the un-interactive reports.

SQL Used to Query the Data for Plots

```
SELECT sum(a.counts) over (partition by beat order by year) as total_cr, a.*, sum(b.trr_count) over (partitio
(SELECT EXTRACT(YEAR FROM incident_date) as year, da.id as beat, median_income::money::numeric as income, cou
FROM data_complainant
LEFT OUTER JOIN data_allegation ON data_complainant.allegation_id = data_allegation.crid
LEFT JOIN data_allegation_areas daa on data_complainant.allegation_id = daa.allegation_id
LEFT OUTER JOIN data_area da on da.id = daa.area_id
where median_income is not null and EXTRACT(YEAR FROM incident_date) >= 2005
group by 1,2,3) a
LEFT JOIN (SELECT SUM(trr_count) as trr_count,community_id,date FROM
            (SELECT count(*) as trr_count,data_area.id AS beat_id, EXTRACT(YEAR FROM trr_datetime) AS date
              LEFT JOIN data_area ON beat::text = data_area.name
             GROUP BY 2,3) a,
            (SELECT DISTINCT ON(1) table1.id as beat_id, table2.id as community_id FROM
              (SELECT * FROM data_area WHERE data_area.are
              (SELECT * FROM data_area WHERE data_area.are
              WHERE ST_Contains(table2.polygon, table1.polygon) or st_intersects(table2.polygon, table1.poly
WHERE a.beat_id = m.beat_id
GROUP BY 2,3
) b ON a.beat = b.community_id and a.year = b.date;
```

```
SELECT json_build_object(
  'type','FeatureCollection',
  'features', json_agg(json_build_object(
    'type','MultiPolygon',
    'coordinates',p.coordinates,
    'properties',json_build_object(
      'name',p.id
    )::jsonb
  )
)
FROM (SELECT ST_AsGeoJSON(data_area.polygon) :: json->'coordinates' AS coordinates ,id
      FROM data_area WHERE data_area.area_type = 'beat' and polygon IS NOT NULL) p;
```

```

SELECT beat_id, CAST(count(CASE WHEN present_for_duty THEN 1 END)AS float)/count(*) AS percent, EXTRACT(YEAR
FROM data_officerassignmentattendance
WHERE start_timestamp IS NOT NULL and beat_id IS NOT NULL
GROUP BY data_officerassignmentattendance.beat_id,date;

```

```

SELECT sum(count) over (partition by name order by date) as value, name, date FROM
(SELECT count(*) as count,name,date
FROM (SELECT data_area.id as name, to_char(trr_datetime, 'YYYY-MM') AS date from trr_trr LEFT
ON beat::text = data_area.name ) AS a
GROUP BY 2, 3 ORDER BY date ASC ) AS b ORDER BY value DESC ;

```