Xingbang Liu • xingbangliu.io
Allegheny College alumni, computer science major. Northwestern University student, MSAI.

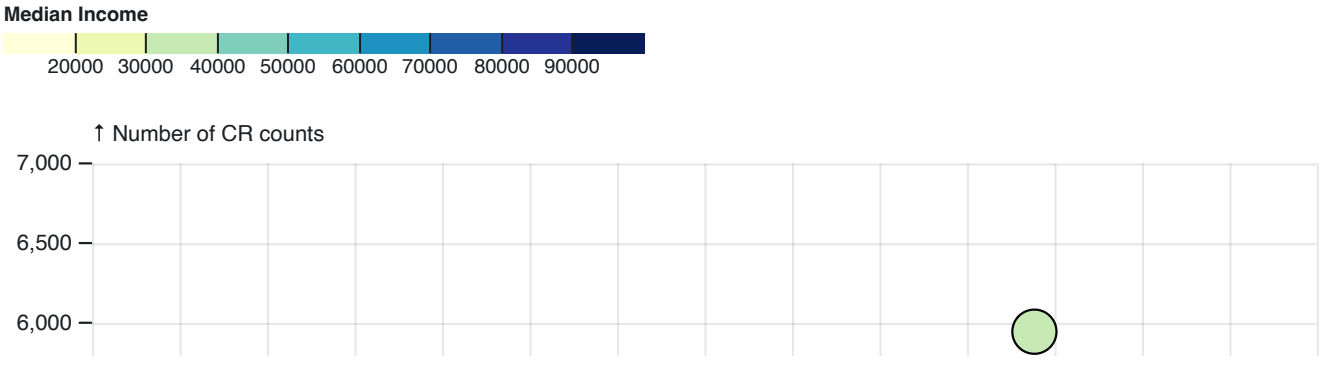
Shared Published (unlisted) Nov 9

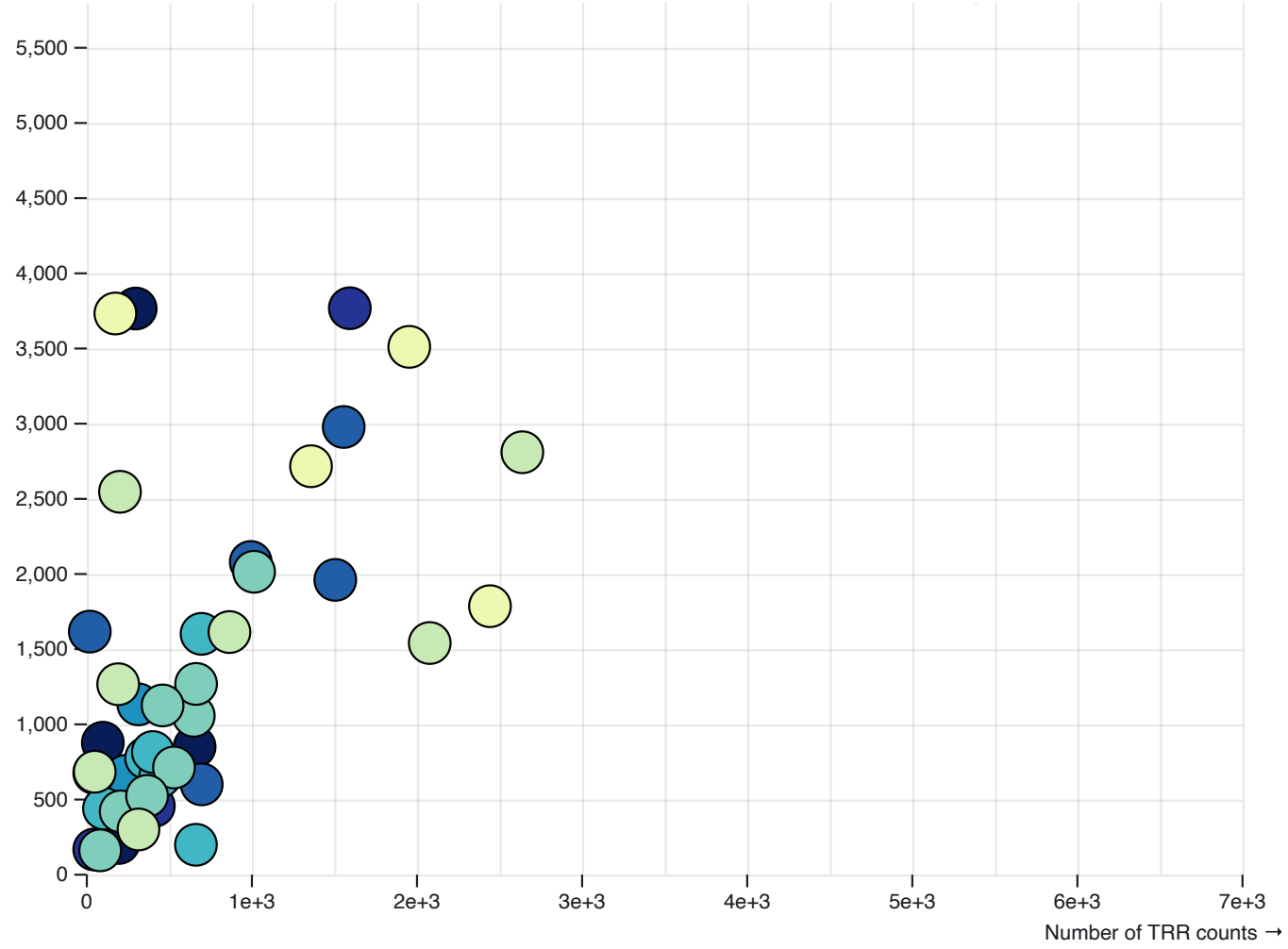
Report on Interactive Visualization Findings

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.

Our theme is about the relationship between geo locations, eco-socio status, and over-policing. In our previous analysis, we have seen the positive correlation between low economic communities and high complaint reports (which indicates more misconduct behaviors). In this report, we would like to further look at key metrics(number of CRs and TRRs) in different communities with different economic social statuses. More importantly, we want to monitor the changes over time, and locate the trend in recent years.

```
Our theme is about the relationship between geo locations, eco-socio status,
and over-policing.
In our previous analysis, we have seen the positive correlation between low
economic communities and high complaint reports (which indicates more misconduct
behaviors).
In this report, we would like to further look at key metrics(number of CRs and
TRRs) in different communities with different economic social statuses. More
importantly, we want to monitor the changes over time, and locate the trend in
recent years.
```





Plot 1

So, we created this plot with multiple dimensions in years, beat areas, median incomes, the number of tactical response reports, and the number of complaint reports.

As you can see, low-income areas(yellow bubbles in the plot) are generally having more CR and TRR reports. High-income areas, despite 2 outliers typically have fewer. From the time perspective, we have seen a reduced increasing speed for both records over the years for most of the beat area. Especially for the "rich" area, most of them stayed around the origin point.

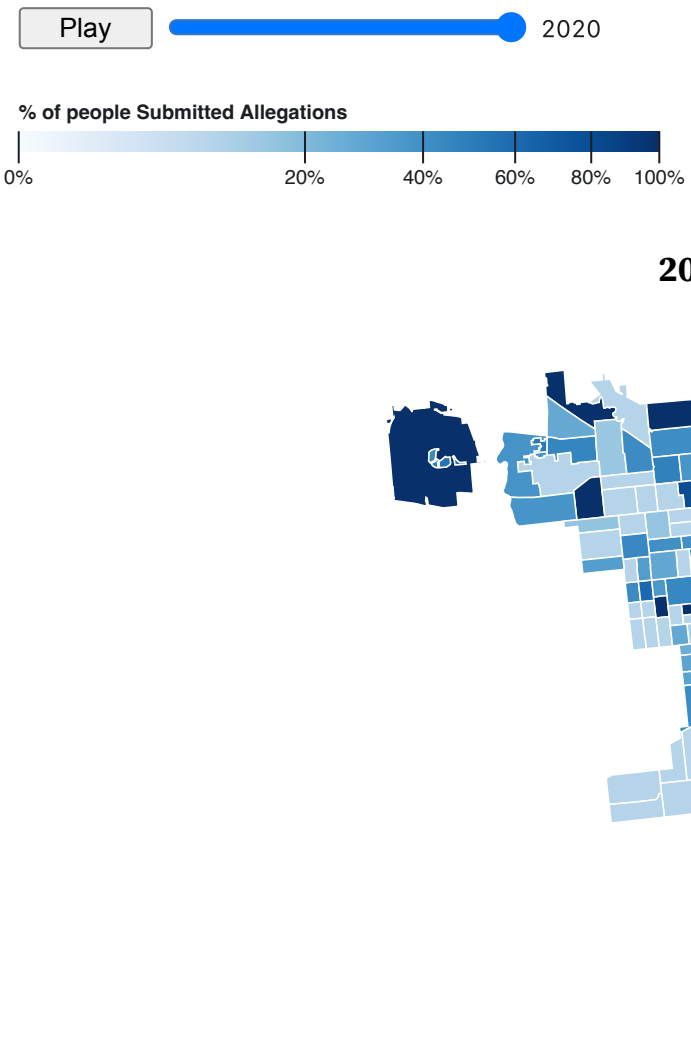
From this chart, we can find there are multiple shallow colored points(low income community) on the top right corner, it means for these areas, police are receiving significant high amount of complains and they would prefer to use technique weapons to agains the citizens. Therefore, we can conclude, there is over-policing in low socio-eco status neighborhoods.

```
Also from this chart, we can find there are multiple shallow colored
```

```
points(low-income community) on the top right corner, which means for these
areas, police are receiving a significantly high amount of complaints and they
would prefer to use technique weapons to against the citizens. Therefore, we
can assume there is over-policing in low socio-eco status neighborhoods.
```

attendance) in different neighborhoods. Set up a time slider to see how it changes over time.

In this section, we discuss the potential relations between officer attendant rate in each beat area each year and the tactical response. We assume that the attendance rate could affect the tactical response rate of the area. To observe the attendance rate each year, we created the animation of officer attendance rate each year, as the following plot shows.

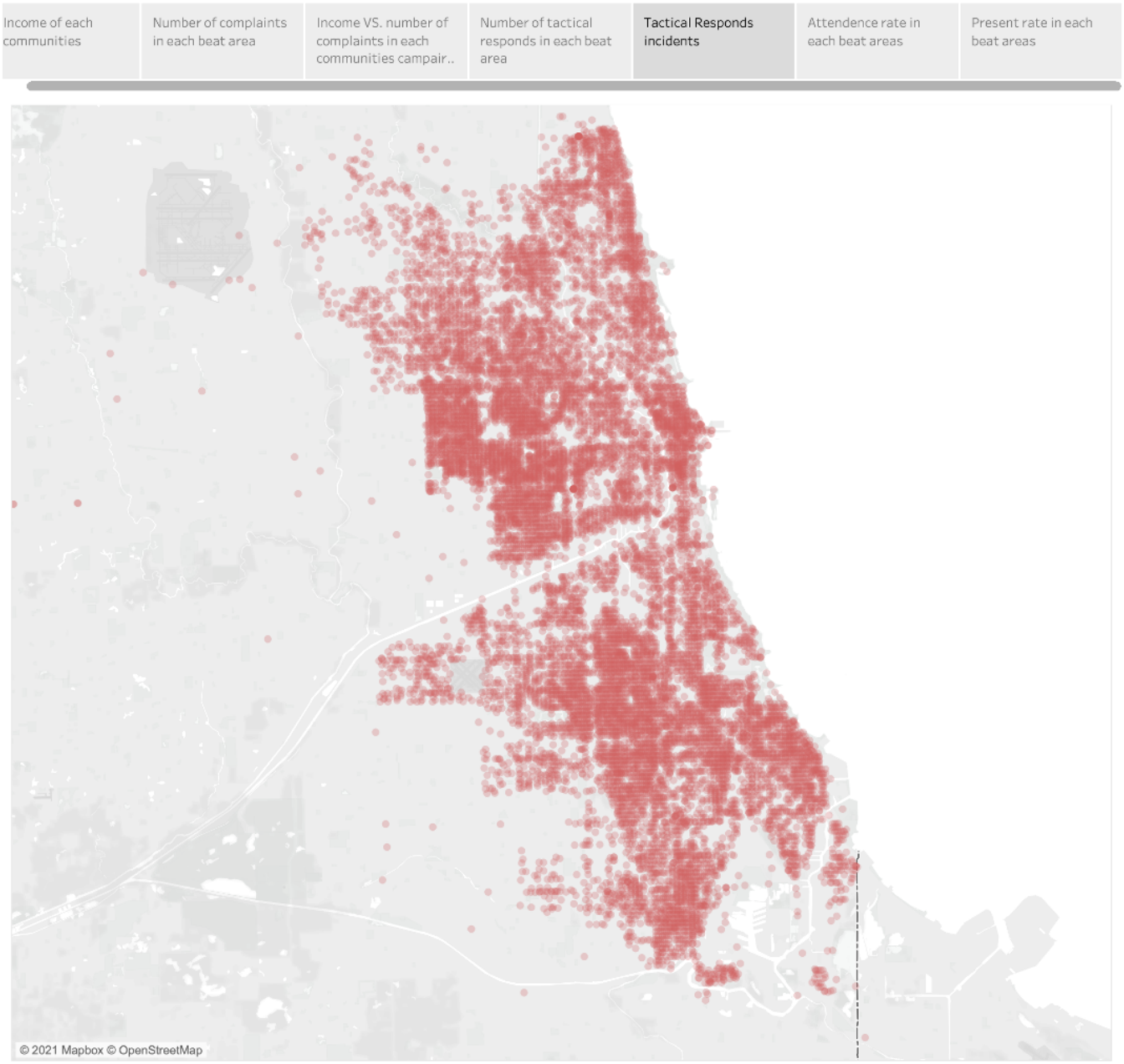


Plot 2

In the graph, we observed that areas such as 123, 271, 101, and 105 have consistently low attendant rates. By comparing with the records in checkpoint 2, we can know that these areas also have low tactical responses and high incomes

that these areas also have low tactical responses and high incomes.

Visualizations



Plot 3

This observation also complemented the findings in the previous question, high-income communities tend to receive less tactical responses from police officers, and they tend to file more complaints to the officers.

Here is another chart for trrs.

Click Here to the full article:

> "With this bar chart race of trr over different beats, we can clearly find such two leading beats, 132 and 65. Associated with Plot 2 in Observable Notebook and the median income map in our Checkpoints 2, we can find that beat 65 and

and the median income map in our Checkpoints 2, we can find that beat 65 and 132, especial beat 132 belongs to the low socio-eco status neighborhoods and it contains a high police attendance rate. The high police attendance rate is important because it implies that police officers pay attention and focus on the security in such areas. With the leading trrs in such low socio-eco status neighborhoods like the beat 132, we can assume that there is over-policing in some low-income areas."

Conclusion

With the above four plots, we can roughly have an idea of the patterns of police misconduct. Police officers tend to give less tactical responses to high-income areas. To avoid biases, police misconduct can be viewed from both public narratives and police narratives, which means an area with both high complaint rates and high tactical response can be viewed as the police misconduct area. From our observation, police misconduct areas tend to have less income.

Appendix

The setups for the plots is as follows.

Plot 1

```
update = undefined
```

```
currentData = ▸ Array(41) [Object, Object, Object, Object, Object, Object, Object
```

```
x = f(n)
```

```
### Conclusion

To avoid biases, police misconduct can be viewed from both public narratives and
police narratives, which means an area with both high complaint rates and high
tactical response can be viewed as the police misconduct area. With the above
four plots, we can roughly have an idea of the patterns of police misconduct.
```

Police officers tend to give less tactical responses to high-income areas and police misconduct areas tend to have less income.

```
color = f(c)
```

```
xAxis = f(g)
```

```
update = chart.update(currentData)
```

```
currentData = dataAt(date)
```

```
x = d3.scaleLinear([0, 7000], [margin.left, width - margin.right])
yAxis = f(g)
```

```
y = d3.scaleLinear([0, 7000], [height - margin.bottom, margin.top])
```

```
radius = d3.scaleSqrt([0, 2e8], [0, width])
```

```
// color = d3.scaleOrdinal(data2.map(d => d.region),
d3.schemeCategory10).unknown("black")
color = d3.scaleThreshold([20000, 30000, 40000, 50000, 60000, 70000, 80000,
90000], d3.schemeYlGnBu[9])
```

```
xAxis = g => g
    .attr("transform", `translate(0,${height - margin.bottom})`)
    .call(d3.axisBottom(x).ticks(width / 80, ","))
    .call(g => g.select(".domain").remove())
    .call(g => g.append("text")
        .attr("x", width)
        .attr("y", margin.bottom - 4)
        .attr("fill", "currentColor")
        .attr("text-anchor", "end")
        .text("Number of TRR counts →"))
```

```
yAxis = g => g
    .attr("transform", `translate(${margin.left},0)`)
    .call(d3.axisLeft(y))
    .call(g => g.select(".domain").remove())
    .call(g => g.append("text")
        .attr("x", 0)
        .attr("y", 10)
        .attr("fill", "currentColor")
        .attr("text-anchor", "start")
        .text("↑ Number of CR counts"))
```

```
grid = g => g
    .attr("stroke", "currentColor")
    .attr("stroke-opacity", 0.1)
    .call(g => g.append("g")
      .selectAll("line")
      .data(x.ticks())
      .join("line")
        .attr("x1", d => 0.5 + x(d))
        .attr("x2", d => 0.5 + x(d))
        .attr("y1", margin.top)
        .attr("y2", height - margin.bottom))
    .call(g => g.append("g")
      .selectAll("line")
      .data(y.ticks())
      .join("line")
        .attr("y1", d => 0.5 + y(d))
        .attr("y2", d => 0.5 + y(d))
        .attr("x1", margin.left)
        .attr("x2", width - margin.right));
```

```
function dataAt(date) {
  return data2.map(d => ({
    name: d.name,
    region: d.region,
    income: valueAt(d.income, date),
    population: valueAt(d.population, date),
    lifeExpectancy: valueAt(d.lifeExpectancy, date)
  }));
}
```

```
interval = ƒ(n)
```

```
function valueAt(values, date) {
  const i = bisectDate(values, date, 0, values.length - 1);
  const a = values[i];
  // if (i > 0) {
  //   const b = values[i - 1];
  //   const t = (date - a[0]) / (b[0] - a[0]);
  //   return a[1] * (1 - t) + b[1] * t;
  // }
  return a[1];
}
```

```
data = (await fetch("https://raw.githubusercontent.com/NU-DSS-The-Dapper-Squirrels/The-Dapper-Squirrels/main/checkpoint-3/data/cr%402.json")).json()
```

```
data2 = data
  .map(({name, region, income, population, lifeExpectancy}) => ({
    name,
    region,
    income: parseSeries(income),
    population: parseSeries(population),
    lifeExpectancy: parseSeries(lifeExpectancy)
  }))
```

```
bisectDate = ƒ(t, n, e, i)
```

```
interval = d3.utcMonth // interval between animation frames
```

```
margin = ▸ Object {top: 20, right: 20, bottom: 35, left: 40}
```

```
dates = interval.range(
  d3.min(data2, d => {
    return d3.min([
      d.income[0],
      d.population[0],
      d.lifeExpectancy[0]
    ], ([date]) => date);
  }),
  d3.min(data2, d => {
    return d3.max([
      d.income[d.income.length - 1],
      d.population[d.population.length - 1],
      d.lifeExpectancy[d.lifeExpectancy.length - 1]
    ], ([date]) => date);
  })
```

```
)
```

```
function parseSeries(series) {
  return series.map(([year, value]) => [new Date(Date.UTC(year, 0, 1)), value]);
}
```

```
bisectDate = d3.bisector(([date]) => date).left
```

```
import {legend} from "@d3/color-legend"
```

```
margin = ({top: 20, right: 20, bottom: 35, left: 40})
```

```
import {Legend, Swatches} from "@d3/color-legend"
```

```
height = 560
```

```
d3 = require("d3@6.7.0/dist/d3.min.js")
```

```
import {Scrubber} from "@mbostock/scrubber"
```

```
trr_data = ▸ Object {2014: Object, 2015: Object, 2016: Object, 2017: Object, 2018
```

```
selectedDateWatcher = {
  map.updateSelectedDate(selectedDate);
  return selectedDate;
}
```

```
d32 = require('d3@5');
```

```
// import {Scrubber} from "@mbostock/scrubber"
```

```
import {legend} from "@d3/color-legend"
```

```
// trr_data = FileAttachment("out1.json").json()
```

```
import {Legend, Swatches} from "@d3/color-legend"
```

```
topojson = require("topojson-client@3")
```

```
districts = (await fetch("https://raw.githubusercontent.com/NU-DSS-The-Dapper-Squirrels/The-Dapper-Squirrels/main/checkpoint-3/data/districts.json")).json()
```

```
trr_data = (await fetch("https://raw.githubusercontent.com/NU-DSS-The-Dapper-Squirrels/The-Dapper-Squirrels/main/checkpoint-3/data/out.json")).json()
```