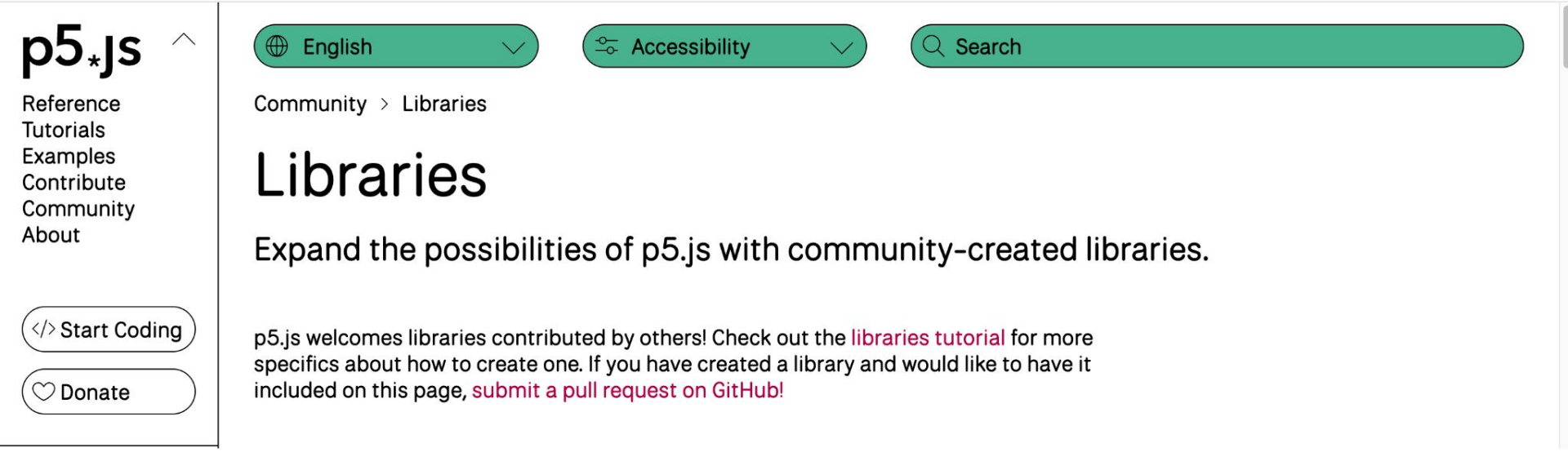
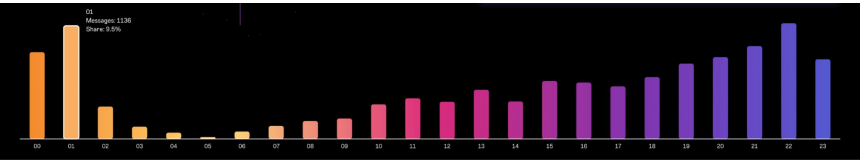
p5.chart.js

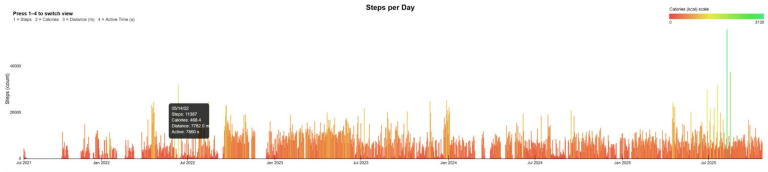
Siddharth Chattoraj



Concept

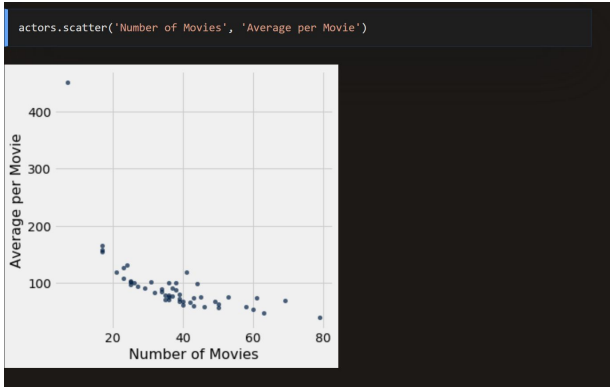
- **p5.chart:** Interactive data visualization and manipulation add-on library for p5.js
  - DataFrames for data analysis: Filtering, grouping, vectorized operations, Boolean indexing, statistical methods, etc.
  - Graphs: Interactive chart constructors for line plots, bar plots, scatter plots, histograms, pie charts, and tables
  - Maps: Data-driven point maps
- DataFrames will enable users to load, clean, transform, summarize, and perform operations on datasets inside p5.js
  - Graphs and maps will be interactive by default, and the base template will enable users to quickly visualize data with few lines of code
  - The library will be accessible to users with only a limited knowledge of JavaScript or coding in general, but advanced users will be able to create detailed interactive charts using the library's in-depth customization features
  - Documentation and formatting will follow the p5.js library creation guidelines

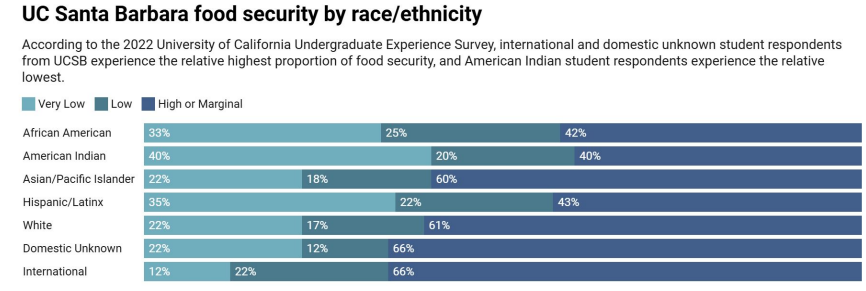


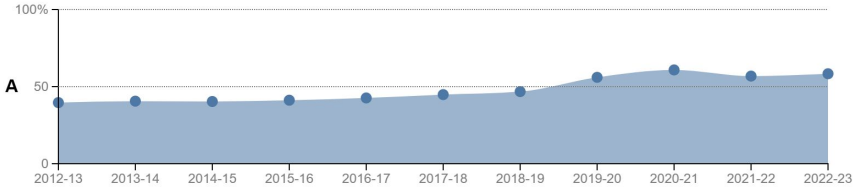


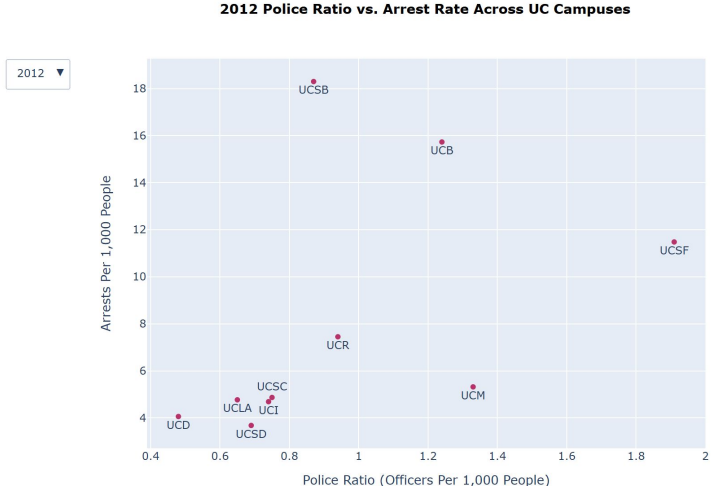
Background

- Data visualization experience across languages
  - Python, R, D3.js, etc.
  - Non-coding tools as well
    - Datawrapper
- ULA for CMPSC 5A
  - Introduction to Data Science 1
  - 9 quarters
  - GE Quantitative Reasoning Course









Motivation

- Beginners face trade-offs between ease of use and flexibility, where some libraries are quick to start but difficult to customize, while others are powerful but intimidating
- Libraries vary widely in abstraction level and technical demands, with tools like Plotly Express enabling quick results and tools like D3.js requiring substantial programming knowledge
- Documentation quality and organization differ significantly, often forcing beginners to combine information from multiple sections to complete basic tasks
- Default interactivity and aesthetics can be both enabling and restrictive, as seen in Bokeh/Plotly Express versus the highly customizable but complex D3.js and Chart.js
- These inconsistencies reveal a gap in beginner-centered design and support, motivating the need for a more accessible and flexible visualization framework

D3.js (JavaScript)

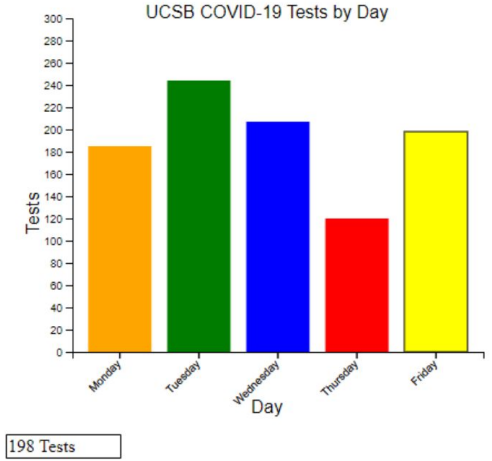
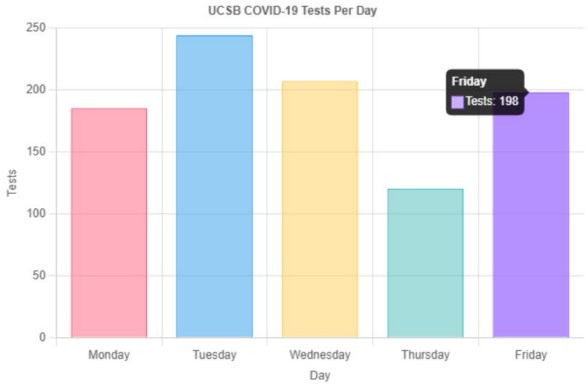


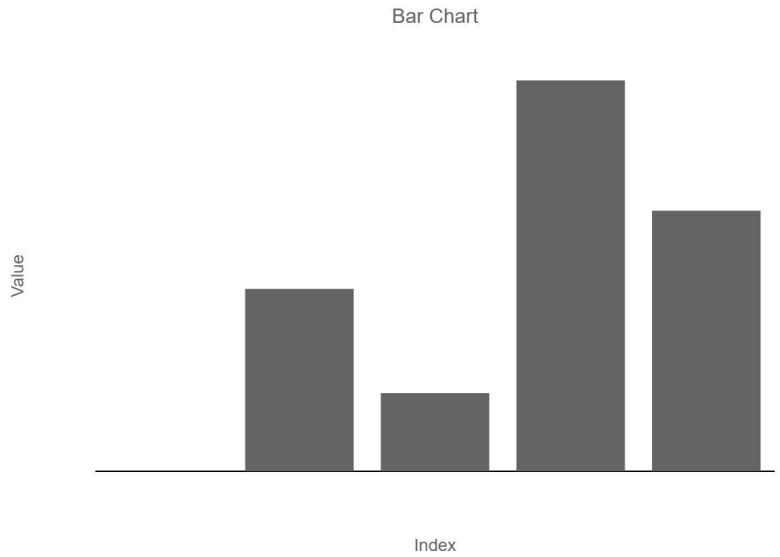
Chart.js (JavaScript)

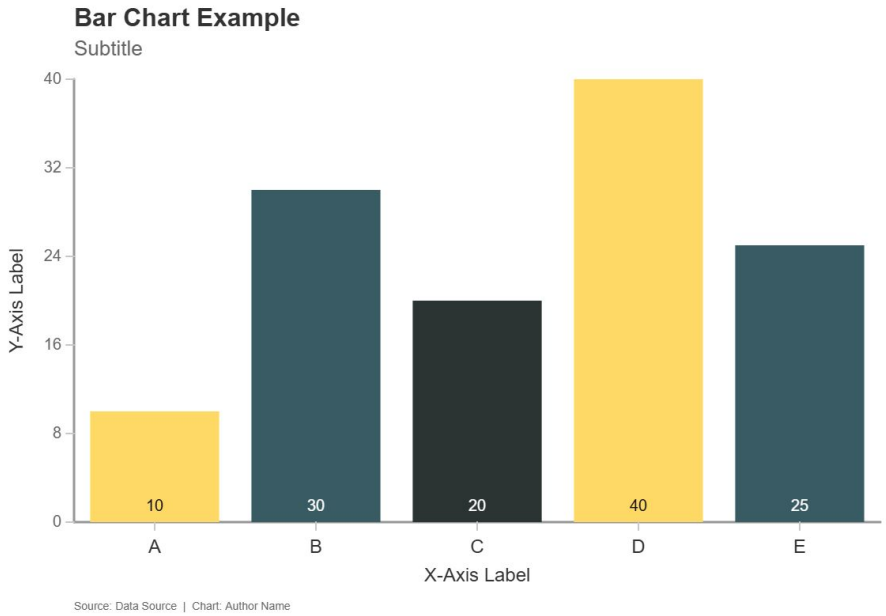


[MAT 237 Final Project]

Process

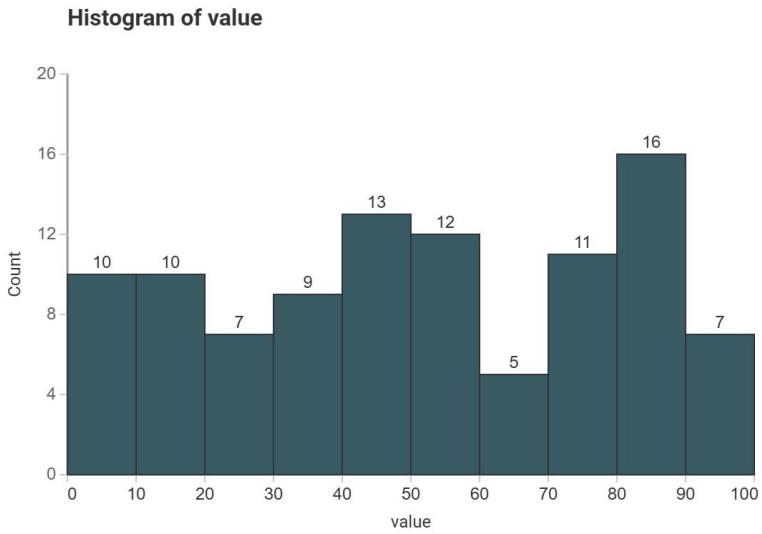
- Iterative Prototyping
- Feature development/Bug-fixing
- Development of aesthetic
- Applying changes cross-library





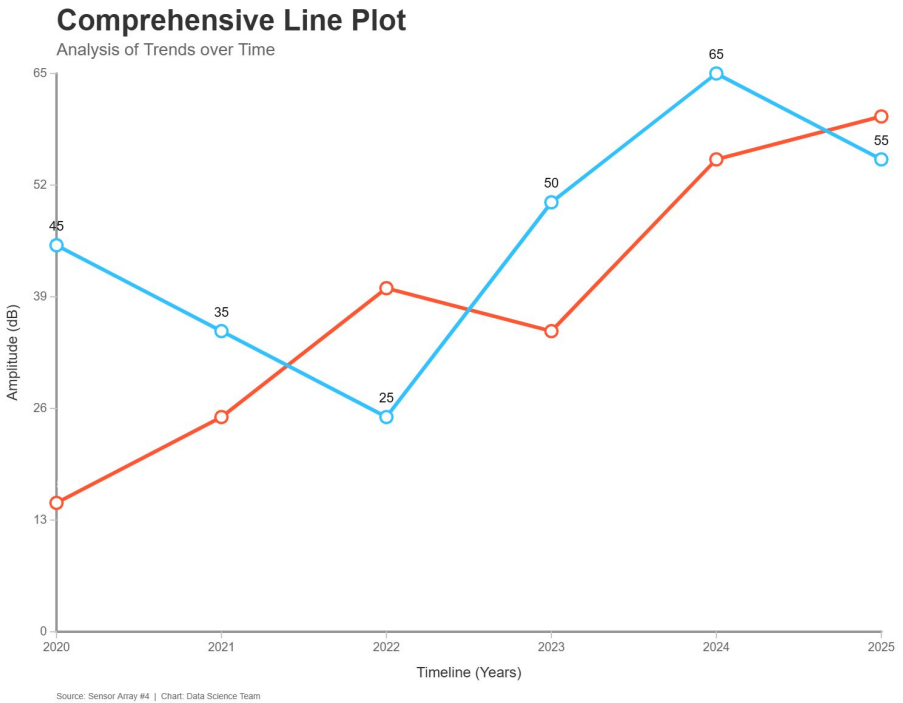
Product

```
function setup() {
  createCanvas(600, 400);
  // Example histogram data: 100
random numbers between 0 and 100
  let data = Array.from({length:
100}, () => ({ value:
Math.floor(Math.random() * 100)
}));
  window.df =
createDataFrame(data);
}

function draw() {
  background(255);
  hist(window.df, {
    x: 'value'
  });
}
```



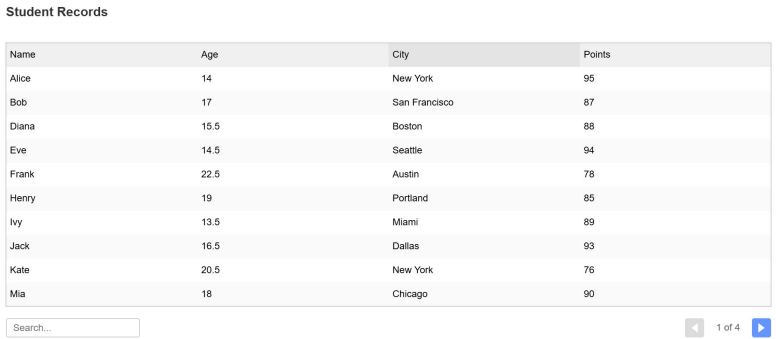
Product

```
let df;

function setup() {
  createCanvas(600, 400);

  let rawData = [
    { red: "A", value: 10 },
    { red: "B", value: 30 },
    { red: "C", value: 20 },
    { red: "D", value: 40 },
    { red: "E", value: 25 }
  ];

  df = createDataFrame(rawData);
}

function draw() {
  background(255);

  bar(df, {
    x: "red",
    y: "value",
    orientation: "vertical",
    mode: "stacked",

    title: "Bar Chart Example",
    subtitle: "Subtitle",
    author: "Author Name",
    source: "Data Source",

    xLabel: "X-Axis Label",
    yLabel: "Y-Axis Label",
    showLabels: true,
    labelPos: "auto",

    width: 600,
    height: 400,
    margin: { top: 60, right: 40, bottom: 60, left: 50
},
    labelSpace: 100,

    background: "#000000",
    palette: ["#FFD966", "#395B64", "#2C3333"],
    showGrid: true,
    gridColor: "#cccccc",

    barWidth: "auto",
    barSpacing: 5,

    hoverEffect: true,
    hoverColor: "#ffcccb",

    font: "Arial",
    textAlign: LEFT,

    titleSize: 16,
    subtitleSize: 13,
    subtitleBold: false,
    authorFontSize: 14,
    xLabelSize: 12,
    yLabelSize: 12,

    debug: true
  });
}
```



Learnings

- Defaults matter
  - If tooltips, labels, or spacing look bad by default, people assume the whole library is broken.
- Edge cases are troublesome
  - Long labels, tiny values, zero bars, and overlapping text caused more bugs than the core bar logic.
- Data is messy in real use, so missing fields, strings instead of numbers, empty datasets, and mixed schemas all broke charts in different ways and I had to account for those
- Default interactivity made the library feel alive. Once hover states and tooltips worked, the charts immediately felt more expressive and usable
- Future
  - Fully using creative aspects of p5.js within each chart
    - More room for play and experimentation
    - Animation/motion



Student Records

Name	Age	City	Points
Alice	14	New York	95
Bob	17	San Francisco	87
Diana	15.5	Boston	88
Eve	14.5	Seattle	94
Frank	22.5	Austin	78
Henry	19	Portland	85
Ivy	13.5	Miami	89
Jack	16.5	Dallas	93
Kate	20.5	New York	76
Mia	18	Chicago	90

Search...

1 of 4