Sam Pottinger
UC Berkeley

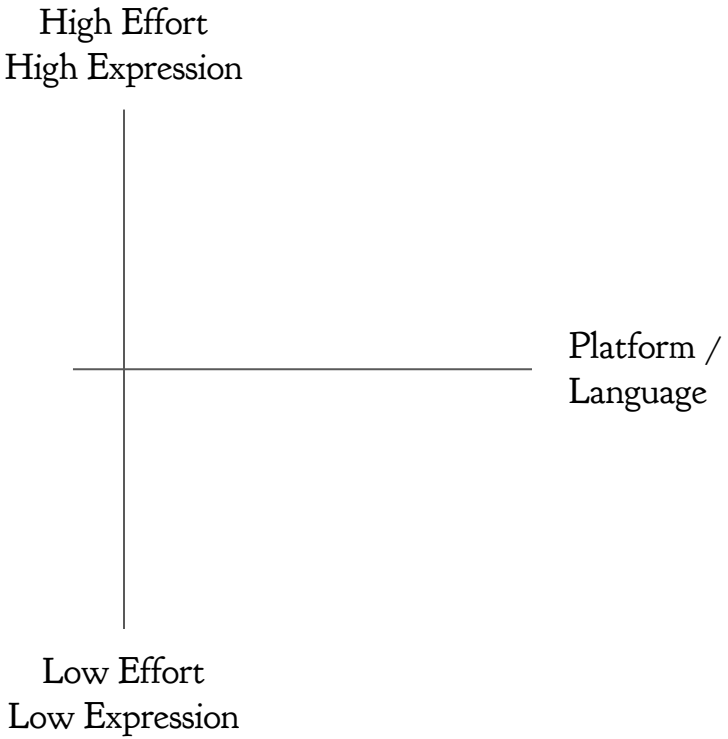
Thinking Through Visualization Tools

Purpose

We are going to talk about different tools available for building visualizations and why we might want to use them in different contexts.

We will also look at the same example dataset visualized using two different approaches.

Data visualization tools in function



There are great functional frameworks out there like in prior work from Krist Wongsuphasawat which provides a great overview of function (effort, outcome, and API design).

I'll tie back into that framework but I want to provide another vantage point.

Tools in function and affordance.

There's some interesting philosophy around tools and thought but an easy example are a paint roller and a brush.

Though their **function** is the same (they both paint), the experience of using them and the type of action they both **afford** are quite different.

The way that they ask you think about painting is different. There's a relationship between tools and the way that they let us think about a task.

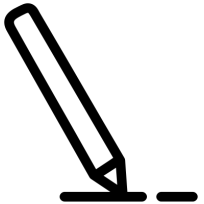
In other words, there's the action we take through the tool and the action the tool takes on us.

Data visualization tools in affordance

The first way of seeing data visualization tools is in how they function (what action we take through them). The other approach is how they influence our thinking about a problem. We will look at tools in both perspectives.

Visualization as language

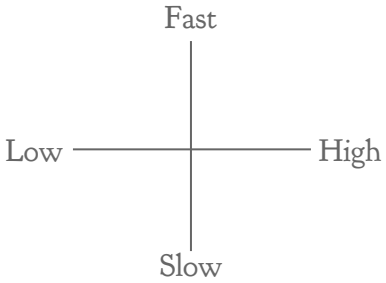
Visualization as representation



Visualization as drawing

A look at tools

We will discuss the functional aspect of these them. How fast it *typically* is to run its course for common visualizations in made with them but also how much typical variation there is in work that they can produce.



We will also discuss how they impact our way of thinking, including by looking at example work done in the type of tool.

What is easy

What is hard

How it influences our thinking

Visualization as language

Typically a set of pre-built chart types. There's nouns: a menu to choose from that have options for configuration but the tool itself defines a basic set of strategies for how information is displayed and how the user is expected to interact with the chart.

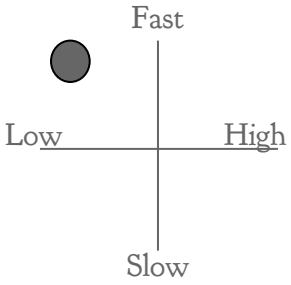


Chart.js

matplotlib

What is easy

Getting something on the page easily in a standardized style.

What is hard

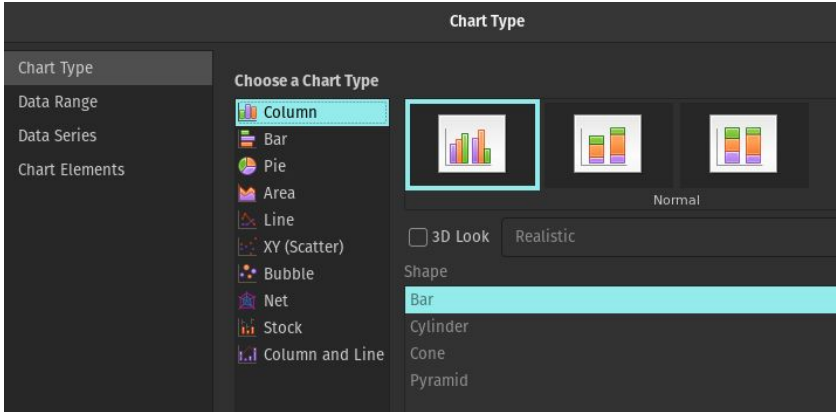
Customization of the structure in which data are displayed, user interaction, and deeply specific styling.

How it influences our thinking

Encourages us to think in specific formats (bar, scatter, doughnut, pie, line) and anything outside those formats becomes unreachable.

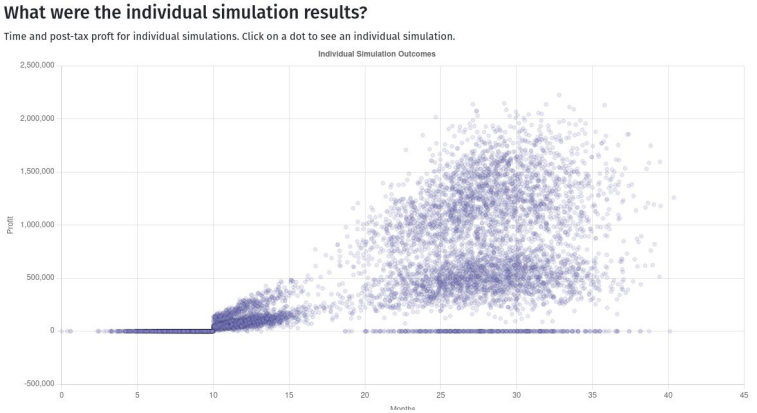
Visualization as language in the world

You might recognize...



LibreOffice Chart Wizard

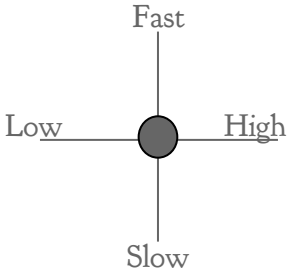
Here's an example for us to consider:



https://startupoptionsbot.com
Using Chart.js

Visualization as representation

"Mapping" different attributes of data like year to horizontal position or cost to vertical size. Tools are often provided for scales and animation. The tool doesn't understand chart types like scatter and line but it is possible to make visual structures that can be difficult to achieve in visualization as language.



What is easy

Representing each data point (like a row in a spreadsheet) visually on a page in sometimes unique / novel visual structures.

What is hard

Morphing between different representations or "movements" (Jonathan Harris), highly customized interaction / animation or unusual drawing strategies.

How it influences our thinking

Allows exploration of unique structures and forces thought on how data map to "encoding mechanisms" but makes it difficult to re-contextualize the data.

Visualization as representation in the world

You might recognize...

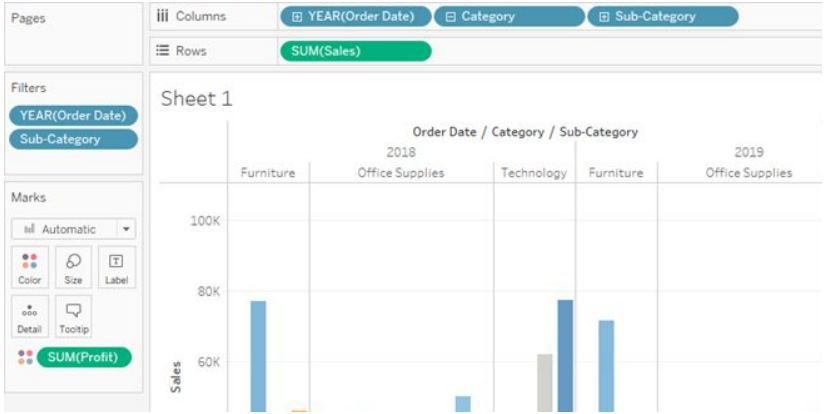
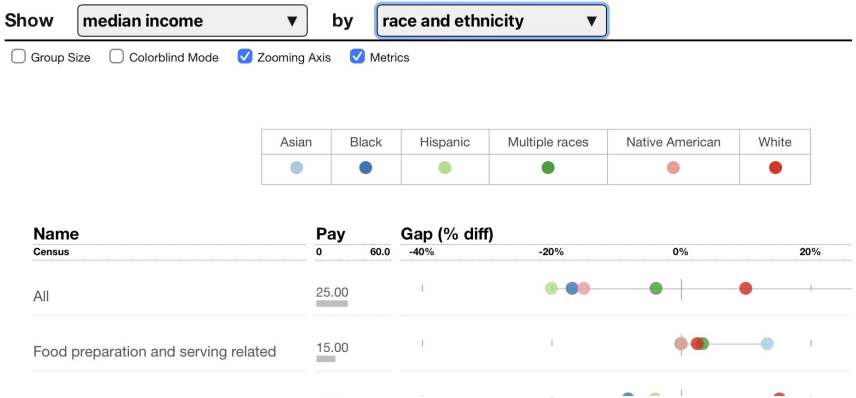


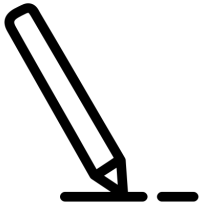
Tableau
(thanks official docs)

Here's an example for us to consider:

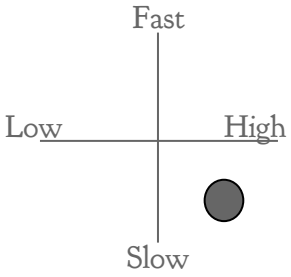


Income Gaps
Using D3

Visualization as drawing



The data guide some drawing process but the method of drawing is left up to the user. Even if there are tools for reading data, the tool doesn't understand what a scale is or define how data are mapped to visual structures. It does however offer tools for drawing complex shapes.



Raphaël

What is easy

Very detailed control over drawing, animation, and interaction pattern. Allows for "movements" or multiple representations and elements not directly mapped from data.

What is hard

These tools often exist in code and it is difficult for non-developers to access these methods.

How it influences our thinking

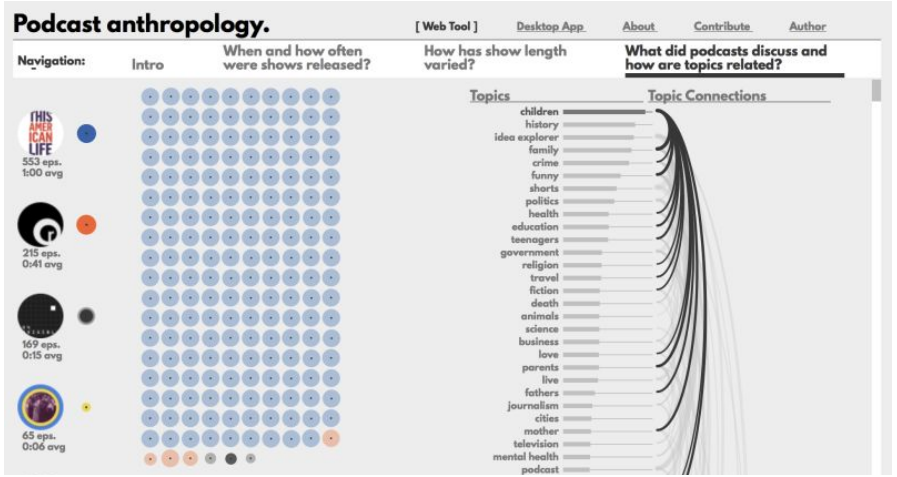
Forces the designer to consider every aspect of the visualization in the problem's context, making every action very intentional.

Visualization as drawing in the world

You might recognize...

[Feltron Annual Reports]
Processing

Here's an example for us to consider:



[Podcast Anthropology]
Using p5.js

What should I use when?

Visualization as language

Don't need it in a very specific format and a simple well established "chart type" is sufficient.

Visualization as representation

There's time and space to explore alternative representations but there's still a clear mapping from data to visual attribute.

Visualization as drawing

Highly specific animation, visual, and interaction design or complex / changing strategy for drawing data.

In general, nothing is "faster" except how quickly the process ends and where it lets you go.

Focus more on what level of depth do you want to provide to the data? How important are they?

Tools can also be used outside their typical affordance.

Example: Can I afford it?

We are going to look at the same dataset two ways: one using the visualization as language and the other using visualization as representation. What are your experiences of the two?

Exploration in Matplotlib

Exploration in D3

Example reflection

- How did your experience in the two tools differ?
- How would you continue your exploration in each?
- How do you feel about the dataset after having gone through each?

What about React?

Visualization code often feels different than the other code on a front-end, especially in the drawing and representation philosophies. There are efforts however to make them more “regular” citizens in the broader codebase.