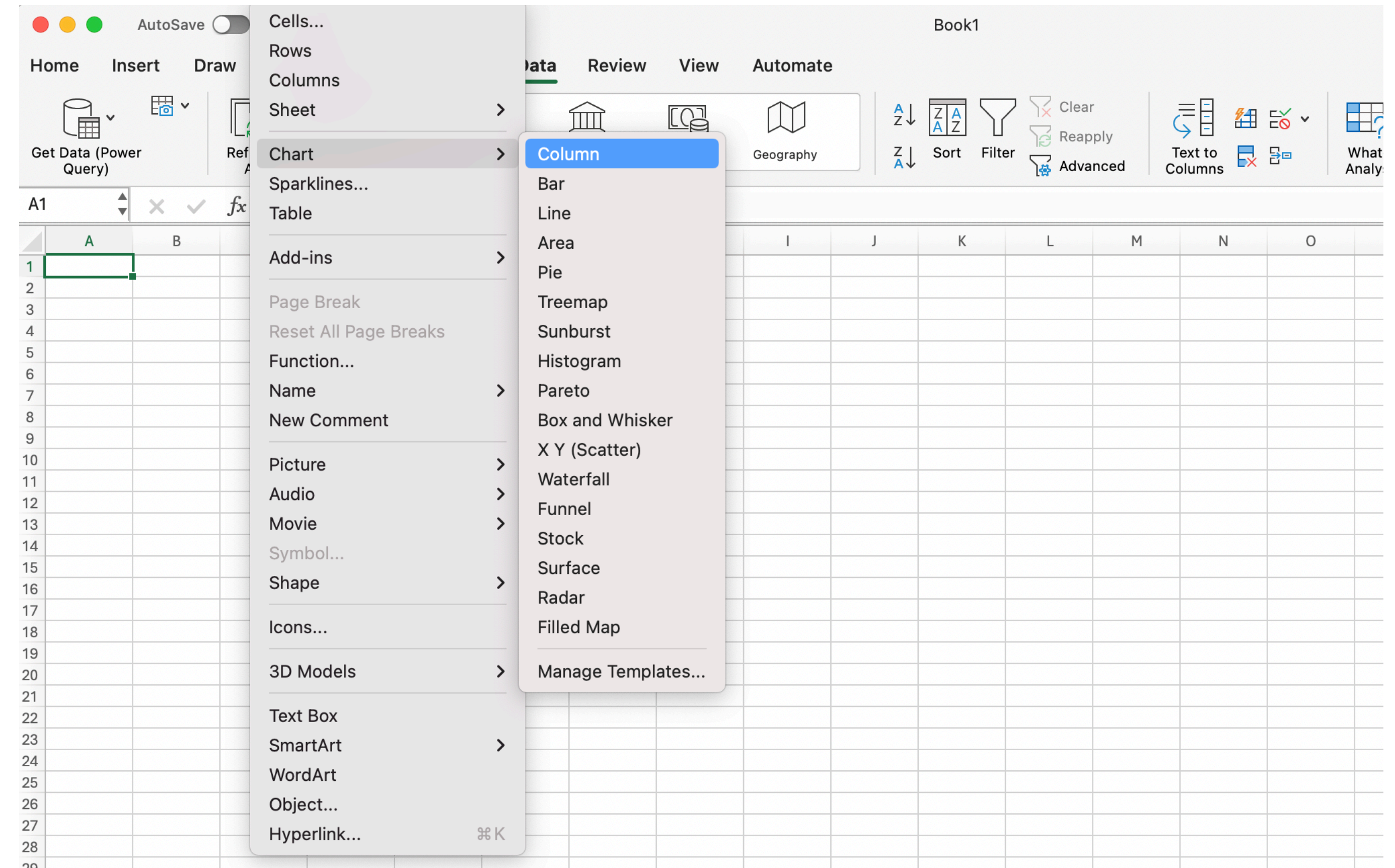


Tools for data visualization

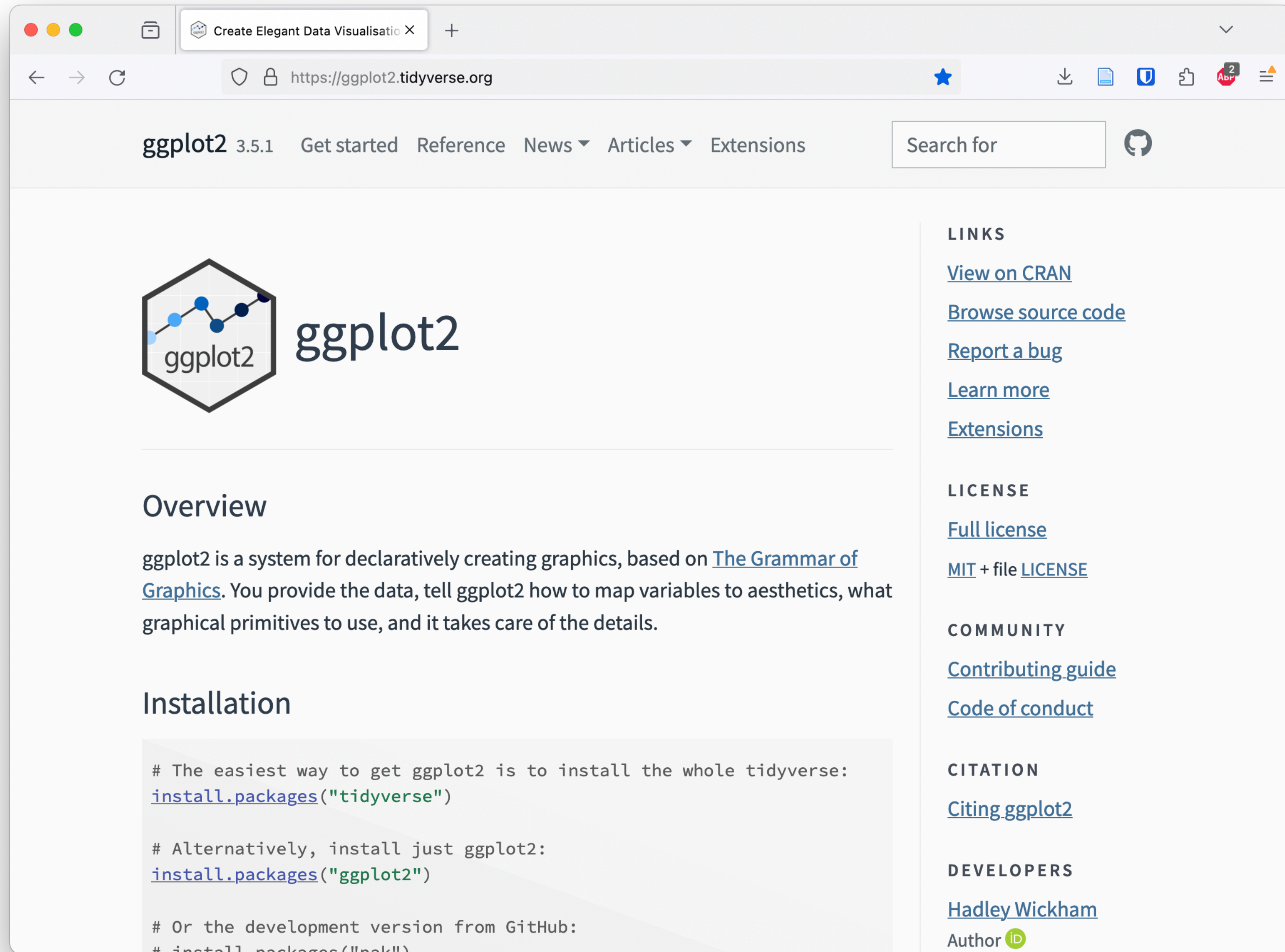
Excel/Google sheets

- Pros:
 - (Almost) everyone has access
 - No code/easy to get started
- Cons:
 - Microsoft products technically do cost money
 - Not an implementation of the Grammar of Graphics/not very customizable




ggplot2 (R package)

- Pros:
 - Free
 - Implementation of the Grammar of Graphics
 - Very customizable
- Cons:
 - Code/can be hard to get started



The screenshot shows the homepage of the ggplot2 R package website. The browser address bar displays "https://ggplot2.tidyverse.org". The navigation menu includes "ggplot2 3.5.1", "Get started", "Reference", "News", "Articles", and "Extensions". A search bar is located on the right side of the navigation menu. The main content area features the ggplot2 logo, which is a hexagon containing a network graph with blue and black nodes and edges, and the text "ggplot2" next to it. Below the logo is the "Overview" section, which describes ggplot2 as a system for declaratively creating graphics based on "The Grammar of Graphics". It states that users provide the data, map variables to aesthetics, and specify graphical primitives, while the package handles the details. The "Installation" section follows, providing R code snippets for installing the package. The code includes instructions for installing the whole tidyverse, just ggplot2, or the development version from GitHub. On the right side of the page, there are several sections of links: "LINKS" with links to "View on CRAN", "Browse source code", "Report a bug", "Learn more", and "Extensions"; "LICENSE" with links to "Full license" and "MIT + file LICENSE"; "COMMUNITY" with links to "Contributing guide" and "Code of conduct"; "CITATION" with a link to "Citing ggplot2"; and "DEVELOPERS" with a link to "Hadley Wickham" and an "Author ID" icon.

ggplot2 3.5.1 Get started Reference News Articles Extensions Search for



ggplot2

Overview

ggplot2 is a system for declaratively creating graphics, based on [The Grammar of Graphics](#). You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Installation

```
# The easiest way to get ggplot2 is to install the whole tidyverse:  
install.packages\("tidyverse"\)  
  
# Alternatively, install just ggplot2:  
install.packages\("ggplot2"\)  
  
# Or the development version from GitHub:  
# install.packages\("pak"\)
```

LINKS

- [View on CRAN](#)
- [Browse source code](#)
- [Report a bug](#)
- [Learn more](#)
- [Extensions](#)

LICENSE

- [Full license](#)
- [MIT + file LICENSE](#)


COMMUNITY

- [Contributing guide](#)
- [Code of conduct](#)

CITATION

- [Citing ggplot2](#)

DEVELOPERS

- [Hadley Wickham](#)
- Author 

ggplot2 (R package)

- Pros:
 - Free
 - Implementation of the Grammar of Graphics
 - Very customizable
- Cons:
 - Code/can be hard to get started

Bar charts — geom_bar · ggplot2

https://ggplot2.tidyverse.org/reference/geom_bar.html

```
# Map class to y instead to flip the orientation
ggplot(mpg) + geom_bar(aes(y = class))
```

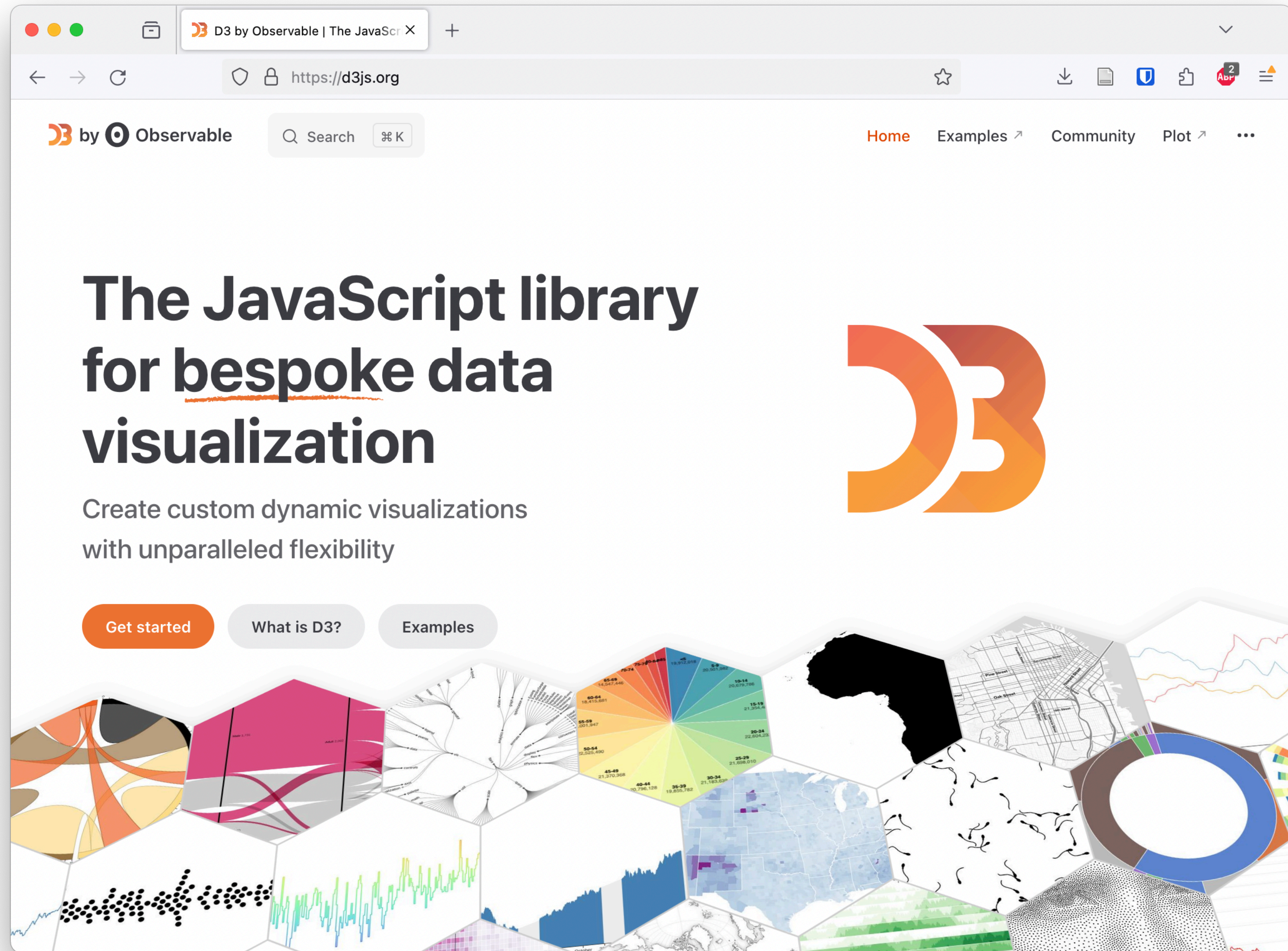
```
# Bar charts are automatically stacked when multiple bars are placed
# at the same location. The order of the fill is designed to match
# the legend
g + geom_bar(aes(fill = drv))
```

ON THIS PAGE

- Usage
- Arguments
- Details
- Orientation
- Aesthetics
- Computed variables
- See also
- Examples

d3.js

- Pros:
 - Free
 - Implementation of the Grammar of Graphics
 - Infinitely customizable
- Cons
 - Code/VERY hard to get started



d3.js

D3 makes things possible, not necessarily easy; even simple things that should be easy are often not. To paraphrase Amanda Cox: "Use D3 if you think it's perfectly normal to write a hundred lines of code for a bar chart."

- Pros:
 - Free
 - [d3 website](#)
 - Implementation of the Grammar of Graphics
 - Infinitely customizable
- Cons
 - Code/VERY hard to get started

Basic barplot in d3.js

https://d3-graph-gallery.com/graph/barplot_basic.html

CHART TYPES QUICK ALL REACT RELATED ABOUT

```
<script>
// set the dimensions and margins of the graph
var margin = {top: 30, right: 30, bottom: 70, left: 60},
    width = 460 - margin.left - margin.right,
    height = 400 - margin.top - margin.bottom;

// append the svg object to the body of the page
var svg = d3.select("#my_dataviz")
    .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform",
        "translate(" + margin.left + "," + margin.top + ")");

// Parse the Data
d3.csv("https://raw.githubusercontent.com/holtzy/data_to_viz/r

// X axis
var x = d3.scaleBand()
    .range([ 0, width ])
    .domain(data.map(function(d) { return d.Country; }))
    .padding(0.2);
svg.append("g")
    .attr("transform", "translate(0," + height + ")")
    .call(d3.axisBottom(x))
    .selectAll("text")
    .attr("transform", "translate(-10,0)rotate(-45)")
    .style("text-anchor", "end");

// Add Y axis
var y = d3.scaleLinear()
```

← Edit me!

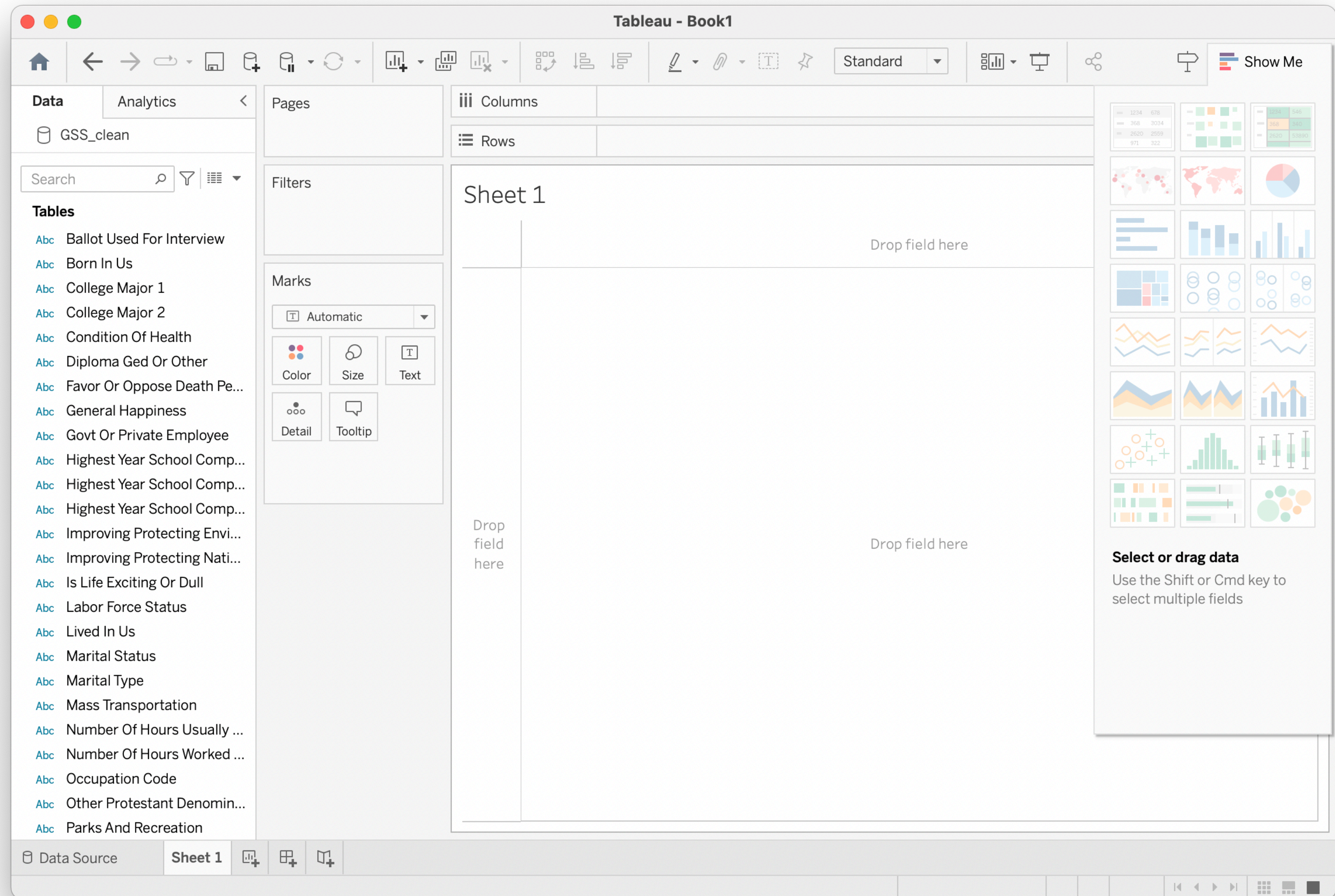
Country	Value
United States	5000
Russia	5000
Germany (FRG)	1800
France	2200
United Kingdom	1200
China	1100
Spain	800
Netherlands	1100
Italy	600
Isrr	1200

Steps:

- The Html part of the code just creates a `div` that will be modified by d3 later on.
- The first part of the javascript code set a `svg` area. It specify the chart size and its margin. [Read more.](#)

Tableau

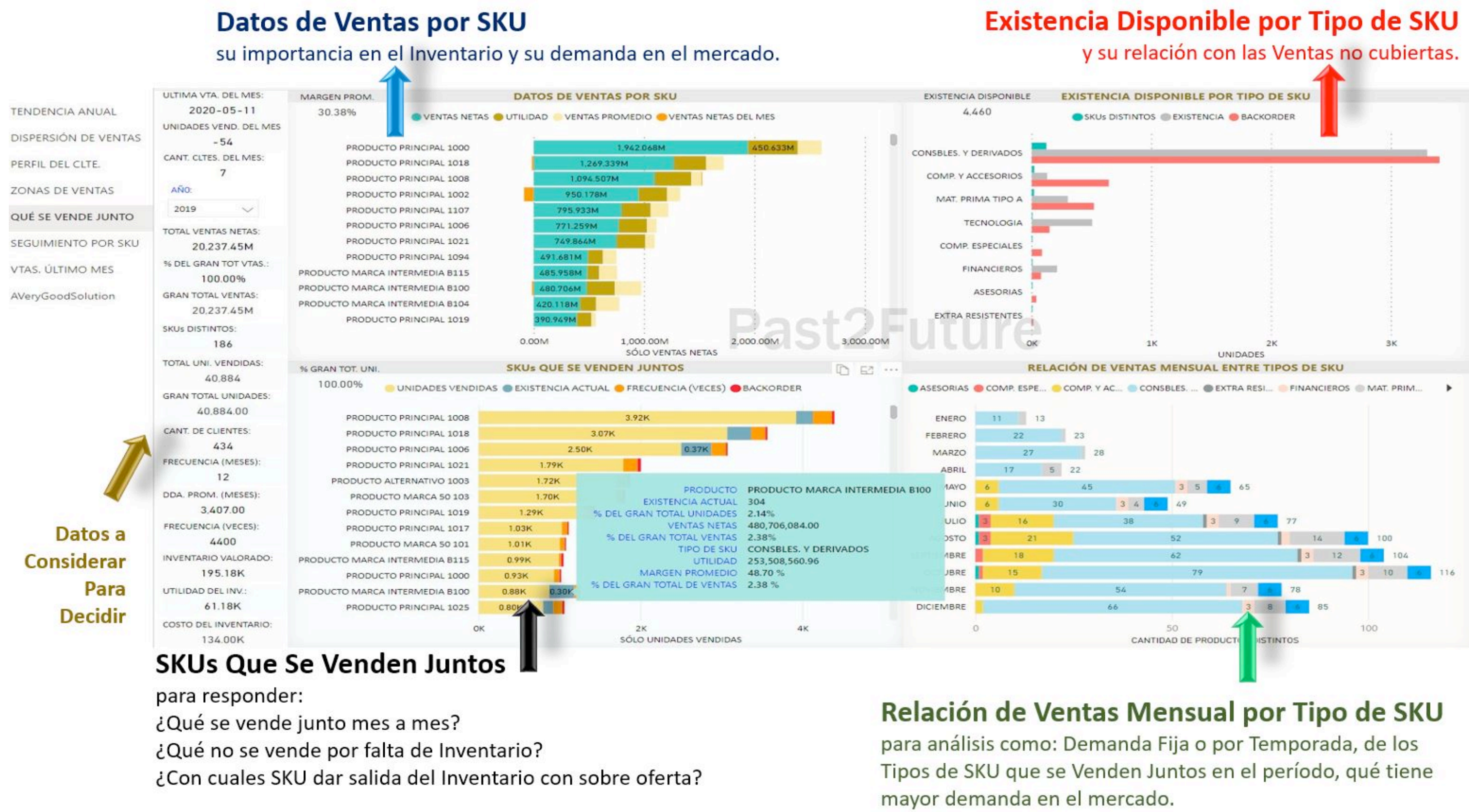
- Pros:
 - No code/easier to get started using
 - Implementation of the Grammar of Graphics
- Cons:
 - Expensive (though they offer free access for educators)
 - Still pretty complicated



Power BI

I'm not very familiar with this one

- Pros:
 - Commonly used in business
 - No code
- Cons:
 - Costs money
 - Hard to customize/may not be an implementation of the Grammar of Graphics



Datawrapper

- Pros:
 - Free
 - No code/easy to get started
- Cons:
 - Not an implementation of the Grammar of Graphics/only customizable to a point
 - Premium features cost money

