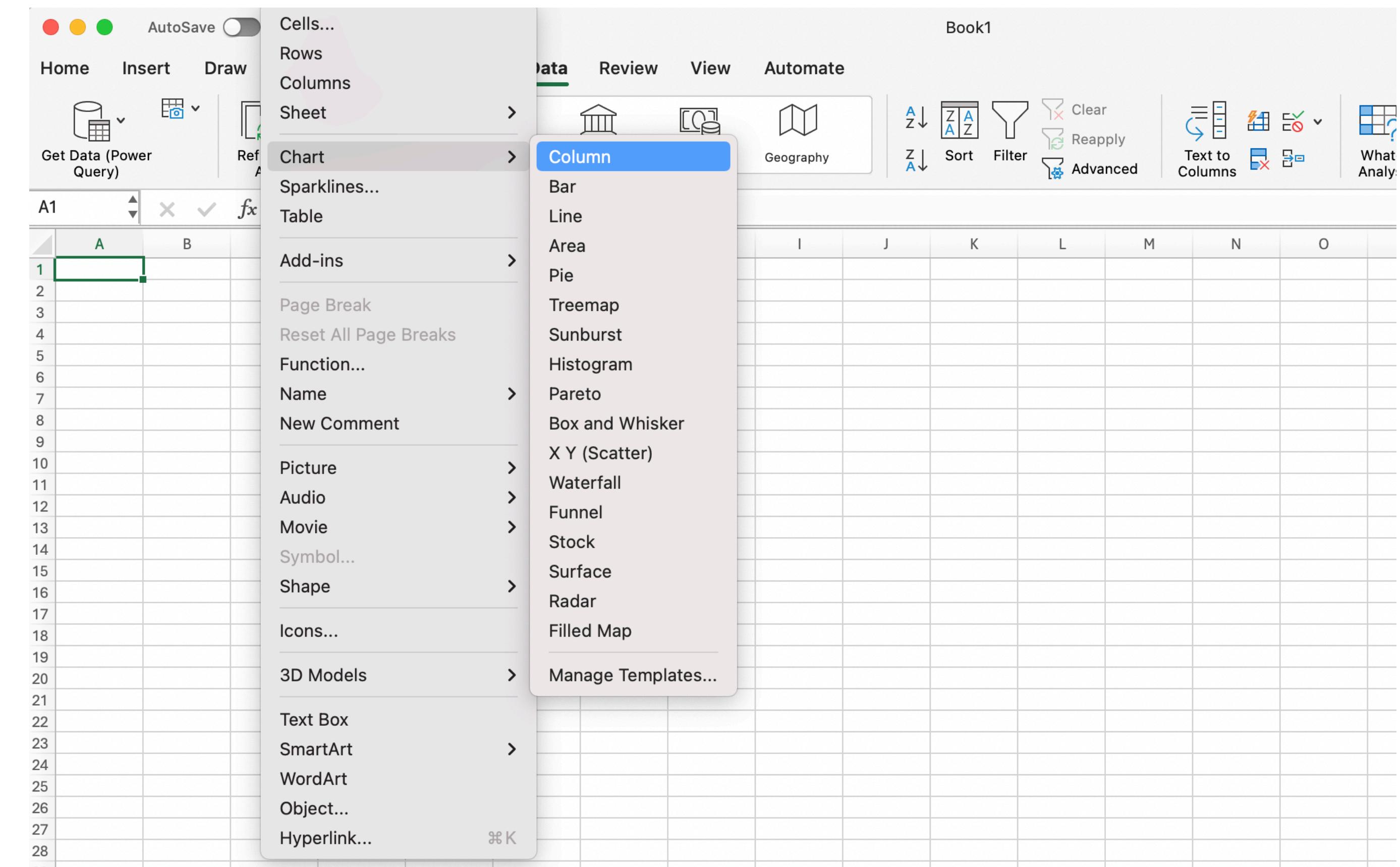


The background of the slide features a large, modern building facade. The facade is covered in a pattern of blue and purple triangles, creating a geometric, star-like appearance. A grey concrete ledge or walkway is visible on the left side of the frame.

# 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 a web browser window displaying the official ggplot2 website at <https://ggplot2.tidyverse.org>. The page features a large hexagonal logo with a grid and several colored dots, followed by the word "ggplot2". The main content includes sections for "Overview" and "Installation", along with a sidebar containing links for various resources like CRAN, source code, and bug reporting.

**Navigation Bar:** ggplot2 3.5.1 | Get started | Reference | News | Articles | Extensions | Search for

**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 ID

**Overview**

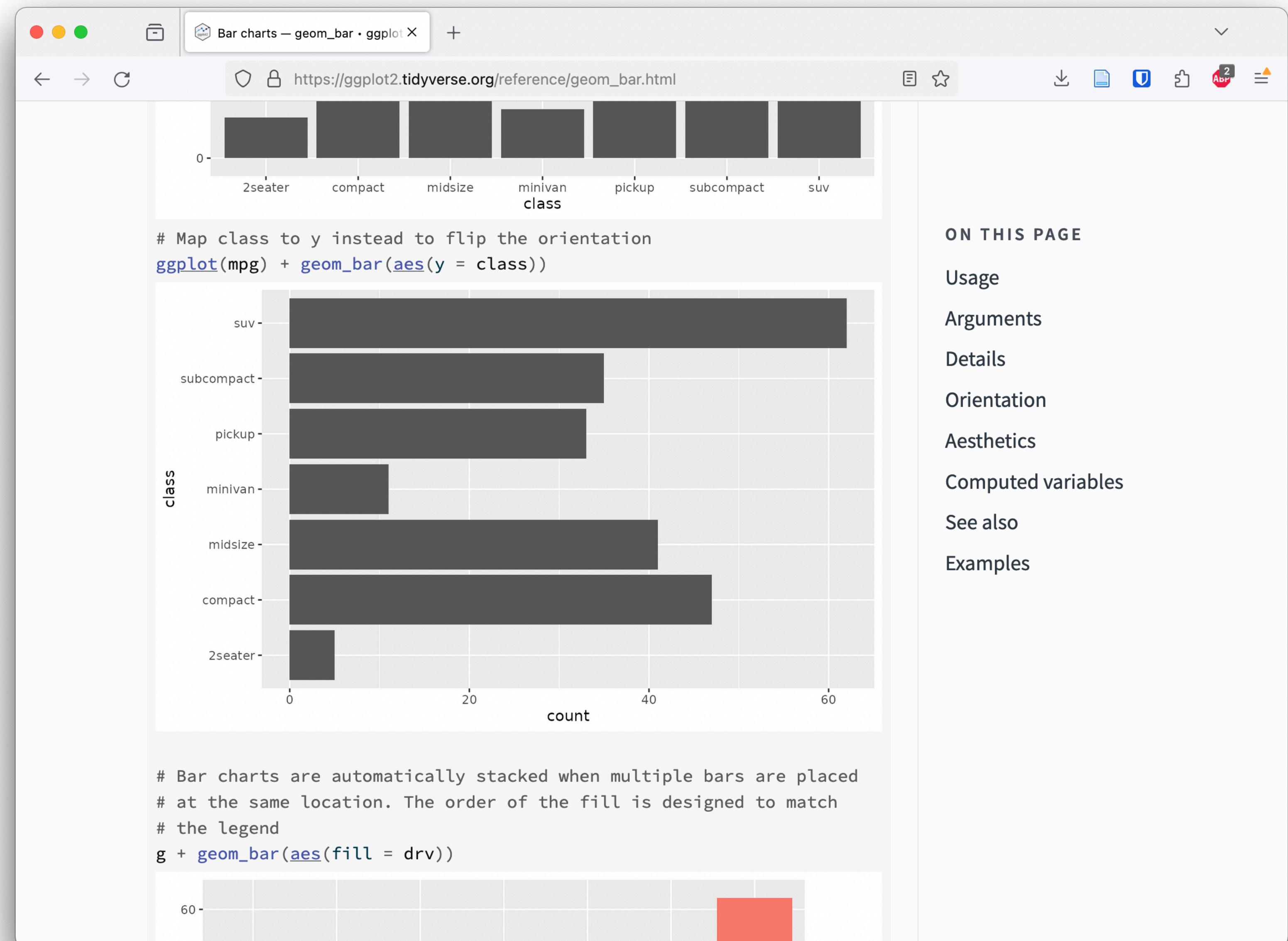
ggplot2 is a system for declaratively creating graphics, based on [The Grammar of Graphics](#). You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

**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\("pak"\)
```

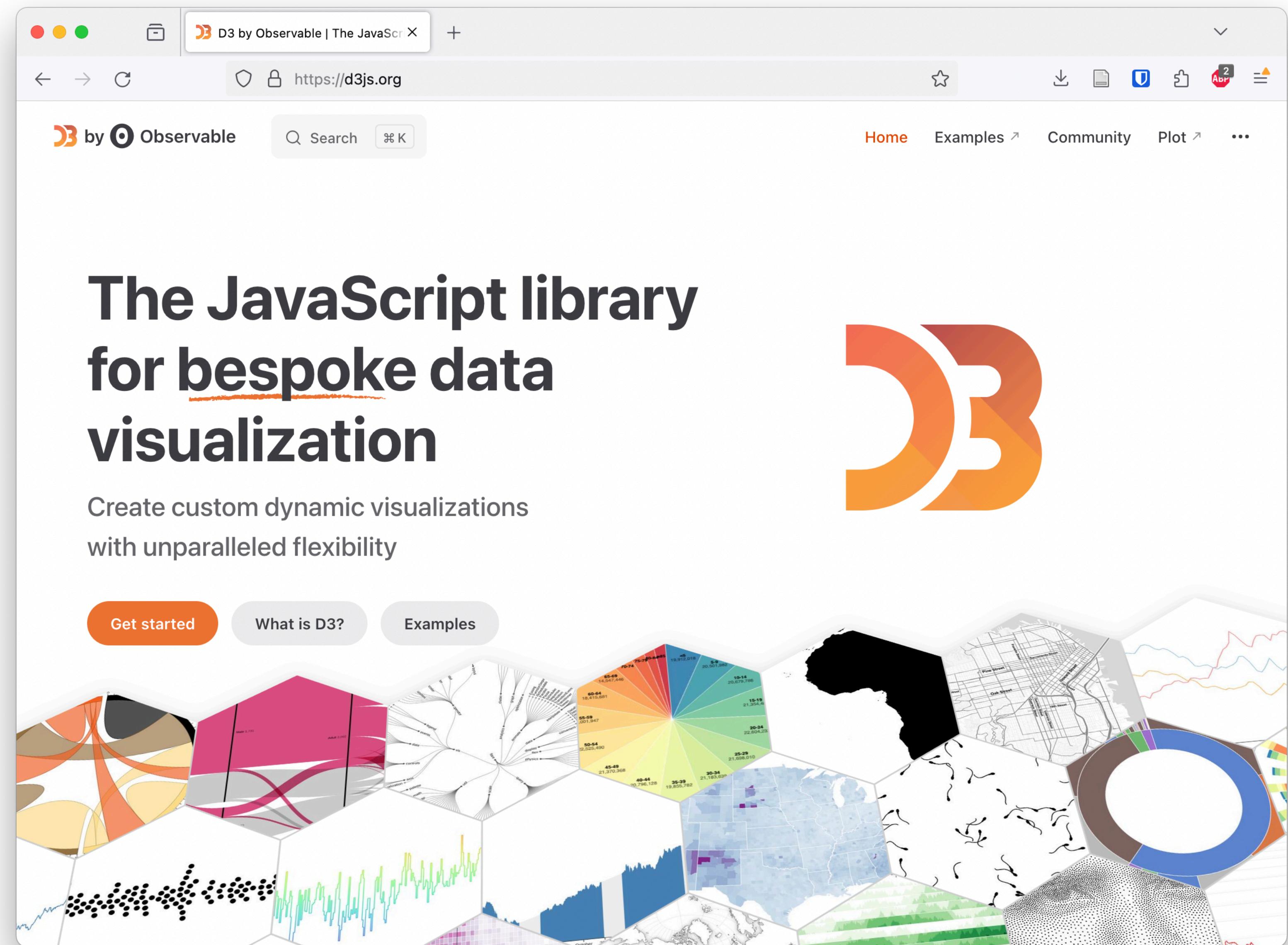
# ggplot2 (R package)

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



# d3.js

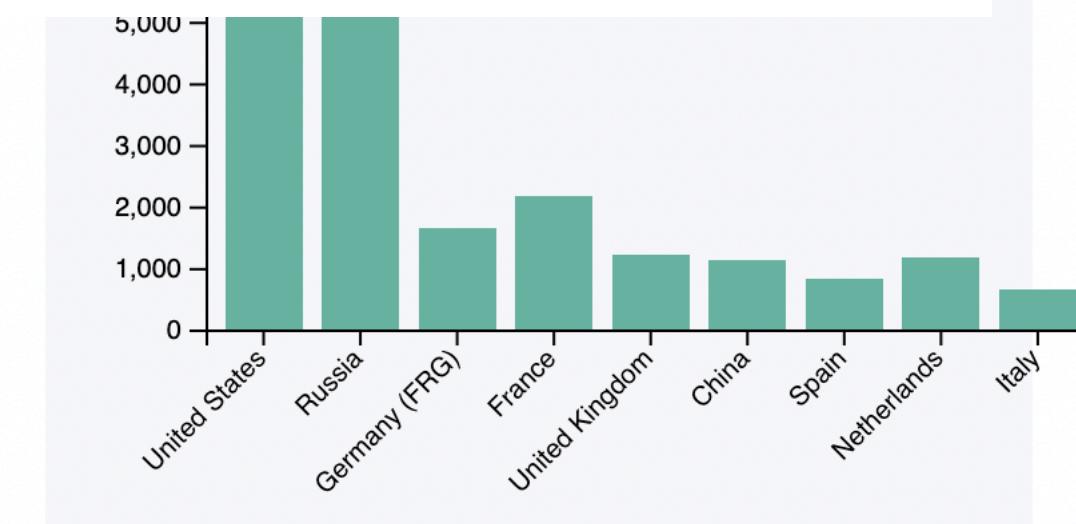
- Pros:
  - Free
  - Implementation of the Grammar of Graphics
  - Infinitely customizable
- Cons
  - Code/VERY hard to get started



# d3.js

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

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.”



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](#).

```
<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 + ")");

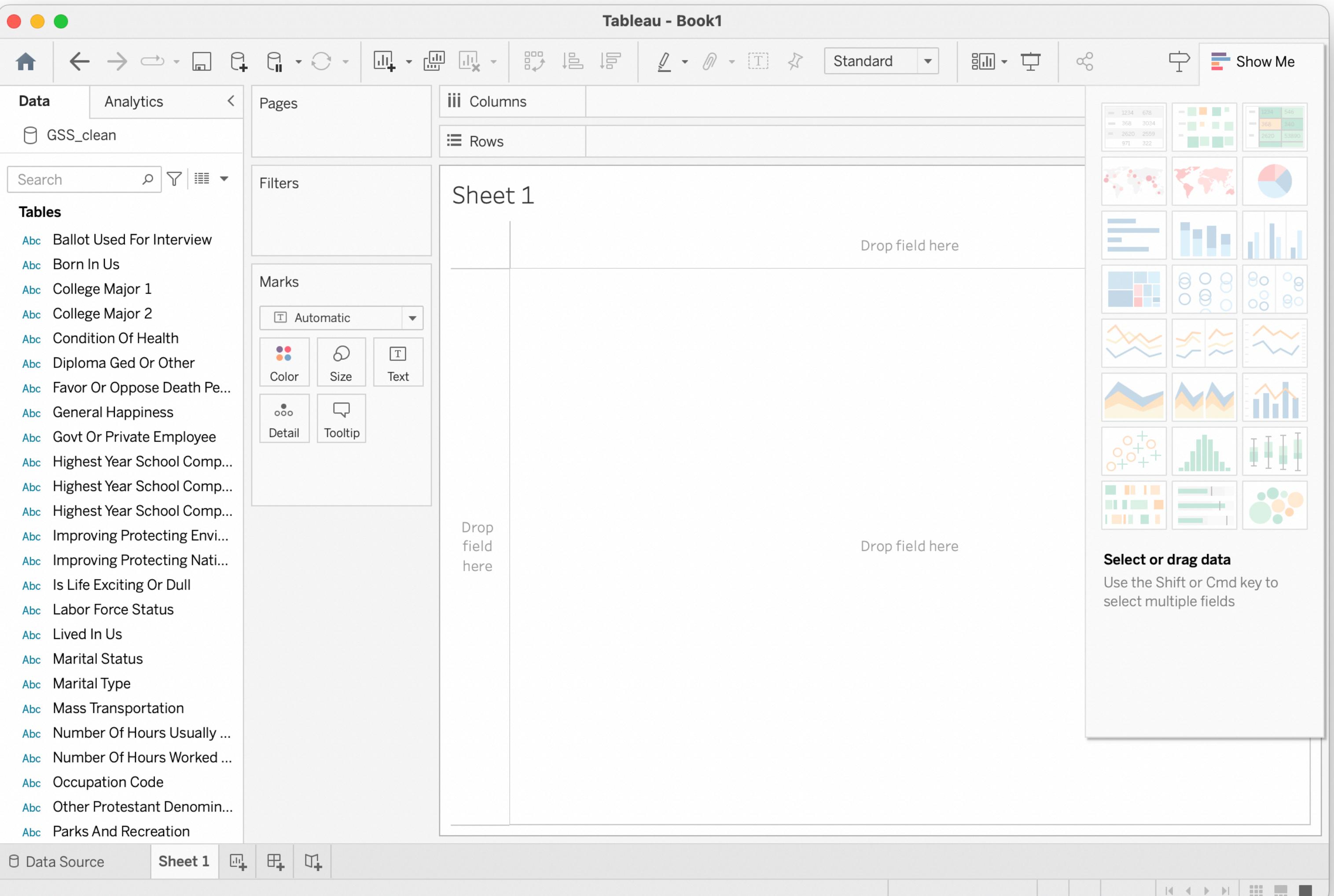
// Parse the Data
d3.csv("https://raw.githubusercontent.com/holtzy/data_to_viz/r

// 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 v = d3.scaleLinear()
```

# 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

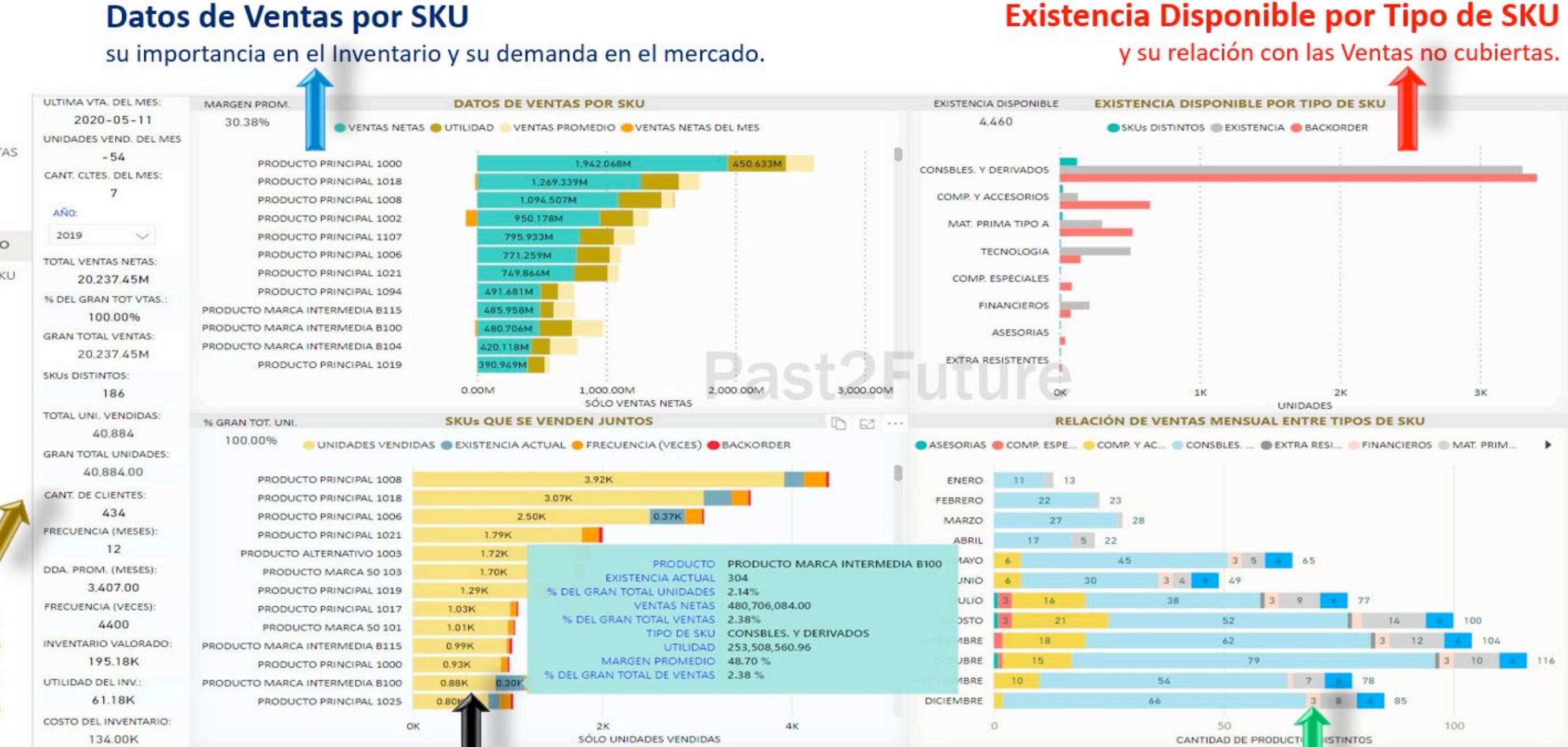


# Power BI

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

**Datos a  
Considerar  
Para  
Decidir**



**Relación de Ventas Mensual por Tipo de SKU**  
para análisis como: Demanda Fija o por Temporada, de los Tipos de SKU que se Venden Juntos en el período, qué tiene mayor demanda en el mercado.

CC BY-SA 4.0 [Usrdewiki](#)

# 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

