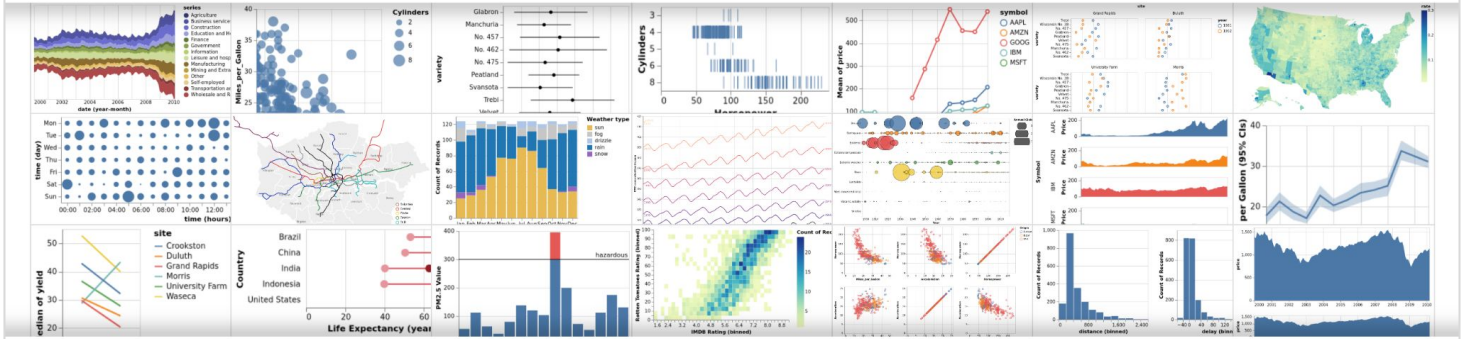
Introduction to Vega-Lite

Gustavo Moreira, University of Illinois Chicago
Marcos Lage, Universidade Federal Fluminense
Nivan Ferreira, Universidade Federal de Pernambuco
Fabio Miranda, University of Illinois Chicago

Vega-Lite – A Grammar of Interactive Graphics



Vega-Lite is a high-level grammar of interactive graphics. It provides a concise, declarative JSON syntax to create an expressive range of visualizations for data analysis and presentation.

Vega-Lite specifications describe visualizations as encoding mappings from data to **properties of graphical marks** (e.g., points or bars). The Vega-Lite compiler **automatically produces visualization components** including axes, legends, and scales. It determines default properties of these components based on a set of **carefully designed rules**. This approach allows Vega-Lite specifications to be concise for quick visualization authoring, while giving user control to override defaults and customize various parts of a visualization. As we also designed Vega-Lite to support data analysis, Vega-Lite supports both **data transformations** (e.g., aggregation, binning, filtering, sorting) and **visual transformations** (e.g., stacking and faceting). Moreover, Vega-Lite specifications can be **composed** into layered and multi-view displays, and made **interactive with selections**.

Get started
Latest Version: 5.16.0

Try online

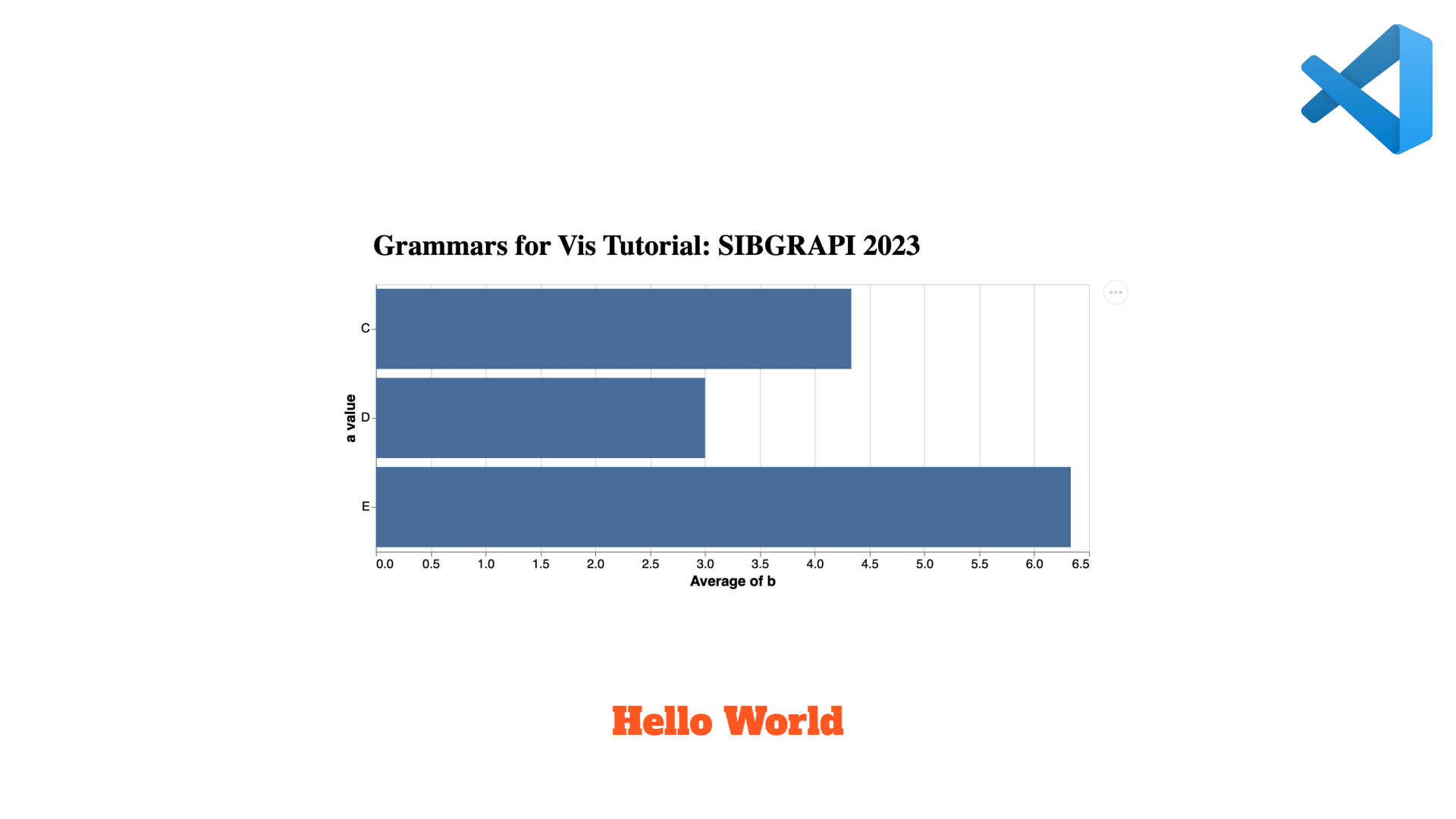
Compared to Vega, Vega-Lite provides a more concise and convenient form to author common visualizations. As Vega-Lite can compile its specifications to Vega specifications, users may use Vega-Lite as the *primary* visualization tool and, if needed, transition to use the lower-level Vega for advanced use cases.

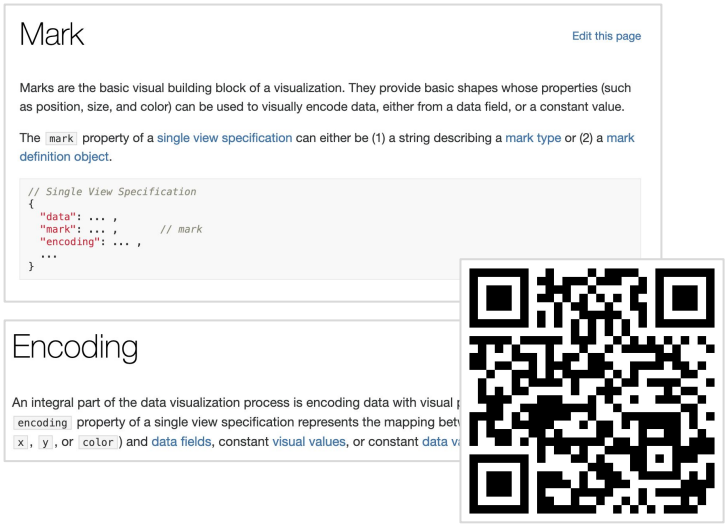
For more information, read our introduction article to Vega-Lite v2 on Medium, watch our OpenVis Conf talk about the new features in Vega-Lite v2, see the documentation and take a look at our example gallery. Follow us on Twitter at @vega_vis to stay informed about updates.





The development of Vega-Lite is led by the alumni and members of the University of Washington Interactive Data Lab (UW IDL), including Kanit “Ham” Wongsuphasawat (now at Databricks), Dominik Moritz (now at CMU / Apple), Arvind Satyanarayan (now at MIT), and Jeffrey Heer (UW IDL).


Grammars for Vis Tutorial: SIBGRAPI 2023
a value
C
D
E
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
Average of b
Hello World



Mark

Edit this page

Marks are the basic visual building block of a visualization. They provide basic shapes whose properties (such as position, size, and color) can be used to visually encode data, either from a data field, or a constant value.

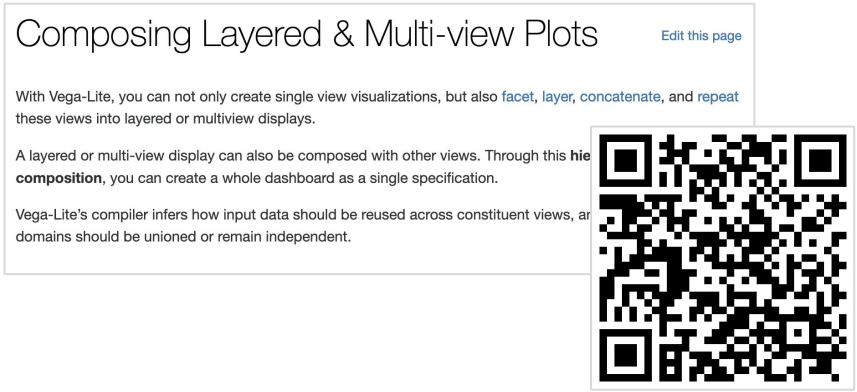
The `mark` property of a single view specification can either be (1) a string describing a mark type or (2) a mark definition object.

```
// Single View Specification
{
  "data": ... ,
  "mark": ... ,      // mark
  "encoding": ... ,
  ...
}
```

Encoding

An integral part of the data visualization process is encoding data with visual p
`encoding` property of a single view specification represents the mapping betw
`x`, `y`, or `color`) and data fields, constant visual values, or constant data va

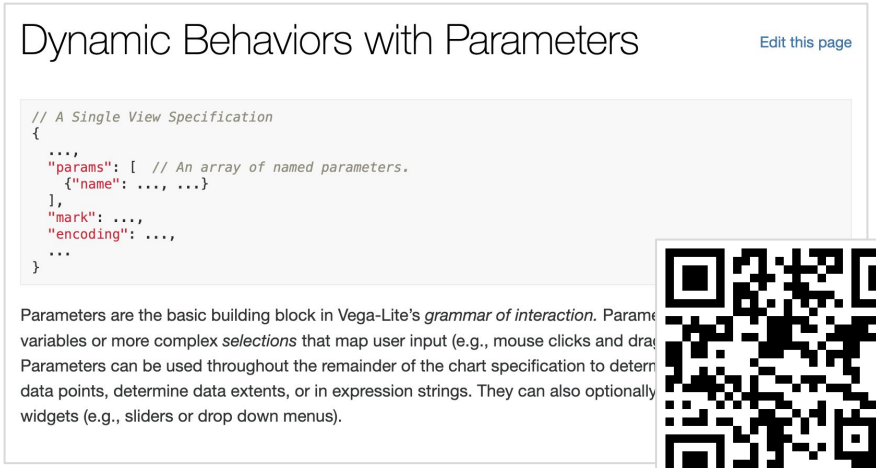
1. Marks and Encodings



2. Layers & Multi-view

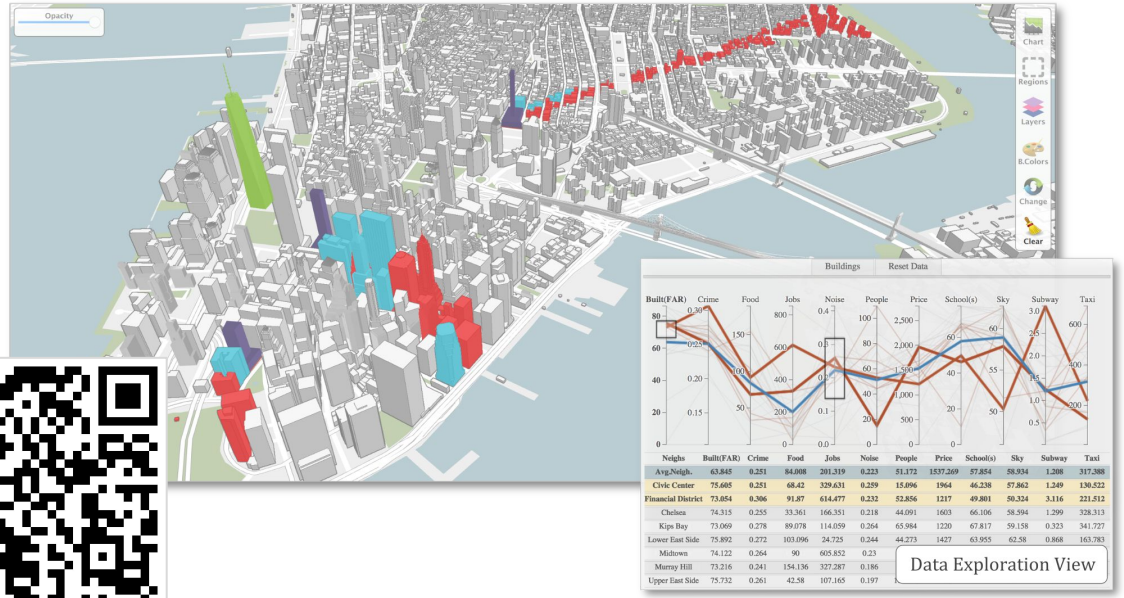
3. Layers & Multi-view

4. Transformations



5. Interactions

6. The best stats you've ever seen, Hans Rosling



7. Urbane, IEEE VIS 2015

Introduction to Vega-Lite

Gustavo Moreira, University of Illinois Chicago
Marcos Lage, Universidade Federal Fluminense
Nivan Ferreira, Universidade Federal de Pernambuco
Fabio Miranda, University of Illinois Chicago