

Using R and ggvis to Create Interactive Graphics for Exploratory Data Analysis

Creating interactive web graphics requires a heterogeneous set of technical skills that can include data munging, analytics, web development and design. Data needs to be acquired, cleaned, transformed and then sent to a web app for rendering as a plot. This creates a barrier to entry for an analyst to effectively explore data and create interactive graphics in a single language in a flexible, iterative and reproducible way. This chapter will cover how a new **R** package, **ggvis**, lets librarians explore and communicate results using interactive graphics without the overhead of tinkering with javascript frameworks. **ggvis** brings together three concepts to make data visualization more consistent: 1) it is based on the *grammar of graphics*, a structure for defining the elements of a graphic; 2) it produces reactive and interactive plotting in a rendered web-based graphic and; 3) it provides support for creating data pipelines making data manipulation and plotting code more readable.

I'll start the chapter by going over the technology suite and dependencies needed to run **ggvis**. I'll also cover the data used in the chapter and

where to obtain it. Once the computing and data sources are covered, I'll introduce how **ggvis** utilizes the *grammar of graphics* to break graphs into components made up of *data*, a *coordinate system*, a *mark type* and *properties*. This modularity allows analysts to better understand graphical elements and swap out different components to make up of a multitude of different plots. Following the *grammar of graphics*, I'll start using **ggvis** to build simple static graphs with built-in datasets in **R**. While demonstrating how to create simple plots, such as, *histograms*, *bar charts*, and *scatter plots*, I'll introduce how **ggvis** incorporates the *data pipelining* syntax to improve code readability and maintenance. I will also show how **ggvis** employs features from the data manipulation package, **dplyr**, to filter, summarize, or transform the data before graphing.

After covering the **ggvis** basic features, I'll introduce interactivity by plotting library datasets (gate count data, ejournal usage, catalog data). Starting simply, I'll will add a bin slider to a histogram . I will then introduce how to create interactive scatterplots and line graphs with various interactive inputs (a third variable fill color, hover events, etc.). Finally, I'll introduce how to embed **ggvis** plots in markdown enabled **R** (Rmarkdown) to create interactive reports. We will also briefly cover how to embed and publish **ggvis** plots in a web application.

ggvis is still very young and currently not recommended for production use. However, **ggvis** is under active development and co-authored by two important **R** programmers with a track record of delivering significant **R**

packages.

ggvis opens the door for a librarian with knowledge of **R** to create interactive graphics without knowing the intricacies of a javascript framework or data format transformations. Furthermore, through the *grammar of graphics* and *data pipelines*, **ggvis** makes graphing code understandable and more reproducible.

Tim Dennis

UCSD Data Librarian

Bio

Tim Dennis is the Data Services and Collections Librarian in UCSD's Geisel Library. He provides consultations on research involving numeric data, including finding and recommending data sources and advising on technical data issues such as file format conversion, web scraping, and basic statistical software use. Prior to UCSD, he was at UC Berkeley where he was actively involved in providing data consultation as part of the UC Berkeley Library Data Lab. He attended the School of Information at UC Berkeley receiving a Master of Information Management and Systems.