

# post02-lindsay-dahlen

Lindsay Dahlen

December 3, 2017

## ggvis: Visualizing data and its properties

### 1. ggvis

In this post I will talk about the data visualization package ggvis. We were required to use ggvis in our interactive shiny app as part of homework 4. I hope to expand on what exactly ggvis is and various ways you can use it, particularly in comparison to the package ggplot2. I will not be using a shiny app in this post, but will be using ggvis in Rstudio.

### 2. Introduction

ggvis is a package that combines the data computing power of R and the graphic rendering power of web browsers, using "Vega". (Vega is a visualization grammar, which I won't be discussing in this post). Rstudio can display ggvis because its a web browser. The graphics will display in the viewer panel.

### 3. Set-up

In the following examples I will be using the Iris dataset, which was created by R. A. Fisher using data collected by E. Anderson. It "iris is a data frame with 150 cases (rows) and 5 variables (columns) named *Sepal.Length*, *Sepal.Width*, *Petal.Length*, *Petal.Width*, and *Species*."

```
# Summary of iris dataset
summary(iris)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
## Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100
## 1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300
## Median :5.800 Median :3.000 Median :4.350 Median :1.300
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199
## 3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800
## Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500
## Species
## setosa :50
## versicolor:50
## virginica :50
##
##
##
```

```
head(iris)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1 5.1 3.5 1.4 0.2 setosa
## 2 4.9 3.0 1.4 0.2 setosa
## 3 4.7 3.2 1.3 0.2 setosa
## 4 4.6 3.1 1.5 0.2 setosa
## 5 5.0 3.6 1.4 0.2 setosa
## 6 5.4 3.9 1.7 0.4 setosa
```

I installed ggvis as follows:

```
# install.packages("ggvis")
library(ggvis)
```

I also loaded the *dplyr* package, which is necessary to write for ggvis. Additionally, I loaded the *ggplot2* package to show comparisons between ggplot and ggvis.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
## filter, lag
```

```
## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
##  
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:ggvis':  
##  
## resolution
```

## 4. The Basics of ggvis

The components of ggvis include **data**, or the data frame; **layer**, or the layers of the plot components; **mappings**, which map between variables and aesthetics; and **~**, which are equivalent to aes mappings in ggplot2.

Unlike ggplot2, ggvis uses three different operators to visualize the data. **\*\* %>%** This is the pipe operator. It passes the result from its left-hand side into the first argument of the function on its right-hand side.

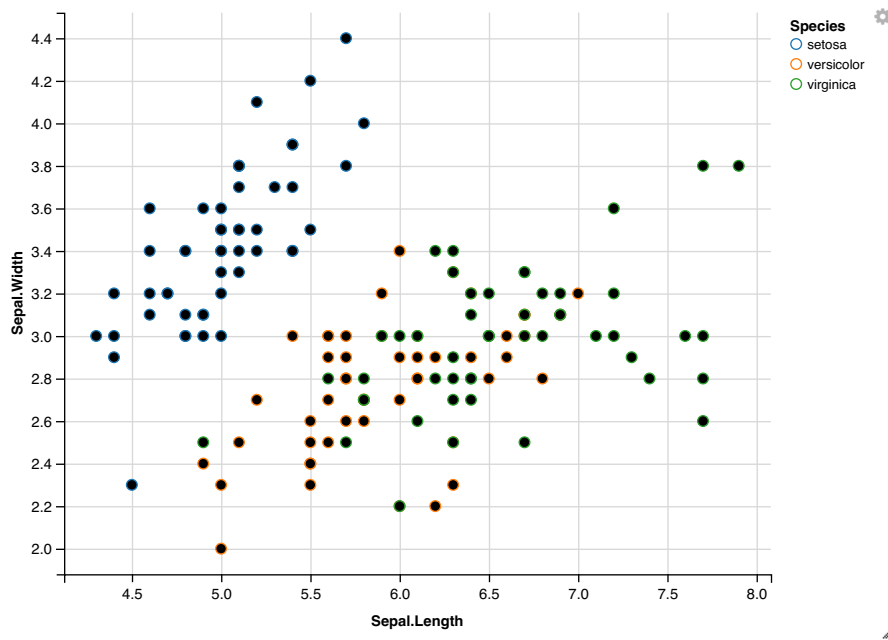
**=** maps a property to a data value or a set of data values. This is how you visualize variation in your data set. ggvis will scale the values appropriately and add a legend that explains how values are mapped to particular instances of the property.

**:=** sets a property to a specific color (or size, width, etc.), and is how you can customize the plot appearance

## 5. Scatterplot in ggvis

Let's start off simple with a scatterplot. To make a scatterplot of sepal lengths versus sepal widths for each flower record using ggvis, set up your code as follows.

```
iris %>% ggvis(~Sepal.Length, ~Sepal.Width, stroke = ~Species) %>% layer_points()
```



As you can see, even without inputting many instructions in the code, ggvis allows you to easily set up a scatterplot that is labeled, color coded by “stroke” (which in this case was set to species), and is easy to understand.

## 6. Interactive ggvis

Because ggvis uses web browsers for its visualization, we can add interactive elements to the graph. For example, let's try adding a slider to the opacity of the points on the scatterplot.

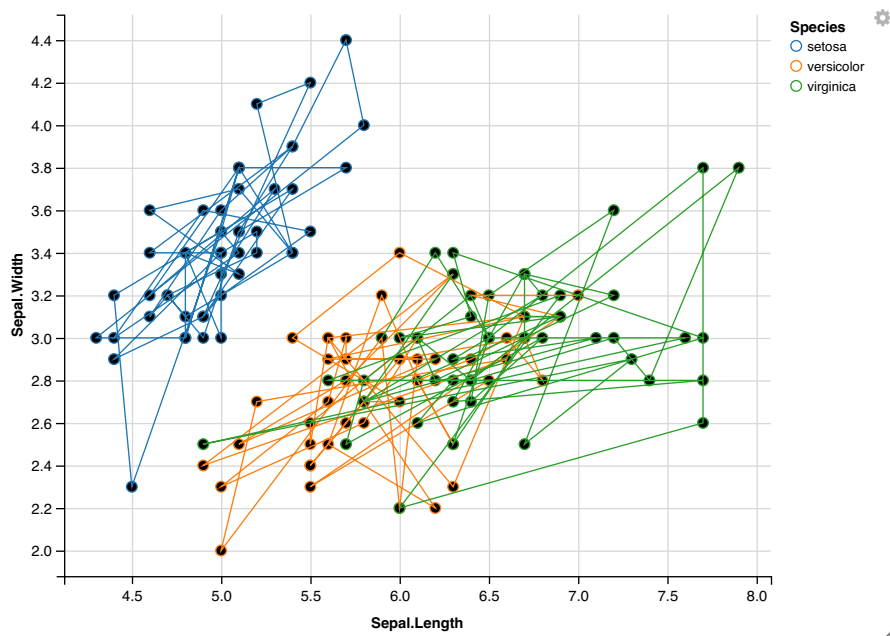
```
slider <- input_slider(10, 250)  
# iris %>% ggvis(~Sepal.Length, ~Sepal.Width, stroke = ~Species) %>% # Remove '#' from code to run  
# layer_points(size := slider) %>%  
# layer_points(fill := "grey4", opacity := 0.5, size := slider)  
# The slider controls both opacity and color of the dots
```

Note: The slider will not operate in html. It will work in RStudio though.

## 7. Grouping in ggvis

Let's try one more example, this time by looking at grouping. To group, use the function “group\_by”

```
iris %>% ggvis(~Sepal.Length, ~Sepal.Width, stroke = ~Species) %>%  
  layer_points() %>%  
  group_by(Species) %>%  
  layer_paths()
```



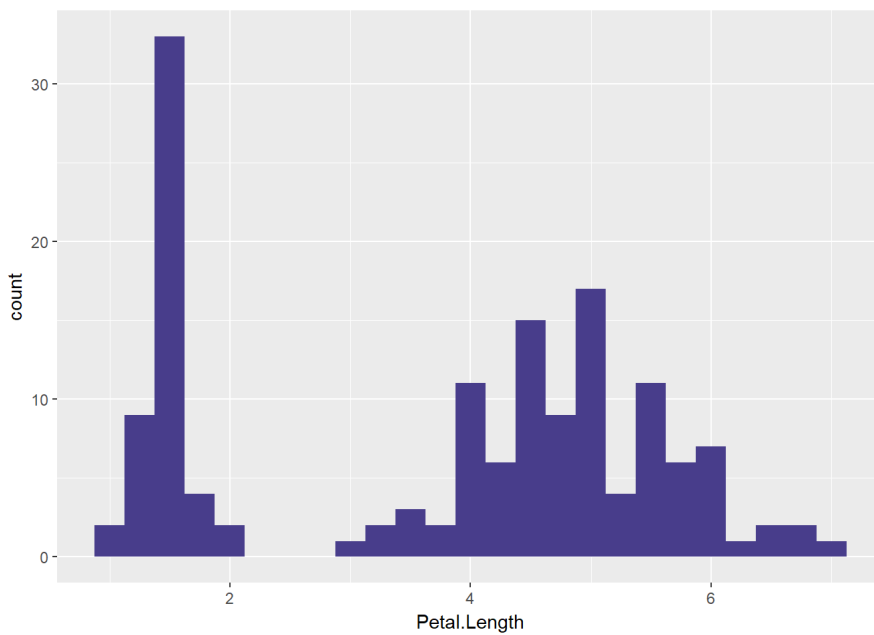
## 8. ggplot2 vs. ggvis

Using ggvis is very similar to using ggplot2. There are some notable distinctions, which I will summarize a few of them as follows.

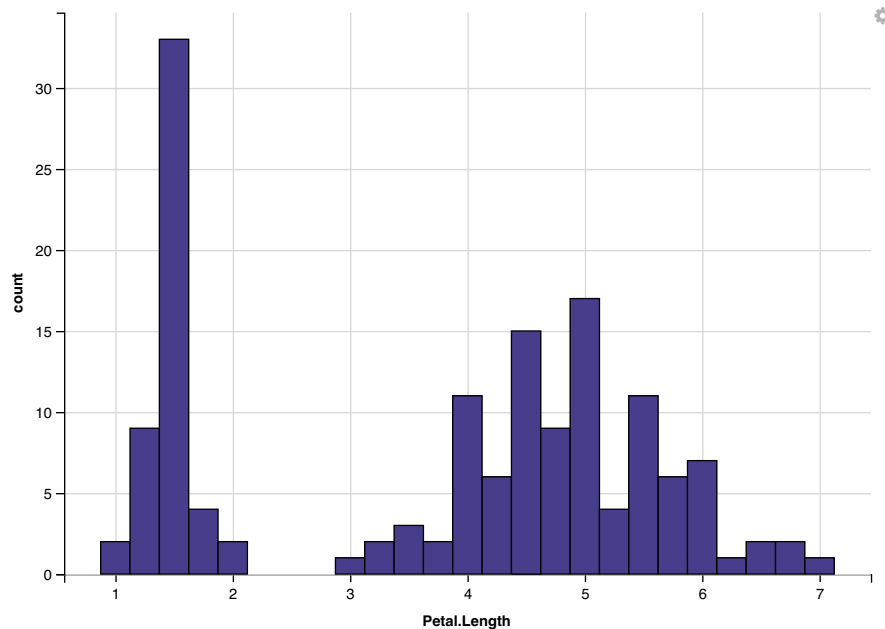
- Faceting is not supported in ggvis. Even when it is, it does not remember faceting in ggplot2.
- ggplot2 syntax makes a distinction between mapping variables and setting constants. In ggvis, everything is a property.
- ggvis has black outlines by default.
- It is not possible to incorrectly map an aesthetic to a scalar.

Let's compare ggplot with ggvis, by looking at a default histograms of petal lengths.

```
## ggplot2
ggplot(iris, aes(x=Petal.Length)) + geom_histogram(binwidth=.25, fill="darkslateblue")
```



```
##ggvis
iris %>% ggvis(~Petal.Length) %>% layer_histograms(width=.25, fill="darkslateblue")
```



Notice that the default ggvis graph is easier to read due to its white background and black outlines.

## 9. Conclusion

ggvis is a very useful package for visualizing data and can be used with shiny apps. It is a newer package, and as such it cannot do all the same functions that ggplot2 can do. Fortunately, the developers are continuing to bring ggvis up to the same level as ggplot2. This is especially nice as ggvis has several default parameters that make it more efficient. Overall, I like the look of ggvis better and I think the coding is easy to learn.

## 10. References

- <https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/iris.html>
- <https://ggvis.rstudio.com/>
- <https://cran.r-project.org/web/packages/ggvis/README.html>
- <http://ggvis.rstudio.com/ggplot2.html>
- <http://jimhester.github.io/ggplot2ToGgvis/>
- <https://ggvis.rstudio.com/ggvis-basics.html>
- <http://ggvis.rstudio.com/0.1/properties-scales.html>