

| ggvis Grammar | Properties for points | Transformations (cont) | Transformations (cont) | | | | | | | | | | |
|---|--|---|---|--|--------------------------------|--|--|---|----------------|---------------|---|---|---|
| <p>The 4 essential components are -</p> <ol style="list-style-type: none"> 1. Data 2. Coordinate system 3. Marks 4. Properties <p>Syntax Example-</p> <pre>faithful %>% ggvis(waiting,eruptions) %>% layer_points() %>% add_axis(" x", title = " Waiting period ", values = de = 9)</pre> | <p>The properties for points are -</p> <pre>fill,x, y, stroke, stroke Width,stroke Opacity</pre> <p>Sample code:</p> <pre>` faithful %>% ` ggvis(waiting, eruptions, ") %>% layer_points() %>% layer_points()</pre> | <p>It returns a dataset with 2 variables, one named <code>pred_</code> and the other <code>resp_</code></p> <pre>fillOpacity = ~exp2</pre> <p>Syntax: <code>compute_smooth()</code></p> <p>Long way: <code>mtcars %>% compute_bin(mpg, wt) %>% ggvis(pred_, ~resp) %>% layer_lines()</code></p> | <p>Similarly, we have <code>compute_count()</code> or the in-built function <code>layer_bars()</code></p> <p>Syntax: <code>compute_smooth()</code></p> <p>Long way: <code>faithful %>% compute_density(waiting) %>% ggvis(pred_, ~resp) %>% layer_lines()</code></p> <p>In-built: <code>faithful %>% ggvis(~waiting, fill := "green") %>% layer_densities()</code></p> | | | | | | | | | | |
| <p>We use the piping operator <code>'%>%'</code> for our syntaxes.</p> | <p>The properties for lines include -</p> <pre>x, y, fill, fillOpacity, idth</pre> | <p>It returns a dataset with 2 variables, one named <code>pred_</code> and the other <code>resp_</code></p> <pre>fillOpacity = ~exp2</pre> <p>Syntax: <code>compute_smooth()</code></p> <p>Long way: <code>mtcars %>% compute_bin(mpg, wt) %>% ggvis(pred_, ~resp) %>% layer_lines()</code></p> <p>In-built: <code>mtcars %>% ggvis(wt,mpg) %>% layer_smooths()</code></p> | <p>Similarly, we have <code>compute_count()</code> or the in-built function <code>layer_bars()</code></p> <p>Syntax: <code>compute_smooth()</code></p> <p>Long way: <code>faithful %>% compute_density(waiting) %>% ggvis(pred_, ~resp) %>% layer_lines()</code></p> <p>In-built: <code>faithful %>% ggvis(~waiting, fill := "green") %>% layer_densities()</code></p> | | | | | | | | | | |
| <p>Mapping Vs Setting propeties</p> <table border="1"> <thead> <tr> <th>Mapping</th> <th>Setting</th> </tr> </thead> <tbody> <tr> <td>= maps a data value</td> <td>:= sets property to a specific size/colour/width</td> </tr> <tr> <td>Used for visualization of data</td> <td>Used for customizing the appearance of plots</td> </tr> </tbody> </table> | Mapping | Setting | = maps a data value | := sets property to a specific size/colour/width | Used for visualization of data | Used for customizing the appearance of plots | <p>Properties for lines</p> <p>The properties for lines include -</p> <pre>x, y, fill, fillOpacity, idth</pre> | <p>Transformations</p> <table border="1"> <thead> <tr> <th>compute_smooth</th> <th>compute_bin()</th> </tr> </thead> <tbody> <tr> <td>It transforms the data to generate a new dataframe.</td> <td>It transforms the data to generate a new dataframe.</td> </tr> </tbody> </table> | compute_smooth | compute_bin() | It transforms the data to generate a new dataframe. | It transforms the data to generate a new dataframe. | <p>Transformations</p> <p>compute_density()</p> <p>A density plot uses a line to display the density of a variable at each point in its range.</p> <p>It returns a data frame with two columns: <code>pred_</code>, the x values of the variable's density line, and <code>resp_</code>, the y values of the variable's density line.</p> |
| Mapping | Setting | | | | | | | | | | | | |
| = maps a data value | := sets property to a specific size/colour/width | | | | | | | | | | | | |
| Used for visualization of data | Used for customizing the appearance of plots | | | | | | | | | | | | |
| compute_smooth | compute_bin() | | | | | | | | | | | | |
| It transforms the data to generate a new dataframe. | It transforms the data to generate a new dataframe. | | | | | | | | | | | | |
| <p>ggvis scales the values to a pre-defined scale of colour/sizes</p> <p>ggvis sends the colour value to <i>vega</i> - a javascript package for further processing</p> | | | | | | | | | | | | | |



By shanly3011

Published 9th April, 2015.

Last updated 12th May, 2016.

Page 1 of 1.

Sponsored by [Readable.com](http://readable.com)

Measure your website readability!

<https://readable.com>