Cheatography

Data Visualization in R: ggvis Cheat Sheet

by shanly3011 via cheatography.com/20988/cs/3865/

ggvis Grammar

The 4 essential components are -

1. Data
2. Coordinate system
3. Marks
4. Properties

Syntax Example-

```
faithful %>%
ggvis( ~wa iti ng, ~er -
upt ions) %>% layer_ -
poi nts() %>%
add_ax is( " x", title =
" Waiting period ",
values = c(1,2, -
3,4 ,5, 6,7), subdivide
= 9)
```

We use the piping operator '%>%' for our syntaxes.

Mapping Vs Setting propeties

Mapping	Setting
= maps property to a data value	:= sets property to a specific size/coluor/width
Used for visualization of data	Used for custom-izing the appearance of plots
ggvis scales the values to a pre-de-fined scale of colour/sizes	ggvis sends the colour value to *vega* - a javascript package for further processing

Properties for points

The properties for points are - `fill,x, y, stroke, stroke Width, stroke - Opa city, fill, opacity, fillOp acity, shape, size`

Sample code:

```
faithful %>% `
ggvis( ~wa iting,
~erupt ions, fillOp -
acity = ~erupt ions,
size := 100, fill := " -
red ", stroke := " -
red ", shape := " cro -
ss") %>%
layer_ poi nts()
```

Properties for lines

The properties for lines include - `x, y, fill, fillOp - acity, opacity, stroke, stroke Dash, stroke - Opa city, and stroke - Width`

Transformations

compute_s-mooth	compute_bin()
It transforms the data to generate a new dataframe.	It transforms the data to generate a new dataframe.

Transformations (cont)

It returns a dataset with 2 variables, one named pred_ and the other resp_	It returns a dataset with 4 variables, x, x2 ,y ,y2.

Syntax: compute_smooth()

Long way:

```
mtcars %>%
comput e_s moo th(mpg ~
wt) %>% ggvis( ~pr -
ed_ ,~r esp_) %>%
layer_ lines()
```

In-built:

```
mtcars %>%
ggvis( ~wt ,~mpg) %>%
layer_ smo oths()
```

Syntax: compute_bin()

Long way:

```
faithful %>%
comput e_b in( ~wa -
iting, width = 5) %>%
ggvis(x = ~ xmin_, x2 =
~ xmax_, y = 0, y2 =
~count_) %>% layer_ -
rects()
```

In-built:

```
faithful %>%
ggvis( ~wa iting) %>%
layer_ his tog ram -
s(width = 5)
```

Transformations

compute_density()

A density plot uses a line to display the density of a variable at each point in its range.

It returns a data frame with two columns: pred_, the x values of the variable's density line, and resp_, the y values of the variable's density line.

Transformations (cont)

Similarly, we have compute_c-ount() or the in- built function layer_bars()

Syntax: compute_smooth()

Long way:

```
faithful %>%
comput e_d ens ity (~w -
aiting) %>% ggvis( -
~pred_, ~resp_) %>%
layer_ lines()
```

In-built:

```
faithful %>%
ggvis( ~wa iting, fill
:= " gre en") %>%
layer_ den sit ies()
```