

Lecture 20: Interactive plots with ggvis

STAT598z: Intro. to computing for statistics

Vinayak Rao

Department of Statistics, Purdue University

ggvis is a simple way to get interactive plots

- provides a simpler interface to shiny
- is still experimental

Like ggplot this expects a dataframe/tibble as an input

Some differences:

- add layers using `%>%` instead of `+`
- instead of `aes (color=group)`, write `color = ~group`
- we still write `color=clr_val`
- aesthetics have different names:
 - `color` becomes `stroke`
 - `alpha` becomes `opacity`

```
In [ ]: library('tidyverse')  
library('ggvis')  
load('~/.RSRCH/DATA/EconData/data/HomeValues.RData')  
HomeValues$qtr <- as.double(HomeValues$qtr)
```

```
In [ ]: plt<- ggvis(HomeValues,x=~qtr,y=~Home.Value,stroke=~State) %>%  
  layer_lines()
```

```
In [ ]: plt <- plt %>% hide_legend('stroke')
```

```
In [ ]: plt %>% layer_points()
```

```
In [ ]: plt %>% layer_points(size=1, fillOpacity=.1) # Bug!
```

ggvis uses both = and := for assignments

Use = to map a variable to a property

- Then use ~ to refer to a column of a dataframe

Use := when we set a property based on a *value*

```
In [ ]: plt %>% layer_points(size:=1, fillOpacity:=.1)
```

In the end, set properties using =~ column or := value

So why use ggvis instead of ggplot?

- Interactive plots!

```
In [ ]: plt %>% layer_points(size:=input_slider(.1,50,1),  
                             fill =~ State, fillOpacity:=.5) %>%  
         hide_legend('fill')
```

```
In [ ]: plt %>% add_tooltip(function(x)
                        {paste(x$State, ":", x$Home.Value)}, 'hover')
```

`add_tooltip` needs a function to read value and return a string

- we used an *anonymous function* to print State, Value

For lines, `add_tooltip` only prints first value

([http://stackoverflow.com/questions/28540504/mouse-hover-](http://stackoverflow.com/questions/28540504/mouse-hover-in-layer-lines-ggvis-r)

[in-layer-lines-ggvis-r](http://stackoverflow.com/questions/28540504/mouse-hover-in-layer-lines-ggvis-r) (<http://stackoverflow.com/questions/28540504/mouse-hover-in-layer-lines-ggvis-r>))

- `add_layer_points()` for all values

```
In [ ]: plt %>% layer_points(size:=input_slider(0,5))%>%
        hide_legend('stroke')
```

```
In [ ]: plt %>% add_tooltip(function(x)
                        {paste(x$State, ":", x$Home.Value)}, 'hover')
```

```
In [ ]: plt <- ggvis(HomeValues x=~qtr,y=~Home.Value,stroke=~State) %>%  
  layer_smooths(span:=input_slider(0,5)) %>%  
  hide_legend('stroke')
```

Error because ggvis doesn't do grouping for you (unlike ggplot)

```
In [ ]: plt <- HomeValues %>% group_by(State) %>%  
  ggvis(x=~qtr,y=~Home.Value, stroke=~State) %>%  
  layer_smooths(span=input_slider(0,2,step=.1)) %>%  
  hide_legend('stroke')
```



```
ggvis(HomeValues) %>% layer_histograms(x=~Home.Value)
```

```
ggvis(HomeValues) %>%  
  layer_histograms(x=~Home.Value,  
                   width=input_slider(min=1000,max=100000))
```

```
ggvis(HomeValues) %>%
  layer_histograms(x=~Home.Value, fill.hover:='red',
    width=input_slider(min=10^3,max=10^5))
```

```
In [ ]: plt<- ggvis(HomeValues,x=~qtr,y=~Home.Value,stroke=~State) %>%  
          layer_lines(stroke.hover:='black') %>% hide_legend('stroke')  
plt %>% add_tooltip(function(x)  
                  {paste(x$State,":",x$Home.Value)}, 'hover')
```

```
In [ ]: plt <- HomeValues %>% group_by(State) %>%  
  ggvis(x=~qtr,y=~Home.Value, stroke=~State) %>%  
  filter(State %in% eval(input_select(choices =  
    unique(as.character(HomeValues$State))),  
    multiple=TRUE, label='States list')) %>%  
  layer_lines(strokeWidth:=2)
```

Note the eval, this is because of we are calling input_select inside filter

```
In [ ]: #https://r2014-mtp.sciencesconf.org/file/92631
dat <- data.frame(time=1:10, value=runif(10))

# Create a reactive that returns a data frame, adding a new
# row every 2 seconds
ddat <- reactive({
  invalidateLater(2000, NULL)
  dat$time <- c(dat$time[-2], dat$time[length(dat$time)] + 1)
  dat$value <- c(dat$value[-1], runif(1))
  dat
})

ddat %>% ggvis(x = ~time, y = ~value, key := ~time) %>%
  layer_points() %>% layer_paths()
```

```
In [ ]: library('ggplot2');library('maps')

my_state_map <- map_data('state');
my_state_map$region <- tolower(my_state_map$region)
get_ab <- function(x)  state.abb[x ==tolower(state.name)]

get_house_pr <- function(st,yr) {
  HomeValues[HomeValues$State==st &
             HomeValues$qtr==yr,2] }

state.name[51] <- "district of columbia"
state.abb[51] <- "DC"

my_state_map$region <- purrr::map_chr(my_state_map$region,
                                     get_ab)

get_yr_pr <- function(yr) {
  pr <- my_state_map$pr
  for(st in state.abb) {
    psn <- my_state_map$region == st
    pr[psn] <- get_house_pr(st,floor(yr))
  }
  pr; }
```

```
In [ ]: yr <- 1976
        stmp <- reactive({invalidateLater(2000,NULL)
          my_state_map$pr <- get_yr_pr(yr)
          yr <-< yr + 4; if(yr>=2013) yr <-< 1976
          print(yr)
          my_state_map })

        stmp %>% ggvis(~long, ~lat,fill=~pr) %>%
          group_by(region) %>%
          layer_paths(strokeOpacity := 0.5,
                     strokeWidth := 0.5) %>%
          hide_axis("x") %>% hide_axis("y") %>%
          set_options(width=960, height=600, keep_aspect=TRUE) %>%
            hide_legend('fill') %>%
            add_tooltip(function(x) {
              paste(x$region)},'hover')
```