

# GGVIS FOR R TUTORIAL

By Leslie McIntosh & Connie Zabarovskaya

# About ggvis

- This ggvis tutorial is based on the notes on the following link, written by Hadley Wickham: <http://ggvis.rstudio.com/ggvis-basics.html>
- GGVIS allows for interactive graphics in R, which makes exploration of data much easier
- Plot data in just a few lines of code
- Outside of RStudio, graphics will be rendered in a web browser.

# Installation

- Download and Install R:

<http://cran.wustl.edu/>

- Install RStudio

<https://www.rstudio.com/ide/>

- Install ggvis: ggvis is not yet (24 May 2014) available on CRAN. So it needs to be installed from GitHub using the following code:

```
> install.packages("devtools")
```

```
> devtools::install_github(c("rstudio/shiny", "rstudio/ggvis"),  
                           build_vignettes = FALSE)
```

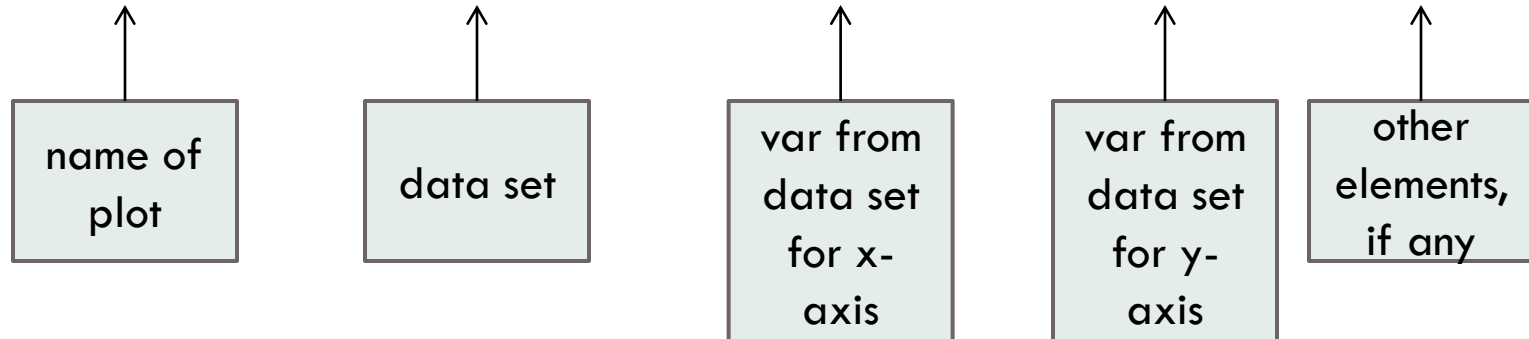
- load ggvis library:

```
> library("ggvis")
```

# Simple ggvis() Call

- call to ggvis

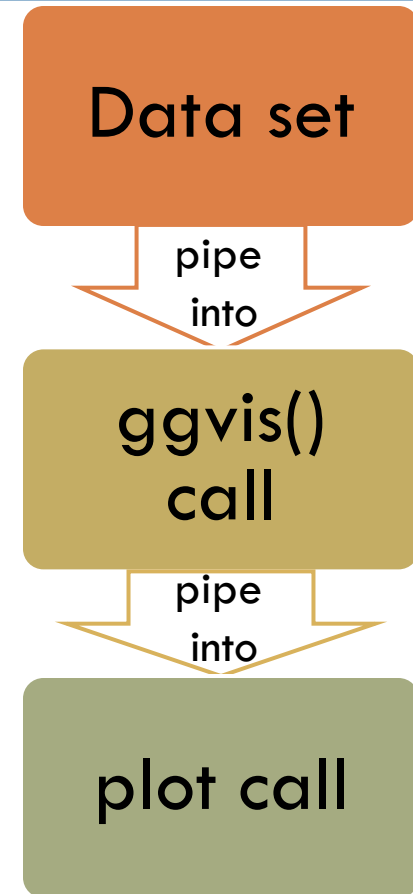
□ > **p** <- ggvis(mtcars, x = ~wt, y = ~mpg, ...)



- to actually plot (display) the data
- > `layer_points(p)`

# ggvis() Call Using 'Pipe'

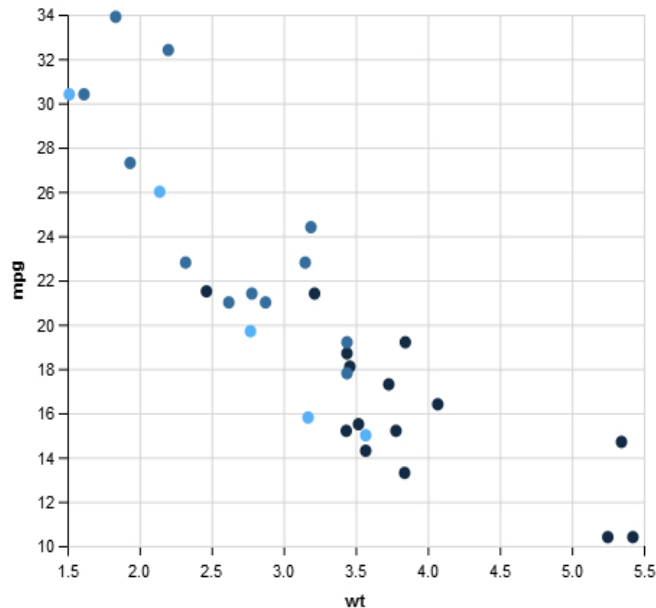
- use 'pipe'\* - `%>%` - to avoid nested functions and “temporary” variables.
- ```
> mtcars %>%  
  ggvis(x = ~wt, y = ~mpg) %>%  
  layer_points()  
OR  
> mtcars %>%  
  ggvis(~wt, ~mpg) %>%  
  layer_points()
```
- `%>%` takes value on the left-hand side and passes it to function or expression on the right-hand side
- \*Pipe comes from the magrittr package  
(<http://cran.r-project.org/web/packages/magrittr/magrittr.pdf>)



# ggvis() Call Using 'Pipe' (Cont'd)

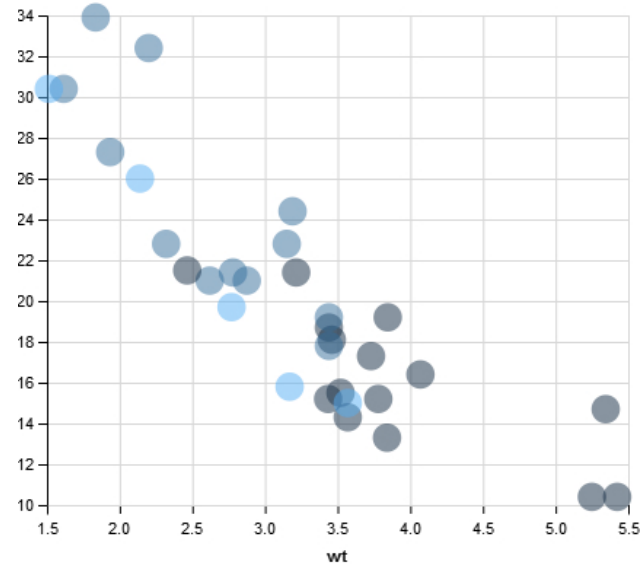
- Using `%>%` ('pipe') also allows you to seamlessly intermingle ggvis and dplyr code
- `> library(dplyr)`
- `> mtcars %>%`  
    `ggvis(x = ~mpg, y = ~disp) %>%`  
    **`mutate(displacement = displacement / 61.0237)`** `%>%`  
    `# convert engine displacement to liters`  
    `layer_points()`

# Visual Properties & Adding More Variables



Other variables can be added using 'fill', 'stroke', 'size', and 'shape'

```
> mtcars %>%  
  ggvis(x=~wt, y=~mpg, fill=  
    ~gear) %>%  
  layer_points()
```



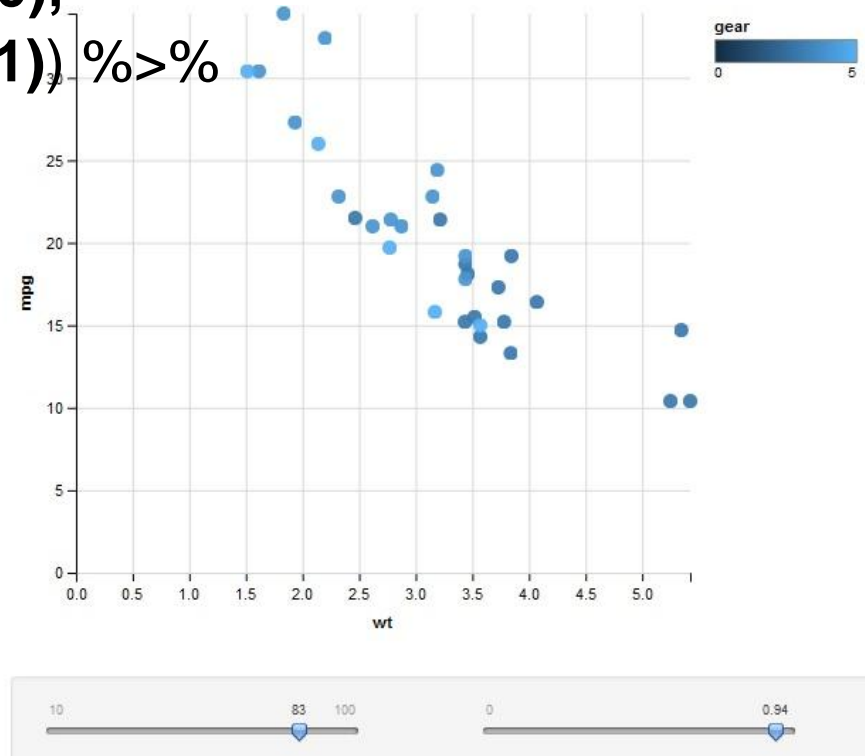
To fix elements (e.g., color, opacity, size, shape, stroke) use ':= ' instead of '='

```
> mtcars %>%  
  ggvis(x=~wt, y=~mpg, fill= ~gear,  
    opacity := 0.5, size := 300) %>%  
  layer_points()
```

# Interaction with ggvis(). Example 1

- To make graphs interactive - connect variables to interactive controls (sliders, drop-down menus, text-boxes, radio buttons, etc.)
- `> mtcars %>%`

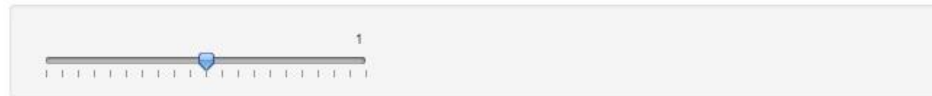
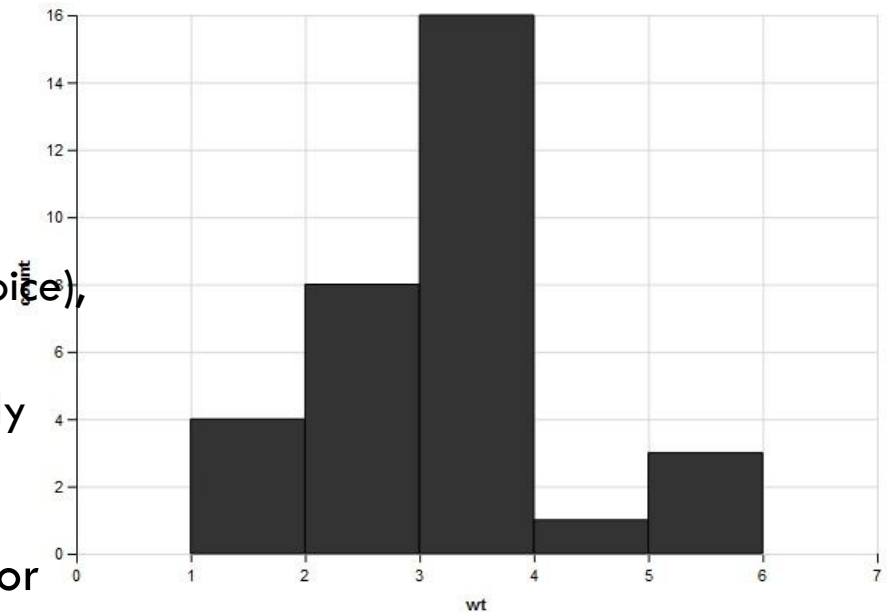
```
ggvis(~wt, ~mpg, fill= ~gear,  
size := input_slider(10, 100),  
opacity := input_slider(0, 1)) %>%  
layer_points()
```





# Interaction with ggvis(). Example 2

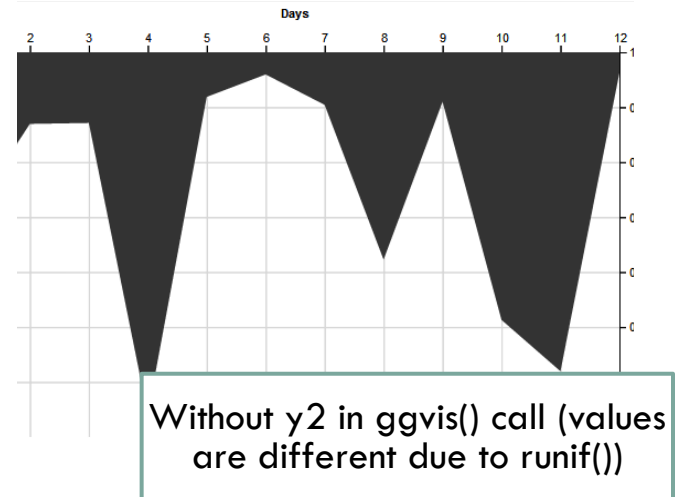
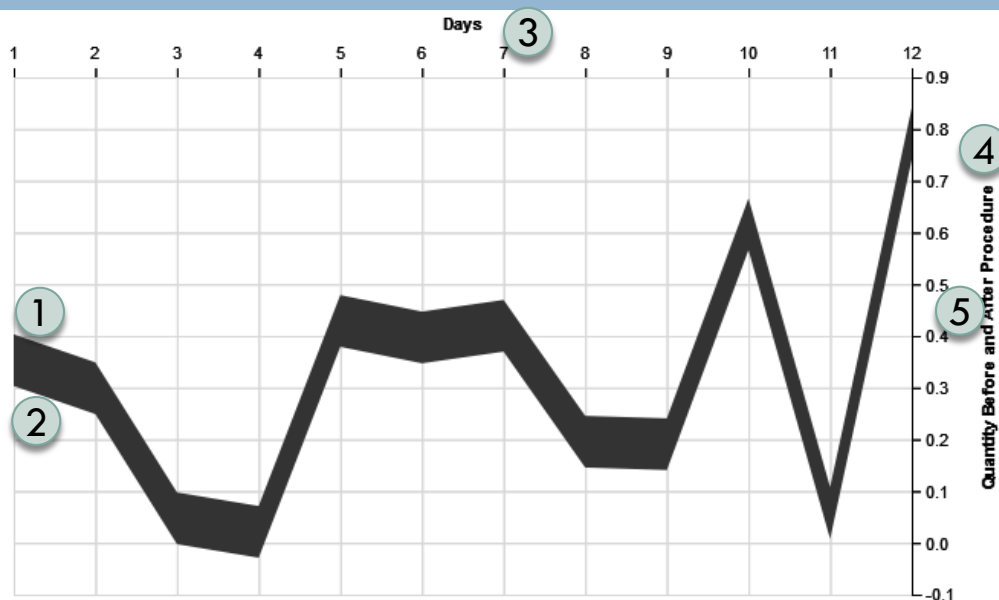
- Interactive Controls available:
  - `input_slider()`,
  - `input_select()` (dropdown menu)
  - `input_checkbox()`,
  - `input_checkboxgroup()` (multiple choice),
  - `input_radiobuttons()`,
  - `input_text()` and `input_numeric()` (only numerical input),
- Use “=” when assigning interactive controls to a var, “:=” is used only for static values
- Example with a Histogram
- ```
> mtcars %>%  
  ggvis(~wt) %>%  
  layer_histogram(binwidth =  
    input_slider(0, 2, step = 0.1))
```



# Layer Options in ggvis()

- Other layer options, besides `layer_points()` and `layer_histograms()`:
  - ▣ `layer_bars()`
  - ▣ `layer_ribbons()` (filled space between two paths or a path and an axis)
  - ▣ `layer_paths()` (all points connected with a line)
  - ▣ `layer_lines()` (equivalent to `arrange(x) %>% layer_paths()`)
  - ▣ `layer_smooths()` (displays predictions with a line)
  - ▣ `layer_rects()` (rectangles)
  - ▣ `layer_text()` (displays text on the chart)
- To display multiple layers on one chart – pipe (`%>%`) them into each other, but include individual parameters (`fill`, `size`, `span`, etc.) inside their parentheses, instead of `ggvis()` (see slide 12)

# Layer\_ribbons() with Axis Formatting



```
> df <- data.frame(x = 1:12, y = runif(12))  
df %>% ggvis(x = ~x, y = ~y, y2 = ~y - 0.1) %>% layer_ribbons() %>%  
  add_guide_axis("x", title = "Days", orient = "top") %>%  
  add_guide_axis("y", title = "Quantity Before and After Procedure", orient = "right",  
    title_offset = 50)
```

# Multiple Layers & Interaction

□ > mtcars %>%

ggvis(~wt, ~mpg) %>%

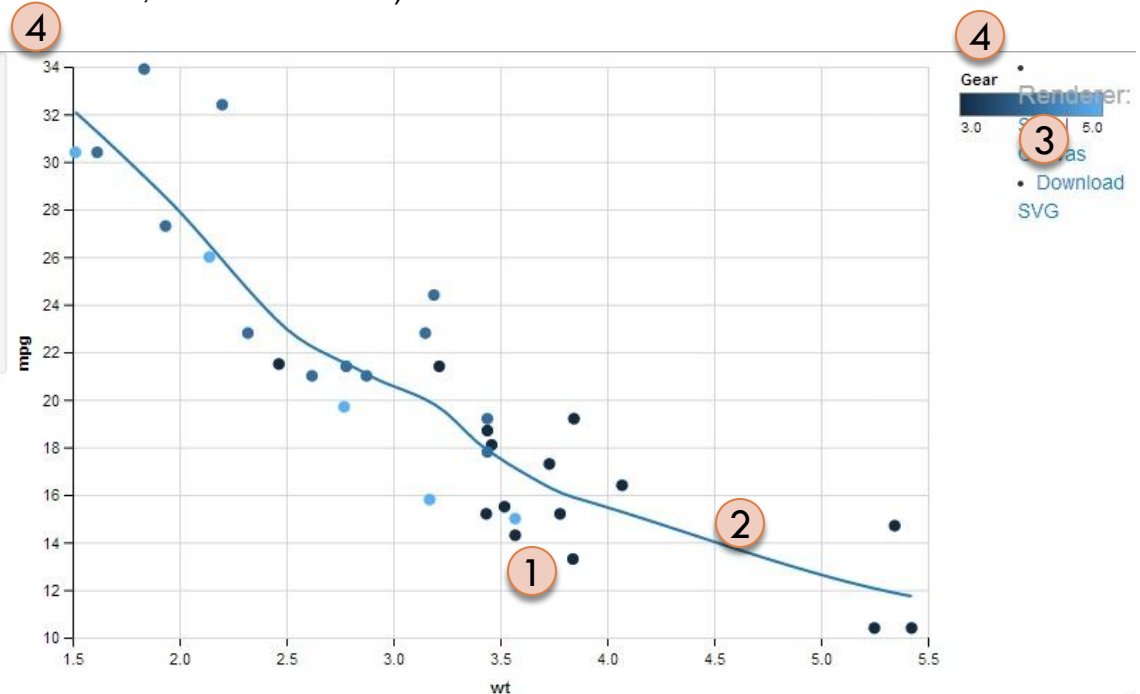
layer\_points(fill = ~gear, size := input\_slider(10, 100, label = "Point Size")) %>%

layer\_smooths(stroke = input\_select(choices = c("Blue" = "blue", "Red" = "red", "Purple" = "purple"), selected = "Blue", label = "Line Color"), span = input\_slider(0.2, 1, value = 0.75, label = "Line Span")) %>%

add\_guide\_legend( fill = "fill", title = "Gear", stroke = "stroke")



NOTE: 'fill = ~gear' could be included into the ggvis() call, instead of layer call



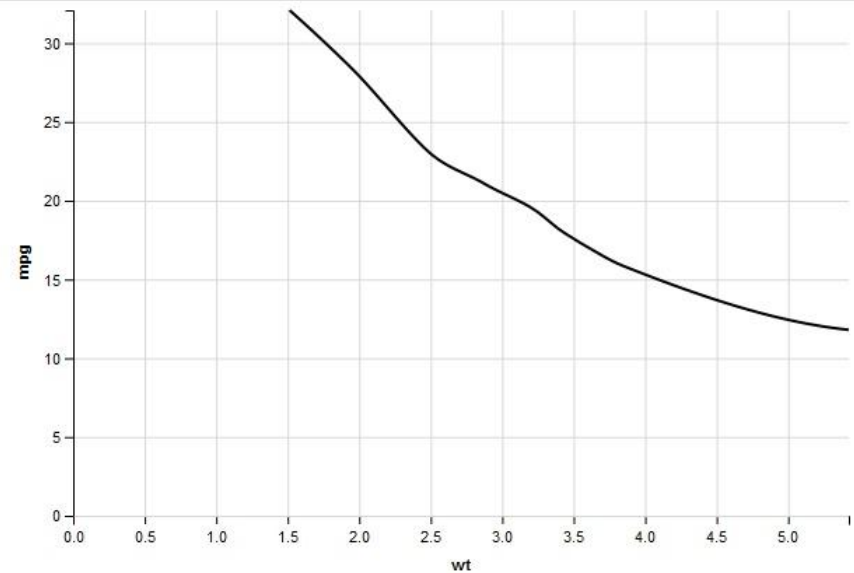
# Interaction with ggvis(). Example 3

- Example with Prediction Model and radio buttons

```
> mtcars %>%
```

```
  ggvis(~wt, ~mpg) %>%
```

```
  layer_model_predictions(model = input_radiobuttons(
    choices = c("Linear" = "lm", "LOESS" = "loess"),
    selected = "loess",
    label = "Model type"))
```



Model type

☐ Linear

☒ LOESS

# Grouping Variables and Smoothers

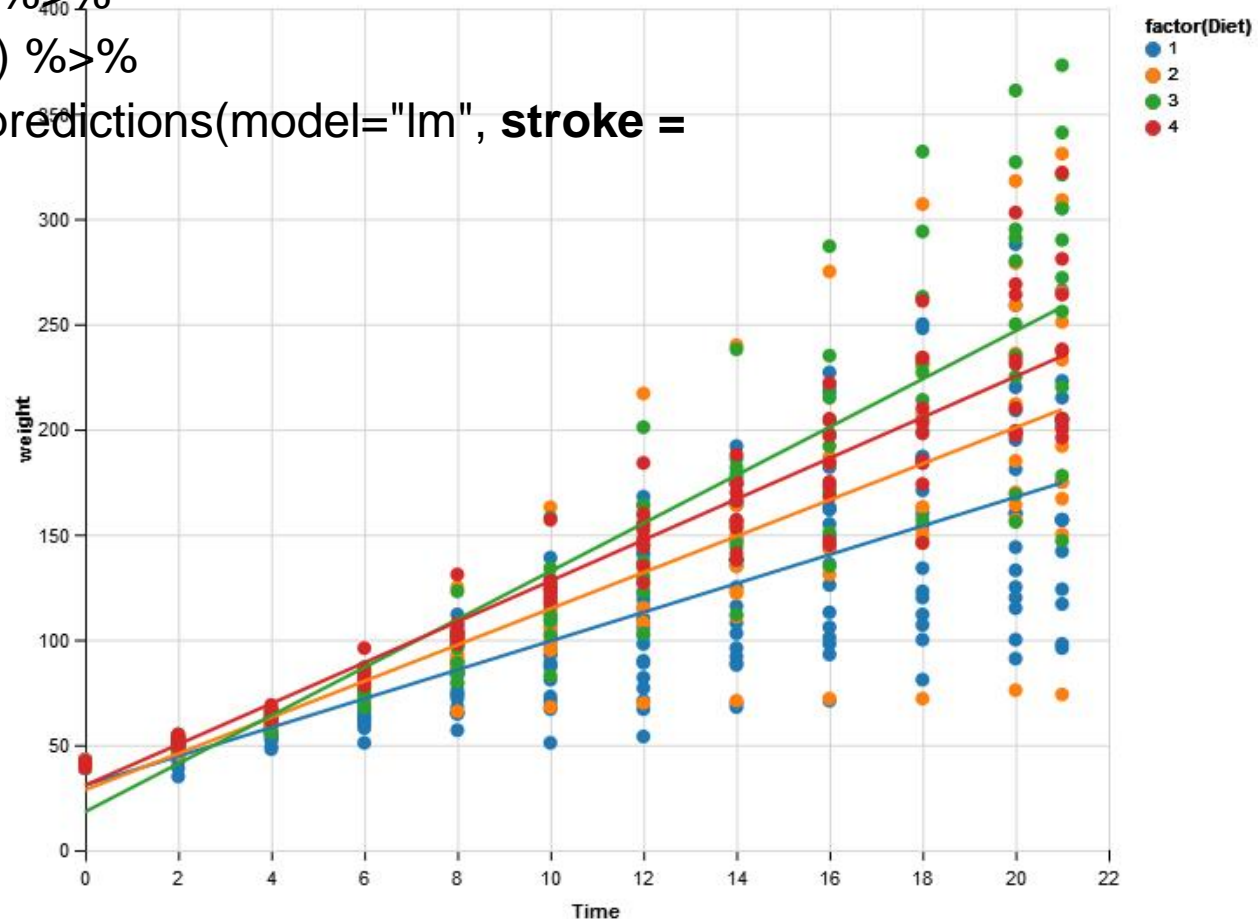
□ > ChickWeight %>%

```
ggvis(x = ~Time, y = ~weight, fill = ~factor(Diet)) %>%
```

```
layer_points() %>%
```

```
group_by(Diet) %>%
```

```
layer_model_predictions(model="lm", stroke =  
~factor(Diet))
```



# More Resources on ggvis()

---

- <http://ggvis.rstudio.com/>
- <http://ggvis.rstudio.com/cookbook.html>
- <http://ggvis.rstudio.com/interactivity.html>



Thank you!  
Questions?