ggvis 0.4 overview

介绍

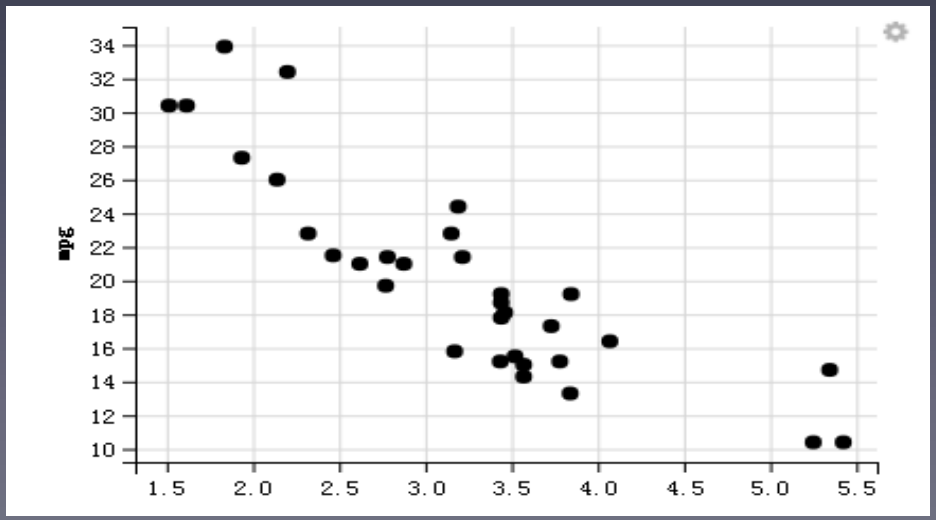
- 介绍
- ggvis的目标是在探索性数据分析时使建立交互式图形变的更容易。ggvis具有ggplot2相似的基本理论（图形的语法），但是在表达上有些不同，并增加了新的功能能够让你的图片互动。ggvis还采用了 shiny的反应式编程模型和dplyr的数据转换语法
- **特点**
- 交互是ggvis包最大的特点，第二大特点是利用%>%可以无缝衔接dplyr包

介绍顺序

- 介绍顺序
- 用ggvis（）进行绘图。
- 交互（Interaction）
- 简单层。
- 建立复杂的多层图形
- **显示方式**
- 交互式图形可以直接显示在RStudio上或在浏览器上。利用 shiny包的基础功能在任何浏览器上显示出来。

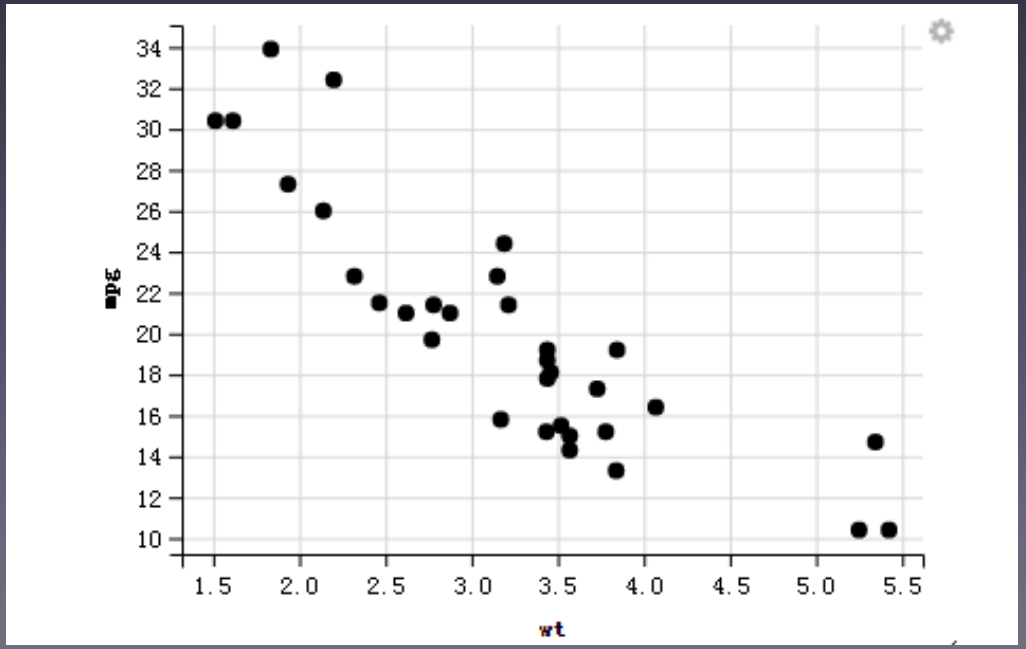
基础介绍

- 第一个参数是要绘制的数据集，其他参数描述如何将变量映射到可视化属性。
- 1 分开写 p <- ggvis(mtcars, x = ~wt, y = ~mpg)
  layer_points(p)
- 2 结合到一起写 layer_points(ggvis(mtcars, x = ~wt, y = ~mpg))



%>%功能

- 1通过使用magrittr包的 %>%功能。允许重写前一个函数调用
- mtcars %>% ggvis(x = ~wt, y = ~mpg) %>% layer_points()



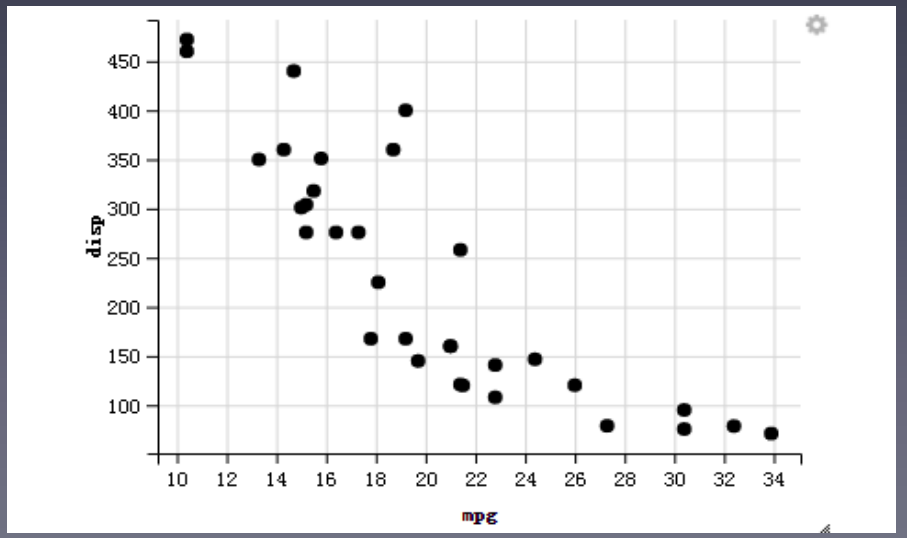
函数引用

2用下面library(dplyr)，让ggvis和dplyr结合

library(dplyr)

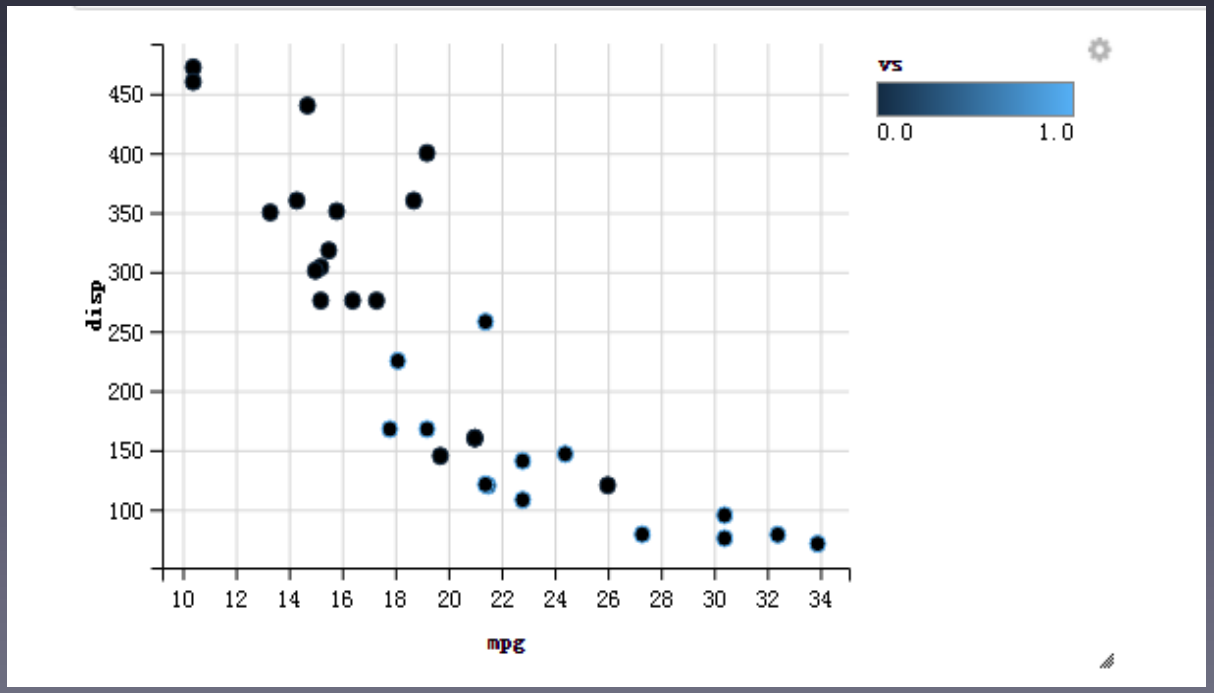
3 ggvis()两参数通常为第一个x和第二个y的~加变量表示替代位置

```
mtcars %>%
 ggvis(~mpg, ~disp) %>%
 layer_points()
```



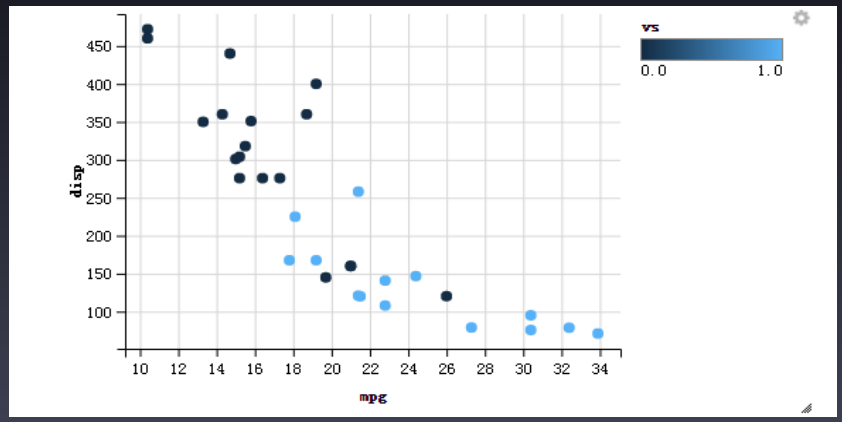
描边 stroke

- 可以通过向图中添加更多变量将它们映射到其他视觉属性（如填充、描边、大小和形状）。
- mtcars %>% ggvis(~mpg, ~disp, stroke = ~vs) %>% layer_points()
- 描边 stroke

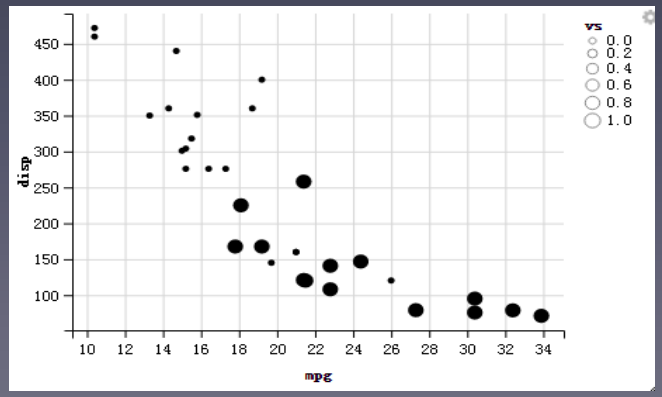


fill 和 size

- mtcars %>% ggvis(~mpg, ~disp, **fill** = ~vs) %>% layer_points()

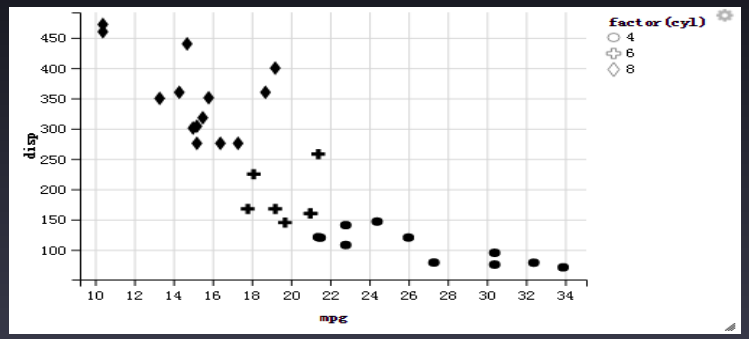


- mtcars %>% ggvis(~mpg, ~disp, size = ~vs) %>% layer_points()

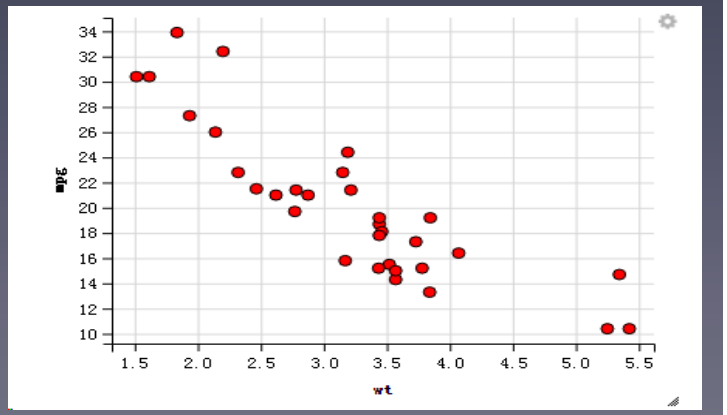


factor(cyl)和 :=

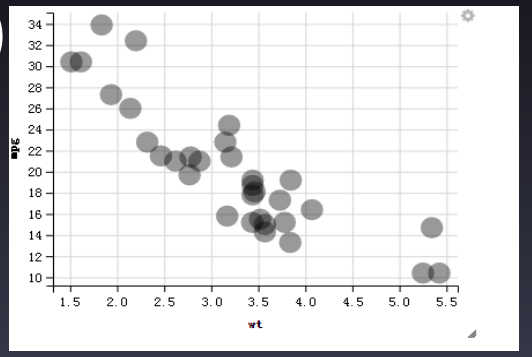
- mtcars %>% ggvis(~mpg, ~disp, **shape = ~factor(cyl))** %>% layer_points()



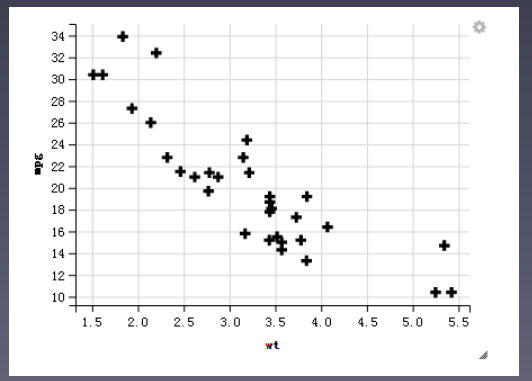
- mtcars %>% ggvis(~wt, ~mpg, **fill := "red", stroke := "black"**) %>% layer_points()
- 用:= 代替= 固定用什么颜色



- mtcars %>% ggvis(~wt, ~mpg, size := 300, opacity := 0.4) %>% layer_points()

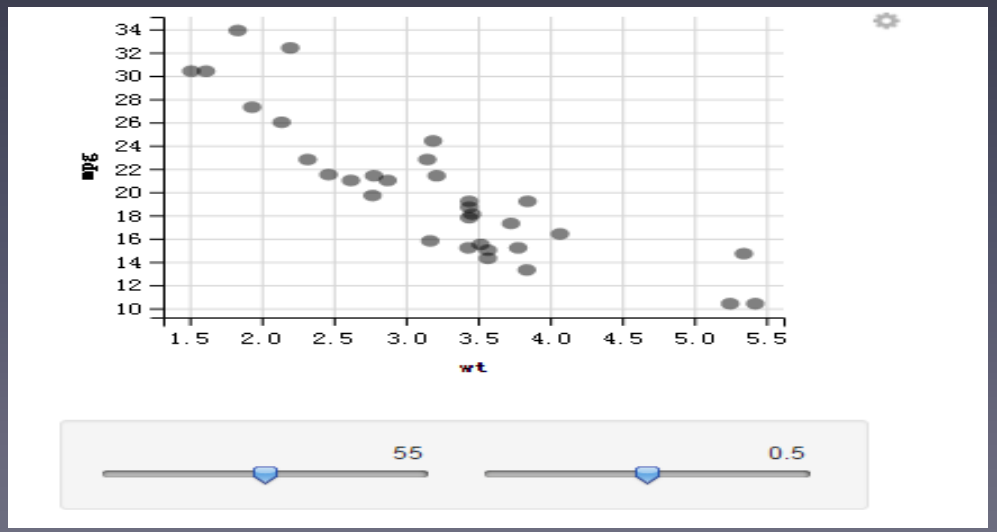


- mtcars %>% ggvis(~wt, ~mpg, shape := "cross") %>% layer_points()



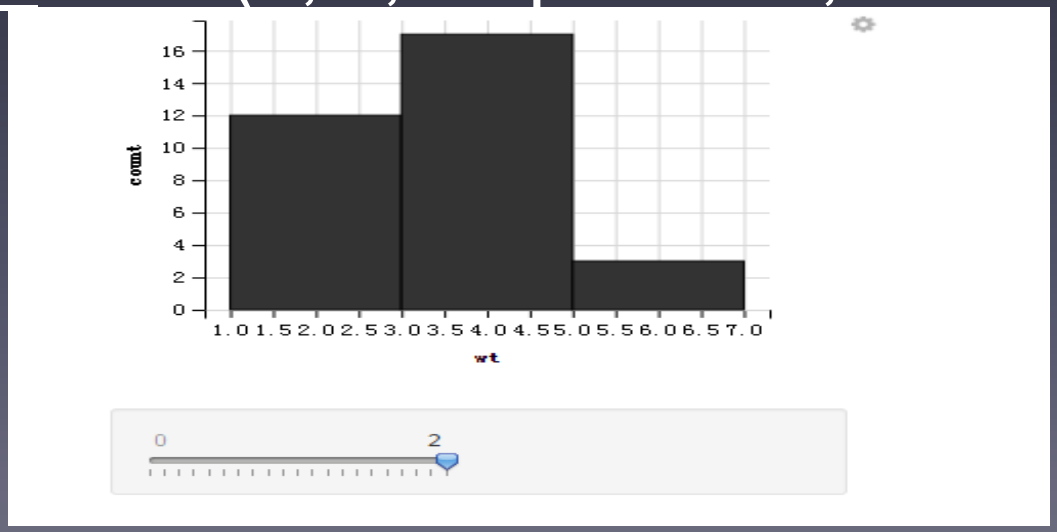
交互（Interaction）

- 散点
- mtcars %>%
- ggvis(~wt, ~mpg,
- size := input_slider(10, 100),
- opacity := input_slider(0, 1)
-) %>%
- layer_points()



直方图

- mtcars %>%
- ggvis(~wt) %>%
- layer_histograms(width = input_slider(0, 2, step = 0.10, label = "width"),
- center = input_slider(0, 2, step = 0.05, label = "center"))

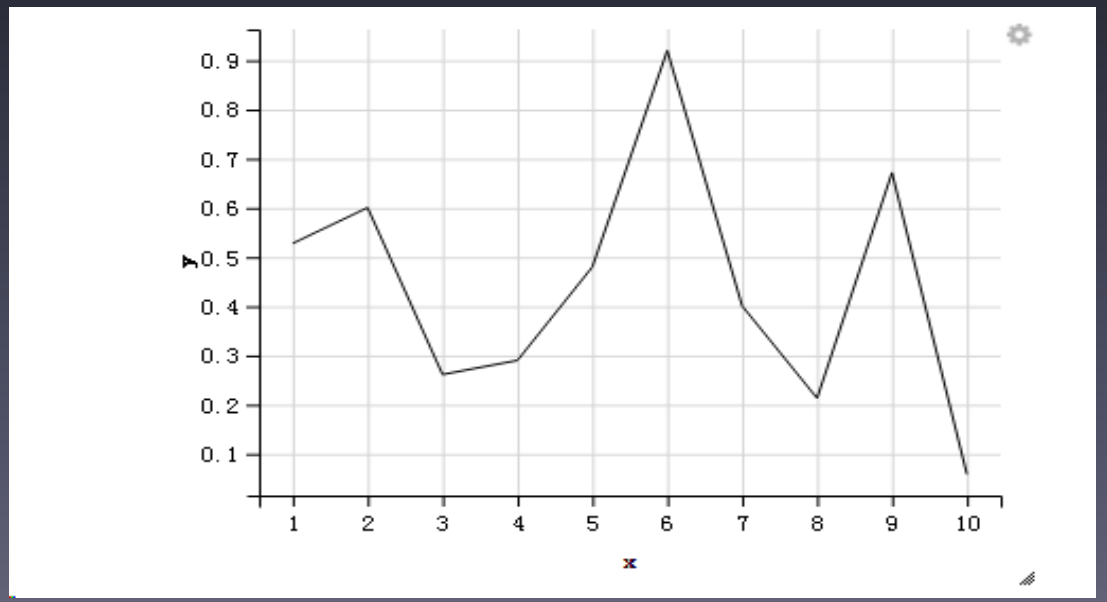


简单层

- 到目前为止，你见过两层功能：layer_points()和layer_histograms()。还有许多其他的层，他们大致可以分为两种类型。
- 简单层：包括基本的点，线和矩形。
- 复杂层：将数据转换与一个或多个简单层相结合的。所有层函数使用复数，而不是单数。

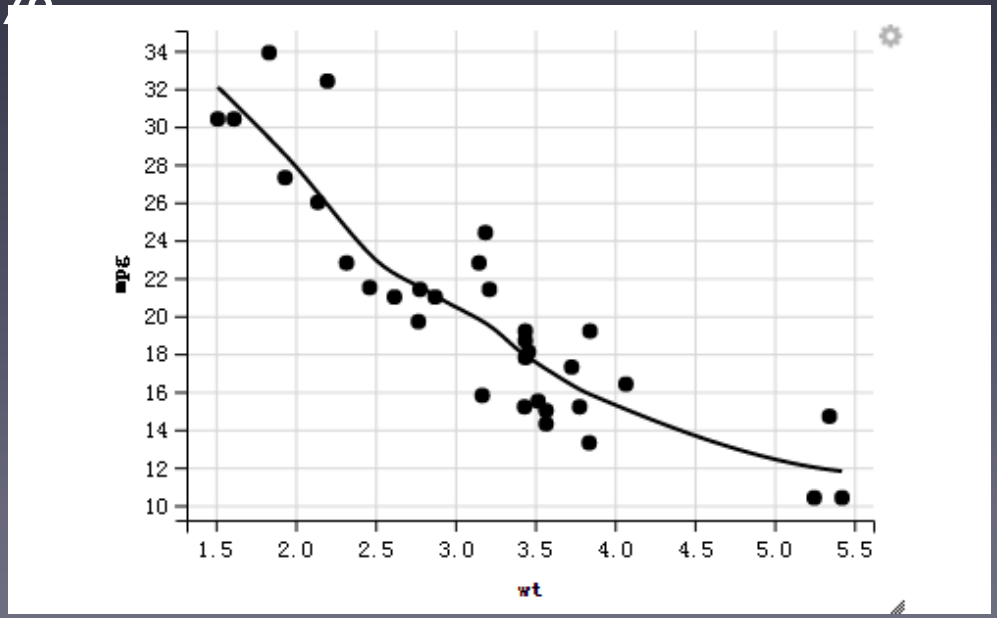
折线图

- layer_paths()
- df <- data.frame(x = 1:10, y = runif(10))
- df %>% ggvis(~x, ~y) %>% layer_paths()

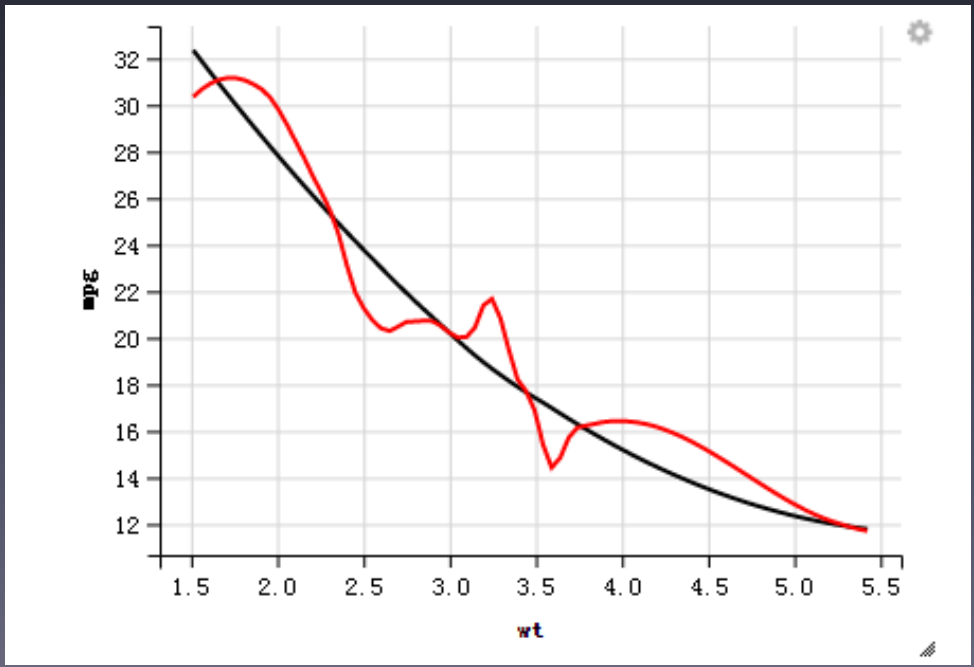


多层次

- 丰富的图形可以创建在同一地块上结合多个层。这更容易做到：只是对多个元素层。
- mtcars %>%
- ggvis(~wt, ~mpg) %>%
- layer_smooths() %>%
- layer_points()



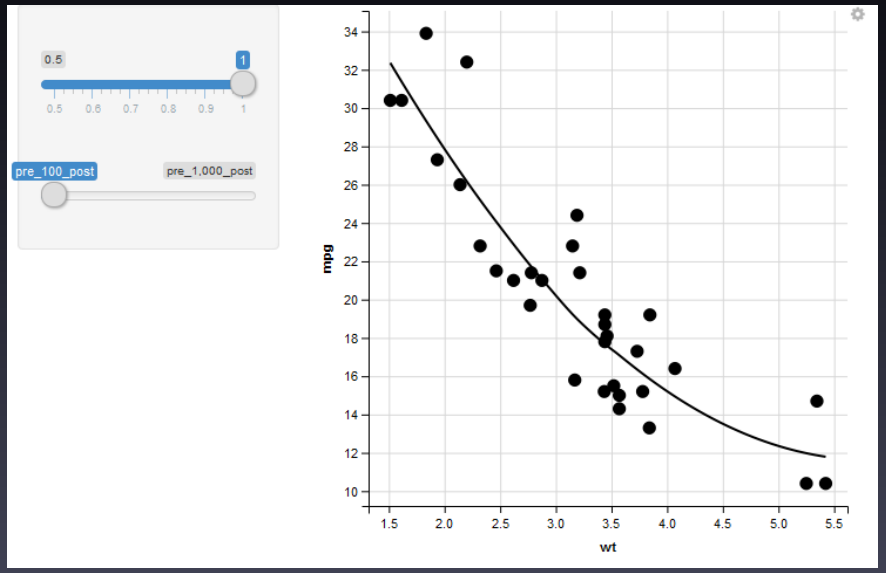
- mtcars %>% ggvis(~wt, ~mpg) %>%
- layer_smooths(span = 1) %>%
- layer_smooths(span = 0.3, stroke := "red")



作业

给出前面代码

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



谢谢