

Adding layers

INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R



Adam Loy

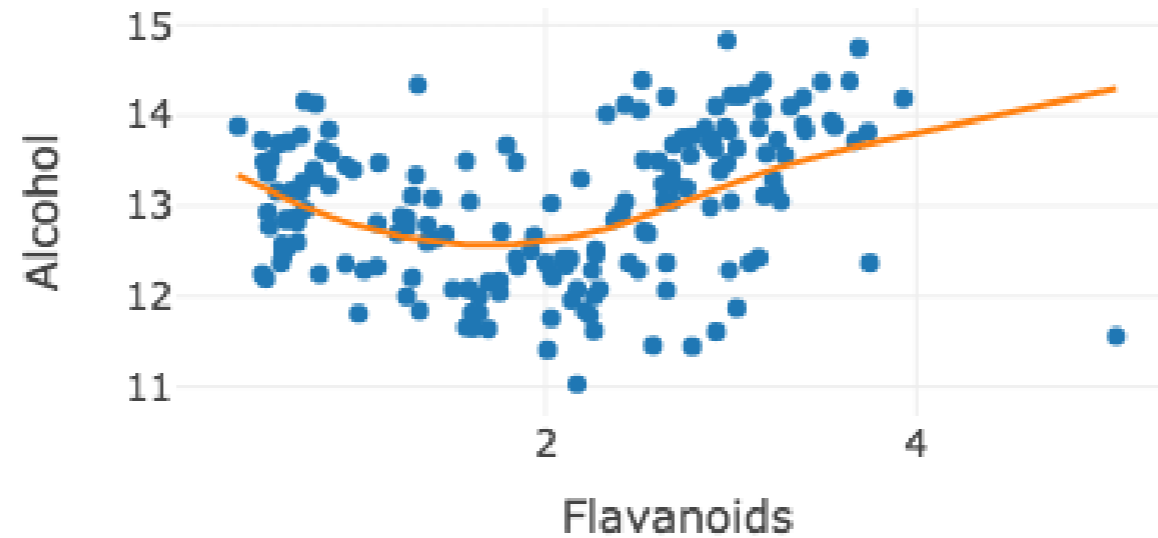
Statistician, Carleton College

Wine data

```
glimpse(wine)
```

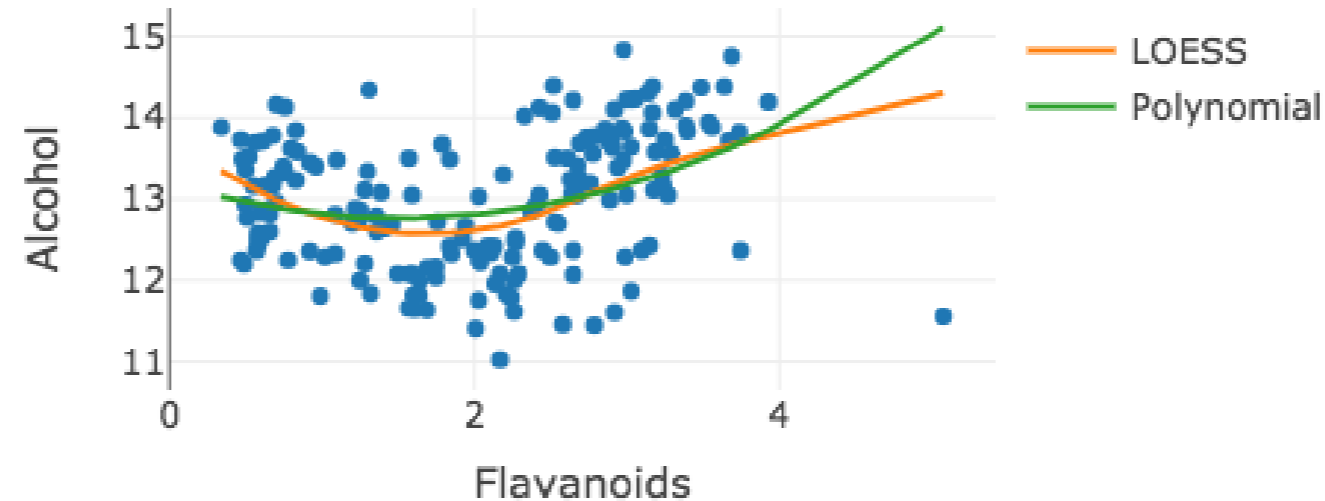
```
Rows: 178
Columns: 14
$ Type      <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ Alcohol  <dbl> 14.23, 13.20, 13.16, 14.37, 13.24, 14.20, 14.3...
$ Malic    <dbl> 1.71, 1.78, 2.36, 1.95, 2.59, 1.76, 1.87, 2.15...
$ Ash      <dbl> 2.43, 2.14, 2.67, 2.50, 2.87, 2.45, 2.45, 2.61...
$ Alkalinity <dbl> 15.6, 11.2, 18.6, 16.8, 21.0, 15.2, 14.6, 17.6...
...
$ Color    <dbl> 5.64, 4.38, 5.68, 7.80, 4.32, 6.75, 5.25, 5.05...
$ Hue      <dbl> 1.04, 1.05, 1.03, 0.86, 1.04, 1.05, 1.02, 1.06...
$ Dilution <dbl> 3.92, 3.40, 3.17, 3.45, 2.93, 2.85, 3.58, 3.58...
$ Proline  <int> 1065, 1050, 1185, 1480, 735, 1450, 1290, 1295,...
```

Adding a smoother



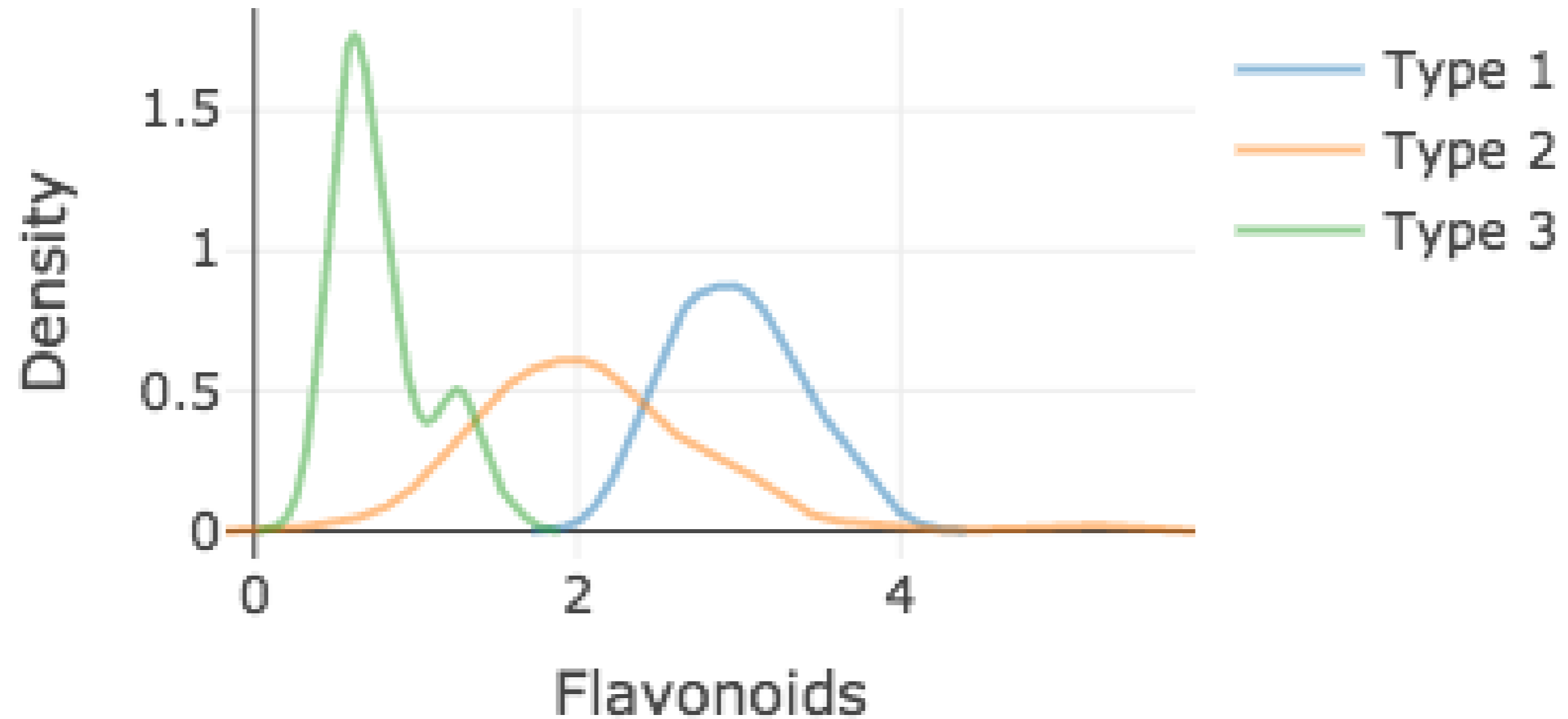
```
m <- loess(Alcohol ~ Flavanoids, data = wine, span = 1.5)
wine %>%
  plot_ly(x = ~Flavanoids, y = ~Alcohol) %>%
  add_markers() %>%
  add_lines(y = ~fitted(m)) %>%
  layout(showlegend = FALSE)
```

Adding a second smoother



```
m2 <- lm(Alcohol ~ poly(Flavanoids, 2), data = wine)
wine %>%
  plot_ly(x = ~Flavanoids, y = ~Alcohol) %>%
  add_markers(showlegend = FALSE) %>%
  add_lines(y = ~fitted(m), name = "LOESS") %>%
  add_lines(y = ~fitted(m2), name = "Polynomial")
```

Layering densities



Layering densities

```
d1 <- filter(wine, Type == 1)
d2 <- filter(wine, Type == 2)
d3 <- filter(wine, Type == 3)
density1 <- density(d1$Flavanoids)
density2 <- density(d2$Flavanoids)
density3 <- density(d3$Flavanoids)
plot_ly(opacity = 0.5) %>%
  add_lines(x = ~density1$x, y = ~density1$y, name = "Type 1") %>%
  add_lines(x = ~density2$x, y = ~density2$y, name = "Type 2") %>%
  add_lines(x = ~density3$x, y = ~density3$y, name = "Type 3") %>%
  layout(xaxis = list(title = 'Flavanoids'),
         yaxis = list(title = 'Density'))
```

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Faceting plotly graphics

INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R



Adam Loy

Statistician, Carleton College

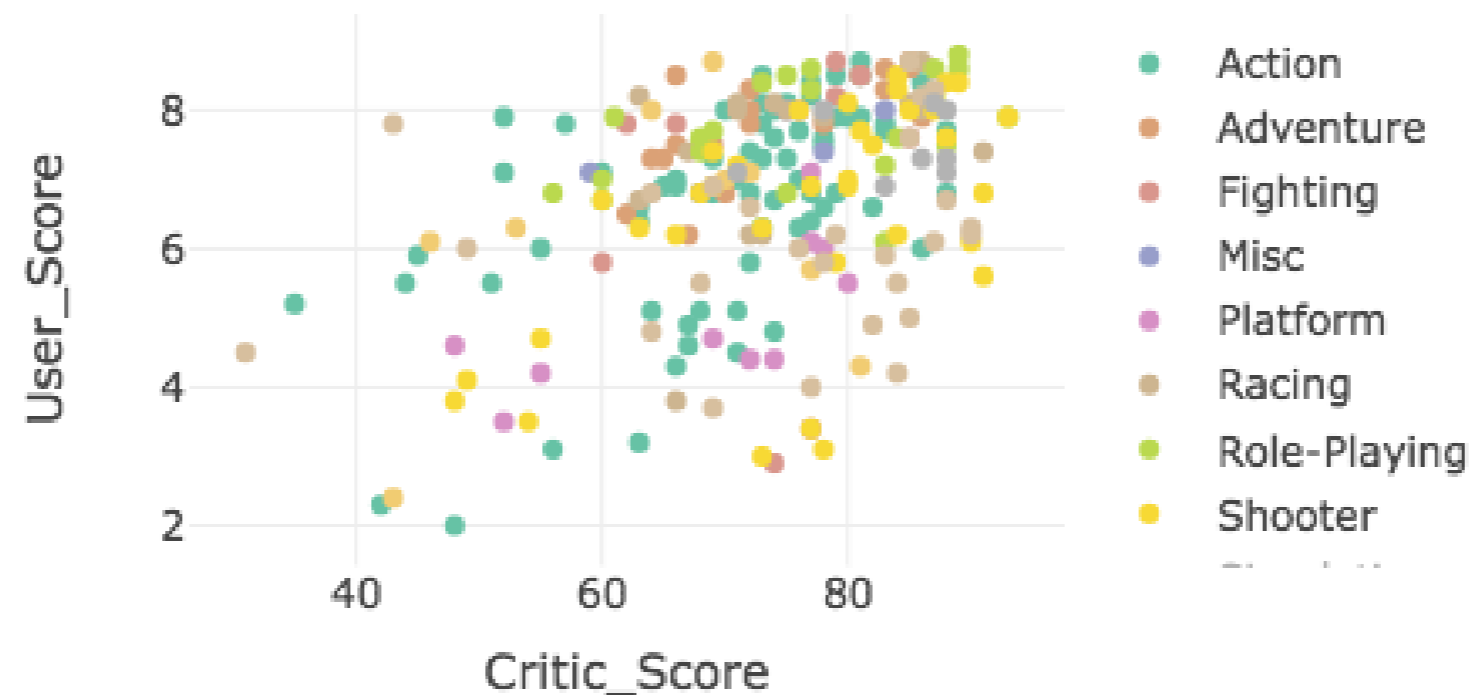
2016 video game sales

```
glimpse(vgsales2016)
```

```
Rows: 502
Columns: 16
$ Name      <fct> FIFA 17, Pokemon Sun/Moon, Unchart...
$ Platform  <fct> PS4, 3DS, PS4, PS4, PS4, PS4, XOne...
$ Year      <int> 2016, 2016, 2016, 2016, 2016, 2016...
$ Genre     <fct> Sports, Role-Playing, Shooter, Sho...
$ Publisher <fct> Electronic Arts, Nintendo, Sony Co...
$ NA_Sales  <dbl> 0.66, 2.98, 1.85, 1.61, 1.10, 1.35...
...
$ User_Score <fct> 5, NA, 7.9, 3.4, 8.4, 7, 5.5, 3.1,...
$ User_Count <int> 398, NA, 7064, 1129, 809, 2219, 20...
$ Developer <fct> EA Sports, EA Vancouver, NA, Naugh...
$ Rating    <fct> E, NA, T, M, M, M, E, M, M, M, M, ...
```

Representing many categories

```
vgsales2016 %>%  
  plot_ly(x = ~Critic_Score, y = ~User_Score, color = ~Genre) %>%  
  add_markers()
```



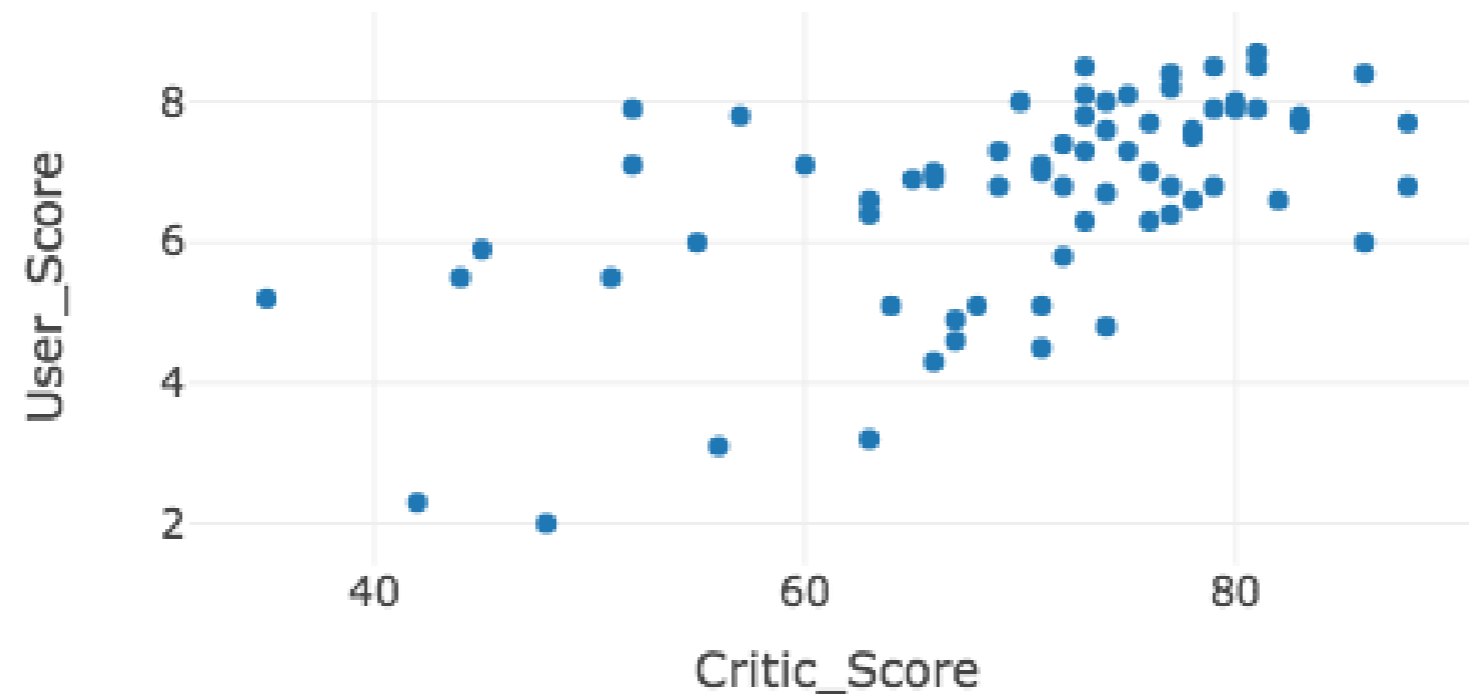
A single subplot

```
library(dplyr)
action_df <- vgsales2016 %>%
  filter(Genre == "Action")
glimpse(action_df)
```

```
Rows: 178
Columns: 16
$ Name      <fct> Far Cry: Primal, Mafia III, No Man's Sky, Yokai Watch 3, Watch Do...
$ Platform  <fct> PS4, PS4, PS4, 3DS, PS4, WiiU, 3DS, PS4, XOne, PS4, PS4, PS4, PS4...
$ Year      <int> 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016,...
$ Genre     <fct> Action, Action, Action, Action, Action, Action, Action, Action, A...
$ Publisher <fct> Ubisoft, Take-Two Interactive, Hello Games, Level 5, Ubisoft, Nin...
$ NA_Sales  <dbl> 0.60, 0.42, 0.63, 0.00, 0.37, 0.56, 0.28, 0.72, 0.47, 0.26, 0.24,...
...
```

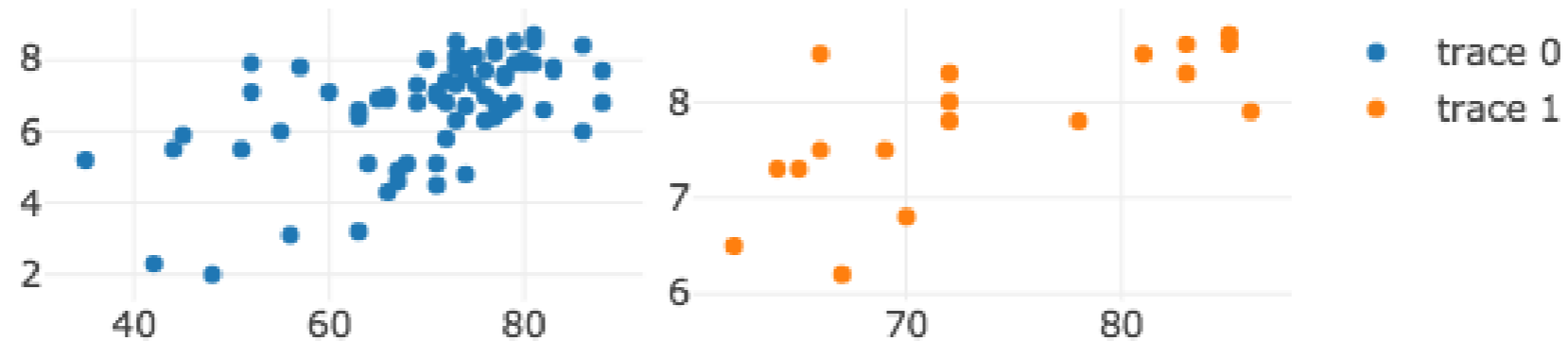
A single subplot

```
action_df %>%  
  plot_ly(x = ~Critic_Score, y = ~User_Score) %>%  
  add_markers()
```



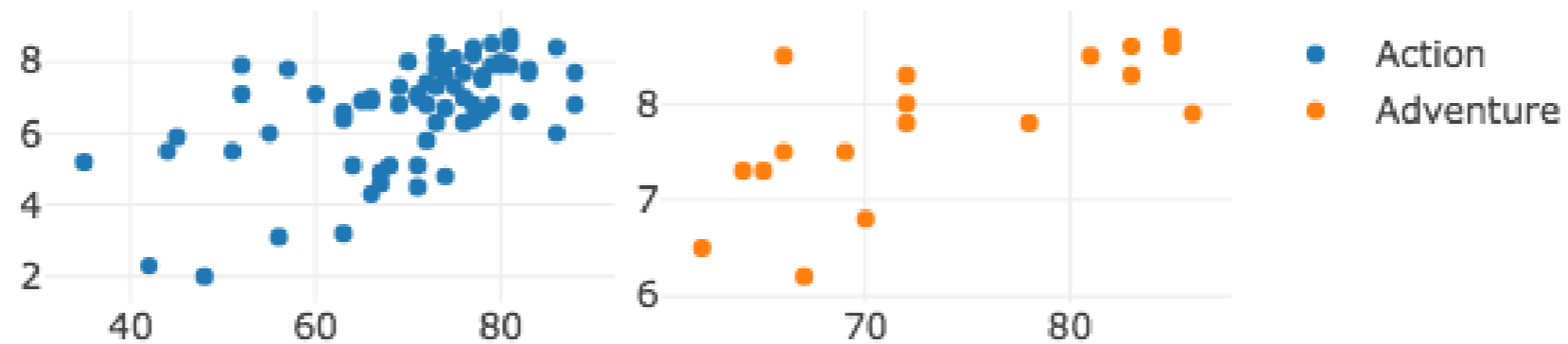
Two subplots

```
p1 <- action_df %>%  
  plot_ly(x = ~Critic_Score, y = ~User_Score) %>%  
  add_markers()  
p2 <- vgsales2016 %>%  
  filter(Genre == "Adventure") %>%  
  plot_ly(x = ~Critic_Score, y = ~User_Score) %>%  
  add_markers()  
subplot(p1, p2, nrows = 1)
```



Legends

```
p1 <- plot_ly(x = ~Critic_Score, y = ~User_Score) %>%  
  add_markers(name = ~Genre)  
  
p2 <- vgsales2016 %>%  
  filter(Genre == "Adventure") %>%  
  plot_ly(x = ~Critic_Score, y = ~User_Score) %>%  
  add_markers(name = ~Genre)  
  
subplot(p1, p2, nrow = 1)
```



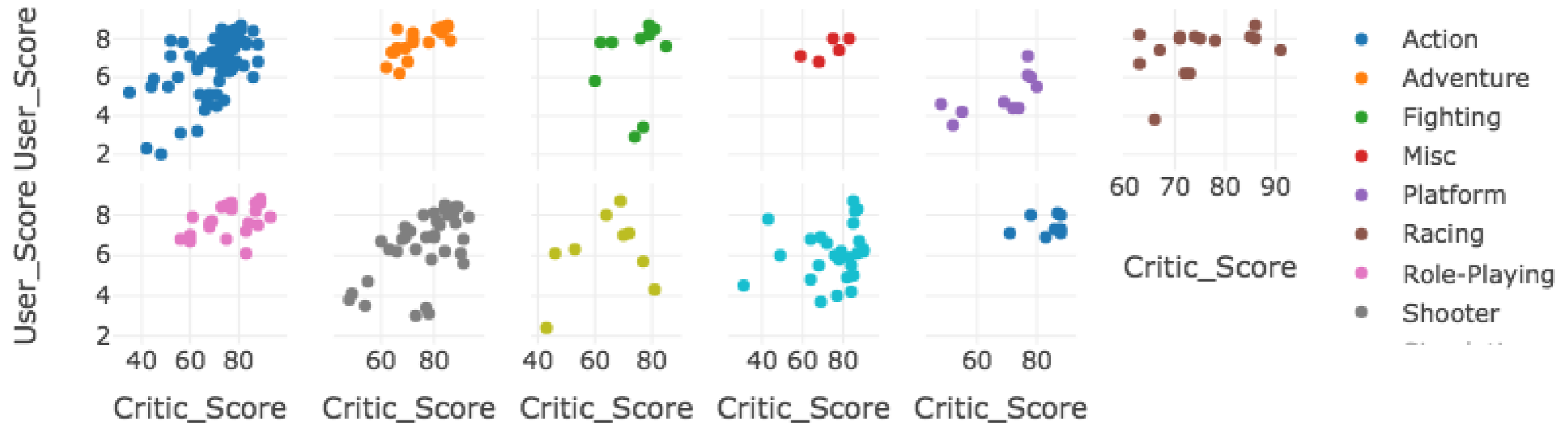
Axis labels

```
subplot(p1, p2, nrow = 1, shareY = TRUE, shareX = TRUE)
```



- Sharing an axis leads to linked interactivity
- If linked interactivity is not desired: use `titleX` and `titleY` arguments

Iterate to automate



Iterate to automate

```
library(tidyverse)
vgsales2016 %>%
  group_by(Genre) %>%
  nest()
```

```
# A tibble: 12 × 2
# Groups:   Genre [12]
  Genre      data
  <chr>      <list>
1 Sports     <tibble [48 × 15]>
2 Role-Playing <tibble [54 × 15]>
3 Shooter    <tibble [47 × 15]>
4 Action     <tibble [178 × 15]>
5 Platform  <tibble [15 × 15]>
6 Fighting  <tibble [16 × 15]>
7 Racing    <tibble [24 × 15]>
8 Adventure  <tibble [56 × 15]>
9 Misc      <tibble [32 × 15]>
```

Iterate to automate

```
library(tidyverse)
vgsales2016 %>%
  group_by(Genre) %>%
  nest() %>%
  mutate(
    plot = map2(data, Genre,
               \(data, Genre)
               plot_ly(data = data,
                       x = ~Critic_Score,
                       y = ~User_Score) %>%
               add_markers(name = ~Genre)
    ))
```

```
# A tibble: 12 × 3
# Groups:   Genre [12]
  Genre      data      plot
  <chr>     <list>  <list>
1 Sports   <tibble [48 × 15]> <plot
2 Role-Playing <tibble [54 × 15]> <plot
3 Shooter  <tibble [47 × 15]> <plot
4 Action   <tibble [178 × 15]> <plot
5 Platform <tibble [15 × 15]> <plot
6 Fighting <tibble [16 × 15]> <plot
7 Racing   <tibble [24 × 15]> <plot
8 Adventure <tibble [56 × 15]> <plot
9 Misc     <tibble [32 × 15]> <plot
```

Iterate to automate

```
vgsales2016 %>%  
  group_by(Genre) %>%  
  nest() %>%  
  mutate(  
    plot = map2(  
      data, Genre,  
      \(data, Genre)  
        plot_ly(data = data,  
          x = ~Critic_Score,  
          y = ~User_Score) %>%  
          add_markers(name = ~Genre)  
    )) %>%  
  subplot(nrows = 2)
```

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Interactive scatterplot matrices

INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R



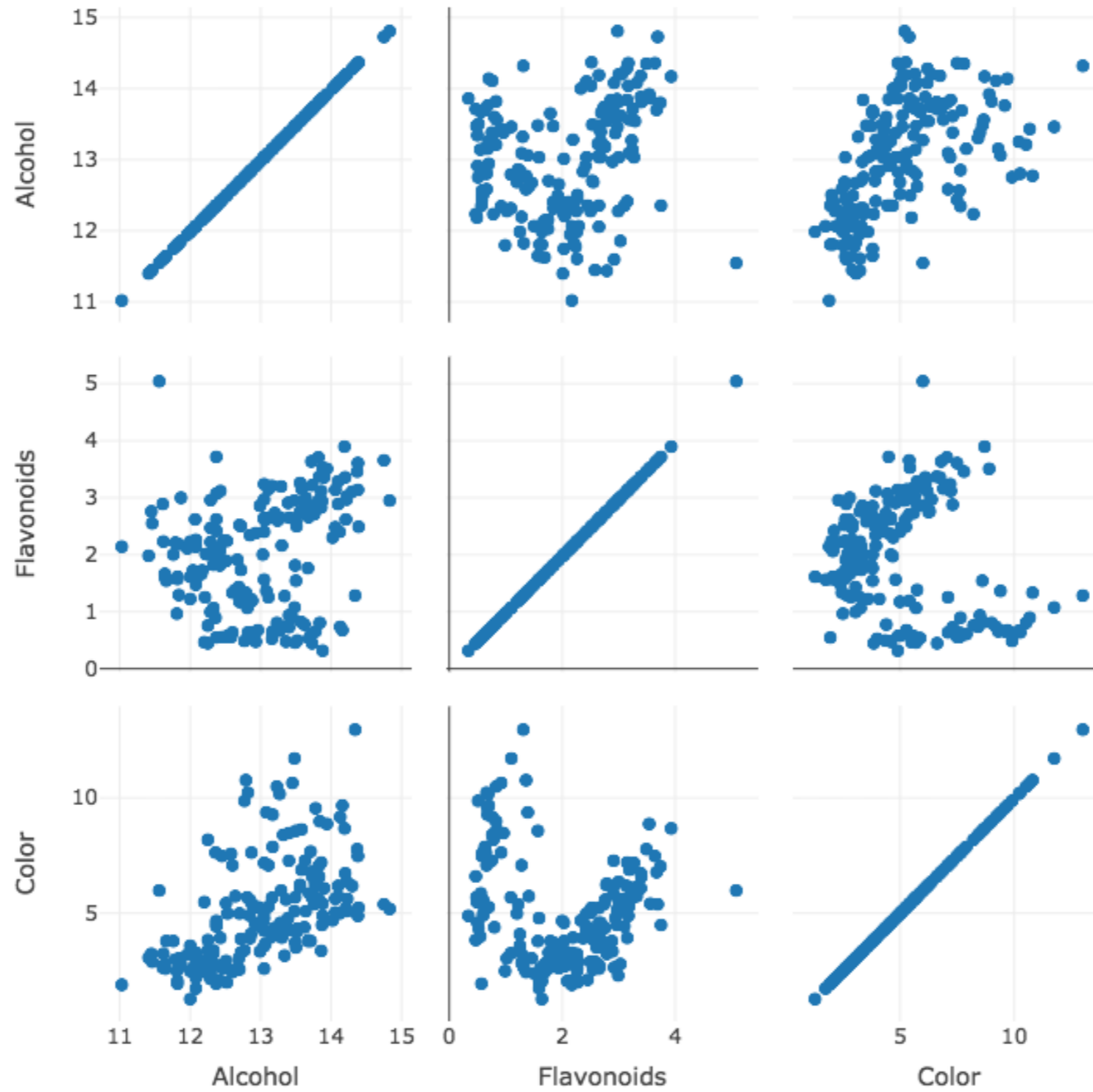
Adam Loy

Statistician, Carleton College

Wine data

```
glimpse(wine)
```

```
Rows: 178
Columns: 14
$ Type      <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ Alcohol  <dbl> 14.23, 13.20, 13.16, 14.37, 13.24, 14.20, 14.3...
$ Malic    <dbl> 1.71, 1.78, 2.36, 1.95, 2.59, 1.76, 1.87, 2.15...
$ Ash      <dbl> 2.43, 2.14, 2.67, 2.50, 2.87, 2.45, 2.45, 2.61...
$ Alkalinity <dbl> 15.6, 11.2, 18.6, 16.8, 21.0, 15.2, 14.6, 17.6...
$ Magnesium <int> 127, 100, 101, 113, 118, 112, 96, 121, 97, 98,...
...
$ Color    <dbl> 5.64, 4.38, 5.68, 7.80, 4.32, 6.75, 5.25, 5.05...
$ Hue      <dbl> 1.04, 1.05, 1.03, 0.86, 1.04, 1.05, 1.02, 1.06...
$ Dilution <dbl> 3.92, 3.40, 3.17, 3.45, 2.93, 2.85, 3.58, 3.58...
$ Proline  <int> 1065, 1050, 1185, 1480, 735, 1450, 1290, 1295,...
```



The template

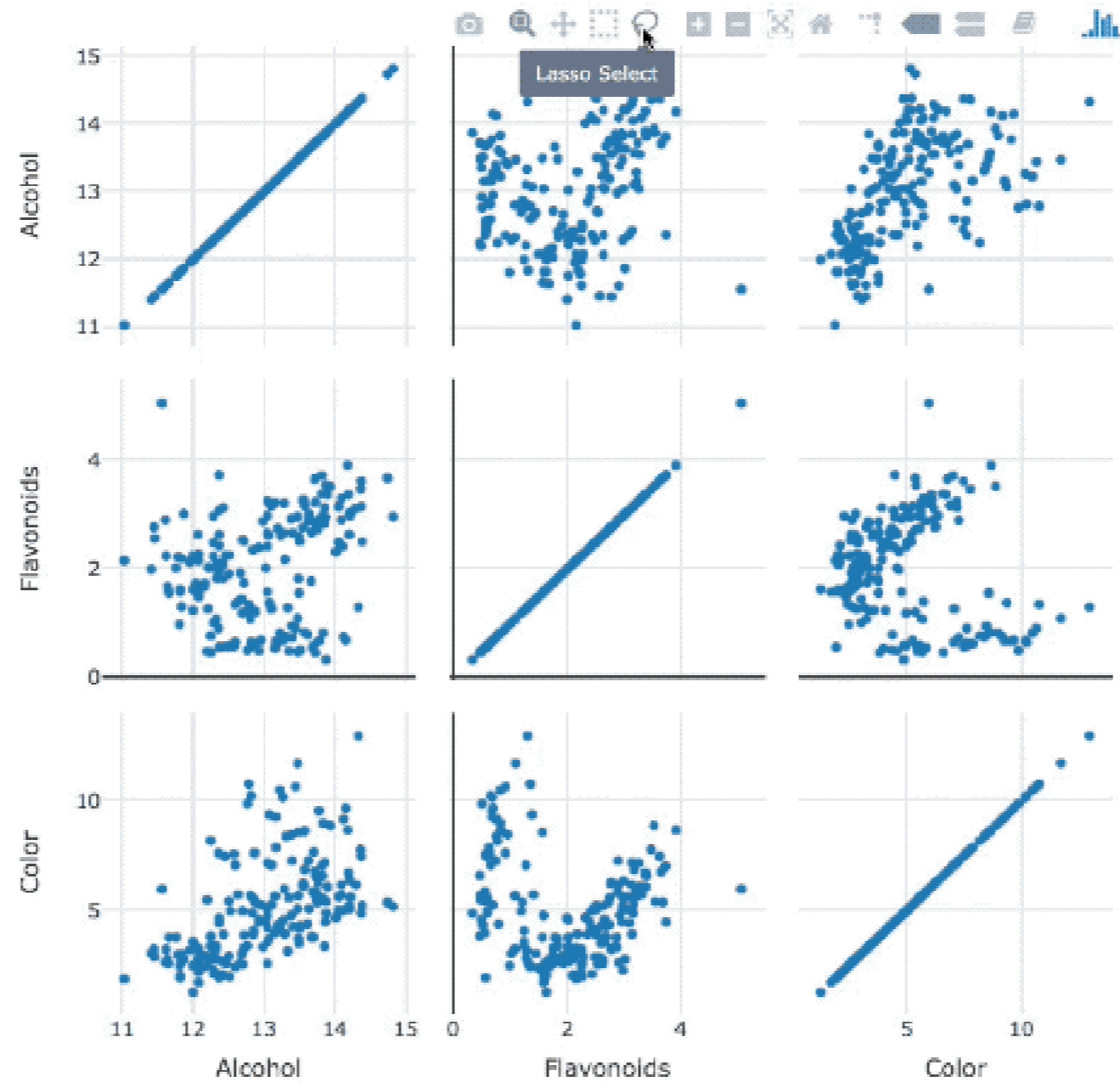
```
data %>%  
  plot_ly() %>%  
  add_trace(  
    type = 'splom',  
    dimensions = list(  
      list(label='string-1', values=~X1),  
      list(label='string-2', values=~X2),  
      .  
      .  
      .  
      list(label='string-n', values=~Xn))  
  )
```

- `add_trace()` to specify variables
- For each variable, two arguments:
 - String for axis label
 - Mapping specifying variable
- `'splom'` trace type = scatterplot matrix

Wine SPLOM

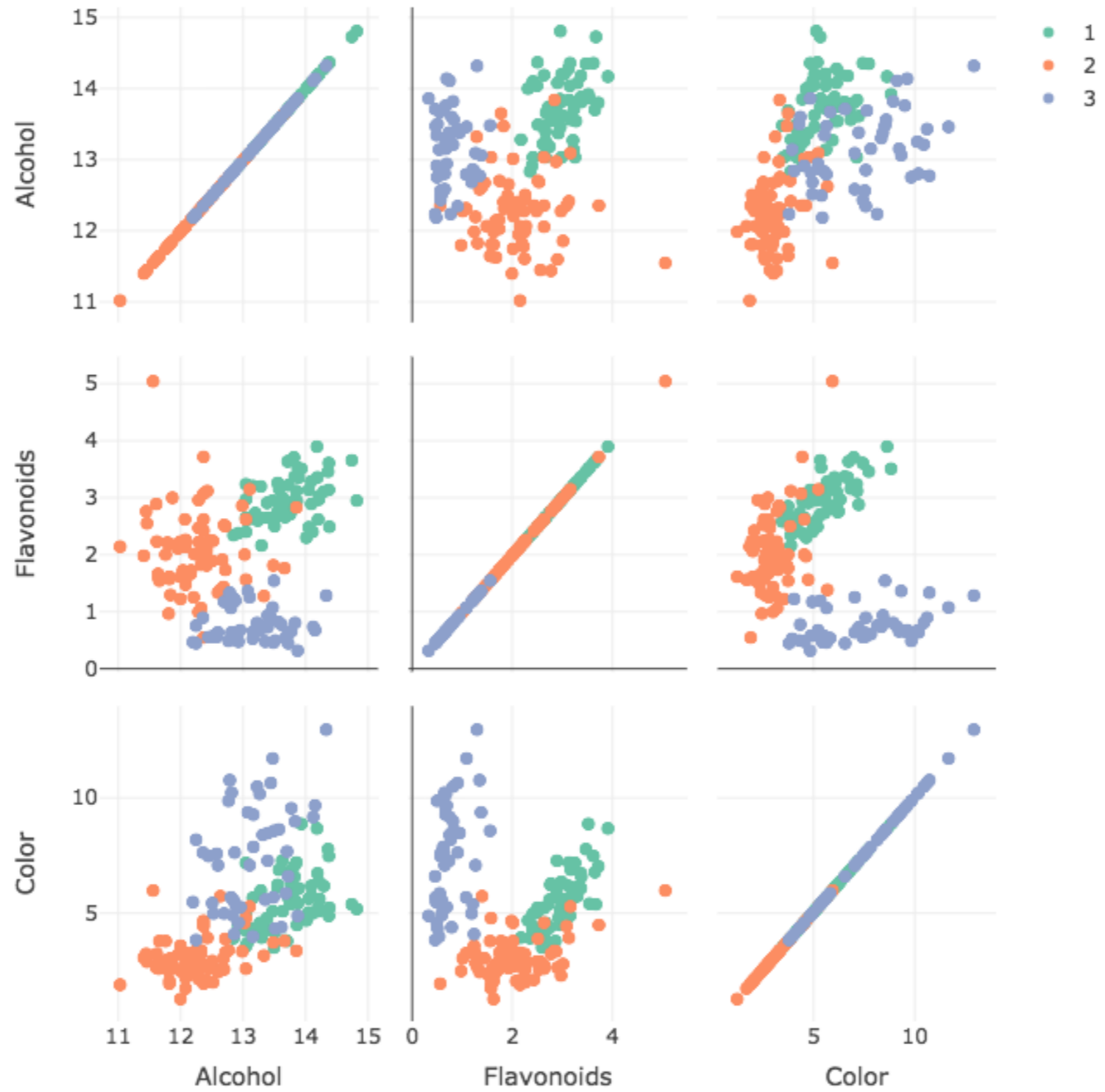
```
wine %>%  
  plot_ly() %>%  
  add_trace(  
    type = 'splom',  
    dimensions = list(  
      list(label='Alcohol', values=~Alcohol),  
      list(label='Flavonoids', values=~Flavonoids),  
      list(label='Color', values=~Color)  
    )  
  )  
)
```

Linked brushing



Adding color

```
wine %>%  
  plot_ly(color = ~Type) %>%  
  add_trace(  
    type = 'splom',  
    dimensions = list(  
      list(label='Alcohol', values=~Alcohol),  
      list(label='Flavonoids', values=~Flavonoids),  
      list(label='Color', values=~Color)  
    )  
  )  
)
```



Let's practice!

INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

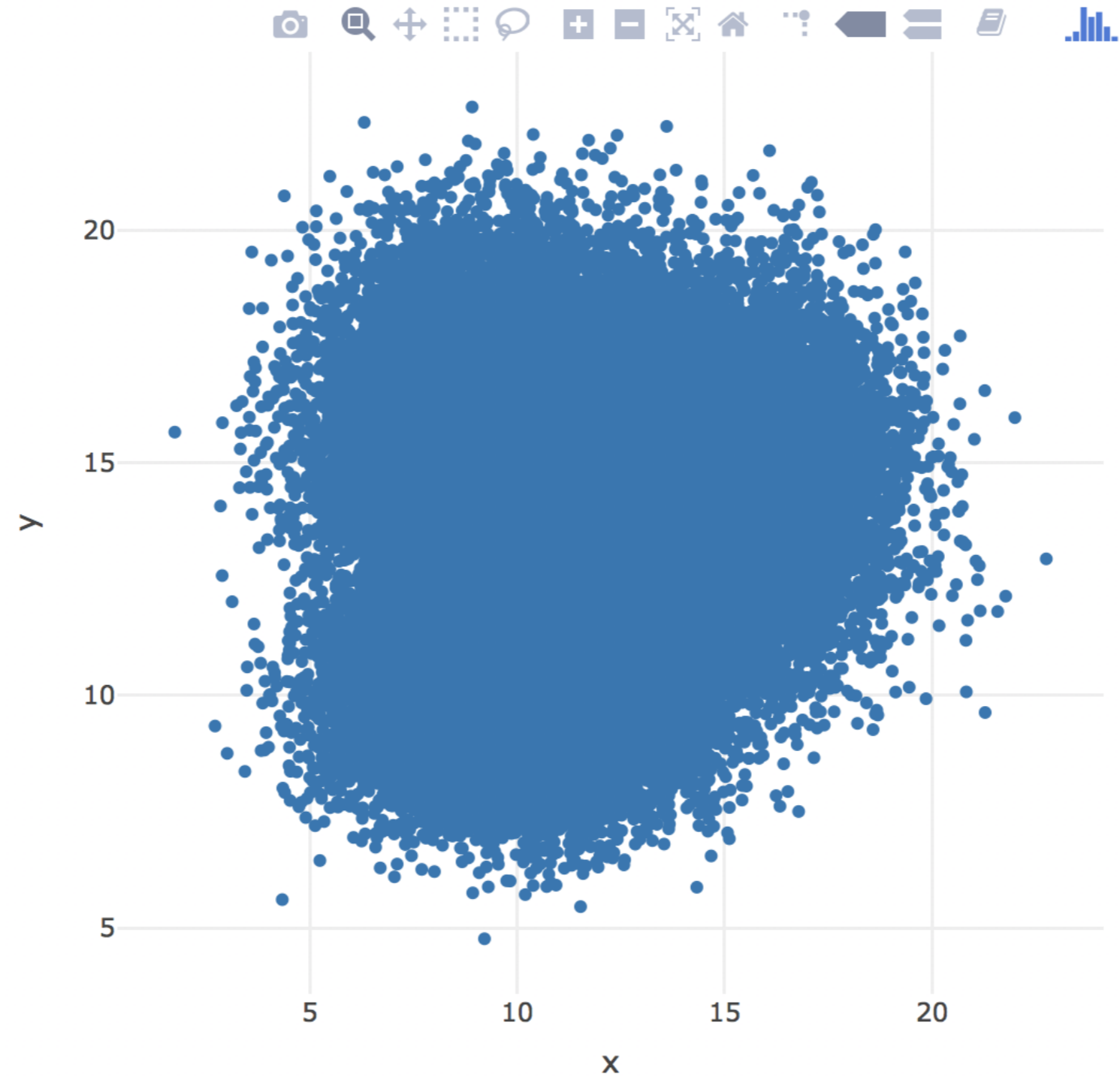
Binned scatterplots

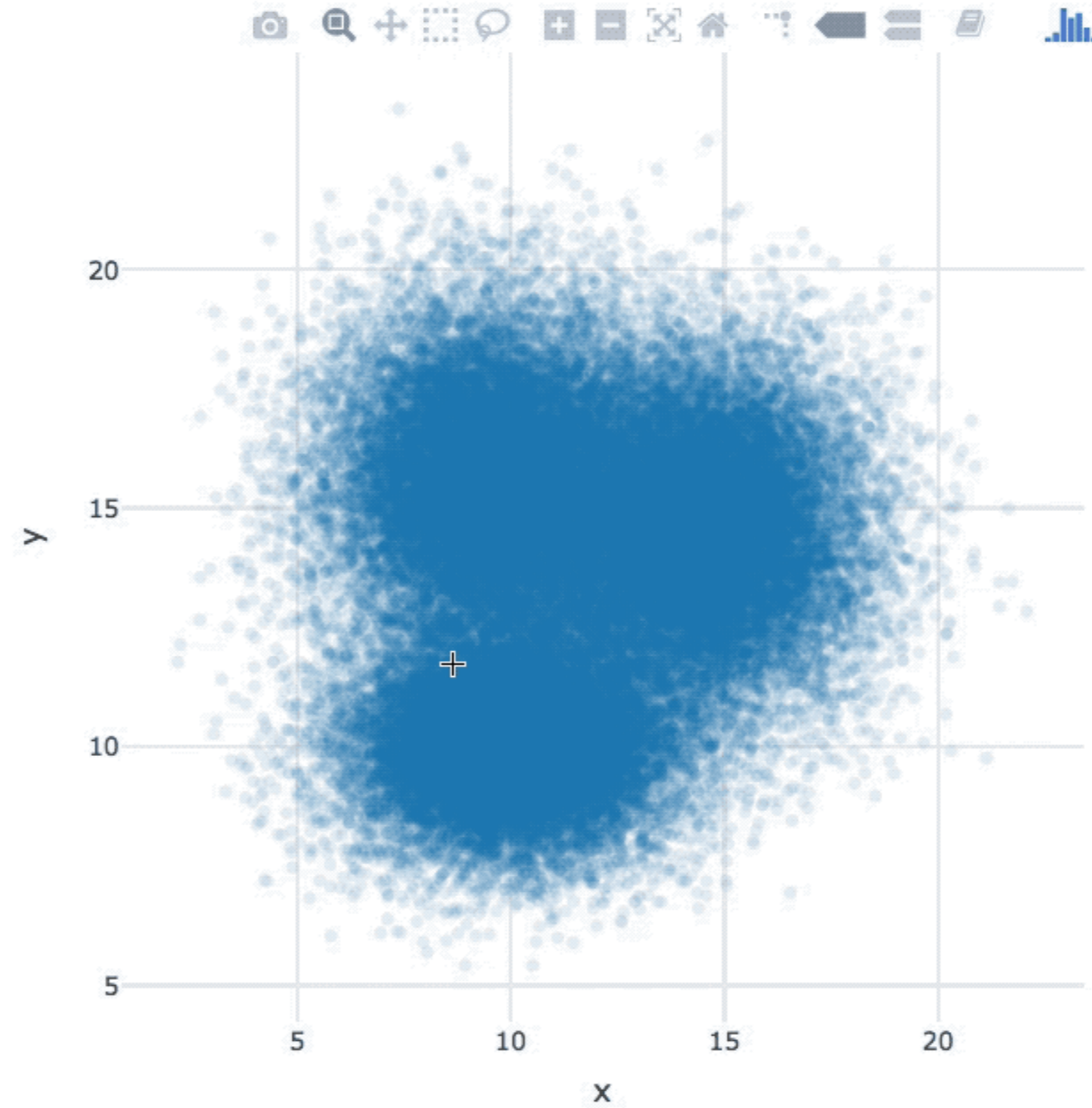
INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

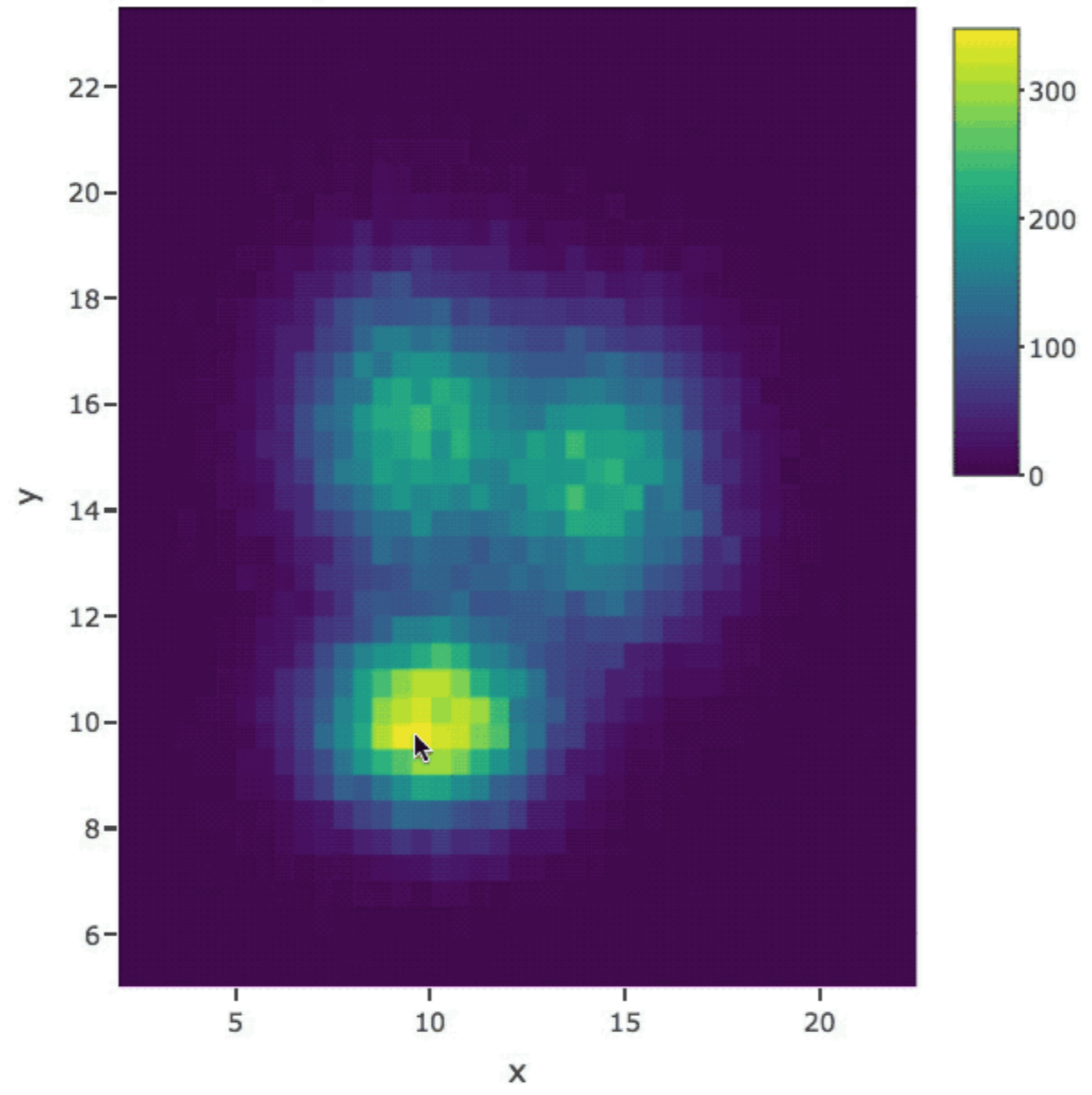


Adam Loy

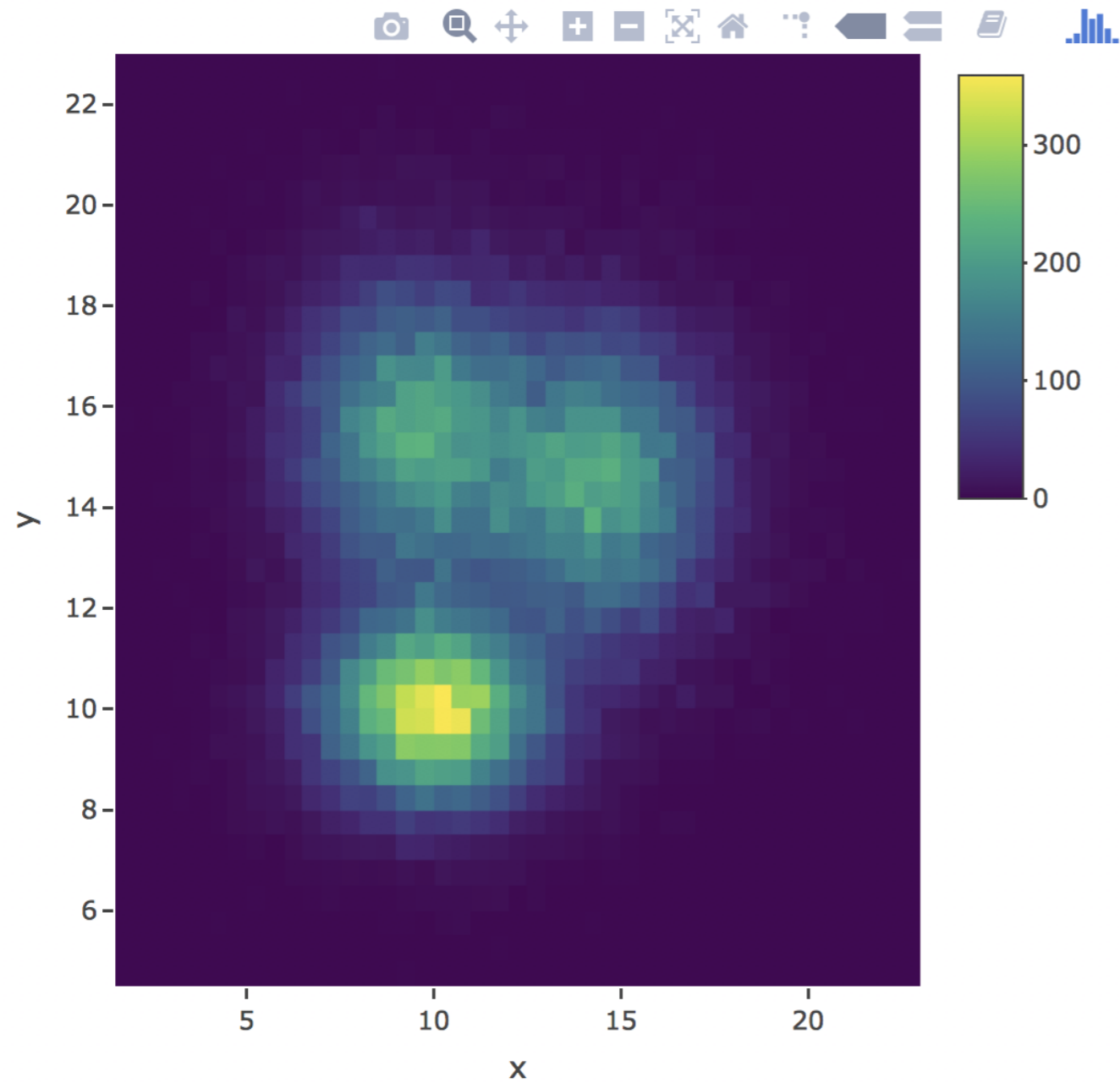
Statistician, Carleton College





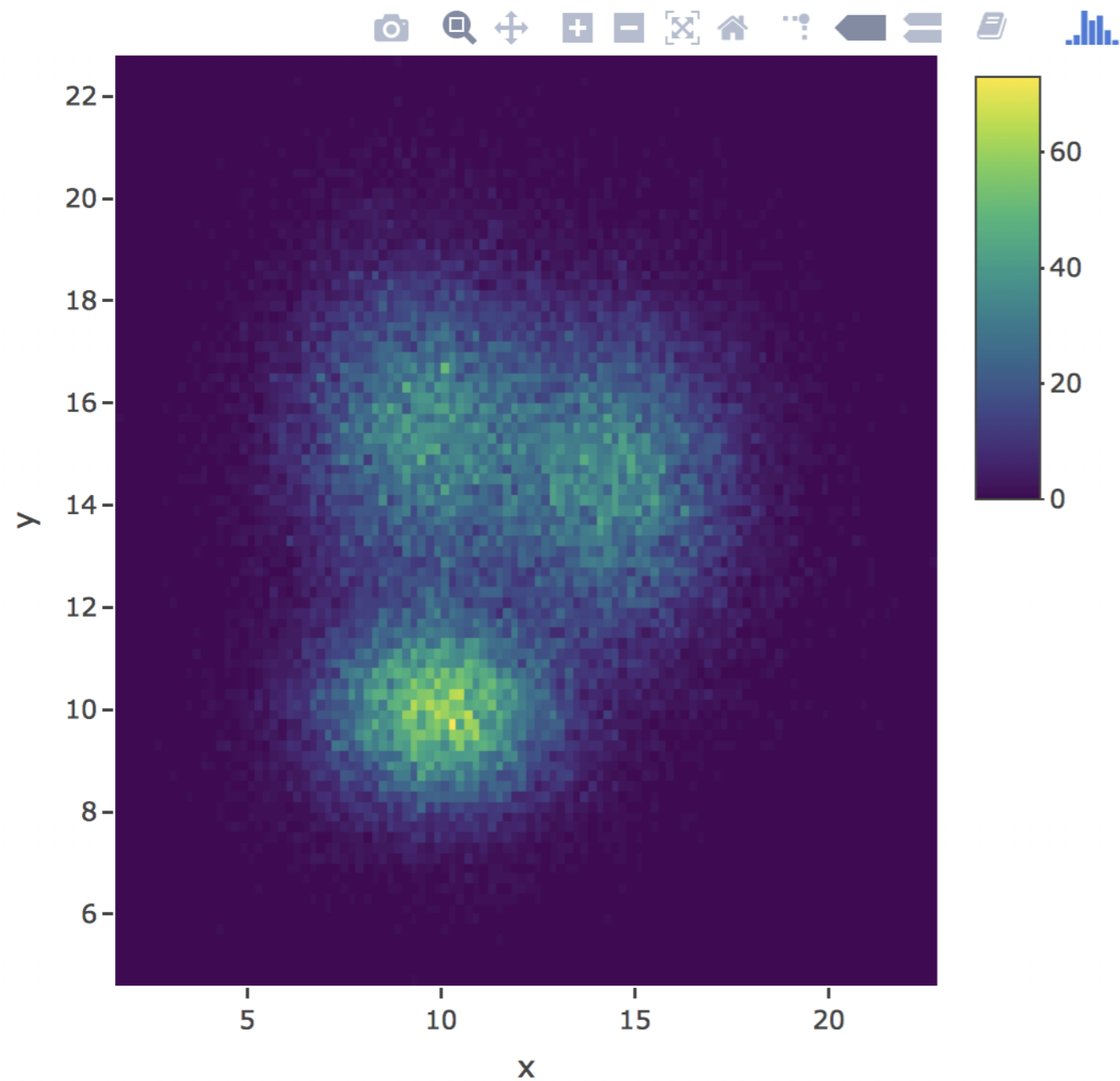


add_histogram2d()



```
sim_data %>%  
  plot_ly(x = ~x, y = ~y) %>%  
  add_histogram2d()
```

Changing the bins



```
sim_data %>%  
  plot_ly(x = ~x, y = ~y) %>%  
  add_histogram2d(nbinsx = 200, nbinsy = 100)
```

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R