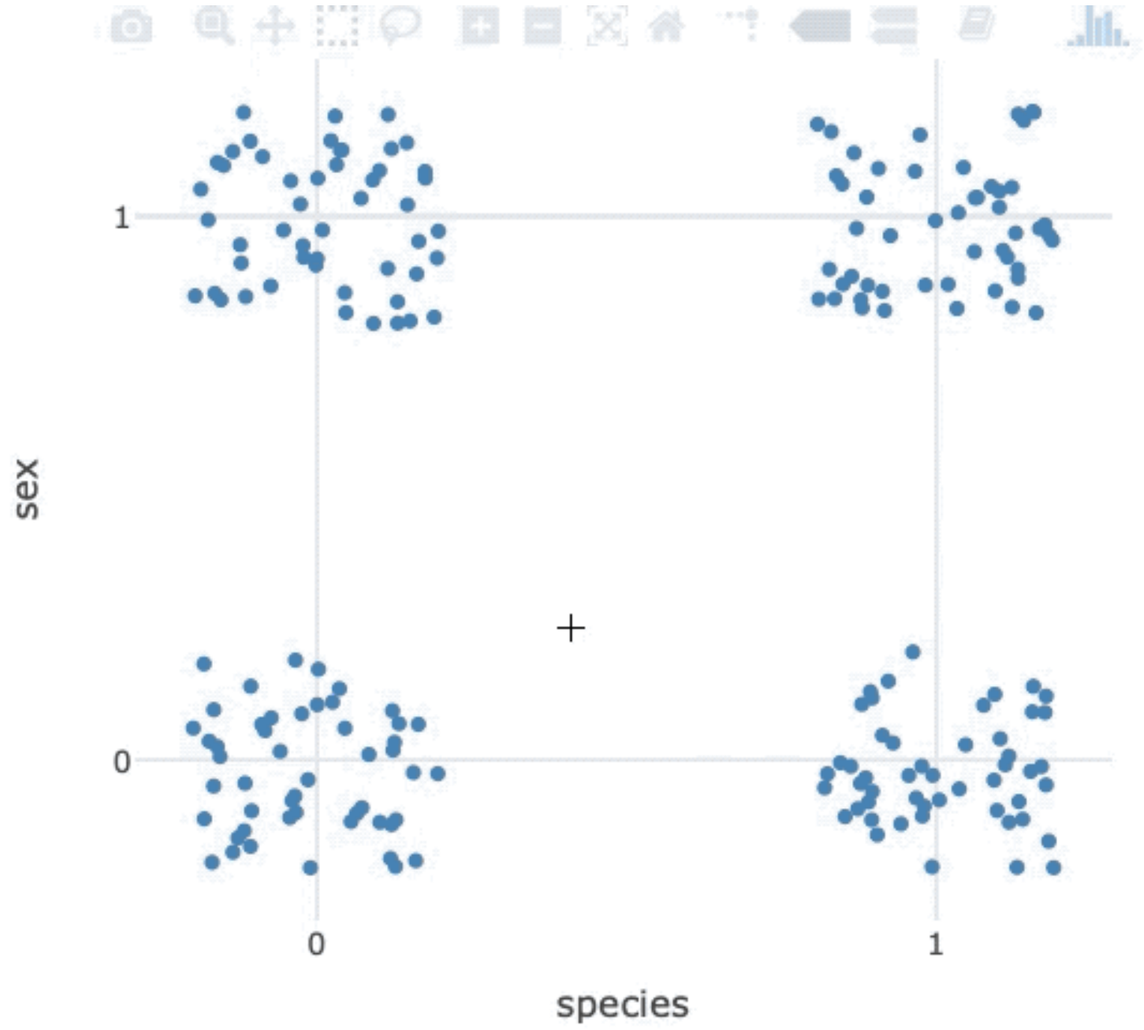
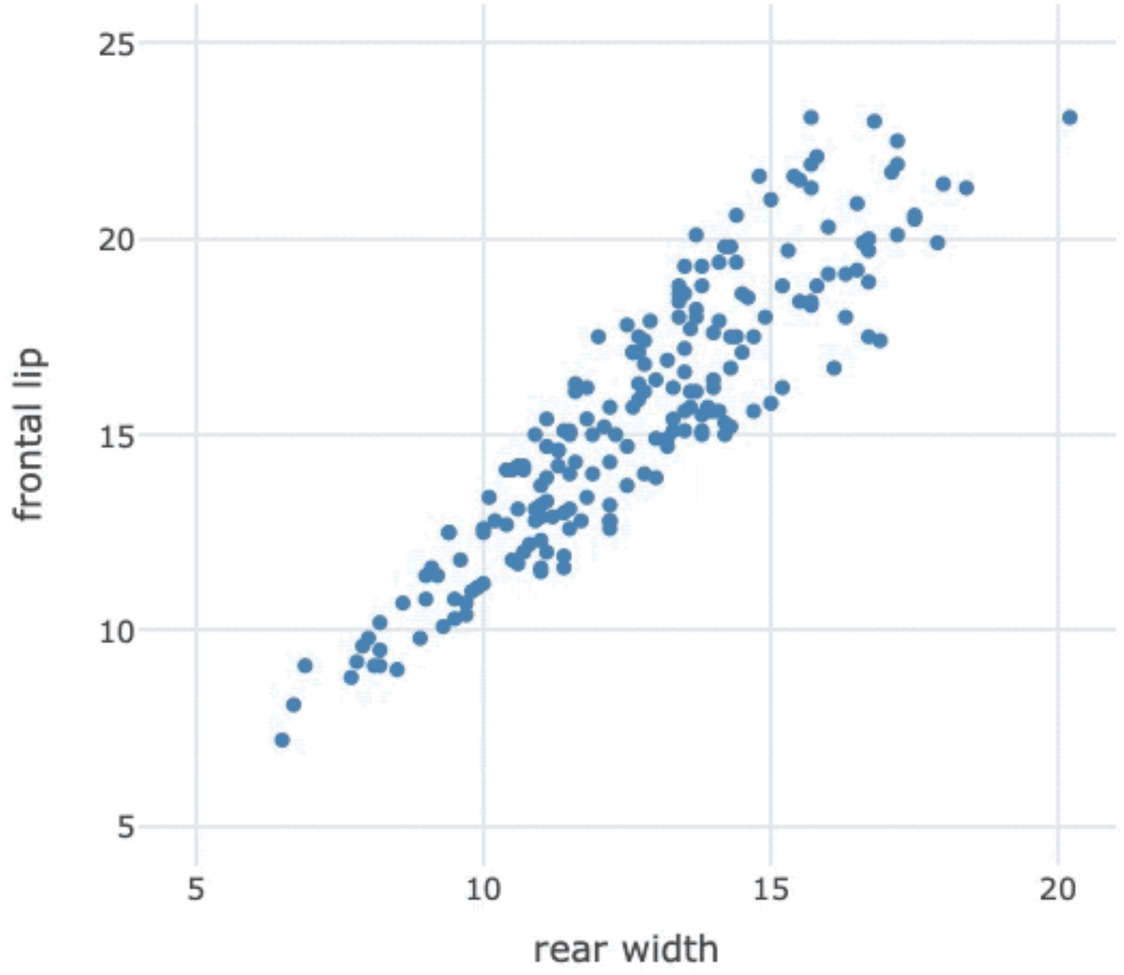
Linking two charts

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

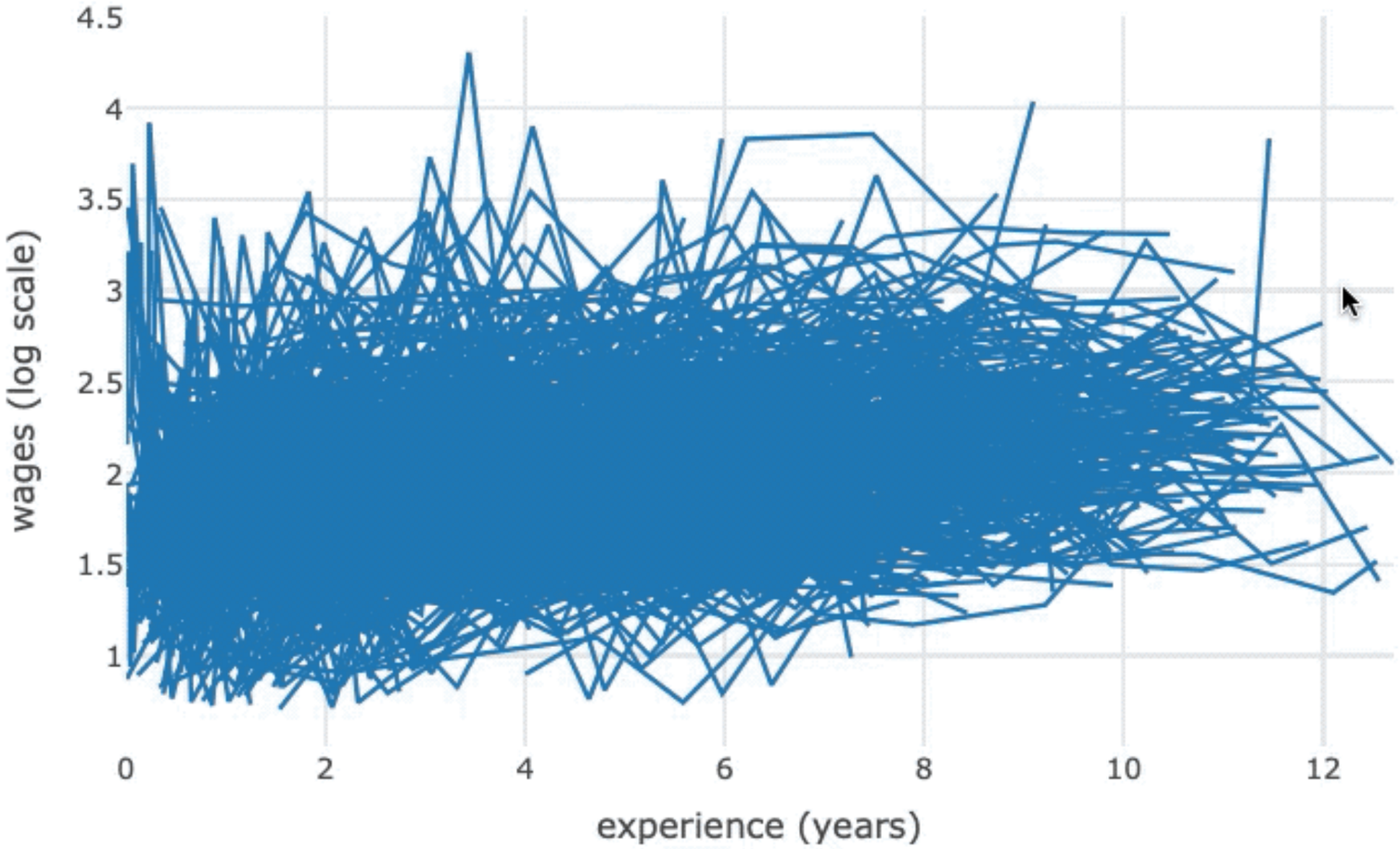
Adam Loy
Statistician, Carleton College

Exploring clusters





Exploring longitudinal data



Crosstalk

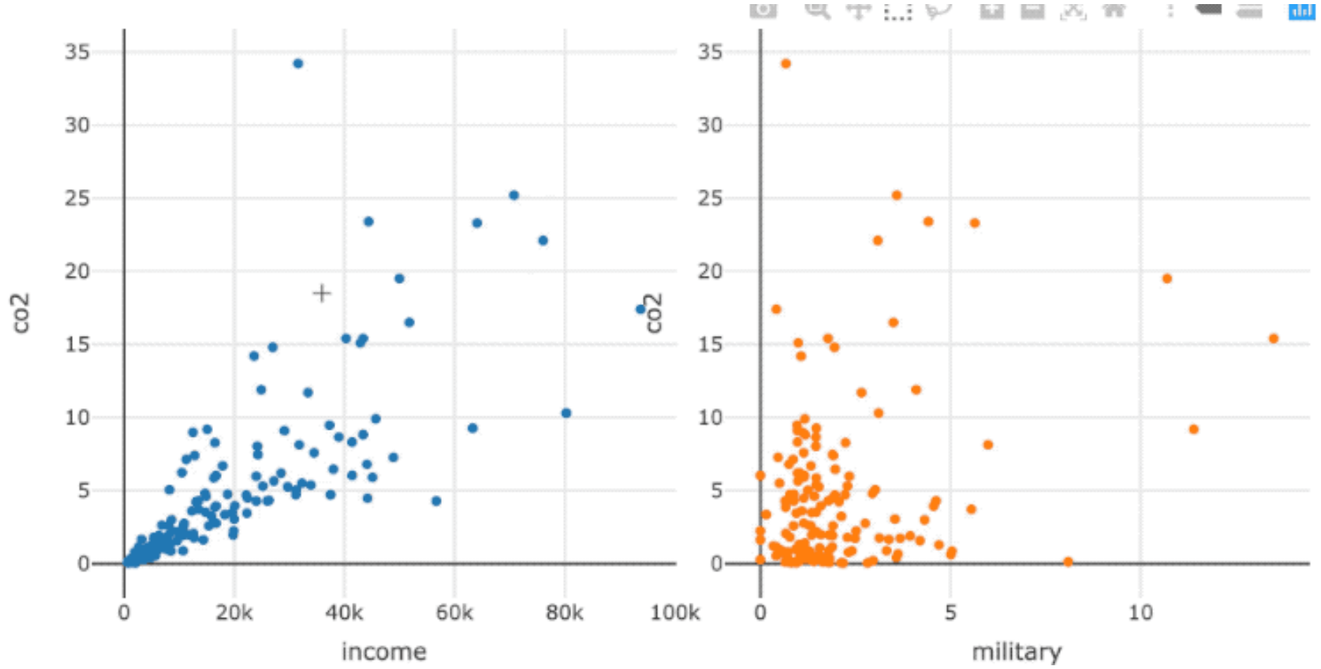
- Enables linked plots via JavaScript
- Creates static HTML files that you can easily host
- Displays in the RStudio viewer pane

2014 world indicators

```
world2014
```

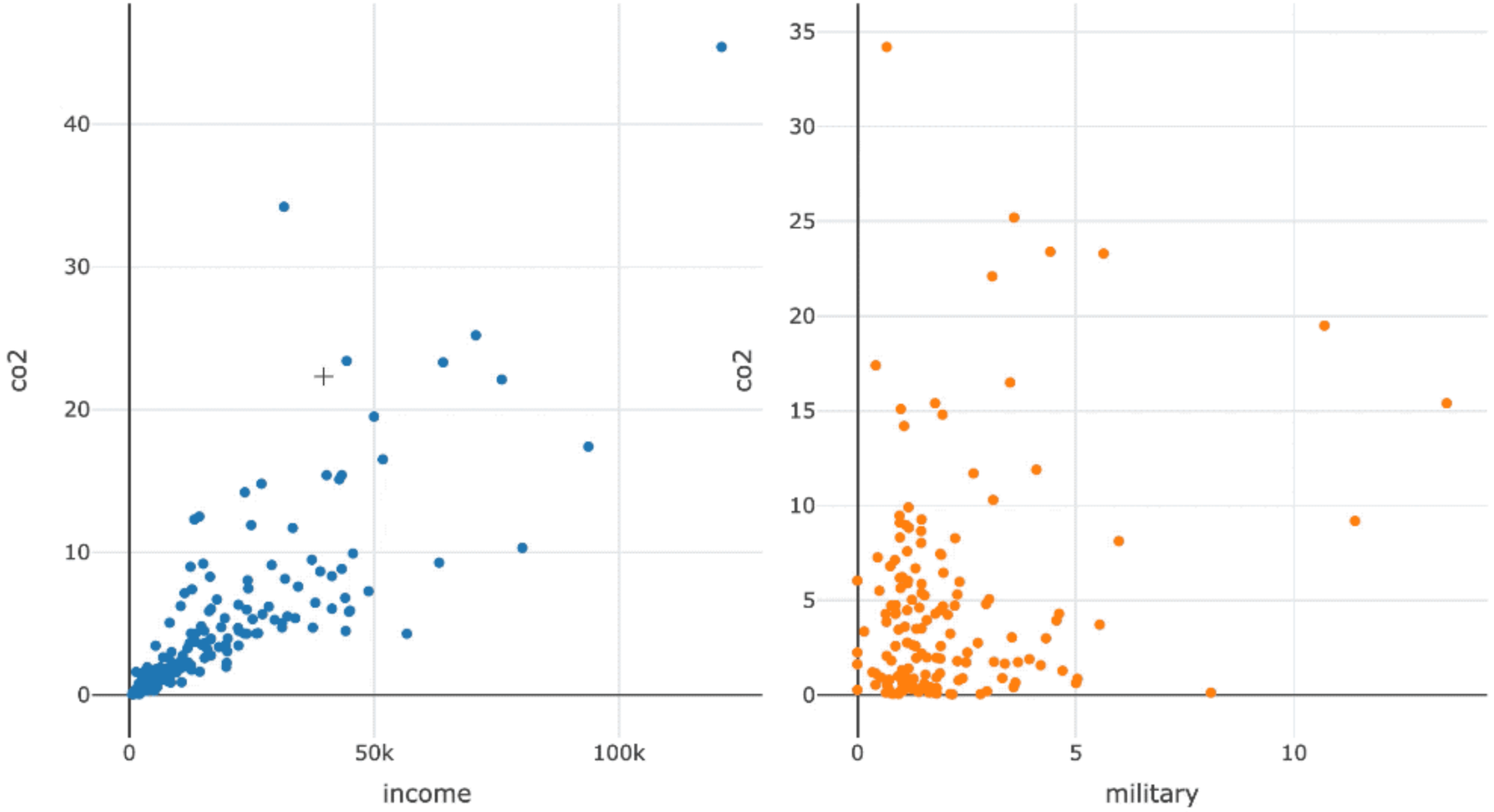
```
# A tibble: 193 x 11
  country  year income   co2 military population  urban life_expectancy four_regions
  <chr>   <dbl>  <dbl> <dbl>    <dbl>      <dbl>  <dbl>           <dbl> <chr>
1 Afghan…  2014   1780 0.299      1.3    32800000 8.05e6            57.8 asia
2 Albania  2014  10700 1.96       1.35    2920000 1.63e6            77.4 europe
3 Algeria  2014  13500 3.72       5.55   39100000 2.75e7            77.1 africa
4 Andorra  2014  44900 5.83      NA         79200 7.01e4            82.6 europe
5 Angola   2014   6260 1.29       4.7    26900000 1.69e7            63.3 africa
6 Antigu…  2014  19500 5.38      NA         98900 2.49e4            77.1 americas
# … with 187 more rows, and 2 more variables: eight_regions <chr>, six_regions <chr>
```

Separate views




```
p1 <- world2014 %>% plot_ly(x = ~income, y = ~co2) %>% add_markers()
p2 <- world2014 %>% plot_ly(x = ~military, y = ~co2) %>% add_markers()
subplot(p1, p2, titleX = TRUE, titleY = TRUE) %>%
  hide_legend()
```

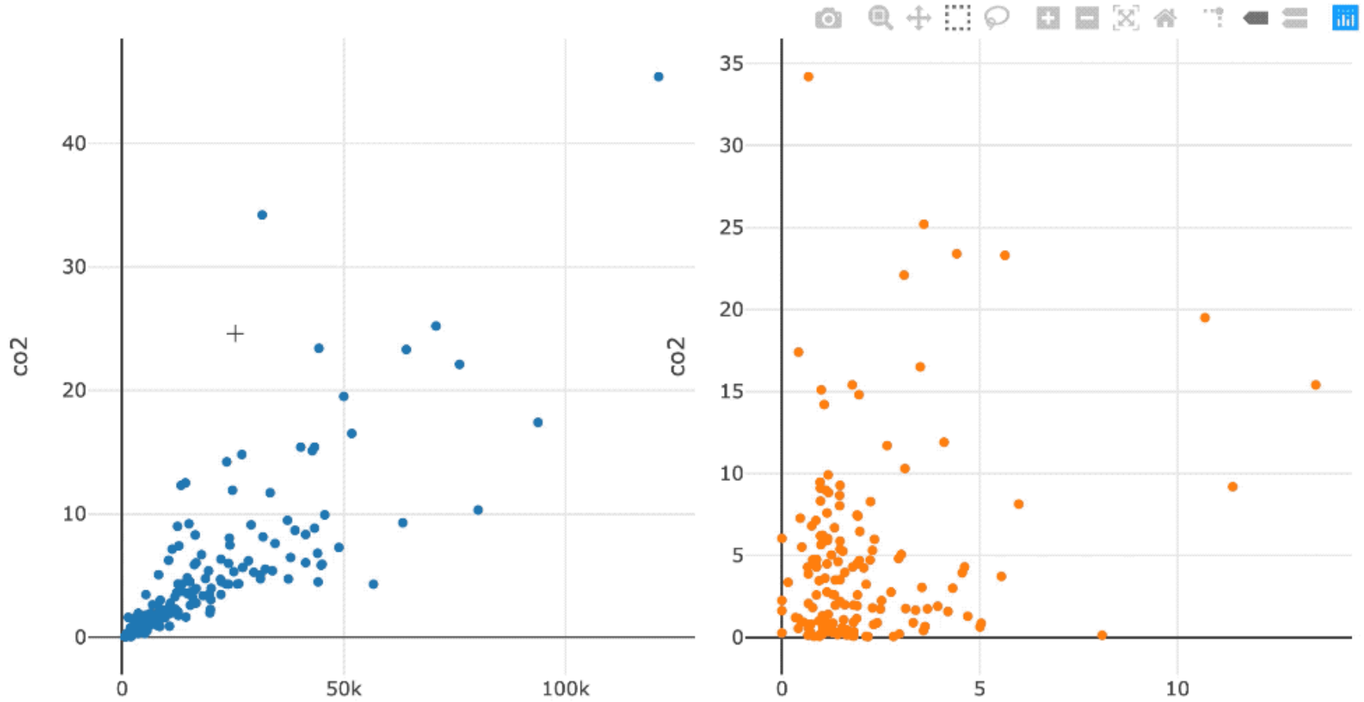
Linked views



Linked views

```
library(crosstalk)
shared_data <- SharedData$new(world2014)
p1 <- shared_data %>%
  plot_ly(x = ~income, y = ~co2) %>%
  add_markers()
p2 <- shared_data %>%
  plot_ly(x = ~military, y = ~co2) %>%
  add_markers()
subplot(p1, p2, titleX = TRUE, titleY = TRUE) %>%
  hide_legend()
```

Linked brushing



Enable linked brushing via `highlight()`

```
subplot(p1, p2, titleX = TRUE, titleY = TRUE) %>%
  hide_legend() %>%
  highlight(on = "plotly_selected")
```

Let's practice!

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Brushing groups

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

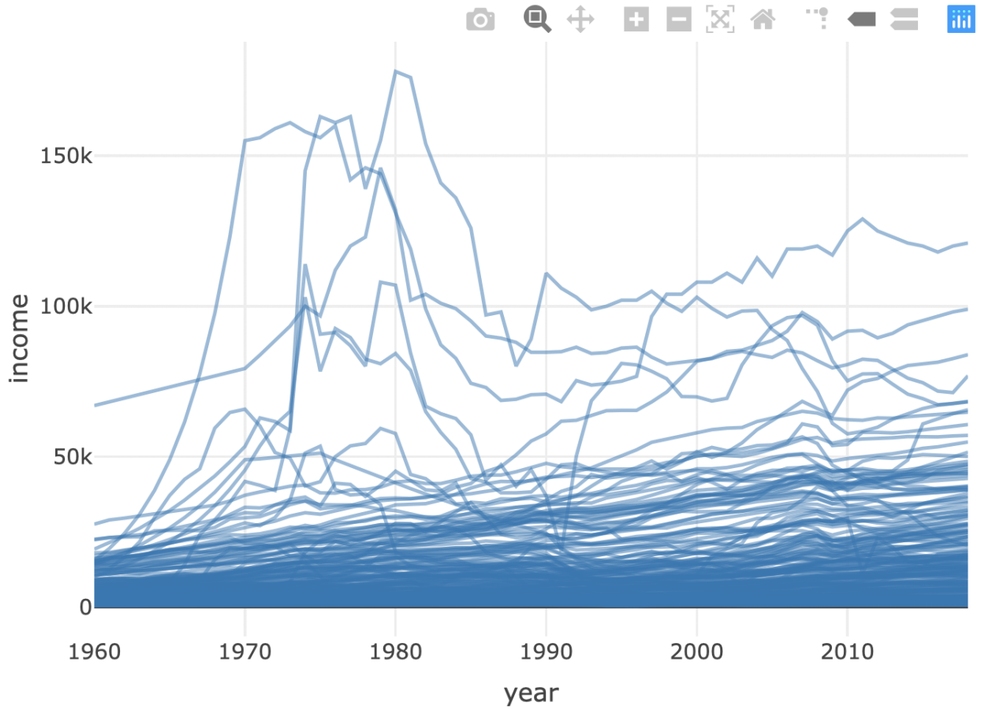
Adam Loy
Statistician, Carleton College

World indicators

```
world_indicators
```

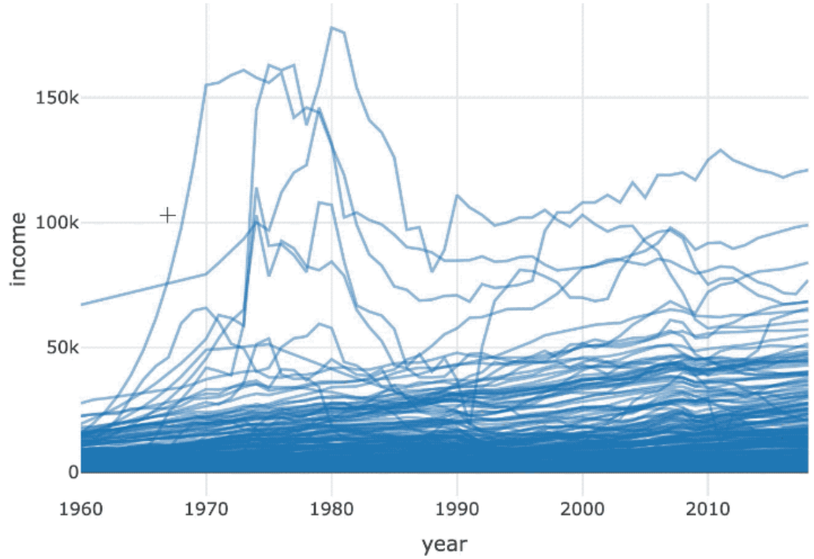
```
# A tibble: 11,387 x 11
  country  year income     co2 military population  urban life_expectancy four_regions
  <chr>   <dbl>  <dbl>   <dbl>    <dbl>      <dbl>  <dbl>           <dbl> <chr>
1 Afghan…  1960   1210  0.0461       NA    9000000 7.56e5            38.6 asia
2 Albania  1960   2790  1.24         NA    1640000 4.94e5            62.7 europe
3 Algeria  1960   6520  0.554        NA   11100000 3.39e6            52   africa
4 Andorra  1960  15200 NA            NA      13400 7.84e3            NA   europe
5 Angola   1960   3860  0.0975       NA    5640000 5.89e5            42.4 africa
6 Antigu…  1960   4420  0.663        NA      55300 2.19e4            62.9 americas
# … with 1.138e+04 more rows, and 2 more variables: eight_regions <chr>,
#   six_regions <chr>
```

Selecting individual time series




```
world_indicators %>%
  plot_ly(x = ~year, y = ~income, alpha = 0.5) %>%
  group_by(country) %>%
  add_lines()
```

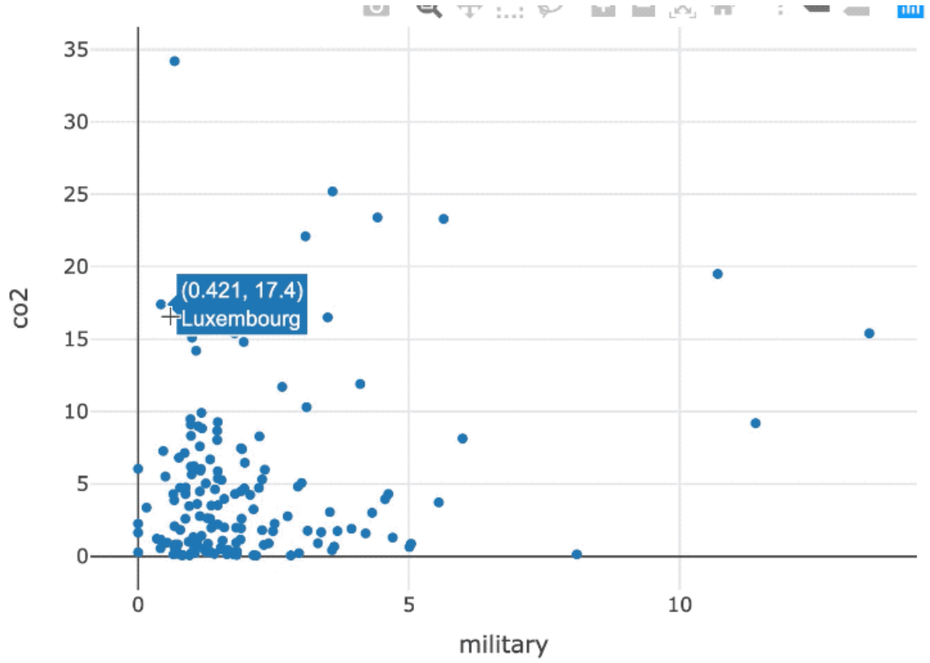
Selecting individual time series



Create a `SharedData` object with a key

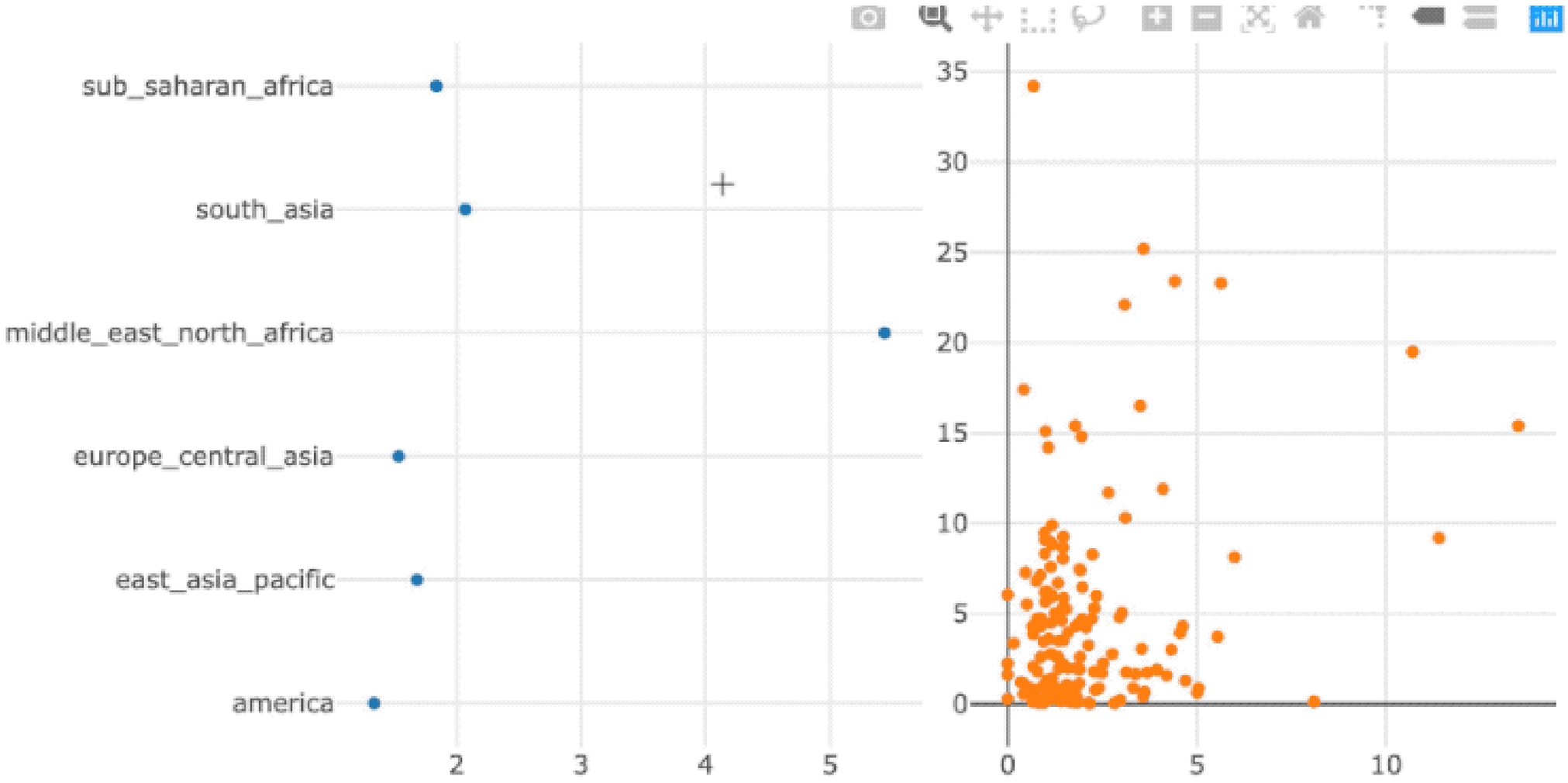
```
world_indicators %>%
  SharedData$new(key = ~country) %>%
  plot_ly(x = ~year, y = ~income, alpha = 0.5) %>%
  group_by(country) %>%
  add_lines()
```

Selecting groups on a scatterplot



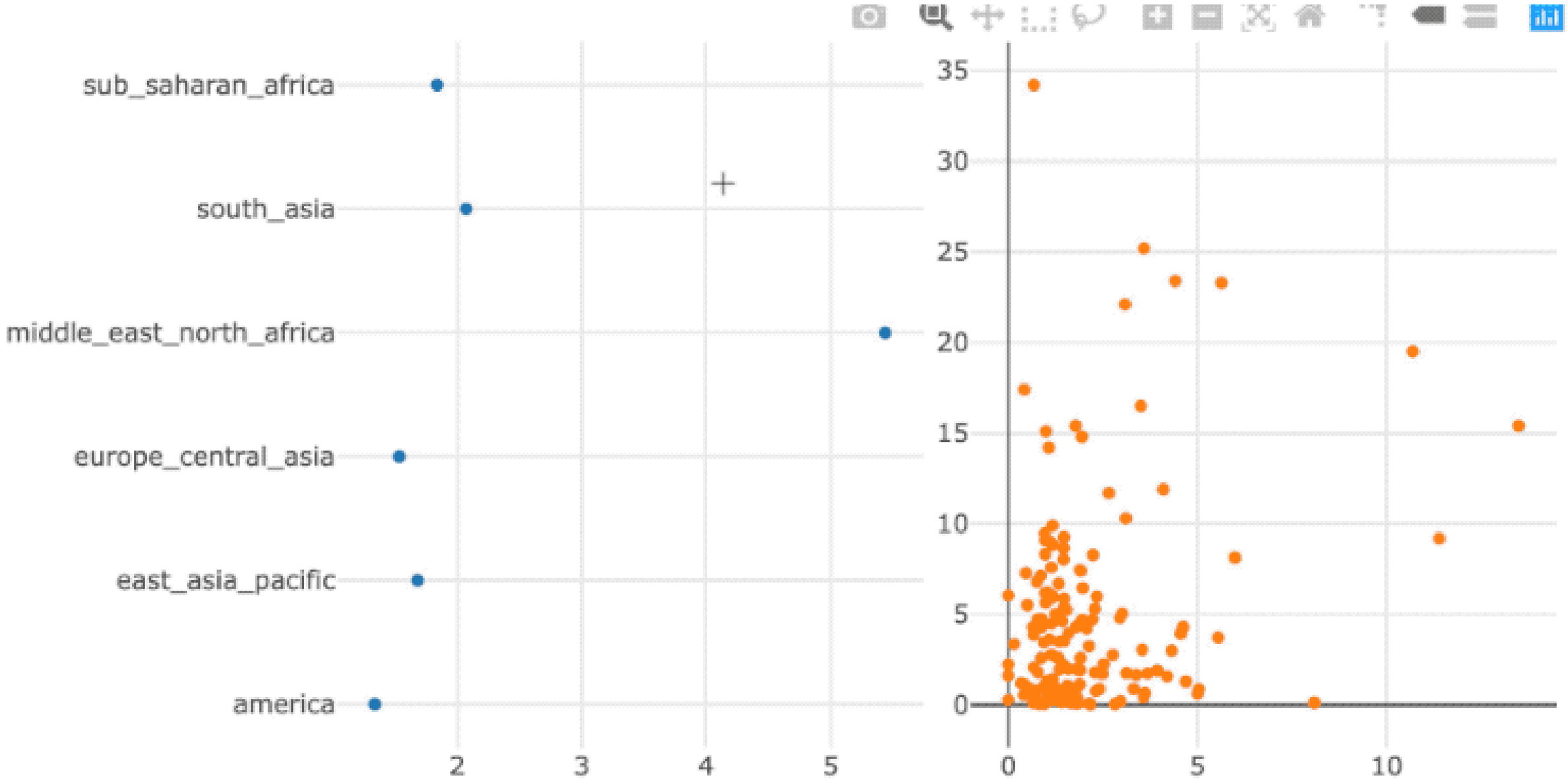

```
world_indicators %>%
  filter(year == 2014) %>%
  SharedData$new(~six_regions) %>%
  plot_ly(x=~military, y = ~co2, text = ~country) %>%
  add_markers()
```

Linking a summary and detailed view



Linking a summary and detailed view

```
shared_data <- world_indicators %>%
  filter(year == 2014) %>%
  SharedData$new(key = ~six_regions)
p1 <- shared_data %>%
  plot_ly() %>%
  group_by(six_regions) %>%
  summarize(avg.military = mean(military, na.rm = TRUE)) %>%
  add_markers(x = ~avg.military, y = ~six_regions)
p2 <- shared_data %>%
  plot_ly(x=~military, y = ~co2, text = ~country) %>%
  add_markers()
subplot(p1, p2) %>% hide_legend()
```


sub_saharan_africa
south_asia
middle_east_north_africa
europe_central_asia
east_asia_pacific
america
2
3
4
5
35
30
25
20
15
10
5
0
0
5
10

Let's practice!

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Selection strategies

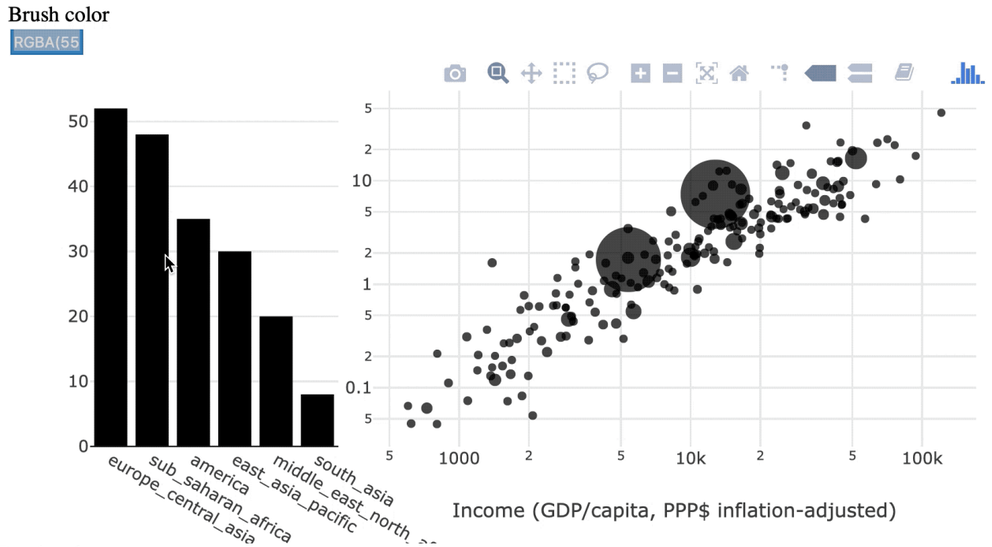
INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Adam Loy
Statistician, Carleton College

Two selection strategies

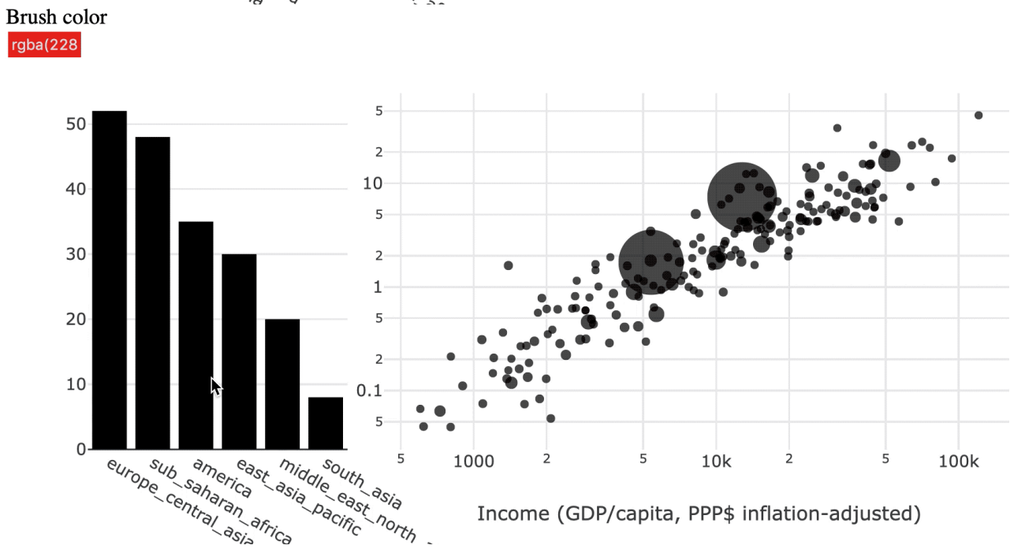
Transient selection

previously selected cases are forgotten

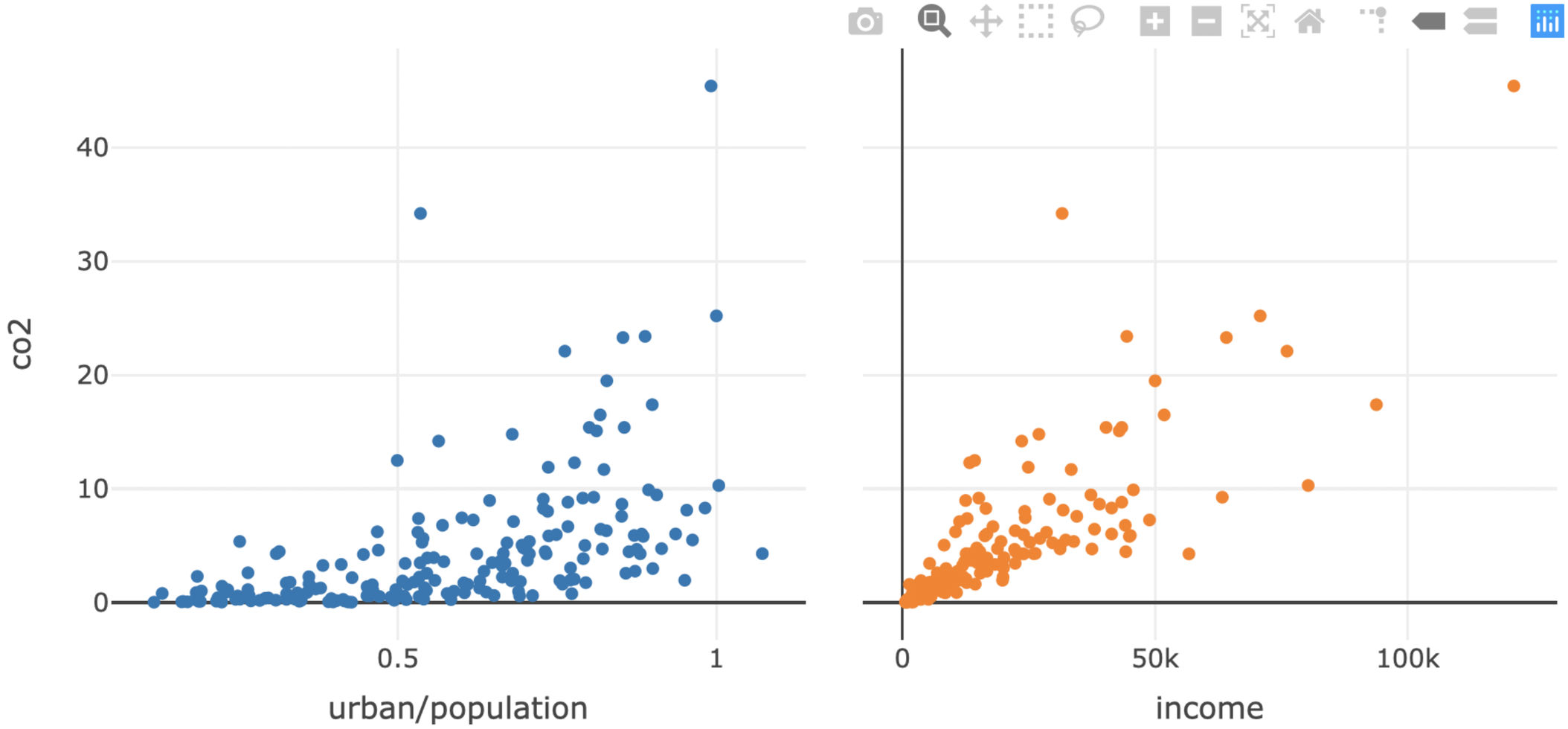


Persistent selection

selected cases accumulate



Example



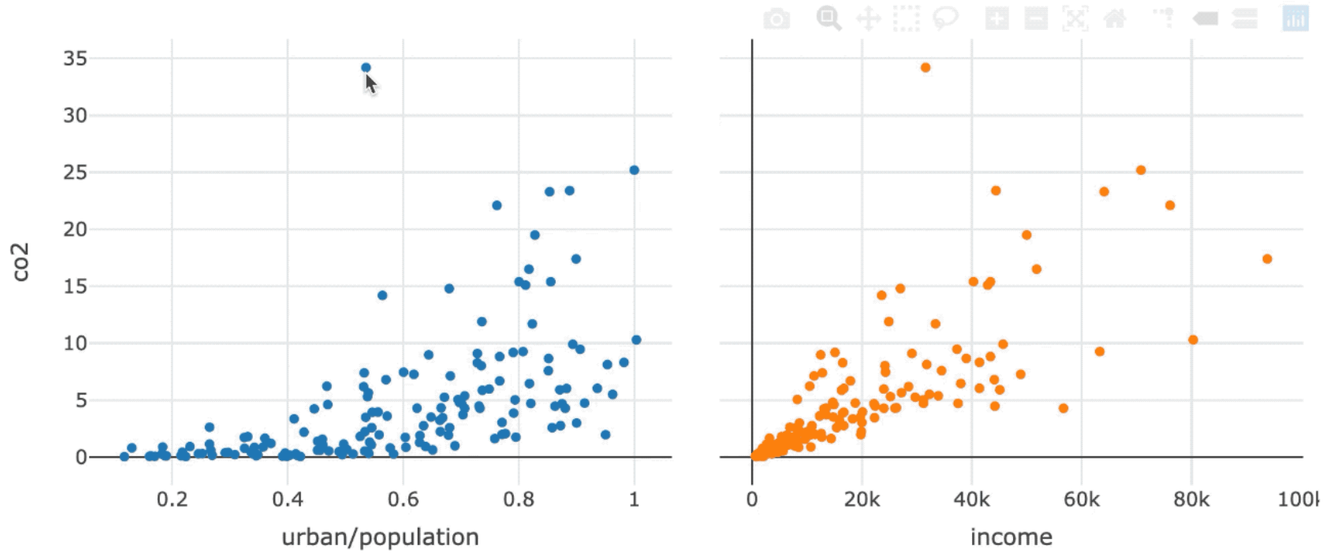
Generate the base chart

```
shared_data <- world2014 %>% SharedData$new()

p1 <- shared_data %>%
  plot_ly(x=~urban/population, y = ~co2, text = ~country) %>%
  add_markers()

p2 <- shared_data %>%
  plot_ly(x=~income, y = ~co2, text = ~country) %>%
  add_markers()

subplot(p1, p2, titleX = TRUE, shareY = TRUE) %>%
  hide_legend()
```

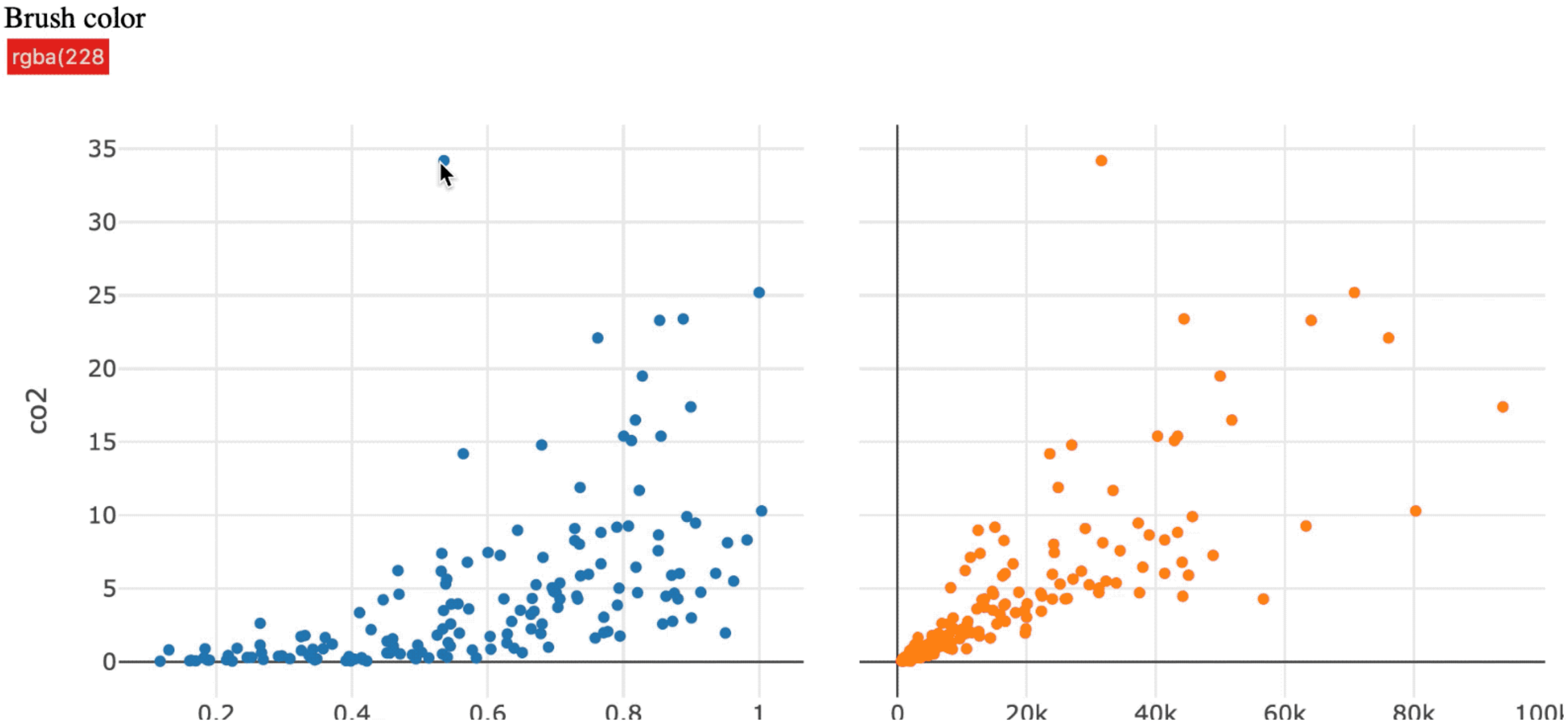
Persistent selection

Activate persistent selection via `highlight()`

```
subplot(p1, p2, titleX = TRUE, shareY = TRUE) %>%
  hide_legend() %>%
  highlight(persistent = TRUE)
```



Highlighting in color



Highlighting in color

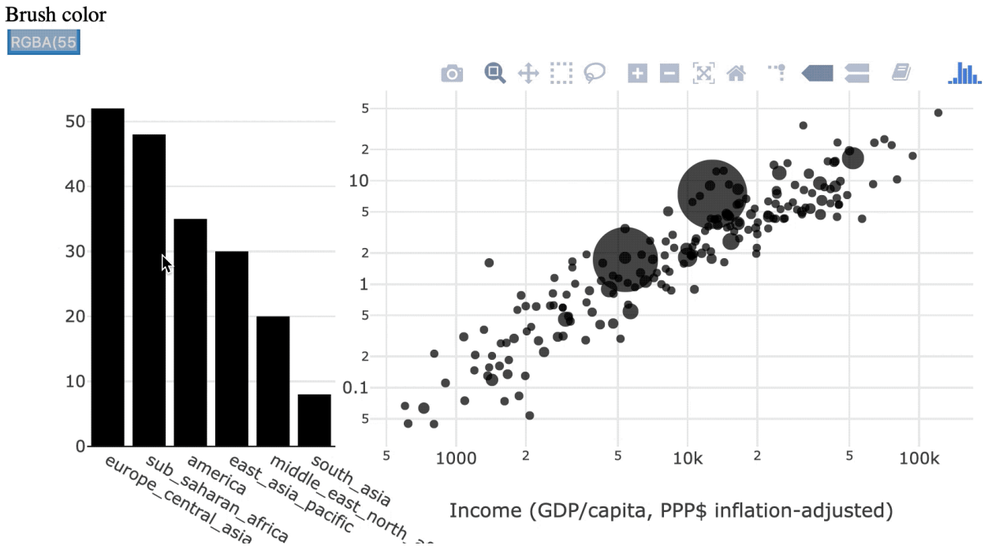
Add `dynamic = TRUE` to activate a color picker

```
subplot(p1, p2, titleX = TRUE, shareY = TRUE) %>%
  hide_legend() %>%
  highlight(persistent = TRUE, dynamic = TRUE)
```

Two manipulation types

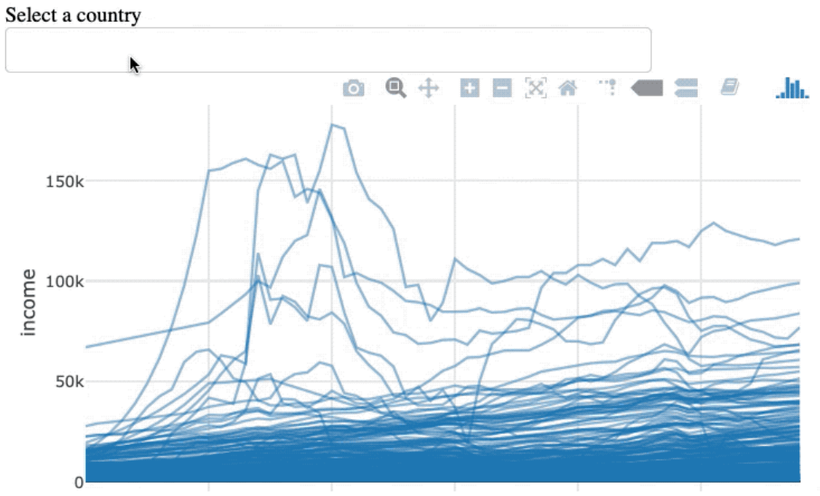
Direct manipulation

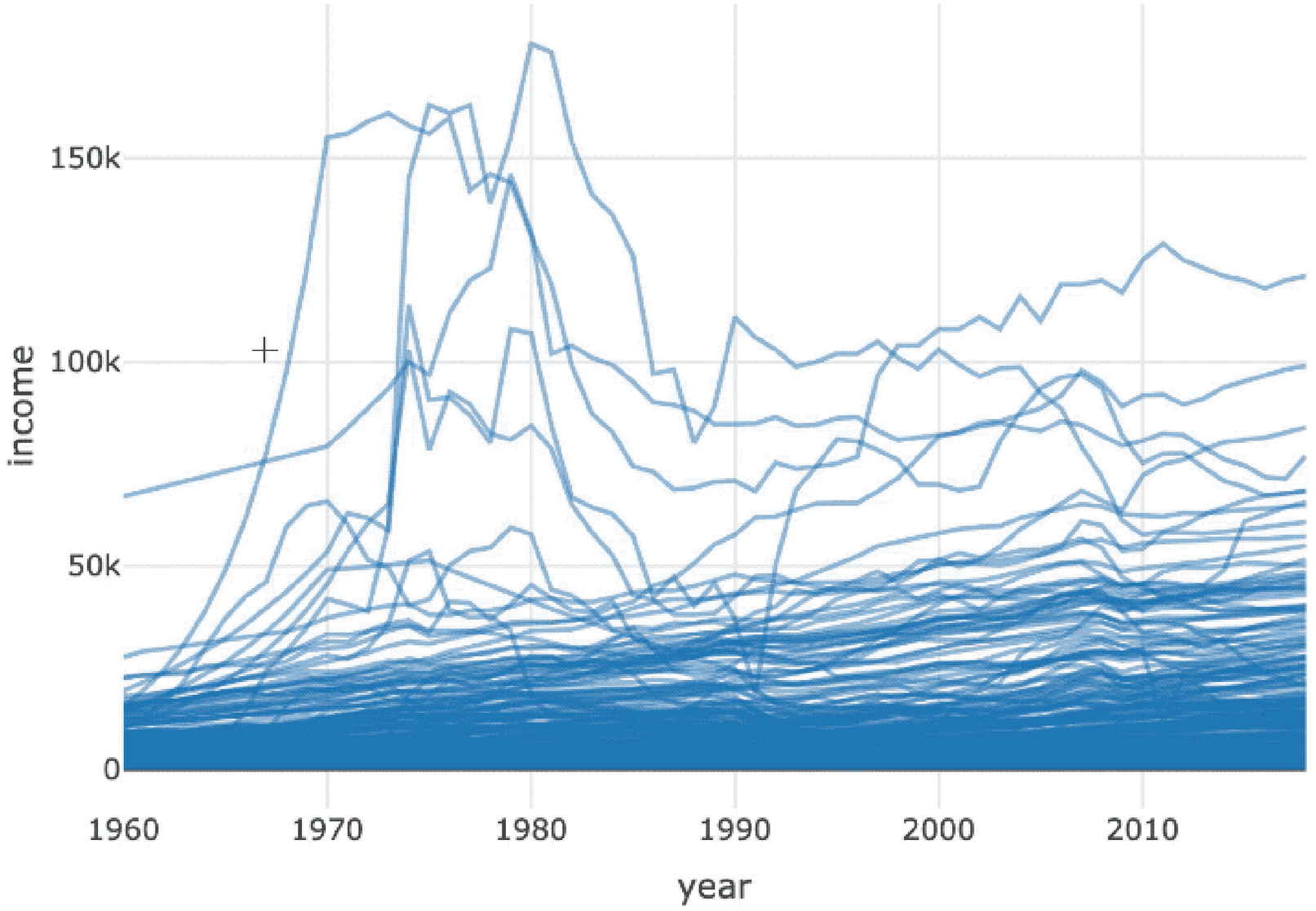
selection performed by interacting with the graphical elements



Indirect manipulation

selection performed via query outside of the chart



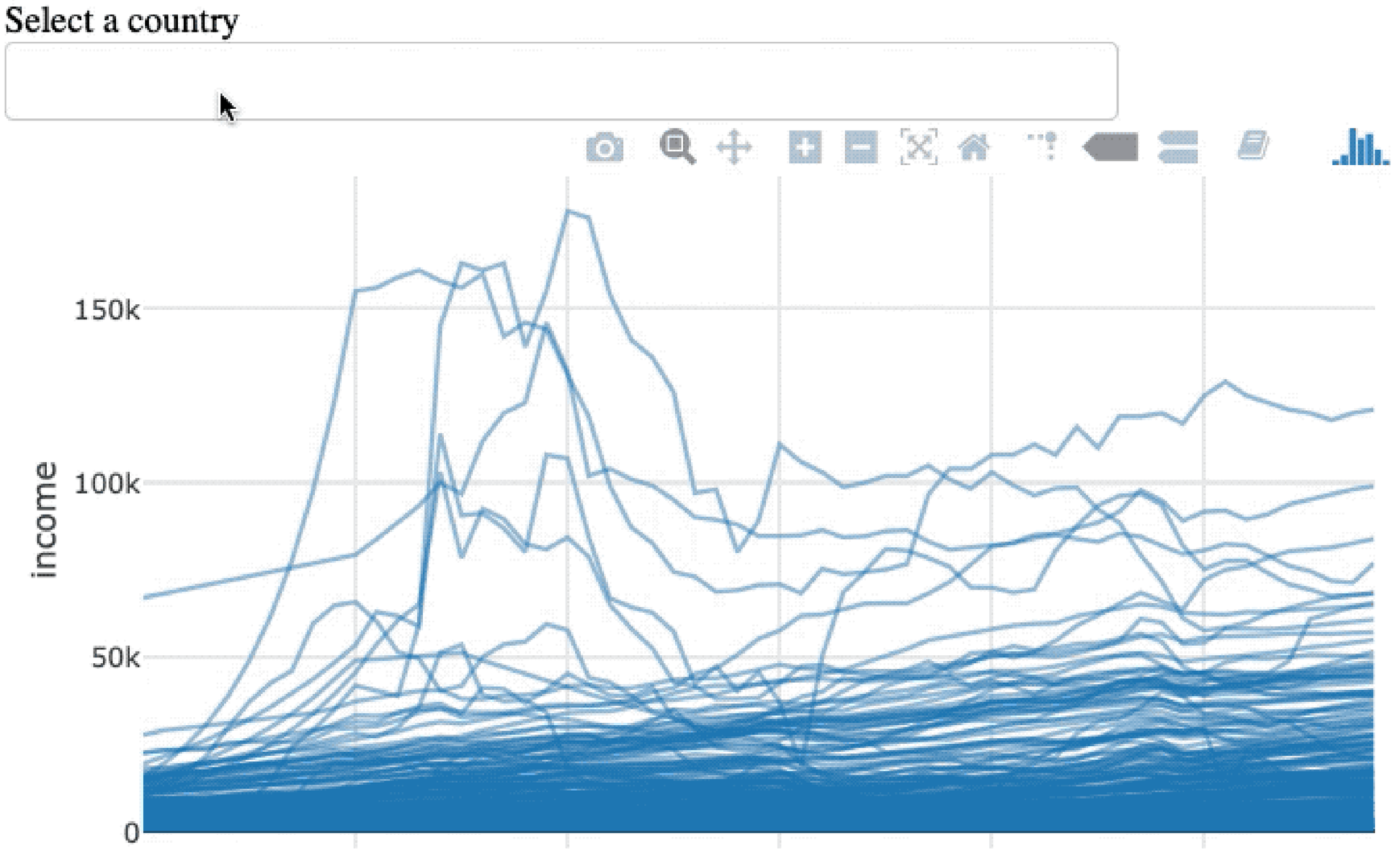

150k
100k
50k
0
income
1960
1970
1980
1990
2000
2010
year

Indirect manipulation

```
world_indicators %>%
  SharedData$new(key = ~country) %>%
  plot_ly(x = ~year, y = ~income, alpha = 0.5) %>%
  group_by(country) %>%
  add_lines()
```

Indirect manipulation

```
world_indicators %>%
  SharedData$new(key = ~country, group = "Select a country") %>%
  plot_ly(x = ~year, y = ~income, alpha = 0.5) %>%
  group_by(country) %>%
  add_lines() %>%
  highlight(selectize = TRUE)
```


Select a country
150k
100k
50k
0
income

Let's practice!

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

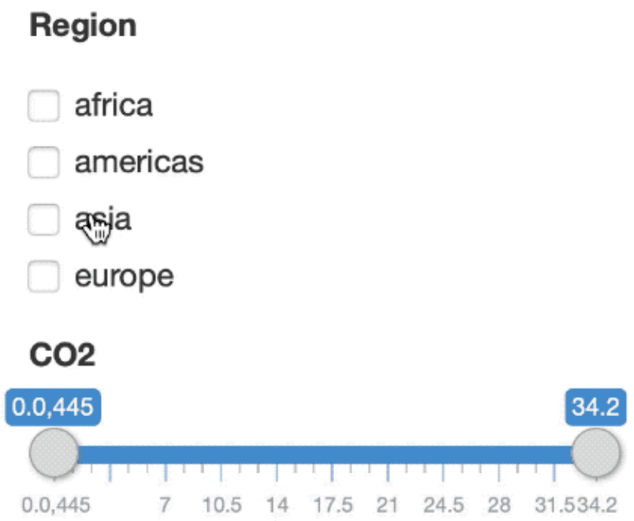
Making shinier charts

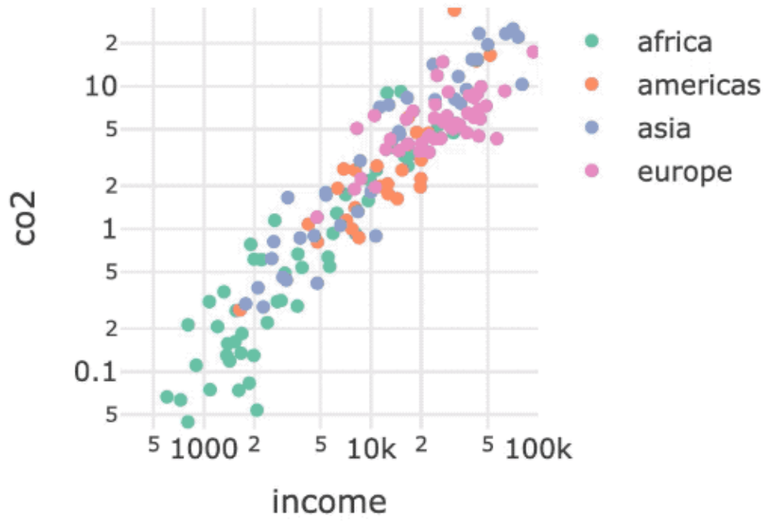
INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Adam Loy
Statistician, Carleton College

datacamp

Apps without shiny

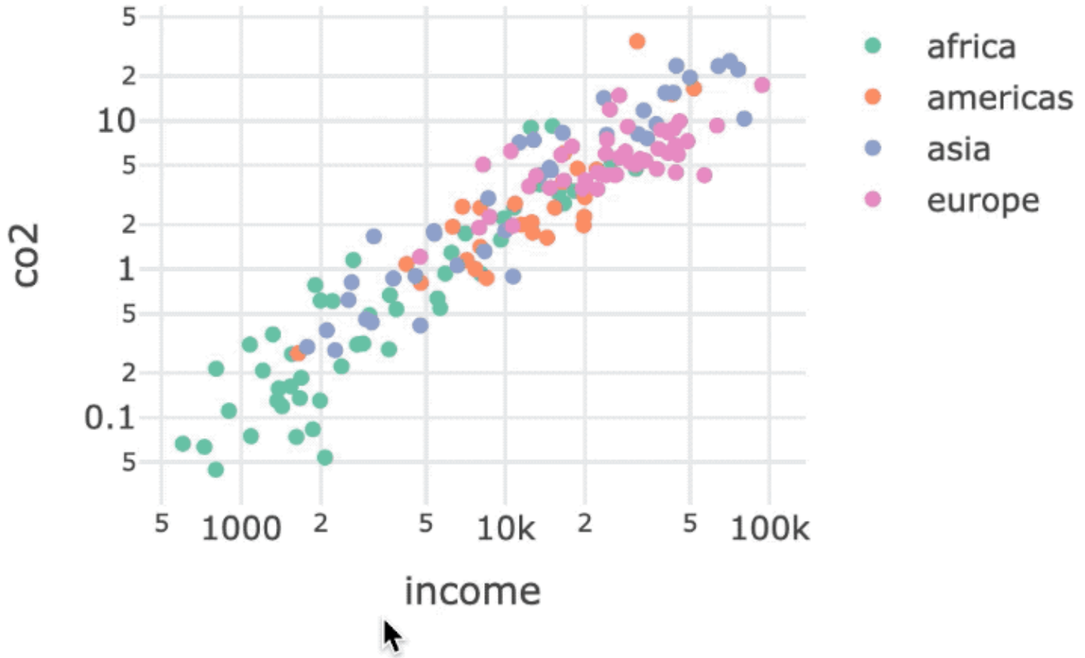


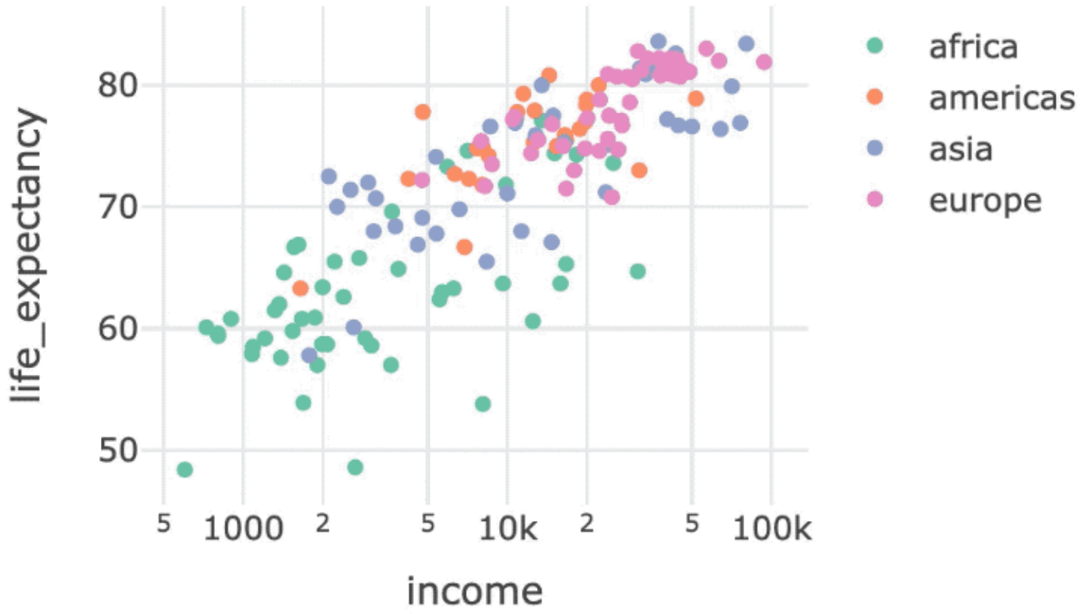


bscols() for column layouts

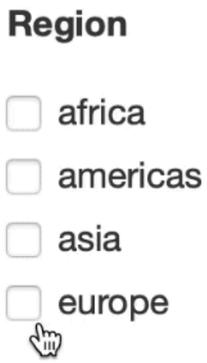
```
library(plotly)
library(crosstalk)
shared_data <- world2014 %>% SharedData$new()
p1 <- shared_data %>%
  plot_ly(x=~income, y = ~co2, color = ~four_regions) %>%
  add_markers() %>%
  layout(xaxis = list(type = "log"), yaxis = list(type = "log"))
p2 <- shared_data %>%
  plot_ly(x=~income, y = ~life_expectancy, color = ~four_regions) %>%
  add_markers() %>%
  layout(xaxis = list(type = "log")))
bscols(p1, p2)
```

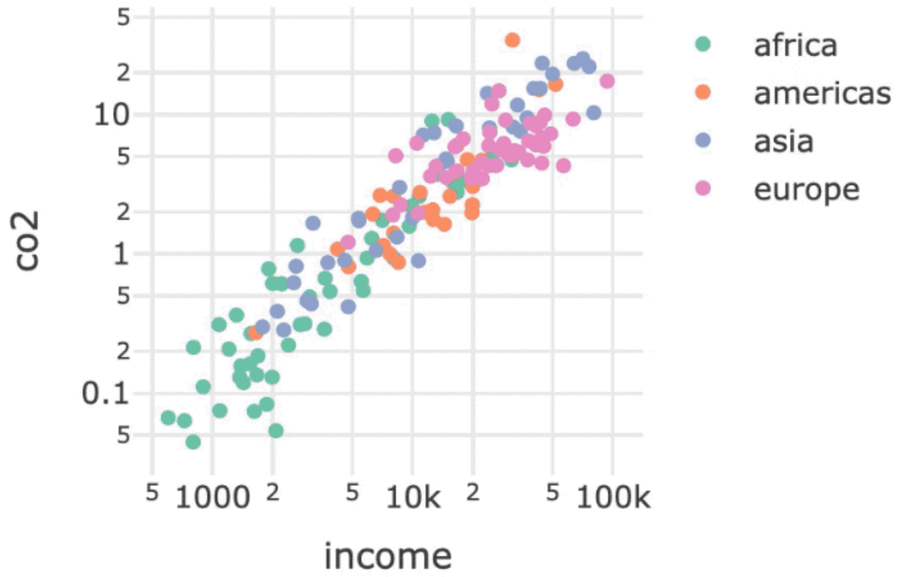
bscols() for column layouts





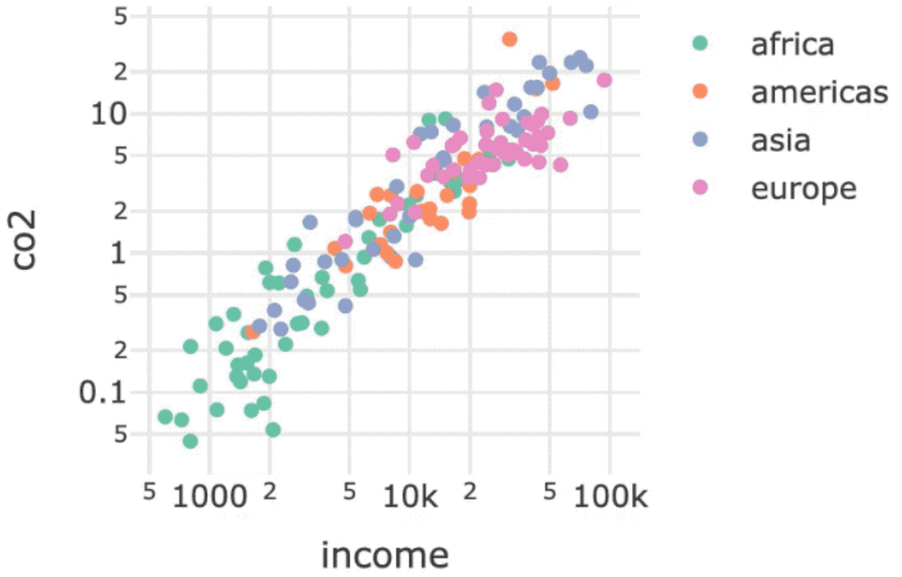
Adding filters: Checkboxes





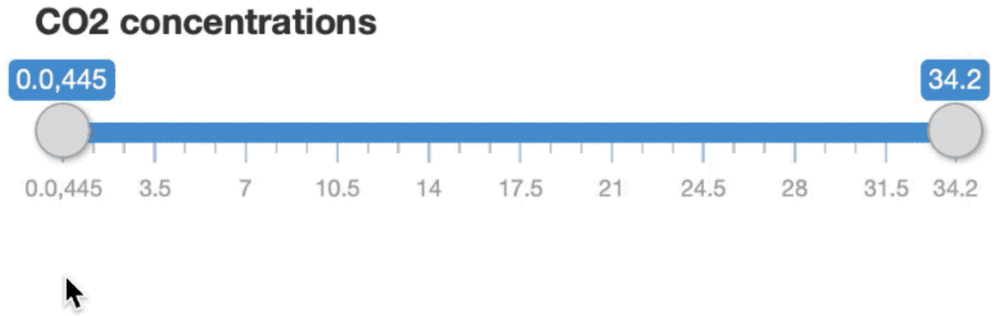

```
bscols(filter_checkbox(id = "four_regions", label = "Region",
                       sharedData = shared_data, group = ~four_regions),
       p1)
```

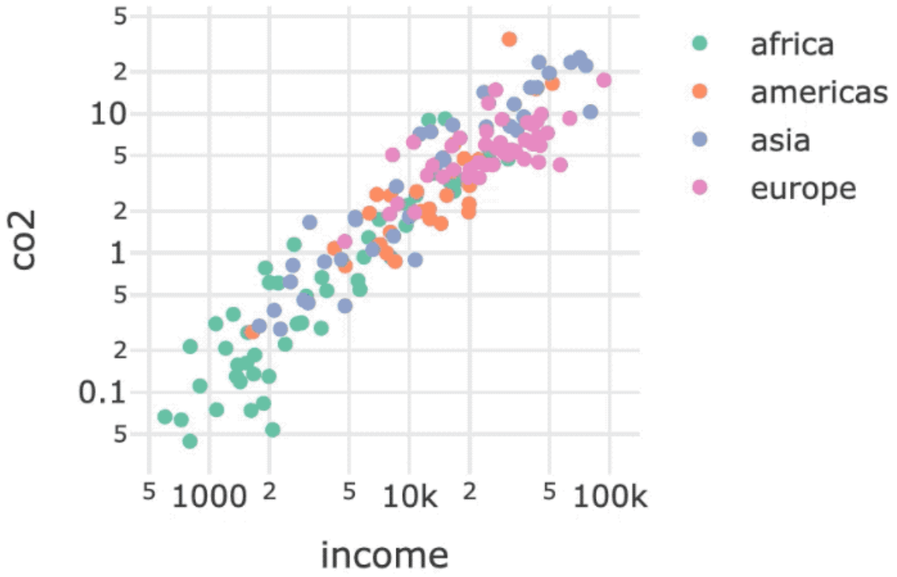
Adding filters: Select box




```
bscols(filter_select(id = "four_regions", label = "Region",
                     sharedData = shared_data, group = ~four_regions),
       p1)
```

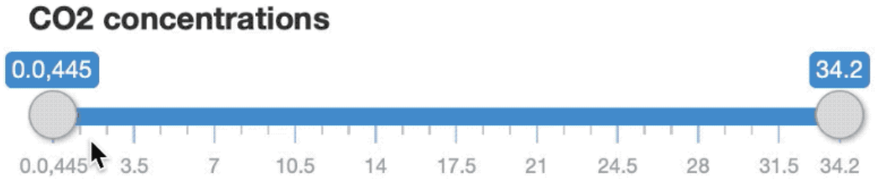
Adding filters: Sliders

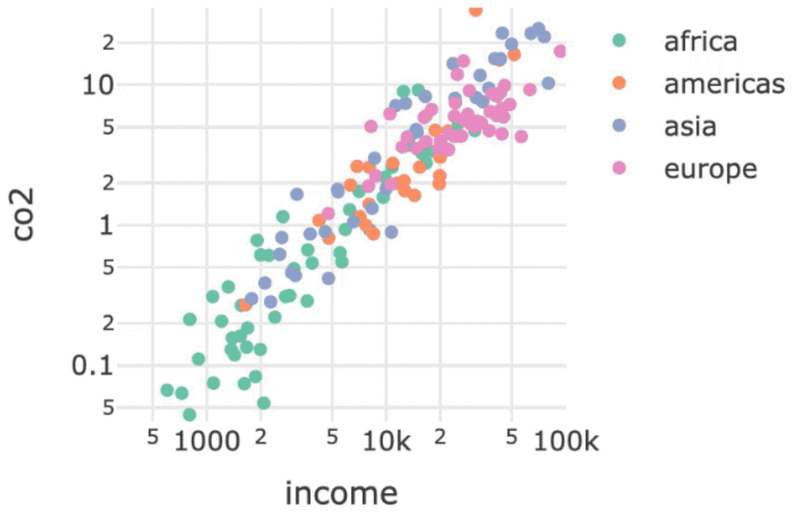





```
bscols(filter_slider(id = "co2", label = "CO2 concentrations",
                     sharedData = shared_data, column = ~co2),
       p1)
```

Fixing the range of your axes



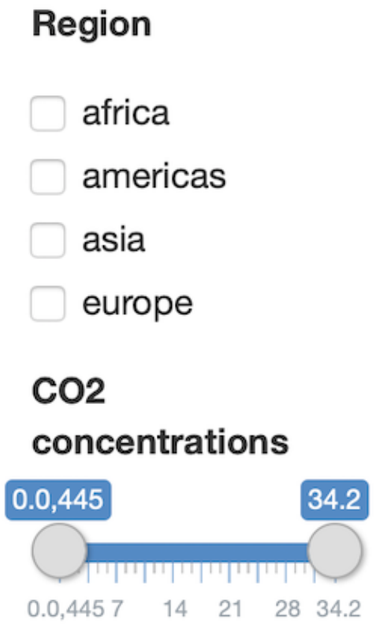


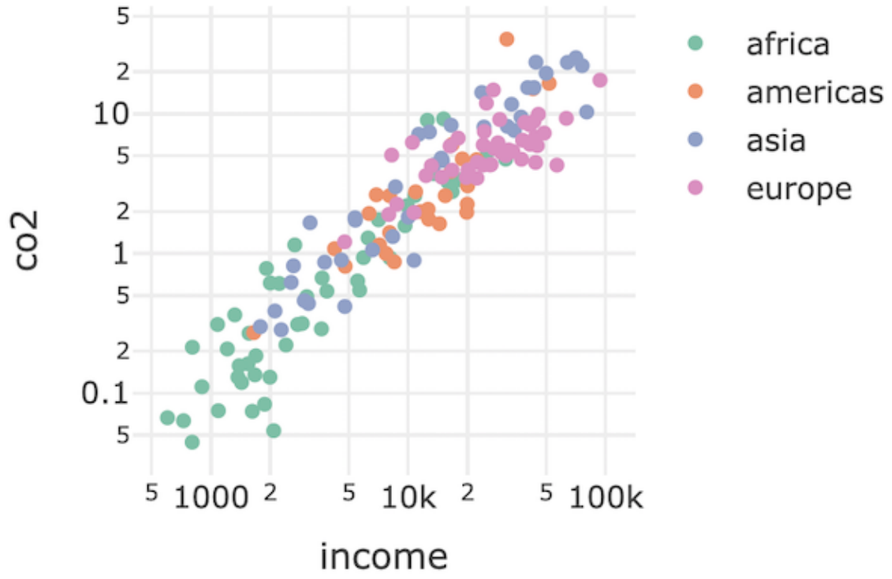

```
bscols(filter_slider(id = "co2", label = "CO2 concentrations",
                     sharedData = shared_data, column = ~co2),
       p1 %>% layout(xaxis = list(range = c(2.5, 5)),
                     yaxis = list(range = c(-1.4, 1.55)))
)
```

Putting the pieces together

```
bscols(widths = c(2, 5, 5),
  list(
    filter_checkbox(
      id = "four_regions", label = "Region",
      sharedData = shared_data, group = ~four_regions
    ),
    filter_slider(
      id = "co2", label = "CO2 concentrations",
      sharedData = shared_data, column = ~co2
    )
  ),
  p1,
  p2
)
```

Putting the pieces together





Let's practice!

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R