Introduction to animation

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

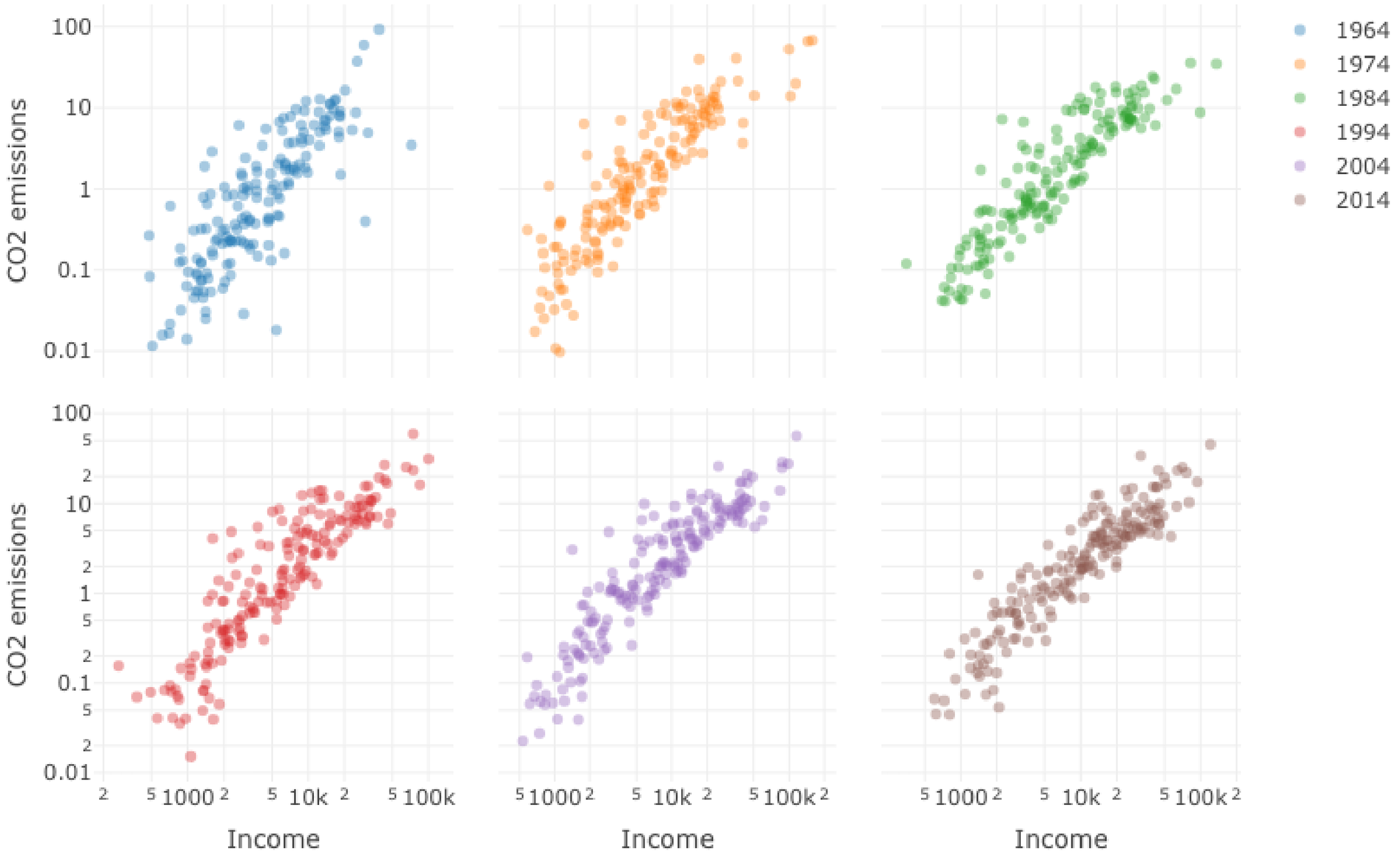
Adam Loy
Statistician, Carleton College

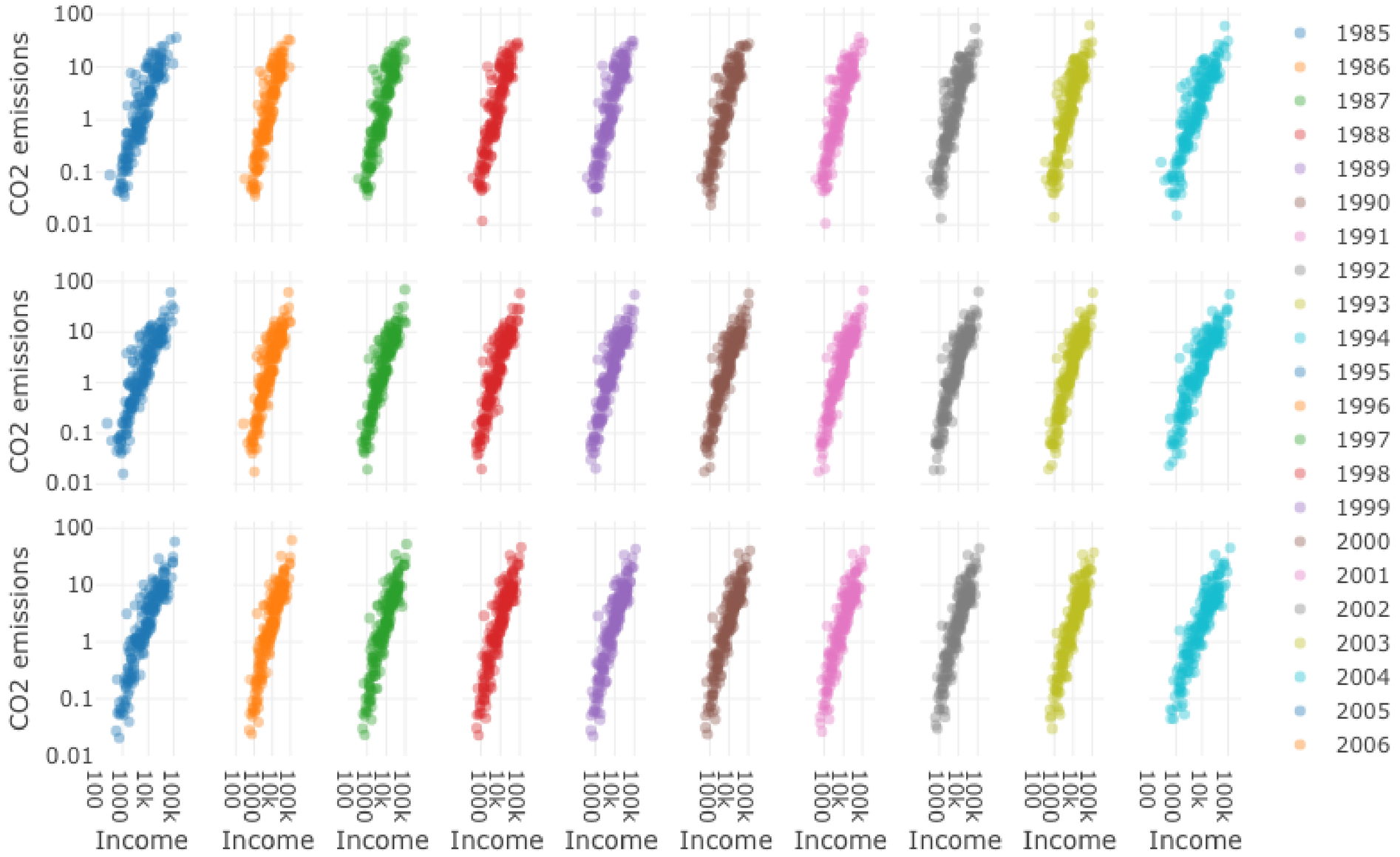
CO2 emissions and income

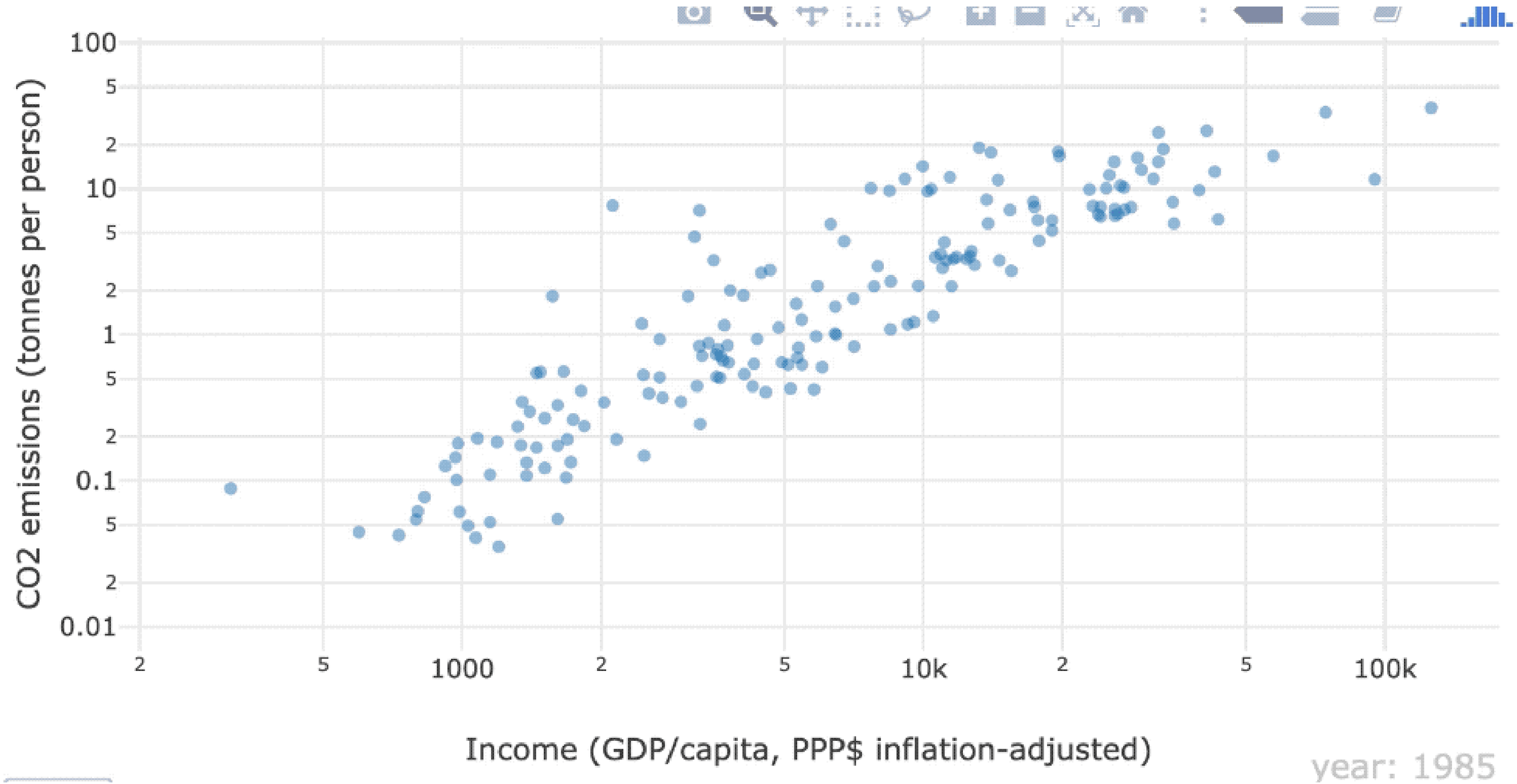
Question: Has the relationship between carbon dioxide emissions and income changed over time?

```
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
# … with 1.138e+04 more rows, and 2 more variables: eight_regions <chr>, six_regions <ch
```

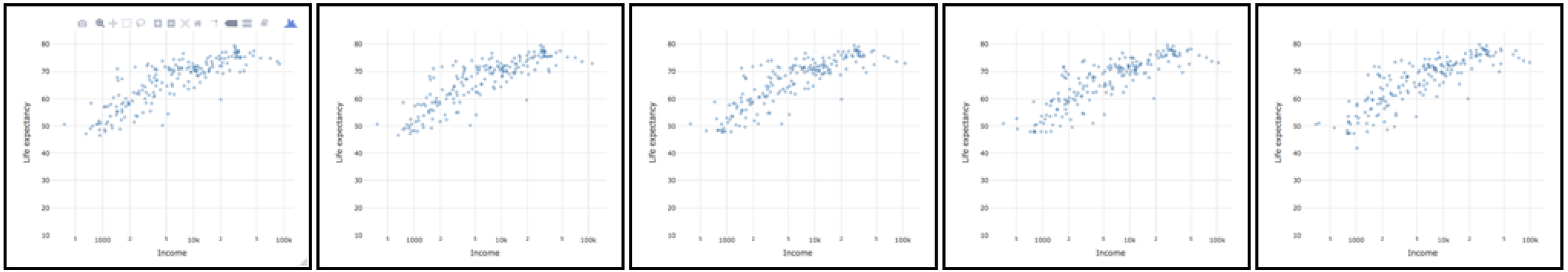

CO2 emissions
100
10
1
0.1
0.01
5
2
1000
10k
100k
Income
1964
1974
1984
1994
2004
2014


CO2 emissions
100
10
1
0.1
0.01
100
1000
10k
100k
Income
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006


CO2 emissions (tonnes per person)
100
5
2
10
5
2
1
5
2
0.1
5
2
0.01
2
5
1000
2
5
10k
2
5
100k
Income (GDP/capita, PPP$ inflation-adjusted)
year: 1985

Keyframe animation

Frame = plot at one time point

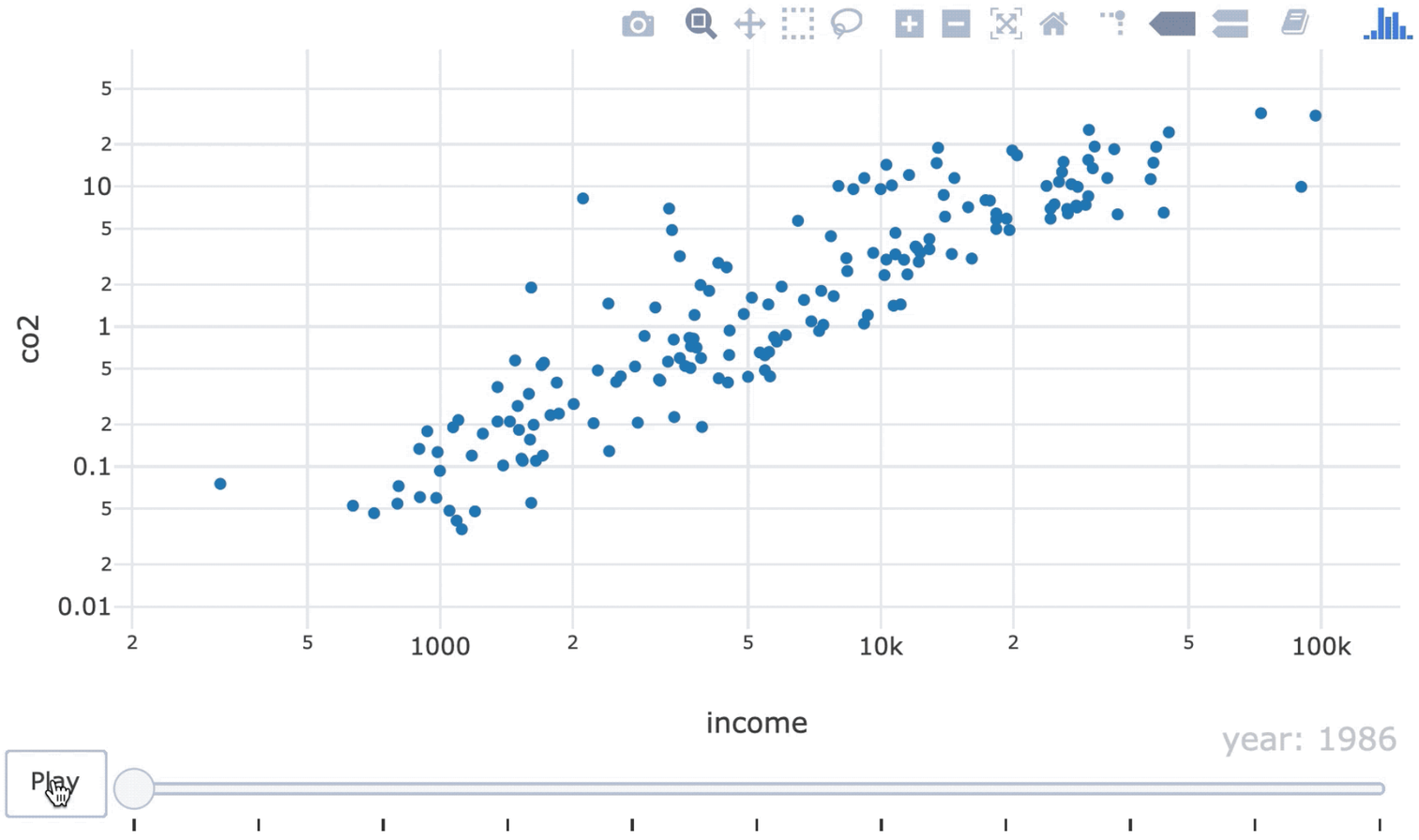


The frame aesthetic

```
world_indicators %>%
  plot_ly(x = ~income, y = ~co2) %>%
  add_markers(frame = ~year, showlegend = FALSE) %>%
  layout(xaxis = list(type = "log"), yaxis = list(type = "log"))
```

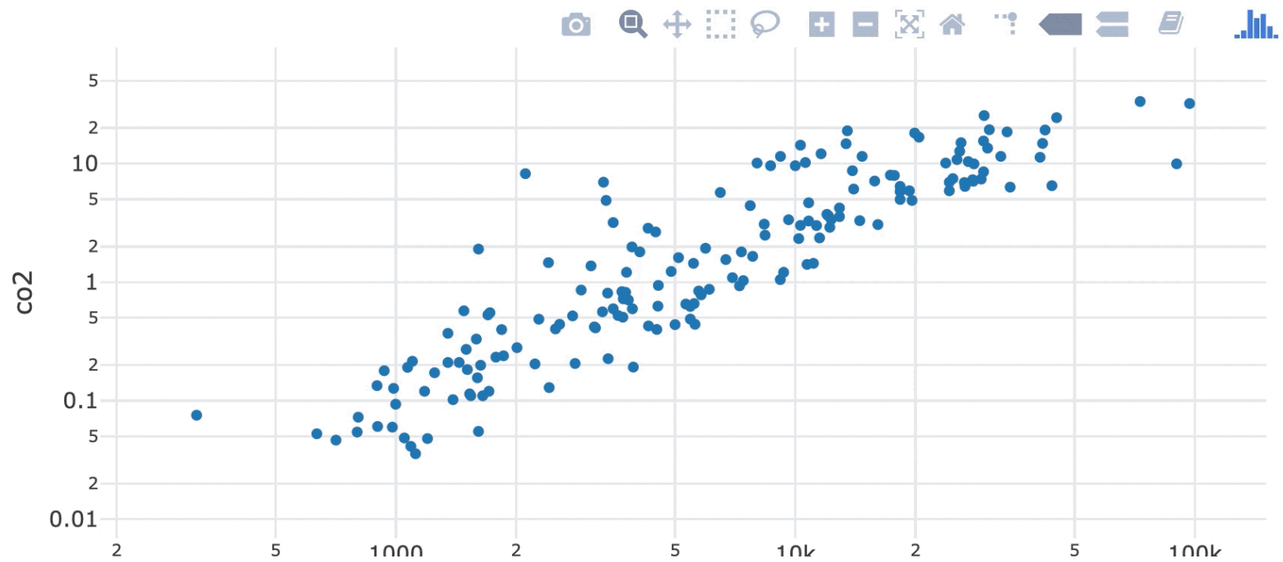
Object constancy

A graphical element (e.g. glyph) should represent a particular data point (e.g. Belgium)



The ids aesthetic

```
world_indicators %>%
  plot_ly(x = ~income, y = ~co2) %>%
  add_markers(frame = ~year, ids = ~country, showlegend = FALSE) %>%
  layout(xaxis = list(type = "log"), yaxis = list(type = "log"))
```



Let's practice!

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Polishing animations

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Adam Loy
Statistician, Carleton College

Customizations

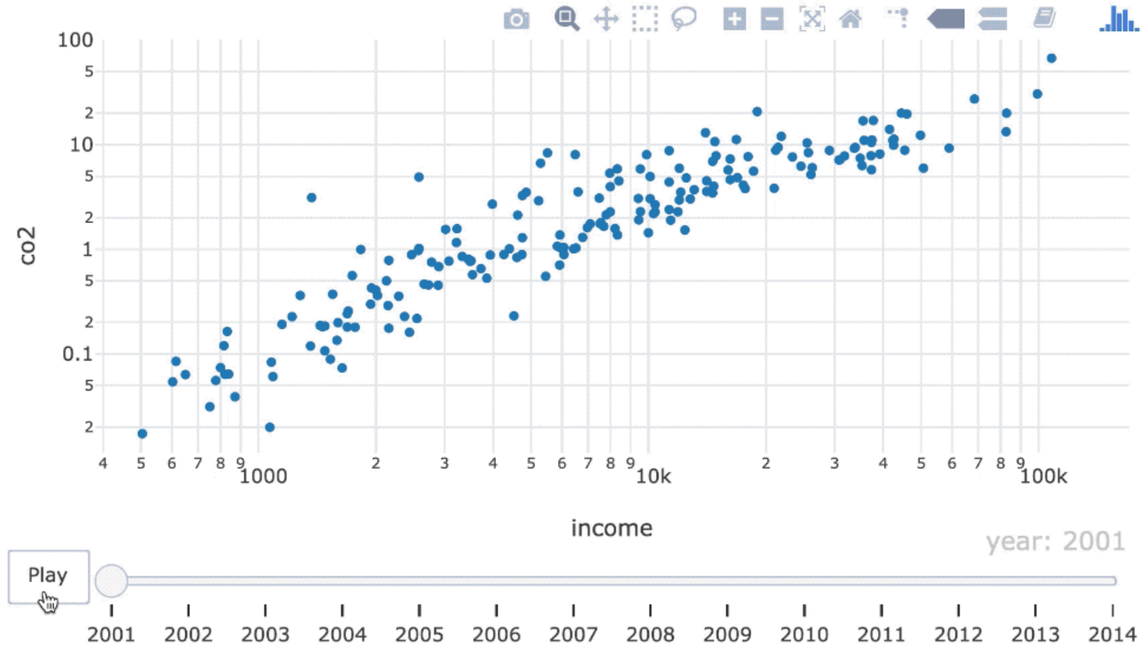
Animation options

- Time between frames
- Frame transitions
- Slider appearance

Plotting options

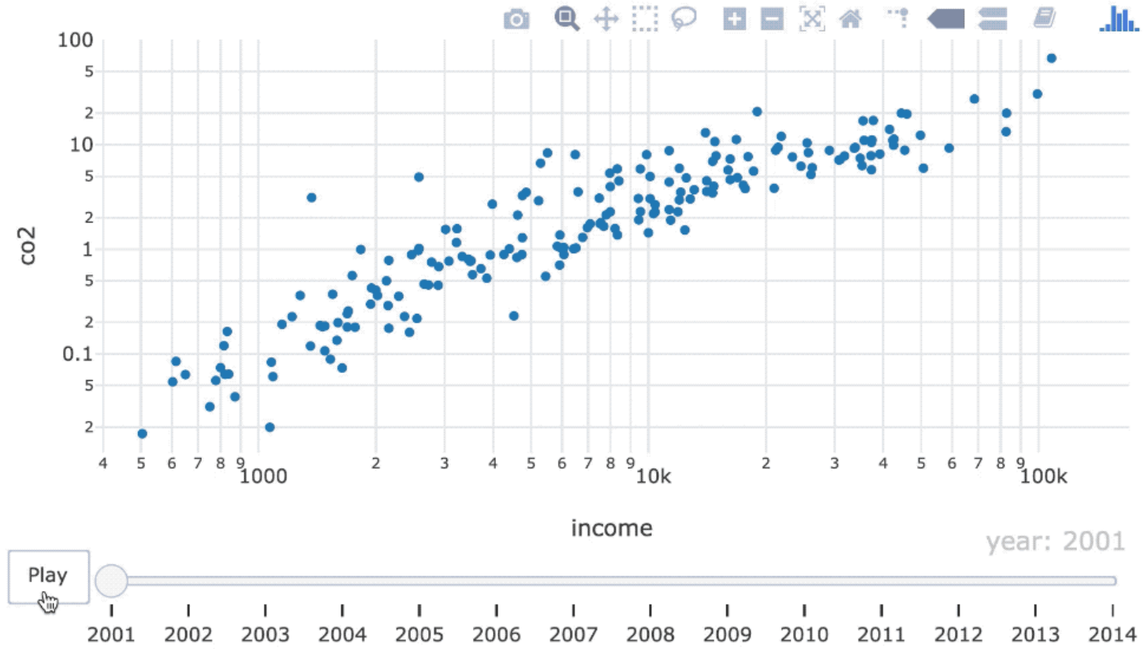
- Glyph color, shape, size
- Axis labels, transformations
- Tools tips (e.g. hover text)

Animation options



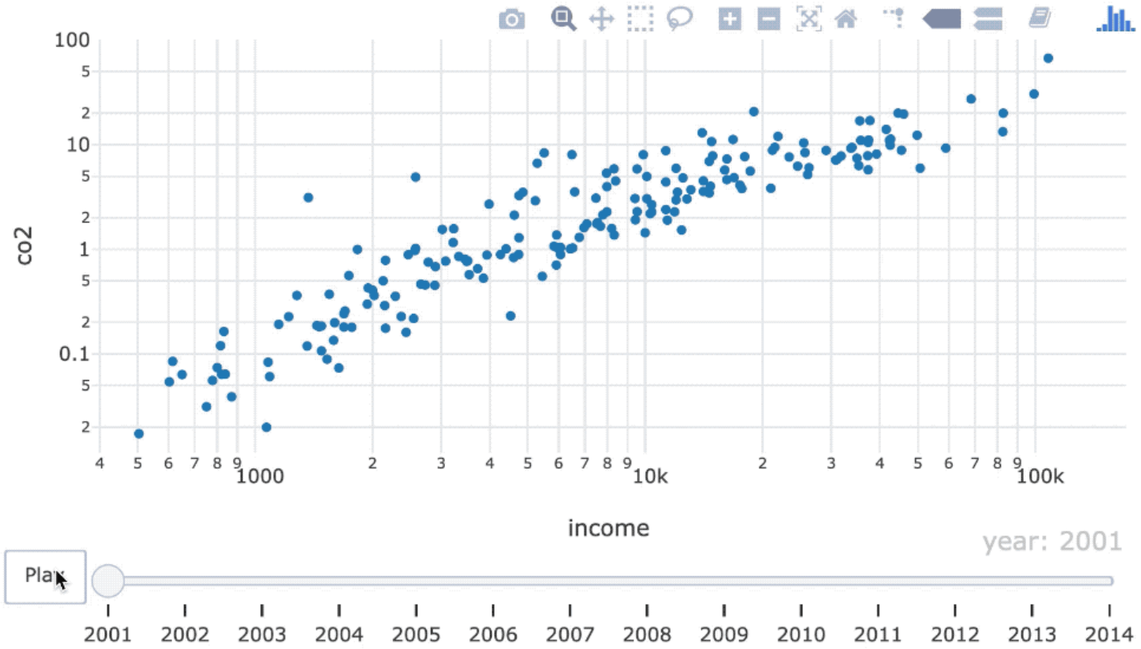

```
ani <- world_indicators %>%
  plot_ly(x = ~income, y = ~co2) %>%
  add_markers(frame = ~year,
              ids = ~country,
              showlegend = FALSE) %>%
  layout(xaxis = list(type = "log"),
         yaxis = list(type = "log"))
```

Animation options



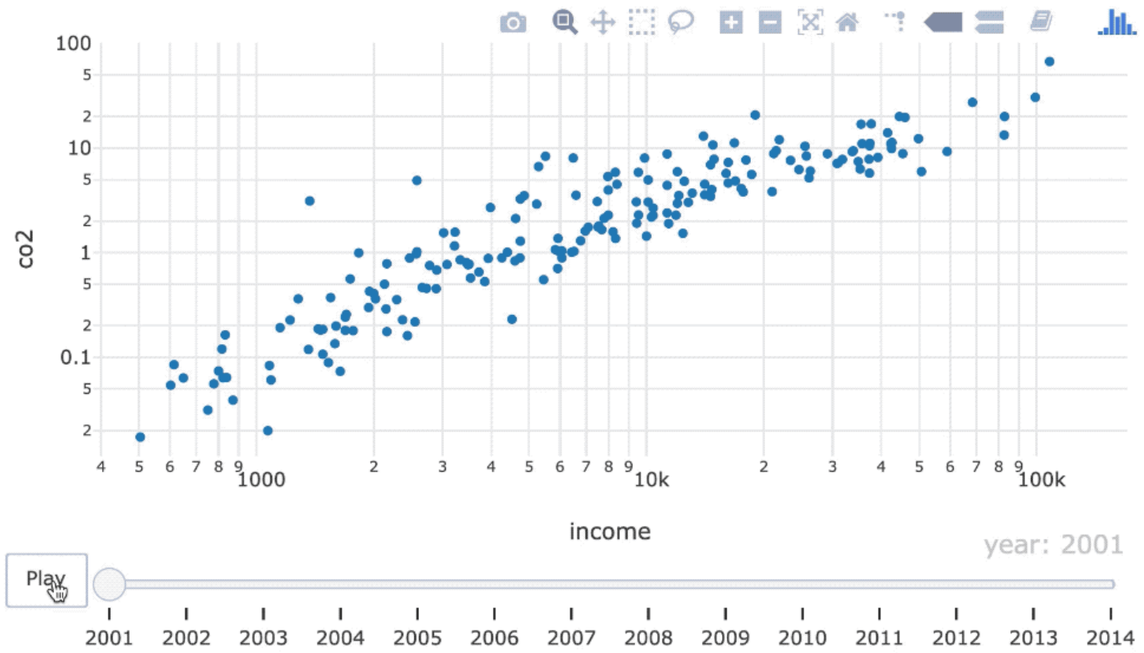

```
ani %>%
  animation_opts(
      frame = 500,
      transition = frame,
      easing = "linear",
      redraw = TRUE
    )
```

Speeding up



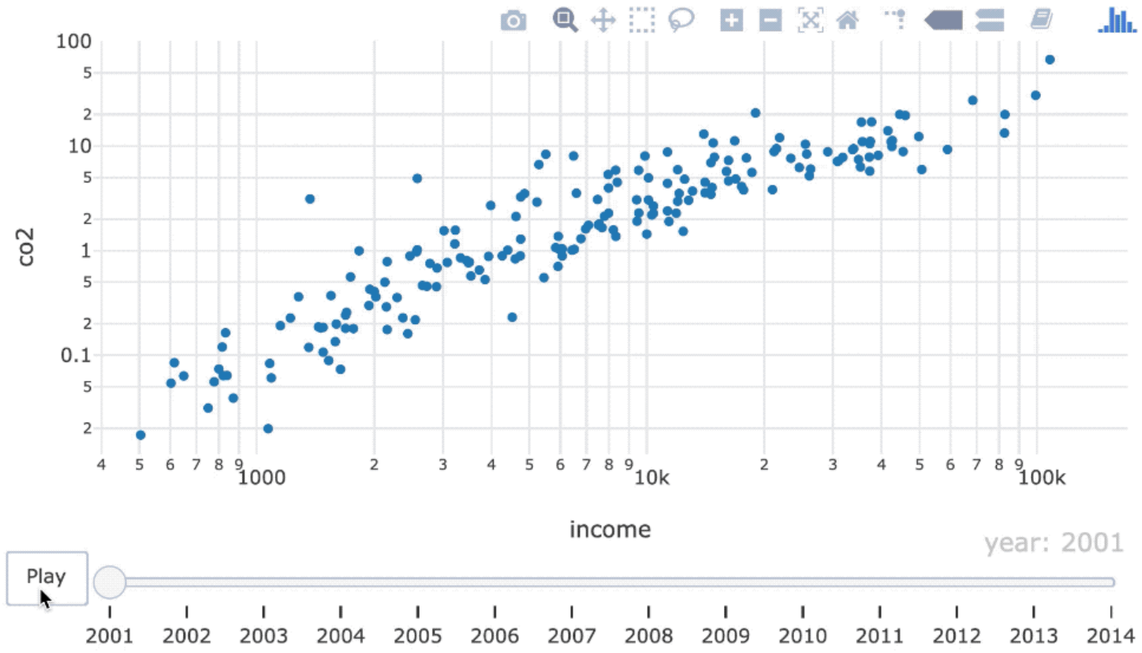

```
ani %>%
  animation_opts(frame = 300)
```

Pausing between frames




```
ani %>%
  animation_opts(
     frame = 700,
     transition = 350
  )
```

Bouncing points

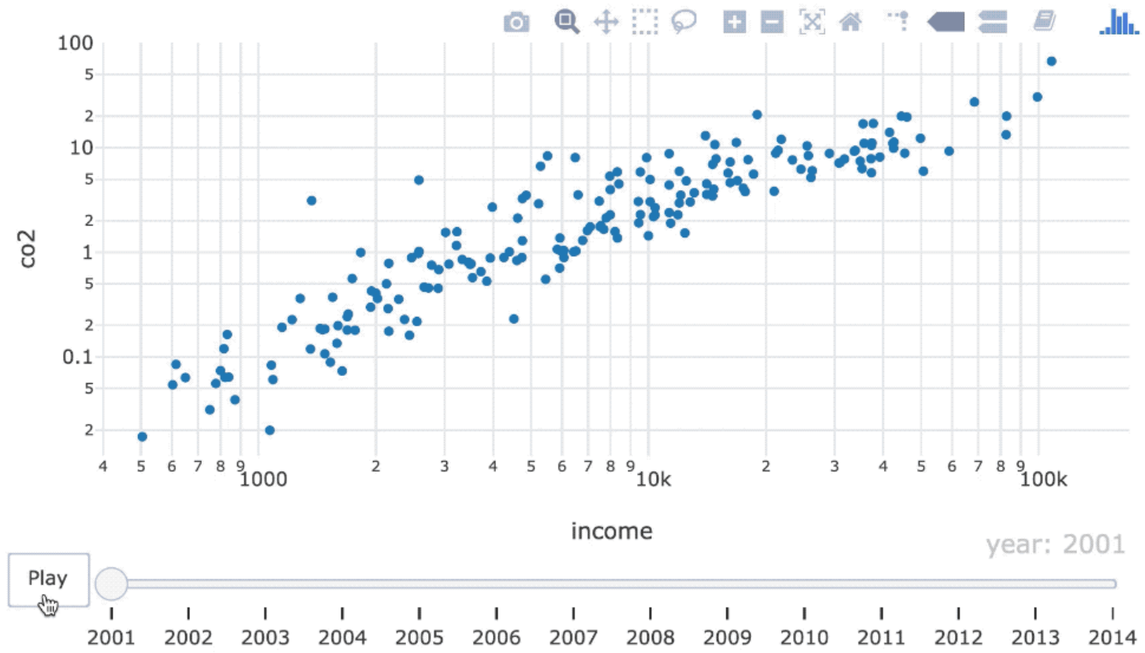



```
ani %>%
  animation_opts(easing = "bounce")
```

Basic easing options:

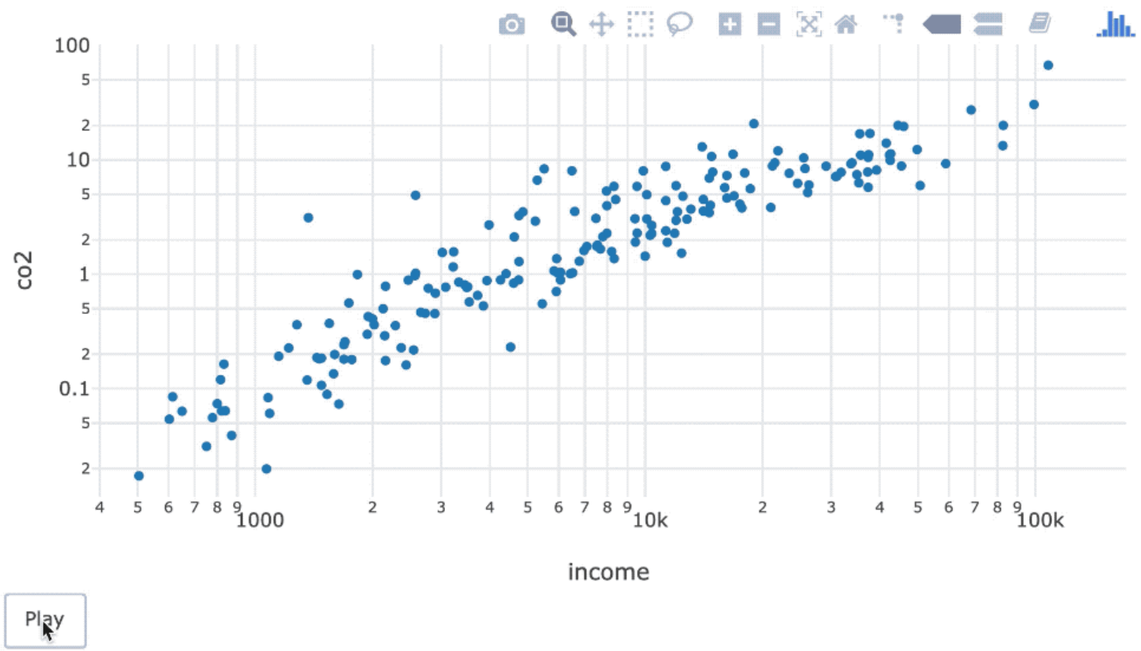
`"linear"` , `"quad"` , `"cubic"` , `"sin"` , `"exp"` , `"circle"` , `"elastic"` , `"back"` , `"bounce"`

Slider options



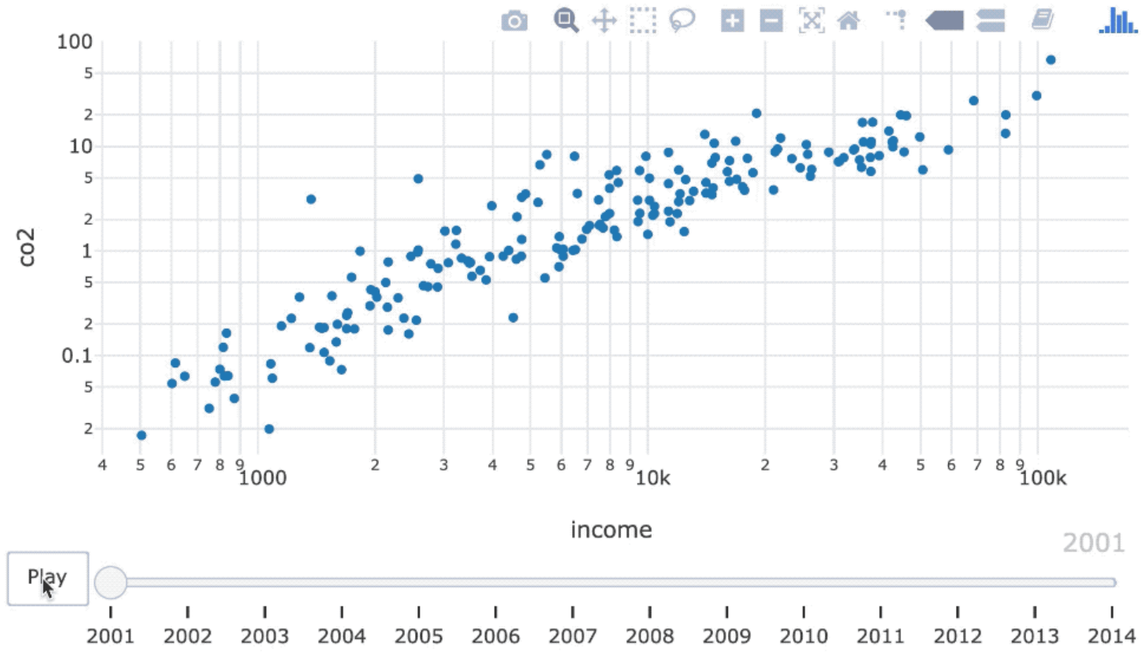

```
ani
```

Removing the slider



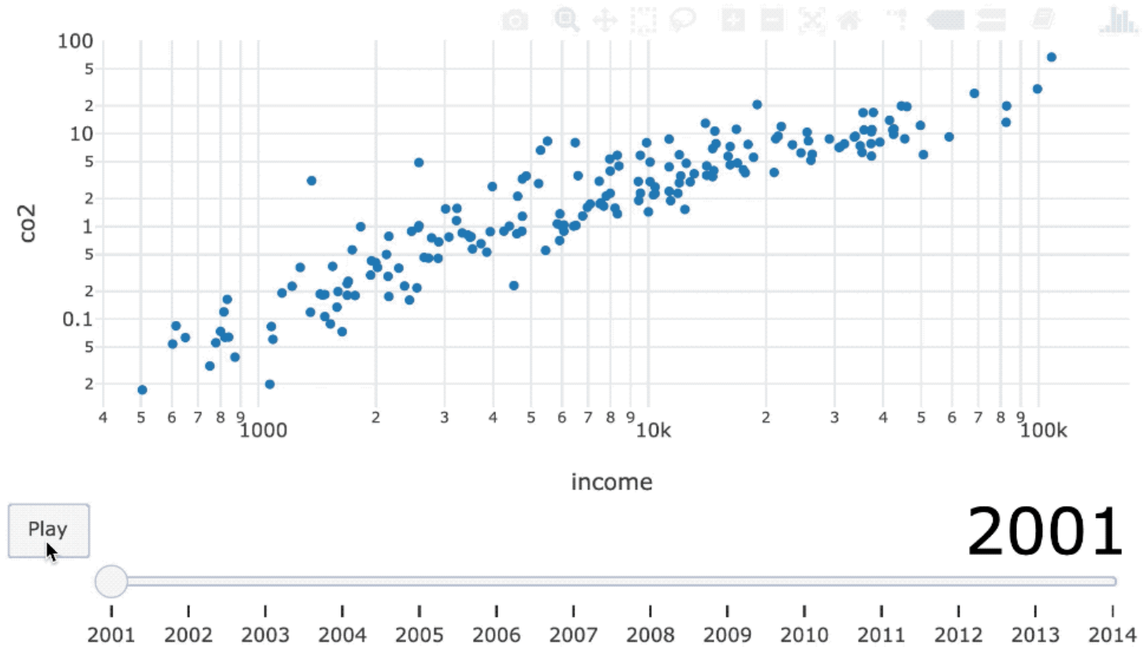

```
ani %>%
  animation_slider(hide = TRUE)
```

Editing slider text




```
ani %>%
  animation_slider(
     currentvalue = list(prefix=NULL)
  )
```

Editing slider text




```
ani %>%
  animation_slider(
     currentvalue = list(prefix=NULL,
        font = list(
           color = "black",
           size = 40
        )
     )
  )
```

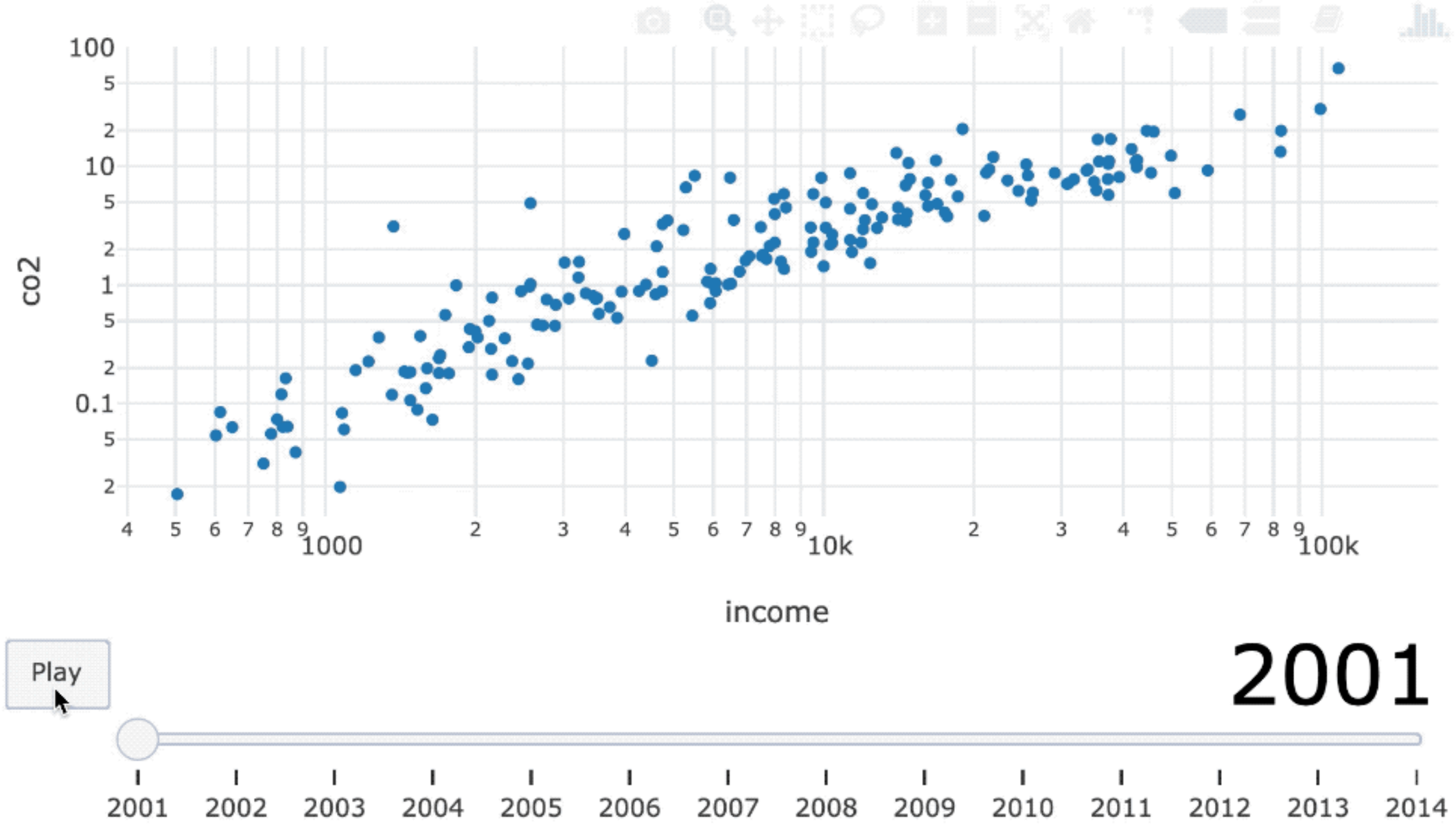
Let's practice!

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Adding layers

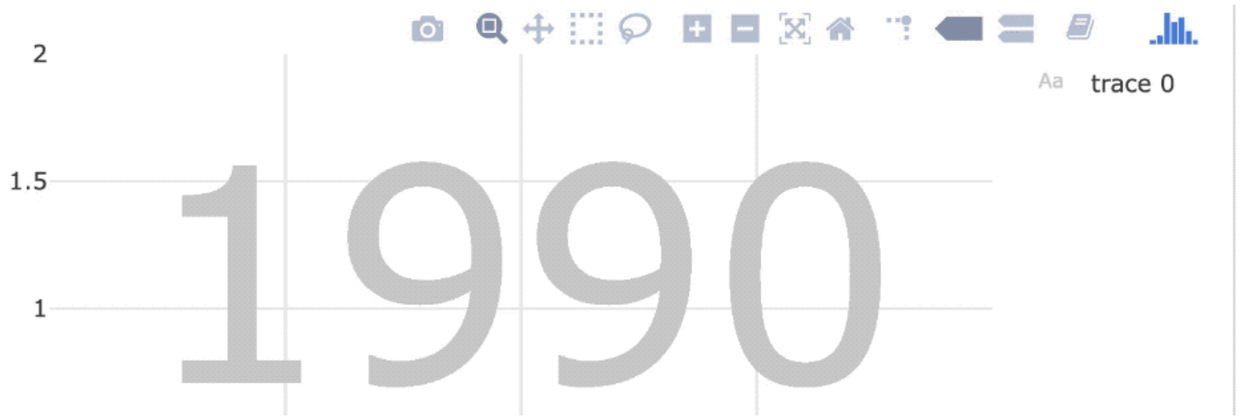
INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Adam Loy
Statistician, Carleton College


co2
100
5
2
10
5
2
1
5
2
0.1
5
2
4 5 6 7 8 9
1000
2 3 4 5 6 7 8 9
10k
2 3 4 5 6 7 8 9
100k
income
Play
2001
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014

Text layer

```
world_indicators %>%
  plot_ly(x = ~income, y = ~co2) %>%
  add_text(
    x = 6500, y = 1, text = ~year, frame = ~year,
    textfont = list(size = 150, color = toRGB("gray80"))
  )
```

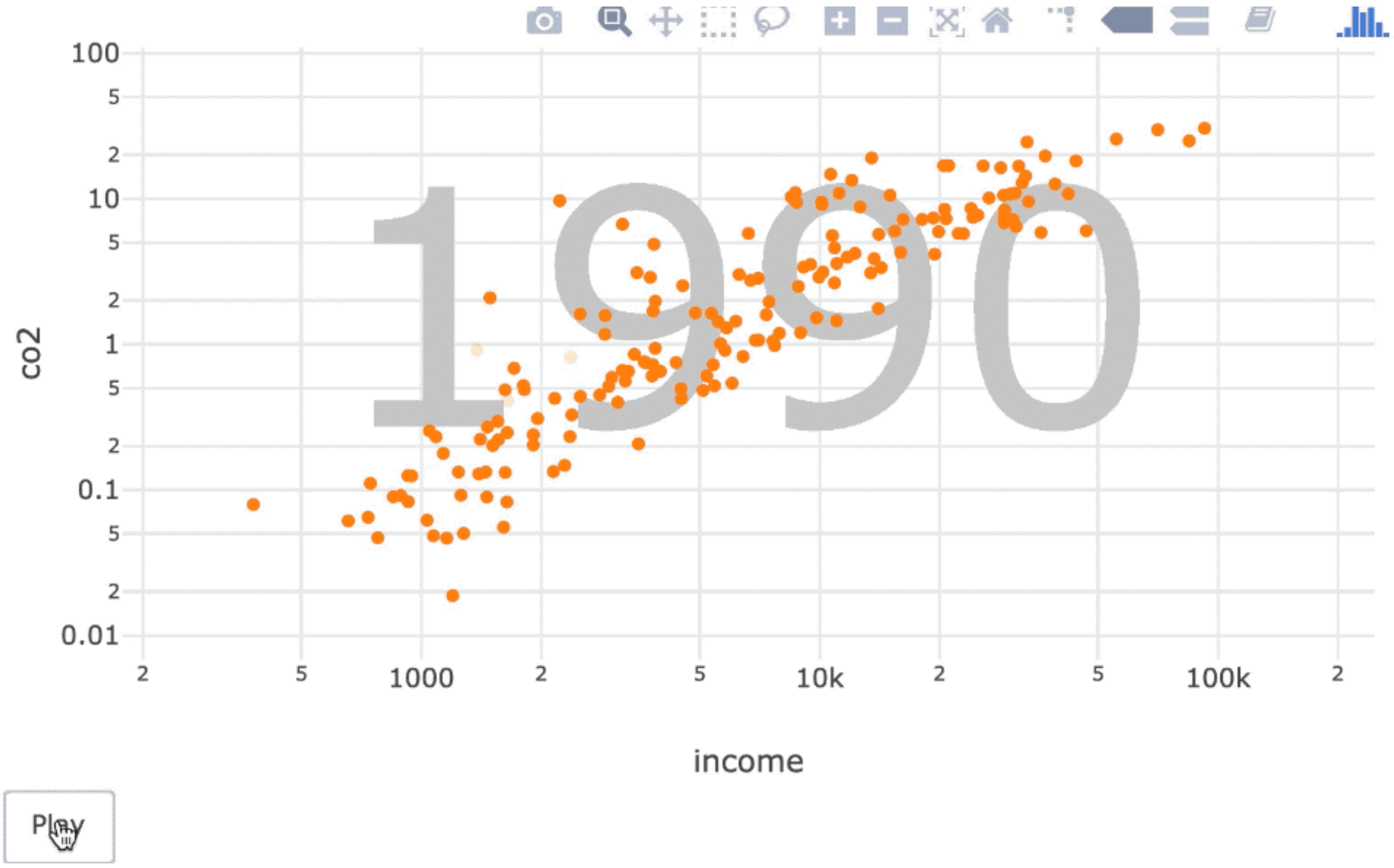


Points layer

```
world_indicators %>%
  plot_ly(x = ~income, y = ~co2) %>%
  add_text(
    x = 6500, y = 1, text = ~year, frame = ~year,
    textfont = list(size = 150, color = toRGB("gray80"))
  ) %>%
  add_markers(frame = ~year, ids = ~country) %>%
  layout(
    xaxis = list(type = "log"), yaxis = list(type = "log")
  )
```

Polishing

```
world_indicators %>%
  plot_ly(x = ~income, y = ~co2) %>%
  add_text(
    x = 6500, y = 1, text = ~year, frame = ~year,
    textfont = list(size = 150, color = toRGB("gray80"))
  ) %>%
  add_markers(frame = ~year, ids = ~country) %>%
  layout(
    xaxis = list(type = "log"), yaxis = list(type = "log"),
    showlegend = FALSE
  ) %>%
  animation_slider(hide = TRUE)
```


100
5
2
10
5
2
1
5
2
0.1
5
2
0.01
co2
1990
2
5
1000
2
5
10k
2
5
100k
2
income
Play

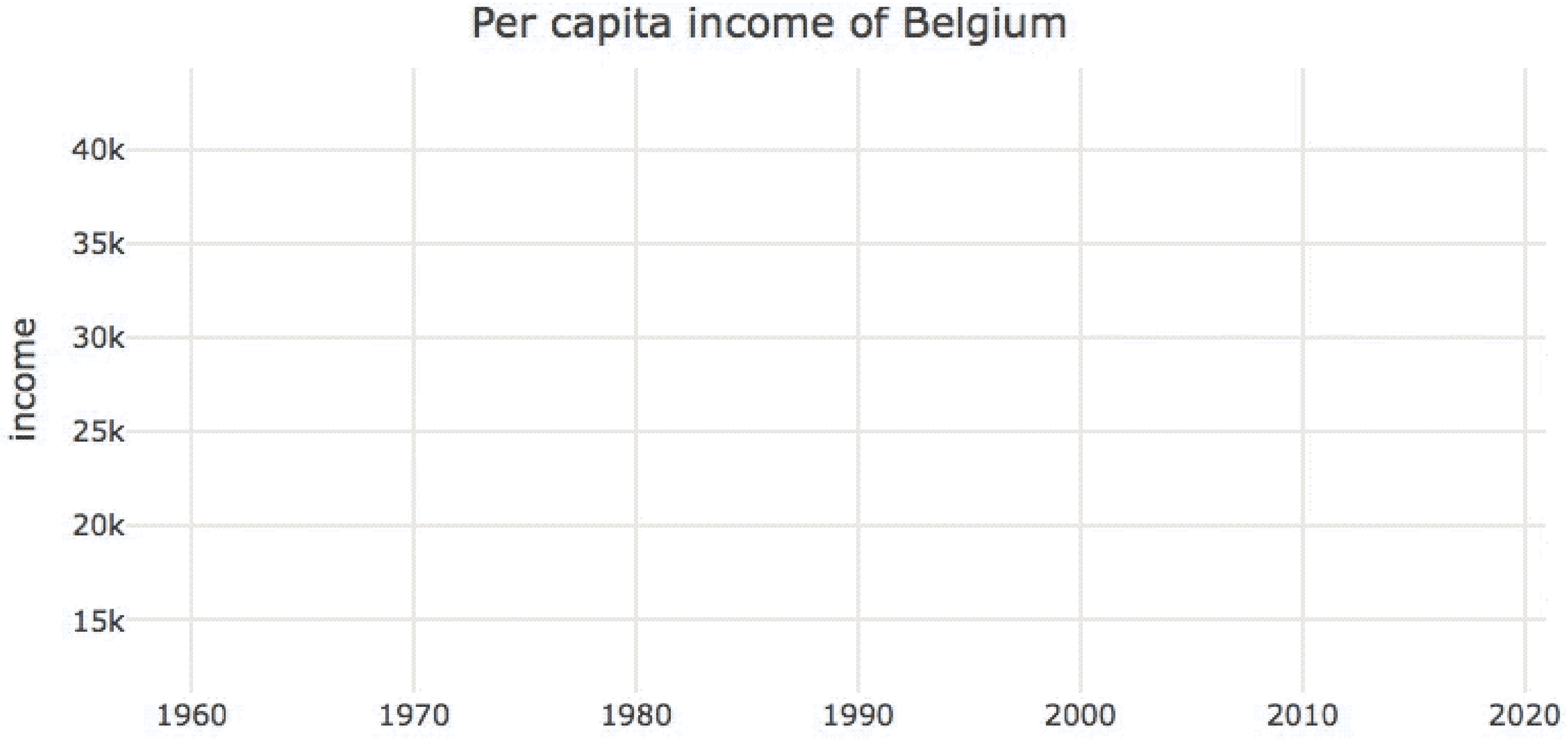
Let's practice!

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Cumulative Animations

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R

Adam Loy
Statistician, Carleton College


Per capita income of Belgium
income
40k
35k
30k
25k
20k
15k
1960
1970
1980
1990
2000
2010
2020

Belgian income data

```
belgium <- world_indicators %>%
  filter(country == "Belgium")
belgium
```

```
# A tibble: 59 x 11
  country  year income   co2 military population  urban life_expectancy four_regions
  <chr>   <dbl>  <dbl> <dbl>    <dbl>      <dbl>  <dbl>           <dbl> <chr>
1 Belgium  1960  12600  9.93     3.4     9170000 8.46e6            69.6 europe
2 Belgium  1961  13100 10.1      3.26    9230000 8.50e6            70.5 europe
3 Belgium  1962  13700 10.6      3.28    9280000 8.55e6            70.2 europe
4 Belgium  1963  14100 11.3      3.22    9340000 8.62e6            70   europe
5 Belgium  1964  15000 11        3.21    9390000 8.72e6            70.7 europe
# … with 54 more rows, and 2 more variables: eight_regions <chr>, six_regions <chr>
```

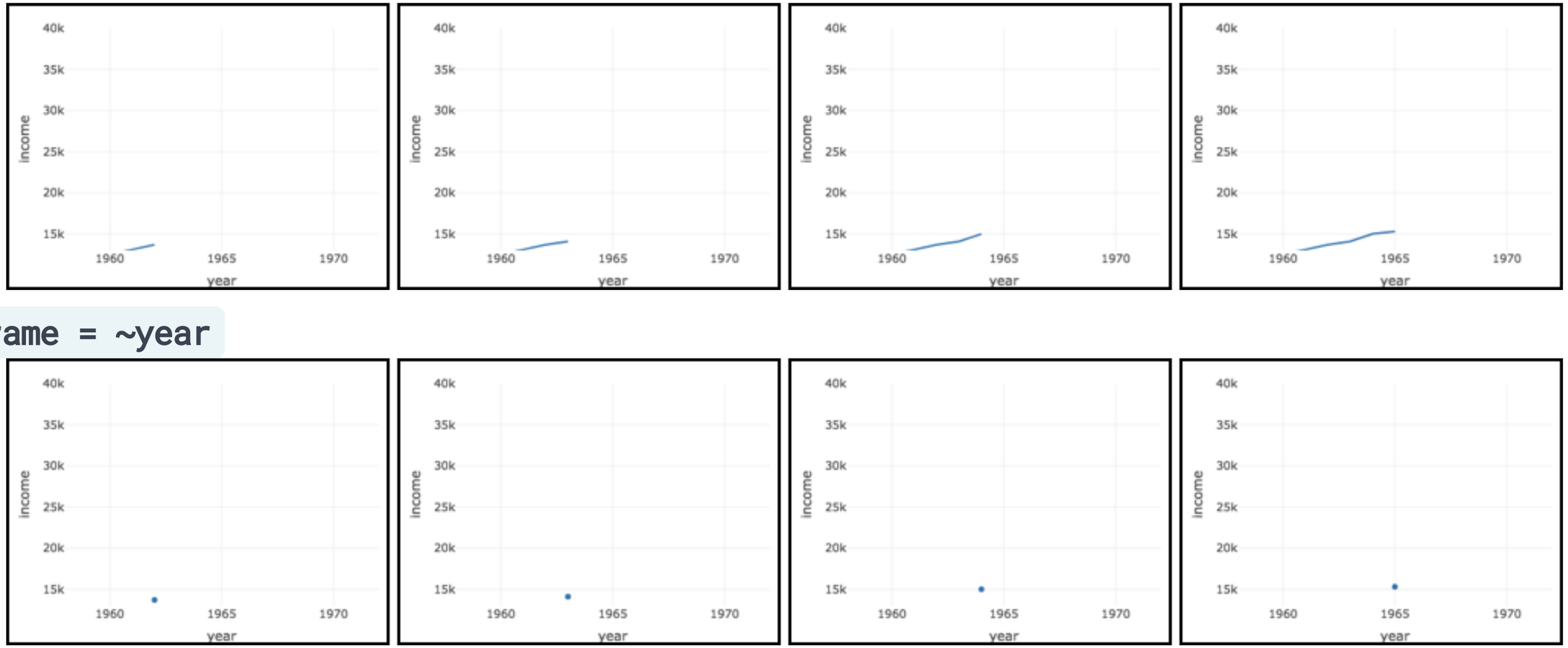
What's the frame?

```
# A tibble: 59 x 11
   country  year income   co2
   <chr>   <dbl>  <dbl> <dbl>
 1 Belgium  1960  12600  9.93
 2 Belgium  1961  13100 10.1
 3 Belgium  1962  13700 10.6
 4 Belgium  1963  14100 11.3
 5 Belgium  1964  15000 11
 6 Belgium  1965  15300 11.2
 # … with 53 more rows, and 7
 #   more variables
```

Does `frame = ~year` ??

What's the frame?

Goal




```
frame = ~year
```

Accumulating data sets

country	year	income
Belgium	1960	12600
Belgium	1961	13100
Belgium	1962	13700
Belgium	1963	14100
Belgium	1964	1500

country	year	income	frame
Belgium	1960	12600	1960

Accumulating data sets

country	year	income
Belgium	1960	12600
Belgium	1961	13100
Belgium	1962	13700
Belgium	1963	14100
Belgium	1964	1500

country	year	income	frame
Belgium	1960	12600	1960
Belgium	1960	12600	1961
Belgium	1961	13100	1961

Accumulating data sets

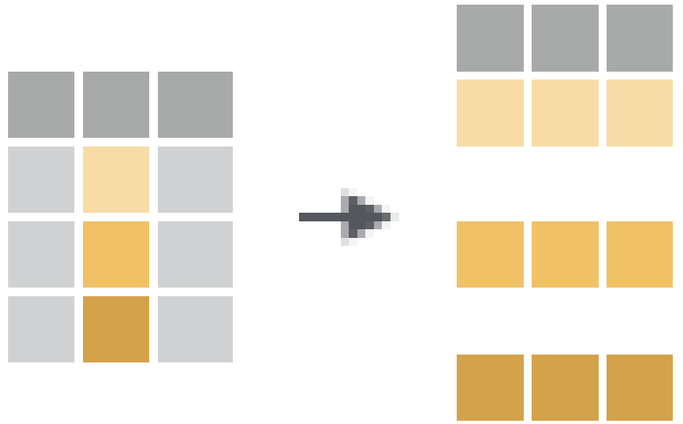
country	year	income
Belgium	1960	12600
Belgium	1961	13100
Belgium	1962	13700
Belgium	1963	14100
Belgium	1964	1500

country	year	income	frame
Belgium	1960	12600	1960
Belgium	1960	12600	1961
Belgium	1961	13100	1961
Belgium	1962	12600	1962
Belgium	1962	13100	1962
Belgium	1962	13700	1962

split()

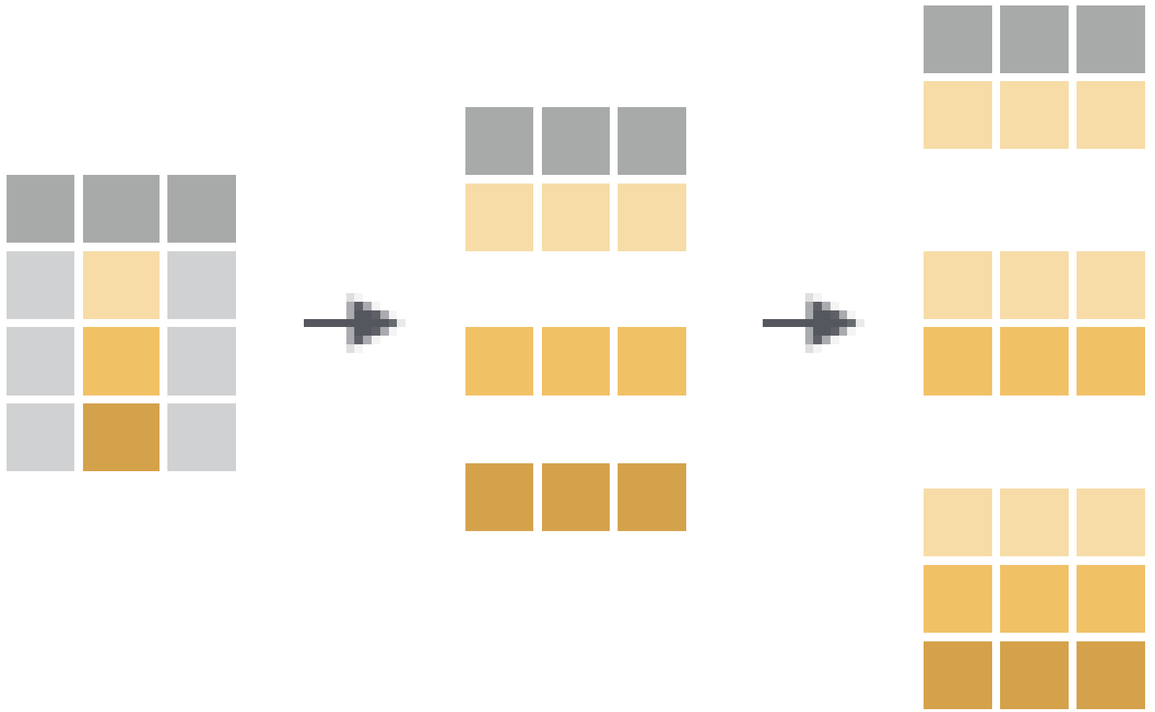
```
library(dplyr)
library(purrr)

belgium %>%
  split(.$year)
```



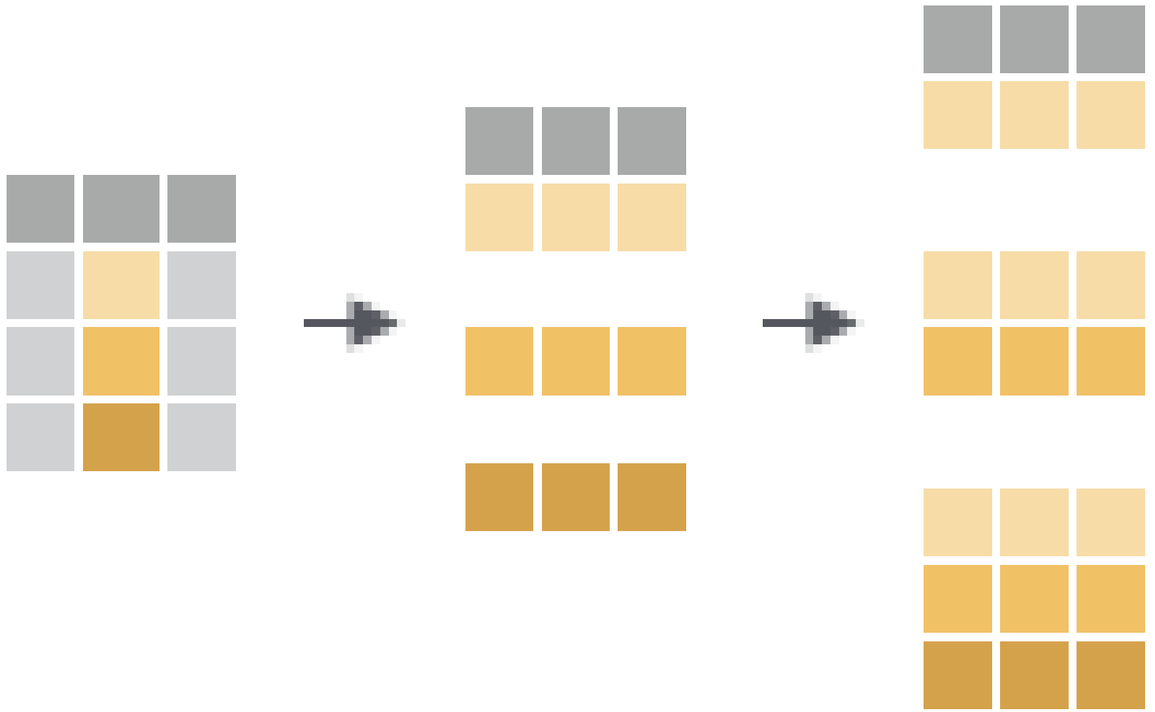
accumulate()

```
library(dplyr)
library(purrr)
belgium %>%
  split(.$year) %>%
  accumulate(~bind_rows(.x, .y))
```



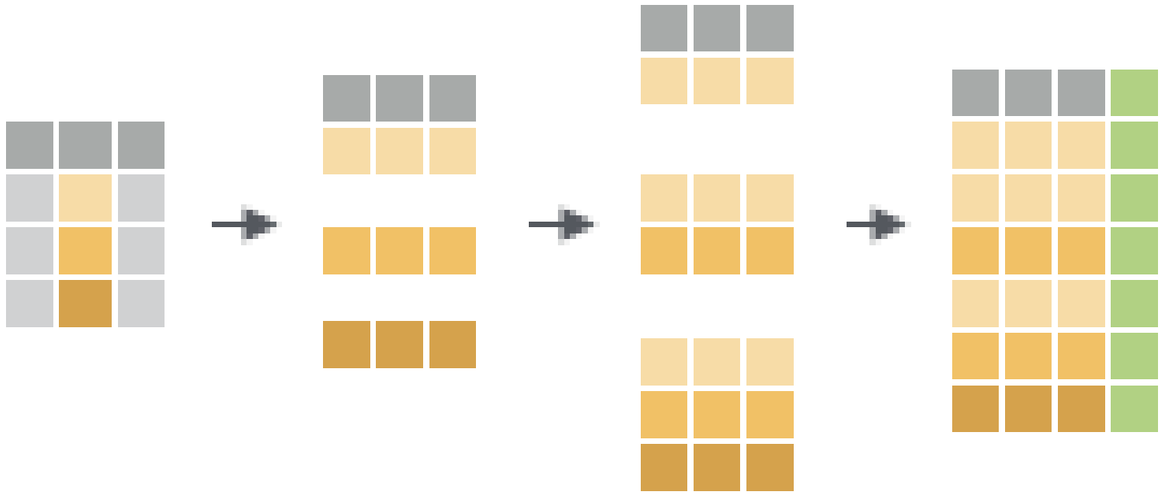
name

```
library(dplyr)
library(purrr)
belgium %>%
  split(.$year) %>%
  accumulate(~bind_rows(.x, .y)) %>%
  set_names(1960:2018)
```



combine

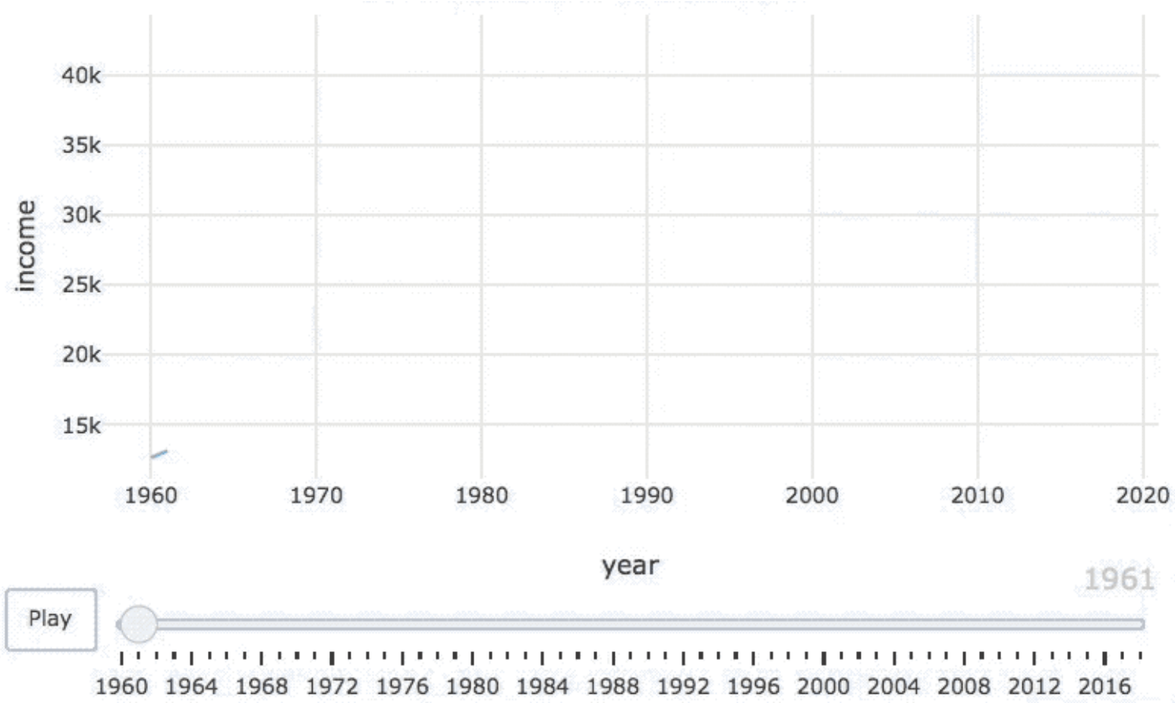
```
library(dplyr)
library(purrr)
belgium %>%
  split(.$year) %>%
  accumulate(~bind_rows(.x, .y)) %>%
  set_names(1960:2018) %>%
  bind_rows(.id = "frame")
```



```
# A tibble: 1,770 x 12
   frame country  year income   co2 military population  urban life_expectancy
   <chr> <chr>   <dbl>  <dbl> <dbl>    <dbl>      <dbl>  <dbl>           <dbl>
 1 1960  Belgium  1960  12600  9.93     3.4     9170000 8.46e6            69.6
 2 1961  Belgium  1960  12600  9.93     3.4     9170000 8.46e6            69.6
 3 1961  Belgium  1961  13100 10.1      3.26    9230000 8.50e6            70.5
 4 1962  Belgium  1960  12600  9.93     3.4     9170000 8.46e6            69.6
 5 1962  Belgium  1961  13100 10.1      3.26    9230000 8.50e6            70.5
 6 1962  Belgium  1962  13700 10.6      3.28    9280000 8.55e6            70.2
 7 1963  Belgium  1960  12600  9.93     3.4     9170000 8.46e6            69.6
 8 1963  Belgium  1961  13100 10.1      3.26    9230000 8.50e6            70.5
 9 1963  Belgium  1962  13700 10.6      3.28    9280000 8.55e6            70.2
10 1963  Belgium  1963  14100 11.3      3.22    9340000 8.62e6            70
# … with 1,760 more rows, and 3 more variables: four_regions <chr>,
#   eight_regions <chr>, six_regions <chr>
```

animate

```
library(dplyr)
library(purrr)
belgium %>%
  split(.$year) %>%
  accumulate(~bind_rows(.x, .y)) %>%
  set_names(1960:2018) %>%
  bind_rows(.id = "frame") %>%
  plot_ly(x = ~year, y = ~income) %>%
  add_lines(
    frame = ~frame, showlegend = FALSE
  )
```



Let's practice!

INTERMEDIATE INTERACTIVE DATA VISUALIZATION WITH PLOTLY IN R