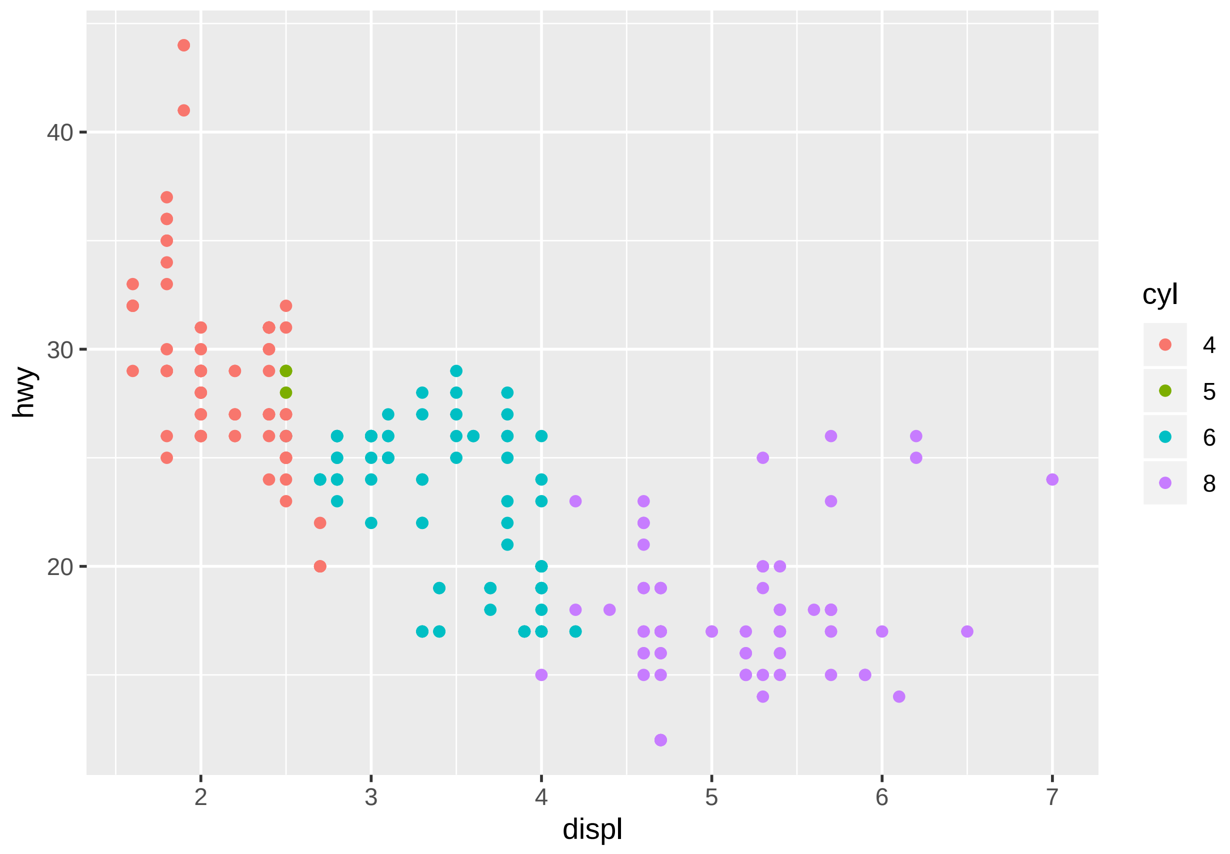
Interactive visualization

01-21-2020

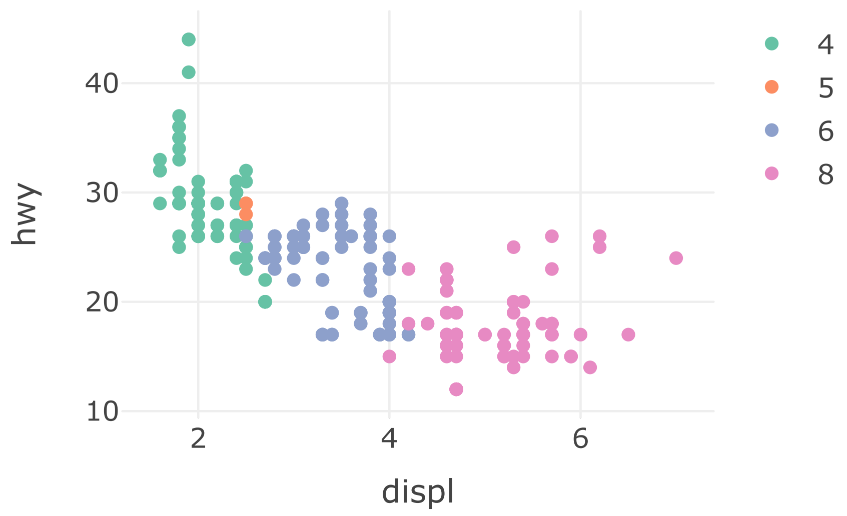
```r
library(tidyverse)
library(plotly)
```

plotly vs ggplot2

```r
mpg %>%
  mutate(cyl = as_factor(cyl)) %>%
  ggplot(aes(displ, hwy, color = cyl)) + geom_point()
```



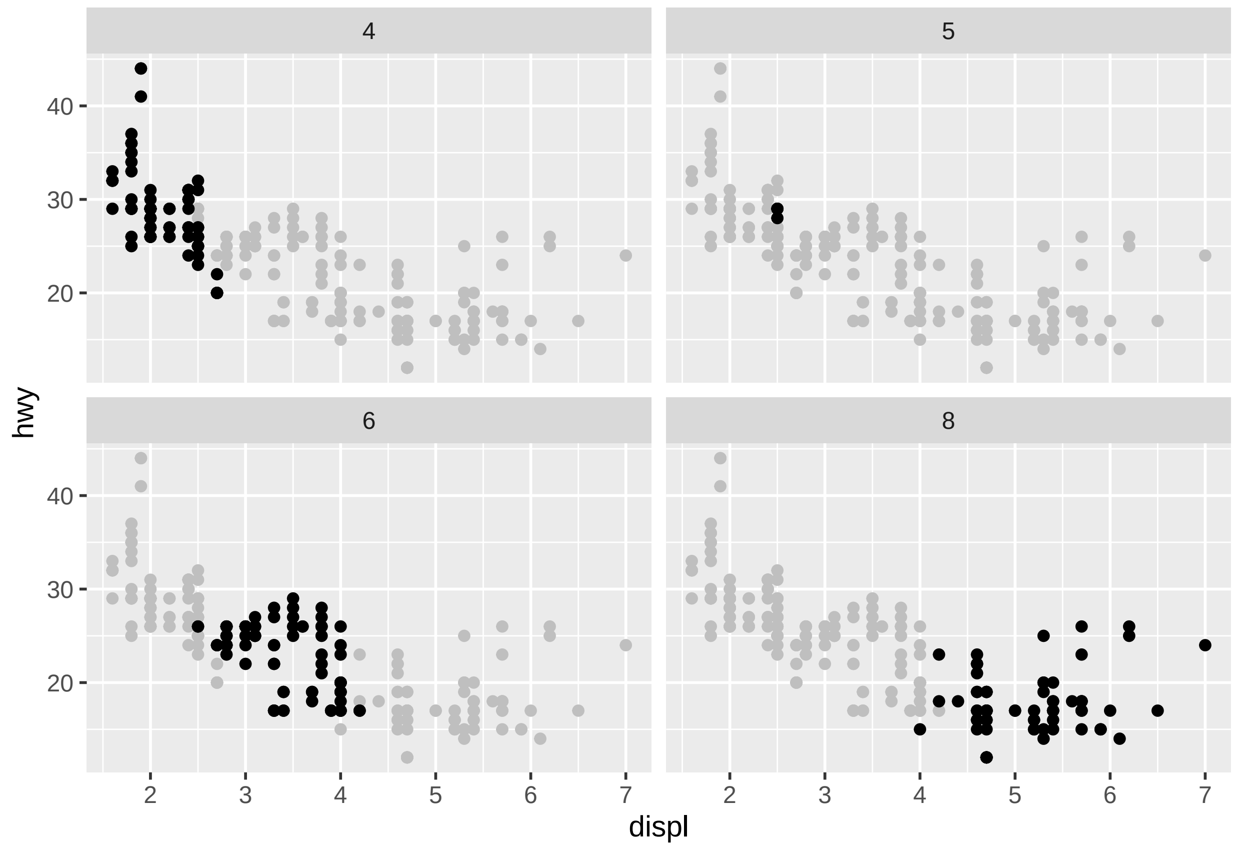
```r
mpg %>%
  plot_ly(x = ~displ, y = ~hwy, color = ~ factor(cyl))
```



Pros: - Interactive - Plotly handles multiple wide data columns (ggplot2 requies long format) - Plotly works for Python, Matlab, and Excel, among other languages - Easy layout customization - 3D charts

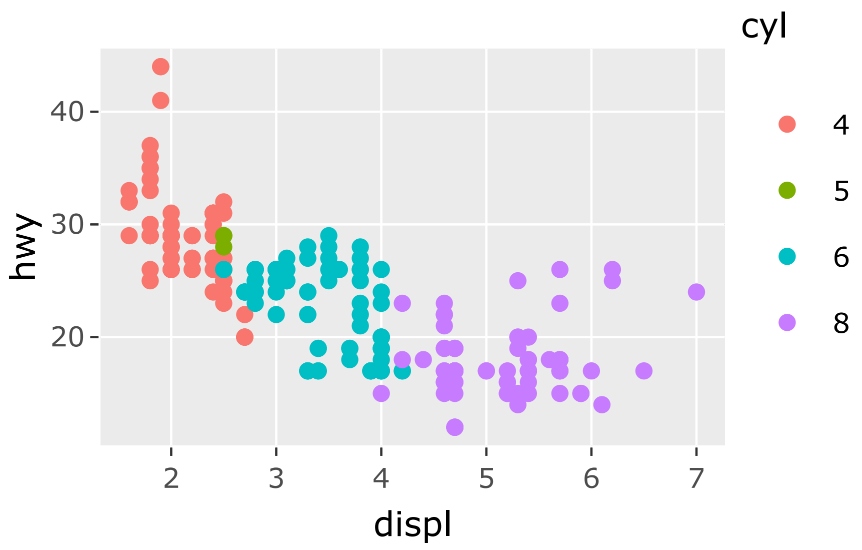
Cons: - Doesn't work very well with pdf - Facet wrapping is a bit complicated compared with ggplot2 - adding legend title is difficult

```r
ggplot(mpg, aes(displ, hwy)) +
  geom_point(data = mutate(mpg, cyl = NULL), color = "grey75") +
  geom_point() +
  facet_wrap(vars(cyl))
```



`ggplotly` function for ggplot2 users

```
p <- mpg %>%
  mutate(cyl = as_factor(cyl)) %>%
  ggplot(aes(displ, hwy, color = cyl)) + geom_point()
ggplotly(p)
```

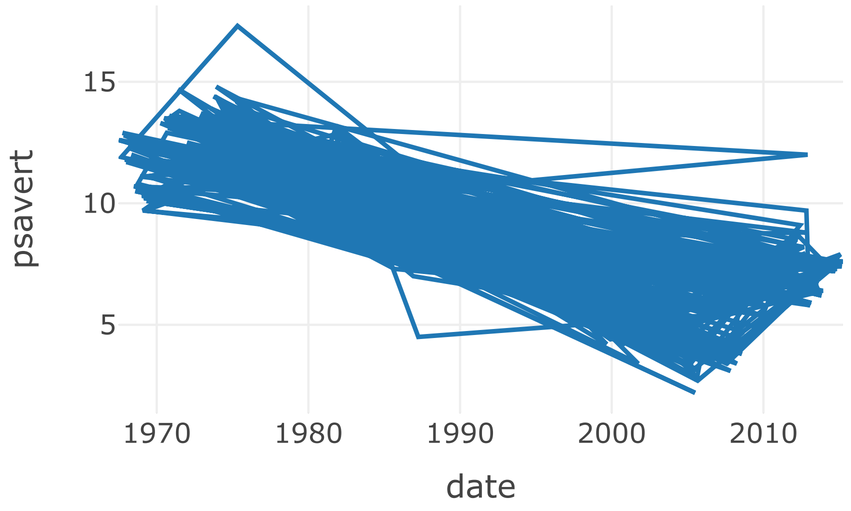


We won't learn (much of)

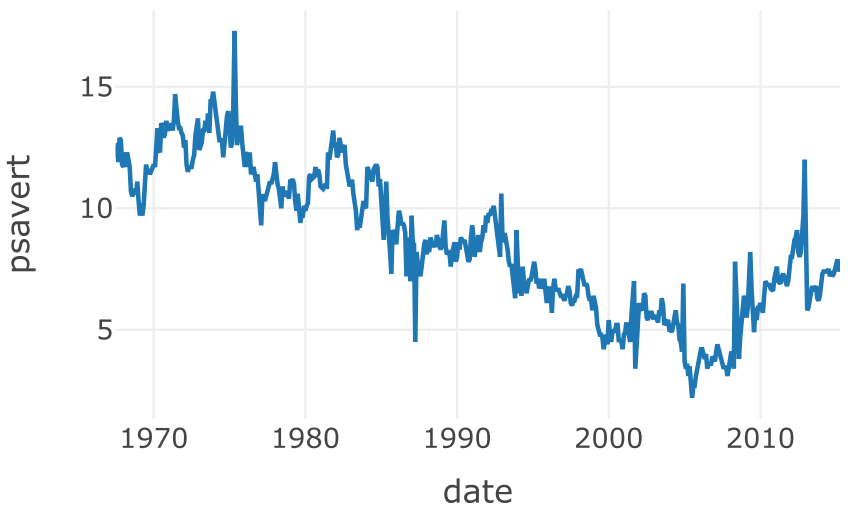
- ggplot2 (or should we?)
- HTML, SVG, CSS, JavaScript
- d3.js (R package r2d3)

Scatter plots + lines

```
p <- economics %>%
  sample_n(n()) %>%
  plot_ly(x = ~date, y = ~psavert)
p %>% add_paths() # using the order of the data frame
```



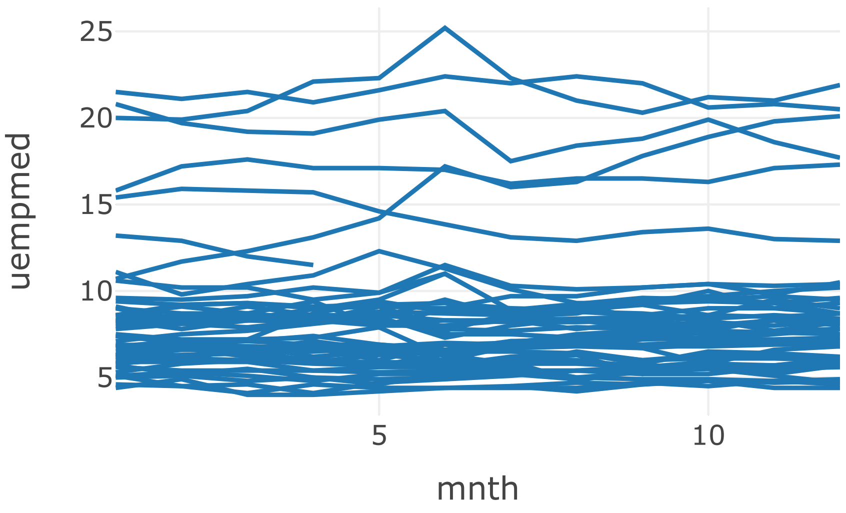
```r
p %>% add_lines()
```



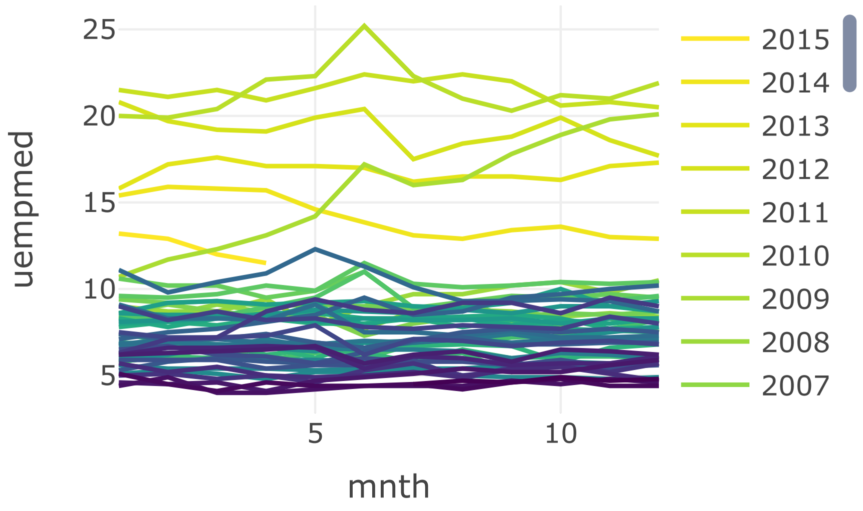
```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
econ <- economics %>%
  mutate(yr = year(date), mnth = month(date))

# One trace (more performant, but less interactive)
econ %>%
  group_by(yr) %>%
  plot_ly(x = ~mnth, y = ~uempmed) %>%
  add_lines(text = ~yr)
```

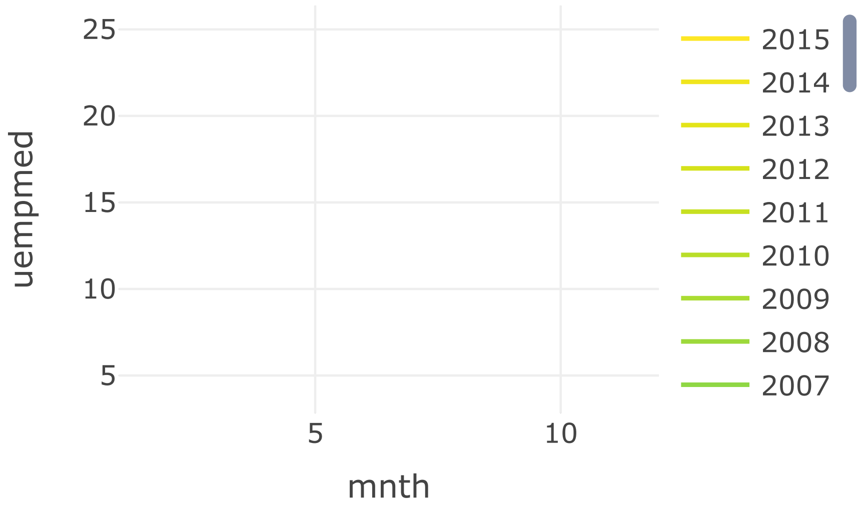


```r
# Multiple traces (less performant, but more interactive)
econ %>%
  mutate(yr = ordered(yr)) %>%
  plot_ly(x = ~mnth, y = ~uempmed) %>%
  add_lines(color = ~yr)
```



Use Canvas rather then SVG for large dataset

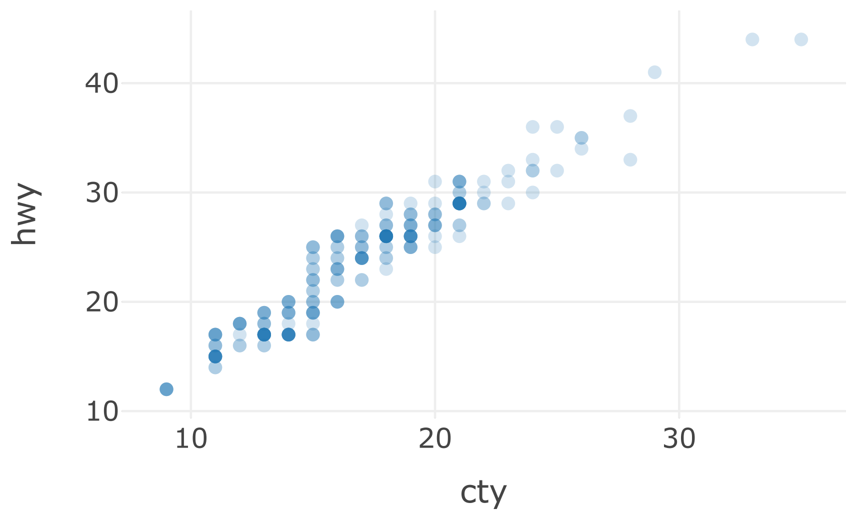
```
plot_ly(econ, x = ~mnth, y = ~uempmed) %>%
  add_lines(color = ~ ordered(yr)) %>%
  toWebGL()
```



Markers

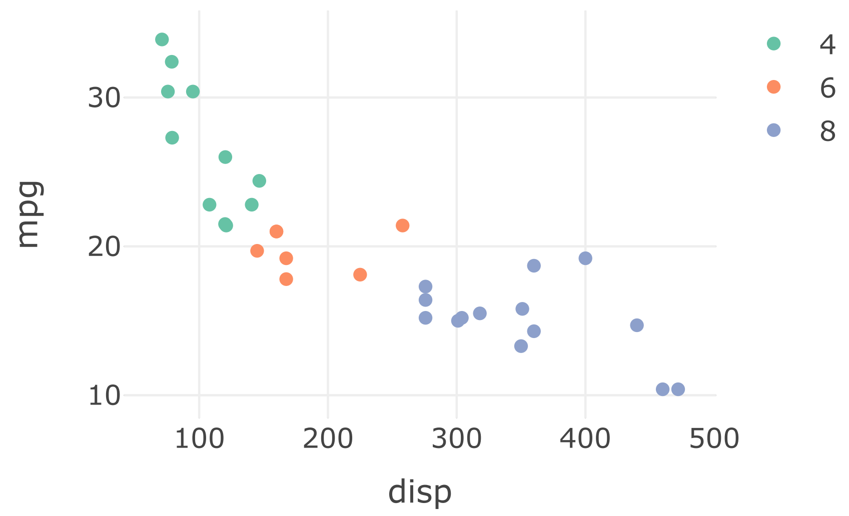
- alpha blending to see replicates

```r
mpg %>%
  plot_ly(x = ~cty, y = ~hwy) %>%
  add_markers(alpha = 0.2)
```



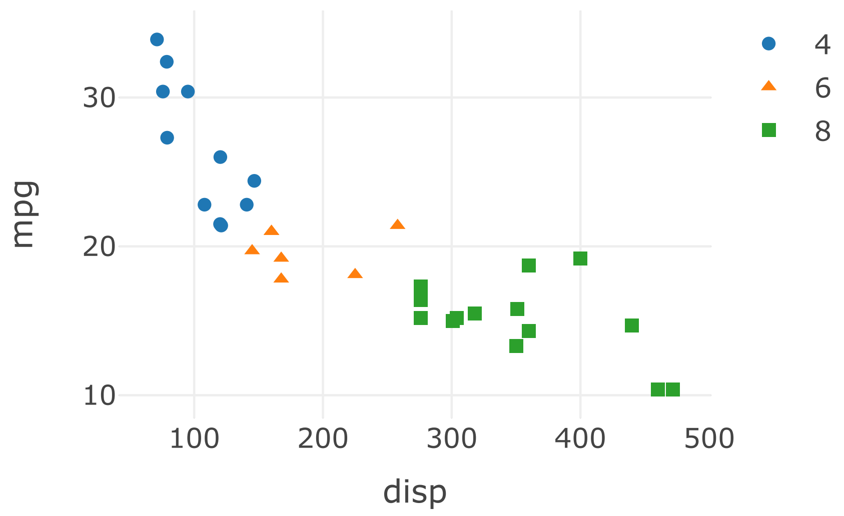
- colors for grouping

```
mtcars %>%
  plot_ly(x = ~disp, y = ~mpg) %>%
  add_markers(color = ~ factor(cyl))
```



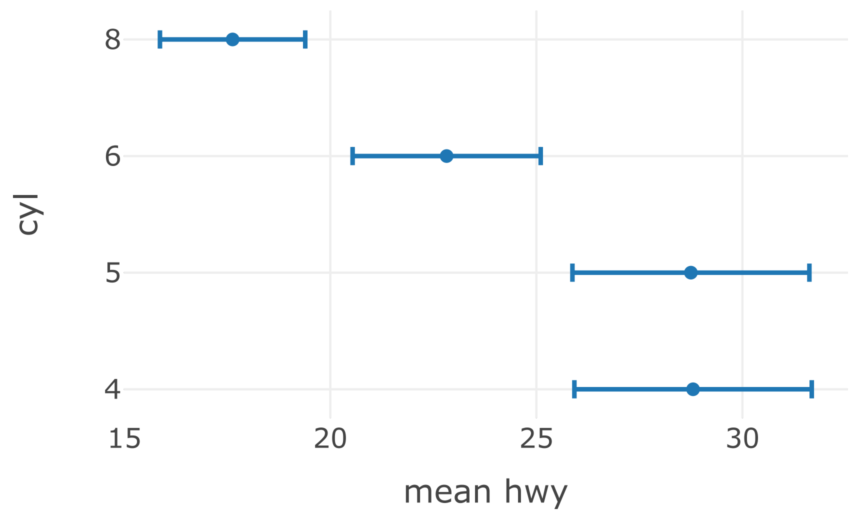
- symbols

```
mtcars %>%
  plot_ly(x = ~disp, y = ~mpg) %>%
  add_markers(symbol = ~ factor(cyl))
```



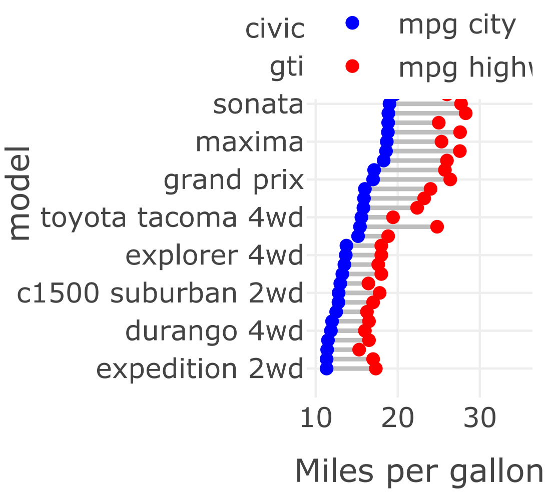
Error bar plot

```r
mpg %>%
  group_by(cyl) %>%
  summarize(mhwy = mean(hwy), se = sd(hwy) / sqrt(n())) %>%
  plot_ly(x = ~mhwy, y = ~ factor(cyl)) %>%
  add_markers(error_x = ~ list(value = se)) %>%
  layout(xaxis = list(title = "mean hwy"), yaxis = list(title = "cyl"))
```



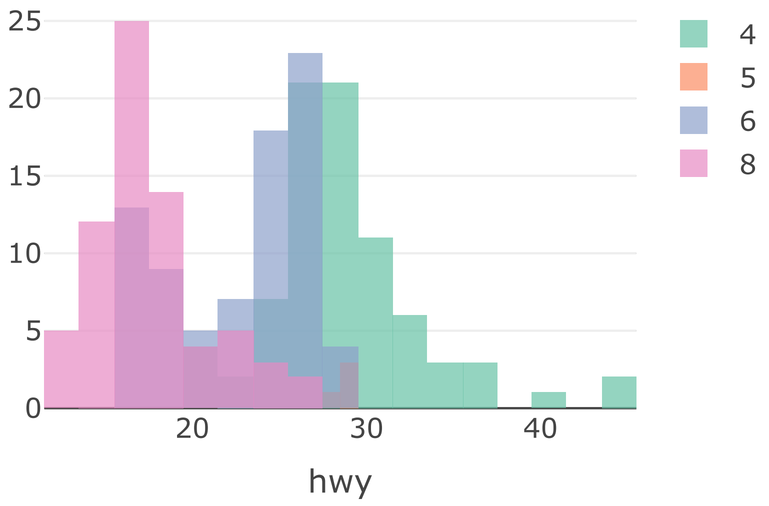
Dumbbell Plot

```
mpg %>%
  group_by(model) %>%
  summarize(c = mean(cty), h = mean(hwy)) %>%
  mutate(model = forcats::fct_reorder(model, c)) %>%
  plot_ly() %>%
  add_segments(
    x = ~c, y = ~model,
    xend = ~h, yend = ~model,
    color = I("gray"), showlegend = FALSE
  ) %>%
  add_markers(
    x = ~c, y = ~model,
    color = I("blue"),
    name = "mpg city"
  ) %>%
  add_markers(
    x = ~h, y = ~model,
    color = I("red"),
    name = "mpg highway"
  ) %>%
  layout(xaxis = list(title = "Miles per gallon"))
```



Histograms

```
mpg %>%
  plot_ly(x = ~hwy, color = ~ factor(cyl)) %>%
  add_histogram(histnorm = "", alpha = 0.7) %>% # histnorm could be "", "probability", "density" and "p
  layout(barmode = "overlay") # barmode could be "overlay", "stack" and "group"
```

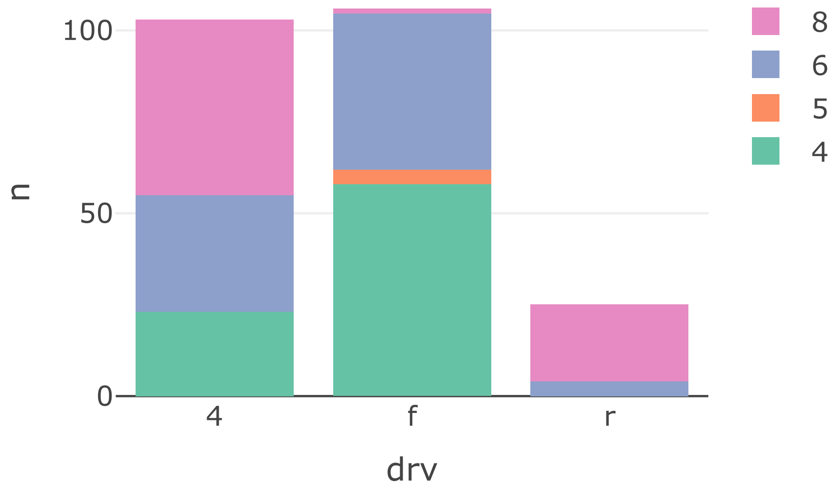


```
names(relig_income) %>%
  paste0(collapse = "\n") %>%
  cat()
```

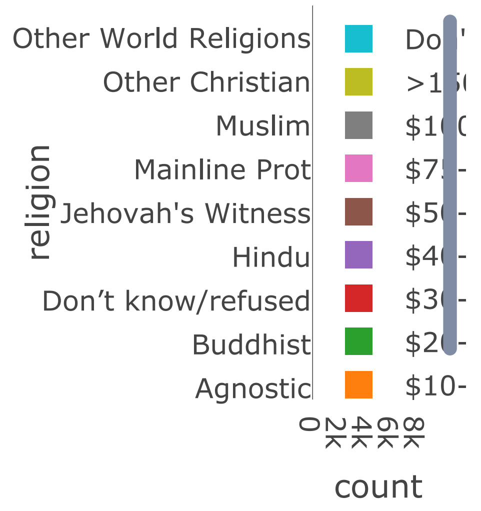
```
## religion
## <$10k
## $10-20k
## $20-30k
## $30-40k
## $40-50k
## $50-75k
## $75-100k
## $100-150k
## >150k
## Don't know/refused
```

Bar plots

```
mpg %>%
  mutate(cyl = as_factor(cyl)) %>%
  count(drv, cyl) %>%
  plot_ly(x = ~drv, y = ~n, color = ~cyl) %>%
  add_bars() %>%
  layout(barmode = "stack")
```

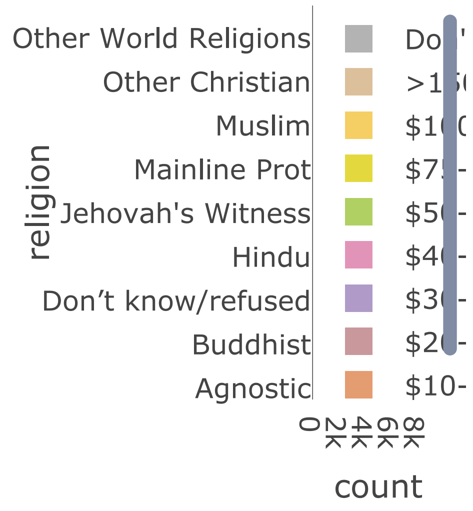


```
# work with wide format directly
relig_income %>%
  mutate(religion = as_factor(religion)) %>%
  plot_ly(y = ~religion) %>%
  add_bars(~`<$10k`, name = "<$10k") %>%
  add_bars(~`$10-20k`, name = "$10-20k") %>%
  add_bars(~`$20-30k`, name = "$20-30k") %>%
  add_bars(~`$30-40k`, name = "$30-40k") %>%
  add_bars(~`$40-50k`, name = "$40-50k") %>%
  add_bars(~`$50-75k`, name = "$50-75k") %>%
  add_bars(~`$75-100k`, name = "$75-100k") %>%
  add_bars(~`$100-150k`, name = "$100-150k") %>%
  add_bars(~`>150k`, name = ">150k") %>%
  add_bars(~`Don't know/refused`, name = "Don't know/refused") %>%
  layout(xaxis = list(title = "count"), barmode = "stack")
```



```r
# may be easier with `pivot_longer`?
relig_income %>%
  mutate(religion = as_factor(religion)) %>%
  pivot_longer(-religion, names_to = "income", values_to = "count") %>%
  mutate(income = fct_inorder(income)) %>%
  plot_ly(x = ~count, y = ~religion, color = ~income) %>%
  add_bars() %>%
  layout(barmode = "stack")
```

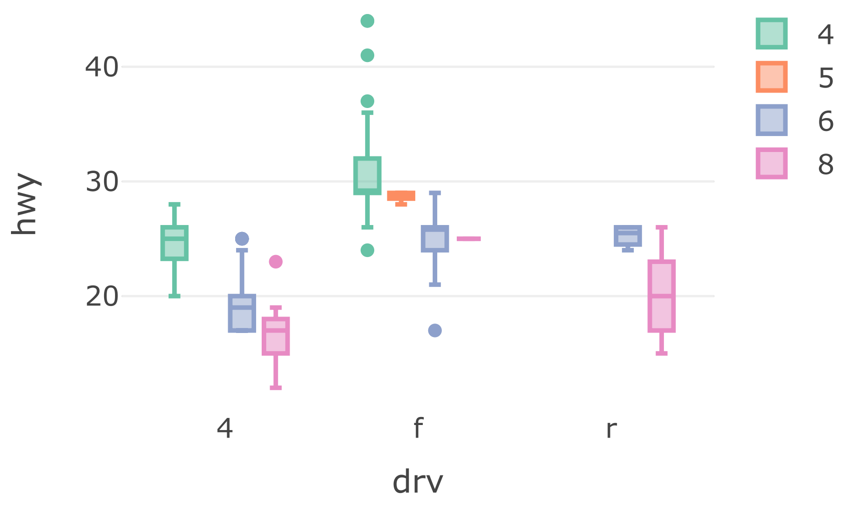
```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors

## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
```



Box plots

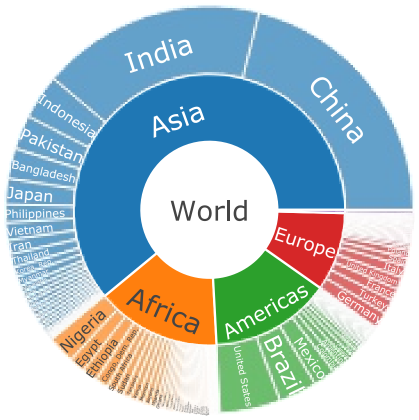
```
mpg %>%
  mutate(cyl = as_factor(cyl)) %>%
  plot_ly(x = ~drv, y = ~hwy, color = ~cyl) %>%
  add_boxplot() %>%
  layout(boxmode = "group")
```

```
## Warning: 'layout' objects don't have these attributes: 'boxmode'
## Valid attributes include:
## 'font', 'title', 'autosize', 'width', 'height', 'margin', 'paper_bgcolor', 'plot_bgcolor', 'separato
```



Sunburst

```
library(gapminder)
gapminder2007 <- gapminder %>%
  filter(year == 2007) %>%
  mutate(pop = as.double(pop))
bind_rows(
  gapminder2007 %>%
    select(children = country, parents = continent, pop = pop),
  gapminder2007 %>%
    group_by(continent) %>%
    summarize(pop = sum(pop)) %>%
    transmute(children = continent, parents = "World", pop = pop),
  gapminder2007 %>%
    summarize(pop = sum(pop)) %>%
    transmute(children = "World", parents = "", pop = pop)
) %>%
  plot_ly(
    ids = ~children,
    labels = ~children,
    parents = ~parents,
    values = ~pop,
    type = "sunburst",
    branchvalues = "total")
```



3D charts and its family

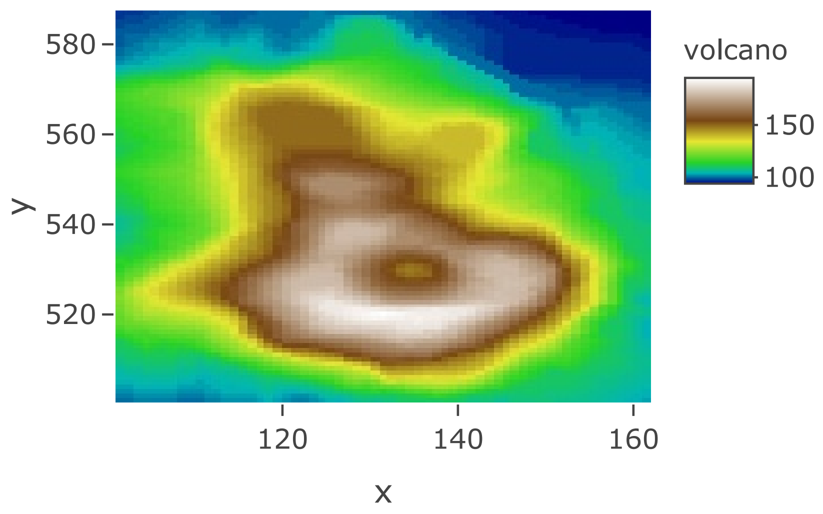
```
mpg %>%
  mutate(cyl = as_factor(cyl)) %>%
  plot_ly(x = ~cty, y = ~hwy, z = ~cyl) %>%
  add_markers(color = ~cyl)
```

WebGL is not supported by your browser - visit https://get.webgl.org for more info

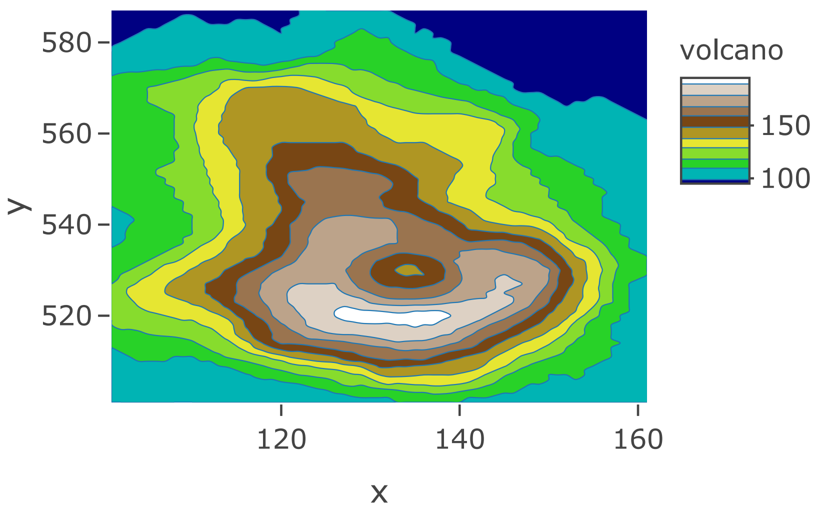
```r
x <- seq_len(nrow(volcano)) + 100
y <- seq_len(ncol(volcano)) + 500
plot_ly(x = ~x, y = ~y, z = ~volcano) %>%
  add_surface(colorscale = "Earth")
```

WebGL is not supported by your browser - visit https://get.webgl.org for more info

```r
# heatmap
plot_ly(x = ~x, y = ~y, z = ~volcano) %>%
  add_heatmap(colorscale = "Earth")
```



```r
plot_ly(x = ~x, y = ~y, z = ~volcano) %>%
  add_contour(colorscale = "Earth")
```

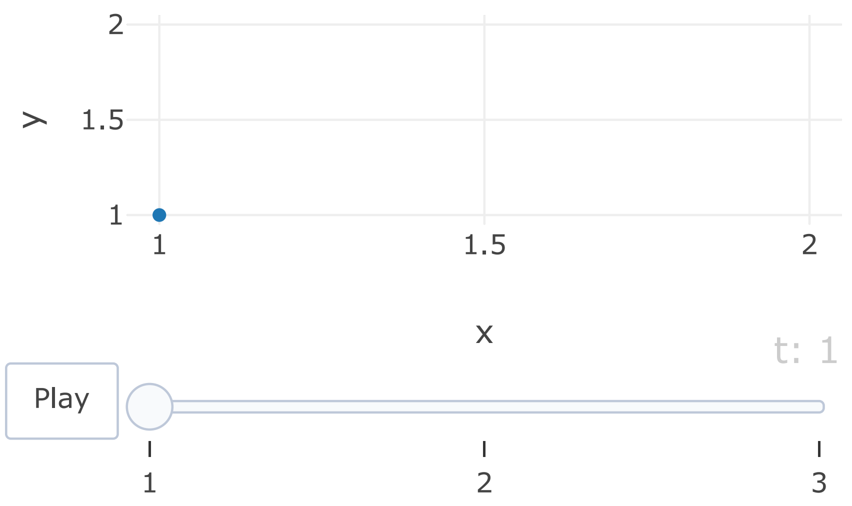


Animating views

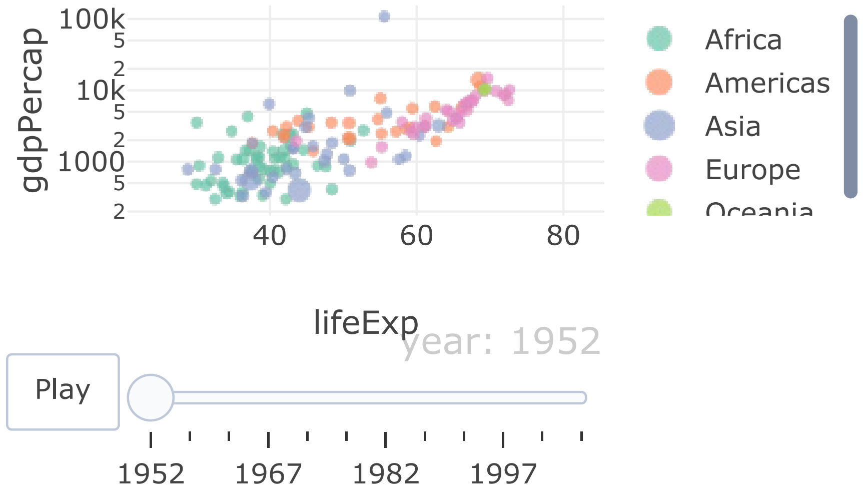
```r
df <- tibble(
  x = c(1, 2, 1),
  y = c(1, 2, 1),
  t = c(1, 2, 3)
)

df %>%
  plot_ly(x = ~x, y = ~y, frame = ~t, showlegend = F)
```

```
## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plot.ly/r/reference/#scatter
```

```
## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plot.ly/r/reference/#scatter-mode
```



```
library(gapminder)
(p <- gapminder %>%
  plot_ly(x = ~lifeExp, y = ~gdpPercap, size = ~pop, color = ~continent, frame = ~year) %>%
  layout(yaxis = list(type = "log")))
```



A more powerful (therefore difficult to master) graphic library `r2d3`

`r2d3` is an R binding to the famous javascript library `d3.js`. Visit https://github.com/d3/d3/wiki/Gallery to see some of the things that `d3.js` is able to create.

References

- Plotly official website: https://plot.ly/r/
- Carson Sievert, Interactive web-based data visualization with R, plotly, and shiny https://plotly-r.com/