Trelliscope in the tidyverse

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Ryan Hafen
Author, TrelliscopeJS

datacamp

Stock market data

```
library(dplyr)
glimpse(stocks)
```

```
Rows: 125,928
Columns: 8
$ symbol   <chr> "TWOU", "TWOU", "TWOU", "TWOU", "TWOU", "TWOU", "TWOU", "T...
$ date     <date> 2016-01-04, 2016-01-05, 2016-01-06, 2016-01-07, 2016-01-0...
$ open     <dbl> 27.21, 26.92, 26.19, 25.84, 25.31, 24.87, 23.82, 23.40, 21...
$ high     <dbl> 27.500, 27.420, 26.530, 26.470, 25.790, 24.870, 24.100, 23...
$ low      <dbl> 26.360, 26.280, 25.950, 24.523, 24.220, 23.220, 22.430, 21...
$ close    <dbl> 27.04, 26.54, 26.38, 25.23, 24.32, 23.72, 23.27, 21.85, 22...
$ volume   <dbl> 530200, 448800, 297200, 635200, 364500, 404900, 1012100, 6...
$ adjusted <dbl> 27.04, 26.54, 26.38, 25.23, 24.32, 23.72, 23.27, 21.85, 22...
```

Visualizing stock data

```
library(dplyr)
library(plotly)
candlestick_plot <- function(d)
  plot_ly(d, x = ~date, type = "candlestick",
    open = ~open, close = ~close,
    high = ~high, low = ~low)
candlestick_plot(filter(stocks, symbol == "AAPL"))
```

Tidyverse: nested data frames

```
by_symbol <- stocks %>%
  group_by(symbol) %>%
  nest()
by_symbol
```

```
# A tibble: 500 x 2
   symbol data
   <chr>  <list>
 1 TWOU   <tibble [252 × 7]>
 2 JOBS   <tibble [252 × 7]>
 3 ABMD   <tibble [252 × 7]>
 4 ACHC   <tibble [252 × 7]>
 5 ACAD   <tibble [252 × 7]>
 6 ACIW   <tibble [252 × 7]>
# ... with 494 more rows
```

Tidyverse: computing on nested data frames

The `purrr` `map_*()` functions

```
by_symbol <- mutate(by_symbol,
  last_close = map_dbl(data, function(x) tail(x$close, 1)))
by_symbol
```

```
# A tibble: 500 x 3
   symbol data               last_close
   <chr>  <list>                  <dbl>
 1 TWOU   <tibble [252 × 7]>       30.2
 2 JOBS   <tibble [252 × 7]>       33.8
 3 ABMD   <tibble [252 × 7]>      113
 4 ACHC   <tibble [252 × 7]>       33.1
 5 ACAD   <tibble [252 × 7]>       28.8
 6 ACIW   <tibble [252 × 7]>       18.2
# ... with 494 more rows
```

Trelliscope in the tidyverse: plot columns

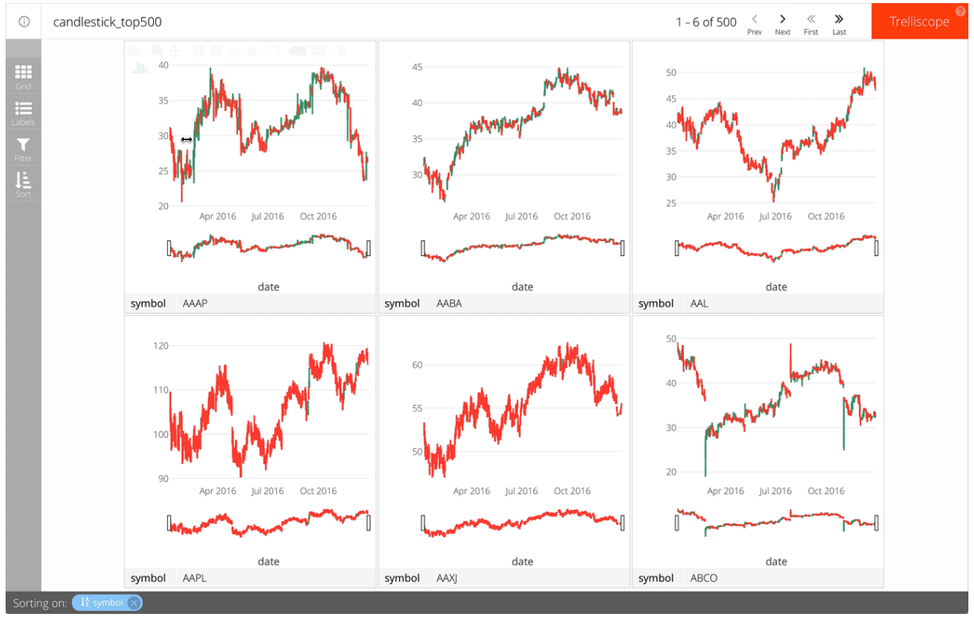
`map_plot()`

```
library(trelliscopejs)
by_symbol <- mutate(by_symbol,
  panel = map_plot(data, candlestick_plot))
```

```
# A tibble: 500 x 4
   symbol data               last_close panel
   <chr>  <list>                  <dbl> <list>
 1 TWOU   <tibble [252 × 7]>       30.2 <S3: plotly>
 2 JOBS   <tibble [252 × 7]>       33.8 <S3: plotly>
 3 ABMD   <tibble [252 × 7]>      113   <S3: plotly>
 4 ACHC   <tibble [252 × 7]>       33.1 <S3: plotly>
 5 ACAD   <tibble [252 × 7]>       28.8 <S3: plotly>
 6 ACIW   <tibble [252 × 7]>       18.2 <S3: plotly>
# ... with 494 more rows
```

Creating the display

```
trelliscope(by_symbol,
  name = "candlestick_top500",
  nrow = 2, ncol = 3)
```



Let's practice!

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Cognostics

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Ryan Hafen
Author, TrelliscopeJS

datacamp

Variables = cognostics

```
by_symbol
```

```
# A tibble: 500 x 4
   symbol data                last_close panel
   <chr>  <list>                   <dbl> <list>
 1 TWOU   <tibble [252 × 7]>       30.2 <S3: plotly>
 2 JOBS   <tibble [252 × 7]>       33.8 <S3: plotly>
 3 ABMD   <tibble [252 × 7]>      113   <S3: plotly>
 4 ACHC   <tibble [252 × 7]>       33.1 <S3: plotly>
 5 ACAD   <tibble [252 × 7]>       28.8 <S3: plotly>
 6 ACIW   <tibble [252 × 7]>       18.2 <S3: plotly>
 7 ATVI   <tibble [252 × 7]>       36.1 <S3: plotly>
 8 ADBE   <tibble [252 × 7]>      103   <S3: plotly>
 9 AAAP   <tibble [252 × 7]>       26.8 <S3: plotly>
10 AEIS   <tibble [252 × 7]>       54.8 <S3: plotly>
# ... with 490 more rows
```

Adding cognostics to the stock data

```
stocks_meta
```

```
# A tibble: 500 x 6
   symbol company                market_cap ipo_year sector   industry
   <chr>  <chr>                       <dbl>    <dbl> <chr>    <chr>
 1 AAPL   Apple Inc.           791730000000     1980 Technol… Computer Manufa…
 2 GOOGL  Alphabet Inc.        668500000000       NA Technol… Computer Softwa…
 3 GOOG   Alphabet Inc.        657890000000     2004 Technol… Computer Softwa…
 4 MSFT   Microsoft Corporation 568960000000     1986 Technol… Computer Softwa…
 5 FB     Facebook, Inc.       490030000000     2012 Technol… Computer Softwa…
 6 AMZN   Amazon.com, Inc.     459430000000     1997 Consume… Catalog/Special…
# ... with 494 more rows
```

```
library(dplyr)
by_symbol <- left_join(by_symbol, stocks_meta)
```

```
by_symbol <- mutate(by_symbol,
  volume_stats = map(data, function(x) {
    data_frame(min_volume = min(x$volume),
               max_volume = max(x$volume))   }))
by_symbol %>% select(symbol, data, volume_stats) %>% head(3)
```

```
# A tibble: 3 x 3
  symbol data               volume_stats
  <chr>  <list>             <list>
1 TWOU   <tibble [252 × 8]> <tibble [1 × 2]>
2 JOBS   <tibble [252 × 8]> <tibble [1 × 2]>
3 ABMD   <tibble [252 × 8]> <tibble [1 × 2]>
```

```
by_symbol$stats[[1]]
```

```
# A tibble: 1 x 2
  min_volume max_volume
       <dbl>      <dbl>
1     139800    1843900
```

Customizing cognostics

`cog()` function allows specification of cognostic behaviors in the viewer:

- `desc` : a free text description
- `default_label` : should this cognostic be shown by default in the viewer?

```
by_symbol <- mutate(by_symbol,
  company = cog(
    val = company,
    desc = "company name",
    default_label = TRUE
  )
)
```

Let's practice!

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Trelliscope options

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Ryan Hafen
Author, TrelliscopeJS

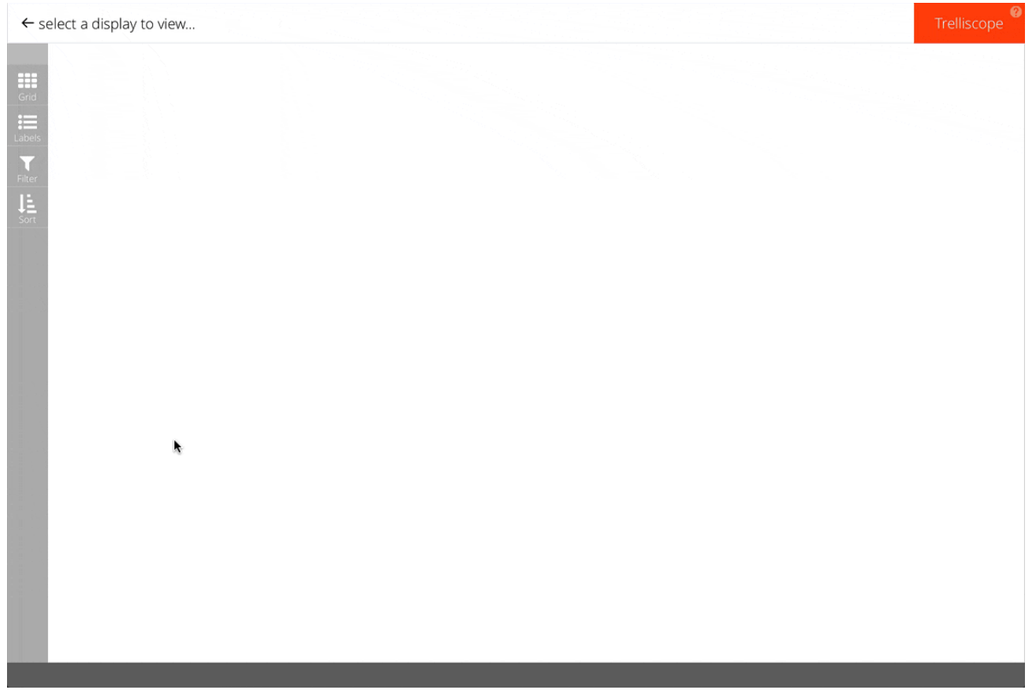
Trelliscope options

`facet_trelliscope()` and `trelliscope()` provide several options for handling the display output:

- The directory in which to store the display
- Storing multiple displays in the same directory
- Providing more detailed descriptions of the display
- The plot aspect ratio
- Specifying the default viewer state

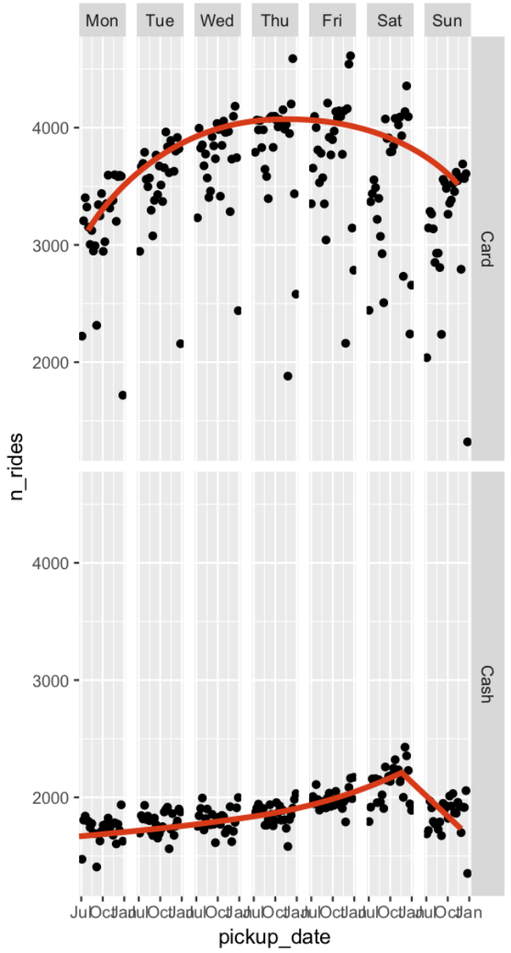
Specifying the output directory

```
trelliscope(dat, path = "...", ...)
ggplot(...) +
  ... +
  facet_trelliscope(path = "...", ...)
```

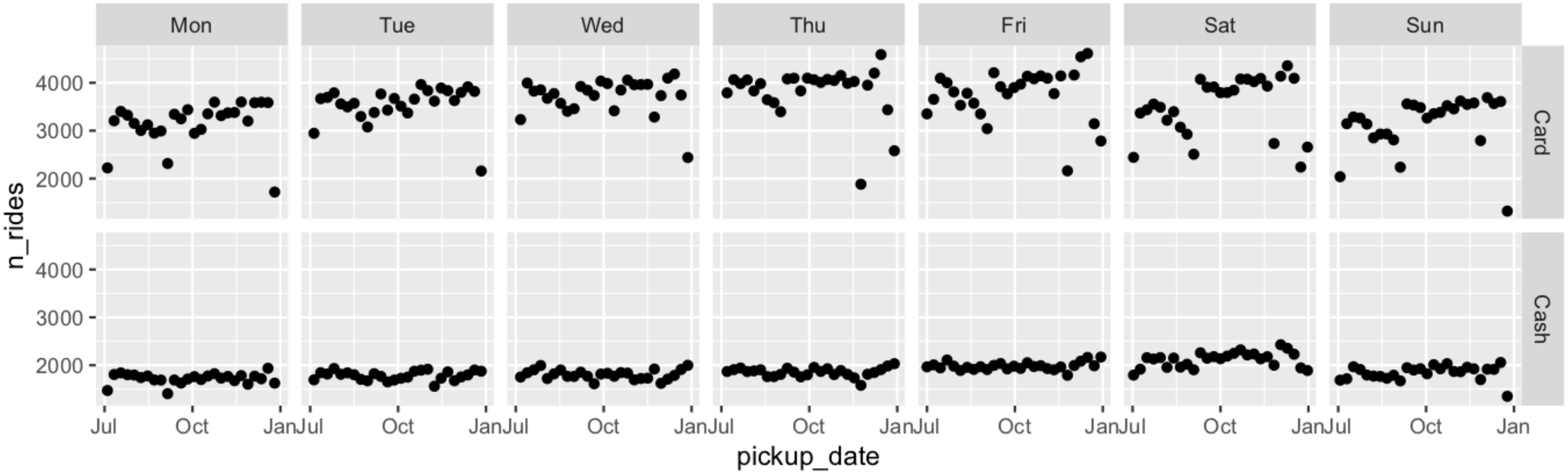


```
trelliscope(by_symbol,
  name = "candlestick_top500",
  desc = "Candlestick plot of the 500 most-traded NASDAQ stocks in 2016",
  md_desc = "
## Candlestick Plot
A [candlestick plot](https://en.wikipedia.org/wiki/Candlestick_chart)
is a financial plot...
..." )
```

Plot aspect ratio



Plot aspect ratio



Plot aspect ratio

```
trelliscope(dat, width = 600, height = 300, ...)
ggplot(...) +
  ... +
  facet_trelliscope(width = 600, height = 300, ...)
```

Default viewer state

```
trelliscope(dat, state = ..., ...)
ggplot(...) +
  ... +
  facet_trelliscope(state = ..., ...)
```

Let's practice!

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Visualizing databases of images

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Ryan Hafen
Author, TrelliscopeJS

```
pokemon
```

```
# A tibble: 801 x 30
   pokemon     id species_id height weight base_experience type_1 type_2 attack
   <chr>    <int> <chr>       <int>  <int>           <int> <chr>  <chr>   <int>
 1 bulbasa…     1 1               7     69              64 grass  poison     49
 2 ivysaur      2 2              10    130             142 grass  poison     62
 3 venusaur     3 3              20   1000             236 grass  poison     82
 4 venusau…     4 3              24   1555             281 grass  poison    100
 5 charman…     5 4               6     85              62 fire   NA         52
# ... with 796 more rows, and 21 more variables: defense <int>, hp <int>,
#   special_attack <int>, special_defense <int>, speed <int>, ability_1 <chr>,
#   ability_2 <chr>, ability_hidden <chr>, color_1 <chr>, color_2 <chr>,
#   color_f <chr>, egg_group_1 <chr>, egg_group_2 <chr>, url_image <chr>,
#   generation_id <chr>, evolves_from_species_id <chr>,
#   evolution_chain_id <chr>, shape_id <chr>, shape <chr>, pokebase <chr>,
#   pokedex <chr>
```

Image panels

```
select(pokemon, url_image)
```

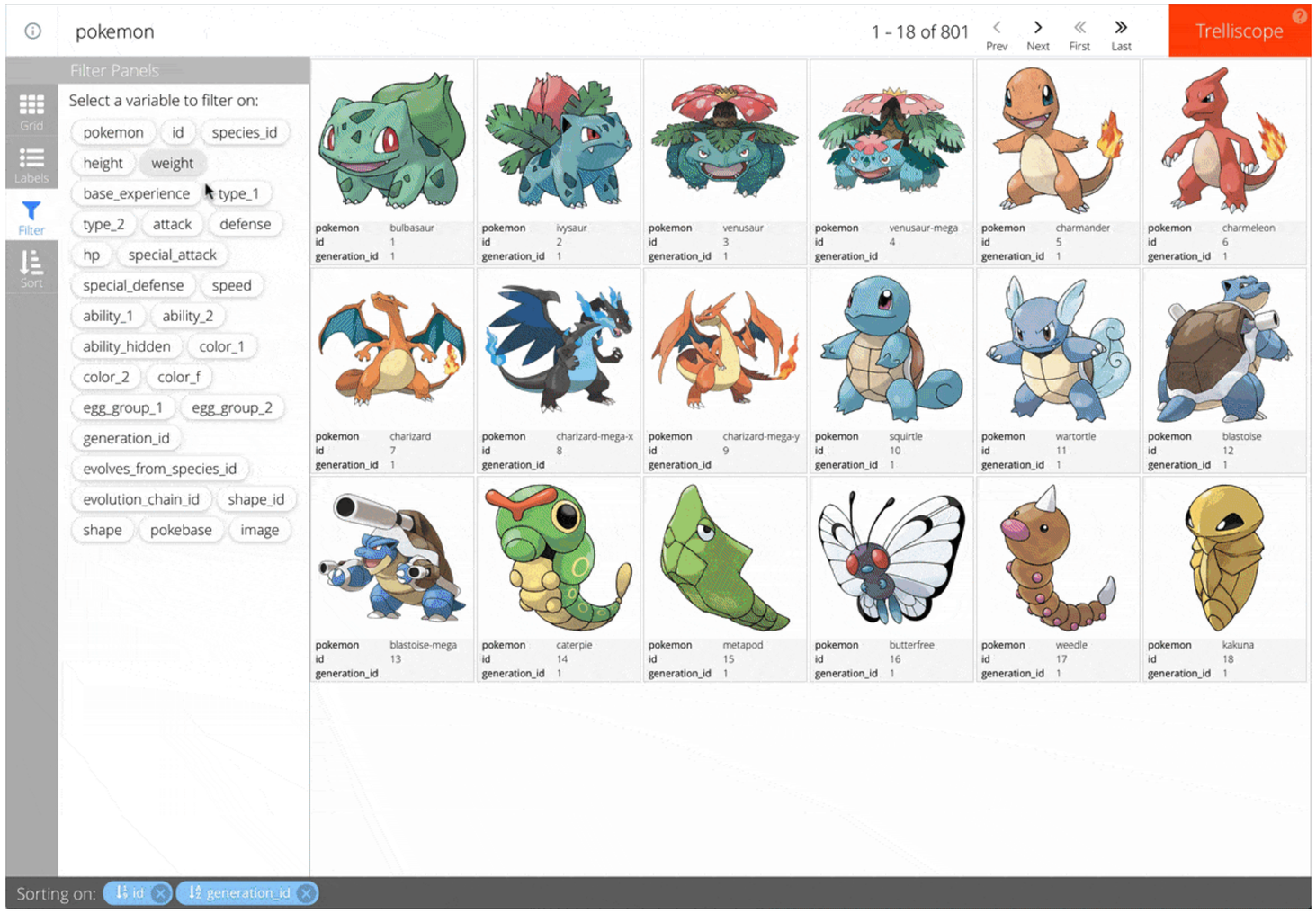
```
# A tibble: 801 x 1
   url_image
   <chr>
 1 http://assets.pokemon.com/assets/cms2/img/pokedex/full/001.png
 2 http://assets.pokemon.com/assets/cms2/img/pokedex/full/002.png
 3 http://assets.pokemon.com/assets/cms2/img/pokedex/full/003.png
 4 http://assets.pokemon.com/assets/cms2/img/pokedex/full/003_f2.png
 5 http://assets.pokemon.com/assets/cms2/img/pokedex/full/004.png
 6 http://assets.pokemon.com/assets/cms2/img/pokedex/full/005.png
 7 http://assets.pokemon.com/assets/cms2/img/pokedex/full/006.png
 8 http://assets.pokemon.com/assets/cms2/img/pokedex/full/006_f2.png
 9 http://assets.pokemon.com/assets/cms2/img/pokedex/full/006_f3.png
10 http://assets.pokemon.com/assets/cms2/img/pokedex/full/007.png
# ... with 791 more rows
```

img_panel()

```
pokemon <- mutate(pokemon,
  panel = img_panel(url_image))
```

```
trelliscope(pokemon, name = "pokemon", nrow = 3, ncol = 6)
```

A database of images



Local images

```
path <- file.path(tempdir(), "pokemon_local")
dir.create(path)
for (url in pokemon$url_image)
  download.file(url, destfile = file.path(path, basename(url)))
```

```
pokemon$image <- basename(pokemon$url_image)
pokemon <- mutate(pokemon,
  panel = img_panel_local(image))

trelliscope(pokemon, name = "pokemon", nrow = 3, ncol = 6,
  path = path)
```

Let's practice!

VISUALIZING BIG DATA WITH TRELLISCOPE IN R