Montreal BIXI bike data

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Ryan Hafen
Author, TrelliscopeJS

BIXI bikeshare data

```
# A tibble: 1,000,000 x 12
   start_date          start_station_c… end_date            end_station_code
   <dttm>                         <int> <dttm>                         <int>
 1 2017-09-18 13:35:00             6906 2017-09-18 13:40:00             6913
 2 2017-08-21 14:31:00             6316 2017-08-21 14:34:00             6316
 3 2017-06-14 16:05:00             6381 2017-06-14 16:20:00             6380
 4 2017-08-26 20:25:00             7067 2017-08-26 20:36:00             7071
 5 2017-10-04 13:29:00             6502 2017-10-04 13:38:00             6359
 6 2017-09-11 06:49:00             6161 2017-09-11 07:03:00             6008
 7 2017-07-24 16:46:00             6406 2017-07-24 16:53:00             7052
 8 2017-10-10 14:15:00             6174 2017-10-10 14:24:00             6078
 9 2017-08-12 15:51:00             6249 2017-08-12 15:56:00             6329
10 2017-06-02 07:37:00             6363 2017-06-02 08:06:00             6043
# ... with 999,990 more rows, and 8 more variables: duration_sec <int>,
#   start_day <date>, start_dow <fct>, weekday <fct>, start_hod <dbl>,
#   start_mon <dbl>, start_wk <dbl>, membership <fct>
```

Let's dive in!

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Summary visualization recap

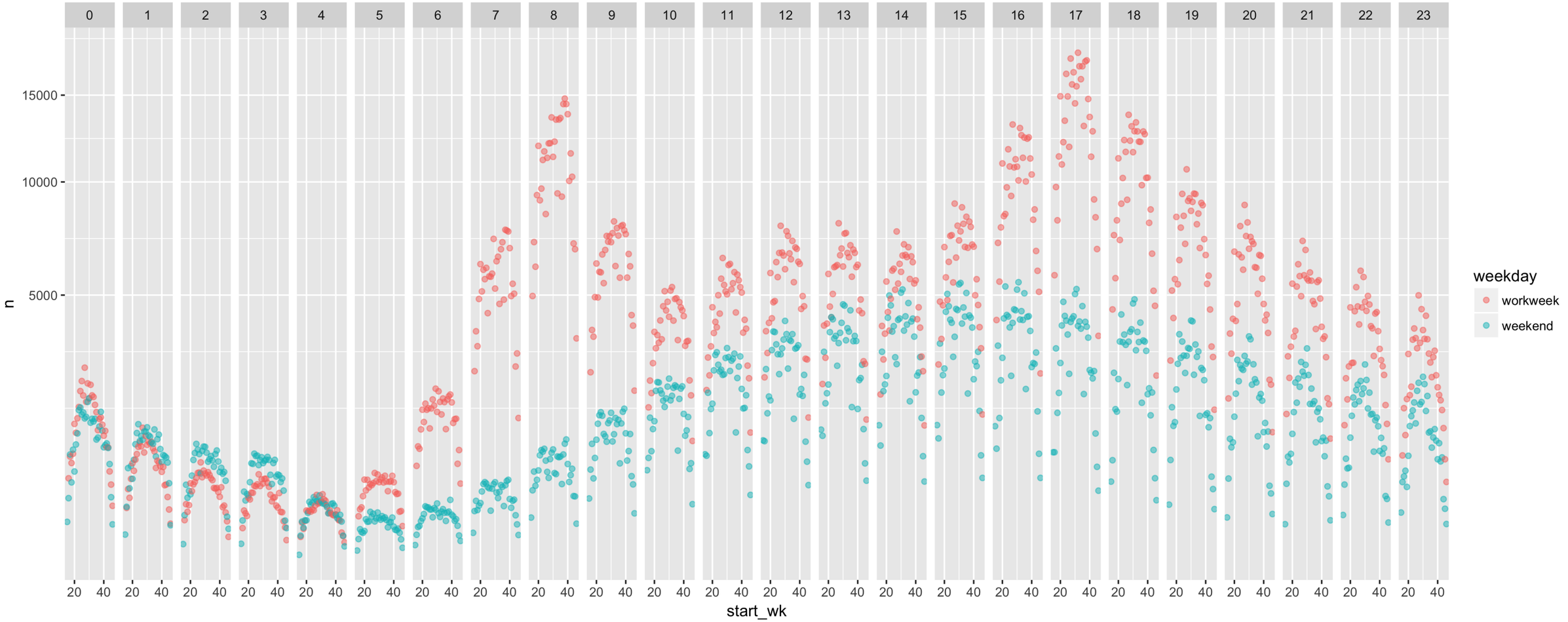
VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Ryan Hafen
Author, TrelliscopeJS

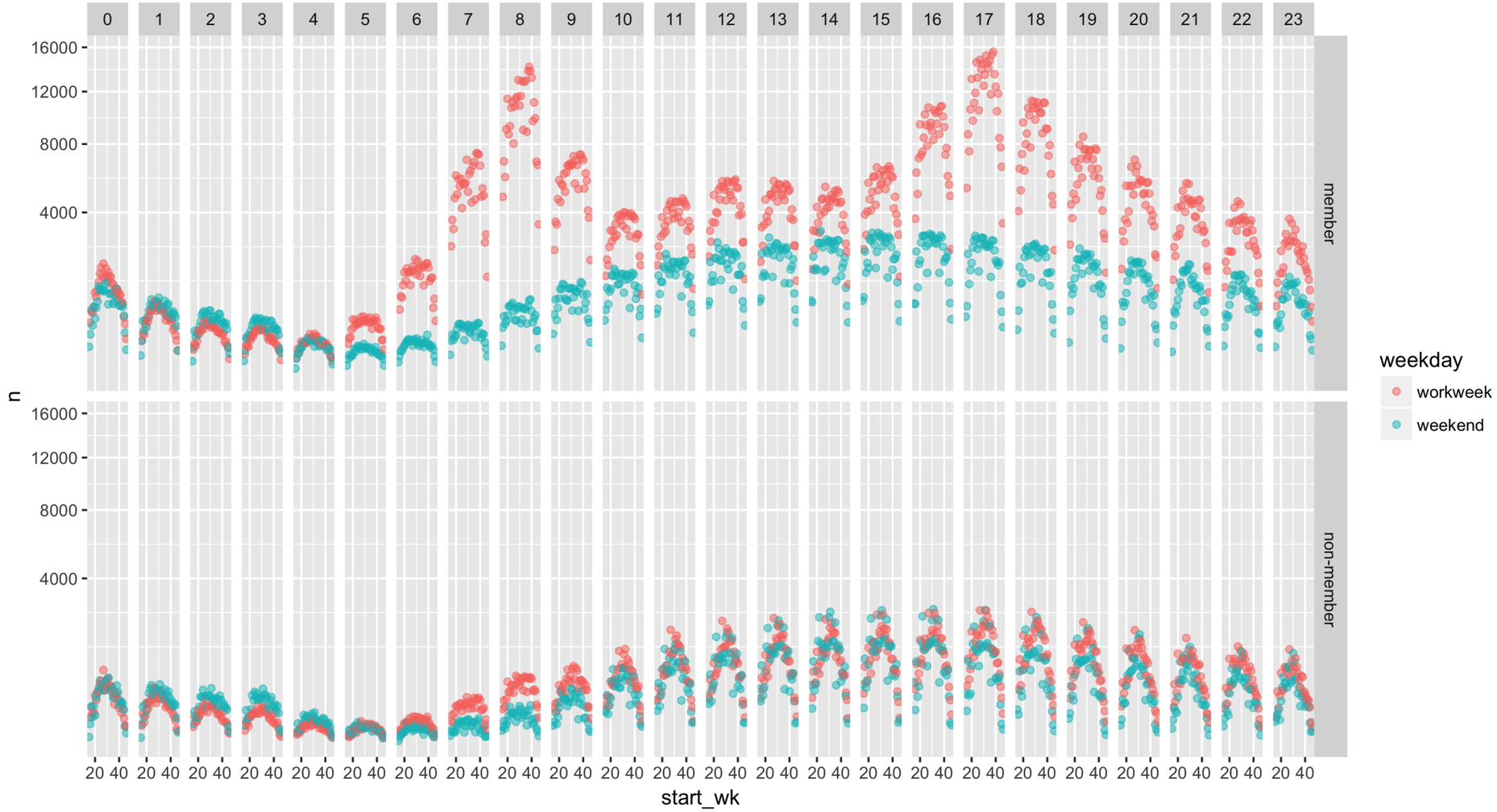
Daily rides


40000
30000
20000
10000
n
May
Jun
Jul
Aug
Sep
Oct
Nov
start_day
weekday
workweek
weekend

Hourly rides over time



Hourly rides over time + membership



Diving deeper

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Top 100 routes dataset

VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Ryan Hafen
Author, TrelliscopeJS

Studying routes

Route frequency

```
route_tab <- bike %>%
  filter(start_station_code != end_station_code) %>%
  group_by(start_station_code, end_station_code) %>%
  summarise(n = n()) %>%
  arrange(-n)
```

```
# A tibble: 193,632 x 3
# Groups:   start_station_code [546]
   start_station_code end_station_code     n
                <int>            <int> <int>
 1               6050             6406  2428
 2               6406             6052  2364
 3               6136             6163  2352
 4               6052             6026  2001
 5               6052             6406  1983
# ... with 193,627 more rows
```

```
top_routes <- paste(
  route_tab$start_station_code[1:100],
  route_tab$end_station_code[1:100])
top100 <- bike %>%
  filter(paste(start_station_code, end_station_code) %in% top_routes)
```

```
# A tibble: 133,786 x 12
   start_date          start_station_code end_date            end_station_code
   <dttm>                           <int> <dttm>                         <int>
 1 2017-04-15 00:10:00               6386 2017-04-15 00:13:00             6393
 2 2017-04-15 00:20:00               6221 2017-04-15 00:24:00             6184
 3 2017-04-15 00:42:00               6206 2017-04-15 00:45:00             6411
 4 2017-04-15 00:48:00               6350 2017-04-15 00:50:00            10002
 5 2017-04-15 02:09:00               6070 2017-04-15 02:12:00             6205
 6 2017-04-15 02:07:00               6221 2017-04-15 02:11:00             6184
# ... with 133,780 more rows, and 8 more variables: duration_sec <int>,
#   start_day <date>, start_dow <fct>, weekday <fct>, start_hod <dbl>,
#   start_mon <dbl>, start_wk <dbl>, membership <fct>
```

```
route_hod <- top100 %>%
  group_by(start_station_code, end_station_code, start_hod, weekday) %>%
  summarise(n = n())
# join station metadata
route_hod <- route_hod %>%
  left_join(start_stations) %>%
  left_join(end_stations)
```

```
# A tibble: 4,114 x 11
# Groups:   start_station_code, end_station_code, start_hod [?]
   start_station_co… end_station_code start_hod weekday     n start_station_na…
               <int>            <int>     <dbl> <fct>   <int> <chr>
 1              6012             6015      0    workwe…    12 Métro St-Laurent…
 2              6012             6015      0    weekend    13 Métro St-Laurent…
 3              6012             6015      1.00 workwe…    11 Métro St-Laurent…
 4              6012             6015      1.00 weekend     2 Métro St-Laurent…
 5              6012             6015      2.00 workwe…     2 Métro St-Laurent…
 6              6012             6015      2.00 weekend     6 Métro St-Laurent…
# ... with 4,108 more rows, and 5 more variables: start_lat <dbl>,
#   start_lon <dbl>, end_station_name <chr>, end_lat <dbl>, end_lon <dbl>
```

Let's visualize!

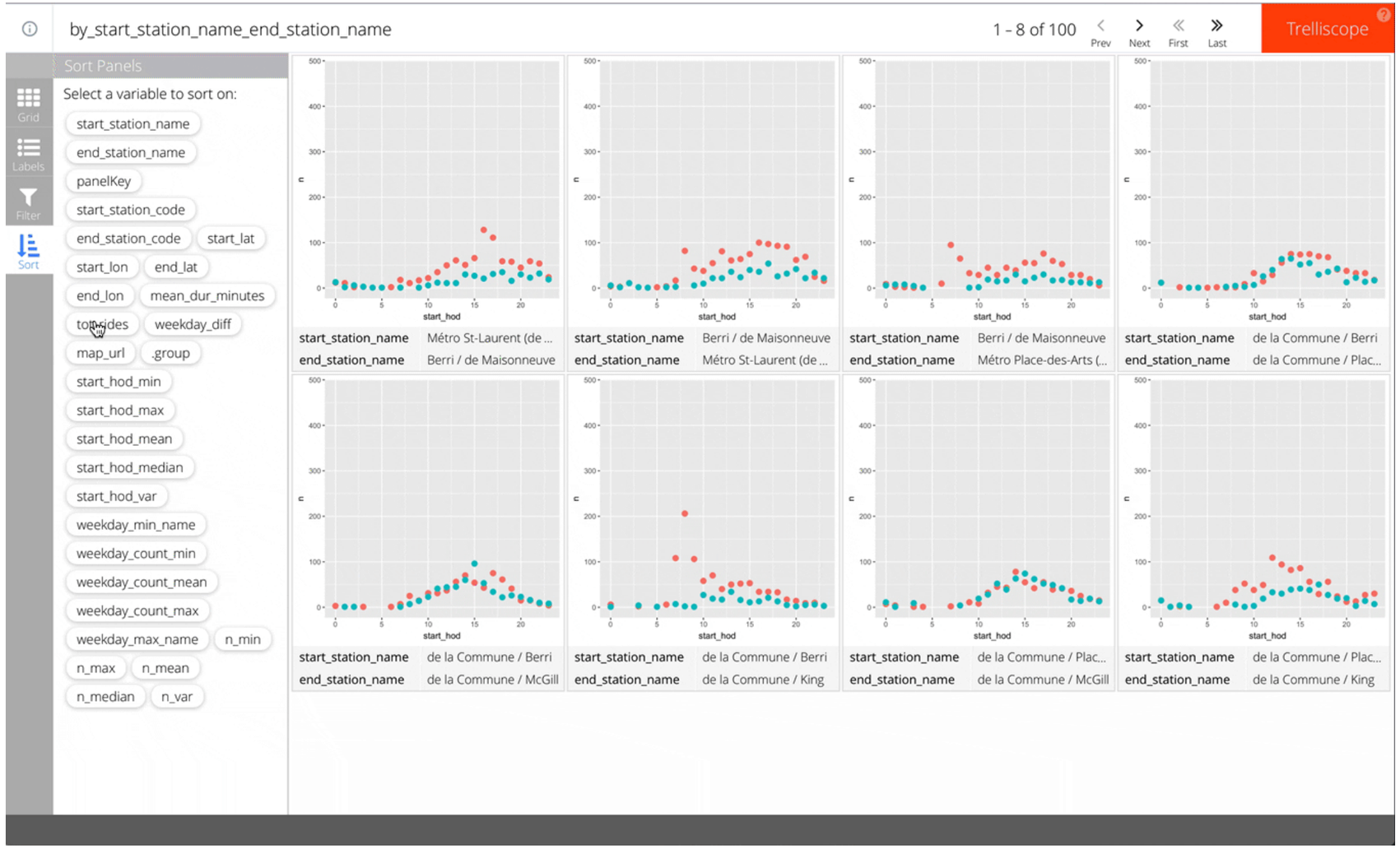
VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Au revoir

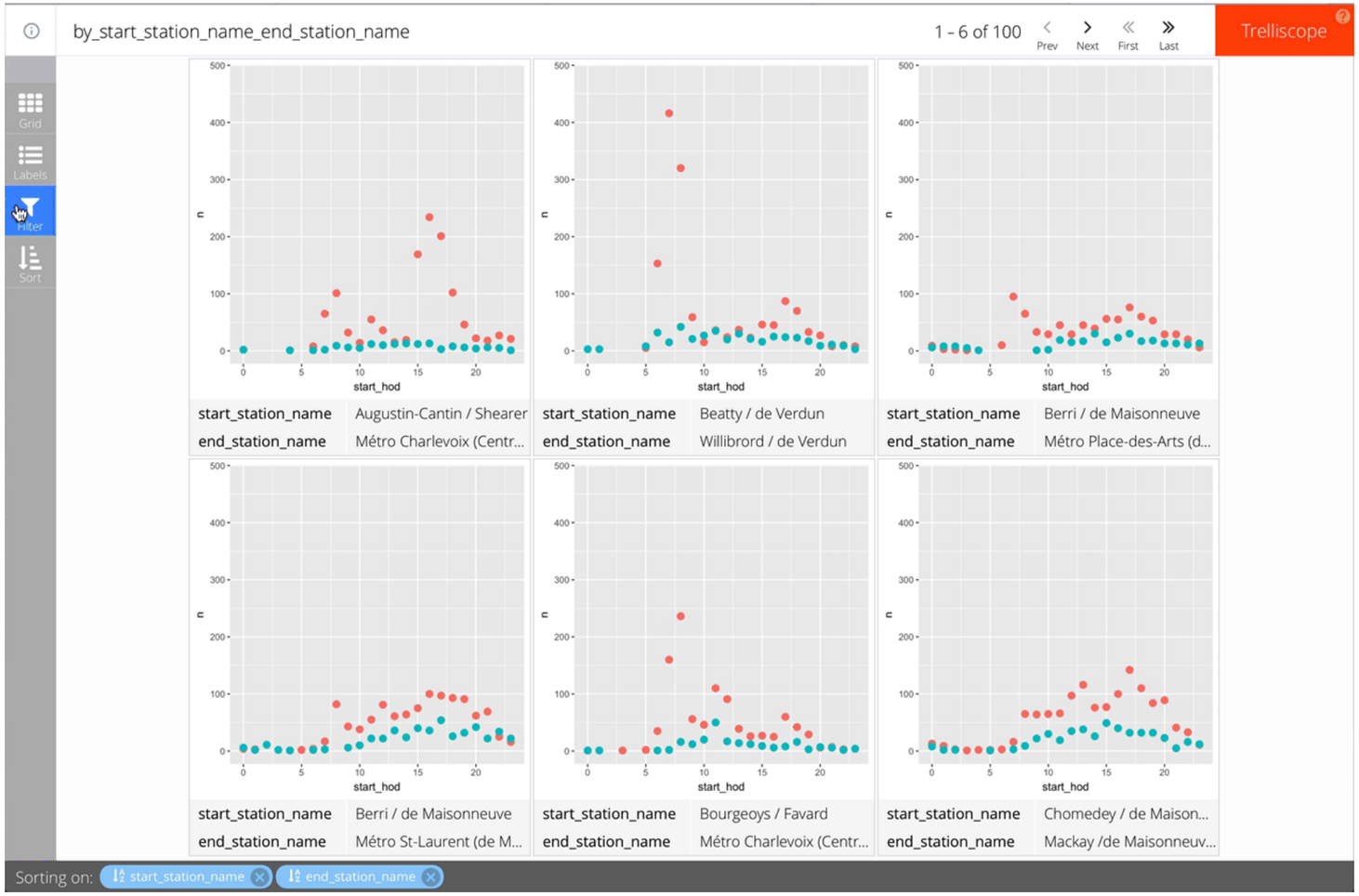
VISUALIZING BIG DATA WITH TRELLISCOPE IN R

Ryan Hafen
Author, TrelliscopeJS

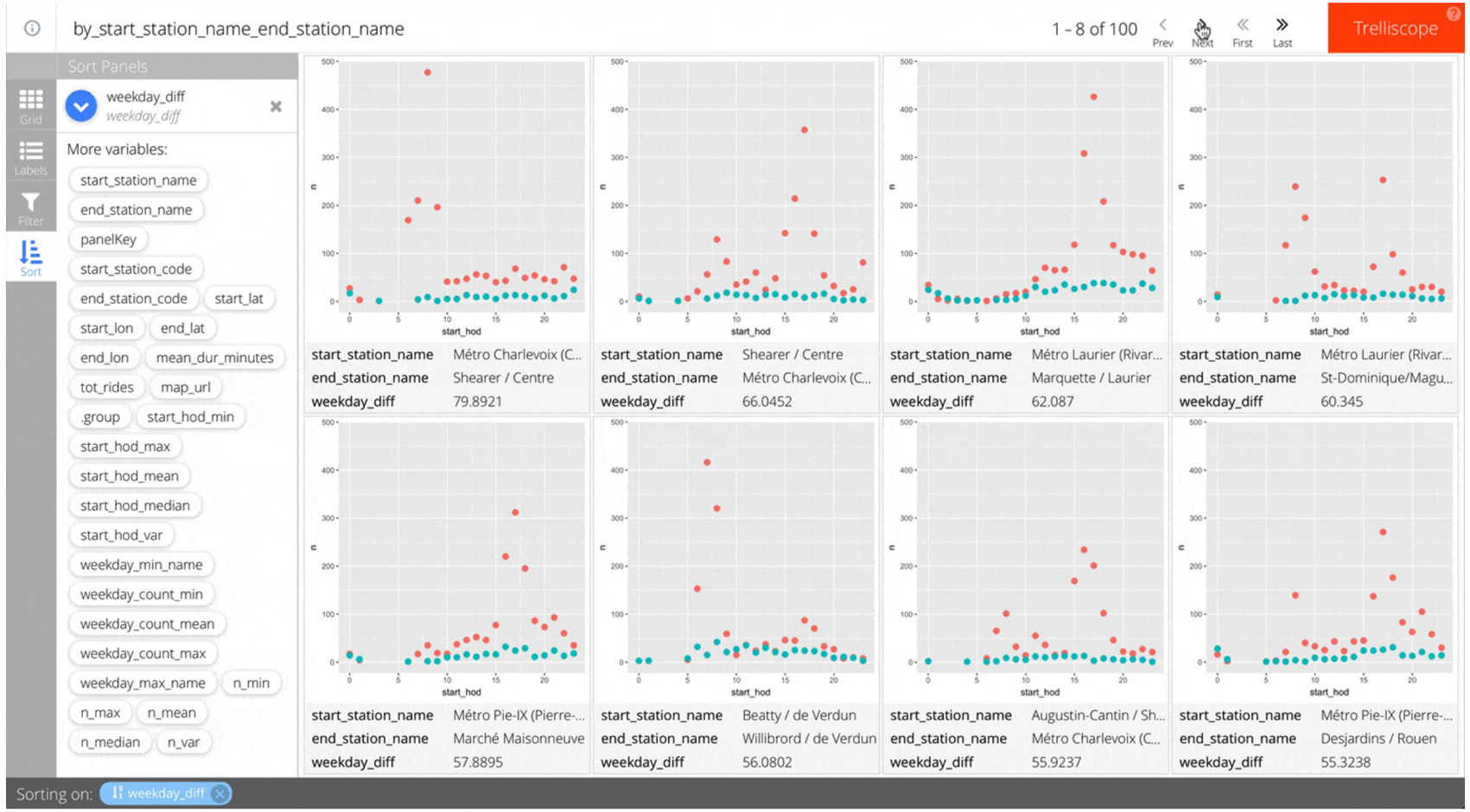
The most popular route



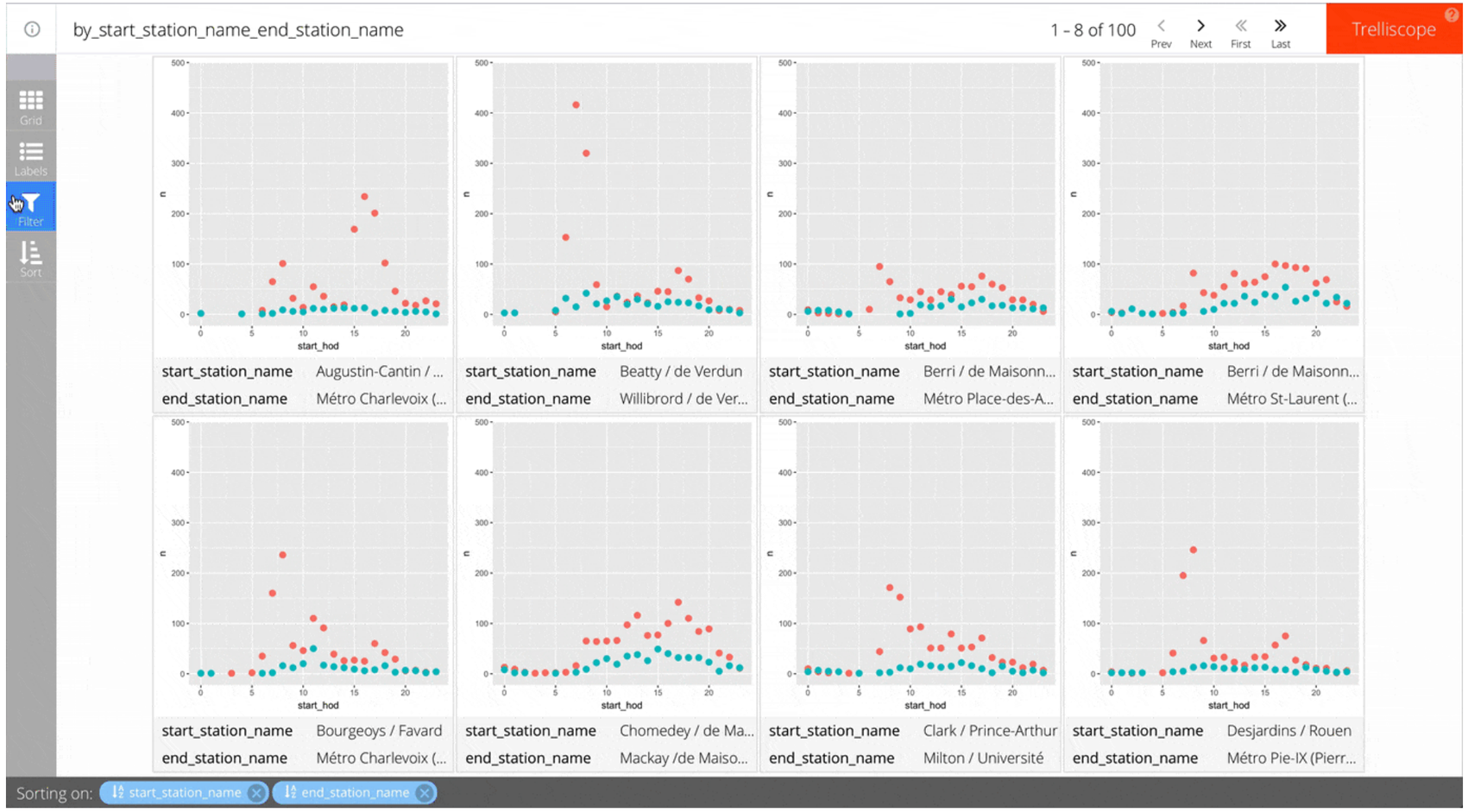
Prominent stations



Commuter vs. non-commuter



Commuter routes are short



More displays

Resources

- Documentation: **https://hafen.github.io/trelliscopejs/**
- Github: **https://github.com/hafen/trelliscopejs**
- Blog: **http://ryanhafen.com/blog/**

Congratulations!

VISUALIZING BIG DATA WITH TRELLISCOPE IN R