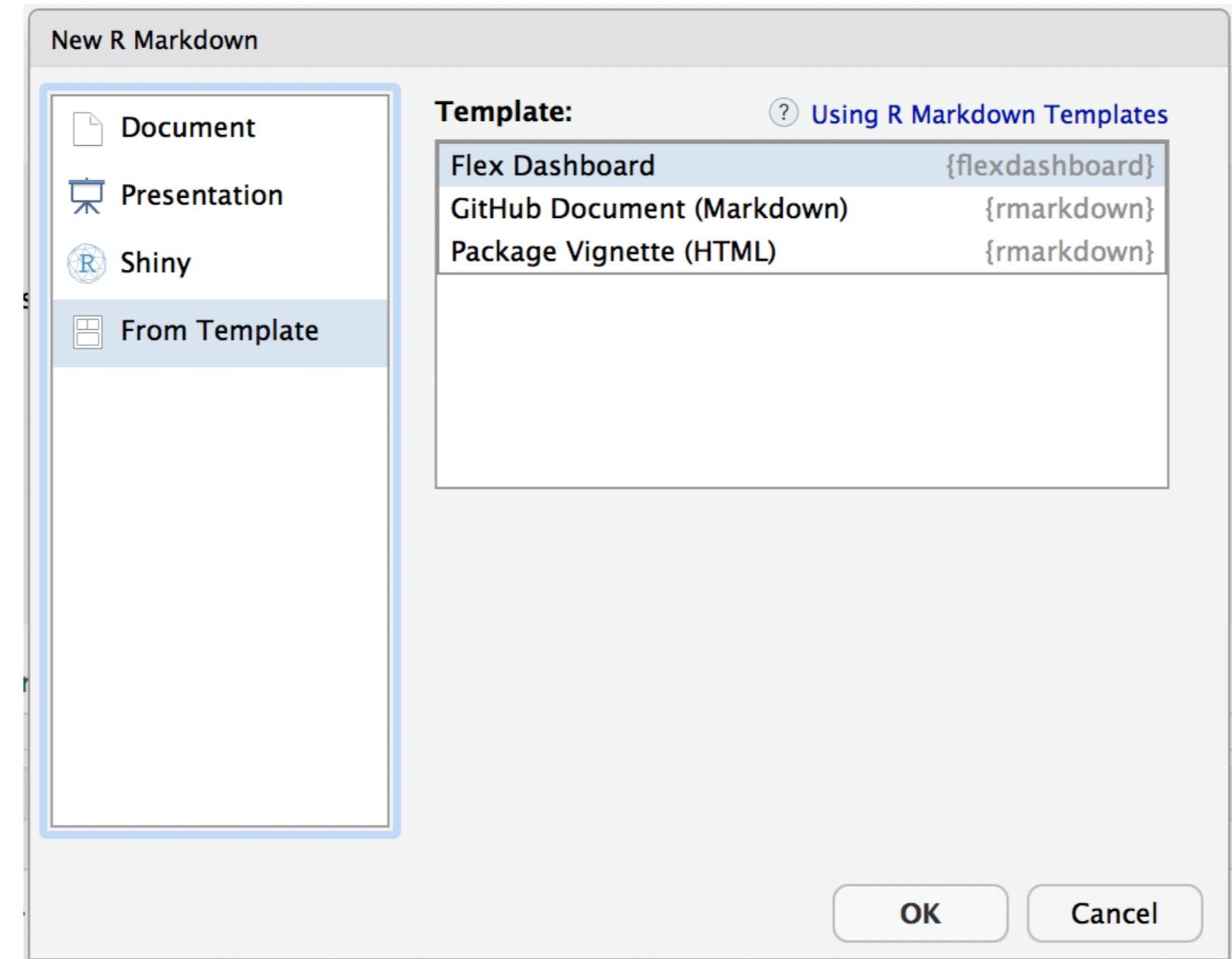


Knit!



... and charts are arranged in columns

```
---
```

```
title: "Bikeshare"
```

```
output:
```

```
  flexdashboard::flex_dashboard:
```

```
    orientation: columns
```

```
--|
```

```
Column {data-width=250}
```

```
---
```

```
## Chart A
```

```
```{r}
```

```
...
```

```
Column {data-width=500}
```

```

```

```
Chart B
```

```
```{r}
```

```
...
```

```
## Chart C
```

```
```{r}
```

```
...
```

```
Column {data-width=250}
```

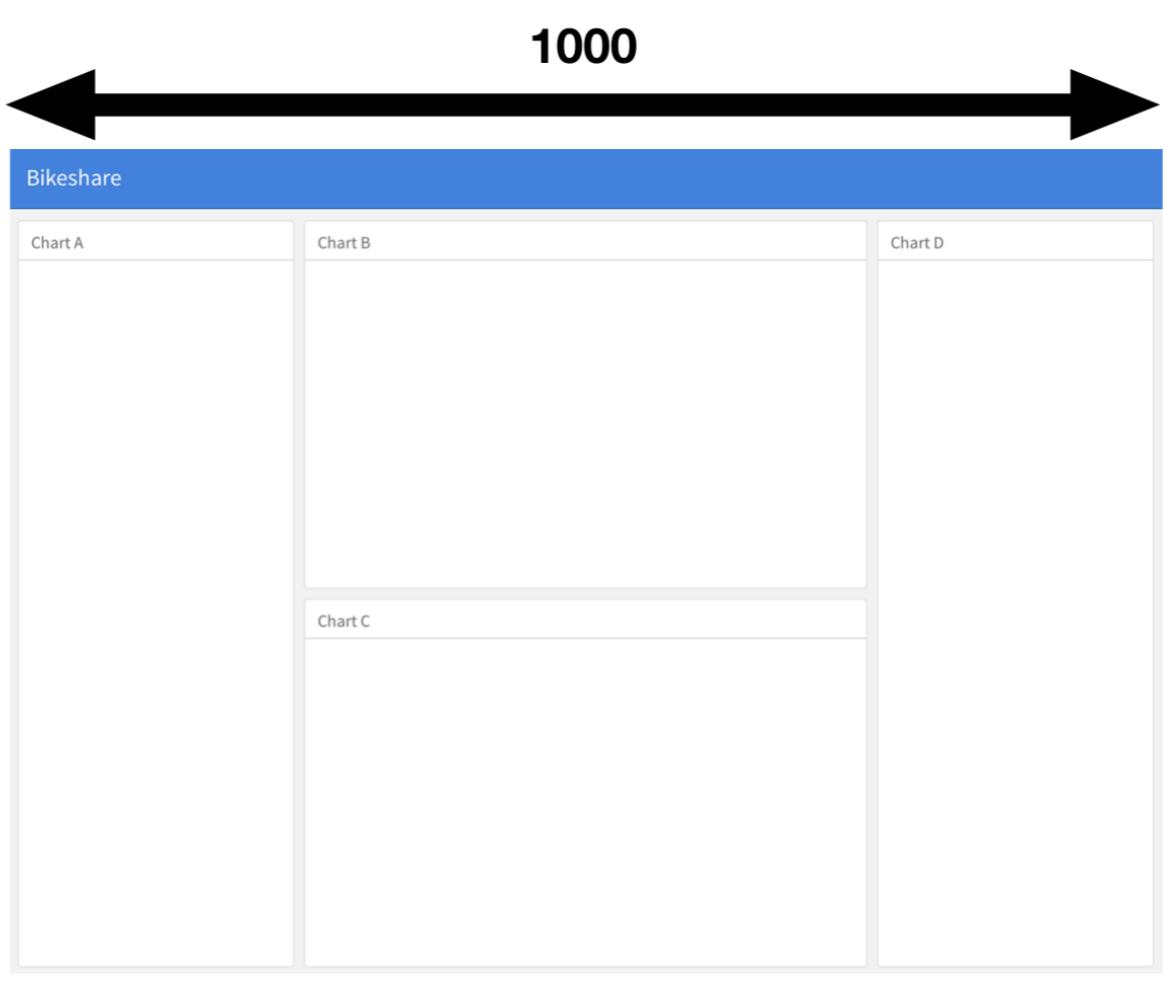
```

```

```
Chart D
```

```
```{r}
```

```
...
```



Row layout

```
---
```

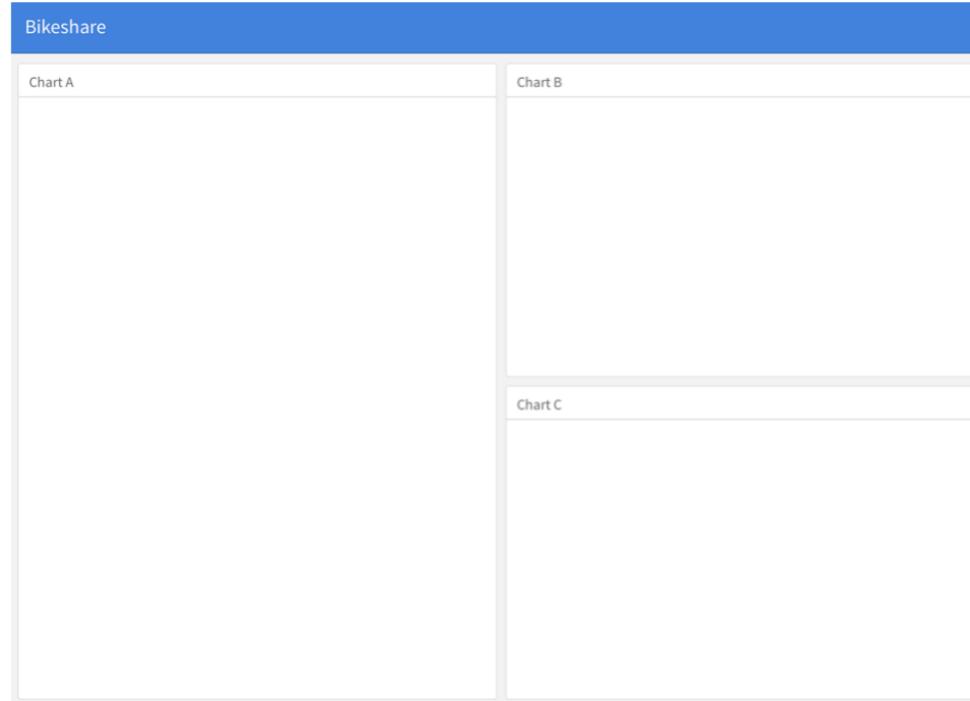
```
title: "Bikeshare"
```

```
output:
```

```
  flexdashboard::flex_dashboard:
```

```
    orientation: columns
```

```
---
```



```
---
```

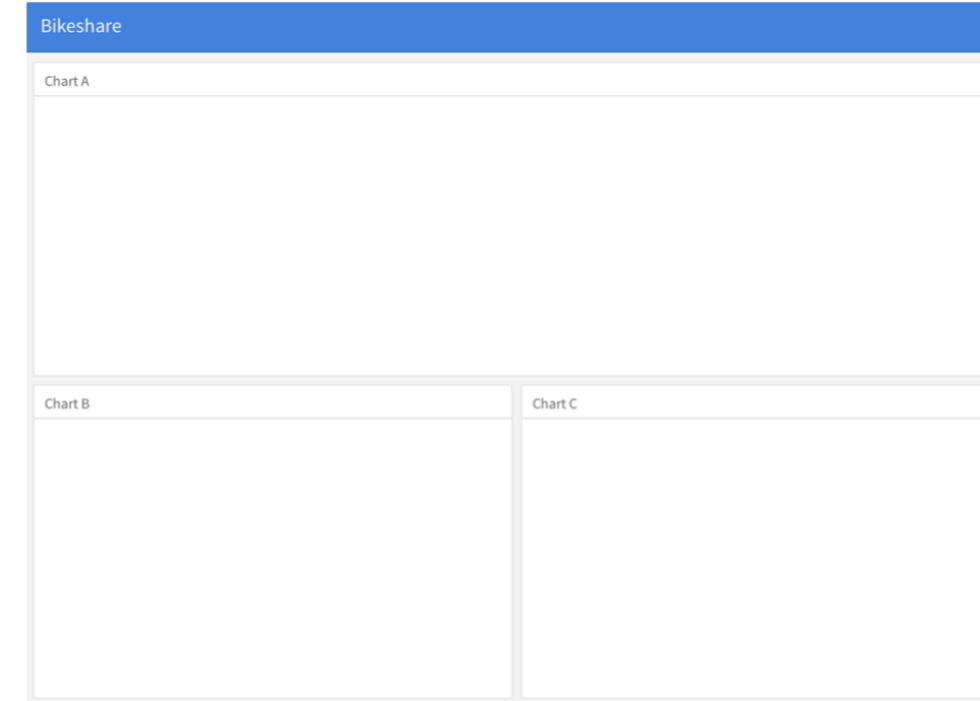
```
title: "Bikeshare"
```

```
output:
```

```
  flexdashboard::flex_dashboard:
```

```
    orientation: rows
```

```
---
```



Scrolling

```
title: "Bikeshare"  
output:  
  flexdashboard::flex_dashboard:  
    orientation: columns  
    vertical_layout: scroll
```

It isn't the best option.

Converting to a Storyboard

```
---
```

```
title: "Bike Shares Daily"
output:
  flexdashboard::flex_dashboard:
    orientation: columns
    vertical_layout: fill
    storyboard: true
---
```

```
### Where do bike share trips start?
```

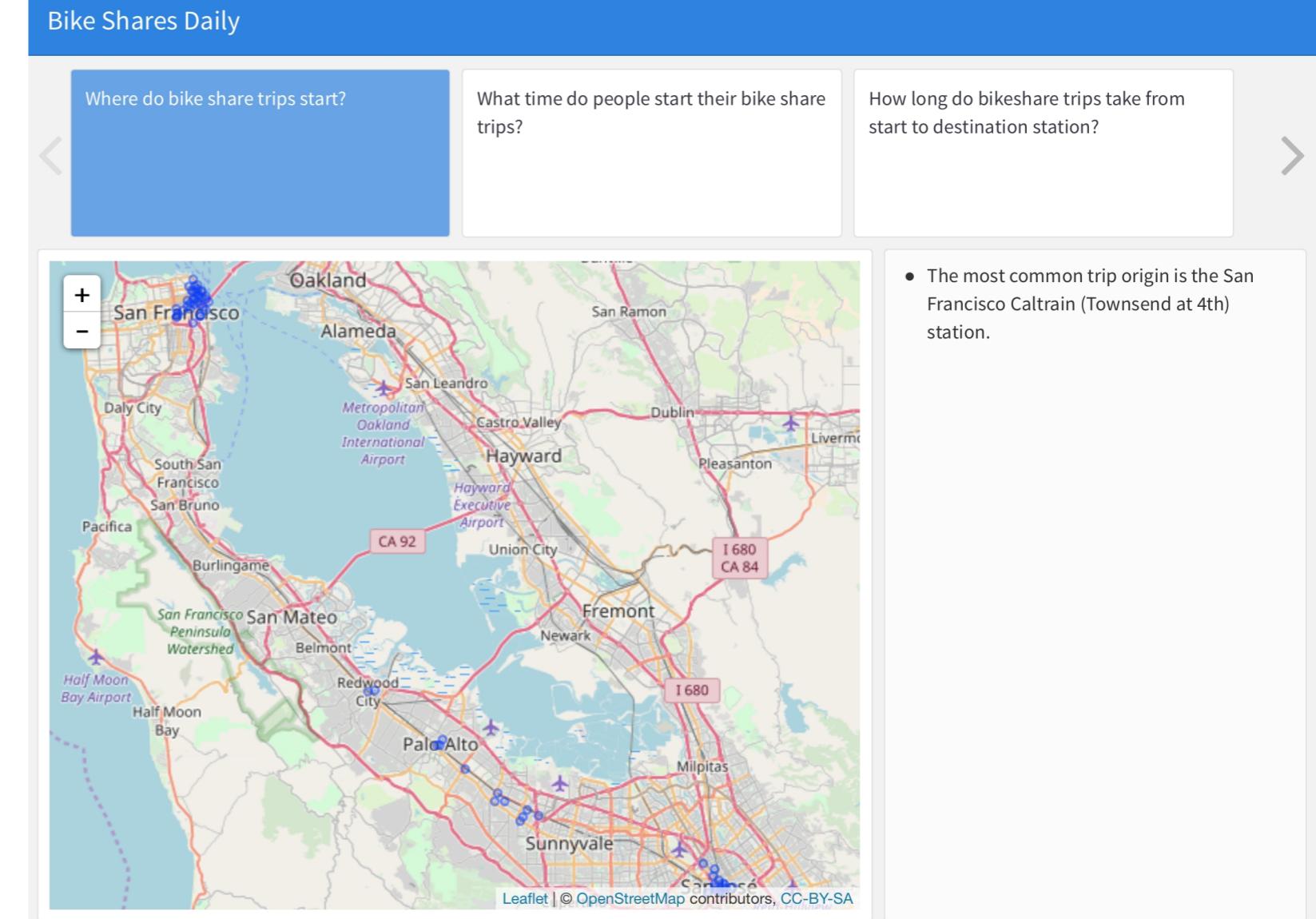
```
```{r}
```

```
...
```

```
What time do people start their bike share trips?
```

```
```{r}
```

```
...
```



Adding Storyboard commentary

```
---
```

```
title: "Bike Shares Daily"
output:
  flexdashboard::flex_dashboard:
    orientation: columns
    vertical_layout: fill
    storyboard: true
---
```

```
### Where do bike share trips start?
```

```
```{r}
```

```
...
```

```

```

Origins and destinations are all station locations.

```
What time do people start their bike share trips?
```

```
```{r}
```

```
...
```

Mixing Storyboards In a page

```
Origin Destination Pairs {.storyboard}
```

```
=====
```

```
### O/D Flow
```

```
```{r}
```

```
...
```

```
Trips by Station
```

```
```{r}
```

```
...
```

```
Trip Raw Data
```

```
=====
```

```
### Trip-Level Data
```

```
```{r}
```

# Tabsets

Column `{.tabset}`

```
Chart A
```{r}
```
Chart B
```{r}
```
Chart C
```{r}
```
Column


```
### Chart D
```{r}
```
### Chart E
```{r}
```

```


The image shows a dashboard titled "Bikeshare". At the top, there is a navigation bar with three tabs: "Chart A", "Chart B", and "Chart C". The "Chart B" tab is highlighted with a black border. Below the navigation bar, the dashboard is divided into two main sections. The top section contains three charts labeled "Chart A", "Chart B", and "Chart C" from left to right. The bottom section contains two charts labeled "Chart D" and "Chart E" from left to right.


```

# Adding Storyboard commentary

```

```

```
title: "Bike Shares Daily"
output:
 flexdashboard::flex_dashboard:
 orientation: columns
 vertical_layout: fill
 storyboard: true

```

```
Where do bike share trips start?
```

```
```{r}
```

```
...
```

```
***
```

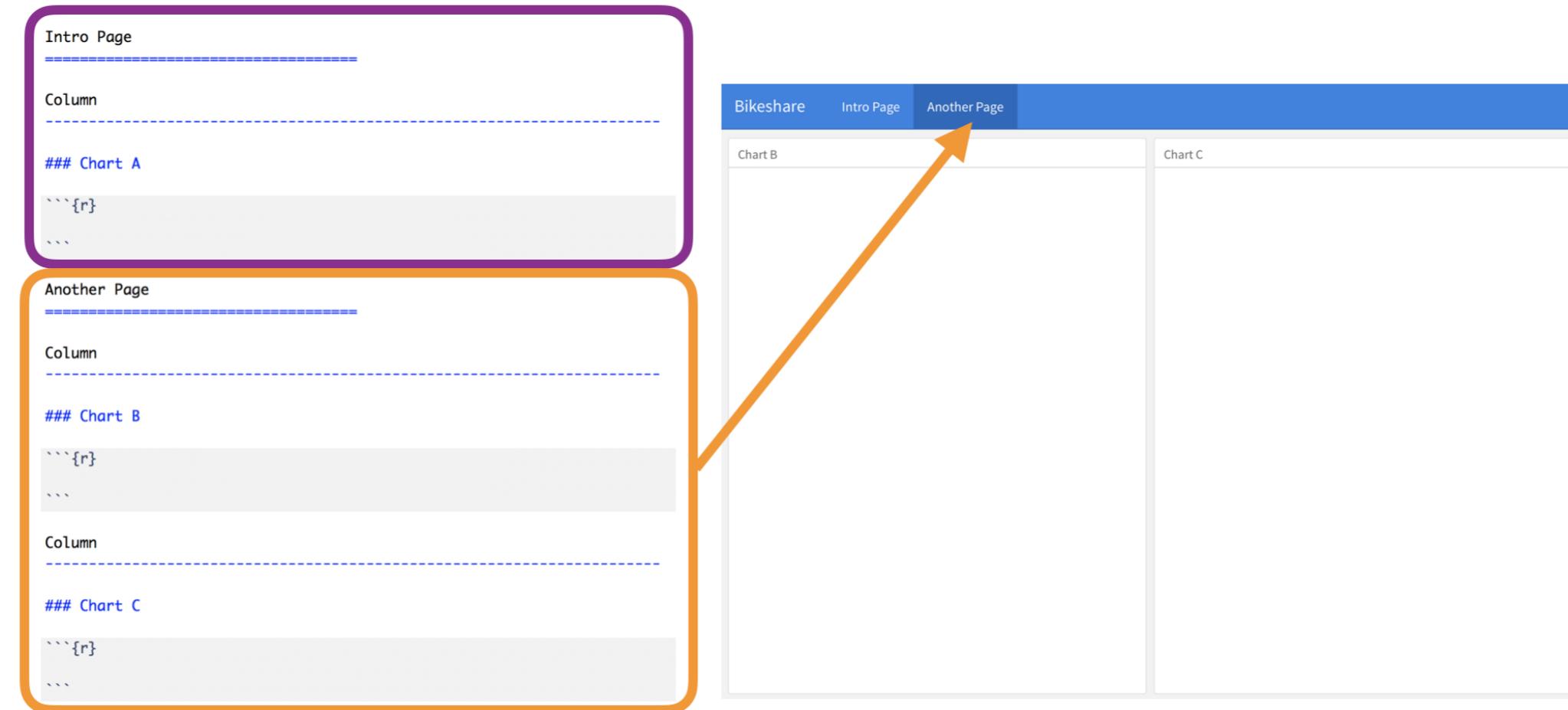
Origins and destinations are all station locations.

```
### What time do people start their bike share trips?
```

```
```{r}
```

```
...
```

# Pages



# Creating menus

```
Intro Page

Column

Chart A
```{r}
```
...
Page 2 {data-navmenu=Details}

Column

Chart B
```{r}
```
...
Page 3 {data-navmenu=Details}

Column

Chart C
```{r}
```
...
Page 4 {data-navmenu="Even More"}

Column

Chart D
```{r}
```
...
```

Bikeshare   Intro Page   Details ▾   Even More ▾

Chart A   Page 2  
Page 3

# Setting page orientation

```
Intro Page
=====
```

```
Column
=====
```

```
Chart A
```

```
```{r}
```

```
```
```

```
Another Page {data-orientation=rows}
=====
```

```
Column
=====
```

```
Chart B
```

```
```{r}
```

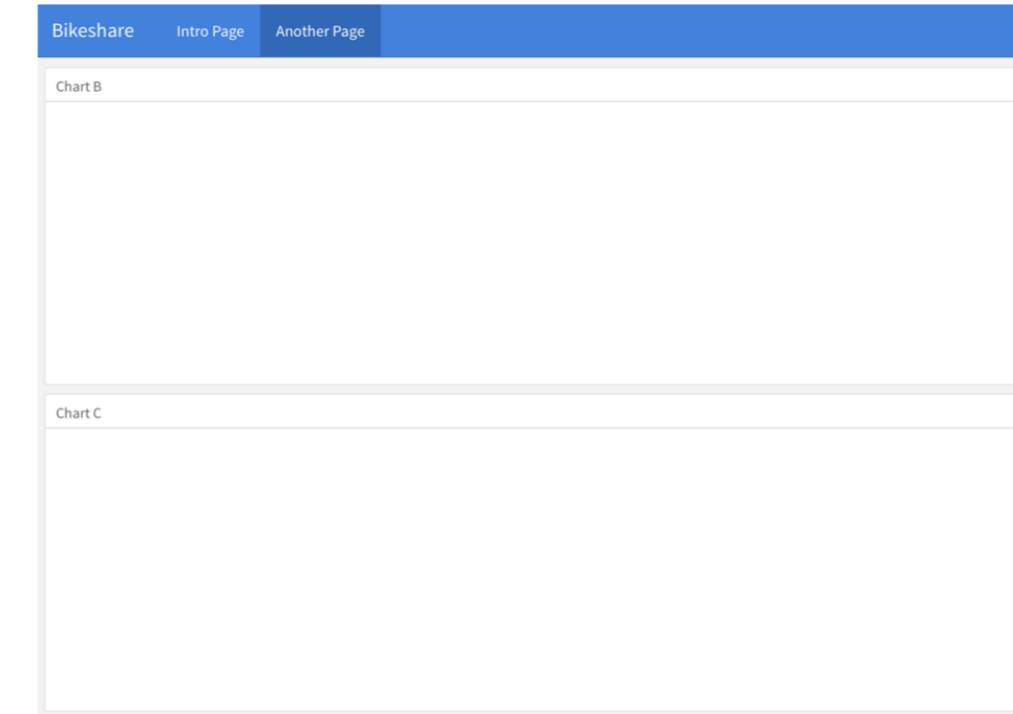
```
```
```

```
Column
=====
```

```
Chart C
```

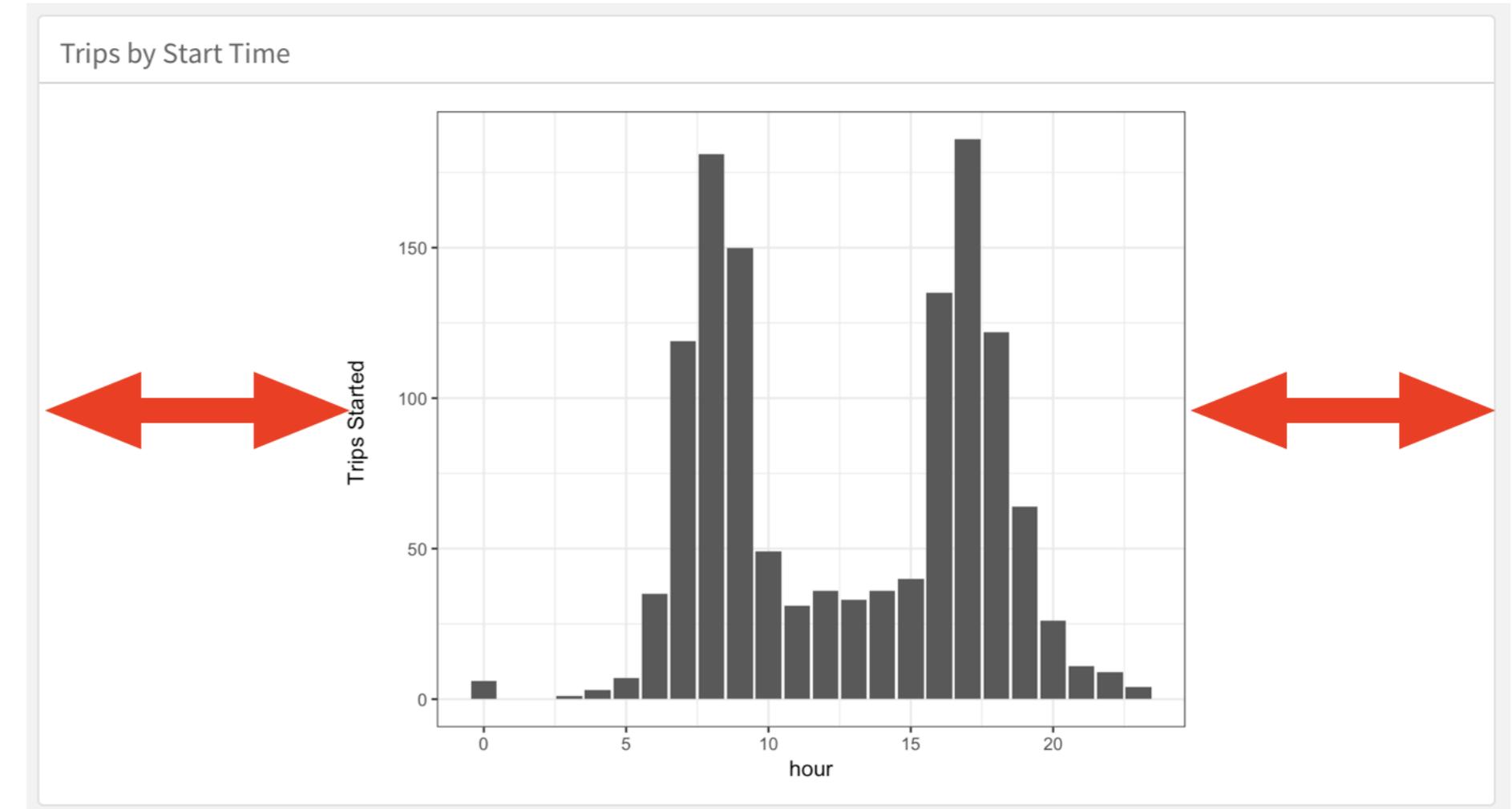
```
```{r}
```

```
```
```



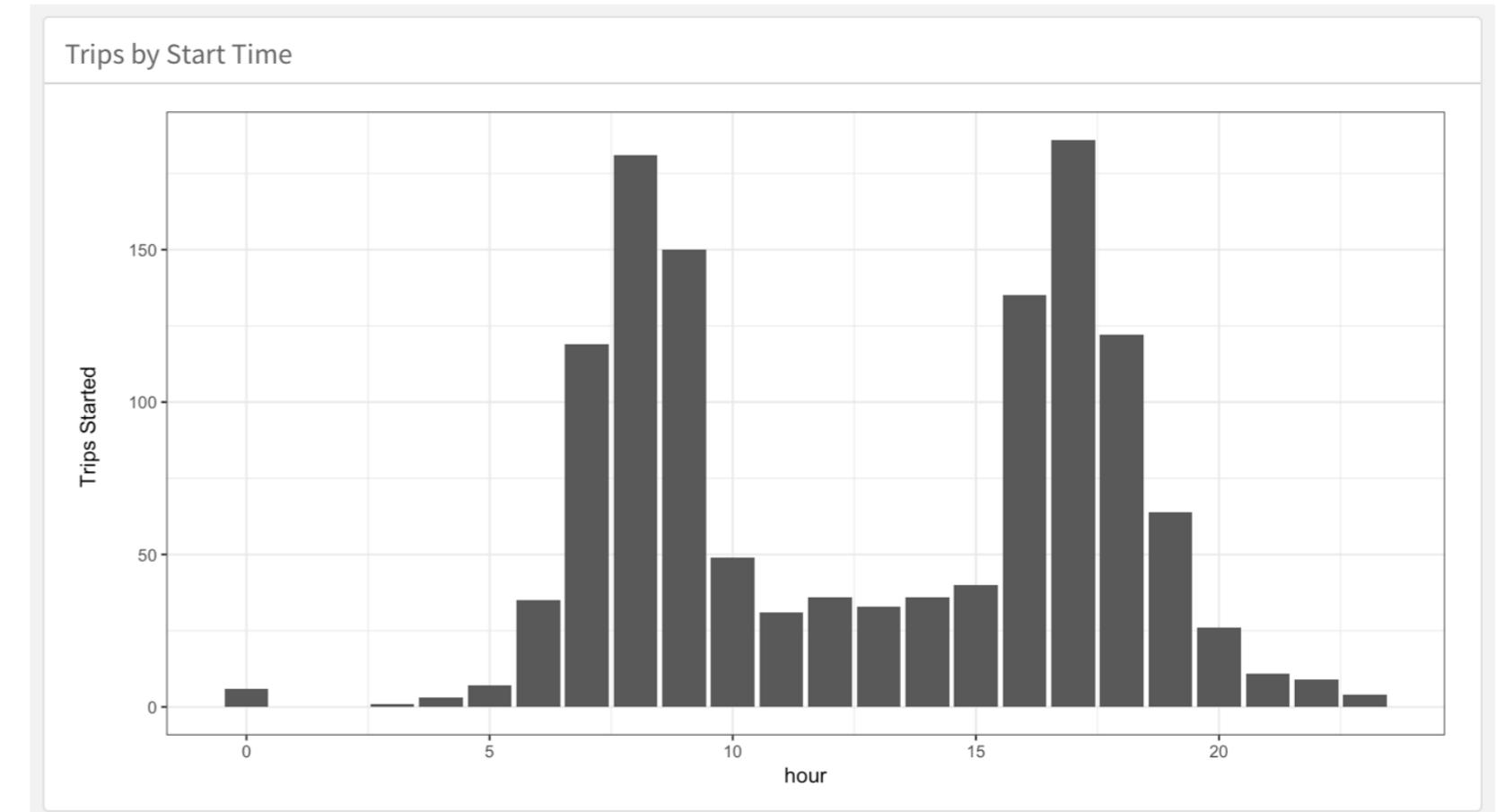
# Adding graphs: the basics

```
My Graph Chart
```{r}  
plot(trips_df$time, trips_df$rides)  
```
```



# Resizing graphs

```
My Graph Chart
`~`{r, fig.width=10, fig.height=5}
plot(trips_df$time, trips_df$rides)
`~`
```



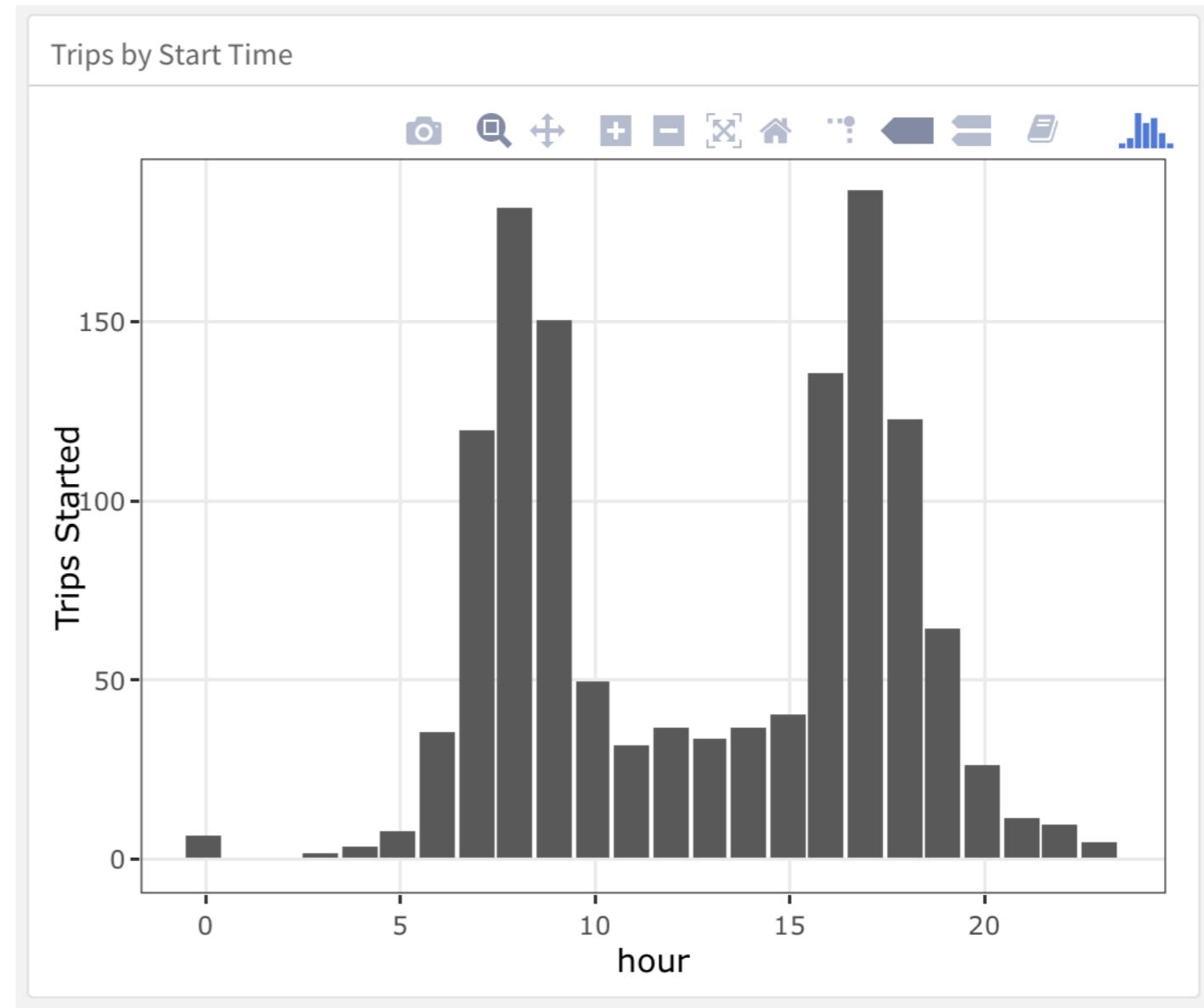
# The magic of plotly

```
library(plotly)

ggplotly(my_ggplot)
```

## Web-friendly options

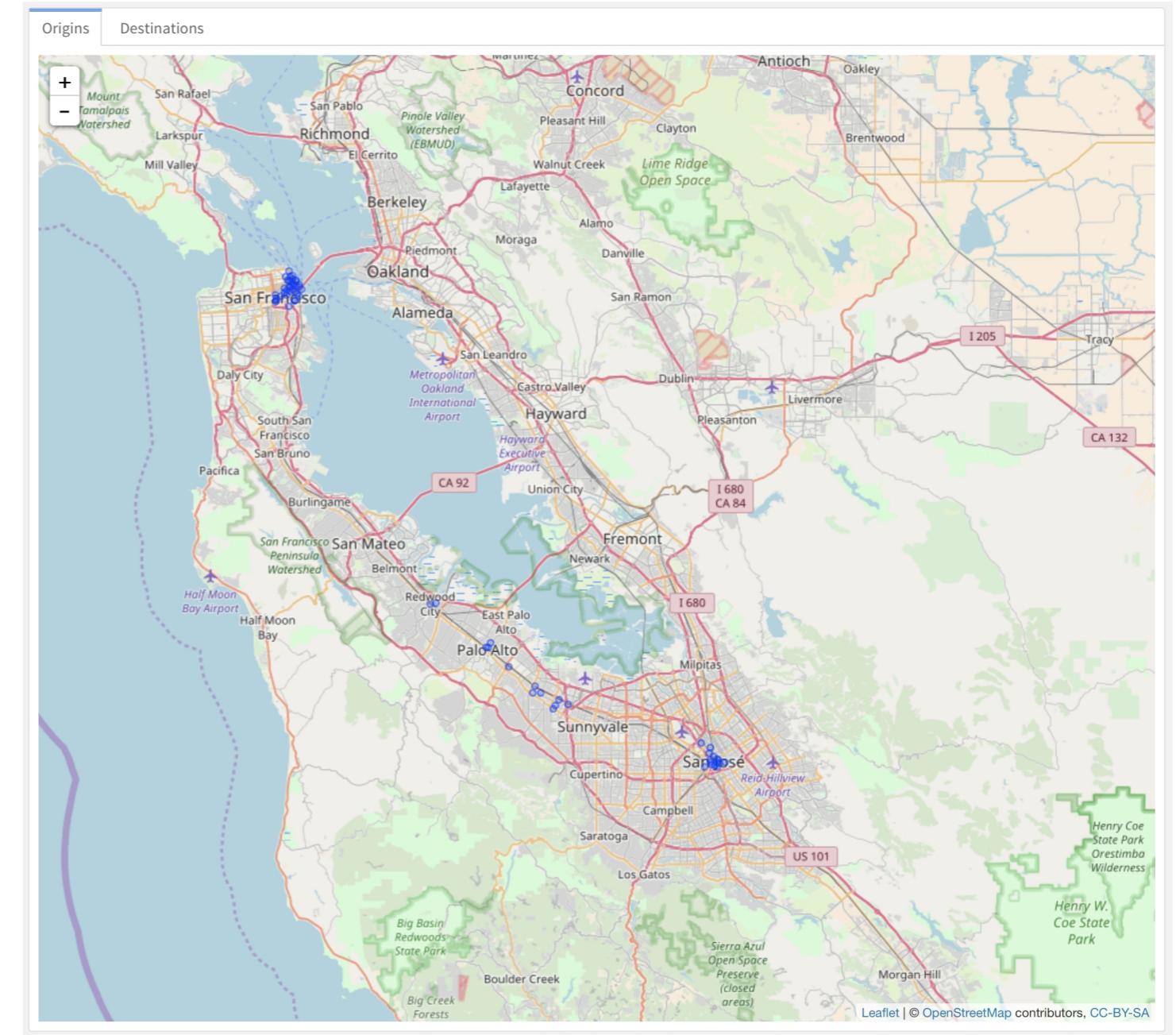
- plotly
- highcharter
- dygraphs
- rbokeh
- ggvis



# leaflet with Dataframes

```
library(leaflet)
```

```
leaflet(data_df) %>%
addTiles() %>%
addMarkers()
```



```
Total Trips
```

```
```{r}  
num_trips <- nrow(trips_df)  
  
valueBox(num_trips)  
  
```
```

1284

Total Trips

```
Total Trips
```

```
```{r}  
num_trips <- nrow(trips_df)  
  
valueBox(num_trips,  
         caption = 'Total Daily Trips',  
         icon = 'fa-bicycle')  
  
```
```

1284

Total Daily Trips



```
Total Trips
```

```
```{r}  
num_trips <- nrow(trips_df)  
  
valueBox(prettyNum(num_trips, big.mark = ','),  
         caption = 'Total Daily Trips',  
         icon = 'fa-bicycle')  
  
```
```

1,284

Total Daily Trips



# Gauges

```
gauge(value = pct_subscriber_trips,
 min = 0,
 max = 100)

gauge(value = pct_subscriber_trips,
 min = 0,
 max = 100,
 sectors = gaugeSectors(
 success = c(90, 100),
 warning = c(70, 89),
 danger = c(0, 69)
),
 symbol = '%')
```

# Links

```
valueBox(prettyNum(num_trips, big.mark = ','),
 caption = 'Total Daily Trips',
 icon = 'fa-bicycle',
 href = '#trip-raw-data')
```

# Adding buttons

```
library(DT)

datatable(
 my_data_df, rownames = FALSE,
 extensions = 'Buttons', options = list(
 dom = 'Bfrtip',
 buttons = c('copy', 'csv', 'excel', 'pdf', 'print')
)
)
```

# Captions with inline code

```
Trip Durations
```

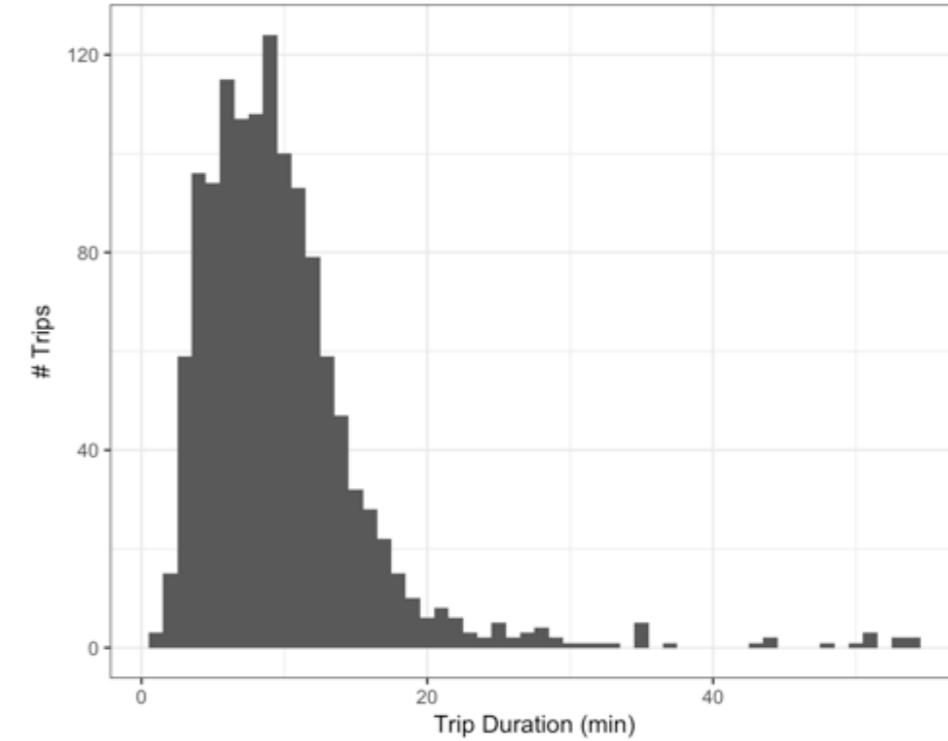
```
```{r}
```

```
trip_durations_graph
```

```
```
```

> Durations of more than 60 minutes are excluded from this graph. `r num\_long\_trips` trips were excluded based on long duration.

## Trip Durations

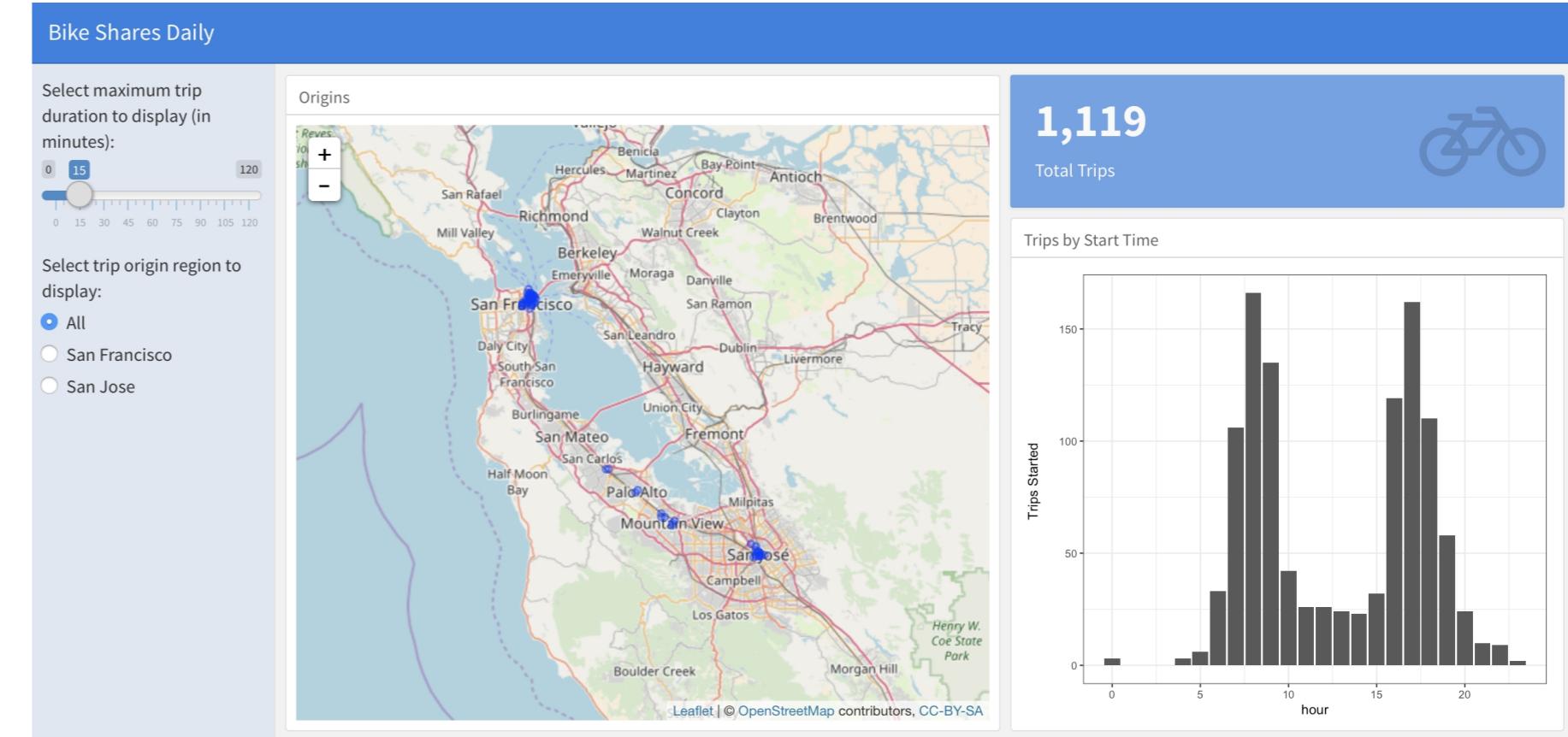


Durations of more than 60 minutes are excluded from this graph. 1269 trips were excluded based on long duration.

# Adding user inputs

Column {data-width=200 .sidebar}

```
```{r}
sliderInput("duration_slider",
  label = "Select maximum trip duration to display (in minutes):",
  min = 0,
  max = 120,
  value = 15,
  step = 5,
  dragRange = TRUE)
```
```



# Making our dataframe reactive

```
show_trips_df <- reactive({

 trips_df %>%
 filter(duration_sec <= input$duration_slider * 60)
})
```

```
Column {data-width=450}

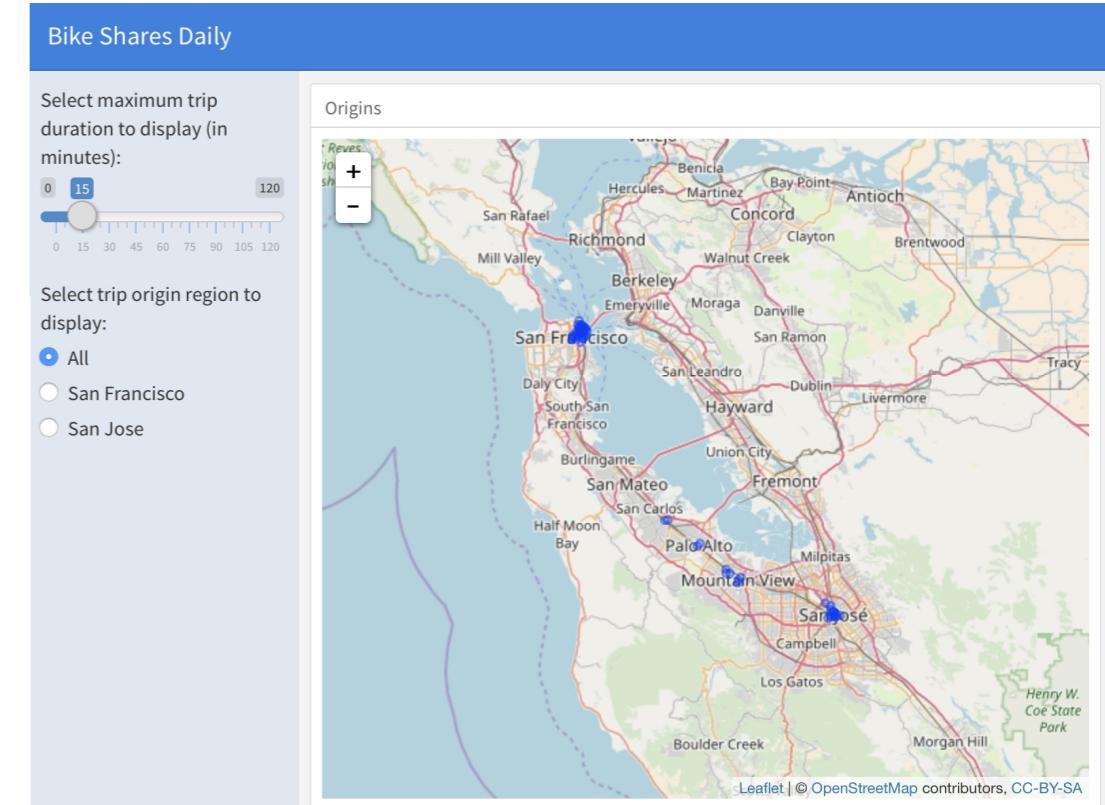
Origins

```{r}  
  
trips_df %>%  
  rename(latitude = start_latitude,  
         longitude = start_longitude) %>%  
  group_by(start_station_id, latitude, longitude) %>%  
  count() %>%  
  leaflet() %>%  
  addTiles() %>%  
  addCircles(radius = ~n)  
  
```
```

```
Column {data-width=450}

Origins

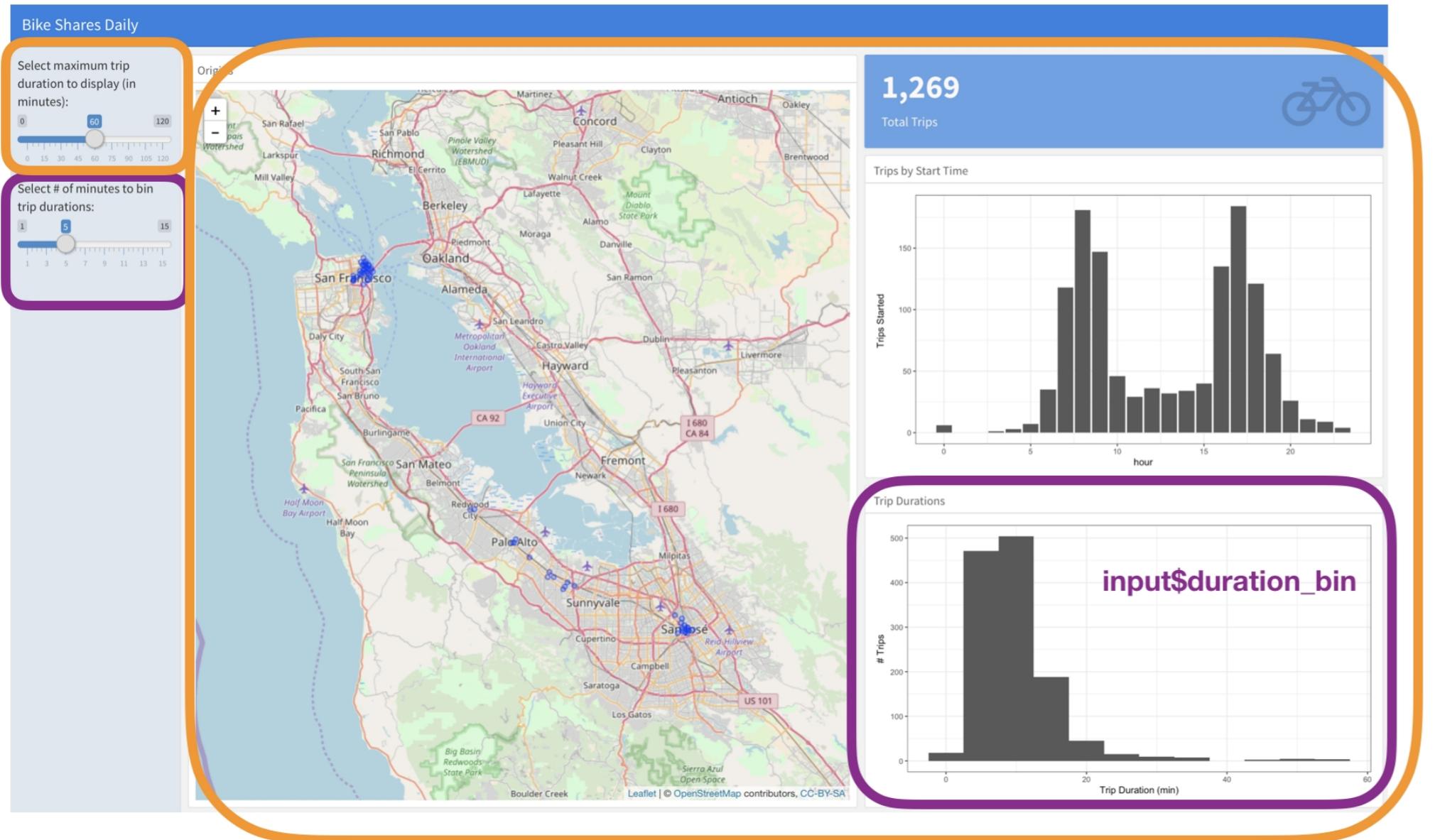
```{r}  
  
renderLeaflet({  
  show_trips_df() %>%  
    rename(latitude = start_latitude,  
           longitude = start_longitude) %>%  
    group_by(start_station_id, latitude, longitude) %>%  
    count() %>%  
    leaflet() %>%  
    addTiles() %>%  
    addCircles(radius = ~n)  
})  
  
```
```



# Steps to the reactive dataframe pattern

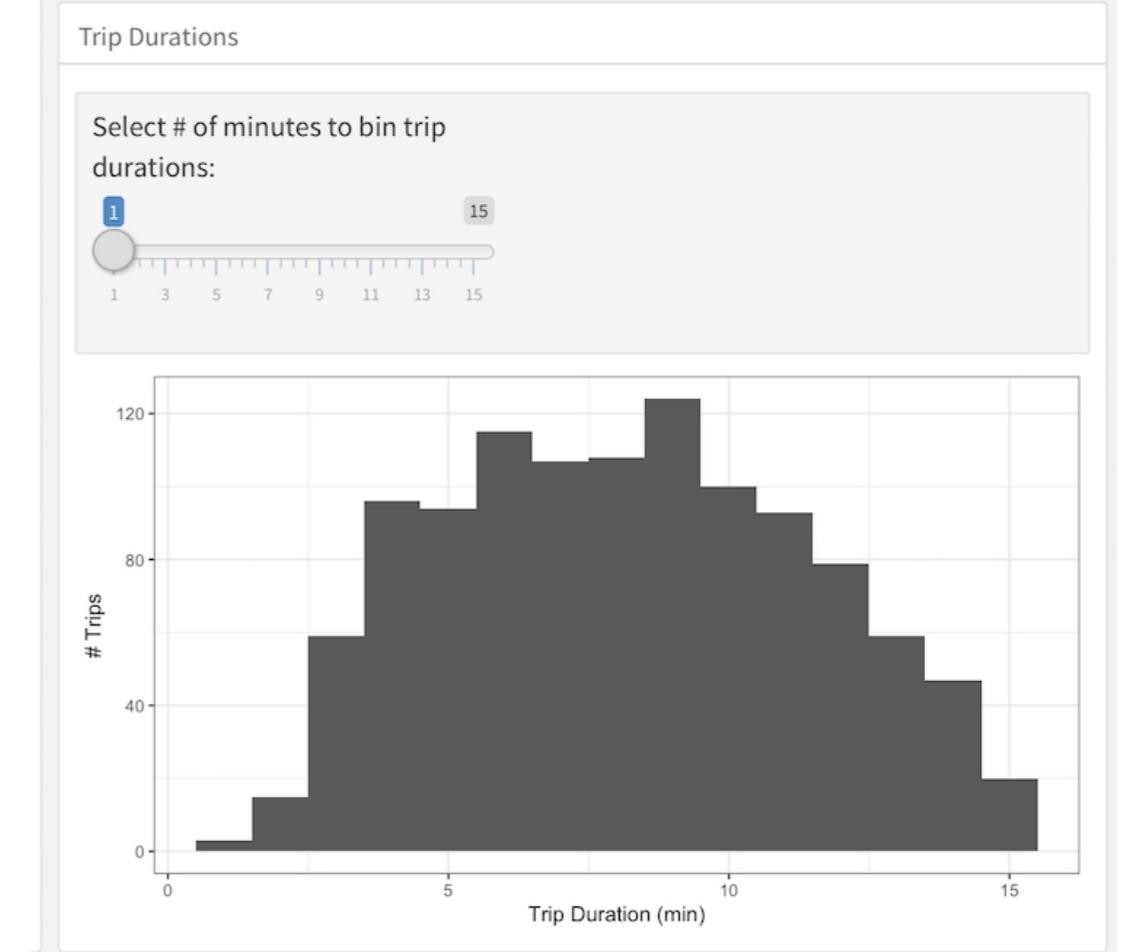
1. Create a sidebar column (using `.sidebar`).
2. Add user inputs to the sidebar (using `xyzInput()` Shiny widgets).
3. Make a "dataframe" that reacts to user inputs (using `reactive()`).
4. Replace the dataframe in the dashboard component code with the reactive version.
5. Wrap each dashboard output with the appropriate Shiny version (`renderXyz()`).

# Chart-specific effects



# Moving inputs into charts

```
```{r}  
→ fillCol(height = 600, flex = c(NA, 1),  
  inputPanel(  
    sliderInput("xyz_input", ...)  
,  
    plotOutput("xyzPlot", height = "100%")  
)  
  
output$xyzPlot <- renderPlot({  
  
})  
```
```



# Setting one sidebar for all pages

```
Global Sidebar { .sidebar }
```

---

```
```{r}
```

```
...
```

Overview

```
Column {data-width=650 .tabset}
```

```
### Origins
```

Resources

- <http://rmarkdown.rstudio.com/exdashboard/>
- <http://www.htmlwidgets.org/>
 - `leaflet`
 - `DT` (datatable)
 - `plotly`
 - `highcharter`