Dashboard LAB

Enrique J. De La Hoz D.

Data Science - UTB

Generating a dashboard

In this exercise, you'll create your first empty flexdashboard! Be sure to expand the html viewer (by clicking the diagonal arrows) after knitting your dashboard to see the whole page.

- Title the dashboard Bikeshare and knit using the title YAML tag.

Adding charts

In this exercise, you'll add content to one of your dashboard charts. Again, be sure to expand the HTML viewer after knitting your dashboard to see the whole page.

- Read the bikeshare.csv dataset
- Be sure to load this packages

```
library(flexdashboard)
library(readr)
library(ggplot2)
library(dplyr)
library(lubridate)
```

- Create a chunk called plot1 and allocate it to the top chart in the second column.

```
trips_df %>%
  mutate(Hour = hour(start_date)) %>%
  group_by(Hour) %>%
  summarize(`Trips Started` = n()) %>%
  ggplot(aes(x = Hour, y = `Trips Started`)) +
  theme_bw() +
  geom_bar(stat = 'identity')
```

- Adjust the widht of the columns and also change the layout of the dashboard to rows. Choose the one you prefer.

Tabsets

In this exercise you'll create a tabset to show two charts in the same layout space.

- Add a second chart to the first column that displays as a second tab.

Pages

In this exercise you'll create a dashboard with two pages.

- Add a chart and make it appear on a second page.
- Title the first page "Overview" and the second page "Details".
- Expand the HTML viewer after knitting and navigate between the two pages.

Menus

- Create 2 pages called 'Details' and 'Data'.
- Change the menu navigation so that the Details and Data pages are listed on a dropdown menu called "More".
- Add the {data-navmenu=More}

Make a plot web-friendly

Let's use ggplotly() on the graph we tried re-sizing.

- Use ggplotly() to make the duration_gg graph in the Trip Durations chart web-friendly. Knit and expand the HTML viewer to see the result. Try resizing the window and observe what happens.

```
duration_gg <- trips_df %>%
  mutate(`Trip Duration (min)` = duration_sec / 60) %>%
  filter(`Trip Duration (min)` <= 60) %>%
  ggplot(aes(x = `Trip Duration (min)`)) +
  theme_bw() +
  geom_histogram(binwidth = 1) +
  ylab('# Trips')

duration_gg
```

Explore web-friendly features

Let's try plotly with another graph and explore the interactive features

- Add the following plot to the Dashboard
- Check the plot
- Use ggplotly() and knit the Dashboard. Check the funcionality.

```
station_df <- trips_df %>%
  select(start_station_name, end_station_name) %>%
  rename(Start = start_station_name, End = end_station_name) %>%
  gather(key = Usage, value = Station)

station_gg <- ggplot(station_df,
                     aes(x = Station, fill = Usage)) +
                     geom_bar(position = 'stack') +
                     theme_bw() +
                     ylab('Trips') +
                     xlab('') +
                     theme(axis.text.x = element_text(angle = 45, hjust = 1))

station_gg
```