# Spatial Data Visualization

Code

## Data Analysis and Visualization

## Instructor: Debopriya Ghosh

Load required packages.

Hide

```
library(dplyr)
```

```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

Hide

```
library(ggplot2)
library(rjson)
library(jsonlite)
```

```
Attaching package: ‘jsonlite’

The following objects are masked from ‘package:rjson’:

    fromJSON, toJSON
```

Hide

```
library(leaflet)
library(RCurl)
```

```
Loading required package: bitops
```

# Interactive Maps with Leaflet

Leaflet package for R allows to overlay your data on top of interactive maps.

Hide

```
rbirthplace <- leaflet() %>%
  addTiles() %>%  # use the default base map which is OpenStreetMap tiles
  addMarkers(lng=174.768, lat=-36.852,
             popup="The birthplace of R")
rbirthplace
```

Let's create your own interactive map using the surface water data.

Data Preprocessing:

1. Request and get the data from the colorado.gov SODA API in R using fromJSON().
2. Address column data types to ensure your quantitative data (number values) data are in fact numeric.
3. Remove NA (missing data) values.

Hide

```
baseurl <- "https://data.colorado.gov/resource/j5pc-4t32.json?"
fullurl <- paste0(baseurl, "station_status=Active",
            "&county=BOULDER")
waterdata <- getURL(URLencode(fullurl))
waterdatadf <- fromJSON(waterdata) %>%
  flatten(recursive = TRUE) # remove the nested data frame
# convert columns to numeric and remove NA values
waterdatadf <- waterdatadf %>%
  mutate_at(vars(amount, location.latitude, location.longitude), funs(as.numeric)) %>%
  filter(!is.na(location.latitude))
```

Once your data are cleaned up, you can create your leaflet map. Here, we are using pipes %>% to set the parameters for the leaflet map.

Hide

```
# create leaflet map
waterlocations <- leaflet(waterdatadf) %>%
  addTiles() %>%
  addCircleMarkers(lng = ~location.longitude,
                   lat = ~location.latitude)
waterlocations
```

# Customize Leaflet Maps

Let's customize the maps:

1. Add custom data-driven popups to your map.
2. Adjust the point symbology.
3. Adjust the basemap.

Hide

```
leaflet(waterdatadf) %>%
  addProviderTiles("CartoDB.Positron") %>%
  addMarkers(lng = ~location.longitude, lat = ~location.latitude,
             popup = ~station_name)
```

# Custom Icons

Hide

```
# Specify custom icon
url <- "http://tinyurl.com/jeybtwj"
water <- makeIcon(url, url, 24, 24)
leaflet(waterdatadf) %>%
  addProviderTiles("Stamen.Terrain") %>%
  addMarkers(lng = ~location.longitude, lat = ~location.latitude, icon=water,
             popup = ~paste0(station_name,
                             "<br/>Discharge: ",
                             amount))
```

# Adding markers

Hide

```
waterdatadf$station_type <- factor(waterdatadf$station_type)
new <- c("red", "green","blue")[waterdatadf$station_type]
icons <- awesomeIcons(
  icon = 'ios-close',
  iconColor = 'black',
  library = 'ion',
  markerColor = new
)
uniquemarkersmap <- leaflet(waterdatadf) %>%
  addProviderTiles("CartoDB.Positron") %>%
  addAwesomeMarkers(lng=~location.longitude, lat=~location.latitude, icon=icons,
                    popup=~station_name,
                    label=~as.character(station_name))
uniquemarkersmap
```

# Adding circle markers

Hide

```
pal <- colorFactor(c("navy", "red", "green"),
                   domain = unique(waterdatadf$station_type))
uniquemarkersmap1 <- leaflet(waterdatadf) %>%
  addProviderTiles("CartoDB.Positron") %>%
  addCircleMarkers(
    color = ~pal(station_type),
    stroke = FALSE, fillOpacity = 0.5,
    lng = ~location.longitude, lat = ~location.latitude,
    label = ~as.character(station_type)
  )
uniquemarkersmap1
```