

# Lab 6

Kiana Fields

Math 241, Week 8

```
# Put all necessary libraries here
library(tidyverse)
library(leaflet)
library(tidycensus)
library(tidyverse)
library(lubridate)
library(ggribbles)
library(dplyr)
library(readr)
```

**Due: Friday, March 22nd at 8:30am**

## Goals of this lab

- Practice creating static and interactive choropleth maps.

### Problem 1: Mapping Bike Rides in Portland

For this problem we will return to the biketown dataset.

- Grab the code from activity 9, Problem 1 to read the data directly from Biketown's API- make sure to keep the longitude and latitude of the start of each ride (`StartLatitude`, `StartLongitude`).

```
bk_jan <- readr::read_csv("https://s3.amazonaws.com/biketown-tripdata-public/2018_01.csv")
bk_jul <- readr::read_csv("https://s3.amazonaws.com/biketown-tripdata-public/2018_07.csv")
bk_nov <- readr::read_csv("https://s3.amazonaws.com/biketown-tripdata-public/2018_11.csv")

bk_jan_cl <- bk_jan %>%
  mutate(StartDate = mdy(StartDate),
        Duration = as.duration(Duration),
        EndDate = mdy(EndDate)) %>%
  mutate(StartTime = ymd_hms(paste(StartDate, StartTime, sep = " ")),
        EndTime = ymd_hms(paste(EndDate, EndTime, sep = " ")))
bk_jul_cl <- bk_jul %>%
  mutate(StartDate = mdy(StartDate),
        Duration = as.duration(Duration),
        EndDate = mdy(EndDate)) %>%
  mutate(StartTime = ymd_hms(paste(StartDate, StartTime, sep = " ")),
        EndTime = ymd_hms(paste(EndDate, EndTime, sep = " ")))
```

```

bk_nov_cl <- bk_nov %>%
  mutate(StartDate = mdy(StartDate),
        Duration = as.duration(Duration),
        EndDate = mdy(EndDate)) %>%
  mutate(StartTime = ymd_hms(paste(StartDate, StartTime, sep = " ")),
        EndTime = ymd_hms(paste(EndDate, EndTime, sep = " ")))

biketown_data <- bind_rows(bk_jan_cl, bk_jul_cl, bk_nov_cl) %>%
  select(StartDate, StartTime, EndDate, EndTime, Distance_Miles,
         BikeID, StartLatitude, StartLongitude, EndLatitude, EndLongitude)

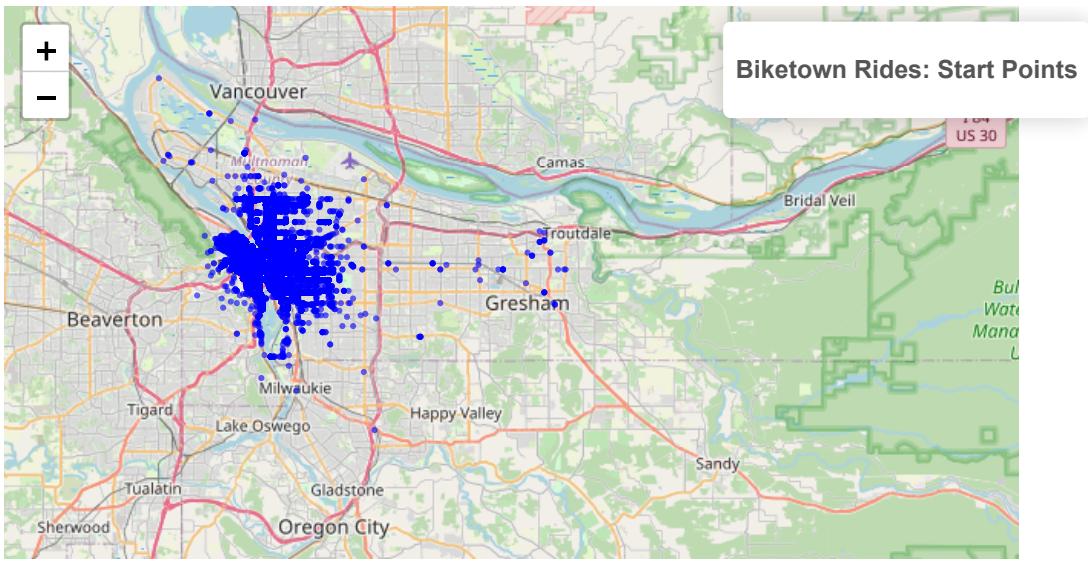
```

- b. Create an interactive map of the start point of the rides using the `leaflet` package. Make sure to include a legend and a title. What do you notice about the distribution of rides?

```

leaflet(biketown_data) %>%
  addTiles() %>%
  addCircleMarkers(
    lng = ~StartLongitude,
    lat = ~StartLatitude,
    popup = ~paste("Start Date:", StartDate, "<br>",
                  "Start Time:", StartTime, "<br>",
                  "Distance (Miles):", Distance_Miles),
    radius = 2,
    color = "blue",
    stroke = FALSE,
    fillOpacity = 0.6
  ) %>%
  addControl(
    html = "<h3 style='text-align:center;'>Biketown Rides: Start Points</h3>",
    position = "topright"
  ) %>%
  addLegend(
    "bottomright",
    title = "Legend",
    colors = "blue",
    labels = "Start Point"
  )

```



Legend  
Start Point

Leaflet | © OpenStreetMap, ODbL

Most of the rides start around downtown (Old Town, Pearl District) as well as the neighborhoods right across the Willamette, from East Moreland up to Boise. I'd be curious to consider the range of the Biketown data set, and where usage is popular.

c. Using the `lubridate` package, create a variable, `month`, indicating the month of each variable.

```
biketown_data <- biketown_data %>%
  mutate(month = month.name[month(StartDate)])
```

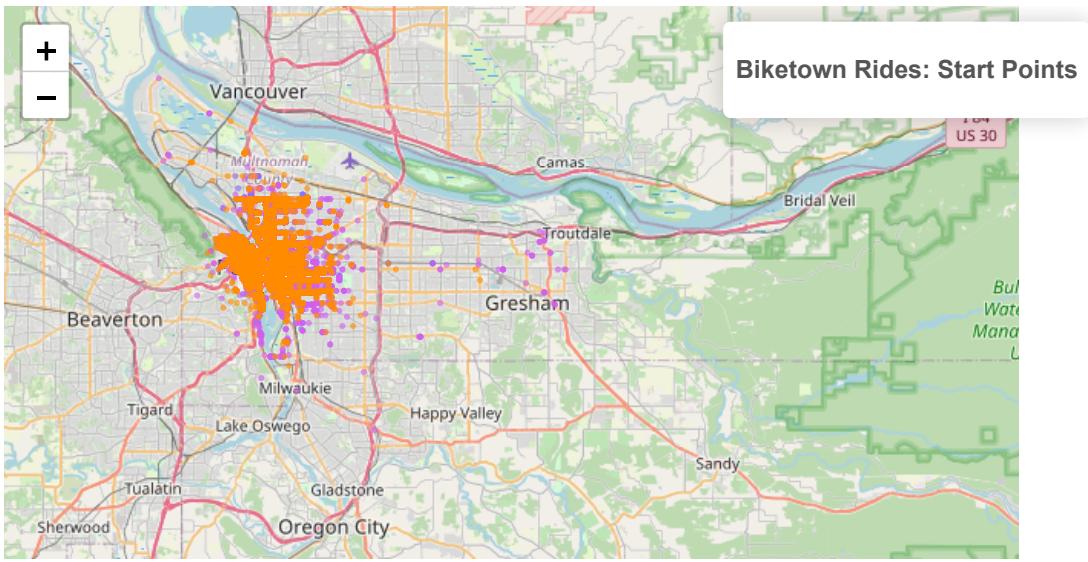
Add this variable to your interactive map using color. Make sure to include a legend and be mindful of your color palette choice. Do ride locations vary by months of the year?

```
library(leaflet)

month_colors <- c("January" = "slateblue1", "July" = "mediumorchid2", "November" = "darkorange")

month_palette <- colorFactor(palette = month_colors,
                               domain = biketown_data$month)

leaflet(biketown_data) %>%
  addTiles() %>%
  addCircleMarkers(
    lng = ~StartLongitude,
    lat = ~StartLatitude,
    popup = ~paste("Start Date:", StartDate, "<br>",
                  "Start Time:", StartTime, "<br>",
                  "Distance (Miles):", Distance_Miles),
    radius = 2,
    color = ~month_palette(month),
    stroke = FALSE,
    fillOpacity = 0.6
  ) %>%
  addControl(
    html = "<h3 style='text-align:center;'>Biketown Rides: Start Points</h3>",
    position = "topright"
  ) %>%
  addLegend(
    "bottomright",
    title = "Month",
    pal = month_palette,
    values = ~month
  )
```



Month
January
July
November

Leaflet | © OpenStreetMap, ODbL

Across all locations, most of the start dates are in November or July, not many in January.

## Problem 2: Choropleth Maps

For this problem, I want you to practice creating choropleth maps. Let's grab some data using `tidycensus`. Remember that you will have to set up an API key.

- a. Let's grab data on the median gross rent (`B25064_001`) from the American Community Survey for Multnomah county, Oregon. I want you to do data pulls at three geography resolutions: county subdivision, tract, and block group.

```
library(sf)
library(tigris)

ORcounty <- get_acs(geography = "county subdivision",
                     year = 2021,
                     variables = c(medGrossRent = "B25064_001"),
                     state = "OR",
                     county = "Multnomah",
                     survey = "acs5",
                     output = "wide")
ORtract <- get_acs(geography = "tract",
                     year = 2021,
                     variables = c(medGrossRent = "B25064_001"),
                     state = "OR",
                     county = "Multnomah",
                     survey = "acs5",
                     output = "wide")
ORblock <- get_acs(geography = "block group",
                     year = 2021,
                     variables = c(medGrossRent = "B25064_001"),
                     state = "OR",
                     county = "Multnomah",
                     survey = "acs5",
                     output = "wide")

sp_counties <- county_subdivisions(state = "OR", cb = TRUE)
```

## Downloading: 16 kB      Downloading: 16 kB      Downloading: 16 kB      Downloading: 16 kB      Download

```
sp_tracts <- tracts(state = "OR", cb = TRUE)
```

## |

```
sp_blocks <- block_groups(state = "OR", cb = TRUE)
```

## Downloading: 100 kB      Downloading: 100 kB      Downloading: 100 kB      Downloading: 100 kB      Download

```
ORcounty <- left_join(ORcounty, sp_counties, by = c("GEOID" = "GEOID"))
ORtract <- left_join(ORtract, sp_tracts, by = c("GEOID" = "GEOID"))
ORblock <- left_join(ORblock, sp_blocks, by = c("GEOID" = "GEOID"))
```

- b. Create three choropleth maps of gross rent, one for each geography resolution. What information can we glean from these maps? Also, which resolution seems most useful for this variable? Justify your answer.

```
library(RColorBrewer)

ORcounty %>%
  st_as_sf() %>%
  ggplot() +
  geom_sf(aes(fill = medGrossRentM)) +
  scale_fill_distiller(palette = "PiYG", direction = -1) +
  labs(fill = "Median Gross Rent") +
  ggtitle("Median Gross Rent in Multnomah County by Subdivision") +
  theme_void()
```

Median Gross Rent in Multnomah County by Subdivision



```
ORtract %>%
  st_as_sf() %>%
  ggplot() +
  geom_sf(aes(fill = medGrossRentM)) +
  scale_fill_distiller(palette = "PiYG", direction = -1) +
  labs(fill = "Median Gross Rent") +
  ggtitle("Median Gross Rent in Multnomah County by Census Tract") +
  theme_void()
```

## Median Gross Rent in Multnomah County by Census Tract



```
ORblock %>%
  st_as_sf() %>%
  ggplot() +
  geom_sf(aes(fill = medGrossRentM)) +
  scale_fill_distiller(palette = "PiYG", direction = -1) +
  labs(fill = "Median Gross Rent") +
  ggtitle("Median Gross Rent in Multnomah County by Block Group") +
  theme_void()
```

## Median Gross Rent in Multnomah County by Block Group



From these maps we can see there's a wide range in rent prices across the county. Most areas seemed to be around 500 with some notable exceptions, like a block group with 2000 as the gross rent. Comparing across these different resolutions it seems like gross rent is generally around 300 across most areas. While there appears to be some missing data, the block group resolution seems most useful for our Rent variable. It clearly shows the wide range in rent in a way the other resolutions don't, possibly because it breaks the county down into smaller groupings than the other resolutions. For example, by subdivision there's only 5 sections and the greatest median rent is 500. This same geographic area in the block group visualization still has a median rent of 500 but we can see there are other block groups with considerably higher median rents.