HW1: Manipulating and visualizing spatio-temporal data

Fern Bromley

DUE: 9/12 11:59pm

Homework Guidelines

Please submit your answers on Gradescope as a PDF with pages matched to question answers.

One way to prepare your solutions to this homework is with R Markdown, which provides a way to include mathematical notation, text, code, and figures in a single document. A template `.Rmd` file is available through D2L.

Make sure all solutions are clearly labeled, and **please utilize the question pairing tool on Gradescope**. You are encouraged to work together, but your solutions, code, plots, and wording should always be your own. Come and see me during office hours or schedule an appointment when you get stuck and can't get unstuck.

I. Time [10 pts]

Let's continue to look at the graffiti dataset from the city of San Diego. We'll focus on the records from 2023 onward.

```
1 data_url <- url(paste0("https://seshat.datasd.org/get_it_done_graffiti/",
2                        "get_it_done_graffiti_requests_datasd.csv"))
3 graffiti <- read.csv(data_url)
4 graffiti$POSIX_requested <-
5   strptime(graffiti$date_requested, format = "________", tz = "America/Los_Angeles")
6 start_date <- as.POSIXlt("2023-01-01 00:00:00", tz = "America/Los_Angeles")
7 graffiti <- graffiti[graffiti$POSIX_requested >= start_date, ]
```

(1) [3 pts] What should be filled in for the `format` argument in line 5 to convert `graffiti$date_requested` to the `POSIXlt` vector `graffiti$POSIX_requested`? In your own words, explain what the code in lines 6 and 7 does.

`POSIXlt` objects have a special method for the function `hist()` in `R`. The following line of code makes a histogram with bins defined by each month of the year.

```
hist(graffiti$POSIX_requested, breaks = "month", xlab = "time", main = "graffiti reports")
abline(v = seq(as.POSIXlt("2023-01-01"), as.POSIXlt("2026-01-01"), "year"),
       col = "darkred", lwd = 2)
```

(2) [3 pts] Based on the histogram, do you think there appears to be any seasonality to the graffiti reports? If so, which months/seasons seem to have the highest rates of reporting? What do you see in the histogram that informs your answer?

(3) [4 pts] Run `?hist.POSIXt` in `R` to look at the help documentation for the `POSIX*t` method of `hist()`. What options are available for the `breaks` argument? **Provide executable code** (**PEC**) that makes a histogram of graffiti reports in time binned by week. Include your figure.

II. Simple features

A. Projections [10 pts]

For many situations, working with spatial data in longitude and latitude works just fine (especially when the locations are near the equator). However, when we need to calculate distances between points, or areas of polygons, it is typically necessary to first *project* the spatial locations onto a flat coordinate system.

(4) [2 pts] Take a look at these examples of map projections: https://xkcd.com/977/. Which one(s) is(are) your favorite(s)? Why?

Let's practice projecting data using the census tracts we looked at in class. We'll start by downloading those census tracts using the `tigris` package.

```r
library(tigris)
sd_tracts <- tracts(state = "CA", county = "San Diego")
```

We can check to see what the current coordinate system for the `SpatialPolygonDataFrame` is with the `st_crs()` function in `sf`.

```r
library(sf)
st_crs(sd_tracts)
```

```
## Coordinate Reference System:
##   User input: NAD83
##   wkt:
## GEOGCRS["NAD83",
##     DATUM["North American Datum 1983",
##         ELLIPSOID["GRS 1980",6378137,298.257222101,
##             LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["latitude",north,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["longitude",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     ID["EPSG",4269]]
```

(5) [2 pts] Use the resources at the homepage for the `sf` package (https://r-spatial.github.io/sf/#cheatsheet) to determine which function can be used to compute the areas of the census tracts in `sd_tracts`. **PEC** that creates a variable called `areas` with those values and verify the range of values match mine.

```r
range(areas)
```

The areas we just calculated use a default method for spatially-referenced data in lat/long coordinates. Now we'll try re-projecting the census tracts for San Diego to a new coordinate reference system and see how the calculated areas change.

(6) [3 pts] Use the `st_transform()` function to transform the census tract polygons to the **Universal Transverse Mercator (UTM) projection for zone 11** (*hint: one way to specify the CRS for UTM zone 11 is* `+proj=utm +zone=11 +datum=WGS84 +units=m +no_defs +type=crs` *but there are others*). Call the new transformed object `sd_tracts_utm`. Compute the areas for each tract in the new coordinate system. How similar/different are the values to your answers from (5)?

(7) [3 pts] One of the columns of data provided by the City of San Diego in `sd_tracts` is labeled `ALAND` for area of land. Compare the values of `ALAND` to the ones we just calculated ourselves. How well do they agree? What do you think might be a reason for discrepancies between the areas?

B. Art density [15 pts]

Now let's look at a new dataset from the City of San Diego. Line 16 downloads and reads into `R` the locations of artwork in the Civic Art Collection.

```
1 art <- read.csv(url("https://seshat.datasd.org/civic_art_collection/public_art_locations_datasd.csv"))
2 art_sf <- st_as_sf(_____, coords = _________)
3 st_crs(art_sf) <- st_crs(sd_tracts)
```

(8) [2 pt] Fill in the blanks on line 17 to create the simple features object `art_sf` with an appropriate geometry defined by the locations in `art`. In your own words, explain what the code in line 18 does.

(9) [2 pt] How many total art installations are there in the collection? How many unique locations are there?

Suppose we're working for the Chief Operating Officer of the City of San Diego, and we're trying to decide where we should spend money on a new mural. There are two candidate locations: College-Roland Library (longitude -117.0561, latitude 32.76941), and Mission Valley Library (-117.1269, 32.7793). One way to help us decide where to commission a new mural might be to look at how much access our residents have to art in their neighborhoods. If there is a scarcity of art in the vicinity of one of these libraries, that might be a good place to consider.

First, we need to figure out how many pieces of art exist near these libraries. One way to summarize that information would be to figure out which census tract each library falls in, and then add up the number of art pieces in that census tract. We can use the function `st_intersection()` to do this.

The code in lines 19–20 creates a simple features object called `CRL` that represents the location of the College-Rolando Library (CRL).

```
1 CRL <- st_as_sf(x = data.frame(lng = -117.0561, lat = 32.76941), coords = c("lng", "lat"))
2 st_crs(CRL) <- st_crs(sd_tracts) ## match the coordinate reference systems
```

(10) [1 pt] **PEC** that creates a simple features object called `MVL` that represents the location of the Mission Valley Library (MVL).

The code in line 21 finds the row in `sd_tracts` that corresponds to the census tract containing the CRL. From the output, we can see that the name of the tract is `29.05`.

```
1 st_intersection(CRL, sd_tracts)
```

```
## Simple feature collection with 1 feature and 13 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -117.0561 ymin: 32.76941 xmax: -117.0561 ymax: 32.76941
## Geodetic CRS:  NAD83
```

```
##   STATEFP COUNTYFP TRACTCE       GEOID              GEOIDFQ  NAME
## 1      06      073  002905 06073002905 1400000US06073002905 29.05
##              NAMELSAD MTFCC FUNCSTAT   ALAND AWATER    INTPTLAT      INTPTLON
## 1 Census Tract 29.05 G5020        S 1146262      0 +32.7711434 -117.0496451
##                       geometry
## 1 POINT (-117.0561 32.76941)
```

(11) [1 pt] What is the name of the census tract containing the MVL?

(12) [4 pts] Fill in the blanks below to create a new variable in the `sd_tracts` dataframe called `n_art` that gives the number of art installations in each census tract. Verify you get 12 installations in the tract with the College-Rolando Library. How many pieces of artwork are in the tract containing the MVL?

```
1 sd_tracts$n_art <- lengths(st______(_____, _____))
2 sd_tracts$n_art[sd_tracts$NAME == "29.05"]
```

(13) [5 pts] Of CLR and MVL, which census tract has a lower density of artwork (i.e., less art per square meter)? Based on your answer to the previous question, which site do you think should be selected for the mural? What's another, *new* piece of information you think would be relevant to this decision?

III. ggmap [5 pts]

The package ggmap is an `R` package the facilitates the use of freely available map tiles from sources like Google Maps, and Stadia Maps. These can provide great contextual visualizations for spatio-temporal data. Both of these maptile services require registration, but they both have free subscription levels that provide all the functionality we'll need. *Stadia does not require you to provide payment information.* Once you have registered, input your key using `ggmap::register_stadiamaps()`. Specifying `write = TRUE` will save your key for future instances of R. If you encounter problems with the CRAN distribution of ggmap, give the development version a try using `devtools::install_github("dkahle/ggmap")`.

The following exercises are designed to help you learn the basic features of ggmap. The code in lines 24–25 loads the ggmap package, then defines the variable `bbox` based on a range of longitudes and latitudes.

```
1 library(ggmap)
2 bbox <- c(left = -117.5, bottom = 32, right = -116.5, top = 33.5)
```

The code in line 26 uses the function `get_stadiamap()` to download tiles from Stadia.

```
1 art_map <- get_stadiamap(bbox = bbox)
```

The code in lines 27–28 plots the artwork locations on a background of map tiles obtained from Stadia. The ggmap package uses the grammar of graphics syntax used in `ggplot2`. Don't worry too much if you're not familiar with it, you'll only need the basics for our class, and there are TONS of tutorials out there if you'd like to learn more. Try out the code below and take a look at the resulting map.

```
1 ggmap(art_map) +
2   geom_point(aes(x = lng, y = lat), data = data.frame(art))
```

(14) [2 pts] The map shown is too zoomed out to be useful for this dataset. Try adjusting the `bbox` argument in the `get_stadiamap()` function until you have a better spatial scale. What are some good values for `bbox`?

Stadia provides some alternative map tiles that can be very visually appealing, if not always strictly necessary. Use `?get_stadiamap` to see some possible values for the arguments `maptype` and `color`.

(15) [3 pts] Explore some other combinations of `maptype` and `color`. Include your maps. Which one(s) is(are) your favorite(s)?

There are LOTS of great tutorials out there on the `ggmap` package. If you find one that resonates with you, let me know so I can post it on D2L for others.