Sewershe_Attachments2

2025-03-18

```
setwd("C:/Users/MaxBo/OneDrive/Documents/Class MQE/Capstone/Data Files/Filtered Park Data Finalized")
library(sf)
```

```
## Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
parks <- st_read("filtered_parks_fix2.geojson")
```

```
## Reading layer 'filtered_parks_fix2' from data source
##   'C:\Users\MaxBo\OneDrive\Documents\Class MQE\Capstone\Data Files\Filtered Park Data Finalized\filt
##   using driver 'GeoJSON'
## Simple feature collection with 168 features and 24 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -80.08152 ymin: 40.36871 xmax: -79.86873 ymax: 40.49287
## Geodetic CRS:  WGS 84
```

```
walksheds <- st_read("Walkshed differences.geojson")
```

```
## Reading layer 'Walkshed differences' from data source
##   'C:\Users\MaxBo\OneDrive\Documents\Class MQE\Capstone\Data Files\Filtered Park Data Finalized\Walks
##   using driver 'GeoJSON'
## Simple feature collection with 168 features and 67 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 1318214 ymin: 383116.4 xmax: 1379567 ymax: 431866.1
## Projected CRS: NAD83 / Pennsylvania South (ftUS)
```

```
sewer <- st_read("Sewershed.geojson")
```

```
## Reading layer 'Sewershed' from data source
##   'C:\Users\MaxBo\OneDrive\Documents\Class MQE\Capstone\Data Files\Filtered Park Data Finalized\Sewe
##   using driver 'GeoJSON'
## Simple feature collection with 182 features and 8 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -80.0918 ymin: 40.37011 xmax: -79.87144 ymax: 40.50073
## Geodetic CRS:  WGS 84
```

```
#Convert nulls to zero priority

sewer$PRIORITYSS[is.na(sewer$PRIORITYSS)] <- 0
```

```r
#Convert priority 1 and 2 to numeric, and deal with problematic obs.

sewer$PRIORITYSS[sewer$PRIORITYSS == "Y - Ph1"] <- 1
sewer$PRIORITYSS[sewer$PRIORITYSS == "Y - Ph2"] <- 2

sewer$PRIORITYSS[sewer$PRIORITYSS == "<Null>"] <- 0
```

```r
#Check class
class(sewer$PRIORITYSS)
```

```
## [1] "character"
```

```r
#Convert characters to numeric
sewer$PRIORITYSS <- as.numeric(sewer$PRIORITYSS)
```

```r
#Disable s2 processing
sf_use_s2(FALSE)
```

```
## Spherical geometry (s2) switched off
```

```r
#Create geopatial overlaps between sewershed and park polygons
overlaps <- st_intersection(parks, sewer)
```

```
## although coordinates are longitude/latitude, st_intersection assumes that they
## are planar

## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries
```

```r
#Some parks do not exist within PWSA Sewershed area
length(unique(overlaps$objectid))
```

```
## [1] 147
```

```r
#Compute area of overlaps
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
```

```r
overlaps <- overlaps %>%
  mutate(overlap_area = st_area(.))
```

```r
#Covnert to Acreage
overlaps$overlaps_acre <- overlaps$overlap_area / 4046.86
```

```r
#Areal Interpolation: Assigns weight to sewershed score based on the amount
#of area the sewershed polygon takes up in the park polygon

park_scores <- overlaps %>%
  mutate(weighted_sewer = (overlaps_acre / acreage) * PRIORITYSS) %>%
  group_by(objectid) %>%  # Replace with actual unique park ID column
  summarise(avg_sewershed_score = sum(weighted_sewer, na.rm = TRUE))
```

```
## although coordinates are longitude/latitude, st_union assumes that they are
## planar
```

```r
#Covnert to Numeric
park_scores$avg_sewershed_score <- as.numeric(park_scores$avg_sewershed_score)
```

```r
# Perform the spatial join using st_join()
parks2 <- st_join(parks, park_scores, by = "objectid")
```

```
## although coordinates are longitude/latitude, st_intersects assumes that they
## are planar
```

```r
#We see 170 observations in the new dataset while we should have 68.
#after some research, we see an error occur with Highland Park
#and Joe Natoli Playground. Both parks are connected to eachother.
#It assigned both scores to both of them.

#After looking at the intersections in QGIS, we can clearly see
#that Joe Natoli Playground has the score of 1 while Highland
#Park has the score of .57. So we will perform a direct removal in python.


nrow(parks2)
```

```
## [1] 170
```

```r
#Replace NULL sewer scores with zero. PWSA doesn't have geodata on them.

parks2$avg_sewershed_score[is.na(parks2$avg_sewershed_score)] <- 0
```

```r
#Check class
class(parks2)
```

```
## [1] "sf"         "data.frame"
```

```r
#De-convert from SF and save as CSV
parks3 <- st_drop_geometry(parks2)
write.csv(parks3, "parks_sewer_scores.csv", row.names = FALSE)
```