Systems Design for Wastewater Surveillance: Information Guide and Training Resources

Dustin T. Hill, Shaz Damani, and David A. Larsen

2025-04-22

Contents

Chapter 1

Introduction

Welcome to this comprehensive guide designed to enhance and strengthen your wastewater surveillance activities. This document serves as a valuable resource to complement the monthly workshops hosted by the New York Center of Excellence (NY CoE) during our regional calls. Within these pages, you will find essential information, resources, and practical tips to support your efforts in wastewater monitoring.

We encourage you to refer to this guide as you embark on or continue your surveillance initiatives. Should you encounter any difficulties accessing the provided links, require additional one-on-one assistance, or have general inquiries, please do not hesitate to reach out to us. We are here to support you in successfully implementing and expanding your wastewater surveillance strategies.

This guide will focus on Systems Design for wastewater surveillance and cover the following topics:

Part 1: How New York State mapped their sewersheds

In Part 1, you will learn about how NYS mapped their sewersheds. The methods can be applied to other jurisdictions looking to produce maps of sewershed boundaries to aid wastewater surveillance for epidemiology.

Part 2: Maps and spatial data for communities connected to public sewer

In Part 2, we will review existing maps for community sewer systems and how to make them for areas that might be missing them. While there is a plethora of digital spatial data on many topics, there is not yet a comprehensive national database for sewers in the United States (US). Using primarily public records, we show how to take an initial inventory of municipal WWTPs as a starting point for mapping a jurisdiction's sewer systems.

Part 3: Adding Census Data to Wastewater-Based Epidemiology

Part 3, we explore how to incorporate United States Census (US Census) data, which can add population demographic variables, to WBE data. I will discuss how to integrate spatial census data to spatial WBE data.

Part 4: Adding other spatial data to WBE

In Part 4, we explore how to add other spatial data to WBE projects including zip codes and point data.

Part 5: System design for outbreak detection

In this last part of our Systems Design review, we share ways to understand an outbreak using wastewater surveillance data. Examples for some pathogens are provided as case studies.

This info guide will also include a comprehensive list of videos, links to data, and other resources in the Resources and Links section.

1.1 Contact information

New York State Center of Excellence Website: https://nywastewatcher.io/nwsscoe.

General concerns email: **COVID.WasteWater@health.ny.gov**.

Chapter 2

How New York State mapped their sewersheds

2.1 Background

In 2020, as part of the emergency response to the COVID-19 pandemic, a pilot wastewater surveillance project was initiated by NYS Department of Health (NYS DOH) and Syracuse University. As part of this pilot, maps were collected for participating wastewater treatment plants (WWTPs). These maps were the initial database of NYS sewersheds and the process used to create them formed the template for mapping all of the state's sewersheds for each wastewater treatment plant (2.1).

In 2021, as part of the scale-up of wastewater surveillance in New York, NYS DOH funded a project to map all of the sewersheds in the state. Using contact information for the state's WWTP operators, a survey was conducted to assess what data already existed for sewershed maps. Geospatial information systems (GIS) digital data were collected from sites that had mapped their systems. In addition, copies of physical maps, photos of maps, address lists for properties connected to sewers, tax rolls, and state digitized parcel records were collected to aid in the mapping effort. All of New York's Sewersheds were mapped regardless of whether the site was selected to participate in the ongoing surveillance projects (2.2)

In this section, you will learn about how NYS mapped their sewersheds and the open-source methods that were used. This overview will be referenced in future sections and the methods can be applied to other jurisdictions looking to produce maps of sewershed boundaries to aid wastewater surveillance for epidemiology.

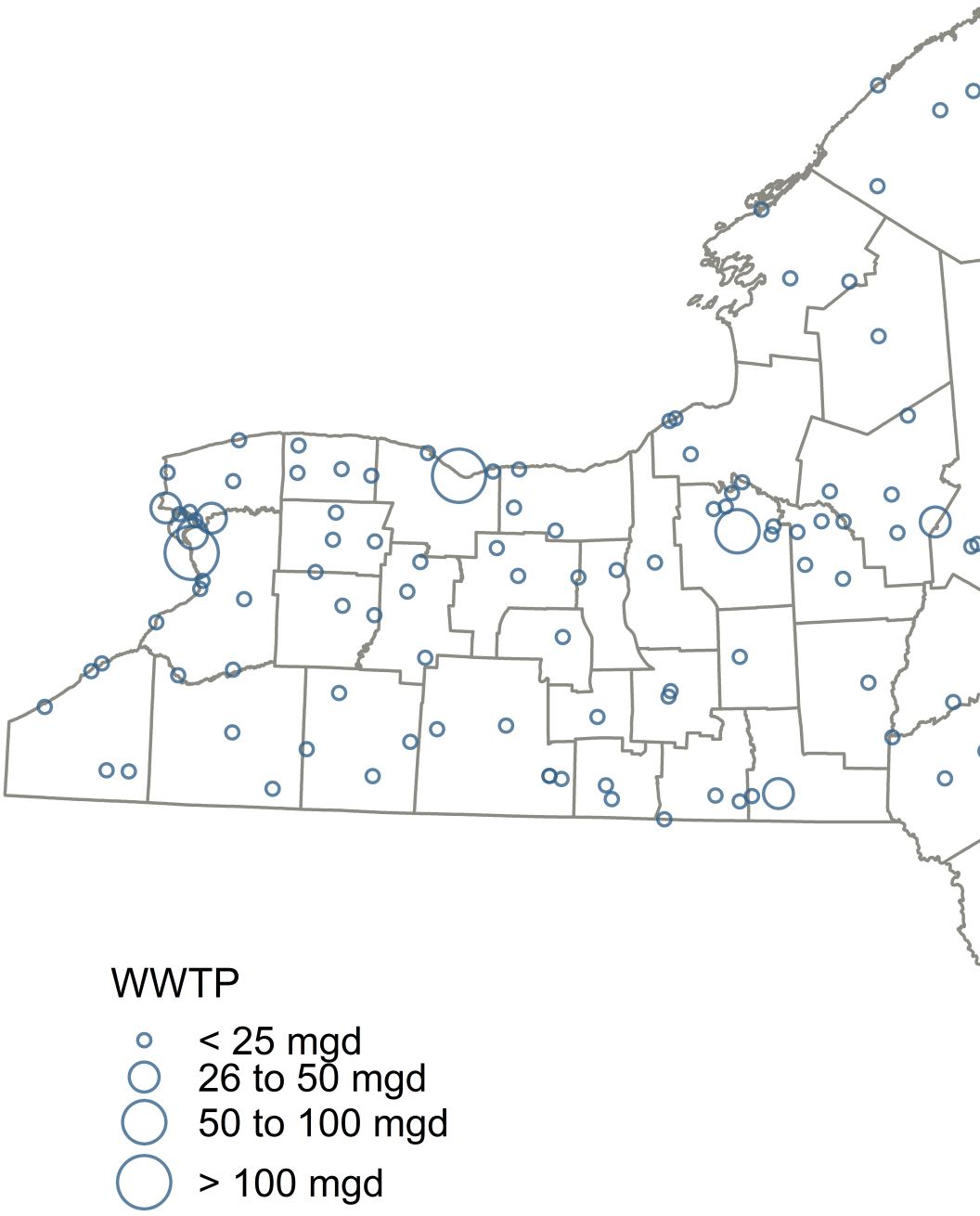


Figure 2.1: Sampling Sites in New York State (2024)

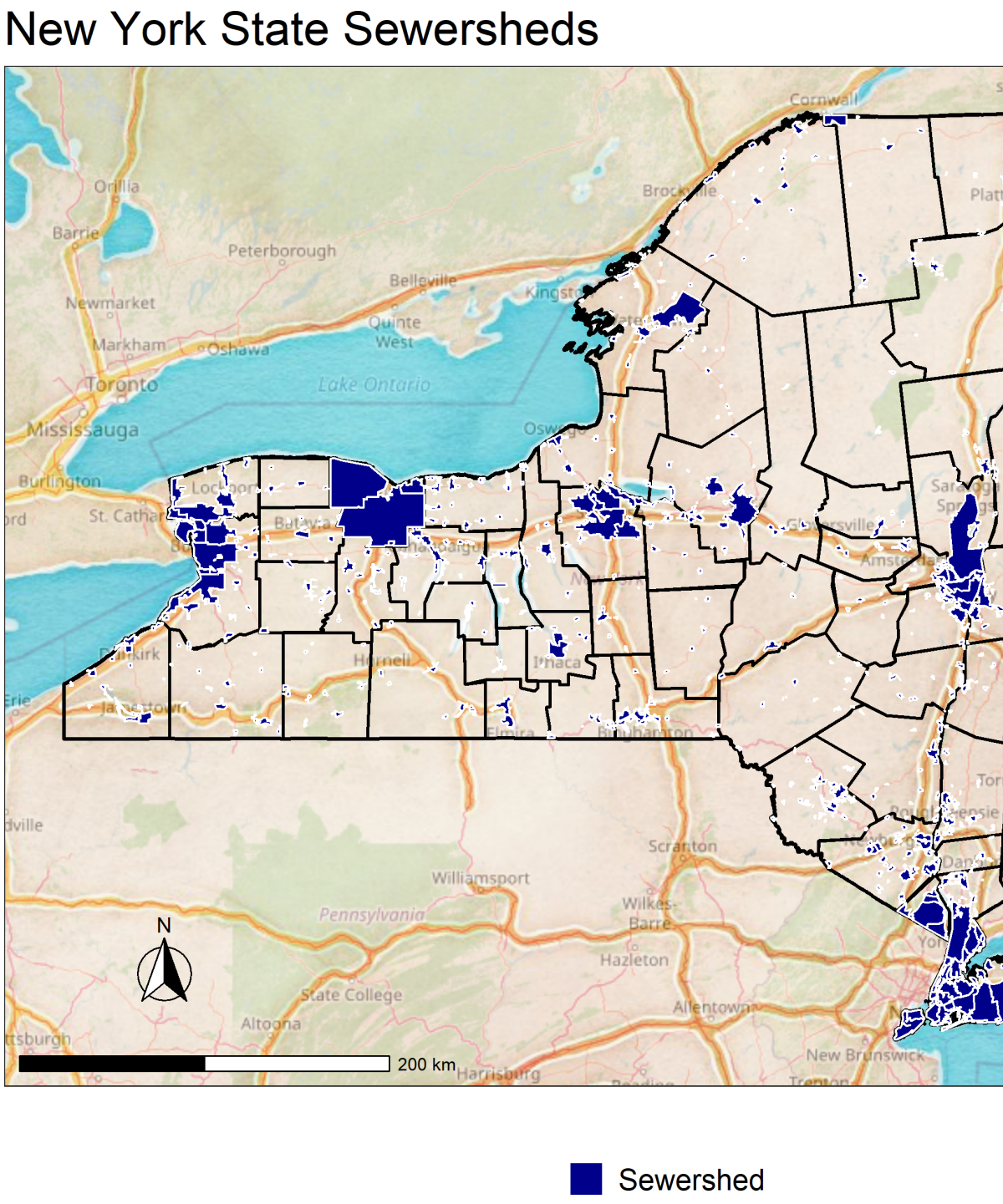


Figure 2.2: New York State Municipal Sewersheds

2.2 Goals for this section

1. Learn about the process that NYS used to map sewersheds.
2. Identify methods used to address challenges in mapping and producing "best" estimates for sewershed boundaries.
3. Learn how NYS is using sewershed boundaries to enhance the utility of wastewater epidemiology.

2.3 Resources

- Video powerpoint for how NY created sewersheds.
- Hill, D, Larsen, DA. 2023. Using geographic information systems to link population estimates to wastewater surveillance data in New York State, USA. PLOS Global Public Health. The paper explains the process for mapping all of NYS's sewersheds including data collection, processing, and estimating population estimates for each sewershed.
- NYS sewershed data (ArcGIS online download). Access and download NYS sewershed data to see how data are recorded.
- NYS sewershed data (GitHub data download). Access and download NYS sewershed data directly into your R session.
- NYS sewershed data viewer. Visually inspect NYS sewersheds and the population served by each. The page includes example code for downloading and adding population data to sewershed shapefiles in the program R.
- Using the Editor toolbar in ArcGIS. ArcGIS can be an excellent way to digitally edit spatial data.
- R spatial tools. R is an open-source program, and most mapping projects and analyses can be completed using the software.

Chapter 3

Maps and spatial data for communities connected to public sewer

3.1 Overview

In this section, we will review existing maps for community sewer and how to go about making them. While there is a plethora of digital spatial data on many topics, there is not yet a comprehensive national database for sewers in the United States (US). Prior to the NYS DOH sewershed mapping project, there was also not a comprehensive map of NY's sewer systems. There are, however, some public databases for locations of publicly owned treatment works including permitted wastewater treatment plants. The US Environmental Protection Agency (EPA) maintains records on permitted WWTPs and so do most state environmental offices. Using these public records, we know where the municipal WWTPs are and this provides a starting point for mapping a state's sewer systems. The next step involves collecting data from other sources like:

- State tax parcel records
- Municipal records for local sewer taxes
- WWTP infrastructure maps or engineering design
- County or local planning documents
- Location of manholes or sewer mains
- Lists of addresses for properties that report paying a sewer tax

Many of these records will be housed in local GIS, tax, planning, or municipal offices. Some will be public, and others might require permission to access or

data use agreements. The final product of mapping a state's sewer systems does not result in any identifiable data being produced, and if you explain to local officials the goals of the project and what the final product will look like, data sharing should not be an issue. Work with your local partners to ensure that their questions are answered, and any concerns are addressed.

3.2 Goals for this section

In this section, you will:

1. Review an example sewershed parcel mapping exercise for one county in New York State
2. Explore loading and manipulating maps in ArcGIS and R
3. Gather data on your state's WWTP locations
4. Contact local offices, WWTP operators, and others to gather data on sewer locations
5. Gather parcel property data for your jurisdiction

3.3 Resources

- National parcel data viewer / data access portal. This is a data portal that lets you explore each state in the US and whether they have tax parcel data. Some links bring you to county or state websites and from there you can view and download the data.

ArcGIS tutorials

- Loading data into Arc
- Add point data from a table
- Using the Editor tool
- Save your shapefile

Q GIS tutorials

- Downloading Q
- Add polygon data to Q
- Add point data from a table
- Edit data in Q
- Split multipart polygons
- Save your data in Q

3.4 Training exercise

This training will show how to use publicly available tax parcel data to draw municipal sewershed boundaries. You will need to download R to run this script on your machine.

For this example, we are going to map sewersheds in Onondaga County, NY. Sewersheds are the area of land that a wastewater treatment plant (WWTP) serves and include public and private properties. Many states, including NY, keep track of residential and commercial sewer access in their state tax records. We will use data downloaded from the NYS Tax Parcel system for Onondaga county.

You can access the file for Onondaga County at the following location: NYS Tax Parcel Data. These data are very large, so for the purposes of this tutorial, I have already downloaded the data and filtered for Onondaga County.

You will also need a list of the municipal wastewater treatment plants for Onondaga County. NYS keeps a record of all permitted WWTPs in the state. You can download the file here.

We will be able to use these data to:

1) Map WWTPs in Onondaga County, NY
2) Select parcels that are on public sewer
3) Link groups of parcels to the town or village they are within
4) Estimate sewershed locations for nearby WWTPs

A note before proceeding: Parcel data are very large and your R session might slow down during some processing steps.

3.4.0.1 Data you will need for this tutorial

You will need to download the following data files for this tutorial (they are stored with this tutorial).

- Onondaga County tax parcels
- NYS WWTP list
- NY county boundaries
- NY town boundaries
- NY village boundaries

3.4.1 Step 1 - Install and load the necessary R packages

Install packages to your computer.

```
install.packages("gridExtra")
install.packages("nngeo")
install.packages("sf")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("tigris")
```

Load packages into your current R session.

```
library(gridExtra)
library(nngeo)
library(sf)
library(dplyr)
library(ggplot2)
library(tigris)
```

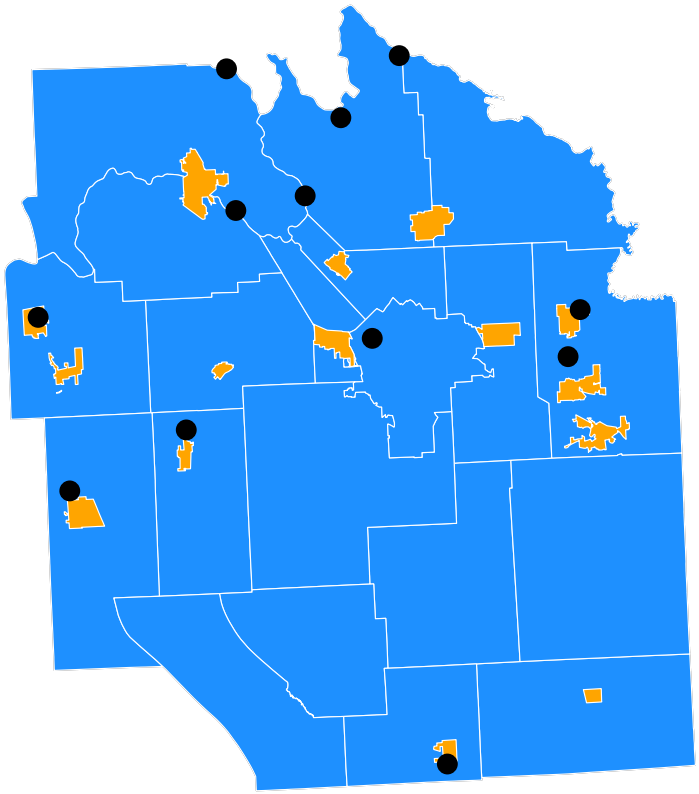
3.4.2 Step 2 - Load the data into your R session

```
# Onondaga county tax parcels
onondaga_parcels <-
  st_read("E:/Data/ny_parcels/onondaga_2020_Tax_Parcels_SHP_2106/onondaga_2020_Tax_Parc

# NY WWTPS
ny_wwtps <- read.csv("data/Wastewater_Treatment_Plants_20250121.csv")

# NY county boundaries - for assistance with mapping
ny_counties <-
  st_read("E:/Dropbox/CEMI/Wastewater/Data/NYS_Civil_Boundaries.shp/Counties_Shoreline

# Filter for Onondaga county
onondaga_border <- ny_counties %>%
  filter(NAME == "Onondaga")

# nys town, city, village boundaries
towns <-
  st_read("E:/Dropbox/CEMI/Wastewater/Data/NYS_Civil_Boundaries.shp/Cities_Towns.shp")

# Onondaga towns
onondaga_towns <- towns %>%
```

```
  filter(COUNTY == "Onondaga")

# villages
villages <-
  st_read("E:/Dropbox/CEMI/Wastewater/Data/NYS_Civil_Boundaries.shp/Villages.shp")

# Onondaga villages
onondaga_villages <- villages %>%
  filter(COUNTY == "Onondaga")
```

3.4.3 Step 3 - Filter the WWTP data for municipal sites only

```
# municipal sites only
wwtp_mun <- ny_wwtps %>%
  filter(Plant.Type == "Municipal")

# make the data spatial for mapping purposes using the coordinates in the file
wwtp_mun_sf <-
  st_as_sf(wwtp_mun,
           coords =  c("Longitude", "Latitude"),
           crs = "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84")

# transform to the same CRS as the Onondaga parcels
wwtp_mun_sf <- st_transform(wwtp_mun_sf, st_crs(onondaga_border))

# filter for Onondaga plants only using st_intersection
wwtp_onondaga_sf <- st_intersection(wwtp_mun_sf, onondaga_border)

# Map out the plants
ggplot()+
  geom_sf(data = onondaga_border, fill = NA)+
  geom_sf(data = onondaga_towns, fill = "dodgerblue", color = "white")+
  geom_sf(data = onondaga_villages, fill = "orange", color = "white")+
  geom_sf(data = wwtp_onondaga_sf, color = "black", size = 3)+
  theme_void()
```



We can see that many WWTPs fall within village or town boundaries. This should help us differentiate the plants that serve different communities. We can also ask the municipalities and local WWTP operators to learn more, but for now, let us assume that we are not sure.

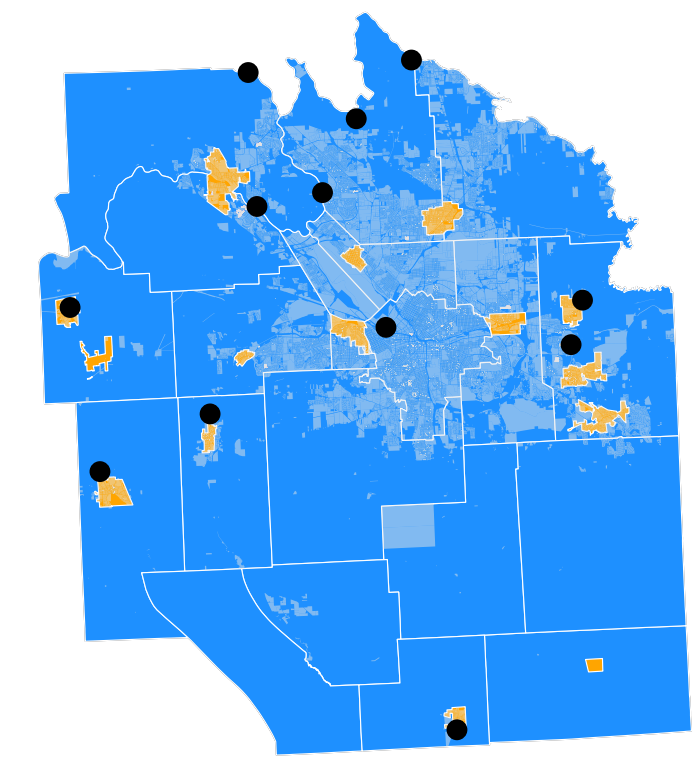
3.4.4 Step 4 - Filter the parcel data for public sewer only

There are two fields in the data for sewer. One is the `SEWER_TYPE` and the other is `SEWER_DESC`. `SEWER_DESC` includes three categories: Comm/public, None, or Private. We will filter for the Comm/public sewer parcels only.

```
# filter for public sewer parcels
table(onondaga_parcels$SEWER_DESC)
```

```
##
## Comm/public        None     Private
##      138461       11440       30452
```

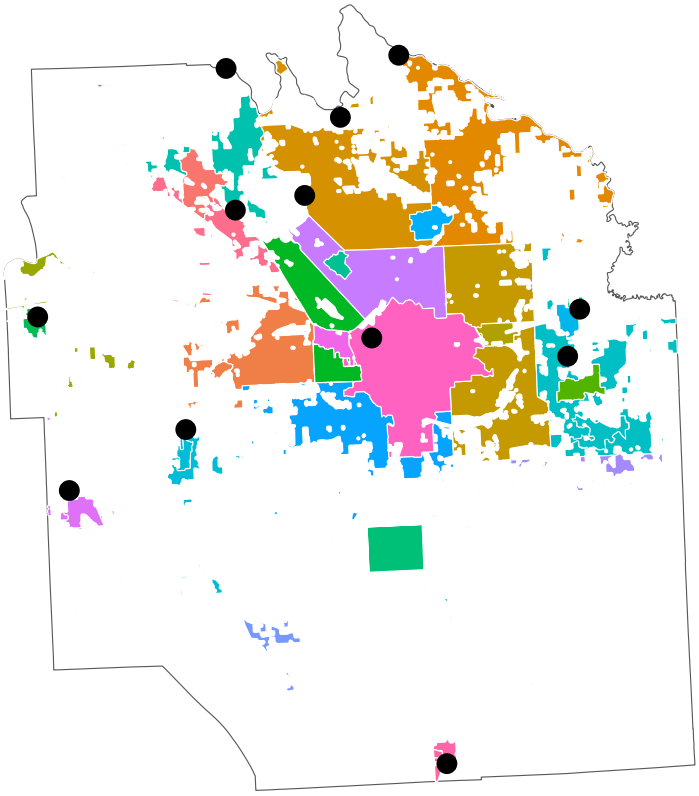
```
onondaga_sewer <- onondaga_parcels %>%
  filter(SEWER_DESC == "Comm/public")

# plot the parcel data
# NOTE - this will take a moment, there are over 100,000 records
ggplot()+
  geom_sf(data = onondaga_border, fill = NA)+
  geom_sf(data = onondaga_towns, fill = "dodgerblue", color = "white")+
  geom_sf(data = onondaga_villages, fill = "orange", color = "white")+
  geom_sf(data = onondaga_sewer, color = NA, alpha = 0.5)+
  geom_sf(data = wwtp_onondaga_sf, color = "black", size = 3)+
  theme_void()+
  labs(title = "Onondaga county parcels connected to public sewer")
```



3.4.5 Step 5 - Link sewer parcels to towns and villages

```
# cut out the village boundaries from the towns
town_no_village <-st_difference(onondaga_towns, st_combine(onondaga_villages))

# intersect the villages/towns with the sewer parcels
sewer_town_intersection <- st_intersection(onondaga_sewer, town_no_village)
sewer_village_intersection <- st_intersection(onondaga_sewer, onondaga_villages)

# group by town / village unit and dissolve
# Note we are using a buffer to help remove extra spaces and lines between
# parcels

# first we buffer
dissolved_towns <- sewer_town_intersection %>%
  st_as_sf() %>%
  st_buffer(100)

# then dissolve
dissolved_towns <- dissolved_towns %>%
  group_by(NAME) %>% # name of the town
  summarise() %>%
  ungroup()

# remove the buffer distance
dissolved_towns <- dissolved_towns %>%
  st_buffer(-100)

# villages

# first we buffer
dissolved_villages <- sewer_village_intersection %>%
  st_as_sf() %>%
  st_buffer(100)

# then dissolve
dissolved_villages <- dissolved_villages %>%
  group_by(NAME) %>%
  summarise() %>%
  ungroup()

# remove the buffer distance
dissolved_villages <- dissolved_villages %>%
  st_buffer(-100)
```

```
# plot to see how they look
ggplot()+
  geom_sf(data = onondaga_border, fill = NA)+
  geom_sf(data = dissolved_towns, color = "white", aes(fill = `NAME`))+
    geom_sf(data = dissolved_villages, color = "white", aes(fill = `NAME`))+
  geom_sf(data = wwtp_onondaga_sf, size = 3, color = "black")+
  theme_void()+
  theme(legend.position = "none")
```



We now have two layers, parcels in each town or city that are on sewer and parcels in each village that are on sewer. The next step can be to take these data to a local county planning office and identify which town sewer systems go to each plant, then group the polygons to create the sewershed for that plant.

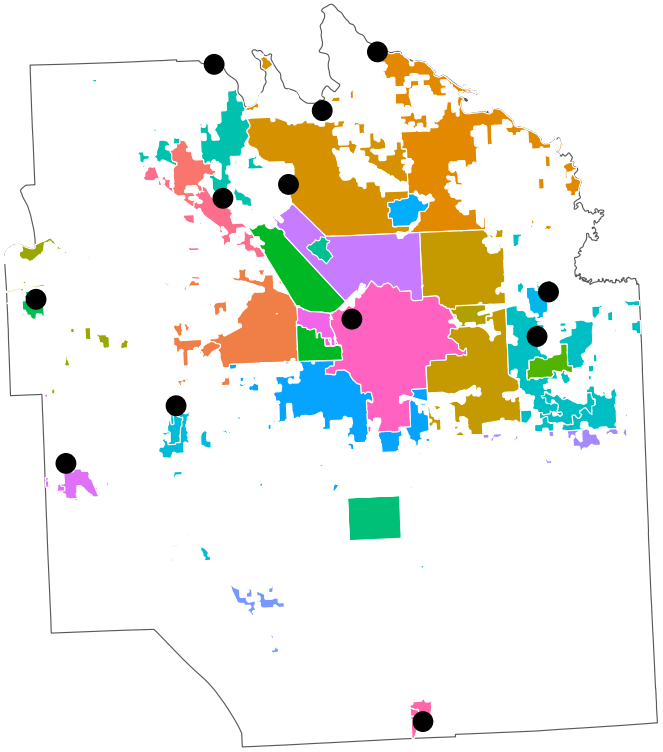
Alternatively, let's do a little more work to make that next step easier. You will notice there are a lot of wayward parcels (parcels that are not near any WWTP or town and look like small "islands"). They likely flow into nearby systems a little ways off. Also, there are a lot of holes in the parcel groups. We will fill in those holes and then intersect the town and village sewer parcels with census blocks to get more robust geometries for the sewersheds. These steps will make

using these data for epidemiology a little easier. If you want to keep the more skeletal boundaries for sewers, you can save that data for future reference.

3.4.5.1 Clean up sewershed boundaries

```
# remove holes

dissolved_towns_filled <- nngeo::st_remove_holes(dissolved_towns)
dissolved_villages_filled <- nngeo::st_remove_holes(dissolved_villages)

sewer_parcel_plot <-
  ggplot()+
  geom_sf(data = onondaga_border, fill = NA)+
  geom_sf(data = dissolved_towns_filled, color = "white", aes(fill = NAME))+
  geom_sf(data = dissolved_villages_filled, color = "white", aes(fill = NAME))+
  geom_sf(data = wwtp_onondaga_sf, size = 3, color = "black")+
  labs(title = "Town and village parcels that are on sewer")+
  theme_void()+
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5))

sewer_parcel_plot
```

Town and village parcels that are on sewer

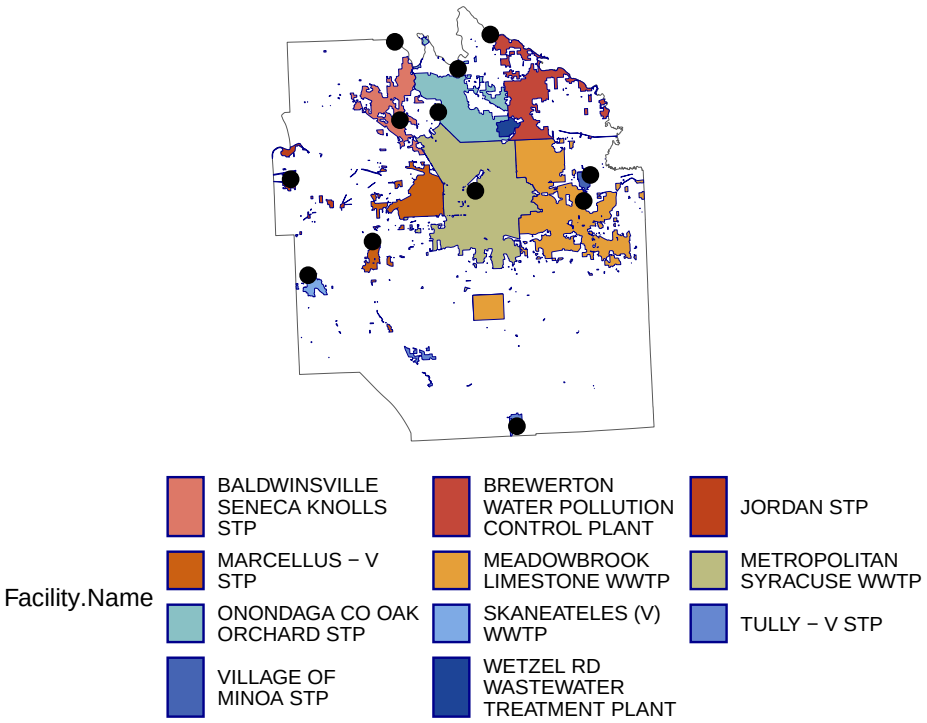


3.4.6 Step 6 - Assign polygons to WWTPs

We want to now bring our village and town parcel polygons together into one shapefile layer and link the polygons to the nearest WWTP.

```
# combine village and town parcels into on R data object
village_town_sewer <- bind_rows(
                                dissolved_towns_filled,
                                dissolved_villages_filled
                                )


# assign each polygon to a point based on proximity
near <- st_nearest_feature(village_town_sewer, wwtp_onondaga_sf)

# using the row numbers for the nearest feature, we can begin to create some
# sewershed boundaries
wwtp_onondaga_sf$wwtp_row_id <- seq(1:nrow(wwtp_onondaga_sf))
```

```
village_town_sewer$parcel_row_id <- seq(1:nrow(village_town_sewer))

# combine the list of near wwtps with the Onondaga sewer dataset
near_df <- as.data.frame(cbind(village_town_sewer, near))

# rename near value to be the wwtp_row_id
near_df$wwtp_row_id <- near_df$near

# to avoid extra "lines" in our dissolved polygons, we buffer the parcels first

# first we buffer
dissolved <- near_df %>%
  st_as_sf() %>%
  st_buffer(10)

# then dissolve
dissolved <- dissolved %>%
  group_by(wwtp_row_id) %>%
  summarise() %>%
  ungroup()

# remove the buffer distance
dissolved <- dissolved %>%
  st_buffer(-10)

# add the wwtp metadata
onondaga_meta <- wwtp_onondaga_sf %>%
  st_drop_geometry()

dissolved <- dissolved %>%
  left_join(onondaga_meta, by = c("wwtp_row_id"))

# plot to see how they look
ggplot()+
  geom_sf(data = onondaga_border, fill = NA)+
  geom_sf(data = dissolved, color = "darkblue", aes(fill = `Facility.Name`))+
  geom_sf(data = wwtp_onondaga_sf, size = 3, color = "black")+
  theme_void()+
  theme(legend.position = "bottom")+
  guides(fill = guide_legend(nrow = 4, byrow=TRUE))+
  scale_fill_manual(values = MetBrewer::met.brewer(name = "Nizami", n = 11),
                    labels = function(x) stringr::str_wrap(x, width = 15))
```



We can save these R objects as shapefiles.

```
st_write(dissolved_towns_filled,
         dsn = "data/onondaga town sewer parcels/onondaga town sewer parcels.shp")

st_write(dissolved_villages_filled,
         dsn = "data/onondaga village sewer parcels/onondaga village sewer parcels.shp")

st_write(dissolved,
         dsn = "data/town village sewer parcels/town village sewer parcels.shp")
```

3.5 Training review

We have used publicly available tax parcel data to estimate sewersheds for municipal WWTPs in Onondaga County, NY. The boundaries are not exact nor are they complete. You will need to work with local planning offices to improve the boundary estimates and finalize assigning each polygon to the WWTP that they flow into.

Some additional data you might gather to create the final boundaries include municipal tax roles, municipal sewer infrastructure maps or GIS data, or your tax parcel data might have what in NY is called a “Special Districts” table.

3.5.1 Appendix 1 - Special districts

The Special Districts table is a dataset that includes more detail on each tax parcel like what water and fire district the parcel is part of. These tables are not always publicly available and you might need permission to work with these data. If you can obtain these data, you can merge the special districts to the parcel data using the municipal tax parcel ID, and then group and filter by special district names for sewer districts. Not all special district table will have sewer data, but many will.

3.5.2 Appendix 2 - create polygons using census blocks

We can also draw sewershed boundaries using census blocks. By intersecting census blocks with sewer parcels, we can map the parts of each town and village on sewer.

An advantage to including the entire block that intersects with a sewershed parcel is that it will make merging these data with census population data easier in the future. Obtaining accurate population estimates for the number of people on sewer is an important part of wastewater-based epidemiology.

3.5.2.1 Download censusblocks for NYS

```
b <- tigris::blocks(state = "New York", county = "Onondaga", year = 2020)
```

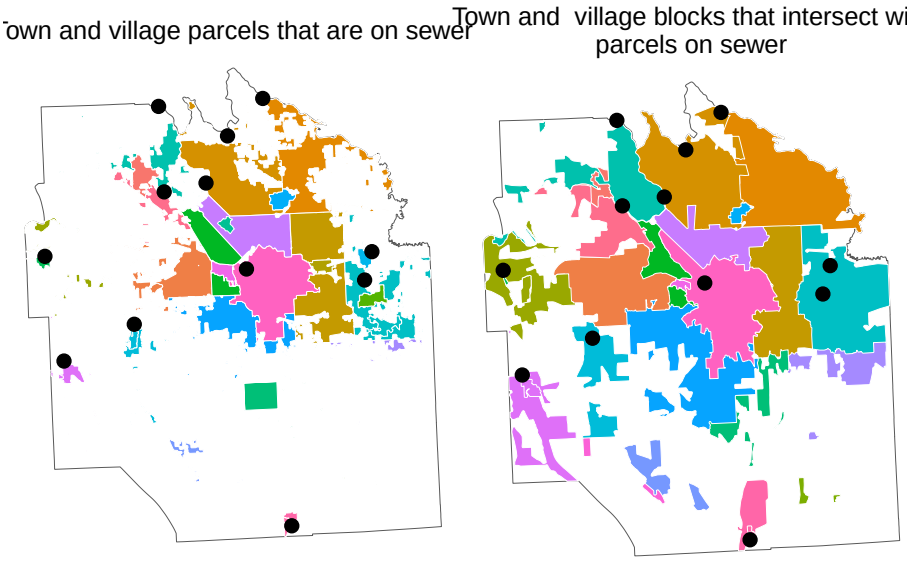
```
# blocks that intersect with town sewer parcels
# transform crs to match
b <- st_transform(b, st_crs(dissolved_towns_filled))

town_sewer_b <- st_intersects(b, dissolved_towns_filled)
town_sewer_b <- as.data.frame(town_sewer_b)

# blocks that intersect with village sewer parcels
village_sewer_b <- st_intersects(b, dissolved_villages_filled)
village_sewer_b <- as.data.frame(village_sewer_b)

# add row ids and then select for row ids in the intersection
b$b_row_id <- seq(1:nrow(b))

village_sewer_blocks <- b %>%
  filter(b_row_id %in% village_sewer_b$row.id)
town_sewer_blocks <- b %>%
   filter(b_row_id %in% town_sewer_b$row.id)
```

```r
# add town / village sewer
dissolved_towns_filled$town_id <- seq(1:nrow(dissolved_towns_filled))
dissolved_villages_filled$village_id <- seq(1:nrow(dissolved_villages_filled))

village_sewer_b <- village_sewer_b %>%
  rename(village_id = col.id,
         b_row_id = row.id) %>%
  left_join(dissolved_villages_filled, by = c("village_id")) %>%
  st_drop_geometry() %>%
  select(-geometry)

# add to village_sewer_blocks
village_sewer_blocks <- left_join(village_sewer_blocks, village_sewer_b,
                                  by = c("b_row_id")
                                  )

# group by village and dissolve

# first we buffer
village_sewer_blocks <- village_sewer_blocks %>%
  st_as_sf() %>%
  st_buffer(100)

# then dissolve
village_sewer_blocks <- village_sewer_blocks %>%
  group_by(NAME) %>%
  summarise() %>%
  ungroup()

# remove the buffer distance
village_sewer_blocks <- village_sewer_blocks %>%
  st_buffer(-100)

## repeat for towns

town_sewer_b <- town_sewer_b %>%
  rename(town_id = col.id,
         b_row_id = row.id) %>%
  left_join(dissolved_towns_filled, by = c("town_id")) %>%
  st_drop_geometry() %>%
  select(-geometry)

# add to village_sewer_blocks
town_sewer_blocks <- left_join(town_sewer_blocks, town_sewer_b,
                               by = c("b_row_id")
```

```
                                  )

# group by village and dissolve

# first we buffer
town_sewer_blocks <- town_sewer_blocks %>%
  st_as_sf() %>%
  st_buffer(100)

# then dissolve
town_sewer_blocks <- town_sewer_blocks %>%
  group_by(NAME) %>%
  summarise() %>%
  ungroup()

# remove the buffer distance
town_sewer_blocks <- town_sewer_blocks %>%
  st_buffer(-100)

# fill holes
town_sewer_blocks_filled <- nngeo::st_remove_holes(town_sewer_blocks)
village_sewer_blocks_filled <- nngeo::st_remove_holes(village_sewer_blocks)

block_sewer_plot <-
  ggplot()+
  geom_sf(data = onondaga_border, fill = NA)+
  geom_sf(data = village_sewer_blocks_filled, color = "white",
          aes(fill = NAME)
          )+
  geom_sf(data = town_sewer_blocks_filled, color = "white",
          aes(fill = NAME)
          )+
  geom_sf(data = wwtp_onondaga_sf, size = 3, color = "black")+
  labs(title = "Town and  village blocks that intersect with\nparcels on sewer")+
  theme_void()+
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5))

# block_sewer_plot
```

So now we have two sets of data to bring to the local planning offices or WWTP operators to help in creating the final boundaries. Let's take a look at them side by side.

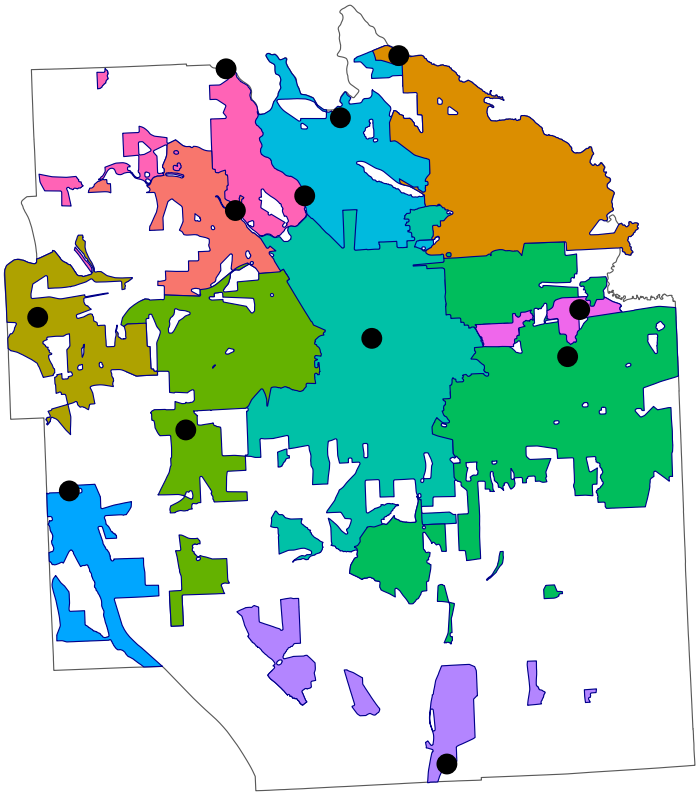
```
gridExtra::grid.arrange(sewer_parcel_plot, block_sewer_plot, nrow = 1)
```



We can also link these data to the nearest WWTP like we did for parcels.

```
# combine into one object
village_town_blocks <- bind_rows(town_sewer_blocks, village_sewer_blocks)

# assign each polygon to a point based on proximity
near <- st_nearest_feature(village_town_blocks, wwtp_onondaga_sf)

# using the rownumbers for the nearest feature, we can begin to create some
# sewershed boundaries
wwtp_onondaga_sf$wwtp_row_id <- seq(1:nrow(wwtp_onondaga_sf))
village_town_blocks$parcel_row_id <- seq(1:nrow(village_town_blocks))

# combine the list of near wwtps with the onondaga sewer dataset
near_df <- as.data.frame(cbind(village_town_blocks, near))

# rename near value to be the wwtp_row_id
near_df$wwtp_row_id <- near_df$near

# to avoid extra "lines" in our dissolved polygons, we buffer the parcels first

# first we buffer
dissolved <- near_df %>%
  st_as_sf() %>%
  st_buffer(10)
```

```r
# then dissolve
dissolved <- dissolved %>%
  group_by(wwtp_row_id) %>%
  summarise() %>%
  ungroup()

# remove the buffer distance
dissolved <- dissolved %>%
  st_buffer(-10)

# add the wwtp metadata
onondaga_meta <- wwtp_onondaga_sf %>%
  st_drop_geometry()

dissolved <- dissolved %>%
  left_join(onondaga_meta, by = c("wwtp_row_id"))

# plot to see how they look
ggplot()+
  geom_sf(data = onondaga_border, fill = NA)+
  geom_sf(data = dissolved, color = "darkblue", aes(fill = `Facility.Name`))+
  geom_sf(data = wwtp_onondaga_sf, size = 3, color = "black")+
  theme_void()+
  theme(legend.position = "none")
```

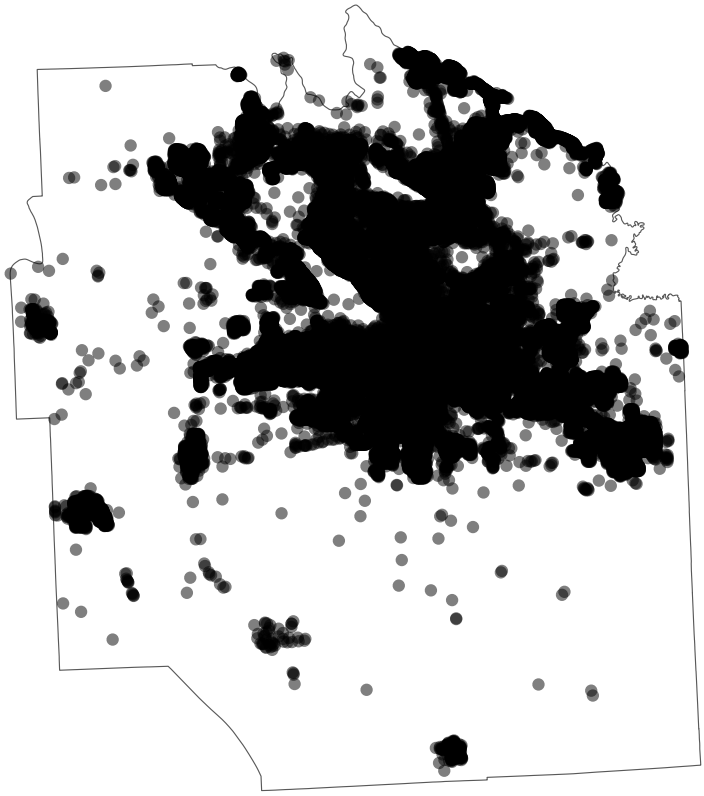


3.5.3 Appendix 3 - Polygons from point data

Some datasets may be made up of points like manhole locations or points for each parcel connected to a sewer system. You can create a polygon from point data using the following code.

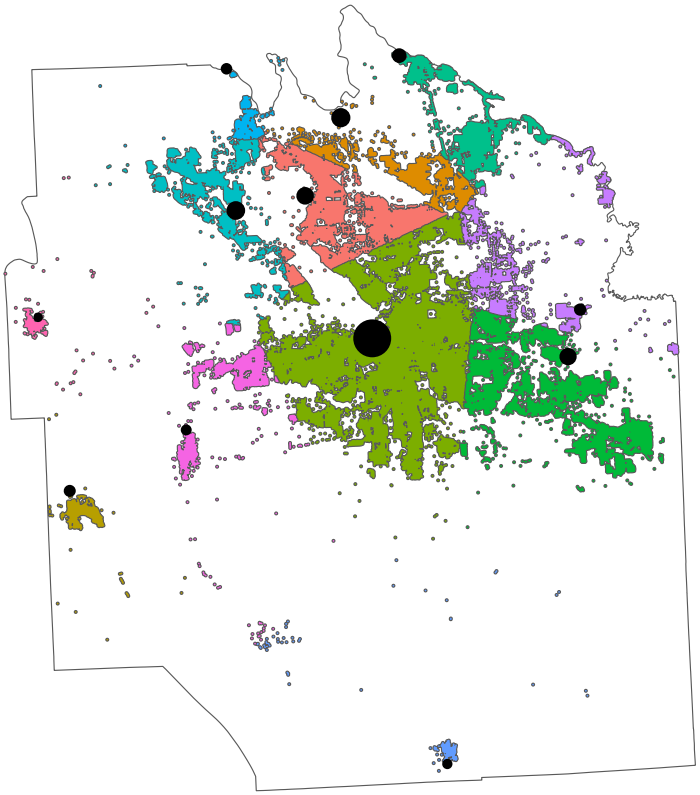
```
# change parcels to centroids
onondaga_sewer_centroids <- st_centroid(onondaga_sewer)

# find the nearest wwtp for each sewer parcel
near <- st_nearest_feature(onondaga_sewer_centroids, wwtp_onondaga_sf)

# using the row numbers for the nearest feature, we can begin to create some
# sewershed boundaries
wwtp_onondaga_sf$wwtp_row_id <- seq(1:nrow(wwtp_onondaga_sf))
onondaga_sewer_centroids$parcel_row_id <- seq(1:nrow(onondaga_sewer_centroids))

# combine the list of near wwtps with the Onondaga sewer dataset
near_df <- as.data.frame(cbind(onondaga_sewer_centroids, near))
```

```
# rename near value to be the wwtp_row_id
near_df$wwtp_row_id <- near_df$near

# plot the points
ggplot()+
  geom_sf(data = onondaga_border, fill = NA)+
  geom_sf(data = st_as_sf(near_df), alpha = 0.5)+
  theme_void()
```



```
# make a buffer around each point
near_df <- near_df %>%
  st_as_sf() %>%
  st_buffer(dist = 100) # you might need to play with this buffer distance

# group by wwtp_row_id and merge into polygons
# then dissolve
dissolved <- near_df %>%
  group_by(wwtp_row_id) %>%
  summarise() %>%
```

```
  ungroup()

# plot
ggplot()+
  geom_sf(data = onondaga_border, fill = NA)+
  geom_sf(data = dissolved, aes(fill = as.factor(wwtp_row_id)))+
  geom_sf(data = wwtp_onondaga_sf, aes(size = `Average.Design.Hydraulic.Flow`))+
  theme_void()+
  theme(legend.position = "none")
```



You can then fill the holes like we did before or do some additional cleaning in ArcGIS or from the special districts tables.

Chapter 4

Adding Census Data to Wastewater-Based Epidemiology

4.1 Overview

Wastewater-based epidemiology (WBE) data can be enhanced by adding other public data sources. One source is United States Census (US Census) data, which can add population demographic variables to WBE data, This vignette demonstrates how to take a spatial WBE dataset of sewershed boundaries in New York State (NYS) and add US Census data using spatial intersection methods.

4.1.1 Install and Load R packages

Install packages to your computer.

```
install.packages("dplyr")
install.packages("ggplot2")
install.packages("sf")
install.packages("tidycensus")
install.packages("units")
```

Load packages into your R session.

```
library(dplyr) # data wrangling
library(sf) # spatial data manipulation
library(ggplot2) # making plots and maps
library(tidycensus) # download tabular census data
library(tigris) # download spatial census data
library(tidyr) # data wrangling
library(units) # handling variables with unit assignment
```

4.2 Load wastewater data

Sewershed boundaries for NYS can be downloaded from ArcGIS online.

```
# load the boundaries
sewersheds <- st_read("data/New York State Sewersheds/New York State Sewersheds.shp")

# select the variables we want to keep
sewersheds <- sewersheds %>%
  dplyr::select(County, City, WWTP_ID, WWTP, Sewershed, SW_ID, geometry)
```

Map your data.

```
# plot to view the data
ggplot()+
  geom_sf(data = sewersheds, fill = "white", color = "black")+
  theme_void()
```

4.3 Load US Census data method - Tidy Census

US Census data can be added to your R session using the packages `tidycensus` and `tigris`. Both packages connect to the US Census API for downloading tabular data or spatial boundaries. You will need an API key to access the data. For information on obtaining a key, visit the `tidycensus` website.

```
# retrieve your API key if stored in the environment
census_api_key(Sys.getenv('CENSUS_API_KEY'))
```

For this example, we are going to download 2020 decennial census data and calculate the total number of individuals that self-identify as black or African American from that year. We will calculate totals and percentage of total population. We are going to use census tracts as our geography.

To download the data, we need to know the name of the variables we want and for that, we use the `load_variables` command.

```
# load the variable names into an R object. cache = TRUE can speed the data download s
census_var_names <- load_variables(year = 2020, dataset = "pl", cache = TRUE)
```

```
# look at the first few rows of data
head(census_var_names)
```

```
## # A tibble: 6 x 3
##   name    label                                                concept
##   <chr>   <chr>                                                <chr>
## 1 H1_001N " !!Total:"                                          OCCUPANCY STATUS
## 2 H1_002N " !!Total:!!Occupied"                                OCCUPANCY STATUS
## 3 H1_003N " !!Total:!!Vacant"                                  OCCUPANCY STATUS
## 4 P1_001N " !!Total:"                                          RACE
## 5 P1_002N " !!Total:!!Population of one race:"                 RACE
## 6 P1_003N " !!Total:!!Population of one race:!!White alone"    RACE
```

This function can load the variable names for many datasets from the US Census and provides the name, the variable description, and the concept group that the variable is part of. Looking at the options, we can see that we want a variable from the `RACE` concept, specifically, `P1_004N`, which is "!!Total:!!Population of one race:!!Black or African American alone". For this exercise, we will not be included those that identify as multiple racial or ethnic groups, but you can adapt the methods presented to make those calculations.

To get the total for each census tract that identify as black or African American, we need this variable. However, to calculate the percent of the population, we also need the total population, which is the variable " !!Total:" or `P1_001N`. We will use the `get_dencennial` function to download the data into our R session.

```
# download the decennial data for the tract geography for NYS
census_data <- get_decennial(
  year = 2020,
  geography = "tract",
  state = "New York",
  variables = c(
    c(population_total = "P1_001N"),
    black_alone = "P1_004N"
  )
)
```

```
# view the first few rows of data
head(census_data)
```

```
## # A tibble: 6 x 4
```

```
##   GEOID       NAME                                       variable         value
##   <chr>       <chr>                                      <chr>            <dbl>
## 1 36001000100 Census Tract 1, Albany County, New York    population_total  2073
## 2 36001000201 Census Tract 2.01, Albany County, New York population_total  3125
## 3 36001000202 Census Tract 2.02, Albany County, New York population_total  2598
## 4 36001000301 Census Tract 3.01, Albany County, New York population_total  3190
## 5 36001000302 Census Tract 3.02, Albany County, New York population_total  3496
## 6 36001000401 Census Tract 4.01, Albany County, New York population_total  2216
```

The resulting dataset is now in our R session with 10,822 rows. We will need to pivot the dataframe to a wider format because it placed the variables in a long format.

```
# pivot the data from long to wide format
census_data <- pivot_wider(census_data,
                           names_from = variable)

# summary of the dataset variables
summary(census_data)
```

```
##     GEOID              NAME           population_total  black_alone
##  Length:5411        Length:5411        Min.   :    0    Min.   :   0.0
##  Class :character   Class :character   1st Qu.: 2490    1st Qu.:  47.5
##  Mode  :character   Mode  :character   Median : 3578    Median : 164.0
##                                        Mean   : 3733    Mean   : 551.9
##                                        3rd Qu.: 4762    3rd Qu.: 667.0
##                                        Max.   :17222    Max.   :7935.0
```

Now we have census counts for individuals that self-identify as black or African American and we want to add these data to our wastewater sewershed boundaries. To do this, we need to add the census geometry (the spatial component) to the R session.

```
# download tracts for NY for 2020 and drop the name column
tracts <- tracts(state = "New York", year = 2020) %>% select(-NAME)

# join the tract geometries to the census data
census_data_sf <- left_join(tracts, census_data, by = c("GEOID"))
```

```
# check the data structure
str(census_data_sf)
```

```
## Classes 'sf' and 'data.frame':   5411 obs. of  15 variables:
##  $ STATEFP         : chr  "36" "36" "36" "36" ...
```

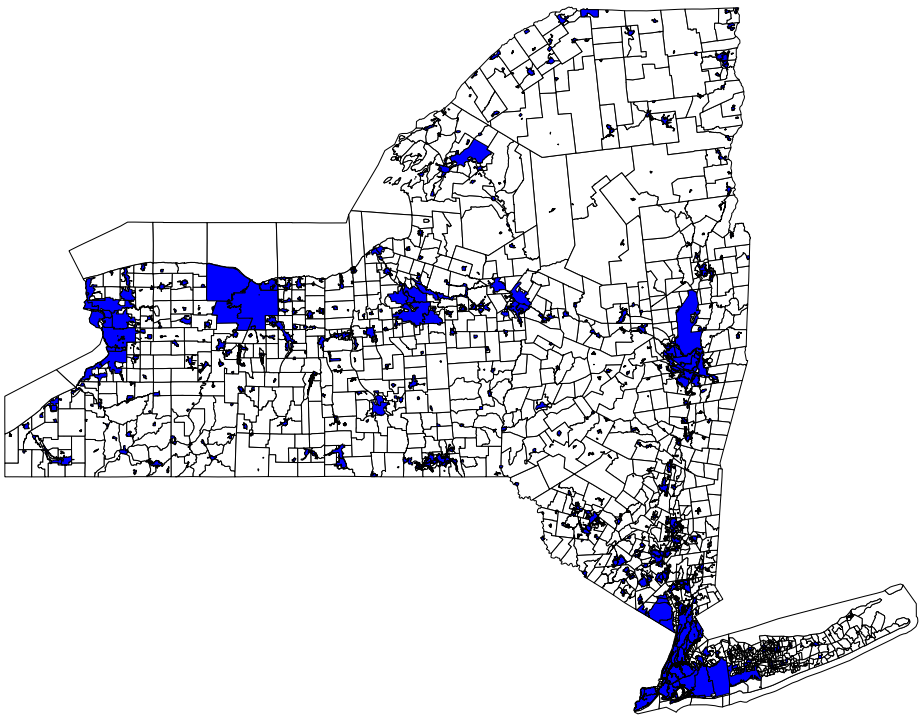
```
##  $ COUNTYFP        : chr  "047" "047" "047" "047" ...
##  $ TRACTCE         : chr  "000700" "000900" "001100" "001300" ...
##  $ GEOID           : chr  "36047000700" "36047000900" "36047001100" "36047001300" .
##  $ NAMELSAD        : chr  "Census Tract 7" "Census Tract 9" "Census Tract 11" "Censu
##  $ MTFCC           : chr  "G5020" "G5020" "G5020" "G5020" ...
##  $ FUNCSTAT        : chr  "S" "S" "S" "S" ...
##  $ ALAND           : num  176774 163469 168507 293167 154138 ...
##  $ AWATER          : num  0 0 0 0 0 ...
##  $ INTPTLAT        : chr  "+40.6923505" "+40.6917206" "+40.6932903" "+40.6976150" .
##  $ INTPTLON        : chr  "-073.9973434" "-073.9916018" "-073.9877087" "-073.988358
##  $ NAME            : chr  "Census Tract 7, Kings County, New York" "Census Tract 9,
##  $ population_total: num  4415 5167 1578 2465 1694 ...
##  $ black_alone     : num  151 197 178 234 172 ...
##  $ geometry        :sfc_MULTIPOLYGON of length 5411; first list element: List of 1
##   ..$ :List of 1
##   .. ..$ : num [1:35, 1:2] -74 -74 -74 -74 -74 ...
##   ..- attr(*, "class")= chr [1:3] "XY" "MULTIPOLYGON" "sfg"
##  - attr(*, "sf_column")= chr "geometry"
##  - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",..: NA NA NA NA NA NA
##   ..- attr(*, "names")= chr [1:14] "STATEFP" "COUNTYFP" "TRACTCE" "GEOID" ...
```

4.4 Adding the Census data to the wastewater data

4.4.1 Spatial transformations

To add our population data to the sewershed data, we have a few steps to complete. First, both of our datasets need to be the same spatial projection so that they spatially overlap and can be mapped on top of each other.

```
# transform the census data CRS to match the sewersheds
census_data_sf <- st_transform(census_data_sf, st_crs(sewersheds))

# plot the data, the layers should overlap
ggplot()+
  geom_sf(data = census_data_sf, fill = "white", color = "black")+
  geom_sf(data = sewersheds, fill = "blue", color = "black")+
  theme_void()
```



4.4.2 Spatial intersection

Now we want to spatially intersect the data. This will assign census count data to the sewersheds that overlap. Sewershed boundaries do not follow census tract boundaries closely, so we have two options for assigning the data. First, we can apportion census data to the sewersheds using the proportion of each census tract area that is within the sewershed area. Alternatively, we can assign the total population for all intersecting tracts to the sewershed.

The first method, called areal apportionment, is generally thought to provide more accurate methods for counts of the data, but assumes that people are evenly distributed within the census tract. The second method could be useful if it is not known how evenly the population is distributed and you want to avoid excluding anyone. We will practice both methods.

4.4.2.1 Areal apportionment

```
# calculate the area of each census tract. Note this field becomes a unit
# variable, we will have to drop those later
census_data_sf$tract_area <- st_area(census_data_sf)
```

```
# turn of spherical geometry to handle overlapping edges and invalid vertices
# warnings
sf::sf_use_s2(FALSE)

# intersect both datasets, creating a new dataset that has slivers of each tract
# that intersects a sewershed
intersection <- st_intersection(sewersheds, census_data_sf)

summary(intersection)
```

```
##     County              City              WWTP_ID              WWTP
##  Length:7963        Length:7963        Length:7963        Length:7963
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   Sewershed            SW_ID              STATEFP            COUNTYFP
##  Length:7963        Length:7963        Length:7963        Length:7963
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    TRACTCE             GEOID              NAMELSAD            MTFCC
##  Length:7963        Length:7963        Length:7963        Length:7963
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    FUNCSTAT             ALAND              AWATER             INTPTLAT
##  Length:7963        Min.   :0.000e+00   Min.   :0.000e+00   Length:7963
##  Class :character   1st Qu.:2.136e+05   1st Qu.:0.000e+00   Class :character
##  Mode  :character   Median :1.176e+06   Median :0.000e+00   Mode  :character
##                     Mean   :1.933e+07   Mean   :2.107e+06
##                     3rd Qu.:6.154e+06   3rd Qu.:1.536e+05
##                     Max.   :1.828e+09   Max.   :1.997e+09
##    INTPTLON             NAME           population_total  black_alone
##  Length:7963        Length:7963        Min.   :    0    Min.   :   0.0
##  Class :character   Class :character   1st Qu.: 2593    1st Qu.:  61.0
##  Mode  :character   Mode  :character   Median : 3672    Median : 211.0
##                                        Mean   : 3803    Mean   : 587.3
##                                        3rd Qu.: 4824    3rd Qu.: 734.5
```

```
##                                      Max.   :17222    Max.   :7935.0
##    tract_area                     geometry
##  Min.   :2.169e+04   GEOMETRYCOLLECTION:  28
##  1st Qu.:2.261e+05   MULTIPOLYGON      :1583
##  Median :1.277e+06   POLYGON           :6352
##  Mean   :2.140e+07   epsg:4269         :   0
##  3rd Qu.:6.565e+06   +proj=long...     :   0
##  Max.   :1.993e+09
```

Our spatial data are now combined into one spatial data object.

```
# calculate a new area for each sliver
intersection$sliver_area <- st_area(intersection)

# calculate the proportion of area that each sliver occupies in the sewershed
intersection$prop_area <- intersection$sliver_area / intersection$tract_area
summary(intersection$prop_area)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.06113 0.88439 0.60984 1.00122 1.00216
```

```
# You will notice some areas are slightly over 1 representing a rounding error. Let's make those

# drop units
intersection$prop_area <- drop_units(intersection$prop_area)
intersection$prop_area <- ifelse(intersection$prop_area > 1,
                                 1,
                                 intersection$prop_area
                                 )
summary(intersection$prop_area)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.06113 0.88439 0.60930 1.00000 1.00000
```

```
# multiply the proportional area by the totals for black or African American and total populatio
intersection$prop_black <- intersection$prop_area * intersection$black_alone
intersection$prop_population <-
  intersection$prop_area * intersection$population_total

# how we will group the data by sewershed and sum the total population and total black populatio
intersection_final <- intersection %>%
  group_by(County, City, WWTP_ID, WWTP, Sewershed, SW_ID) %>%
  summarize(sum_black_alone = sum(prop_black, na.rm = TRUE),
            sum_total_population = sum(prop_population)
```

```
            ) %>%
  ungroup()

# our resulting dataset has 674 rows, one for each sewershed, and proportional estimat
summary(intersection_final)
```

```
##     County              City              WWTP_ID              WWTP
##  Length:674         Length:674         Length:674         Length:674
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   Sewershed            SW_ID           sum_black_alone     sum_total_population
##  Length:674         Length:674         Min.   :     0.00   Min.   :      0.2
##  Class :character   Class :character   1st Qu.:     0.74   1st Qu.:     66.6
##  Mode  :character   Mode  :character   Median :    12.91   Median :    595.6
##                                        Mean   :  4932.45   Mean   :  27214.5
##                                        3rd Qu.:   381.10   3rd Qu.:   7356.8
##                                        Max.   :307461.96   Max.   :1154606.8
##                geometry
##  GEOMETRYCOLLECTION:  2
##  MULTIPOLYGON      :160
##  POLYGON           :512
##  epsg:4269         :  0
##  +proj=long...     :  0
##
```

We have estimated totals, which can be useful, but often, percentage of the population is more informative for analyses. We can get an estimated proportion of the population by dividing the total black alone variable by the total population value.Also, you will notice that the values are no longer integers, meaning that there are estimates for total number of people that include decimals. These can be rounded to the nearest whole number to return the data back to integers.

```
# calculate the proportion
intersection_final$prop_black <-
  intersection_final$sum_black_alone / intersection_final$sum_total_population

# plot the percentages
ggplot()+
  geom_sf(data = intersection_final, aes(fill = prop_black), color = "black")+
  theme_void()
```

4.4.2.2 Intersection assignment

The other method for assigning data is to assign the total population for all census tracts that intersect with a sewershed.

```
# st_intersects returns a list of the polygon rows that intersect from each
# object
intersect_list <- st_intersects(sewersheds, census_data_sf)
```

```
# make it a dataframe
intersect_df <- as.data.frame(intersect_list)

# view the top view rows
head(intersect_df)
```

```
##   row.id col.id
## 1      1   1142
## 2      1   1145
## 3      1   1146
## 4      1   2664
## 5      1   3692
## 6      1   5333
```

Column 1 are the row ids for the sewersheds and column 2 are the row ids for the tracts that intersect them. We will add the row ids to each data object, then use those to merge the census counts to the spatial data and sum to totals.

```
# add row ids
sewersheds$sewer_row_id <- as.integer(rownames(sewersheds))
census_data_sf$census_row_id <- as.integer(rownames(census_data_sf))


# rename intersect column and merge to spatial sewersheds
colnames(intersect_df) <- c("sewer_row_id", "census_row_id")
intersect_combined <- full_join(sewersheds, intersect_df,
                                by = c("sewer_row_id")
                                )

# change the census data from a spatial object to a dataframe
census_data_df <- st_drop_geometry(census_data_sf)

# merge census data -> note that we have to remove the spatial data from the
# census data
intersect_combined <- left_join(intersect_combined, census_data_df,
                                by = c("census_row_id")
                                )

# sum the total number of people that are in census tracts intersecting each
# sewershed
intersect_final <- intersect_combined%>%
  group_by(County, City, WWTP_ID, WWTP, Sewershed, SW_ID) %>%
  summarize(sum_black_alone = sum(black_alone, na.rm = TRUE),
            sum_total_population = sum(population_total, na.rm = TRUE)
            ) %>%
  ungroup()

# calculate the proportions
intersect_final$prop_black <-
  intersect_final$sum_black_alone / intersect_final$sum_total_population

# make a map
ggplot()+
  geom_sf(data = intersect_final, aes(fill = prop_black), color = "black")+
  theme_void()
```

4.5 Training review

This tutorial demonstrated a process for adding US Census data to WBE spatial data (sewersheds). We calculated the percentage of each sewershed population that identified as black or African American in the 2020 decennial census. These methods can be used to estimate other data available from the US Census. For more information visit the US Census website or tidycensus.

Chapter 5

Adding other spatial data to WBE

5.1 Overview

In this training, you will link zip code boundaries to sewersheds.

5.1.0.1 Data you will need for this training

- New York State Sewershed boundaries

5.1.1 Install and Load R packages

Install packages

```
install.packages("dplyr")
install.packages("ggplot2")
install.packages("sf")
install.packages("tigris")
install.packages("units")
```

Load packages into your current R session.

```
library(ggplot2)
library(tigris)
library(sf)
library(dplyr)
```

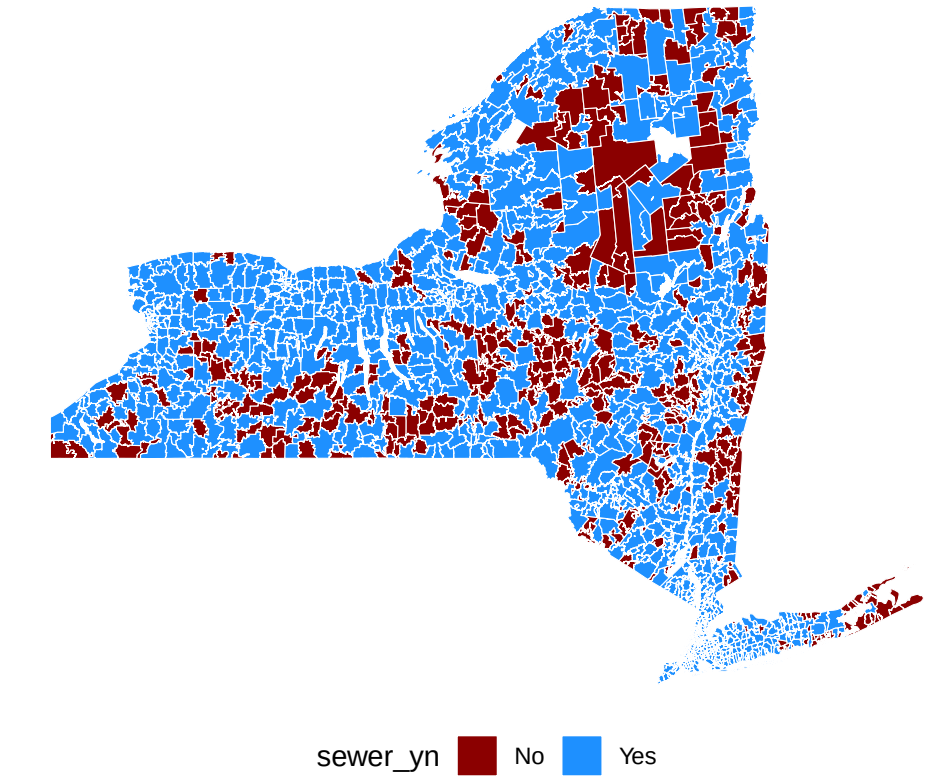
5.1.2 Load data into R

```
# sewershed boundaries for NYS
ny_sewersheds <- st_read("data/New York State Sewersheds/New York State Sewersheds.shp"

ny_zips <- zctas(state = "New York", year = 2010)
```

5.1.3 Intersect zip code TAs with sewersheds

```
# Sewershed boundaries - spatial data #

# Sewersheds need to remove upstream
ny_sewersheds <- ny_sewersheds %>% filter(Method == "Influent")

# there are overlapping sewersheds, so the geometry needs to be made valid
ny_sewersheds <- st_make_valid(ny_sewersheds)

# spatial transform the county data
ny_zips <- st_transform(ny_zips, st_crs(ny_sewersheds))

# intersect the datasets
intersection <- st_intersects(ny_zips, ny_sewersheds)

# the result is a list object of each zip code and what row from the sewershed
# object the zip code matches with

# make it a dataframe
intersection <- as.data.frame(intersection)

# the first column are the rows for the zip codes
# the second column are the rows for the sewersheds

# add row ids to each object so we can make a table for each zip code and what
# sewershed it intersects with
ny_sewersheds$sewer_row_id <- seq(1:nrow(ny_sewersheds))
ny_zips$zip_row_id <- seq(1:nrow(ny_zips))

# rename the columns in the intersection object
intersection <- intersection %>%
  rename(zip_row_id = row.id,
         sewer_row_id = col.id)
```

```
# merge the intersection object with the zip code object
ny_zips_merge <- left_join(ny_zips, intersection, by = c("zip_row_id"))

# add the sewershed identifying information
sewer_id_list <- ny_sewersheds %>%
  st_drop_geometry() %>%
  select(SW_ID, sewer_row_id)

# merge to the zip code object
ny_zips_merge <- left_join(ny_zips_merge, sewer_id_list, by = c("sewer_row_id"))

# add sewer indicator
ny_zips_merge$sewer_yn <- ifelse(is.na(ny_zips_merge$SW_ID), "No", "Yes")

# make a table that could be exported
ny_zip_table <- ny_zips_merge %>%
  select(ZCTA5CE10, sewer_yn, SW_ID) %>%
  rename(zip_code = ZCTA5CE10) %>%
  st_drop_geometry()

# map the zip codes indicating if they intersect with a sewershed or not

ggplot()+
  geom_sf(data = ny_zips_merge, aes(fill = sewer_yn), color = "white")+
  theme_void()+
  theme(legend.position = "bottom")+
  labs(title = "NY Zip codes that are on sewer")+
  scale_fill_manual(values = c("Yes" = "dodgerblue",
                               "No" = "darkred"))
```




```
head(ny_zip_table)
```

```
##   zip_code sewer_yn                SW_ID
## 1    12205      Yes 36001NY0026875ACWWL
## 2    12205      Yes 36001NY0027758ACWWM
## 3    12009      Yes 36001NY0031534ACWWD
## 4    12009      Yes 36001NY0022217ACWWE
## 5    12009      Yes 36001NY0022225ACWWF
## 6    14804       No                 <NA>
```

5.2 Training review

This data file can then be used to link other zip code level data to sewersheds. If a zip code intersects with a sewershed, then residents of that zip code likely live on or near the sewer system.

Chapter 6

System design for outbreak detection

Understanding outbreak detection in wastewater surveillance

Chapter 7

Resources and links

7.1 Training Videos

The videos below are available on our public YouTube channel. They are suggested to view prior to starting the training exercise.

include video library links for each video

include list of data sources and links

7.2 Data

All the data used in these tutorials are publicly available at the following links:

- Systems Design GitHub Page. Click on the **data** folder.
- NYS sewershed data (ArcGIS online download). Access and download NYS sewershed data to see how data are recorded.
- NYS sewershed data (GitHub data download). Access and download NYS sewershed data directly into your R session.
- National parcel data viewer / data access portal. This is a data portal that lets you explore each state in the US and whether they have tax parcel data. Some links bring you to county or state websites and from there you can view and download the data.

7.3 Other Resources

- NYS sewershed data viewer. Visually inspect NYS sewersheds and the population served by each. The page includes example code for downloading and adding population data to sewershed shapefiles in the program R.
- Using the Editor toolbar in ArcGIS. ArcGIS can be an excellent way to digitally edit spatial data.
- R spatial tools. R is an open-source program, and most mapping projects and analyses can be completed using the software.