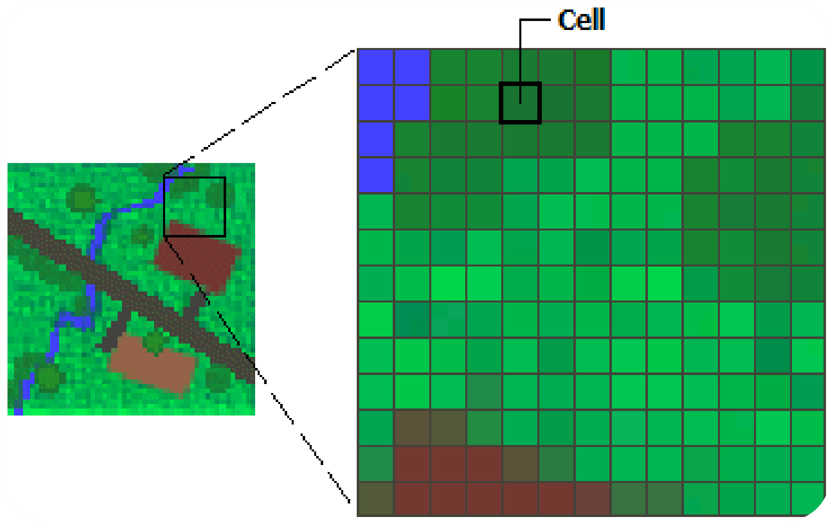
ENVIRONMENTAL DATA ANALYTICS: M9 – SPATIAL ANALYSIS - *LAB*

ENV 872

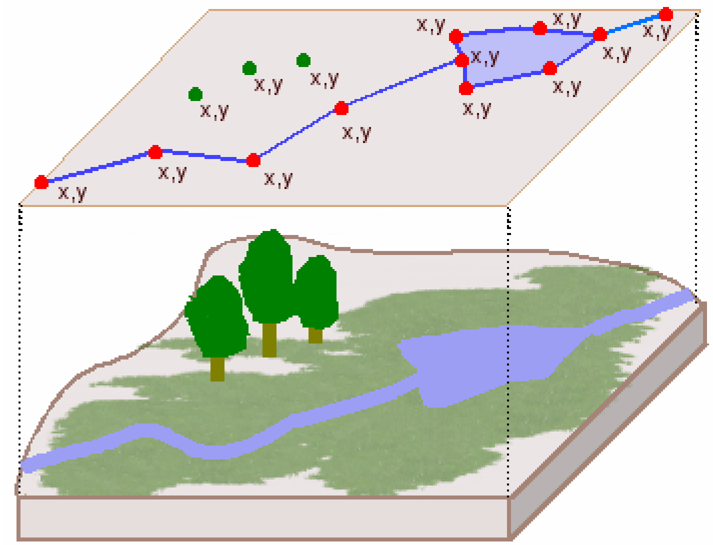
Nicholas School of the Environment - Duke University

Spatial data models

How are spatial features represented in code?



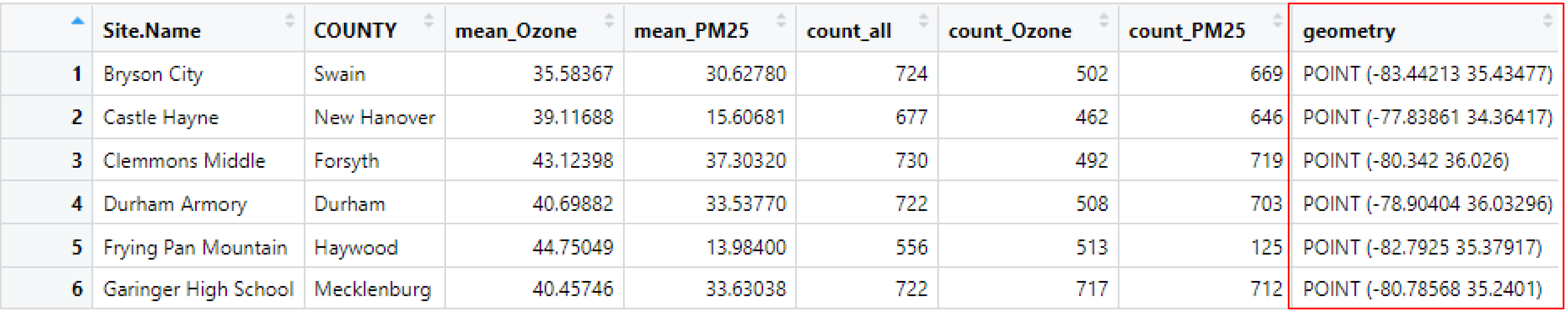
RASTER Data Model



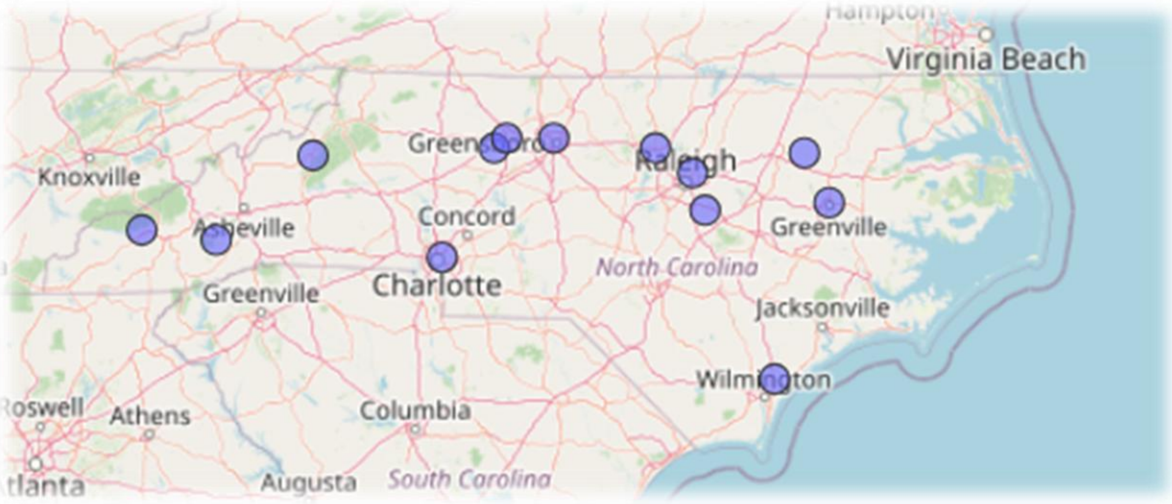
VECTOR Data Model

Vector data & the SF package

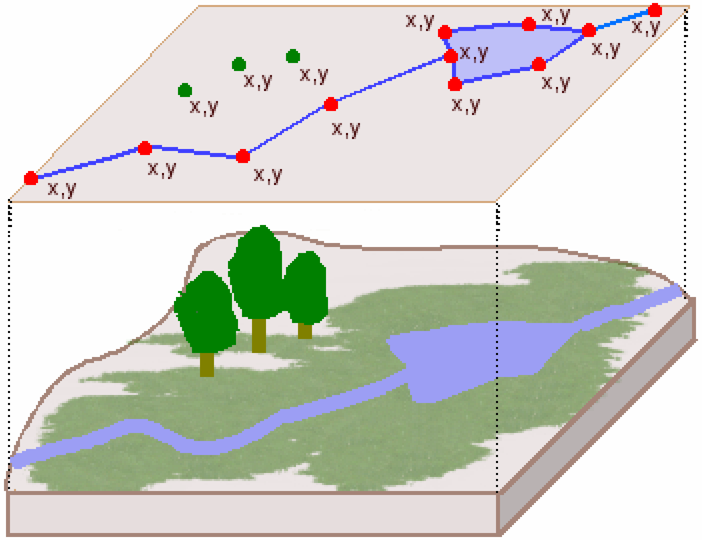
- Simple feature ("*sf*") objects, aka "spatial dataframes"

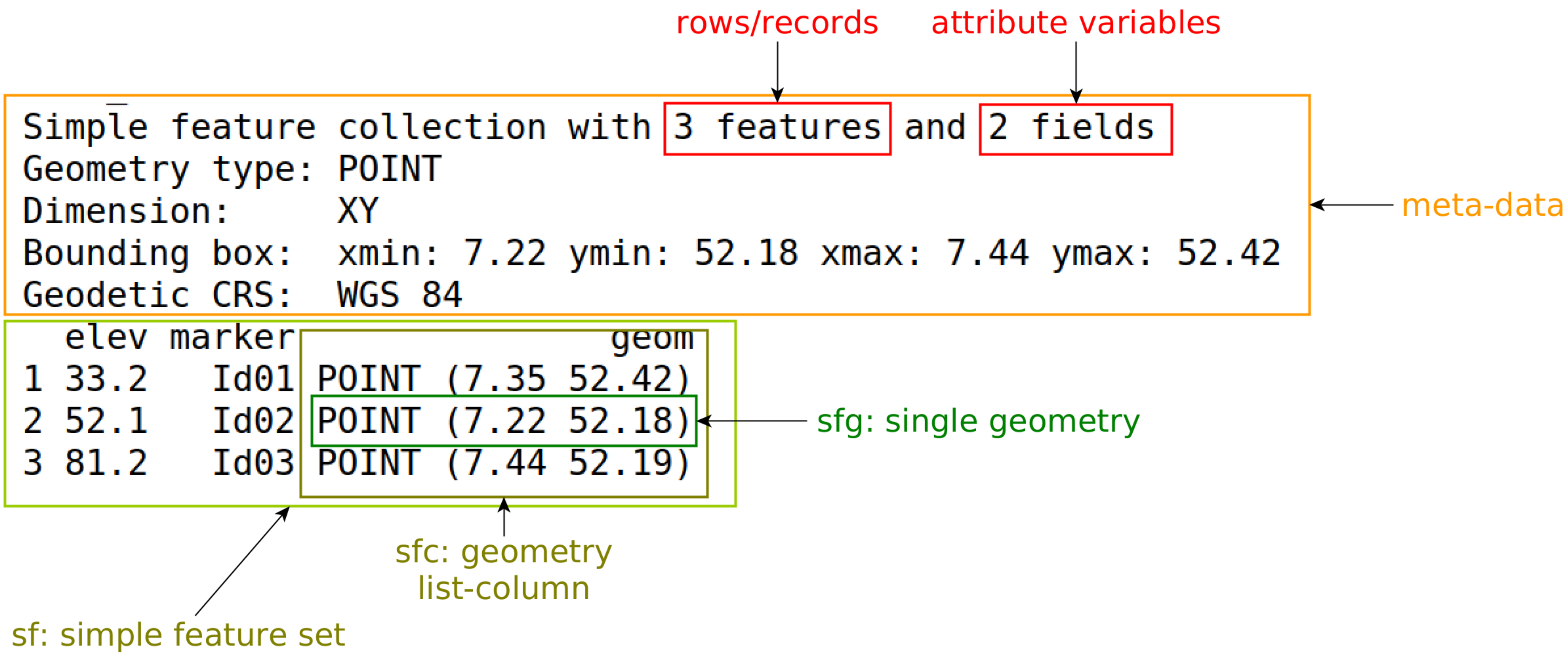


	Site.Name	COUNTY	mean_Ozone	mean_PM25	count_all	count_Ozone	count_PM25	geometry
1	Bryson City	Swain	35.58367	30.62780	724	502	669	POINT (-83.44213 35.43477)
2	Castle Hayne	New Hanover	39.11688	15.60681	677	462	646	POINT (-77.83861 34.36417)
3	Clemmons Middle	Forsyth	43.12398	37.30320	730	492	719	POINT (-80.342 36.026)
4	Durham Armory	Durham	40.69882	33.53770	722	508	703	POINT (-78.90404 36.03296)
5	Frying Pan Mountain	Haywood	44.75049	13.98400	556	513	125	POINT (-82.7925 35.37917)
6	Garinger High School	Mecklenburg	40.45746	33.63038	722	717	712	POINT (-80.78568 35.2401)



Simple Features…



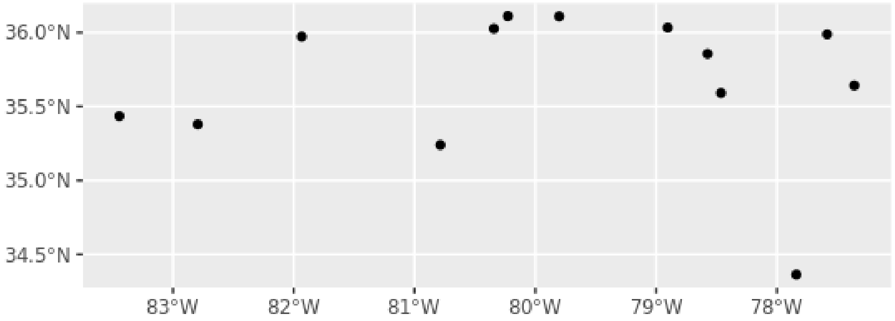


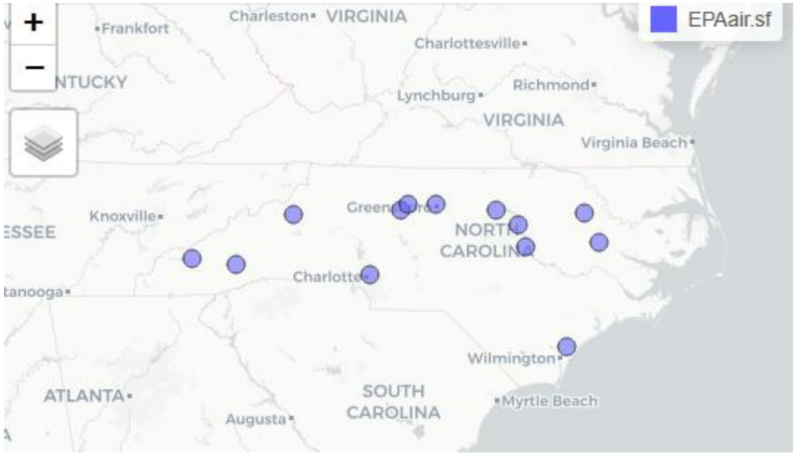
```
Simple feature collection with 3 features and 2 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 7.22 ymin: 52.18 xmax: 7.44 ymax: 52.42
Geodetic CRS:  WGS 84
  elev marker                 geom
1 33.2    Id01 POINT (7.35 52.42)
2 52.1    Id02 POINT (7.22 52.18)
3 81.2    Id03 POINT (7.44 52.19)
```

Exercise:

Dataframe → Spatial dataframe

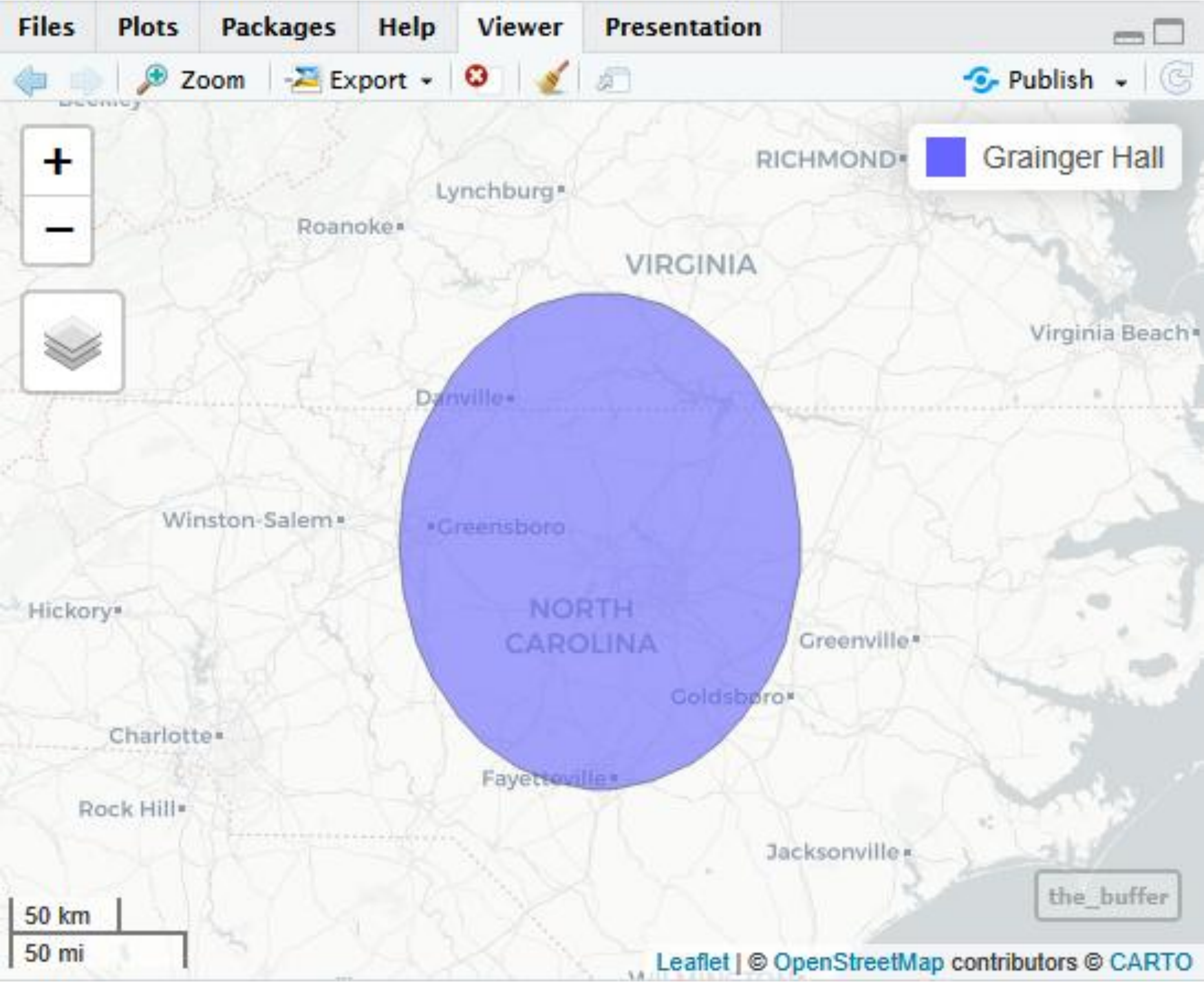
- □ Create dataframe from *EPAair_O3_PM25_NC1819_Processed.csv*
- □ Wrangle so only 1 record for each site
- □ Convert to spatial dataframe (aka "sf object")
- □ Plot:



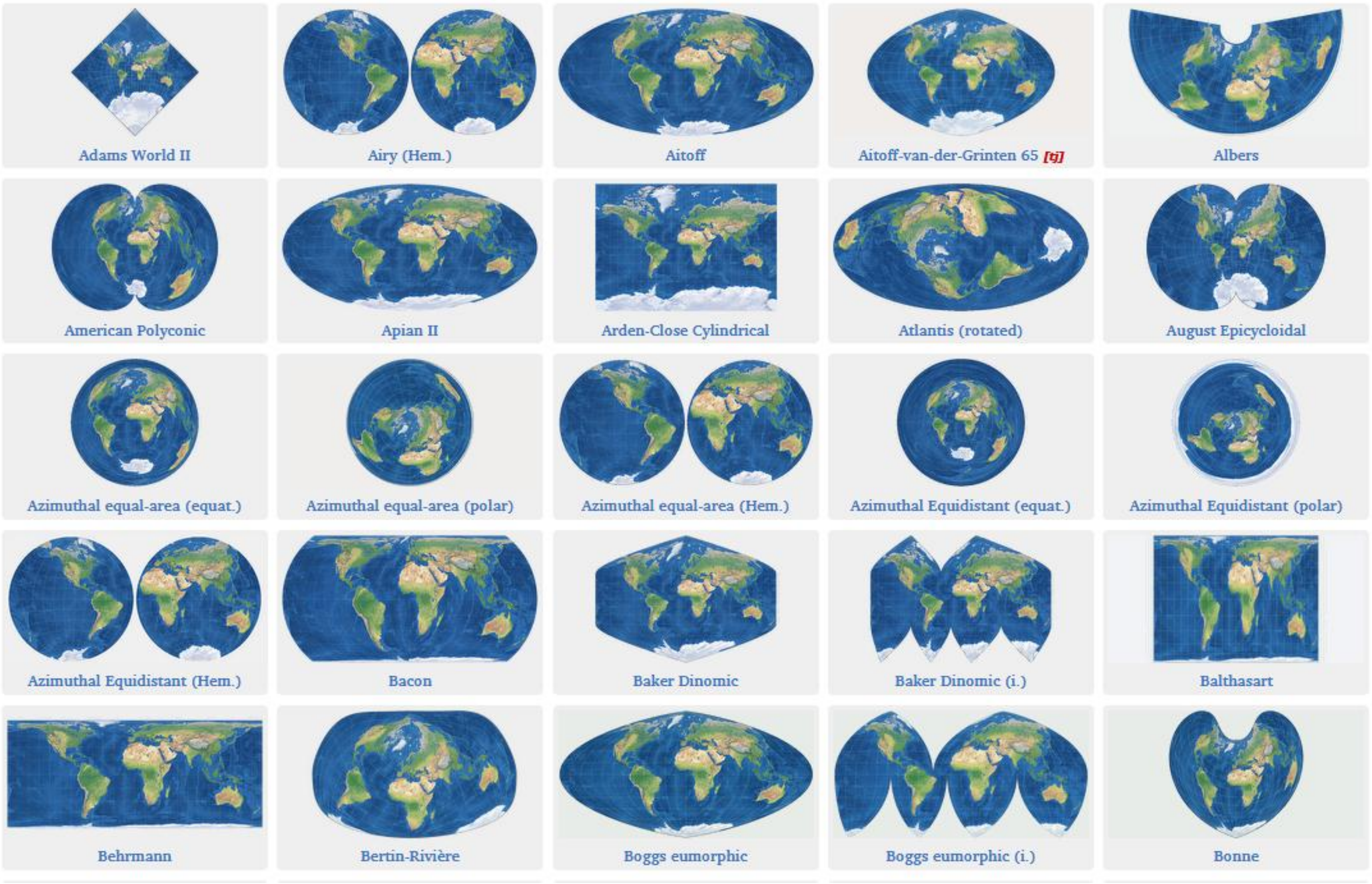


Coordinate Reference Systems

- *Why oval?*



Coordinate Reference Systems



https://www.nceas.ucsb.edu/sites/default/files/2020-04/OverviewCoordinateReferenceSystems.pdf

ENVIRONMENTAL DATA ANALYTICS: M9 – SPATIAL ANALYSIS

ENV 872

Nicholas School of the Environment - Duke University

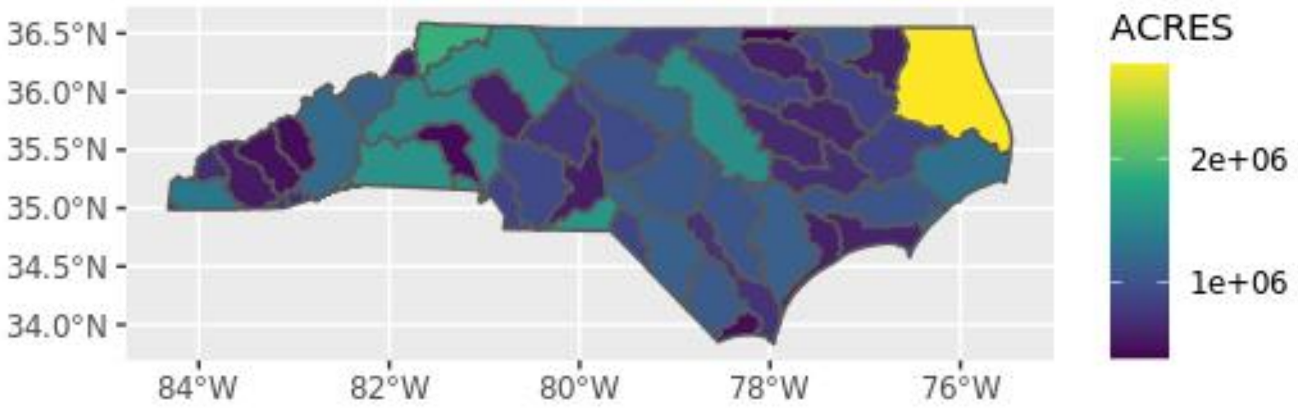
Part 2 – Spatial Analysis

Reading spatial data into R	st_read()
Attribute joins	left_join(), inner_join(),…
Spatial aggregation	group_by() & summarize()
Coordinate system transformations	st_transform()
Intersecting data	st_filter(…,.pred)
Clipping data	st_intersection()
Spatial Selection	st_intersects()

Exercise 2.2.2

- □ Read in the NC HUC-8 shapefile & filter for : `./Data/Spatial/NCHUC8.shp`

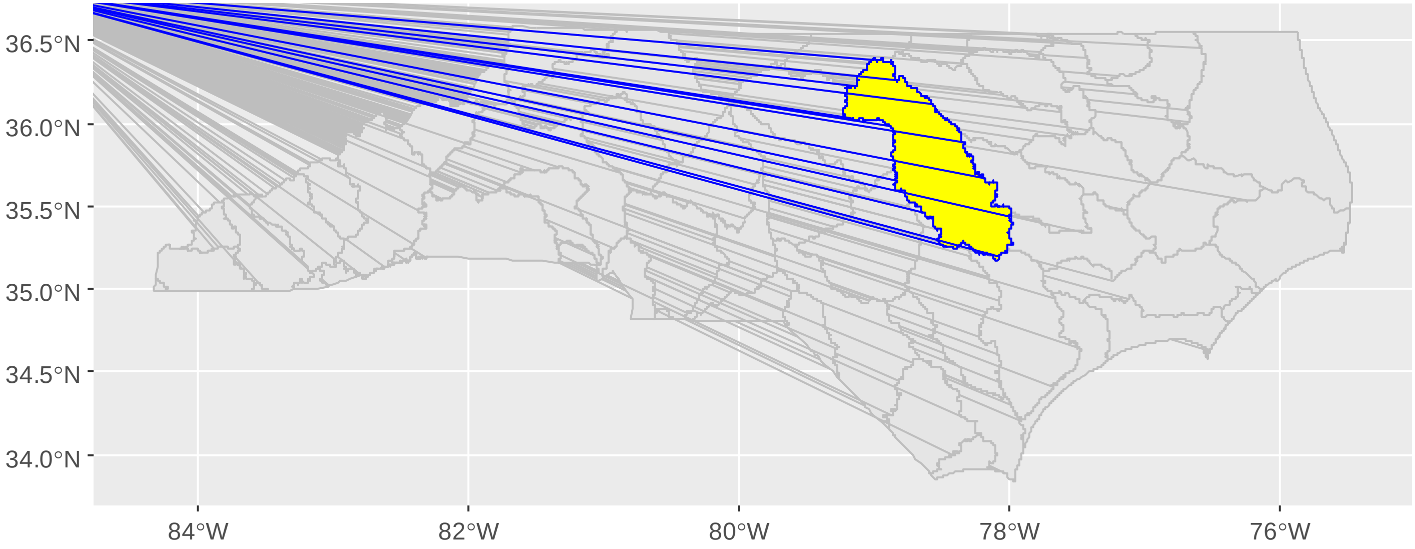
- □ View features, colored by ACRES



Exercise 2.2.3 - Challenge

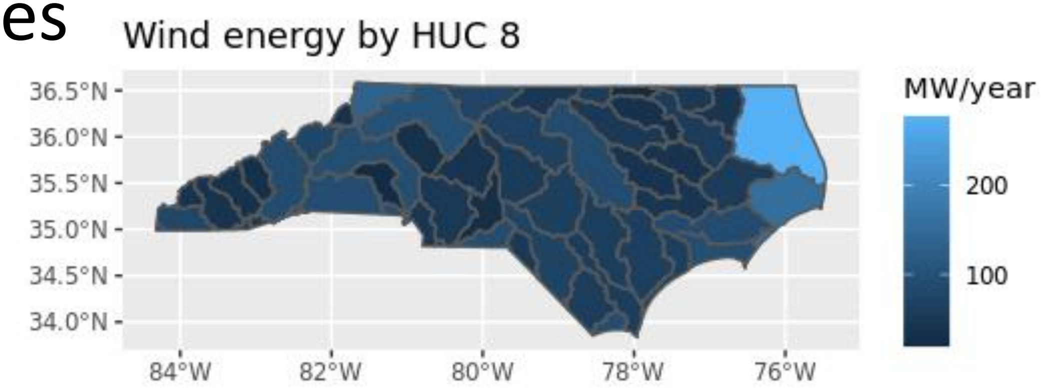
- □ Read in the NC HUC-8 shapefile & *filter for SUBBASIN is "Upper Neuse"*

- □ View all HUCs in gray, Upper Neuse in yellow w/blue border



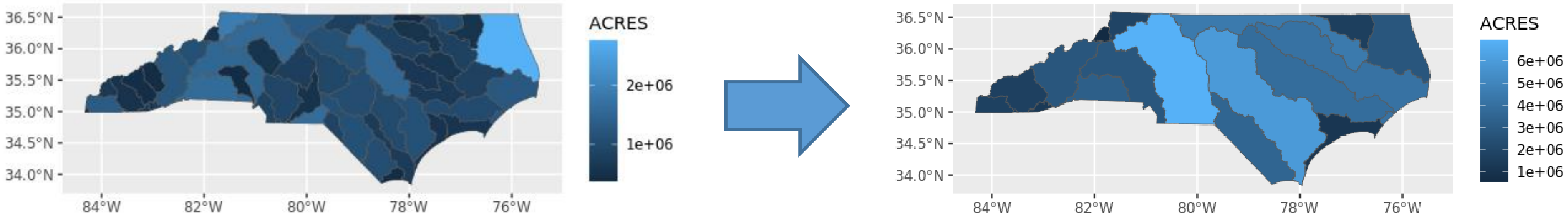
Exercise 3.1.1

- Read an online CSV file into a dataframe\
  https://raw.githubusercontent.com/ENV859/EnviroAtlasData/main/Wind_Energy.csv
  - Set `HUC12` column to be a factor (*colClasses*)
  - Compute `HUC8` from `HUC12` (*substr*)
  - Group on `HUC8`
  - Compute sum of AvgWindEnergy for each HUC8
- Join to HUC8 features



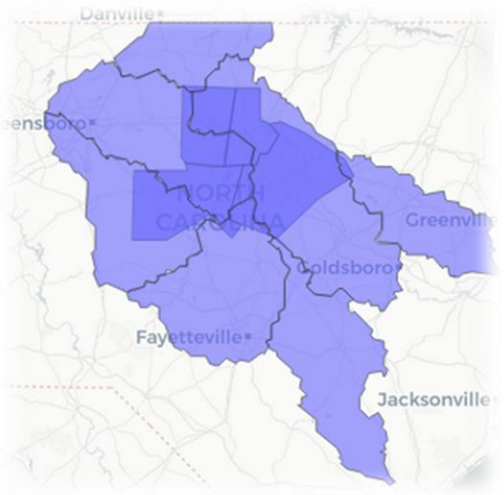
Exercise 3.1.2

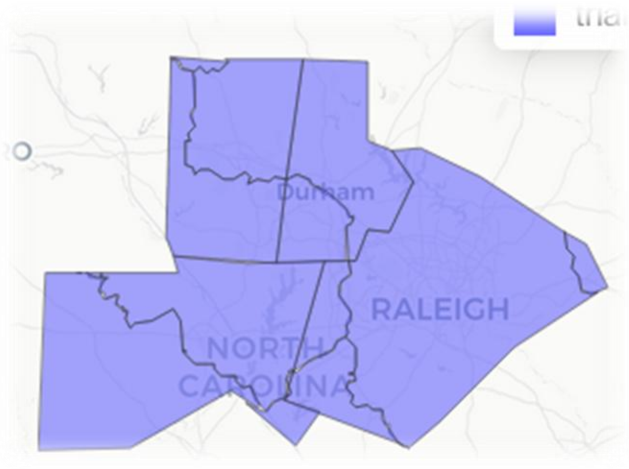
- Aggregate HUC 8 to HUC 6 (“DWQ Sub-basin”)
  - Compute aggregate area in acres and square miles



Exercise 3.3.1

- Select Triangle counties from all counties: *Chatham, Durham, Orange,* and *Wake*
- Select HUC8s that intersect the Triangle counties
- Intersect (clip) the HUC8 areas falling w/in Triangle





A NOTE ON KNITTING…

Maps created with MapView and Leaflet don’t knit well to PDF!

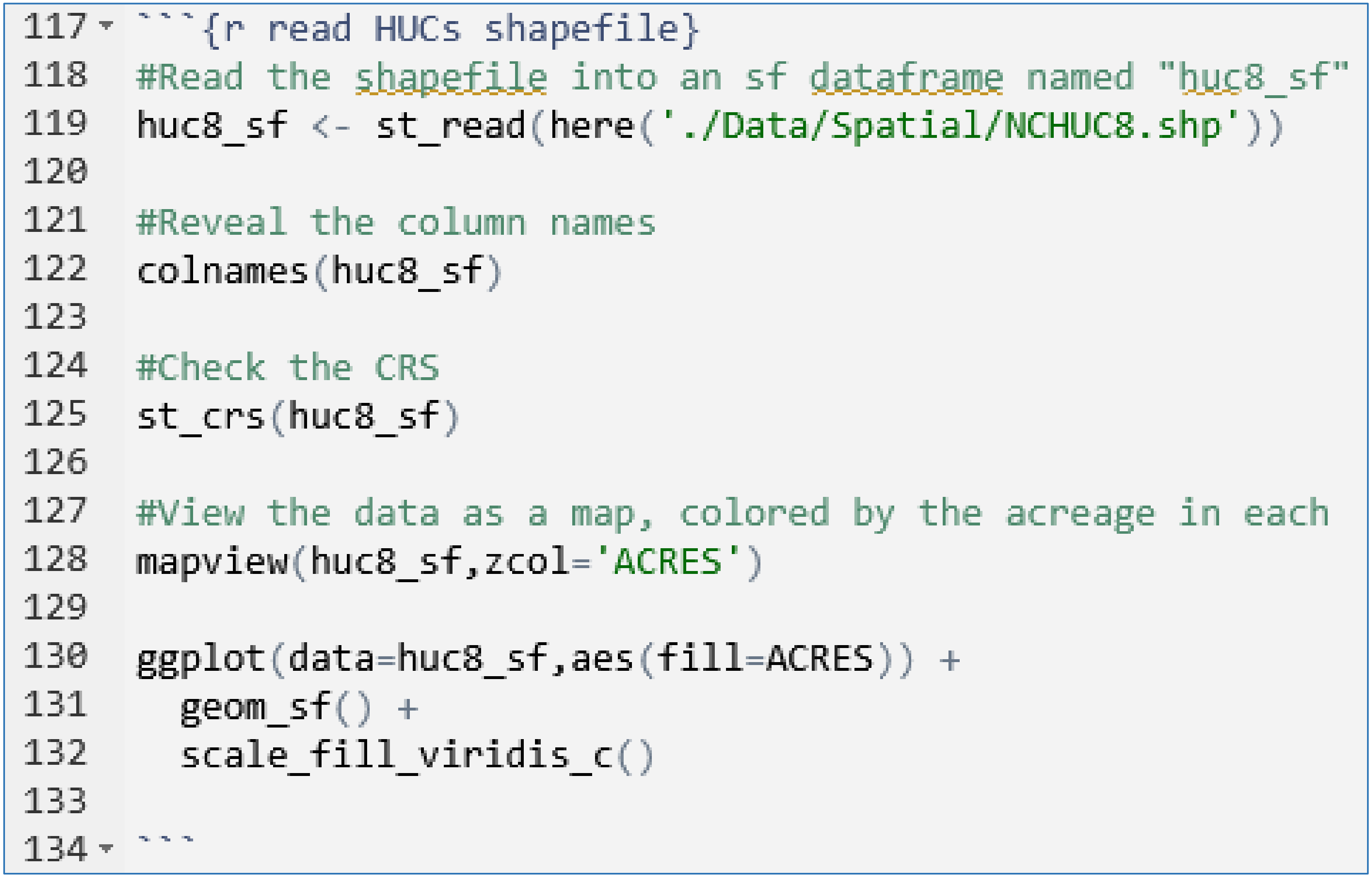
- **Solutions**:
  - Use ggplot…
  - Knit to HTML…

M9.3 – Spatial Data Visualization

- ggplot() + geom_sf()
- mapview
- leaflet

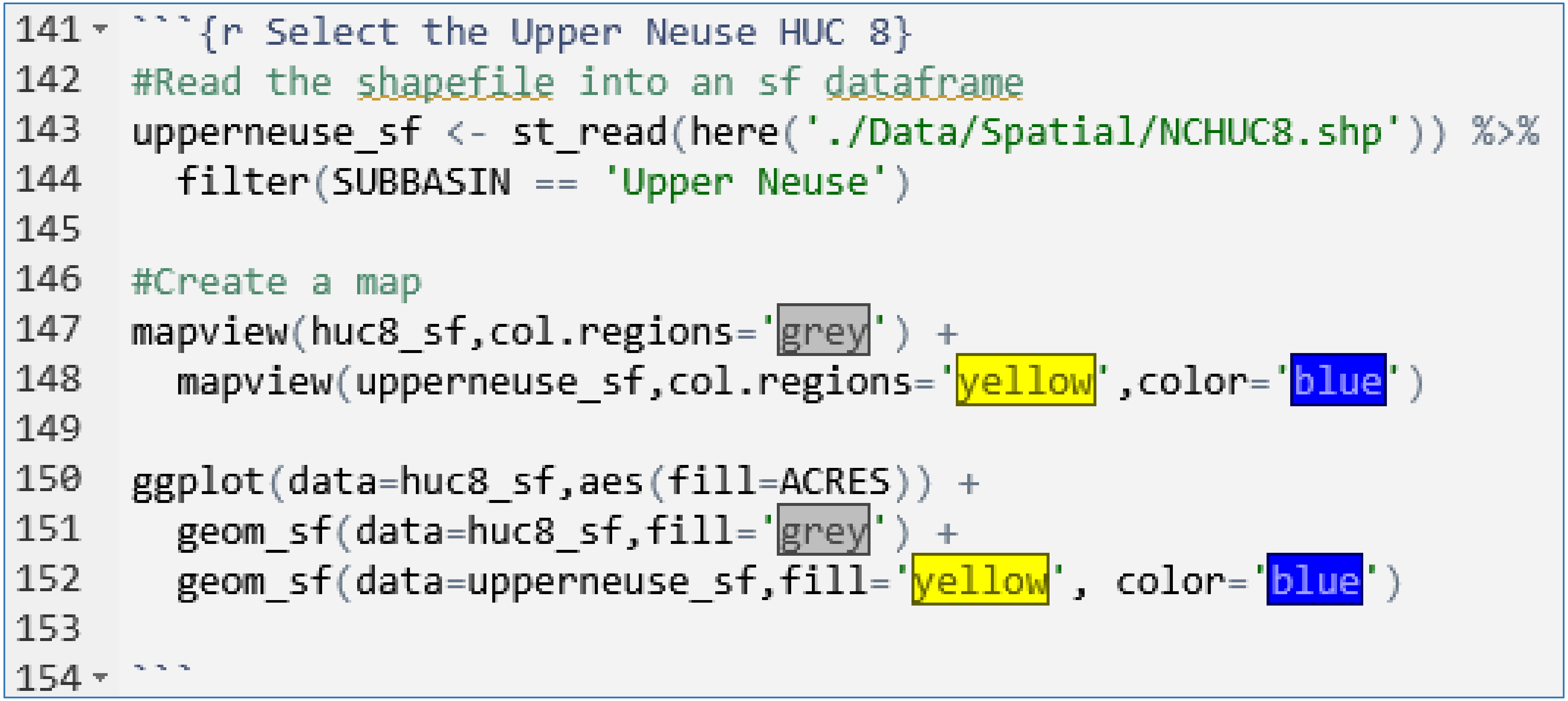
Solutions…

2.2.2 Read HUCs shapefile



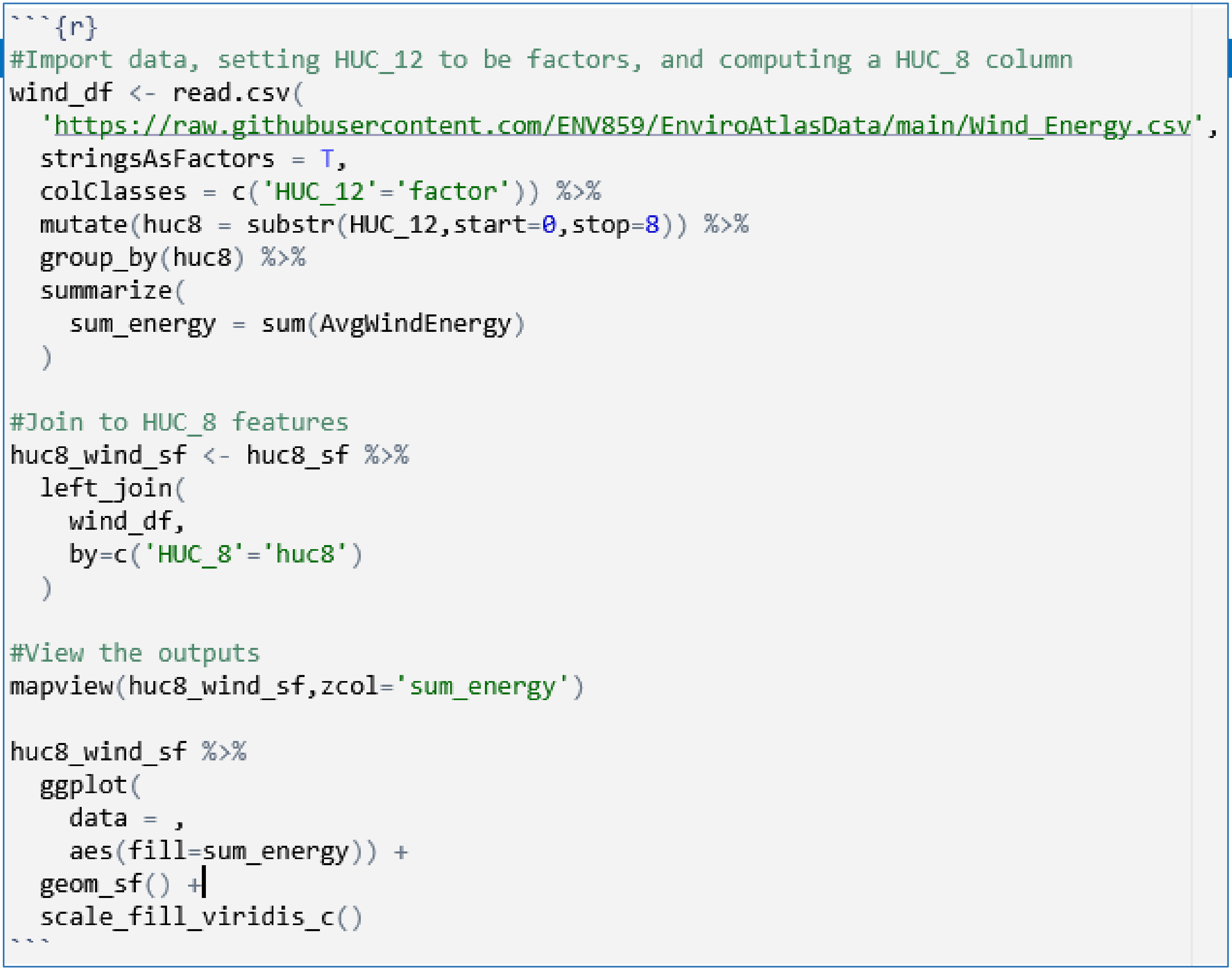
~~~
```{r read HUCs shapefile}
#Read the shapefile into an sf dataframe named "huc8_sf"
huc8_sf <- st_read(here('./Data/Spatial/NCHUC8.shp'))

#Reveal the column names
colnames(huc8_sf)

#Check the CRS
st_crs(huc8_sf)

#View the data as a map, colored by the acreage in each
mapview(huc8_sf,zcol='ACRES')

ggplot(data=huc8_sf,aes(fill=ACRES)) +
  geom_sf() +
  scale_fill_viridis_c()

```
~~~

2.2.3 Challenge



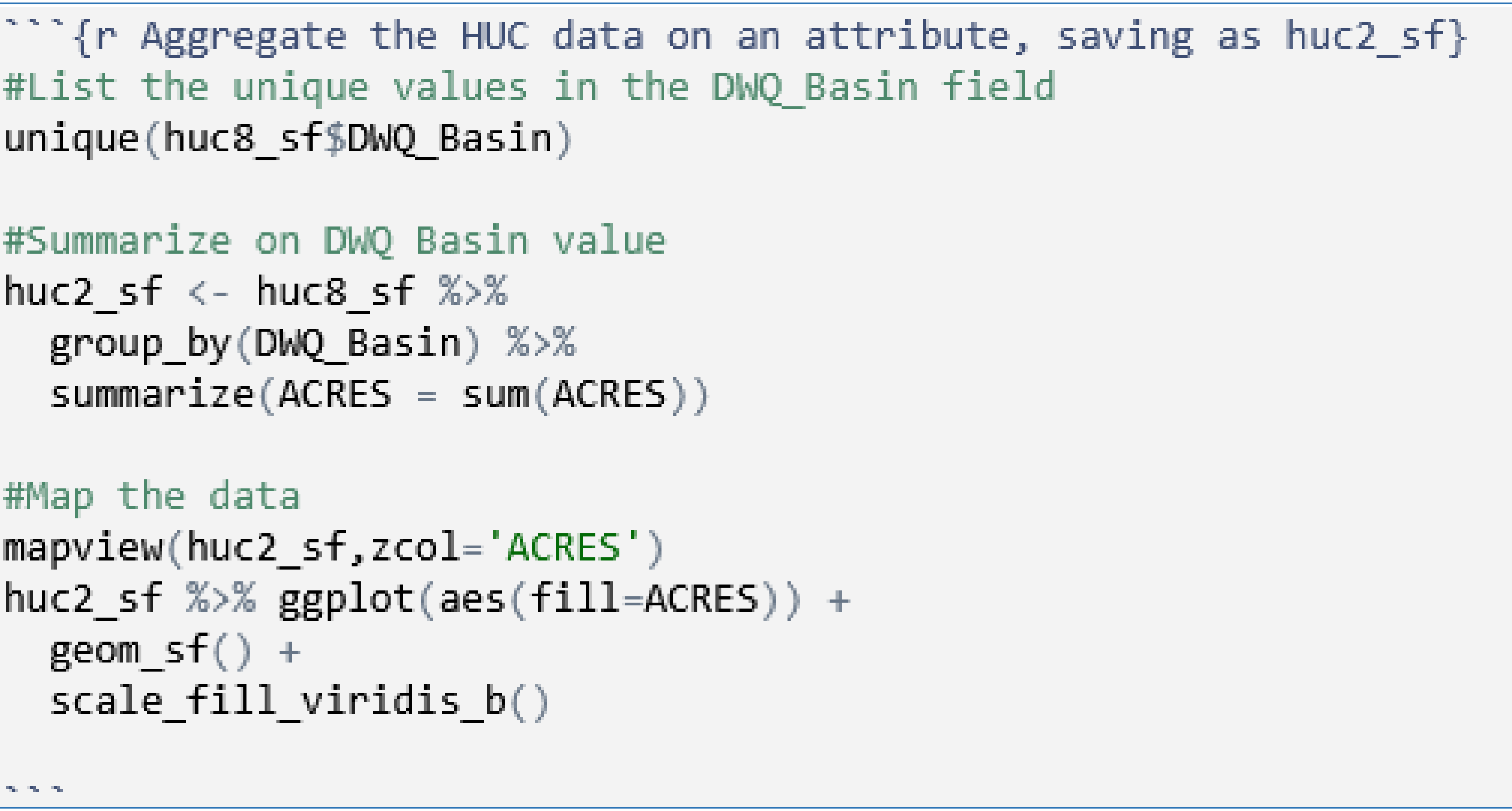
~~~
```{r Select the Upper Neuse HUC 8}
#Read the shapefile into an sf dataframe
upperneuse_sf <- st_read(here('./Data/Spatial/NCHUC8.shp')) %>%
  filter(SUBBASIN == 'Upper Neuse')

#Create a map
mapview(huc8_sf,col.regions='grey') +
  mapview(upperneuse_sf,col.regions='yellow',color='blue')

ggplot(data=huc8_sf,aes(fill=ACRES)) +
  geom_sf(data=huc8_sf,fill='grey') +
  geom_sf(data=upperneuse_sf,fill='yellow', color='blue')

```
~~~

3.1 Join wind data to HUC8s



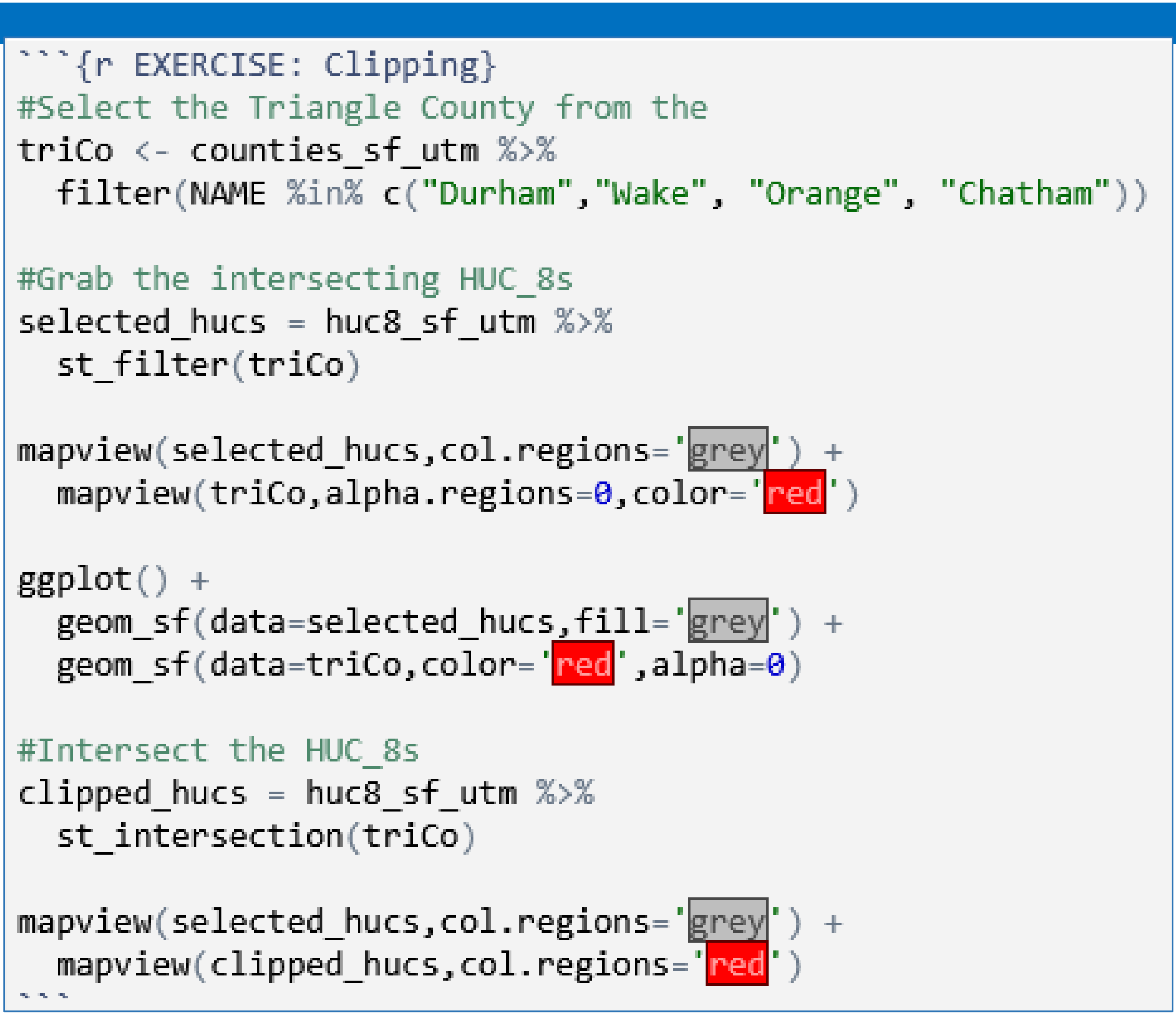
~~~
```{r}
#Import data, setting HUC_12 to be factors, and computing a HUC_8 column
wind_df <- read.csv(
  'https://raw.githubusercontent.com/ENV859/EnviroAtlasData/main/Wind_Energy.csv',
  stringsAsFactors = T,
  colClasses = c('HUC_12'='factor')) %>%
  mutate(huc8 = substr(HUC_12,start=0,stop=8)) %>%
  group_by(huc8) %>%
  summarize(
    sum_energy = sum(AvgWindEnergy)
  )

#Join to HUC_8 features
huc8_wind_sf <- huc8_sf %>%
  left_join(
    wind_df,
    by=c('HUC_8'='huc8')
  )

#View the outputs
mapview(huc8_wind_sf,zcol='sum_energy')

huc8_wind_sf %>%
  ggplot(
    data = ,
    aes(fill=sum_energy)) +
  geom_sf() +
  scale_fill_viridis_c()
```
~~~

3.1 Aggregate HUC-8s to HUC-2s



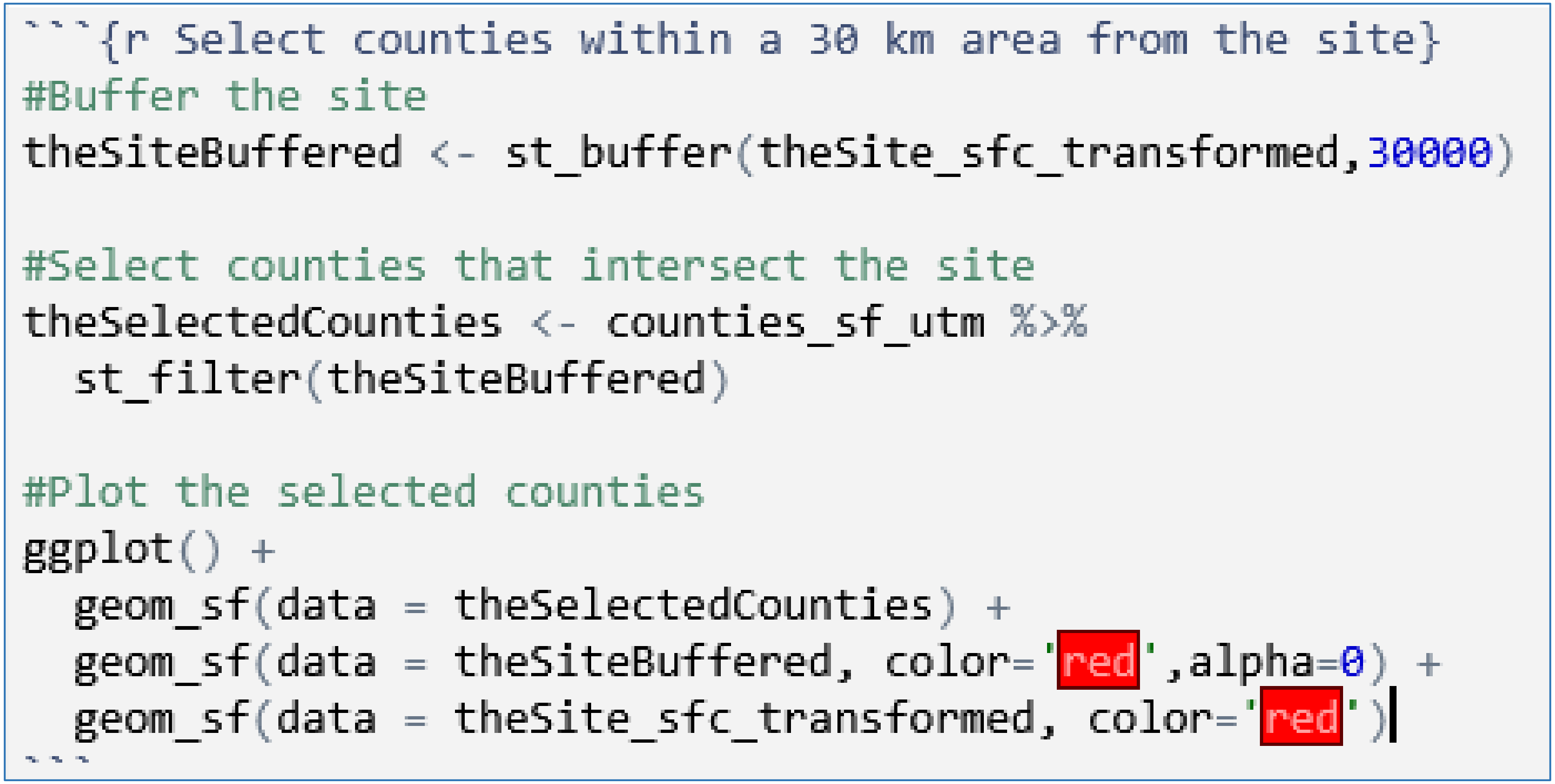
~~~
```{r Aggregate the HUC data on an attribute, saving as huc2_sf}
#List the unique values in the DWQ_Basin field
unique(huc8_sf$DWQ_Basin)

#Summarize on DWQ Basin value
huc2_sf <- huc8_sf %>%
  group_by(DWQ_Basin) %>%
  summarize(ACRES = sum(ACRES))

#Map the data
mapview(huc2_sf,zcol='ACRES')
huc2_sf %>% ggplot(aes(fill=ACRES)) +
  geom_sf() +
  scale_fill_viridis_b()

```
~~~

3.3 Select HUC8s in the Triangle



~~~
```{r EXERCISE: Clipping}
#Select the Triangle County from the
triCo <- counties_sf_utm %>%
  filter(NAME %in% c("Durham","Wake", "Orange", "Chatham"))

#Grab the intersecting HUC_8s
selected_hucs = huc8_sf_utm %>%
  st_filter(triCo)

mapview(selected_hucs,col.regions='grey') +
  mapview(triCo,alpha.regions=0,color='red')

ggplot() +
  geom_sf(data=selected_hucs,fill='grey') +
  geom_sf(data=triCo,color='red',alpha=0)

#Intersect the HUC_8s
clipped_hucs = huc8_sf_utm %>%
  st_intersection(triCo)

mapview(selected_hucs,col.regions='grey') +
  mapview(clipped_hucs,col.regions='red')
```
~~~

3.4 Select counties w/in 30 km of pt



~~~
```{r Select counties within a 30 km area from the site}
#Buffer the site
theSiteBuffered <- st_buffer(theSite_sfc_transformed,30000)

#Select counties that intersect the site
theSelectedCounties <- counties_sf_utm %>%
  st_filter(theSiteBuffered)

#Plot the selected counties
ggplot() +
  geom_sf(data = theSelectedCounties) +
  geom_sf(data = theSiteBuffered, color='red',alpha=0) +
  geom_sf(data = theSite_sfc_transformed, color='red')
```
~~~