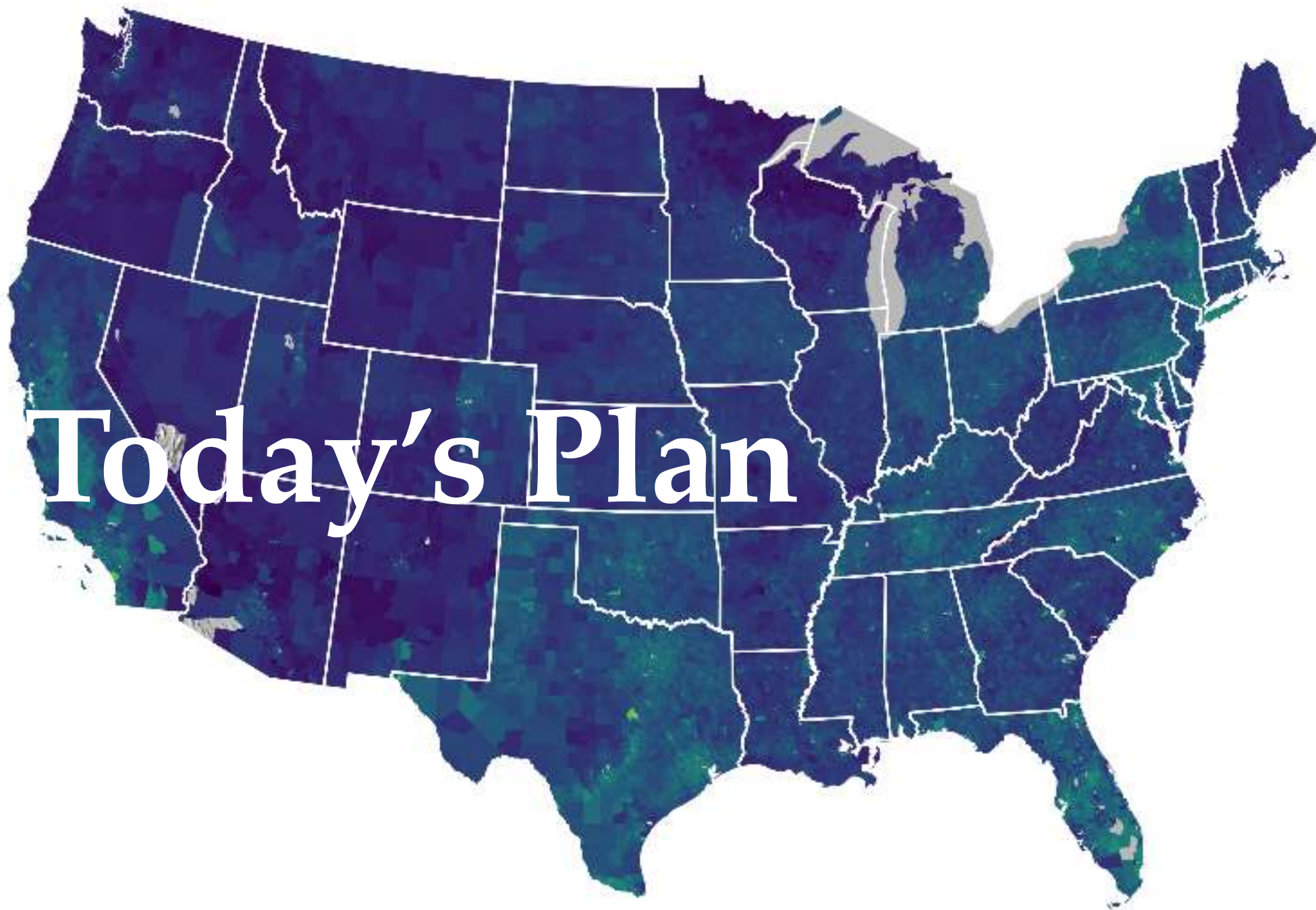# Integrating Rasters and Vector Data

HES 505 Fall 2023: Session 16

Matt Williamson

Today's Plan

Objectives

- Use `dplyr` with `predicates` and `measures` to subset and manipulate data

- Use `extract` to access raster data

- Use `zonal` to summarize access data

- Join data into a single analyzable dataframe

Motivating Question

How do Collaborative Forest Landscape Restoration projects compare to other National Forest lands with respect to social and wildfire risks?

Thinking about the data

- **Datasets** - Forest Service Boundaries, CFLRP Boundaries, Wildfire Risk Raster, CEJST shapefile

- **Dependent Variable** - CFLRP (T or F)

- **Independent Variables** - Wildfire hazard, income, education, housing burdent

Building some Pseudocode

```
1  1. Load Libraries
2  2. Load data
3  3. Check validity and alignment
4  4. Subset to relevant geographies
5  5. Select relevant attributes
6  6. Extract wildfire risk
7  7. CFLRP T or F
```

Load libraries

```r
1  library(sf)
2  library(terra)
3  library(tidyverse)
4  library(tmap)
```

Load the data

- Downloading USFS data using tempfiles and `unzip`

```r
1   ### FS Boundaries
2   tmp <- tempfile()
3   fs.url <- "https://data.fs.usda.gov/geodata/edw/edw_resources/shp/S_USA.Adm
4   download.file(fs.url, tmp)
5   tmp2 <- tempfile()
6   unzip(zipfile=tmp, exdir = tmp2 )
7
8   fs.bdry <- read_sf(tmp2)
9
10  ### CFLRP Data
11  tmp <- tempfile()
12  cflrp.url <- "https://data.fs.usda.gov/geodata/edw/edw_resources/shp/S_USA.
13  download.file(cflrp.url, tmp)
14  tmp2 <- tempfile()
15  unzip(zipfile=tmp, exdir = tmp2 )
16  cflrp.bdry <- read_sf(tmp2)
```

Load the data

- From our class folder

```
1  ### Wildfire Hazard Data
2  wildfire.haz <- rast("opt/data/2023/assignment01/wildfire_hazard_agg.tif")
3
4  ## CEJST data
5  cejst <- read_sf("opt/data/2023/assignment01/cejst_pnw.shp")
```

Check Validity

- The USFS datasets are new; let's check the geometries

```r
1  all(st_is_valid(fs.bdry))
```

`[1] FALSE`

```r
1  all(st_is_valid(cflrp.bdry))
```

`[1] FALSE`

- Make them valid

```r
1  fs.bdry.valid <- st_make_valid(fs.bdry)
2  all(st_is_valid(fs.bdry.valid))
```

`[1] TRUE`

```r
1  cflrp.bdry.valid <- st_make_valid(cflrp.bdry)
2  all(st_is_valid(cflrp.bdry.valid))
```

`[1] TRUE`

Set Projection

- We know these are in different CRS

- Project to the CRS of the `raster`

- Using `%>%` to pipe data through the function

```
1  cflrp.proj <- cflrp.bdry.valid %>% st_transform(., crs=crs(wildfire.haz))
2  cejst.proj <- cejst %>% st_transform(., crs=crs(wildfire.haz))
3  fs.proj <- fs.bdry.valid %>% st_transform(., crs=crs(wildfire.haz))
```
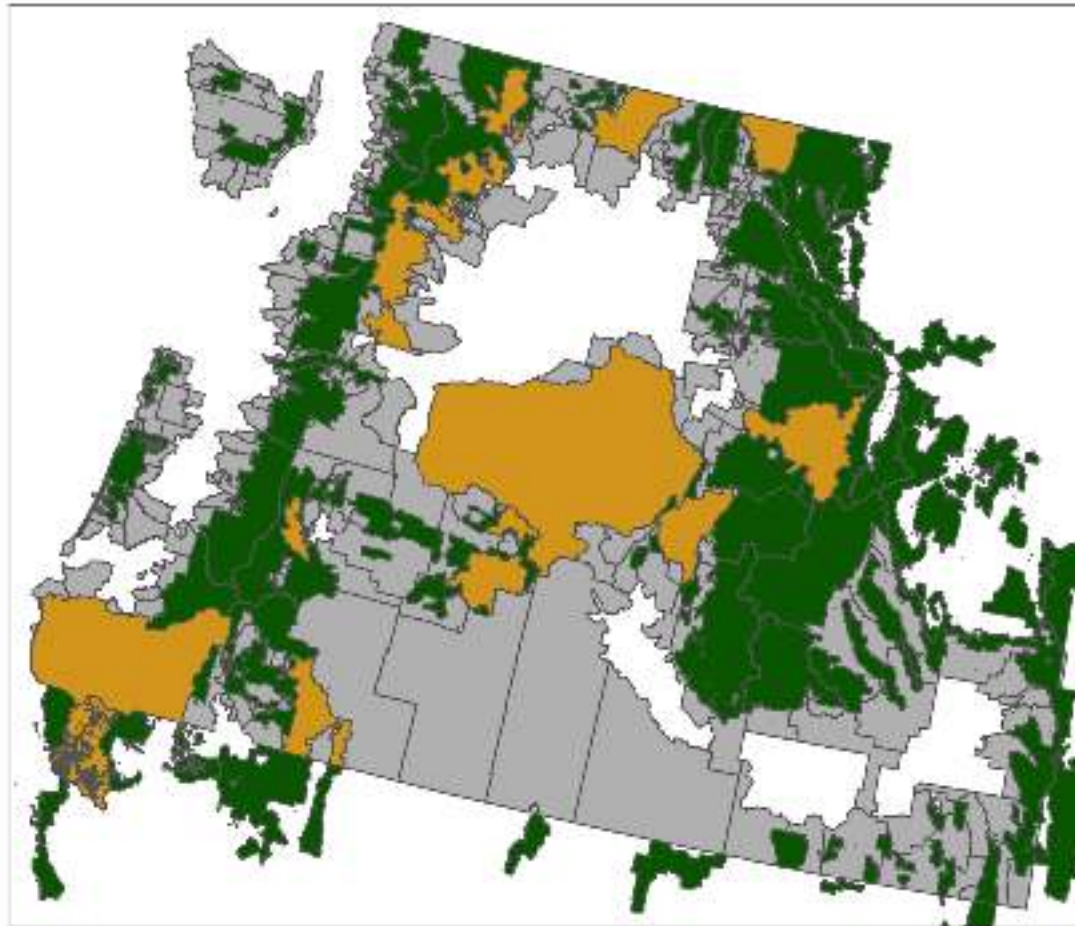
Subset Geometries

- We can use the [] notation to subset the one dataset based on the geometry of the other

- Need USFS and CFLRP within the region

- Then need tracts that overlap USFS

```
1  fs.subset <- fs.proj[cejst.proj, ]
2  cflrp.subset <- cflrp.proj[cejst.proj, ]
3  cejst.subset <- cejst.proj[fs.subset, ]
```

Subset geometries

```r
1  sub.map <- tm_shape(cejst.subset) +
2    tm_polygons(col="gray") +
3    tm_shape(fs.subset) +
4    tm_polygons(col="darkgreen") +
5    tm_shape(cflrp.subset) +
6    tm_polygons(col="goldenrod")
```

Select Relevant Columns

- Use the codebook to identify the right columns

- Then use `select` from `dplyr`

- geometries are sticky!

```
1  cejst.df <- cejst.subset %>%
2    select(GEOID10, LMI_PFS, LHE, HBF_PFS)
3  head(cejst.df)
```

```
Simple feature collection with 6 features and 4 fields
Geometry type: MULTIPOLYGON
Dimension:      XY
Bounding box:  xmin: -1598729 ymin: 2388182 xmax: -1475201 ymax: 3000813
Projected CRS: unnamed
# A tibble: 6 × 5
  GEOID10      LMI_PFS    LHE HBF_PFS
geometry
  <chr>          <dbl> <int>   <dbl>                              <MULTIPOLYGON
[m]>
1 16025970100    0.75      0     0.6  (((-1485848 2427049, -1485813 2426977,
```

–148…
2 16057005500   0.43   0   0.44 (((–1567845 2843218, –1567803 2843209,
–156…
3 16057005600   0.3   0   0.05 (((–1573408 2823058, –1573412 2823071,

Extract wildfire data

- Can use `zonal` for one summary statistic

- Or `extract` for multiple

```r
1  wildfire.zones <- terra::zonal(wildfire.haz, vect(cejst.df), fun="mean", na
2
3  head(wildfire.zones)
```

```
     WHP_ID
1 2997.7951
2  182.8864
3  386.9580
4  173.1703
5  193.4199
6  210.4406
```