Geospatial for Scott

Matthew J Berens

2023-08-25

Introduction and Data Access

The boundary of the EFPC watershed is accessed by the `get_huc()` function in the `nhdplusTools` package. The remaining shapefiles were accessed externally through the NHDPlus dataset and manually processed in QGIS. At this point, NHDPlus data for the EFPC watershed are not available through the `nhdplusTools` package.

```
EFPC_HUC <- st_set_crs(get_huc(id = "060102070302", buffer = 0.5,
                              type = "huc12"), 4326)
EFPC_catchments <- st_set_crs(read_sf("GIS_data/EFPC Catchments.shp"), 4326)
EFPC_fl <- st_set_crs(read_sf("GIS_data/EFPC Flowlines.shp"), 4326)
EFPC_main <- st_set_crs(read_sf("GIS_data/East Fork Poplar Creek.shp"), 4326)
```

Land use data can be directly obtained from the National Land Classification Database (NLCD) through the `get_nlcd()` function in the `FedData` package. The data are available for select years from 2001-2019. The year can be selected by changing the `year = ####` option in the `get_nlcd()` function.

```
nlcd_2019 <- FedData::get_nlcd(template = EFPC_HUC, label = "EFK NLCD 2019",
                              year = 2019)
```

This chunk generates the correct color scheme for the NLCD data.

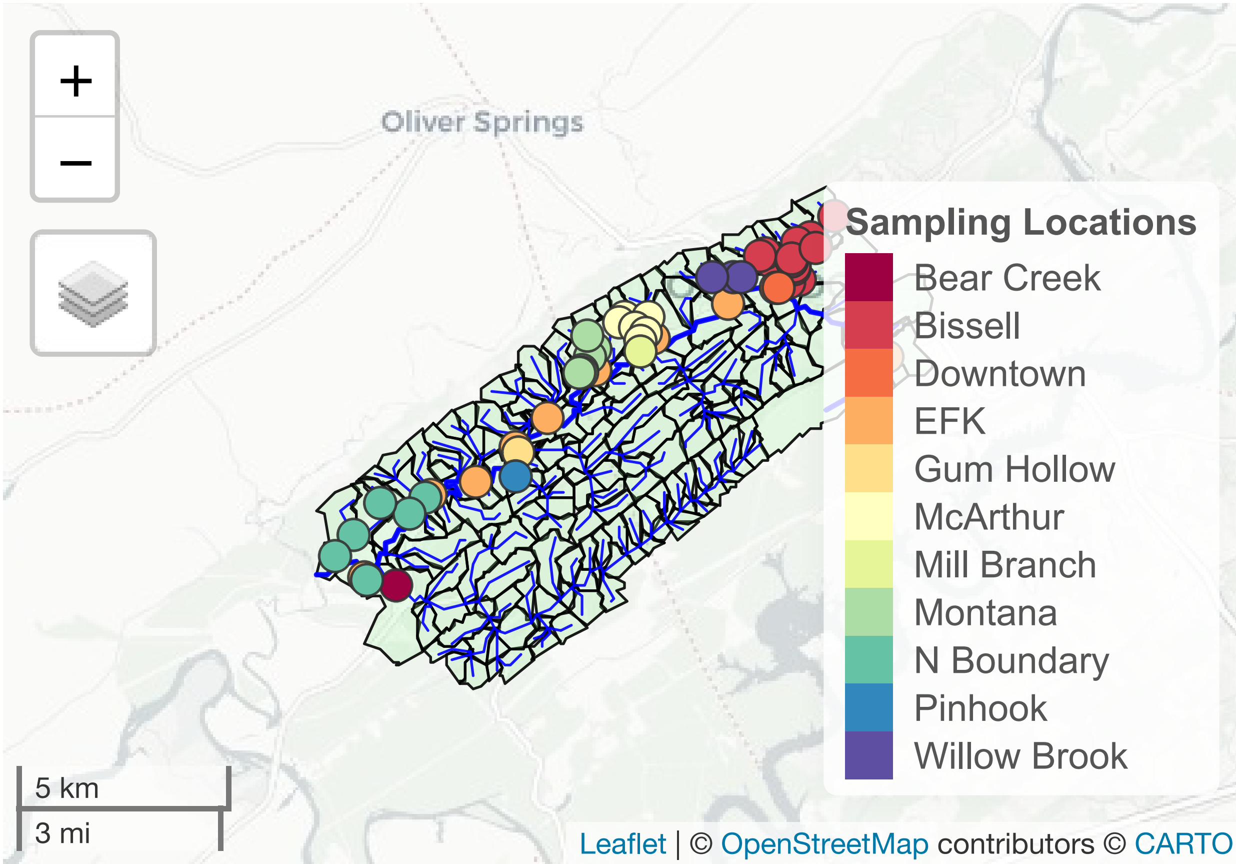
The WaDE SFA sampling locations are stored in a csv in the GitHub directory. The points must be imported and converted to a spatial object with the `st_as_sf()` function. Ensure that the CRS is set to 4326.

```
WaDE_sites <- read.csv("raw/WaDE SYNOPTIC_SITES.csv") %>%
   st_as_sf(coords = c('longitude', 'latitude')) %>%
    st_set_crs(4326)
```

Overview of Study Area

Here is a simple map that uses the `mapview` function to visualize the study area for the WaDE SFA. I colorized the sampling locations based on their tributary subcatchment. The NLCD data can be toggled on/off from the legend. Each of the shapefiles for the catchments, flowlines, etc. are added as separate mapview objects.

```
mapview(WaDE_sites, zcol = "network", layer.name = "Sampling Locations",
        alpha.region = 1, label = "site_name", col.regions =   RColorBrewer::brewer.pal(length(unique(Wa
mapview(EFPC_catchments, alpha.regions = 0.2, col.region = "lightgreen",
        color = "black", legend = FALSE, lwd=1, layer.name = "EFK Catchments") +
mapview(EFPC_fl, color = "blue", legend = FALSE, lwd = 1,
        layer.name = "NHD Flowlines") +
mapview(EFPC_main, color = "blue", legend = FALSE, lwd = 2,
        layer.name = "EFK Main Branch") +
mapview(nlcd_2019, col.regions = data, att = "Class",
        layer.name = "Landcover Class", label = FALSE, legend = TRUE,
        alpha = 0.5, hide = TRUE)
```



This function overlaps the WaDE sampling locations with the catchment boundary shapefile to determine which catchment each sampling location is in. The `WaDE_sites` data frame is then updataed to include the catchment data

```
WaDE_sites <- st_intersection(WaDE_sites, EFPC_catchments)
```

Exploring Land Use in the EFPC Watershed

This chunk calculates the catchment-specific land use percentages from the NLCD and then bins them into larger categorical groupings (e.g., developed, water, forested). I save them as geopackages so they can be easily accessed later. Sometimes saving things as .shp files in R messes up the names of the data frames.

```
#This code calculates the zonal statistics for each of the EFPC catchments.
#The results are exported as a df that is saved as landuse_fracs

landuse_fracs <- exact_extract(nlcd_2019, EFPC_catchments, fun = 'frac',
                               force_df = TRUE)

#Assign the catchments their correct OBJECTID
landuse_fracs$OBJECTID <- EFPC_catchments$OBJECTID

EFPC_catchments_landuse <- merge(EFPC_catchments, landuse_fracs,
                                 by = "OBJECTID") %>%
                                dplyr::select(-c(NHDPlusID, SourceFC,
                                GridCode, VPUID, SHAPE_Leng, SHAPE_Area)) %>%
  mutate(Developed = frac_21 + frac_22 + frac_23 + frac_24,
         Water = frac_11,
```

```r
         Forest = frac_41 + frac_42 + frac_43,
         Shrub = frac_52,
         Grassland = frac_71,
         Pasture = frac_81,
         Wetland = frac_90 + frac_95,
         Herbaceous = frac_52 + frac_71 + frac_81,
         OBJECTID = as.character(OBJECTID))


#Merge the catchment land use percentages with the sampling site locations
WaDE_sites_landuse <- st_intersection(EFPC_catchments_landuse, WaDE_sites) %>%
  dplyr::select(-c(AreaSqKm.1))

#Export combined land use calculations with sampling locations
st_write(EFPC_catchments_landuse, "GIS_data/EFPC_catchments_landuse.gpkg",
         append = FALSE)

st_write(WaDE_sites_landuse, "GIS_data/WaDE_sites_landuse.gpkg", append = FALSE)
```

Next, I manually grouped the subcatchments together into one larger catchment that encompasses and entire upstream portion of the tributary network for each site

```r
catchment_bins <- rbind(
data.frame("site_name" = "NB01", "OBJECTID" = "89458"),
data.frame("site_name" = "NB02", "OBJECTID" = c("238691","100959")),
data.frame("site_name" = "NB03", "OBJECTID" = "100959"),
data.frame("site_name" = "NB04", "OBJECTID" = "102476"),
data.frame("site_name" = "NB05", "OBJECTID" = c("231934","242844",
                                                "98890","99751","99669")),
data.frame("site_name" = "NB06", "OBJECTID" = "101341"),
data.frame("site_name" = "MTN4", "OBJECTID" = "90316"),
data.frame("site_name" = "MTN3", "OBJECTID" = c("90316","257888","90620")),
data.frame("site_name" = "MTN2", "OBJECTID" = c("90316","257888","90620")),
data.frame("site_name" = "MTN1.3", "OBJECTID" = "93594"),
data.frame("site_name" = "MTN1.2", "OBJECTID" = c("90316","257888","90620")),
data.frame("site_name" = "MTN1.1", "OBJECTID" = c("245543","93594","90316",
                                                  "257888","90620")),
data.frame("site_name" = "MCA4", "OBJECTID" = c("238228","91132","90546")),
data.frame("site_name" = "MCA3.3", "OBJECTID" = c("89979")),
data.frame("site_name" = "MCA3.2", "OBJECTID" = c("91501")),
data.frame("site_name" = "MCA3.1", "OBJECTID" = c("231863", "91501", "89979")),
data.frame("site_name" = "MCA2" , "OBJECTID" = c("238322","238228","91132",
                                                 "90546","231863", "91501",
                                                 "89979")),
data.frame("site_name" = "MCA1" , "OBJECTID" = c("238322","238228","91132",
                                                 "90546","231863", "91501",
                                                 "89979","90467", "90031")),
data.frame("site_name" = "WBK1" , "OBJECTID" = c("245547","94989")),
data.frame("site_name" = "WBK2" , "OBJECTID" = c("245547","94989")),
data.frame("site_name" = "WBK3" , "OBJECTID" = c("93685")),
data.frame("site_name" = "BSL8" , "OBJECTID" = c("81824")),
data.frame("site_name" = "BSL7" , "OBJECTID" = c("73872")),
data.frame("site_name" = "BSL6" , "OBJECTID" = c("85970")),
data.frame("site_name" = "BSL5.3" , "OBJECTID" = c("258680")),
```

```r
data.frame("site_name" = "BSL5.2" , "OBJECTID" = c("72273")),
data.frame("site_name" = "BSL5.1" , "OBJECTID" = c("238341","258680","72273")),
data.frame("site_name" = "BSL4.3" , "OBJECTID" = c("73872")),
data.frame("site_name" = "BSL4.2" , "OBJECTID" = c("81824")),
data.frame("site_name" = "BSL4.1" , "OBJECTID" = c("245291","81824","73872")),
data.frame("site_name" = "BSL3.3" , "OBJECTID" = c("238341","258680","72273")),
data.frame("site_name" = "BSL3.2" , "OBJECTID" = c("85970")),
data.frame("site_name" = "BSL3.1" , "OBJECTID" = c("85970","238341","258680",
                                                   "72273")),
data.frame("site_name" = "BSL2.3" , "OBJECTID" = c("85970","238341","258680",
                                                   "72273")),
data.frame("site_name" = "BSL2.2" , "OBJECTID" = c("245291","81824","73872")),
data.frame("site_name" = "BSL2.1" , "OBJECTID" = c("85970","238341","258680",
                                                   "72273",
                                                   "245291","81824","73872")),
data.frame("site_name" = "BSL1.3" , "OBJECTID" = c("85970","238341","258680",
                                                   "72273",
                                                   "245291","81824","73872")),
data.frame("site_name" = "BSL1.2" , "OBJECTID" = c("241710")),
data.frame("site_name" = "BSL1.1" , "OBJECTID" = c("85970","238341","258680","72273",
                                                   "245291","81824","73872","95129",
                                                   "241710"))
)

catchments_binned <- merge(EFPC_catchments, catchment_bins, by = "OBJECTID") %>%
    group_by(site_name) %>%
    summarise(geometry = sf::st_union(geometry)) %>%
    ungroup() %>%
    st_as_sf() %>%
    mutate(AreaSqKm = st_area(geometry)/1000000)

#This code calculates the zonal statistics for each of the EFPC catchments.
#The results are exported as a df that is saved as landuse_bins
landuse_bins <- exact_extract(nlcd_2019, catchments_binned, fun = 'frac',
                              force_df = TRUE)

landuse_bins$site_name = catchments_binned$site_name

EFPC_catchments_binned <- merge(catchments_binned, landuse_bins,
                                by = "site_name") %>%

  mutate(Developed = frac_21 + frac_22 + frac_23 + frac_24,
         Water = frac_11,
         Forest = frac_41 + frac_42 + frac_43,
         Shrub = frac_52,
         Grassland = frac_71,
         Pasture = frac_81,
         Wetland = frac_90 + frac_95,
         Herbaceous = frac_52 + frac_71 + frac_81)

#Export combined land use calculations with sampling locations
st_write(EFPC_catchments_binned, "GIS_data/EFPC_LU_catchments_binned.gpkg",
         append = FALSE)
```