12: Mapping

Water Data Analytics | Kateri Salk

Spring 2022

Lesson Objectives

1. Delineate watersheds and flowlines using nhdplusTools
2. Extract spatial information from delineated watersheds
3. Create maps of river networks

NHDPlus in R

The National Hydrography Dataset (NHD), now operating as NHDPlus, is a geospatial hydrology dataset developed by EPA and USGS. The `nhdplusTools` R package allows us to access NHDPlus information for use in spatial applications for rivers and streams in the US. This package works well when used in concert with the `dataRetrieval` and `sf` packages.

Key terms from the nhdplusTools guide:

"**Flowline:** The NHD name for a hydrographic representation of a flowing body of water. Flowline is generally used when referring to geometry."

"**Flowpath:** The HY_Features name for a hydrologic feature that is the primary path water follows through a catchment; either from headwater to outlet or inlet to outlet. Flowpath is used when describing aspects of the abstract flowpath featuretype, generally in relation to a flowpath's relationship to a catchment."

"**Catchment:** The most abstract unit of hydrology in HY_Features is the catchment. It is a physiographic unit with zero or one inlets and one outlet. It does not inherently have any conceptual realizations. Rather, a given catchment can be realized in a number of ways; flowpath, divide, and networks of flowpaths and divides are the primary realizations."

"**Catchment divide:** NHD "catchment" polygons are more accurately described as "catchment divide" features. Because of the overlap with the HY_Features abstract "catchment" feature type, "catchment divide" is used for polygon representations of catchments."

We will be going through a basic example today, but there are numerous extensions that can be explored. A good start guide can be found here

Watershed and flowpath delineation

Let's delineate the upstream watershed for the sampling and gage site: Neuse River at Kinston, NC.

```
# Recall there are a lot of parameters measured at the site
NeuseParams <- whatNWISdata(siteNumbers = "02089500")
```

```
## Warning: 1 failed to parse.
```

```
# Extract latitude and longitude for the site
NeuseCoords <- NeuseParams %>%
  select(site_no, dec_lat_va, dec_long_va) %>%
  distinct()
```

```
# Define the gage site as the starting point
start_point <- st_sfc(st_point(c(NeuseCoords$dec_long_va, NeuseCoords$dec_lat_va)),
                      crs = 4269) # NAD83, commonly  used by US agencies
start_comid <- discover_nhdplus_id(start_point)
# start_point2 <- st_as_sf(data.frame(x = NeuseCoords$dec_long_va, y =  NeuseCoords$dec_lat_va),
#                              coords = c("x", "y"), crs = 4269)

# Navigate the NLDI network
NLDI <- navigate_nldi(list(featureSource = "comid", featureID = start_comid),
                          mode = "upstreamTributaries",
                          distance_km = 1000)

# Extract watershed and flowpath information
subset_file <- tempfile(fileext = ".gpkg")
subset <- subset_nhdplus(comids = as.integer(NLDI$UT$nhdplus_comid),
                         output_file = subset_file,
                         nhdplus_data = "download",
                         flowline_only = FALSE,
                         return_data = TRUE, overwrite = TRUE)
```

```
## All intersections performed in latitude/longitude.
```

```
## Reading NHDFlowline_Network
```

```
## Warning in get_flowline_subset(nhdplus_data, comids, output_file, status, :
## Download functionality not tested for this many comids
```

```
## Writing NHDFlowline_Network
```

```
## Reading CatchmentSP
```

```
## Found invalid geometry, attempting to fix.
```

```
## Writing CatchmentSP
```

```
## Found invalid geometry, attempting to fix.
## Found invalid geometry, attempting to fix.
```

```
# Create data frames
flowline <- subset$NHDFlowline_Network
catchment <- subset$CatchmentSP
waterbody <- subset$NHDWaterbody

class(flowline)
```

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

```
class(catchment)
```

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

```
class(waterbody)
```

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

```
# find gages near watershed
gages <- get_nwis(AOI = catchment)
```

```
## Warning in if (AOI_type == "POINT") {: the condition has length > 1 and only the
## first element will be used
```

```
## Warning in if (AOI_type == "POINT") {: the condition has length > 1 and only the
## first element will be used
```

```
class(gages)
```

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

```
# find gages only within watershed
gages <- st_intersection(gages, catchment)
```

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

Mapping the network

The `plot_nhdplus` function is a ready-made option that includes a basemap. We can also use ggplot and the `geom_sf` function to create plots from the ground up.

Extensions:

- Advanced Network Attributes including hydrosequence and level path
- Indexing and Referencing
- Advanced plotting

```
plot_nhdplus("USGS-02089500", streamorder = 3)
```

```
## Found invalid geometry, attempting to fix.
## Found invalid geometry, attempting to fix.
## Found invalid geometry, attempting to fix.
```

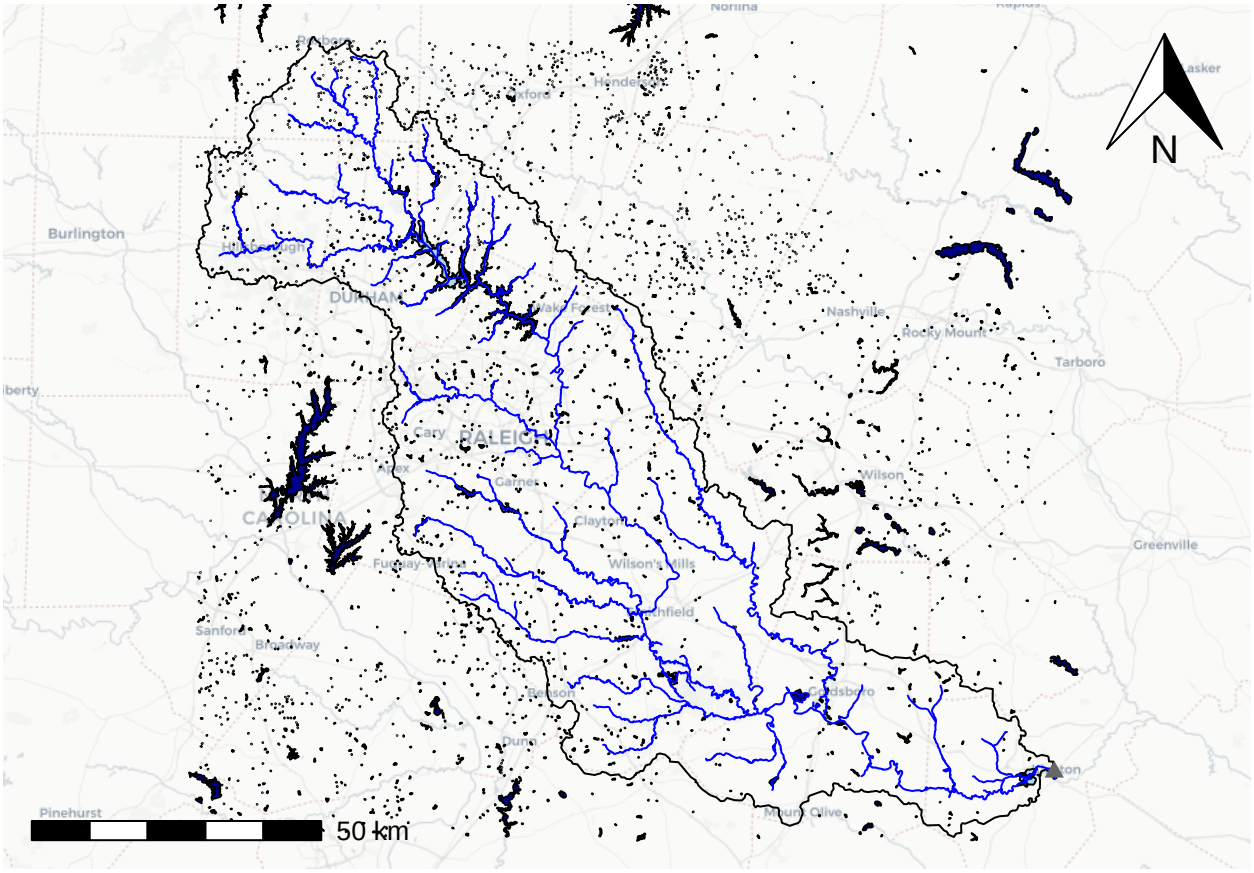
```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj =
## prefer_proj): Discarded ellps WGS 84 in Proj4 definition: +proj=merc +a=6378137
## +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +k=1 +units=m +nadgrids=@null
## +wktext +no_defs
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj =
## prefer_proj): Discarded datum WGS_1984 in Proj4 definition
```

```
## Zoom: 9
```

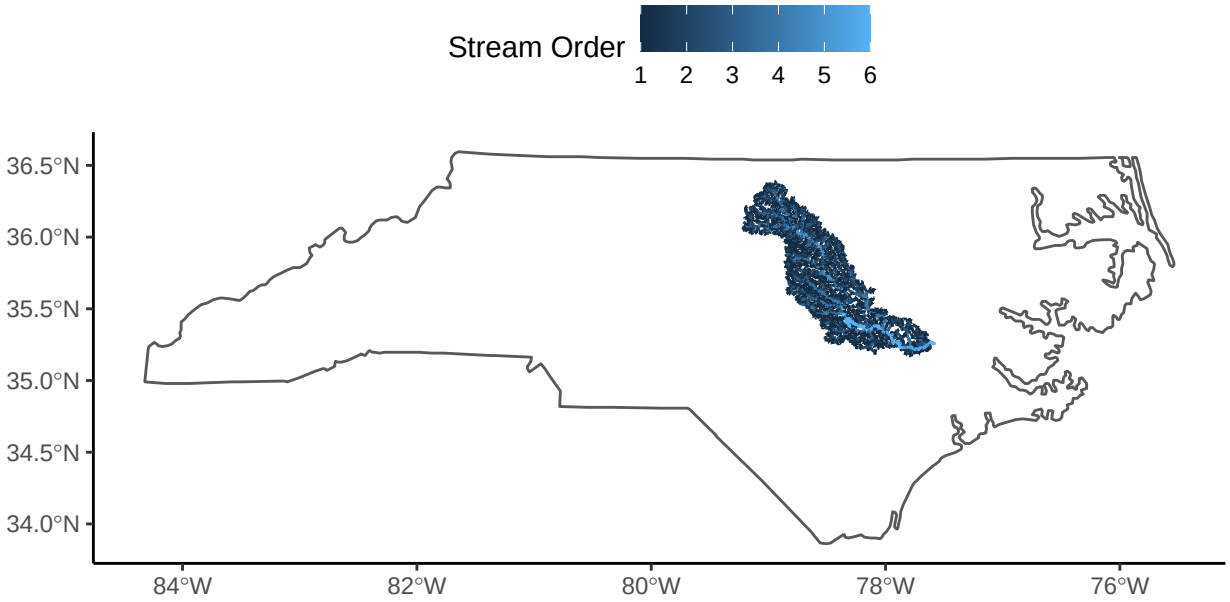
```
## Map tiles by Carto, under CC BY 3.0. Data by OpenStreetMap, under ODbL.
```

```
## Audotdetect projection: assuming Google Mercator (epsg 3857)
```

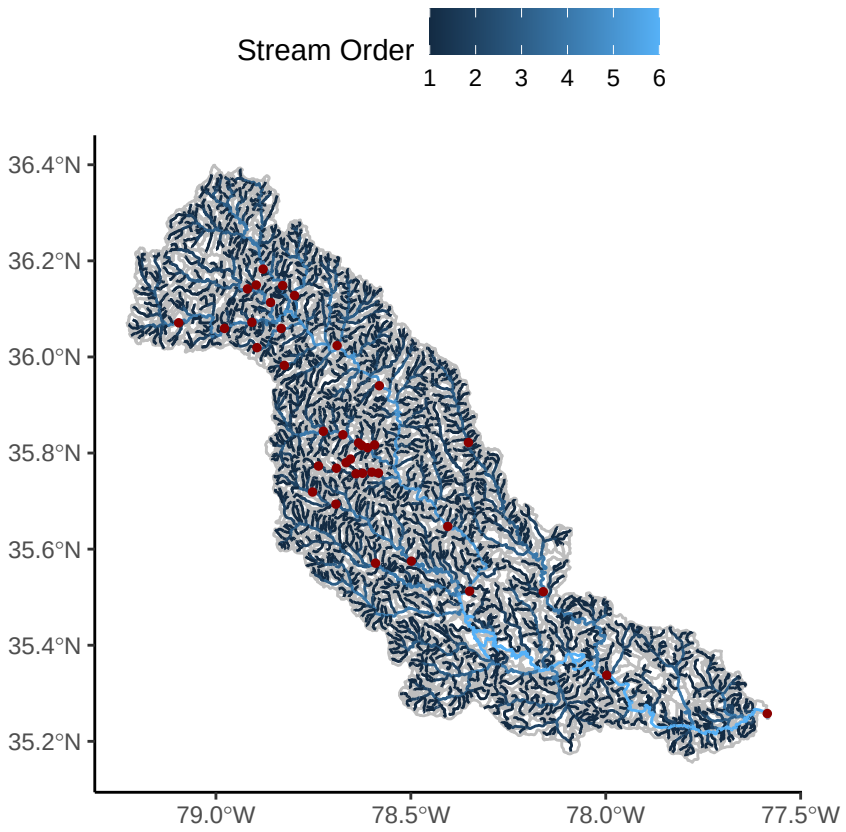



```
states <- st_as_sf(map(database = "state", plot = FALSE, fill = TRUE, col = "white"))
nc <- filter(states, ID == "north carolina")

ggplot(nc) +
  geom_sf(fill = "white") +
  geom_sf(data = flowline, aes(color = streamorde)) +
  labs(color = "Stream Order") +
  theme(legend.position = "top")
```



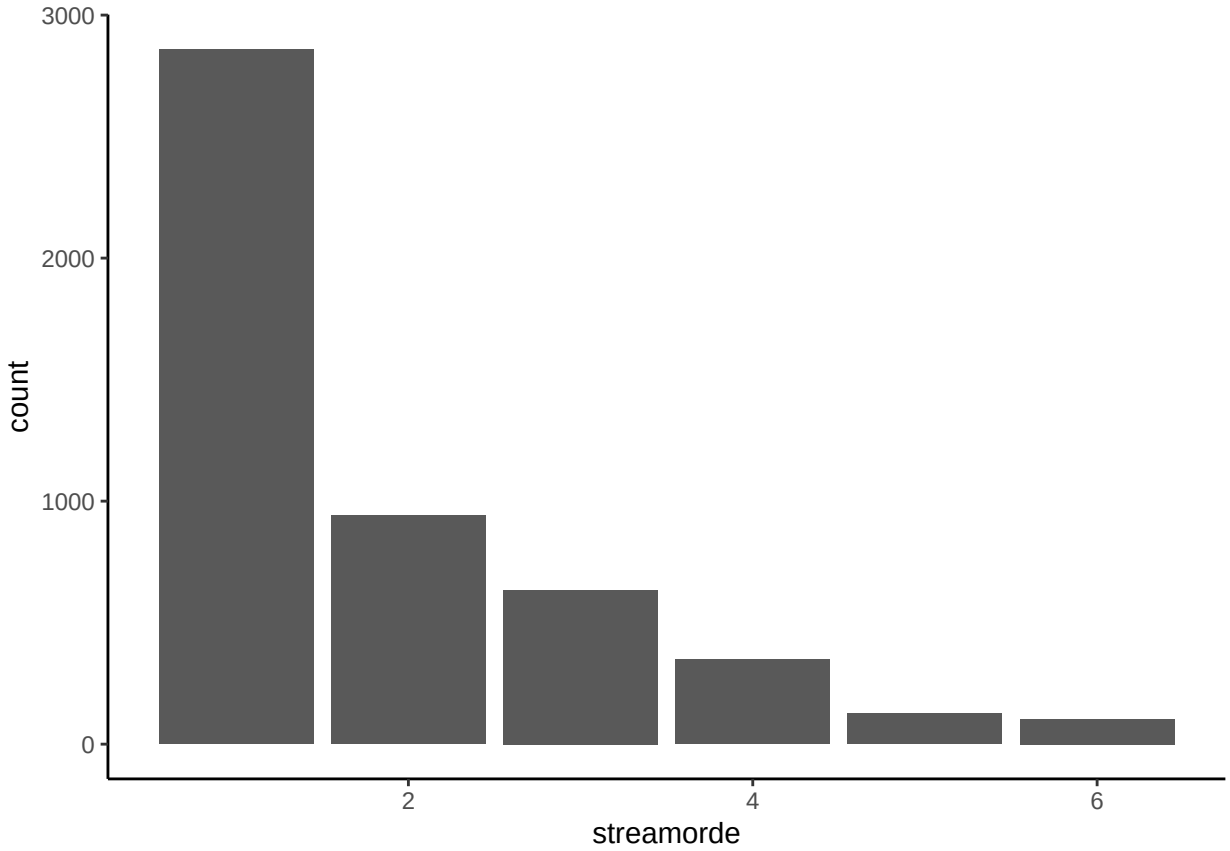

```
ggplot(catchment) +
  geom_sf(fill = "white", color = "gray", lwd = 0.5) +
  geom_sf(data = flowline, aes(color = streamorde)) +
  geom_sf(data = gages, color = "darkred", size = 1) +
  labs(color = "Stream Order") +
  theme(legend.position = "top")
```



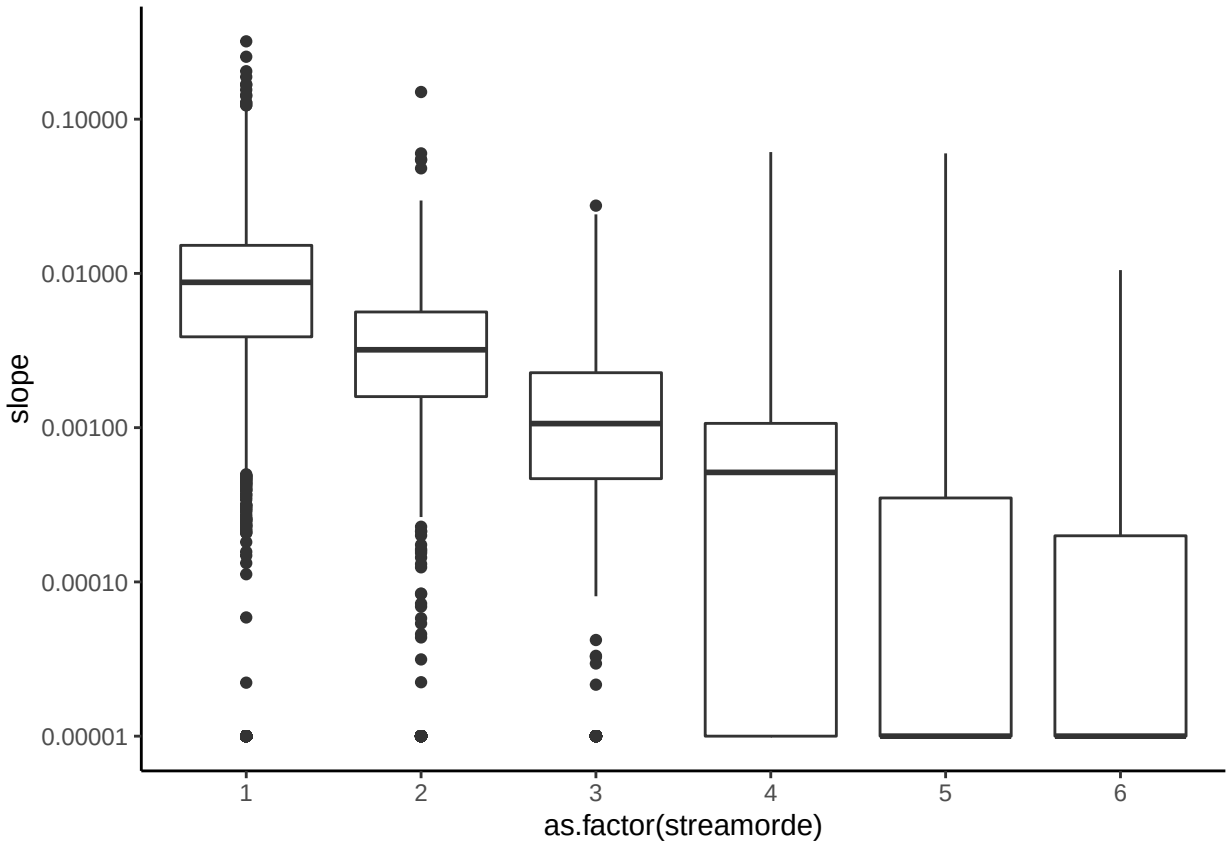
Analyzing tabular information from the network

1. How are flowlines distributed with respect to stream order?
2. How do flowline slopes compare across stream order?
3. Total drainage area

```
ggplot(flowline, aes(x = streamorde)) +
  geom_bar()
```




```
ggplot(subset(flowline, slope >= 0),
       aes(x = as.factor(streamorde), y =slope)) +
  geom_boxplot() +
  scale_y_log10()
```




```
max(flowline$totdasqkm)
```

```
## [1] 7020.258
```