

Figure 1: Exploratory plots

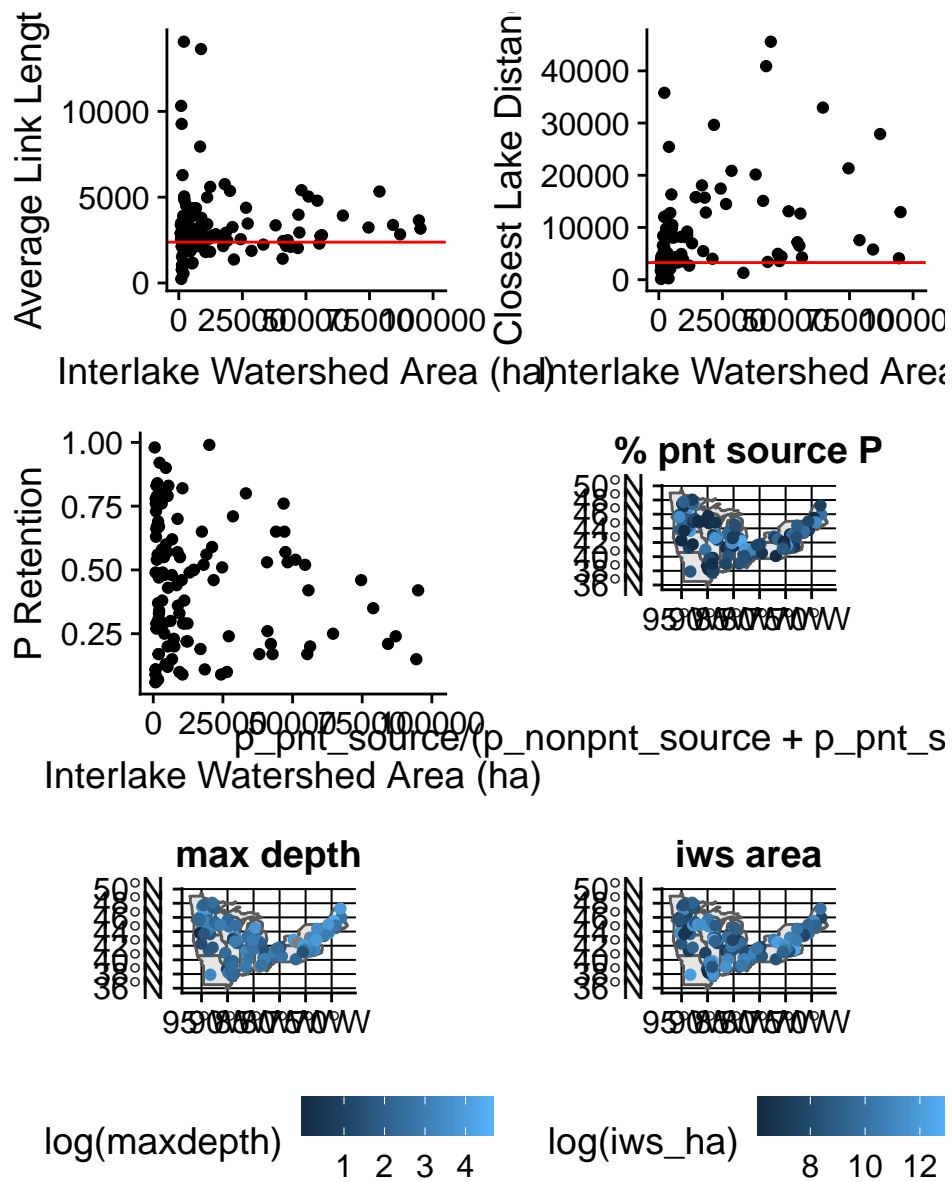


Figure 2: Exploratory plots cont

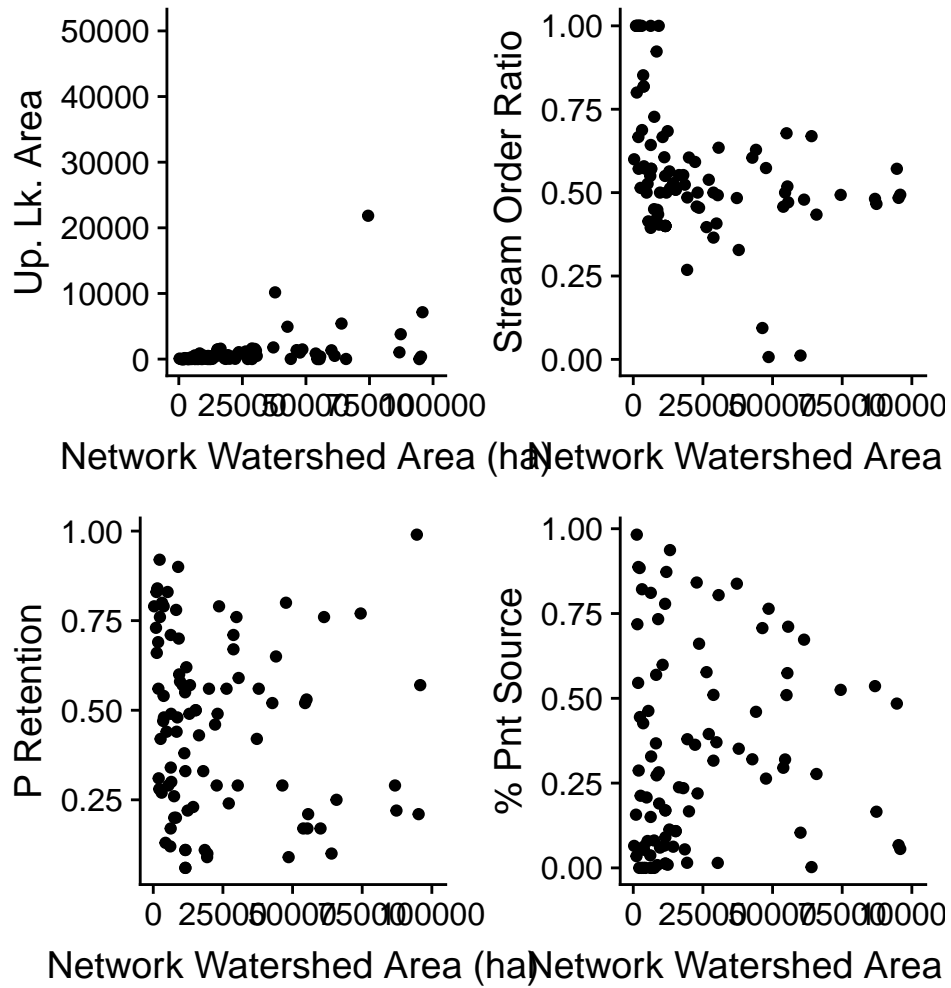
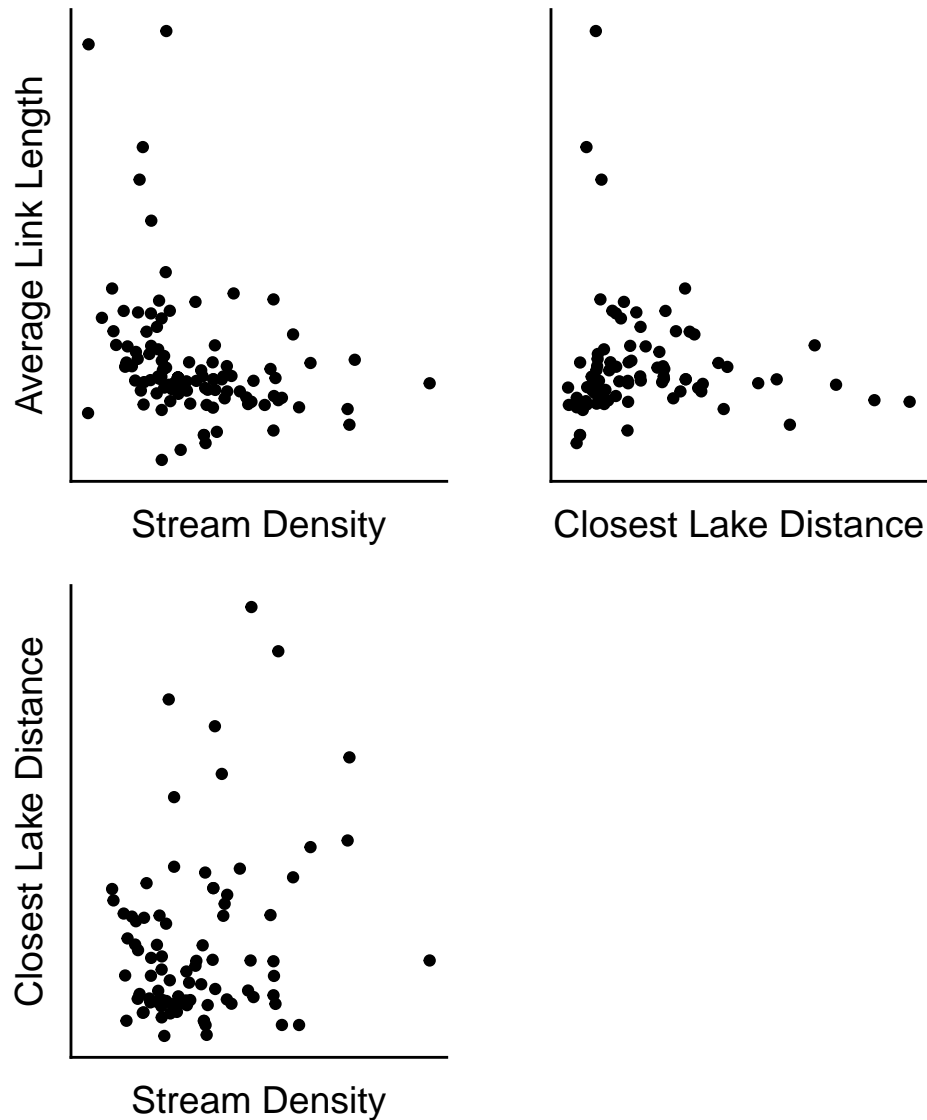


Figure 3: Exploratory plots cont x 2



```
# cran
library(concaveman)
library(sf)
library(dplyr)
library(mapview)
library(sp)
library(ggsn)

# github
library(vapour)
library(nhdR)

## Loading required package: maps

library(LAGOSNE)
library(streamnet)

pull_iws <- function(lagoslakeid, maxsteps = 15){
  gdb_path <- path.expand("~/local/share/LAGOS-GIS/lagos-ne_gis.gpkg")
```

```

crs <- "+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96 +x_0=0 +y_0=0 +datum=NAD83 +units=m +n

layer_name <- "LAGOS_NE_All_Lakes_4ha"
query      <- paste0("SELECT * FROM ", layer_name,
                     " WHERE lagoslakeid = '",
                     lagoslakeid, "'")

lakepoly_geom <- st_as_sfc(vapour_read_geometry(gdb_path, sql = query))
lakepoly      <- st_sf(vapour_read_attributes(gdb_path, sql = query),
                     geometry = lakepoly_geom, crs = crs)
lakepoly <- st_zm(lakepoly)

layer_name <- "IWS"
query      <- paste0("SELECT * FROM ", layer_name,
                     " WHERE lagoslakeid = '",
                     lagoslakeid, "'")

poly_geom <- st_as_sfc(vapour_read_geometry(gdb_path, sql = query))
poly      <- st_sf(vapour_read_attributes(gdb_path, sql = query),
                     geometry = poly_geom, crs = crs)
poly      <- st_zm(poly)
poly      <- st_union(poly, lakepoly)

lines <- nhd_plus_query(poly = poly, dsn = "NHDFlowLine",
                       approve_all_dl = TRUE)$sp$NHDFlowLine

lg      <- lagosne_load("1.087.1")
wb_coords <- as.numeric(
  lake_info(lg, lagoslakeid = lagoslakeid)[,c("nhd_long", "nhd_lat")])

names(lines) <- tolower(names(lines))
extract_network(wb_coords[1], wb_coords[2], lines = lines, maxsteps = maxsteps)
}

pull_nws <- function(lagoslakeid, map = FALSE, maxsteps = 15, buffer_dist = 0.01){
  print(lagoslakeid)
  lg      <- lagosne_load("1.087.1")
  wb_coords <- as.numeric(
    LAGOSNE::lake_info(lg, lagoslakeid = lagoslakeid)[,c("nhd_long", "nhd_lat")])

  extract_network(wb_coords[1], wb_coords[2], maxsteps = maxsteps,
                 buffer_dist = buffer_dist, approve_all_dl = TRUE)
}

pull_lakes <- function(lines){
  poly <- concaveman::concaveman(st_cast(lines, "POINT"))$polygons
  nhd_plus_query(poly = poly,
                 dsn = "NHDWaterbody", approve_all_dl = TRUE)$sp$NHDWaterbody
}

run_lake <- function(lagoslakeid, lines, lakes){
  print(lagoslakeid)
  calc_metrics(lines, lakes)
}

```

```

}

pull_nws_metrics <- function(lakes){

  if(nrow(lakes) == 0){
    list(baseflow = NA,
         stream_density = NA)
  }else{
    # pull llds
    lg <- lagosne_load("1.087.1")
    poly <- concaveman::concaveman(
      st_cast(lakes, "POINT"))$polygons
    lg_sf <- st_transform(coordinate(lg$locus), st_crs(poly))
    poly_lake_intersect <- st_intersects(lg_sf, poly)
    res <- lg_sf[which(unlist(lapply(poly_lake_intersect,
      function(x) length(x) > 0))),]
    res <- data.frame(lagoslakeid = res$lagoslakeid, stringsAsFactors = FALSE)

    # join metrics

    # baseflowindex - area-weighted average
    hu12s <- dplyr::filter(lg$locus, lagoslakeid %in% res$lagoslakeid)

    hu12_ids <- unique(hu12s$hu12_zoneid)
    hu12_areas <- dplyr::filter(lg$hu12, hu12_zoneid %in% hu12_ids)$hu12_ha
    hu12_baseflow <- dplyr::filter(
      dplyr::select(lg$hu12.chag, hu12_zoneid, hu12_baseflowindex_mean),
      hu12_zoneid %in% hu12_ids)$hu12_baseflowindex_mean

    baseflow <- sum(hu12_areas * hu12_baseflow) / sum(hu12_areas)

    # stream density - area-weighted average
    res <- dplyr::left_join(res, dplyr::select(lg$iws, lagoslakeid, iws_ha))
    res <- dplyr::left_join(res, dplyr::select(lg$iws.conn, lagoslakeid,
      iws_streamdensity_streams_density_mperha))
    res <- res[!is.na(res$iws_ha),]

    nws_ha <- sum(res$iws_ha)

    stream_density <-
      sum(res$iws_ha * res$iws_streamdensity_streams_density_mperha) / nws_ha

    list(baseflow = baseflow,
         stream_density = stream_density,
         nws_ha = nws_ha)
  }
}

```

```

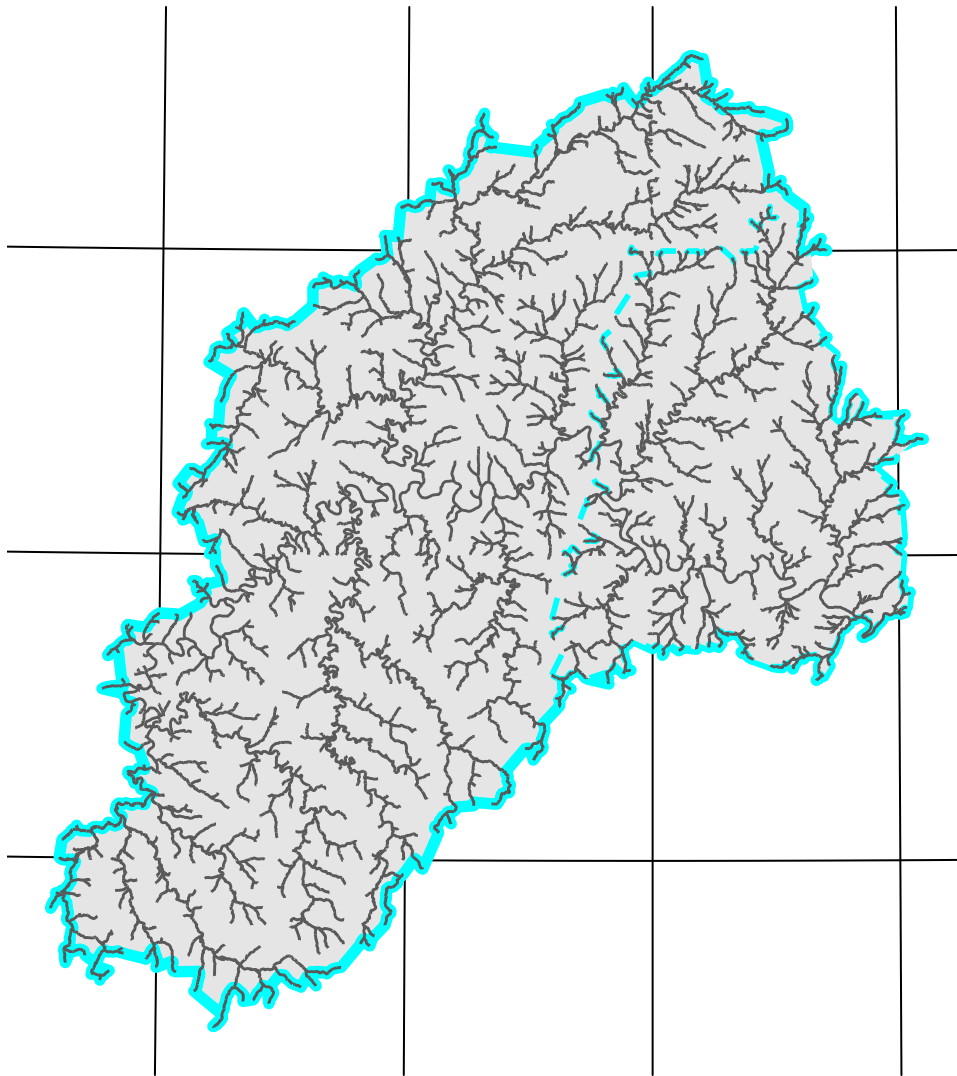
## Warning in st_cast.sf(iws, "POINT"): repeating attributes for all sub-
## geometries for which they may not be constant

```

```

## Warning in st_cast.sf(nws, "POINT"): repeating attributes for all sub-
## geometries for which they may not be constant

```



```
## although coordinates are longitude/latitude, st_intersects assumes that they are planar
```

Average
link length

