Data Prep - Summer Temperatures

Libraries

```
library(sf)
library(tidyverse)
library(spmodel)
library(data.table)
library(ggplot2)
library(StreamCatTools)
library(tigris)
library(prism)
library(terra)
```

Data Development

Stream Temperature (st) Observations

- These data represent raw daily mean values from USGS loggers/stations
- We applied QA/QC process to flag and remove records that can represent a variety of issues (see QA documentation in SI)
- This code does the following:

1. Reads raw flagged data; removes those that are flagged to be removed
2. Calculates mean monthly values for July and August for sites with >20 days of record for those months
3. Converts data to simple feature spatial object

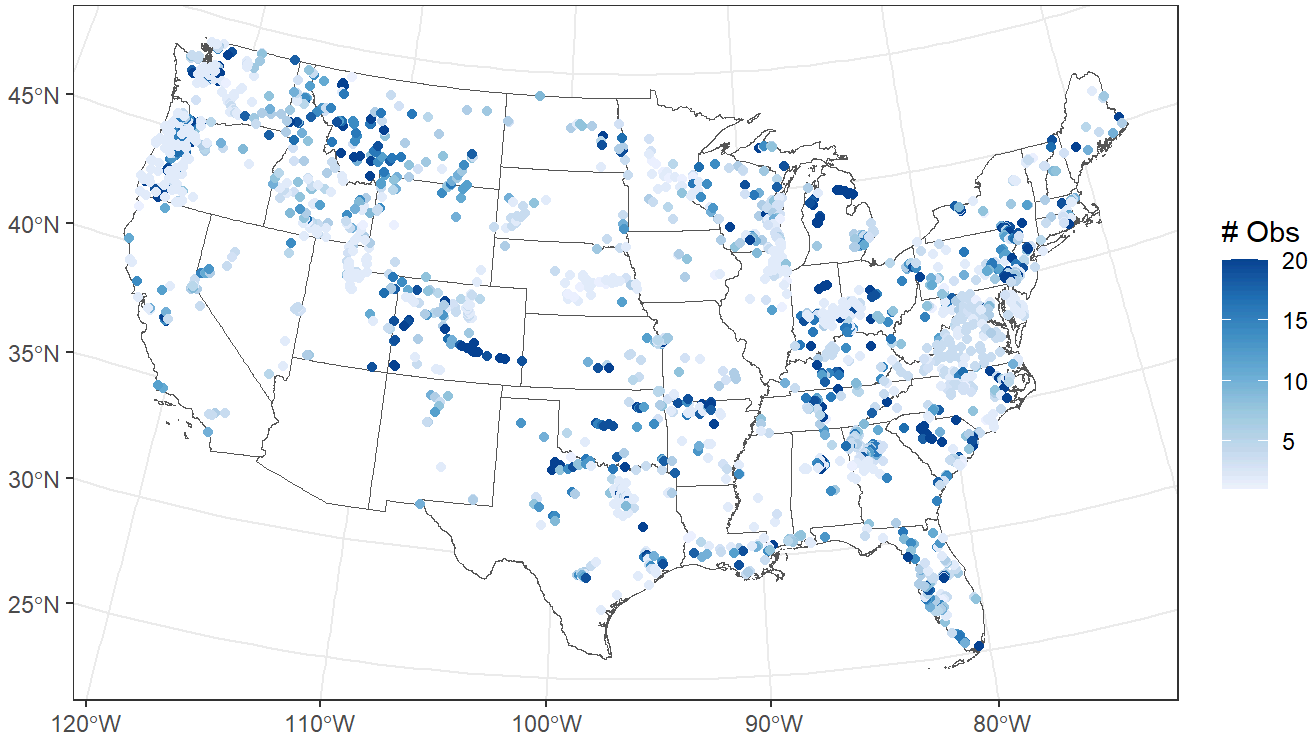
```
pts <- read_rds('../data/pts_sf.rds')

st <- read_rds('../data/stream_temperatures_raw_flagged.rds') %>%

  # Filter anything with remove flag
  filter(Flag_Remove == "") %>%

  # Add year and month
  mutate(date = lubridate::ymd(YYYYMMDD),
         year = year(date),
         month = month(date)) %>%

  # Select just necessary columns (DV_VALUE = daily temperatures)
  dplyr::select(SITECODE, DV_VALUE, year, month) %>%

  # Calc mean for year/month by sites
  group_by(SITECODE, year, month) %>%
```

```
  summarise(wtmp_mo = mean(DV_VALUE, na.rm = TRUE),
            count = n()) %>%

  # Filter to just July/August w/ greater than/equal 20 days
  filter(month == 7 | month == 8,
         count >= 20) %>%

  # Join to locations by ID
  left_join(pts, join_by(SITECODE)) %>%

  st_as_sf() %>%

  st_transform(crs = 5070)
```

Map observed values

Map temperature sites and color by number of observations (months with data)

```
states <- tigris::states(cb = TRUE, progress_bar = FALSE)  %>%
  filter(!STUSPS %in% c('HI', 'PR', 'AK', 'MP', 'GU', 'AS', 'VI'))  %>%
  st_transform(crs = 5070)

ggplot() +
  geom_sf(data = states,
          fill = 'white') +
  geom_sf(data = st %>%
            group_by(SITECODE) %>%
            summarise(n = n()),
          aes(color = n)) +
  scale_color_distiller(name = '# Obs',
                        palette = 'Blues',
                        direction = 2) +
  theme_bw()
```




```
ggsave(file = '../figures/number_summer_temperature_obs.png',
       width = 8,
       height = 5,
       units = 'in',
       dpi = 600)
```

Summary of model data table

```
# Number of monthly observations across all sites
nrow(st)
```

```
[1] 16157
```

```
# Number of records for July and August
table(st$month)
```

```
   7    8 
8051 8106 
```

```
# Number of records for each year
```

```
table(st$year)
```

```
1999 2000 2001 2002 2003 2004 2005 2006 2007 2008
1177 1322 1322 1499 1739 1766 1661 1716 2041 1914
```

```
# Summary of data
summary(st)
```

```
  SITECODE              year          month          wtmp_mo
Length:16157       Min.   :1999   Min.   :7.000   Min.   : 3.60
Class :character   1st Qu.:2002   1st Qu.:7.000   1st Qu.:18.31
Mode  :character   Median :2004   Median :8.000   Median :22.26
                   Mean   :2004   Mean   :7.502   Mean   :21.72
                   3rd Qu.:2006   3rd Qu.:8.000   3rd Qu.:25.63
                   Max.   :2008   Max.   :8.000   Max.   :33.92
    count                geometry
Min.   :20.00   POINT        :16157
1st Qu.:31.00   epsg:5070    :    0
Median :31.00   +proj=aea ...:    0
Mean   :30.24
3rd Qu.:31.00
Max.   :31.00
```

USGS flow metrics

- Modeled monthly (July and August) flow estimates for each site (source: USGS).
- Data not easily accessible for new sites.
- We used table to filter stations with data issues that were identified by USGS.

```
flow <- fread('../data/comid_matches_daren_usgs_flow.csv') %>%
  dplyr::select(SITECODE, COMID, Comment, July.Q.mn,
                August.Q.mn, July.Q.md, August.Q.md)
```

NHDPlus flow metrics

Modeled monthly (July and August) flow estimates from NHDPlus

- Data available for calibration sites and USGS/EPA fish sites.
- Flow values are very correlated with USGS estimates of flow from above.
- USGS values included some very large values, but inspection of streams in Google Maps suggested that NHDPlus flow estimates of river size were more accurate.

```
nhd_dir <- 'C:/Users/RHill04/WorkFolder/GIS/NHDPlusV21/NHDPlusNationalData/NHDPlusV21_National_Sea

nhd_flow <-
  st_read(dsn = paste0(nhd_dir),
          layer = 'NHDFlowline_Network') %>%
```

```
  st_drop_geometry() %>%
  dplyr::select(COMID, QE_07, QE_08) %>%
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'nhdflow') %>%
  mutate(month = str_replace(tmpcol, 'QE_0', '') %>%
           as.integer()) %>%
  dplyr::select(-tmpcol)
```

```
Reading layer `NHDFlowline_Network' from data source

`C:\Users\RHill04\WorkFolder\GIS\NHDPlusV21\NHDPlusNationalData\NHDPlusV21_National_Seamless_Flat
tened_Lower48.gdb'
  using driver `OpenFileGDB'
Simple feature collection with 2691339 features and 137 fields
Geometry type: MULTILINESTRING
Dimension:     XYZM
Bounding box:  xmin: -124.7332 ymin: 24.63052 xmax: -66.94983 ymax: 49.37661
z_range:       zmin: 0 zmax: 0
m_range:       mmin: -2.35e-05 mmax: 100
Geodetic CRS:  NAD83
```

StreamCat (sc) static metrics

Static watershed/local catchment metrics:

- Elevation (Cat)
- Calcium oxide content of underlying lithology (Ws)
- Base flow index (Ws)
- Water table depth (Ws)
- Watershed area (Ws)
- Runoff (ws)
- Clay soil content (Ws)
- Sand soil content (Ws)
- Topographic wetness index (Ws)
- National Anthropogenic Barriers dam density (screened dams of NID) (Ws)
- Hydrologic conductivity (HydrlCond) (Ws)

```
comids <- flow$COMID %>%
  na.omit() %>%
  unique()

#Pull in static watershed metrics
sc <-
  sc_get_data(metric = 'HydrlCond,Runoff,Clay,Sand,WtDep,WetIndex,NABD_Dens,NABD_NRMSTOR,BFI,PRECI
              aoi = 'catchment,watershed',
              comid = comids) %>%
  dplyr::select(COMID, ELEVCAT, CAOWS, BFIWS, WTDEPWS,
                WSAREASQKM, RUNOFFWS, CLAYWS, SANDWS, WETINDEXWS,
                NABD_DENSCAT, NABD_DENSWS, NABD_NRMSTORWS,
                PRECIP8110WS, HYDRLCONDWS) %>%
```

```
  mutate(dam_prescat = ifelse(NABD_DENSCAT > 0, 1, 0),
         dam_presws = ifelse(NABD_DENSWS > 0, 1, 0))
```

StreamCat Year-Specific NLCD data

Riparian forest cover (catchment)

1. Extracts yrs. 2001-2008 NLCD from StreamCat for riparian (~100m buffer) watersheds.
2. Filters data to just CONIF, DECID, or MXFST types.
3. Pivots table to include year of NLCD and % riparian forest column.

```
riparian_forest <-
  sc_nlcd(year = '2001, 2004, 2006, 2008',
          aoi = 'riparian_watershed',
          comid = comids) %>%
  dplyr::select(COMID,
                grep('CONIF|DECID|MXFST', names(.))) %>%
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'PCTFSTXXXXWSRP100') %>%
  mutate(year = as.integer(
    str_replace_all(tmpcol, 'PCTMXFST|PCTDECID|PCTCONIF|WSRP100', ''))) %>%
  group_by(COMID, year) %>%
  summarise(PCTFSTXXXXWSRP100 = sum(PCTFSTXXXXWSRP100))
```

Crop cover (watershed)

Same process as riparian forest cover, but for NLCD type CROP.

```
crop <-
  sc_nlcd(year = '2001, 2004, 2006, 2008',
          aoi = 'watershed',
          comid = comids) %>%
  dplyr::select(COMID,
                grep('CROP', names(.))) %>%
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'PCTCROPXXXXWS') %>%
  mutate(year = as.integer(
    str_replace_all(tmpcol, 'PCTCROP|WS', ''))) %>%
  dplyr::select(-tmpcol)
```

Urban cover (watershed)

Same process as riparian forest cover, but for NLCD type PCTURBLO, PCTURBMD, or PCTURBHI.

```
urban <-
  sc_nlcd(year = '2001, 2004, 2006, 2008',
          aoi = 'watershed',
          comid = comids) %>%
  dplyr::select(COMID,
                grep('PCTURBLO|PCTURBMD|PCTURBHI', names(.))) %>%
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'PCTURBXXXXWS') %>%
```

```
  mutate(year = as.integer(
    str_replace_all(tmpcol, 'PCTURBLO|PCTURBMD|PCTURBHI|WS', ''))) %>%
  group_by(COMID, year) %>%
  summarise(PCTURBXXXXWS = sum(PCTURBXXXXWS))
```

Lake/Reservoir (open water) in watershed (watershed)

Same process as riparian forest cover, but for NLCD type PCTOW.

Variable added to interact with dam presence/absence to account for stations that occur below natural lakes or man made reservoirs.

```
water <-
  sc_nlcd(year = '2001, 2004, 2006, 2008',
          aoi = 'watershed',
          comid = comids) %>%
  dplyr::select(COMID,
                grep('PCTOW', names(.))) %>%
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'PCTOWXXXXWS') %>%
  mutate(year = as.integer(
    str_replace_all(tmpcol, 'PCTOW|WS', ''))) %>%
  group_by(COMID, year) %>%
  summarise(PCTOWXXXXWS = sum(PCTOWXXXXWS))
```

PRISM Climate Data

Air temperature

```
years <- 1999:2008

# Set the PRISM directory (creates directory in not present)
prism_set_dl_dir("../data/prism_data", create = TRUE)

# Download monthly PRISM rasters (tmean)
get_prism_monthlys('tmean',
                   years = years,
                   mon = 7:8,
                   keepZip = FALSE)
```

```
  |
  |                                                                      |   0%
  |
  |====                                                                  |   5%
  |
  |=======                                                               |  10%
  |
  |==========                                                            |  15%
  |
```

```
  |==============                                                        |  20%
  |
  |==================                                                    |  25%
  |
  |======================                                                |  30%
  |
  |========================                                              |  35%
  |
  |============================                                          |  40%
  |
  |================================                                      |  45%
  |
  |==================================                                    |  50%
  |
  |====================================                                  |  55%
  |
  |========================================                              |  60%
  |
  |============================================                          |  65%
  |
  |==============================================                        |  70%
  |
  |================================================                      |  75%
  |
  |==================================================                    |  80%
  |
  |======================================================                |  85%
  |
  |========================================================              |  90%
  |
  |============================================================          |  95%
  |
  |======================================================================| 100%
```

```r
# Create stack of PRISM climate rasters to extract values
tmn <- pd_stack((prism_archive_subset("tmean","monthly",
                                      years = years,
                                      mon = 7:8)))

# Extract tmean at sample points and massage data
tmn <- terra::extract(tmn,
                      # Transform pts to CRS of PRISM on the fly
                      pts %>%
                        st_transform(crs = st_crs(tmn))) %>%

  # Add site IDs to extracted values
  data.frame(SITECODE = pts$SITECODE, .) %>%

  # Remove front and back text from PRISM year/month in names
  rename_with( ~ stringr::str_replace_all(., 'PRISM_tmean_stable_4kmM3_|_bil', '')) %>%
```

```
  # Pivot to long table and calle column tmeanPRISM
  pivot_longer(!SITECODE, names_to = 'year_month',
               values_to = 'tmeanPRISM') %>%

  # Create new column of year
  mutate(year = year(ym(year_month)),
         month = month(ym(year_month))) %>%

  dplyr::select(-year_month)
```

Precipitation

```
get_prism_monthlys('ppt',
                   years = years,
                   mon = 7:8,
                   keepZip = FALSE)
```

```
  |
  |                                                                      |   0%
  |
  |====                                                                  |   5%
  |
  |=======                                                               |  10%
  |
  |==========                                                            |  15%
  |
  |==============                                                        |  20%
  |
  |==================                                                    |  25%
  |
  |======================                                                |  30%
  |
  |=========================                                             |  35%
  |
  |=============================                                         |  40%
  |
  |================================                                      |  45%
  |
  |====================================                                  |  50%
  |
  |========================================                              |  55%
  |
  |============================================                          |  60%
  |
  |================================================                      |  65%
  |
  |====================================================                  |  70%
```

```
  |
  |=================================================                     |  75%
  |
  |=====================================================                 |  80%
  |
  |=========================================================             |  85%
  |
  |=============================================================         |  90%
  |
  |=================================================================     |  95%
  |
  |=====================================================================| 100%
```

```
ppt <- pd_stack((prism_archive_subset("ppt","monthly",
                                      years = years,
                                      mon = 7:8)))

ppt <- terra::extract(ppt,
                      pts %>%
                        st_transform(crs = st_crs(ppt))) %>%
  data.frame(SITECODE = pts$SITECODE, .) %>%
  rename_with( ~ stringr::str_replace_all(., 'PRISM_ppt_stable_4kmM3_|_bil', '')) %>%
  pivot_longer(!SITECODE, names_to = 'year_month',
               values_to = 'pptPRISM') %>%
  mutate(year = year(ym(year_month)),
         month = month(ym(year_month))) %>%

  dplyr::select(-year_month)
```

Combine data for modeling

- Code creates crosswalk that matches the closest temperature years and NLCD years.
- All geospatial metrics are then joined to location (COMID)/month/year combinations of observed water temperatures.

```
nlcd_years <-
  data.table(year = c(2001, 2004, 2006, 2008)) %>%
  mutate(merge = year) %>%
  setkeyv('merge')

st_years <-
  data.table(year = 1999:2008) %>%
  mutate(merge = year) %>%
  setkeyv('merge')

nearest <-
  nlcd_years[st_years, roll = 'nearest'] %>%
  dplyr::select(year, i.year) %>%
  rename(nlcd_year = year,
         year = i.year)
```

```r
st <- st %>%
  left_join(nearest, join_by(year)) %>%
  left_join(tmn,
            join_by(SITECODE, year, month)) %>%
  left_join(ppt,
            join_by(SITECODE, year, month)) %>%
  left_join(flow, join_by(SITECODE)) %>%
  left_join(sc, join_by(COMID)) %>%
  left_join(riparian_forest,
            join_by(COMID == COMID,
                    nlcd_year == year)) %>%
  left_join(crop,
            join_by(COMID == COMID,
                    nlcd_year == year)) %>%
  left_join(urban,
            join_by(COMID == COMID,
                    nlcd_year == year)) %>%
  left_join(water,
            join_by(COMID == COMID,
                    nlcd_year == year)) %>%
  left_join(nhd_flow,
            join_by(COMID == COMID,
                    month == month)) %>%
  mutate(q_mn = ifelse(month == 7,
                       July.Q.mn,
                       August.Q.mn),
         q_md = ifelse(month == 7,
                       July.Q.md,
                       August.Q.md)) %>%
  dplyr::select(-July.Q.mn:-August.Q.md)

# Write output file for modeling
write_rds(st,
          file = '../data/summer_data.2024.08.08.rds',
          compress = "xz")
```

Canidate Models

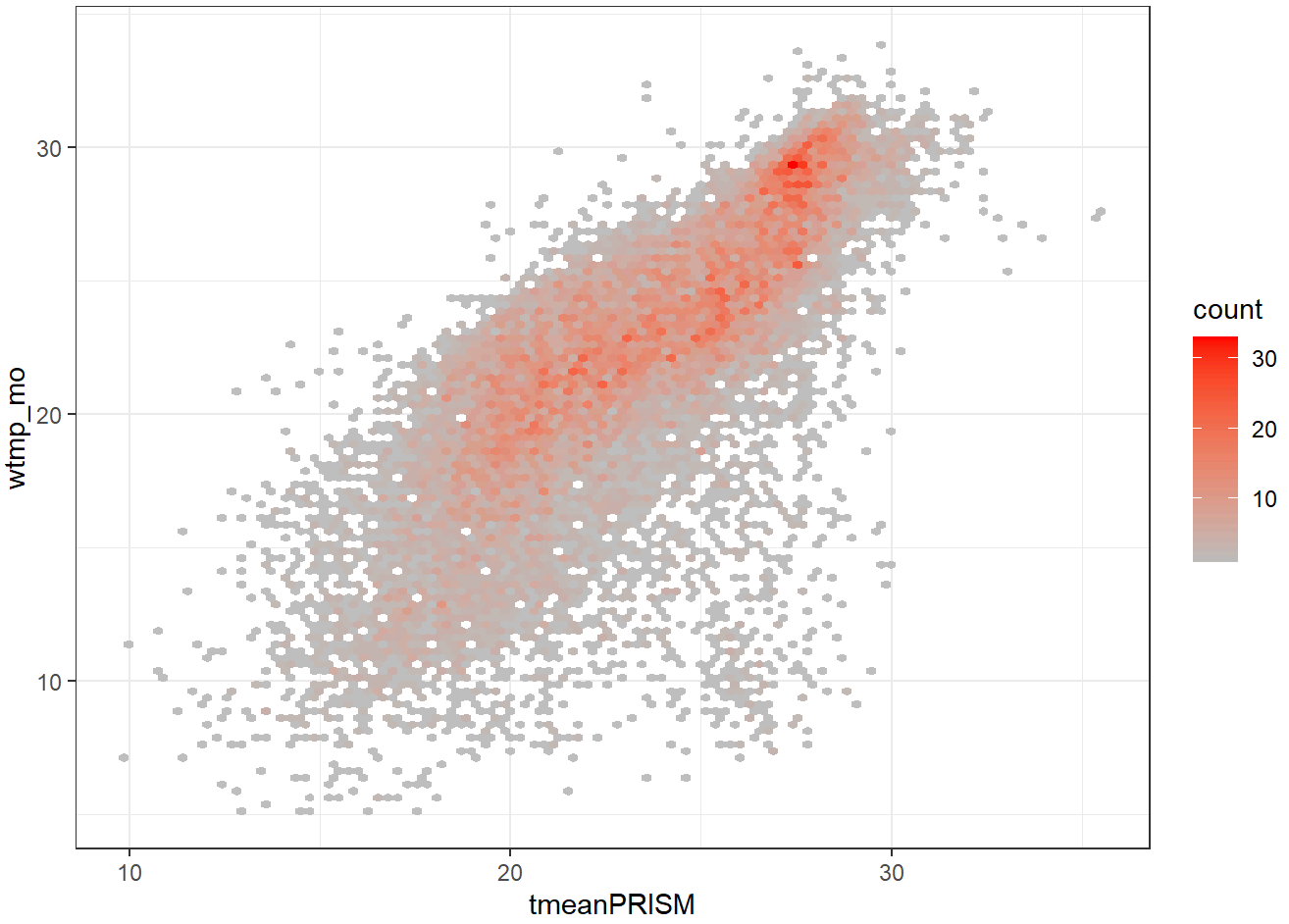
Libraries

```
library(data.table)
library(MASS)
library(lmerTest)
library(car)
library(tidyverse)
library(sf)
```

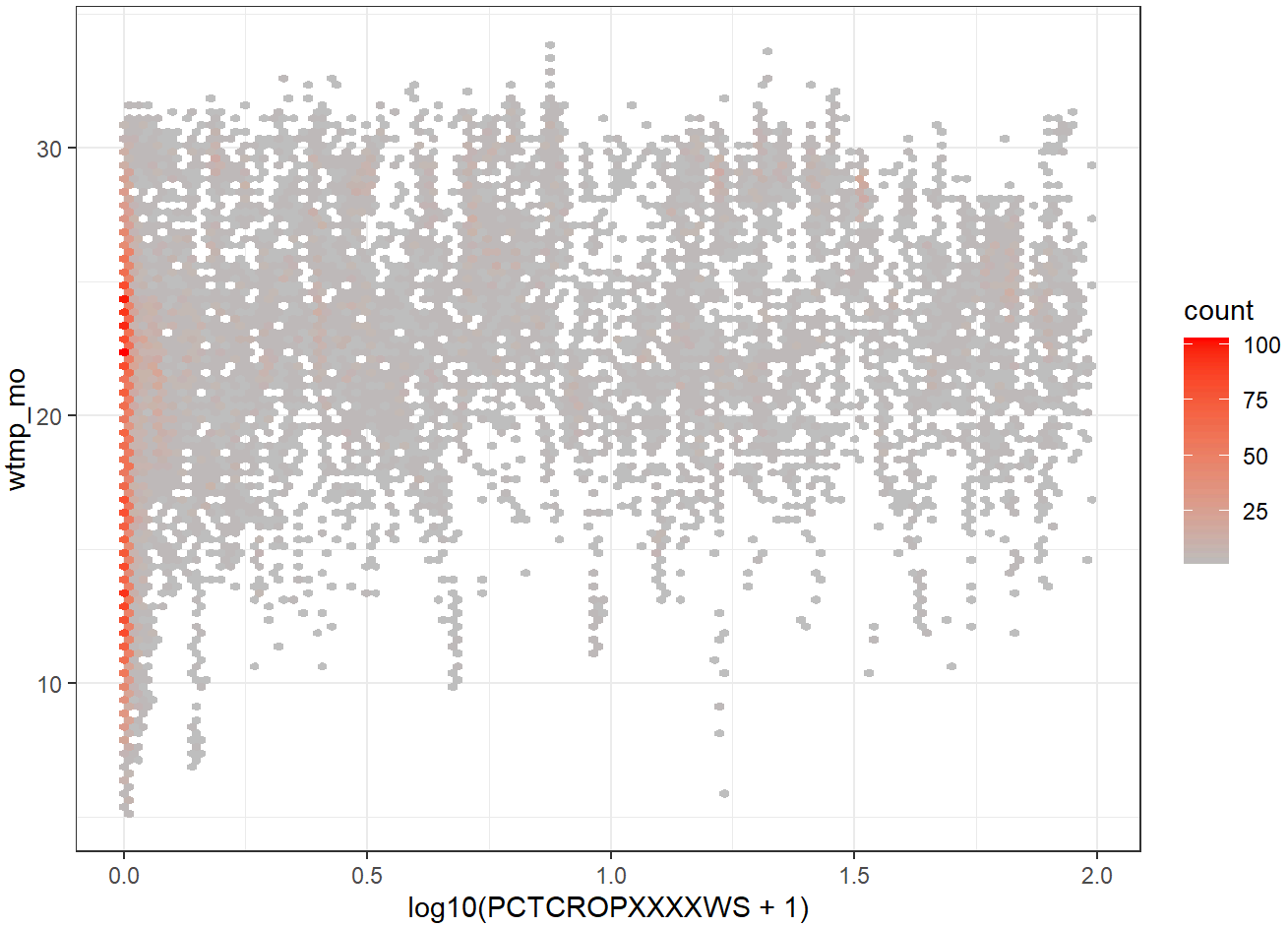
Initial Model Selection

- A fully-expressed formula is defined with multiple potential interactions with air temperature and month.
- Two selection procedures are then used to identify two candidate models. These procedures include:
- Backward selection (lmerTest::step) with a random intercept (COMID) included.
- Backward/Forward selection without a random intercept (COMID) included.
- Candidate models are then saved.

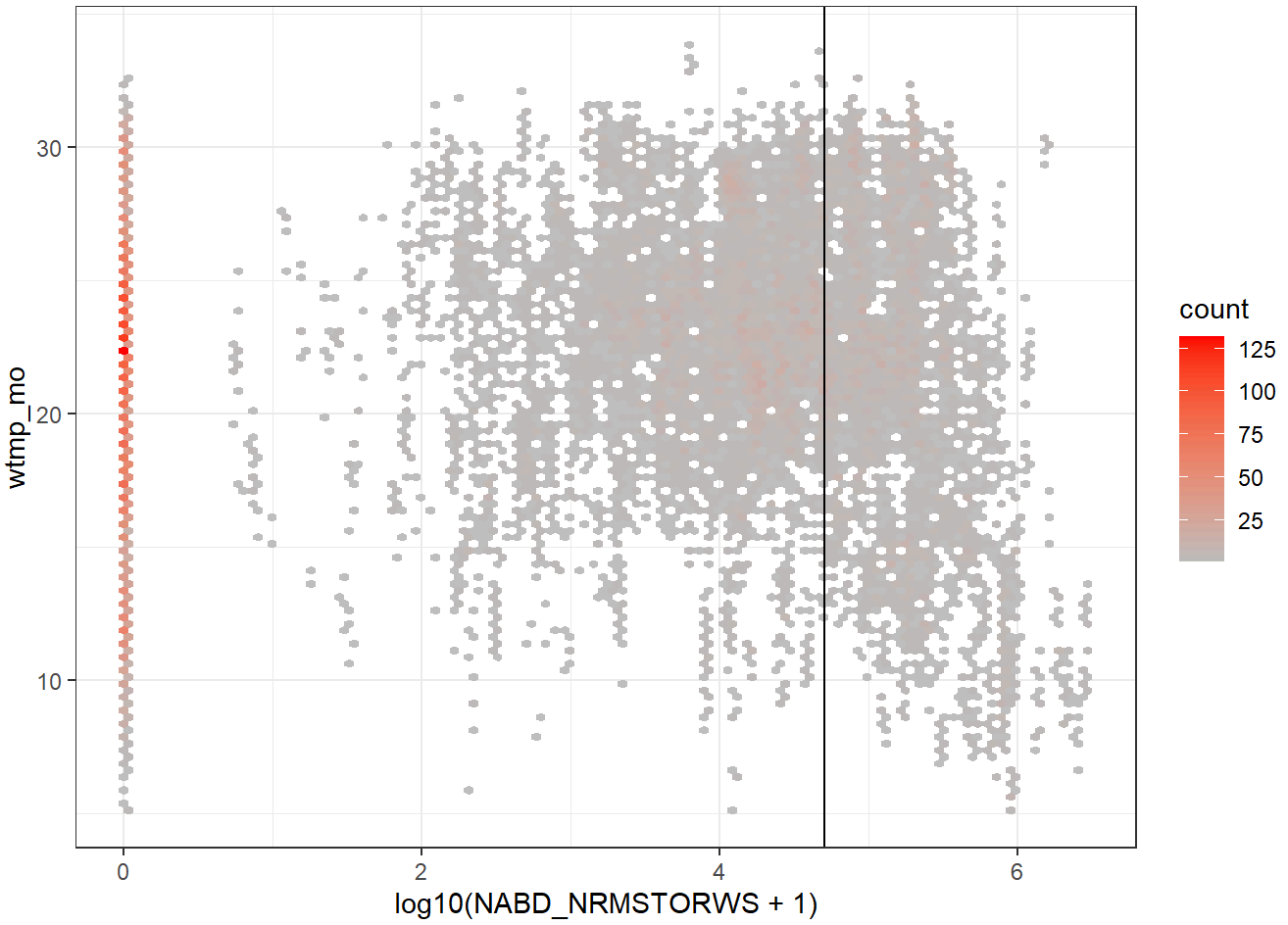
```
st <- read_rds('../data/summer_data.2024.08.08.rds') %>%
  dplyr::mutate(month = as.character(month),
                COMID = as.character(COMID)) %>%
  na.omit() %>%
  dplyr::ungroup() %>%
  st_drop_geometry() %>%
  dplyr::mutate(log10_dam = log10(NABD_NRMSTORWS + 1),
                large_dam = ifelse(log10_dam > 5, 1, 0) %>%
                  as.character(),
                no_dam = ifelse(log10_dam == 0, 1, 0) %>%
                  as.character())

ggplot(data = st,
       aes(x = tmeanPRISM,
           y = wtmp_mo)) +
  geom_hex(bins = 100) +
  scale_fill_gradient(low = "grey", high = "red") +
  theme_bw()
```



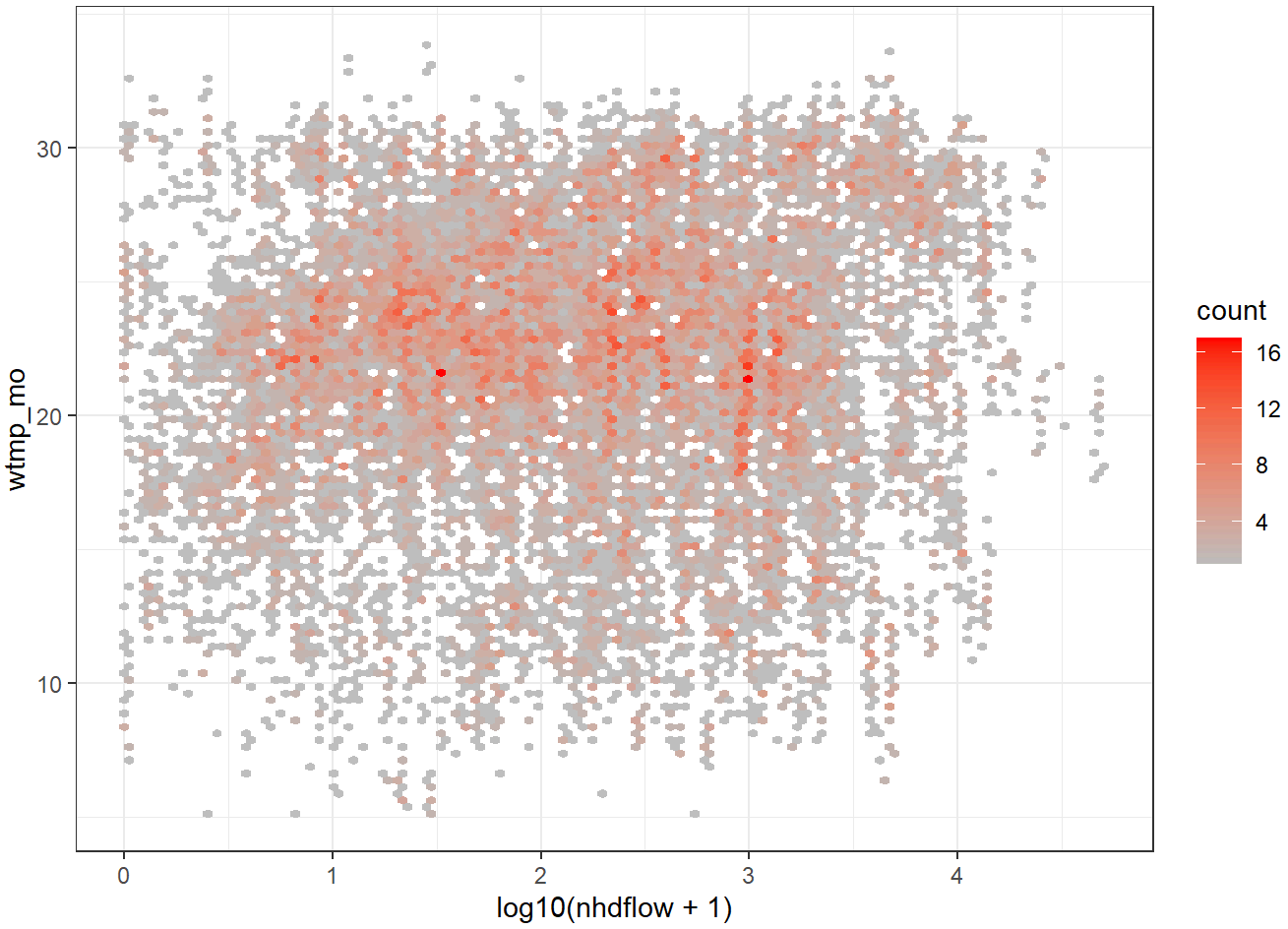

```
ggplot(data = st,
       aes(x = log10(PCTCROPXXXXWS+1),
           y = wtmp_mo)) +
  geom_hex(bins = 100) +
  scale_fill_gradient(low = "grey", high = "red") +
  theme_bw()
```




```
ggplot(data = st,
       aes(x = log10(NABD_NRMSTORWS + 1),
           y = wtmp_mo)) +
  geom_hex(bins = 100) +
  scale_fill_gradient(low = "grey", high = "red") +
  theme_bw() +
  geom_vline(xintercept = 4.7)
```




```
ggplot(data = st,
       aes(x = log10(nhdflow+1),
           y = wtmp_mo)) +
  geom_hex(bins = 100) +
  scale_fill_gradient(low = "grey", high = "red") +
  theme_bw()
```




```
formula <-
  wtmp_mo ~
  tmeanPRISM*month +
  tmeanPRISM*PCTOWXXXXWS*I(log10(NABD_NRMSTORWS+1))*large_dam*month +
  tmeanPRISM*ELEVCAT*month +
  tmeanPRISM*BFIWS*month +
  tmeanPRISM*I(log10(PCTCROPXXXXWS+1))*month +
  tmeanPRISM*WTDEPWS*month +
  tmeanPRISM*PCTFSTXXXXWSRP100*month +
  tmeanPRISM*PCTURBXXXXWS*month +
  tmeanPRISM*SANDWS*month +
  tmeanPRISM*WETINDEXWS*month +
  tmeanPRISM*I(log10(nhdflow+1))*month +
  tmeanPRISM*RUNOFFWS*month +
  tmeanPRISM*CAOWS*month +
  tmeanPRISM*BFIWS*month +
  tmeanPRISM*pptPRISM*month +
  (1|COMID)

initial.lmer <- lmerTest::lmer(formula,
                               data = st)

reduced <- lmerTest::step(initial.lmer)
```

```r
lmer_formula <-
  lmerTest::get_model(reduced) %>%
  formula() %>%
  update(. ~ .
         -(1 | COMID))

initial.lm <-
  lm(formula = update(formula, . ~ . - (1 | COMID)),
     data = st)

initial.stepaic <-
  MASS::stepAIC(initial.lm,
                direction = "both",
                trace = FALSE)

aic_formula <-
  formula(initial.stepaic) %>%
  update(. ~ .
         -(1 | COMID))

formula <-
  formula %>%
  update(. ~ .
         -(1 | COMID))

write_rds(lmer_formula, '../data/base_lmer_formula.rds')
write_rds(aic_formula, '../data/base_formula_stepaic.rds')
write_rds(formula, '../data/fullset_formula.rds')
```

Summer Model Assessment

Libraries & Setup

- Documents function to conduct 10-fold cross validation of stream temperature data.
- Leave-one-out cross validation is not adequate because of repeated measurements at stations.
- This code withholds all sites with the same location ID (COMID) from the fold that contains the ID.
- Doing so, prevents repeated measurements at the same site from being included in both training and testing data.

▼ Code

```
library(sf)
library(tidyverse)
library(spmodel)
library(data.table)
library(ggplot2)
library(caret)
library(knitr)
library(parallel)

# Read in stream temperature data
st <-
  readr::read_rds('../data/summer_data.2024.08.08.rds') %>%
  dplyr::mutate(month = as.character(month),
                COMID = as.character(COMID)) %>%
  na.omit() %>%
  dplyr::ungroup() %>%
  dplyr::mutate(log10_dam = log10(NABD_NRMSTORWS + 1),
                large_dam = ifelse(log10_dam > 5, 1, 0) %>%
                  as.character(),
                no_dam = ifelse(log10_dam == 0, 1, 0) %>%
                  as.character())

# Create list of unique COMIDs in from stream temperature data
siteids <-
  st %>%
  dplyr::ungroup() %>%
  sf::st_drop_geometry() %>%
  dplyr::select(COMID) %>%
  dplyr::distinct() %>%
  dplyr::pull()

# Read in candidate models (formulas)
lmer_formula <- read_rds('../data/base_lmer_formula.rds')
aic_formula <- read_rds('../data/base_formula_stepaic.rds')
fullset_formula <- read_rds('../data/fullset_formula.rds')
```

```r
# Function to calculate RMSE from observed and predicted values
rmse <- function(observed, predicted){
  sqrt(sum((predicted - observed)^2) / length(observed))
}

# Function to calculate median (or other quantile) absolute error
quant_ae <- function(observed, predicted, quantile){
  quantile(abs(observed - predicted), probs = quantile)
}

# Create folds or assessment of models
set.seed(20240731)
folds <-caret::createFolds(siteids)

# Function to generate 10-fold cross validated assessments of candidate models
foldcv <- function(x, siteids, st, formula, spcov, rando = ~ (1 | COMID)){

  library(dplyr); library(sf); library(spmodel)

  y <- siteids[x]

  # Generate training dataset from siteids
  train <-
    st %>%
    dplyr::filter(!(COMID %in% y))

  # Generate test datset from siteids
  test <-
    st %>%
    dplyr::filter(COMID %in% y)

  # Create spatial model
  model <- spmodel::splm(formula,
                         data = train,
                         spcov_type = spcov,
                         random = rando,
                         local = TRUE)

  # predict to test data
  tmp <- predict(model,
                 newdata = test,
                 local = TRUE)

  # Add predicted column to test data
  test <-
    test %>%
    dplyr::mutate(predicted = tmp) %>%
    sf::st_drop_geometry() %>%
    dplyr::select(COMID, month, year, wtmp_mo, predicted)
  return(test)
```

```
}

performance <- function(x){
  rmse <-
    x %>%
    summarise(rmse = rmse(wtmp_mo, predicted))
  mdae <-
    x %>%
    summarise(mdae = quant_ae(wtmp_mo, predicted, 0.5))
  cor2 <-
    cor(x$wtmp_mo, x$predicted)^2

  out <- data.frame(rmse = rmse, mdae = mdae, cor2 = cor2)
  return(out)
}

model_performance <- list()
cl <- parallel::makeCluster(30)
```

Model Assessment

Fullset model (no variable selection, non-spatial)

▼ Code

```
formula <- fullset_formula

eval <-
  parLapply(cl,
            folds,
            foldcv,
            siteids = siteids,
            st = st,
            formula = formula,
            spcov = "none") %>%
  bind_rows()
#parallel::stopCluster(cl)

model_performance[[1]] <- performance(eval)
```

Fullset model (no variable selection, spatial)

▼ Code

```
formula <- fullset_formula

eval <-
  parLapply(cl,
```

```
            folds,
            foldcv,
            siteids = siteids,
            st = st,
            formula = formula,
            spcov = "exponential") %>%
  bind_rows()

model_performance[[2]] <- performance(eval)
```

lmerTest::step() model (non-spatial)

▼ Code

```
formula <- lmer_formula

eval <-
  parLapply(cl,
            folds,
            foldcv,
            siteids = siteids,
            st = st,
            formula = formula,
            spcov = "none") %>%
  bind_rows()

model_performance[[3]] <- performance(eval)
```

lmerTest::step() model (spatial)

▼ Code

```
formula <- lmer_formula

eval <-
  parLapply(cl,
            folds,
            foldcv,
            siteids = siteids,
            st = st,
            formula = formula,
            spcov = "exponential") %>%
  bind_rows()

model_performance[[4]] <- performance(eval)
```

MASS::stepAIC() model (non-spatial)

▼ Code

```
formula <- aic_formula

eval <-
  parLapply(cl,
            folds,
            foldcv,
            siteids = siteids,
            st = st,
            formula = formula,
            spcov = "none") %>%
  bind_rows()

model_performance[[5]] <- performance(eval)
```

MASS::stepAIC() model (spatial)

▼ Code

```
formula <- aic_formula

eval <-
  parLapply(cl,
            folds,
            foldcv,
            siteids = siteids,
            st = st,
            formula = formula,
            spcov = "exponential") %>%
  bind_rows()

model_performance[[6]] <- performance(eval)
```

Random Forest Models

- Includes standard random forest and random forest + spatial model of residuals

▼ Code

```
formula <-
  wtmp_mo ~
  tmeanPRISM+
  month +
  NABD_NRMSTORWS +
  ELEVCAT +
  BFIWS +
  PCTCROPXXXXWS +
  WTDEPWS +
  PCTFSTXXXXWSRP100 +
  PCTURBXXXXWS +
```

```
  SANDWS +
  WETINDEXWS +
  nhdflow +
  RUNOFFWS +
  CAOWS +
  BFIWS +
  pptPRISM

foldcv_rf <- function(x, siteids, st, formula){

  library(dplyr); library(sf); library(spmodel)

  y <- siteids[x]

  train <-
    st %>%
    dplyr::filter(!(COMID %in% y)) %>%
    dplyr::mutate(lon = sf::st_coordinates(.)[,1],
                  lat = sf::st_coordinates(.)[,2]) %>%
    sf::st_drop_geometry()
  test <-
    st %>%
    dplyr::filter(COMID %in% y) %>%
    dplyr::mutate(lon = sf::st_coordinates(.)[,1],
                  lat = sf::st_coordinates(.)[,2]) %>%
    sf::st_drop_geometry()
  model <- ranger::ranger(formula,
                          data=train)

  train$prediction <- model$predictions
  train$resids <- train$wtmp_mo - train$prediction
  train.splm <- splm(resids ~ 1,
                     data = train,
                     spcov_type = "exponential",
                     xcoord = lon,
                     ycoord = lat,
                     local = TRUE)

  test$predicted_rf <- predict(model, data = test)$prediction
  test$predicted_sprf <- test$predicted_rf + predict(train.splm,
                                                     newdata = test,
                                                     local = TRUE)

  test <-
    test %>%
    #sf::st_drop_geometry() %>%
    dplyr::select(COMID, month, year, wtmp_mo, predicted_rf, predicted_sprf)
  return(test)
}

eval_rf <-
```

```
  parLapply(cl,
            folds,
            foldcv_rf,
            siteids = siteids,
            st = st,
            formula = formula) %>%
  bind_rows()

# Non-spatial
rmserf <- eval_rf %>%
  #group_by(month) %>%
  summarise(rmse = rmse(wtmp_mo, predicted_rf))

mdaerf <- eval_rf %>%
  #group_by(month) %>%
  summarise(mdae = quant_ae(wtmp_mo, predicted_rf, 0.5))

cor2rf <-
  cor(eval_rf$wtmp_mo, eval_rf$predicted_rf)^2

rfns <- data.frame(rmse = rmserf, mdae = mdaerf, cor2 = cor2rf)

# Spatial
rmserf <- eval_rf %>%
  #group_by(month) %>%
  summarise(rmse = rmse(wtmp_mo, predicted_sprf))

mdaerf <- eval_rf %>%
  #group_by(month) %>%
  summarise(mdae = quant_ae(wtmp_mo, predicted_sprf, 0.5))

cor2rf <-
  cor(eval_rf$wtmp_mo, eval_rf$predicted_sprf)^2

sprf <- data.frame(rmse = rmserf, mdae = mdaerf, cor2 = cor2rf)
```

Combined model assessments

The models have almost identical performances, but the model selected with MASS::stepAIC() achieves this performance with far fewer parameters. We, therefore, selected this model as our final model.

▼ Code

```
model_performance <-
  model_performance %>%
  bind_rows(rfns, sprf)


model <- c(
  'fullset non-spatial',
```

```
  'fullset spatial',
  'lmerTest non-spatial',
  'lmerTest spatial',
  'stepAIC non-spatial',
  'stepAIC spatial',
  'random forest non-spatial',
  'random forest spatial')

model_performance <-
  tibble(model=model, model_performance) %>%
  arrange(rmse)

kable(model_performance)
```

model	rmse	mdae	cor2
random forest non-spatial	2.371226	1.123107	0.8047128
random forest spatial	2.371255	1.122805	0.8047063
fullset spatial	2.500423	1.223853	0.7809956
lmerTest spatial	2.502277	1.223105	0.7806449
stepAIC spatial	2.522697	1.229727	0.7770130
lmerTest non-spatial	2.834327	1.450812	0.7185669
stepAIC non-spatial	2.837282	1.452145	0.7179759
fullset non-spatial	2.842531	1.455480	0.7169222

Final model

▼ Code

```
formula <- aic_formula

eval_final <-
  parLapply(cl,
            folds,
            foldcv,
            siteids = siteids,
            st = st,
            formula = formula,
            spcov = "exponential") %>%
  bind_rows()
parallel::stopCluster(cl)

final.rmse <- eval_final %>%
  summarise(rmse = rmse(wtmp_mo, predicted)) %>%
  pull(rmse)
```

```r
final.mdae <- eval_final %>%
  summarise(mdae = quant_ae(wtmp_mo, predicted, 0.5)) %>%
  pull(mdae)

final.cor2 <- cor(eval_final$wtmp_mo, eval_final$predicted)^2

final.cor2 <- c(
  final.cor2,

  eval_final %>%
    summarise(wtmp_mo = mean(wtmp_mo),
              predicted = mean(predicted),
              .by = c(COMID, year)) %>%
    summarise(cor2 = cor(wtmp_mo, predicted)) %>%
    pull(cor2),

  eval_final %>%
    summarise(wtmp_mo = mean(wtmp_mo),
              predicted = mean(predicted),
              .by = c(COMID)) %>%
    summarise(cor2 = cor(wtmp_mo, predicted)) %>%
    pull(cor2)
)

final.rmse <- c(
  final.rmse,

  eval_final %>%
    summarise(wtmp_mo = mean(wtmp_mo),
              predicted = mean(predicted),
              .by = c(COMID, year)) %>%
    summarise(rmse = rmse(wtmp_mo, predicted)) %>%
    pull(rmse),

  eval_final %>%
    summarise(wtmp_mo = mean(wtmp_mo),
              predicted = mean(predicted),
              .by = c(COMID)) %>%
    summarise(rmse = rmse(wtmp_mo, predicted)) %>%
    pull(rmse)
)

final.mdae <- c(
  final.mdae,

  eval_final %>%
    summarise(wtmp_mo = mean(wtmp_mo),
              predicted = mean(predicted),
              .by = c(COMID, year)) %>%
    summarise(mdae = quant_ae(wtmp_mo, predicted, 0.5)) %>%
    pull(mdae),
```

```
  eval_final %>%
    summarise(wtmp_mo = mean(wtmp_mo),
              predicted = mean(predicted),
              .by = c(COMID)) %>%
    summarise(mdae = quant_ae(wtmp_mo, predicted, 0.5)) %>%
    pull(mdae)
)

n <- c(
  nrow(eval_final),

  eval_final %>%
    summarise(wtmp_mo = mean(wtmp_mo),
              predicted = mean(predicted),
              .by = c(COMID, year)) %>%
    nrow(),

  eval_final %>%
    summarise(wtmp_mo = mean(wtmp_mo),
              predicted = mean(predicted),
              .by = c(COMID)) %>%
    nrow()
)

period <- c(
  'YEAR-MONTH-COMID',
  'YEAR-COMID',
  'COMID'
)

definition <- c(
  'Prediction of July or August temperatures within years at a COMID',
  'Predictions of temperature within years (July/Aug averaged) at a COMID',
  'Predictions of temperature at a COMID (July/Aug/years averaged)'
)

model_performance2 <- tibble(period, definition, n, final.rmse, final.mdae, final.cor2)

final_splm <- spmodel::splm(formula,
                            data = st,
                            spcov_type = "exponential",
                            random = ~ (1 | COMID),
                            local = list(size = 50,
                                         method = "kmeans",
                                         parallel = TRUE,
                                         ncores = 30,
                                         var_adjust = "none"))

summary(final_splm)
```

```
Call:
spmodel::splm(formula = formula, data = st, spcov_type = "exponential",
    random = ~(1 | COMID), local = list(size = 50, method = "kmeans",
        parallel = TRUE, ncores = 30, var_adjust = "none"))

Residuals:
     Min       1Q   Median       3Q      Max
-17.3834  -1.3085   0.2102   1.5555  11.1393

Coefficients (fixed):
                                                                  Estimate
(Intercept)                                                     8.115e+00
tmeanPRISM                                                      7.330e-01
month8                                                         -1.694e+00
PCTOWXXXXWS                                                     2.110e-01
I(log10(NABD_NRMSTORWS + 1))                                    4.057e-01
large_dam1                                                     -3.247e+01
ELEVCAT                                                        -5.751e-04
BFIWS                                                          -1.058e-01
I(log10(PCTCROPXXXXWS + 1))                                     2.960e-01
WTDEPWS                                                         2.965e-02
PCTFSTXXXXWSRP100                                              -1.647e-02
PCTURBXXXXWS                                                   -1.733e-02
SANDWS                                                          4.096e-03
WETINDEXWS                                                      2.158e-03
I(log10(nhdflow + 1))                                           3.877e-01
RUNOFFWS                                                       -2.744e-03
CAOWS                                                          -3.094e-02
pptPRISM                                                       -1.630e-03
tmeanPRISM:month8                                               4.551e-02
tmeanPRISM:PCTOWXXXXWS                                         -2.974e-03
tmeanPRISM:I(log10(NABD_NRMSTORWS + 1))                        -5.810e-03
PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1))                        8.676e-03
tmeanPRISM:large_dam1                                           2.739e+00
PCTOWXXXXWS:large_dam1                                          3.555e+00
I(log10(NABD_NRMSTORWS + 1)):large_dam1                         6.693e+00
month8:large_dam1                                               8.218e-02
tmeanPRISM:ELEVCAT                                             -7.916e-05
month8:ELEVCAT                                                  4.194e-04
tmeanPRISM:BFIWS                                                2.103e-03
tmeanPRISM:I(log10(PCTCROPXXXXWS + 1))                         -2.668e-02
month8:I(log10(PCTCROPXXXXWS + 1))                             1.165e-01
tmeanPRISM:WTDEPWS                                             -1.365e-03
month8:WTDEPWS                                                 -2.033e-03
tmeanPRISM:PCTFSTXXXXWSRP100                                    1.057e-04
month8:PCTFSTXXXXWSRP100                                        7.831e-03
tmeanPRISM:PCTURBXXXXWS                                         7.488e-04
month8:PCTURBXXXXWS                                             3.895e-03
tmeanPRISM:SANDWS                                              -5.731e-04
```

tmeanPRISM:WETINDEXWS	1.798e-05
tmeanPRISM:I(log10(nhdflow + 1))	4.645e-03
month8:I(log10(nhdflow + 1))	1.046e-01
month8:RUNOFFWS	4.201e-04
tmeanPRISM:CAOWS	4.971e-04
month8:CAOWS	4.765e-02
tmeanPRISM:pptPRISM	-1.800e-05
tmeanPRISM:PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1))	-1.629e-03
tmeanPRISM:PCTOWXXXXWS:large_dam1	-9.771e-02
tmeanPRISM:I(log10(NABD_NRMSTORWS + 1)):large_dam1	-5.634e-01
PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1)):large_dam1	-7.365e-01
tmeanPRISM:month8:CAOWS	-1.864e-03
tmeanPRISM:PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1)):large_dam1	2.482e-02
	Std. Error
(Intercept)	3.274e+00
tmeanPRISM	1.304e-01
month8	2.395e-01
PCTOWXXXXWS	1.179e-01
I(log10(NABD_NRMSTORWS + 1))	1.181e-01
large_dam1	1.085e+01
ELEVCAT	3.674e-04
BFIWS	1.449e-02
I(log10(PCTCROPXXXXWS + 1))	4.549e-01
WTDEPWS	6.770e-03
PCTFSTXXXXWSRP100	1.054e-02
PCTURBXXXXWS	1.410e-02
SANDWS	1.590e-02
WETINDEXWS	3.203e-03
I(log10(nhdflow + 1))	2.179e-01
RUNOFFWS	2.662e-04
CAOWS	2.950e-02
pptPRISM	1.416e-03
tmeanPRISM:month8	8.545e-03
tmeanPRISM:PCTOWXXXXWS	4.700e-03
tmeanPRISM:I(log10(NABD_NRMSTORWS + 1))	4.890e-03
PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1))	4.154e-02
tmeanPRISM:large_dam1	4.591e-01
PCTOWXXXXWS:large_dam1	2.353e+00
I(log10(NABD_NRMSTORWS + 1)):large_dam1	2.015e+00
month8:large_dam1	4.588e-02
tmeanPRISM:ELEVCAT	1.580e-05
month8:ELEVCAT	4.696e-05
tmeanPRISM:BFIWS	5.696e-04
tmeanPRISM:I(log10(PCTCROPXXXXWS + 1))	1.791e-02
month8:I(log10(PCTCROPXXXXWS + 1))	4.062e-02
tmeanPRISM:WTDEPWS	2.655e-04
month8:WTDEPWS	5.212e-04
tmeanPRISM:PCTFSTXXXXWSRP100	4.324e-04
month8:PCTFSTXXXXWSRP100	1.086e-03
tmeanPRISM:PCTURBXXXXWS	5.557e-04
month8:PCTURBXXXXWS	1.280e-03

```
tmeanPRISM:SANDWS                                                 6.313e-04
tmeanPRISM:WETINDEXWS                                             1.256e-04
tmeanPRISM:I(log10(nhdflow + 1))                                  8.742e-03
month8:I(log10(nhdflow + 1))                                      1.971e-02
month8:RUNOFFWS                                                   6.289e-05
tmeanPRISM:CAOWS                                                  1.162e-03
month8:CAOWS                                                      1.892e-02
tmeanPRISM:pptPRISM                                               5.663e-05
tmeanPRISM:PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1))               1.674e-03
tmeanPRISM:PCTOWXXXXWS:large_dam1                                 1.003e-01
tmeanPRISM:I(log10(NABD_NRMSTORWS + 1)):large_dam1                8.514e-02
PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1)):large_dam1               4.204e-01
tmeanPRISM:month8:CAOWS                                           7.667e-04
tmeanPRISM:PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1)):large_dam1    1.790e-02
                                                                 z value Pr(>|z|)
(Intercept)                                                        2.478 0.013201
tmeanPRISM                                                         5.622 1.89e-08
month8                                                            -7.074 1.51e-12
PCTOWXXXXWS                                                        1.790 0.073532
I(log10(NABD_NRMSTORWS + 1))                                       3.435 0.000592
large_dam1                                                        -2.992 0.002774
ELEVCAT                                                           -1.566 0.117421
BFIWS                                                             -7.299 2.89e-13
I(log10(PCTCROPXXXXWS + 1))                                        0.651 0.515206
WTDEPWS                                                            4.380 1.19e-05
PCTFSTXXXXWSRP100                                                 -1.563 0.118034
PCTURBXXXXWS                                                      -1.229 0.219000
SANDWS                                                             0.258 0.796771
WETINDEXWS                                                         0.674 0.500405
I(log10(nhdflow + 1))                                              1.779 0.075216
RUNOFFWS                                                         -10.308  < 2e-16
CAOWS                                                             -1.049 0.294273
pptPRISM                                                          -1.151 0.249771
tmeanPRISM:month8                                                  5.326 1.00e-07
tmeanPRISM:PCTOWXXXXWS                                            -0.633 0.526821
tmeanPRISM:I(log10(NABD_NRMSTORWS + 1))                           -1.188 0.234823
PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1))                           0.209 0.834535
tmeanPRISM:large_dam1                                              5.965 2.44e-09
PCTOWXXXXWS:large_dam1                                             1.511 0.130838
I(log10(NABD_NRMSTORWS + 1)):large_dam1                            3.322 0.000893
month8:large_dam1                                                  1.791 0.073277
tmeanPRISM:ELEVCAT                                                -5.010 5.44e-07
month8:ELEVCAT                                                     8.932  < 2e-16
tmeanPRISM:BFIWS                                                   3.693 0.000222
tmeanPRISM:I(log10(PCTCROPXXXXWS + 1))                            -1.490 0.136266
month8:I(log10(PCTCROPXXXXWS + 1))                                 2.869 0.004115
tmeanPRISM:WTDEPWS                                                -5.140 2.75e-07
month8:WTDEPWS                                                    -3.902 9.56e-05
tmeanPRISM:PCTFSTXXXXWSRP100                                       0.245 0.806824
month8:PCTFSTXXXXWSRP100                                           7.211 5.56e-13
tmeanPRISM:PCTURBXXXXWS                                            1.347 0.177877
```

```
month8:PCTURBXXXXWS                                          3.043 0.002341
tmeanPRISM:SANDWS                                           -0.908 0.363981
tmeanPRISM:WETINDEXWS                                        0.143 0.886187
tmeanPRISM:I(log10(nhdflow + 1))                             0.531 0.595139
month8:I(log10(nhdflow + 1))                                 5.307 1.12e-07
month8:RUNOFFWS                                              6.680 2.40e-11
tmeanPRISM:CAOWS                                             0.428 0.668851
month8:CAOWS                                                 2.519 0.011777
tmeanPRISM:pptPRISM                                         -0.318 0.750629
tmeanPRISM:PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1))         -0.973 0.330485
tmeanPRISM:PCTOWXXXXWS:large_dam1                           -0.974 0.329867
tmeanPRISM:I(log10(NABD_NRMSTORWS + 1)):large_dam1          -6.617 3.67e-11
PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1)):large_dam1         -1.752 0.079778
tmeanPRISM:month8:CAOWS                                     -2.431 0.015069
tmeanPRISM:PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1)):large_dam1   1.387 0.165589

(Intercept)                                                 *
tmeanPRISM                                                  ***
month8                                                      ***
PCTOWXXXXWS                                                 .
I(log10(NABD_NRMSTORWS + 1))                                ***
large_dam1                                                  **
ELEVCAT
BFIWS                                                       ***
I(log10(PCTCROPXXXXWS + 1))
WTDEPWS                                                     ***
PCTFSTXXXXWSRP100
PCTURBXXXXWS
SANDWS
WETINDEXWS
I(log10(nhdflow + 1))                                       .
RUNOFFWS                                                    ***
CAOWS
pptPRISM
tmeanPRISM:month8                                           ***
tmeanPRISM:PCTOWXXXXWS
tmeanPRISM:I(log10(NABD_NRMSTORWS + 1))
PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1))
tmeanPRISM:large_dam1                                       ***
PCTOWXXXXWS:large_dam1
I(log10(NABD_NRMSTORWS + 1)):large_dam1                     ***
month8:large_dam1                                           .
tmeanPRISM:ELEVCAT                                          ***
month8:ELEVCAT                                              ***
tmeanPRISM:BFIWS                                            ***
tmeanPRISM:I(log10(PCTCROPXXXXWS + 1))
month8:I(log10(PCTCROPXXXXWS + 1))                          **
tmeanPRISM:WTDEPWS                                          ***
month8:WTDEPWS                                              ***
tmeanPRISM:PCTFSTXXXXWSRP100
month8:PCTFSTXXXXWSRP100                                    ***
```

```
tmeanPRISM:PCTURBXXXXWS
month8:PCTURBXXXXWS                                          **
tmeanPRISM:SANDWS
tmeanPRISM:WETINDEXWS
tmeanPRISM:I(log10(nhdflow + 1))
month8:I(log10(nhdflow + 1))                                 ***
month8:RUNOFFWS                                              ***
tmeanPRISM:CAOWS
month8:CAOWS                                                 *
tmeanPRISM:pptPRISM
tmeanPRISM:PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1))
tmeanPRISM:PCTOWXXXXWS:large_dam1
tmeanPRISM:I(log10(NABD_NRMSTORWS + 1)):large_dam1           ***
PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1)):large_dam1          .
tmeanPRISM:month8:CAOWS                                      *
tmeanPRISM:PCTOWXXXXWS:I(log10(NABD_NRMSTORWS + 1)):large_dam1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Pseudo R-squared: 0.4016

Coefficients (exponential spatial covariance):
       de        ie     range
    3.216     1.024 25068.002

Coefficients (random effects):
1 | COMID
    2.476
```

▼ Code

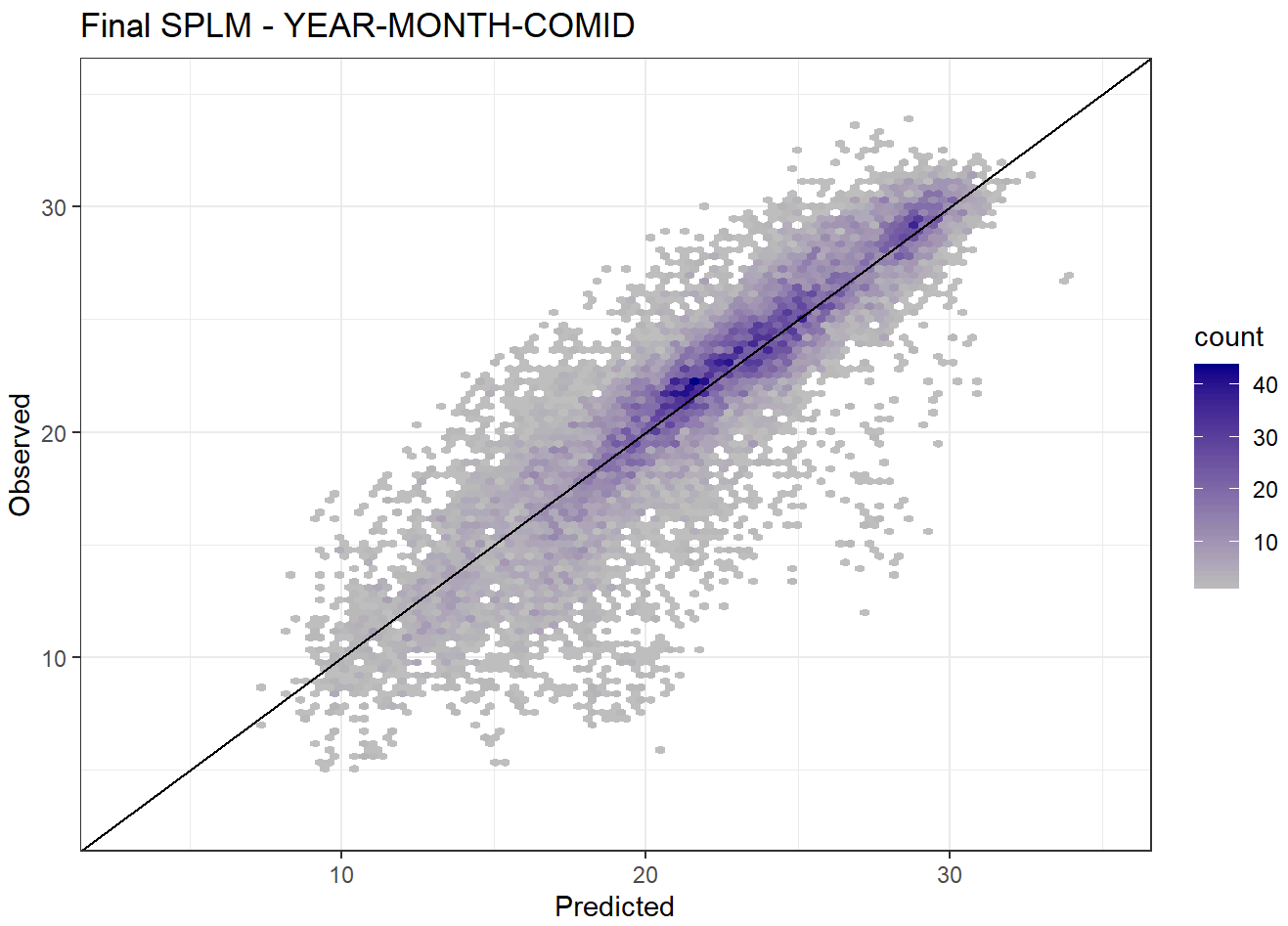
```
write_rds(final_splm, '../data/splm_selected.2024.08.08.rds')

kable(model_performance2)
```

period	definition	n	final.rmse	final.mdae	final.cor2
YEAR-MONTH-COMID	Prediction of July or August temperatures within years at a COMID	15403	2.499545	1.227332	0.7811244
YEAR-COMID	Predictions of temperature within years (July/Aug averaged) at a COMID	7918	2.446388	1.191278	0.8861542
COMID	Predictions of temperature at a COMID (July/Aug/years averaged)	2036	2.115167	1.068819	0.9082763

Plot of observed vs. predicted

▼ Code

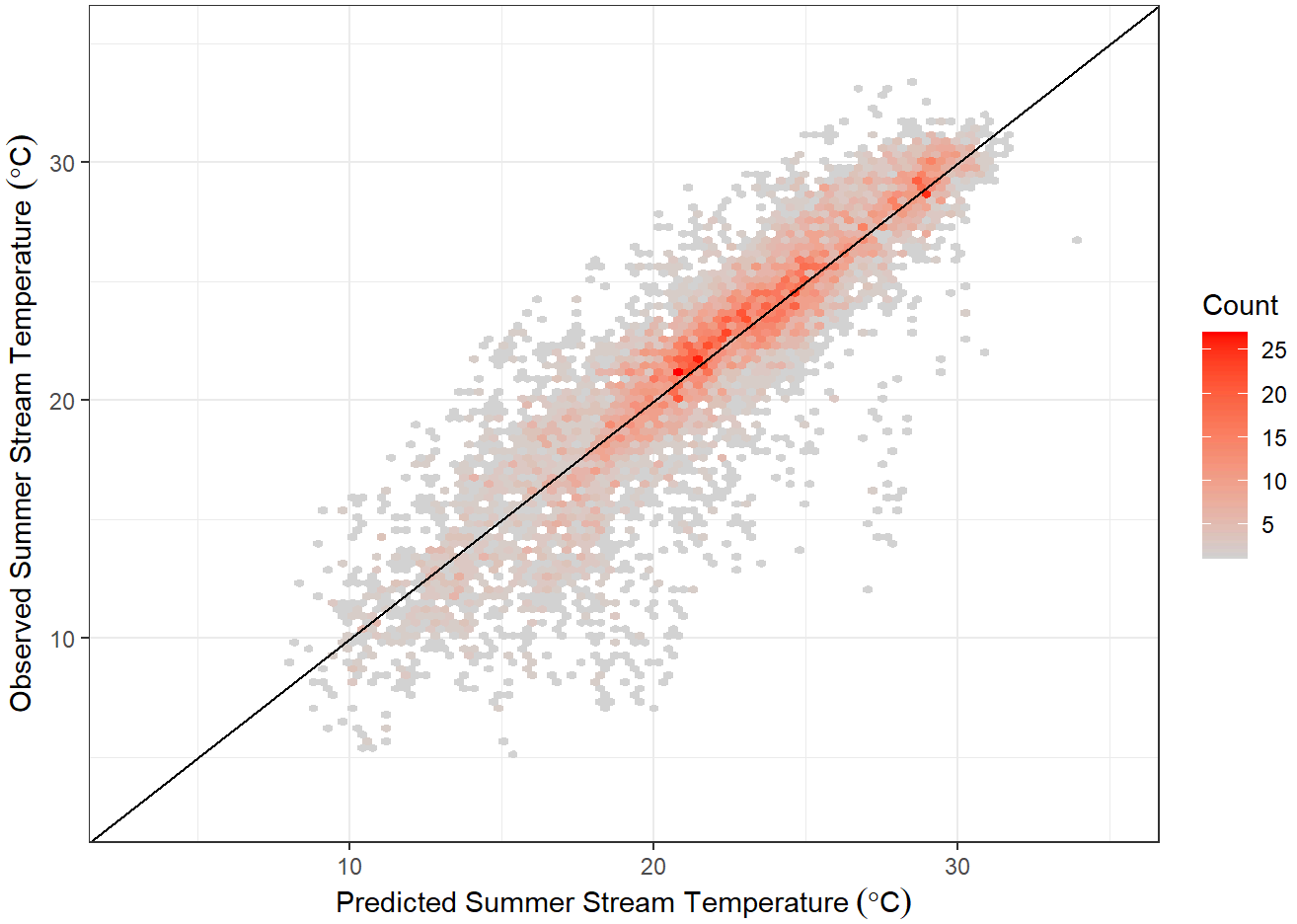
```
ggplot(data = eval_final,
       aes(x = predicted,
           y = wtmp_mo)) +
  geom_hex(bins = 100) +
  ggtitle("Final SPLM - YEAR-MONTH-COMID") +
  xlim(3, 35) + ylim(3, 35) +
  xlab('Predicted') + ylab('Observed') +
  scale_fill_gradient(low = "grey", high = "darkblue") +
  geom_abline(color='black') +
  theme_bw()
```



▼ Code

```
ggplot(data = eval_final %>%
         summarise(wtmp_mo = mean(wtmp_mo),
                   predicted = mean(predicted),
                   .by = c(COMID, year)),
       aes(x = predicted,
           y = wtmp_mo)) +
  geom_hex(bins = 100) +
  #ggtitle("Final Spatial Model") +
  xlim(3, 35) + ylim(3, 35) +
  xlab(expression("Predicted Summer Stream Temperature"~(degree*C))) +
  ylab(expression("Observed Summer Stream Temperature"~(degree*C))) +
```

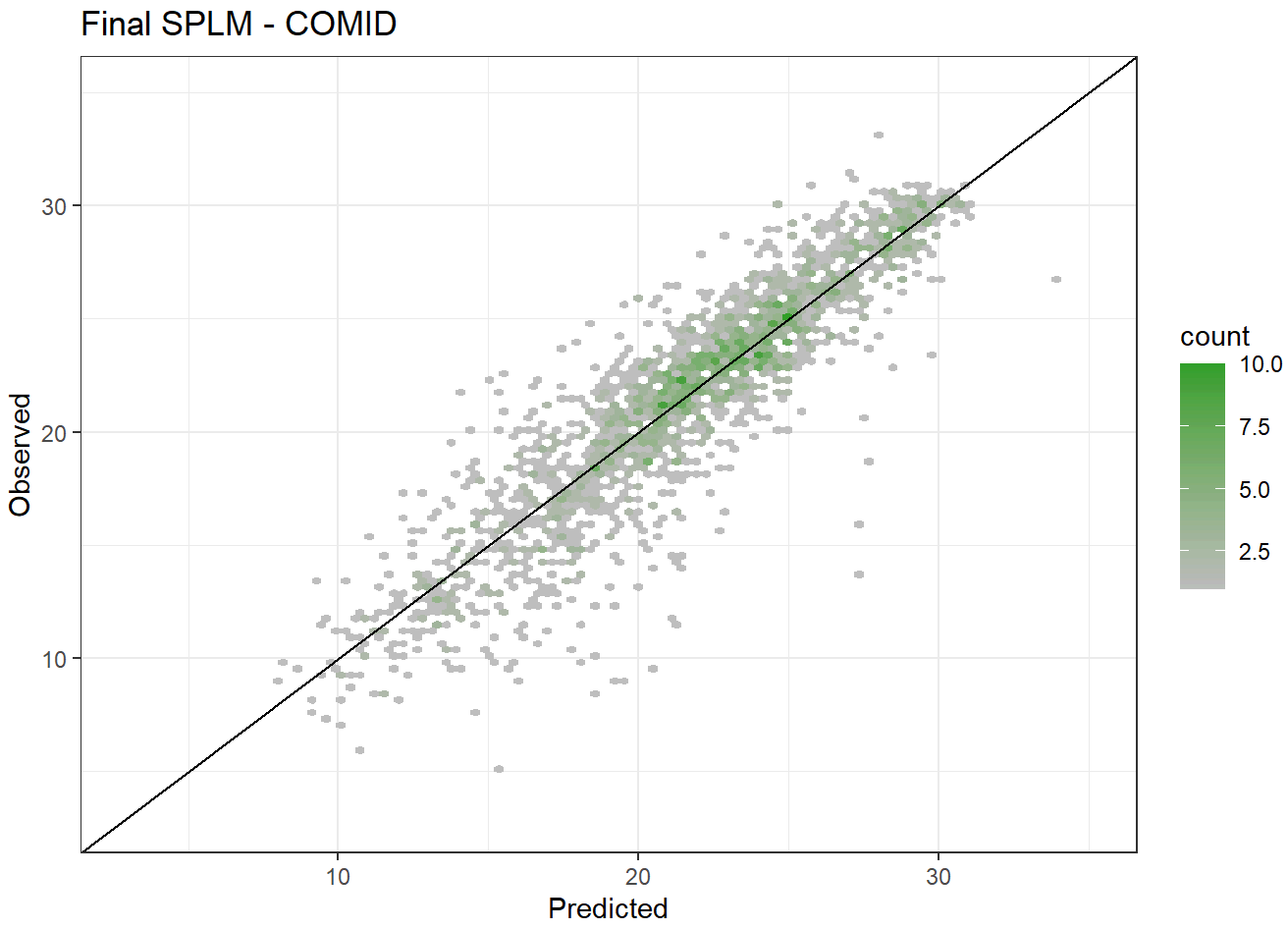
```
  #ylab('Observed Summer Stream Temperature') +
  scale_fill_gradient(name = 'Count',
                      low = "lightgrey",
                      high = "red") +
  geom_abline(color='black') +
  theme_bw()
```



▼ Code

```
ggsave(file = '../figures/observed-predicted.png',
       width = 5,
       height = 4,
       units = 'in',
       dpi = 1000)

ggsave(file = '../figures/observed-predicted.pdf',
       width = 5,
       height = 4,
       units = 'in')

ggplot(data = eval_final %>%
         summarise(wtmp_mo = mean(wtmp_mo),
                   predicted = mean(predicted),
                   .by = c(COMID)),
```

```
      aes(x = predicted,
          y = wtmp_mo)) +
geom_hex(bins = 100) +
ggtitle("Final SPLM - COMID") +
xlim(3, 35) + ylim(3, 35) +
xlab('Predicted') + ylab('Observed') +
scale_fill_gradient(low = "grey", high = "#33a02c") +
geom_abline(color='black') +
theme_bw()
```



Stream Temperature Model Application - NRSA/NAWQA

Libraries & Directories

```
library(StreamCatTools)
library(tidyverse)
library(data.table)
library(sf)
library(prism)
library(lubridate)
library(knitr)
library(spmodel)
library(readr)

# NHDPlus directory
nhd_dir <- 'C:/Users/RHill04/WorkFolder/GIS/NHDPlusV21/'
```

Prep data for Powell Center analysis

```
pts <- fread('../data/FishSiteCOMIDs.csv')  %>%
  left_join(
    read_rds(paste0(nhd_dir, 'cat-pour-points.rds'))  %>%
      rename(COMID = FEATUREID)  %>%
      dplyr::select(-AreaSqKM, -SOURCEFC, -GRIDCODE) %>%
      na.omit()
  ) %>%
  st_as_sf(coords = c('LON_DD', 'LAT_DD'),
           crs = 4269)

comids <- pts %>%
  pull(COMID) %>%
  na.omit()
```

NHDPlus flow metrics

Modeled monthly (July and August) flow estimates for each site (source: USGS)

```
nhd_dir <-
  paste0(nhd_dir, 'NHDPlusNationalData/NHDPlusV21_National_Seamless_Flattened_Lower48.gdb')

nhd_flow <-
  st_read(dsn = paste0(nhd_dir),
          layer = 'NHDFlowline_Network') %>%
  st_drop_geometry() %>%
  dplyr::select(COMID, QE_07, QE_08) %>%
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'nhdflow') %>%
  mutate(month = str_replace(tmpcol, 'QE_0', '') %>%
```

```
          as.integer()) %>%
  dplyr::select(-tmpcol)
```

```
Reading layer `NHDFlowline_Network' from data source

`C:\Users\RHill04\WorkFolder\GIS\NHDPlusV21\NHDPlusNationalData\NHDPlusV21_National_Seamless_Flat
tened_Lower48.gdb'
  using driver `OpenFileGDB'
Simple feature collection with 2691339 features and 137 fields
Geometry type: MULTILINESTRING
Dimension:     XYZM
Bounding box:  xmin: -124.7332 ymin: 24.63052 xmax: -66.94983 ymax: 49.37661
z_range:       zmin: 0 zmax: 0
m_range:       mmin: -2.35e-05 mmax: 100
Geodetic CRS:  NAD83
```

StreamCat (sc) static metrics

Static watershed/local catchment metrics:

- Elevation (Cat)
- Calcium oxide content of underlying lithology (Ws)
- Base flow index (Ws)
- Water table depth (Ws)
- Watershed area (Ws)
- Runoff (ws)
- Sand soil content (Ws)
- Topographic wetness index (Ws)
- National Anthropogenic Barriers dam normal storage (screened dams of NID) (Ws)

```
# comids_missing <- flow$COMID %>%
#   na.omit()

#Pull in static watershed metrics
sc <-
  sc_get_data(metric = 'Runoff,Sand,WtDep,WetIndex,NABD_NRMSTOR,BFI,ELEV,CAO',
              aoi = 'catchment,watershed',
              comid = comids) %>%
  dplyr::select(COMID, ELEVCAT, CAOWS, BFIWS, WTDEPWS,
                WSAREASQKM, RUNOFFWS, SANDWS, WETINDEXWS,
                NABD_NRMSTORWS)
```

StreamCat Year-Specific NLCD data

Riparian forest cover (catchment)

```
riparian_forest <-
  sc_nlcd(year = '2001, 2004, 2006, 2008, 2011, 2013, 2016, 2019',
```

```
          aoi = 'riparian_watershed',
          comid = comids) %>%
  dplyr::select(COMID,
                grep('CONIF|DECID|MXFST', names(.))) %>%
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'PCTFSTXXXXWSRP100') %>%
  mutate(year = as.integer(
    str_replace_all(tmpcol, 'PCTMXFST|PCTDECID|PCTCONIF|WSRP100', ''))) %>%
  group_by(COMID, year) %>%
  summarise(PCTFSTXXXXWSRP100 = sum(PCTFSTXXXXWSRP100))
```

Crop cover (watershed)

```
crop <-
  sc_nlcd(year = '2001, 2004, 2006, 2008, 2011, 2013, 2016, 2019',
          aoi = 'watershed',
          comid = comids) %>%
  dplyr::select(COMID,
                grep('CROP', names(.))) %>%
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'PCTCROPXXXXWS') %>%
  mutate(year = as.integer(
    str_replace_all(tmpcol, 'PCTCROP|WS', ''))) %>%
  dplyr::select(-tmpcol)
```

Urban cover (watershed)

```
urban <-
  sc_nlcd(year = '2001, 2004, 2006, 2008, 2011, 2013, 2016, 2019',
          aoi = 'watershed',
          comid = comids) %>%
  dplyr::select(COMID,
                grep('PCTURBLO|PCTURBMD|PCTURBHI', names(.))) %>%
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'PCTURBXXXXWS') %>%
  mutate(year = as.integer(
    str_replace_all(tmpcol, 'PCTURBLO|PCTURBMD|PCTURBHI|WS', ''))) %>%
  group_by(COMID, year) %>%
  summarise(PCTURBXXXXWS = sum(PCTURBXXXXWS))
```

Lake/Reservoir (open water) in watershed (watershed)

Variable added to interact with reservoir size to account for stations that occur below natural lakes or man made reservoirs.

```
water <-
  sc_nlcd(year = '2001, 2004, 2006, 2008, 2011, 2013, 2016, 2019',
          aoi = 'watershed',
          comid = comids) %>%
  dplyr::select(COMID,
                grep('PCTOW', names(.))) %>%
```

```
  pivot_longer(!COMID, names_to = 'tmpcol', values_to = 'PCTOWXXXXWS') %>%
  mutate(year = as.integer(
    str_replace_all(tmpcol, 'PCTOW|WS', ''))) %>%
  group_by(COMID, year) %>%
  summarise(PCTOWXXXXWS = sum(PCTOWXXXXWS))
```

PRISM Climate Data

Air temperature

```
years <- 1990:2020

# Set the PRISM directory (creates directory in not present)
prism_set_dl_dir("../data/prism_data", create = TRUE)

# Download monthly PRISM rasters (tmean)
get_prism_monthlys('tmean',
                   years = years,
                   mon = 7:8,
                   keepZip = FALSE)
```

```
  |
  |                                                                      |   0%

  |
  |=                                                                     |   2%

  |
  |==                                                                    |   3%

  |
  |===                                                                   |   5%

  |
  |=====                                                                 |   6%

  |
  |======                                                                |   8%

  |
  |=======                                                               |  10%
```

```
|
|========                                                 |  11%

|
|=========                                                |  13%

|
|==========                                               |  15%

|
|===========                                              |  16%

|
|============                                             |  18%

|
|==============                                           |  19%

|
|===============                                          |  21%

|
|================                                         |  23%

|
|==================                                       |  24%

|
|===================                                      |  26%

|
|====================                                     |  27%

|
|======================                                   |  29%

|
|=======================                                  |  31%
```

```
|
|=======================                                               |  32%

|
|========================                                              |  34%

|
|=========================                                             |  35%

|
|==========================                                            |  37%

|
|===========================                                           |  39%

|
|============================                                          |  40%

|
|=============================                                         |  42%

|
|==============================                                        |  44%

|
|===============================                                       |  45%

|
|================================                                      |  47%

|
|=================================                                     |  48%

|
|==================================                                    |  50%

|
|===================================                                   |  52%
```

```
|
|=================================== |  53%

|
|==================================== |  55%

|
|====================================== |  56%

|
|======================================= |  58%

|
|========================================= |  60%

|
|========================================== |  61%

|
|============================================ |  63%

|
|============================================= |  65%

|
|============================================== |  66%

|
|================================================ |  68%

|
|================================================= |  69%

|
|================================================== |  71%

|
|==================================================== |  73%
```

```
|
|================================================                  |  74%

|
|=================================================                 |  76%

|
|==================================================                |  77%

|
|====================================================              |  79%

|
|=====================================================             |  81%

|
|======================================================            |  82%

|
|========================================================          |  84%

|
|=========================================================         |  85%

|
|===========================================================       |  87%

|
|============================================================      |  89%

|
|=============================================================     |  90%

|
|===============================================================   |  92%

|
|================================================================  |  94%
```

```
  |
  |=================================================================    |  95%

  |
  |=================================================================    |  97%

  |
  |=================================================================    |  98%

  |
  |=================================================================    | 100%
```

```
tmn <- pd_stack((prism_archive_subset("tmean","monthly",
                                      years = years,
                                      mon = 7:8)))

# Extract tmean at sample points and massage data
tmn <- terra::extract(tmn,
                      # Transform pts to CRS of PRISM on the fly
                      pts %>%
                        st_transform(crs = st_crs(tmn))) %>%

  # Add site IDs to extracted values
  data.frame(COMID = pts$COMID, .) %>%

  # Remove front and back text from PRISM year/month in names
  rename_with( ~ stringr::str_replace_all(., 'PRISM_tmean_stable_4kmM3_|_bil', '')) %>%

  # Pivot to long table and calle column tmeanPRISM
  pivot_longer(!COMID, names_to = 'year_month',
               values_to = 'tmeanPRISM') %>%

  # Create new column of year
  mutate(year = year(ym(year_month)),
         month = month(ym(year_month))) %>%

  dplyr::select(-year_month)
```

Precipitation

```
get_prism_monthlys('ppt',
                   years = years,
                   mon = 7:8,
                   keepZip = FALSE)
```

```
|
|                                                                      |   0%

|
|=                                                                     |   2%

|
|==                                                                    |   3%

|
|===                                                                   |   5%

|
|=====                                                                 |   6%

|
|======                                                                |   8%

|
|=======                                                               |  10%

|
|========                                                              |  11%

|
|=========                                                             |  13%

|
|==========                                                            |  15%

|
|============                                                          |  16%

|
|=============                                                         |  18%

|
|==============                                                        |  19%
```

```
|
|===============                                                   |  21%

|
|================                                                  |  23%

|
|=================                                                 |  24%

|
|==================                                                |  26%

|
|===================                                               |  27%

|
|====================                                              |  29%

|
|======================                                            |  31%

|
|=======================                                           |  32%

|
|========================                                          |  34%

|
|=========================                                         |  35%

|
|==========================                                        |  37%

|
|===========================                                       |  39%

|
|============================                                      |  40%
```

```
|
|============================                                     |  42%

|
|=============================                                    |  44%

|
|==============================                                   |  45%

|
|================================                                 |  47%

|
|=================================                                |  48%

|
|==================================                               |  50%

|
|====================================                             |  52%

|
|=====================================                            |  53%

|
|======================================                           |  55%

|
|=======================================                          |  56%

|
|========================================                         |  58%

|
|=========================================                        |  60%

|
|==========================================                       |  61%
```

```
|
|========================================                              |  63%

|
|=========================================                             |  65%

|
|==========================================                            |  66%

|
|===========================================                           |  68%

|
|============================================                          |  69%

|
|=============================================                         |  71%

|
|==============================================                        |  73%

|
|===============================================                       |  74%

|
|================================================                      |  76%

|
|=================================================                     |  77%

|
|==================================================                    |  79%

|
|===================================================                   |  81%

|
|====================================================                  |  82%
```

```
  |
  |=============================================================          |  84%

  |
  |==============================================================         |  85%

  |
  |================================================================       |  87%

  |
  |=================================================================      |  89%

  |
  |==================================================================     |  90%

  |
  |====================================================================   |  92%

  |
  |=====================================================================  |  94%

  |
  |======================================================================== |  95%

  |
  |========================================================================== |  97%

  |
  |============================================================================ |  98%

  |
  |==============================================================================| 100%
```

```
ppt <- pd_stack((prism_archive_subset("ppt","monthly",
                                      years = years,
                                      mon = 7:8)))

ppt <- terra::extract(ppt,
                      pts %>%
                        st_transform(crs = st_crs(ppt))) %>%
```

```
  data.frame(COMID = pts$COMID, .) %>%
  rename_with( ~ stringr::str_replace_all(., 'PRISM_ppt_stable_4kmM3_|_bil', '')) %>%
  pivot_longer(!COMID, names_to = 'year_month',
               values_to = 'pptPRISM') %>%
  mutate(year = year(ym(year_month)),
         month = month(ym(year_month))) %>%

  dplyr::select(-year_month)
```

Combine data for modeling

```
nlcd_years <-
  data.table(year = c(2001, 2004, 2006, 2008, 2011, 2013, 2016, 2019)) %>%
  mutate(merge = year) %>%
  setkeyv('merge')

st_years <-
  data.table(year = 1990:2020) %>%
  mutate(merge = year) %>%
  setkeyv('merge')

nearest <-
  nlcd_years[st_years, roll = 'nearest'] %>%
  dplyr::select(year, i.year) %>%
  rename(nlcd_year = year,
         year = i.year)

powell.data <- tmn %>%
  left_join(nearest, join_by(year)) %>%
  left_join(ppt,
            join_by(COMID, year, month)) %>%
  left_join(sc, join_by(COMID)) %>%
  left_join(riparian_forest,
            join_by(COMID == COMID,
                    nlcd_year == year)) %>%
  left_join(crop,
            join_by(COMID == COMID,
                    nlcd_year == year)) %>%
  left_join(urban,
            join_by(COMID == COMID,
                    nlcd_year == year)) %>%
  left_join(nhd_flow,
            join_by(COMID == COMID,
                    month == month)) %>%
  left_join(water,
            join_by(COMID == COMID,
                    nlcd_year == year)) %>%
  left_join(pts,
            join_by(COMID == COMID)) %>%
  mutate(month = as.character(month),
```

```
         COMID = as.character(COMID)) %>%
  st_as_sf(crs = 4269) %>%
  st_transform(crs = 5070)

# Grab values from nearest site for WTDEP NA value

ptna <- powell.data  %>%
  filter(is.na(WTDEPWS)) %>%
  dplyr::select(COMID) %>%
  distinct()

dist_matrix <- st_distance(pts)
names(dist_matrix) <- pts$COMID
rownames(dist_matrix) <- pts$COMID

nearest <- dist_matrix[, grep(ptna$COMID, names(dist_matrix))]
nearest <- data.frame(nearest) %>%
  distinct() %>%
  arrange(nearest) %>%
  slice_head(n=6) %>%
  row.names()

powell.data$WTDEPWS[powell.data$COMID == ptna$COMID] <- mean(powell.data$WTDEPWS[powell.data$COMID

# Write output file for modeling
write_rds(powell.data,
          file = '../data/powell_data.2024.07.31.rds',
          compress = "xz")
```

Predict stream temperatures

```
sp.mod <- read_rds('../data/splm_selected.2024.08.08.rds')

powell.pred <- predict(sp.mod,
                       newdata = powell.data,
                       se.fit = TRUE,
                       local = list(parallel = TRUE,
                                    ncores = 30))


outdf <-
  powell.data %>%
  dplyr::select(COMID, year, month) %>%
  mutate(wt_pred = powell.pred$fit,
         wt_se.fit = powell.pred$se.fit) %>%
  st_drop_geometry()

length(comids)

test <- powell.data %>%
```

```
  pull(COMID) %>%
  na.omit() %>% unique()
```

Write output file

```
write_rds(outdf,
          file = '../data/powell-long-term-water-temperature-predictions.2024.08.08.rds',
          compress = "xz")
```