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

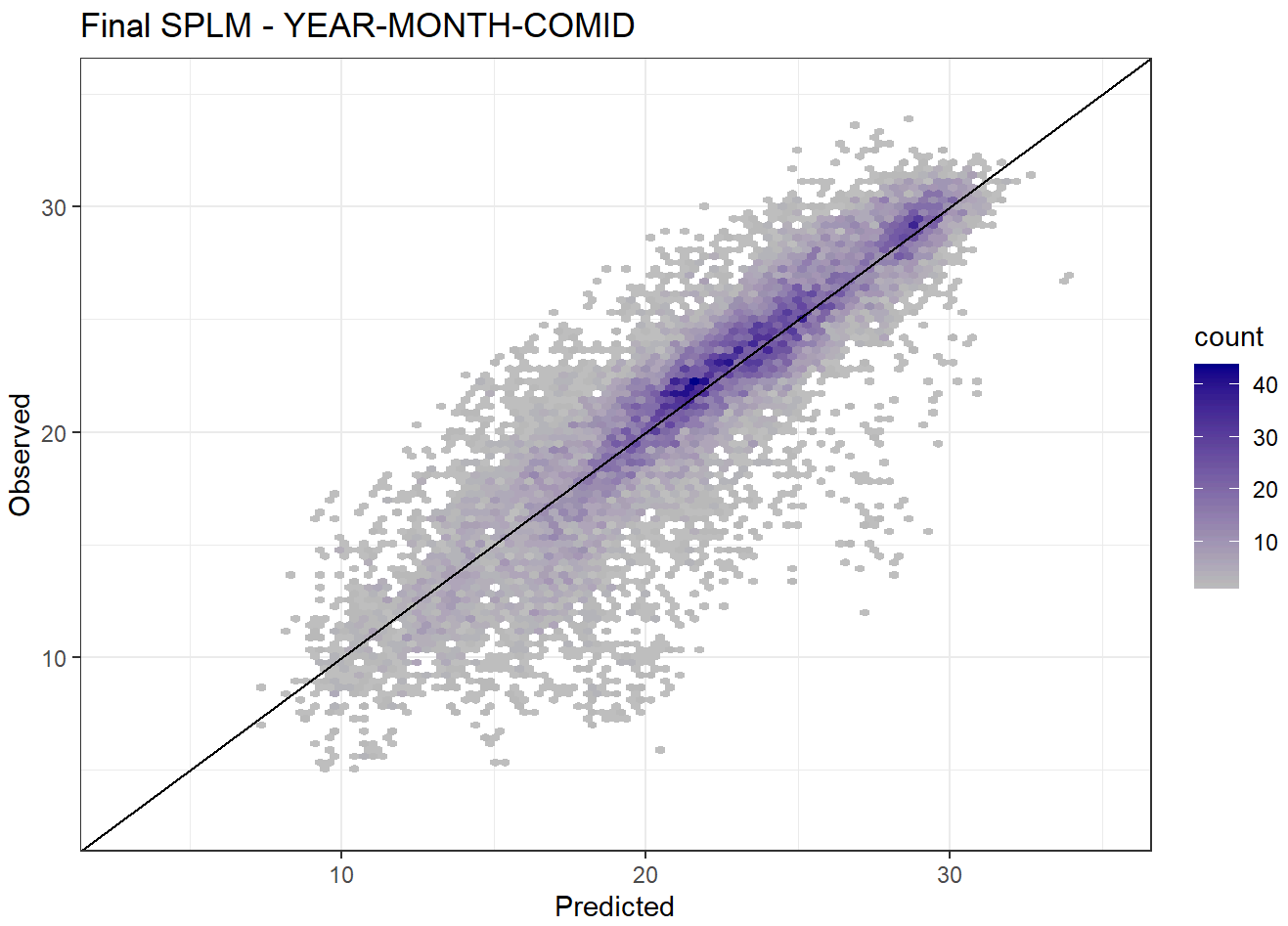
```
write_rds(final_splm, '../data/splm_selected.2024.08.08.rds')

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

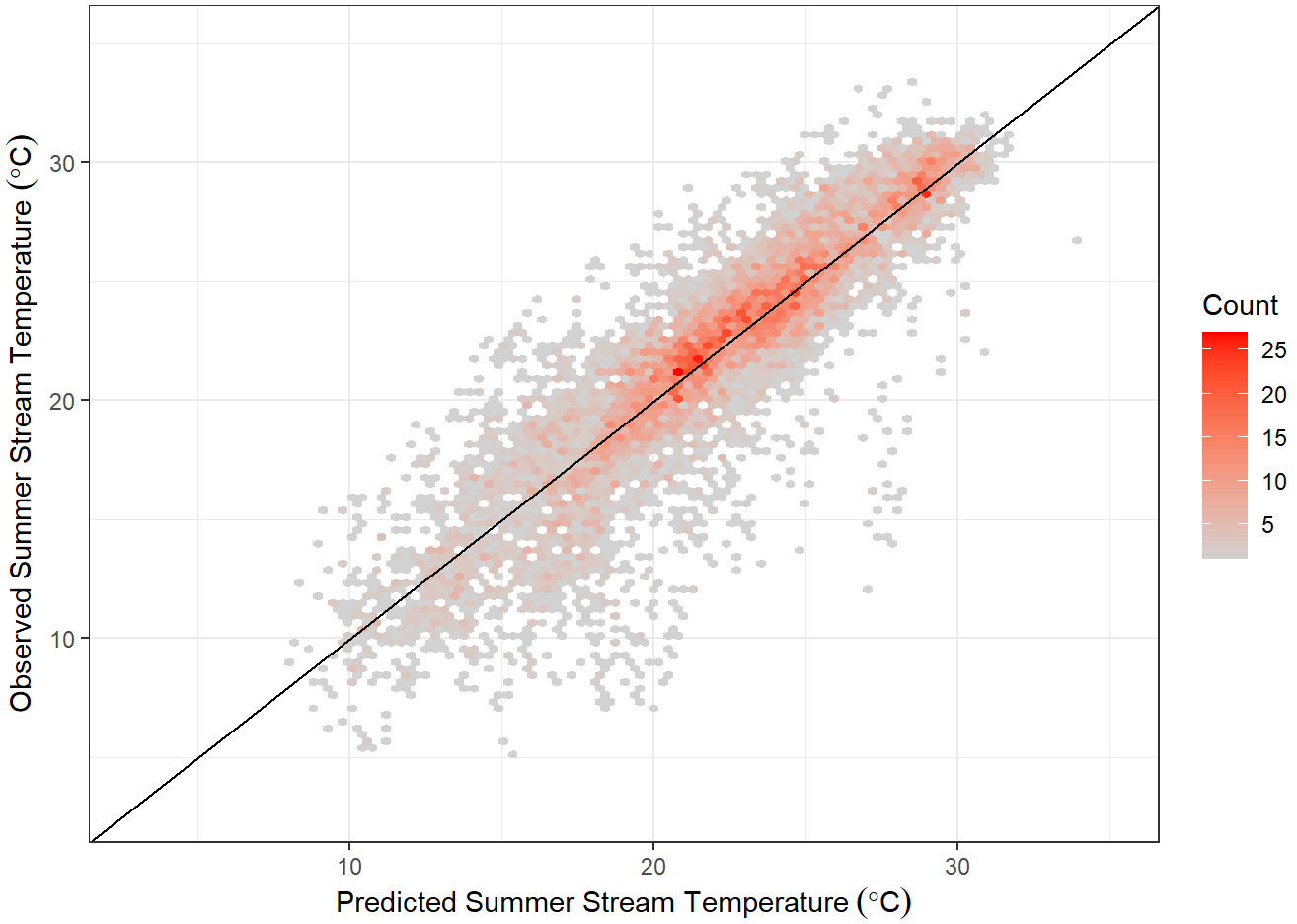
```
ggplot(data = eval_final,
       aes(x = predicted,
           y = wtmp_mo)) +
  geom_hex(bins = 100) +
  ggtitle("Final SPLM - YEAR-MONTH-COMID") +
  xlim(3, 35) + ylim(3, 35) +
  xlab('Predicted') + ylab('Observed') +
  scale_fill_gradient(low = "grey", high = "darkblue") +
  geom_abline(color='black')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