

Final Project

Bailey McNichol

2021 December 12

```
# Load Libraries
library(tidyverse)
library(sf)
library(tmap)
library(sp)
library(rgdal)
library(readxl)
library(daymetr)
library(lubridate)
library(purrr)
library(pals)
library(raster)

# Read in FIA data by state
ne <- read_excel("NE tree spp of interest.xlsx")
wi <- read_excel("WI tree spp of interest.xlsx")
mo <- read_excel("MO tree spp of interest.xlsx")
sd <- read_excel("SD tree spp of interest.xlsx")
oh <- read_excel("OH tree spp of interest.xlsx")
nd <- read_excel("ND tree spp of interest.xlsx")
mi <- read_excel("MI tree spp of interest.xlsx")
ks <- read_excel("KS tree spp of interest.xlsx")
ind <- read_excel("IN tree spp of interest.xlsx")
il <- read_excel("IL tree spp of interest.xlsx")
ia <- read_excel("IA tree spp of interest.xlsx")
mn <- read_excel("MN tree spp of interest.xlsx")

# Read in Common/Scientific Names for Each Spp
spp <- read_excel("sppNiobraraFIAMidwest.xlsx")

# Read in CSVs with Monthly Climate data for each state
ne_month <- read.csv("neMonthlyClim.csv")
wi_month <- read.csv("wiMonthlyClim.csv")
mo_month <- read.csv("moMonthlyClim.csv")
sd_month <- read.csv("sdMonthlyClim.csv")
oh_month <- read.csv("ohMonthlyClim.csv")
nd_month <- read.csv("ndMonthlyClim.csv")
mi_month <- read.csv("miMonthlyClim.csv")
ks_month <- read.csv("ksMonthlyClim.csv")
in_month <- read.csv("inMonthlyClim.csv")
il_month <- read.csv("ilMonthlyClim.csv")
```

```

ia_month <- read.csv("iaMonthlyClim.csv")
mn_month <- read.csv("mnMonthlyClim.csv")

## Process for Creating State Boundary Shapefile for
# Midwestern States only

# Read in shapefile with boundaries for all continental US
states <- sf::read_sf("./tl_2012_us_state/tl_2012_us_state.shp") %>% st_make_valid()

# Transform CRS to NAD83
state <- st_transform(states, "NAD83")

# Filter shapefile to only include Midwest states with State FIPS
midwestST <- state %>%
  dplyr::filter(STATEFP == "17" |
    STATEFP == "18" |
    STATEFP == "19" |
    STATEFP == "20" |
    STATEFP == "26" |
    STATEFP == "27" |
    STATEFP == "29" |
    STATEFP == "31" |
    STATEFP == "38" |
    STATEFP == "39" |
    STATEFP == "46" |
    STATEFP == "55")

# Remove ALAND and AWATER to reduce file size
mwS2 <- dplyr::select(midwestST, -ALAND, -AWATER)

# Write shapefile for mapping
sf::write_sf(mwS2, "./County Data/midwestSB.shp")

# Read in Shapefile with Midwest state boundaries
state <- sf::read_sf("./County Data/midwestSB.shp")

##### Extract Plot-level Lat/Longs (Averaged for plot) for obtaining
# weather data

##### Nebraska
# Average Lat/Longs for each plot for climate data
ne_ll <- ne %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                  plot_lon = mean(LON))
colnames(ne_ll) <- c("site", "lat", "lon")
#write.csv(ne_ll, "ne_latlong.csv", row.names=FALSE)

##### Wisconsin
# Average Lat/Longs for each plot for climate data
wi_ll <- wi %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),

```

```

        plot_lon = mean(LON))
colnames(wi_ll) <- c("site", "lat", "lon")
#write.csv(wi_ll, "wi_latlong.csv", row.names=FALSE)

##### Missouri
# Average Lat/Longs for each plot for climate data
mo_ll <- mo %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                  plot_lon = mean(LON))
colnames(mo_ll) <- c("site", "lat", "lon")
#write.csv(mo_ll, "mo_latlong.csv", row.names=FALSE)

##### South Dakota
# Average Lat/Longs for each plot for climate data
sd_ll <- sd %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                  plot_lon = mean(LON))
colnames(sd_ll) <- c("site", "lat", "lon")
#write.csv(sd_ll, "sd_latlong.csv", row.names=FALSE)

##### Ohio
# Average Lat/Longs for each plot for climate data
oh_ll <- oh %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                  plot_lon = mean(LON))
colnames(oh_ll) <- c("site", "lat", "lon")
#write.csv(oh_ll, "oh_latlong.csv", row.names=FALSE)

##### North Dakota
# Average Lat/Longs for each plot for climate data
nd_ll <- nd %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                  plot_lon = mean(LON))
colnames(nd_ll) <- c("site", "lat", "lon")
#write.csv(nd_ll, "nd_latlong.csv", row.names=FALSE)

##### Michigan
# Average Lat/Longs for each plot for climate data
mi_ll <- mi %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                  plot_lon = mean(LON))
colnames(mi_ll) <- c("site", "lat", "lon")
#write.csv(mi_ll, "mi_latlong.csv", row.names=FALSE)

##### Kansas
# Average Lat/Longs for each plot for climate data
ks_ll <- ks %>%
  group_by(PLOT) %>%

```

```

dplyr::summarise(., plot_lat = mean(LAT),
                 plot_lon = mean(LON))
colnames(ks_ll) <- c("site", "lat", "lon")
#write.csv(ks_ll, "ks_latlong.csv", row.names=FALSE)

##### Indiana
# Average Lat/Longs for each plot for climate data
ind_ll <- ind %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                 plot_lon = mean(LON))
colnames(ind_ll) <- c("site", "lat", "lon")
#write.csv(ind_ll, "ind_latlong.csv", row.names=FALSE)

##### Illinois
# Average Lat/Longs for each plot for climate data
il_ll <- il %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                 plot_lon = mean(LON))
colnames(il_ll) <- c("site", "lat", "lon")
#write.csv(il_ll, "il_latlong.csv", row.names=FALSE)

##### Iowa
# Average Lat/Longs for each plot for climate data
ia_ll <- ia %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                 plot_lon = mean(LON))
colnames(ia_ll) <- c("site", "lat", "lon")
#write.csv(ia_ll, "ia_latlong.csv", row.names=FALSE)

##### Minnesota
# Average Lat/Longs for each plot for climate data
mn_ll <- mn %>%
  group_by(PLOT) %>%
  dplyr::summarise(., plot_lat = mean(LAT),
                 plot_lon = mean(LON))
colnames(mn_ll) <- c("site", "lat", "lon")
#write.csv(mn_ll, "mn_latlong.csv", row.names=FALSE)

```

```
# Download Climate Data for each state from 1980-2020
```

```

### General function to extract climate data
get_daymet <- function(site, lat, lon){
  # Run download_daymet on each location
  temp_daymet <- download_daymet(site,
                                   lat,
                                   lon,
                                   start = 1980,
                                   end = 2020) %>%
  # Extract df with environmental vars from list
  .\$data %>%

```

```

# Convert to tibble
as_tibble() %>%
# Only include the variables you want
dplyr::select(., -dayl..s., -swe..kg.m.2.) %>%
# Give each its plot_id
mutate(site_id = site) %>%
# Get date from day of the year using lubridate
mutate(date = as.Date(paste(year, yday, sep = "-"), "%Y-%j")) %>%
# Add column with month
mutate(month = month(date, label = TRUE, abbr = FALSE))
return(temp_daymet)
}

##### Obtain Midwestern State FIA Plot Climate Data from DayMet
## Variables: Total daily precipitation (mm), max daily temperature (C),
# min daily temperature (C), daily average water vapor pressure (Pa)

##### Nebraska
ne_clim <- ne_ll %>% purrr::pmap_dfr(.,get_daymet)

##### Wisconsin
wi_clim <- wi_ll %>% purrr::pmap_dfr(.,get_daymet)

##### Missouri
mo_clim <- mo_ll %>% purrr::pmap_dfr(.,get_daymet)

##### South Dakota
sd_clim <- sd_ll %>% purrr::pmap_dfr(.,get_daymet)

##### Ohio
oh_clim <- oh_ll %>% purrr::pmap_dfr(.,get_daymet)

##### North Dakota
nd_clim <- nd_ll %>% purrr::pmap_dfr(.,get_daymet)

##### Michigan
mi_clim <- mi_ll %>% purrr::pmap_dfr(.,get_daymet)

##### Kansas
ks_clim <- ks_ll %>% purrr::pmap_dfr(.,get_daymet)

##### Indiana
in_clim <- ind_ll %>% purrr::pmap_dfr(.,get_daymet)

##### Illinois
il_clim <- il_ll %>% purrr::pmap_dfr(.,get_daymet)

##### Iowa
ia_clim <- ia_ll %>% purrr::pmap_dfr(.,get_daymet)

##### Minnesota
mn_clim <- mn_ll %>% purrr::pmap_dfr(.,get_daymet)

```

```

##### Calculate monthly average climate variables for Midwestern
# State FIA Plots from 1980-2020

##### Nebraska
ne_month <- ne_clim %>%
  group_by(site_id, year, month) %>%
  dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                  minTempY = mean(tmin..deg.c.),
                  maxTempY = mean(tmax..deg.c.),
                  meanSRadY = mean(srad..W.m.2.),
                  meanVPY = mean(vp..Pa.)) %>%
  group_by(site_id, month) %>%
  dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                  minTemp = mean(minTempY),
                  maxTemp = mean(maxTempY),
                  meanSR = mean(meanSRadY),
                  meanVP = mean(meanVPY))
#write.csv(ne_month, "neMonthlyClim.csv")

##### Wisconsin (no month variable)
wi_month <- wi_clim %>%
  group_by(site_id, year, month) %>%
  dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                  minTempY = mean(tmin..deg.c.),
                  maxTempY = mean(tmax..deg.c.),
                  meanSRadY = mean(srad..W.m.2.),
                  meanVPY = mean(vp..Pa.)) %>%
  group_by(site_id, month) %>%
  dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                  minTemp = mean(minTempY),
                  maxTemp = mean(maxTempY),
                  meanSR = mean(meanSRadY),
                  meanVP = mean(meanVPY))
#write.csv(wi_month, "wiMonthlyClim.csv")

##### Missouri
mo_month <- mo_clim %>%
  group_by(site_id, year, month) %>%
  dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                  minTempY = mean(tmin..deg.c.),
                  maxTempY = mean(tmax..deg.c.),
                  meanSRadY = mean(srad..W.m.2.),
                  meanVPY = mean(vp..Pa.)) %>%
  group_by(site_id, month) %>%
  dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                  minTemp = mean(minTempY),
                  maxTemp = mean(maxTempY),
                  meanSR = mean(meanSRadY),
                  meanVP = mean(meanVPY))
#write.csv(mo_month, "moMonthlyClim.csv")

##### South Dakota
sd_month <- sd_clim %>%

```

```

group_by(site_id, year, month) %>%
dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                 minTempY = mean(tmin..deg.c.),
                 maxTempY = mean(tmax..deg.c.),
                 meanSRadY = mean(srad..W.m.2.),
                 meanVPY = mean(vp..Pa.)) %>%
group_by(site_id, month) %>%
dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                 minTemp = mean(minTempY),
                 maxTemp = mean(maxTempY),
                 meanSR = mean(meanSRadY),
                 meanVP = mean(meanVPY))
#write.csv(sd_month, "sdMonthlyClim.csv")

##### Ohio
oh_month <- oh_clim %>%
  group_by(site_id, year, month) %>%
  dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                   minTempY = mean(tmin..deg.c.),
                   maxTempY = mean(tmax..deg.c.),
                   meanSRadY = mean(srad..W.m.2.),
                   meanVPY = mean(vp..Pa.)) %>%
  group_by(site_id, month) %>%
  dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                   minTemp = mean(minTempY),
                   maxTemp = mean(maxTempY),
                   meanSR = mean(meanSRadY),
                   meanVP = mean(meanVPY))
#write.csv(oh_month, "ohMonthlyClim.csv")

##### North Dakota
nd_month <- nd_clim %>%
  group_by(site_id, year, month) %>%
  dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                   minTempY = mean(tmin..deg.c.),
                   maxTempY = mean(tmax..deg.c.),
                   meanSRadY = mean(srad..W.m.2.),
                   meanVPY = mean(vp..Pa.)) %>%
  group_by(site_id, month) %>%
  dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                   minTemp = mean(minTempY),
                   maxTemp = mean(maxTempY),
                   meanSR = mean(meanSRadY),
                   meanVP = mean(meanVPY))
#write.csv(nd_month, "ndMonthlyClim.csv")

##### Michigan
mi_month <- mi_clim %>%
  group_by(site_id, year, month) %>%
  dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                   minTempY = mean(tmin..deg.c.),
                   maxTempY = mean(tmax..deg.c.),
                   meanSRadY = mean(srad..W.m.2.),

```

```

        meanVPY = mean(vp..Pa.)) %>%
group_by(site_id, month) %>%
dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                 minTemp = mean(minTempY),
                 maxTemp = mean(maxTempY),
                 meanSR = mean(meanSRadY),
                 meanVP = mean(meanVPY))
#write.csv(mi_month, "miMonthlyClim.csv")

##### Kansas
ks_month <- ks_clim %>%
group_by(site_id, year, month) %>%
dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                 minTempY = mean(tmin..deg.c.),
                 maxTempY = mean(tmax..deg.c.),
                 meanSRadY = mean(srad..W.m.2.),
                 meanVPY = mean(vp..Pa.)) %>%
group_by(site_id, month) %>%
dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                 minTemp = mean(minTempY),
                 maxTemp = mean(maxTempY),
                 meanSR = mean(meanSRadY),
                 meanVP = mean(meanVPY))
#write.csv(ks_month, "ksMonthlyClim.csv")

##### Indiana
in_month <- in_clim %>%
group_by(site_id, year, month) %>%
dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                 minTempY = mean(tmin..deg.c.),
                 maxTempY = mean(tmax..deg.c.),
                 meanSRadY = mean(srad..W.m.2.),
                 meanVPY = mean(vp..Pa.)) %>%
group_by(site_id, month) %>%
dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                 minTemp = mean(minTempY),
                 maxTemp = mean(maxTempY),
                 meanSR = mean(meanSRadY),
                 meanVP = mean(meanVPY))
#write.csv(in_month, "inMonthlyClim.csv")

##### Illinois
il_month <- il_clim %>%
group_by(site_id, year, month) %>%
dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                 minTempY = mean(tmin..deg.c.),
                 maxTempY = mean(tmax..deg.c.),
                 meanSRadY = mean(srad..W.m.2.),
                 meanVPY = mean(vp..Pa.)) %>%
group_by(site_id, month) %>%
dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                 minTemp = mean(minTempY),
                 maxTemp = mean(maxTempY),

```

```

        meanSR = mean(meanSRadY),
        meanVP = mean(meanVPY))
#write.csv(il_month, "ilMonthlyClim.csv")

##### Iowa
ia_month <- ia_clim %>%
  group_by(site_id, year, month) %>%
  dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                  minTempY = mean(tmin..deg.c.),
                  maxTempY = mean(tmax..deg.c.),
                  meanSRadY = mean(srad..W.m.2.),
                  meanVPY = mean(vp..Pa.)) %>%
  group_by(site_id, month) %>%
  dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                  minTemp = mean(minTempY),
                  maxTemp = mean(maxTempY),
                  meanSR = mean(meanSRadY),
                  meanVP = mean(meanVPY))
#write.csv(ia_month, "iaMonthlyClim.csv")

##### Minnesota
mn_month <- mn_clim %>%
  group_by(site_id, year, month) %>%
  dplyr::summarise(., cumPrecipY = sum(prcp..mm.day.),
                  minTempY = mean(tmin..deg.c.),
                  maxTempY = mean(tmax..deg.c.),
                  meanSRadY = mean(srad..W.m.2.),
                  meanVPY = mean(vp..Pa.)) %>%
  group_by(site_id, month) %>%
  dplyr::summarise(., cumPrecip = mean(cumPrecipY),
                  minTemp = mean(minTempY),
                  maxTemp = mean(maxTempY),
                  meanSR = mean(meanSRadY),
                  meanVP = mean(meanVPY))
#write.csv(mn_month, "mnMonthlyClim.csv")

```

```

##### Merge Plot Lat/Longs with Monthly Climate Data
ne_cLL <- merge(ne_month, ne_ll, by.x = "site_id", by.y = "site")
wi_cLL <- merge(wi_month, wi_ll, by.x = "site_id", by.y = "site")
mo_cLL <- merge(mo_month, mo_ll, by.x = "site_id", by.y = "site")
sd_cLL <- merge(sd_month, sd_ll, by.x = "site_id", by.y = "site")
oh_cLL <- merge(oh_month, oh_ll, by.x = "site_id", by.y = "site")
nd_cLL <- merge(nd_month, nd_ll, by.x = "site_id", by.y = "site")
mi_cLL <- merge(mi_month, mi_ll, by.x = "site_id", by.y = "site")
ks_cLL <- merge(ks_month, ks_ll, by.x = "site_id", by.y = "site")
in_cLL <- merge(in_month, ind_ll, by.x = "site_id", by.y = "site")
il_cLL <- merge(il_month, il_ll, by.x = "site_id", by.y = "site")
ia_cLL <- merge(ia_month, ia_ll, by.x = "site_id", by.y = "site")
mn_cLL <- merge(mn_month, mn_ll, by.x = "site_id", by.y = "site")

```

```

##### Filter trees in each state to only include live trees

```

```

##### Nebraska

```

```

ne_tree <- filter(ne, STATUSCD==1)
##### Wisconsin
wi_tree <- filter(wi, STATUSCD==1)
##### Missouri
mo_tree <- filter(mo, STATUSCD==1)
##### South Dakota
sd_tree <- filter(sd, STATUSCD==1)
##### Ohio
oh_tree <- filter(oh, STATUSCD==1)
##### North Dakota
nd_tree <- filter(nd, STATUSCD==1)
##### Michigan
mi_tree <- filter(mi, STATUSCD==1)
##### Kansas
ks_tree <- filter(ks, STATUSCD==1)
##### Indiana
in_tree <- filter(ind, STATUSCD==1)
##### Illinois
il_tree <- filter(il, STATUSCD==1)
##### Iowa
ia_tree <- filter(ia, STATUSCD==1)
##### Minnesota
mn_tree <- filter(mn, STATUSCD==1)

##### Merge tree data with plot-level climate data
##### Nebraska
ne_tc <- merge(ne_tree, ne_cLL, by.x = "PLOT", by.y = "site_id")
##### Wisconsin
wi_tc <- merge(wi_tree, wi_cLL, by.x = "PLOT", by.y = "site_id")
##### Missouri
mo_tc <- merge(mo_tree, mo_cLL, by.x = "PLOT", by.y = "site_id")
##### South Dakota
sd_tc <- merge(sd_tree, sd_cLL, by.x = "PLOT", by.y = "site_id")
##### Ohio
oh_tc <- merge(oh_tree, oh_cLL, by.x = "PLOT", by.y = "site_id")
##### North Dakota
nd_tc <- merge(nd_tree, nd_cLL, by.x = "PLOT", by.y = "site_id")
##### Michigan
mi_tc <- merge(mi_tree, mi_cLL, by.x = "PLOT", by.y = "site_id")
##### Kansas
ks_tc <- merge(ks_tree, ks_cLL, by.x = "PLOT", by.y = "site_id")
##### Indiana
in_tc <- merge(in_tree, in_cLL, by.x = "PLOT", by.y = "site_id")
##### Illinois
il_tc <- merge(il_tree, il_cLL, by.x = "PLOT", by.y = "site_id")
##### Iowa
ia_tc <- merge(ia_tree, ia_cLL, by.x = "PLOT", by.y = "site_id")
##### Minnesota
mn_tc <- merge(mn_tree, mn_cLL, by.x = "PLOT", by.y = "site_id")

##### Calculating minimum and maximum for climatic variables, by species
##### Nebraska

```

```

## Make species code a factor
ne_tc$SPCD <- as.factor(ne_tc$SPCD)
# Add state
ne_tc$State <- rep("NE",times = length(nrow(ne_tc)))
# Group by species/location and summarize to obtain extreme values
ne_r2 <- ne_tc %>%
  group_by(SPCD,State,PLOT,lat,lon) %>%
  dplyr::summarise(., minTC = min(minTemp),
                    maxTC = max(maxTemp),
                    minRain = min(cumPrecip),
                    maxRain = max(cumPrecip),
                    minAVP = min(meanVP),
                    maxAVP = max(meanVP),
                    minASR = min(meanSR),
                    maxASR = max(meanSR))
# Min SR
ne_minSR <- ne_r2 %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
ne_maxSR <- ne_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
ne_minTC <- ne_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
ne_maxTC <- ne_r2 %>%
  group_by(SPCD) %>%
  filter(., maxTC == max(maxTC)) %>%
  dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
ne_minR <- ne_r2 %>%
  group_by(SPCD) %>%
  filter(., minRain == min(minRain)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall
ne_maxR <- ne_r2 %>%
  group_by(SPCD) %>%
  filter(., maxRain == max(maxRain)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min VP
ne_minV <- ne_r2 %>%
  group_by(SPCD) %>%
  filter(., minAVP == min(minAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
ne_maxV <- ne_r2 %>%
  group_by(SPCD) %>

```

```

    filter(.,maxAVP == max(maxAVP)) %>%
    dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### Wisconsin
## Make species code a factor
wi_tc$SPCD <- as.factor(wi_tc$SPCD)
# Add state
wi_tc$State <- rep("WI",times = length(nrow(wi_tc)))
# Group by species/location and summarize to obtain extreme values
wi_r2 <- wi_tc %>%
  group_by(SPCD,State,PLOT,lat,lon) %>%
  dplyr::summarise(., minTC = min(minTemp),
                    maxTC = max(maxTemp),
                    minRain = min(cumPrecip),
                    maxRain = max(cumPrecip),
                    minAVP = min(meanVP),
                    maxAVP = max(meanVP),
                    minASR = min(meanSR),
                    maxASR = max(meanSR))

# Min SR
wi_minSR <- wi_r2 %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
wi_maxSR <- wi_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
wi_minTC <- wi_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
wi_maxTC <- wi_r2 %>%
  group_by(SPCD) %>%
  filter(.,maxTC == max(maxTC)) %>%
  dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
wi_minR <- wi_r2 %>%
  group_by(SPCD) %>%
  filter(.,minRain == min(minRain)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall
wi_maxR <- wi_r2 %>%
  group_by(SPCD) %>%
  filter(.,maxRain == max(maxRain)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min VP
wi_minV <- wi_r2 %>%
  group_by(SPCD) %>%
  filter(.,minAVP == min(minAVP)) %>%

```

```

dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
wi_maxV <- wi_r2 %>%
  group_by(SPCD) %>%
  filter(., maxAVP == max(maxAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### Missouri
## Make species code a factor
mo_tc$SPCD <- as.factor(mo_tc$SPCD)
# Add state
mo_tc$State <- rep("MO", times = length(nrow(mo_tc)))
# Group by species/location and summarize to obtain extreme values
mo_r2 <- mo_tc %>%
  group_by(SPCD, State, PLOT, lat, lon) %>%
  dplyr::summarise(., minTC = min(minTemp),
                    maxTC = max(maxTemp),
                    minRain = min(cumPrecip),
                    maxRain = max(cumPrecip),
                    minAVP = min(meanVP),
                    maxAVP = max(meanVP),
                    minASR = min(meanSR),
                    maxASR = max(meanSR))

# Min SR
mo_minSR <- mo_r2 %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
mo_maxSR <- mo_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
mo_minTC <- mo_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
mo_maxTC <- mo_r2 %>%
  group_by(SPCD) %>%
  filter(., maxTC == max(maxTC)) %>%
  dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
mo_minR <- mo_r2 %>%
  group_by(SPCD) %>%
  filter(., minRain == min(minRain)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall
mo_maxR <- mo_r2 %>%
  group_by(SPCD) %>%
  filter(., maxRain == max(maxRain)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)

```

```

# Min VP
mo_minV <- mo_r2 %>%
  group_by(SPCD) %>%
  filter(., minAVP == min(minAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
mo_maxV <- mo_r2 %>%
  group_by(SPCD) %>%
  filter(., maxAVP == max(maxAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### South Dakota
## Make species code a factor
sd_tc$SPCD <- as.factor(sd_tc$SPCD)
# Add state
sd_tc$State <- rep("SD", times = length(nrow(sd_tc)))
# Group by species/location and summarize to obtain extreme values
sd_r2 <- sd_tc %>%
  group_by(SPCD, State, PLOT, lat, lon) %>%
  dplyr::summarise(., minTC = min(minTemp),
                    maxTC = max(maxTemp),
                    minRain = min(cumPrecip),
                    maxRain = max(cumPrecip),
                    minAVP = min(meanVP),
                    maxAVP = max(meanVP),
                    minASR = min(meanSR),
                    maxASR = max(meanSR))

# Min SR
sd_minSR <- sd_r2 %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
sd_maxSR <- sd_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
sd_minTC <- sd_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
sd_maxTC <- sd_r2 %>%
  group_by(SPCD) %>%
  filter(., maxTC == max(maxTC)) %>%
  dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
sd_minR <- sd_r2 %>%
  group_by(SPCD) %>%
  filter(., minRain == min(minRain)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall

```

```

sd_maxR <- sd_r2 %>%
  group_by(SPCD) %>%
  filter(., maxRain == max(maxRain)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min VP
sd_minV <- sd_r2 %>%
  group_by(SPCD) %>%
  filter(., minAVP == min(minAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
sd_maxV <- sd_r2 %>%
  group_by(SPCD) %>%
  filter(., maxAVP == max(maxAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### Ohio
## Make species code a factor
oh_tc$SPCD <- as.factor(oh_tc$SPCD)
# Add state
oh_tc$State <- rep("OH", times = length(nrow(oh_tc)))
# Group by species/location and summarize to obtain extreme values
oh_r2 <- oh_tc %>%
  group_by(SPCD, State, PLOT, lat, lon) %>%
  dplyr::summarise(., minTC = min(minTemp),
                    maxTC = max(maxTemp),
                    minRain = min(cumPrecip),
                    maxRain = max(cumPrecip),
                    minAVP = min(meanVP),
                    maxAVP = max(meanVP),
                    minASR = min(meanSR),
                    maxASR = max(meanSR))

# Min SR
oh_minSR <- oh_r2 %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
oh_maxSR <- oh_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
oh_minTC <- oh_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
oh_maxTC <- oh_r2 %>%
  group_by(SPCD) %>%
  filter(., maxTC == max(maxTC)) %>%
  dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
oh_minR <- oh_r2 %>%

```

```

group_by(SPCD) %>%
  filter(., minRain == min(minRain)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall
oh_maxR <- oh_r2 %>%
  group_by(SPCD) %>%
  filter(., maxRain == max(maxRain)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min VP
oh_minV <- oh_r2 %>%
  group_by(SPCD) %>%
  filter(., minAVP == min(minAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
oh_maxV <- oh_r2 %>%
  group_by(SPCD) %>%
  filter(., maxAVP == max(maxAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### North Dakota
## Make species code a factor
nd_tc$SPCD <- as.factor(nd_tc$SPCD)
# Add state
nd_tc$State <- rep("ND", times = length(nrow(nd_tc)))
# Group by species/location and summarize to obtain extreme values
nd_r2 <- nd_tc %>%
  group_by(SPCD, State, PLOT, lat, lon) %>%
  dplyr::summarise(., minTC = min(minTemp),
                   maxTC = max(maxTemp),
                   minRain = min(cumPrecip),
                   maxRain = max(cumPrecip),
                   minAVP = min(meanVP),
                   maxAVP = max(meanVP),
                   minASR = min(meanSR),
                   maxASR = max(meanSR))

# Min SR
nd_minSR <- nd_r2 %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
nd_maxSR <- nd_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
nd_minTC <- nd_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
nd_maxTC <- nd_r2 %>%
  group_by(SPCD) %>%

```

```

filter(.,maxTC == max(maxTC)) %>%
dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
nd_minR <- nd_r2 %>%
group_by(SPCD) %>%
filter(.,minRain == min(minRain)) %>%
dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall
nd_maxR <- nd_r2 %>%
group_by(SPCD) %>%
filter(.,maxRain == max(maxRain)) %>%
dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min VP
nd_minV <- nd_r2 %>%
group_by(SPCD) %>%
filter(.,minAVP == min(minAVP)) %>%
dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
nd_maxV <- nd_r2 %>%
group_by(SPCD) %>%
filter(.,maxAVP == max(maxAVP)) %>%
dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### Michigan

## Make species code a factor
mi_tc$SPCD <- as.factor(mi_tc$SPCD)
# Add state
mi_tc$State <- rep("MI",times = length(nrow(mi_tc)))
# Group by species/location and summarize to obtain extreme values
mi_r2 <- mi_tc %>%
group_by(SPCD,State,PLOT,lat,lon) %>%
dplyr::summarise(., minTC = min(minTemp),
maxTC = max(maxTemp),
minRain = min(cumPrecip),
maxRain = max(cumPrecip),
minAVP = min(meanVP),
maxAVP = max(meanVP),
minASR = min(meanSR),
maxASR = max(meanSR))

# Min SR
mi_minSR <- mi_r2 %>%
group_by(SPCD) %>%
filter(., minASR == min(minASR)) %>%
dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
mi_maxSR <- mi_r2 %>%
group_by(SPCD) %>%
filter(., maxASR == max(maxASR)) %>%
dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
mi_minTC <- mi_r2 %>%
group_by(SPCD) %>%

```

```

filter(., minTC == min(minTC)) %>%
dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
mi_maxTC <- mi_r2 %>%
group_by(SPCD) %>%
filter(., maxTC == max(maxTC)) %>%
dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
mi_minR <- mi_r2 %>%
group_by(SPCD) %>%
filter(., minRain == min(minRain)) %>%
dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall
mi_maxR <- mi_r2 %>%
group_by(SPCD) %>%
filter(., maxRain == max(maxRain)) %>%
dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min VP
mi_minV <- mi_r2 %>%
group_by(SPCD) %>%
filter(., minAVP == min(minAVP)) %>%
dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
mi_maxV <- mi_r2 %>%
group_by(SPCD) %>%
filter(., maxAVP == max(maxAVP)) %>%
dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### Kansas

## Make species code a factor
ks_tc$SPCD <- as.factor(ks_tc$SPCD)
# Add state
ks_tc$State <- rep("KS", times = length(nrow(ks_tc)))
# Group by species/location and summarize to obtain extreme values
ks_r2 <- ks_tc %>%
group_by(SPCD, State, PLOT, lat, lon) %>%
dplyr::summarise(., minTC = min(minTemp),
                 maxTC = max(maxTemp),
                 minRain = min(cumPrecip),
                 maxRain = max(cumPrecip),
                 minAVP = min(meanVP),
                 maxAVP = max(meanVP),
                 minASR = min(meanSR),
                 maxASR = max(meanSR))

# Min SR
ks_minSR <- ks_r2 %>%
group_by(SPCD) %>%
filter(., minASR == min(minASR)) %>%
dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
ks_maxSR <- ks_r2 %>%
group_by(SPCD) %>%

```

```

filter(., maxASR == max(maxASR)) %>%
dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
ks_minTC <- ks_r2 %>%
group_by(SPCD) %>%
filter(., minTC == min(minTC)) %>%
dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
ks_maxTC <- ks_r2 %>%
group_by(SPCD) %>%
filter(., maxTC == max(maxTC)) %>%
dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
ks_minR <- ks_r2 %>%
group_by(SPCD) %>%
filter(., minRain == min(minRain)) %>%
dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall
ks_maxR <- ks_r2 %>%
group_by(SPCD) %>%
filter(., maxRain == max(maxRain)) %>%
dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min VP
ks_minV <- ks_r2 %>%
group_by(SPCD) %>%
filter(., minAVP == min(minAVP)) %>%
dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
ks_maxV <- ks_r2 %>%
group_by(SPCD) %>%
filter(., maxAVP == max(maxAVP)) %>%
dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### Indiana
## Make species code a factor
in_tc$SPCD <- as.factor(in_tc$SPCD)
# Group by species/location and summarize to obtain extreme values
# Add state
in_tc$State <- rep("IN", times = length(nrow(in_tc)))
in_r2 <- in_tc %>%
group_by(SPCD, State, PLOT, lat, lon) %>%
dplyr::summarise(., minTC = min(minTemp),
                 maxTC = max(maxTemp),
                 minRain = min(cumPrecip),
                 maxRain = max(cumPrecip),
                 minAVP = min(meanVP),
                 maxAVP = max(meanVP),
                 minASR = min(meanSR),
                 maxASR = max(meanSR))

# Min SR
in_minSR <- in_r2 %>%
group_by(SPCD) %>%
filter(., minASR == min(minASR)) %>%

```

```

dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
in_maxSR <- in_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
in_minTC <- in_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
in_maxTC <- in_r2 %>%
  group_by(SPCD) %>%
  filter(., maxTC == max(maxTC)) %>%
  dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
in_minR <- in_r2 %>%
  group_by(SPCD) %>%
  filter(., minRain == min(minRain)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall
in_maxR <- in_r2 %>%
  group_by(SPCD) %>%
  filter(., maxRain == max(maxRain)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min VP
in_minV <- in_r2 %>%
  group_by(SPCD) %>%
  filter(., minAVP == min(minAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
in_maxV <- in_r2 %>%
  group_by(SPCD) %>%
  filter(., maxAVP == max(maxAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### Illinois
## Make species code a factor
il_tc$SPCD <- as.factor(il_tc$SPCD)
# Add state
il_tc$State <- rep("IL", times = length(nrow(il_tc)))
# Group by species/location and summarize to obtain extreme values
il_r2 <- il_tc %>%
  group_by(SPCD, State, PLOT, lat, lon) %>%
  dplyr::summarise(., minTC = min(minTemp),
                  maxTC = max(maxTemp),
                  minRain = min(cumPrecip),
                  maxRain = max(cumPrecip),
                  minAVP = min(meanVP),
                  maxAVP = max(meanVP),
                  minASR = min(meanSR),
                  maxASR = max(meanSR))

```

```

# Min SR
il_minSR <- il_r2 %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)
# Max SR
il_maxSR <- il_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)
# Min temp
il_minTC <- il_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max temp
il_maxTC <- il_r2 %>%
  group_by(SPCD) %>%
  filter(., maxTC == max(maxTC)) %>%
  dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min rainfall
il_minR <- il_r2 %>%
  group_by(SPCD) %>%
  filter(., minRain == min(minRain)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Max rainfall
il_maxR <- il_r2 %>%
  group_by(SPCD) %>%
  filter(., maxRain == max(maxRain)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)
# Min VP
il_minV <- il_r2 %>%
  group_by(SPCD) %>%
  filter(., minAVP == min(minAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)
# Max VP
il_maxV <- il_r2 %>%
  group_by(SPCD) %>%
  filter(., maxAVP == max(maxAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### Iowa
## Make species code a factor
ia_tc$SPCD <- as.factor(ia_tc$SPCD)
# Add state
ia_tc$State <- rep("IA", times = length(nrow(ia_tc)))
# Group by species/location and summarize to obtain extreme values
ia_r2 <- ia_tc %>%
  group_by(SPCD, State, PLOT, lat, lon) %>%
  dplyr::summarise(., minTC = min(minTemp),
                  maxTC = max(maxTemp),
                  minRain = min(cumPrecip),
                  maxRain = max(cumPrecip),

```

```

    minAVP = min(meanVP),
    maxAVP = max(meanVP),
    minASR = min(meanSR),
    maxASR = max(meanSR))

# Min SR
ia_minSR <- ia_r2 %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)

# Max SR
ia_maxSR <- ia_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)

# Min temp
ia_minTC <- ia_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)

# Max temp
ia_maxTC <- ia_r2 %>%
  group_by(SPCD) %>%
  filter(., maxTC == max(maxTC)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)

# Min rainfall
ia_minR <- ia_r2 %>%
  group_by(SPCD) %>%
  filter(., minRain == min(minRain)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)

# Max rainfall
ia_maxR <- ia_r2 %>%
  group_by(SPCD) %>%
  filter(., maxRain == max(maxRain)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)

# Min VP
ia_minV <- ia_r2 %>%
  group_by(SPCD) %>%
  filter(., minAVP == min(minAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)

# Max VP
ia_maxV <- ia_r2 %>%
  group_by(SPCD) %>%
  filter(., maxAVP == max(maxAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

##### Minnesota
## Make species code a factor
mn_tc$SPCD <- as.factor(mn_tc$SPCD)
# Add state
mn_tc$State <- rep("MN", times = length(nrow(mn_tc)))
# Group by species/location and summarize to obtain extreme values
mn_r2 <- mn_tc %>%
  group_by(SPCD, State, PLOT, lat, lon) %>%

```

```

dplyr::summarise(., minTC = min(minTemp),
                 maxTC = max(maxTemp),
                 minRain = min(cumPrecip),
                 maxRain = max(cumPrecip),
                 minAVP = min(meanVP),
                 maxAVP = max(meanVP),
                 minASR = min(meanSR),
                 maxASR = max(meanSR))

# Min SR
mn_minSR <- mn_r2 %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -maxASR)

# Max SR
mn_maxSR <- mn_r2 %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR)

# Min temp
mn_minTC <- mn_r2 %>%
  group_by(SPCD) %>%
  filter(., minTC == min(minTC)) %>%
  dplyr::select(., -maxTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)

# Max temp
mn_maxTC <- mn_r2 %>%
  group_by(SPCD) %>%
  filter(., maxTC == max(maxTC)) %>%
  dplyr::select(., -minTC, -minRain, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)

# Min rainfall
mn_minR <- mn_r2 %>%
  group_by(SPCD) %>%
  filter(., minRain == min(minRain)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minAVP, -maxAVP, -minASR, -maxASR)

# Max rainfall
mn_maxR <- mn_r2 %>%
  group_by(SPCD) %>%
  filter(., maxRain == max(maxRain)) %>%
  dplyr::select(., -minTC, -maxTC, -minRain, -minAVP, -maxAVP, -minASR, -maxASR)

# Min VP
mn_minV <- mn_r2 %>%
  group_by(SPCD) %>%
  filter(., minAVP == min(minAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -maxAVP, -minASR, -maxASR)

# Max VP
mn_maxV <- mn_r2 %>%
  group_by(SPCD) %>%
  filter(., maxAVP == max(maxAVP)) %>%
  dplyr::select(., -minTC, -maxTC, -maxRain, -minRain, -minAVP, -minASR, -maxASR)

```

```

##### Aggregate Min/Max Values Across States and make lat/long spatial features
# Min Monthly Temperatures
midwMinTC <- rbind(ne_minTC, wi_minTC, mo_minTC, sd_minTC,
                     oh_minTC, nd_minTC, mi_minTC, ks_minTC,

```

```

        in_minTC, il_minTC, ia_minTC, mn_minTC)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwMinTC$Common <- spp$common[match(midwMinTC$SPCD, spp$SPCD)]
midwMinTC$Species <- spp$scientific[match(midwMinTC$SPCD, spp$SPCD)]
# Write file
#write.csv(midwMinTC, "sppMinTCstate.csv")
# Convert lat/long to sf
MinTC_sf = st_as_sf(midwMinTC, coords = c("lon", "lat"),
                     crs = 4269)

# Max Monthly Temperatures
midwMaxTC <- rbind(ne_maxTC, wi_maxTC, mo_maxTC, sd_maxTC,
                     oh_maxTC, nd_maxTC, mi_maxTC, ks_maxTC,
                     in_maxTC, il_maxTC, ia_maxTC, mn_maxTC)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwMaxTC$Common <- spp$common[match(midwMaxTC$SPCD, spp$SPCD)]
midwMaxTC$Species <- spp$scientific[match(midwMaxTC$SPCD, spp$SPCD)]
# Write file
#write.csv(midwMaxTC, "sppMaxTCstate.csv")
# Convert lat/long to sf
MaxTC_sf = st_as_sf(midwMaxTC, coords = c("lon", "lat"),
                     crs = 4269)

# Minimum Monthly Cumulative Rainfall
midwMinR <- rbind(ne_minR, wi_minR, mo_minR, sd_minR,
                     oh_minR, nd_minR, mi_minR, ks_minR,
                     in_minR, il_minR, ia_minR, mn_minR)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwMinR$Common <- spp$common[match(midwMinR$SPCD, spp$SPCD)]
midwMinR$Species <- spp$scientific[match(midwMinR$SPCD, spp$SPCD)]
# Write file
#write.csv(midwMinR, "sppMinRstate.csv")
# Convert lat/long to sf
MinR_sf = st_as_sf(midwMinR, coords = c("lon", "lat"),
                     crs = 4269)

# Max Monthly Cumulative Rainfall
midwMaxR <- rbind(ne_maxR, wi_maxR, mo_maxR, sd_maxR,
                     oh_maxR, nd_maxR, mi_maxR, ks_maxR,
                     in_maxR, il_maxR, ia_maxR, mn_maxR)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwMaxR$Common <- spp$common[match(midwMaxR$SPCD, spp$SPCD)]
midwMaxR$Species <- spp$scientific[match(midwMaxR$SPCD, spp$SPCD)]
# Write file
#write.csv(midwMaxR, "sppMaxRstate.csv")
# Convert lat/long to sf
MaxR_sf = st_as_sf(midwMaxR, coords = c("lon", "lat"),
                     crs = 4269)

# Minimum Avg Vapor Pressure
midwMinV <- rbind(ne_minV, wi_minV, mo_minV, sd_minV,
                     oh_minV, nd_minV, mi_minV, ks_minV,
                     in_minV, il_minV, ia_minV, mn_minV)

```

```

# Add Species/Common Names to Each Row based on species' code (SPCD)
midwMinV$Common <- spp$common[match(midwMinV$SPCD, spp$SPCD)]
midwMinV$Species <- spp$scientific[match(midwMinV$SPCD, spp$SPCD)]
# Write file
#write.csv(midwMinV, "sppMinVstate.csv")
# Convert lat/long to sf
MinV_sf = st_as_sf(midwMinV, coords = c("lon", "lat"),
                    crs = 4269)

# Maximum Avg Vapor Pressure
midwMaxV <- rbind(ne_maxV, wi_maxV, mo_maxV, sd_maxV,
                    oh_maxV, nd_maxV, mi_maxV, ks_maxV,
                    in_maxV, il_maxV, ia_maxV, mn_maxV)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwMaxV$Common <- spp$common[match(midwMaxV$SPCD, spp$SPCD)]
midwMaxV$Species <- spp$scientific[match(midwMaxV$SPCD, spp$SPCD)]
# Write file
#write.csv(midwMaxV, "sppMaxVstate.csv")
# Convert lat/long to sf
MaxV_sf = st_as_sf(midwMaxV, coords = c("lon", "lat"),
                    crs = 4269)

# Minimum Shortwave Radiation
midwMinSR <- rbind(ne_minSR, wi_minSR, mo_minSR, sd_minSR,
                     oh_minSR, nd_minSR, mi_minSR, ks_minSR,
                     in_minSR, il_minSR, ia_minSR, mn_minSR)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwMinSR$Common <- spp$common[match(midwMinSR$SPCD, spp$SPCD)]
midwMinSR$Species <- spp$scientific[match(midwMinSR$SPCD, spp$SPCD)]
# Write file
#write.csv(midwMinSR, "sppMinSRstate.csv")
# Convert lat/long to sf
MinSR_sf = st_as_sf(midwMinSR, coords = c("lon", "lat"),
                     crs = 4269)

# Maximum Shortwave Radiation
midwMaxSR <- rbind(ne_maxSR, wi_maxSR, mo_maxSR, sd_maxSR,
                     oh_maxSR, nd_maxSR, mi_maxSR, ks_maxSR,
                     in_maxSR, il_maxSR, ia_maxSR, mn_maxSR)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwMaxSR$Common <- spp$common[match(midwMaxSR$SPCD, spp$SPCD)]
midwMaxSR$Species <- spp$scientific[match(midwMaxSR$SPCD, spp$SPCD)]
# Write file
#write.csv(midwMaxSR, "sppMaxSRstate.csv")
# Convert lat/long to sf
MaxSR_sf = st_as_sf(midwMaxSR, coords = c("lon", "lat"),
                     crs = 4269)

```

Mapping Min/Max Environmental Variables by Species Across States

```

##### Minimum Temperature
# Subset to only include minimum for each species
minTCspp <- MinTC_sf %>%

```

```

group_by(SPCD) %>%
  filter(., minTC == min(minTC))

# Create color palette with 17 distinct colors
kelly(n = 18) # Not using first light gray color

## [1] "#F2F3F4" "#222222" "#F3C300" "#875692" "#F38400" "#A1CAF1" "#BE0032"
## [8] "#C2B280" "#848482" "#008856" "#E68FAC" "#0067A5" "#F99379" "#604E97"
## [15] "#F6A600" "#B3446C" "#DCD300" "#882D17"

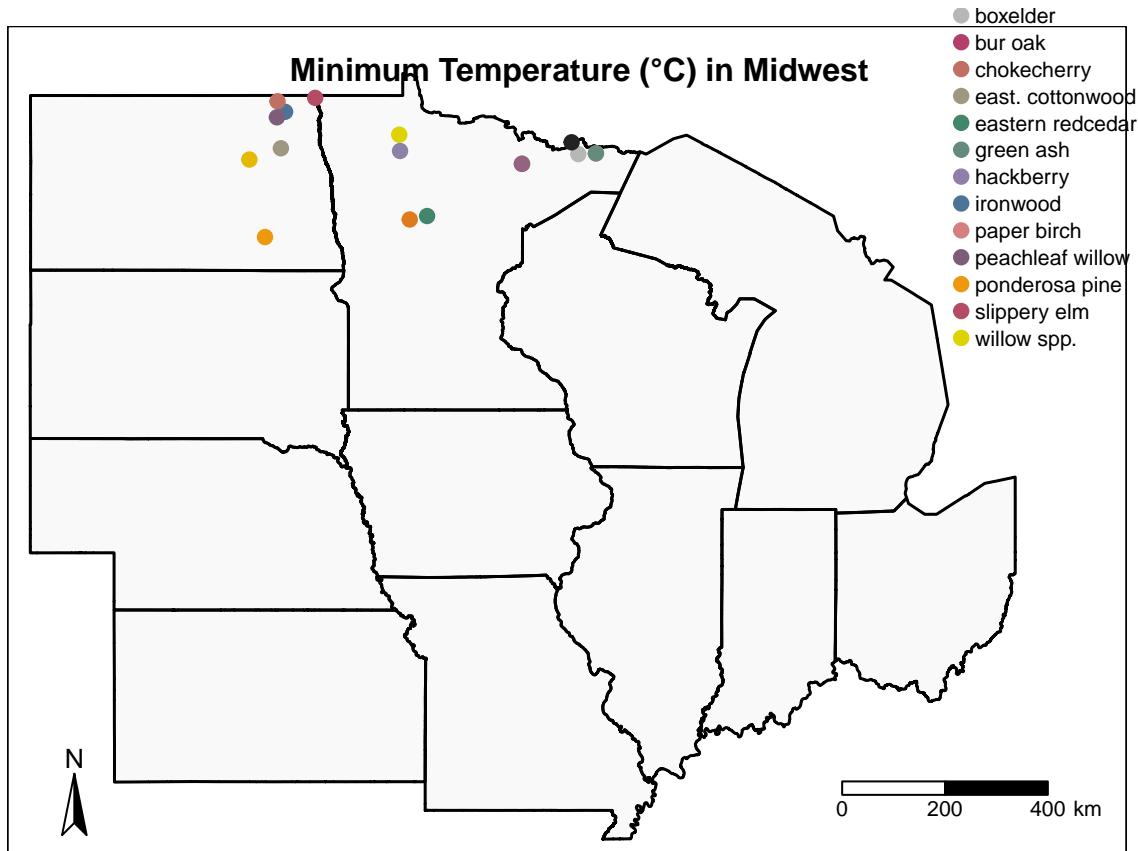
pal <- c("#222222", "#F3C300", "#875692", "#F38400",
        "#A1CAF1", "#BE0032", "#C2B280", "#848482",
        "#008856", "#E68FAC", "#0067A5", "#F99379",
        "#604E97", "#F6A600", "#B3446C", "#DCD300")

# Make mapping space larger - create new bounding box
bbox_new <- st_bbox(state)

# Specify range of values of current box
xrange <- bbox_new$xmax - bbox_new$xmin # range of x values
yrange <- bbox_new$ymax - bbox_new$ymin # range of y values
# Expand the bounding box to create more space top/right
bbox_new[3] <- bbox_new[3] + (0.09 * xrange) # xmax - right
bbox_new[4] <- bbox_new[4] + (0.04 * yrange) # ymax - top
# Make new bounding box a sf polygon
bbox_new <- bbox_new %>%
  st_as_sfc()

# Map - Minimum temps for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
    tm_borders(col = "black", lwd = 1.5) +
# Add dots for the min temp for each state
tm_shape(minTCspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
          palette = pal) +
# Add legend and title and specify their positions
tm_layout(legend.outside = FALSE,
          legend.position = c(0.84, 0.6),
          legend.text.size = 0.7,
          title = "Minimum Temperature (°C) in Midwest",
          title.size = 1,
          title.fontface = "bold",
          title.position = c("center", "top")) +
# Add a north arrow
tm_compass(type = "arrow", position = c("left", "bottom"),
            size = 2) +
# Add a scale bar
tm_scale_bar(breaks = c(0, 200, 400), text.size = 0.7)

```



```
##### Mapping Min/Max Environmental Variables by Species Across States
```

Maximum Temperature

```
# Increase space at top of map - create new bounding box (using a
# different bounding box than the previous figure to include these as
# Panels A and B of the same figure - see manuscript)
```

```
bbox_new2 <- st_bbox(state)
```

```
# Specify range of values of current box
```

```
yrange2 <- bbox_new2$ymax - bbox_new2$ymin # range of y values
```

```
# Expand the bounding box to create more space at top
```

```
bbox_new2[4] <- bbox_new2[4] + (0.04 * yrange2) # ymax - top
```

```
# Make new bounding box a sf polygon
```

```
bbox_new2 <- bbox_new2 %>%
```

```
st_as_sf()
```

```
# Subset to only include minimum for each species
```

```
maxTCspp <- MaxTC_sf %>%
```

```
group_by(SPCD) %>%
```

```
filter(., maxTC == max(maxTC))
```

```
# Map - Max temps for midwest states
```

```
# Plot Midwest state boundaries in more spacious bounding box
```

```
tm_shape(state, bbox = bbox_new2) +
```

```
tm_polygons(col = "#f9f9f9") +
```

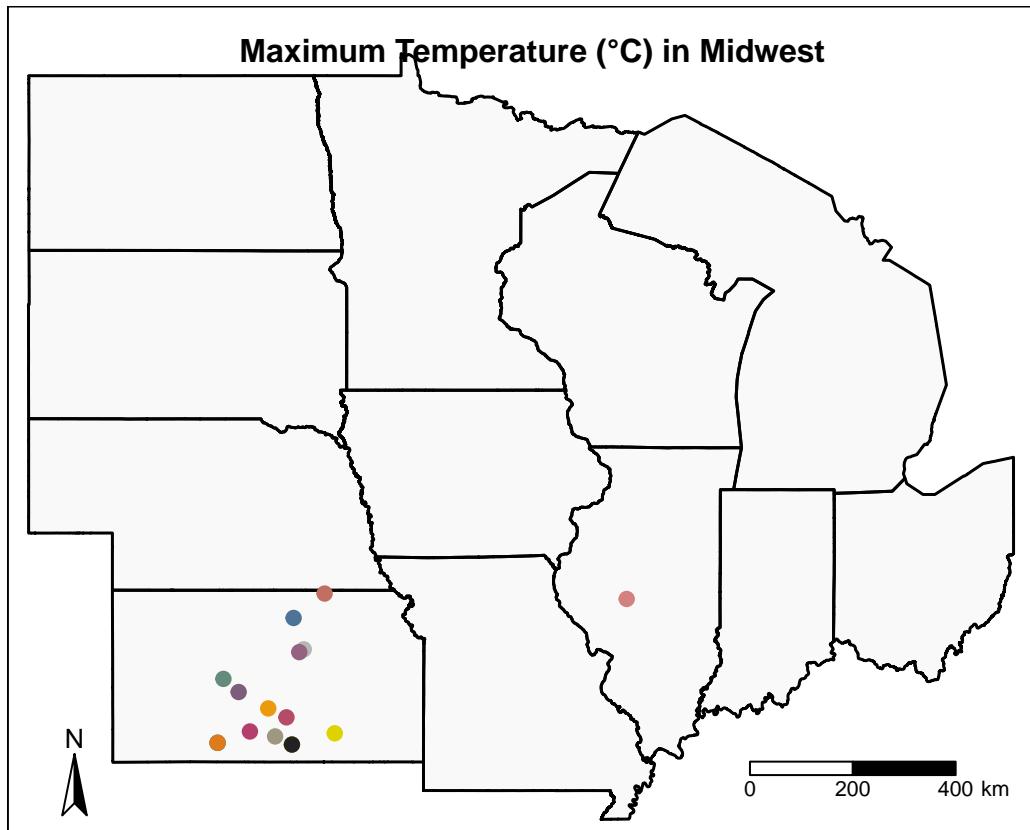
```
tm_shape(state) +
```

```
tm_borders(col = "black", lwd = 1.5) +
```

```

# Add dots for the min temp for each state
tm_shape(maxTCspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
    palette = pal, legend.show = FALSE) +
  # Add legend and title and specify their positions
  tm_layout(title = "Maximum Temperature (°C) in Midwest",
    title.size = 1,
    title.fontface = "bold",
    title.position = c("center", "top")) +
  # Add a north arrow
  tm_compass(type = "arrow", position = c("left", "bottom"),
    size = 2) +
  # Add a scale bar
  tm_scale_bar(breaks = c(0, 200, 400), text.size = 0.7)

```



```

##### Mapping Min/Max Environmental Variables by Species Across States

##### Minimum Rainfall
# Subset to only include minimum for each species
minRspp <- MinR_sf %>%
  group_by(SPCD) %>%
  filter(., minRain == min(minRain))

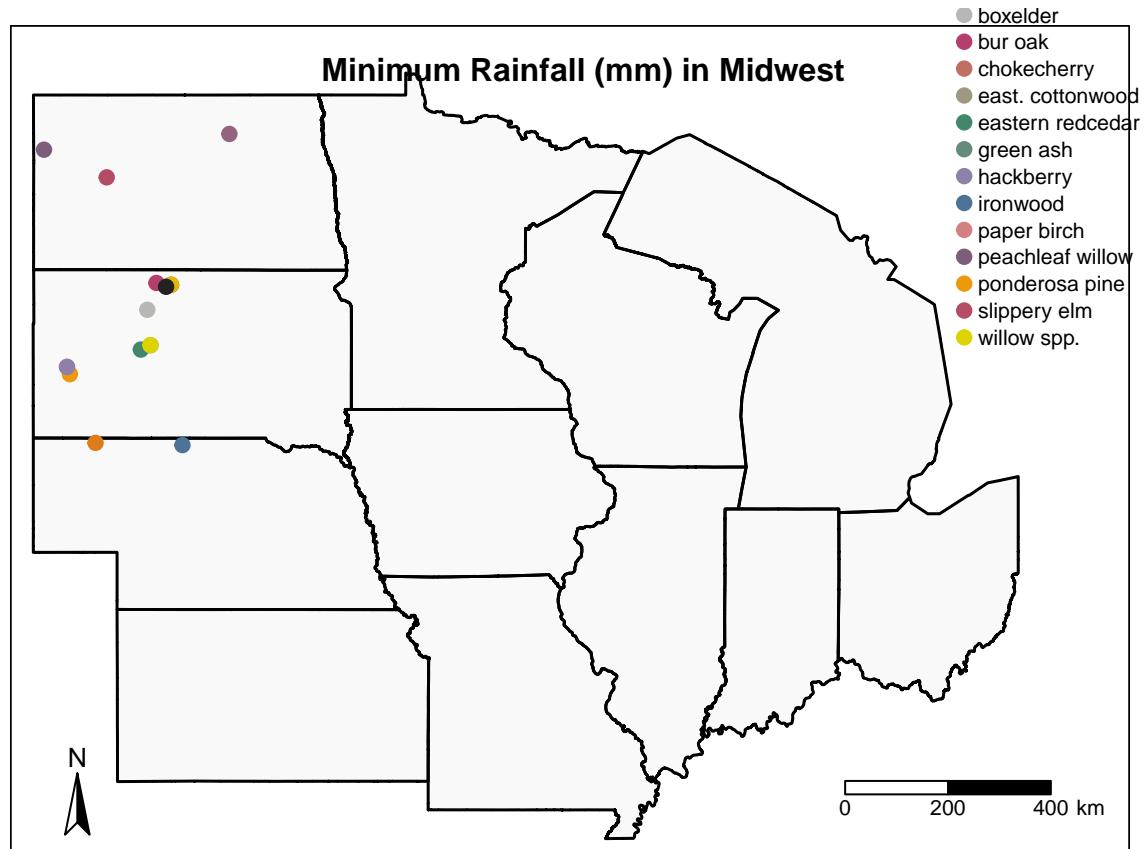
# Map - Minimum rainfall for midwest states
# Plot Midwest state boundaries in more spacious bounding box

```

```

tm_shape(state, bbox = bbox_new) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +
  # Add dots for the min temp for each state
  tm_shape(minRspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
          palette = pal) +
  # Add legend and title and specify their positions
  tm_layout(legend.outside = FALSE,
            legend.position = c(0.84,0.6),
            legend.text.size = 0.7,
            title = "Minimum Rainfall (mm) in Midwest",
            title.size = 1,
            title.fontface = "bold",
            title.position = c("center","top")) +
  # Add a north arrow
  tm_compass(type = "arrow", position = c("left", "bottom"),
              size = 2) +
  # Add a scale bar
  tm_scale_bar(breaks = c(0,200,400), text.size = 0.7)

```



Mapping Min/Max Environmental Variables by Species Across States

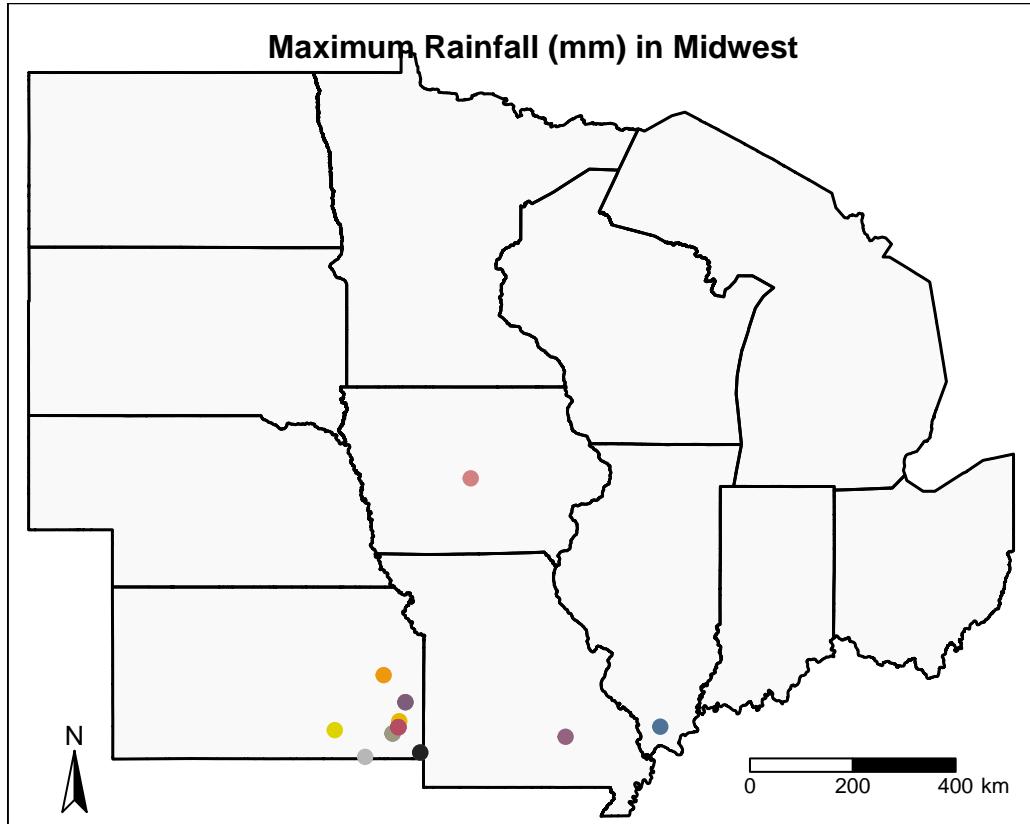
Maximum Rainfall

```

# Subset to only include minimum for each species
maxRspp <- MaxR_sf %>%
  group_by(SPCD) %>%
  filter(., maxRain == max(maxRain))

# Map - Max rainfall for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new2) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +
  # Add dots for the min temp for each state
  tm_shape(maxRspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
          palette = pal, legend.show = FALSE) +
  # Add legend and title and specify their positions
  tm_layout(title = "Maximum Rainfall (mm) in Midwest",
            title.size = 1,
            title.fontface = "bold",
            title.position = c("center", "top")) +
  # Add a north arrow
  tm_compass(type = "arrow", position = c("left", "bottom"),
             size = 2) +
  # Add a scale bar
  tm_scale_bar(breaks = c(0, 200, 400), text.size = 0.7)

```



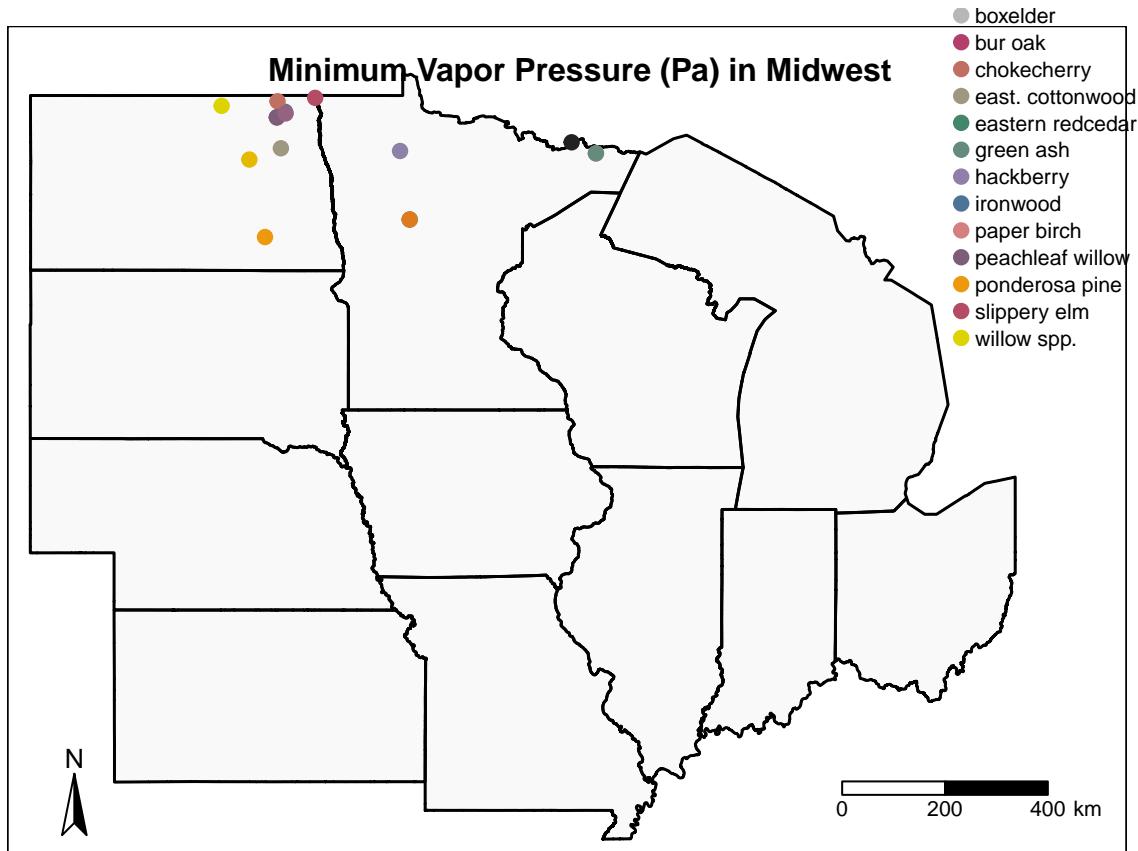
```

##### Mapping Min/Max Environmental Variables by Species Across States

##### Minimum Vapor Pressure
# Subset to only include minimum for each species
minVspp <- MinV_sf %>%
  group_by(SPCD) %>%
  filter(., minAVP == min(minAVP))

# Map - Minimum vapor pressure for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +
  # Add dots for the min temp for each state
  tm_shape(minVspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
          palette = pal) +
  # Add legend and title and specify their positions
  tm_layout(legend.outside = FALSE,
            legend.position = c(0.84,0.6),
            legend.text.size = 0.7,
            title = "Minimum Vapor Pressure (Pa) in Midwest",
            title.size = 1,
            title.fontface = "bold",
            title.position = c("center","top")) +
  # Add a north arrow
  tm_compass(type = "arrow", position = c("left", "bottom"),
             size = 2) +
  # Add a scale bar
  tm_scale_bar(breaks = c(0,200,400), text.size = 0.7)

```

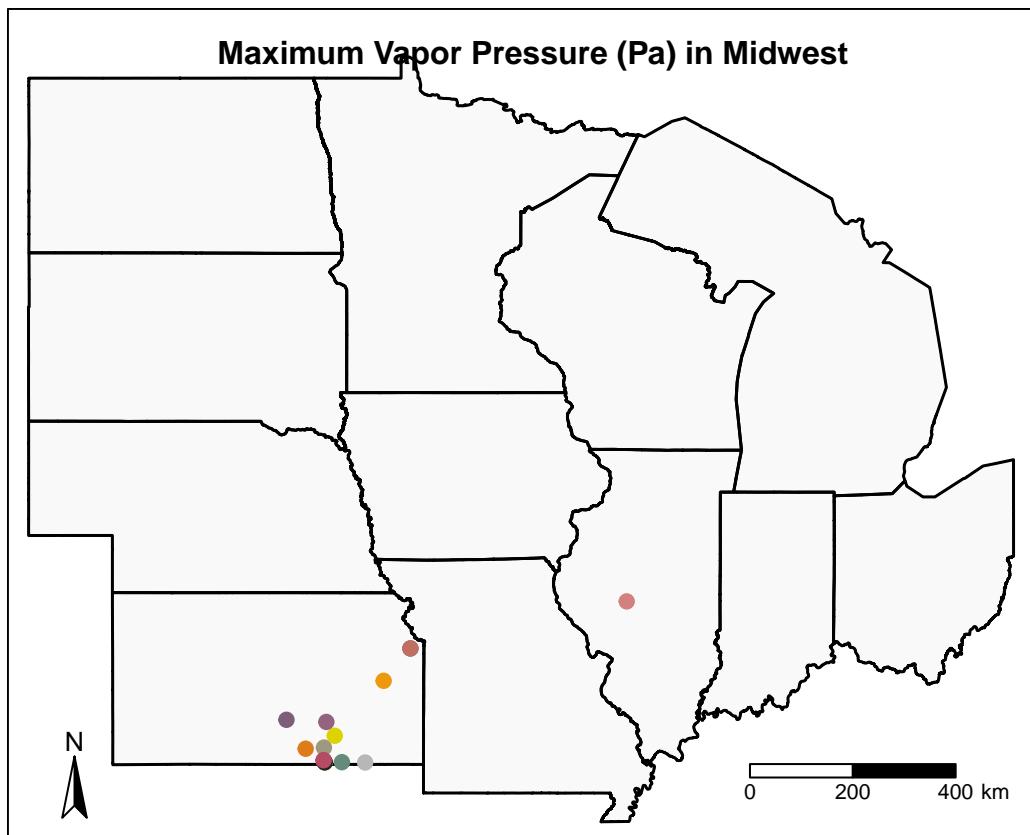


```
##### Mapping Min/Max Environmental Variables by Species Across States
```

```
##### Maximum Vapor Pressure
# Subset to only include minimum for each species
maxVspp <- MaxV_sf %>%
  group_by(SPCD) %>%
  filter(., maxAVP == max(maxAVP))

# Map - Max vapor pressure for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new2) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +
  # Add dots for the min temp for each state
  tm_shape(maxVspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
         palette = pal, legend.show = FALSE) +
  # Add legend and title and specify their positions
  tm_layout(title = "Maximum Vapor Pressure (Pa) in Midwest",
            title.size = 1,
            title.fontface = "bold",
            title.position = c("center", "top")) +
  # Add a north arrow
  tm_compass(type = "arrow", position = c("left", "bottom"),
             size = 2) +
```

```
# Add a scale bar
tm_scale_bar(breaks = c(0,200,400), text.size = 0.7)
```



```
##### Mapping Min/Max Environmental Variables by Species Across States
```

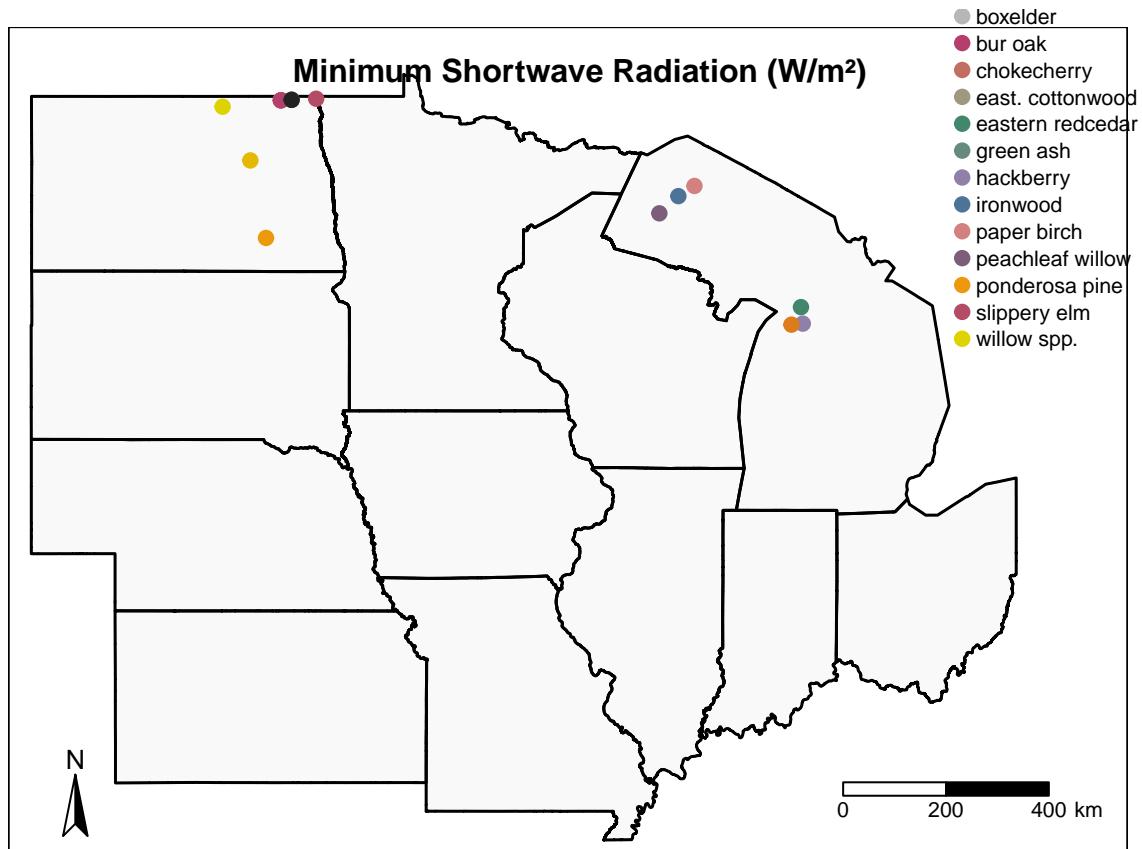
```
##### Minimum Shortwave Radiation
# Subset to only include minimum for each species
minSRspp <- MinSR_sf %>%
  group_by(SPCD) %>%
  filter(., minASR == min(minASR))

# Map - Minimum shortwave radiation for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +
  # Add dots for the min temp for each state
  tm_shape(minSRspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
         palette = pal) +
  # Add legend and title and specify their positions
  tm_layout(legend.outside = FALSE,
            legend.position = c(0.84,0.6),
            legend.text.size = 0.7,
```

```

title = "Minimum Shortwave Radiation (W/m2)",
  title.size = 1,
  title.fontface = "bold",
  title.position = c("center", "top")) +
# Add a north arrow
tm_compass(type = "arrow", position = c("left", "bottom"),
  size = 2) +
# Add a scale bar
tm_scale_bar(breaks = c(0,200,400), text.size = 0.7)

```



```
##### Mapping Min/Max Environmental Variables by Species Across States
```

```
##### Maximum Shortwave Radiation
```

```

# Subset to only include minimum for each species
maxSRspp <- MaxSR_sf %>%
  group_by(SPCD) %>%
  filter(., maxASR == max(maxASR))

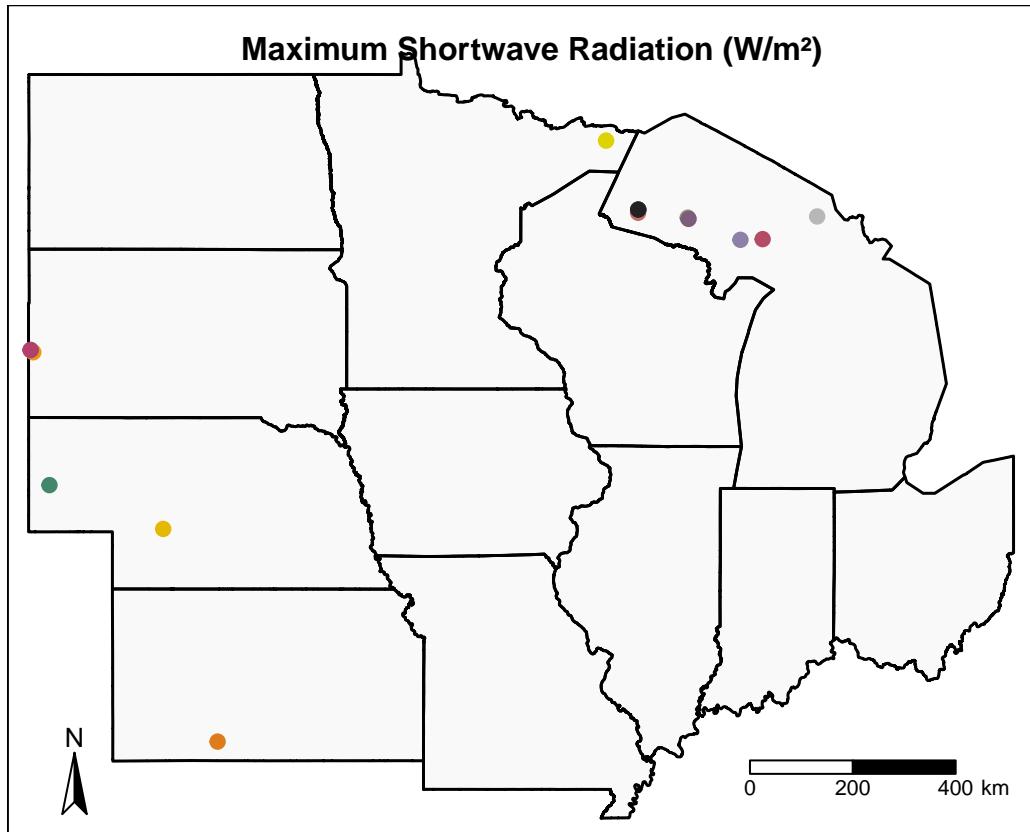
# Map - Max shortwave radiation for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new2) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +

```

```

# Add dots for the min temp for each state
tm_shape(maxSRspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
    palette = pal, legend.show = FALSE) +
  # Add legend and title and specify their positions
  tm_layout(title = "Maximum Shortwave Radiation (W/m2)",
    title.size = 1,
    title.fontface = "bold",
    title.position = c("center", "top")) +
  # Add a north arrow
  tm_compass(type = "arrow", position = c("left", "bottom"),
    size = 2) +
  # Add a scale bar
  tm_scale_bar(breaks = c(0, 200, 400), text.size = 0.7)

```



```
##### Procedure to Create Map of Midwestern FIA Plot Locations
```

```

## I created this as Figure 1 in the manuscript - it takes a very
# long time to run because it is tons of data, so I have not evaluated
# the code to run/produce this map

```

```

## Add State to each dataset
ne_11$state <- rep("NE", times = length(nrow(ne_11)))
wi_11$state <- rep("WI", times = length(nrow(wi_11)))
mo_11$state <- rep("MO", times = length(nrow(mo_11)))

```

```

sd_ll$state <- rep("SD",times = length(nrow(sd_ll)))
oh_ll$state <- rep("OH",times = length(nrow(oh_ll)))
nd_ll$state <- rep("ND",times = length(nrow(nd_ll)))
mi_ll$state <- rep("MI",times = length(nrow(mi_ll)))
ks_ll$state <- rep("KS",times = length(nrow(ks_ll)))
ind_ll$state <- rep("IN",times = length(nrow(ind_ll)))
il_ll$state <- rep("IL",times = length(nrow(il_ll)))
ia_ll$state <- rep("IA",times = length(nrow(ia_ll)))
mn_ll$state <- rep("MN",times = length(nrow(mn_ll)))

## Midwest FIA Plot Locations
midwestFIA <- rbind(ne_ll, wi_ll, mo_ll, sd_ll,
                      oh_ll, nd_ll, mi_ll, ks_ll,
                      ind_ll, il_ll, ia_ll, mn_ll)
# Convert Lat/Longs to spatial features - NAD83 to match FIA procedure
dat_sf = st_as_sf(midwestFIA, coords = c("lon", "lat"),
                   crs = 4269)

# Create color palettes for plots in each state
statePal <- tol(n = 12)

# Add state polygons/boundaries
tm_shape(state, bbox = bbox_new) +
  tm_polygons(col = "#f9f9f9") +
tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +
# Add the dots for each FIA plot
tm_shape(dat_sf) +
  tm_dots(col = "state", size = 0.05, title = "State",
          palette = statePal) +
# Add legend and title and specify their positions
tm_layout(legend.outside = FALSE,
          legend.position = c(0.84,0.6),
          legend.text.size = 0.7,
          title = "Locations of Midwest FIA Plots",
          title.size = 1,
          title.fontface = "bold",
          title.position = c("center","top")) +
# Add a north arrow
tm_compass(type = "arrow", position = c("left", "bottom"),
            size = 2) +
# Add a scale bar
tm_scale_bar(breaks = c(0,200,400), text.size = 0.7)

```

```

### Variation in Soil Environment (Soil Organic Carbon and
# Available Water Storage) for Species

# Procedure for resampling the soil raster layers from 500x500 m to
# 1000x1000 m to match resolution of climate data

## Function used to aggregate resolution of soil rasters from
#500 x 500 m (resolution when the TIFs were created in ArcGIS)
# to the 1000 x 1000 m scale

```

```

myfun <- function(x){
  # Call files from this folder
  r1 <- raster::raster(paste0("./State Soil Tifs/", x))
  # Use aggregate() with a factor of two and the mean function
  # to decrease resolution to 1000 x 1000 m
  rs<-raster::aggregate(r1, fact = 2, fun = mean)
  # Write the new files into this folder
  writeRaster(rs, paste0("./Resampled Rasters/rs_", x), overwrite=T)
}

## Run each file through the map function to resample each raster tif
tifName %>% map(., myfun)

## Read in resampled (1000 x 1000 m) soil organic carbon (SOC)
#and soil water availability (soilW) rasters for each state - files
# labeled as 500m but are all 1000 x 1000 m

##### Nebraska
NESoc <- raster::raster("./Resampled Rasters/rs_NEsoc_150cm_500m.tif")
NEaws <- raster::raster("./Resampled Rasters/rs_NEsoilW_150cm_500m.tif")

##### Wisconsin
WIsoc <- raster::raster("./Resampled Rasters/rs_WIsoc_150cm_500m.tif")
WIaws <- raster::raster("./Resampled Rasters/rs_WIsoilW_150cm_500m.tif")

##### Missouri
M0soc <- raster::raster("./Resampled Rasters/rs_M0soc_150cm_500m.tif")
M0aws <- raster::raster("./Resampled Rasters/rs_M0soilW_150cm_500m.tif")

##### South Dakota
SDsoc <- raster::raster("./Resampled Rasters/rs_SDsoc_150cm_500m.tif")
SDaws <- raster::raster("./Resampled Rasters/rs_SDsoilW_150cm_500m.tif")

##### Ohio
OHsoc <- raster::raster("./Resampled Rasters/rs_OHsoc_150cm_500m.tif")
OHaws <- raster::raster("./Resampled Rasters/rs_OHsoilW_150cm_500m.tif")

##### North Dakota
NDsoc <- raster::raster("./Resampled Rasters/rs_NDsoc_150cm_500m.tif")
NDaws <- raster::raster("./Resampled Rasters/rs_NDsoilW_150cm_500m.tif")

##### Michigan
MIsoc <- raster::raster("./Resampled Rasters/rs_MIsoc_150cm_500m.tif")
MIaws <- raster::raster("./Resampled Rasters/rs_MIsoilW_150cm_500m.tif")

##### Kansas
KSsoc <- raster::raster("./Resampled Rasters/rs_KSsoc_150cm_500m.tif")
KSaws <- raster::raster("./Resampled Rasters/rs_KSsoilW_150cm_500m.tif")

##### Indiana
INsoc <- raster::raster("./Resampled Rasters/rs_INsoc_150cm_500m.tif")
INaws <- raster::raster("./Resampled Rasters/rs_INsoilW_150cm_500m.tif")

```

```

##### Illinois
ILsoc <- raster::raster("./Resampled Rasters/rs_ILsoc_150cm_500m.tif")
ILaws <- raster::raster("./Resampled Rasters/rs_ILsoilW_150cm_500m.tif")

##### Iowa - no SOC data and only AWS between 20-50 cm depth
IAaws <- raster::raster("./Resampled Rasters/rs_IAsoilW_20_50_500m.tif")

##### Minnesota
MNsoc <- raster::raster("./Resampled Rasters/rs_MNsoc_150cm_500m.tif")
MNaws <- raster::raster("./Resampled Rasters/rs_MNsoilW_150cm_500m.tif")

### For each state:
### 1) Make average plot coordinates (lat/long) into spatial
# features for each state
### 2) Group data by Plot, Species Codes (SPCD), and Month to only
# include one tree per species per plot per month (reduce dataset
# size - i.e., create presence/absence by spp)
### 3) Use the extract function to create a new column with the Soil
# Organic Carbon and Available Soil Water for each plot
### 4) Group by species/plot and find min and max

#####
##### Nebraska
# Coordinates to sf
neTC_sf <- st_as_sf(ne_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot for each month
neTC_sf2 <- neTC_sf %>%
  group_by(PLOT, SPCD, month) %>%
  slice(1)
# Extract SOC for each species/plot
neTC_sf2$SOC <- raster::extract(NEsoc, neTC_sf2)
# Extract AWS for each species/plot
neTC_sf2$AWS <- raster::extract(NEaws, neTC_sf2)

##### Make species code a factor
neTC_sf2$SPCD <- as.factor(neTC_sf2$SPCD)
# Add state
neTC_sf2$State <- rep("NE", times = length(nrow(neTC_sf2)))
# Group by species/location and summarize to obtain extreme values
ne_soil <- neTC_sf2 %>%
  group_by(SPCD, State, PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC, na.rm = TRUE),
                  maxSOC = max(SOC, na.rm = TRUE),
                  minAWS = min(AWS, na.rm = TRUE),
                  maxAWS = max(AWS, na.rm = TRUE))
## Determine min/max for both SOC/AWS
# Min NE SOC
ne_minSOC <- ne_soil %>%
  group_by(SPCD) %>%
  filter(., minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max NE SOC

```

```

ne_maxSOC <- ne_soil %>%
  group_by(SPCD) %>%
  filter(., maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min NE AWS
ne_minAWS <- ne_soil %>%
  group_by(SPCD) %>%
  filter(., minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max NE AWS
ne_maxAWS <- ne_soil %>%
  group_by(SPCD) %>%
  filter(., maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### Wisconsin
wiTC_sf <- st_as_sf(wi_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot for each month
wiTC_sf2 <- wiTC_sf %>%
  group_by(PLOT, SPCD, month) %>%
  slice(1)
# Extract SOC for each species/plot
wiTC_sf2$SOC <- raster::extract(WIsoc, wiTC_sf2)
# Extract AWS for each species/plot
wiTC_sf2$AWS <- raster::extract(WIaws, wiTC_sf2)

##### Make species code a factor
wiTC_sf2$SPCD <- as.factor(wiTC_sf2$SPCD)
# Add state
wiTC_sf2$State <- rep("WI", times = length(nrow(wiTC_sf2)))
# Group by species/location and summarize to obtain extreme values
wi_soil <- wiTC_sf2 %>%
  group_by(SPCD, State, PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC, na.rm = TRUE),
                  maxSOC = max(SOC, na.rm = TRUE),
                  minAWS = min(AWS, na.rm = TRUE),
                  maxAWS = max(AWS, na.rm = TRUE))
## Determine min/max for both SOC/AWS
# Min WI SOC
wi_minSOC <- wi_soil %>%
  group_by(SPCD) %>%
  filter(., minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max wi SOC
wi_maxSOC <- wi_soil %>%
  group_by(SPCD) %>%
  filter(., maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min wi AWS
wi_minAWS <- wi_soil %>%
  group_by(SPCD) %>%
  filter(., minAWS == min(minAWS)) %>%

```

```

dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max wi AWS
wi_maxAWS <- wi_soil %>%
  group_by(SPCD) %>%
  filter(., maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### Missouri
moTC_sf <- st_as_sf(mo_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot
moTC_sf2 <- moTC_sf %>%
  group_by(PLOT, SPCD, month) %>%
  slice(1)
# Extract SOC for each species/plot
moTC_sf2$SOC <- raster::extract(MOsoc, moTC_sf2)
# Extract AWS for each species/plot
moTC_sf2$AWS <- raster::extract(MOaws, moTC_sf2)

##### Make species code a factor
moTC_sf2$SPCD <- as.factor(moTC_sf2$SPCD)
# Add state
moTC_sf2$State <- rep("MO", times = length(nrow(moTC_sf2)))
# Group by species/location and summarize to obtain extreme values
mo_soil <- moTC_sf2 %>%
  group_by(SPCD, State, PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC, na.rm = TRUE),
                   maxSOC = max(SOC, na.rm = TRUE),
                   minAWS = min(AWS, na.rm = TRUE),
                   maxAWS = max(AWS, na.rm = TRUE))
## Determine min/max for both SOC/AWS
# Min mo SOC
mo_minSOC <- mo_soil %>%
  group_by(SPCD) %>%
  filter(., minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max mo SOC
mo_maxSOC <- mo_soil %>%
  group_by(SPCD) %>%
  filter(., maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min mo AWS
mo_minAWS <- mo_soil %>%
  group_by(SPCD) %>%
  filter(., minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max mo AWS
mo_maxAWS <- mo_soil %>%
  group_by(SPCD) %>%
  filter(., maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### South Dakota

```

```

sdTC_sf <- st_as_sf(sd_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot
sdTC_sf2 <- sdTC_sf %>%
  group_by(PLOT,SPCD,month) %>%
  slice(1)
# Extract SOC for each species/plot
sdTC_sf2$SOC <- raster::extract(SDsoc, sdTC_sf2)
# Extract AWS for each species/plot
sdTC_sf2$AWS <- raster::extract(SDaws, sdTC_sf2)

##### Make species code a factor
sdTC_sf2$SPCD <- as.factor(sdTC_sf2$SPCD)
# Add state
sdTC_sf2$State <- rep("SD",times = length(nrow(sdTC_sf2)))
# Group by species/location and summarize to obtain extreme values
sd_soil <- sdTC_sf2 %>%
  group_by(SPCD,State,PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC),
                  maxSOC = max(SOC),
                  minAWS = min(AWS),
                  maxAWS = max(AWS))
## Determine min/max for both SOC/AWS
# Min sd SOC
sd_minSOC <- sd_soil %>%
  group_by(SPCD) %>%
  filter(.,minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max sd SOC
sd_maxSOC <-sd_soil %>%
  group_by(SPCD) %>%
  filter(.,maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min sd AWS
sd_minAWS <- sd_soil %>%
  group_by(SPCD) %>%
  filter(.,minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max sd AWS
sd_maxAWS <- sd_soil %>%
  group_by(SPCD) %>%
  filter(.,maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### Ohio
ohTC_sf <- st_as_sf(oh_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot
ohTC_sf2 <- ohTC_sf %>%
  group_by(PLOT,SPCD,month) %>%
  slice(1)
# Extract SOC for each species/plot
ohTC_sf2$SOC <- raster::extract(OHsoc, ohTC_sf2)

```

```

# Extract AWS for each species/plot
ohTC_sf2$AWS <- raster::extract(OHaws, ohTC_sf2)

##### Make species code a factor
ohTC_sf2$SPCD <- as.factor(ohTC_sf2$SPCD)
# Add state
ohTC_sf2$State <- rep("OH",times = length(nrow(ohTC_sf2)))
# Group by species/location and summarize to obtain extreme values
oh_soil <- ohTC_sf2 %>%
  group_by(SPCD,State,PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC),
                  maxSOC = max(SOC),
                  minAWS = min(AWS),
                  maxAWS = max(AWS))
## Determine min/max for both SOC/AWS
# Min oh SOC
oh_minSOC <- oh_soil %>%
  group_by(SPCD) %>%
  filter(.,minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max oh SOC
oh_maxSOC <- oh_soil %>%
  group_by(SPCD) %>%
  filter(.,maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min oh AWS
oh_minAWS <- oh_soil %>%
  group_by(SPCD) %>%
  filter(.,minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max oh AWS
oh_maxAWS <- oh_soil %>%
  group_by(SPCD) %>%
  filter(.,maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### North Dakota
ndTC_sf <- st_as_sf(nd_tc, coords = c("lon", "lat"),
                     crs = 4269)
# Only include first tree per species in each plot
ndTC_sf2 <- ndTC_sf %>%
  group_by(PLOT,SPCD,month) %>%
  slice(1)
# Extract SOC for each species/plot
ndTC_sf2$SOC <- raster::extract(NDsoc, ndTC_sf2)
# Extract AWS for each species/plot
ndTC_sf2$AWS <- raster::extract(NDaws, ndTC_sf2)

##### Make species code a factor
ndTC_sf2$SPCD <- as.factor(ndTC_sf2$SPCD)
# Add state
ndTC_sf2$State <- rep("ND",times = length(nrow(ndTC_sf2)))
# Group by species/location and summarize to obtain extreme values

```

```

nd_soil <- ndTC_sf2 %>%
  group_by(SPCD,State,PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC),
                  maxSOC = max(SOC),
                  minAWS = min(AWS),
                  maxAWS = max(AWS))
## Determine min/max for both SOC/AWS
# Min nd SOC
nd_minSOC <- nd_soil %>%
  group_by(SPCD) %>%
  filter(.,minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max nd SOC
nd_maxSOC <-nd_soil %>%
  group_by(SPCD) %>%
  filter(.,maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min nd AWS
nd_minAWS <- nd_soil %>%
  group_by(SPCD) %>%
  filter(.,minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max nd AWS
nd_maxAWS <- nd_soil %>%
  group_by(SPCD) %>%
  filter(.,maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### Michigan
miTC_sf <- st_as_sf(mi_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot
miTC_sf2 <- miTC_sf %>%
  group_by(PLOT,SPCD,month) %>%
  slice(1)
# Extract SOC for each species/plot
miTC_sf2$SOC <- raster::extract(MIsoc, miTC_sf2)
# Extract AWS for each species/plot
miTC_sf2$AWS <- raster::extract(MIaws, miTC_sf2)

##### Make species code a factor
miTC_sf2$SPCD <- as.factor(miTC_sf2$SPCD)
# Add state
miTC_sf2$State <- rep("MI",times = length(nrow(miTC_sf2)))
# Group by species/location and summarize to obtain extreme values
mi_soil <- miTC_sf2 %>%
  group_by(SPCD,State,PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC),
                  maxSOC = max(SOC),
                  minAWS = min(AWS),
                  maxAWS = max(AWS))
## Determine min/max for both SOC/AWS
# Min mi SOC

```

```

mi_minSOC <- mi_soil %>%
  group_by(SPCD) %>%
  filter(., minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max mi SOC

mi_maxSOC <- mi_soil %>%
  group_by(SPCD) %>%
  filter(., maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min mi AWS

mi_minAWS <- mi_soil %>%
  group_by(SPCD) %>%
  filter(., minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max mi AWS

mi_maxAWS <- mi_soil %>%
  group_by(SPCD) %>%
  filter(., maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### Kansas
ksTC_sf <- st_as_sf(ks_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot
ksTC_sf2 <- ksTC_sf %>%
  group_by(PLOT, SPCD, month) %>%
  slice(1)
# Extract SOC for each species/plot
ksTC_sf2$SOC <- raster::extract(KSsoc, ksTC_sf2)
# Extract AWS for each species/plot
ksTC_sf2$AWS <- raster::extract(KSaws, ksTC_sf2)

##### Make species code a factor
ksTC_sf2$SPCD <- as.factor(ksTC_sf2$SPCD)
# Add state
ksTC_sf2$State <- rep("KS", times = length(nrow(ksTC_sf2)))
# Group by species/location and summarize to obtain extreme values
ks_soil <- ksTC_sf2 %>%
  group_by(SPCD, State, PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC),
                  maxSOC = max(SOC),
                  minAWS = min(AWS),
                  maxAWS = max(AWS))
## Determine min/max for both SOC/AWS

# Min ks SOC
ks_minSOC <- ks_soil %>%
  group_by(SPCD) %>%
  filter(., minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max ks SOC
ks_maxSOC <- ks_soil %>%
  group_by(SPCD) %>%
  filter(., maxSOC == max(maxSOC)) %>%

```

```

dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min ks AWS
ks_minAWS <- ks_soil %>%
  group_by(SPCD) %>%
  filter(., minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max ks AWS
ks_maxAWS <- ks_soil %>%
  group_by(SPCD) %>%
  filter(., maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### Indiana
inTC_sf <- st_as_sf(in_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot
inTC_sf2 <- inTC_sf %>%
  group_by(PLOT, SPCD, month) %>%
  slice(1)
# Extract SOC for each species/plot
inTC_sf2$SOC <- raster::extract(INsoc, inTC_sf2)
# Extract AWS for each species/plot
inTC_sf2$AWS <- raster::extract(INaws, inTC_sf2)

##### Make species code a factor
inTC_sf2$SPCD <- as.factor(inTC_sf2$SPCD)
# Add state
inTC_sf2$State <- rep("IN", times = length(nrow(inTC_sf2)))
# Group by species/location and summarize to obtain extreme values
in_soil <- inTC_sf2 %>%
  group_by(SPCD, State, PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC),
                   maxSOC = max(SOC),
                   minAWS = min(AWS),
                   maxAWS = max(AWS))
## Determine min/max for both SOC/AWS
# Min in SOC
in_minSOC <- in_soil %>%
  group_by(SPCD) %>%
  filter(., minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max in SOC
in_maxSOC <- in_soil %>%
  group_by(SPCD) %>%
  filter(., maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min in AWS
in_minAWS <- in_soil %>%
  group_by(SPCD) %>%
  filter(., minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max in AWS
in_maxAWS <- in_soil %>%

```

```

group_by(SPCD) %>%
  filter(., maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### Illinois
ilTC_sf <- st_as_sf(il_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot
ilTC_sf2 <- ilTC_sf %>%
  group_by(PLOT, SPCD, month) %>%
  slice(1)
# Extract SOC for each species/plot
ilTC_sf2$SOC <- raster::extract(ILsoc, ilTC_sf2)
# Extract AWS for each species/plot
ilTC_sf2$AWS <- raster::extract(ILaws, ilTC_sf2)

##### Make species code a factor
ilTC_sf2$SPCD <- as.factor(ilTC_sf2$SPCD)
# Add state
ilTC_sf2$State <- rep("IL", times = length(nrow(ilTC_sf2)))
# Group by species/location and summarize to obtain extreme values
il_soil <- ilTC_sf2 %>%
  group_by(SPCD, State, PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC),
                   maxSOC = max(SOC),
                   minAWS = min(AWS),
                   maxAWS = max(AWS))
## Determine min/max for both SOC/AWS
# Min il SOC
il_minSOC <- il_soil %>%
  group_by(SPCD) %>%
  filter(., minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max il SOC
il_maxSOC <- il_soil %>%
  group_by(SPCD) %>%
  filter(., maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min il AWS
il_minAWS <- il_soil %>%
  group_by(SPCD) %>%
  filter(., minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max il AWS
il_maxAWS <- il_soil %>%
  group_by(SPCD) %>%
  filter(., maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

##### Iowa
iaTC_sf <- st_as_sf(ia_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot

```

```

iaTC_sf2 <- iaTC_sf %>%
  group_by(PLOT,SPCD,month) %>%
  slice(1)
# Extract AWS for each species/plot - only 20-50 cm
iaTC_sf2$AWS <- raster::extract(IAaws, iaTC_sf2)

##### Make species code a factor
iaTC_sf2$SPCD <- as.factor(iaTC_sf2$SPCD)
# Add state
iaTC_sf2$State <- rep("IA",times = length(nrow(iaTC_sf2)))
# Group by species/location and summarize to obtain extreme values
ia_soil <- iaTC_sf2 %>%
  group_by(SPCD,State,PLOT) %>%
  dplyr::summarise(., minAWS = min(AWS),
                   maxAWS = max(AWS))
## Determine min/max for AWS
# Min ia AWS
ia_minAWS <- ia_soil %>%
  group_by(SPCD) %>%
  filter(.,minAWS == min(minAWS)) %>%
  dplyr::select(., -maxAWS)
# Max ia AWS
ia_maxAWS <- ia_soil %>%
  group_by(SPCD) %>%
  filter(.,maxAWS == max(maxAWS)) %>%
  dplyr::select(., -minAWS)

##### Minnesota
mnTC_sf <- st_as_sf(mn_tc, coords = c("lon", "lat"),
                      crs = 4269)
# Only include first tree per species in each plot
mnTC_sf2 <- mnTC_sf %>%
  group_by(PLOT,SPCD,month) %>%
  slice(1)
# Extract SOC for each species/plot
mnTC_sf2$SOC <- raster::extract(MNsoc, mnTC_sf2)
# Extract AWS for each species/plot
mnTC_sf2$AWS <- raster::extract(MNaws, mnTC_sf2)

##### Make species code a factor
mnTC_sf2$SPCD <- as.factor(mnTC_sf2$SPCD)
# Add state
mnTC_sf2$State <- rep("MN",times = length(nrow(mnTC_sf2)))
# Group by species/location and summarize to obtain extreme values
mn_soil <- mnTC_sf2 %>%
  group_by(SPCD,State,PLOT) %>%
  dplyr::summarise(., minSOC = min(SOC),
                   maxSOC = max(SOC),
                   minAWS = min(AWS),
                   maxAWS = max(AWS))
## Determine min/max for both SOC/AWS
# Min mn SOC
mn_minSOC <- mn_soil %>

```

```

group_by(SPCD) %>%
  filter(., minSOC == min(minSOC)) %>%
  dplyr::select(., -maxSOC, -minAWS, -maxAWS)
# Max mn SOC
mn_maxSOC <- mn_soil %>%
  group_by(SPCD) %>%
  filter(., maxSOC == max(maxSOC)) %>%
  dplyr::select(., -minSOC, -minAWS, -maxAWS)
# Min mn AWS
mn_minAWS <- mn_soil %>%
  group_by(SPCD) %>%
  filter(., minAWS == min(minAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -maxAWS)
# Max mn AWS
mn_maxAWS <- mn_soil %>%
  group_by(SPCD) %>%
  filter(., maxAWS == max(maxAWS)) %>%
  dplyr::select(., -maxSOC, -minSOC, -minAWS)

```

Aggregate Min/Max Values for SOC and AWS Across States

```

##### Min Soil Organic Carbon - No data for IA so excluded
midwminSOC <- rbind(ne_minSOC, wi_minSOC, mo_minSOC, sd_minSOC,
                      oh_minSOC, nd_minSOC, mi_minSOC, ks_minSOC,
                      in_minSOC, il_minSOC, mn_minSOC)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwminSOC$Common <- spp$common[match(midwminSOC$SPCD, spp$SPCD)]
midwminSOC$Species <- spp$scientific[match(midwminSOC$SPCD, spp$SPCD)]
# Write file
#write.csv(midwminSOC, "sppMinSOCstate.csv")

##### Max Soil Organic Carbon - No data for IA so excluded
midwmaxSOC <- rbind(ne_maxSOC, wi_maxSOC, mo_maxSOC, sd_maxSOC,
                      oh_maxSOC, nd_maxSOC, mi_maxSOC, ks_maxSOC,
                      in_maxSOC, il_maxSOC, mn_maxSOC)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwmaxSOC$Common <- spp$common[match(midwmaxSOC$SPCD, spp$SPCD)]
midwmaxSOC$Species <- spp$scientific[match(midwmaxSOC$SPCD, spp$SPCD)]
# Write file
#write.csv(midwmaxSOC, "sppMaxSOCstate.csv")

##### Min Available Water Storage - Excluding IA b/c data is not across the same soil depth (so values
midwminAWS <- rbind(ne_minAWS, wi_minAWS, mo_minAWS, sd_minAWS,
                      oh_minAWS, nd_minAWS, mi_minAWS, ks_minAWS,
                      in_minAWS, il_minAWS, mn_minAWS)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwminAWS$Common <- spp$common[match(midwminAWS$SPCD, spp$SPCD)]
midwminAWS$Species <- spp$scientific[match(midwminAWS$SPCD, spp$SPCD)]
# Write file
#write.csv(midwminAWS, "sppMinAWSstate.csv")

##### Max Available Water Storage - Excluding IA b/c data is not across the same soil depth (so values
midwmaxAWS <- rbind(ne_maxAWS, wi_maxAWS, mo_maxAWS, sd_maxAWS,

```

```

    oh_maxAWS, nd_maxAWS, mi_maxAWS, ks_maxAWS,
    in_maxAWS, il_maxAWS, mn_maxAWS)
# Add Species/Common Names to Each Row based on species' code (SPCD)
midwmaxAWS$Common <- spp$common[match(midwmaxAWS$SPCD, spp$SPCD)]
midwmaxAWS$Species <- spp$scientific[match(midwmaxAWS$SPCD, spp$SPCD)]
# Write file
#write.csv(midwmaxAWS, "sppMaxAWSstate.csv")

```

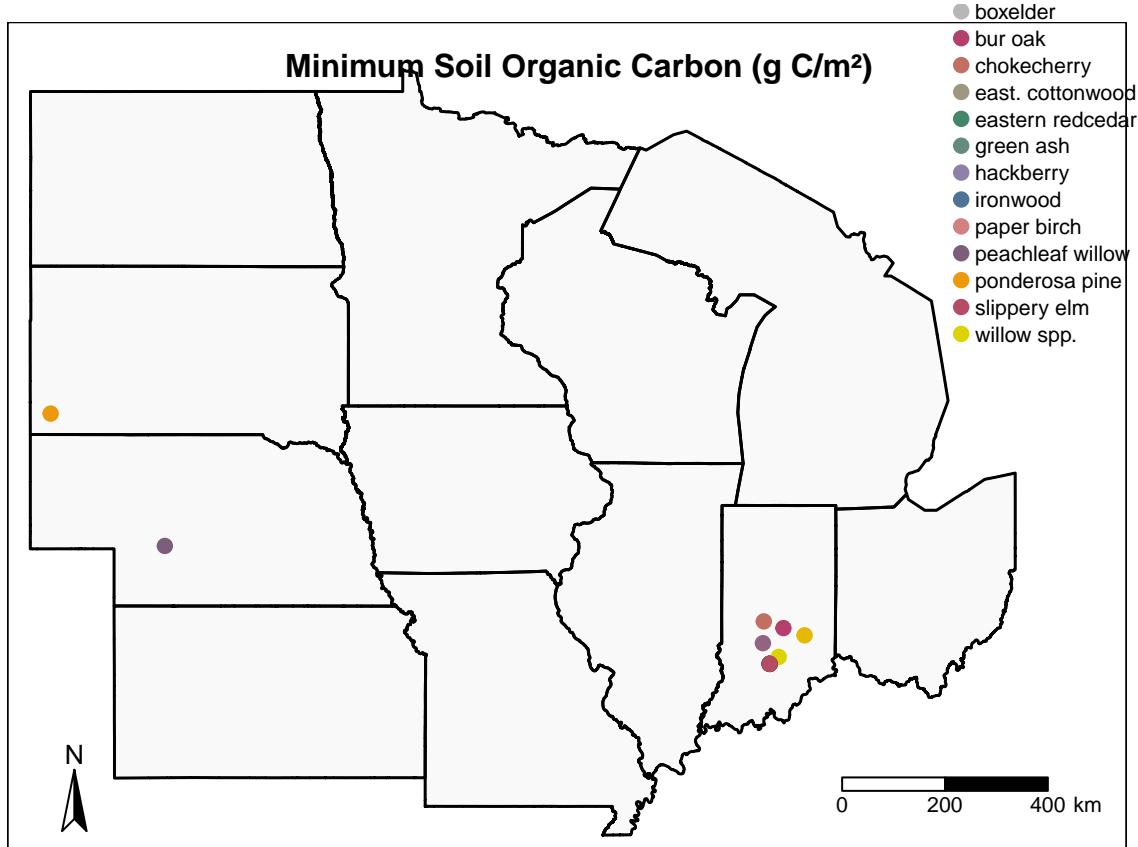
```

### Filter to determine minimum/maximum SOC and AWS per species across Midwest, and plot these data

##### Minimum Soil Organic Carbon (g C/m2)
# Subset to only include minimum for each species
minSOCspp <- midwminSOC %>%
  group_by(SPCD) %>%
  filter(., minSOC == min(minSOC))

# Map - Min SOC for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +
  # Add dots for the min temp for each state
  tm_shape(minSOCspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
          palette = pal) +
  # Add legend and title and specify their positions
  tm_layout(legend.outside = FALSE,
            legend.position = c(0.84,0.6),
            legend.text.size = 0.7,
            title = "Minimum Soil Organic Carbon (g C/m2)",
            title.size = 1,
            title.fontface = "bold",
            title.position = c("center","top")) +
  # Add a north arrow
  tm_compass(type = "arrow", position = c("left", "bottom"),
             size = 2) +
  # Add a scale bar
  tm_scale_bar(breaks = c(0,200,400), text.size = 0.7)

```



```
### Filter to determine minimum/maximum SOC and AWS per species
# across Midwest, and plot these data
```

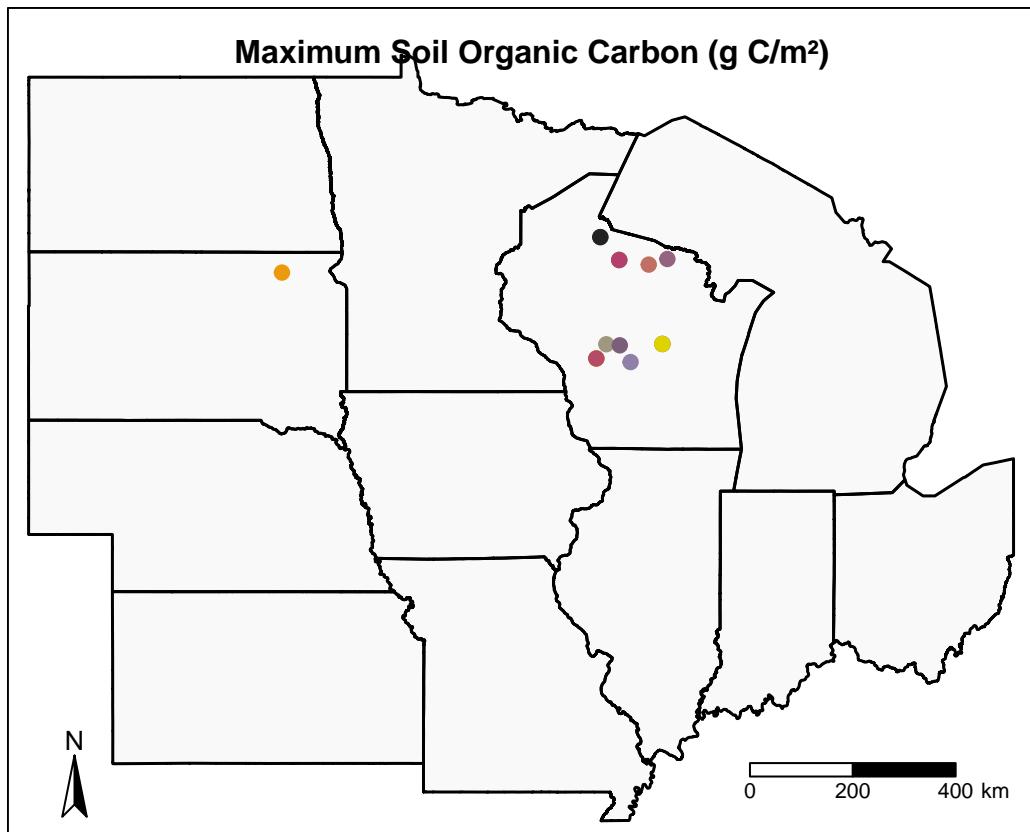
```
##### Maximum Soil Organic Carbon (g C/m²)
# Subset to only include maximum for each species
maxSOCspp <- midwmaxSOC %>%
  group_by(SPCD) %>%
  filter(., maxSOC == max(maxSOC))

# Map - Max SOC for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new2) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +
  # Add dots for the min temp for each state
  tm_shape(maxSOCspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
         palette = pal, legend.show = FALSE) +
  # Add legend and title and specify their positions
  tm_layout(title = "Maximum Soil Organic Carbon (g C/m²)",
            title.size = 1,
            title.fontface = "bold",
            title.position = c("center", "top")) +
  # Add a north arrow
  tm_compass(type = "arrow", position = c("left", "bottom"),
```

```

size = 2) +
# Add a scale bar
tm_scale_bar(breaks = c(0,200,400), text.size = 0.7)

```



```

### Filter to determine minimum/maximum SOC and AWS per species
# across Midwest, and plot these data

```

```

##### Minimum Available Water Storage (mm)
# Subset to only include minimum for each species
minAWSspp <- midwminAWS %>%
  group_by(SPCD) %>%
  filter(., minAWS == min(minAWS))

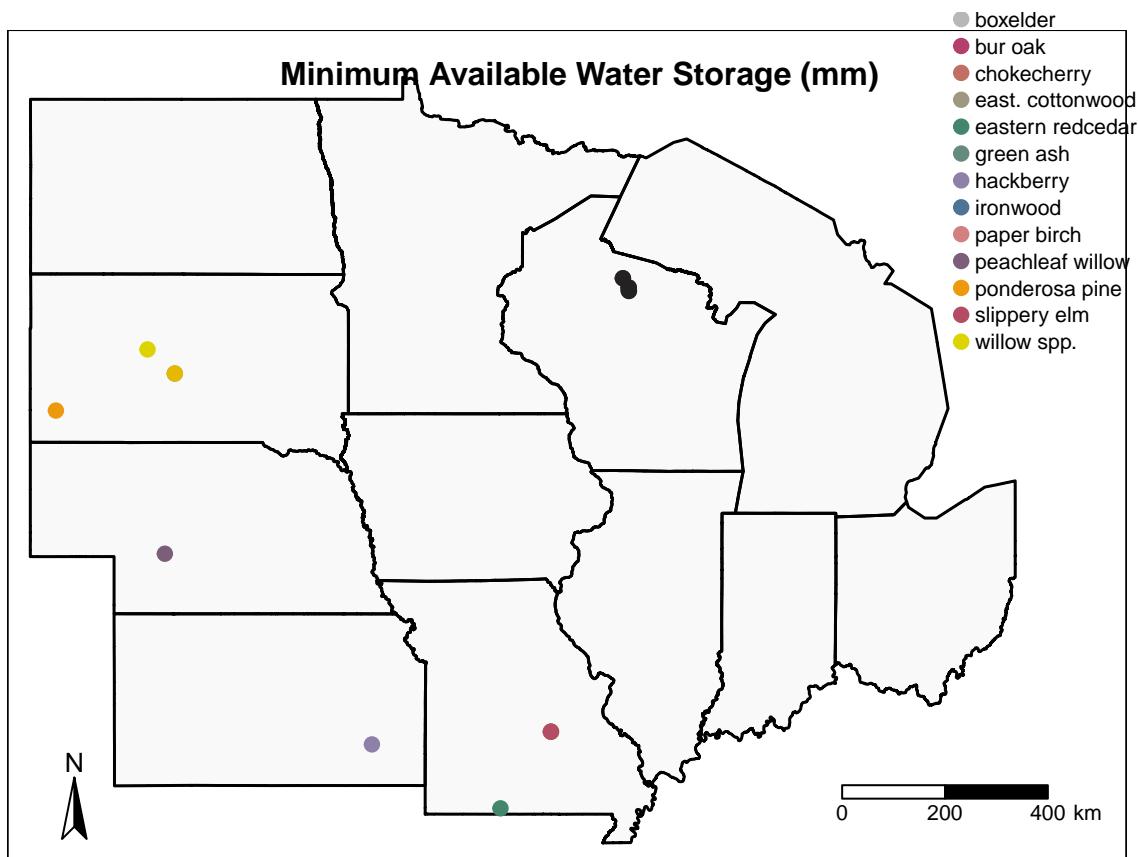
# Map - Min AWS for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new) +
  tm_polygons(col = "#f9f9f9") +
  tm_shape(state) +
  tm_borders(col = "black", lwd = 1.5) +
  # Add dots for the min temp for each state
  tm_shape(minAWSspp) +
  tm_dots(col = "Common", size = 0.25, title = "Species",
          palette = pal) +
  # Add legend and title and specify their positions
  tm_layout(legend.outside = FALSE,

```

```

legend.position = c(0.84,0.6),
legend.text.size = 0.7,
title = "Minimum Available Water Storage (mm)",
title.size = 1,
title.fontface = "bold",
title.position = c("center","top")) +
# Add a north arrow
tm_compass(type = "arrow", position = c("left", "bottom"),
size = 2) +
# Add a scale bar
tm_scale_bar(breaks = c(0,200,400), text.size = 0.7)

```



```

#### Filter to determine minimum/maximum SOC and AWS per species
# across Midwest, and plot these data

##### Maximum Available Water Storage (mm)
# Subset to only include maximum for each species
maxAWSspp <- midwmaxAWS %>%
  group_by(SPCD) %>%
  filter(., maxAWS == max(maxAWS))

# Map - Max AWS for midwest states
# Plot Midwest state boundaries in more spacious bounding box
tm_shape(state, bbox = bbox_new2) +
  tm_polygons(col = "#f9f9f9") +

```

```

tm_shape(state) +
tm_borders(col = "black", lwd = 1.5) +
# Add dots for the min temp for each state
tm_shape(maxAWSspp) +
tm_dots(col = "Common", size = 0.25, title = "Species",
palette = pal, legend.show = FALSE) +
# Add legend and title and specify their positions
tm_layout(title = "Maximum Available Water Storage (mm)",
title.size = 1,
title.fontface = "bold",
title.position = c("center", "top")) +
# Add a north arrow
tm_compass(type = "arrow", position = c("left", "bottom"),
size = 2) +
# Add a scale bar
tm_scale_bar(breaks = c(0,200,400), text.size = 0.7)

```

