Drosophila-climatology

Sam Curnow research project - Matt Olson

2023-09-29

Terra and gridded data

Most climate datasets are in a gridded format. The latest and best R package to work with gridded data is the Terra package. Libraries like this require additional, external geospatial libraries installed on your OS - e.g. GDAL, GEOS, and PROJ.4. Installing Terra will vary based on your OS. **Be sure to read** the documentation so that you properly install any dependencies.

Working with geospatial libraries can be tricky, and I am willing to help you troubleshoot if you run into issues.

```
library(terra);library(dplyr)
```

I'll also read in the csv file you shared and filter for a few key variables. Note you'll want to set your working directory or work from a project.

```
# read in data - omit NA values
df <- read.csv("climate_data_drosophila_June.csv")
df2 <- df %>% dplyr::select(Species, Subgenus, MbDNA_Male, MbDNA_Female, Tmax,Tmin) %>% na.omi
t()
head(df2)
```

```
##                    Species   Subgenus MbDNA_Male MbDNA_Female  Tmax  Tmin
## 3  Drosophila_acanthoptera                 175.4        173.9 34.07  7.01
## 4       Drosophila_affinis Sophophora      165.6        168.7 35.06  5.49
## 6     Drosophila_algonquin Sophophora      170.0        156.2 34.81 13.14
## 8     Drosophila_ananassae Sophophora      179.9        169.0 30.13  6.72
## 9        Drosophila_anceps Drosophila      171.8        159.1 29.14  9.03
## 10    Drosophila_arawakana Drosophila      187.5        165.5 28.80  3.60
```

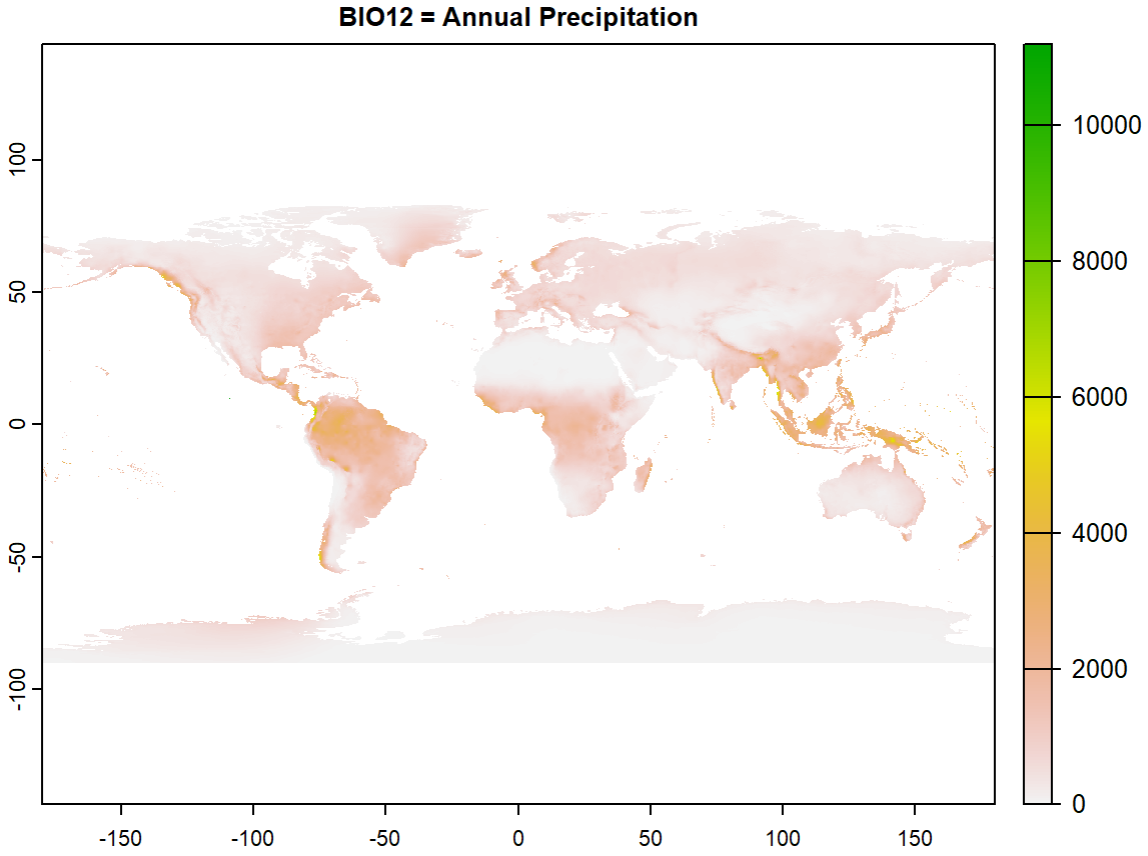
Bioclimatic data

WorldClim hosts several easy to access datasets. Although these are not necessarily the most accurate for some studies, they should be perfectly sufficient for others. R has an easy-to-use library for accessing basic geographic and bioclimatic global variables. Climate-related variables are generally averages over a 30-year time-period (1970-2000).

```
library(geodata)
# define output folder for data download
if (!file.exists("data")){dir.create("data")}
outpath = "data"
# download WorldClim global bioclimatic variables
bioclim19 <- worldclim_global('bio', res=10, path=outpath)
cat("Object has",nlyr(bioclim19),"geographic layers")
```

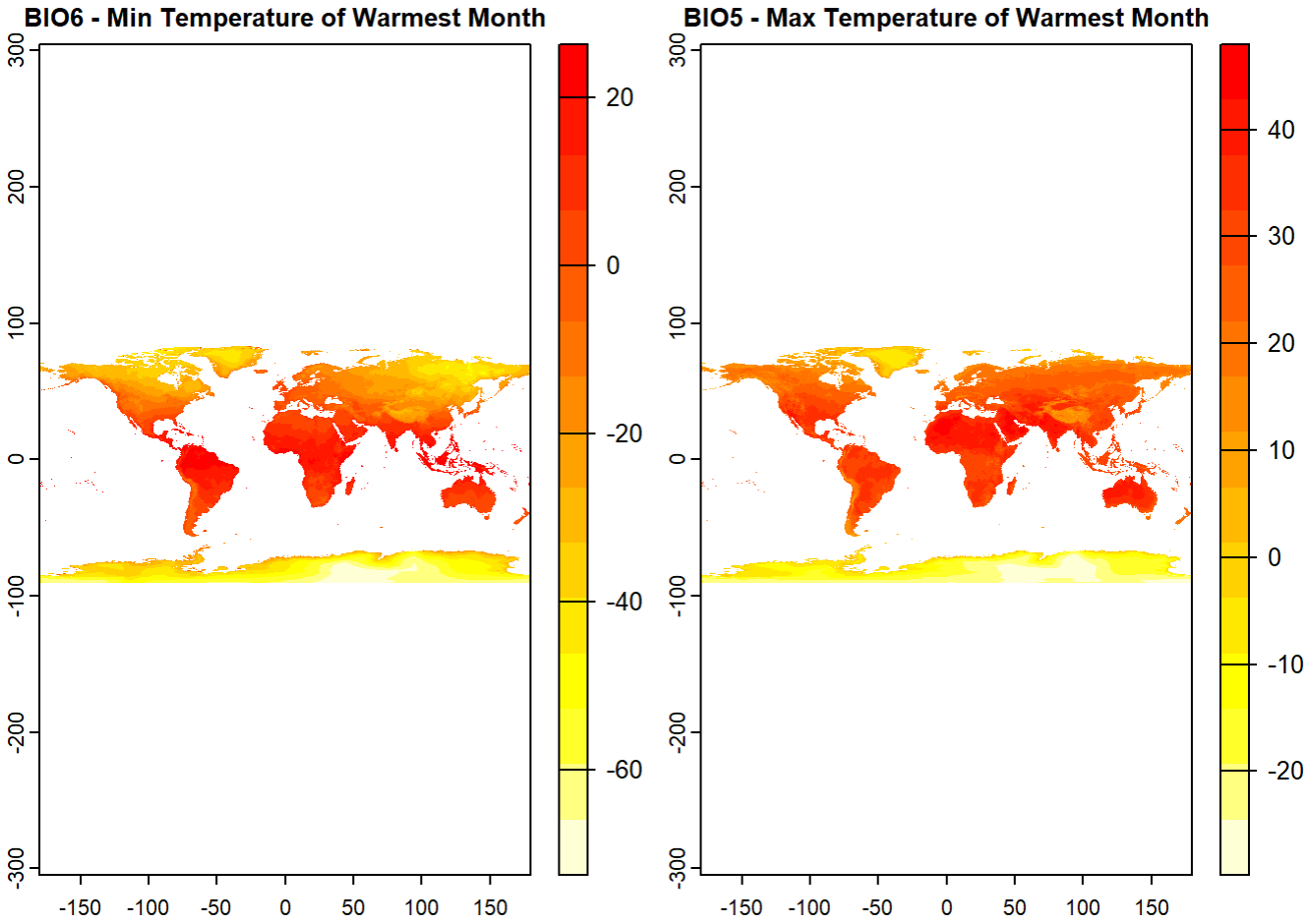
```
## Object has 19 geographic layers
```

```
plot(subset(bioclim19, 12), main="BIO12 = Annual Precipitation")
```



We can further subset these variables into individual layers

```
# subset variables of interest
tmax <- subset(bioclim19, 5);names(tmax) <- "BIO5 - Max Temperature of Warmest Month"
tmin <- subset(bioclim19, 6);names(tmin) <- "BIO6 - Min Temperature of Warmest Month"
plot(c(tmin,tmax), col=rev(heat.colors(15)))
```



Subsetting geographic datasets

Fortunately your task is relatively simple. We can just subset these gridded datasets based on the temperature thresholds in your csv file.

There are a few steps involved so I've created two functions. The code could probably be more efficient, but this is what I wrote. Several spatial functions are applied to the species in the csv file.

```
# function to apply spatial bounds for all species
spat_filter <- function(csv.file, grid.var, var.name, bounds=c("lower","upper")){
  grid.cpy <- rast(replicate(nrow(csv.file),grid.var))
  names(grid.cpy) <- csv.file$Species # rename layers
  # apply threshold from bounds arg
  lowval=-Inf;upval=-Inf
  if (bounds=='lower'){
    t.dist <- rast(sapply(1:nrow(csv.file), function(x) clamp(subset(grid.cpy, x), lower=csv.f
ile[,var.name][x], value=FALSE)))}
  else{
    t.dist <- rast(sapply(1:nrow(csv.file), function(x) clamp(subset(grid.cpy, x), upper=csv.f
ile[,var.name][x], value=FALSE)))}
  return(t.dist)
}

# function to combine layers (if needed)
grid_comb <-  function(stack.a, stack.b, original.layer.a){
```

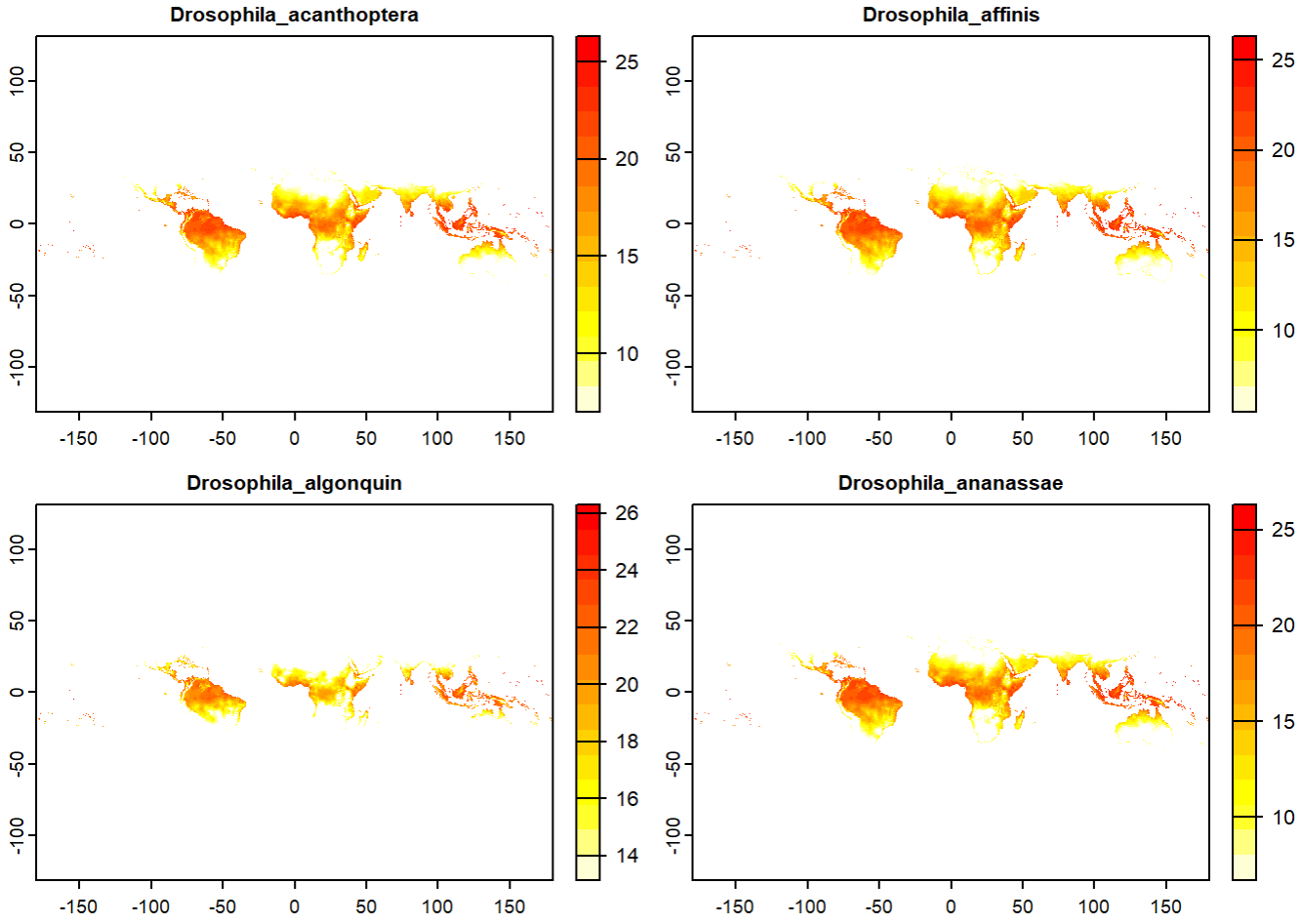
```
  # boolean operator over all land surface where both conditions meet
  comb.grid <- !is.na(stack.a) & !is.na(stack.b) & !is.na(rast(replicate(nlyr(stack.a),origina
l.layer.a)))
  return(comb.grid)
}
```

Mapping results

Now we can run these functions and plot the results for one of the variables. I'll just stick to simple map plots.

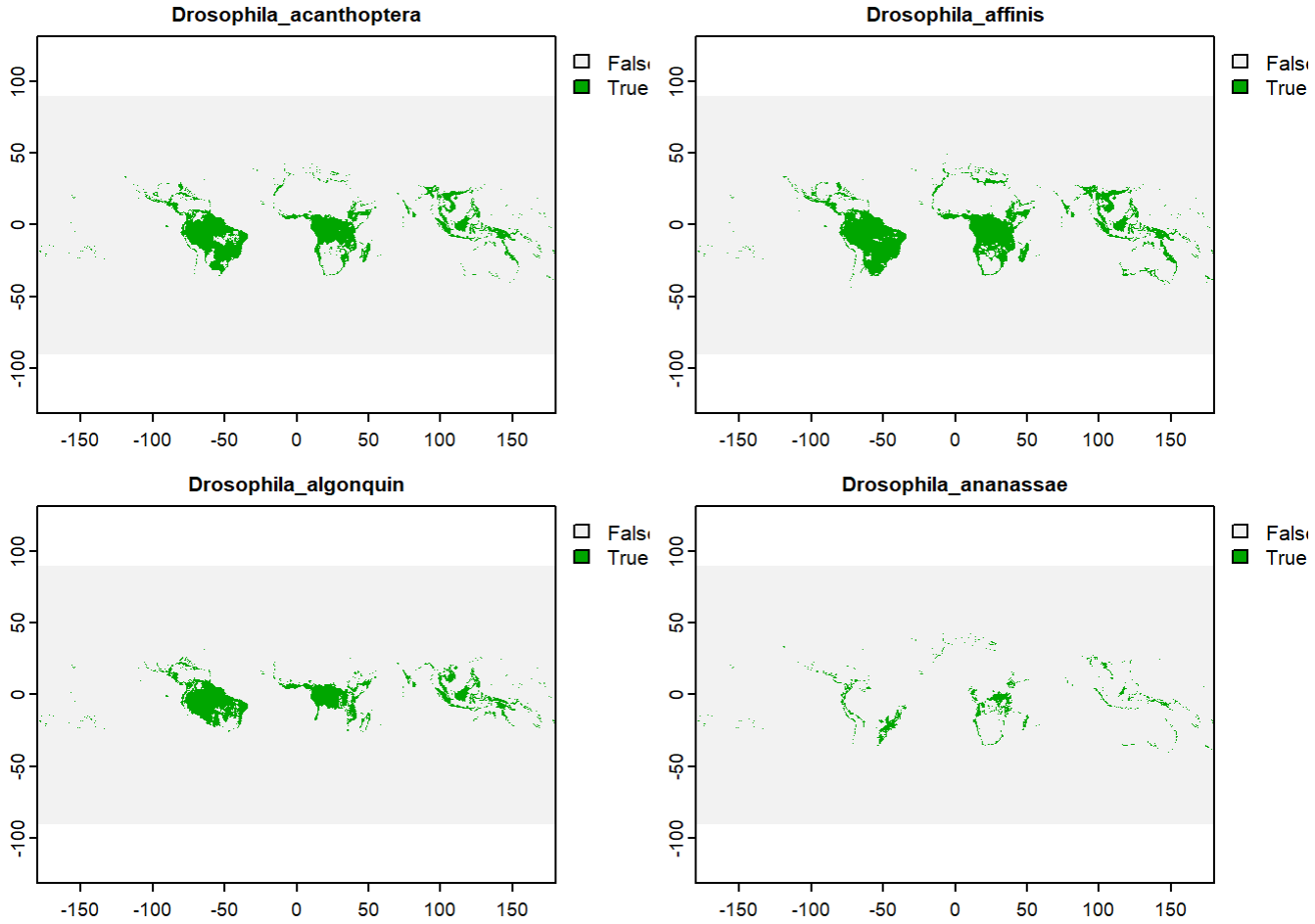
```
# run the function for both variables
tmax.stack <-  spat_filter(df2, tmax, "Tmax", bounds="upper")
tmin.stack <-  spat_filter(df2, tmin, "Tmin", bounds="lower")

# tmin distribution of first four species
plot(tmin.stack[[1:4]], col=rev(heat.colors(15)))
```



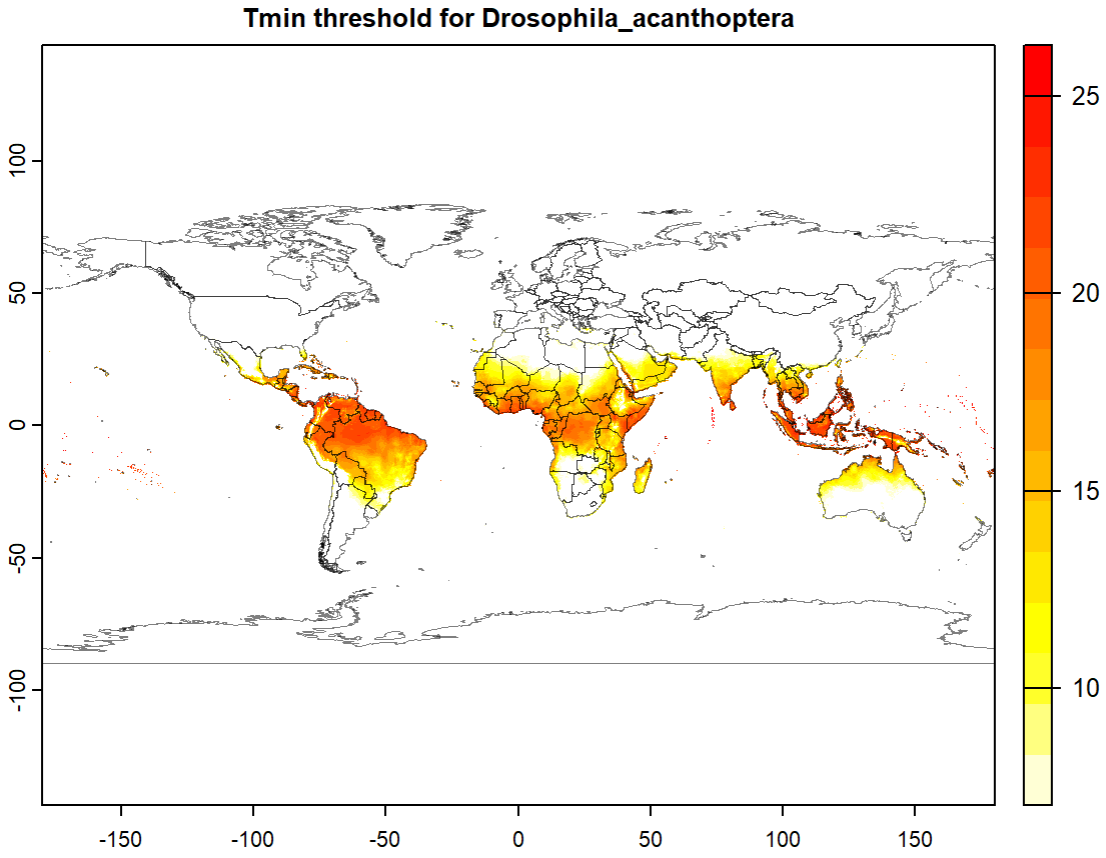
And combine the layers to show where all conditions are met.

```
# combine layers and plot distribution
t.dist <- grid_comb(tmax.stack,tmin.stack,tmax)
# plot the distribution for the first four
plot(t.dist[[1:4]])
```



You can also add country boundaries to give greater context to a map. This function is within the same package we loaded above.

```
# plot first species with world map
library(ggplot2)
worldmap <- world(resolution=5,level=0,outpath,version="latest")
plot(subset(tmin.stack,1), col=rev(heat.colors(15)), main=paste("Tmin threshold for",df2$Speci
es[1]) )
plot(worldmap, add=TRUE, lwd=0.5, border=alpha(rgb(0,0,0), 0.5))
```



Plotting results

Finally, you can generate some statistics about the spatial distribution of these species.

```
# generate some statistics from the first species only
crds(t.dist[[1]], df=TRUE) %>% filter(values(t.dist[[1]])==1) %>%
  summarize(n=n(), latmean=mean(y), latmax=max(y), latmin=min(y) )
```

```
##       n   latmean   latmax    latmin
## 1 96099 -2.541921 48.41667 -48.08333
```

We can build a function and calculate these same stats over the entire dataset, re-inserting the values into our dataframe. Again, the code could probably be more efficient. This step may take a minute, so you may want to refill your coffee.

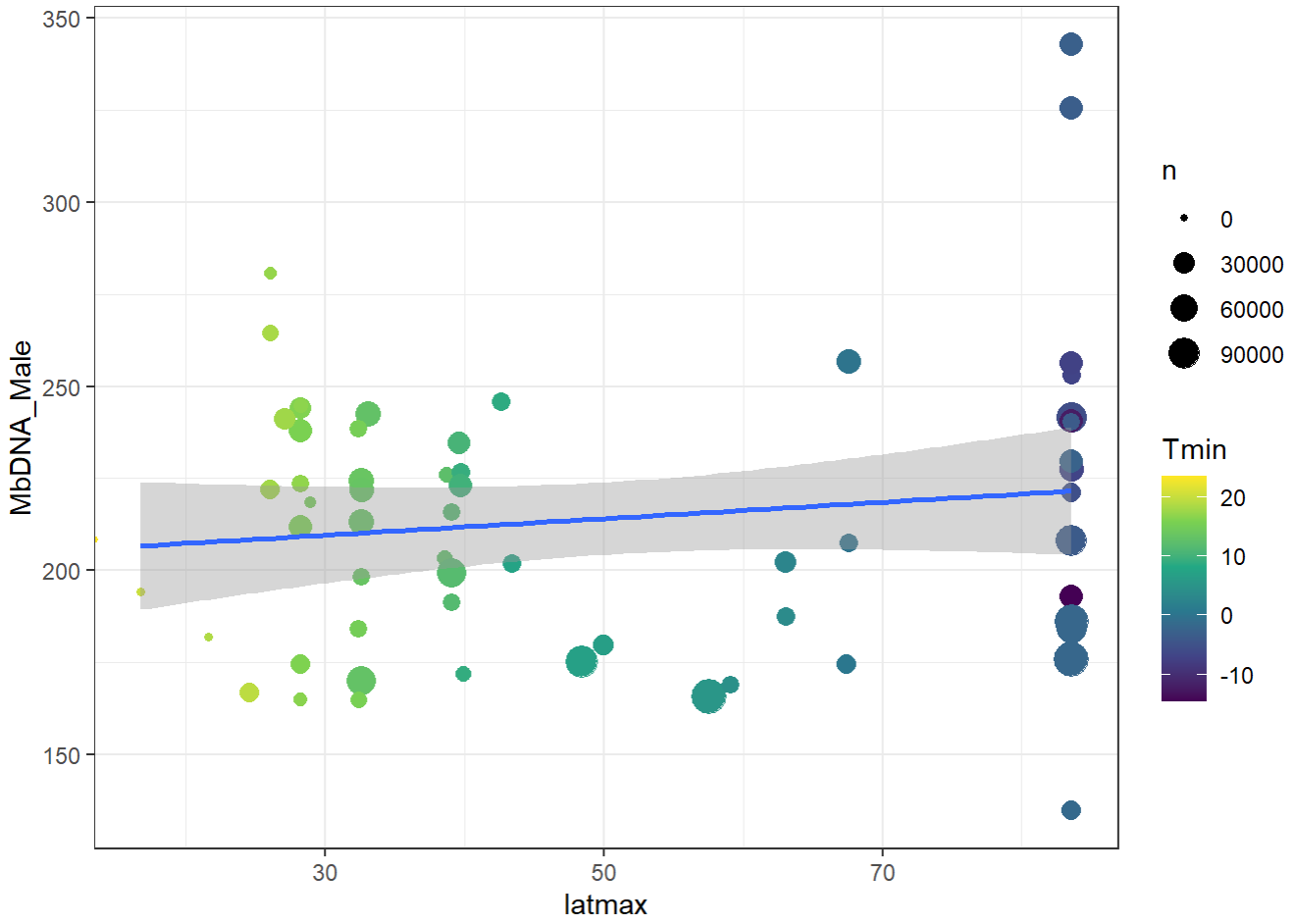
```
# function to extract stats from a given layer
lyr_summary <- function(lyr){
  rowinfo <- crds(lyr, df=TRUE) %>% filter(values(lyr)==1) %>% summarize(n=n(), latmean=mean(y
), latmax=max(y), latmin=min(y) )
  return(rowinfo)
}
# apply this to all layers
t.df2 <- sapply(1:nlyr(t.dist), function(x) lyr_summary(t.dist[[x]])  )
# insert layers into dataframe and a clean
df3 <- as.data.frame(t(t.df2)) %>% mutate(Species = df2$Species, .before=n) %>% full_join(df2,
```

```
 by='Species') %>%
  mutate(n = unlist(n), latmean = unlist(latmean),latmax = unlist(latmax),latmin = unlist(latm
in) )
head(df3)
```

```
##                   Species      n   latmean   latmax    latmin   Subgenus
## 1 Drosophila_acanthoptera  96099 -2.541921 48.41667 -48.08333
## 2      Drosophila_affinis 117630 -2.697064 57.58333 -48.08333 Sophophora
## 3    Drosophila_algonquin  74469 -1.119279 32.58333 -31.58333 Sophophora
## 4    Drosophila_ananassae  24717 -5.506854 49.91667 -48.08333 Sophophora
## 5       Drosophila_anceps  11212 -4.743222 39.91667 -40.41667 Drosophila
## 6    Drosophila_arawakana  18680 -5.992666 63.08333 -50.91667 Drosophila
##   MbDNA_Male MbDNA_Female  Tmax  Tmin
## 1      175.4        173.9 34.07  7.01
## 2      165.6        168.7 35.06  5.49
## 3      170.0        156.2 34.81 13.14
## 4      179.9        169.0 30.13  6.72
## 5      171.8        159.1 29.14  9.03
## 6      187.5        165.5 28.80  3.60
```

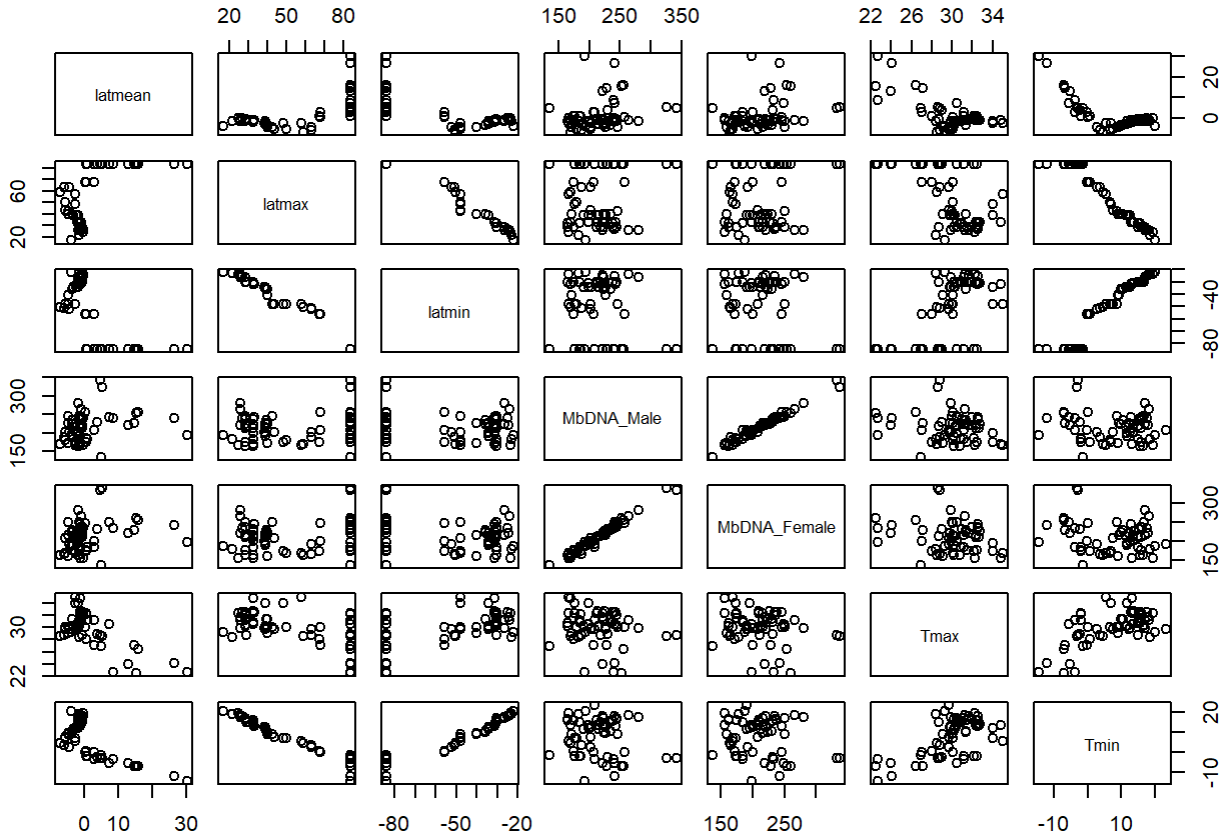
And a quick plot:

```
#quick plot of two variables
df3 %>%
  ggplot(aes(x=latmax, y=MbDNA_Male, colour=Tmin)) +
  geom_point(aes(size=n)) + scale_color_continuous(type = 'viridis') +
  geom_smooth(method='lm', formula='y~x') + theme_bw()
```



And all variables that we selected:

```
df3 %>% dplyr::select(latmean, latmax, latmin, MbDNA_Male, MbDNA_Female, Tmax, Tmin) %>% pairs
()
```



I'll let you take it from here Sam! Hopefully this is what you had in mind when you asked for help. You can replace other variables in the steps above to include annual precipitation, and even monthly temperature information. There are many other spatial questions that you could ask with such a dataset. **Good luck!**