

Supplementary FS3

16 December 2024

This file documents our reanalysis of the dataset used to examine biodiversity and conflict relationships in the Southern Philippines presented in this paper <https://doi.org/10.1038/s44185-024-00044-8>. Here, we consider spatial autocorrelation.

Files needed

```
##1. "cdh" folder containing the raster files of tree cover, tree density and forest canopy height  
##2. "landcover" folder containing the raster files of 1988 and 2020 land cover types  
##3. "mobios_tab.csv" or the biodiversity data  
##4. "conflict_tab_89-21.csv" or the conflict data  
##5. "functions.R" for calculating non-significant autocorrelated average distance  
##6. "1988_lc_classmap.csv" 1988 land cover types code  
##7. "2020_lc_classmap.csv" 2020 land cover types code  
##8. "mindanao_adm_geo.gpkg" Mindanao boundary
```

Load required packages

```
## make sure to install all packages including required dependencies prior to use  
re.lib <- c("dplyr", "tidyverse", "ggplot2", "patchwork", "GeoThinneR", "ecodist",  
         "sf", "sp", "spdep", "raster", "MASS", "terra")  
  
new_packages <- re.lib[!(re.lib %in% installed.packages()[, "Package"])]  
if (length(new_packages)) install.packages(new_packages)  
  
# load packages  
invisible(lapply(re.lib, library, character.only = TRUE))
```

Prepare datasets

```
## biodiversity data obtained from https://doi.org/10.15468/rtedgk  
## conflict data obtained from https://data.humdata.org/dataset/ucdp-data-for-philippines?  
## note that only a subset of this dataset (i.e., those from Mindanao and adjacent islands) was used  
  
## read biodiversity data  
bio <- read.csv("data/mobios_tab.csv", header = TRUE, sep = ",")  
#subset data  
bio.sub <- subset(bio, select = c(12, 23, 20, 15, 16))  
bio.sub <- bio.sub %>% filter(county!="") #remove rows with no county information  
bio.sub <- bio.sub %>% filter(class!="Malacostraca") #exclude this taxon  
bio.sub <- bio.sub %>% filter(class!="Bivalvia") #exclude this taxon  
bio.sub <- bio.sub %>% filter(class!="Gastropoda")#exclude this taxon  
  
## check province name  
## there are rows where the name of the province is uncertain as indicated by "/"  
## "Zamboanga-†del Norte" was changed to "Zamboanga del Norte" prior to import of data  
unique(bio.sub$county)  
  
## [1] "Lanao del Norte"
```

```

## [2] "South Cotabato"
## [3] "North Cotabato"
## [4] "Camiguin Island"
## [5] "Maguindanao"
## [6] "Agusan del Sur"
## [7] "Bukidnon"
## [8] "Davao Oriental"
## [9] "Davao"
## [10] "Misamis Occidental"
## [11] "Surigao del Sur"
## [12] "Davao del sur"
## [13] "Lanao del Sur"
## [14] "Sarangani"
## [15] "Misamis Oriental"
## [16] "Surigao del Norte"
## [17] "Davao de Oro"
## [18] "Davao del Norte"
## [19] "Zamboanga del Sur"
## [20] "Surigao del Norte/Agusan del Sur"
## [21] "Dinagat Island"
## [22] "Basilan"
## [23] "Tawi-Tawi"
## [24] "Sulu"
## [25] "Sultan Kudarat"
## [26] "Agusan del Norte"
## [27] "Cagayan de Oro"
## [28] "Bukidnon/Camiguin"
## [29] "Bukidnon/Misamis Oriental/Iligan"
## [30] "Zamboanga del Norte"
## [31] "Zamboanga Sibugay"
## [32] "Misamis Occidental/Misamis Oriental/Camiguin/Bukidnon"
## [33] "Zamboanga City"
## [34] "General Santos"
## [35] "North Cotabato/Davao City"

## remove uncertain province names
bio.sub <- bio.sub %>% filter(!grepl('/', county))
head(bio.sub)

```

	county	class	scientificName	decimalLatitude
## 1	Lanao del Norte	Aves	Buceros hydrocorax	8.2122
## 2	Lanao del Norte	Mammalia	Eonycteris spelaea	8.3218
## 3	Lanao del Norte	Mammalia	Hipposideros diadema	8.3218
## 4	Lanao del Norte	Mammalia	Pipistrellus javanicus	8.3218
## 5	Lanao del Norte	Mammalia	Rhinolophus inops	8.3218
## 6	Lanao del Norte	Mammalia	Rousettus amplexicaudatus	8.3218
##		decimalLongitude		
## 1		124.3835		
## 2		124.2940		
## 3		124.2940		
## 4		124.2940		
## 5		124.2940		
## 6		124.2940		

```

## write data
write.csv(bio.sub, "bio.sub.csv")

## read conflict data
con <- read.csv("data/conflict_tab_89-21.csv", header = TRUE, sep = ",")

## subset to get relevant data
con.sub <- subset(con, select = c(1,3,15,18,30,32,33))
head(con.sub)

##      id year           side_a side_b      PROVINCE latitude
## 1 124029 1989 Government of Philippines    CPP   Cotabato 6.951400
## 2 125199 1989 Government of Philippines    CPP   Maguindanao 7.270000
## 3 123871 1989 Government of Philippines    CPP Surigao del Sur 8.666670
## 4 122819 1989 Government of Philippines    CPP Agusan del Sur 8.500000
## 5 123283 1989 Government of Philippines    CPP Lanao del Norte 8.000000
## 6 123344 1989 Government of Philippines    CPP Davao Oriental 7.081413
##   longitude
## 1 125.0992
## 2 124.3106
## 3 126.0000
## 4 125.8333
## 5 124.0000
## 6 126.1299

## export filtered conflict data
write.csv(con.sub, "con.sub.csv")

```

Address spatial autocorrelation in both biodiversity and conflict data

```

## this part was not done in the paper, which may affect the analysis
## first, filter species with duplicate coordinates
bio.unique <- bio.sub %>%
  group_by(scientificName) %>%
  distinct(decimalLongitude, decimalLatitude, .keep_all = TRUE)

## then find the non-significant autocorrelated average distance to thin the data\
## based on the relationship between land cover types

## use the spatialautocorrelation function
## from https://github.com/jorgeassis/spatialAutocorrelation
source("data/functions.R")

## read data
files <- list.files("data/landcover/", pattern = "tif", full.names=TRUE)
raster.files <- lapply(files, raster)

raster.files

## [[1]]
## class       : RasterLayer
## dimensions : 736, 903, 664608  (nrow, ncol, ncell)
## resolution : 0.008000454, 0.007996592  (x, y)
## extent     : 119.3807, 126.6051, 4.58649, 10.47198  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs
## source     : 1988_lc_rast1km.tif

```

```

## names      : X1988_lc_rast1km
##
##
## [[2]]
## class      : RasterLayer
## dimensions : 736, 903, 664608  (nrow, ncol, ncell)
## resolution : 0.008000454, 0.007996592  (x, y)
## extent     : 119.3807, 126.6051, 4.58649, 10.47198  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs
## source     : 2020_lc_rast1km.tif
## names      : X2020_lc_rast1km

## make rasters to have common extent
common_extent <- do.call(union, lapply(raster.files, extent))

raster.files <- lapply(raster.files, function(r) {
  extend(r, common_extent)
  extent(r) <- common_extent
  r
})

## check extent
lapply(raster.files, extent)

## [[1]]
## class      : Extent
## xmin       : 119.3807
## xmax       : 126.6051
## ymin       : 4.58649
## ymax       : 10.47198
##
## [[2]]
## class      : Extent
## xmin       : 119.3807
## xmax       : 126.6051
## ymin       : 4.58649
## ymax       : 10.47198

## resample raster files to have a uniform dimension/resolution
standard <- raster.files[[1]]
raster.files <- lapply(raster.files, function(r) {
  resample(r, standard, method = "bilinear")
})

## check resolution
lapply(raster.files, res)

## [[1]]
## [1] 0.008000454 0.007996592
##
## [[2]]
## [1] 0.008000454 0.007996592

## stack the raster files
lc.data <- stack(raster.files)

```

```

## get longitude and latitude columns
bio.coor <- as.data.frame(subset(bio.unique, select = c("decimalLongitude", "decimalLatitude")))

## define the distance class (in km)
autocorrelationClassDistance <- 2

## define the maximum distance (in km)
autocorrelationMaxDistance <- 10

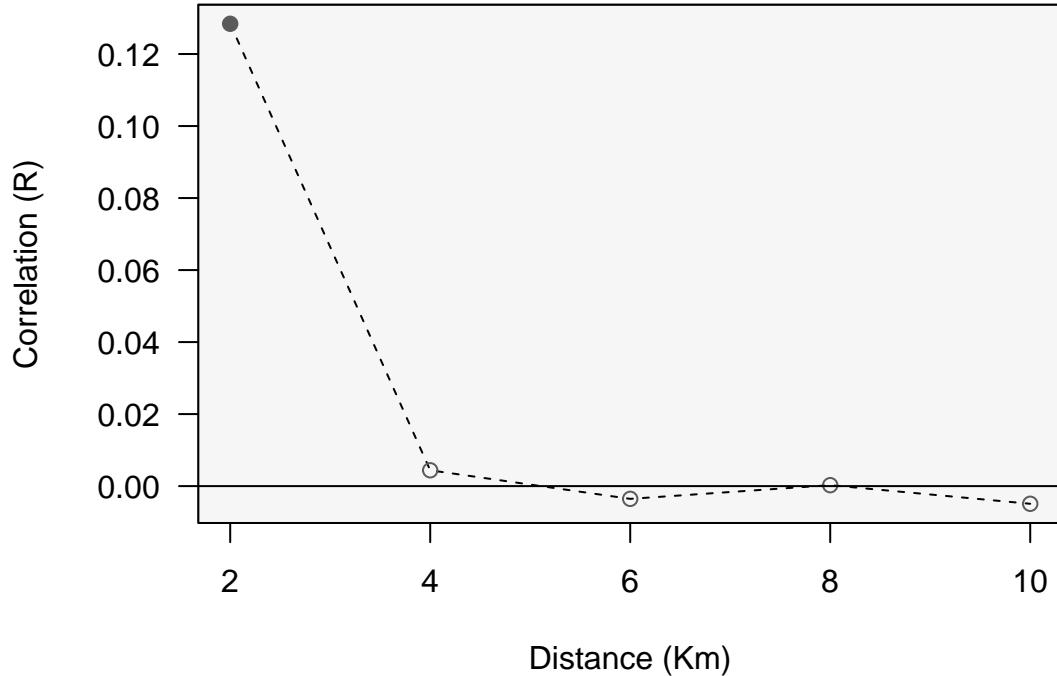
## define the significance level of the test
autocorrelationSignif <- 0.05

distanceUncorr <- data.frame(Predictor=names(lc.data), Distance=NA)

for( i in 1:length(names(lc.data)))
  distanceUncorr[i,2] <- spatialAutocorrelation(occurrenceRecords=bio.coor,subset(lc.data,i),
    autocorrelationClassDistance,autocorrelationMaxDistance,autocorrelationSignif)

##
##
## More than 1000 occurrence records.
## Using a maximum of 1000 random records.

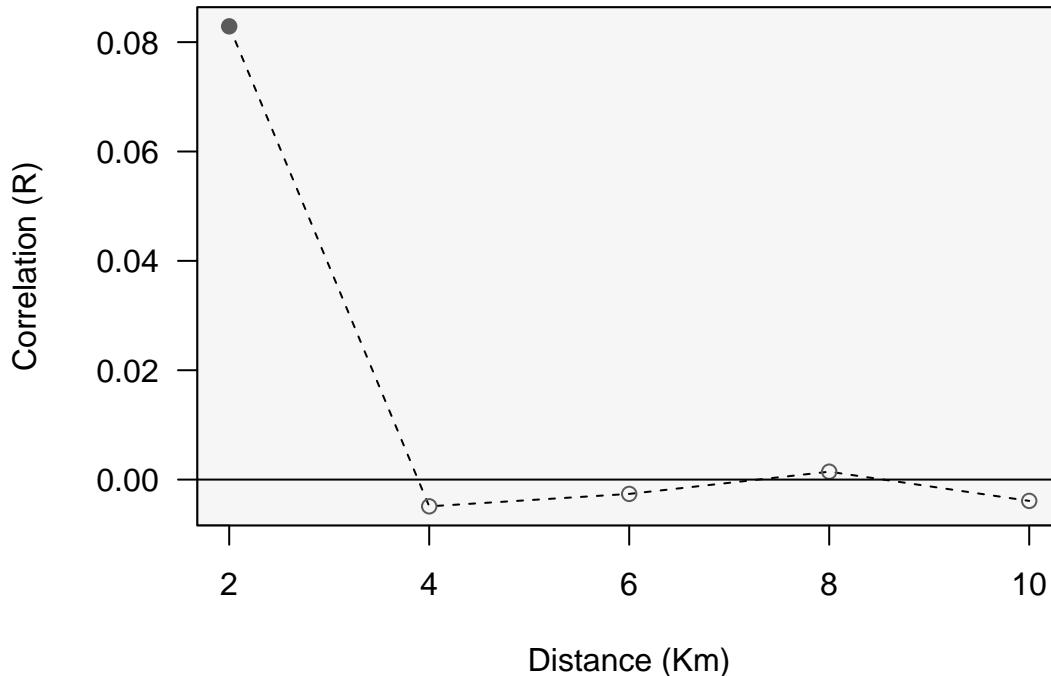
```



```

##
##
## First non-correlated distance: 4 km
##
## More than 1000 occurrence records.
## Using a maximum of 1000 random records.

```



```
##  
##  
## First non-correlated distance: 4 km  
meanCorrDistance <- mean(distanceUncorr[,2])  
  
meanCorrDistance  
  
## [1] 4  
## thin every species data using the mean distance  
## species name  
sp.name <- unique(bio.unique$scientificName)  
  
## create an empty list to store the results  
thin.data <- list()  
  
## go over each species  
for (spp in sp.name) {  
  bio.spp <- subset(bio.unique, scientificName == spp)  
  bio.thin <- thin_points(  
    data = bio.spp,  
    long_col = "decimalLongitude",  
    lat_col = "decimalLatitude",  
    method = "brute_force",  
    thin_dist = meanCorrDistance, # thinning distance in km  
    trials = 1, # number of replicates  
    all_trials = TRUE, # return all trials  
    seed = 123 # seed for reproducibility  
  )  
  
  ## convert results to a data frame  
  bio.thin.df <- as.data.frame(bio.thin[1])
```

```

## add the species name
bio.thin.df$scientificName <- spp

## append the data to the list
thin.data[[spp]] <- bio.thin.df
}

## combine all thinned
bio.thin.df <- do.call(rbind, thin.data)

## write to a csv
write.csv(bio.thin.df, "bio.thin.csv")

## convert thin data to sf object
bio.thin.sf <- st_as_sf(bio.thin.df, coords = c("decimalLongitude", "decimalLatitude"), crs = 4326)

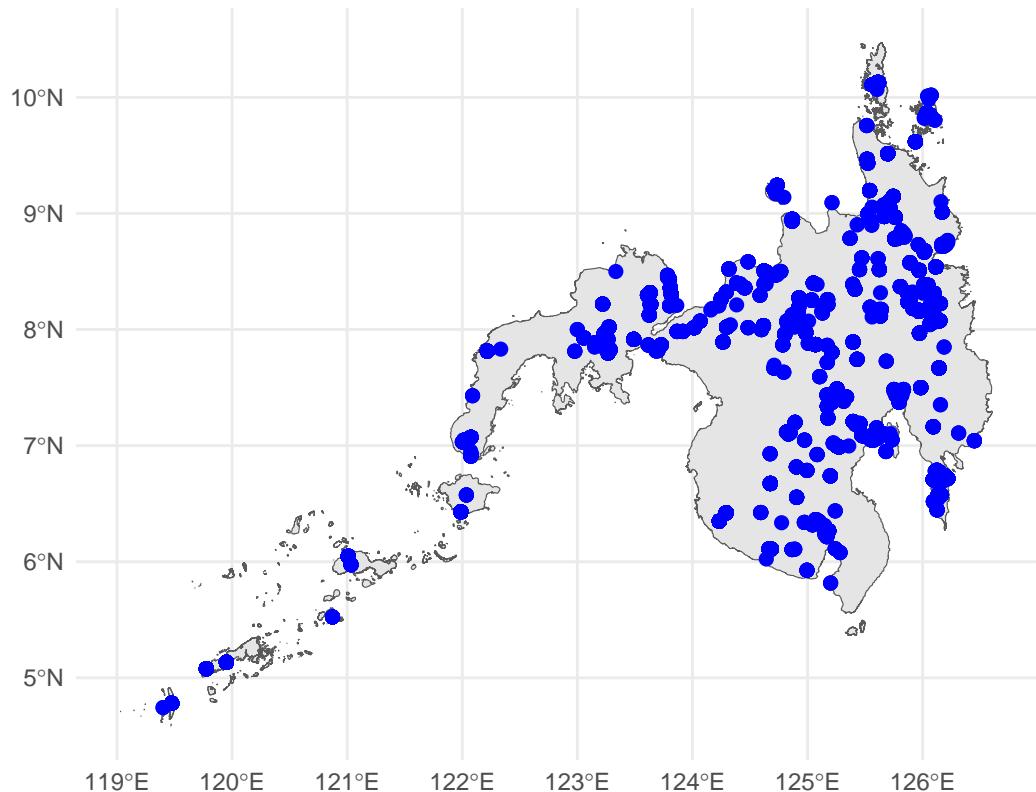
## plot biodiversity data
## read Mindanao admin boundary
ph.poly <- st_read("data/mindanao_adm_geo.gpkg")

## Reading layer `mindanao_adm__single_parts' from data source
##   `/home/christian/Downloads/Supplementary_FS3/data/mindanao_adm_geo.gpkg'
##   using driver `GPKG'
## Simple feature collection with 1838 features and 3 fields
## Geometry type: POLYGON
## Dimension:      XY
## Bounding box:  xmin: 119.0302 ymin: 4.587184 xmax: 126.6051 ymax: 10.47158
## Geodetic CRS:  WGS 84

## plot
ggplot() +
  geom_sf(data = ph.poly) +
  geom_sf(data = bio.thin.sf, color = "blue", size = 2) +
  theme_minimal() +
  labs(title = "Thinned Biodiversity Data")

```

Thinned Biodiversity Data



```
## check the number of records (original vs thinned data)
nrow(bio.sub) # original data

## [1] 10038
nrow(bio.thin.df) # thinned data

## [1] 8074

## apply the same approach to conflict data
con.unique <- con.sub %>%
  group_by(PROVINCE) %>%
  distinct(longitude, latitude, .keep_all = TRUE)

## get longitude and latitude columns
con.coor <- as.data.frame(subset(con.unique, select = c("longitude", "latitude")))

## define the distance class
autocorrelationClassDistance <- 2

## define the maximum distance (in km)
autocorrelationMaxDistance <- 10

## define the significance level of the test
autocorrelationSignif <- 0.05

distanceUncorr <- data.frame(Predictor=names(lc.data), Distance=NA)

for( i in 1:length(names(lc.data)))
```

```

distanceUncorr[i,2] <- spatialAutocorrelation(occurrenceRecords=con.coor,subset(lc.data,i),
    autocorrelationClassDistance,autocorrelationMaxDistance,autocorrelationSignif)

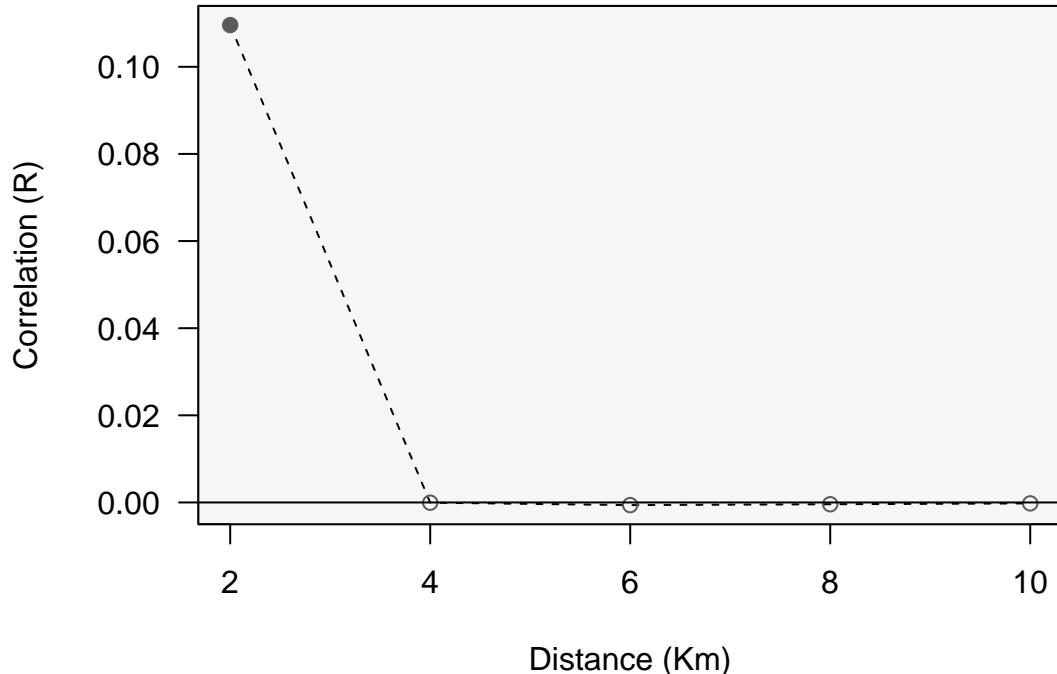
##  

##  

## More than 1000 occurrence records.  

## Using a maximum of 1000 random records.

```



```

##  

##  

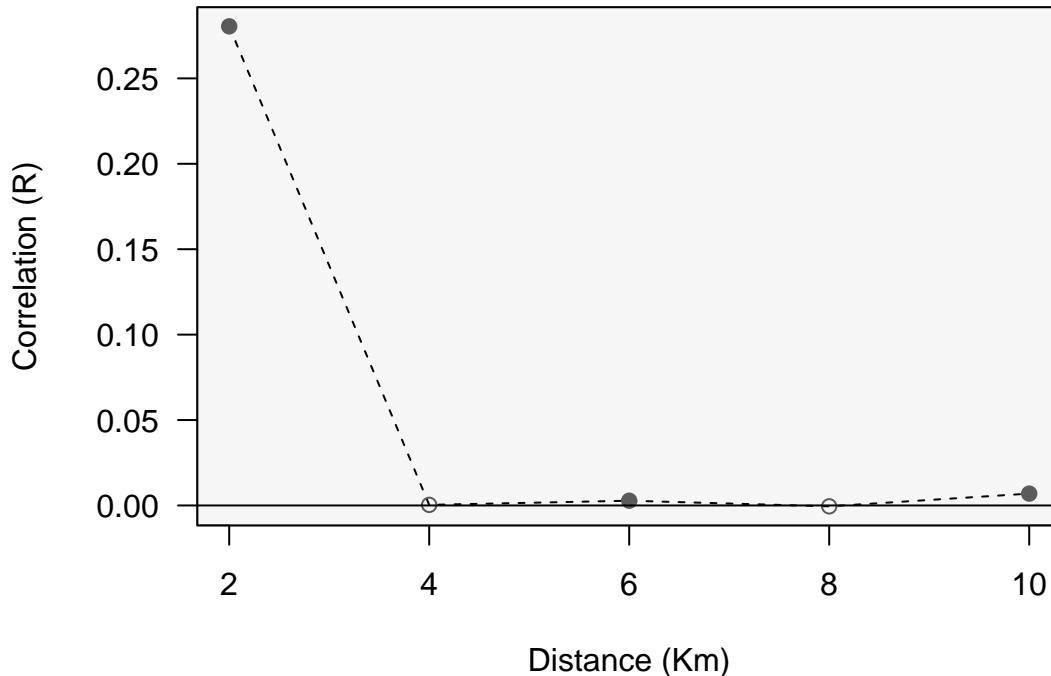
## First non-correlated distance: 4 km  

##  

## More than 1000 occurrence records.  

## Using a maximum of 1000 random records.

```



```
##  
##  
## First non-correlated distance: 4 km  
meanCorrDistance <- mean(distanceUncorr[,2])  
  
meanCorrDistance  
  
## [1] 4  
## thin every province data using the mean distance  
## use province name  
prov.name <- unique(con.unique$PROVINCE)  
  
## create an empty list to store the results  
thin.data <- list()  
  
## go over each province  
for (prov in prov.name) {  
  con.prov <- subset(con.unique, PROVINCE == prov)  
  con.thin <- thin_points(  
    data = con.prov,  
    long_col = "longitude",  
    lat_col = "latitude",  
    method = "brute_force",  
    thin_dist = meanCorrDistance, # thinning distance in km  
    trials = 1, # number of replicates  
    all_trials = TRUE, # return all trials  
    seed = 123 # seed for reproducibility  
  )  
  
  ## convert results to a data frame  
  con.thin.df <- as.data.frame(con.thin[1])
```

```

## add the province name
con.thin.df$PROVINCE <- prov

## append the data to the list
thin.data[[prov]] <- con.thin.df
}

## combine all thinned
con.thin.df <- do.call(rbind, thin.data)

## write to a csv
write.csv(con.thin.df, "con.thin.csv")

## convert thin data to sf object
con.thin.sf <- st_as_sf(con.thin.df, coords = c("longitude", "latitude"), crs = 4326)

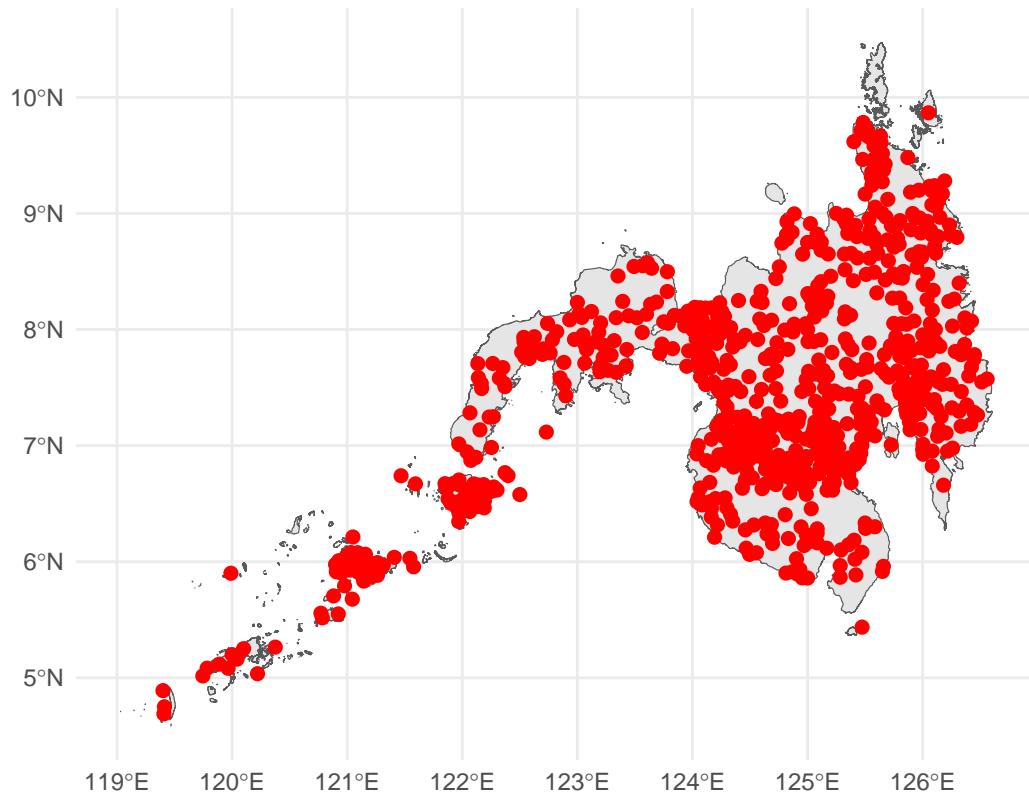
## plot conflict data
## read Mindanao admin boundary
ph.poly <- st_read("data/mindanao_adm_geo.gpkg")

## Reading layer `mindanao_adm__single_parts' from data source
##   `/home/christian/Downloads/Supplementary_FS3/data/mindanao_adm_geo.gpkg'
##   using driver `GPKG'
## Simple feature collection with 1838 features and 3 fields
## Geometry type: POLYGON
## Dimension:      XY
## Bounding box:  xmin: 119.0302 ymin: 4.587184 xmax: 126.6051 ymax: 10.47158
## Geodetic CRS:  WGS 84

## plot
ggplot() +
  geom_sf(data = ph.poly) +
  geom_sf(data = con.thin.sf, color = "red", size = 2) +
  theme_minimal() +
  labs(title = "Thinned Conflict Data")

```

Thinned Conflict Data



```
## check the number of records (original vs thinned data)
nrow(con.sub) # original data

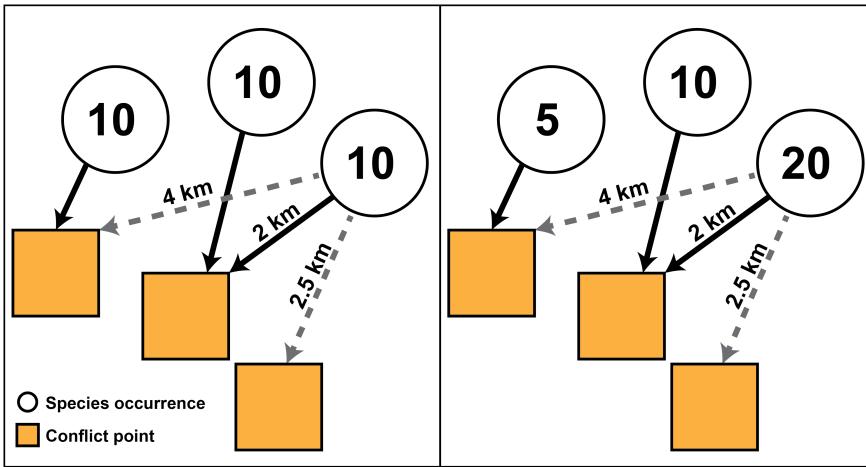
## [1] 2583
nrow(con.thin.df) # thinned data

## [1] 748

Count recorded species per taxon in each point and within province

## the procedures for conducting species counts in each site were not mentioned in the paper
## according to the paper, analysis was done at the provincial level
## there could be two ways that species counts were done (see image below)
## this part is extremely important, which the authors failed to discuss
## the left panel of the figure shows that species counts were aggregated at the provincial level
## each point (within the province) receives the same value of species count (or species richness)
## the right panel shows that each point has a unique number of species count
## note that these values are crucial for other downstream analyses
## particularly when relationships of species counts
## and distance to the nearest conflict sites are considered
## we generated the values for these two scenarios below
## and test whether any of them would match the results presented in the paper

knitr::include_graphics("images/bioconflict.png")
```



```
## count unique records (i.e., species richness) per point
bio.data.sum <- bio.thin.df %>%
  group_by(county, class, decimalLatitude, decimalLongitude) %>%
  mutate(NumbSp = n_distinct(scientificName))

head(bio.data.sum)

## # A tibble: 6 x 6
## # Groups:   county, class, decimalLatitude, decimalLongitude [6]
##   county      class scientificName decimalLatitude decimalLongitude NumbSp
##   <chr>       <chr> <chr>           <dbl>            <dbl>    <int>
## 1 Lanao del Norte Aves  Buceros hydro~     8.21             124.     1
## 2 North Cotabato  Aves  Buceros hydro~     6.82             125.     23
## 3 Agusan del Sur Aves  Buceros hydro~     8.51             126.     76
## 4 Davao de Oro    Aves  Buceros hydro~     7.67             126.     59
## 5 Sultan Kudarat Aves  Buceros hydro~     6.42             124.     85
## 6 Agusan del Norte Aves  Buceros hydro~     9.47             126.     24

## write to a csv
write.csv(bio.data.sum, "bio.data.sum.csv")

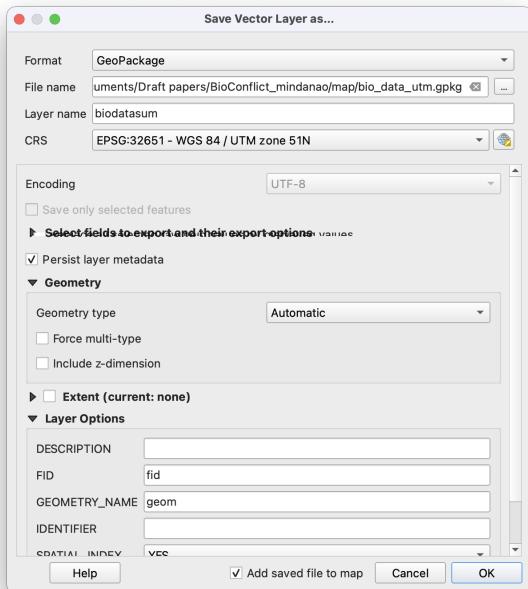
## count species records per province (this lumps all the data within each province,
## thus each point will have a single species richness)
prov.bio.data.sum <- bio.thin.df %>%
  group_by(county, class) %>%
  mutate(NumbSp = n_distinct(scientificName))
head(prov.bio.data.sum)

## # A tibble: 6 x 6
## # Groups:   county, class [6]
##   county      class scientificName decimalLatitude decimalLongitude NumbSp
##   <chr>       <chr> <chr>           <dbl>            <dbl>    <int>
## 1 Lanao del Norte Aves  Buceros hydro~     8.21             124.     1
## 2 North Cotabato  Aves  Buceros hydro~     6.82             125.    111
## 3 Agusan del Sur Aves  Buceros hydro~     8.51             126.    196
## 4 Davao de Oro    Aves  Buceros hydro~     7.67             126.     67
## 5 Sultan Kudarat Aves  Buceros hydro~     6.42             124.     85
## 6 Agusan del Norte Aves  Buceros hydro~     9.47             126.     65
```

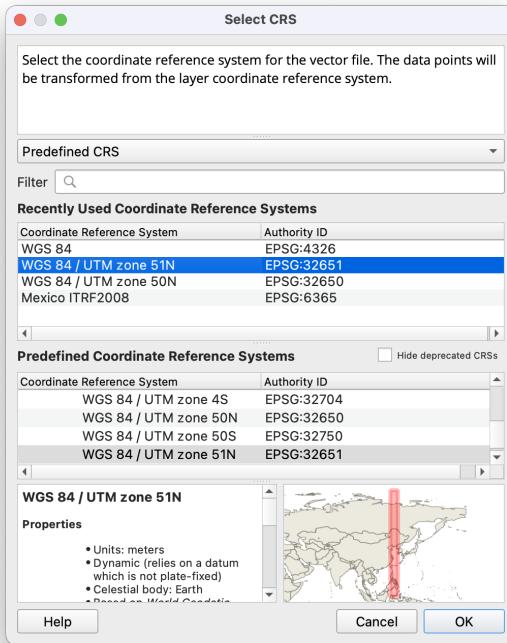
```
## write to a csv  
write.csv(prov.bio.data.sum, "prov.data.sum.csv")
```

Calculate distance of species record to the nearest conflict sites using QGIS

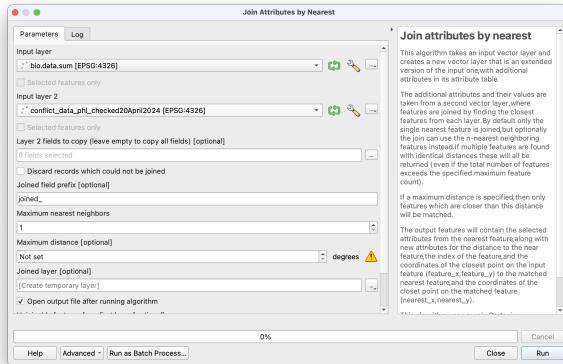
```
## in QGIS, import the "bio.data.sum.csv" and "con.thin.csv" from R as delimited text files.  
## export these files as GeoPackage or Shapefile to convert the data to meters prior to calculation  
## use the CRS UTM Zone 51N  
knitr:::include_graphics("images/qgis_geo.png")
```



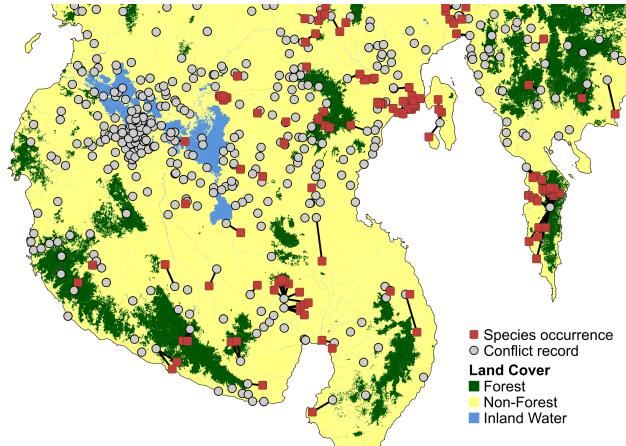
```
knitr:::include_graphics("images/qgis_utm.png")
```



```
## import back the newly saved data to QGIS
## get the distance of nearest conflict areas using the "Joint attributes by nearest" plugin
## QGIS > Processing Toolbox > Join attributes by nearest
knitr:::include_graphics("images/qgis_join.png")
```



```
## distance can be calculated as well without joining the attributes using the "Distance to nearest hub"
## after calculating the distance, a new layer will be created.
## the new layer contains the calculated distance in the attribute table
## this also includes the x and y coordinates of the nearest conflict site
## then export this new layer as a csv file so we can use the data in R for other analysis
## repeat the entire process for provincial data (i.e., prov.bio.data.sum.csv) if necessary
## figure below shows the nearest conflict point to species record (black line)
## notice the remainder of conflict points are not included
knitr:::include_graphics("images/hub.png")
```



Calculate average distance

```

## from now on, we'll solely use specie richness data per point
## read the file with calculated distance from QGIS
bio.con.dist <- read.csv("data/bio.con.dist.csv", header = TRUE, sep = ",")

## get average distance by province for each taxon (i.e., class column)
mean.dist <- bio.con.dist %>%
  group_by(county, class, NumbSp, decimalLatitude, decimalLongitude) %>%
  summarize(meanDistance = mean(distance, na.rm = TRUE))

## `summarise()` has grouped output by 'county', 'class', 'NumbSp',
## 'decimalLatitude'. You can override using the `.`groups` argument.
head(mean.dist)

## # A tibble: 6 x 6
## # Groups:   county, class, NumbSp, decimalLatitude [6]
#>   county       class   NumbSp decimalLatitude decimalLongitude meanDistance
#>   <chr>        <chr>    <int>      <dbl>            <dbl>           <dbl>
#> 1 Agusan del Norte Amphib~     2        9.05          126.        8659.
#> 2 Agusan del Norte Amphib~     2        9.07          126.        6472.
#> 3 Agusan del Norte Amphib~     5        9.00          126.        5108.
#> 4 Agusan del Norte Amphib~    10        9.07          126.        5517.
#> 5 Agusan del Norte Amphib~    18        9.05          126.        8052.
#> 6 Agusan del Norte Amphib~    20        8.97          126.        1594.

## write to a csv
write.csv(mean.dist, "mean.distance.csv")

```

Plot species record vs average distance

```

## get taxa names
key.lab <- unique(mean.dist$class)

## assign point shapes and colors
plot.color <- c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
point.shape <- c(0,1,2,3,4,5,6,7,8,9,10)

## plot species richness vs mean distance
p <- ggplot(mean.dist, (aes(x=meanDistance, y=NumbSp, color=class, shape=class))) + theme_bw() +
  theme(axis.text=element_text(size=15), axis.title=element_text(size=15, face="bold"),

```

```

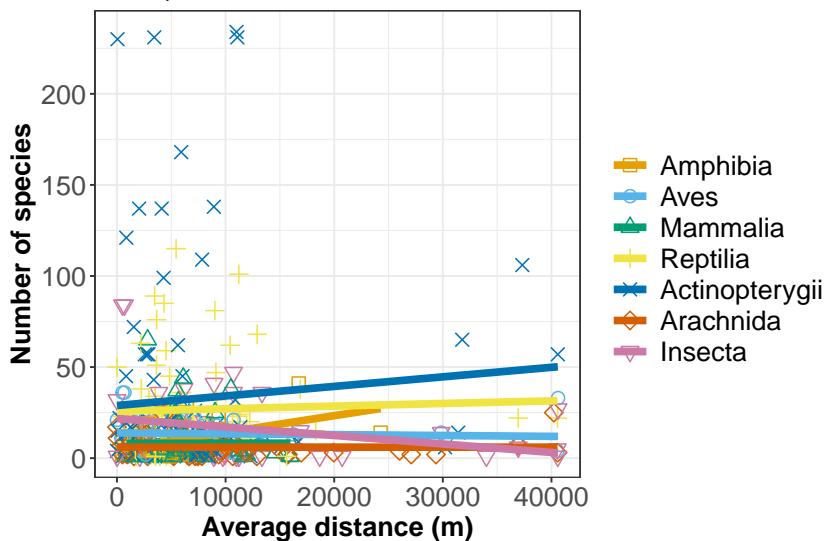
    legend.key.width = unit(0.8, units = "cm"), legend.text = element_text(size=15),
    legend.key=element_blank()) +
  ylab("Number of species") +
  xlab("Average distance (m)") +
  labs(title = "Point species richness") +
  geom_point(size = 3, stroke = 0.6) +
  geom_smooth(method=lm, se=FALSE, fullrange=FALSE, linewidth = 2) +
  scale_color_manual(name = "", labels = key.lab, values = plot.color) +
  scale_shape_manual(name = "", labels = key.lab, values = point.shape)

```

p

```
## `geom_smooth()` using formula = 'y ~ x'
```

Point species richness



```
## write plot to a tiff file
```

```
tiff("sp_dist.tif", res=300, width = 7, height = 7, unit="in")
ggplot(mean.dist, (aes(x=meanDistance, y=NumbSp, color=class, shape=class))) + theme_bw() +
  theme(axis.text=element_text(size=15), axis.title=element_text(size=15, face="bold"),
        legend.key.width = unit(0.8, units = "cm"), legend.text = element_text(size=15),
        legend.key=element_blank()) +
  ylab("Number of species") +
  xlab("Average distance (m)") +
  labs(title = "Lumped species richness") +
  geom_point(size = 3, stroke = 0.6) +
  geom_smooth(method=lm, se=FALSE, fullrange=FALSE, linewidth = 2) +
  scale_color_manual(name = "", labels = key.lab, values = plot.color) +
  scale_shape_manual(name = "", labels = key.lab, values = point.shape)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
dev.off()
```

```
## RStudioGD
##      2
```

Get number of conflict sites per province

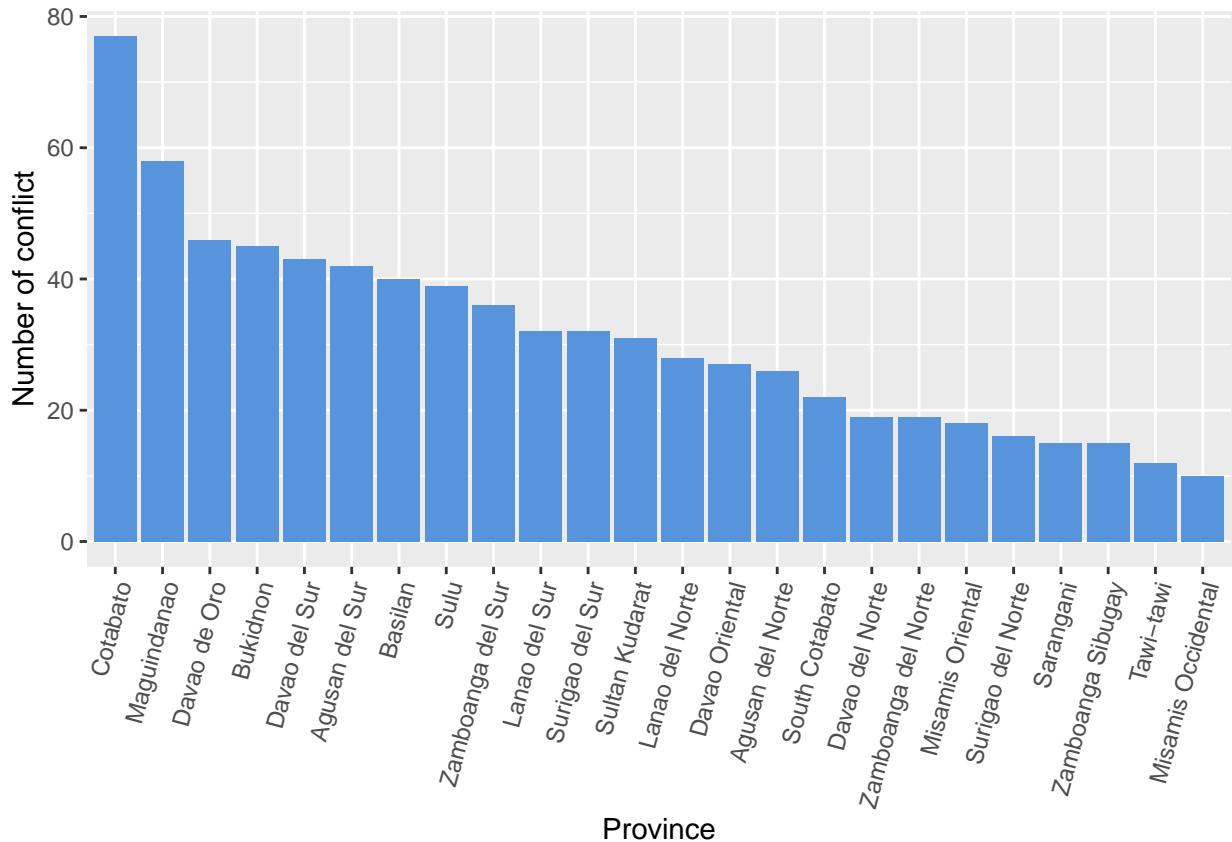
```

## we are not sure how the frequency of conflicts were scored in the paper as it was not mentioned
## so lets assume that count was done per province
con.sub.thin.df2 <- con.thin.df
con.sub.thin.df2 <- con.sub.thin.df2 %>%
  filter(PROVINCE!="") # filter empty rows

con.freq <- con.sub.thin.df2 %>%
  group_by(PROVINCE) %>%
  count()
## write to a csv
write.csv(con.freq, "conflict.frequency.rows.csv")

## plot number of conflict
ggplot(data=con.freq, aes(x=reorder(PROVINCE, -n), y=n)) +
  geom_bar(stat="identity", fill="#5794db") +
  xlab("Province") +
  ylab("Number of conflict") +
  guides(x = guide_axis(angle = 75))

```



```

## combine biodiversity and conflict data (per point)
perpoint.comb <- mean.dist %>% mutate(conflictFreq = if_else(county == "Agusan del Norte", 24,
  if_else(county == "Agusan del Sur", 40,
  if_else(county == "Basilan", 31,
  if_else(county == "Bukidnon", 44,
  if_else(county == "Cagayan de Oro", 0,
  if_else(county == "Camiguin Island", 0,
  if_else(county == "Cotabato", 64,

```

Perform Spatial Mixed Models

```

## perform spatial mixed model with tree cover, tree density and forest canopy height
## these data can be obtained from the links below
## forest canopy height: https://glad.umd.edu/dataset/gedi
## tree density: https://elischolar.library.yale.edu/yale\_fes\_data/1/
## tree cover: https://earthenginepartners.appspot.com/science-2013-global-forest/download\_v1.7.html

## read the file with calculated distance from QGIS
bio.con.dist <- read.csv("data/bio.con.dist.csv", header = TRUE, sep = ",")

lon.lat <- subset(bio.con.dist, select = c("decimalLongitude", "decimalLatitude"))

## read raster files
raster.files <- list.files("data/cdh/", pattern = "\\.tif$", full.names = TRUE)

cov.den.hei <- stack(raster.files)

## extract values of tree cover, tree density and forest canopy height

values <- extract(cov.den.hei , lon.lat)

## bind data and get only the required data
bio.con.dist.cdh <- cbind(bio.con.dist, values)

## add conflict frequency data per province

```

```

## count conflict frequency per province
con.sub.2 <- con.thin.df
con.sub.2 <- con.sub.2 %>% filter(PROVINCE!="") #filter empty rows

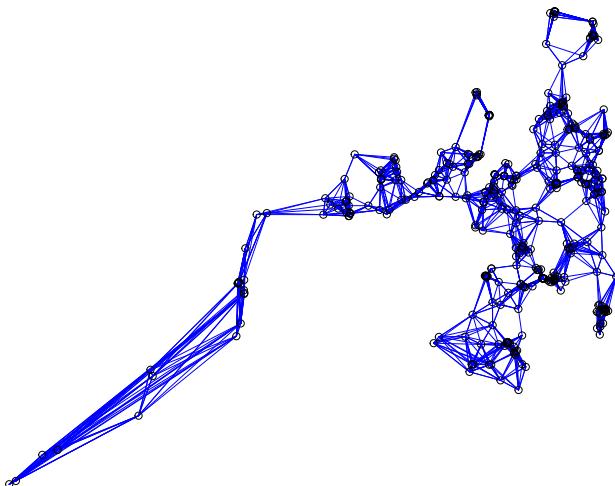
con.freq <- con.sub.2 %>%
  group_by(PROVINCE) %>%
  count()
con.freq

## # A tibble: 24 x 2
## # Groups:   PROVINCE [24]
##       PROVINCE      n
##       <chr>        <int>
## 1 Agusan del Norte    26
## 2 Agusan del Sur     42
## 3 Basilan            40
## 4 Bukidnon           45
## 5 Cotabato           77
## 6 Davao Oriental     27
## 7 Davao de Oro       46
## 8 Davao del Norte    19
## 9 Davao del Sur      43
## 10 Lanao del Norte   28
## # i 14 more rows

## enter the frequency value manually
bio.con.dist.cdh.cfreq <- bio.con.dist.cdh %>%
  mutate(conflictFreq = if_else(county == "Agusan del Norte", 24,
  if_else(county == "Agusan del Sur", 40,
  if_else(county == "Basilan", 31,
  if_else(county == "Bukidnon", 44,
  if_else(county == "Cagayan de Oro", 0,
  if_else(county == "Camiguin Island", 0,
  if_else(county == "Cotabato", 64,
  if_else(county == "Davao", 0,
  if_else(county == "Davao Oriental", 25,
  if_else(county == "Davao de Oro", 43,
  if_else(county == "Davao del Norte", 19,
  if_else(county == "Davao del sur", 38,
  if_else(county == "Dinagat Island", 0,
  if_else(county == "General Santos", 0,
  if_else(county == "Lanao del Norte", 27,
  if_else(county == "Lanao del Sur", 28,
  if_else(county == "Maguindanao", 50,
  if_else(county == "Misamis Occidental", 10,
  if_else(county == "Misamis Oriental", 16,
  if_else(county == "North Cotabato", 0,
  if_else(county == "Sarangani", 15,
  if_else(county == "South Cotabato", 20,
  if_else(county == "Sultan Kudarat", 29,
  if_else(county == "Sulu", 33,
  if_else(county == "Surigao del Norte", 15,
  if_else(county == "Surigao del Sur", 28,
  if_else(county == "Tawi-Tawi", 12,

```


Spatial Weight Matrix



```
## set seed and turn off scientific notation
set.seed(123)
options(scipen = 10)

## assign class and year columns as factors
smm.data.sf$class <- as.factor(smm.data.sf$class)
smm.data.sf$Year <- as.factor(smm.data.sf$Year)

## provide the model equation
eq.mod <- NumbSp.log ~ class + Year + distance.log + frequency.log + forestHeight.log +
  treeCov.log + treeDensity.log

## fit model
mod <- lm(eq.mod, data=smm.data.sf)

## test regression residuals to examine the residuals of the model using the spatial relationship matrix
## use Morn's correlation
## the null hypothesis states that there is no spatial correlation in the residuals
lm.morantest(mod, l.w)

##
## Global Moran I for regression residuals
##
## data:
## model: lm(formula = eq.mod, data = smm.data.sf)
## weights: l.w
##
## Moran I statistic standard deviate = 1.258, p-value = 0.1042
## alternative hypothesis: greater
## sample estimates:
## Observed Moran I      Expectation      Variance
## -0.0014328919    -0.0238590381    0.0003178115
## or use LaGrange Multiplier
lm.RStests(mod, l.w, test="all")
```

```

## 
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
## 
## data:
## model: lm(formula = eq.mod, data = smm.data.sf)
## test weights: l.w
## 
## RSerr = 0.0043289, df = 1, p-value = 0.9475
## 
## 
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
## 
## data:
## model: lm(formula = eq.mod, data = smm.data.sf)
## test weights: l.w
## 
## RSlag = 0.19928, df = 1, p-value = 0.6553
## 
## 
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
## 
## data:
## model: lm(formula = eq.mod, data = smm.data.sf)
## test weights: l.w
## 
## adjRSerr = 1.1077, df = 1, p-value = 0.2926
## 
## 
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
## 
## data:
## model: lm(formula = eq.mod, data = smm.data.sf)
## test weights: l.w
## 
## adjRSlag = 1.3027, df = 1, p-value = 0.2537
## 
## 
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
## 
## data:
## model: lm(formula = eq.mod, data = smm.data.sf)
## test weights: l.w
## 
## SARMA = 1.307, df = 2, p-value = 0.5202
## stop and use the model results above if there is no spatial correlation (p-value >0.05)
## otherwise, proceed with spatial lag model, spatial error models and etc

## find the best model if necessary

```

```

mod <- lm(eq.mod, data = smm.data.sf)

## fit models with fewer predictors
mod1 <- update(mod, . ~ . - class)
mod2 <- update(mod, . ~ . - Year)
mod3 <- update(mod, . ~ . - distance.log)
mod4 <- update(mod, . ~ . - frequency.log)
mod5 <- update(mod, . ~ . - forestHeight.log)
mod6 <- update(mod, . ~ . - treeCov.log)
mod7 <- update(mod, . ~ . - treeDensity.log)

## compare AIC values
aic.values <- AIC(mod, mod1, mod2, mod3, mod4, mod5, mod6, mod7)
aic.values

##      df     AIC
## mod  41 1066.206
## mod1 35 1127.966
## mod2 13 1055.632
## mod3 40 1066.131
## mod4 40 1068.172
## mod5 40 1068.025
## mod6 40 1064.411
## mod7 40 1064.743

## identify the model with the lowest AIC
best.model <- rownames(aic.values)[which.min(aic.values$AIC)]
best.model

## [1] "mod2"

## use model 2 results to check for spatial correlation
lm.morantest(mod2, l.w)

##
## Global Moran I for regression residuals
##
## data:
## model: lm(formula = NumbSp.log ~ class + distance.log + frequency.log +
##           + forestHeight.log + treeCov.log + treeDensity.log, data =
##           smm.data.sf)
## weights: l.w
##
## Moran I statistic standard deviate = 1.4616, p-value = 0.07193
## alternative hypothesis: greater
## sample estimates:
## Observed Moran I      Expectation      Variance
## 0.0220101742    -0.0079451347   0.0004200423

## or use LaGrange Multiplier
lm.RStests(mod2, l.w, test="all")

##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:

```

```

## model: lm(formula = NumbSp.log ~ class + distance.log + frequency.log
## + forestHeight.log + treeCov.log + treeDensity.log, data =
## smm.data.sf)
## test weights: l.w
##
## RSerr = 1.0214, df = 1, p-value = 0.3122
##
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = NumbSp.log ~ class + distance.log + frequency.log
## + forestHeight.log + treeCov.log + treeDensity.log, data =
## smm.data.sf)
## test weights: l.w
##
## RSlag = 1.2552, df = 1, p-value = 0.2626
##
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = NumbSp.log ~ class + distance.log + frequency.log
## + forestHeight.log + treeCov.log + treeDensity.log, data =
## smm.data.sf)
## test weights: l.w
##
## adjRSerr = 0.0017726, df = 1, p-value = 0.9664
##
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = NumbSp.log ~ class + distance.log + frequency.log
## + forestHeight.log + treeCov.log + treeDensity.log, data =
## smm.data.sf)
## test weights: l.w
##
## adjRSlag = 0.23551, df = 1, p-value = 0.6275
##
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = NumbSp.log ~ class + distance.log + frequency.log
## + forestHeight.log + treeCov.log + treeDensity.log, data =
## smm.data.sf)
## test weights: l.w
##
## SARMA = 1.2569, df = 2, p-value = 0.5334

```

```

## show model 2 summary
## this summary shows some significant results for several classes based on p-value
## however, t-values are too low; thus, these predictors do not contribute much to the model
summary(get(best.model))

##
## Call:
## lm(formula = NumbSp.log ~ class + distance.log + frequency.log +
##     forestHeight.log + treeCov.log + treeDensity.log, data = smm.data.sf)
##
## Residuals:
##    Min      1Q  Median      3Q      Max
## -2.3034 -0.6753 -0.0280  0.5313  3.2076
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.15796   0.47090  4.583 0.00000636 ***
## classAmphibia 0.19500   0.22912  0.851  0.39528
## classArachnida -0.49226   0.21773 -2.261  0.02437 *
## classAves 0.63404   0.21804  2.908  0.00387 **
## classInsecta 0.42120   0.20485  2.056  0.04050 *
## classMammalia -0.61133   0.22116 -2.764  0.00600 **
## classReptilia -0.67736   0.32069 -2.112  0.03537 *
## distance.log 0.006665  0.05039  0.132  0.89509
## frequency.log -0.10354  0.04460 -2.321  0.02082 *
## forestHeight.log 0.09520   0.06179  1.541  0.12427
## treeCov.log -0.00469  0.03700 -0.127  0.89919
## treeDensity.log 0.01365  0.01681  0.812  0.41721
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9928 on 357 degrees of freedom
## Multiple R-squared: 0.2047, Adjusted R-squared: 0.1802
## F-statistic: 8.352 on 11 and 357 DF, p-value: 0.00000000000043

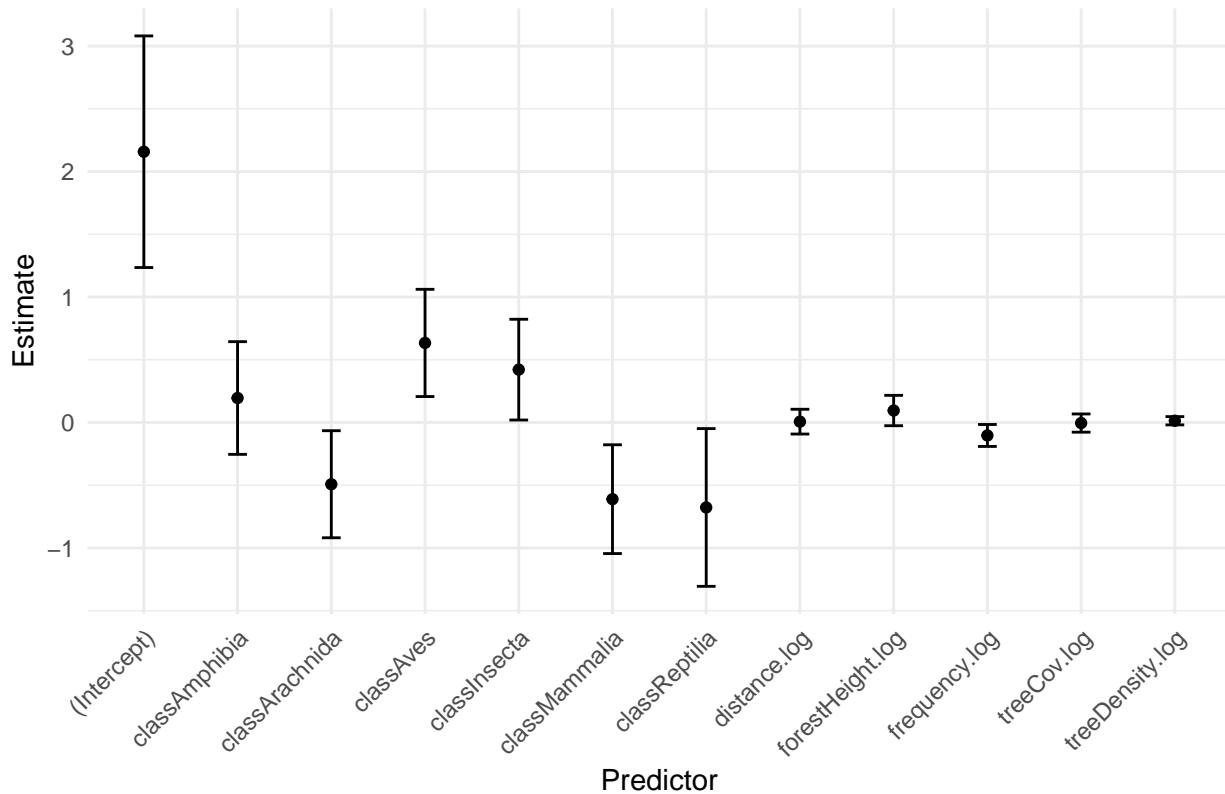
## plot results
## get fixed effects coefficients and their confidence intervals
fix.eff <- summary(get(best.model))$coefficients

# create a data frame
fix.eff_df <- as.data.frame(fix.eff)
fix.eff_df$term <- rownames(fix.eff_df)

## plot coefficients
ggplot(fix.eff_df, aes(x = term, y = Estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Estimate - 1.96 * `Std. Error`,
                    ymax = Estimate + 1.96 * `Std. Error`), width = 0.2) +
  theme_minimal() +
  labs(title = "Fixed Effects Estimates", x = "Predictor", y = "Estimate") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

Fixed Effects Estimates



```

## uncomment to reorder plot from lowest to highest coefficients
##fix.eff_df$term <- factor(fix.eff_df$term,
##                             levels = fix.eff_df$term[order(fix.eff_df$Estimate)])
```

```

##ggplot(fix.eff_df, aes(x = term, y = Estimate)) +
##  geom_point() +
##  geom_errorbar(aes(ymin = Estimate - 1.96 * `Std. Error`,
##                    ymax = Estimate + 1.96 * `Std. Error`), width = 0.2) +
##  theme_minimal() +
##  labs(title = "Fixed Effects Estimates", x = "Predictor", y = "Estimate") +
##  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```

## perform spatial mixed models with land cover data
## land cover data of 1988 was from Swedish Space Corporation (SSC 1988)
## retrieved from
## https://data.humdata.org/dataset/29a3760f-3170-4555-b5d7-1fdbd6cfb5a69?force_layout=desktop
## land cover of 2020 was from the Philippine National Mapping and Resource Information Authority
## data was retrieved from geoportal PH (https://www.geoportal.gov.ph/)
## the national land cover data is ground-truthed
## this is only for simplistic illustration of sampling artifact
## the original land cover format is shapefile
## we converted it to raster file for fast data processing
## data conversion was done in QGIS using the plugin "rasterize"
## raster pixel size is 0.0008 or approximately 100 square meters
## all raster files have uniform coordinate reference system, resolution and extent
```

```

## read the file with calculated distance from QGIS
```

```

bio.con.dist <- read.csv("data/bio.con.dist.csv", header = TRUE, sep = ",")  

lon.lat <- subset(bio.con.dist, select = c("decimalLongitude", "decimalLatitude"))  

## extract land cover value  

values <- extract(lc.data, lon.lat)  

## bind data and get only the required data  

bio.con.dist.lc <- cbind(bio.con.dist, values)  

colnames(bio.con.dist.lc)[24] <- "LC_1988"  

colnames(bio.con.dist.lc)[25] <- "LC_2020"  

bio.con.dist.lc$LC_1988 <- as.integer(bio.con.dist.lc$LC_1988)  

bio.con.dist.lc$LC_2020 <- as.integer(bio.con.dist.lc$LC_2020)  

## add land cover descriptions  

bio.con.dist.lc.mod <- bio.con.dist.lc %>%
  mutate(class_1988 = if_else(LC_1988 == 1, "Closed Canopy",
    if_else(LC_1988 == 2, "Cultivated Area mixed with brushland/grassland",
    if_else(LC_1988 == 3, "Crop land mixed with coconut plantation",
    if_else(LC_1988 == 4, "Unclassified",
    if_else(LC_1988 == 5, "Open Canopy",
    if_else(LC_1988 == 6, "Arable land, crops mainly cereals and sugar",
    if_else(LC_1988 == 7, "Fishponds derived from mangrove",
    if_else(LC_1988 == 8, "Coral Reef",
    if_else(LC_1988 == 9, "Grassland, grass covering > 70 percent",
    if_else(LC_1988 == 10, "Lake",
    if_else(LC_1988 == 11, "Mangrove vegetation",
    if_else(LC_1988 == 12, "Built-up Area",
    if_else(LC_1988 == 13, "Quarry",
    if_else(LC_1988 == 14, "Coconut plantations",
    if_else(LC_1988 == 15, "Siltation pattern in lake",
    if_else(LC_1988 == 16, "Marshy area and swamp",
    if_else(LC_1988 == 17, "Riverbeds",
    if_else(LC_1988 == 18, "Crop land mixed with other plantation",
    if_else(LC_1988 == 19, "Other plantations", "No data")))))))))))))  

bio.con.dist.lc.mod <- bio.con.dist.lc.mod %>%
  mutate(class_2020 = if_else(LC_2020 == 1, "Grassland",
  if_else(LC_2020 == 2, "Annual Crop",
  if_else(LC_2020 == 3, "Marshland/Swamp",
  if_else(LC_2020 == 4, "Perennial Crop",
  if_else(LC_2020 == 5, "Built-up",
  if_else(LC_2020 == 6, "Fishpond",
  if_else(LC_2020 == 7, "Inland Water",
  if_else(LC_2020 == 8, "Mangrove Forest",
  if_else(LC_2020 == 9, "Brush/Shrubs",
  if_else(LC_2020 == 10, "Closed Forest",
  if_else(LC_2020 == 11, "Open Forest",
  if_else(LC_2020 == 12, "Open/Barren", "No data")))))))))  

## add conflict frequency data per province  

## count conflict frequency per province

```

```

con.sub.2 <- con.thin.df
con.sub.2 <- con.sub.2 %>% filter(PROVINCE!="") #filter empty rows

con.freq <- con.sub.2 %>%
  group_by(PROVINCE) %>%
  count()
con.freq

## # A tibble: 24 x 2
## # Groups:   PROVINCE [24]
##       PROVINCE     n
##       <chr>      <int>
## 1 Agusan del Norte    26
## 2 Agusan del Sur     42
## 3 Basilan            40
## 4 Bukidnon           45
## 5 Cotabato           77
## 6 Davao Oriental     27
## 7 Davao de Oro        46
## 8 Davao del Norte     19
## 9 Davao del Sur       43
## 10 Lanao del Norte    28
## # i 14 more rows

## enter the frequency value manually
bio.con.dist.lc.cfreq <- bio.con.dist.lc.mod %>%
  mutate(conflictFreq = if_else(county == "Agusan del Norte", 24,
    if_else(county == "Agusan del Sur", 40,
    if_else(county == "Basilan", 31,
    if_else(county == "Bukidnon", 44,
    if_else(county == "Cagayan de Oro", 0,
    if_else(county == "Camiguin Island", 0,
    if_else(county == "Cotabato", 64,
    if_else(county == "Davao", 0,
    if_else(county == "Davao Oriental", 25,
    if_else(county == "Davao de Oro", 43,
    if_else(county == "Davao del Norte", 19,
    if_else(county == "Davao del sur", 38,
    if_else(county == "Dinagat Island", 0,
    if_else(county == "General Santos", 0,
    if_else(county == "Lanao del Norte", 27,
    if_else(county == "Lanao del Sur", 28,
    if_else(county == "Maguindanao", 50,
    if_else(county == "Misamis Occidental", 10,
    if_else(county == "Misamis Oriental", 16,
    if_else(county == "North Cotabato", 0,
    if_else(county == "Sarangani", 15,
    if_else(county == "South Cotabato", 20,
    if_else(county == "Sultan Kudarat", 29,
    if_else(county == "Sulu", 33,
    if_else(county == "Surigao del Norte", 15,
    if_else(county == "Surigao del Sur", 28,
    if_else(county == "Tawi-Tawi", 12,
    if_else(county == "Zamboanga City", 0,

```



```

## fit model for 1988 land cover
mod <- lm(eq.88, data=smm.data.sf)

## test regression residuals to examine the residuals of the model
## use the spatial relationship matrix
lm.morantest(mod, l.w)

##
## Global Moran I for regression residuals
##
## data:
## model: lm(formula = eq.88, data = smm.data.sf)
## weights: l.w
##
## Moran I statistic standard deviate = 2.1504, p-value = 0.01576
## alternative hypothesis: greater
## sample estimates:
## Observed Moran I      Expectation      Variance
##        0.0194843728    -0.0376268922    0.0007053682

## or use LaGrange Multiplier
lm.RStests(mod, l.w, test="all")

##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = eq.88, data = smm.data.sf)
## test weights: l.w
##
## RSerr = 0.40876, df = 1, p-value = 0.5226
##
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = eq.88, data = smm.data.sf)
## test weights: l.w
##
## RSlag = 0.73005, df = 1, p-value = 0.3929
##
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = eq.88, data = smm.data.sf)
## test weights: l.w
##
## adjRSerr = 0.055325, df = 1, p-value = 0.814
##
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial

```

```

## dependence
##
## data:
## model: lm(formula = eq.88, data = smm.data.sf)
## test weights: l.w
##
## adjRSlag = 0.37661, df = 1, p-value = 0.5394
##
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = eq.88, data = smm.data.sf)
## test weights: l.w
##
## SARMA = 0.78537, df = 2, p-value = 0.6752
## find the best model if necessary
mod <- lm(eq.88, data = smm.data.sf)

## fit models with fewer predictors
mod1 <- update(mod, . ~ . - class_1988)
mod2 <- update(mod, . ~ . - class)
mod3 <- update(mod, . ~ . - Year)
mod4 <- update(mod, . ~ . - distance.log)
mod5 <- update(mod, . ~ . - frequency.log)

## compare AIC values
aic.values <- AIC(mod, mod1, mod2, mod3, mod4, mod5)

## Warning in AIC.default(mod, mod1, mod2, mod3, mod4, mod5): models are not all
## fitted to the same number of observations
aic.values

##      df      AIC
## mod  49 1081.309
## mod1 38 1108.407
## mod2 43 1150.043
## mod3 21 1071.468
## mod4 48 1080.915
## mod5 48 1084.627
## identify the model with the lowest AIC
best.model <- rownames(aic.values)[which.min(aic.values$AIC)]
best.model

## [1] "mod3"
## use model 3
summary(get(best.model))

##
## Call:
## lm(formula = NumbSp.log ~ class_1988 + class + distance.log +
##     frequency.log, data = smm.data.sf)
##

```

```

## Residuals:
##      Min     1Q   Median     3Q    Max
## -2.54307 -0.62998 -0.07342  0.57378  3.03949
##
## Coefficients:
##                               Estimate Std. Error
## (Intercept)                2.5959501  0.4781908
## class_1988Arable land, crops mainly cereals and sugar -0.2219161  0.2144229
## class_1988Built-up Area          -0.7267695  0.3367929
## class_1988Coconut plantations          -0.6408453  0.3271912
## class_1988Crop land mixed with coconut plantation          -0.1724935  0.1513684
## class_1988Crop land mixed with other plantation          -0.1900014  0.3434825
## class_1988Fishponds derived from mangrove          -0.5549514  0.7098303
## class_1988Grassland, grass covering > 70 percent          0.1046473  0.2776135
## class_1988Lake          -0.8677348  0.7073919
## class_1988No data          0.3193736  0.1713498
## class_1988Siltation pattern in lake          0.0885759  0.7113779
## class_1988Unclassified          -0.1471678  0.1579725
## classAmphibia          0.2568196  0.2239176
## classArachnida          -0.4011883  0.2124153
## classAves          0.6585197  0.2137386
## classInsecta          0.5467317  0.2003961
## classMammalia          -0.5697266  0.2143977
## classReptilia          -0.7447523  0.3091205
## distance.log          -0.0009307  0.0498792
## frequency.log          -0.1159963  0.0442316
##
## t value  Pr(>|t|)
## (Intercept) 5.429 0.000000106 ***
## class_1988Arable land, crops mainly cereals and sugar -1.035 0.30140
## class_1988Built-up Area          -2.158 0.03161 *
## class_1988Coconut plantations          -1.959 0.05094 .
## class_1988Crop land mixed with coconut plantation          -1.140 0.25524
## class_1988Crop land mixed with other plantation          -0.553 0.58050
## class_1988Fishponds derived from mangrove          -0.782 0.43485
## class_1988Grassland, grass covering > 70 percent          0.377 0.70643
## class_1988Lake          -1.227 0.22076
## class_1988No data          1.864 0.06317 .
## class_1988Siltation pattern in lake          0.125 0.90098
## class_1988Unclassified          -0.932 0.35218
## classAmphibia          1.147 0.25218
## classArachnida          -1.889 0.05975 .
## classAves          3.081 0.00222 **
## classInsecta          2.728 0.00668 **
## classMammalia          -2.657 0.00823 **
## classReptilia          -2.409 0.01649 *
## distance.log          -0.019 0.98512
## frequency.log          -2.622 0.00911 **
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9849 on 354 degrees of freedom
##   (8 observations deleted due to missingness)
## Multiple R-squared:  0.2399, Adjusted R-squared:  0.1991
## F-statistic: 5.88 on 19 and 354 DF, p-value: 0.0000000000006982

```

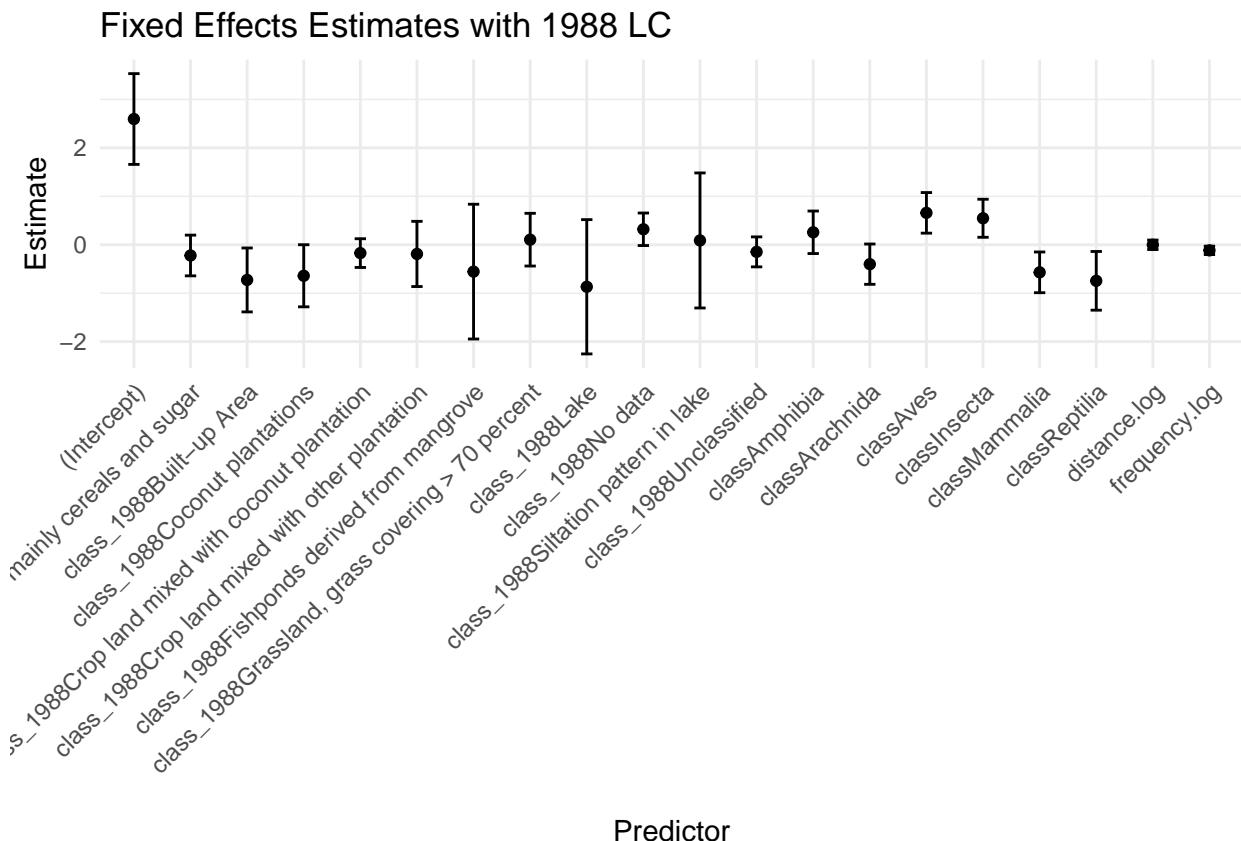
```

## plot results
## get fixed effects coefficients and their confidence intervals
fix.eff <- summary(best.model)$coefficients

# create a data frame
fix.eff_df <- as.data.frame(fix.eff)
fix.eff_df$term <- rownames(fix.eff_df)

## plot results
ggplot(fix.eff_df, aes(x = term, y = Estimate)) +
  geom_point() +
  geom_errorbar(aes(ymax = Estimate + 1.96 * `Std. Error`,
                    ymin = Estimate - 1.96 * `Std. Error`), width = 0.2) +
  theme_minimal() +
  labs(title = "Fixed Effects Estimates with 1988 LC", x = "Predictor", y = "Estimate") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```



```

## equation with 2020 land cover
eq.20 <- NumbSp.log ~ class_2020 + class + Year + distance.log + frequency.log

## create a spatial weights matrix using nearest neighbors
nb.w <- knn2nb(knearneigh(st_coordinates(smm.data.sf), k = 10))
l.w <- nb2listw(nb.w, style = "W")

#plot(nb.w, st_coordinates(smm.data.sf), lwd=.2, col="blue", cex = .5)
#title(main = "Spatial Weight Matrix")

```

```

## fit model for 2020 land cover
mod <- lm(eq.20, data=smm.data.sf)

## test regression residuals to examine the residuals of the model
## use the spatial relationship matrix
lm.morantest(mod, l.w)

##
## Global Moran I for regression residuals
##
## data:
## model: lm(formula = eq.20, data = smm.data.sf)
## weights: l.w
##
## Moran I statistic standard deviate = 0.76535, p-value = 0.222
## alternative hypothesis: greater
## sample estimates:
## Observed Moran I      Expectation      Variance
## -0.0107719927    -0.0242294255    0.0003091767

## or use LaGrange Multiplier
lm.RStests(mod, l.w, test="all")

##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = eq.20, data = smm.data.sf)
## test weights: l.w
##
## RSerr = 0.24959, df = 1, p-value = 0.6174
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = eq.20, data = smm.data.sf)
## test weights: l.w
##
## RSlag = 0.0019833, df = 1, p-value = 0.9645
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = eq.20, data = smm.data.sf)
## test weights: l.w
##
## adjRSerr = 1.3423, df = 1, p-value = 0.2466
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial

```

```

## dependence
##
## data:
## model: lm(formula = eq.20, data = smm.data.sf)
## test weights: l.w
##
## adjRSlag = 1.0947, df = 1, p-value = 0.2954
##
##
## Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
## dependence
##
## data:
## model: lm(formula = eq.20, data = smm.data.sf)
## test weights: l.w
##
## SARMA = 1.3443, df = 2, p-value = 0.5106
## find the best model if necessary
mod <- lm(eq.20, data = smm.data.sf)

## fit models with fewer predictors
mod1 <- update(mod, . ~ . - class_2020)
mod2 <- update(mod, . ~ . - class)
mod3 <- update(mod, . ~ . - Year)
mod4 <- update(mod, . ~ . - distance.log)
mod5 <- update(mod, . ~ . - frequency.log)

## compare AIC values
aic.values <- AIC(mod, mod1, mod2, mod3, mod4, mod5)

## Warning in AIC.default(mod, mod1, mod2, mod3, mod4, mod5): models are not all
## fitted to the same number of observations
aic.values

##      df      AIC
## mod  46 1087.627
## mod1 38 1108.407
## mod2 40 1157.120
## mod3 18 1071.257
## mod4 45 1087.172
## mod5 45 1090.358

## identify the model with the lowest AIC
best.model <- rownames(aic.values)[which.min(aic.values$AIC)]
best.model

## [1] "mod3"
## use model 3
summary(get(best.model))

##
## Call:
## lm(formula = NumbSp.log ~ class_2020 + class + distance.log +
##     frequency.log, data = smm.data.sf)
##

```

```

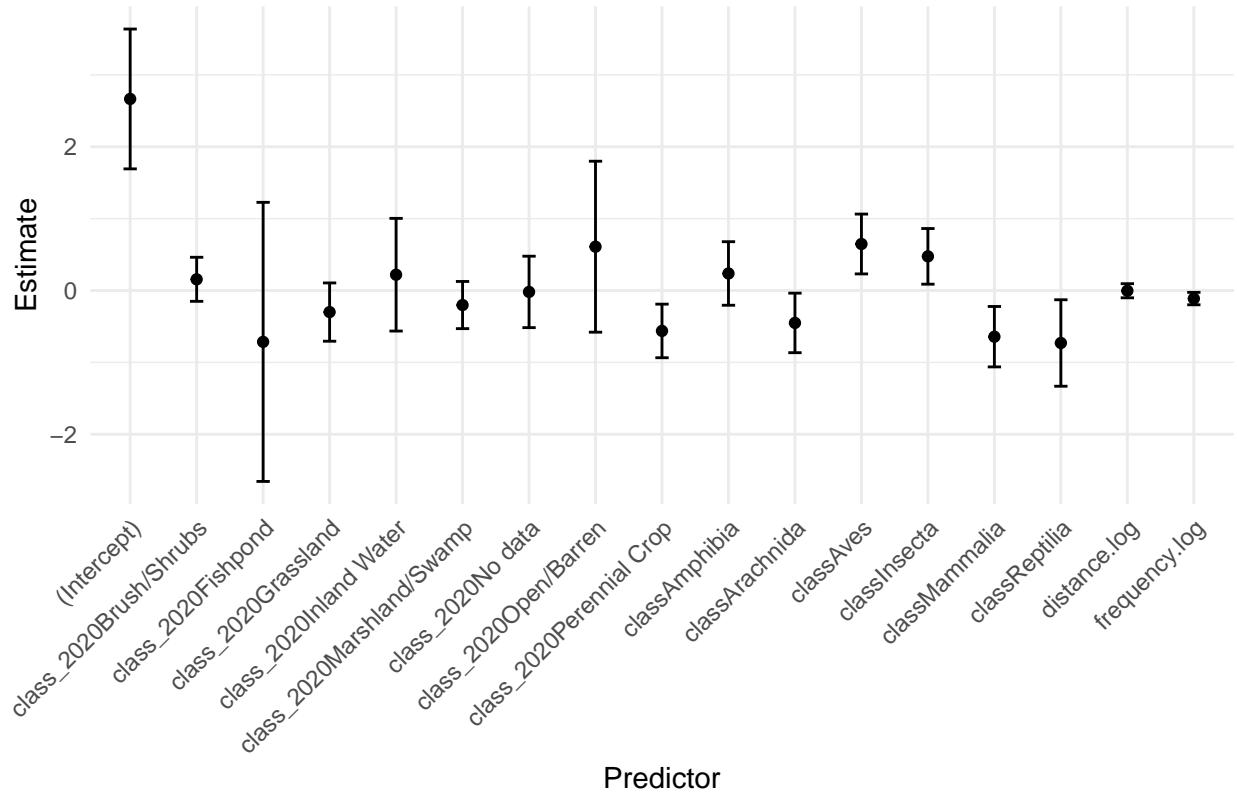
## Residuals:
##      Min       1Q   Median      3Q      Max
## -2.42376 -0.60253 -0.02423  0.57666  3.06664
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.664949  0.496338   5.369 0.000000142 ***
## class_2020Brush/Shrubs    0.155412  0.156370   0.994  0.32095
## class_2020Fishpond       -0.715221  0.990639  -0.722  0.47077
## class_2020Grassland      -0.300421  0.206912  -1.452  0.14739
## class_2020Inland Water   0.219356  0.400028   0.548  0.58379
## class_2020Marshland/Swamp -0.202816  0.167434  -1.211  0.22657
## class_2020No data        -0.019859  0.253476  -0.078  0.93759
## class_2020Open/Barren     0.609297  0.606805   1.004  0.31600
## class_2020Perennial Crop -0.562984  0.190210  -2.960  0.00328 **
## classAmphibia            0.236626  0.225617   1.049  0.29497
## classArachnida          -0.451327  0.211343  -2.136  0.03339 *
## classAves                 0.646714  0.212369   3.045  0.00250 **
## classInsecta              0.474940  0.197506   2.405  0.01669 *
## classMammalia             0.642938  0.214440  -2.998  0.00290 **
## classReptilia             0.730594  0.307020  -2.380  0.01785 *
## distance.log               -0.003723  0.050113  -0.074  0.94082
## frequency.log             -0.112711  0.044117  -2.555  0.01103 *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9738 on 361 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.2451, Adjusted R-squared:  0.2117
## F-statistic: 7.327 on 16 and 361 DF,  p-value: 7.859e-15
##
## plot results
## get fixed effects coefficients and their confidence intervals
fix.eff <- summary(best.model)$coefficients

# create a data frame
fix.eff_df <- as.data.frame(fix.eff)
fix.eff_df$term <- rownames(fix.eff_df)

## plot results
ggplot(fix.eff_df, aes(x = term, y = Estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Estimate - 1.96 * `Std. Error`,
                     ymax = Estimate + 1.96 * `Std. Error`), width = 0.2) +
  theme_minimal() +
  labs(title = "Fixed Effects Estimates with 2020 LC", x = "Predictor", y = "Estimate") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

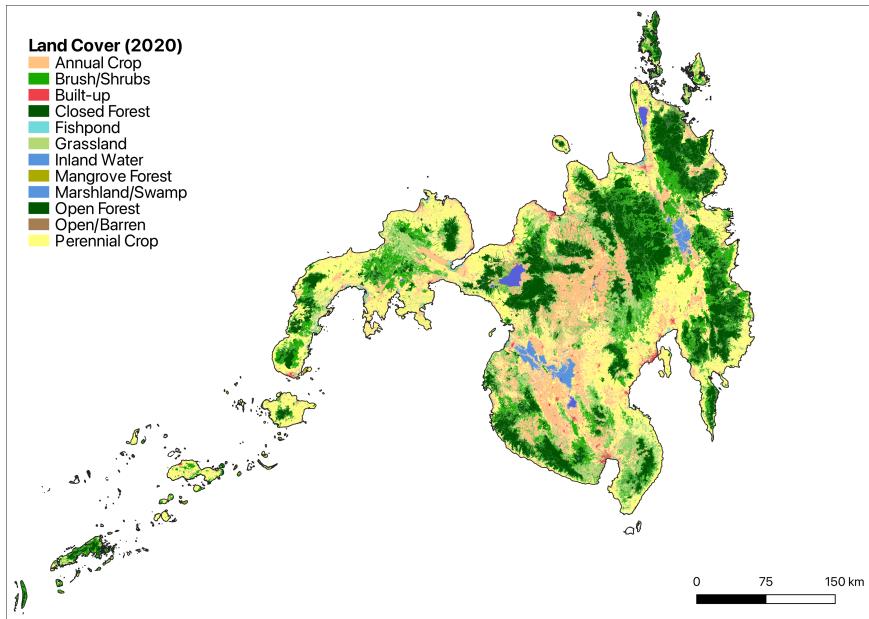
Fixed Effects Estimates with 2020 LC



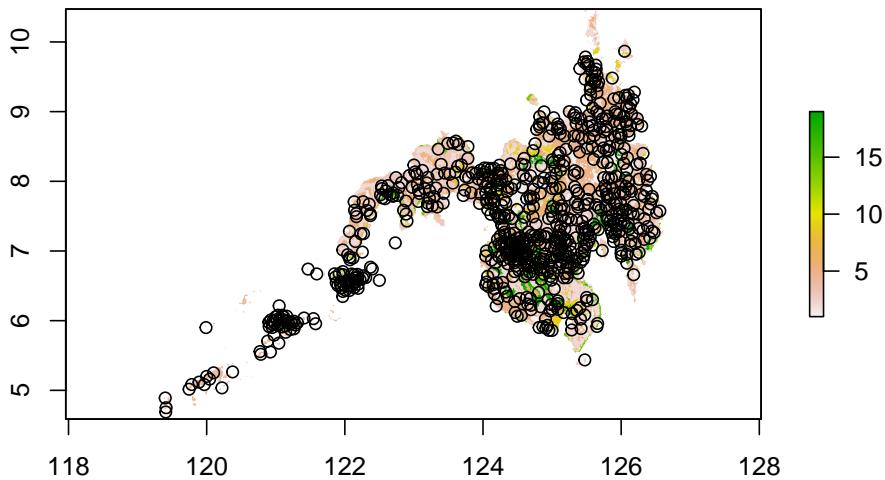
Calculate proportion of species occurrence records and conflict sites positioned in different land cover types

```
## here we count the number of conflict sites found in different land cover types
## this demonstrates the artifact of sampling, which was not considered in the paper
```

```
## here is the 2020 land cover of Mindanao
knitr:::include_graphics("images/lc_2020.png")
```



```
## plot data
plot(lc.data$X1988_lc_rast1km)
points(con.thin.df$longitude, con.thin.df$latitude)
```



```
con.coor <- cbind(con.thin.df$longitude, con.thin.df$latitude)

## extract data from land cover using conflict data coordinates
con.lc.data <- extract(lc.data, con.coor)
colnames(con.lc.data)[1] <- "LC_1988"
colnames(con.lc.data)[2] <- "LC_2020"

## remove NAs
con.lc.data <- as.data.frame(na.omit(con.lc.data))
con.lc.data$LC_1988 <- as.integer(con.lc.data$LC_1988)
con.lc.data$LC_2020 <- as.integer(con.lc.data$LC_2020)

head(con.lc.data)

## LC_1988 LC_2020
```

```

## 1      18      3
## 2      1      0
## 3      1      3
## 4      3      3
## 5      6      1
## 6      6      3

## read land cover types code
lc.88.d <- read.csv("data/1988_lc_classmap.csv")
lc.88.d

##                                     category code
## 1                      Closed Canopy    1
## 2 Cultivated Area mixed with brushland/grassland    2
## 3          Crop land mixed with coconut plantation    3
## 4                      Unclassified    4
## 5                      Open Canopy    5
## 6 Arable land, crops mainly cereals and sugar    6
## 7 Fishponds derived from mangrove    7
## 8          Coral Reef    8
## 9 Grassland, grass covering > 70 percent    9
## 10                     Lake    10
## 11        Mangrove vegetation    11
## 12        Built-up Area    12
## 13           Quarry    13
## 14        Coconut plantations    14
## 15 Siltation pattern in lake    15
## 16        Marshy area and swamp    16
## 17        Riverbeds    17
## 18        Crop land mixed with other plantation    18
## 19        Other plantations    19

lc.20.d <- read.csv("data/2020_lc_classmap.csv")
lc.20.d

##                                     category code
## 1          Grassland    1
## 2     Annual Crop    2
## 3 Marshland/Swamp    3
## 4 Perennial Crop    4
## 5        Built-up    5
## 6       Fishpond    6
## 7 Inland Water    7
## 8   Mangrove Forest    8
## 9     Brush/Shrubs    9
## 10    Closed Forest   10
## 11    Open Forest   11
## 12 Open/Barren    12

## add land cover descriptions
con.lc.data.m1 <- con.lc.data %>%
  mutate(class_1988 = if_else(LC_1988 == 1, "Closed Canopy",
  if_else(LC_1988 == 2, "Cultivated Area mixed with brushland/grassland",
  if_else(LC_1988 == 3, "Crop land mixed with coconut plantation",
  if_else(LC_1988 == 4, "Unclassified",
  if_else(LC_1988 == 5, "Open Canopy",

```

```

if_else(LC_1988 == 6, "Arable land, crops mainly cereals and sugar",
if_else(LC_1988 == 7, "Fishponds derived from mangrove",
if_else(LC_1988 == 8, "Coral Reef",
if_else(LC_1988 == 9, "Grassland, grass covering > 70 percent",
if_else(LC_1988 == 10, "Lake",
if_else(LC_1988 == 11, "Mangrove vegetation",
if_else(LC_1988 == 12, "Built-up Area",
if_else(LC_1988 == 13, "Quarry",
if_else(LC_1988 == 14, "Coconut plantations",
if_else(LC_1988 == 15, "Siltation pattern in lake",
if_else(LC_1988 == 16, "Marshy area and swamp",
if_else(LC_1988 == 17, "Riverbeds",
if_else(LC_1988 == 18, "Crop land mixed with other plantation",
if_else(LC_1988 == 19, "Other plantations", "No data"))))))))))))))))))))

con.lc.data.m2 <- con.lc.data %>%
  mutate(class_2020 = if_else(LC_2020 == 1, "Grassland",
  if_else(LC_2020 == 2, "Annual Crop",
  if_else(LC_2020 == 3, "Marshland/Swamp",
  if_else(LC_2020 == 4, "Perennial Crop",
  if_else(LC_2020 == 5, "Built-up",
  if_else(LC_2020 == 6, "Fishpond",
  if_else(LC_2020 == 7, "Inland Water",
  if_else(LC_2020 == 8, "Mangrove Forest",
  if_else(LC_2020 == 9, "Brush/Shrubs",
  if_else(LC_2020 == 10, "Closed Forest",
  if_else(LC_2020 == 11, "Open Forest",
  if_else(LC_2020 == 12, "Open/Barren", "No data)))))))))))

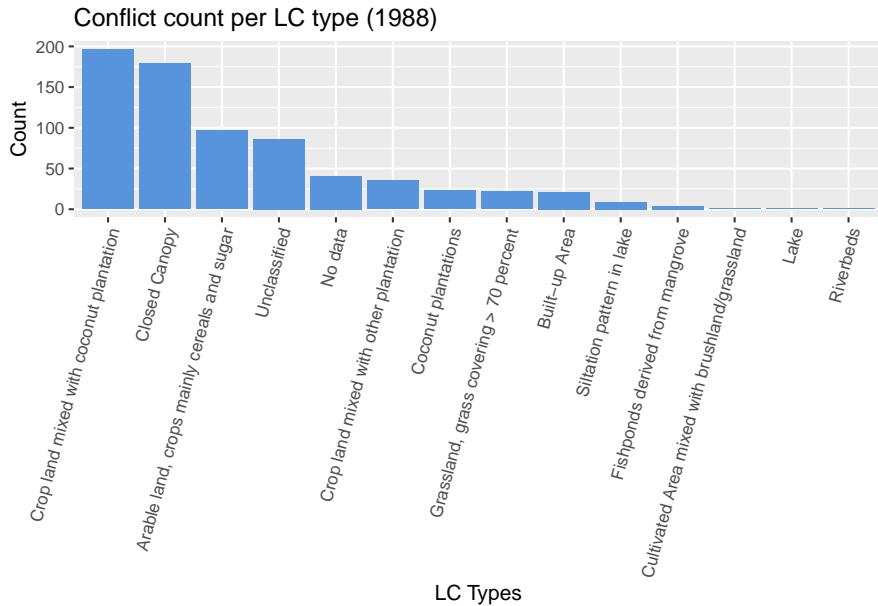
## count the number of conflict sites in each land cover type
## 1988 data
con.lc.count.88 <- con.lc.data.m2 %>%
  group_by(class_1988) %>%
  summarise(n())
colnames(con.lc.count.88)[2] <- "count"

con.lc.count.88

## # A tibble: 14 x 2
##   class_1988               count
##   <chr>                  <int>
## 1 Arable land, crops mainly cereals and sugar      97
## 2 Built-up Area                           21
## 3 Closed Canopy                          179
## 4 Coconut plantations                   23
## 5 Crop land mixed with coconut plantation    196
## 6 Crop land mixed with other plantation     35
## 7 Cultivated Area mixed with brushland/grassland    1
## 8 Fishponds derived from mangrove            4
## 9 Grassland, grass covering > 70 percent       22
## 10 Lake                                1
## 11 No data                            41
## 12 Riverbeds                           1
## 13 Siltation pattern in lake                9

```

```
## 14 Unclassified 86
ggplot(data=con.lc.count.88, aes(x=reorder(class_1988, -count), y=count)) +
  geom_bar(stat="identity", fill="#5794db") +
  labs(title = "Conflict count per LC type (1988)", x = "LC Types", y = "Count") +
  guides(x = guide_axis(angle = 75))
```



```
## write to a csv
con.lc.count.88 <- as.data.frame(con.lc.count.88)
write.csv(con.lc.count.88, "con.lc.count.88.csv")
```

```
#2020 data
con.lc.count.20 <- con.lc.data.m2 %>%
  group_by(class_2020) %>%
  summarise(n())
colnames(con.lc.count.20)[2] <- "count"
```

```
con.lc.count.20
```

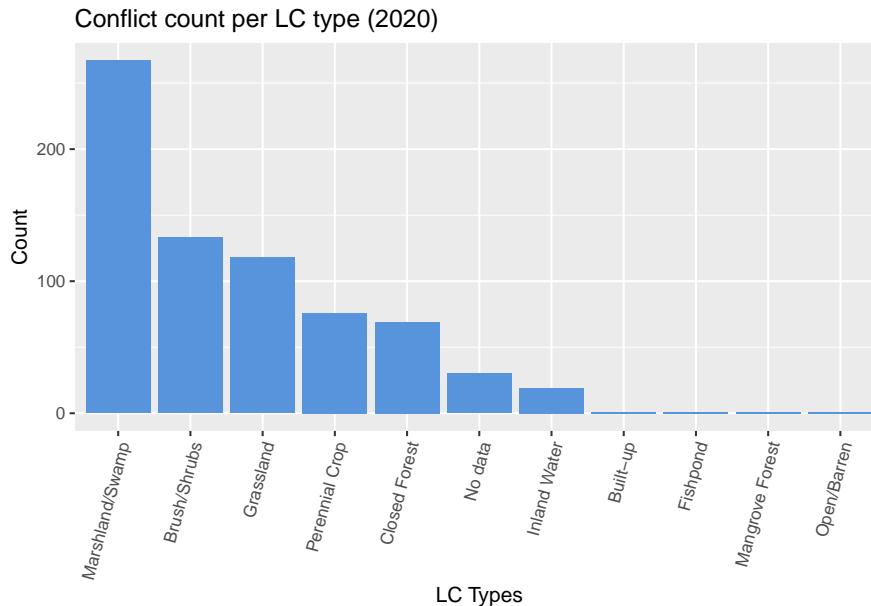
```
## # A tibble: 11 x 2
##   class_2020      count
##   <chr>        <int>
## 1 Brush/Shrubs     133
## 2 Built-up          1
## 3 Closed Forest     69
## 4 Fishpond          1
## 5 Grassland         118
## 6 Inland Water      19
## 7 Mangrove Forest    1
## 8 Marshland/Swamp    267
## 9 No data            30
## 10 Open/Barren       1
## 11 Perennial Crop    76
```

```
ggplot(data=con.lc.count.20, aes(x=reorder(class_2020, -count), y=count)) +
  geom_bar(stat="identity", fill="#5794db") +
```

```

  labs(title = "Conflict count per LC type (2020)", x = "LC Types", y = "Count") +
  guides(x = guide_axis(angle = 75))

```



```

## write to a csv
con.lc.count.20 <- as.data.frame(con.lc.count.20)
write.csv(con.lc.count.20, "con.lc.count.20.csv")

## extract data from land cover using biodiversity data coordinates
bio.coor <- cbind(bio.thin.df$decimalLongitude, bio.thin.df$decimalLatitude)

bio.lc.data <- extract(lc.data, bio.coor)

colnames(bio.lc.data )[1] <- "LC_1988"
colnames(bio.lc.data )[2] <- "LC_2020"

bio.lc.data <- as.data.frame(na.omit(bio.lc.data))

bio.lc.data$LC_1988 <- as.integer(bio.lc.data$LC_1988)
bio.lc.data$LC_2020 <- as.integer(bio.lc.data$LC_2020)

head(bio.lc.data)

##    LC_1988 LC_2020
## 1       1       0
## 2       6       1
## 3       6       3
## 4       1       9
## 5      18       1
## 6       9       7

## add land cover descriptions
bio.lc.data.m1 <- bio.lc.data %>%
  mutate(class_1988 = if_else(LC_1988 == 1, "Closed Canopy",
  if_else(LC_1988 == 2,"Cultivated Area mixed with brushland/grassland",
  if_else(LC_1988 == 3, "Crop land mixed with coconut plantation",

```

```

if_else(LC_1988 == 4, "Unclassified",
if_else(LC_1988 == 5, "Open Canopy",
if_else(LC_1988 == 6, "Arable land, crops mainly cereals and sugar",
if_else(LC_1988 == 7, "Fishponds derived from mangrove",
if_else(LC_1988 == 8, "Coral Reef",
if_else(LC_1988 == 9, "Grassland, grass covering > 70 percent",
if_else(LC_1988 == 10, "Lake",
if_else(LC_1988 == 11, "Mangrove vegetation",
if_else(LC_1988 == 12, "Built-up Area",
if_else(LC_1988 == 13, "Quarry",
if_else(LC_1988 == 14, "Coconut plantations",
if_else(LC_1988 == 15, "Siltation pattern in lake",
if_else(LC_1988 == 16, "Marshy area and swamp",
if_else(LC_1988 == 17, "Riverbeds",
if_else(LC_1988 == 18, "Crop land mixed with other plantation",
if_else(LC_1988 == 19, "Other plantations", "No data"))))))))))))))))))))

bio.lc.data.m2<- bio.lc.data %>%
  mutate(bio.lc.data.m1, class_2020 = if_else(LC_2020 == 1, "Grassland",
  if_else(LC_2020 == 2, "Annual Crop",
  if_else(LC_2020 == 3, "Marshland/Swamp",
  if_else(LC_2020 == 4, "Perennial Crop",
  if_else(LC_2020 == 5, "Built-up",
  if_else(LC_2020 == 6, "Fishpond",
  if_else(LC_2020 == 7, "Inland Water",
  if_else(LC_2020 == 8, "Mangrove Forest",
  if_else(LC_2020 == 9, "Brush/Shrubs",
  if_else(LC_2020 == 10, "Closed Forest",
  if_else(LC_2020 == 11, "Open Forest",
  if_else(LC_2020 == 12, "Open/Barren", "No data")))))))))))))

## count the number of biodiversity data points in each land cover type
#1988 data
bio.lc.count.88 <- bio.lc.data.m2 %>%
  group_by(class_1988) %>%
  summarise(n())
colnames(bio.lc.count.88)[2] <- "count"

bio.lc.count.88

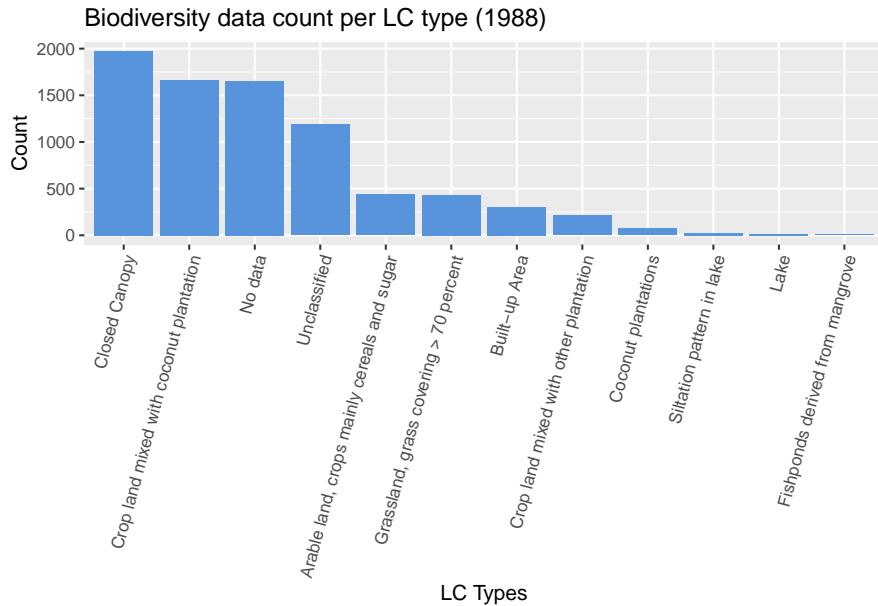
## # A tibble: 12 x 2
##   class_1988                      count
##   <chr>                           <int>
## 1 Arable land, crops mainly cereals and sugar    436
## 2 Built-up Area                      304
## 3 Closed Canopy                     1976
## 4 Coconut plantations                  74
## 5 Crop land mixed with coconut plantation    1667
## 6 Crop land mixed with other plantation      214
## 7 Fishponds derived from mangrove            10
## 8 Grassland, grass covering > 70 percent     434
## 9 Lake                                11
## 10 No data                            1654
## 11 Siltation pattern in lake                 28

```

```

## 12 Unclassified 1186
ggplot(data=bio.lc.count.88, aes(x=reorder(class_1988, -count), y=count)) +
  geom_bar(stat="identity", fill="#5794db") +
  labs(title = "Biodiversity data count per LC type (1988)", x = "LC Types", y = "Count") +
  guides(x = guide_axis(angle = 75))

```



```

bio.lc.count.88 <- as.data.frame(bio.lc.count.88)
write.csv(bio.lc.count.88, "bio.lc.count.88.csv")

```

```

#2020 data
bio.lc.count.20 <- bio.lc.data.m2 %>%
  group_by(class_2020) %>%
  summarise(n())
colnames(bio.lc.count.20)[2] <- "count"

bio.lc.count.20

```

```

## # A tibble: 9 x 2
##   class_2020      count
##   <chr>        <int>
## 1 Brush/Shrubs    2754
## 2 Closed Forest    1373
## 3 Fishpond         17
## 4 Grassland        696
## 5 Inland Water     118
## 6 Marshland/Swamp  1246
## 7 No data          443
## 8 Open/Barren       288
## 9 Perennial Crop   1059

```

```

bio.lc.count.20 <- as.data.frame(bio.lc.count.20)

```

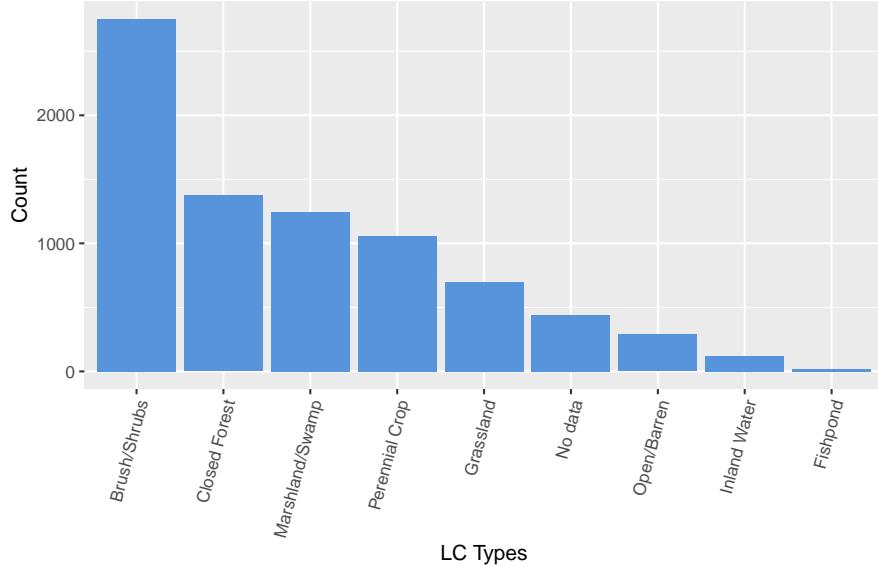
```

ggplot(data=bio.lc.count.20, aes(x=reorder(class_2020, -count), y=count)) +
  geom_bar(stat="identity", fill="#5794db") +
  labs(title = "Biodiversity data count per LC type (2020)", x = "LC Types", y = "Count") +

```

```
guides(x = guide_axis(angle = 75))
```

Biodiversity data count per LC type (2020)



```
bio.lc.count.20 <- as.data.frame(bio.lc.count.20)
write.csv(bio.lc.count.20, "bio.lc.count.20.csv")
```