Combined 2015 and 2023 Data Setup

Mark C. Otto

2024-07-01

## Introduction

We combine the 2015 and 2023 survey files: The spatial files are: MEBoundrySF, StrSF, StrPlotSF, ColonySF

The data files are GRTSLoc tblFlightInfo tblPlotDetails tblObservations tblStaff, RecObs.

Reference files: StrArea

These are combined over years. The flight, plot, and observation files combine into RecObs. Do I add the begplot and endplot in rRecObs` when I can get that information from Danielle’s tables? Why don’t I use her tables and create the list, plot, and caphist files for information from them.

I follow Danielle D’Auria’s convention of using \_PK of a primary key variable and \_FK for a foreign key variable.

The result can be an SQLite3 database and a R Savefile.

library(readr)  
library(readxl)  
library(sf)

Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf\_use\_s2() is TRUE

library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':  
  
 filter, lag

The following objects are masked from 'package:base':  
  
 intersect, setdiff, setequal, union

library(stringr)  
library(tidyr)  
library(purrr)  
library(ggplot2)  
library(RSQLite)  
source(here::here("R/RDB2Data.R"))  
  
# Survey years  
CurrentYr <- 2023  
cYear <- c(2015, CurrentYr)  
nYear <- length(cYear)  
  
# Survey frames  
cFrame <- c("List", "Area")  
nFrame <- length(cFrame)  
  
# Survey strata  
cDFStr <- c("SC", "Co", "ND")  
nStr <- length(cDFStr)  
  
# Make the year-stratum levels  
cYrStr <- apply(expand\_grid(Year = cYear, Str = cDFStr), 1, function(x)  
 paste(x[1], x[2], sep = "."))  
nYrStr <- length(cYrStr)  
  
# Make the year-frame-stratum levels.  
# Should call cYFS.  
cFS <- apply(expand\_grid(Year = cYear, Frame = cFrame, Str = cDFStr), 1, function(x)  
 paste(x[1], x[2], x[3], sep = "."))  
nFS <- length(cFS)  
  
lWrite <- TRUE

## CRS Projections

The spatial projections or CRS codes do not seem to define the projections correctly, so save them: Alpers equal area, UTS, and WGS84 (World Geodetic System 1984).

CRS.WGS84 <- st\_crs("EPSG:4326")  
tStrPlotSF <- st\_read("~/Profesional/GBHE.2023/GIS/StrPlotSp.shp")

Reading layer `StrPlotSp' from data source   
 `/Users/motto/Profesional/GBHE.2023/GIS/StrPlotSp.shp' using driver `ESRI Shapefile'  
Simple feature collection with 1084 features and 5 fields  
Geometry type: MULTIPOLYGON  
Dimension: XY  
Bounding box: xmin: 1931486 ymin: 2500109 xmax: 2260561 ymax: 3013336  
Projected CRS: North\_America\_Albers\_Equal\_Area\_Conic

CRS.Alpers <- st\_crs(tStrPlotSF)  
tColonySp <- st\_read("~/Profesional/GBHE.2023/GIS/2015/GBHE\_ColonyPlotJoin.shp")

Reading layer `GBHE\_ColonyPlotJoin' from data source   
 `/Users/motto/Profesional/GBHE.2023/GIS/2015/GBHE\_ColonyPlotJoin.shp'   
 using driver `ESRI Shapefile'  
Simple feature collection with 215 features and 31 fields  
Geometry type: POINT  
Dimension: XY  
Bounding box: xmin: 342393 ymin: 4776074 xmax: 629280 ymax: 5170445  
Projected CRS: NAD83 / UTM zone 19N

CRS.UTM19N <- st\_crs(tColonySp)  
if (lWrite) {  
 save(CRS.WGS84, file = here::here("data/CRS.WGS84.RData"))  
 save(CRS.Alpers, file = here::here("data/CRS.Alpers.RData"))  
 save(CRS.UTM19N, file = here::here("data/CRS.UTM19N.RData"))  
}

## Maine State Boundary Layer

The state boundary polygon, [Maine State Boundary Polygon Feature](https://maine.hub.arcgis.com/datasets/maine::maine-state-boundary-polygon-feature/explore), is WGS 84. This is just a reference layer, so leave the variables as is. Transform it to Alpers and save as a KML file

tMEBoundarySF <- st\_read("~/Profesional/GBHE.2023/GIS/Maine\_State\_Boundary\_Polygon\_Feature.shp")

Reading layer `Maine\_State\_Boundary\_Polygon\_Feature' from data source   
 `/Users/motto/Profesional/GBHE.2023/GIS/Maine\_State\_Boundary\_Polygon\_Feature.shp'   
 using driver `ESRI Shapefile'  
Simple feature collection with 6204 features and 9 fields  
Geometry type: POLYGON  
Dimension: XY  
Bounding box: xmin: -71.08392 ymin: 42.97703 xmax: -66.94942 ymax: 47.45986  
Geodetic CRS: WGS 84

MEBoundarySF <- st\_transform(x = tMEBoundarySF, crs = CRS.Alpers)  
if (lWrite) {  
 st\_write(  
 MEBoundarySF,  
 dsn = here::here("gis/MEBoundarySF.kml"),  
 delete\_layer = TRUE  
 )  
}

Writing layer `MEBoundarySF' to data source   
 `/Users/motto/Profesional/GBHE/gis/MEBoundarySF.kml' using driver `KML'  
Writing 6204 features with 9 fields and geometry type Polygon.

rm(tMEBoundarySF)

## Stratum Layer

Stratum, StrSF. Don’t worry about the complication of list and area strata that are not involved in this survey.

Maine is supposed to have 91,646km². Danielle may have removed some water area. Use the areas in StrArea.

StrArea <- RDB2Data(Tbl = "StrArea",  
 RowNames = "DFStr",  
 RDB = "~/Profesional/GBHE.2023/extdata/2023/DFS2023.sq3") %>%  
 select(Str\_PK = DFStr, cStratum, KMSq)  
fStr <- factor(StrArea$Str\_PK, StrArea$Str\_PK)  
  
tStrSF <- st\_read("~/Profesional/GBHE.2023/GIS/GBHE\_Stratum.shp") %>%  
 mutate(KMSq = AREA / 10 ^ 6, cStratum = NAME2\_) %>%  
 select(cStratum, KMSq)

Reading layer `GBHE\_Stratum' from data source   
 `/Users/motto/Profesional/GBHE.2023/GIS/GBHE\_Stratum.shp' using driver `ESRI Shapefile'  
Simple feature collection with 3 features and 3 fields  
Geometry type: POLYGON  
Dimension: XY  
Bounding box: xmin: 336611.8 ymin: 4766602 xmax: 664711.9 ymax: 5255774  
Projected CRS: NAD83 / UTM zone 19N

# `dplyr` joins removes the `sf` class.  
tStrSF2 <- st\_as\_sf(inner\_join(tibble::as\_tibble(  
 x = list(Str\_FK = fStr, cStratum = StrArea$cStratum)  
), tStrSF, by = "cStratum") %>%  
 arrange(Str\_FK))  
  
StrSF <- st\_transform(x = tStrSF2, crs = CRS.Alpers)  
if (lWrite) {  
 st\_write(StrSF,  
 dsn = here::here("gis/StrSF.kml"),  
 delete\_layer = TRUE)  
}

Writing layer `StrSF' to data source   
 `/Users/motto/Profesional/GBHE/gis/StrSF.kml' using driver `KML'  
Writing 3 features with 3 fields and geometry type Polygon.

rm(tStrSF, tStrSF2)

## Colony List for both years

We should start with the colony list we had in 2015, add the new colonies found in the 2015 survey and any in the following eight years.

What is odd is ME.0712 was dropped from the list and we had new colonies at least found in the survey. If ME.0712 was in active since the last survey, then there is good reason to drop it.

The AnnObs table is a good way to create the list colonies for each survey, but we don’t have a current set of those observations.

To add the 2015 new colonies, use the locations and match them to the AllColonies list. Ask Danielle.

Make the spatial layer out of the .csv files.

tColony2015 <- RDB2Data(Tbl = "Nest",  
 RowNames = "FWSNestID",  
 RDB = "~/Profesional/GBHE.2023/extdata/2015/DFS2015.sq3") %>%  
 select(  
 FWSColonyID\_PK = FWSNestID,  
 Str\_FK = LstStr,  
 PlotID\_FK = PlotID,  
 Lng,  
 Lat,  
 LastYr,  
 Status,  
 nNest,  
 nActive,  
 Comment  
 )  
  
# These are all the colonies. 445 of them so select from the last list  
# Where are the new colonies found in the last survey.  
tColony2023 <- inner\_join(  
 RDB2Data(  
 Tbl = "Nest",  
 RowNames = "FWSNestID",  
 RDB = "~/Profesional/GBHE.2023/extdata/2023/DFS2023.sq3"  
 ) %>%  
 select(  
 FWSColonyID\_PK = FWSNestID,  
 Str\_FK = LstStr,  
 PlotID\_FK = PlotID,  
 Lng,  
 Lat,  
 LastYr,  
 Status,  
 nNest,  
 nActive,  
 Comment  
 )  
 ,  
 read\_csv(  
 "~/Profesional/GBHE.2023/extdata/2023/GBHE\_ColonyPlotJoin.csv",  
 col\_types = "------i---------"  
 ) %>%  
 mutate(FWSColonyID\_PK = sprintf("ME.%04d", Colony)) %>%  
 select(FWSColonyID\_PK),  
 by = "FWSColonyID\_PK"  
)  
setdiff(tColony2023$FWSColonyID\_PK, tColony2015$FWSColonyID\_PK)

character(0)

setdiff(tColony2015$FWSColonyID\_PK, tColony2023$FWSColonyID\_PK)

[1] "ME.0712"

tColony2015SF <- inner\_join(  
 st\_read("~/Profesional/GBHE.2023/GIS/2015/GBHE\_ColonyPlotJoin.shp") %>%  
 mutate(FWSColonyID\_PK = sprintf("ME.%04d", COLONYNO)) %>%  
 select(FWSColonyID\_PK),  
 tColony2015,  
 by = "FWSColonyID\_PK"  
)

Reading layer `GBHE\_ColonyPlotJoin' from data source   
 `/Users/motto/Profesional/GBHE.2023/GIS/2015/GBHE\_ColonyPlotJoin.shp'   
 using driver `ESRI Shapefile'  
Simple feature collection with 215 features and 31 fields  
Geometry type: POINT  
Dimension: XY  
Bounding box: xmin: 342393 ymin: 4776074 xmax: 629280 ymax: 5170445  
Projected CRS: NAD83 / UTM zone 19N

# For now, make an entry for each year. We should get these from  
# and annual observations file like `AnnObs`.  
ColonySF <- st\_as\_sf(bind\_rows(  
 tColony2015SF %>%  
 mutate(Year = 2015),  
 tColony2015SF %>%  
 mutate(Year = 2023)  
)) %>%  
 select(  
 FWSColonyID\_PK,  
 Year,  
 Str\_FK,  
 PlotID\_FK,  
 Lng,  
 Lat,  
 LastYr,  
 Status,  
 nNest,  
 nActive,  
 Comment  
 )  
  
if(lWrite) {  
 st\_write(ColonySF,  
 dsn = here::here("gis/ColonySF.kml"),  
 delete\_layer = TRUE)  
}

Writing layer `ColonySF' to data source   
 `/Users/motto/Profesional/GBHE/gis/ColonySF.kml' using driver `KML'  
Writing 430 features with 11 fields and geometry type Point.

rm(tColony2015, tColony2023, tColony2015SF)

## Non-Spatial Colony List

Colony <- st\_drop\_geometry(ColonySF) %>%  
 select(  
 Year,  
 FWSColonyID = FWSColonyID\_PK,  
 Str = Str\_FK,  
 PlotID = PlotID\_FK,  
 Lng,  
 Lat,  
 nNest,  
 nActive  
 )  
if (lWrite) {  
 readr::write\_csv(Colony, here::here("data/Colony.csv"), na = "")  
}

## Stratum-Plot Layer

I intersected or spatially joined the stratum with the plot layers, so use StrPlotSF. Later create the lat-long using st\_centroid(). Also copy over the Plot table that is non spatial but has number of colonies and km2. The row and col come from RC2XY made in the Bald Eagle national grid.

tStrPlotSF <- st\_read("~/Profesional/GBHE.2023/GIS/StrPlotSp.shp") %>%  
 select(Str\_FK = Str, PlotID\_FK = PlotID, KMSq, Col, Row)

Reading layer `StrPlotSp' from data source   
 `/Users/motto/Profesional/GBHE.2023/GIS/StrPlotSp.shp' using driver `ESRI Shapefile'  
Simple feature collection with 1084 features and 5 fields  
Geometry type: MULTIPOLYGON  
Dimension: XY  
Bounding box: xmin: 1931486 ymin: 2500109 xmax: 2260561 ymax: 3013336  
Projected CRS: North\_America\_Albers\_Equal\_Area\_Conic

tLatLong <- st\_transform(x = st\_centroid(tStrPlotSF), crs = CRS.WGS84)

Warning: st\_centroid assumes attributes are constant over geometries

LatLong <- st\_drop\_geometry(tLatLong %>%  
 mutate(Lng = unlist(map(  
 tLatLong$geometry, 1  
 )), Lat = unlist(map(  
 tLatLong$geometry, 2  
 ))) %>%  
 select(Lng, Lat))  
  
# Add the long-lat to the nestlist  
StrPlotSF <- bind\_cols(tStrPlotSF, LatLong) %>%  
 select(Str\_FK, PlotID\_FK, KMSq, Lng, Lat, Col, Row)  
if(lWrite) {  
 st\_write(StrPlotSF, here::here("GIS/StrPlotSF.kml"), delete\_layer = TRUE)  
}

Writing layer `StrPlotSF' to data source   
 `/Users/motto/Profesional/GBHE/GIS/StrPlotSF.kml' using driver `KML'  
Writing 1084 features with 7 fields and geometry type Multi Polygon.

rm(tStrPlotSF, tLatLong, LatLong)

## All Known Colonies

We need allcolonies to incorporate the 2015 new nests into the 2023 list and some colonies not in the original list were observered.

# NestList is the colony list. Need to remove all the colonies without  
# locations.  
tColonySF <- read\_csv(file = "~/Profesional/GBHE.2023/extdata/2023/COLONIES.csv", col\_types = "ic--c---------cccc---dd---c-d------ii") %>%  
 filter(!is.na(X\_UTMNAD83)) %>%  
 mutate(  
 FWSColonyID = sprintf("ME.%04d", COLONYNO),  
 Year = ifelse(Found2009, "2009", "<2009"),  
 Comment = paste(  
 SiteID,  
 ColName,  
 HabitatDesc,  
 Directions,  
 Comments,  
 History,  
 sep = ";"  
 )  
 ) %>%  
 select(FWSColonyID, Nest = COLSIZE, Year, Comment, X\_UTMNAD83, Y\_UTMNAD83) %>%  
 # 0919 was a duplicate.  
 distinct() %>%  
 st\_as\_sf(coords = c("X\_UTMNAD83", "Y\_UTMNAD83"),  
 crs = CRS.UTM19N)  
  
tLatLong <- st\_transform(x = tColonySF, crs = CRS.WGS84)  
  
LatLong <- st\_drop\_geometry(tLatLong %>%  
 mutate(Lng = unlist(map(  
 tLatLong$geometry, 1  
 )), Lat = unlist(map(  
 tLatLong$geometry, 2  
 ))) %>%  
 select(Lng, Lat))  
  
# Add the long-lat to the nestlist  
tColonySF2 <- bind\_cols(tColonySF, LatLong)  
  
AllColonySF <- st\_intersection(  
 StrPlotSF %>%  
 select(Str = Str\_FK, PlotID = PlotID\_FK),  
 st\_transform(x = tColonySF2, crs = st\_crs(StrPlotSF))  
) %>%  
 select(FWSColonyID, Year, Str, PlotID, Nest, Lng, Lat, Comment)

Warning: attribute variables are assumed to be spatially constant throughout  
all geometries

if (lWrite) {  
 st\_write(AllColonySF,  
 here::here("GIS/AllColonySF.kml"),  
 delete\_layer = TRUE)  
}

Writing layer `AllColonySF' to data source   
 `/Users/motto/Profesional/GBHE/GIS/AllColonySF.kml' using driver `KML'  
Writing 375 features with 8 fields and geometry type Point.

rm(tColonySF, tColonySF2, tLatLong, LatLong)

## List Plot List

Doesn’t this depend on the year the sample was taken as to what is put in the list? For now we use all colonies.

List <- st\_drop\_geometry(ColonySF) %>%  
 select(Str = Str\_FK,  
 PlotID = PlotID\_FK,  
 nNest,  
 # Should standardize: nActive or nAct  
 nAct = nActive) %>%  
 group\_by(Str, PlotID) %>%  
 summarize(  
 nCol = n(),  
 nNest = sum(nNest, na.rm = TRUE),  
 nAct = sum(nAct, na.rm = TRUE),  
 .groups = "drop"  
 )  
if (lWrite) {  
 readr::write\_csv(List, here::here("data/List.csv"), na = "")  
}

## Identify the colonies used for known colonies.

tColUsed <- read\_csv("~/Profesional/GBHE.2023/extdata/2023/GBHE\_ColonyPlotJoin.csv",  
 col\_types = "---idd------ii----------------c-") %>%  
 mutate(FWSColonyID = sprintf("ME.%04d", COLONYNO),  
 DatEntered = "2022-01-01") %>%  
 select(FWSColonyID, DatEntered)  
  
# # Identfy the colonies used by 2022 for the `DatEntered` and 2008 otherwise.  
# ColoniesSF <- left\_join(tColonySF2, tColUsed, by = "FWSColonyID") %>%  
# mutate(  
# St = "ME",  
# StNestID = as.integer(sub("ME.", "", FWSColonyID)),  
# LocAcc = .1,  
# Location = "",  
# Access = "",  
# Owner = "Danielle D'Auria",  
# ListYr = 2008,  
# DatEntered = ifelse(is.na(DatEntered), "2008-01-01", DatEntered),  
# Comment = ""  
# ) %>%  
# select(  
# FWSColonyID ,  
# St ,  
# StNestID ,  
# PlotID ,  
# Str ,  
# Lng ,  
# Lat ,  
# LocAcc ,  
# Location ,  
# Access ,  
# Owner ,  
# ListYr ,  
# DatEntered,  
# Comment  
# ) %>%  
# arrange(FWSColonyID)  
# if (lWrite) {  
# st\_write(ColonySF, here::here("GIS/AllColonySF.kml"), delete\_layer = TRUE)  
# }

## Plot Non-Spatial

Add the nests too.

tPlot <- st\_drop\_geometry(StrPlotSF) %>%  
 select(Str = Str\_FK, PlotID = PlotID\_FK, KMSq, Lng, Lat)  
  
Plot <- left\_join(tPlot, List, by = c("Str", "PlotID")) %>%  
 mutate(  
 nCol = ifelse(is.na(nCol), 0, nCol),  
 nNest = ifelse(is.na(nNest), 0, nNest),  
 nAct = ifelse(is.na(nAct), 0, nAct)  
 ) %>%  
 select(Str, PlotID, Lng, Lat, KMSq, nCol, nNest, nAct)  
if (lWrite) {  
 write\_csv(Plot, here::here("GIS/Plot.csv"), na = "")  
}  
rm(tPlot)

## GRTS Sample

Combine the current year GRTS sample file with the past years. Add the year onto the tables. Archive GRTSLoc that has all the information used for the sample design. Make GRTSSmp that identifies the samples flown and keep that for the estimation.

L79 was sampled, so correct EvalStatus.

tGRTSLoc <- RDB2Data(  
 Tbl = "GRTSLoc",  
 RowNames = "siteID",  
 RDB = "~/Profesional/GBHE.2023/extdata/2015/DFS2015.sq3",  
 envir = NULL  
) %>%  
 mutate(Year = 2015)  
  
tGRTSCurrent <- RDB2Data(  
 Tbl = "GRTSLoc",  
 RowNames = "siteID",  
 RDB = "~/Profesional/GBHE.2023/extdata/2023/DFS2023.sq3",  
 envir = NULL  
) %>%  
 mutate(Year = CurrentYr)  
  
  
tGRTSLoc2 <- bind\_rows(tGRTSLoc, tGRTSCurrent) %>%  
 mutate(  
 Str = stratum,  
 siteID = Plot,  
 EvalStatus = ifelse(is.na(EvalStatus), FALSE, EvalStatus == "Evaluated"),  
 EvalStatus = ifelse(siteID == "L79", TRUE, EvalStatus)  
 )  
  
GRTSLoc <- inner\_join(tGRTSLoc2,  
 Plot %>%  
 select(Str, PlotID, KMSq, nAct),  
 by = c("Str", "PlotID")) %>%  
 select(  
 Year,  
 siteID,  
 Str,  
 PlotID,  
 SmpType,  
 xcoord,  
 ycoord,  
 mdcaty,  
 wgt,  
 KMSq,  
 nAct,  
 panel,  
 EvalStatus,  
 EvalReason,  
 Comment  
 )  
if (lWrite) {  
 readr::write\_csv(GRTSLoc, here::here("data/GRTSLoc.csv"), na = "")  
}  
rm(tGRTSLoc, tGRTSLoc2, tGRTSCurrent)

## GRTSSmp for the Estimation

GRTSSmp <- GRTSLoc %>%  
 filter(EvalStatus) %>%  
 select(Year, siteID, Str, PlotID, SmpType, KMSq, nAct)  
if (lWrite) {  
 readr::write\_csv(GRTSSmp, here::here("data/GRTSSmp.csv"), na = "")  
}

The colony observations, RecObs are created from tblFlightInfo, tblObservations, tblPlotDetails, and tblStaff.

First combine the flight and plot files. Problems with commas in the weather notes, WxNotes in the 2023 table. Put quotes around the field.

tblFlightInfo2015 <- readr::read\_csv("~/Profesional/GBHE.2015/ExtData/GBHEDualFrame2/tblFlightInfo.csv",  
 col\_types = "iiiiccccccc")  
  
tblFlightInfo2023 <- read\_csv("~/Profesional/GBHE.2023/extdata/2023/tblFlightInfo.csv",  
 col\_types = "iiiiccccccc")  
# %>%mutate(Start = as.character(Start), End = as.character(End))  
tblFlightInfo <- bind\_rows(tblFlightInfo2015, tblFlightInfo2023) %>%  
 mutate(  
 Date = sprintf("%04d-%02d-%02d", DateYear, DateMonth, DateDay),  
 Start = as.POSIXct(Start, format = "%H:%M:%S"),  
 End = as.POSIXct(End, format = "%H:%M:%S"),  
 Comment = paste(ifelse(is.na(WxNotes), "", WxNotes), ifelse(is.na(Comments), "", Comments), sep = "; ")  
 ) %>%  
 select(  
 Flight\_PK,  
 Year = DateYear,  
 Date,  
 Start,  
 End,  
 Pilot,  
 Front = FObs,  
 Rear = RObs,  
 Comment  
 )  
if (lWrite) {  
 readr::write\_csv(tblFlightInfo, here::here("data/tblFlightInfo.csv"), na = "")  
}  
rm(tblFlightInfo2015, tblFlightInfo2023)

## Merge the plot tables

Combine with GRTSLoc to add Str and PlotID. The OOS will not have stratum or plot ID.

tblPlotDetails2015 <- readr::read\_csv("~/Profesional/GBHE.2015/ExtData/GBHEDualFrame2/tblPlotDetails.csv",  
 col\_types = "iclccccci") %>%  
 mutate(Year = 2015)  
  
tblPlotDetails2023 <- read\_csv("~/Profesional/GBHE.2023/extdata/2023/tblPlotDetails.csv",  
 col\_types = "iiclccccc") %>%  
 mutate(Year = CurrentYr)  
  
ttblPlotDetails <- bind\_rows(tblPlotDetails2015, tblPlotDetails2023) %>%  
 mutate(  
 BegArea = as.POSIXct(BegArea, format = "%H:%M:%S"),  
 EndArea = as.POSIXct(EndArea, format = "%H:%M:%S"),  
 BegList = as.POSIXct(BegList, format = "%H:%M:%S"),  
 EndList = as.POSIXct(EndList, format = "%H:%M:%S"),  
 Comment = ifelse(`Combined?`, paste("Combined.", ifelse(  
 is.na(Comments), "", Comments  
 )), Comments)  
 )  
  
tblPlotDetails <-  
 left\_join(  
 ttblPlotDetails,  
 GRTSLoc %>%  
 select(Year, SiteID = siteID, Str, PlotID),  
 by = c("Year", "SiteID")  
 ) %>%  
 select(  
 PlotDetails\_PK,  
 Year,  
 Str,  
 Flight\_FK,  
 SiteID,  
 PlotID,  
 BegArea,  
 EndArea,  
 BegList,  
 EndList,  
 Comment  
 )  
if (lWrite) {  
 readr::write\_csv(tblPlotDetails, here::here("data/tblPlotDetails.csv"), na = "")  
}  
rm(tblPlotDetails2015, tblPlotDetails2023, ttblPlotDetails)

## 2015 Observations

I am assuming the coordinates are in UTM 19N. Convert them to lat-longs for the few that have coordinates. These are important since the GBHE ones are the new colonies and we have to add them to the 2023 colony list.

Save LatLong to correct RecObs2015 too.

ttblObservations2015 <- readr::read\_csv(  
 "~/Profesional/GBHE.2015/ExtData/GBHEDualFrame2/tblObservations.csv",  
 col\_types = "iclicciiiicccicii"  
) %>%  
 mutate(Year = 2015) %>%  
 select(  
 Obs\_PK,  
 PlotDetails\_FK,  
 WaypointID,  
 `NewColony?`,  
 WBC\_ID,  
 Strata,  
 Time,  
 CaptHist,  
 FObsTot,  
 FObsAct,  
 RObsTot,  
 RObsAct,  
 Species,  
 Habitat,  
 Comments,  
 Latitude,  
 Longitude,  
 Year  
 )  
  
# For only those observations with locations.  
Loc2015SF <- ttblObservations2015 %>%  
 filter(!is.na(Latitude)) %>%  
 mutate(xUTM = Longitude, yUTM = Latitude) %>%  
 st\_as\_sf(coords = c("Longitude", "Latitude"), crs = CRS.UTM19N) %>%  
 st\_transform(crs = CRS.WGS84)  
  
LatLong <- st\_drop\_geometry(Loc2015SF %>%  
 mutate(Lng = unlist(map(  
 Loc2015SF$geometry, 1  
 )), Lat = unlist(map(  
 Loc2015SF$geometry, 2  
 ))) %>%  
 select(Obs\_PK, xUTM, yUTM, Lat, Lng))  
  
tblObservations2015 <- left\_join(ttblObservations2015, LatLong, by = "Obs\_PK") %>%  
 select(  
 Obs\_PK,  
 PlotDetails\_FK,  
 WaypointID,  
 `NewColony?`,  
 WBC\_ID,  
 Strata,  
 Time,  
 CaptHist,  
 FObsTot,  
 FObsAct,  
 RObsTot,  
 RObsAct,  
 Species,  
 Habitat,  
 Comments,  
 Latitude = Lat,  
 Longitude = Lng,  
 Year  
 )  
  
rm(ttblObservations2015, Loc2015SF)

## 2023 Observations

Make edits that I would have made on RecObs.

tblObservations2023 <- read\_csv("~/Profesional/GBHE.2023/extdata/2023/tblObservations.csv",  
 col\_types = "iicliccciiiicccii") %>%  
 mutate(Year = CurrentYr)

Warning: One or more parsing issues, call `problems()` on your data frame for details,  
e.g.:  
 dat <- vroom(...)  
 problems(dat)

## Combine Observations

Need to edit the 2023 observations like Danielle and I did with RecObs. and add all the locations for the known colonies. Also, many of the nests are not on the colony list we used for 2023. 1. They don’t have the 2015 new nests and 2. observed ones that are not on the list we used but are on the all colonies list.

To get this done, combine RecObs.

tblObservations <- bind\_rows(tblObservations2015, tblObservations2023) %>%  
 mutate(  
 FWSColonyID = ifelse(is.na(WBC\_ID), NA, ifelse(  
 `NewColony?`,  
 sprintf("new.%04d", WBC\_ID),  
 sprintf("ME.%04d", WBC\_ID)  
 )),  
 Str = Strata,  
 CapHist = tolower(CaptHist),  
 ) %>%  
 select(  
 Year,  
 Obs\_PK,  
 PlotDetails\_FK,  
 Lng = Longitude,  
 Lat = Latitude,  
 AOU = Species,  
 CapHist,  
 Location = Habitat,  
 FWSColonyID,  
 fNest = FObsTot,  
 rNest = RObsTot,  
 fAct = FObsAct,  
 rAct = RObsAct,  
 Comment = Comments  
 )  
if (lWrite) {  
 readr::write\_csv(tblObservations,  
 here::here("data/tblObservations.csv"),  
 na = "")  
}  
  
rm(tblObservations2015, tblObservations2023)

## Combine RecObs from tbl\* data files

tRecObs <- inner\_join(tblFlightInfo,  
 tblPlotDetails,  
 by = c(Flight\_PK = "Flight\_FK", "Year")) %>%  
 mutate(Comment = paste(Comment.x, Comment.y, sep = ";")) %>%  
 select(  
 Flight\_PK,  
 PlotDetails\_PK,  
 Year,  
 Date,  
 BegFlight = Start,  
 EndFlight = End,  
 Pilot ,  
 Front,  
 Rear,  
 Str ,  
 siteID = SiteID,  
 PlotID,  
 BegArea,  
 EndArea,  
 BegList,  
 EndList,  
 Comment  
 )  
  
# I should do a spatial join for every colony observed in case there  
# are any strata that split sample plots.  
tRecObs2 <- left\_join(  
 tRecObs %>%  
 select(  
 Flight\_PK,  
 PlotDetails\_PK,  
 Year,  
 siteID,  
 PlotID,  
 Date,  
 Pilot,  
 Front,  
 Rear,  
 Comment  
 ),  
 GRTSLoc %>%  
 select(Year, siteID, PlotID, Str),  
 by = c("Year", "siteID", "PlotID")  
) %>%  
 # Should not need this  
 distinct()  
  
# Merge the flight, plot, and observation tables to make `RecObs`.  
# Keep all the flight, plot, and observation keys for reference.  
# All the sampled colonies have a stratum and plot.  
# Don't match the bald eagle RecObs. Obs\_PK is equivalent to OBJECTID.  
tRecObs3 <- inner\_join(tRecObs2,  
 tblObservations,  
 by = c(PlotDetails\_PK = "PlotDetails\_FK", "Year")) %>%  
 mutate(AOU = "GBHE",  
 Comment = paste(Comment.x, Comment.y, sep = ";")) %>%  
 select(  
 Obs\_PK,  
 Year,  
 Str,  
 siteID,  
 PlotID,  
 DatTim = Date,  
 Pilot,  
 Front,  
 Rear,  
 Lng,  
 Lat,  
 AOU,  
 CapHist,  
 Location,  
 FWSColonyID,  
 fNest,  
 rNest,  
 fAct,  
 rAct,  
 Comment,  
 Flight\_PK,  
 PlotDetails\_PK  
 )  
  
tRecObs4 <- left\_join(  
 tRecObs3,  
 ColonySF %>%  
 select(  
 FWSColonyID = FWSColonyID\_PK,  
 Lng,  
 Lat #,  
 # Str = Str\_FK,  
 # PlotID = PlotID\_FK  
 ),  
 by = "FWSColonyID"  
) %>%  
 distinct() %>%  
 mutate(  
 Lng = ifelse(is.na(Lng.x), Lng.y, Lng.x),  
 Lat = ifelse(is.na(Lat.x), Lat.y, Lat.x) #,  
 # Str = ifelse(is.na(Str.x), Str.y, ifelse(  
 # Str.x == Str.y, Str.x, paste(Str.x, Str.y, sep = "!=")  
 # )),  
 # PlotID = ifelse(  
 # is.na(PlotID.x),  
 # PlotID.y,  
 # ifelse(  
 # PlotID.x == PlotID.y,  
 # PlotID.x,  
 # paste(PlotID.x, PlotID.y, sep = "!=")  
 # )  
 # )  
 ) %>%  
 select(  
 Obs\_PK,  
 Year,  
 Str,  
 siteID,  
 PlotID,  
 DatTim,  
 Pilot,  
 Front,  
 Rear,  
 Lng,  
 Lat,  
 AOU,  
 CapHist,  
 Location,  
 FWSColonyID,  
 fNest,  
 rNest,  
 fAct,  
 rAct,  
 Comment,  
 Flight\_PK,  
 PlotDetails\_PK  
 ) %>%  
 arrange(Obs\_PK)

Warning in left\_join(tRecObs3, ColonySF %>% select(FWSColonyID = FWSColonyID\_PK, : Detected an unexpected many-to-many relationship between `x` and `y`.  
ℹ Row 1 of `x` matches multiple rows in `y`.  
ℹ Row 200 of `y` matches multiple rows in `x`.  
ℹ If a many-to-many relationship is expected, set `relationship =  
 "many-to-many"` to silence this warning.

RecObs <- st\_drop\_geometry(tRecObs4)  
  
if (lWrite) {  
 readr::write\_csv(RecObs, here::here("data/RecObs.csv"), na = "")  
}  
  
rm(tRecObs, tRecObs2, tRecObs3)

## Combine RecObs from year-specific RecObs

Use Obs\_FK rather than OBJECTID.  
In 2015, change the UTMs to lat-longs. Add the locations for 2015 using the colony list.

RecObs2015 <- RDB2Data(Tbl = "RecObs",  
 RowNames = "OBJECTID",  
 RDB = "~/Profesional/GBHE.2023/extdata/2015/DFS2015.sq3") %>%  
 mutate(Yr = 2015,  
 siteID = ifelse(siteID == "outplot", "OOS", siteID)) %>%  
 mutate(FWSColonyID = FWSNestID)  
  
RecObs2023 <- RDB2Data(Tbl = "RecObs",  
 RowNames = "OBJECTID",  
 RDB = "~/Profesional/GBHE.2023/extdata/2023/DFS2023.sq3") %>%  
 mutate(FWSColonyID = stringr::str\_replace(FWSNestID, "new\\.[:alpha:][:alpha:]", "ME"))  
  
tRecObs <- bind\_rows(RecObs2015, RecObs2023) %>%  
 select(  
 OBJECTID,  
 Year = Yr,  
 St,  
 Svy,  
 DatTim,  
 Pilot,  
 Front,  
 Rear,  
 Str,  
 PlotID,  
 siteID,  
 Lng,  
 Lat,  
 AOU,  
 CapHist,  
 Location,  
 FWSColonyID,  
 fNest,  
 rNest,  
 fAct,  
 rAct,  
 Comment,  
 Edit,  
 File,  
 Rec,  
 FWSNestID  
 )  
  
tRecObs2 <- left\_join(tRecObs,  
 AllColonySF %>% select(FWSColonyID, Lng, Lat, Str, PlotID),  
 by = "FWSColonyID") %>%  
 mutate(  
 Lng = ifelse(is.na(Lng.x), Lng.y, Lng.x / 10 ^ 4),  
 Lat = ifelse(is.na(Lat.x), Lat.y, Lat.x / 10 ^ 5),  
 Str = ifelse(is.na(Str.x), Str.y, ifelse(  
 Str.x == Str.y, Str.x, paste(Str.x, Str.y, sep = "!=")  
 )),  
 PlotID = ifelse(  
 is.na(PlotID.x),  
 PlotID.y,  
 ifelse(  
 PlotID.x == PlotID.y,  
 PlotID.x,  
 paste(PlotID.x, PlotID.y, sep = "!=")  
 )  
 )  
 ) %>%  
 select(  
 OBJECTID,  
 Year,  
 St,  
 Svy,  
 DatTim,  
 Pilot,  
 Front,  
 Rear,  
 Str,  
 PlotID,  
 siteID,  
 Lng,  
 Lat,  
 AOU,  
 CapHist,  
 Location,  
 FWSColonyID = FWSNestID,  
 fNest,  
 rNest,  
 fAct,  
 rAct,  
 Comment,  
 Edit,  
 File,  
 Rec  
 )  
  
RecObsYS <- tRecObs2

## Edit 2023 Observations

# MCO: The ME.0647 seems to be off c6.1v vs. c4.1r (2 over and 3 up).  
# DDA: This should be c6.1v.  
# This is probably due to the original data entry error where it was entered as 105.  
# MCO: ColList: ME.0647 c6.1v SC -70.78627 43.17239  
# It was in a list plot, L63, with no nests.  
# c6.1v is not a sample plot, but this was the colony that was sampled?  
# ColList: ME.0105 c4.1r SC -70.82679 43.61651  
# 105 is in the c4.1r or the L63 sample plot. Is 105 647?  
# RecObs[25,"FWSColonyID"]  
iObs <- which(RecObs$FWSColonyID == "ME.0647")  
RecObs[iObs, "PlotID"] <- "c6.1v"  
RecObs[iObs, "siteID"] <- "OOS"  
RecObs[iObs, "Edit"] <- "Changed PlotID from c4.1r and siteID from L63."  
  
# MCO: new.SC.0917 may be just over the border, so go with what you marked it as.  
# DDA: This should be ci.18.  
# MCO: So it is outside the A16 ci.19 sample plot?  
# That would make it OOS. I will change it.  
# RecObs[81,"FWSColonyID"]  
iObs <- which(RecObs$FWSColonyID == "new.SC.0917")  
RecObs[iObs, "PlotID"] <- "ci.18"  
RecObs[iObs, "siteID"] <- "OOS"  
RecObs[iObs, "Edit"] <- "Changed PlotID from ci.19 and siteID from A18."  
  
# MCO: ch.1c is a split plot, so go with the Co part of the plot?  
# DDA: Yes, this colony (866) is in Co stratum.  
# MCO: Good  
# RecObs[184,"FWSColonyID"]  
iObs <- which(RecObs$FWSColonyID == "ME.0866")  
RecObs[iObs, "Str"] <- "Co"  
RecObs[iObs, "Edit"] <- "Changed Str from SC because it was a plot split between strata."  
  
# MCO: ME.0721 is just over in the next plot, so should we stick with the  
# assigned plot as it is the sample plot?  
# DDA: This should be c4.16  
# MCO: c4.16 is not a sample plot, but it was a list plot and had no nests.  
# RecObs[217,"FWSColonyID"]  
iObs <- which(RecObs$FWSColonyID == "ME.0721")  
RecObs[iObs, "PlotID"] <- "c4.16"  
RecObs[iObs, "siteID"] <- "OOS"  
RecObs[iObs, "Edit"] <- "Changed PlotID from c4.15 and siteID from L73."  
  
# MCO: cd.1l (1 down and 5 up) is a ways off. Not sure what to do for ME.0739.  
# DDA: This should be cd.1l  
# MCO: cd.1l is the listover. I can fix that.  
# RecObs[221,"FWSColonyID"]  
iObs <- which(RecObs$FWSColonyID == "ME.0739")  
RecObs[iObs, "PlotID"] <- "cd.1l"  
RecObs[iObs, "siteID"] <- "L79"  
RecObs[iObs, "Edit"] <- "Changed PlotID from c4.16 and siteID from L03."  
  
# New nests from talking to DDA 27 June 2024:  
# OBJECTID, siteID, ColonyID  
# 12 A27 848 ce.1i? Co?  
iObs <- grep("A27", RecObs$siteID)  
  
# 115 A37 850  
# 126 A18 851  
# 137 A33 852  
# OOS 846

## Spatial RecObs

# Make a spatial file. `tRecObs4` is a spatial file   
tRecObsSF2 <- tRecObs4 %>%  
 filter(!is.na(Lng)) %>%  
 mutate(  
 Lng2 = Lng,  
 Lat2 = Lat  
 ) %>%  
 st\_as\_sf(coords = c("Lng2", "Lat2"), crs = CRS.WGS84)  
  
# This only had 137 observations, so it some locations are off.  
tRecObsSF3 <- st\_intersection(StrPlotSF %>%  
 select(Str\_FK, PlotID\_FK),  
 st\_transform(x = tRecObsSF2, crs = st\_crs(StrPlotSF))) %>%  
 mutate(Str = ifelse(is.na(Str), Str\_FK, Str),  
 PlotID = ifelse(is.na(PlotID), PlotID\_FK, PlotID))

Warning: attribute variables are assumed to be spatially constant throughout  
all geometries

RecObsSF <- st\_as\_sf(tRecObsSF2 %>%  
 filter(!is.na(Lng)),  
 coords = c("Lng", "Lat"),  
 crs = CRS.WGS84) %>%  
 st\_transform(crs = CRS.UTM19N)  
if (lWrite) {  
 st\_write(RecObsSF, here::here("GIS/RecObsSF.kml"), delete\_layer = TRUE)  
}

Writing layer `RecObsSF' to data source   
 `/Users/motto/Profesional/GBHE/GIS/RecObsSF.kml' using driver `KML'  
Writing 275 features with 22 fields and geometry type Point.

rm(  
 # RecObs2015,  
 # RecObs2023,  
 # tRecObs,  
 # tRecObs2,  
 tRecObs4,  
 tRecObsSF2,  
 tRecObsSF3)

## Check RecObs

Check stratum and plot for those that have it and add it where they are missing.

tRecObsSF <- st\_as\_sf(RecObs, coords = c("Lng", "Lat"), crs = st\_crs(4326))  
tRecObsSF2 <- st\_intersection(StrPlotSF %>%  
 select(tStr = Str, tPlotID = PlotID),   
 st\_transform(x = tRecObsSF, crs = st\_crs(StrPlotSF))) %>%  
 mutate(  
 Str = ifelse(is.na(Str), tStr, Str),  
 PlotID = ifelse(is.na(PlotID), tPlotID, PlotID)  
 )  
  
# Check of stratum and plot differences. Resolve.  
tRecObsSF2 %>%  
 filter(tStr != Str | tPlotID != PlotID)  
  
# Edit RecObs, to avoid the complications of dropping the spatial  
# and rearranging the columns.  
  
RecObs$Str <- ifelse(is.na(RecObs$Str), tRecObsSF2$Str, RecObs$Str)  
RecObs$PlotID <- ifelse(is.na(RecObs$PlotID), tRecObsSF2$PlotID, RecObs$PlotID)  
RecObs$FWSColonyID <- ifelse(  
 grepl("\\. \\.", RecObs$FWSColonyID),  
 stringr::str\_replace(RecObs$FWSColonyID, " ", RecObs$Str),  
 RecObs$FWSColonyID  
)

## Ground Observations

Add the ground observations. Make a spatial file by adding the colony information.

GroundObs2015 <- read\_excel(path = "~/Profesional/GBHE.2015/Ori/GroundObs\_DualFrame.xlsx", sheet = "Sheet1", skip = 1) %>%  
 mutate(  
 # OBJECTID = row\_number(),  
 Year = 2015,  
 Frame = "List",  
 Svy = "GBHE",  
 DatTim = as.character(Date),  
 FWSColonyID = sprintf("ME.%04d", Colony),  
 Obsr = "Gnd",  
 Seat = "Dirt",  
 Comment = NA  
 ) %>%  
 rename(Nest = `Total Nests`,  
 Act = `Active Nests`) %>%  
 select(Year,  
 Frame,  
 Svy,  
 DatTim,  
 FWSColonyID,  
 Obsr,  
 Seat,  
 Nest,  
 Act,  
 Comment)  
  
GroundObs2023 <- read\_csv(file = "~/Profesional/GBHE.2023/extdata/2023/GroundCounts.csv", col\_types = "cciiii------cc") %>%  
 mutate(  
 # OBJECTID = row\_number(),  
 Year = 2023,  
 Frame = "List",  
 Svy = "GBHE",  
 DatTim = Date,  
 FWSColonyID = sprintf("ME.%04d", Colony),  
 Obsr = "Gnd",  
 Seat = "Dirt",  
 Nest = Total,  
 Act = Active,  
 Comment = paste(GN, Notes, sep = ";")  
 ) %>%  
 select(Year,  
 Frame,  
 Svy,  
 DatTim,  
 FWSColonyID,  
 Obsr,  
 Seat,  
 Nest,  
 Act,  
 Comment)  
  
tGroundObs <- bind\_rows(GroundObs2015, GroundObs2023) %>%  
 mutate(OBJECTID = row\_number()) %>%  
 select(OBJECTID,  
 Year,  
 Frame,  
 Svy,  
 DatTim,  
 FWSColonyID,  
 Obsr,  
 Seat,  
 Nest,  
 Act,  
 Comment)  
  
tGroundObsSF <- left\_join(tGroundObs,  
 AllColonySF %>%  
 select(FWSColonyID, Str, PlotID, Lng, Lat, geometry),  
 by = "FWSColonyID") %>%  
 select(  
 OBJECTID,  
 Year,  
 FWSColonyID,  
 Frame,  
 Svy,  
 Str,  
 PlotID,  
 Lng,  
 Lat,  
 DatTim,  
 Obsr,  
 Seat,  
 Nest,  
 Act,  
 Comment,  
 geometry  
 )  
  
# These 21 ground observations are not in the colony list I have  
tGroundObsSF %>%  
 filter(is.na(Lng))

# A tibble: 21 × 16  
 OBJECTID Year FWSColonyID Frame Svy Str PlotID Lng Lat DatTim Obsr   
 <int> <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <chr> <chr>  
 1 14 2015 ME.0712 List GBHE <NA> <NA> NA NA 2015-0… Gnd   
 2 44 2023 ME.0895 List GBHE <NA> <NA> NA NA 2023-0… Gnd   
 3 49 2023 ME.0868 List GBHE <NA> <NA> NA NA 2023-0… Gnd   
 4 53 2023 ME.0908 List GBHE <NA> <NA> NA NA 2023-0… Gnd   
 5 57 2023 ME.0916 List GBHE <NA> <NA> NA NA 2023-1… Gnd   
 6 59 2023 ME.0902 List GBHE <NA> <NA> NA NA 2023-0… Gnd   
 7 66 2023 ME.0900 List GBHE <NA> <NA> NA NA 2023-0… Gnd   
 8 67 2023 ME.0899 List GBHE <NA> <NA> NA NA 2023-0… Gnd   
 9 69 2023 ME.0890 List GBHE <NA> <NA> NA NA 2023-0… Gnd   
10 74 2023 ME.0913 List GBHE <NA> <NA> NA NA 2023-0… Gnd   
# ℹ 11 more rows  
# ℹ 5 more variables: Seat <chr>, Nest <dbl>, Act <dbl>, Comment <chr>,  
# geometry <POINT [m]>

GroundObsSF <- tGroundObsSF %>%  
 filter(!is.na(Lng)) %>%  
 st\_sf()  
  
if (lWrite) {  
 st\_write(GroundObsSF,  
 here::here("GIS/GroundObsSF.kml"),  
 delete\_layer = TRUE)  
}

Writing layer `GroundObsSF' to data source   
 `/Users/motto/Profesional/GBHE/GIS/GroundObsSF.kml' using driver `KML'  
Writing 80 features with 15 fields and geometry type Point.

rm(GroundObs2015, GroundObs2023, tGroundObs, tGroundObsSF)

## Non-Spatial Ground Observations

GroundObs <- st\_drop\_geometry(GroundObsSF)  
if (lWrite) {  
 readr::write\_csv(GroundObs, here::here("data/GroundObs.csv"), na = "")  
}

## Save Data

Save all the tables we need for the estimation. The spatial data are already in gis and the non-spatial tables in data, so they can be accessed easily by locdata which is more general than base::data. Loading the save file makes data access easy as if this were an R project.

save(  
 CRS.WGS84,  
 CRS.Alpers,  
 CRS.UTM19N,  
 cYear,  
 nYear,  
 cFrame,  
 nFrame,  
 cDFStr,  
 nStr,  
 cYrStr,  
 nYrStr,  
 cFS,  
 nFS,  
 MEBoundarySF,  
 StrSF,  
 ColonySF,  
 Colony,  
 StrPlotSF,  
 List,  
 Plot,  
 GRTSSmp,  
 tblFlightInfo,  
 tblPlotDetails,  
 tblObservations,  
 RecObs,  
 GroundObsSF,  
 GroundObs,  
 file = here::here("output/GBHE.RData")  
)