

Quantifying Sampled Wetlands on BLM Land

Elin Binck

2022-03-25

Contents

1	Introduction	5
2	Hello bookdown	7
2.1	A section	7
3	Remove Duplicates	9
3.1	Load the data	9
3.2	Break up lpi_tall to investigate duplicates	10
3.3	Create objects for duplicates for each state	11
3.4	Remove plots with duplicates	12
4	Correct Species Codes	15
4.1	Load the rest of the data	15
4.2	Join lpi_tall2 with other species lists	16

Chapter 1

Introduction

This is a compilation of the code I have written thus far for my master's research, quantifying wetlands that have been sampled with BLM's Terrestrial Assessment, Inventory, and Monitoring (AIM) program. While the Terrestrial program aims to sample upland ecosystems, over 40,000 sites have been sampled, meaning even with minimal error, there are likely hundreds of sites that qualify as wetlands. In this bookdown, I use a number of criteria to identify "wetland" sites in the Terrestrial AIM database.

```
bookdown::render_book()
```

```
bookdown::serve_book()
```


Chapter 2

Hello bookdown

All chapters start with a first-level heading followed by your chapter title, like the line above. There should be only one first-level heading (#) per .Rmd file.

2.1 A section

All chapter sections start with a second-level (##) or higher heading followed by your section title, like the sections above and below here. You can have as many as you want within a chapter.

An unnumbered section

Chapters and sections are numbered by default. To un-number a heading, add a {.unnumbered} or the shorter {-} at the end of the heading, like in this section.

Chapter 3

Remove Duplicates

I was provided all of the AIM data by the BLM in a tall table for easy analysis. However, after beginning to work with the data, I realized there were a number of duplicate records for some reason. As a result, my first step was to remove data from all sites that had any duplicate records. Even if there was one duplicate reading in a site, the reasons seemed to be variable, and I determined it was more efficient to remove all of the data for those sites than to try to fix the issue. Additionally, I decided it was better to omit the data than try to alter it in a way that may not be accurate in relation to on the ground field conditions.

3.1 Load the data

```
lpi_tall<-readRDS("/Users/elinbinck/Documents/Grad_School/Thesis/R_project/Thesis_Research/data/A  
  rename(SpeciesCode = code)  
  
header <- readRDS("/Users/elinbinck/Documents/Grad_School/Thesis/R_project/Thesis_Research/data/A
```

3.1.1 Join lpi with corresponding states

```
PrimKeyState <- header[,c("PrimaryKey", "State")]  
  
lpi_tall2<-lpi_tall %>%  
  left_join(PrimKeyState) %>%  
  rename(SpeciesState = State) %>%  
  select(-STATE, -SAGEBRUSH_SPP)  
  
## Joining, by = "PrimaryKey"
```

3.2 Break up `lpi_tall` to investigate duplicates

```
#write a function to calculate the number of unique vs total rows for a given state
state_dups<-function(state) {
  dups<-lpi_tall2%>%
    filter(SpeciesState == state)
  print(n_distinct(dups))
  print(nrow(dups))
}

states<-unique(lpi_tall2$SpeciesState)
states_list<-setNames(vector("list", length(states)), states)

#use a for loop to quickly calculate them for each state
for (i in seq_along(states)){
  N<-state_dups(states[i])
  print(states[i])
  print(N)
}
```

```
## [1] 2329459
## [1] 2329459
## [1] "NV"
## [1] 2329459
## [1] 1291815
## [1] 1291856
## [1] "CO"
## [1] 1291856
## [1] 84073
## [1] 84073
## [1] "AK"
## [1] 84073
## [1] 220373
## [1] 220373
## [1] "AZ"
## [1] 220373
## [1] 622846
## [1] 622846
## [1] "CA"
## [1] 622846
## [1] 82541
## [1] 82546
## [1] "SD"
## [1] 82546
## [1] 1035623
## [1] 1035628
```

```
## [1] "ID"
## [1] 1035628
## [1] 1187917
## [1] 1187917
## [1] "WY"
## [1] 1187917
## [1] 10844
## [1] 10844
## [1] "ND"
## [1] 10844
## [1] 984421
## [1] 984421
## [1] "UT"
## [1] 984421
## [1] 804749
## [1] 807959
## [1] "MT"
## [1] 807959
## [1] 1427307
## [1] 1427307
## [1] "OR"
## [1] 1427307
## [1] 361284
## [1] 361284
## [1] "WA"
## [1] 361284
## [1] 0
## [1] 0
## [1] NA
## [1] 0
## [1] 785171
## [1] 785171
## [1] "NM"
## [1] 785171
```

3.3 Create objects for duplicates for each state

3.3.0.1 South Dakota

```
SDrows<-lpi_tall2%>%
  filter(SpeciesState == "SD")

SDdups<- SDrows[duplicated(SDrows),]
```

3.3.0.2 Montana

```
MTrows<-lpi_tall2%>%  
  filter(SpeciesState == "MT")  
  
MTdups<- MTrows[duplicated(MTrows),]
```

3.3.0.3 Colorado

```
COrows<-lpi_tall2%>%  
  filter(SpeciesState == "CO")  
  
COdups<- COrows[duplicated(COrows),]
```

3.3.0.4 Idaho

```
IDrows<-lpi_tall2%>%  
  filter(SpeciesState == "ID")  
  
IDDups<- IDrows[duplicated(IDrows),]
```

3.4 Remove plots with duplicates

3.4.1 Create a df with all the PrimaryKeys from each dup file for each state

```
uniqueSDdups<-SDdups %>%  
  distinct(PrimaryKey)  
  
uniqueMTdups<-MTdups %>%  
  distinct(PrimaryKey)  
  
uniqueCOdups<-COdups %>%  
  distinct(PrimaryKey)  
  
uniqueIDDups<-IDDups %>%  
  distinct(PrimaryKey)  
  
dupPrimaryKeys<-rbind(uniqueSDdups, uniqueMTdups, uniqueCOdups, uniqueIDDups)  
  
dupPrimaryKeys<-as.vector(dupPrimaryKeys$PrimaryKey)
```

3.4.2 Remove all plots that have any duplicate values

```
for (i in seq_along(dupPrimaryKeys)){  
  lpi_tall2<-lpi_tall2 %>%  
    filter(PrimaryKey !=dupPrimaryKeys[i])  
}  
  
#confirm that that worked and removed all entries with those PrimaryKeys  
#also check to see how many plots were removed - it looks like less than 100 which is good  
  
n_distinct(lpi_tall$PrimaryKey)  
  
## [1] 36314  
  
n_distinct(lpi_tall2$PrimaryKey)  
  
## [1] 36232
```

3.4.3 Check for more duplicates

```
#No more duplicates exist!  
n_distinct(lpi_tall2)  
  
## [1] 11229988
```


Chapter 4

Correct Species Codes

```
library(tidyverse)
library(tidyr)
library(readr)

knitr::opts_chunk$set(echo = TRUE)
```

4.1 Load the rest of the data

4.1.0.1 LPI data (dups removed)

```
lpi_tall2<- read.csv("/Users/elinbinck/Documents/Grad_School/Thesis/R_project/Thesis_Research/dat
```

4.1.0.2 State Species List

This is the data that has the “correct” codes to replace incorrect ones.

```
StateSpecies<-read.csv("/Users/elinbinck/Documents/Grad_School/Thesis/R_project/Thesis_Research/c
  select(SpeciesCode,
         ScientificName,
         UpdatedSpeciesCode,
         SpeciesState)

#change some of the incorrectly formatted data

StateSpecies["SpeciesCode"][StateSpecies["SpeciesCode"] == "7-Feb"] <- "FEBR7"

StateSpecies["SpeciesCode"][StateSpecies["SpeciesCode"] == "2-Feb"] <- "FEBR2"

StateSpecies["SpeciesCode"][StateSpecies["SpeciesCode"] == "2-Mar"] <- "MARCH2"
```

4.1.0.2 Wetland AIM Master Species List

```
WetAIMmasterlist <- read.csv("/Users/elinbinck/Documents/Grad_School/Thesis/R_project/Thesis_R/
  select(Symbol,
          WMVC_WetStatus,
          AW_WetStatus,
          GP_WetStatus,
          Scientific.Name)
```

4.1.0.4 USDA Plant List

```
USDAlist <- read.csv("/Users/elinbinck/Documents/Grad_School/Thesis/R_project/Thesis_R/

USDAlist["Symbol"][USDAlist["Symbol"] == "7-Feb"] <- "FEBR7"

USDAlist["Symbol"][USDAlist["Symbol"] == "5-Feb"] <- "FEBR5"

USDAlist["Symbol"][USDAlist["Symbol"] == "2-Feb"] <- "FEBR2"

USDAlist["Symbol"][USDAlist["Symbol"] == "2-Mar"] <- "MARCH2"

USDAlist["Symbol"][USDAlist["Symbol"] == "Dec-70"] <- "DECE70"

USDAlist["Symbol"][USDAlist["Symbol"] == "5-Jun"] <- "JUNE5"

USDAlist["Symbol"][USDAlist["Symbol"] == "2-Nov"] <- "NOVE2"
```

4.2 Join lpi_tall2 with other species lists

4.2.1 Join with USDA plant list

#First, create a new column with EITHER the Symbol or the Synonym to prevent duplicati

```
USDAlist_oneCode<- USDAlist %>%
  mutate(SpeciesCode =
    if_else(Synonym.Symbol == "",Symbol, Synonym.Symbol)) %>%
  select(-Synonym.Symbol)

lpi_USDA <- lpi_tall2 %>%
  left_join(USDAlist_oneCode, by = "SpeciesCode")
```


4.2.2 Join with StateSpecies

4.2.2.1 Prep StateSpecies to join

Here I need to remove all duplicate combinations of state and species code. To do so, I used anti join to find all of the records that are redundant because they match up with codes from the USDA list. This also conveniently removed all duplicates, so when I join with the lpi data, this should give me a species name and possibly an “updated code” for any listings that did not already match with the USDA list.

#Investigate how many of the codes from the StateSpecies list show up in the USDA list w/ synonyms

```
USDA_StateSpecies <- USDAlist_oneCode %>%
  inner_join(StateSpecies, by = "SpeciesCode")
```

#there are only 2,500ish records that don't match. Let's look at those:

```
antiUSDA_StateSpecies <-StateSpecies %>%
  anti_join(USDAlist_oneCode, by = "SpeciesCode")
```

#It looks like there are no duplicates anymore also! So now I will join this to the lpi data:

```
unique(antiUSDA_StateSpecies$UpdatedSpeciesCode)
```

```
## [1] ""      "JUALN"  "SASP91" "SPHAG2" "ASROM"  "2FUNGI" "CAMI10"
## [8] "CEIS60" "CILA70" "CLST7"  "DRINC"  "HUPER"  "OROB"    "PLAL8"
## [15] "SPC070" "TEMN70" "AMMEI2" "GALIU"  "RIMO2"  "ACTH7"   "LILE3"
## [22] "<Null>"  "VEW02"  "RIBES"  "MAGR2"  "SATR12" "ANPA4"   "ORST2"
## [29] "SACA52" "FRAL2"  "ALTE"   "THAR5"  "COLI2"  "HECO26"  "ALPR3"
## [36] "JUHO2"  "CRPO5"  "BOER4"  "SALIX"  "SCLER10" "SEDUM"   "ERIGE2"
## [43] "GIIN2"  "HEPA11" "PUTR2"  "RUMEX"  "EUPHO"  "PENST"   "POLYG4"
## [50] "ARFR4"  "ARTRW8" "ANST2"  "ARTRT"  "NOCU"   "AMAL2"   "ACMI2"
## [57] "AGGL"   "ALLIU"  "AMBRO"  "ANTEN"  "ARABI2" "CRAC2"   "DRABA"
## [64] "SPORO"  "TAOF"   "TRDU"   "TRIFO"
```

4.2.2.2 Join the data

*#Join the lpi data that was joined with the USDA codes with the state species codes that are not
#so, this should produce the same number of records as lpi_tall2/lpi_USDA - and it does!*

#Add the column with the "correct code," so basically any codes that differ between the UpdatedSp

```
lpi_USDA_StateSpecies<-lpi_USDA %>%
  left_join(antiUSDA_StateSpecies) %>%
  mutate(CorrectSpeciesCode =
    if_else(UpdatedSpeciesCode == "", SpeciesCode, UpdatedSpeciesCode, missing = SpeciesCode)
```

```
## Joining, by = c("SpeciesCode", "SpeciesState")
```

4.2.3 Join with USDA list again

This time I will be joining by scientific name, to try to catch any records where codes were just not right for some reason, but scientific names were.

#I shouldn't need to do this with USDAlist_oneCode, since the Wetland AIM master list

```
USDAlist2<- USDAlist %>%
  rename("Symbol2" = "Symbol",
         "Sci.Name"="Scientific.Name.with.Author") %>%
  select(-"Common.Name",
         - "Family",
         - "Synonym.Symbol")
```

#Now, join the USDA list again, joining with its scientific name column (Sci.Name) to

```
lpi_USDA2_StateSpecies<- lpi_USDA_StateSpecies %>%
  left_join(USDAlist2, by = c("ScientificName" = "Sci.Name")) %>%
  mutate(CorrectSpeciesCode2 =
         if_else(!is.na(Symbol2), Symbol2, CorrectSpeciesCode)) %>%
  select(-Symbol2,
         -Symbol,
         -CorrectSpeciesCode,
         -UpdatedSpeciesCode,
         -ScientificName)
```

#No duplicates, which is a great sign!

#this adds a correct symbol column (Symbol2) directly from the USDA list that should m