Fric et al. critiques: data curation

Elise Larsen & Vaughn Shirey

Updated 3-Dec-2020; Begun Feb-2020

Here we explore the occurrence data from Fric et al. (2020)

This gives a detailed account of some data curation issues we observed in the Fric et al. data and curation.

```
rm(list=ls())
# load libraries
library(tidyverse)
library(readxl)
library(ggplot2)
library(ggExtra)
library(gridExtra)
library(lubridate)
# install.packages("viridis")
library(viridis)
```

```
## Warning: package 'viridis' was built under R version 4.0.3
```

Data Input

```r
#raw data
all.data <- readLines("fric_supplements/data.csv")

#identify header rows
all.header.rows<-grep("decimalLongitude", all.data)

#check headers for consistency
uniqueheaders<-unique(all.data[all.header.rows])

# 2 versions! -> Get row numbers for "header 1"
header.rows1<-grep(uniqueheaders[1], all.data)
#Get row numbers for "header 2"
header.rows2<-setdiff(all.header.rows, header.rows1)

#Create row identifiers:
#0 is a header row, 1 is format 1 data, 2 is format 2 data
j<-rep(0,length(all.data))
for (i in all.header.rows) {
  #set index to the next header if it's not the last header; otherwise set to end of datafile + 1
  if(i<max(all.header.rows)) {
    next_index<-min(all.header.rows[all.header.rows>i])
  }else { next_index<-length(all.data)+1 }

  #for data between header rows, set row index
  j[(i+1):(next_index-1)]<-ifelse(i%in%header.rows1,1,2)
}

#need to add a row index to the header text for new data files
newheader1<-paste('"row.index\",' ,uniqueheaders[1], sep="")
newheader2<-paste('"row.index\",' ,uniqueheaders[2], sep="")

#write data file
formatteddatafile1<-file("data/fric_data_header_1.txt")
writeLines(c(newheader1,all.data[which(j==1)]), formatteddatafile1)
close(formatteddatafile1)

formatteddatafile2<-file("data/fric_data_header_2.txt")
writeLines(c(newheader2,all.data[which(j==2)]), formatteddatafile2)
close(formatteddatafile2)
rm(list=ls())
```

```
#read back in the formatted data
data1<-read_csv("data/fric_data_header_1.txt")
```

```
## Parsed with column specification:
## cols(
##   row.index = col_double(),
##   name = col_character(),
##   decimalLongitude = col_double(),
##   decimalLatitude = col_double(),
##   year = col_double(),
##   month = col_double(),
##   country = col_character(),
##   day = col_double(),
##   SuccDay = col_double(),
##   rndLat = col_double(),
##   alt = col_double()
## )
```

```
data2<-read_csv("data/fric_data_header_2.txt")
```

```
## Parsed with column specification:
## cols(
##   row.index = col_double(),
##   name = col_character(),
##   decimalLongitude = col_double(),
##   decimalLatitude = col_double(),
##   year = col_double(),
##   month = col_double(),
##   day = col_double(),
##   country = col_character(),
##   SuccDay = col_double(),
##   rndLat = col_double(),
##   alt = col_double()
## )
```

```
paste( nrow(data1), "records in format 1;", nrow(data2), "records in format 2")
```

```
## [1] "49243 records in format 1; 233201 records in format 2"
```

```
alldata<-bind_rows(data1,data2)
rm(data1,data2)
```

New code 11/25/2020: day of year reconciliation

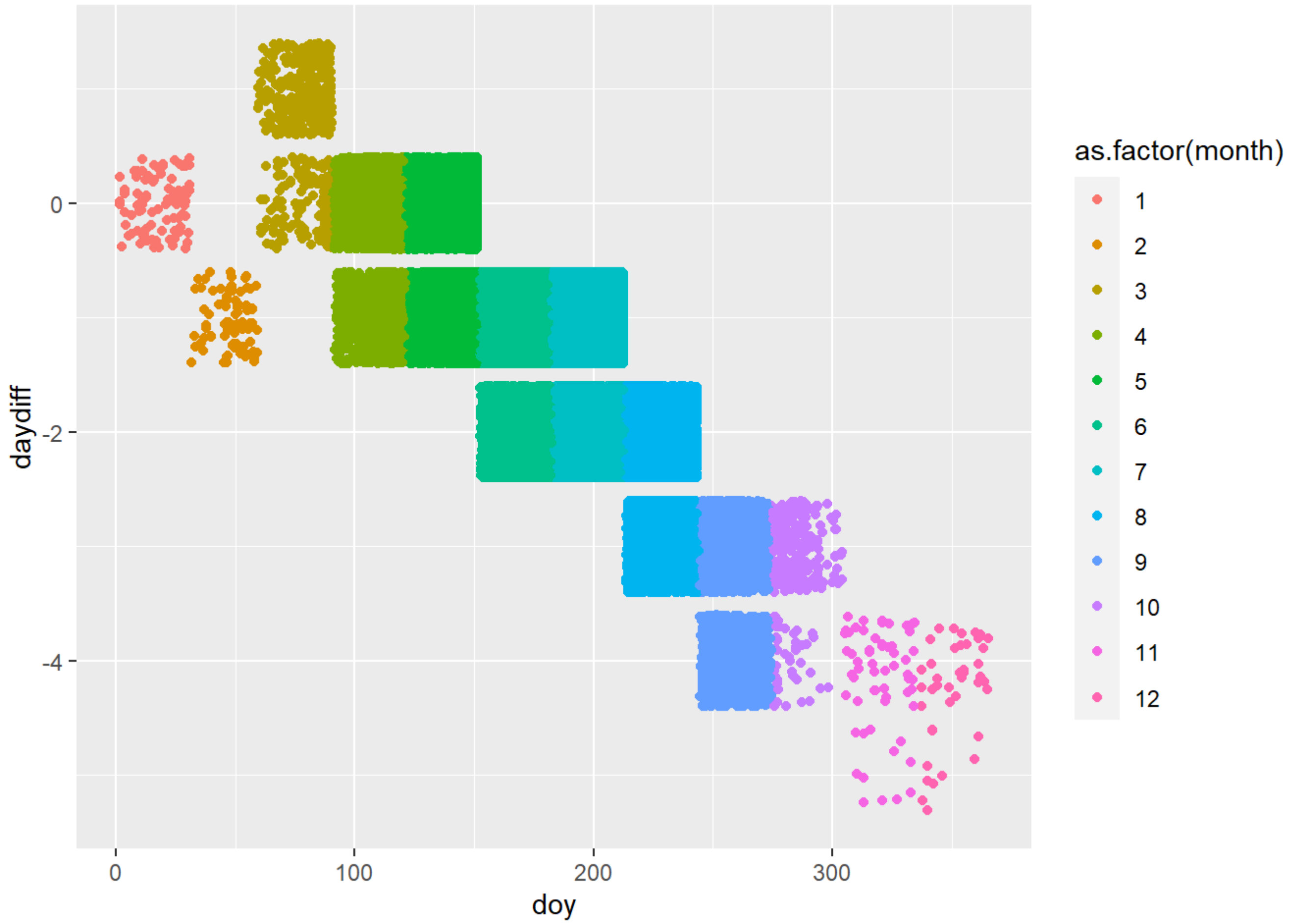
Until today, we had assumed that the “SuccDay” values were a consistent index for day of year. However, we had not documented our initial spot-checking of altitudes. While identifying GBIF records for documented sopt-checking, we found some inconsistencies in the SuccDay value. Here we identify how “SuccDay” was calculated.

```
#DOES SUCCDAY MATCH DOY?

alldata<-na.omit(alldata)
checkdays<-alldata %>%
  mutate(doy=yday(as.Date(paste(year,month,day, sep="-"), "%Y-%m-%d")),
         daydiff=SuccDay-doy, fricday=(month-1)*30+day) %>%
  select(name,day,month,year,SuccDay,doy,daydiff,fricday)
#summary(checkdays)
table(checkdays$fricday-checkdays$SuccDay)
```

```
##
##      0
## 282386
```

```
ggplot(data=checkdays, aes(y=daydiff, x=doy, color=as.factor(month))) + geom_jitter()
```




```
#we'd prefer to use calendar day
alldata<-alldata %>%mutate(doy=yday(as.Date(paste(year,month,day, sep="-"),"%Y-%m-%d")))
```

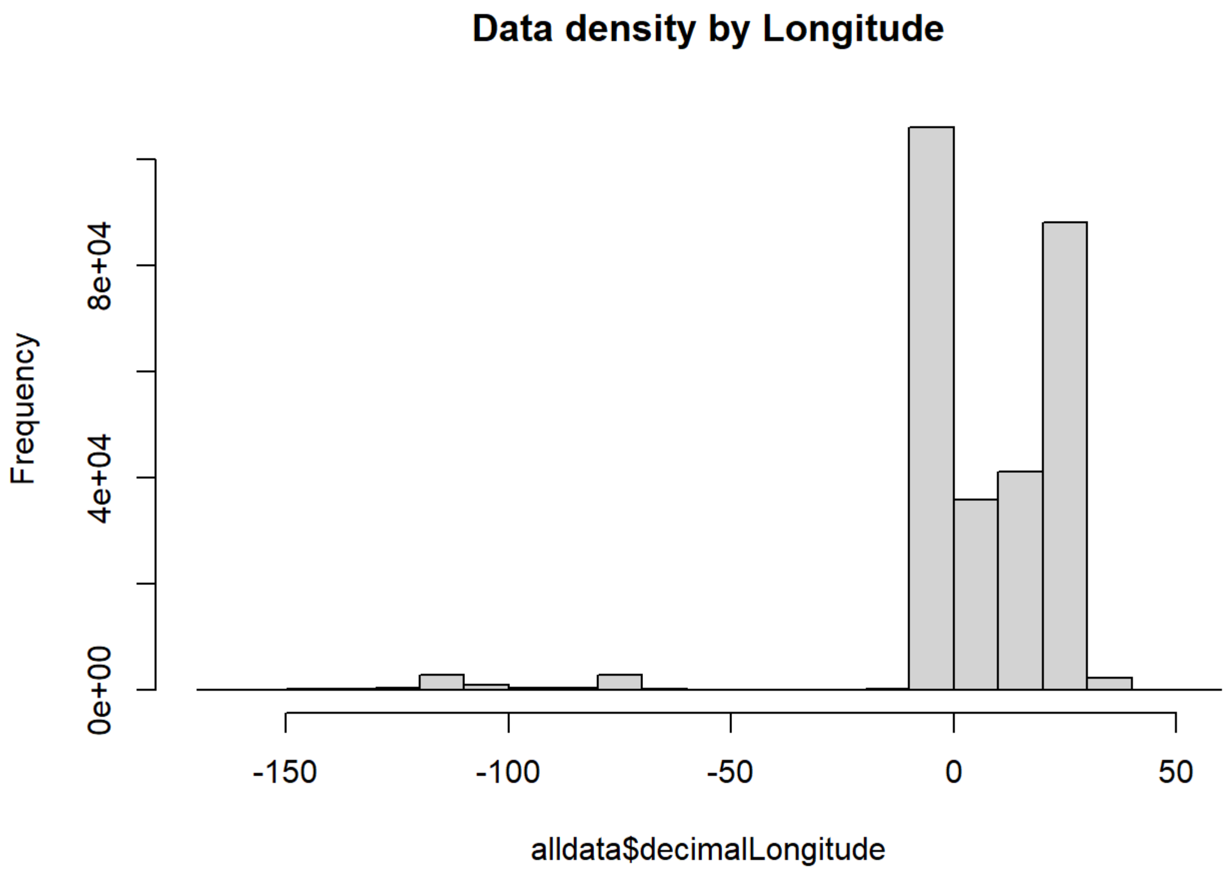
Data exploration 1

Now we assign region, reconcile names that don't match between the data file and results files provided in the original supplement, and filter the Fric dataset to remove first day of the month records to obtain the dataset used in Fric et al.

```
summary(alldata)
```

```
##    row.index          name           decimalLongitude   decimalLatitude
##  Min.   :    1   Length:282386      Min.   :-162.559   Min.   : 5.787
##  1st Qu.: 2369   Class :character   1st Qu.:  -2.782   1st Qu.:52.784
##  Median : 7008   Mode  :character   Median :   9.398   Median :55.628
##  Mean   :14819                      Mean   :   6.317   Mean   :56.271
##  3rd Qu.:20216                      3rd Qu.:  23.573   3rd Qu.:60.624
##  Max.   :85273                      Max.   :  59.333   Max.   :71.216
##       year          month          country               day
##  Min.   :1616   Min.   : 1.000   Length:282386      Min.   : 1.00
##  1st Qu.:1992   1st Qu.: 6.000   Class :character   1st Qu.: 9.00
##  Median :2002   Median : 7.000   Mode  :character   Median :16.00
##  Mean   :1996   Mean   : 6.517                      Mean   :16.15
##  3rd Qu.:2009   3rd Qu.: 7.000                      3rd Qu.:24.00
##  Max.   :2015   Max.   :12.000                      Max.   :31.00
##     SuccDay          rndLat           alt                 doy
##  Min.   :  2.0   Min.   : 6.00   Min.   :-2666.74   Min.   :  2.0
##  1st Qu.:163.0   1st Qu.:53.00   1st Qu.:   23.25   1st Qu.:165.0
##  Median :186.0   Median :56.00   Median :   64.33   Median :187.0
##  Mean   :181.7   Mean   :56.21   Mean   :  114.22   Mean   :182.9
##  3rd Qu.:202.0   3rd Qu.:61.00   3rd Qu.:  111.09   3rd Qu.:203.0
##  Max.   :361.0   Max.   :71.00   Max.   : 4305.17   Max.   :365.0
```

```
##Fric et al identifies datasets by region (N. America, Europe), but the data file does not include this information. We lab
el data by region using longitude:
## visualize data density by longitude
hist(alldata$decimalLongitude, main="Data density by Longitude")
```




```
#We label everything East of -40 as Europe, the rest as N. America
alldata<-alldata %>%
  mutate(region=ifelse(decimalLongitude>=(-40),"Europe","N. America"))

#We expect 100 species names, based on the manuscript.
length(unique(alldata$name))
```

```
## [1] 108
```

```
#What are the names in the dataset?
datanames<-sort(unique(alldata$name))
data.gs<-strsplit(datanames," ")
data.names <-as.data.frame(cbind(datanames,matrix(unlist(strsplit(datanames," ")),ncol=2,byrow=T)))
names(data.names)<-c("data.name","genus","spep")

#Which of these names shows up in the results?
result.names<-unique(na.omit(read_excel("fric_supplements/ele13419-sup-0003-tables2.xlsx", sheet="~latitude", range="A3:A11
3"))$Species)
resultnames<-(strsplit(result.names, " "))
result.names<-tibble(name=character(),genus=character(),spep=character())
for(i in 1:length(resultnames)) {
  genus<-paste(resultnames[[i]][1])
  spep<-paste(resultnames[[i]][2])
  name<-paste(genus,spep,sep=" ")
  temp.names<-tibble(name=as.character(name),genus=as.character(genus),spep=as.character(spep))
  result.names<-bind_rows(result.names,temp.names)
}
#which names match
which(data.names$data.name%in%result.names$name)
```

```
##  [1]   1   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
## [20]  21  22  23  25  26  27  28  29  31  32  33  34  35  36  37  38  39  40  41
## [39]  42  44  45  46  47  48  49  50  51  53  54  55  57  58  60  61  62  64  65
## [58]  67  68  69  70  72  73  74  75  76  78  79  80  81  82  83  84  85  86  87
## [77]  88  89  90  91  92  93  94  97  98 100 101 102 103 105 106 107
```

```r
#not matched
names1<-data.names[which(!data.names$data.name%in%result.names$name),]
names2<-result.names[which(!result.names$name%in%data.names$data.name),]
names1$result.name<-NA

#First let's try fuzzy matching
for (i in 1:nrow(names1)) {
    if(length(agrep(names1$data.name[i], names2$name, ignore.case = TRUE, value = TRUE, max.distance = 0.1))>0) {
      names1$result.name[i]<-agrep(names1$data.name[i], names2$name, ignore.case = TRUE, value = TRUE, max.distance = 0.2)
    }
}
#names1 #looks good

#now let's match on specific epithets
which(names2$spep%in%names1$spep[is.na(names1$result.name)])
```

```
## [1] 2 5 7 8
```

```r
names1$result.name[which(names1$spep%in%names2$spep)]<-names2$name[match(names1$spep[which(names1$spep%in%names2$spep)],name
s2$spep)]
names1 #looks good
```

```
##                data.name        genus       spep         result.name
## 2      Agriades optilete     Agriades   optilete Vacciniina optilete
## 24        Boloria selene      Boloria     selene                <NA>
## 30     Callophrys polios   Callophrys     polios    Callophrys polia
## 43       Cupido amyntula       Cupido   amyntula                <NA>
## 52         Erynnis tages      Erynnis      tages                <NA>
## 56    Euphydryas aurinia   Euphydryas    aurinia                <NA>
## 59     Fabriciana adippe   Fabriciana     adippe     Argynnis adippe
## 63  Incisalia augustinus    Incisalia augustinus                <NA>
## 66        Lethe eurydice        Lethe   eurydice  Satyrodes eurydice
## 71        Lycaeides idas    Lycaeides       idas                <NA>
## 77       Maculinea arion    Maculinea      arion                <NA>
## 95  Phyciodes campestris    Phyciodes campestris                <NA>
## 96      Phyciodes tharos    Phyciodes     tharos                <NA>
## 99    Plebejus saepiolus     Plebejus  saepiolus  Icaricia saepiolus
## 104  Scolitantides orion Scolitantides     orion                <NA>
## 108   Thymelicus lineola   Thymelicus    lineola Thymelicus lineolus
```

```
print("The species names in the results that are not present in the data are:")
```

```
## [1] "The species names in the results that are not present in the data are:"
```

```
names2$name[!names2$name%in%names1$result.name]
```

```
## [1] "Phyciodes cocyta"    "Phyciodes pratensis"
```

```
#GBIF considers Phyciodes cocyta a synonym of Phyciodes tharos (https://www.gbif.org/species/1918971)
#GBIF considers Phyciodes pratensis a synonym of Phyciodes campestris (https://www.gbif.org/fr/species/1918960)
names1$result.name[names1$data.name=="Phyciodes tharos"]<-"Phyciodes cocyta"
names1$result.name[names1$data.name=="Phyciodes campestris"]<-"Phyciodes pratensis"

#Now we can match data specific epithets to other results specific epithets
shared.spep<-result.names$spep[which(result.names$spep%in%names1$spep[is.na(names1$result.name)])]

names1$result.name[which(names1$spep%in%shared.spep)]<-result.names$name[which(result.names$spep%in%shared.spep)]

names1
```

```
##                data.name        genus       spep           result.name
## 2      Agriades optilete     Agriades   optilete   Vacciniina optilete
## 24        Boloria selene      Boloria     selene                  <NA>
## 30     Callophrys polios   Callophrys     polios      Callophrys polia
## 43       Cupido amyntula       Cupido   amyntula                  <NA>
## 52         Erynnis tages      Erynnis      tages                  <NA>
## 56    Euphydryas aurinia   Euphydryas    aurinia                  <NA>
## 59     Fabriciana adippe   Fabriciana     adippe       Argynnis adippe
## 63  Incisalia augustinus    Incisalia augustinus Callophrys augustinus
## 66        Lethe eurydice        Lethe   eurydice    Satyrodes eurydice
## 71        Lycaeides idas    Lycaeides       idas          Plebejus idas
## 77       Maculinea arion    Maculinea      arion       Phengaris arion
## 95  Phyciodes campestris    Phyciodes campestris   Phyciodes pratensis
## 96      Phyciodes tharos    Phyciodes     tharos      Phyciodes cocyta
## 99    Plebejus saepiolus     Plebejus  saepiolus    Icaricia saepiolus
## 104  Scolitantides orion Scolitantides     orion                  <NA>
## 108   Thymelicus lineola   Thymelicus    lineola   Thymelicus lineolus
```

```
#It is unclear if any other species names in the data contribute to the results.
#Euphydryas aurinia is removed by Fric et al.
names1$result.name[names1$data.name=="Euphydryas aurinia"]<-""
#This leaves four species names, which we will not address.

write.csv(names1, file="data/name_changes.csv")
# this file can now be used for correcting names in the main file

for(namei in 1:nrow(names1)) {
  alldata$name[alldata$name==names1$data.name[namei]]<-names1$result.name[namei]
}

fricdata<-alldata %>% filter(alldata$name %in% result.names$name)
rm(name_changes, resultnames, result.names, data.names, namei, names_1, names_2, nmatch)
```

```
## Warning in rm(name_changes, resultnames, result.names, data.names, namei, :
## object 'name_changes' not found
```

```
## Warning in rm(name_changes, resultnames, result.names, data.names, namei, :
## object 'names_1' not found
```

```
## Warning in rm(name_changes, resultnames, result.names, data.names, namei, :
## object 'names_2' not found
```

```
## Warning in rm(name_changes, resultnames, result.names, data.names, namei, :
## object 'nmatch' not found
```

```
#Fric et al removed all 1st of month observations.
fricdata<-filter(fricdata, day!=1)

summary(fricdata)
```

```
##    row.index         name           decimalLongitude   decimalLatitude
##  Min.   :    1   Length:257919      Min.   :-162.559   Min.   : 5.787
##  1st Qu.: 2343   Class :character   1st Qu.:  -2.676   1st Qu.:52.711
##  Median : 7277   Mode  :character   Median :   9.551   Median :55.640
##  Mean   :15627                      Mean   :   6.548   Mean   :56.300
##  3rd Qu.:22572                      3rd Qu.:  23.672   3rd Qu.:60.650
##  Max.   :85273                      Max.   :  59.333   Max.   :71.216
##       year          month           country               day
##  Min.   :1616   Min.   : 1.000   Length:257919      Min.   : 2.00
##  1st Qu.:1992   1st Qu.: 6.000   Class :character   1st Qu.: 9.00
##  Median :2002   Median : 7.000   Mode  :character   Median :16.00
##  Mean   :1996   Mean   : 6.519                      Mean   :16.19
##  3rd Qu.:2009   3rd Qu.: 7.000                      3rd Qu.:24.00
##  Max.   :2015   Max.   :12.000                      Max.   :31.00
##     SuccDay          rndLat            alt                 doy
##  Min.   :  2.0   Min.   : 6.00   Min.   :-2666.74   Min.   :  2
##  1st Qu.:165.0   1st Qu.:53.00   1st Qu.:   23.25   1st Qu.:166
##  Median :187.0   Median :56.00   Median :   64.24   Median :188
##  Mean   :181.8   Mean   :56.24   Mean   :  114.23   Mean   :183
##  3rd Qu.:202.0   3rd Qu.:61.00   3rd Qu.:  109.48   3rd Qu.:203
##  Max.   :361.0   Max.   :71.00   Max.   : 4305.17   Max.   :365
##     region
##  Length:257919
##  Class :character
##  Mode  :character
##
##
##
```

Data exploration: altitude (elevation)

(We defer to the Fric et al use of “altitude” for clarity)

Early on in data exploration we were concerned with the range of altitude values in the data. One aspect of our data exploration for altitude involved examining outliers and spot-checking specific occurrence records in GBIF, which were either below 0m or in the top quartile of altitudes. Looking at these records led us to understand that

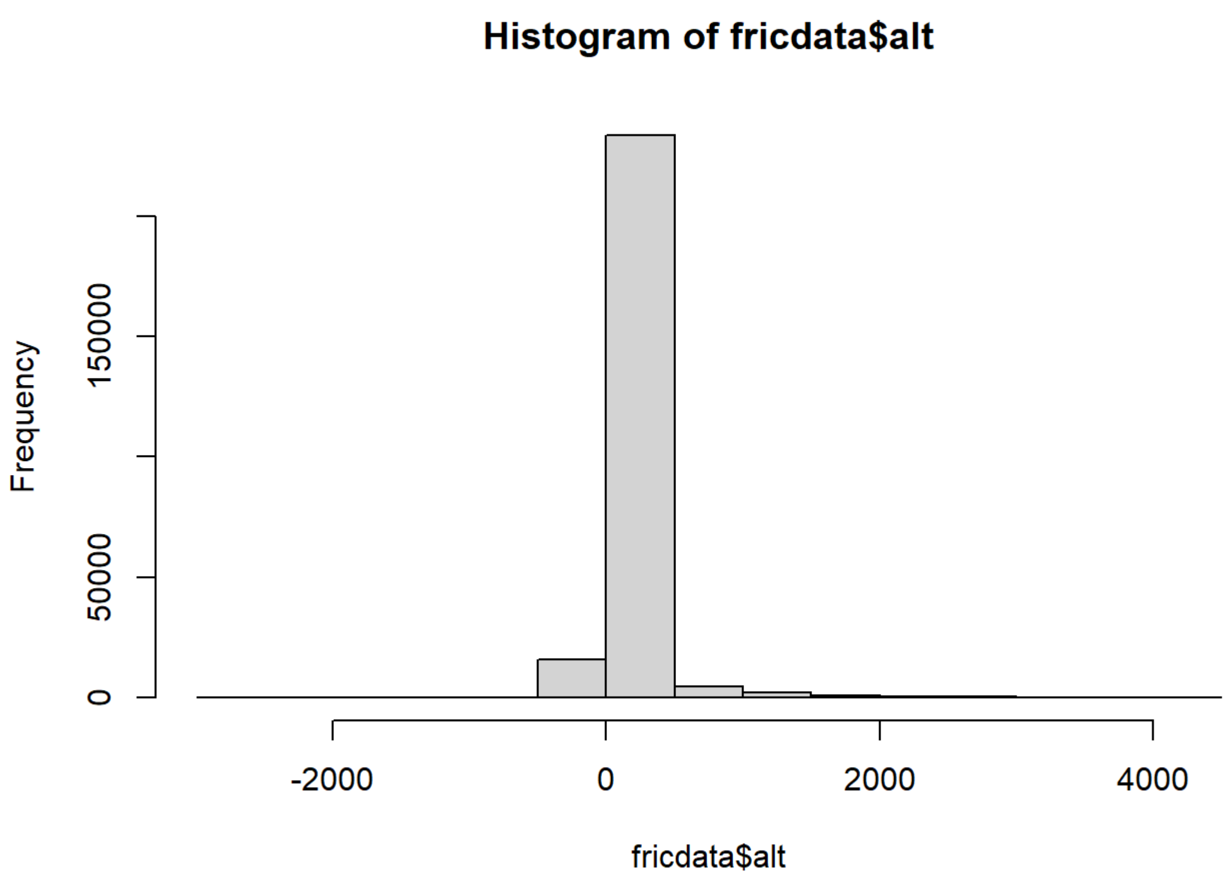
- 1. GIS coordinates had often been assigned by placename, or were otherwise inaccurate, and
  2. 2. altitudes obtained by using the Google API to extract altitude for coordinates did not provide reliable altitudes for the underlying occurrences.

Here we examine broad patterns and specific outlier cases.

```
#basic range & frequency in data
summary(fricdata$alt)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2666.74    23.25    64.24   114.23   109.48  4305.17
```

```
hist(fricdata$alt)
```



```
#how many records below 0?
print(paste(nrow(filter(fricdata,alt<0)),"records below sea level represent", round(nrow(filter(fricdata,alt<0))/nrow(fricda
ta)*100,2),"percent of all ocurrence records. We examined lat/long for many of these records and all examined locations were
in bodies of water.",sep=" "))
```

```
## [1] "9974 records below sea level represent 3.87 percent of all ocurrence records. We examined lat/long for many of these
records and all examined locations were in bodies of water."
```

```
#how many records are above 500m?
print(paste(nrow(filter(fricdata,alt>500)),"records above 500m represent", round(nrow(filter(fricdata,alt>500))/nrow(fricdat
a)*100,2),"percent of all ocurrence records. We examined lat/long and location for a small subset of high altitude records a
nd found vague place names had been used for geolocation.",sep=" "))
```

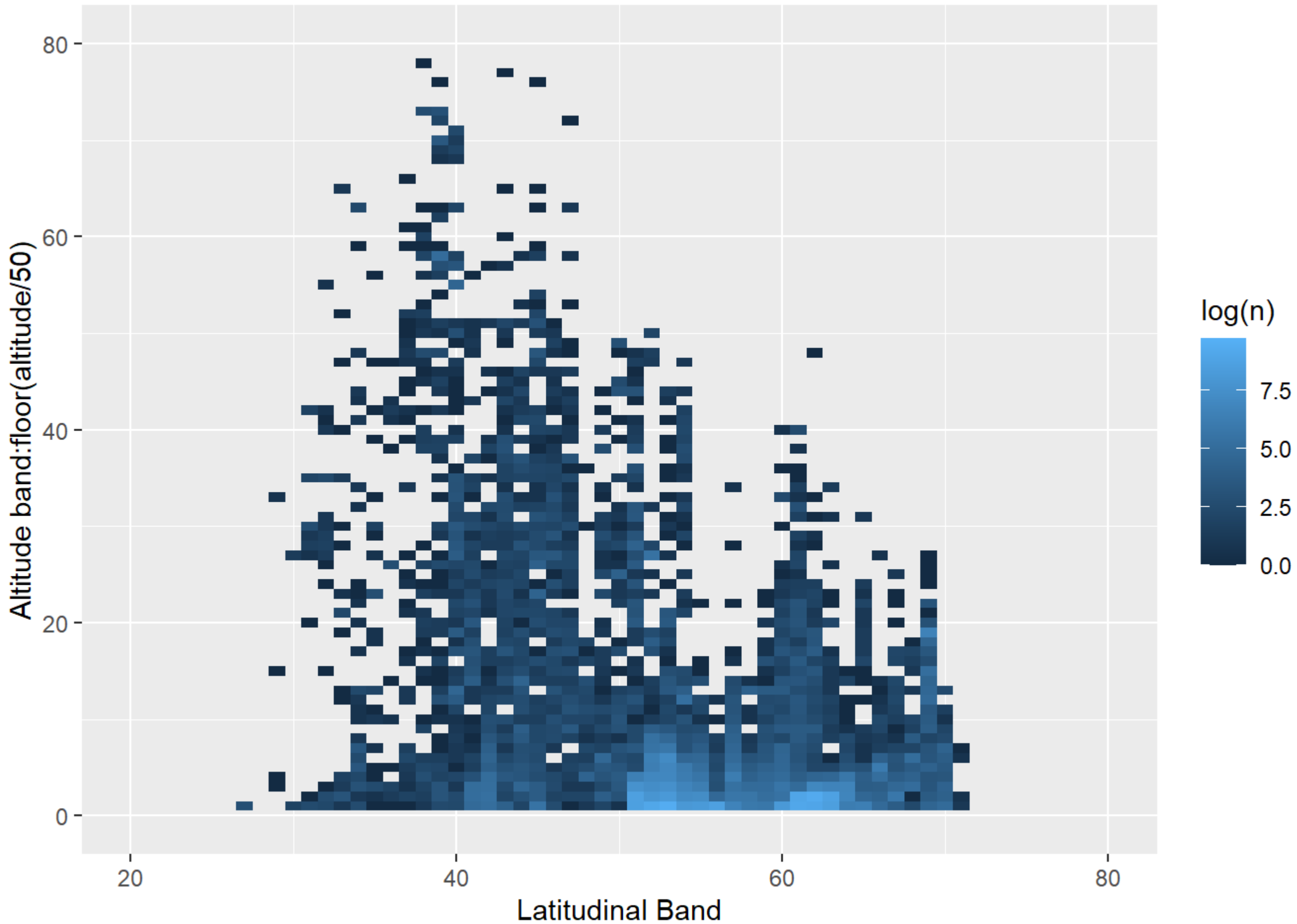
```
## [1] "8620 records above 500m represent 3.34 percent of all ocurrence records. We examined lat/long and location for a sma
ll subset of high altitude records and found vague place names had been used for geolocation."
```

```
#How many in the 0-500m range
print(paste(nrow(filter(fricdata,between(alt,0,500))),"records  within 0-500m represent", round(nrow(filter(fricdata,between
(alt,0,500)))/nrow(fricdata)*100,2),"percent of all ocurrence records. For reanalysis, we can constrain data to these record
s with minimal impact on data density. ",sep=" "))
```

```
## [1] "239325 records  within 0-500m represent 92.79 percent of all ocurrence records. For reanalysis, we can constrain dat
a to these records with minimal impact on data density. "
```

```
altdata<-fricdata %>% mutate(alt.grp=floor(alt/50))  %>%
  group_by(alt.grp, rndLat) %>% tally()
# Heatmap
ggplot(altdata, aes(rndLat, alt.grp, fill= log(n))) +
  geom_tile() + labs(x="Latitudinal Band", y="Altitude band:floor(altitude/50)") +
  xlim(20,80) + ylim(0,80)
```

```
## Warning: Removed 37 rows containing missing values (geom_tile).
```



Outliers appear to be a problem with altitude. Reviewing GBIF records, this appears to be primarily due to the assumption by Fric et al. that the GIS coordinates are precise and that the google API would provide accurate and reliable altitude metrics. Based on the records we spot-checked, when GBIF includes elevation, the values do not match those used in the analysis.

A few examples including the lowest and highest alt records, as well as some additional records selected arbitrarily from the extreme quantiles of altitude:

- 1953 Anthocharis sara record (row.index 166; altitude -525.96m) is from https://www.gbif.org/occurrence/1039154960 (https://www.gbif.org/occurrence/1039154960); geocoordinates were assigned via vertnet in 2015. These coordinates are located in the ocean. The GBIF record traces to https://collections.peabody.yale.edu/search/Record/YPM-ENT-729028 (https://collections.peabody.yale.edu/search/Record/YPM-ENT-729028) which simply gives a locality of “North America; USA; California; Los Angeles County; Rolling Hills”. Rolling Hills, CA is ~10km east of the given lat/long according to our estimation using googlemaps.
- 1991 Parnassius smintheus record (row.index 38; altitude 4048m) is from https://www.gbif.org/occurrence/1039027733 (https://www.gbif.org/occurrence/1039027733) (which gives elevation of 3810m). The GBIF record traces to

https://collections.peabody.yale.edu/search/Record/YPM-ENT-430824 (https://collections.peabody.yale.edu/search/Record/YPM-ENT-430824) which gives a locality of "North America; USA; Colorado; Summit County; Loveland Pass, 3810 m". The actual collection altitude is provided by the source, and is different than that used in the analysis.

- 1918 Euphydryas chalcedona record (row.index 139; altitude 4305m) is the highest record in the data. It's from https://www.gbif.org/occurrence/1039181223 (https://www.gbif.org/occurrence/1039181223). The GBIF record traces to https://collections.peabody.yale.edu/search/Record/YPM-ENT-819202 (https://collections.peabody.yale.edu/search/Record/YPM-ENT-819202) which gives a locality of "North America; USA; California; Siskiyou County; Mount Shasta" There is a city named Mount Shasta, CA that incorporated in 1905 that is at elevation 1100m and the peak of Mount Shasta is 4320. It is unclear whether the locality refers to the mountain or to the city; either way it is unlikely that an altitude so close to the peak of the mountain is the best choice for this specimen.

So far those examples are all North America - does this problem exist in Europe too?

- A Lycaena hippothoe record from 1995 (row.index 2160; altitude 3274m) is from https://www.gbif.org/occurrence/2570253925 (https://www.gbif.org/occurrence/2570253925) which lists an inferred elevation of 2000m.
- A Lycaena virgaureae record from 2002 (row.index 4501; altitude -85.8m) appears to match https://www.gbif.org/occurrence/173651704 (https://www.gbif.org/occurrence/173651704) which is located in the Gulf of Bothnia, though GBIF assigns an elevation of 0m. Considering the lat/long are (65,23) most likely those coordinates are imprecise.

Altitude ~ Latitude collinearity

Fric et al. used regression of residuals for corrected analyses. Regression of residuals is not recommended, particularly if there could be collinearity among explanatory variables. We examined the collinearity between altitude and latitude, which would indicate the regression of residuals analysis would produce biased parameter estimates.

```
#Additional issues with altitude
#Given the use of regression of residuals, we were concerned that collinearity among independent variables could have led to biased results.

#How many datasets have significant collinearity between altitude and latitude?
templms<-NULL
datasets<-fricdata %>% group_by(name, region)  %>% tally()
for (spi in 1:nrow(datasets)) {
  tempdata<-fricdata %>% filter(name==datasets$name[spi],region==datasets$region[spi])
  spilm<-summary(lm(rndLat~alt, data=tempdata))
  templms<-rbind(templms,c(nrow(tempdata), spilm$coefficients[2,1],  spilm$coefficients[2,4], spilm$r.squared))
}
templms<-as.data.frame(templms)
names(templms)<-c("n","coef","pval","r2")
hist(templms$coef)
```


```
summary(templms)
```

```
##        n              coef               pval               r2
##  Min.   :   15   Min.   :-0.019376   Min.   :0.00000   Min.   :0.0000222
##  1st Qu.:   78   1st Qu.:-0.006861   1st Qu.:0.00000   1st Qu.:0.0311301
##  Median :  186   Median :-0.004516   Median :0.00000   Median :0.1936878
##  Mean   : 2456   Mean   :-0.003844   Mean   :0.06384   Mean   :0.2828444
##  3rd Qu.: 1067   3rd Qu.:-0.001088   3rd Qu.:0.00851   3rd Qu.:0.5261002
##  Max.   :51819   Max.   : 0.014623   Max.   :0.80204   Max.   :0.8487862
```

```
round(nrow(filter(templms,pval<0.05))/nrow(templms),2)
```

```
## [1] 0.85
```

```
#How many datasets have significant collinearity
print(paste(nrow(filter(templms,pval<0.05)),"datasets have significant collinearity, representing", round(nrow(filter(templm
s,pval<0.05))/nrow(templms)*100,1),"percent of all datasets. For datasets with significant collinearity, the mean coefficien
t is",round(mean(templms$coef[templms$pval<0.05]),3),"(which translates to a slope of", round(1/mean(templms$coef[templms$pv
al<0.05]),0),"meters per degree latitude) and mean r-squared is",round(mean(templms$r2[templms$pval<0.05]),3)," - therefore
 regression of residuals is likely producing bias parameters.",sep=" "))
```

```
## [1] "89 datasets have significant collinearity, representing 84.8 percent of all datasets. For datasets with significant
collinearity, the mean coefficient is -0.004 (which translates to a slope of -224 meters per degree latitude) and mean r-squ
ared is 0.33  - therefore regression of residuals is likely producing bias parameters."
```

Data exploration: data density

- In Fric et al. (2020), datasets were analysed with as few as 15 ocurrence records.
- We examine the prevalence of singleton ocurrences, when just one ocurrence was available in a latitudinal band.

```
lat.summary1<-fricdata %>%
  group_by(name, region, rndLat) %>%
  summarize(lat.samplesize=n(),singleton=ifelse(lat.samplesize==1,1,0),dur=max(SuccDay)-min(SuccDay))
```

```
## `summarise()` regrouping output by 'name', 'region' (override with `.groups` argument)
```

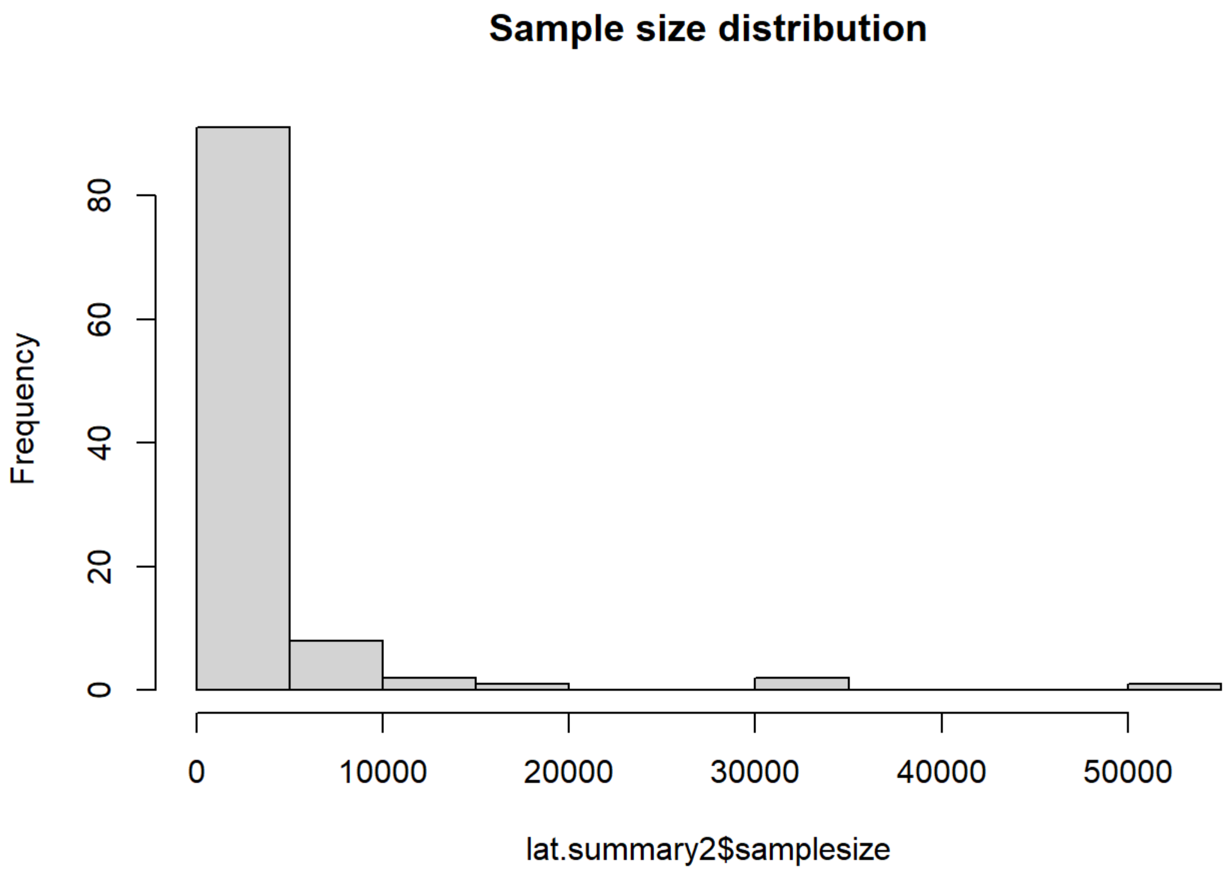
```
lat.summary2<-lat.summary1 %>%
  group_by(name,region) %>%
  summarize(samplesize=sum(lat.samplesize),latspan=max(rndLat)-min(rndLat),nlats=length(unique(rndLat)),n.singletons=sum(sin
gleton),prop.singletons=n.singletons/nlats)
```

```
## `summarise()` regrouping output by 'name' (override with `.groups` argument)
```

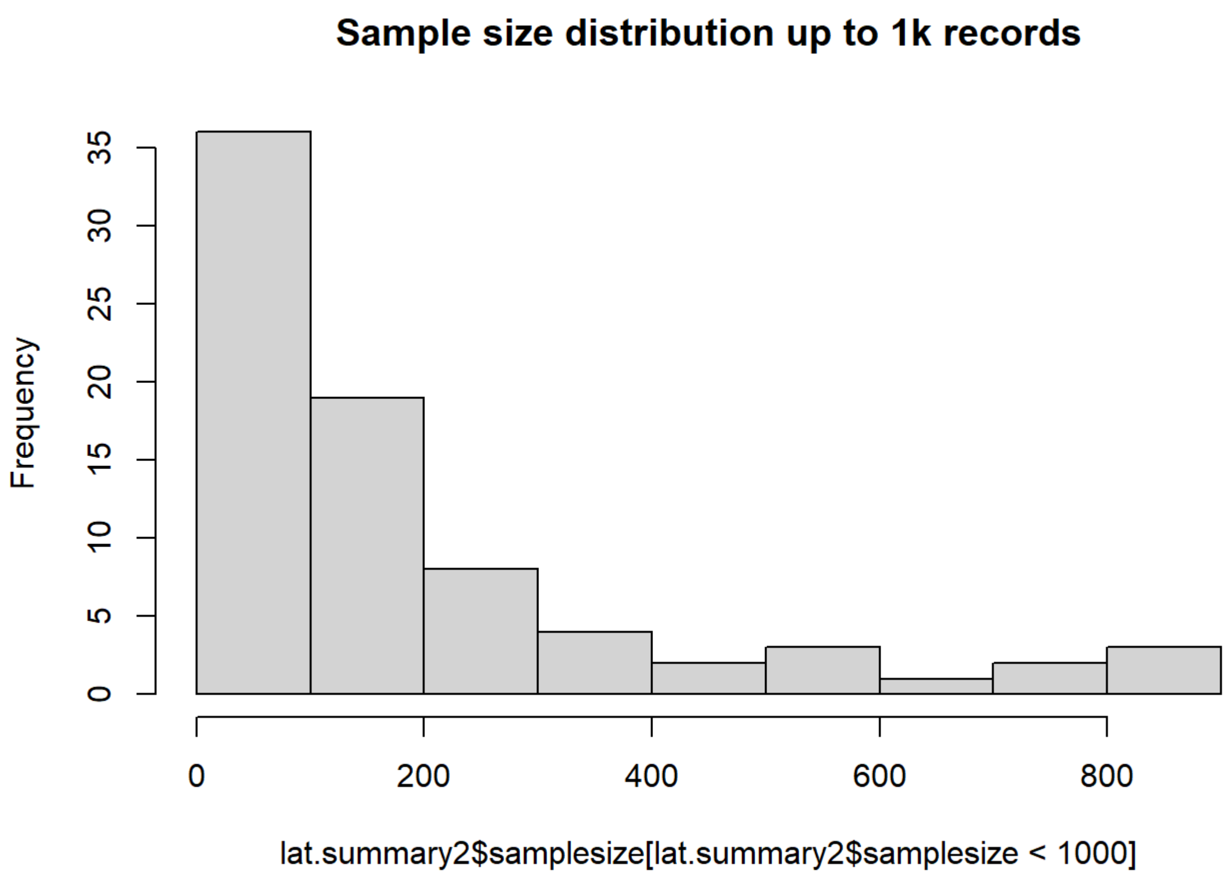
```
summary(lat.summary2)
```

```
##      name              region           samplesize      latspan
##  Length:105         Length:105         Min.   :   15   Min.   :10.0
##  Class :character   Class :character   1st Qu.:   78   1st Qu.:24.0
##  Mode  :character   Mode  :character   Median :  186   Median :27.0
##                                        Mean   : 2456   Mean   :26.3
##                                        3rd Qu.: 1067   3rd Qu.:30.0
##                                        Max.   :51819   Max.   :64.0
##      nlats        n.singletons    prop.singletons
##  Min.   : 5.00   Min.   : 0.000   Min.   :0.00000
##  1st Qu.:13.00   1st Qu.: 2.000   1st Qu.:0.09524
##  Median :18.00   Median : 3.000   Median :0.18750
##  Mean   :18.89   Mean   : 3.438   Mean   :0.20907
##  3rd Qu.:25.00   3rd Qu.: 5.000   3rd Qu.:0.33333
##  Max.   :33.00   Max.   :10.000   Max.   :0.60000
```

```
#Visualize range of sample sizes
hist(lat.summary2$samplesize, main="Sample size distribution")
```




```
#look at the lower end of sample sizes, where most datasets are
hist(lat.summary2$samplesize[lat.summary2$samplesize<1000], main="Sample size distribution up to 1k records")
```



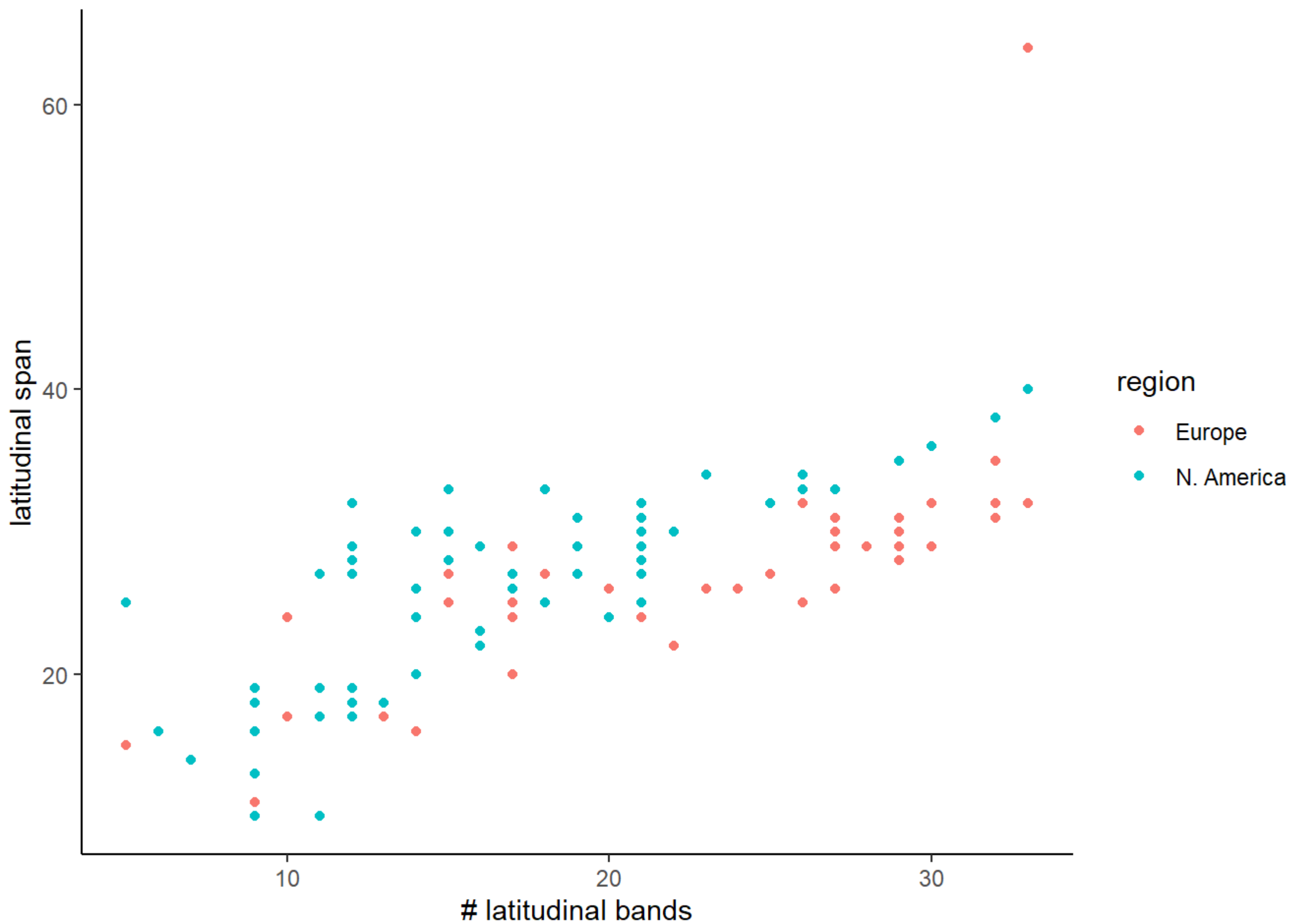

```
nrow(lat.summary2 %>% filter(samplesize<100))
```

```
## [1] 36
```

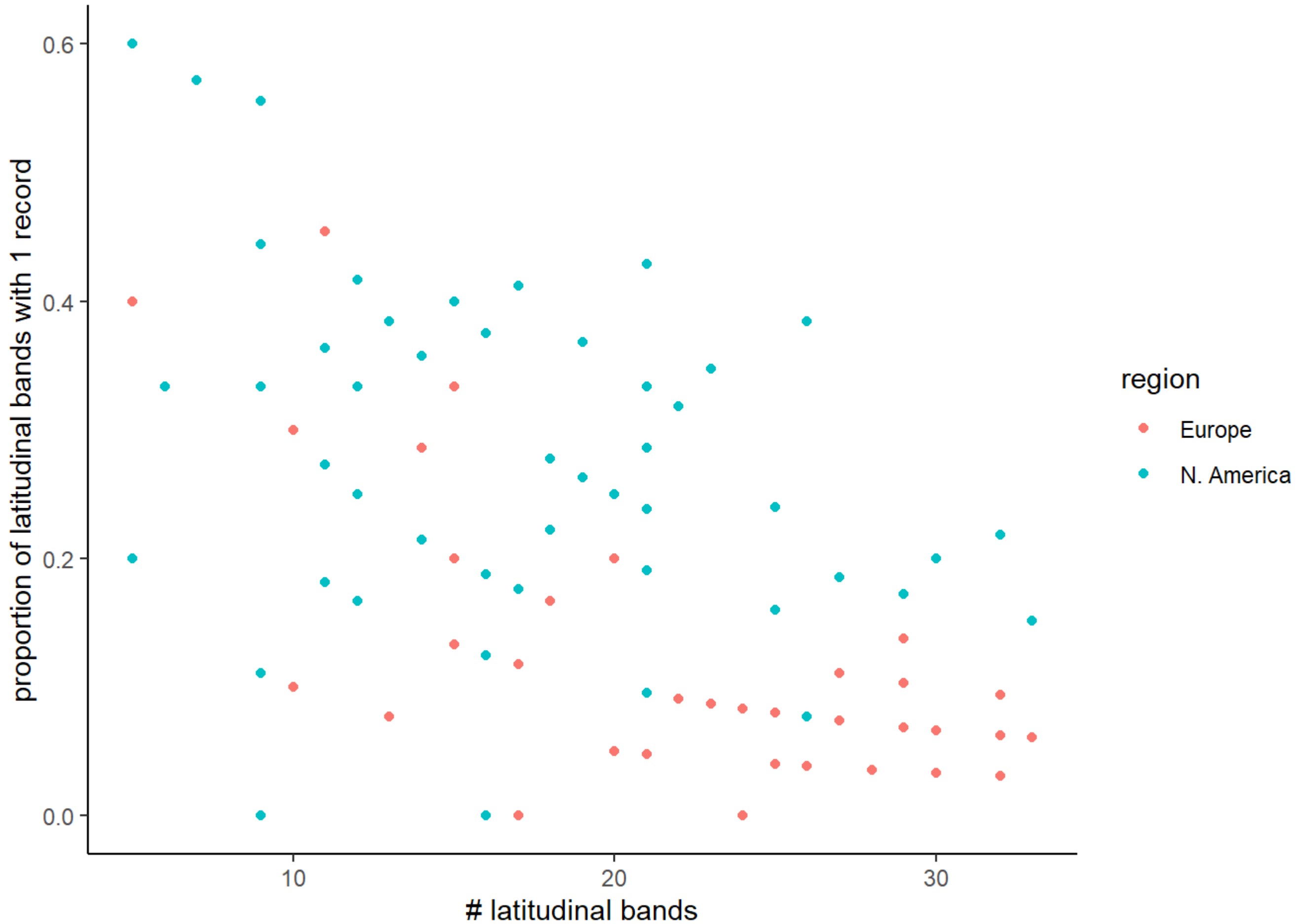
```
print(paste(nrow(lat.summary2 %>% filter(samplesize<100)),"datasets have less than 100 ocurrence records."))
```

```
## [1] "36 datasets have less than 100 ocurrence records."
```

```
ggplot(data=lat.summary2, aes(x=nlats, y=latspan, color=region)) + geom_point() + theme_classic() +
  labs(x="# latitudinal bands", y="latitudinal span")
```



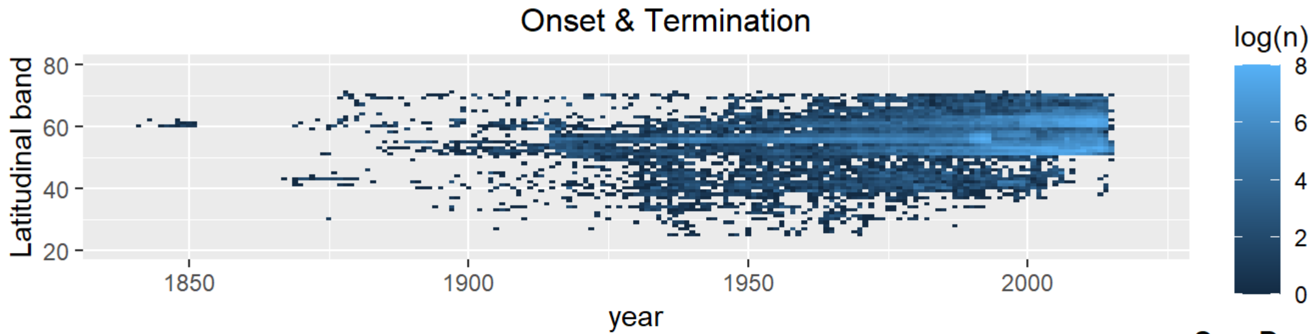

```
ggplot(data=lat.summary2, aes(x=nlats, y=prop.singletons, color=region)) + geom_point() + theme_classic() +
  labs(x="# latitudinal bands", y="proportion of latitudinal bands with 1 record")
```

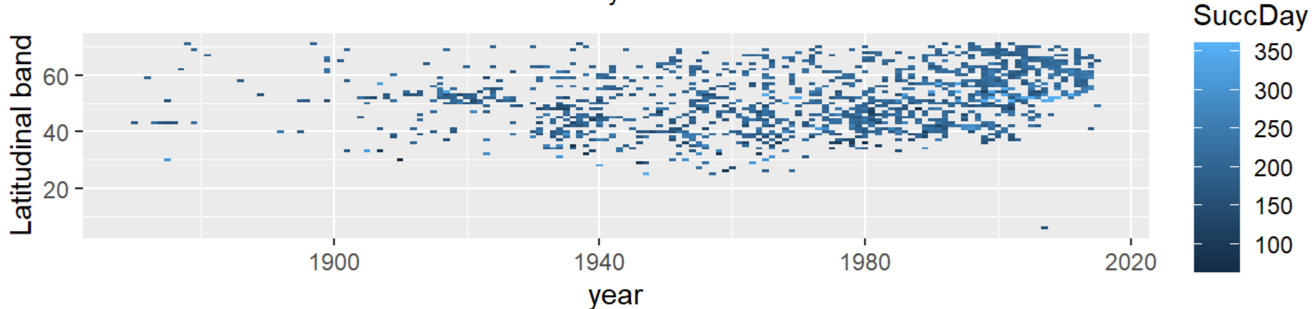


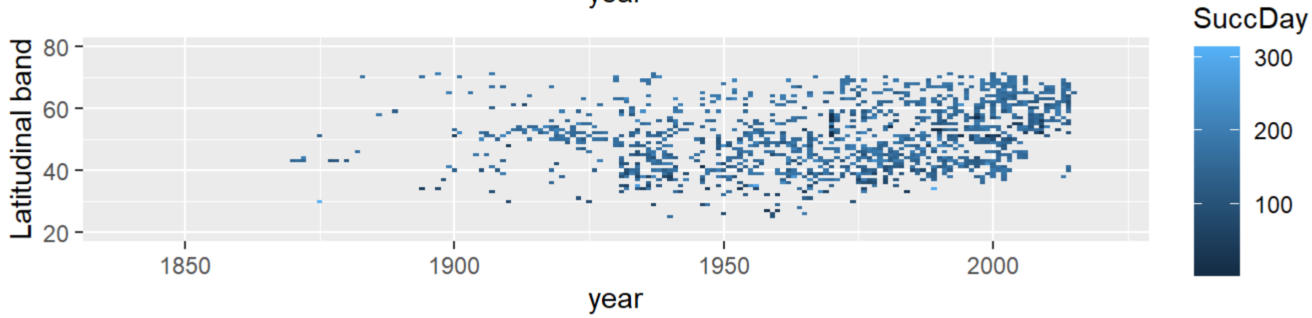
Data exploration: year

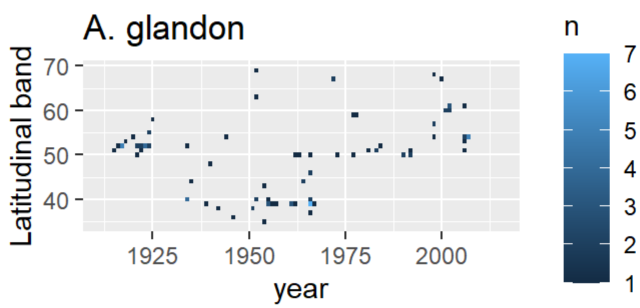
As expected, most data are quite recent. By selecting the min and max day of year per latitudinal band as onset & termination, the authors vastly decrease their sample size and remove most of the variation along the year and altitude axes

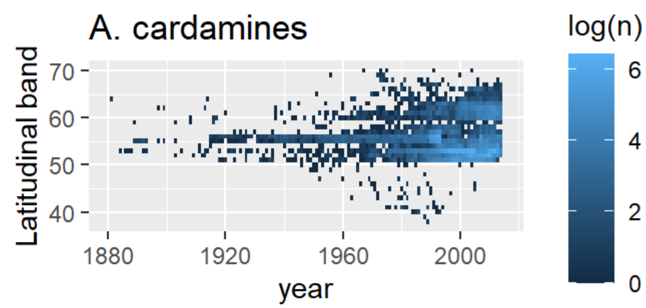
We arbitrarily selected two species, one with a low sample size and one with a large sample size, to visualize.

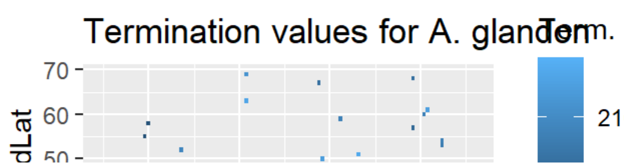


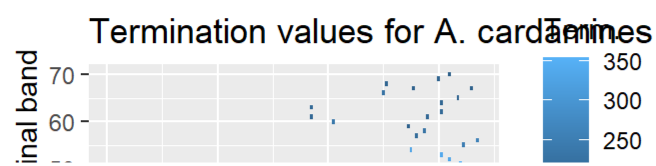


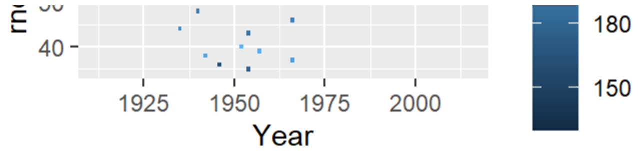


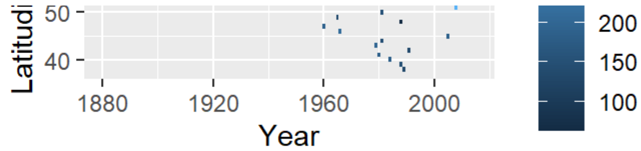


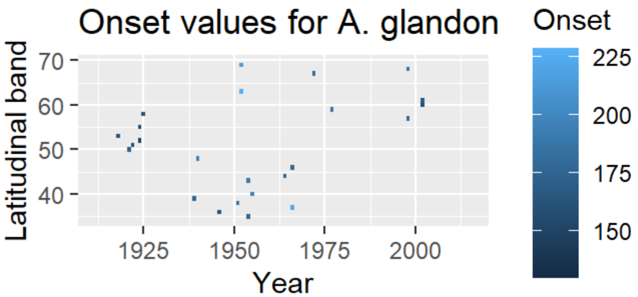


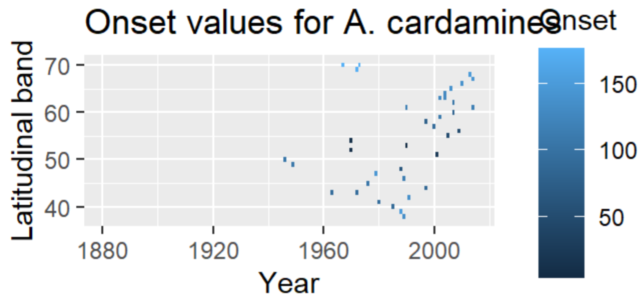


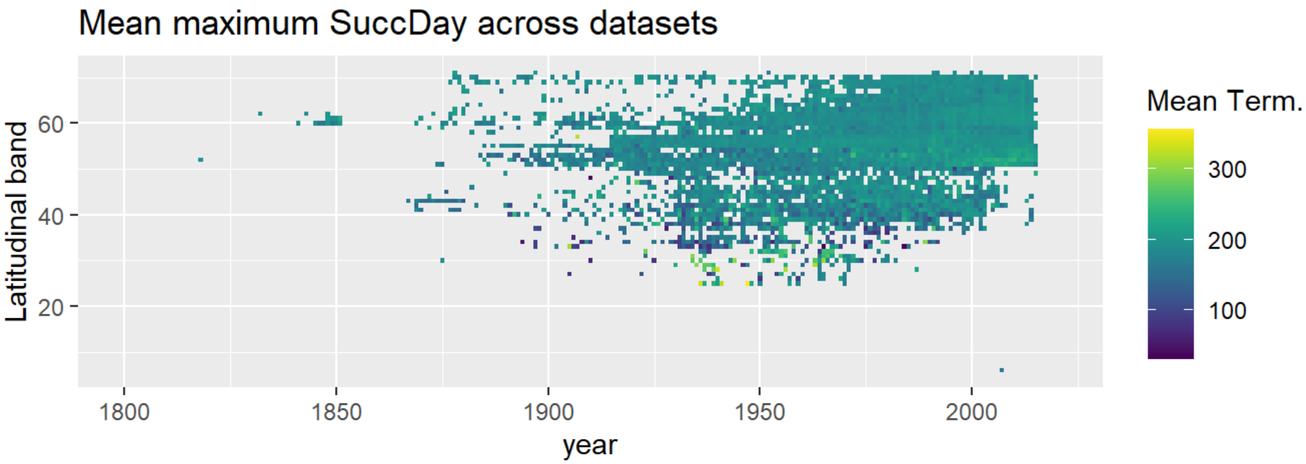


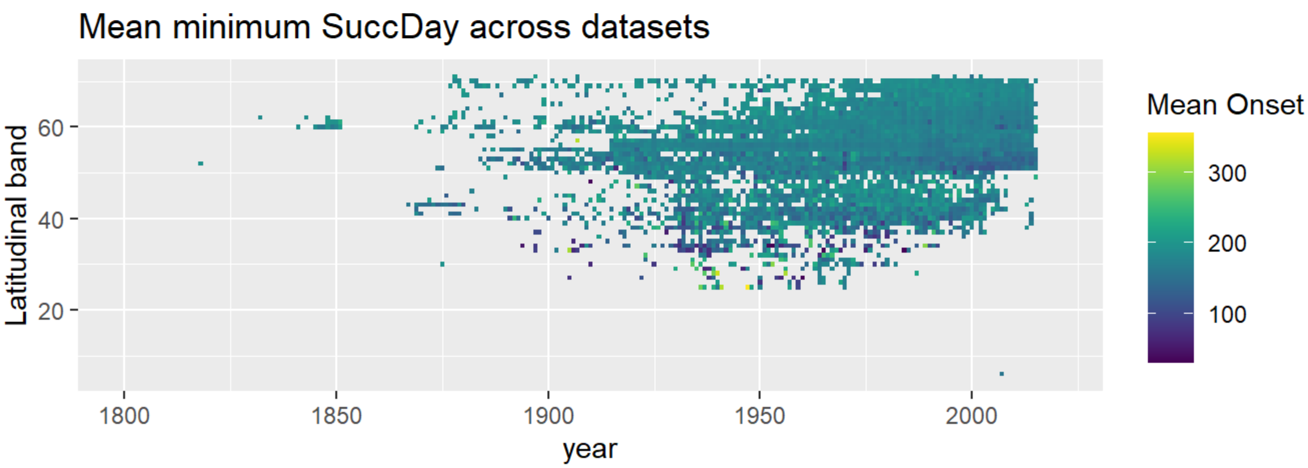

40
1925
1950
1975
2000
Year
180
150

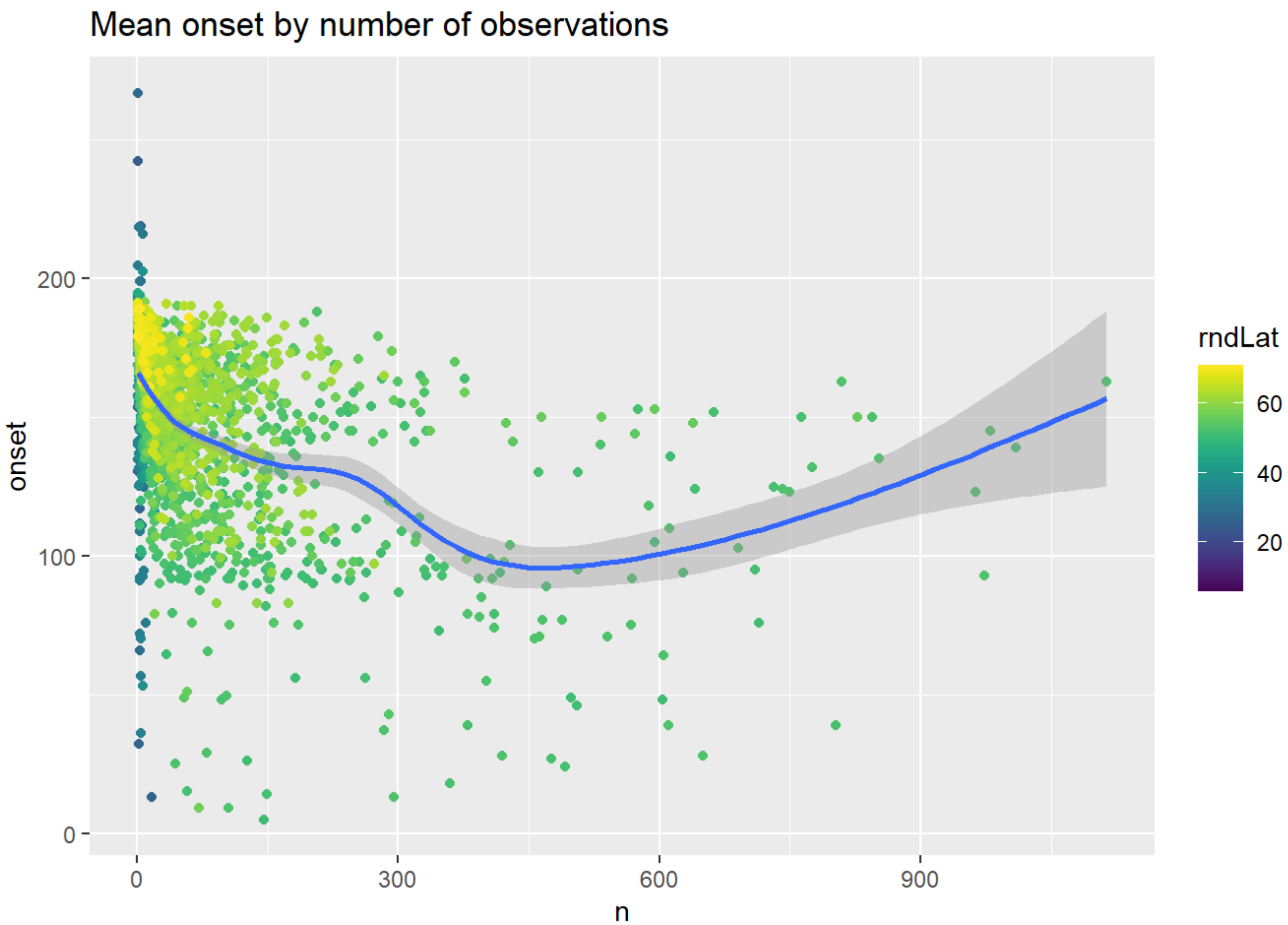

50
40
1880
1920
1960
2000
Year
200
150
100

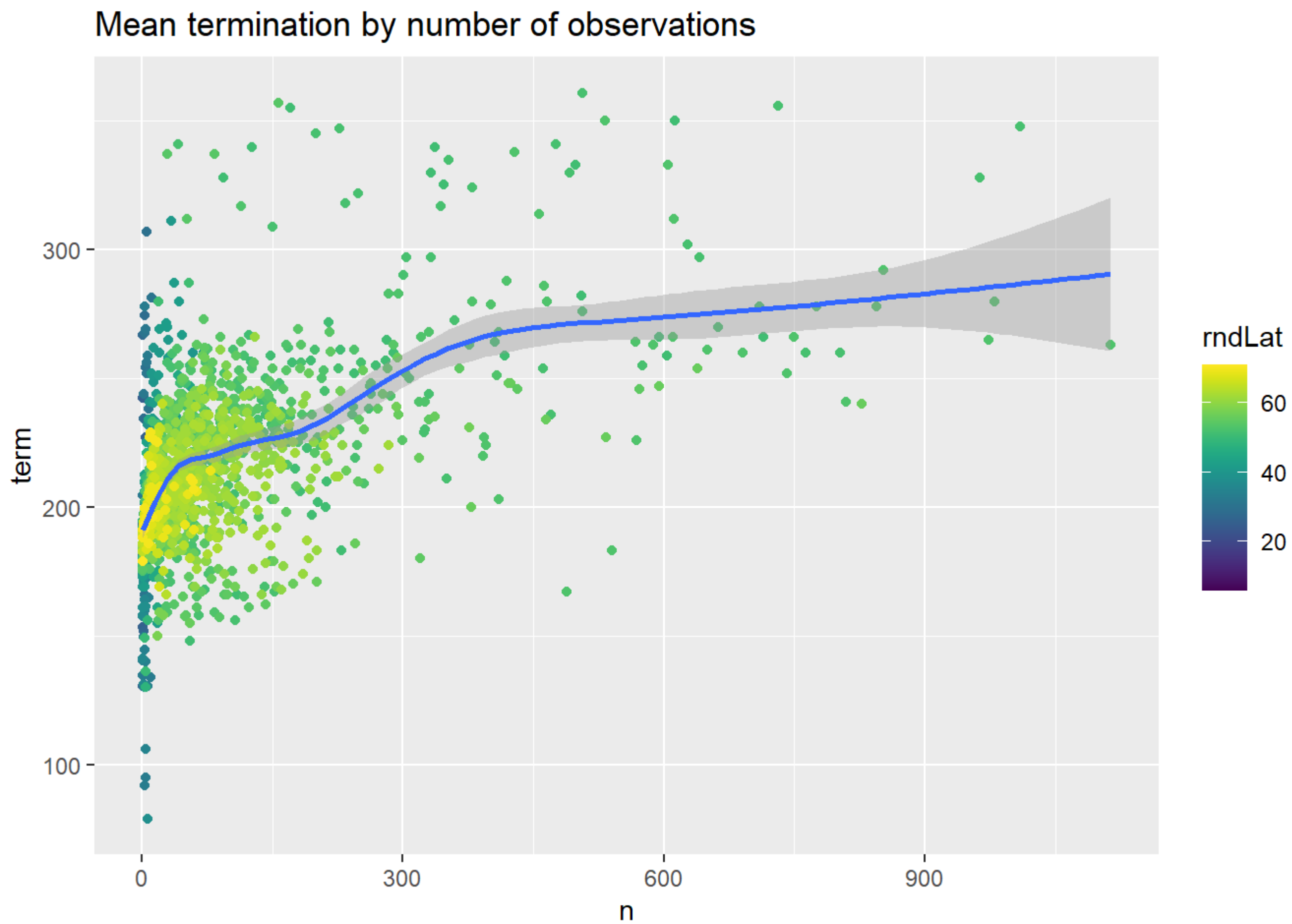

Onset values for A. glandon
Onset
Latitudinal band
70
60
50
40
1925
1950
1975
2000
Year
225
200
175
150


Onset values for A. cardamines
Onset
Latitudinal band
70
60
50
40
1880
1920
1960
2000
Year
150
100
50


Mean maximum SuccDay across datasets
Latitudinal band
60
40
20
1800
1850
1900
1950
2000
year
Mean Term.
300
200
100


Mean minimum SuccDay across datasets
Latitudinal band
60
40
20
1800
1850
1900
1950
2000
year
Mean Onset
300
200
100


Mean onset by number of observations
onset
200
100
0
0
300
600
900
n
rndLat
60
40
20


Mean termination by number of observations
term
300
200
100
0
300
600
900
n
rndLat
60
40
20

End of File.