Fric et al. Re-analysis Code

Vaughn Shirey & Elise Larsen

Current version 7-Dec-2020; initiated Feb-2020

*

###Begin Analysis
This code chunk sets up the workspace and loads necessary packages. If phest is not already installed, remove comment from install line.

This file imports data/occurrences.RData, data/voltinism.csv, and data/fric_results.RData. This file outputs several files to the outputs directory: LarsenShirey_Fig1.pdf, LarsenShirey_Fig2.pdf, Larsen_Shirey_stats_supp_table1.csv (which contributes to Supplemental Table 1), and LarsenShite_SuppTable2.csv)

```
#start fresh
rm(list=ls())
# load libraries
library(tidyverse)
library(ggplot2)
library(ggExtra)
library(gridExtra)
#library(devtools); install_github("willpearse/phest")
library(phest)
library(readxl)
library(lubridate)
```

*

Data Import and Formatting

data.csv file was downloaded from https://doi.org/10.6084/m9.figshare.9946934 (https://doi.org/10.6084/m9.figshare.9946934) (https://figshare.com/articles/Phenology_responses_of_temperate_butterflies_-_Supplementary_data/9946934 (https://figshare.com/articles/Phenology_responses_of_temperate_butterflies_-_Supplementary_data/9946934))

This cvs file contains the occurrence data used in Fric et al. (2020), which they downloaded from gbif. The file includes separate data tables for each dataset, which have been concatenated into one file. These data tables have the same fields but are not formatted as a single data table; individual datasets were all written into one data file, including headers and row indices in each dataset. This first set of code reformats the data & writes formatted data files.

```
load("data/occurrences.RData")
#load("data/occurrences_FricAnalysis.RData") our analysis suggests day 1 observations in months 2-12 were included
load("data/fric_results.RData")
```

*

Data Exploration

Here we use the alldata tibble and filter it to species in the Fric results. We calculate summary statistics for supplemental table 1.

```
#Tally the number of observations per dataset & calculate how each dataset spans latitude, year, altitude

alldata<-filter(alldata, name %in% fric.results$name)
spans.summary<-alldata %>%
  group_by(name, region) %>%
  add_count(name="fric_n") %>% ## n. records
  group_by(name, region, fric_n) %>%
  summarize(lat_span=(max(rndLat, na.rm=T)-min(rndLat, na.rm=T)),
         year_span=(max(year, na.rm=T)-min(year, na.rm=T)),
         alt_span=round((max(alt, na.rm=T)-min(alt, na.rm=T)),0))
```

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

```
#calculate # latitudes, onsets, terminations, flight curves = 0
endpt.summary<-alldata %>%
  group_by(name, region, rndLat) %>%
  # count no. records by latitudinal band
  add_count(name="n_recs") %>%
  #filter to onset & offset dates and label onset dates and offset dates
  filter(SuccDay==min(SuccDay) | SuccDay==max(SuccDay)) %>%
  mutate(onset=ifelse(SuccDay==min(SuccDay),1,0),    term=ifelse(SuccDay==max(SuccDay),1,0)) %>%
  group_by(name, region) %>%
  #create summary statistics by species & region
  summarize(n_lat=length(unique(rndLat)), n_onset=sum(onset), n_term=sum(term), n_flightcurve0s=sum(n_recs==1) )
```

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

```
#combine summary tables
fric.data.summary<-merge(spans.summary, endpt.summary, by=intersect(names(spans.summary), names(endpt.summary)))
rm(spans.summary)
summary(fric.data.summary)
```

```
##      name              region             fric_n         lat_span
##  Length:105         Length:105         Min.   :   15   Min.   :10.00
##  Class :character   Class :character   1st Qu.:   82   1st Qu.:24.00
##  Mode  :character   Mode  :character   Median :  192   Median :27.00
##                                        Mean   : 2461   Mean   :26.39
##                                        3rd Qu.: 1067   3rd Qu.:30.00
##                                        Max.   :51819   Max.   :64.00
##    year_span        alt_span        n_lat          n_onset          n_term
##  Min.   : 64.0   Min.   : 530   Min.   : 5.00   Min.   : 5.00   Min.   : 5.0
##  1st Qu.:102.0   1st Qu.:2113   1st Qu.:13.00   1st Qu.:15.00   1st Qu.:14.0
##  Median :117.0   Median :2737   Median :18.00   Median :19.00   Median :20.0
##  Mean   :127.3   Mean   :2740   Mean   :19.02   Mean   :20.58   Mean   :20.3
##  3rd Qu.:141.0   3rd Qu.:3495   3rd Qu.:25.00   3rd Qu.:27.00   3rd Qu.:26.0
##  Max.   :399.0   Max.   :5163   Max.   :34.00   Max.   :39.00   Max.   :36.0
##  n_flightcurve0s
##  Min.   :0.000
##  1st Qu.:2.000
##  Median :3.000
##  Mean   :3.429
##  3rd Qu.:5.000
##  Max.   :9.000
```

*

Explore data by altitude & latitude

This code chunk explores the spatiotemporal representation in the fric dataset.

Create Figure 1: Occurrences by altitude & latitude

This code outputs Larsen & Shirey Figure 1, which uses the 4 species presented in Fric et al. Figure 1, to demonstrate the spatiotemporal biases as well as the prevalence of flight periods with a duration of 0 days.

```
summary(alldata$alt)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2666.74    23.25    64.24   115.19   109.93  4305.17
```

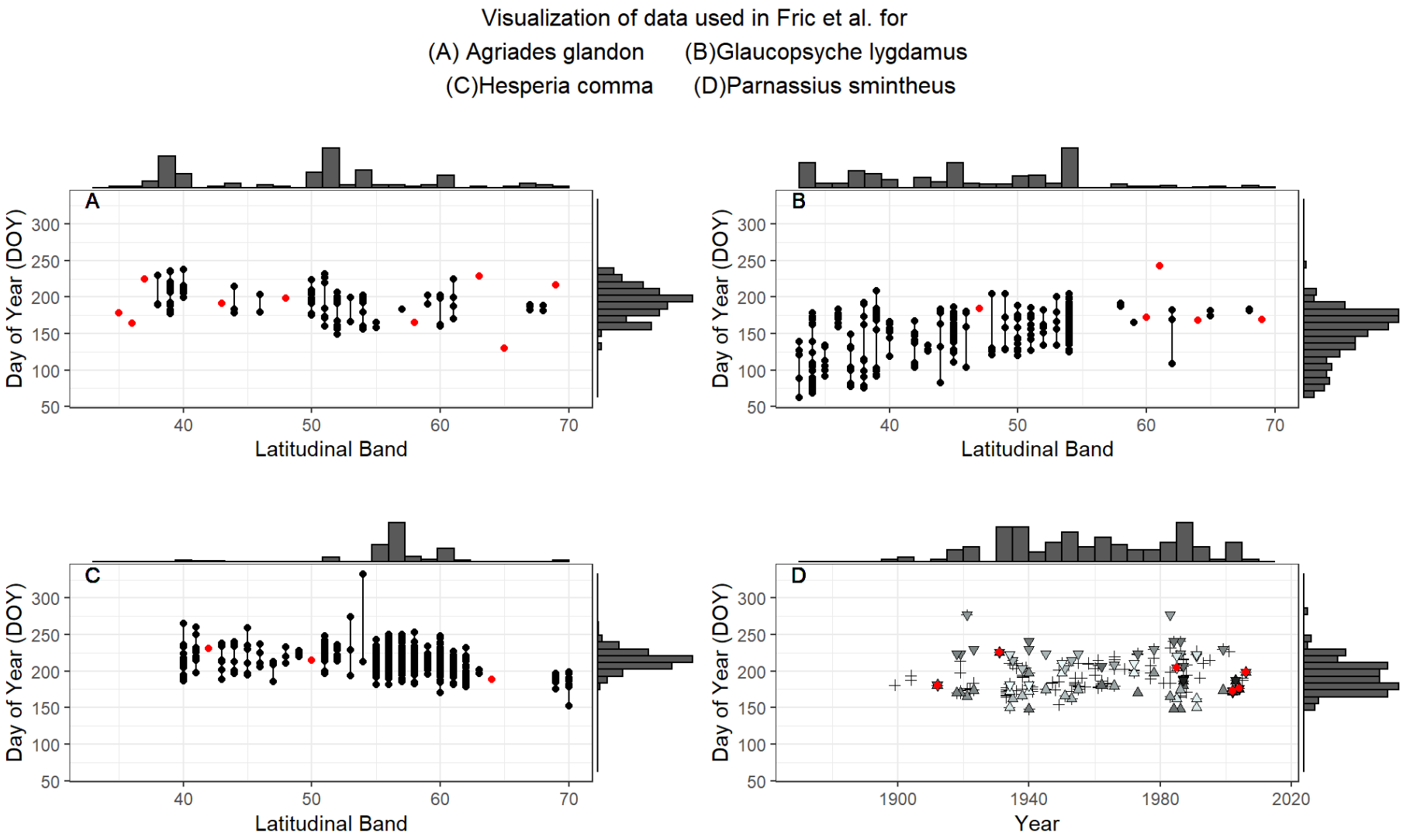
```
summary(alldata$decimalLatitude)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.787  52.690  55.631  56.278  60.635  71.216
```

```
##Create Figure 1
#species list
fric.datasets<-alldata %>% group_by(name, region) %>% tally()

fig1sp<-c("Agriades glandon","Glaucopsyche lygdamus","Hesperia comma","Parnassius smintheus")

#Filter data to these species
fig1data<-alldata %>%
  filter(name %in% fig1sp)

#Get onset & termination dates (SuccDay)
f1.pheno.data<-fig1data %>%
  group_by(name, region, rndLat) %>%
  mutate(onset=min(SuccDay), term=max(SuccDay), fp=term-onset, singles=ifelse(length(SuccDay)==1,1,0))

f1.pheno.data2<-f1.pheno.data %>%
  filter(SuccDay==onset | SuccDay==term)

#A list to store plot panels
tempplot<-list()
fig1panels<-list()

tags<-c("A","B","C","D")

#Create Panels
for(i in 1:2) {
  #paneltitle<-paste(fig1sp[i],"N. America")
  tempplot[[i]] <- ggplot(filter(f1.pheno.data, name==fig1sp[i], region=="N. America"), aes(x=rndLat, y=SuccDay, color=as.fa
ctor(singles))) +
    theme_bw() +
    theme(legend.position="none", plot.margin = margin(1,1,1,1, "in")) +
    geom_segment(data=filter(f1.pheno.data2, name==fig1sp[i], region=="N. America"), aes(x=rndLat, y=onset, xend=rndLat, yen
d=term)) +
    geom_point(aes(color=as.factor(singles))) +
    scale_color_manual(values=c("black","red")) +
    xlim(min(f1.pheno.data$rndLat),max(f1.pheno.data$rndLat)) + ylim(min(f1.pheno.data$SuccDay),max(f1.pheno.data$SuccDay))
 +
    labs(x="Latitudinal Band", y="Day of Year (DOY)", title="") + geom_text(x=min(f1.pheno.data$rndLat), y=max(f1.pheno.data
$SuccDay), label=tags[i])
  # with marginal histograms
```

```
  fig1panels[[i]] <- ggMarginal(tempplot[[i]], type="histogram")
  }

i<-3 #H. comma panel in Fric et al. is from Europe
#paneltitle<-paste(fig1sp[i],"Europe")
  tempplot[[i]] <- ggplot(filter(f1.pheno.data, name==fig1sp[i], region=="Europe"), aes(x=rndLat, y=SuccDay, color=as.factor
(singles))) +
    theme_bw() +
    theme(legend.position="none", plot.margin = margin(1,1,1,1, "in")) +
    geom_segment(data=filter(f1.pheno.data2, name==fig1sp[i], region=="Europe"), aes(x=rndLat, y=onset, xend=rndLat, yend=te
rm)) +
    geom_point(aes(color=as.factor(singles))) +
    scale_color_manual(values=c("black","red")) +
    xlim(min(f1.pheno.data$rndLat),max(f1.pheno.data$rndLat)) + ylim(min(f1.pheno.data$SuccDay),max(f1.pheno.data$SuccDay))
 +
    labs(x="Latitudinal Band", y="Day of Year (DOY)", title="") + geom_text(x=min(f1.pheno.data$rndLat), y=max(f1.pheno.data
$SuccDay), label=tags[i])

  # with marginal histogram
  fig1panels[[i]] <- ggMarginal(tempplot[[i]], type="histogram")

##### Figure 1d  2020-07-29 update uses YEAR and DAY to mirror Fric et al.
i<-4
#paneltitle<-paste(fig1sp[i],"N. America")
tempplot[[i]]<- ggplot(filter(f1.pheno.data, name==fig1sp[i], region=="N. America"), aes(x=year, y=SuccDay, fill=decimalLati
tude)) +
  geom_point(shape=3) +
  theme_bw() +
  theme(legend.position="none", plot.margin = margin(1,1,1,1, "in")) +
  geom_point(data=filter(f1.pheno.data2, name==fig1sp[i], region=="N. America"), aes(x=year, y=onset, fill=decimalLatitude),
shape=24) +
  geom_point(data=filter(f1.pheno.data2, name==fig1sp[i], region=="N. America"), aes(x=year, y=term, fill=decimalLatitude),
 shape=25) +
  scale_fill_gradient(low="azure1", high="black") +
  geom_point(data=filter(f1.pheno.data2, name==fig1sp[i], region=="N. America", singles==1), aes(x=year, y=SuccDay), color=
"red", shape=16) +
  xlim(min(f1.pheno.data$year),max(f1.pheno.data$year)) + ylim(min(f1.pheno.data$SuccDay),max(f1.pheno.data$SuccDay)) +
  labs(x="Year", y="Day of Year (DOY)", title="") + geom_text(x=min(f1.pheno.data$year), y=max(f1.pheno.data$SuccDay), label
=tags[i])

# with marginal histogram
```

```
fig1panels[[i]]  <- ggMarginal(tempplot[[i]], type="histogram")

grid.arrange(grobs=fig1panels[c(1:4)], nrow=2, ncol=2, top="Visualization of data used in Fric et al. for \n (A) Agriades gl
andon      (B)Glaucopsyche lygdamus \n (C)Hesperia comma      (D)Parnassius smintheus")
```



```
#Used to create figure 1 pdf
pdf_filename<-("outputs/LarsenShirey_Fig1.pdf")
ggsave(pdf_filename, grid.arrange(grobs=fig1panels[c(1:4)], nrow=2, ncol=2, top=" \n ", bottom=" \n ", left=" \n  \n ", righ
t=" \n \n " ), width=8, height=8, units="in", scale=1,dpi=600)
```

*

Data curation

The following code chunk applies the filters used in the Larsen & Shirey reanalysis and calculates summary data density statistics for all species present in Fric's results to output to Supplemental Table 1.

Our reanalysis excludes datasets along two axes - data density, and voltinism. This code examines data along the data density axis. Unlike Fric et al., we include first day of the month records. We curate raw occurrence data with the following filters prior to estimating phenometrics:

- 1 - remove Euphydryas aurinia (as Fric et al. did)
- 2 - altitude in [0m,500m]
- 3 - DOY in (60,330) which corresponds to start of march to late november
- 4 - 10 or more records for calculating phenometrics for data grouped by species, region, year, and latitudinal band

```
#Summarize data availability for Larsen & Shirey re-analysis
#Now, filter data for altitude & for cases with 10 or more records by species-region-year-latitude
all.datasets<-alldata %>% group_by(name, region) %>% tally()
new.data.summary<-alldata %>%
  filter(between(alt,0,500), name!="Euphydryas aurinia", doy %in% c(60:330)) %>%
  # calculate data availability by species, region, latitude & year
  group_by(name, region, rndLat, year) %>%
  add_count(name="group_n") %>% ## n. observations per group
  filter(group_n>=10) %>% ### filter by 10 or more observations in group
  # calculate reanalysis statistics by species & region
  group_by(name, region) %>%
  add_count(name="curated_n_obs") %>%
  group_by(name, region, curated_n_obs) %>%
  #calculate summary statistics applying data filters
  summarize(curated_n_lat=length(unique(rndLat)),  curated_n_fcurve=length(unique(paste(rndLat,year))),
            curated_lat_span=(max(rndLat, na.rm=T)-min(rndLat, na.rm=T)),
            curated_year_span=(max(year, na.rm=T)-min(year, na.rm=T)),
            curated_alt_span=round((max(alt, na.rm=T)-min(alt, na.rm=T)),0))
```

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

```
#combine summary tables
supptable1<-merge(fric.data.summary, new.data.summary, by=intersect(names(fric.data.summary), names(new.data.summary)), all.
x=T)
head(supptable1)
```

```
##                     name     region fric_n lat_span year_span alt_span n_lat
## 1       Agriades glandon N. America    113       34       103     4042    26
## 2    Amblyscirtes vialis N. America     97       29       133     2775    21
## 3 Anthocharis cardamines     Europe  31849       32       168     2595    33
## 4       Anthocharis sara N. America    229       28       111     4417    21
## 5  Aphantopus hyperantus     Europe  30598       25       399     2102    26
## 6        Aporia crataegi     Europe   5172       64       165     2520    33
##   n_onset n_term n_flightcurve0s curated_n_obs curated_n_lat curated_n_fcurve
## 1      27     27               9            NA            NA              NA
## 2      22     21               9            NA            NA              NA
## 3      39     35               2         29134            17             393
## 4      22     22               6            NA            NA              NA
## 5      27     28               1         27878            15             330
## 6      34     35               2          4055            10             108
##   curated_lat_span curated_year_span curated_alt_span
## 1               NA                NA               NA
## 2               NA                NA               NA
## 3               16                80              499
## 4               NA                NA               NA
## 5               14                79              488
## 6               13                74              268
```

```
summary(supptable1)
```

```
##      name              region             fric_n         lat_span
##  Length:105         Length:105         Min.   :   15   Min.   :10.00
##  Class :character   Class :character   1st Qu.:   82   1st Qu.:24.00
##  Mode  :character   Mode  :character   Median :  192   Median :27.00
##                                        Mean   : 2461   Mean   :26.39
##                                        3rd Qu.: 1067   3rd Qu.:30.00
##                                        Max.   :51819   Max.   :64.00
##
##    year_span        alt_span        n_lat          n_onset          n_term
##  Min.   : 64.0   Min.   : 530   Min.   : 5.00   Min.   : 5.00   Min.   : 5.0
##  1st Qu.:102.0   1st Qu.:2113   1st Qu.:13.00   1st Qu.:15.00   1st Qu.:14.0
##  Median :117.0   Median :2737   Median :18.00   Median :19.00   Median :20.0
##  Mean   :127.3   Mean   :2740   Mean   :19.02   Mean   :20.58   Mean   :20.3
##  3rd Qu.:141.0   3rd Qu.:3495   3rd Qu.:25.00   3rd Qu.:27.00   3rd Qu.:26.0
##  Max.   :399.0   Max.   :5163   Max.   :34.00   Max.   :39.00   Max.   :36.0
##
##  n_flightcurve0s curated_n_obs     curated_n_lat curated_n_fcurve
##  Min.   :0.000   Min.   :   10.0   Min.   : 1    Min.   :  1.00
##  1st Qu.:2.000   1st Qu.:   42.5   1st Qu.: 2    1st Qu.:  3.00
##  Median :3.000   Median :  361.0   Median : 3    Median : 23.00
##  Mean   :3.429   Mean   : 3920.1   Mean   : 6    Mean   : 76.35
##  3rd Qu.:5.000   3rd Qu.: 3928.0   3rd Qu.:10    3rd Qu.:124.50
##  Max.   :9.000   Max.   :47611.0   Max.   :17    Max.   :393.00
##                  NA's   :54        NA's   :54    NA's   :54
##  curated_lat_span curated_year_span curated_alt_span
##  Min.   : 0.000   Min.   :  0.00    Min.   :  0.0
##  1st Qu.: 1.000   1st Qu.:  9.50    1st Qu.:228.5
##  Median : 5.000   Median : 34.00    Median :379.0
##  Mean   : 6.569   Mean   : 43.63    Mean   :330.9
##  3rd Qu.:11.500   3rd Qu.: 74.00    3rd Qu.:467.5
##  Max.   :18.000   Max.   :123.00    Max.   :499.0
##  NA's   :54       NA's   :54        NA's   :54
```

```
#output summary table to csv file
write_csv(supptable1, "Larsen&Shirey_stats_supp_table1.csv")
rm(fric.data.summary, new.data.summary, endpt.summary)
rm(fig1sp, fig1data, f1.pheno.data, f1.pheno.data2, fig1panels, tempplot, pdf_filename, tags)
```

*

Data curation for reanalysis

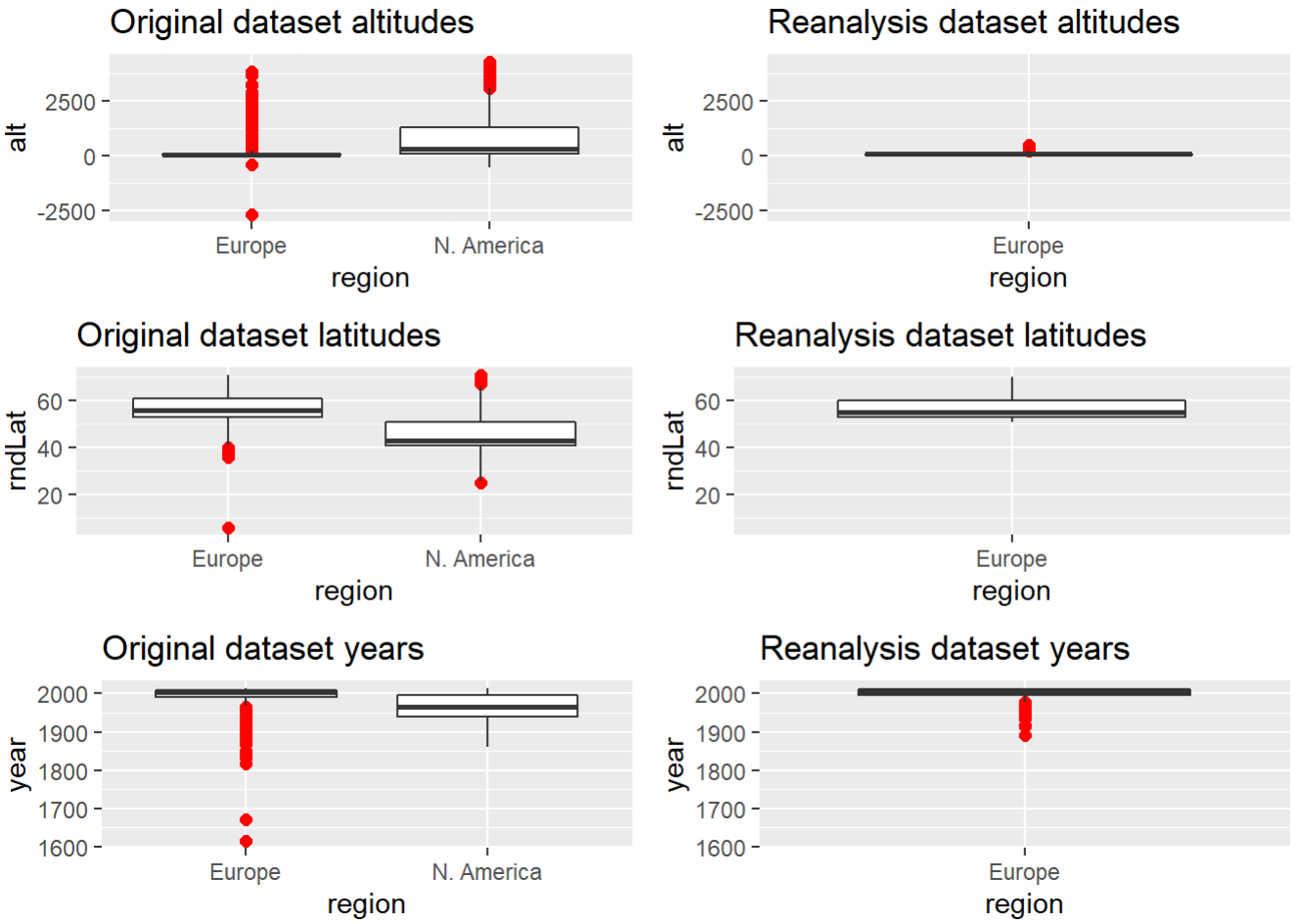
This code filters occurrence data for reanalysis by voltinism and data density, and visualizes some differences between datasets curated for the original analysis and this reanalysis. We only include datasets with sufficient data for calculating phenometrics at 3 or more distinct latitudinal bands, so that a linear model can be applied.

```
#FILTER DATA BY VOLTINISM

#get species list without evidence of multiple generations
#Euphydryas aurinia is not included in the voltinism file
voltindata<-read_csv("data/voltinism.csv")
```

```
## Parsed with column specification:
## cols(
##   id = col_double(),
##   name_datafile = col_character(),
##   name_resultsfile = col_character(),
##   region = col_character(),
##   Voltinism = col_character(),
##   Voltinism_source = col_character(),
##   `In reanalysis?` = col_double(),
##   Why_excluded = col_character()
## )
```

```
voltindata<-na.omit(voltindata[,c(1:8)])
voltindata<-voltindata %>% select(name=name_resultsfile,region,Voltinism)
multi<-c("Univoltine","Univoltine, sometimes biennial","Not determined")
univoltine<-filter(voltindata, Voltinism %in% multi)
rm(voltindata, multi)

#filter occurrence dataset to these species
reanalysis.data<-merge(alldata, univoltine, by=intersect(names(alldata),names(univoltine)))
rm(univoltine)

#filter data by altitude and data density
reanalysis.data<-reanalysis.data %>%
  filter(between(alt,0,500), doy %in% c(60:330)) %>%
  # calculate data availability by species, region, latitude & year
  group_by(name, region, rndLat, year) %>%
  add_count(name="group_n") %>% ## n. observations per group
  filter(group_n>=10) %>% #only groups with at least 10 observations
  group_by(name, region) %>% #group by "dataset"
  mutate(nlat=length(unique(rndLat))) %>% #count how many distinct latitudinal bands included
  filter(nlat>=3) # need at least 3 latitudinal bands

#visualize some differences
plotcompar<-list()
plotcompar[[1]]<-ggplot(data=alldata, aes(x=region, y=alt) ) +
  geom_boxplot(outlier.colour="red", outlier.shape=16, outlier.size=2, notch=FALSE) + ggtitle(label="Original dataset altitu
des")

plotcompar[[2]]<-ggplot(data=reanalysis.data, aes(x=region, y=alt) ) +
  geom_boxplot(outlier.colour="red", outlier.shape=16, outlier.size=2, notch=FALSE) + ggtitle(label="Reanalysis dataset alti
tudes") + ylim(min(alldata$alt),max(alldata$alt))

plotcompar[[3]]<-ggplot(data=alldata, aes(x=region, y=rndLat) ) +
  geom_boxplot(outlier.colour="red", outlier.shape=16, outlier.size=2, notch=FALSE) + ggtitle(label="Original dataset latitu
des")

plotcompar[[4]]<-ggplot(data=reanalysis.data, aes(x=region, y=rndLat) ) +
  geom_boxplot(outlier.colour="red", outlier.shape=16, outlier.size=2, notch=FALSE) + ggtitle(label="Reanalysis dataset lati
tudes") + ylim(min(alldata$rndLat), max(alldata$rndLat))

plotcompar[[5]]<-ggplot(data=filter(alldata, !is.na(year)), aes(x=region, y=year) ) +
```

```
  geom_boxplot(outlier.colour="red", outlier.shape=16, outlier.size=2, notch=FALSE) + ggtitle(label="Original dataset years"
)

plotcompar[[6]]<-ggplot(data=reanalysis.data, aes(x=region, y=year) ) +
  geom_boxplot(outlier.colour="red", outlier.shape=16, outlier.size=2, notch=FALSE) + ggtitle(label="Reanalysis dataset year
s") + ylim(min(alldata$year, na.rm=T), max(alldata$year, na.rm=T))

grid.arrange(grobs=plotcompar[c(1:6)], nrow=3)
```



*

Estimate phenometrics using phest

This chunk of code estimates onset and offset phenometrics by species-region-year-latitudinal_band using curated data.

We use the phest package to estimate onset and offset of flight periods based on occurrence data, when at least 10 observations exist for a species-region-year-latitudinal_band unit. The phest package applies a weibull distribution. Please note that this chunk does take a few minutes to run. Also, warnings are automatically generated by “phest” when a correction is applied to the phenometric estimate. Additionally, “phest” throws a warning for CI estimation. We have explored these warnings and don’t believe that there is any problem continuing with the estimates produced; therefore we have suppressed the warning messages here.

We have added an easy way to select whether to estimate the phenometrics directly or to load them from a saved .RData file. To bypass the weibull estimation, set calc.new.metrics to FALSE. To run the weibull estimation, set calc.new.metrics to TRUE.

```
rm(plotcompar)

#If you want to just load the previously estimated phenometrics, set this to FALSE.
calc.new.metrics<-TRUE

datasets.ls<-reanalysis.data %>% group_by(name, region) %>% tally()

#For each species & region, calculate phenometrics
if(calc.new.metrics) {
  pheno.est<-data.frame(name=character(0),region=character(0),year=integer(0),rndLat=integer(0),onset.est=numeric(0),onset.l
ow=numeric(0),onset.high=numeric(0),offset.est=numeric(0),offset.low=numeric(0),offset.high=numeric(0))

  for(rowi in 1:nrow(datasets.ls)){ # for each unique dataset
    namei<-datasets.ls$name[rowi]
    regi<-datasets.ls$region[rowi]
    index <- 1 # create/reset an indexer
    pheno.estimates <- list() # create/refresh a blank list per group
    rowi.data<-filter(reanalysis.data, name==namei, region==regi)
    for(yr in unique(rowi.data$year)){ # and each unique year
      for(lat in unique(rowi.data$rndLat)){ # and each unique latitude
        temp <- filter(rowi.data, rndLat==lat, year==yr) # filter the occurrence data for each group

        if(nrow(temp) > 9){ # if there are at least 10 occurrences, then...
          estimates <- c(namei, regi, yr, lat, nrow(temp),
                         suppressWarnings(weib.limit(temp$doy, upper=FALSE, alpha=0.05)),  suppressWarnings(weib.limit(temp$do
y, upper=TRUE, alpha=0.05))) # calculate estimates for the group: onset, offset
          pheno.estimates[[index]] <- estimates # shuttle those into a list
          index <- index+1
        } #end if enough occurrences
      } #end lat
    } #end yr
    df <- data.frame(matrix(unlist(pheno.estimates), nrow=length(pheno.estimates), byrow=TRUE),stringsAsFactors=FALSE)
    names(df)<-c("name","region","year","rndLat","n","onset.est","onset.low","onset.high","offset.est","offset.low","offset.
high")
    pheno.est<-rbind(pheno.est, df)
  }
  for(coli in 3:11) {
    pheno.est[,coli]<-as.numeric(pheno.est[,coli])
  }
```

```
  #Format & store data
  pheno.data<-pheno.est %>%
    mutate(unit=paste(name, rndLat, year,sep="-")) %>%
    select(unit,onset.est,offset.est,name,region,rndLat,year,n) %>%
    mutate(onset=round(onset.est,0),term=round(offset.est,0))
  pheno.data<-na.omit(pheno.data)
  #Weibull estimator doesn't bound so
  #We bounded all onset & termination metrics y [60,330], limiting flight periods to March - November
  nrow(pheno.data[pheno.data$onset<60,])
  nrow(pheno.data[pheno.data$term>330,])
  pheno.data$onset[pheno.data$onset<60]<-60
  pheno.data$term[pheno.data$term>330]<-330

  save(pheno.data, file="data/phenometrics.RData")
  rm(estimates, index,lat,namei,regi,rowi,yr,coli,calc.new.metrics,temp,df,pheno.est, pheno.estimates)
  } else {
  #If we want to skip phest and phenometric estimation:
  load("data/phenometrics.RData")
}
```

*

Statistical models for phenometrics

This code uses estimated onset and offset phenometrics in linear models to examine phenological patterns with latitude and year. Other statistical models may be more appropriate for a de novo analysis, but here we want our statistical model to parallel the Fric et al. model in intention, but using multiple regression instead of residual regression.

```
datasets<-pheno.data %>%
  group_by(name, region) %>%
  tally()
pheno.data<-na.omit(pheno.data)
fric_FP<-alldata %>%
  group_by(name,region,rndLat) %>%
  summarize(onset=min(SuccDay),term=max(SuccDay),FP=term-onset)
```

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

```
verify.order<-pheno.data %>%
  mutate(FP=term-onset)
summary(verify.order$FP)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6.00   44.00   56.00   62.01   71.00  252.00
```

```
print(paste("Across datasets our estimated flight periods average ", round(mean(verify.order$FP, na.rm=T))," days, and range from ", min(verify.order$FP, na.rm=T), " days to ",max(verify.order$FP, na.rm=T), " days. In the original analysis, the average flight period duration was ", round(mean(fric_FP$FP, na.rm=T)), " days, with a range of ",min(fric_FP$FP, na.rm=T),"-", max(fric_FP$FP, na.rm=T), " days.",sep=""))
```

```
## [1] "Across datasets our estimated flight periods average 62 days, and range from 6 days to 252 days. In the original analysis, the average flight period duration was 49 days, with a range of 0-359 days."
```

```
rm(verify.order)

#Loop through datasets, run model for phenology by species & region, and store LM parameters
onsetpheno<-list()
termpheno<-list()
onset1<-NULL
term1<-NULL
axeso<-NULL
axest<-NULL

for(rowi in 1:nrow(datasets)) {
  pheno.rowi<-pheno.data %>%
    filter(name==datasets$name[rowi], region==datasets$region[rowi])
#estimate model params for onset
  onset.lm<-summary(lm(onset~rndLat+year, data=pheno.rowi))$coefficients #estimate model params for termination
  term.lm<-summary(lm(term~rndLat+year, data=pheno.rowi))$coefficients
#store
  onsetpheno[[rowi]]<-onset.lm
  termpheno[[rowi]]<-term.lm

#onset
  temponset<-matrix(unlist(onset.lm[c(2:3),]), ncol=4, byrow=F)
  onset1<-rbind(onset1, temponset)
  axeso<-c(axeso,row.names(onset.lm)[c(2:3)])
#termination
  tempterm<-matrix(unlist(term.lm[c(2:3),]), ncol=4, byrow=F)
  term1<-rbind(term1, tempterm)
  axest<-c(axest,row.names(term.lm)[c(2:3)])
  rm(pheno.rowi,onset.lm,term.lm,temponset,tempterm)
  }

#Create results dataframes: onset
onset1<-as.data.frame(onset1)
colnames(onset1)<-c("param.est","param.se","param.t","param.p")
onset1$param<-axeso
onset1$metric<-"onset"
onset1$name<-rep(datasets$name, each=2)
onset1$region<-rep(datasets$region, each=2)
onset1$n<-rep(datasets$n, each=2)
```

```
#Create results dataframes: termination
term1<-as.data.frame(term1)
colnames(term1)<-c("param.est","param.se","param.t","param.p")
term1$param<-axest
term1$metric<-"termination"
term1$name<-rep(datasets$name, each=2)
term1$region<-rep(datasets$region, each=2)
term1$n<-rep(datasets$n, each=2)

result<-bind_rows(onset1, term1)
result<-result %>%
  mutate(response=ifelse(param.p<0.05,ifelse(param.est>0,1,-1),0))
```

*

Compare statistical results to Fric et al.

This code uses model outputs and compares them to the results of the Fric et al. analysis. It outputs Figure 2.

```
##Results and visualizations

datasets$set<-paste(datasets$name,datasets$region,sep="-")
##Import Fric results:
load("data/fric_results.RData")
fric.results <- fric.results %>%
  mutate(reanalyzed=ifelse(set%in%datasets$set & model %in%c("lat","corr"),1,0))

#Model 1 = Fric Direct regression, all species
fric1<-fric.results %>%
  filter(model=="lat") %>%
  mutate(modelnum=1, modelname='SR-105') %>%
  select(name,region,onset.coef,onset.response,term.coef,term.response,modelnum,modelname)

#Model 3 = Fric Direct regression, reanalyzed species
fric3<-fric.results %>%
  filter(model=="lat", reanalyzed==1) %>%
  mutate(modelnum=3, modelname='SR-22') %>%
  select(name,region,onset.coef,onset.response,term.coef,term.response,modelnum,modelname)

#Model 2 = Fric residual regression, all species
fric2<-fric.results %>%
  filter(model=="corr") %>%
  mutate(modelnum=2, modelname='RR-105') %>%
  select(name,region,onset.coef,onset.response,term.coef,term.response,modelnum,modelname)

#Model 4 = Fric residual regression, reanalyzed species
fric4<-fric.results %>%
  filter(model=="corr", reanalyzed==1) %>%
  mutate(modelnum=4, modelname='RR-22') %>%
  select(name,region,onset.coef,onset.response,term.coef,term.response,modelnum,modelname)

#Model 5 = Reanalysis multiple regression
temp<-pivot_wider(filter(result, param=="rndLat"), id_cols =c(name, region),names_from=metric,values_from=c(param.est,param.
p, response) )
print("The reanalysis result table has fields:")
```

```
## [1] "The reanalysis result table has fields:"
```

```
names(result)
```

```
##  [1] "param.est" "param.se"  "param.t"   "param.p"   "param"     "metric"
##  [7] "name"      "region"    "n"         "response"
```

```
print("From which the following fields are created using pivot_wider:")
```

```
## [1] "From which the following fields are created using pivot_wider:"
```

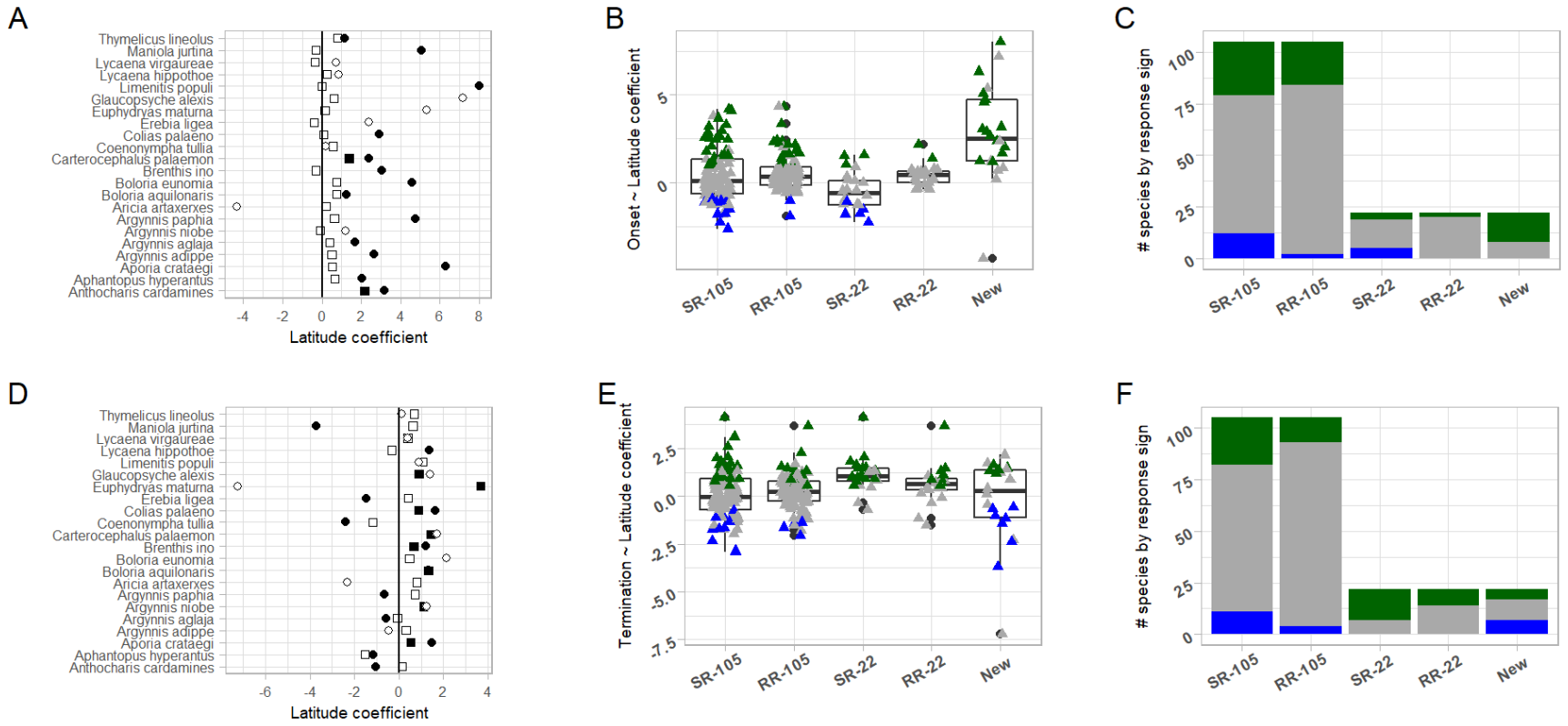
```
names(temp)
```

```
## [1] "name"                  "region"                "param.est_onset"
## [4] "param.est_termination" "param.p_onset"         "param.p_termination"
## [7] "response_onset"        "response_termination"
```

```
#Here we select the fields we need and name them to correspond to the Fric result tables
result5<-temp %>%
  select(name, region, onset.coef=param.est_onset, onset.response=response_onset, term.coef=param.est_termination, term.resp
onse=response_termination) %>%
  mutate(modelnum=5, modelname="New")
rm(temp)

#Combine all results into 1 data frame
result.compar<-as.data.frame(rbind(fric1,fric2,fric3,fric4,result5))
#This field is used to create stacked barplots
result.compar$s1<-1

##Create Figure 2: parameters
colorscheme<-c("blue", "darkgray", "darkgreen")
ts<-8
ar=2/3
ar1=1

#Panels A, D: compare coefficients
#Panel A: Onset coefficients
onset.sp<-ggplot(data=filter(result.compar, as.numeric(modelnum)>3), aes(x=name, y=onset.coef, shape=as.factor(modelnum), fi
ll=as.factor(onset.response))) +
  geom_point(color="black") +
  scale_shape_manual(values=c(22,21)) +
  scale_fill_manual(values=c("white","black")) +
  geom_hline(yintercept=0) +
  scale_y_continuous(breaks=seq(-8,8,2)) +
  labs(x="", y="Latitude coefficient") + coord_flip() +
  theme_light()  +
  theme(legend.position = "none", axis.title=element_text(size=ts-1),  axis.text=element_text(size=ts-2), aspect.ratio=ar1,
 plot.margin = margin(0.25, 0.25, 0.25, 0.25, unit = "cm"))
#Panel D: Termination coefficients
term.sp<-ggplot(data=filter(result.compar, as.numeric(modelnum)>3), aes(x=name, y=term.coef, shape=as.factor(modelnum), fill
=as.factor(term.response))) +
  geom_point(color="black") +
  scale_shape_manual(values=c(22,21)) +
  scale_fill_manual(values=c("black","white","black")) +
  geom_hline(yintercept=0) +
  scale_y_continuous(breaks=seq(-8,8,2)) +
  labs(x="", y="Latitude coefficient") + coord_flip() +
```

```
  theme_light()  +
  theme(legend.position = "none", axis.title=element_text(size=ts-1),  axis.text=element_text(size=ts-2), aspect.ratio=ar1,
 plot.margin = margin(0.25, 0.25, 0.25, 0.25, unit = "cm"))

#Panels B, E: response boxplots
#Panel B: Onset
onset.c<-ggplot(data=result.compar, aes(x=reorder(modelname,modelnum), y=onset.coef)) +
  geom_boxplot(aes(group=reorder(modelname,modelnum))) +
  geom_jitter(data=filter(result.compar),  aes(x=reorder(modelname,modelnum), y=onset.coef, color=as.factor(onset.respons
e)), width=0.2, height=0, shape=17) +
  labs(x="", y="Onset ~ Latitude coefficient") +
  scale_color_manual(values=colorscheme) +
  theme_light() +
  theme(legend.position = "none", axis.title=element_text(size=ts-1, face="plain"), axis.text=element_text(size=ts-1, angle=
30, hjust=0.8, face="bold"), aspect.ratio=ar, plot.margin = margin(0.25, 0.25, 0.25, 0.25, unit = "cm"))
#Panel E: termination
term.c<-ggplot(data=result.compar, aes(x=reorder(modelname,modelnum), y=term.coef)) +
  geom_boxplot(aes(group=reorder(modelname,modelnum))) +
  geom_jitter(data=filter(result.compar),  aes(x=reorder(modelname,modelnum), y=term.coef, color=as.factor(term.response)),
 width=0.2, height=0, shape=17) +
  labs(x="", y="Termination ~ Latitude coefficient") +
  scale_color_manual(values=colorscheme) +
  theme_light() +
  theme(legend.position = "none", axis.title=element_text(size=ts-1, face="plain"), axis.text=element_text(size=ts-1, angle=
30, hjust=0.8, face="bold"), aspect.ratio=ar, plot.margin = margin(0.25, 0.25, 0.25, 0.25, unit = "cm"))

#Panels C, F: stacked barplots
#Panel c: Onset responses
onset.st<-ggplot(data=result.compar, aes(x=(reorder(modelname,modelnum)), y=s1, fill=as.factor(onset.response))) +
  geom_bar(position=position_stack(reverse=T), stat="identity")  +
  scale_fill_manual(values=colorscheme) +
  labs(x="", y="# species by response sign") + theme_light() +
  theme(legend.position = "none", axis.title=element_text(size=ts-1, face="plain"), axis.text=element_text(size=ts-1, angle=
30, hjust=0.8, face="bold"), aspect.ratio=ar, plot.margin = margin(0.25, 0.25, 0.25, 0.25, unit = "cm"))
#Panel F: Termination responses
term.st<-ggplot(data=result.compar, aes(x=reorder(modelname,modelnum), y=s1, fill=as.factor(term.response))) +
  geom_bar(position=position_stack(reverse=T), stat="identity") +
  scale_fill_manual(values=colorscheme) +
  theme_light() + labs(x="", y="# species by response sign") +
  theme(legend.position = "none", axis.title=element_text(size=ts-1, face="plain"), axis.text=element_text(size=ts-1, angle=
30, hjust=0.8, face="bold"), aspect.ratio=ar, plot.margin = margin(0.25, 0.25, 0.25, 0.25, unit = "cm"))
```

```
##Combine panels into Figure 2:
p1<-onset.sp+labs(tag="A")
p2<-onset.c+labs(tag="B")
p3<-onset.st+labs(tag="C")
p4<-term.sp+labs(tag="D")
p5<-term.c+labs(tag="E")
p6<-term.st+labs(tag="F")

#pdf_filename2<-("outputs/LarsenShirey_Fig2.pdf")
grid.arrange(ncol=3, grobs=list(p1, p2, p3, p4, p5, p6),  widths=c(1.2,1,1), bottom="These figures show the difference betwe
en the results of our reanalysis ('New') and Fric et al.'s \n results (SR=Single Regression, RR=Regression of Residuals; 105
= all 105 datasets, 22 =  reanalyzed datasets).")
```



These figures show the difference between the results of our reanalysis ('New') and Fric et al.'s results (SR=Single Regression, RR=Regression of Residuals; 105 = all 105 datasets, 22 = reanalyzed datasets).

```
#fig2<-grid.arrange(ncol=3, grobs=list(p1, p2, p3, p4, p5, p6), widths=c(1.05,1,1), top="\n\n", bottom="\n\n", left="\n\n",
 right="\n\n", width=10, height=5)
#ggsave(pdf_filename2, arrangeGrob(fig2, nrow=1), width=10, height=6, scale=1, dpi=600,units="in")
```

*

Create statistics for results table (Supplemental Table 2)

This code outputs a results table that is a partial Supplemental Table 2 - it is currently missing the ‘year’ analyses from Fric et al., as our focus is on the latitudinal patterns.

```
rm(onset.c,onset.sp,onset.st,onsetpheno,p1,p2,p3,p4,p5,p6,axeso,term.c,term.sp,term.st,axest,rowi.data)
#Here we are building supplemental table 2 with fields: name_resultsfile, region, phenometric, indep.variable, Fric_singleRegression_Sign, Fric_resid.regress_sign, Reanalysis_sign, Reanalysis_p, Reanalysis_coefficient, Fric_resid.regress_p, Fric_resid.regress_coefficient, Fric_singleRegression_p, Fric_singleRegression_coefficient

#Reanalysis results
table2<-result %>%
  select(name_resultsfile=name, region, phenometric=metric, indep.variable=param, Reanalysis_sign=response,Reanalysis_p=param.p,Reanalysis_coef=param.est) %>%
  mutate(indep.variable=ifelse(indep.variable=="rndLat","latitude","year"), unit=paste(name_resultsfile,region,indep.variable,sep="."))

#Fric results
fric.results <- fric.results %>%
  filter(set%in%datasets$set) %>%
  mutate(regtype=ifelse(model%in%c("lat","year"),"sr","rr"), param=ifelse(model%in%c("lat","corr"),"latitude","year"))

fric.wide<-fric.results %>%
  pivot_wider(
    id_cols = c(name, region, param),
    names_from = regtype,
    names_sep = ".",
    values_from = c(onset.p_mean, onset.coef, onset.response, term.p_mean, term.coef, term.response)
  ) %>% mutate(unit=paste(name,region,param,sep="."))

#combine
onset2<-merge(filter(table2,phenometric=="onset"),fric.wide[,c(4:9,16)],by="unit")
onset2<-onset2 %>% select(name_resultsfile:Reanalysis_coef,Fric_SR_sign=onset.response.sr,Fric_SR_p=onset.p_mean.sr,Fric_SR_coef=onset.coef.sr,Fric_RR_sign=onset.response.rr,Fric_RR_p=onset.p_mean.rr,Fric_RR_coef=onset.coef.rr)
term2<-merge(filter(table2,phenometric=="termination"),fric.wide[,c(10:16)],by="unit")
term2<-term2 %>% select(name_resultsfile:Reanalysis_coef,Fric_SR_sign=term.response.sr,Fric_SR_p=term.p_mean.sr,Fric_SR_coef=term.coef.sr,Fric_RR_sign=term.response.rr,Fric_RR_p=term.p_mean.rr,Fric_RR_coef=term.coef.rr)
table2<-bind_rows(onset2,term2)

summary(table2)
```

```
##  name_resultsfile      region          phenometric        indep.variable
##  Length:88          Length:88          Length:88          Length:88
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Reanalysis_sign    Reanalysis_p      Reanalysis_coef    Fric_SR_sign
##  Min.   :-1.0000   Min.   :0.000000   Min.   :-7.2315   Min.   :-1.0000
##  1st Qu.: 0.0000   1st Qu.:0.008117   1st Qu.:-0.1581   1st Qu.: 0.0000
##  Median : 0.0000   Median :0.093213   Median : 0.1699   Median : 0.0000
##  Mean   : 0.1136   Mean   :0.224371   Mean   : 0.6807   Mean   : 0.1705
##  3rd Qu.: 1.0000   3rd Qu.:0.433347   3rd Qu.: 1.3533   3rd Qu.: 0.0000
##  Max.   : 1.0000   Max.   :0.999200   Max.   : 8.0260   Max.   : 1.0000
##    Fric_SR_p          Fric_SR_coef      Fric_RR_sign       Fric_RR_p
##  Min.   :0.0000006   Min.   :-2.2694   Min.   :-1.0000   Min.   :0.0001309
##  1st Qu.:0.0277003   1st Qu.:-0.3598   1st Qu.: 0.0000   1st Qu.:0.1256775
##  Median :0.1713357   Median : 0.1184   Median : 0.0000   Median :0.3625378
##  Mean   :0.2948753   Mean   : 0.1714   Mean   : 0.1136   Mean   :0.4010582
##  3rd Qu.:0.5142356   3rd Qu.: 0.7875   3rd Qu.: 0.0000   3rd Qu.:0.6475207
##  Max.   :0.9870071   Max.   : 4.1062   Max.   : 1.0000   Max.   :0.9996094
##   Fric_RR_coef
##  Min.   :-1.5254
##  1st Qu.:-0.1506
##  Median : 0.1700
##  Mean   : 0.2557
##  3rd Qu.: 0.6302
##  Max.   : 3.6603
```

```
#write.csv(table2,file="outputs/LarsenShirey_SuppTable2.csv")
```

This is the end of the main analysis. Code for Supplemental Figure 1 has been moved to a separate Rmarkdown file.

Here is the model with our data curation but Fric et al.'s estimation of onset & termiation. Here we don't use Weibull-based phenometrics; instead we take the min and max days per year-latitude with at least 10 records.

```
#Loop through datasets, run model for phenology by species & region, and store LM parameters
onsetpheno<-list()
termpheno<-list()
onset1<-NULL
term1<-NULL
axeso<-NULL
axest<-NULL

for(rowi in 1:nrow(datasets)) {
  pheno.rowi<-reanalysis.data %>%
    filter(name==datasets$name[rowi], region==datasets$region[rowi]) %>%
    group_by(name, region, rndLat, year) %>%
    summarize(onset=min(doy, na.rm=T),term=max(doy,na.rm=T))
#estimate model params for onset
  onset.lm<-summary(lm(onset~rndLat+year, data=pheno.rowi))$coefficients #estimate model params for termination
  term.lm<-summary(lm(term~rndLat+year, data=pheno.rowi))$coefficients
#store
  onsetpheno[[rowi]]<-onset.lm
  termpheno[[rowi]]<-term.lm

#onset
  temponset<-matrix(unlist(onset.lm[c(2:3),]), ncol=4, byrow=F)
  onset1<-rbind(onset1, temponset)
  axeso<-c(axeso,row.names(onset.lm)[c(2:3)])
#termination
  tempterm<-matrix(unlist(term.lm[c(2:3),]), ncol=4, byrow=F)
  term1<-rbind(term1, tempterm)
  axest<-c(axest,row.names(term.lm)[c(2:3)])
  rm(pheno.rowi,onset.lm,term.lm,temponset,tempterm)
  }
```

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

```
#Create results dataframes: onset
onset1<-as.data.frame(onset1)
colnames(onset1)<-c("param.est","param.se","param.t","param.p")
onset1$param<-axeso
onset1$metric<-"onset"
onset1$name<-rep(datasets$name, each=2)
onset1$region<-rep(datasets$region, each=2)
onset1$n<-rep(datasets$n, each=2)

#Create results dataframes: termination
term1<-as.data.frame(term1)
colnames(term1)<-c("param.est","param.se","param.t","param.p")
term1$param<-axest
term1$metric<-"termination"
term1$name<-rep(datasets$name, each=2)
term1$region<-rep(datasets$region, each=2)
term1$n<-rep(datasets$n, each=2)

result.ex<-bind_rows(onset1, term1)
result.ex<-result.ex %>%
  mutate(response=ifelse(param.p<0.05,ifelse(param.est>0,1,-1),0))
(result.ex<-result.ex %>% group_by(param, metric, response) %>% tally())
```

```
## # A tibble: 10 x 4
## # Groups:   param, metric [4]
##    param  metric      response     n
##    <chr>  <chr>          <dbl> <int>
##  1 rndLat onset              0     7
##  2 rndLat onset              1    15
##  3 rndLat termination       -1     6
##  4 rndLat termination        0    11
##  5 rndLat termination        1     5
##  6 year   onset             -1    11
##  7 year   onset              0    10
##  8 year   onset              1     1
##  9 year   termination        0    14
## 10 year   termination        1     8
```

These response patterns are broadly comparable to those in our main reanalysis, and do not validate the original analysis in any way.

Author notes - Future updates should:
Remove variables when we’re done with them
See if we can suppress geom_smooth() messages

End of file.