

warmXtrophic Project: Seed Set Phenology Analyses

Moriah Young

December 17, 2021

Load in packages & data

```
# clear all existing data
rm(list = ls())

# Load packages
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.3      v purrr    0.3.4
## v tibble   3.0.6      v dplyr    1.0.4
## v tidyr    1.1.2      v stringr  1.4.0
## v readr    1.4.0      vforcats  0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()

library(ggplot2)
library(lme4)

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyR':
## 
##     expand, pack, unpack

library(lmerTest)

##
## Attaching package: 'lmerTest'
```

```

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step

library(emmeans)
library(vegan)

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-7

library(car)

## Loading required package: carData

## Registered S3 methods overwritten by 'car':
##   method           from
##   influence.merMod      lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod      lme4
##   dfbetas.influence.merMod     lme4

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some

library(rstatix)

##
## Attaching package: 'rstatix'

## The following object is masked from 'package:stats':
##
##     filter

```

```

library(scales)

##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

library(fitdistrplus)

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:rstatix':
##
##     select

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: survival

library(moments) # for calculating skewness of data
library(ggpubr)
library(jtools) # summ() function
library(predictmeans)

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:lme4':
##
##     lmList

## The following object is masked from 'package:dplyr':
##
##     collapse

## Loading required package: parallel

```

```

library(olsrr)

##
## Attaching package: 'olsrr'

## The following object is masked from 'package:MASS':
##
##     cement

## The following object is masked from 'package:datasets':
##
##     rivers

library(car)
library(fitdistrplus)
library(ggpubr)
library(interactions)
library(sjPlot)

## Learn more about sjPlot with 'browseVignettes("sjPlot")'.

library(effects)

##
## Use the command
##     lattice::trellis.par.set(effectsTheme())
## to customize lattice options for effects plots.
## See ?effectsTheme for details.

library(glmmTMB)
library(GGally)  # ggpairs() function

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

##
## Attaching package: 'GGally'

## The following object is masked from 'package:emmeans':
##
##     pigs

library(bbmle)  # AICtab() function

## Loading required package: stats4

##
## Attaching package: 'bbmle'

```

```

## The following object is masked from 'package:dplyr':
##
##     slice

# Set working directory
Sys.getenv("L1DIR")

## [1] "/Volumes/GoogleDrive/Shared drives/SpaCE_Lab_warmXtrophic/data/L1"

L1_dir <- Sys.getenv("L1DIR")
L2_dir <- Sys.getenv("L2DIR")

# Set ggplot2 plotting This code for ggplot2 sets the theme to mostly black and
# white (Arial font, and large font, base size=24)
theme_set(theme_bw(14))
theme_update(axis.text.x = element_text(size = 12, angle = 90), axis.text.y = element_text(size = 12))

# Read in data
sd_species <- read.csv(file.path(L2_dir, "phenology/final_sd_species_L2.csv")) # species level data
sd_plot <- read.csv(file.path(L2_dir, "phenology/final_sd_plot_L2.csv")) # plot level data

# get rid of 'X' column that shows up
sd_species$X <- NULL
sd_plot$X <- NULL
# take a look at the data to see if looks good
View(sd_species)
View(sd_plot)

sd_species$julian_min <- as.numeric(as.integer(sd_species$julian_min))
sd_plot$julian_min <- as.numeric(as.integer(sd_plot$julian_min))

# Order warmed and ambient so that warmed shows up first in plotting (and is
# default is red = warmed; blue = ambient). First make it a factor
sd_species$state <- as.factor(sd_species$state)
levels(sd_species$state)

## [1] "ambient" "warmed"

# [1] 'ambient' 'warmed'
sd_species$state <- factor(sd_species$state, levels(sd_species$state)[c(2, 1)])
levels(sd_species$state)

## [1] "warmed"   "ambient"

# [1] 'warmed' 'ambient'

# again for plot level data
sd_plot$state <- as.factor(sd_plot$state)
levels(sd_plot$state)

## [1] "ambient" "warmed"

```

```

# [1] 'ambient' 'warmed'
sd_plot$state <- factor(sd_plot$state, levels(sd_plot$state)[c(2, 1)])
levels(sd_plot$state)

## [1] "warmed" "ambient"

# [1] 'warmed' 'ambient'

# make the year_factor a factor
sd_plot$year_factor <- as.factor(sd_plot$year_factor)
sd_species$year_factor <- as.factor(sd_species$year_factor)

umbs_sd_plot <- subset(sd_plot, site == "umbs") # pull out umbs only data at plot level
umbs_sd_spp <- subset(sd_species, site == "umbs") # pull out umbs only data at species level
kbs_sd_plot <- subset(sd_plot, site == "kbs") # pull out kbs only data at plot level
kbs_sd_spp <- subset(sd_species, site == "kbs") # pull out kbs only data at species level

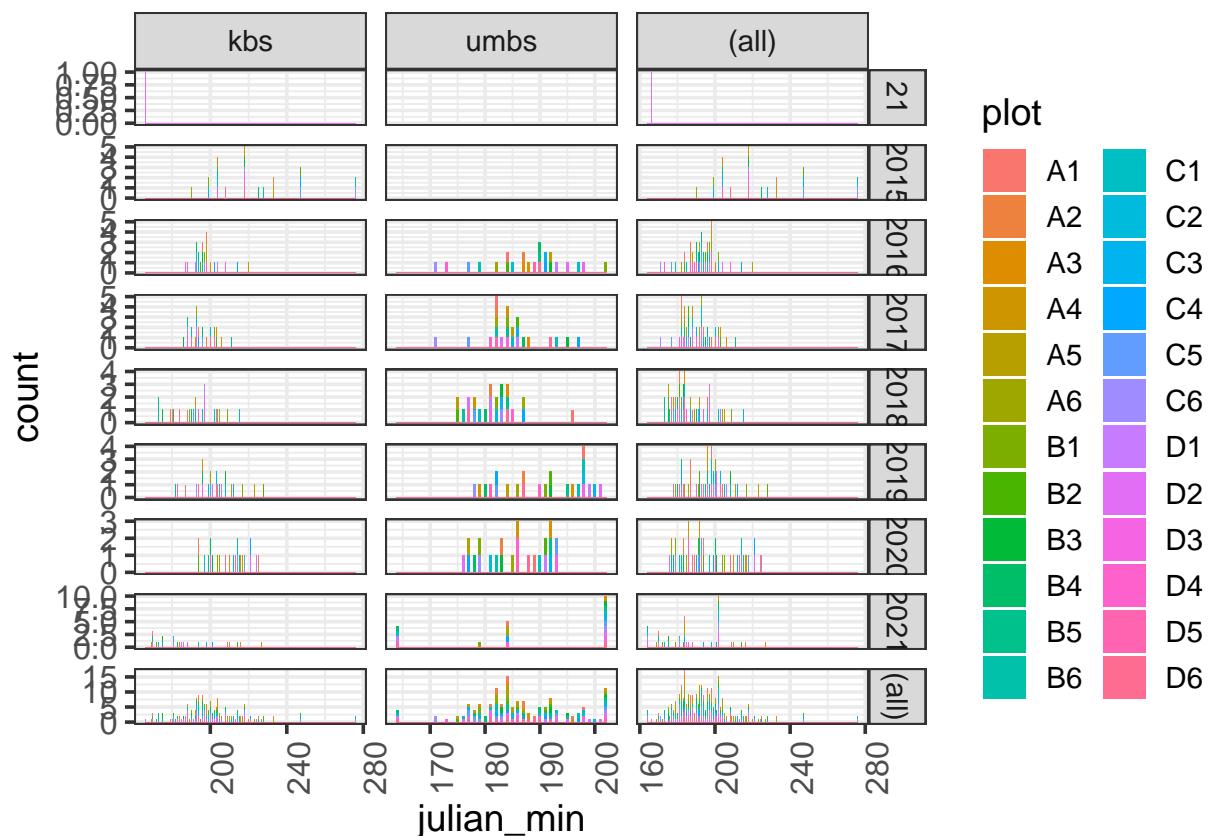
```

Data exploration for minimum (first) Julian date of seed set at the PLOT for both sites

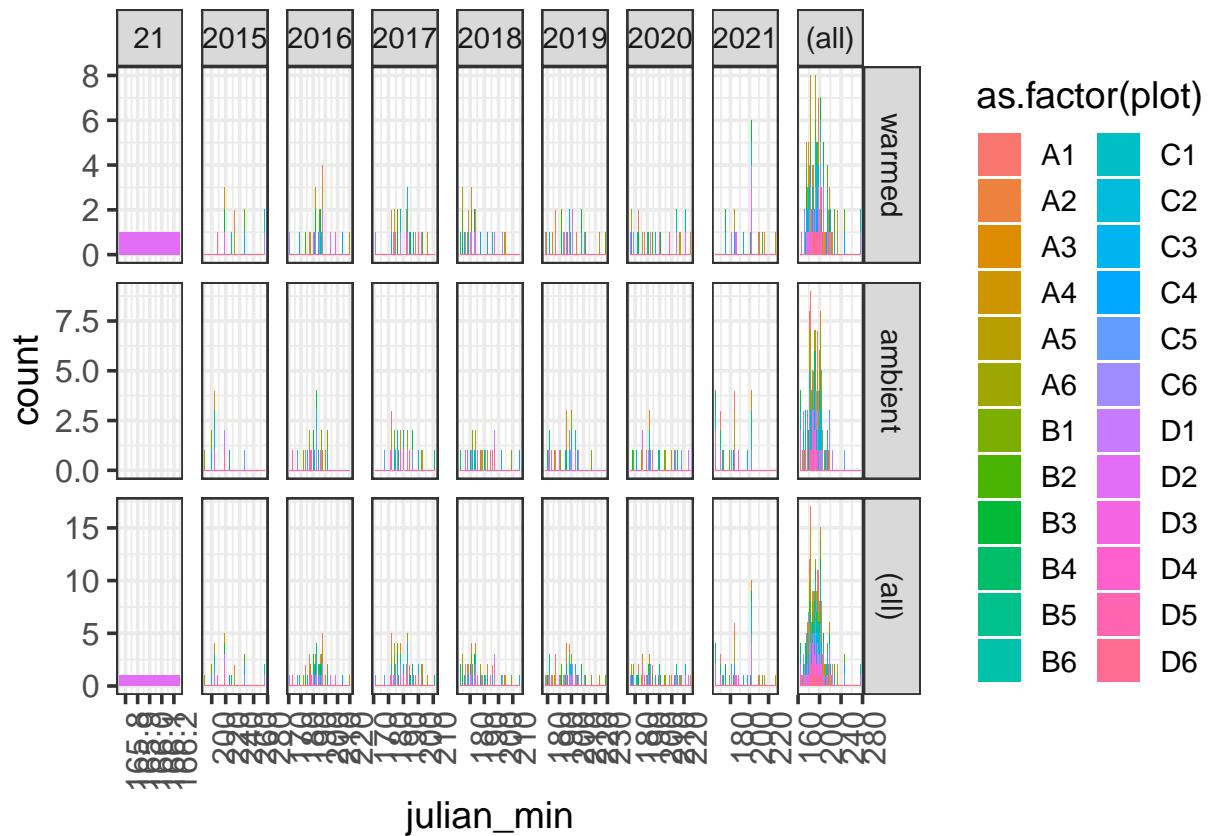
```

# Visualizing avg minimum Julian date for both sites at the PLOT LEVEL
ggplot(sd_plot, aes(julian_min, fill = plot)) + geom_histogram(binwidth = 0.5) +
  facet_grid(year ~ site, margins = TRUE, scales = "free")

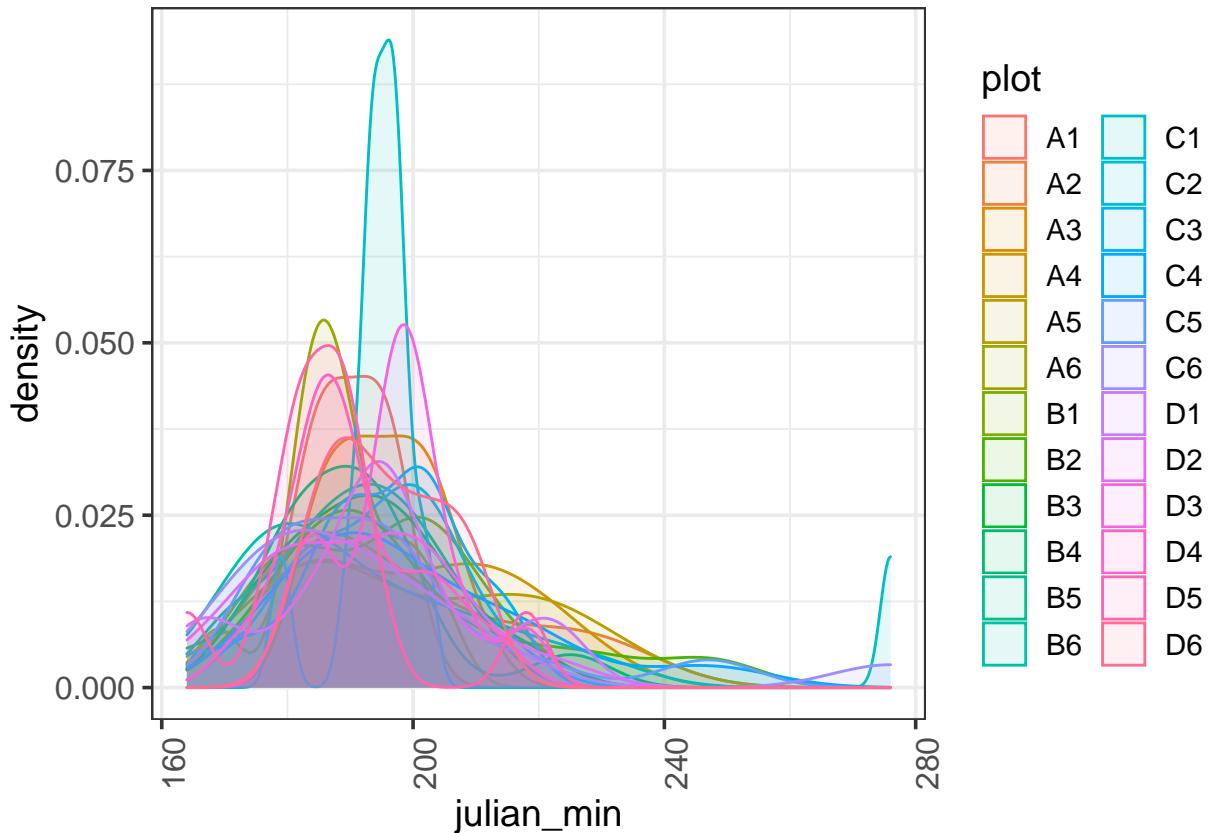
```



```
ggplot(sd_plot, aes(julian_min, fill = as.factor(plot))) + geom_histogram(binwidth = 0.5) +
  facet_grid(state ~ year, margins = TRUE, scales = "free")
```



```
ggplot(sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



```
ggplot(sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year)
```

```
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
## Warning: Groups with fewer than two data points have been dropped.
```



```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```

```

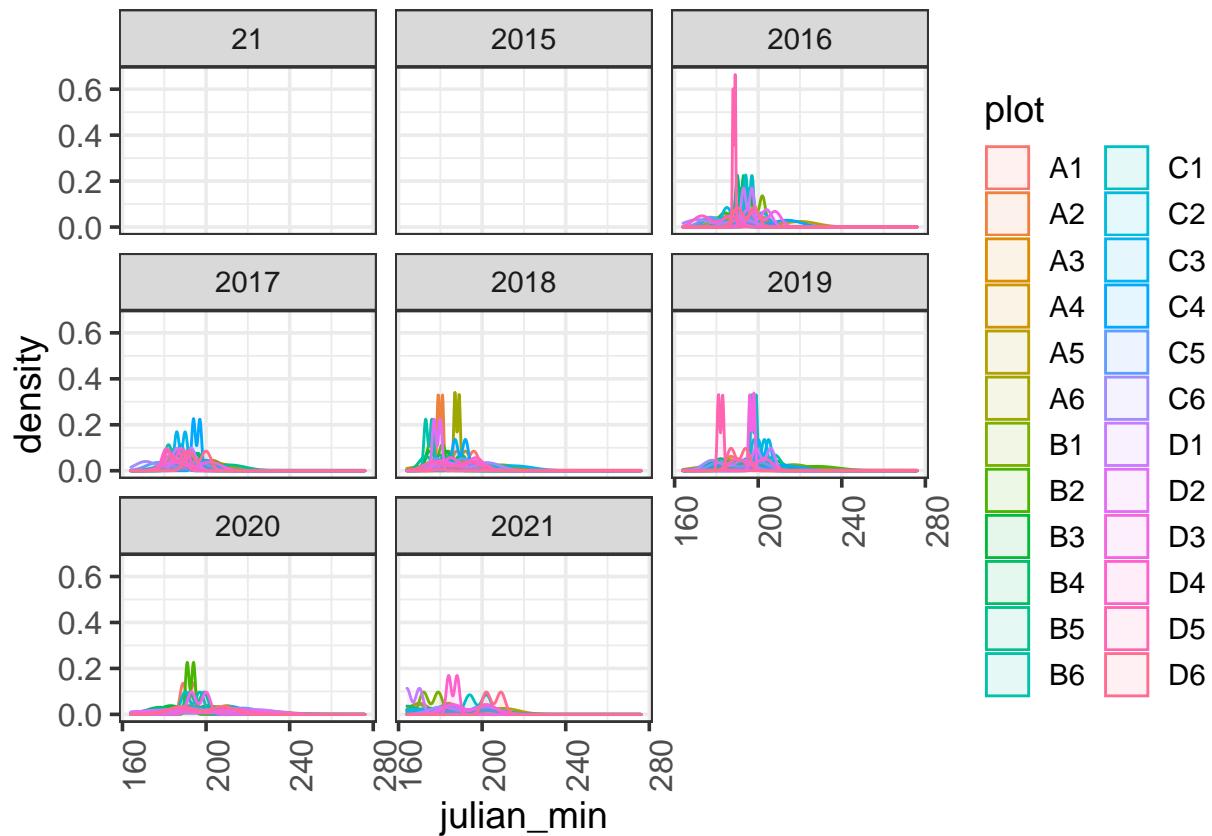
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

```



```

ggplot(sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)

```

```

## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.

```



```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

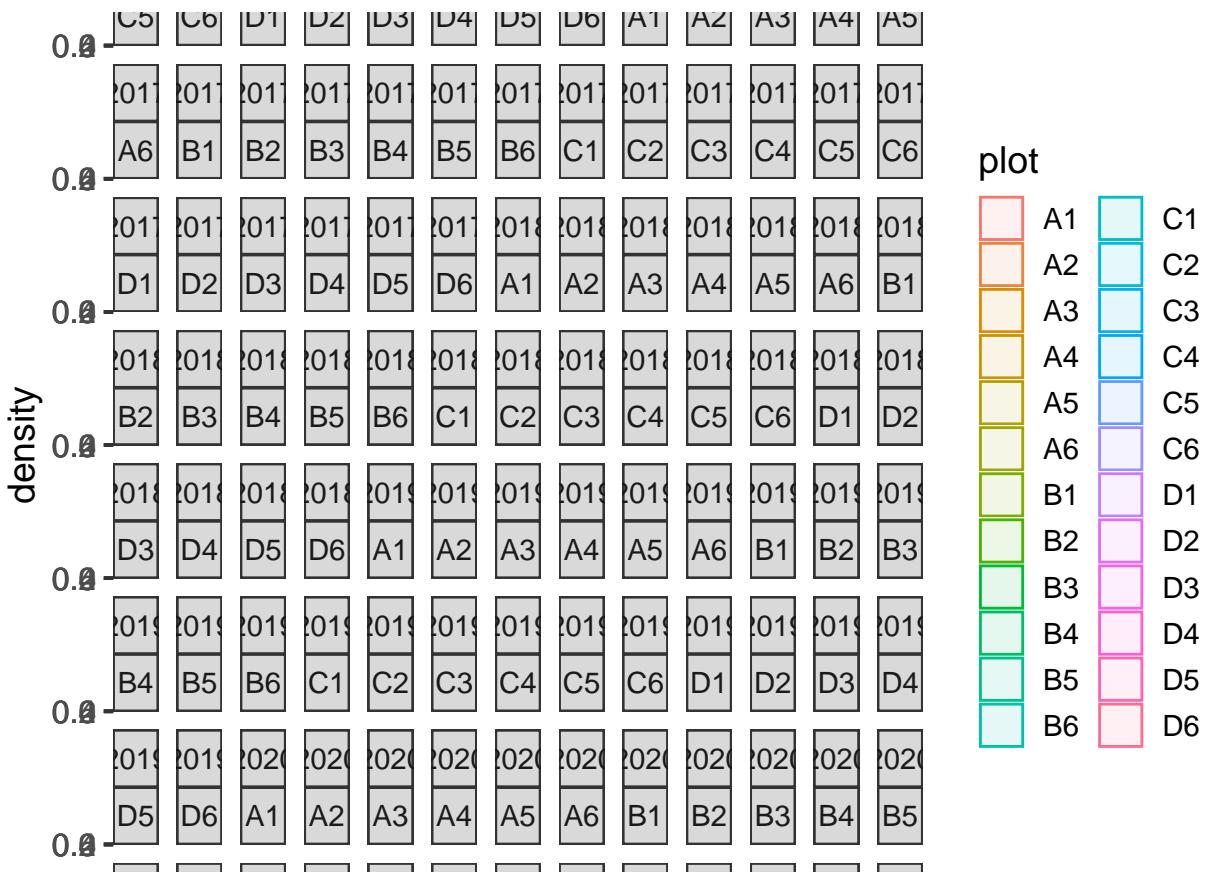
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

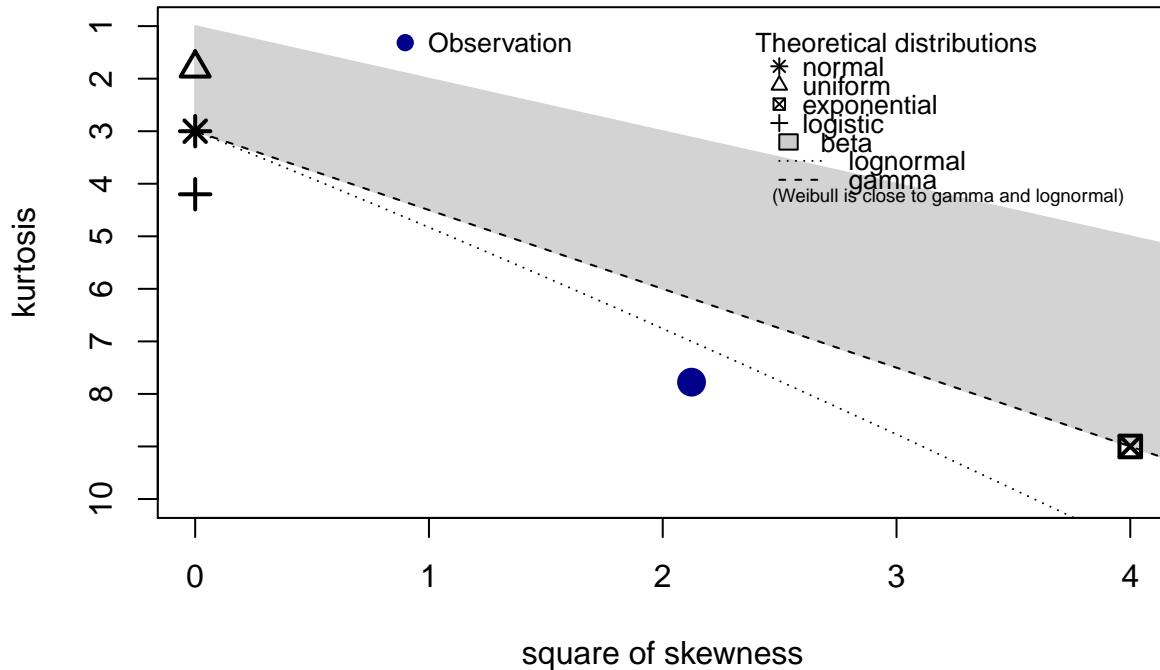
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```



```
descdist(sd_plot$julian_min, discrete = FALSE)
```

Cullen and Frey graph

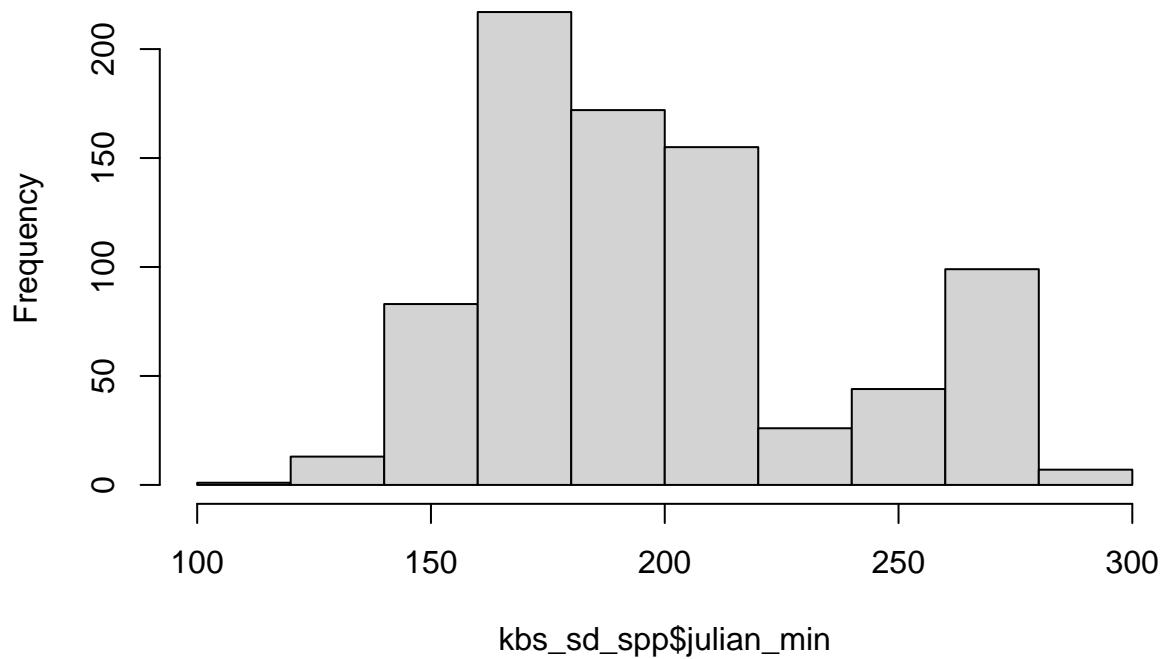


```
## summary statistics
## -----
## min: 164   max: 276
## median: 193
## mean: 194.2361
## estimated sd: 15.68764
## estimated skewness: 1.45713
## estimated kurtosis: 7.776184
```

**KBS SPECIES LEVEL - Looking at MINIMUM JULIAN DAY
(aka date of observed first seed set)**

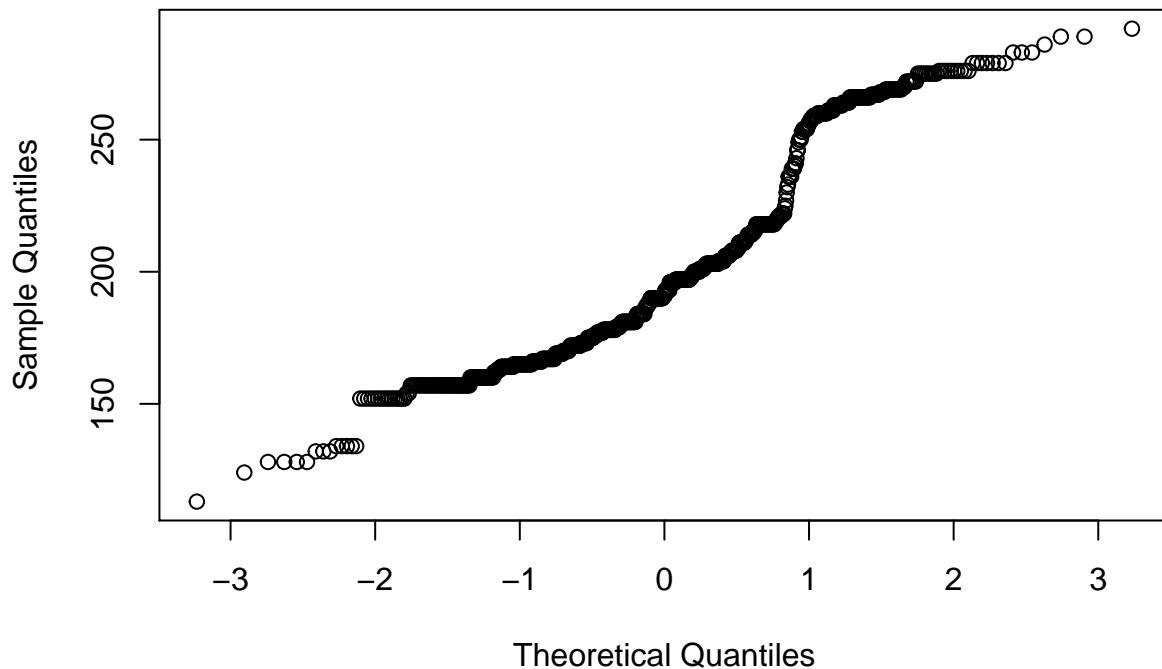
```
### KBS ####
hist(kbs_sd_spp$julian_min)
```

Histogram of kbs_sd_spp\$julian_min



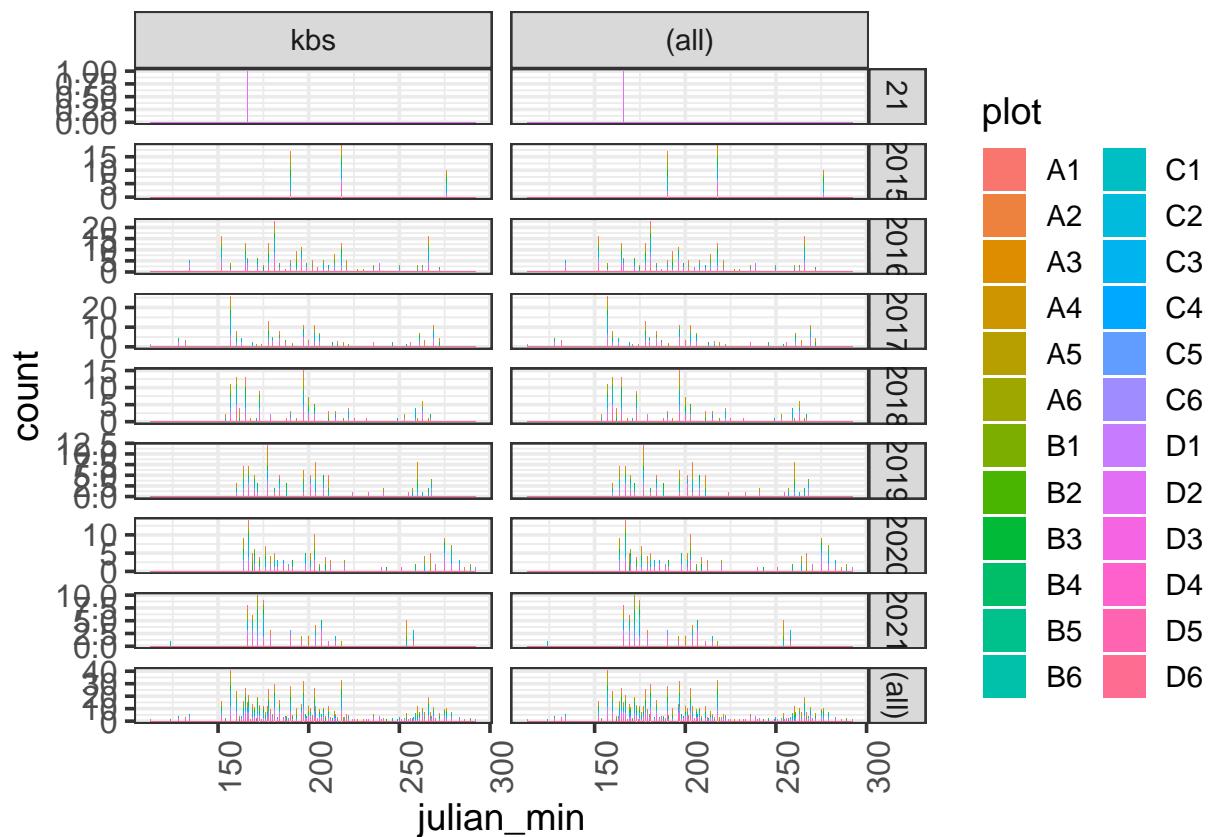
```
qqnorm(kbs_sd_spp$julian_min)
```

Normal Q-Q Plot

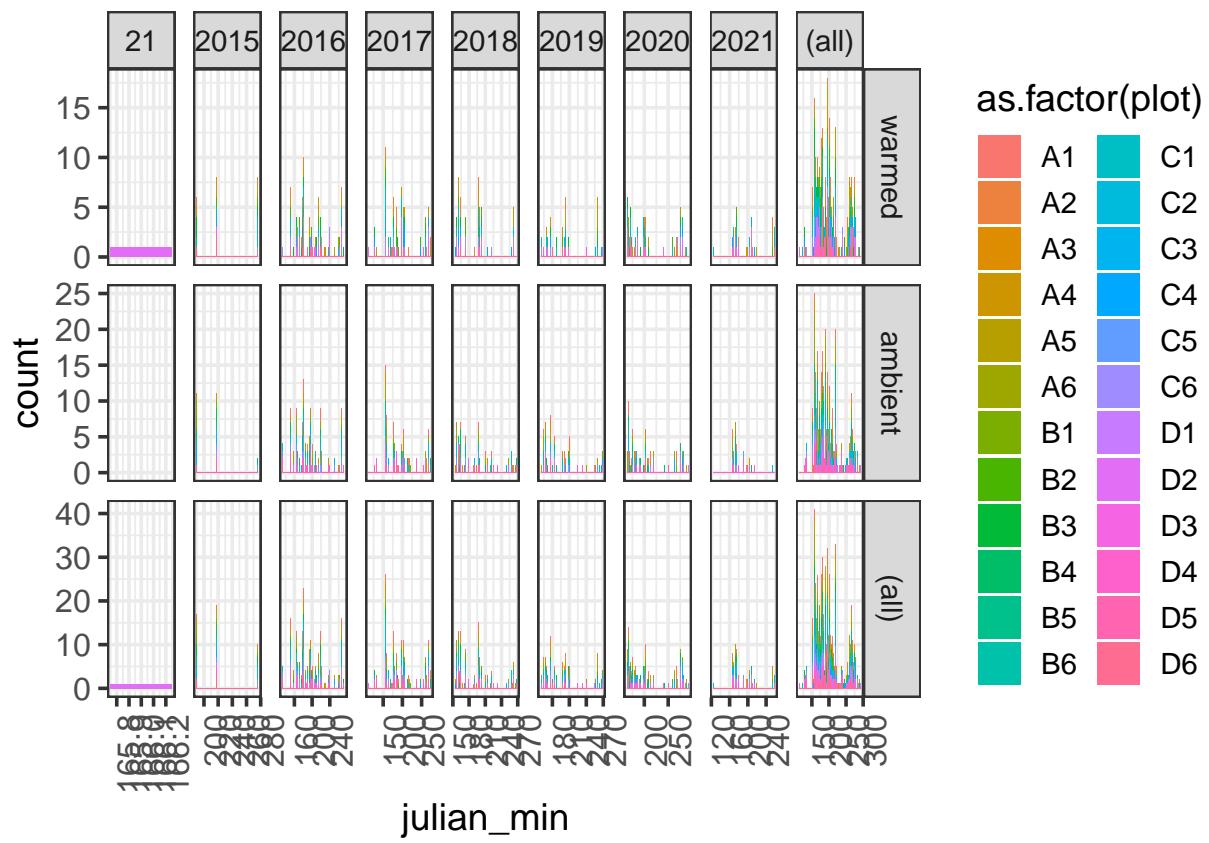


```
shapiro.test(kbs_sd_spp$julian_min) # pvalue is < 0.05 so we reject the null hypothesis that the data
```

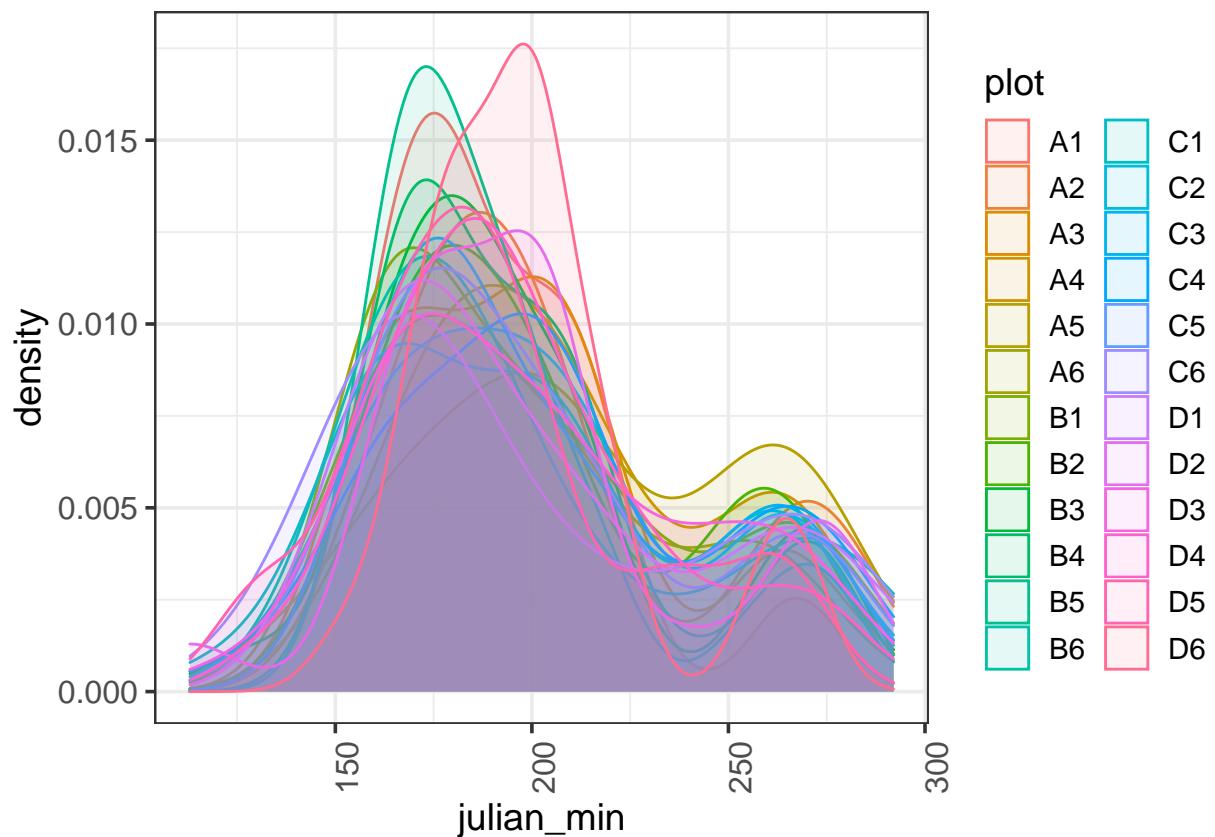
```
##  
## Shapiro-Wilk normality test  
##  
## data: kbs_sd_spp$julian_min  
## W = 0.91991, p-value < 2.2e-16  
  
# Visualizing first Julian date for kbs at the SPECIES LEVEL  
ggplot(kbs_sd_spp, aes(julian_min, fill = plot)) + geom_histogram(binwidth = 0.5) +  
  facet_grid(year ~ site, margins = TRUE, scales = "free")
```



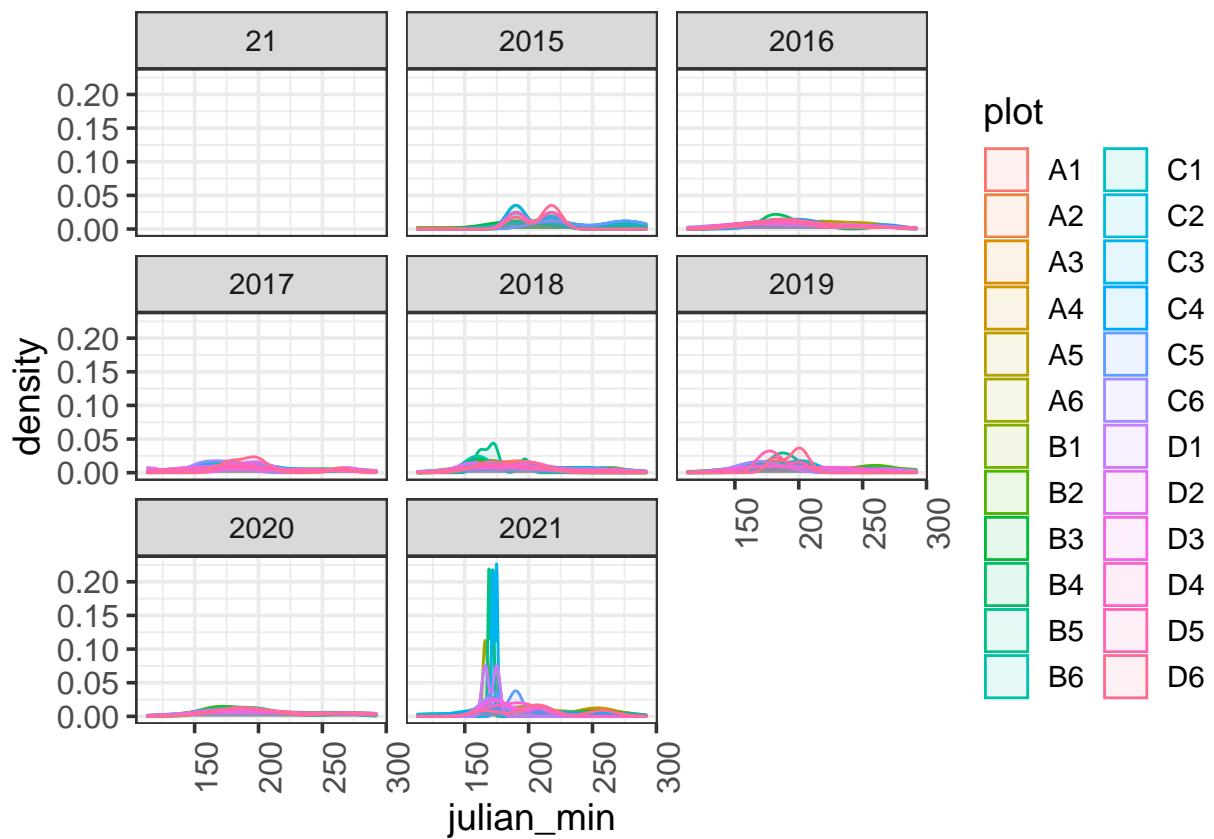
```
ggplot(kbs_sd_spp, aes(julian_min, fill = as.factor(plot))) + geom_histogram(binwidth = 0.5) +  
  facet_grid(state ~ year, margins = TRUE, scales = "free")
```



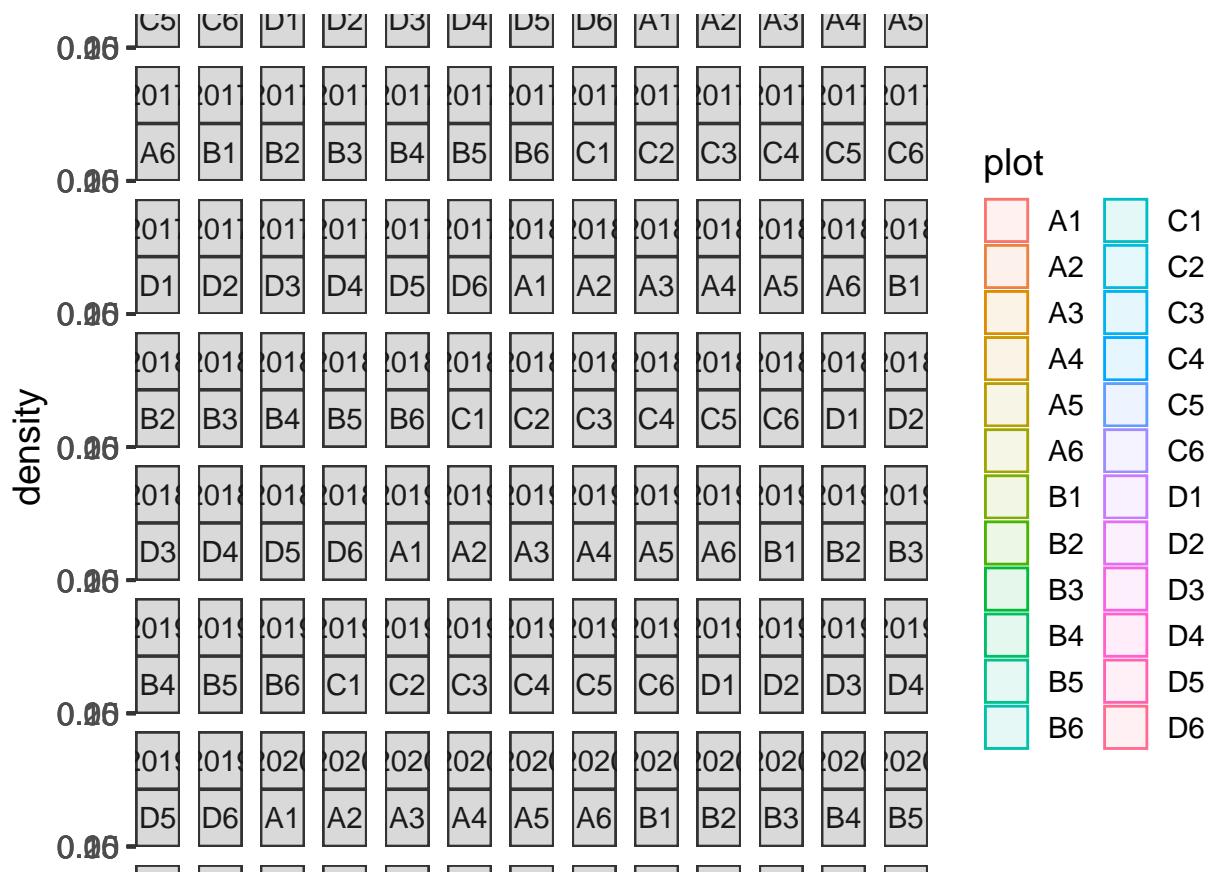
```
ggplot(kbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



```
ggplot(kbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year)
```

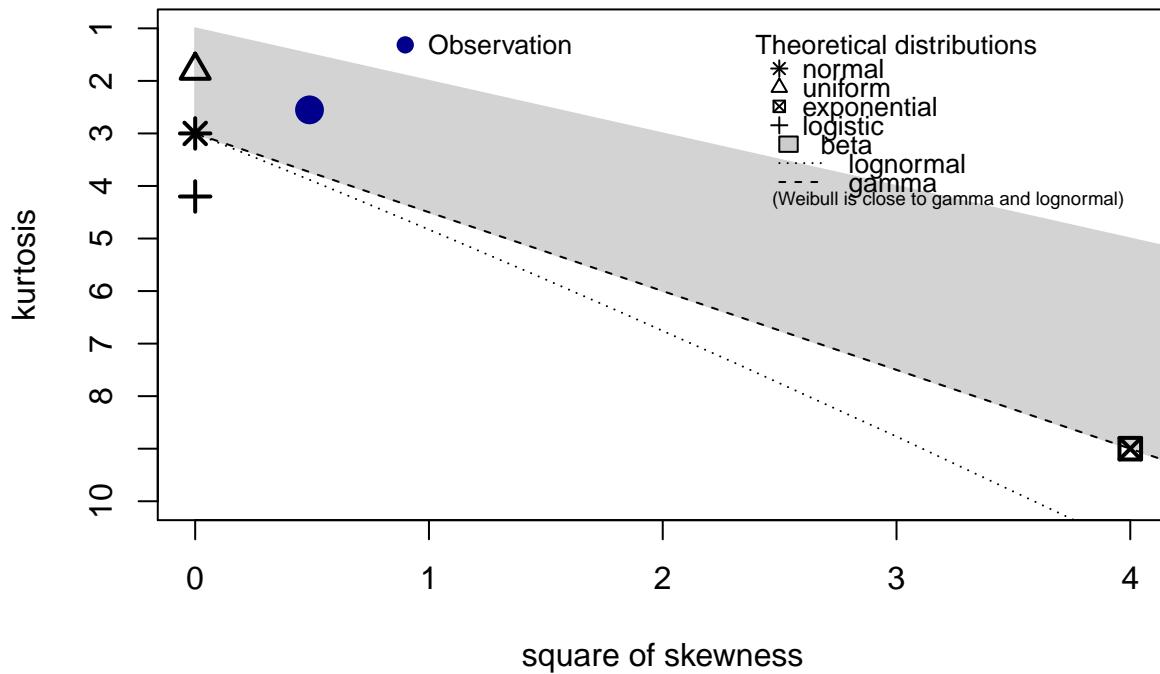


```
ggplot(kbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



```
# Exploring distributions  
descdist(kbs sd spp$julian min, discrete = FALSE)
```

Cullen and Frey graph

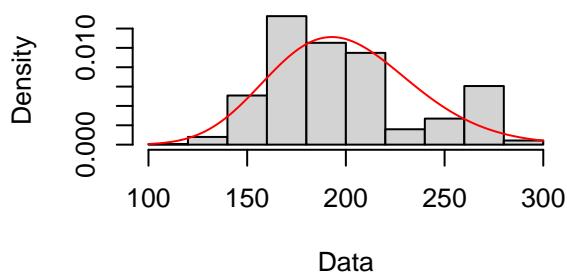
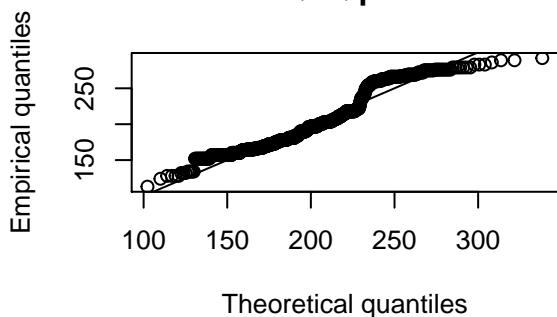
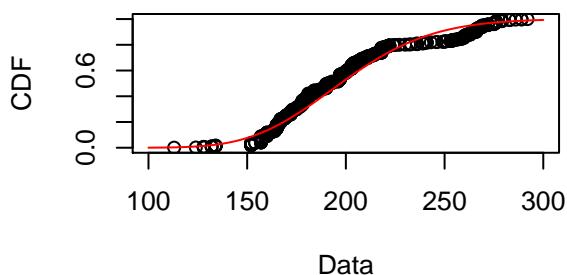
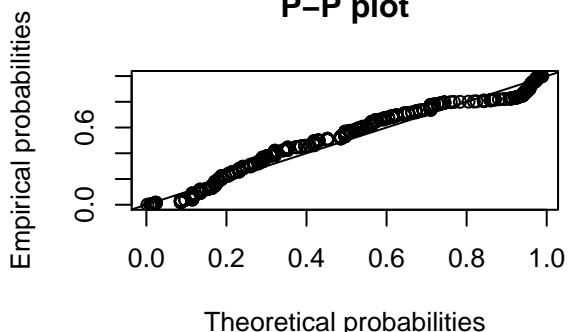


```

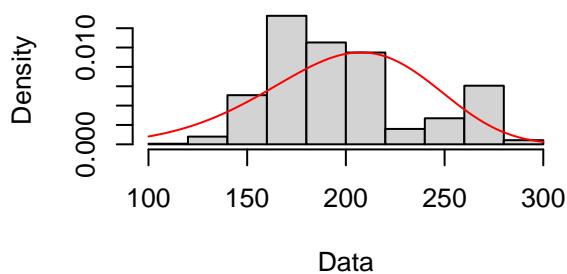
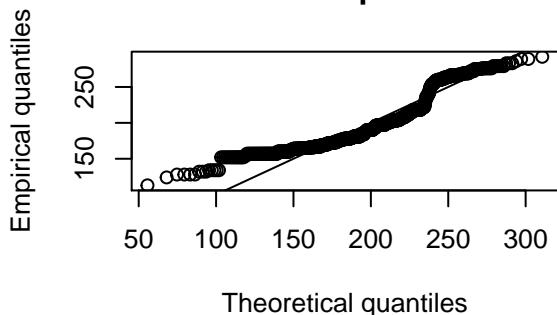
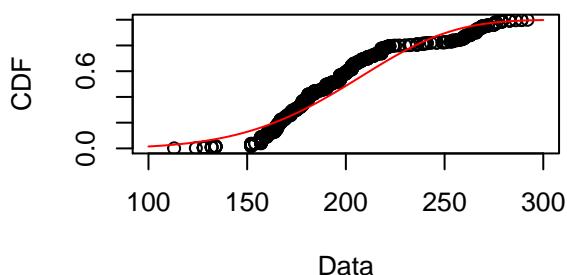
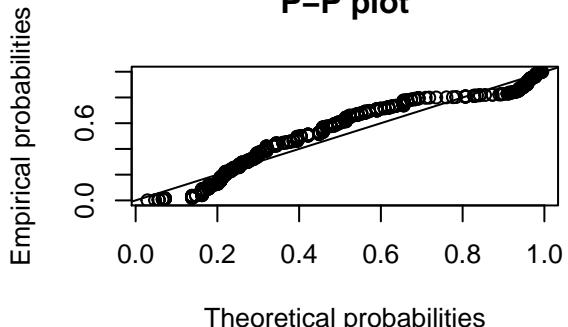
## summary statistics
## -----
## min: 113  max: 292
## median: 191
## mean: 199.5165
## estimated sd: 37.4115
## estimated skewness: 0.6993432
## estimated kurtosis: 2.551332

# Gamma distribution
fit.gamma <- fitdist(kbs_sd_spp$julian_min, "gamma")
plot(fit.gamma)

```

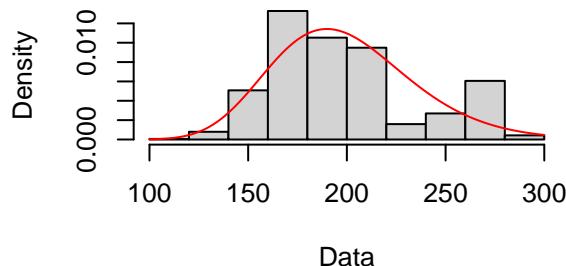
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

```
# Weibull distribution
fit.weibull <- fitdist(kbs_sd_spp$julian_min, "weibull")
plot(fit.weibull)
```

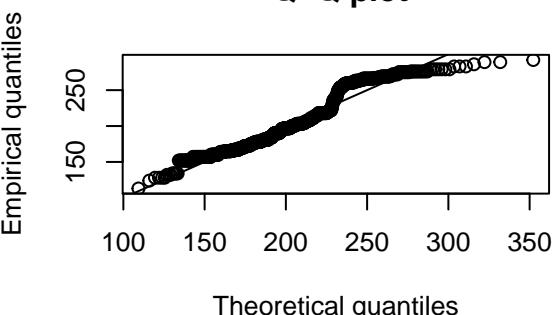
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

```
# Lognormal distribution
fit.ln <- fitdist(kbs_sd_spp$julian_min, "lnorm")
plot(fit.ln)
```

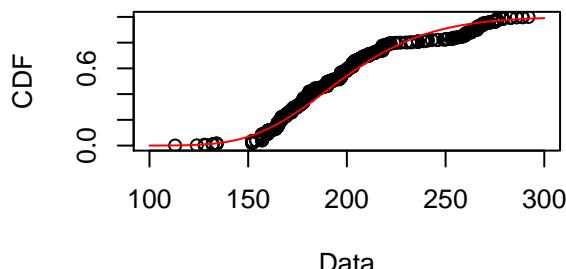
Empirical and theoretical dens.



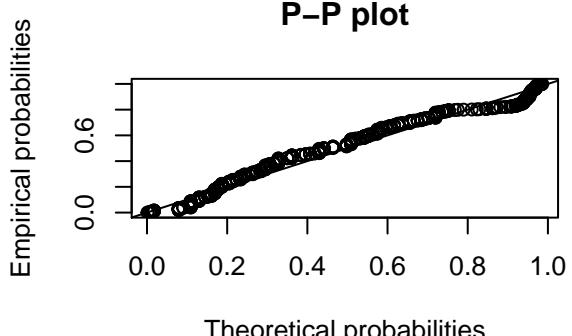
Q-Q plot



Empirical and theoretical CDFs

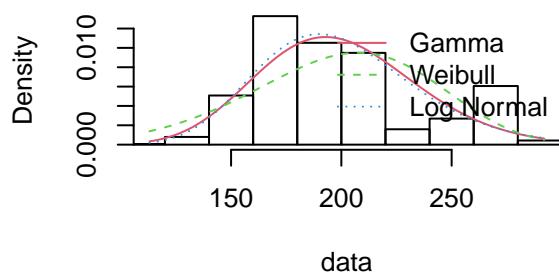


P-P plot

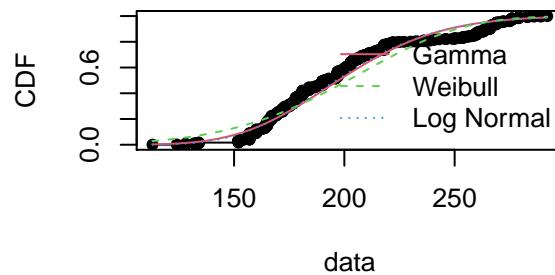


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
```

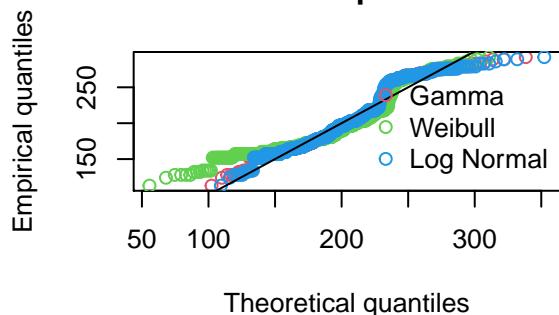
Histogram and theoretical densities



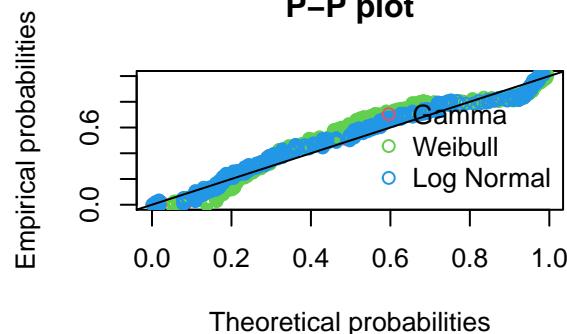
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```
gofstat(list(fit.gamma, fit.weibull, fit.ln), fitnames = c("Gamma", "Weibull", "Log Normal"))
```

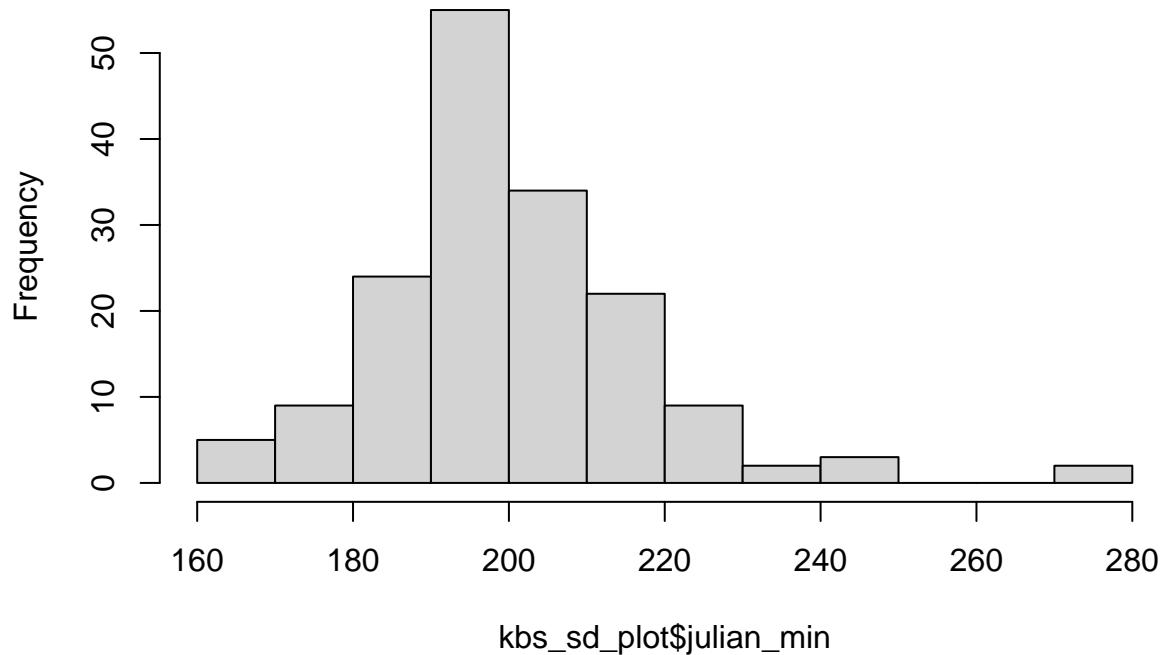
```
## Goodness-of-fit statistics
##                                     Gamma   Weibull  Log Normal
## Kolmogorov-Smirnov statistic 0.09988396 0.1360783 0.09317306
## Cramer-von Mises statistic   2.41253973 5.0192692 1.88913912
## Anderson-Darling statistic   17.07536252 30.9119372 14.02047891
##
## Goodness-of-fit criteria
##                                     Gamma   Weibull  Log Normal
## Akaike's Information Criterion 8177.095 8325.080 8155.683
## Bayesian Information Criterion 8186.506 8334.491 8165.094
```

```
# Lognormal is better - going with a lognormal transformation for kbs seed set at
# the species level
```

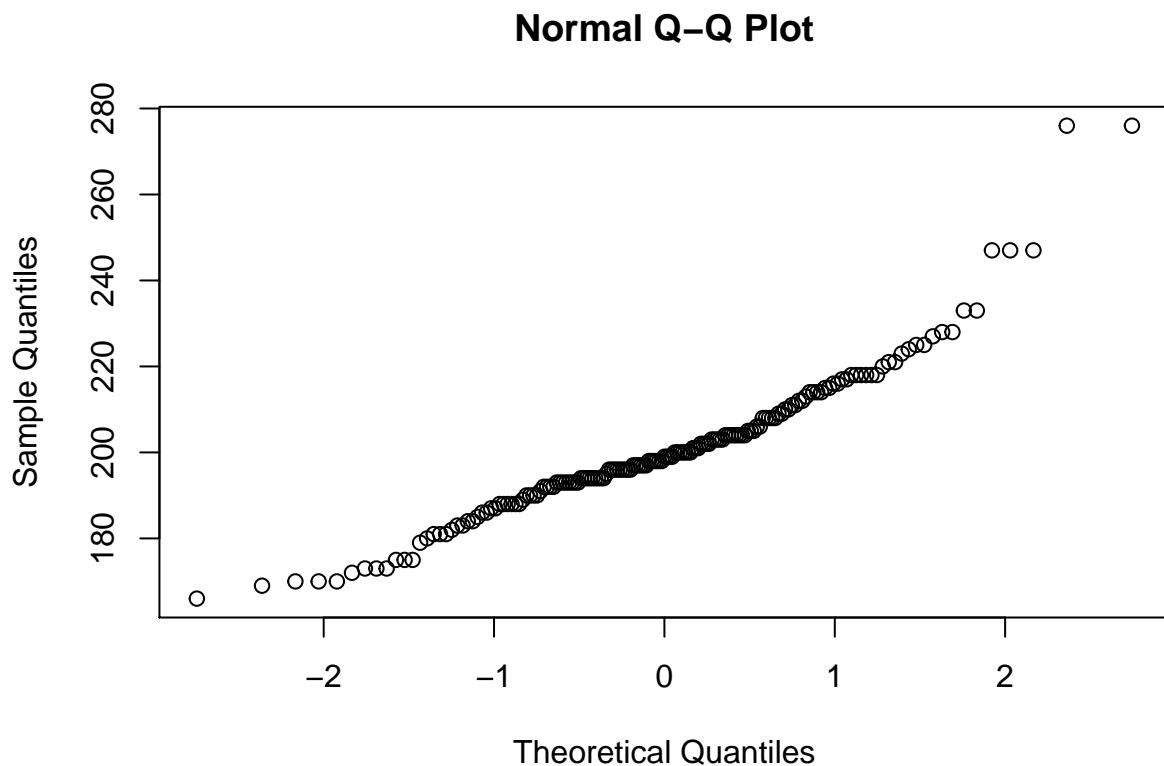
KBS PLOT LEVEL - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

```
### KBS ####
hist(kbs_sd_plot$julian_min)
```

Histogram of kbs_sd_plot\$julian_min



```
qqnorm(kbs_sd_plot$julian_min)
```



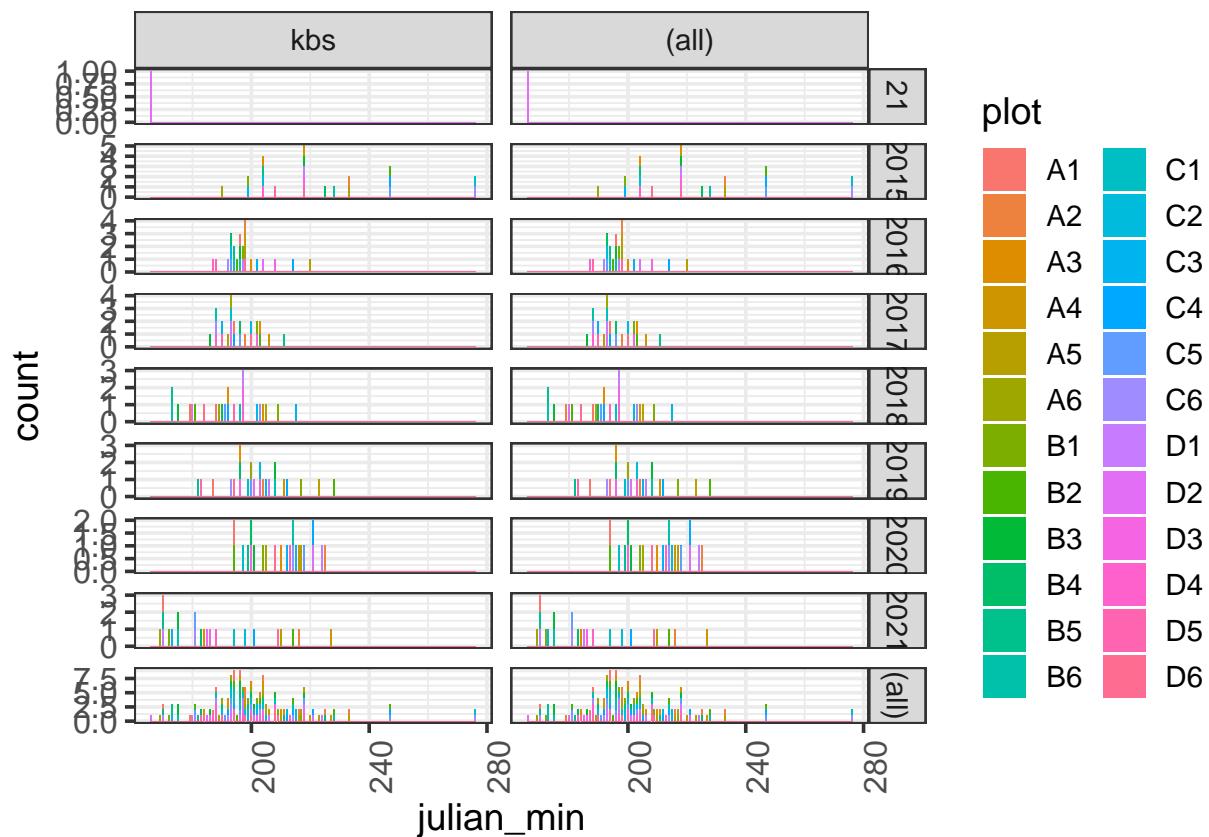
```

shapiro.test(kbs_sd_plot$julian_min) # pvalue is < 0.05 so we reject the null hypothesis that the data

## 
## Shapiro-Wilk normality test
## 
## data: kbs_sd_plot$julian_min
## W = 0.92701, p-value = 2.091e-07

# Visualizing first Julian date for kbs at the PLOT LEVEL
ggplot(kbs_sd_plot, aes(julian_min, fill = plot)) + geom_histogram(binwidth = 0.5) +
  facet_grid(year ~ site, margins = TRUE, scales = "free")

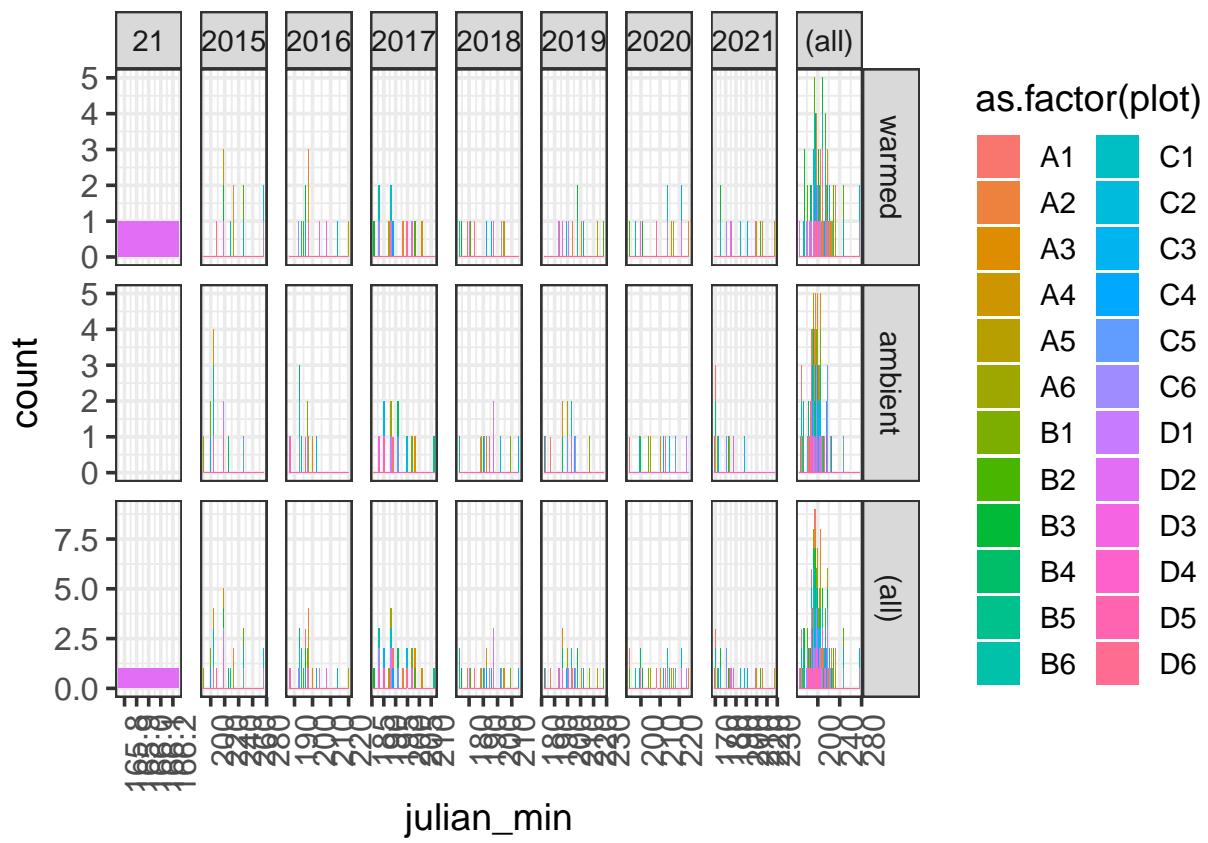
```



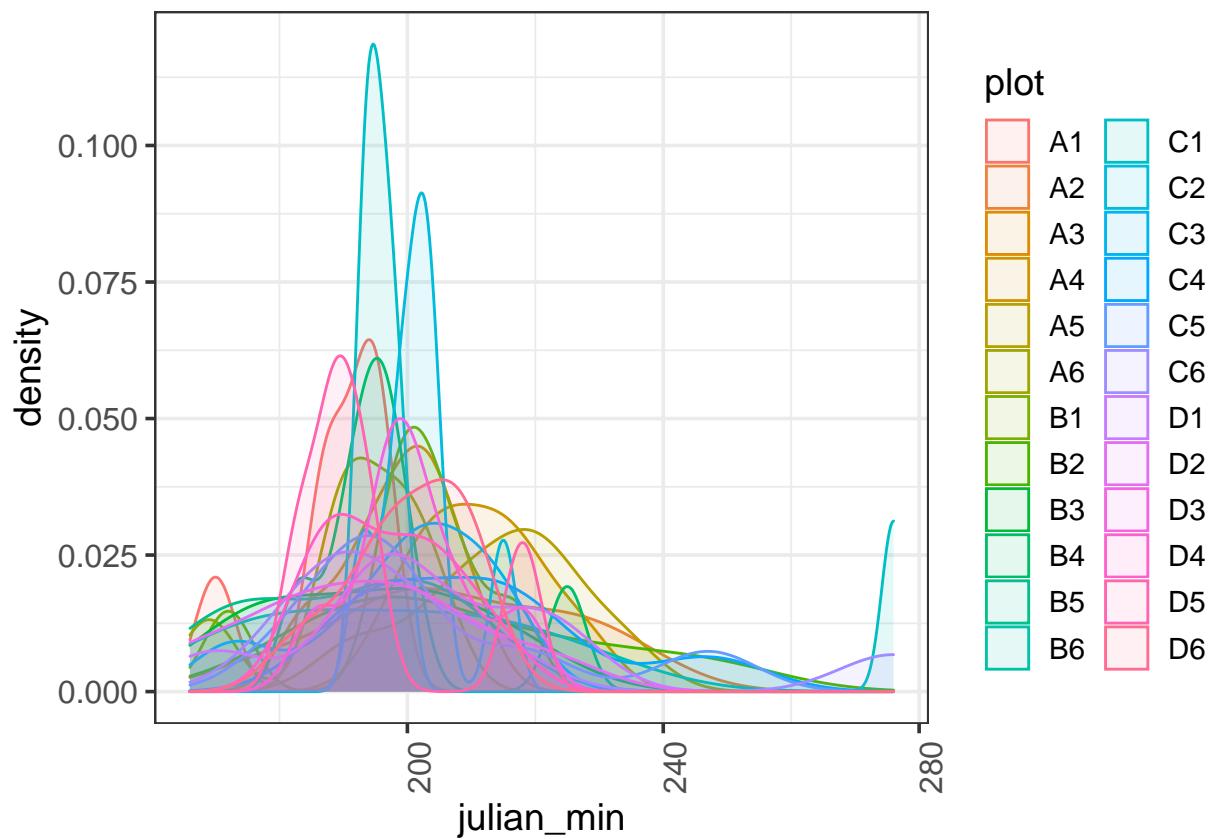
```

ggplot(kbs_sd_plot, aes(julian_min, fill = as.factor(plot))) + geom_histogram(binwidth = 0.5) +
  facet_grid(state ~ year, margins = TRUE, scales = "free")

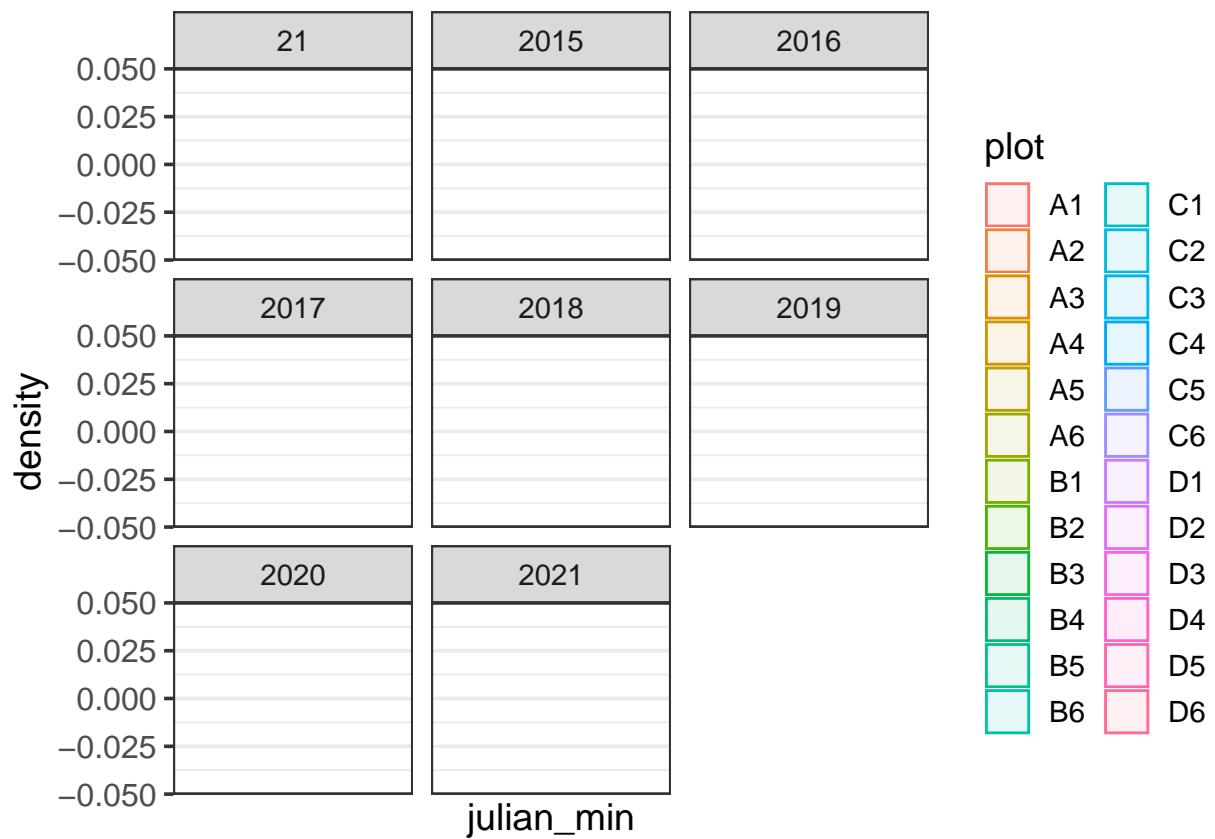
```



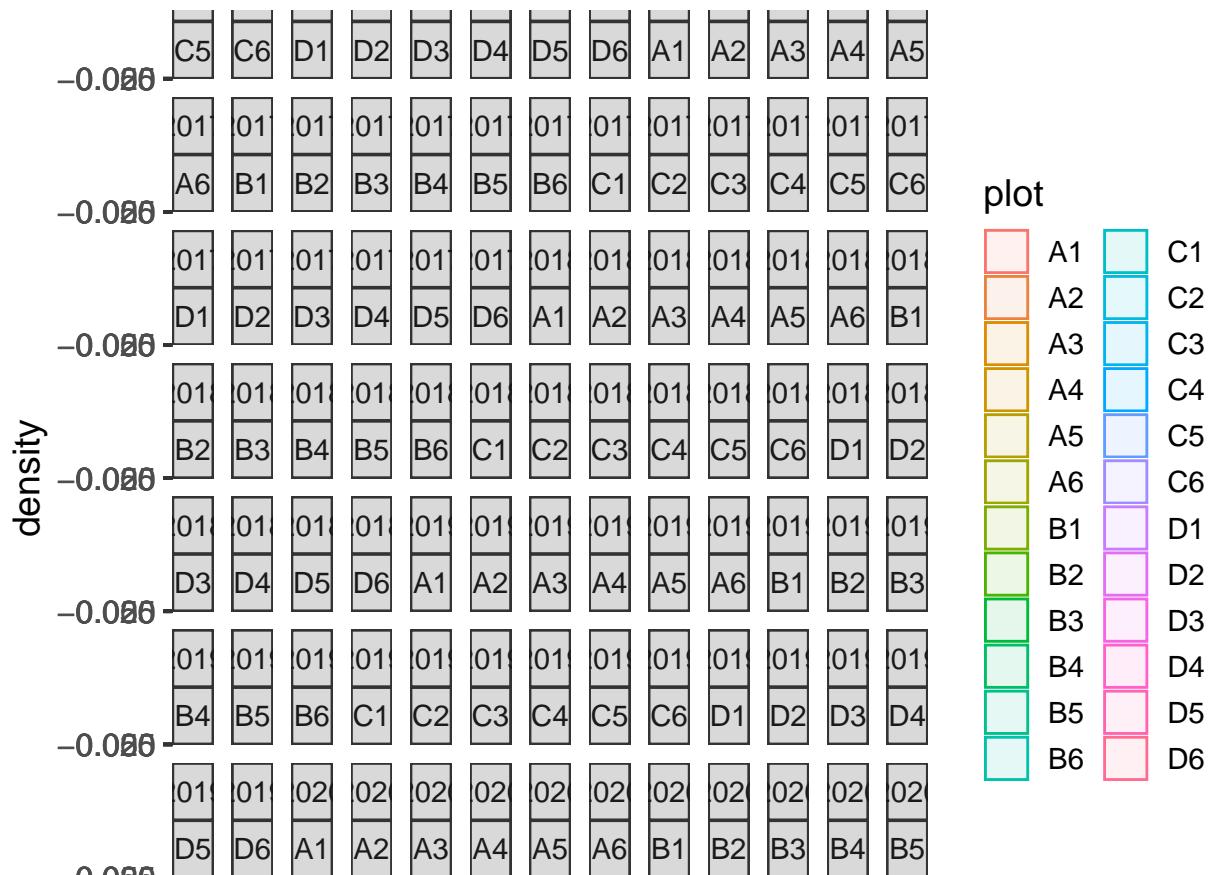
```
ggplot(kbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



```
ggplot(kbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year)
```

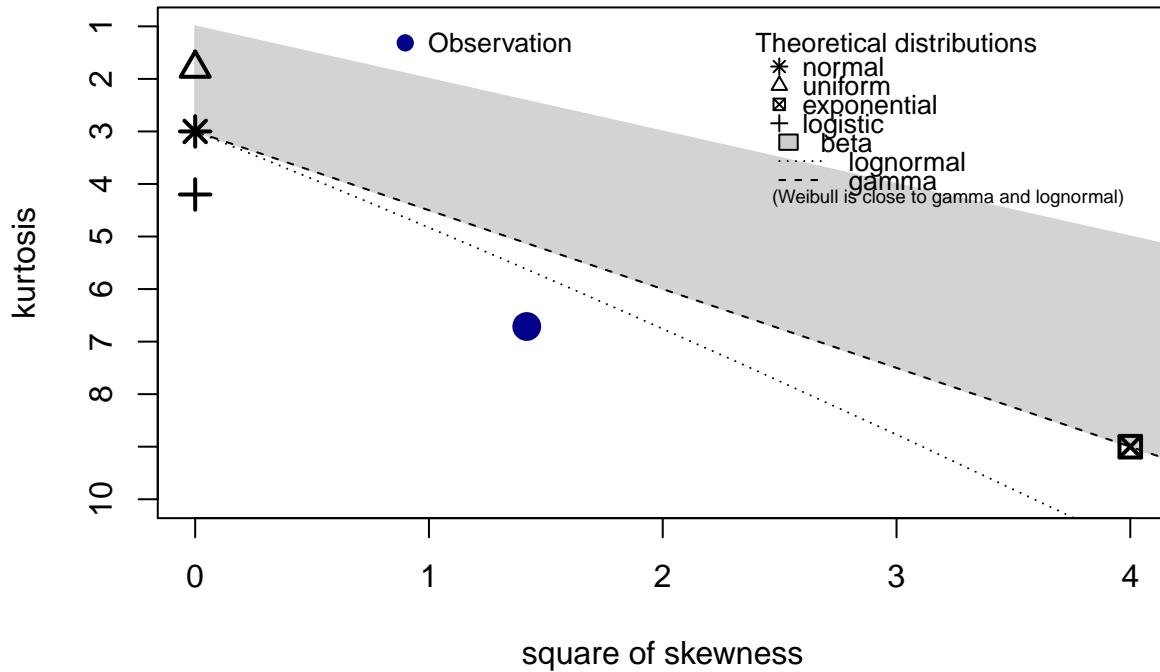


```
ggplot(kbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



```
# Exploring distributions for these right-skewed data:  
descdist(kbs_sd_plot$julian_min, discrete = FALSE)
```

Cullen and Frey graph

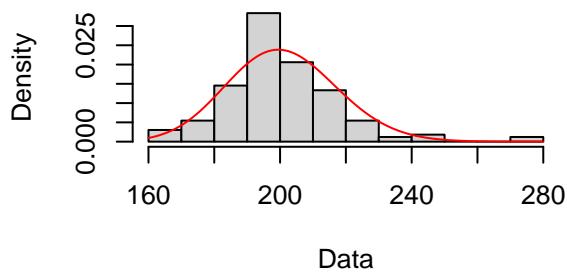
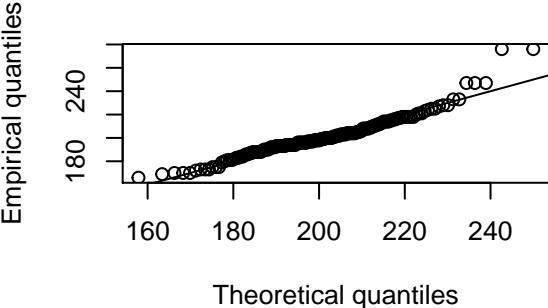
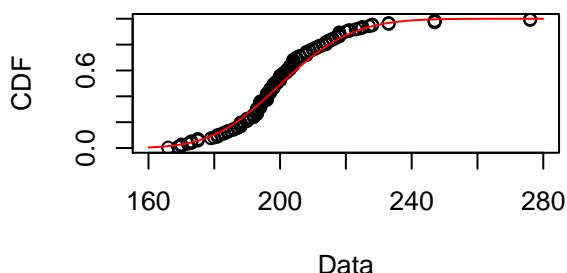
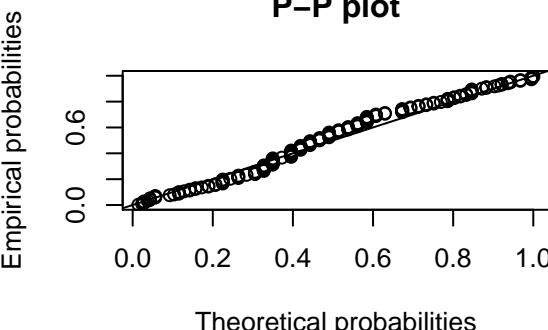


```

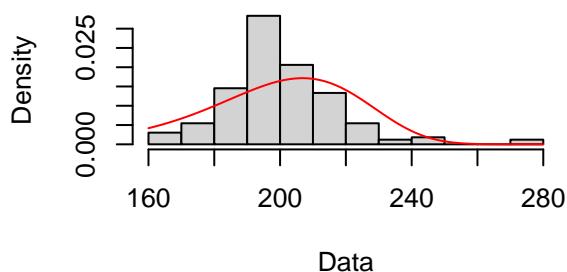
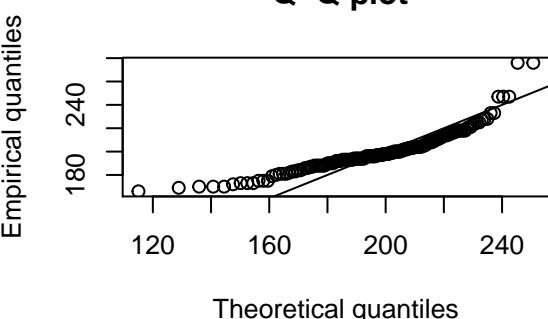
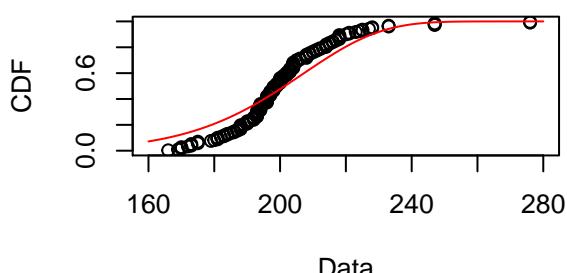
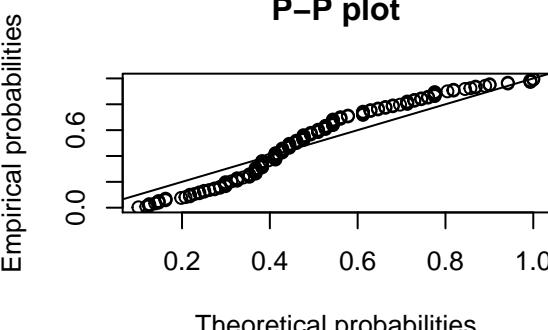
## summary statistics
## -----
## min: 166   max: 276
## median: 199
## mean: 200.8727
## estimated sd: 17.24966
## estimated skewness: 1.190949
## estimated kurtosis: 6.712314

# Gamma distribution
fit.gamma <- fitdist(kbs_sd_plot$julian_min, "gamma")
plot(fit.gamma)

```

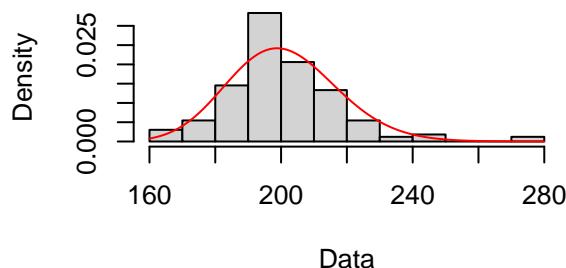
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

```
# Weibull distribution
fit.weibull <- fitdist(kbs_sd_plot$julian_min, "weibull")
plot(fit.weibull)
```

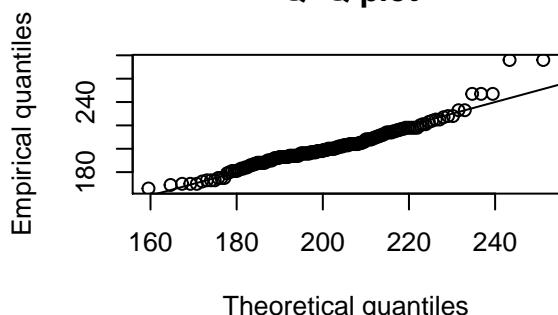
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

```
# Lognormal distribution
fit.ln <- fitdist(kbs_sd_plot$julian_min, "lnorm")
plot(fit.ln)
```

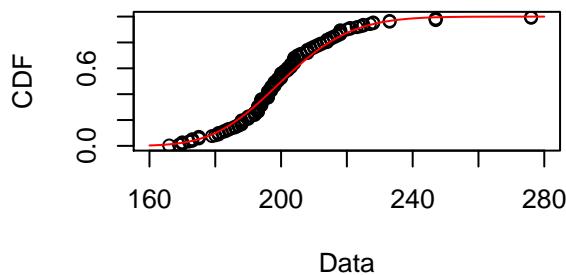
Empirical and theoretical dens.



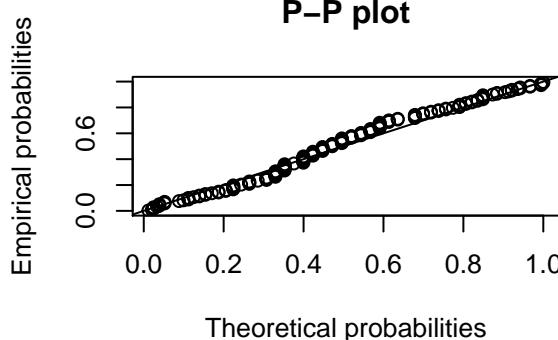
Q-Q plot



Empirical and theoretical CDFs

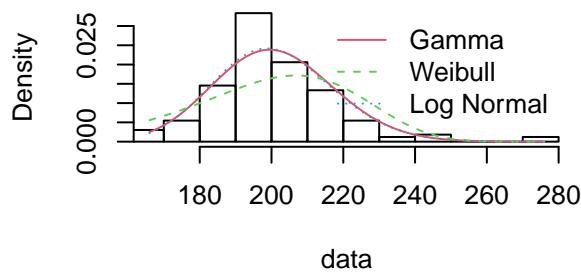


P-P plot

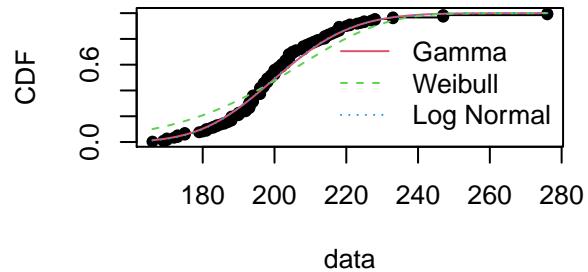


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
```

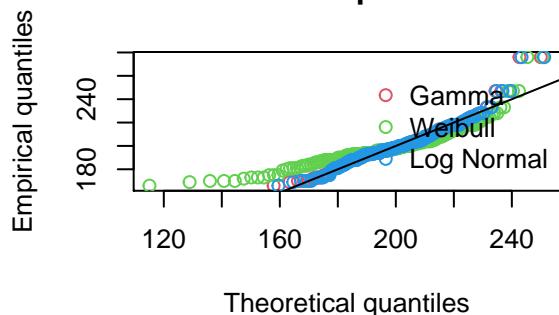
Histogram and theoretical densities



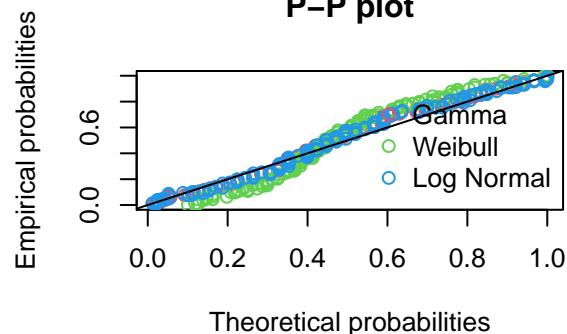
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```

gofstat(list(fit.gamma, fit.weibull, fit.ln), fitnames = c("Gamma", "Weibull", "Log Normal"))

## Goodness-of-fit statistics
##                                     Gamma   Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.1004296 0.1422884 0.09428196
## Cramer-von Mises statistic   0.2949814 1.3843015 0.25734148
## Anderson-Darling statistic   1.6842999 8.4222832 1.47267311
##
## Goodness-of-fit criteria
##                                     Gamma   Weibull Log Normal
## Akaike's Information Criterion 1402.059 1473.678 1398.466
## Bayesian Information Criterion 1408.271 1479.890 1404.678

# Lognormal is better - going with a lognormal transformation for kbs seed set at
# the plot level

```

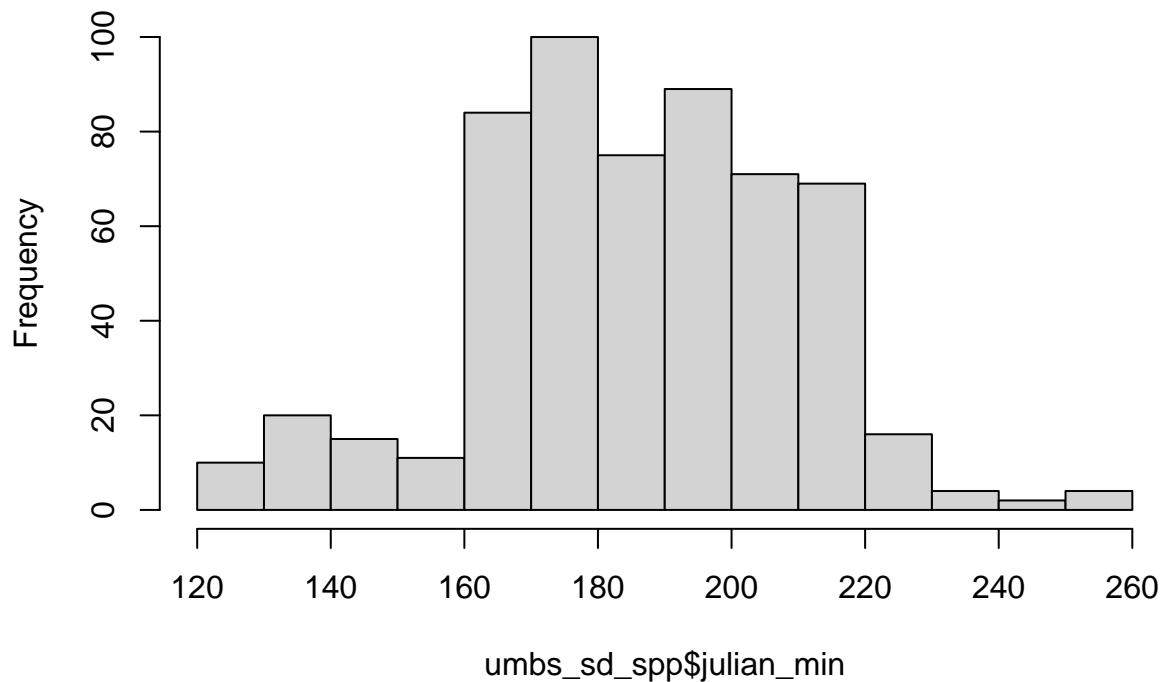
UMBS SPECIES LEVEL - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

```

### UMBS ####
hist(umbs_sd_spp$julian_min)

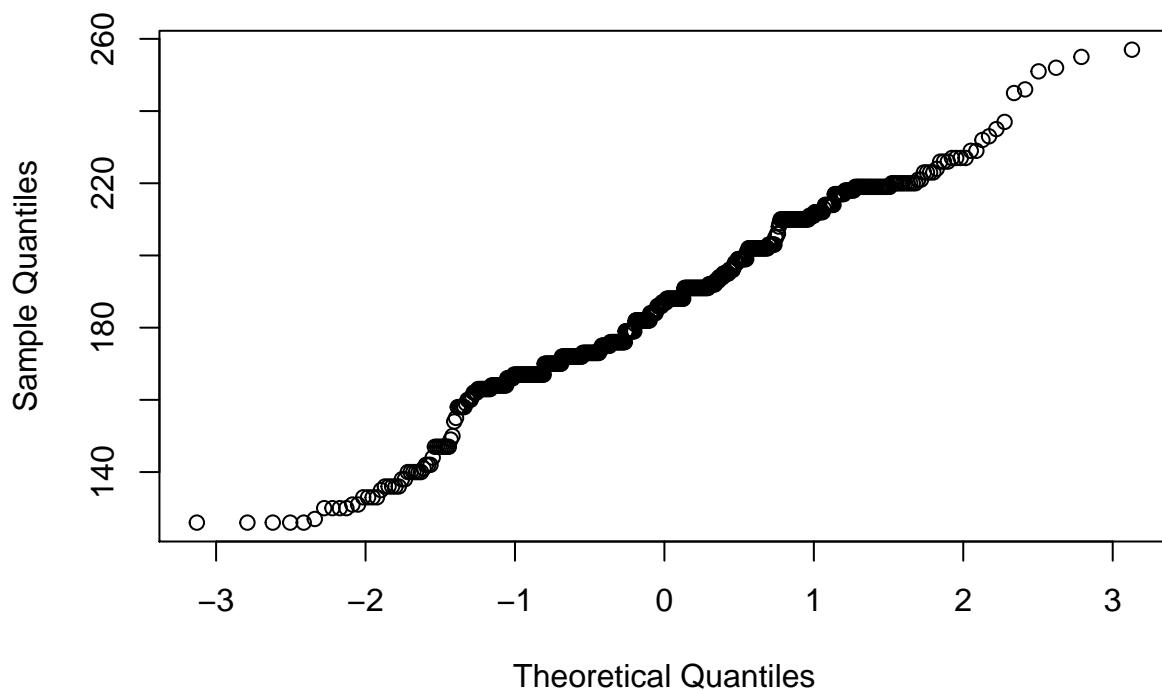
```

Histogram of umbs_sd_spp\$julian_min



```
qqnorm(umbs_sd_spp$julian_min)
```

Normal Q-Q Plot



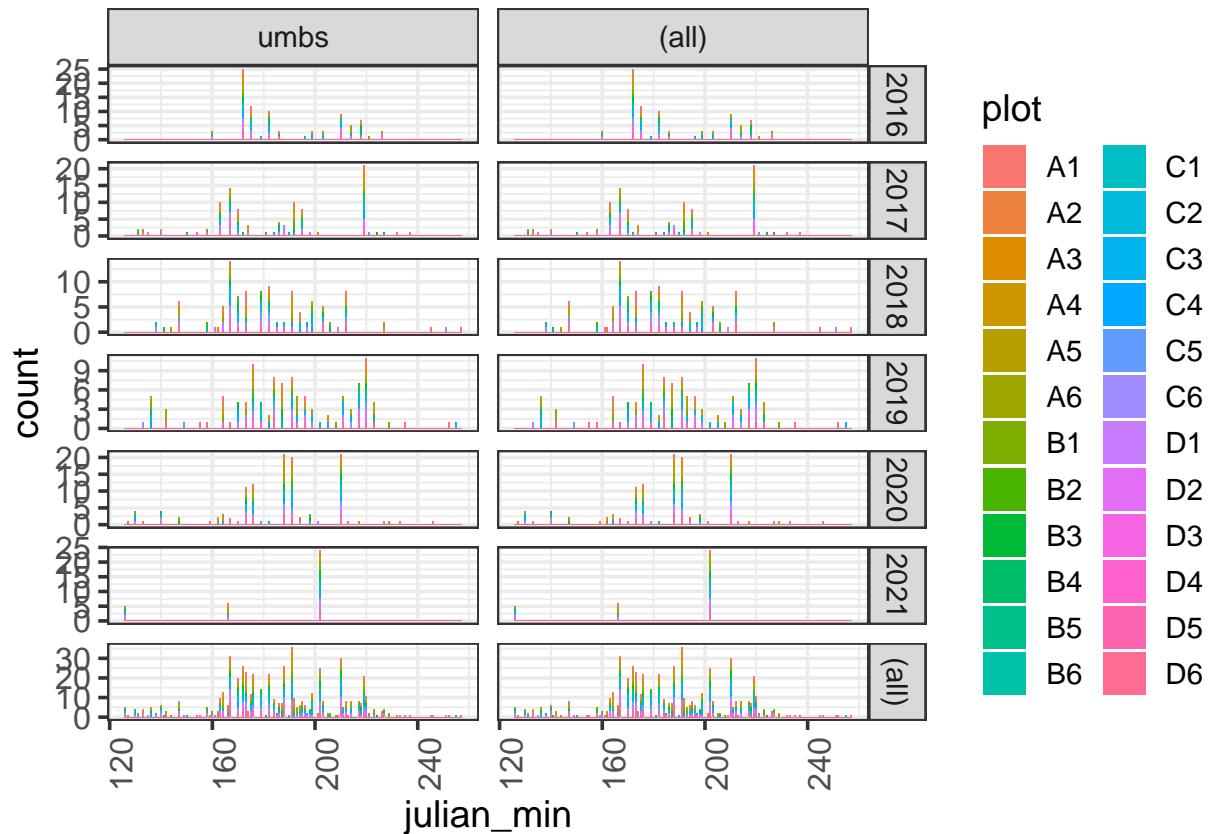
```

shapiro.test(umbs_sd_spp$julian_min) # pvalue is < 0.05 so we reject the null hypothesis that the data

##
## Shapiro-Wilk normality test
##
## data: umbs_sd_spp$julian_min
## W = 0.98456, p-value = 9.79e-06

# Visualizing median Julian date for umbs at the PLOT LEVEL
ggplot(umbs_sd_spp, aes(julian_min, fill = plot)) + geom_histogram(binwidth = 0.5) +
  facet_grid(year ~ site, margins = TRUE, scales = "free")

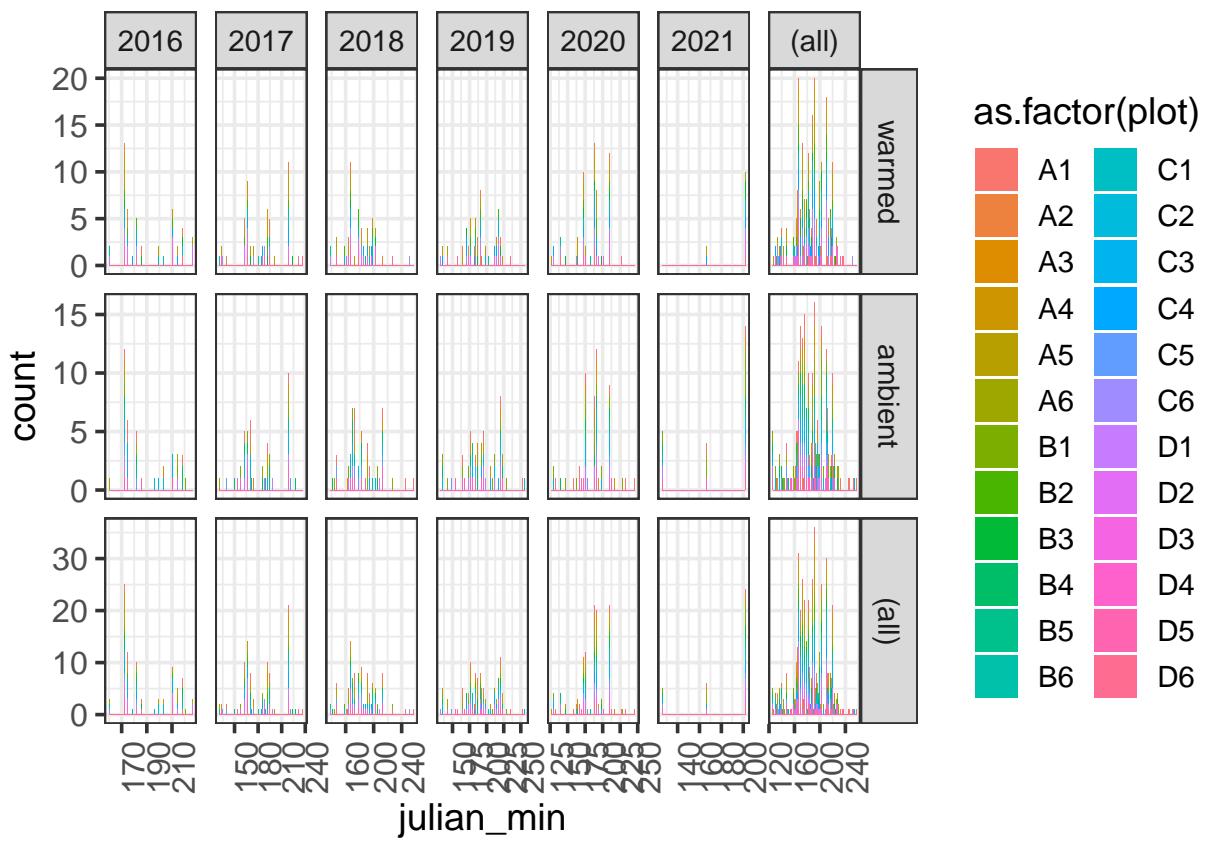
```



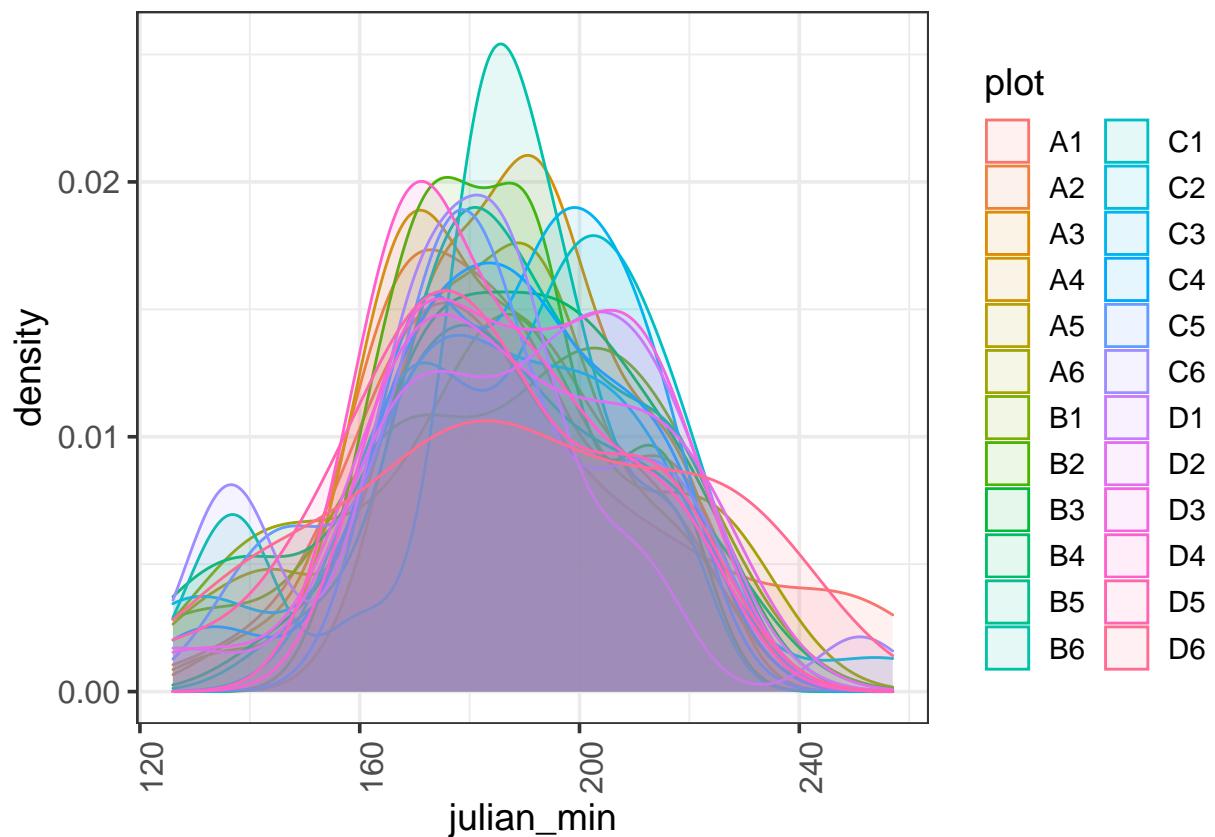
```

ggplot(umbs_sd_spp, aes(julian_min, fill = as.factor(plot))) + geom_histogram(binwidth = 0.5) +
  facet_grid(state ~ year, margins = TRUE, scales = "free")

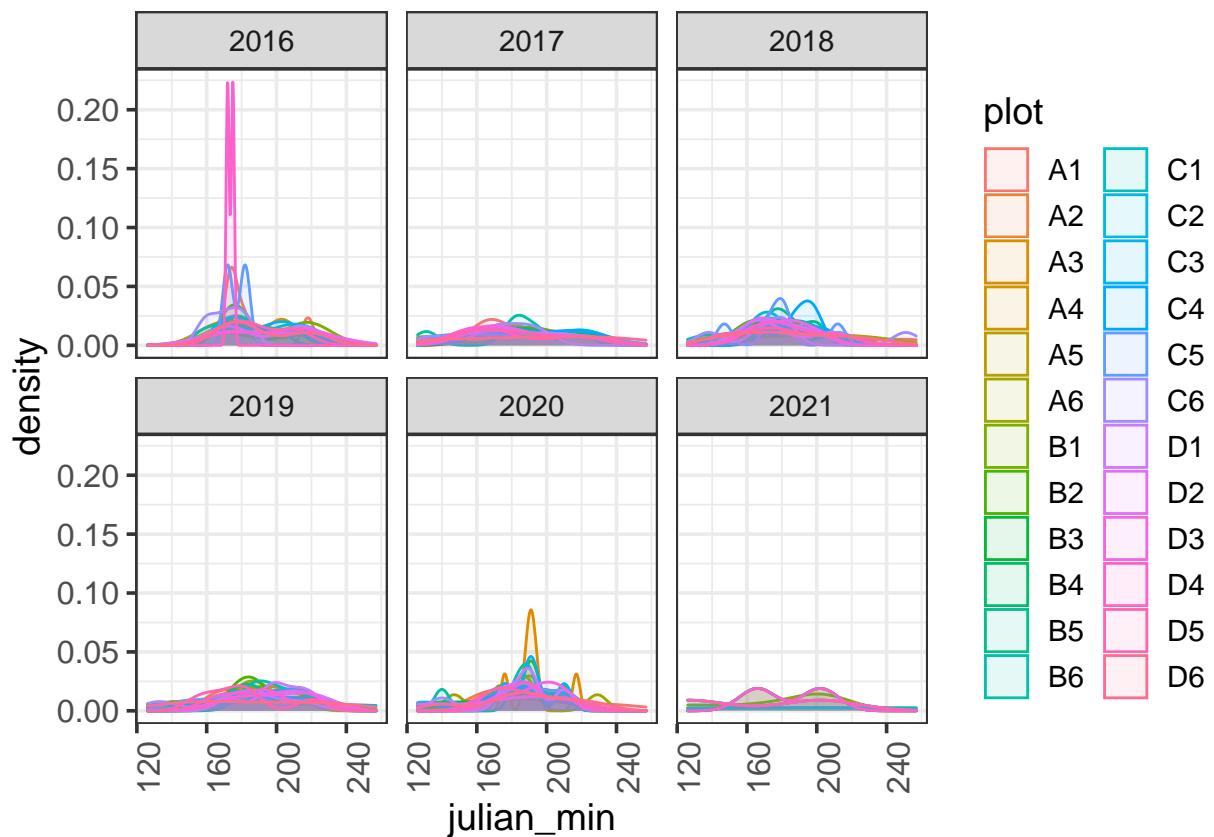
```



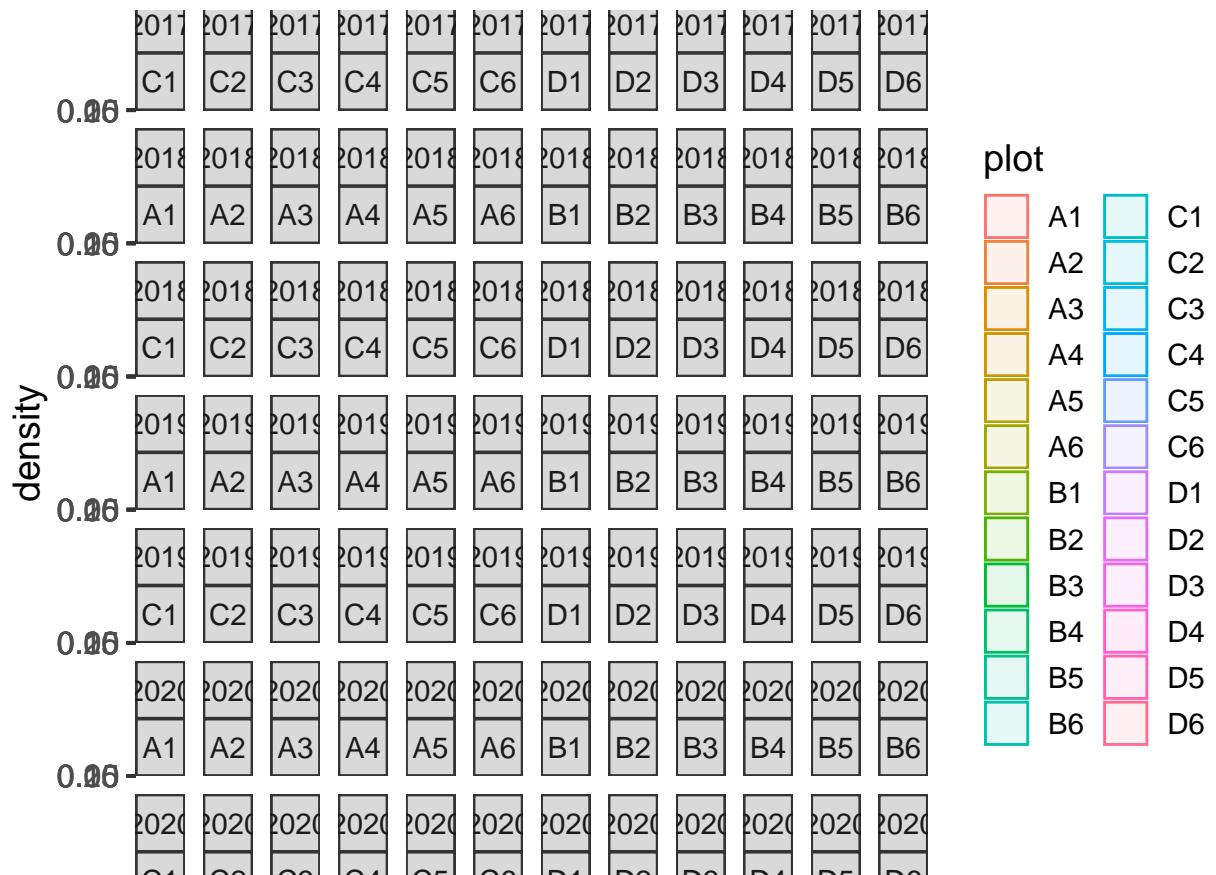
```
ggplot(umbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



```
ggplot(umbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year)
```

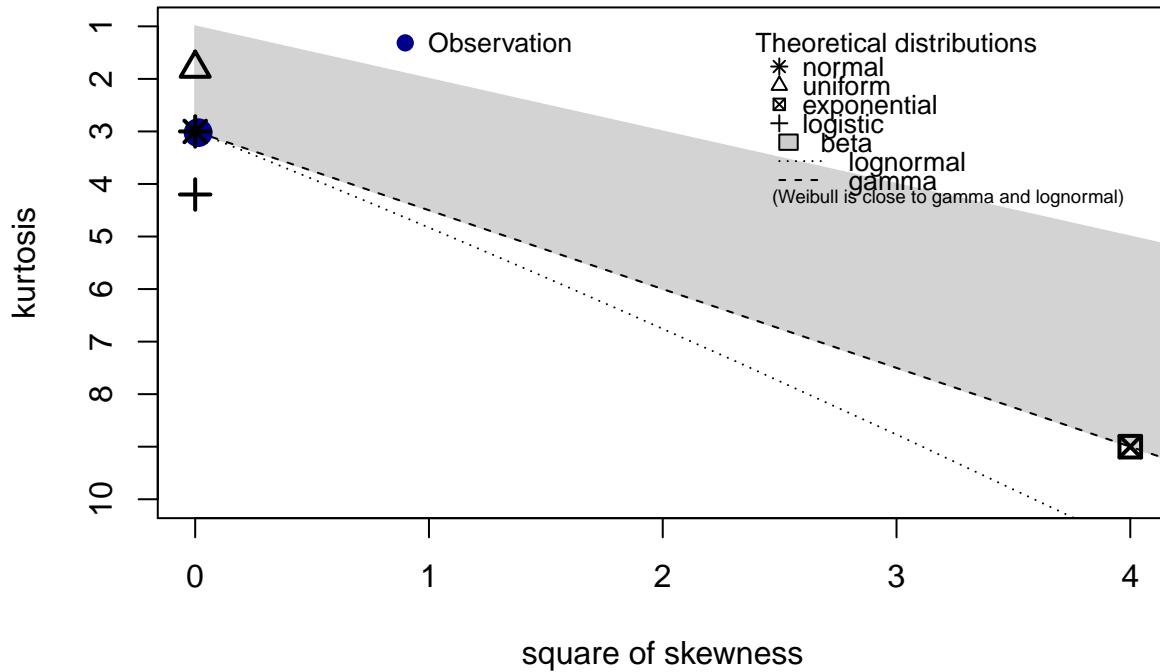


```
ggplot(umbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



```
# Exploring distributions for these right-skewed data:
descdist(umbs_sd_spp$julian_min, discrete = FALSE) # normal!
```

Cullen and Frey graph

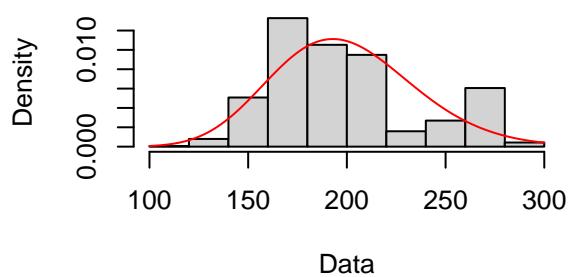
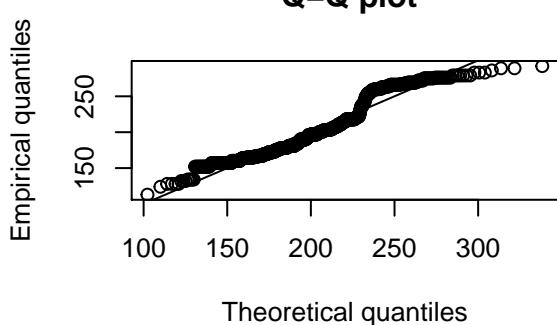
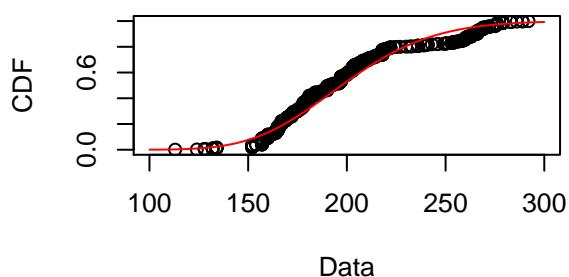
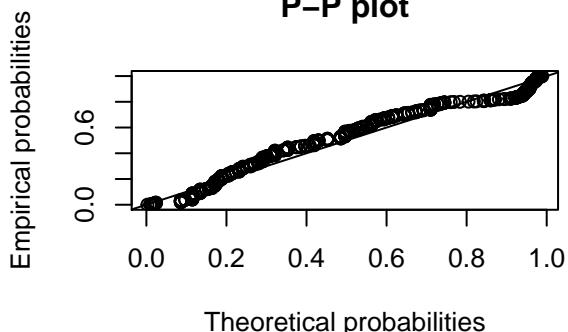


```

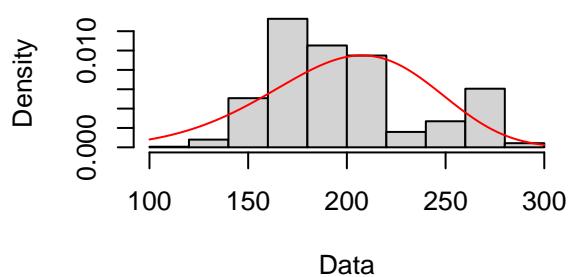
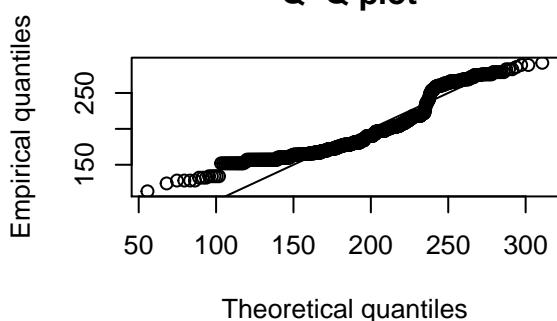
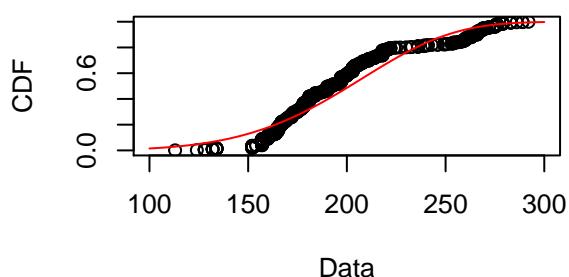
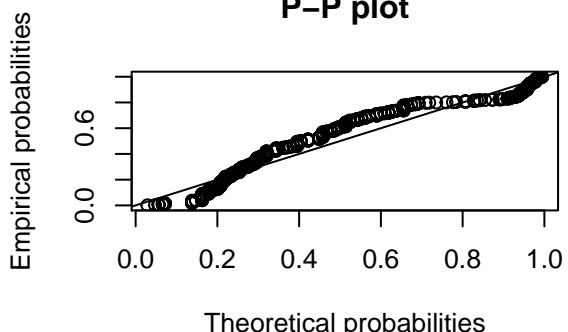
## summary statistics
## -----
## min: 126   max: 257
## median: 187
## mean: 186.1596
## estimated sd: 23.86446
## estimated skewness: -0.1125617
## estimated kurtosis: 3.023487

# Gamma distribution
fit.gamma <- fitdist(kbs_sd_spp$julian_min, "gamma")
plot(fit.gamma)

```

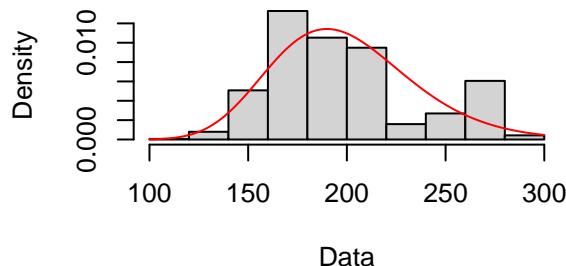
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

```
# Weibull distribution
fit.weibull <- fitdist(kbs_sd_spp$julian_min, "weibull")
plot(fit.weibull)
```

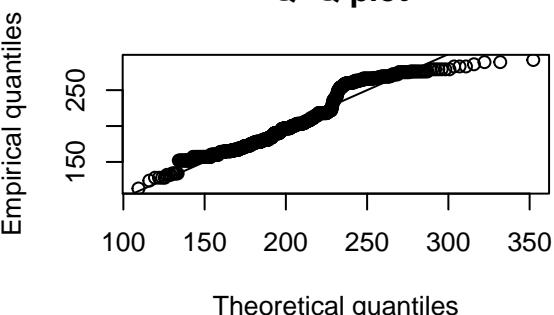
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

```
# Lognormal distribution
fit.ln <- fitdist(kbs_sd_spp$julian_min, "lnorm")
plot(fit.ln)
```

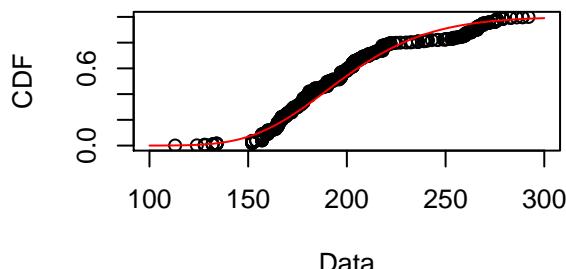
Empirical and theoretical dens.



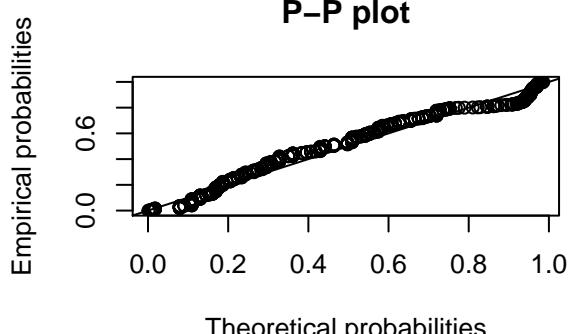
Q–Q plot



Empirical and theoretical CDFs

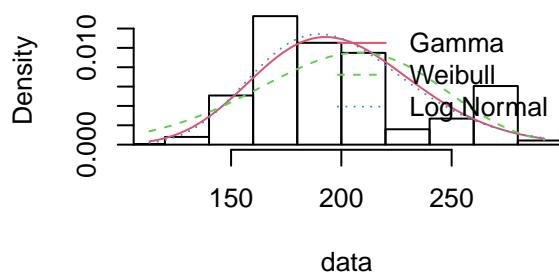


P–P plot

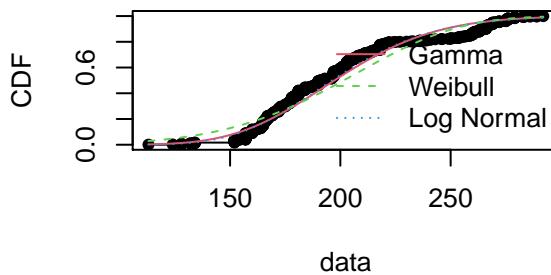


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
```

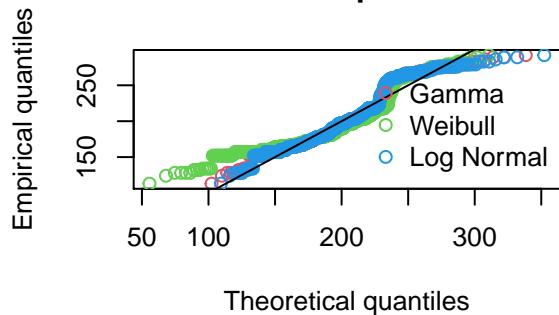
Histogram and theoretical densities



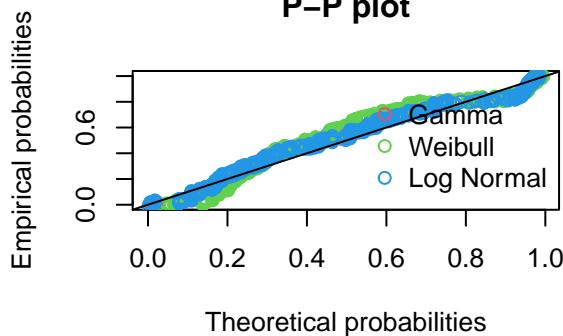
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```
gofstat(list(fit.gamma, fit.weibull, fit.ln), fitnames = c("Gamma", "Weibull", "Log Normal"))
```

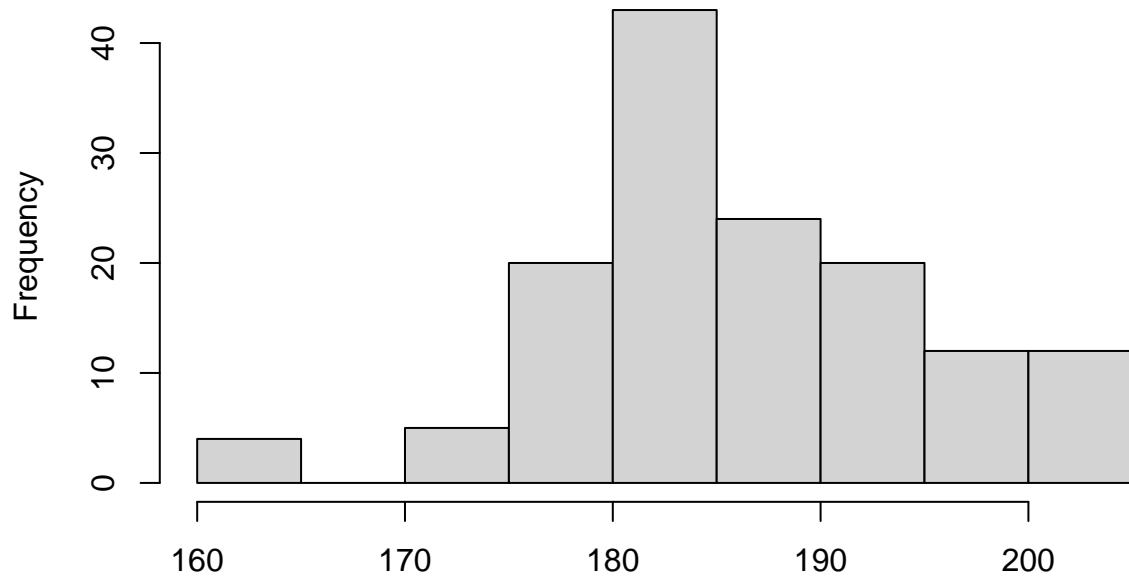
```
## Goodness-of-fit statistics
##                                     Gamma   Weibull  Log Normal
## Kolmogorov-Smirnov statistic 0.09988396 0.1360783 0.09317306
## Cramer-von Mises statistic   2.41253973 5.0192692 1.88913912
## Anderson-Darling statistic  17.07536252 30.9119372 14.02047891
##
## Goodness-of-fit criteria
##                                     Gamma   Weibull  Log Normal
## Akaike's Information Criterion 8177.095 8325.080 8155.683
## Bayesian Information Criterion 8186.506 8334.491 8165.094
```

*# Lognormal is slightly better than gamma - going with a lognormal transformation
for umbs seed set at the species level*

UMBS PLOT LEVEL - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

```
### UMBS ####
hist(umbs_sd_plot$julian_min)
```

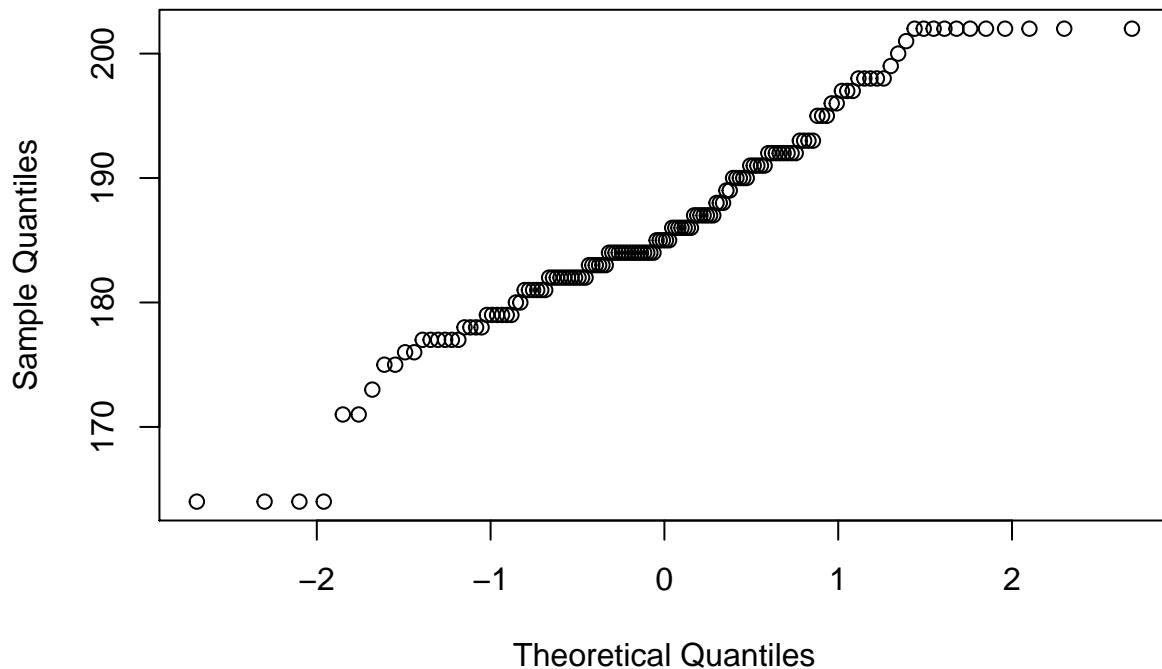
Histogram of umbs_sd_plot\$julian_min



umbs_sd_plot\$julian_min

```
qqnorm(umbs_sd_plot$julian_min)
```

Normal Q-Q Plot



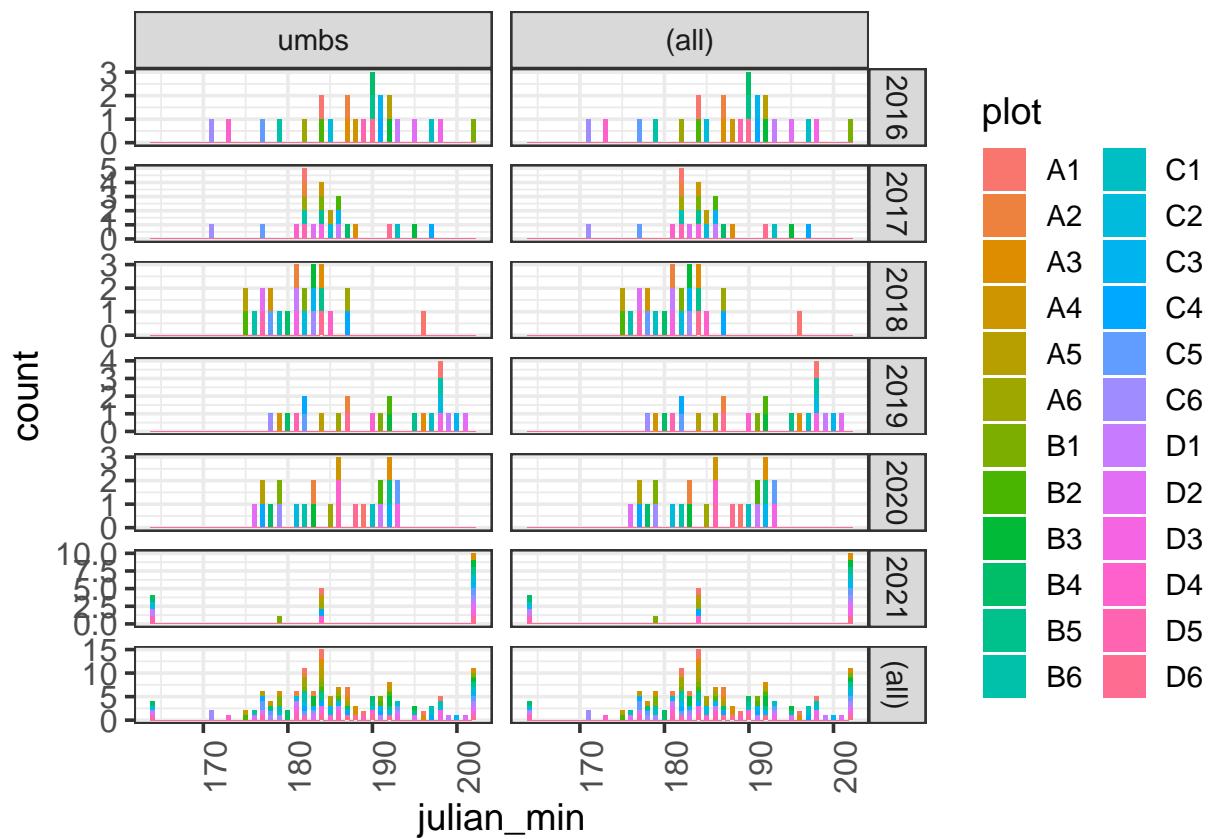
```

shapiro.test(umbs_sd_plot$julian_min) # pvalue is < 0.05 so we reject the null hypothesis that the data

## 
## Shapiro-Wilk normality test
## 
## data: umbs_sd_plot$julian_min
## W = 0.96627, p-value = 0.001563

# Visualizing first Julian date for umbs at the PLOT LEVEL
ggplot(umbs_sd_plot, aes(julian_min, fill = plot)) + geom_histogram(binwidth = 0.5) +
  facet_grid(year ~ site, margins = TRUE, scales = "free")

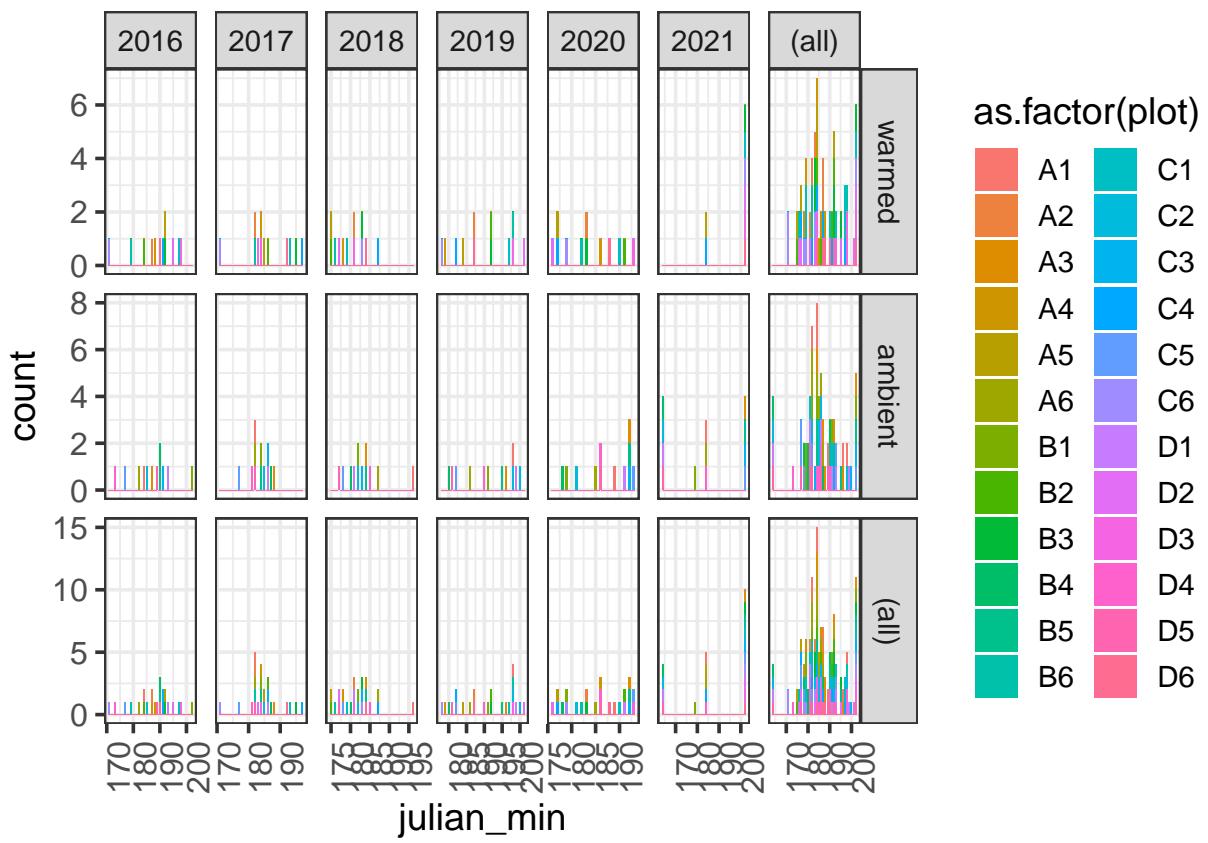
```



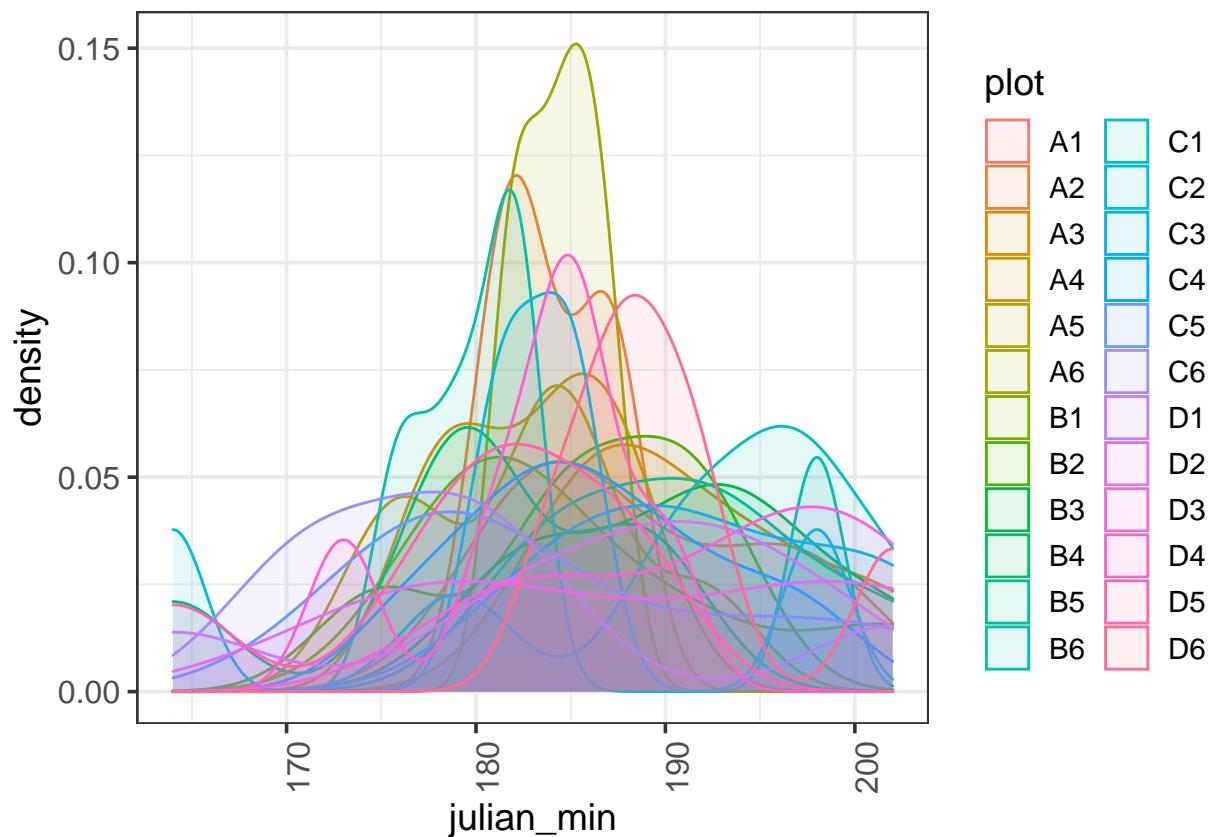
```

ggplot(umbs_sd_plot, aes(julian_min, fill = as.factor(plot))) + geom_histogram(binwidth = 0.5) +
  facet_grid(state ~ year, margins = TRUE, scales = "free")

```



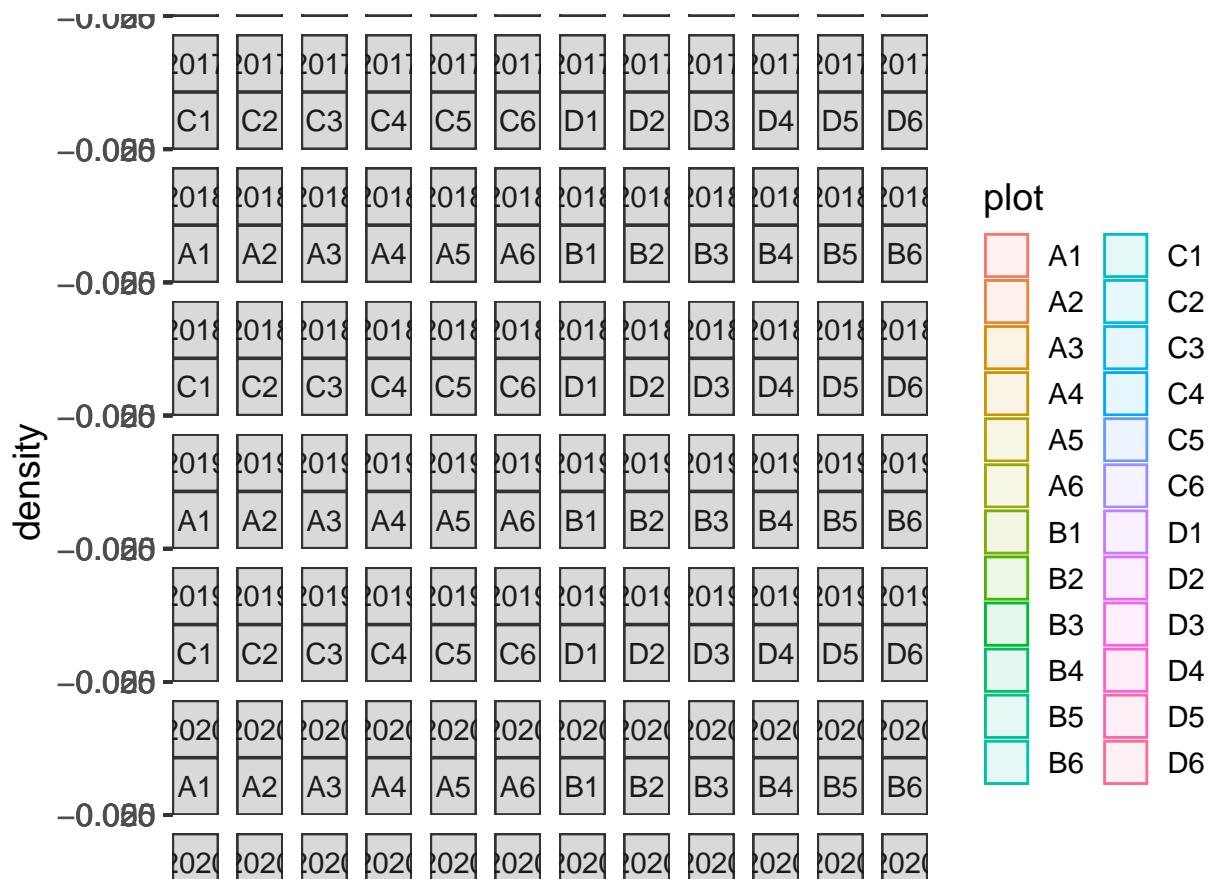
```
ggplot(umbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



```
ggplot(umbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year)
```

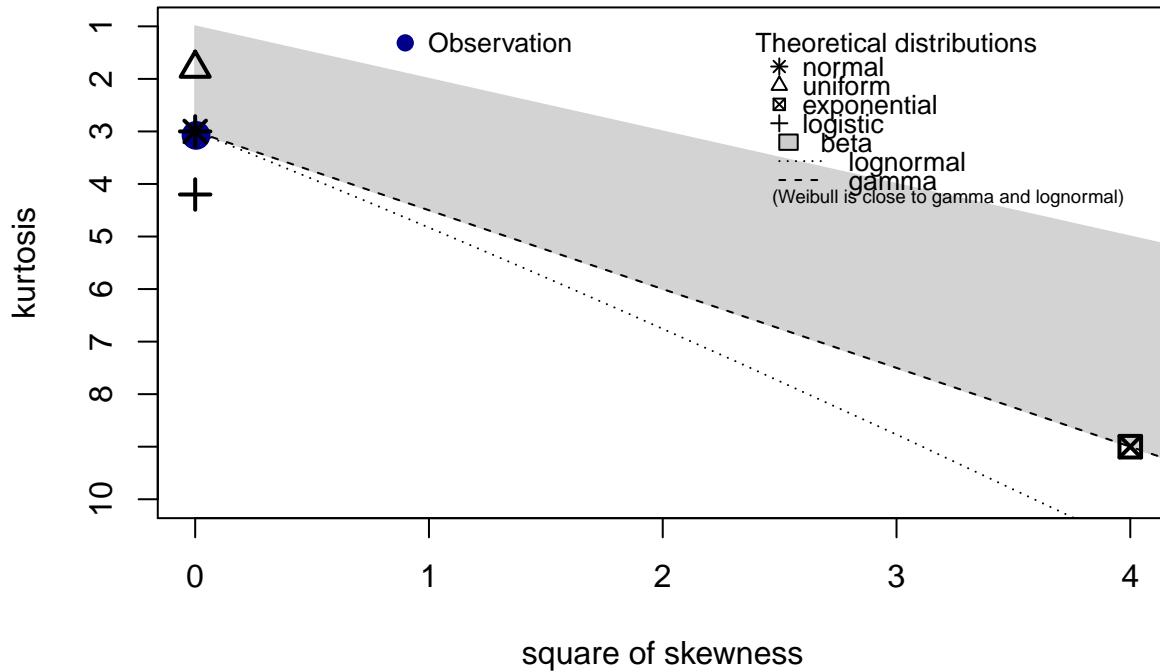


```
ggplot(umbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



```
# Exploring distributions for these right-skewed data:  
descdist(umbs_sd_plot$julian_min, discrete = FALSE) # normal
```

Cullen and Frey graph



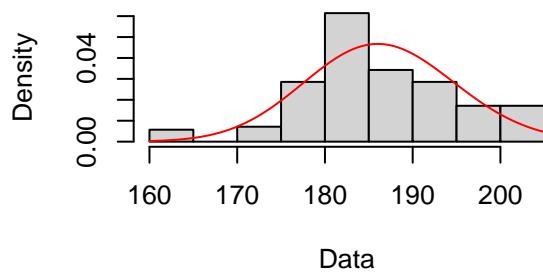
```

## summary statistics
## -----
## min: 164   max: 202
## median: 185
## mean: 186.4143
## estimated sd: 8.559555
## estimated skewness: -0.06050833
## estimated kurtosis: 3.07523

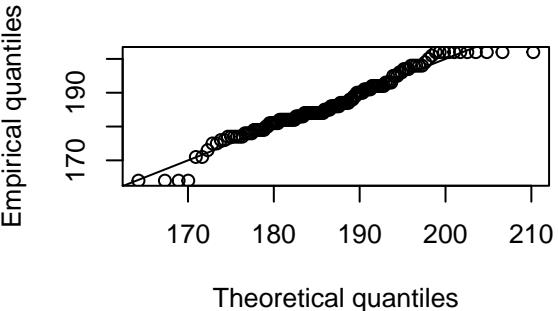
# Gamma distribution
fit.gamma <- fitdist(umbs_sd_plot$julian_min, "gamma")
plot(fit.gamma)

```

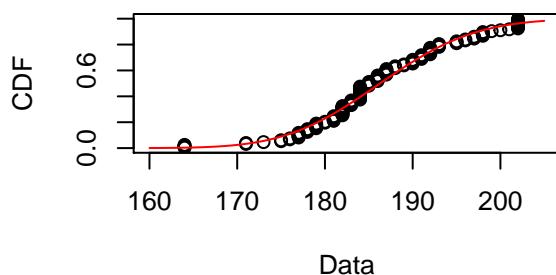
Empirical and theoretical dens.



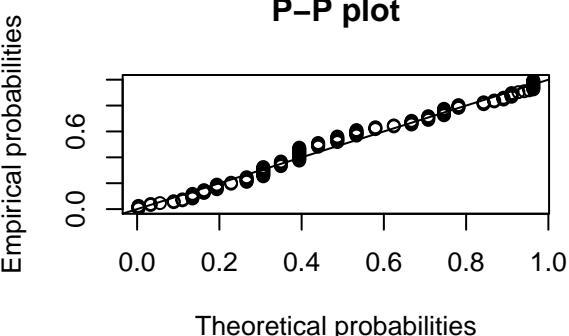
Q-Q plot



Empirical and theoretical CDFs

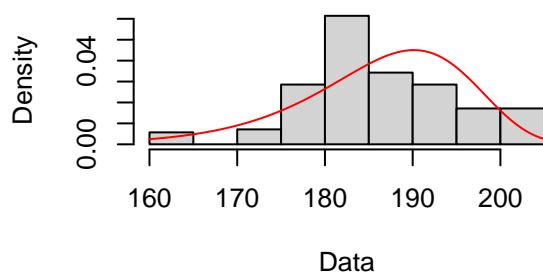


P-P plot

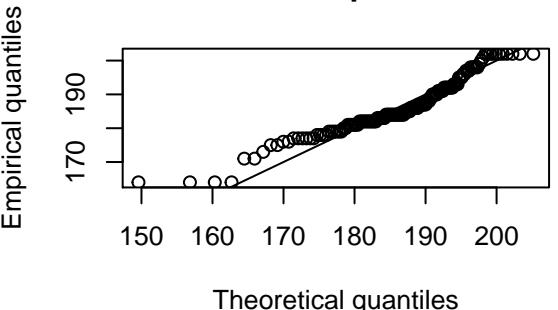


```
# Weibull distribution
fit.weibull <- fitdist(umbs_sd_plot$julian_min, "weibull")
plot(fit.weibull)
```

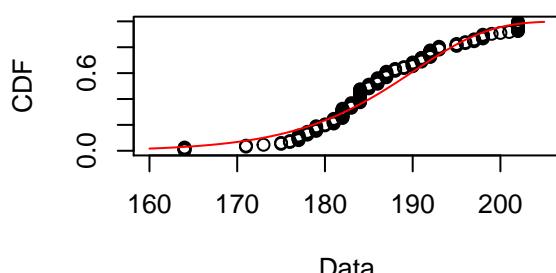
Empirical and theoretical dens.



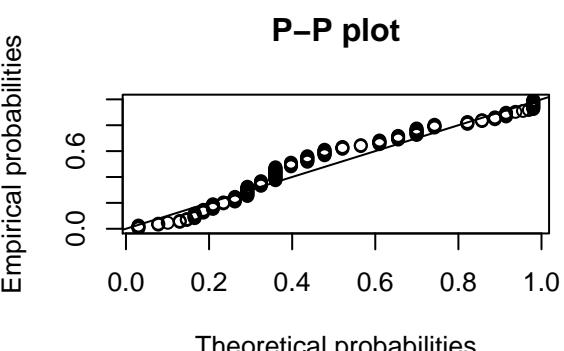
Q-Q plot



Empirical and theoretical CDFs

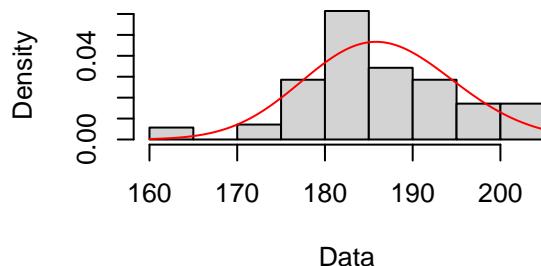


P-P plot

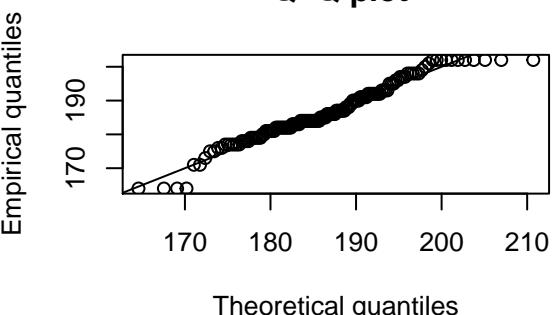


```
# Lognormal distribution
fit.ln <- fitdist(umbs_sd_plot$julian_min, "lnorm")
plot(fit.ln)
```

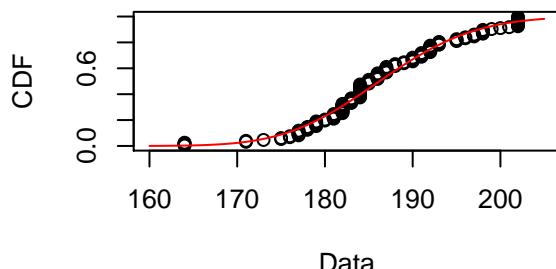
Empirical and theoretical dens.



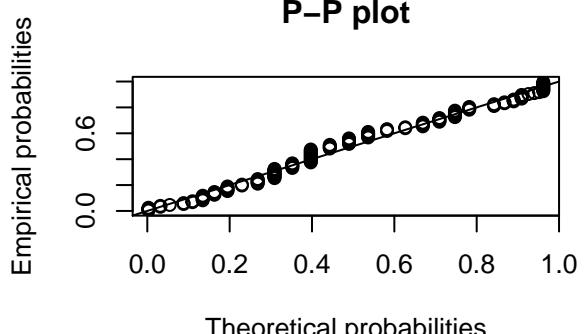
Q-Q plot



Empirical and theoretical CDFs

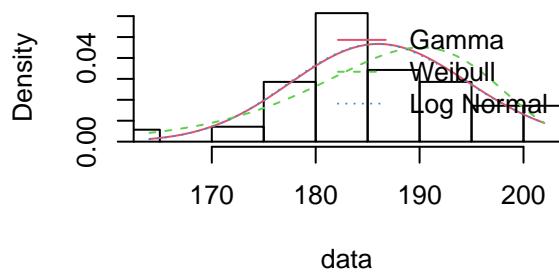


P-P plot

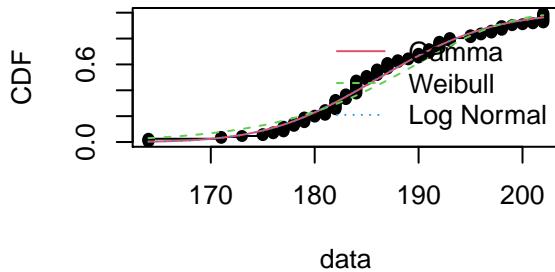


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
```

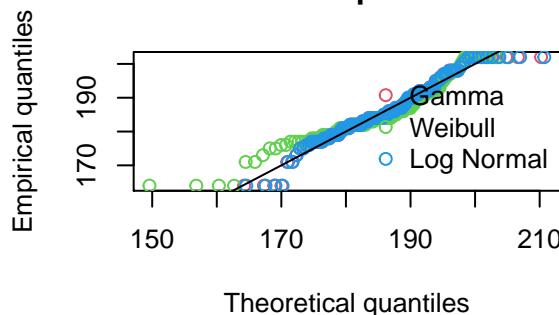
Histogram and theoretical densities



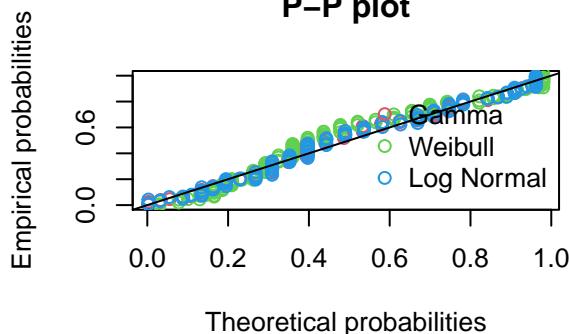
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```
gofstat(list(fit.gamma, fit.weibull, fit.ln), fitnames = c("Gamma", "Weibull", "Log Normal"))
```

```
## Goodness-of-fit statistics
##                                     Gamma   Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.08433184 0.1363419 0.08146899
## Cramer-von Mises statistic   0.16388469 0.5217951 0.15545965
## Anderson-Darling statistic   1.16140603 3.0890479 1.13032756
##
## Goodness-of-fit criteria
##                                     Gamma   Weibull Log Normal
## Akaike's Information Criterion 1001.939 1013.962 1002.326
## Bayesian Information Criterion 1007.822 1019.846 1008.210
```

```
# Gamma is slightly better here but going to go with a lognormal transformation
# for umbs seed set at the plot level
```

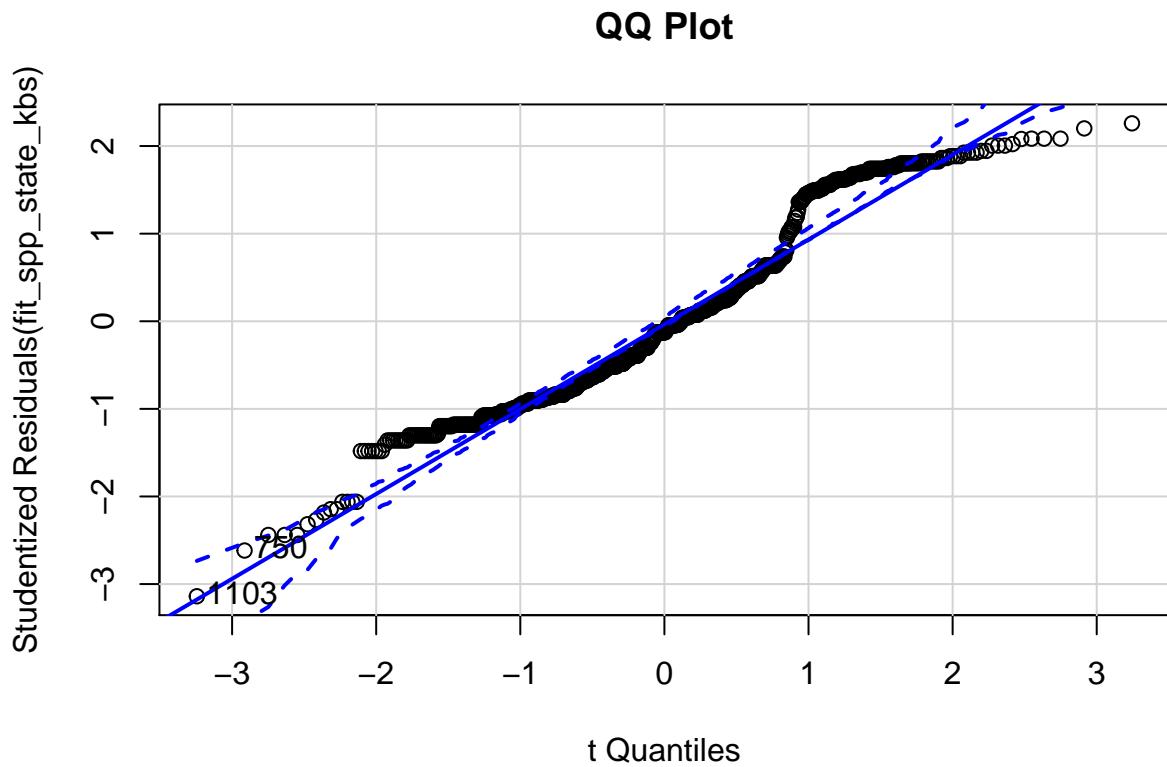
Leverage plots and detecting Outliers. <https://www.statmethods.net/stats/rdiagnostics.html>

These illustrate whether certain data points have more leverage (more influence), and thus could be outliers. It's a way of detecting outliers. Leverage plots can help identify whether a point has high or low influence, based on its leverage and residual and determining model fit with and without the point in question. Ultimately you decide whether the points are outliers or not, based on the knowledge of the system and how much it changes the model when included vs. excluded from the data used to fit the model. Here is a good overview of the combination of leverage and residual: scroll down to sections beginning at “13.3 Unusual Observations”: <https://daviddalpiaz.github.io/appliedstats/model-diagnostics.html>

```
# KBS species level data KBS State-only model
fit_spp_state_kbs <- lm(log(julian_min) ~ state, data = kbs_sd_spp)
outlierTest(fit_spp_state_kbs) # no outliers
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferroni p
## 1103 -3.139037      0.0017562        NA
```

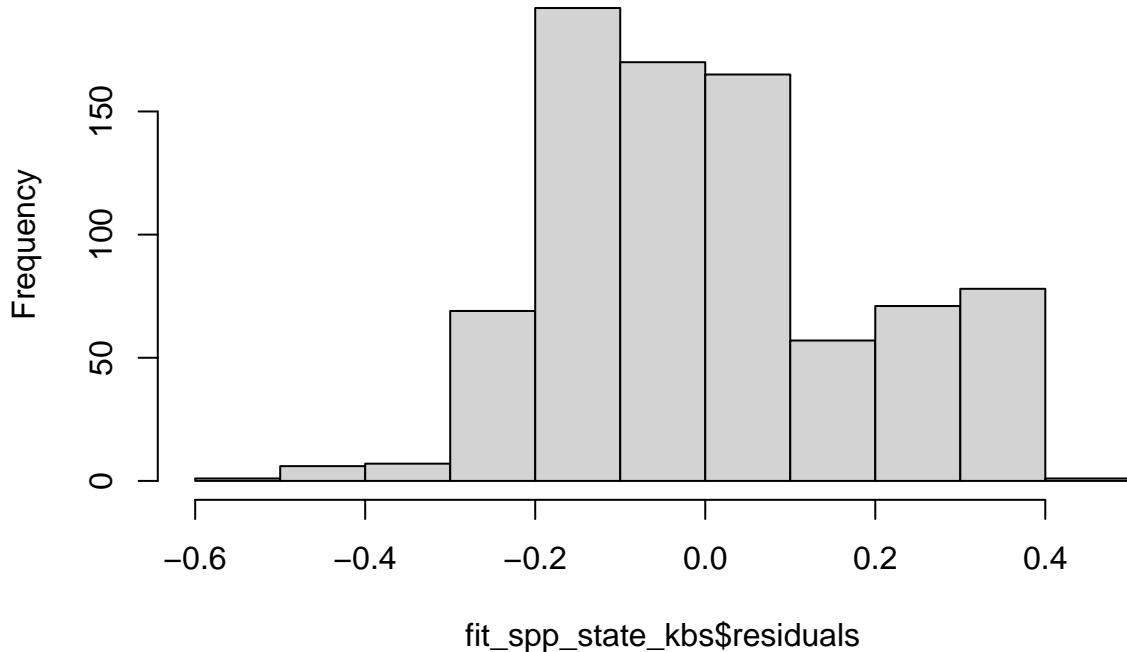
```
qqPlot(fit_spp_state_kbs, main = "QQ Plot")
```



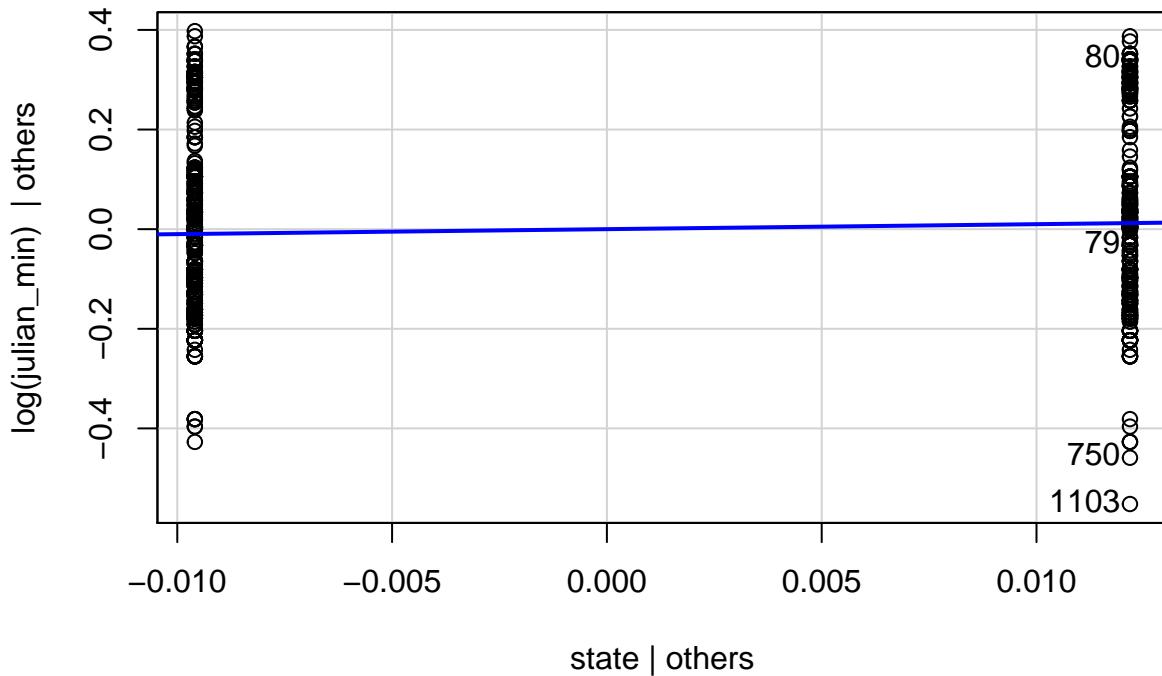
```
##   750 1103
##   433 642
```

```
hist(fit_spp_state_kbs$residuals)
```

Histogram of fit_spp_state_kbs\$residuals



```
leveragePlots(fit_spp_state_kbs)
```



```
ols_test_normality(fit_spp_state_kbs)
```

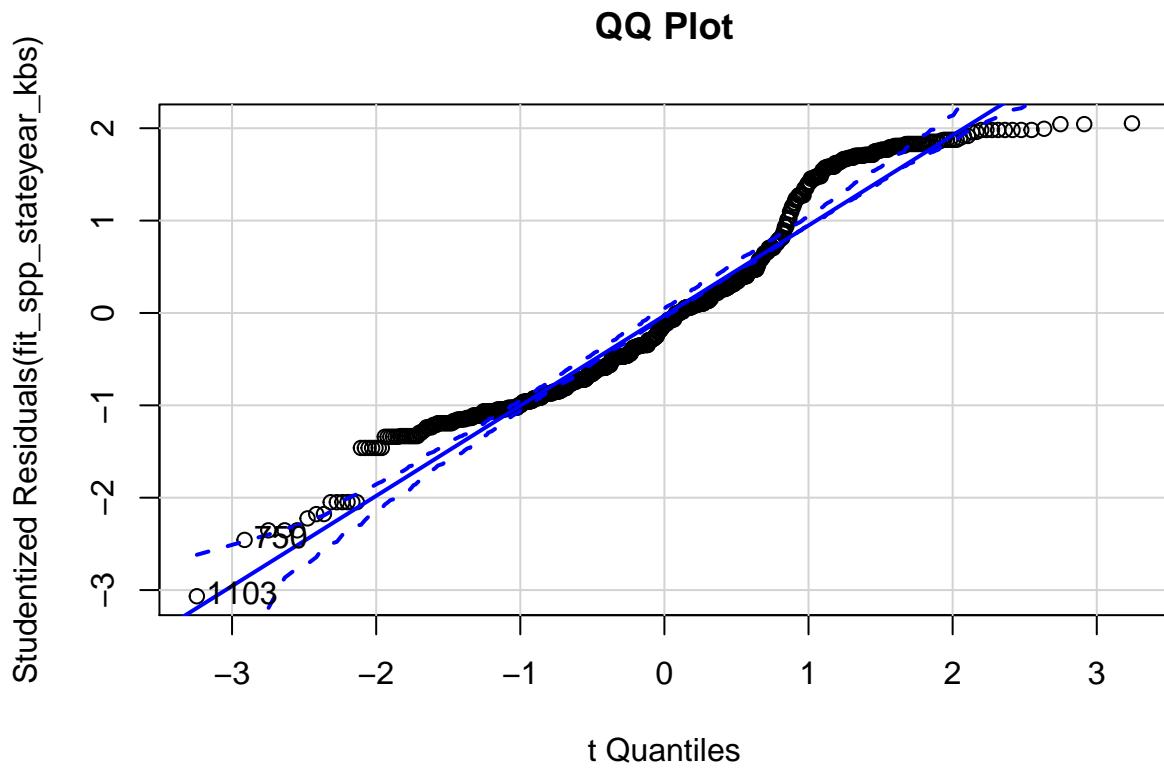
```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
## -----
##      Test          Statistic       pvalue
## -----
## Shapiro-Wilk      0.9542       0.0000
## Kolmogorov-Smirnov 0.0901       0.0000
## Cramer-von Mises 183.3113      0.0000
## Anderson-Darling 13.2382       0.0000
## -----
```

```
# KBS State and year model
fit_spp_stateyear_kbs <- lm(log(julian_min) ~ state + year_factor, data = kbs_sd_spp)
outlierTest(fit_spp_stateyear_kbs) # no outliers
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##           rstudent unadjusted p-value Bonferroni p
## 1103     -3.066819             0.0022357      NA
```

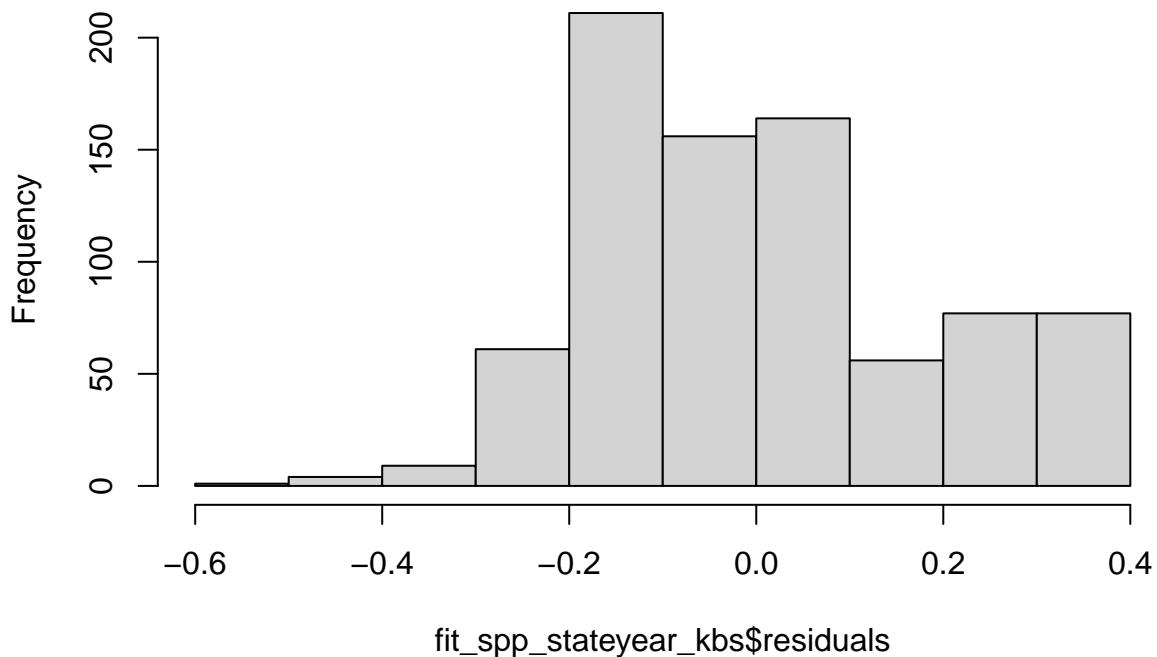
```
qqPlot(fit_spp_stateyear_kbs, main = "QQ Plot")
```



```
## 750 1103
## 433 642
```

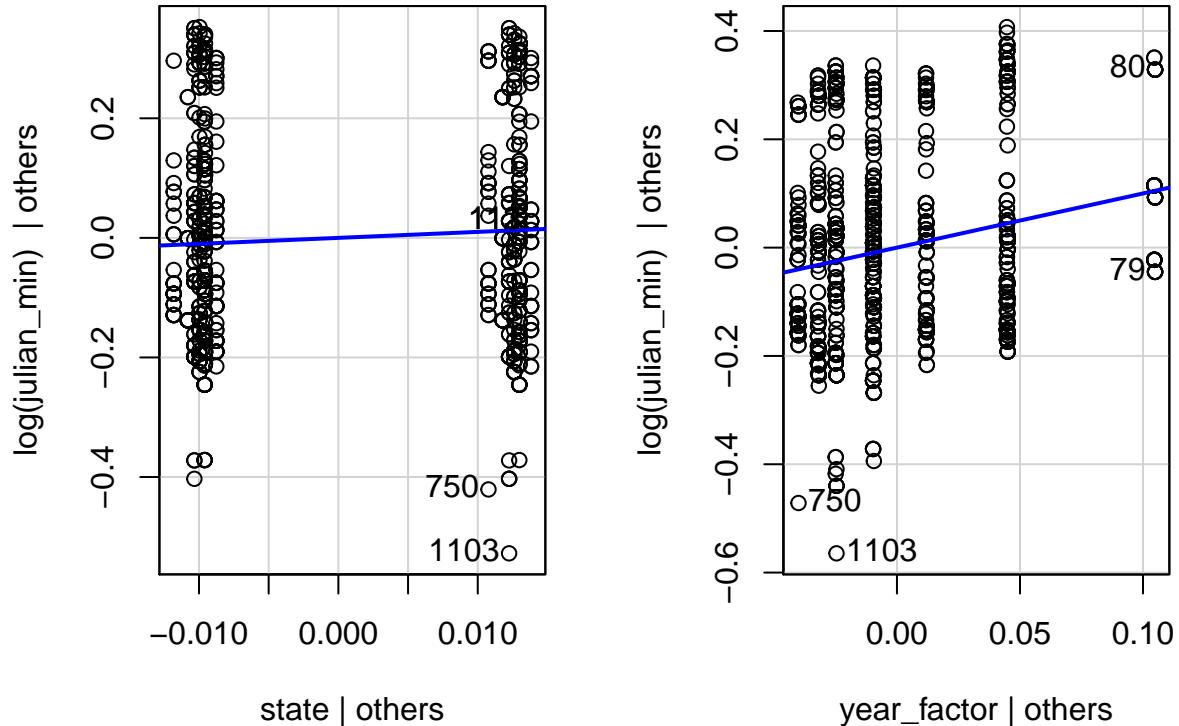
```
hist(fit_spp_stateyear_kbs$residuals)
```

Histogram of fit_spp_stateyear_kbs\$residuals



```
leveragePlots(fit_spp_stateyear_kbs)
```

Leverage Plots



```
ols_test_normality(fit_spp_stateyear_kbs)
```

```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
## -----
##      Test       Statistic     pvalue
## -----
## Shapiro-Wilk      0.9499    0.0000
## Kolmogorov-Smirnov 0.0903    0.0000
## Cramer-von Mises   184.0438   0.0000
## Anderson-Darling    14.1724    0.0000
## -----
```

```
# Interaction plot (ignore for now the repeated measures with species); see:
# https://cran.r-project.org/web/packages/interactions/vignettes/interactions.html
# and: https://interactions.jacob-long.com/
```

```
# these wont work below
fit3 <- lm(log(julian_min) ~ state + year_factor + species, data = kbs_sd_spp)
interact_plot(fit3, pred = year_factor, modx = state)
```

```
## Using data kbs_sd_spp from global environment. This could cause incorrect
## results if kbs_sd_spp has been altered since the model was fit. You can
## manually provide the data to the "data =" argument.
```

```

## Warning: year_factor and state are not included in an interaction with one another
## in the model.

## Error in interact_plot(fit3, pred = year_factor, modx = state): Focal predictor ("pred") cannot be a

fit4 <- lm(log(julian_min) ~ state * year_factor + species, data = kbs_sd_spp)
interact_plot(fit4, pred = year_factor, modx = state, mod2 = species)

## Using data kbs_sd_spp from global environment. This could cause incorrect
## results if kbs_sd_spp has been altered since the model was fit. You can
## manually provide the data to the "data =" argument.

## Warning: year_factor and state and species are not included in an interaction with
## one another in the model.

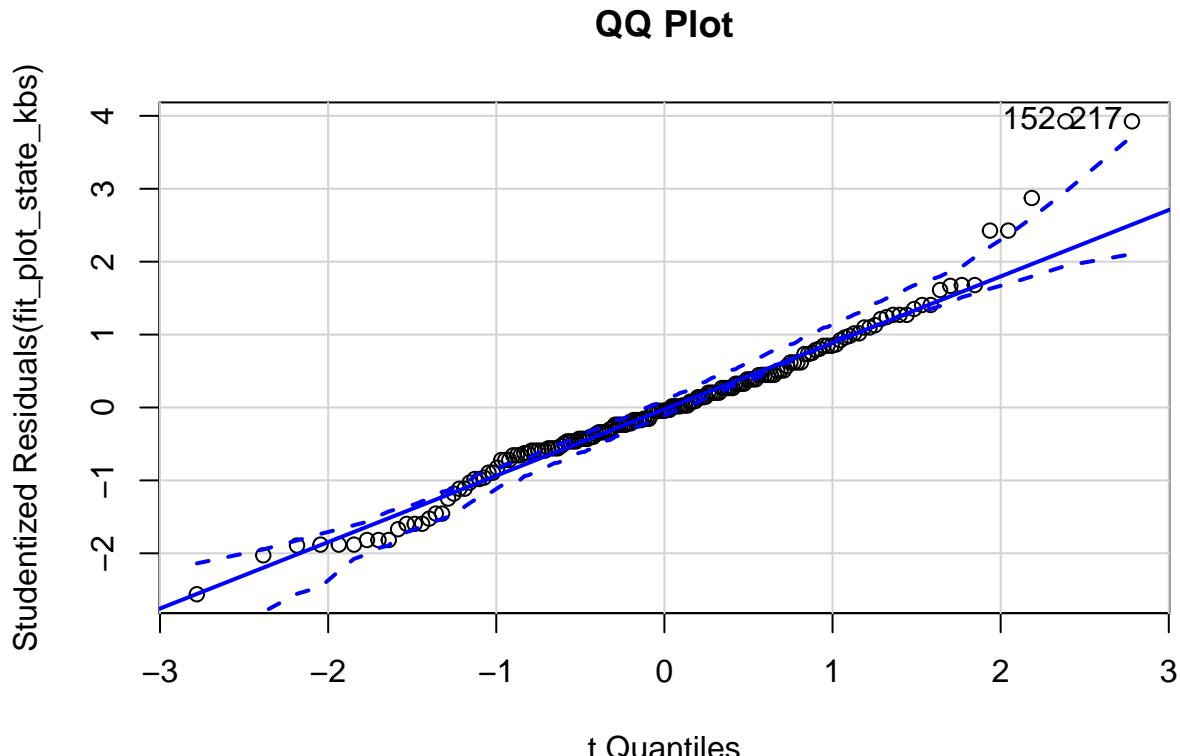
## Error in interact_plot(fit4, pred = year_factor, modx = state, mod2 = species): Focal predictor ("pr

# KBS Plot level data KBS State-only model
fit_plot_state_kbs <- lm(log(julian_min) ~ state, data = kbs_sd_plot)
outlierTest(fit_plot_state_kbs) # outliers - 152, 217

##      rstudent unadjusted p-value Bonferroni p
## 152 3.924672      0.00012813     0.021141
## 217 3.924672      0.00012813     0.021141

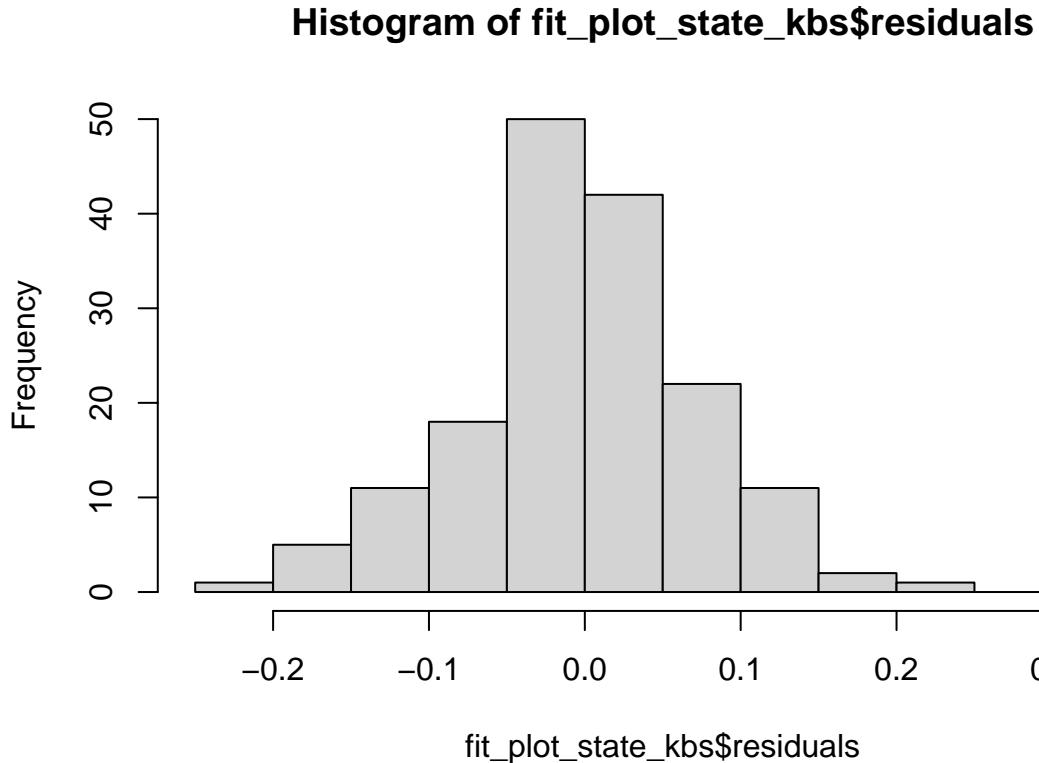
qqPlot(fit_plot_state_kbs, main = "QQ Plot")

```

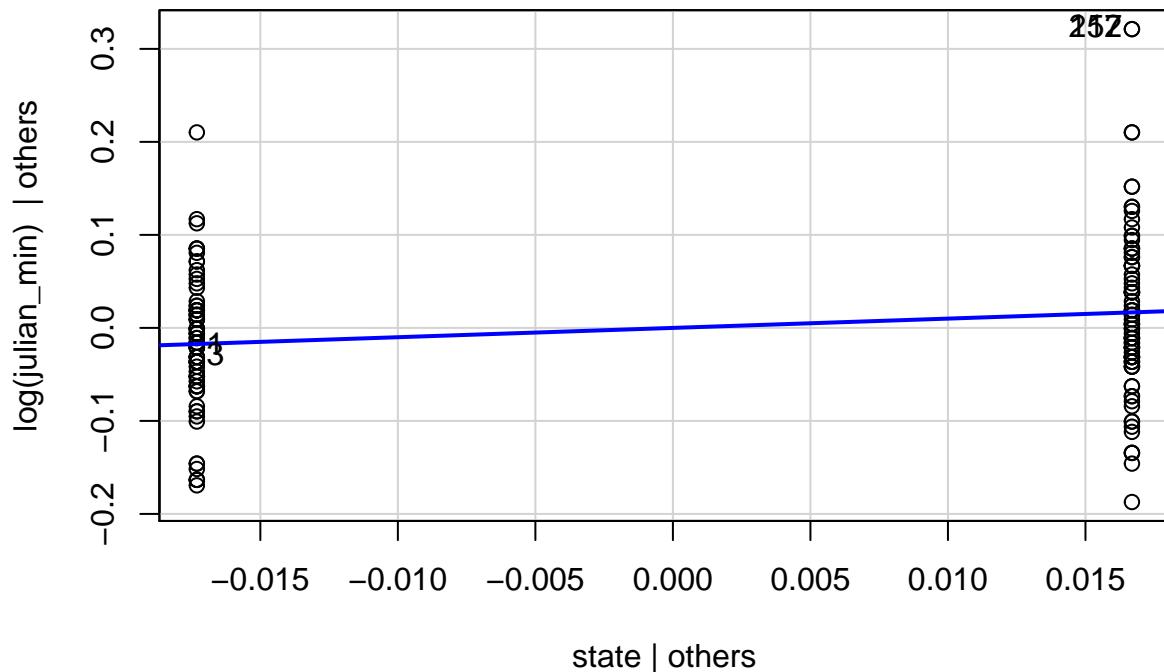


```
## 152 217  
## 84 119
```

```
hist(fit_plot_state_kbs$residuals)
```



```
leveragePlots(fit_plot_state_kbs)
```



```

ols_test_normality(fit_plot_state_kbs)

## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test

## -----
##              Test      Statistic     pvalue
## -----
## Shapiro-Wilk      0.9648    3e-04
## Kolmogorov-Smirnov   0.0744    0.3205
## Cramer-von Mises     46.9551   0.0000
## Anderson-Darling      1.1738    0.0045
## -----
# KBS State and year model
fit_plot_stateyear_kbs <- lm(log(julian_min) ~ state + year_factor, data = kbs_sd_plot)
outlierTest(fit_spp_stateyear_kbs) # no outliers

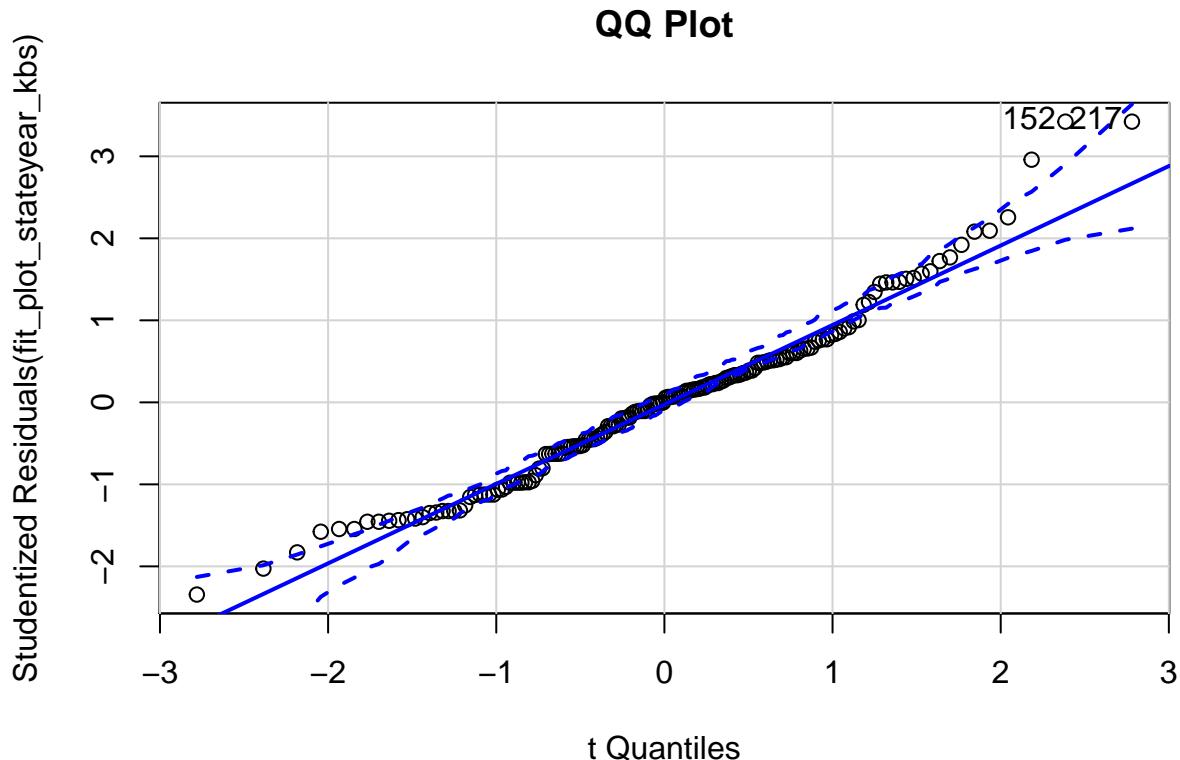
```

```

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##           rstudent unadjusted p-value Bonferroni p
## 1103 -3.066819          0.0022357        NA

```

```
qqPlot(fit_plot_stateyear_kbs, main = "QQ Plot")
```



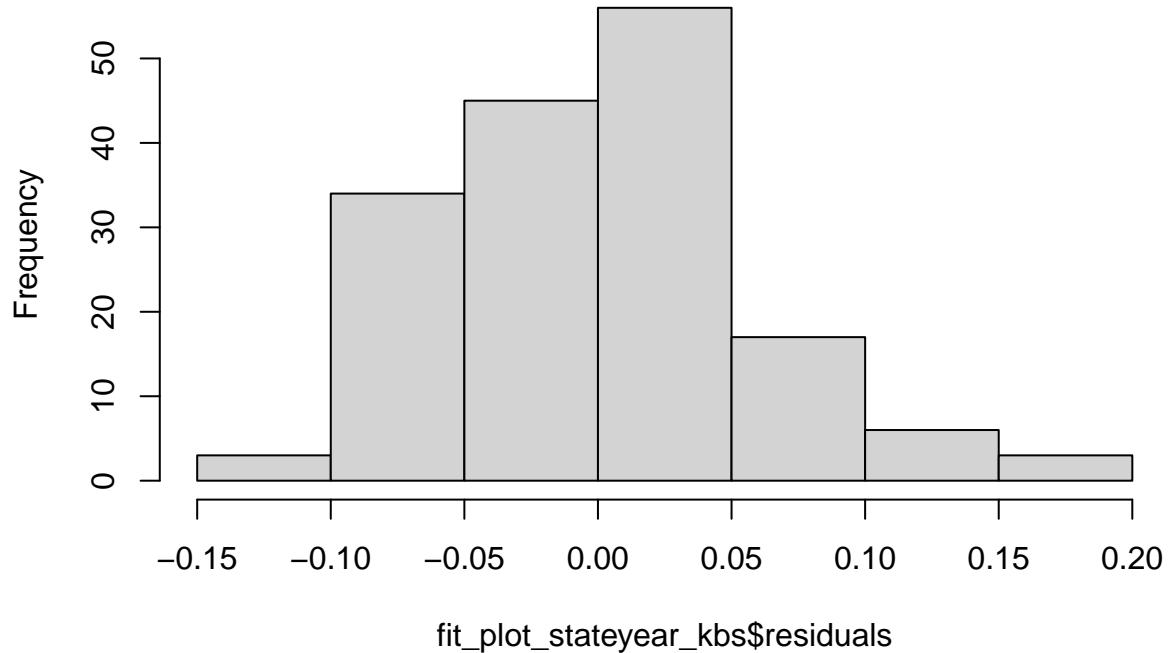
```

## 152 217
## 84 119

```

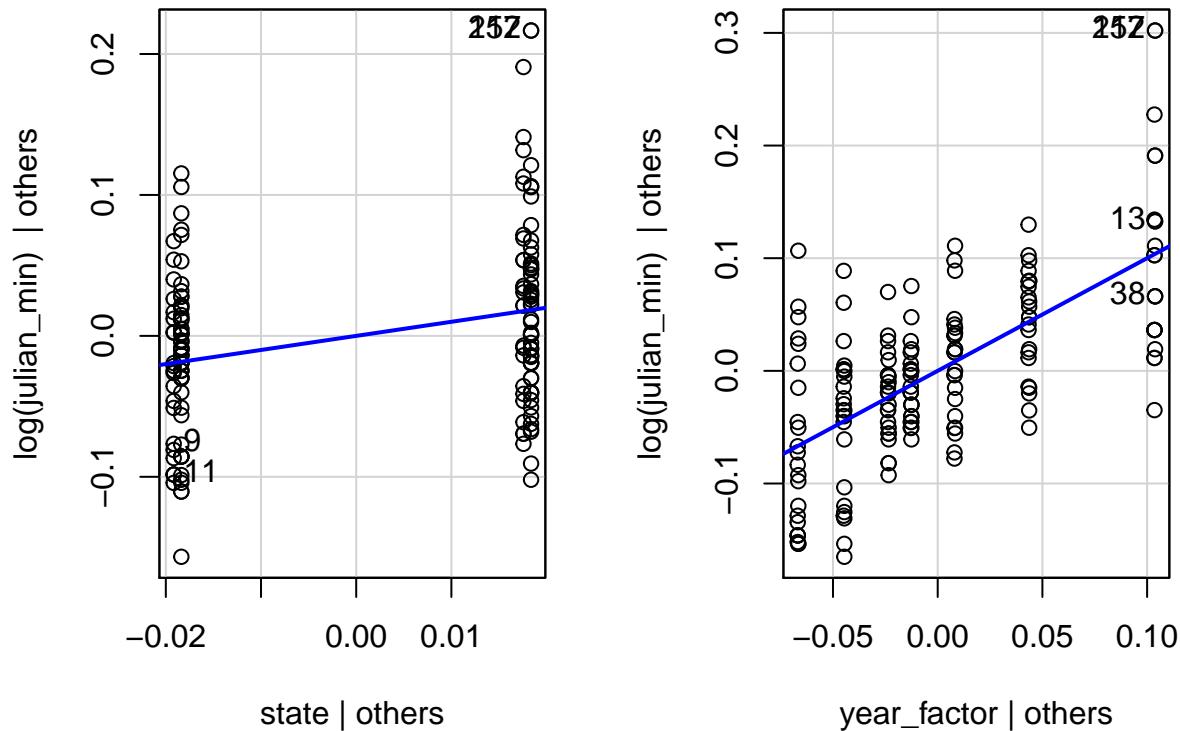
```
hist(fit_plot_stateyear_kbs$residuals)
```

Histogram of fit_plot_stateyear_kbs\$residuals



```
leveragePlots(fit_plot_stateyear_kbs)
```

Leverage Plots



```
ols_test_normality(fit_plot_stateyear_kbs)
```

```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
## -----
##          Test      Statistic     pvalue
## -----
## Shapiro-Wilk      0.9747    0.0042
## Kolmogorov-Smirnov   0.0632    0.5286
## Cramer-von Mises  48.4006    0.0000
## Anderson-Darling   0.8621    0.0263
## -----
```

```
# Interaction plot (ignore for now the repeated measures with species); see:
# https://cran.r-project.org/web/packages/interactions/vignettes/interactions.html
# and: https://interactions.jacob-long.com/
```

```
fit3 <- lm(log(julian_min) ~ state * year_factor, data = kbs_sd_plot)
interact_plot(fit3, pred = year_factor, modx = state) # this doesn't work
```

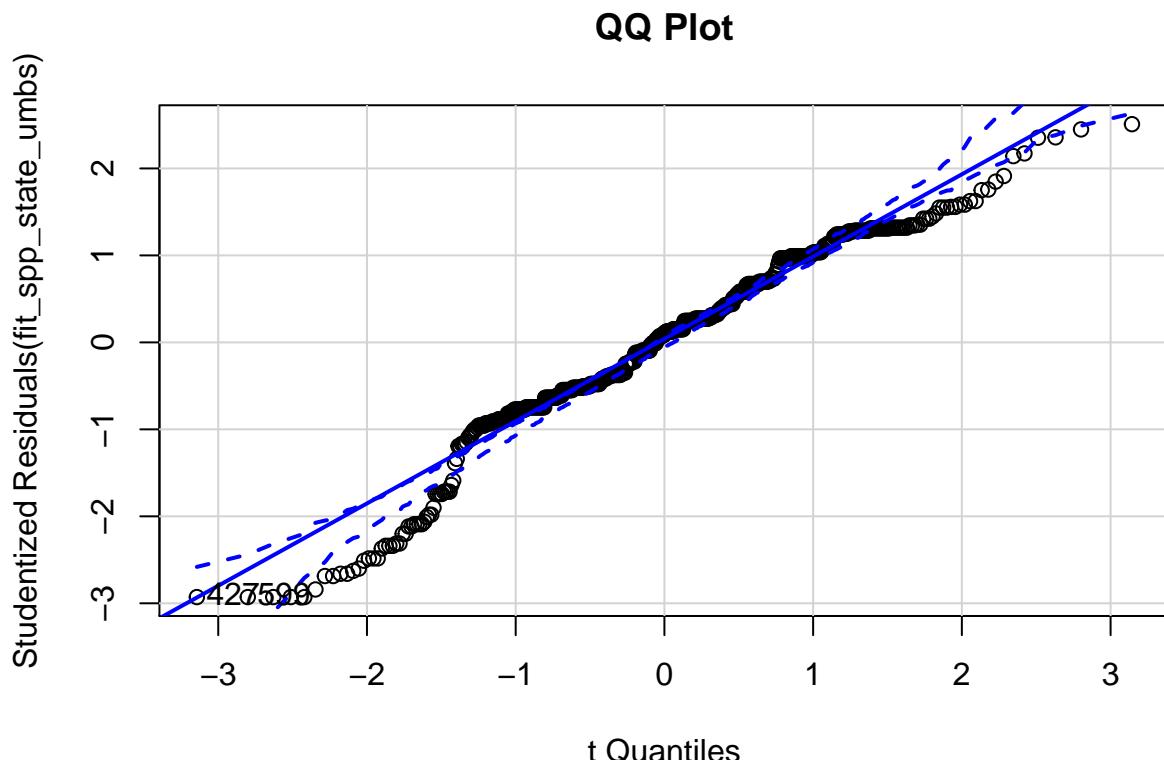
```
## Using data kbs_sd_plot from global environment. This could cause incorrect
## results if kbs_sd_plot has been altered since the model was fit. You can
## manually provide the data to the "data =" argument.
```

```
## Error in interact_plot(fit3, pred = year_factor, modx = state): Focal predictor ("pred") cannot be a
```

```
# UMBS species level data UMBS State-only model
fit_spp_state_umbs <- lm(log(julian_min) ~ state, data = umbs_sd_spp)
outlierTest(fit_spp_state_umbs) # no outliers
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferroni p
## 427 -2.928397          0.003544        NA
```

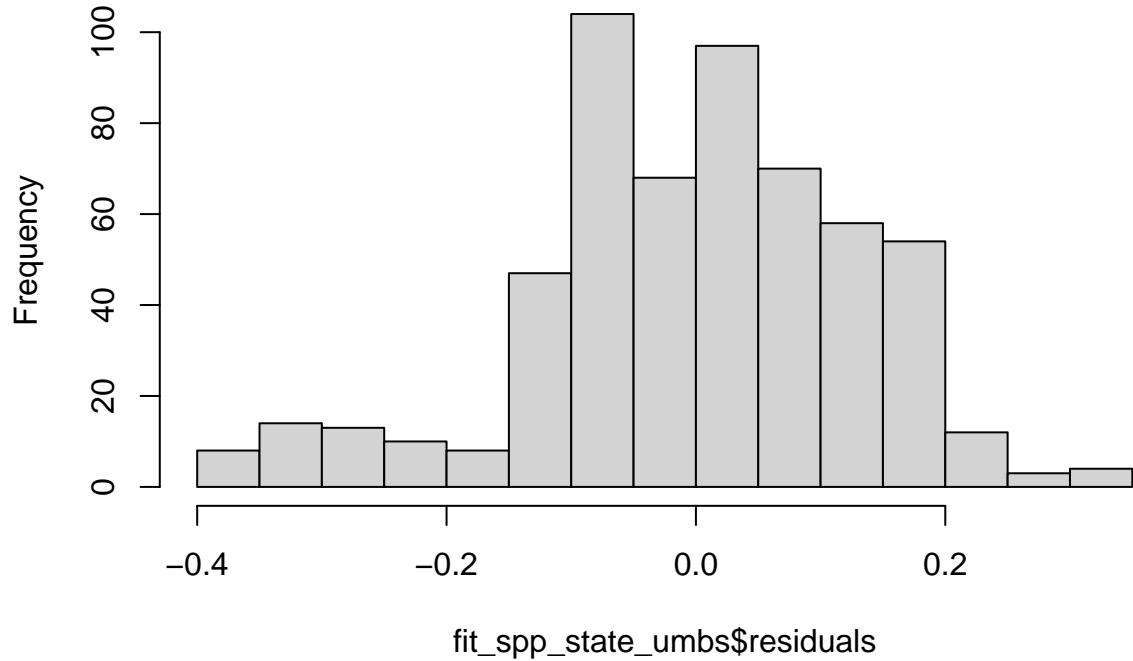
```
qqPlot(fit_spp_state_umbs, main = "QQ Plot")
```



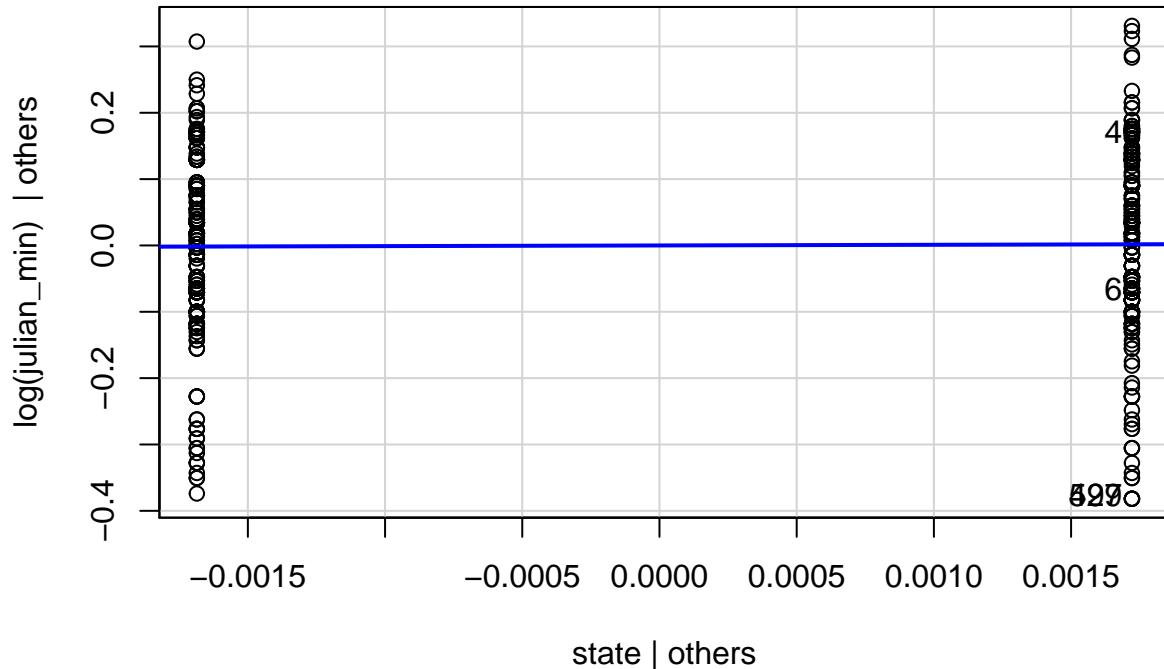
```
## 427 599
## 194 260
```

```
hist(fit_spp_state_umbs$residuals)
```

Histogram of fit_spp_state_umbs\$residuals



```
leveragePlots(fit_spp_state_umbs)
```



```
ols_test_normality(fit_spp_state_umbs)
```

```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test
```

```

## -----
##          Test      Statistic     pvalue
## -----
## Shapiro-Wilk       0.97    0.0000
## Kolmogorov-Smirnov 0.0644   0.0176
## Cramer-von Mises   144.572  0.0000
## Anderson-Darling    4.0617  0.0000
## -----

```

```

# UMBS State and year model
fit_spp_stateyear_umbs <- lm(log(julian_min) ~ state + year_factor, data = umbs_sd_spp)
outlierTest(fit_spp_stateyear_umbs) # no outliers

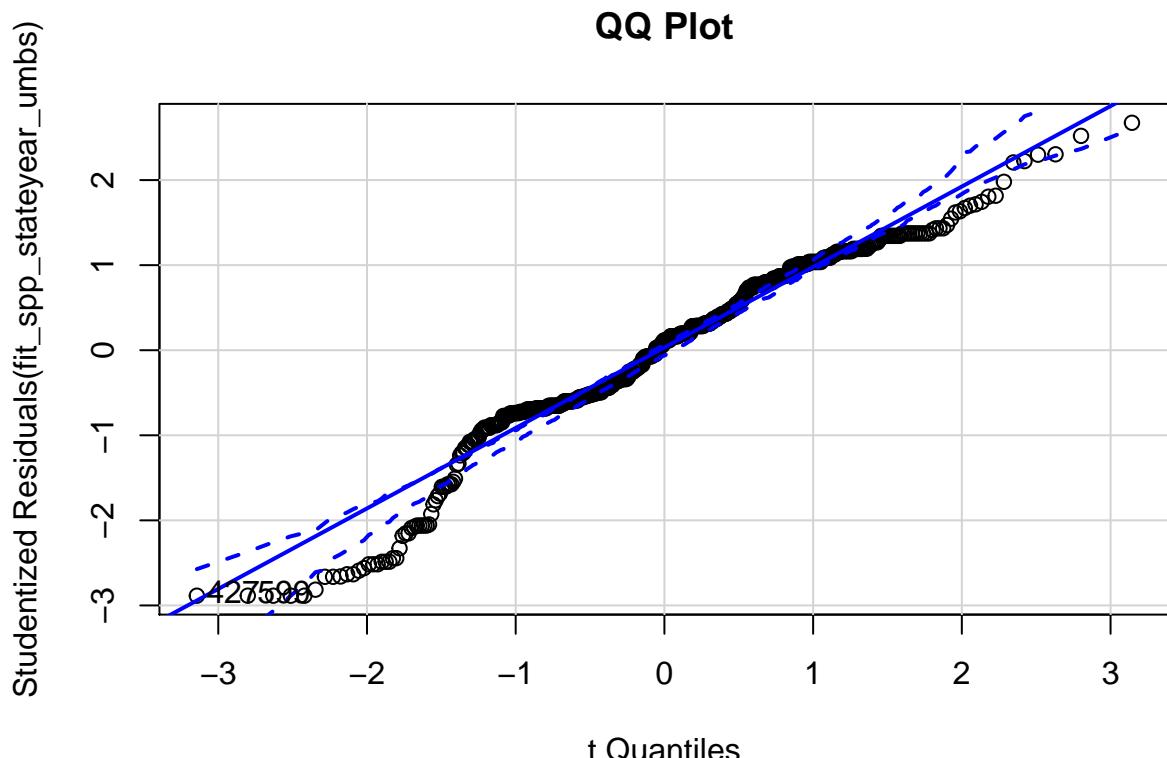
```

```

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##           rstudent unadjusted p-value Bonferroni p
## 427 -2.885452          0.0040586        NA

```

```
qqPlot(fit_spp_stateyear_umbs, main = "QQ Plot")
```



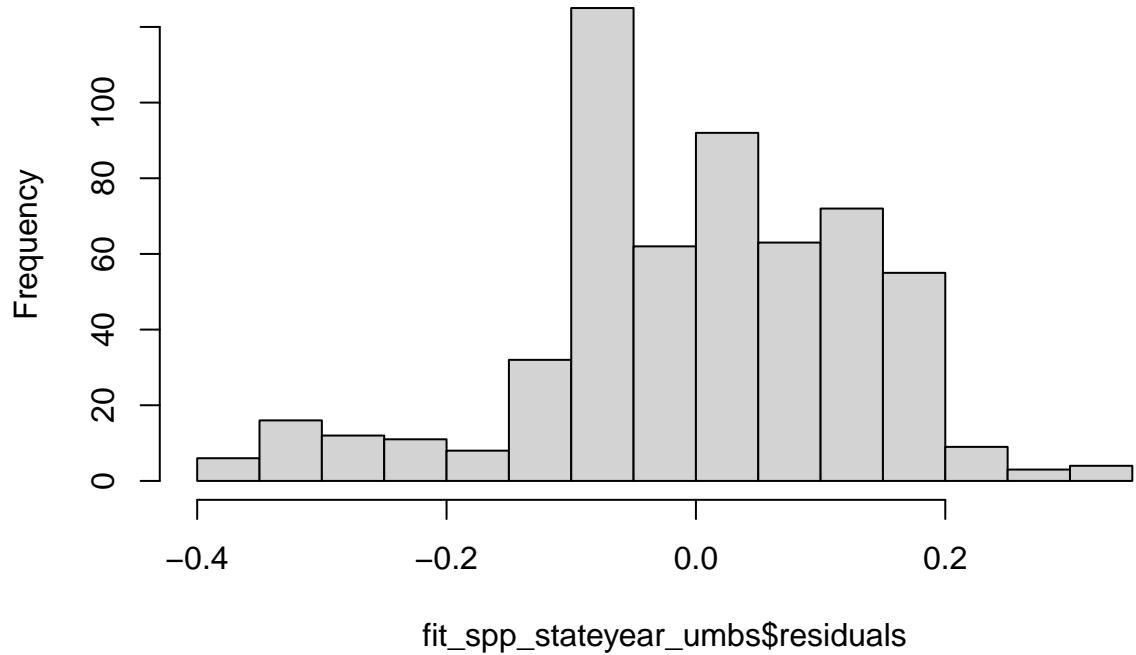
```

## 427 599
## 194 260

```

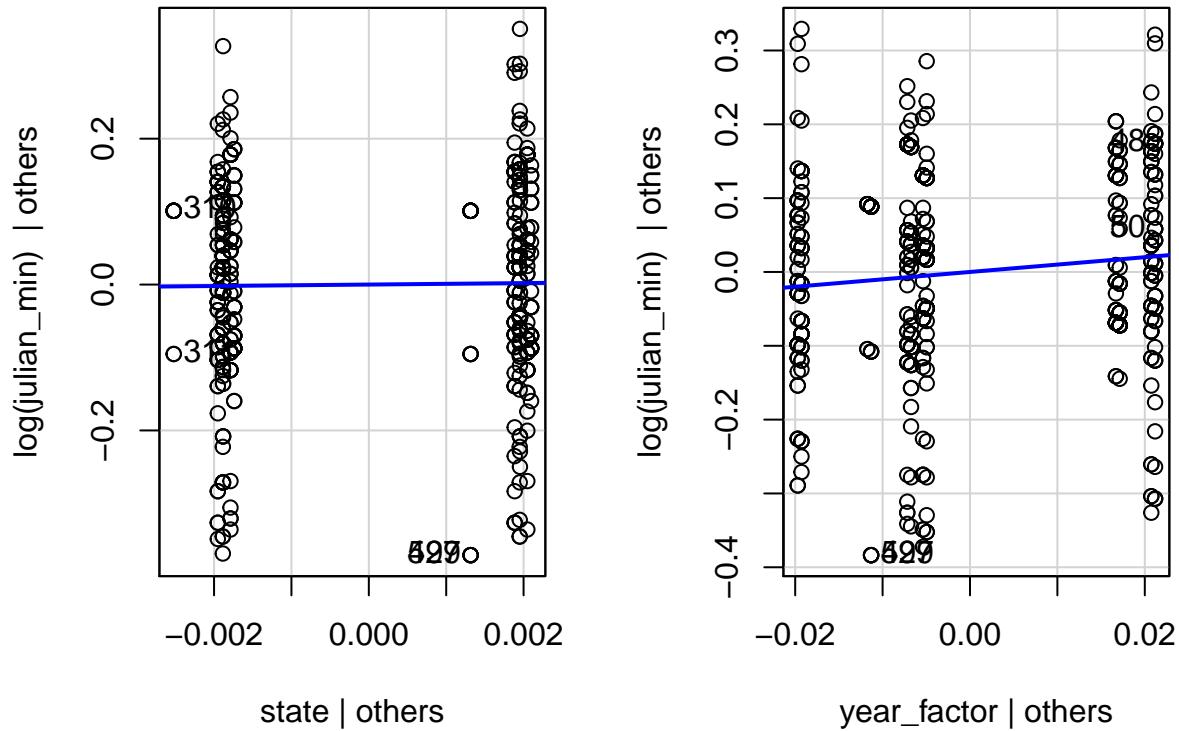
```
hist(fit_spp_stateyear_umbs$residuals)
```

Histogram of fit_spp_stateyear_umbs\$residuals



```
leveragePlots(fit_spp_stateyear_umbs)
```

Leverage Plots



```

ols_test_normality(fit_spp_stateyear_umbs)

## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test

## -----
##          Test           Statistic      pvalue
## -----
## Shapiro-Wilk       0.9681      0.0000
## Kolmogorov-Smirnov 0.0811      0.0011
## Cramer-von Mises   144.5732     0.0000
## Anderson-Darling    4.6029      0.0000
## -----


# Interaction plot (ignore for now the repeated measures with species); see:
# https://cran.r-project.org/web/packages/interactions/vignettes/interactions.html
# and: https://interactions.jacob-long.com/

# these wont work below
fit3_spp_umbs <- lm(log(julian_min) ~ state + year_factor + species, data = umbs_sd_spp)
interact_plot(fit3, pred = year_factor, modx = state)

## Using data kbs_sd_plot from global environment. This could cause incorrect
## results if kbs_sd_plot has been altered since the model was fit. You can
## manually provide the data to the "data =" argument.

## Error in interact_plot(fit3, pred = year_factor, modx = state): Focal predictor ("pred") cannot be a

fit4_spp_umbs <- lm(log(julian_min) ~ state * year_factor + species, data = umbs_sd_spp)
interact_plot(fit4, pred = year_factor, modx = state, mod2 = species)

## Using data kbs_sd_spp from global environment. This could cause incorrect
## results if kbs_sd_spp has been altered since the model was fit. You can
## manually provide the data to the "data =" argument.

## Warning: year_factor and state and species are not included in an interaction with
## one another in the model.

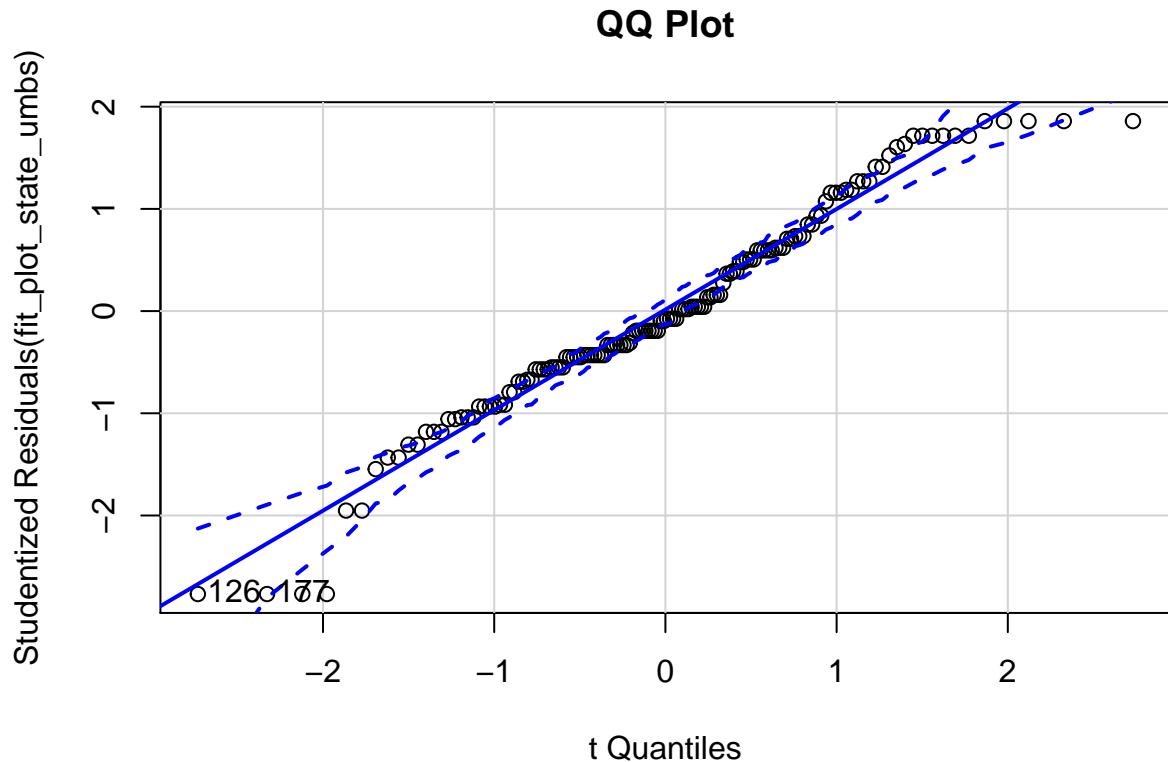
## Error in interact_plot(fit4, pred = year_factor, modx = state, mod2 = species): Focal predictor ("pr

# Plot level data UMBS State-only model
fit_plot_state_umbs <- lm(log(julian_min) ~ state, data = umbs_sd_plot)
outlierTest(fit_plot_state_umbs) # no outliers

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferroni p
## 126 -2.768466      0.0064122      0.89771

```

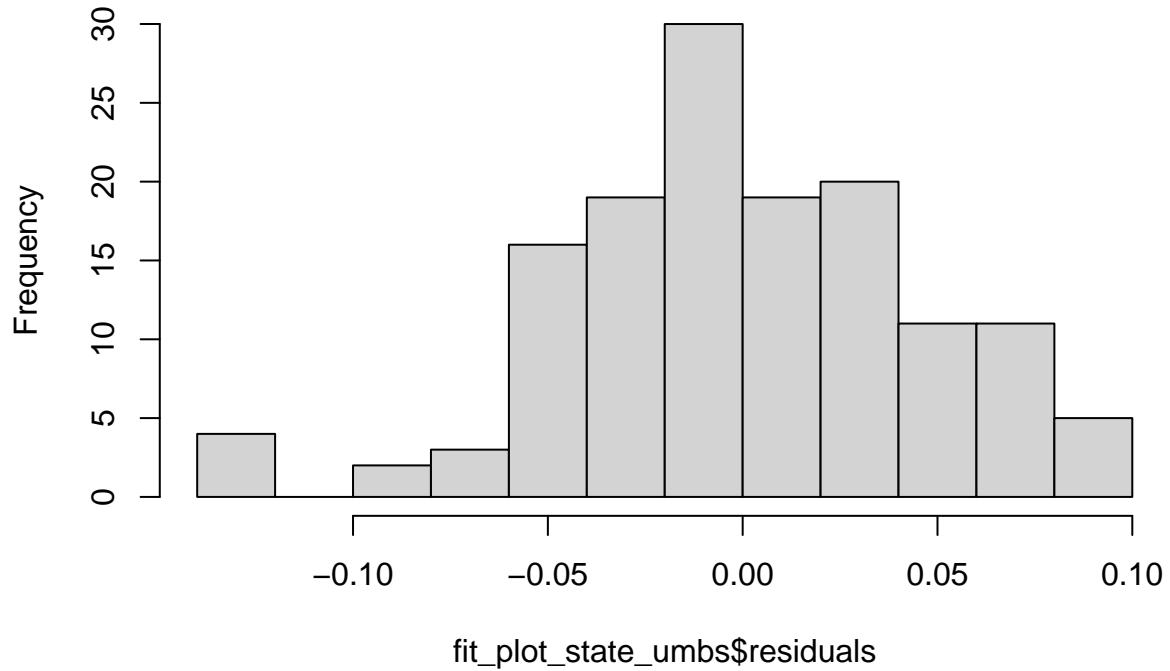
```
qqPlot(fit_plot_state_umbs, main = "QQ Plot")
```



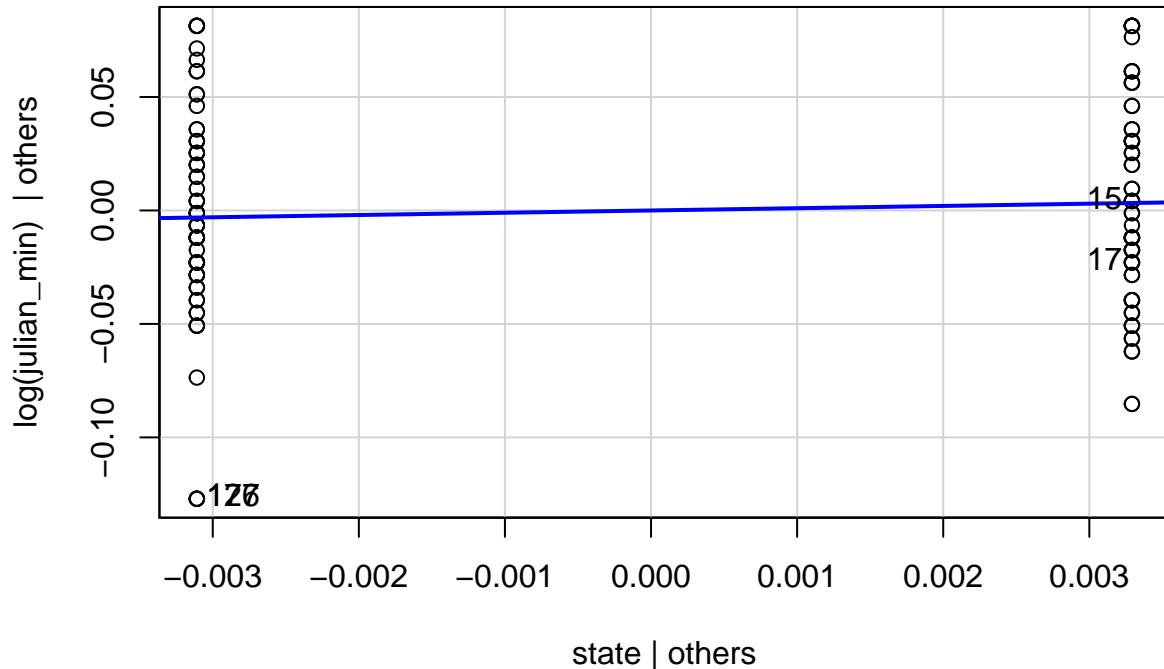
```
## 126 177  
## 57 80
```

```
hist(fit_plot_state_umbs$residuals)
```

Histogram of fit_plot_state_umbs\$residuals



```
leveragePlots(fit_plot_state_umbs)
```



```
ols_test_normality(fit_plot_state_umbs)
```

```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test
```

```

## -----
##          Test      Statistic     pvalue
## -----
## Shapiro-Wilk      0.9712    0.0046
## Kolmogorov-Smirnov 0.0761    0.3926
## Cramer-von Mises   42.4719   0.0000
## Anderson-Darling    0.9037    0.0207
## -----
# UMBS State and year model
fit_plot_stateyear_umbs <- lm(log(julian_min) ~ state + year_factor, data = umbs_sd_plot)
outlierTest(fit_spp_stateyear_umbs)

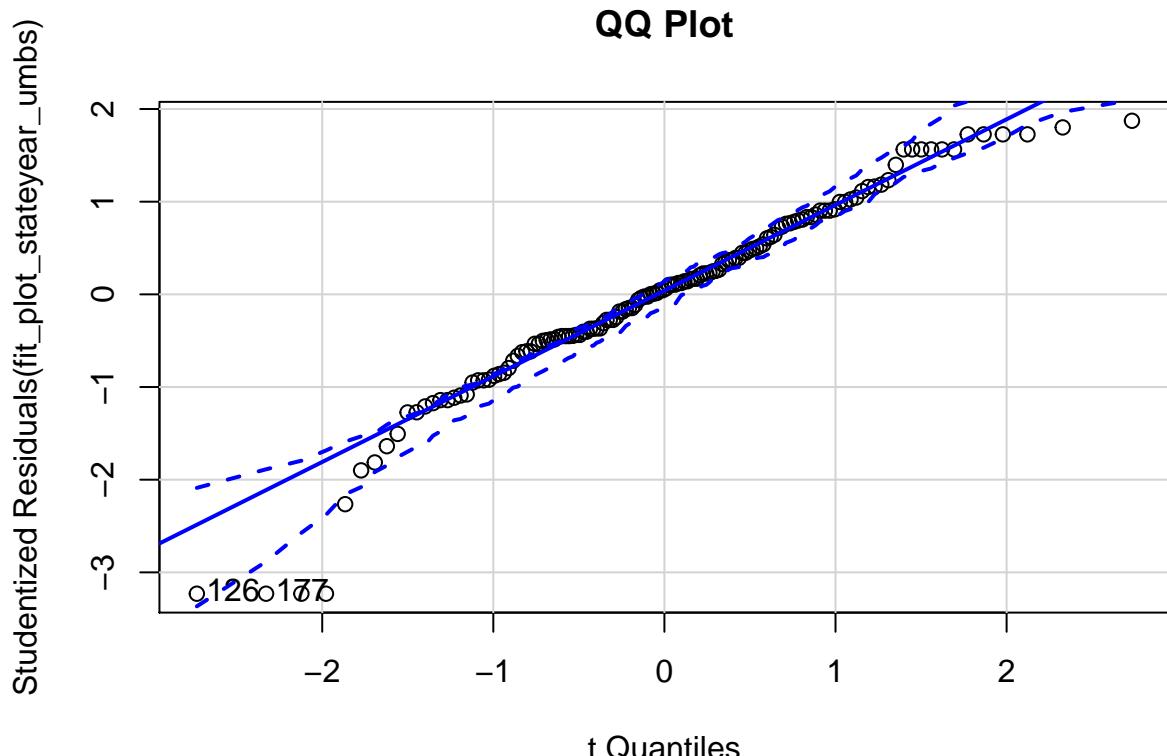
```

```

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##           rstudent unadjusted p-value Bonferroni p
## 427 -2.885452          0.0040586        NA

```

```
qqPlot(fit_plot_stateyear_umbs, main = "QQ Plot")
```



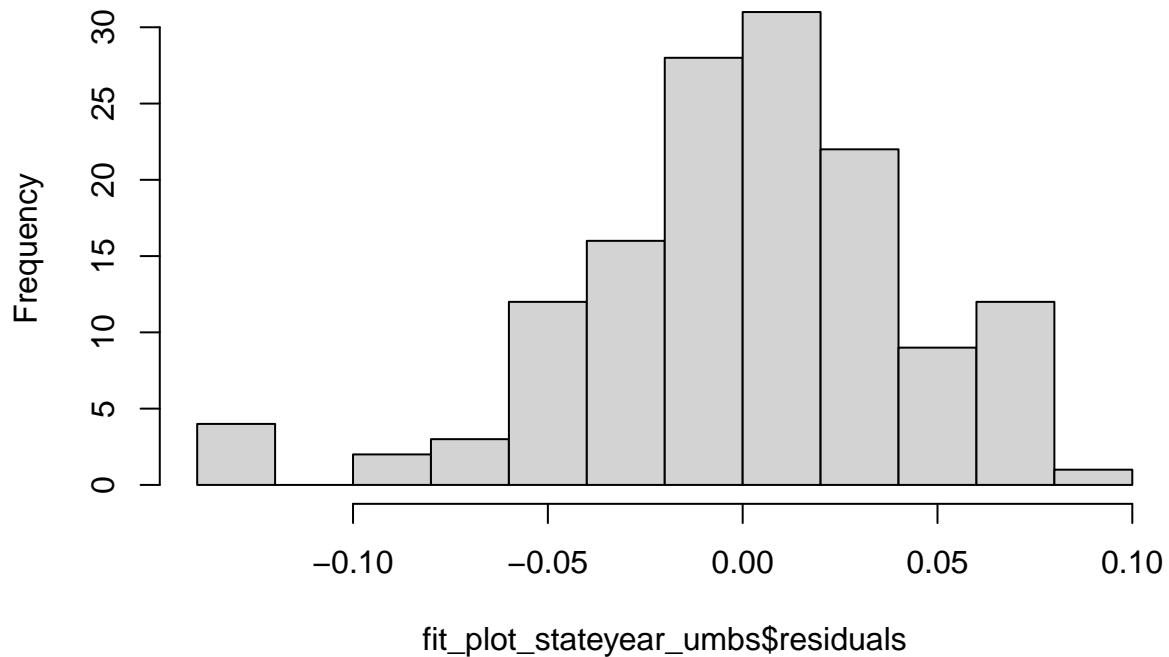
```

## 126 177
## 57 80

```

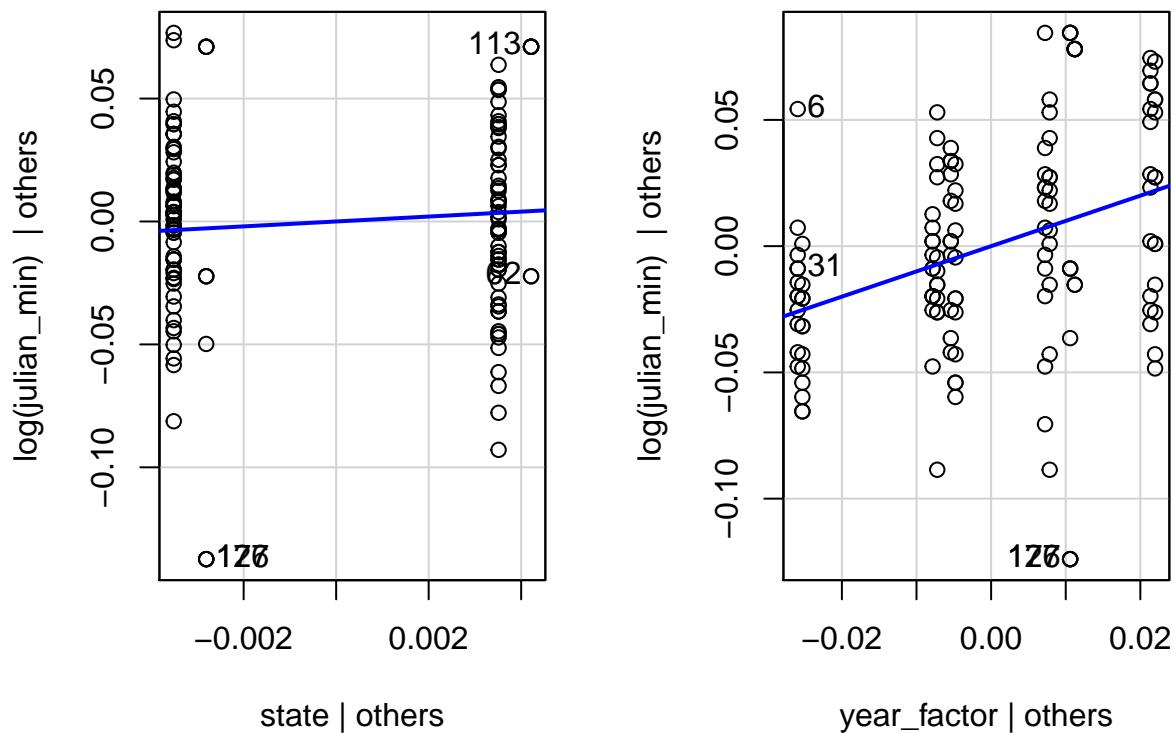
```
hist(fit_plot_stateyear_umbs$residuals)
```

Histogram of fit_plot_stateyear_umbs\$residuals



```
leveragePlots(fit_plot_stateyear_umbs)
```

Leverage Plots



```

ols_test_normality(fit_plot_stateyear_umbs)

## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test

## -----
##          Test      Statistic     pvalue
## -----
## Shapiro-Wilk      0.9588    3e-04
## Kolmogorov-Smirnov   0.0754    0.4035
## Cramer-von Mises    42.8802   0.0000
## Anderson-Darling     0.872     0.0248
## -----
```

Interaction plot (ignore for now the repeated measures with species); see:
<https://cran.r-project.org/web/packages/interactions/vignettes/interactions.html>
and: <https://interactions.jacob-long.com/>

```

fit3_plot_umbs <- lm(log(julian_min) ~ state * year_factor, data = umbs_sd_plot)
interact_plot(fit3_plot_umbs, pred = year_factor, modx = state)
```

Using data umbs_sd_plot from global environment. This could cause incorrect
results if umbs_sd_plot has been altered since the model was fit. You can
manually provide the data to the "data =" argument.

Error in interact_plot(fit3_plot_umbs, pred = year_factor, modx = state): Focal predictor ("pred") c

MODEL BUILDING

KBS Species-level Mixed Effects Models

Start by replicating (almost) what we did in the Decologia 2018 paper. The only
difference here is that we have multiple years, so we are also including year
as a fixed effect and as an interactive term. Our goal here is to find a model
that is the best fit to the data. We also want to find a model that is the most
parsimonious (one that has the fewest parameters).

Do we need to include plot as a random effect with the UMBS models?
mod1 <- lmer(log(julian_min) ~ state * year_factor + insecticide * year_factor +
 (1 | species) + (1 | plot), kbs_sd_spp, REML = FALSE)

boundary (singular) fit: see ?isSingular

```

mod2 <- lmer(log(julian_min) ~ state * year_factor + insecticide * year_factor +
  (1 | species), kbs_sd_spp, REML = FALSE)
# Run analysis of variance on each model (see this for more explanation on how
# anova on a linear mixed effects model is similar to an anova on a regular
# linear model: https://m-clark.github.io/docs/mixedModels/anovamixed.html)
anova(mod1)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##                               Sum Sq  Mean Sq NumDF DenDF F value Pr(>F)
## state                      0.00671 0.006706     1 794.06 2.8796 0.0901 .
## year_factor                 0.72591 0.120986     6 794.41 51.9480 <2e-16 ***
## insecticide                  0.00474 0.004742     1 794.09 2.0359 0.1540
## state:year_factor            0.00928 0.001546     6 794.03 0.6640 0.6789
## year_factor:insecticide      0.01238 0.002063     6 794.06 0.8857 0.5048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
anova(mod2)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##                               Sum Sq  Mean Sq NumDF DenDF F value Pr(>F)
## state                      0.00671 0.006706     1 794.06 2.8796 0.0901 .
## year_factor                 0.72591 0.120986     6 794.41 51.9480 <2e-16 ***
## insecticide                  0.00474 0.004742     1 794.09 2.0359 0.1540
## state:year_factor            0.00928 0.001546     6 794.03 0.6640 0.6789
## year_factor:insecticide      0.01238 0.002063     6 794.06 0.8857 0.5048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# Run an ANOVA to test the 2 models to test whether the more complex model is
# significantly better at capturing the data than the simpler model. If the
# resulting p-value is sufficiently low (usually less than 0.05), we conclude
# that the more complex model is significantly better than the simpler model, and
# thus favor the more complex model. If the p-value is not sufficiently low
# (usually greater than 0.05), we should favor the simpler model.
# https://bookdown.org/ndphillips/YaRrrr/comparing-regression-models-with-anova.html
anova(mod1, mod2) # models are not different from each other so go with simpler model = model 2

```

```

## Data: kbs_sd_spp
## Models:
## mod2: log(julian_min) ~ state * year_factor + insecticide * year_factor +
## mod2:          (1 | species)
## mod1: log(julian_min) ~ state * year_factor + insecticide * year_factor +
## mod1:          (1 | species) + (1 | plot)
##      npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## mod2   23 -2458.1 -2349.9 1252.1  -2504.1
## mod1   24 -2456.1 -2343.2 1252.1  -2504.1      0  1           1

```

```
summary(mod1)
```

```

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state * year_factor + insecticide * year_factor +
##           (1 | species) + (1 | plot)
## Data: kbs_sd_spp
##
##      AIC      BIC logLik deviance df.resid
## -2456.1 -2343.2 1252.1 -2504.1      792
## 

```

```

## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -4.5749 -0.5062 -0.0168  0.4582  7.6160
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   plot     (Intercept) 0.000000 0.00000
##   species (Intercept) 0.040350 0.20087
##   Residual           0.002329 0.04826
## Number of obs: 816, groups: plot, 24; species, 22
##
## Fixed effects:
##                               Estimate Std. Error      df t value
##   ## (Intercept)            5.36220  0.04483 25.74146 119.607
##   ## stateambient          0.02061  0.01433 793.79819   1.438
##   ## year_factor2         -0.09614  0.01421 793.86859  -6.768
##   ## year_factor3         -0.10098  0.01466 794.21173  -6.887
##   ## year_factor4         -0.10740  0.01515 793.88740  -7.088
##   ## year_factor5         -0.07735  0.01543 793.82667  -5.014
##   ## year_factor6         -0.06531  0.01461 793.84647  -4.470
##   ## year_factor7         -0.07777  0.01661 793.88823  -4.683
##   ## insecticideno_insects 0.03017  0.01427 793.79100   2.114
##   ## stateambient:year_factor2 -0.02182  0.01594 793.83440  -1.369
##   ## stateambient:year_factor3 -0.01223  0.01638 793.87744  -0.747
##   ## stateambient:year_factor4 -0.02685  0.01700 793.95450  -1.580
##   ## stateambient:year_factor5 -0.01148  0.01751 793.81399  -0.656
##   ## stateambient:year_factor6 -0.01401  0.01666 793.96220  -0.841
##   ## stateambient:year_factor7 -0.01151  0.01871 793.82712  -0.615
##   ## year_factor2:insecticideno_insects -0.03334  0.01584 793.79868  -2.105
##   ## year_factor3:insecticideno_insects -0.03443  0.01634 793.99170  -2.107
##   ## year_factor4:insecticideno_insects -0.02605  0.01686 794.01578  -1.545
##   ## year_factor5:insecticideno_insects -0.02812  0.01734 793.80149  -1.622
##   ## year_factor6:insecticideno_insects -0.02782  0.01656 793.81823  -1.680
##   ## year_factor7:insecticideno_insects -0.02310  0.01887 793.87303  -1.224
##                               Pr(>|t|)
##   ## (Intercept)            < 2e-16 ***
##   ## stateambient          0.1507
##   ## year_factor2          2.53e-11 ***
##   ## year_factor3          1.16e-11 ***
##   ## year_factor4          3.01e-12 ***
##   ## year_factor5          6.59e-07 ***
##   ## year_factor6          8.95e-06 ***
##   ## year_factor7          3.32e-06 ***
##   ## insecticideno_insects 0.0348 *
##   ## stateambient:year_factor2 0.1714
##   ## stateambient:year_factor3 0.4555
##   ## stateambient:year_factor4 0.1146
##   ## stateambient:year_factor5 0.5123
##   ## stateambient:year_factor6 0.4006
##   ## stateambient:year_factor7 0.5388
##   ## year_factor2:insecticideno_insects 0.0356 *
##   ## year_factor3:insecticideno_insects 0.0354 *
##   ## year_factor4:insecticideno_insects 0.1228
##   ## year_factor5:insecticideno_insects 0.1053

```

```

## year_factor6:insecticideno_insects   0.0934 .
## year_factor7:insecticideno_insects   0.2212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 21 > 12.
## Use print(x, correlation=TRUE)  or
##      vcov(x)           if you need it

## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

summary(mod2)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state * year_factor + insecticide * year_factor +
##          (1 | species)
## Data: kbs_sd_spp
##
##       AIC     BIC   logLik deviance df.resid
## -2458.1 -2349.9  1252.1   -2504.1      793
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.5749 -0.5062 -0.0168  0.4582  7.6160
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## species (Intercept) 0.040350 0.20087
## Residual            0.002329 0.04826
## Number of obs: 816, groups: species, 22
##
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                  5.36220  0.04483 25.74146 119.607
## stateambient                 0.02061  0.01433 793.79818  1.438
## year_factor2                -0.09614  0.01421 793.86859 -6.768
## year_factor3                -0.10098  0.01466 794.21173 -6.887
## year_factor4                -0.10740  0.01515 793.88740 -7.088
## year_factor5                -0.07735  0.01543 793.82667 -5.014
## year_factor6                -0.06531  0.01461 793.84647 -4.470
## year_factor7                -0.07777  0.01661 793.88823 -4.683
## insecticideno_insects       0.03017  0.01427 793.79100  2.114
## stateambient:year_factor2  -0.02182  0.01594 793.83440 -1.369
## stateambient:year_factor3  -0.01223  0.01638 793.87744 -0.747
## stateambient:year_factor4  -0.02685  0.01700 793.95450 -1.580
## stateambient:year_factor5  -0.01148  0.01751 793.81399 -0.656
## stateambient:year_factor6  -0.01401  0.01666 793.96220 -0.841
## stateambient:year_factor7  -0.01151  0.01871 793.82712 -0.615
## year_factor2:insecticideno_insects -0.03334  0.01584 793.79868 -2.105
## year_factor3:insecticideno_insects -0.03443  0.01634 793.99170 -2.107

```

```

## year_factor4:insecticideno_insects -0.02605  0.01686 794.01578 -1.545
## year_factor5:insecticideno_insects -0.02812  0.01734 793.80149 -1.622
## year_factor6:insecticideno_insects -0.02782  0.01656 793.81823 -1.680
## year_factor7:insecticideno_insects -0.02310  0.01887 793.87303 -1.224
##
##                                     Pr(>|t|)
## (Intercept)                  < 2e-16 ***
## stateambient                 0.1507
## year_factor2                2.53e-11 ***
## year_factor3                1.16e-11 ***
## year_factor4                3.01e-12 ***
## year_factor5                6.59e-07 ***
## year_factor6                8.95e-06 ***
## year_factor7                3.32e-06 ***
## insecticideno_insects        0.0348 *
## stateambient:year_factor2   0.1714
## stateambient:year_factor3   0.4555
## stateambient:year_factor4   0.1146
## stateambient:year_factor5   0.5123
## stateambient:year_factor6   0.4006
## stateambient:year_factor7   0.5388
## year_factor2:insecticideno_insects 0.0356 *
## year_factor3:insecticideno_insects 0.0354 *
## year_factor4:insecticideno_insects 0.1228
## year_factor5:insecticideno_insects 0.1053
## year_factor6:insecticideno_insects 0.0934 .
## year_factor7:insecticideno_insects 0.2212
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

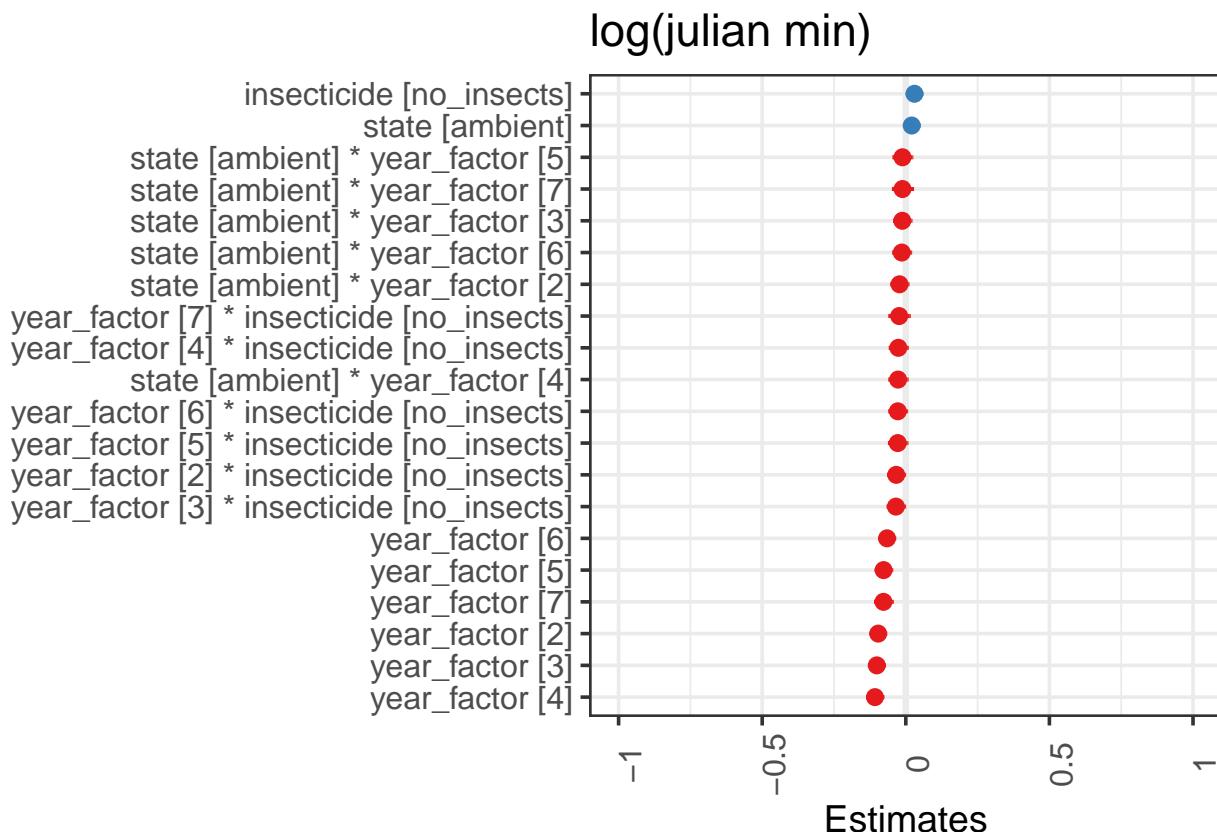
##
## Correlation matrix not shown by default, as p = 21 > 12.
## Use print(x, correlation=TRUE)  or
##      vcov(x)      if you need it

```

```

# Next, plot the model. There are multiple variables but here's one way to do it
# based on this package sjPlot:
# https://strengejacke.github.io/sjPlot/articles/plot\_model\_estimates.html
# Annoyingly, this package somehow overwrites the factor order in its plotting so
# we will have to modify the code to get warmed = red. I haven't figured this out
# yet. It does seem to work on some of the plots. hmm. ?plot_model Plot the fixed
# effects estimates for different models these are the fixed effects estimates
plot_model(mod2, sort.est = TRUE)

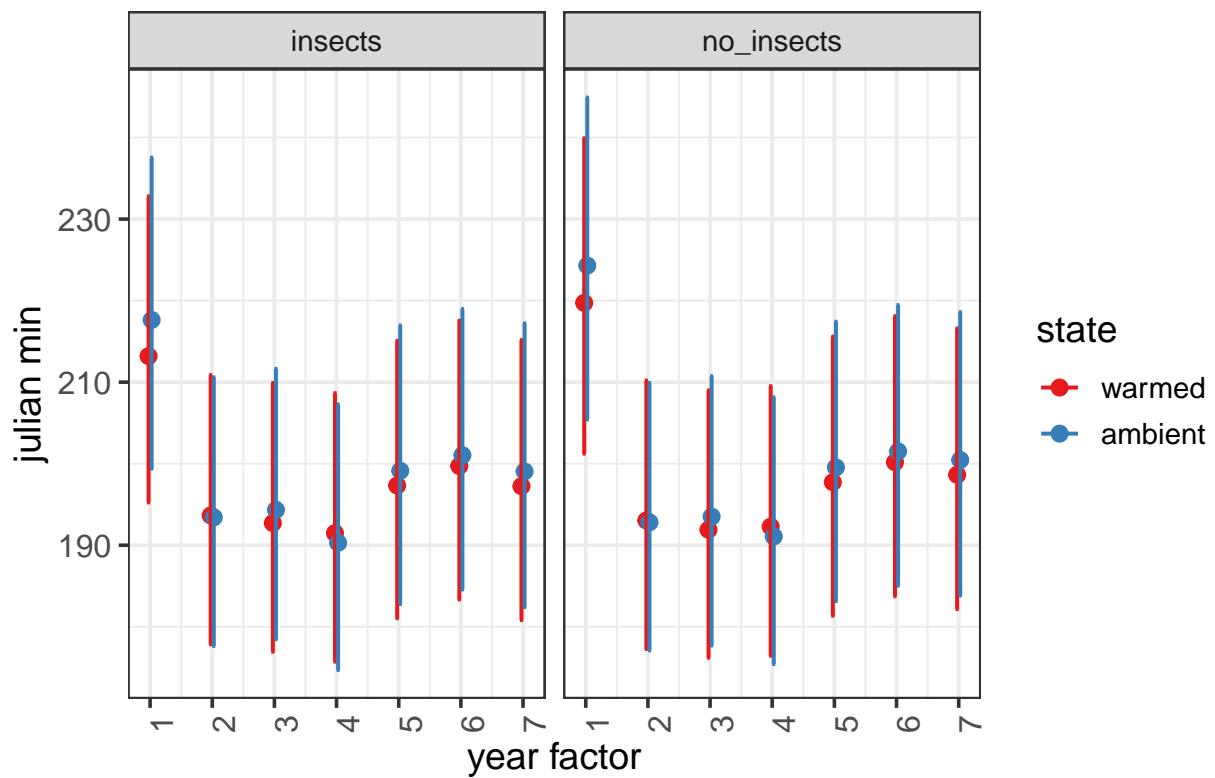
```



```
# these are the fixed predicted values:
plot_model(mod2, type = "pred", terms = c("year_factor", "state", "insecticide"))
```

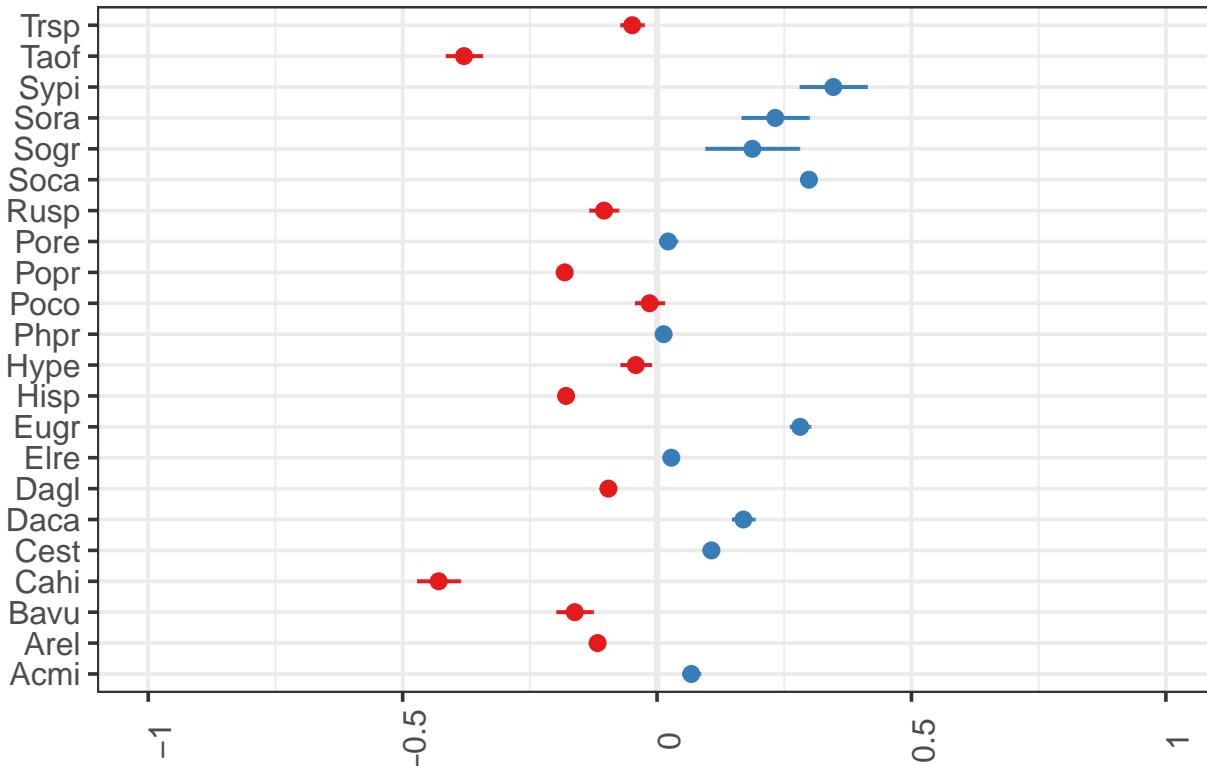
```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standardized
```

Predicted values of julian min



```
# these are the random effects estimates
plot_model(mod2, type = "re", terms = c("species"))
```

Random effects



```
# Do we need to include insecticide?
mod3 <- lmer(log(julian_min) ~ state * year_factor + (1 | species), kbs_sd_spp, REML = FALSE)
anova(mod1, mod3) # Don't need insecticide, continue with mod3
```

```
## Data: kbs_sd_spp
## Models:
## mod3: log(julian_min) ~ state * year_factor + (1 | species)
## mod1: log(julian_min) ~ state * year_factor + insecticide * year_factor +
## mod1: (1 | species) + (1 | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod3   16 -2466.5 -2391.2 1249.3   -2498.5
## mod1   24 -2456.1 -2343.2 1252.1   -2504.1 5.6243  8     0.6892
```

```
# Does year need to be interactive with insecticide? - already removed
# insecticide mod4 <- lmer(log(julian_min) ~ state*year_factor + insecticide +
# (1/species) + (1/plot), kbs_sd_spp, REML=FALSE)
```

```
# Does year need to be interactive with state?
mod5 <- lmer(log(julian_min) ~ state + year_factor + (1 | species) + (1 | plot),
kbs_sd_spp, REML = FALSE)
```

```
## boundary (singular) fit: see ?isSingular
```

```
anova(mod3, mod5)
```

```

## Data: kbs_sd_spp
## Models:
## mod5: log(julian_min) ~ state + year_factor + (1 | species) + (1 |
## mod5:     plot)
## mod3: log(julian_min) ~ state * year_factor + (1 | species)
##      npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## mod5   11 -2472.7 -2420.9 1247.3   -2494.7
## mod3   16 -2466.5 -2391.2 1249.3   -2498.5 3.8516  5       0.571

```

```
AICctab(mod3, mod5, weights = T) # go with mod5
```

```

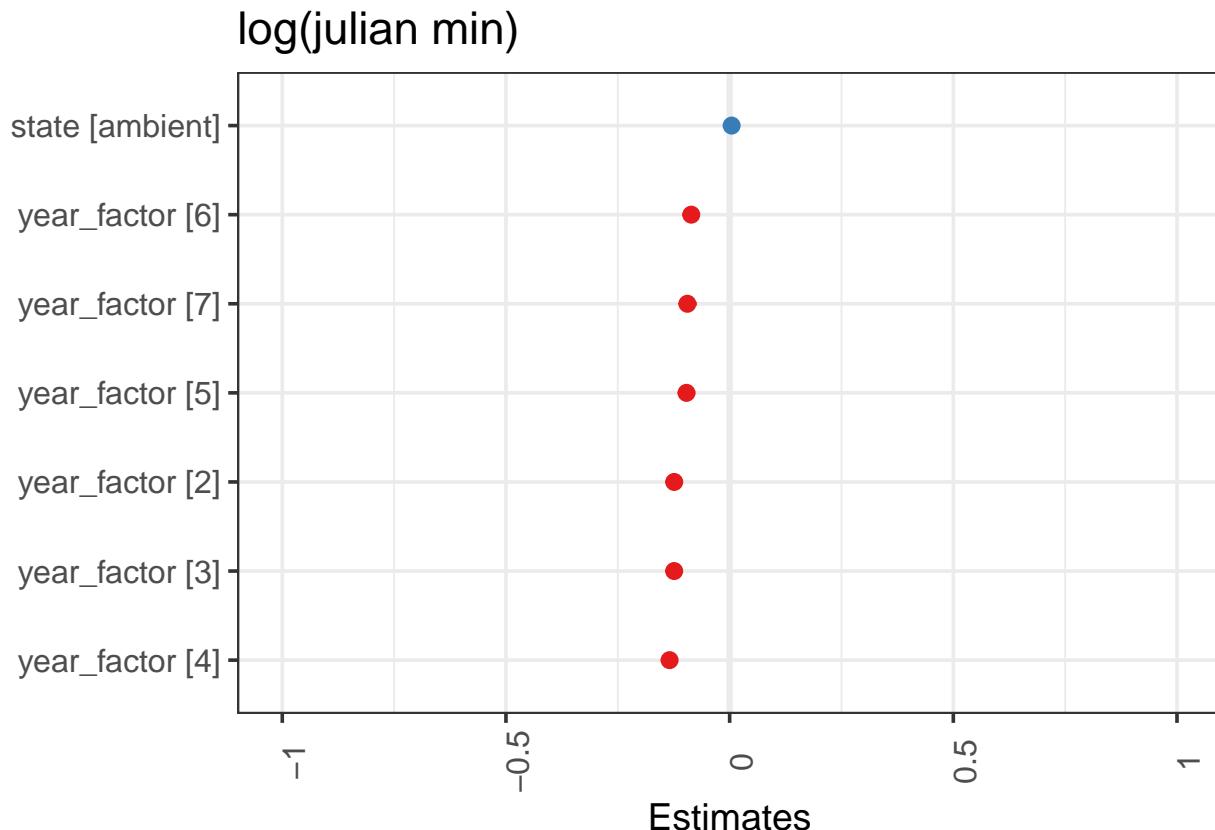
##      dAICc df weight
## mod5  0.0  11  0.963
## mod3  6.5  16  0.037

```

```

# Plot the fixed effects estimates for different models these are the fixed
# effects estimates from summary(mod5)
plot_model(mod5, sort.est = TRUE)

```



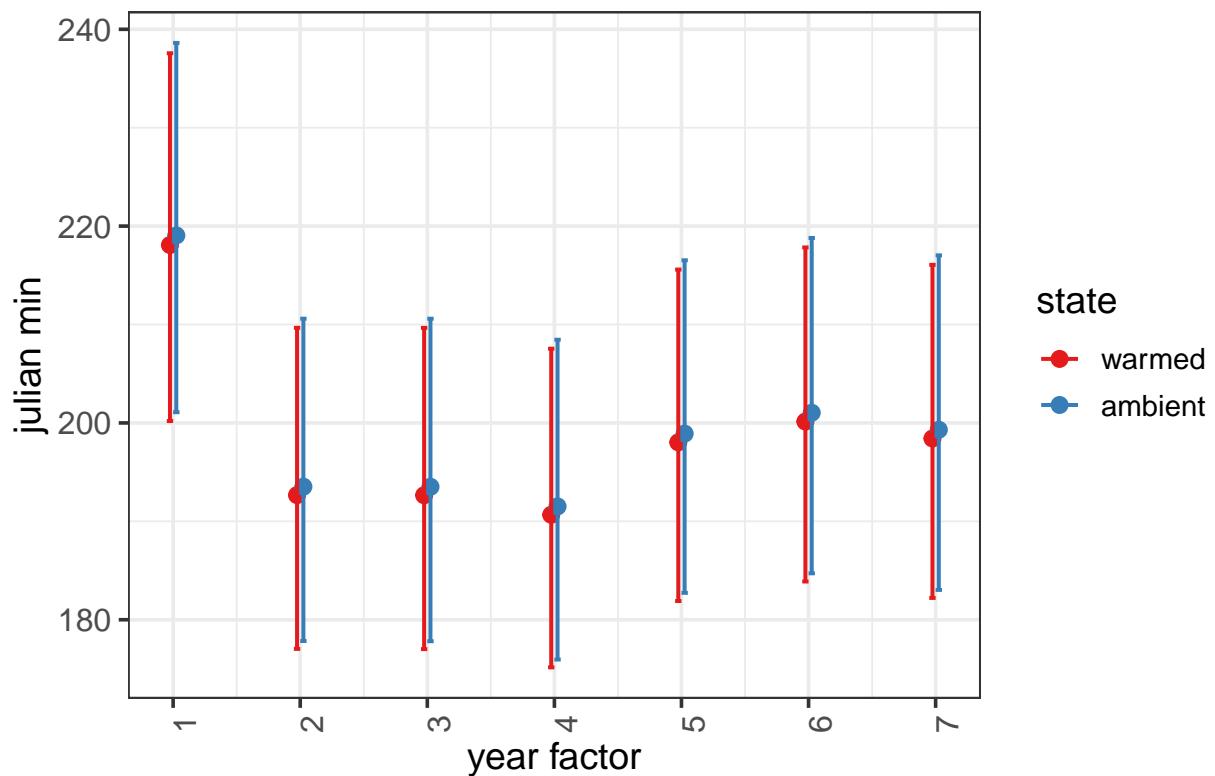
```

# these are the fixed predicted values:
plot_model(mod5, type = "pred", terms = c("year_factor", "state"))

```

```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standard
```

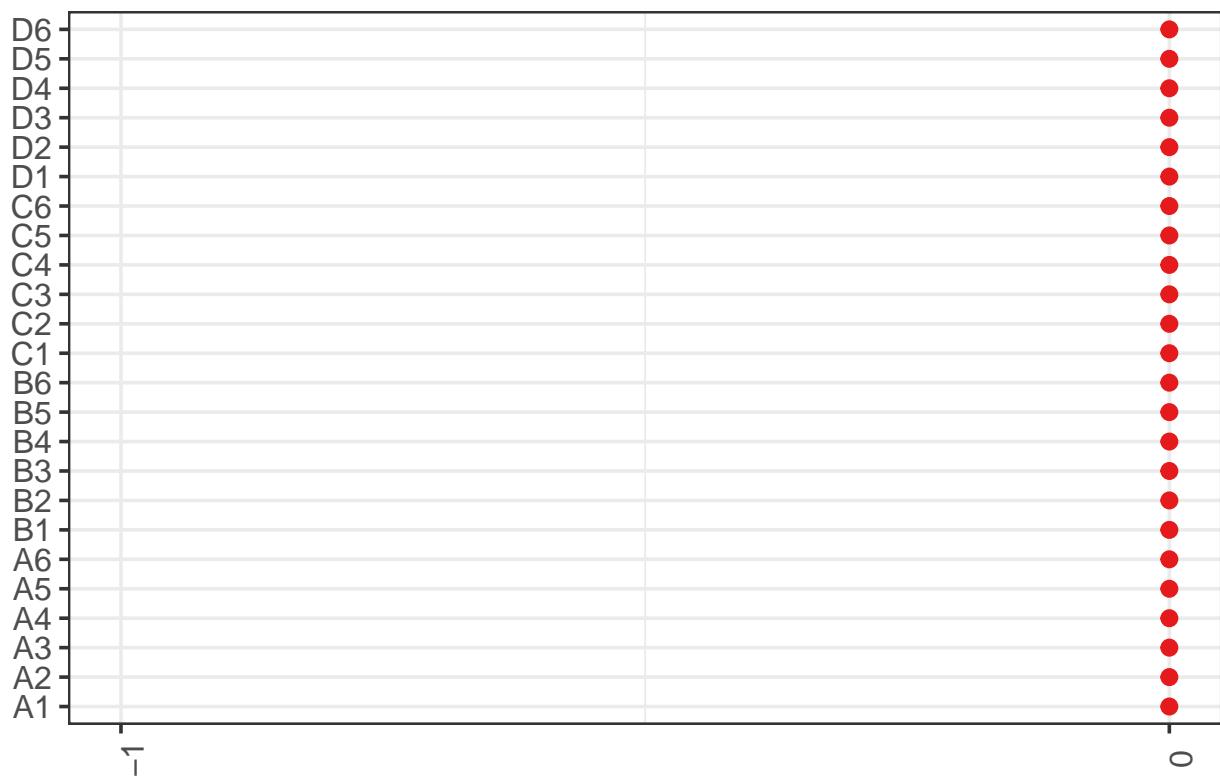
Predicted values of julian min



```
# these are the random effects estimates
plot_model(mod5, type = "re", terms = c("species"))
```

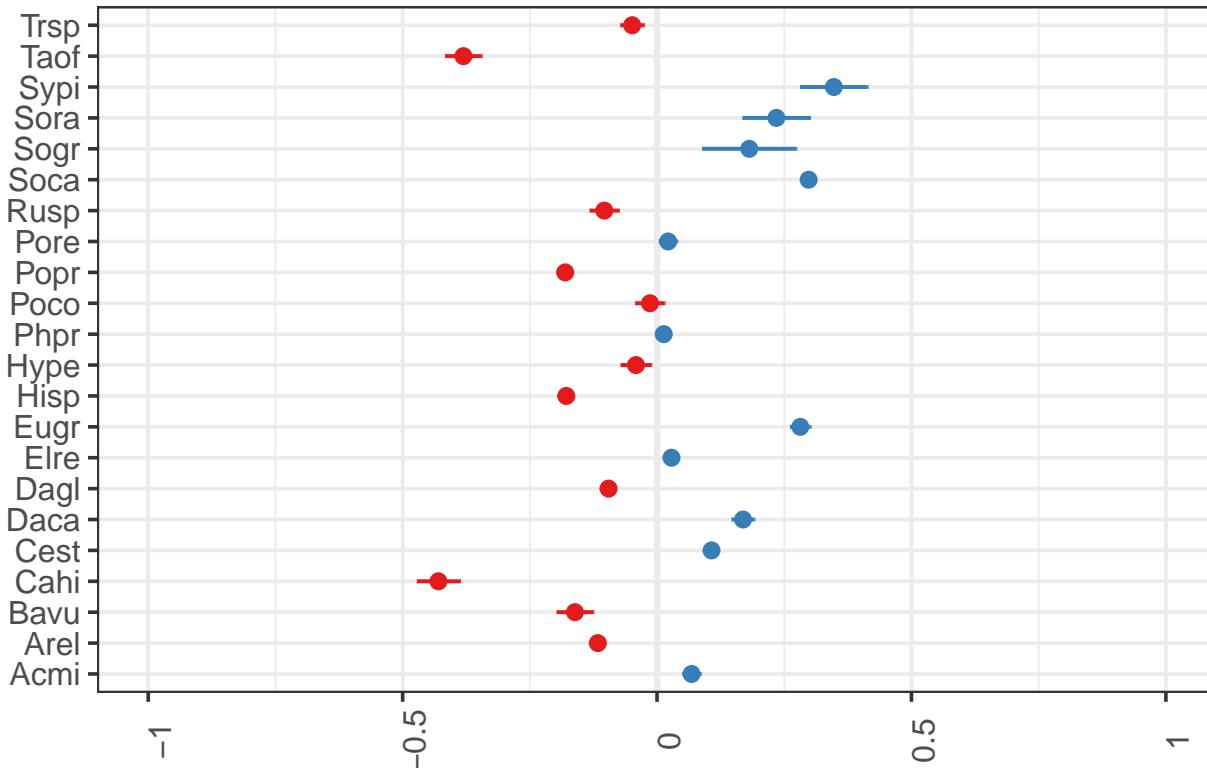
```
## [[1]]
```

Random effects



```
##  
## [[2]]
```

Random effects



```
# If we wanted to include plots nested within year it would look like this:
mod6 <- lmer(log(julian_min) ~ state + year_factor + (1 | species) + (1 + year |
  plot), kbs_sd_spp, REML = FALSE)
```

```
## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 1 negative eigenvalue: -2.5e-01

anova(mod5, mod6) #stick with mod5
```

```
## Data: kbs_sd_spp
## Models:
## mod5: log(julian_min) ~ state + year_factor + (1 | species) + (1 |
## mod5:   plot)
## mod6: log(julian_min) ~ state + year_factor + (1 | species) + (1 +
## mod6:   year | plot)
##      npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## mod5    11 -2472.7 -2420.9 1247.3   -2494.7
## mod6    13 -2379.0 -2317.8 1202.5   -2405.0     0  2       1
```

```
anova(mod5)
```

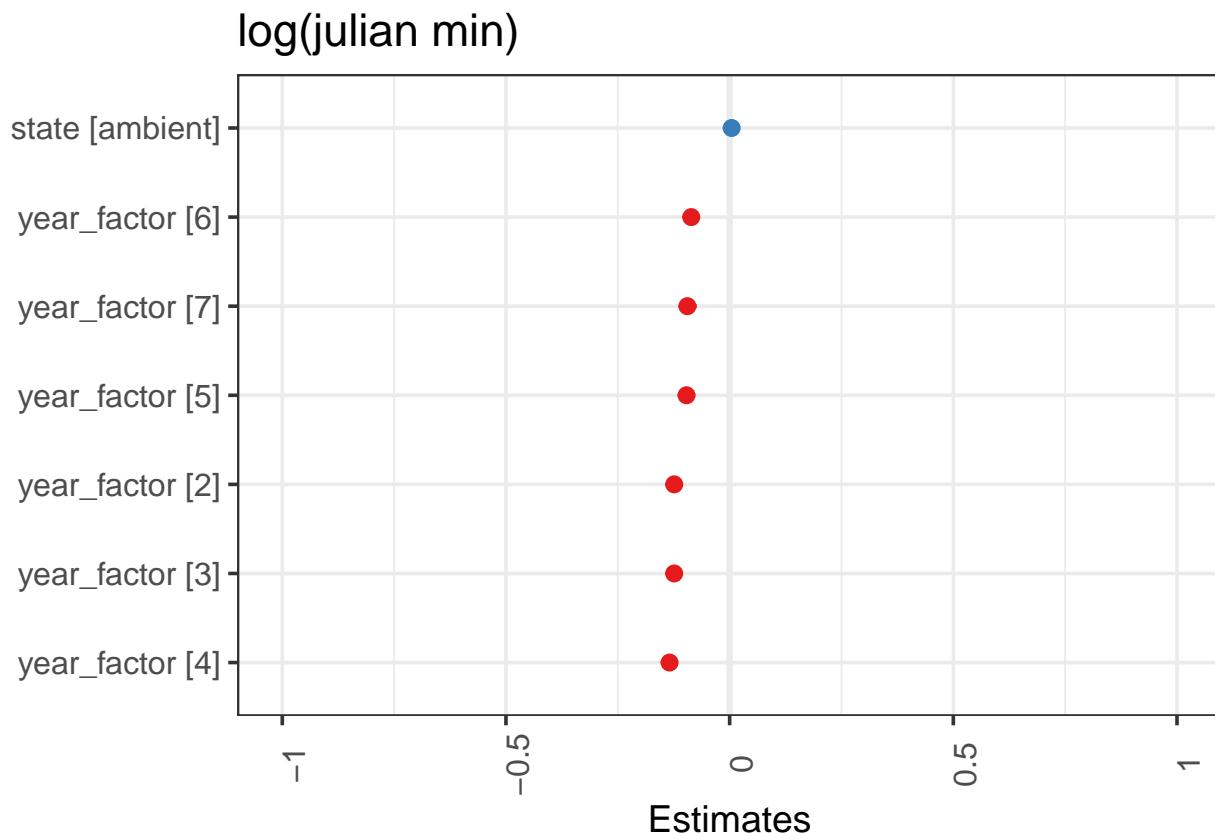
```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq  Mean Sq NumDF DenDF F value Pr(>F)
## state     0.00370 0.003698     1 794.25  1.569 0.2107
```

```

## year_factor 0.73853 0.123088      6 794.44  52.223 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

plot_model(mod5, sort.est = TRUE)

```



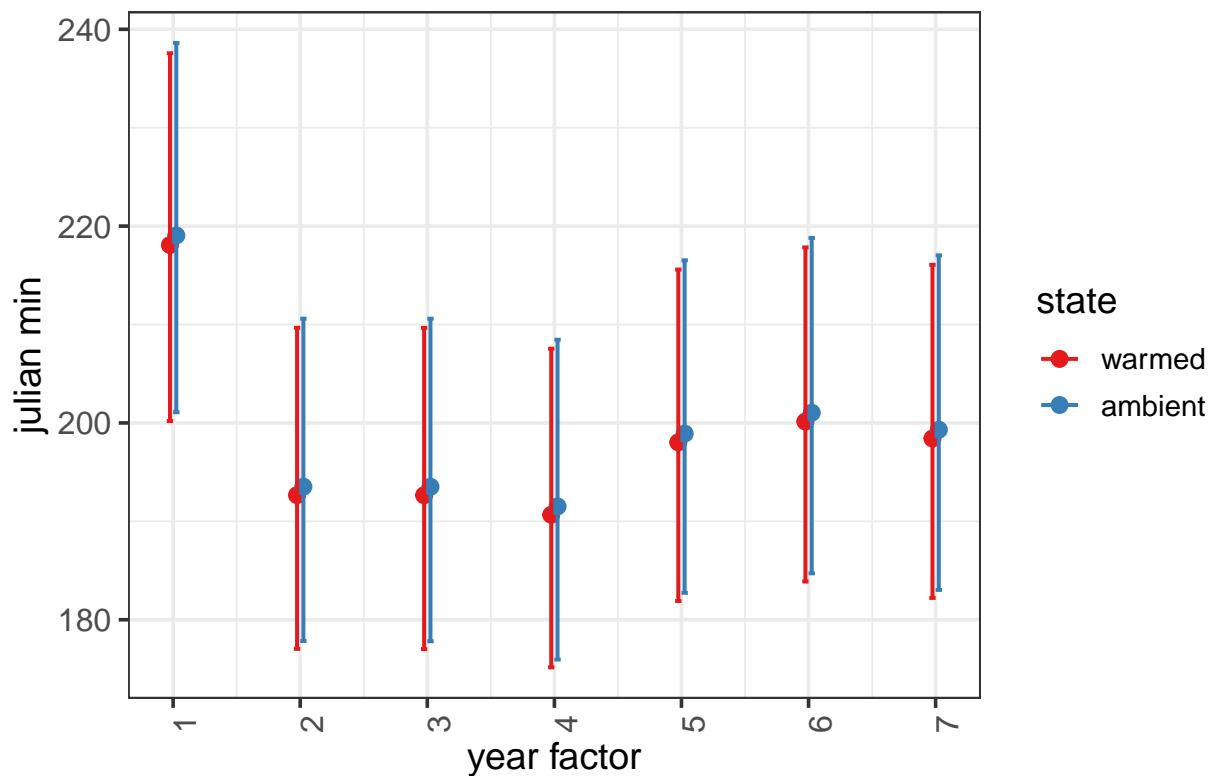
```

# these are the fixed predicted values:
plot_model(mod5, type = "pred", terms = c("year_factor", "state"))

```

Model has log-transformed response. Back-transforming predictions to original response scale. Standardized residuals.

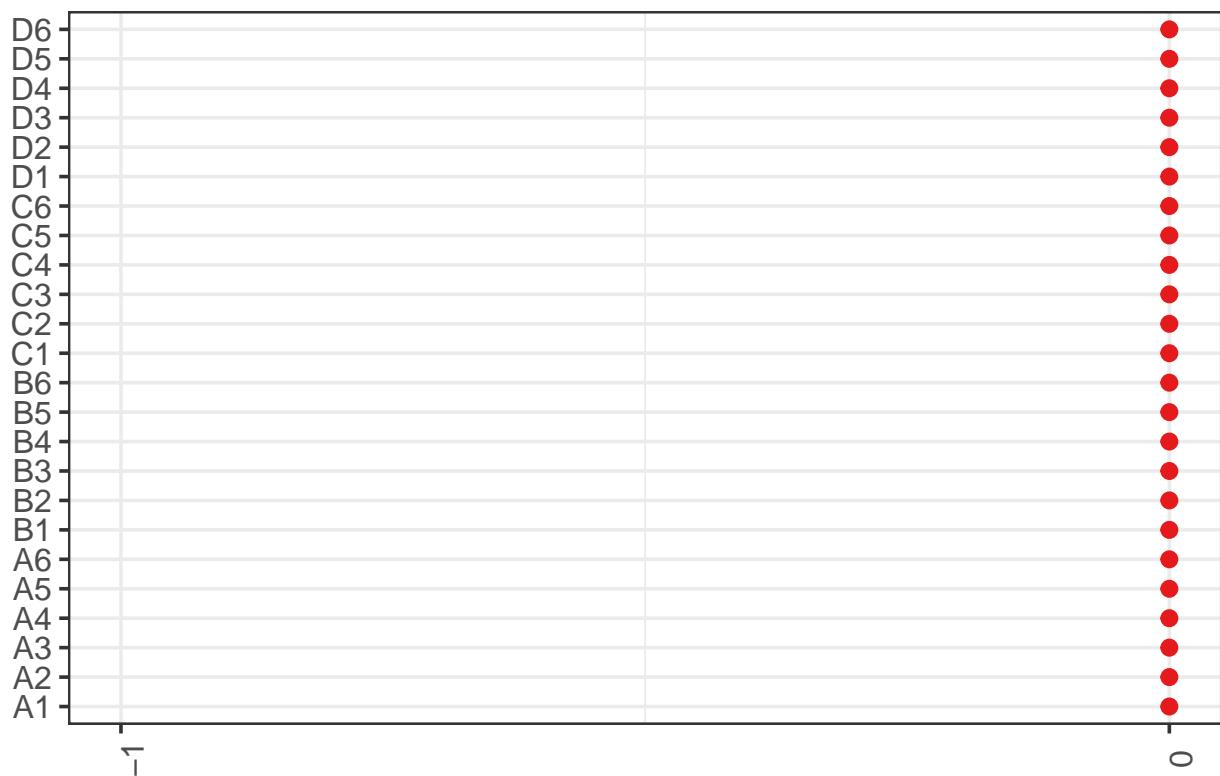
Predicted values of julian min



```
# these are the random effects estimates
plot_model(mod5, type = "re", terms = c("species", "plot"))
```

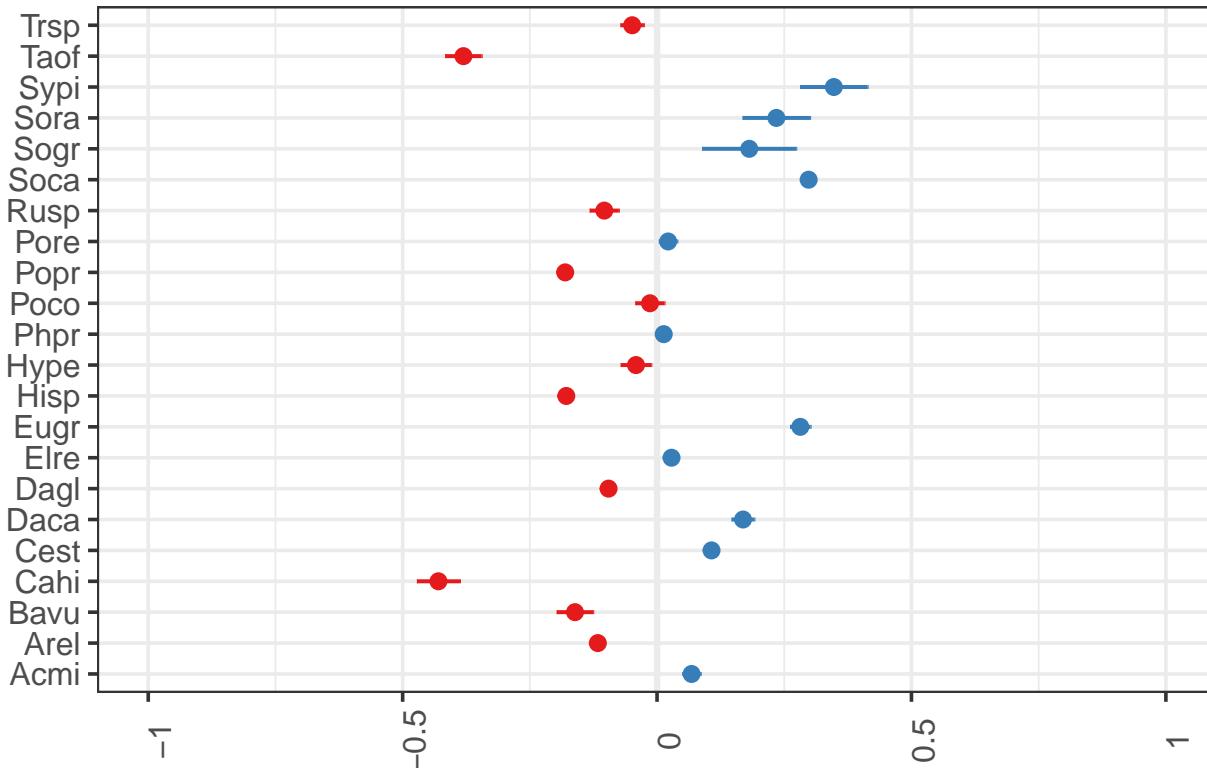
```
## [[1]]
```

Random effects



```
##  
## [[2]]
```

Random effects



```
# mod4 (and mod5) are pretty complex in terms of interpretation (they actually
# don't have many parameters though). We could consider an alternative model
# that's simpler to understand and also one that provides more insight about the
# species. That would be something like this:
mod7 <- lmer(log(julian_min) ~ state + species + (1 + year_factor | plot), kbs_sd_spp,
  REML = FALSE)

## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 2 negative eigenvalues: -3.4e-02 -3.2e+00

mod7a <- lmer(log(julian_min) ~ state + species + year_factor + (1 | plot), kbs_sd_spp,
  REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod7b <- lmer(log(julian_min) ~ state * year_factor + species + (1 | plot), kbs_sd_spp,
  REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod7c <- lmer(log(julian_min) ~ state + species + year_factor + insecticide + (1 |
  plot), kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular
```

```

anova(mod5, mod7) # model 7 is a better fit to data

## Data: kbs_sd_spp
## Models:
## mod5: log(julian_min) ~ state + year_factor + (1 | species) + (1 |
## mod5:      plot)
## mod7: log(julian_min) ~ state + species + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod5    11 -2472.7 -2420.9 1247.3   -2494.7
## mod7    52 -2485.0 -2240.4 1294.5   -2589.0 94.377 41  4.315e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
anova(mod7, mod7a) #mod 7a
```

```

## Data: kbs_sd_spp
## Models:
## mod7a: log(julian_min) ~ state + species + year_factor + (1 | plot)
## mod7: log(julian_min) ~ state + species + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod7a   31 -2581.8 -2436.0 1321.9   -2643.8
## mod7    52 -2485.0 -2240.4 1294.5   -2589.0      0 21           1

```

```
anova(mod7a, mod7b) #mod 7a
```

```

## Data: kbs_sd_spp
## Models:
## mod7a: log(julian_min) ~ state + species + year_factor + (1 | plot)
## mod7b: log(julian_min) ~ state * year_factor + species + (1 | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod7a   31 -2581.8 -2436.0 1321.9   -2643.8
## mod7b   37 -2573.8 -2399.7 1323.9   -2647.8 3.9597  6     0.6821

```

```
anova(mod7a, mod7c) #mod7a
```

```

## Data: kbs_sd_spp
## Models:
## mod7a: log(julian_min) ~ state + species + year_factor + (1 | plot)
## mod7c: log(julian_min) ~ state + species + year_factor + insecticide +
## mod7c:      (1 | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod7a   31 -2581.8 -2436.0 1321.9   -2643.8
## mod7c   32 -2580.2 -2429.6 1322.1   -2644.2 0.3422  1     0.5586

```

```
summary(mod7a)
```

```

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + species + year_factor + (1 | plot)
## Data: kbs_sd_spp

```

```

##
##      AIC      BIC logLik deviance df.resid
## -2581.8 -2436.0   1321.9 -2643.8      785
##
## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -4.5921 -0.5306 -0.0049  0.4909  7.6708
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.0000000 0.00000
## Residual           0.002293 0.04789
## Number of obs: 816, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)  5.452875  0.011990 816.00000 454.793 < 2e-16 ***
## stateambient 0.004370  0.003507 816.00000  1.246 0.213101  
## speciesArel -0.184443  0.010465 816.00000 -17.624 < 2e-16 ***
## speciesBavu -0.230672  0.020305 816.00000 -11.360 < 2e-16 ***
## speciesCahi -0.502774  0.023467 816.00000 -21.425 < 2e-16 ***
## speciesCest  0.039215  0.011618 816.00000  3.375 0.000772 ***
## speciesDaca  0.101541  0.014495 816.00000  7.005 5.16e-12 ***
## speciesDagl -0.163366  0.010831 816.00000 -15.083 < 2e-16 ***
## speciesElre -0.039521  0.011372 816.00000 -3.475 0.000537 ***
## speciesEugr  0.214379  0.013900 816.00000 15.422 < 2e-16 ***
## speciesHisp -0.246588  0.011127 816.00000 -22.161 < 2e-16 ***
## speciesHype -0.109633  0.017723 816.00000 -6.186 9.75e-10 ***
## speciesPhpr -0.054832  0.010547 816.00000 -5.199 2.54e-07 ***
## speciesPoco -0.082002  0.017606 816.00000 -4.658 3.73e-06 ***
## speciesPopr -0.248610  0.010203 816.00000 -24.368 < 2e-16 ***
## speciesPore -0.046105  0.013012 816.00000 -3.543 0.000418 ***
## speciesRusp -0.172216  0.017181 816.00000 -10.024 < 2e-16 ***
## speciesSoca  0.230129  0.010169 816.00000 22.630 < 2e-16 ***
## speciesSogr  0.123923  0.048964 816.00000  2.531 0.011564 *  
## speciesSora  0.173318  0.035168 816.00000  4.928 1.00e-06 ***
## speciesSypi  0.289872  0.035383 816.00000  8.192 9.86e-16 ***
## speciesTaof -0.451748  0.020315 816.00000 -22.237 < 2e-16 ***
## speciesTrsp -0.117081  0.014949 816.00000 -7.832 1.49e-14 ***
## year_factor2 -0.123848  0.008143 816.00000 -15.210 < 2e-16 ***
## year_factor3 -0.123874  0.008310 816.00000 -14.906 < 2e-16 ***
## year_factor4 -0.134431  0.008450 816.00000 -15.909 < 2e-16 ***
## year_factor5 -0.096438  0.008651 816.00000 -11.148 < 2e-16 ***
## year_factor6 -0.086017  0.008330 816.00000 -10.327 < 2e-16 ***
## year_factor7 -0.094345  0.009449 816.00000 -9.985 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 29 > 12.
## Use print(x, correlation=TRUE)  or
##      vcov(x)      if you need it

## optimizer (nloptwrap) convergence code: 0 (OK)

```

```

## boundary (singular) fit: see ?isSingular

anova(mod7a) # investigates whether at least one of the levels within each factor is significantly dif

## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state      0.0036 0.00356     1    816  1.5527 0.2131
## species    23.6339 1.12542    21    816 490.7854 <2e-16 ***
## year_factor 0.7373 0.12289     6    816  53.5895 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Yes, at least one of the species is different (they do not all have the median
# first flwr dates).
emmeans(mod7a, list(pairwise ~ state + year_factor), adjust = "tukey")

```

```

## boundary (singular) fit: see ?isSingular

## $`emmeans of state, year_factor`
##   state year_factor emmean       SE   df lower.CL upper.CL
##   warmed 1           5.386 0.008638 731.1   5.369   5.403
##   ambient 1          5.390 0.008527 717.0   5.373   5.407
##   warmed 2          5.262 0.005434 323.6   5.251   5.273
##   ambient 2          5.266 0.005230 282.4   5.256   5.277
##   warmed 3          5.262 0.005630 346.9   5.251   5.273
##   ambient 3          5.266 0.005453 302.0   5.255   5.277
##   warmed 4          5.251 0.006294 469.0   5.239   5.264
##   ambient 4          5.256 0.006062 404.7   5.244   5.268
##   warmed 5          5.289 0.006753 535.0   5.276   5.303
##   ambient 5          5.294 0.006446 460.0   5.281   5.306
##   warmed 6          5.300 0.005916 451.5   5.288   5.311
##   ambient 6          5.304 0.005715 378.3   5.293   5.315
##   warmed 7          5.291 0.007527 606.6   5.277   5.306
##   ambient 7          5.296 0.007543 571.4   5.281   5.311
##
## Results are averaged over the levels of: species
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $`pairwise differences of state, year_factor`
##   1           estimate       SE   df t.ratio p.value
##   warmed 1 - ambient 1 -4.37e-03 0.00358 25.9 -1.220 0.9914
##   warmed 1 - warmed 2  1.24e-01 0.00830 840.6 14.917 <.0001
##   warmed 1 - ambient 2  1.19e-01 0.00903 509.1 13.235 <.0001
##   warmed 1 - warmed 3  1.24e-01 0.00847 841.6 14.618 <.0001
##   warmed 1 - ambient 3  1.20e-01 0.00920 517.6 12.994 <.0001
##   warmed 1 - warmed 4  1.34e-01 0.00861 839.0 15.606 <.0001
##   warmed 1 - ambient 4  1.30e-01 0.00928 528.5 14.018 <.0001
##   warmed 1 - warmed 5  9.64e-02 0.00882 838.2 10.937 <.0001
##   warmed 1 - ambient 5  9.21e-02 0.00940 540.7  9.789 <.0001
##   warmed 1 - warmed 6  8.60e-02 0.00849 838.6 10.131 <.0001

```

```

## warmed 1 - ambient 6 8.16e-02 0.00919 516.5 8.882 <.0001
## warmed 1 - warmed 7 9.43e-02 0.00964 843.5 9.789 <.0001
## warmed 1 - ambient 7 9.00e-02 0.01039 597.9 8.663 <.0001
## ambient 1 - warmed 2 1.28e-01 0.00906 509.1 14.156 <.0001
## ambient 1 - ambient 2 1.24e-01 0.00830 840.6 14.917 <.0001
## ambient 1 - warmed 3 1.28e-01 0.00920 524.6 13.934 <.0001
## ambient 1 - ambient 3 1.24e-01 0.00847 841.6 14.618 <.0001
## ambient 1 - warmed 4 1.39e-01 0.00938 555.2 14.796 <.0001
## ambient 1 - ambient 4 1.34e-01 0.00861 839.0 15.606 <.0001
## ambient 1 - warmed 5 1.01e-01 0.00963 581.7 10.469 <.0001
## ambient 1 - ambient 5 9.64e-02 0.00882 838.2 10.937 <.0001
## ambient 1 - warmed 6 9.04e-02 0.00924 547.9 9.783 <.0001
## ambient 1 - ambient 6 8.60e-02 0.00849 838.6 10.131 <.0001
## ambient 1 - warmed 7 9.87e-02 0.01018 610.2 9.700 <.0001
## ambient 1 - ambient 7 9.43e-02 0.00964 843.5 9.789 <.0001
## warmed 2 - ambient 2 -4.37e-03 0.00358 25.9 -1.220 0.9914
## warmed 2 - warmed 3 2.51e-05 0.00547 841.2 0.005 1.0000
## warmed 2 - ambient 3 -4.35e-03 0.00656 228.8 -0.663 1.0000
## warmed 2 - warmed 4 1.06e-02 0.00591 839.9 1.792 0.8797
## warmed 2 - ambient 4 6.21e-03 0.00686 258.7 0.906 0.9998
## warmed 2 - warmed 5 -2.74e-02 0.00624 844.9 -4.392 0.0011
## warmed 2 - ambient 5 -3.18e-02 0.00707 269.7 -4.497 0.0008
## warmed 2 - warmed 6 -3.78e-02 0.00568 846.0 -6.664 <.0001
## warmed 2 - ambient 6 -4.22e-02 0.00670 225.3 -6.298 <.0001
## warmed 2 - warmed 7 -2.95e-02 0.00749 845.3 -3.937 0.0068
## warmed 2 - ambient 7 -3.39e-02 0.00845 392.3 -4.008 0.0056
## ambient 2 - warmed 3 4.40e-03 0.00652 232.8 0.674 1.0000
## ambient 2 - ambient 3 2.51e-05 0.00547 841.2 0.005 1.0000
## ambient 2 - warmed 4 1.50e-02 0.00696 288.4 2.149 0.6665
## ambient 2 - ambient 4 1.06e-02 0.00591 839.9 1.792 0.8797
## ambient 2 - warmed 5 -2.30e-02 0.00733 321.8 -3.145 0.0982
## ambient 2 - ambient 5 -2.74e-02 0.00624 844.9 -4.392 0.0011
## ambient 2 - warmed 6 -3.35e-02 0.00673 252.2 -4.976 0.0001
## ambient 2 - ambient 6 -3.78e-02 0.00568 846.0 -6.664 <.0001
## ambient 2 - warmed 7 -2.51e-02 0.00816 394.8 -3.080 0.1159
## ambient 2 - ambient 7 -2.95e-02 0.00749 845.3 -3.937 0.0068
## warmed 3 - ambient 3 -4.37e-03 0.00358 25.9 -1.220 0.9914
## warmed 3 - warmed 4 1.06e-02 0.00615 831.5 1.716 0.9100
## warmed 3 - ambient 4 6.19e-03 0.00706 295.2 0.877 0.9998
## warmed 3 - warmed 5 -2.74e-02 0.00649 843.2 -4.228 0.0021
## warmed 3 - ambient 5 -3.18e-02 0.00727 301.0 -4.375 0.0014
## warmed 3 - warmed 6 -3.79e-02 0.00600 845.5 -6.309 <.0001
## warmed 3 - ambient 6 -4.22e-02 0.00696 258.5 -6.066 <.0001
## warmed 3 - warmed 7 -2.95e-02 0.00767 846.0 -3.851 0.0094
## warmed 3 - ambient 7 -3.39e-02 0.00859 424.8 -3.944 0.0070
## ambient 3 - warmed 4 1.49e-02 0.00718 320.8 2.079 0.7155
## ambient 3 - ambient 4 1.06e-02 0.00615 831.5 1.716 0.9100
## ambient 3 - warmed 5 -2.31e-02 0.00755 348.2 -3.055 0.1246
## ambient 3 - ambient 5 -2.74e-02 0.00649 843.2 -4.228 0.0021
## ambient 3 - warmed 6 -3.35e-02 0.00702 282.6 -4.773 0.0002
## ambient 3 - ambient 6 -3.79e-02 0.00600 845.5 -6.309 <.0001
## ambient 3 - warmed 7 -2.52e-02 0.00833 423.0 -3.019 0.1355
## ambient 3 - ambient 7 -2.95e-02 0.00767 846.0 -3.851 0.0094
## warmed 4 - ambient 4 -4.37e-03 0.00358 25.9 -1.220 0.9914

```

```

##  warmed 4 - warmed 5 -3.80e-02 0.00672 836.3 -5.653 <.0001
##  warmed 4 - ambient 5 -4.24e-02 0.00754 350.2 -5.620 <.0001
##  warmed 4 - warmed 6 -4.84e-02 0.00634 837.9 -7.633 <.0001
##  warmed 4 - ambient 6 -5.28e-02 0.00732 319.3 -7.211 <.0001
##  warmed 4 - warmed 7 -4.01e-02 0.00789 841.4 -5.078 <.0001
##  warmed 4 - ambient 7 -4.45e-02 0.00885 479.0 -5.025 0.0001
##  ambient 4 - warmed 5 -3.36e-02 0.00769 372.8 -4.371 0.0013
##  ambient 4 - ambient 5 -3.80e-02 0.00672 836.3 -5.653 <.0001
##  ambient 4 - warmed 6 -4.40e-02 0.00725 320.3 -6.076 <.0001
##  ambient 4 - ambient 6 -4.84e-02 0.00634 837.9 -7.633 <.0001
##  ambient 4 - warmed 7 -3.57e-02 0.00849 457.5 -4.208 0.0025
##  ambient 4 - ambient 7 -4.01e-02 0.00789 841.4 -5.078 <.0001
##  warmed 5 - ambient 5 -4.37e-03 0.00358 25.9 -1.220 0.9914
##  warmed 5 - warmed 6 -1.04e-02 0.00660 835.6 -1.580 0.9511
##  warmed 5 - ambient 6 -1.48e-02 0.00762 360.1 -1.941 0.8023
##  warmed 5 - warmed 7 -2.09e-03 0.00812 837.5 -0.258 1.0000
##  warmed 5 - ambient 7 -6.46e-03 0.00911 518.6 -0.709 1.0000
##  ambient 5 - warmed 6 -6.05e-03 0.00739 338.9 -0.819 0.9999
##  ambient 5 - ambient 6 -1.04e-02 0.00660 835.6 -1.580 0.9511
##  ambient 5 - warmed 7 2.28e-03 0.00863 481.4 0.264 1.0000
##  ambient 5 - ambient 7 -2.09e-03 0.00812 837.5 -0.258 1.0000
##  warmed 6 - ambient 6 -4.37e-03 0.00358 25.9 -1.220 0.9914
##  warmed 6 - warmed 7 8.33e-03 0.00767 839.7 1.087 0.9984
##  warmed 6 - ambient 7 3.96e-03 0.00861 464.4 0.460 1.0000
##  ambient 6 - warmed 7 1.27e-02 0.00831 437.8 1.529 0.9618
##  ambient 6 - ambient 7 8.33e-03 0.00767 839.7 1.087 0.9984
##  warmed 7 - ambient 7 -4.37e-03 0.00358 25.9 -1.220 0.9914
##
## Results are averaged over the levels of: species
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 14 estimates

```

```
emmeans(mod7a, list(pairwise ~ year_factor), adjust = "tukey")
```

```
## boundary (singular) fit: see ?isSingular
```

```

## $`emmeans of year_factor`
##  year_factor emmean      SE   df lower.CL upper.CL
##  1           5.388 0.008394 821.4    5.371   5.404
##  2           5.264 0.005023 519.7    5.254   5.274
##  3           5.264 0.005245 537.0    5.254   5.274
##  4           5.253 0.005914 655.2    5.242   5.265
##  5           5.291 0.006353 701.1    5.279   5.304
##  6           5.302 0.005534 656.2    5.291   5.313
##  7           5.294 0.007319 747.7    5.279   5.308
##
## Results are averaged over the levels of: state, species
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $`pairwise differences of year_factor`
```

```

##   1      estimate      SE  df t.ratio p.value
## 1 - 2  1.24e-01 0.00830 841 14.917 <.0001
## 1 - 3  1.24e-01 0.00847 842 14.618 <.0001
## 1 - 4  1.34e-01 0.00861 839 15.606 <.0001
## 1 - 5  9.64e-02 0.00882 838 10.937 <.0001
## 1 - 6  8.60e-02 0.00849 839 10.131 <.0001
## 1 - 7  9.43e-02 0.00964 843  9.789 <.0001
## 2 - 3  2.51e-05 0.00547 841  0.005 1.0000
## 2 - 4  1.06e-02 0.00591 840  1.792 0.5542
## 2 - 5 -2.74e-02 0.00624 845 -4.392 0.0003
## 2 - 6 -3.78e-02 0.00568 846 -6.664 <.0001
## 2 - 7 -2.95e-02 0.00749 845 -3.937 0.0017
## 3 - 4  1.06e-02 0.00615 832  1.716 0.6053
## 3 - 5 -2.74e-02 0.00649 843 -4.228 0.0005
## 3 - 6 -3.79e-02 0.00600 846 -6.309 <.0001
## 3 - 7 -2.95e-02 0.00767 846 -3.851 0.0024
## 4 - 5 -3.80e-02 0.00672 836 -5.653 <.0001
## 4 - 6 -4.84e-02 0.00634 838 -7.633 <.0001
## 4 - 7 -4.01e-02 0.00789 841 -5.078 <.0001
## 5 - 6 -1.04e-02 0.00660 836 -1.580 0.6954
## 5 - 7 -2.09e-03 0.00812 838 -0.258 1.0000
## 6 - 7  8.33e-03 0.00767 840  1.087 0.9321
##
## Results are averaged over the levels of: state, species
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 7 estimates

```

```
emmeans(mod7a, list(pairwise ~ species), adjust = "tukey")
```

```
## boundary (singular) fit: see ?isSingular
```

```

## $`emmeans of species`
##   species emmean      SE  df lower.CL upper.CL
##   Acmi    5.361 0.009503 766.2    5.342   5.380
##   Arel    5.176 0.004916 533.2    5.167   5.186
##   Bavu    5.130 0.018595 821.7    5.094   5.167
##   Cahí    4.858 0.022232 845.8    4.815   4.902
##   Cest    5.400 0.007665 754.1    5.385   5.415
##   Daca    5.462 0.011681 820.5    5.440   5.485
##   Dagl    5.198 0.005888 667.2    5.186   5.209
##   Elre    5.321 0.006853 758.1    5.308   5.335
##   Eugr    5.575 0.010748 820.3    5.554   5.596
##   Hisp    5.114 0.006754 722.6    5.101   5.128
##   Hype    5.251 0.015643 829.6    5.221   5.282
##   Phpr    5.306 0.005261 567.8    5.296   5.316
##   Poco    5.279 0.015173 839.7    5.249   5.309
##   Popr    5.112 0.004615 543.9    5.103   5.121
##   Pore    5.315 0.009684 827.1    5.296   5.334
##   Rusp    5.189 0.014945 781.0    5.159   5.218
##   Soca    5.591 0.004507 525.4    5.582   5.600
##   Sogr    5.485 0.049100 846.1    5.388   5.581
##   Sora    5.534 0.034789 846.0    5.466   5.603

```

```

##   Sypi      5.651 0.034843 846.0      5.582      5.719
##   Taof      4.909 0.018683 844.5      4.873      4.946
##   Trsp      5.244 0.012360 845.0      5.220      5.268
##
## Results are averaged over the levels of: state, year_factor
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $'pairwise differences of species'
##   1           estimate      SE  df t.ratio p.value
##   Acmi - Arel  0.18444 0.01071 840  17.228 <.0001
##   Acmi - Bavu  0.23067 0.02071 844  11.137 <.0001
##   Acmi - Cahи  0.50277 0.02395 846  20.993 <.0001
##   Acmi - Cest -0.03921 0.01188 843  -3.300 0.1254
##   Acmi - Daca -0.10154 0.01480 846  -6.863 <.0001
##   Acmi - Dagl  0.16337 0.01106 846  14.768 <.0001
##   Acmi - Elre  0.03952 0.01163 839   3.398 0.0948
##   Acmi - Eugr -0.21438 0.01419 846  -15.105 <.0001
##   Acmi - Hisp  0.24659 0.01137 845  21.687 <.0001
##   Acmi - Hype  0.10963 0.01812 838   6.051 <.0001
##   Acmi - Phpr  0.05483 0.01079 839   5.083 0.0001
##   Acmi - Poco  0.08200 0.01796 845   4.565 0.0012
##   Acmi - Popr  0.24861 0.01044 838  23.809 <.0001
##   Acmi - Pore  0.04611 0.01328 846   3.471 0.0762
##   Acmi - Rusp  0.17222 0.01756 843   9.809 <.0001
##   Acmi - Soca -0.23013 0.01040 843  -22.127 <.0001
##   Acmi - Sogr -0.12392 0.05001 846  -2.478 0.6494
##   Acmi - Sora -0.17332 0.03591 845  -4.826 0.0004
##   Acmi - Sypi -0.28987 0.03612 846  -8.025 <.0001
##   Acmi - Taof  0.45175 0.02074 846  21.779 <.0001
##   Acmi - Trsp  0.11708 0.01524 841   7.681 <.0001
##   Arel - Bavu  0.04623 0.01925 835   2.402 0.7067
##   Arel - Cahи  0.31833 0.02282 844  13.950 <.0001
##   Arel - Cest -0.22366 0.00911 842  -24.561 <.0001
##   Arel - Daca -0.28598 0.01266 846  -22.581 <.0001
##   Arel - Dagl -0.02108 0.00767 846  -2.748 0.4375
##   Arel - Elre -0.14492 0.00847 846  -17.106 <.0001
##   Arel - Eugr -0.39882 0.01181 845  -33.777 <.0001
##   Arel - Hisp  0.06214 0.00833 846   7.457 <.0001
##   Arel - Hype -0.07481 0.01642 844  -4.557 0.0012
##   Arel - Phpr -0.12961 0.00720 840  -17.991 <.0001
##   Arel - Poco -0.10244 0.01588 846  -6.449 <.0001
##   Arel - Popr  0.06417 0.00672 845   9.551 <.0001
##   Arel - Pore -0.13834 0.01080 843  -12.804 <.0001
##   Arel - Rusp -0.01223 0.01569 812  -0.779 1.0000
##   Arel - Soca -0.41457 0.00669 846  -62.011 <.0001
##   Arel - Sogr -0.30837 0.04934 846  -6.250 <.0001
##   Arel - Sora -0.35776 0.03516 846  -10.175 <.0001
##   Arel - Sypi -0.47431 0.03516 846  -13.492 <.0001
##   Arel - Taof  0.26730 0.01930 846  13.847 <.0001
##   Arel - Trsp -0.06736 0.01329 844  -5.068 0.0001
##   Bavu - Cahи  0.27210 0.02883 846   9.439 <.0001
##   Bavu - Cest -0.26989 0.02005 836  -13.458 <.0001

```

```

##  Bavu - Daca -0.33221 0.02185 845 -15.206 <.0001
##  Bavu - Dagl -0.06731 0.01948 839 -3.456 0.0798
##  Bavu - Elre -0.19115 0.01975 838 -9.679 <.0001
##  Bavu - Eugr -0.44505 0.02136 845 -20.836 <.0001
##  Bavu - Hisp 0.01592 0.01969 843 0.808 1.0000
##  Bavu - Hype -0.12104 0.02416 842 -5.010 0.0001
##  Bavu - Phpr -0.17584 0.01931 836 -9.109 <.0001
##  Bavu - Poco -0.14867 0.02396 842 -6.206 <.0001
##  Bavu - Popr 0.01794 0.01914 835 0.937 1.0000
##  Bavu - Pore -0.18457 0.02092 841 -8.824 <.0001
##  Bavu - Rusp -0.05846 0.02374 844 -2.463 0.6613
##  Bavu - Soca -0.46080 0.01908 839 -24.149 <.0001
##  Bavu - Sogr -0.35460 0.05258 846 -6.744 <.0001
##  Bavu - Sora -0.40399 0.03937 845 -10.261 <.0001
##  Bavu - Sypi -0.52054 0.03952 845 -13.172 <.0001
##  Bavu - Taof 0.22108 0.02625 846 8.420 <.0001
##  Bavu - Trsp -0.11359 0.02220 845 -5.118 0.0001
##  Cahи - Cest -0.54199 0.02341 845 -23.156 <.0001
##  Cahи - Daca -0.60431 0.02495 845 -24.224 <.0001
##  Cahи - Dagl -0.33941 0.02295 844 -14.789 <.0001
##  Cahи - Elre -0.46325 0.02318 846 -19.985 <.0001
##  Cahи - Eugr -0.71715 0.02471 845 -29.019 <.0001
##  Cahи - Hisp -0.25619 0.02309 843 -11.097 <.0001
##  Cahи - Hype -0.39314 0.02698 844 -14.574 <.0001
##  Cahи - Phpr -0.44794 0.02283 843 -19.621 <.0001
##  Cahи - Poco -0.42077 0.02700 845 -15.585 <.0001
##  Cahи - Popr -0.25416 0.02267 845 -11.212 <.0001
##  Cahи - Pore -0.45667 0.02419 844 -18.878 <.0001
##  Cahи - Rusp -0.33056 0.02676 846 -12.351 <.0001
##  Cahи - Soca -0.73290 0.02261 845 -32.419 <.0001
##  Cahи - Sogr -0.62670 0.05393 843 -11.621 <.0001
##  Cahи - Sora -0.67609 0.04085 839 -16.550 <.0001
##  Cahи - Sypi -0.79265 0.04141 846 -19.142 <.0001
##  Cahи - Taof -0.05103 0.02900 844 -1.760 0.9804
##  Cahи - Trsp -0.38569 0.02546 843 -15.150 <.0001
##  Cest - Daca -0.06233 0.01375 842 -4.533 0.0014
##  Cest - Dagl 0.20258 0.00956 844 21.197 <.0001
##  Cest - Elre 0.07874 0.01016 844 7.749 <.0001
##  Cest - Eugr -0.17516 0.01305 843 -13.420 <.0001
##  Cest - Hisp 0.28580 0.00991 845 28.830 <.0001
##  Cest - Hype 0.14885 0.01727 840 8.617 <.0001
##  Cest - Phpr 0.09405 0.00918 846 10.250 <.0001
##  Cest - Poco 0.12122 0.01716 846 7.065 <.0001
##  Cest - Popr 0.28782 0.00877 844 32.832 <.0001
##  Cest - Pore 0.08532 0.01205 846 7.080 <.0001
##  Cest - Rusp 0.21143 0.01676 787 12.614 <.0001
##  Cest - Soca -0.19091 0.00876 844 -21.794 <.0001
##  Cest - Sogr -0.08471 0.04962 846 -1.707 0.9862
##  Cest - Sora -0.13410 0.03555 846 -3.773 0.0284
##  Cest - Sypi -0.25066 0.03567 846 -7.027 <.0001
##  Cest - Taof 0.49096 0.01999 845 24.556 <.0001
##  Cest - Trsp 0.15630 0.01415 840 11.048 <.0001
##  Daca - Dagl 0.26491 0.01299 844 20.400 <.0001
##  Daca - Elre 0.14106 0.01345 846 10.487 <.0001

```

```

##  Daca - Eugr -0.11284 0.01574 846 -7.170 <.0001
##  Daca - Hisp  0.34813 0.01326 846 26.248 <.0001
##  Daca - Hype  0.21117 0.01938 844 10.894 <.0001
##  Daca - Phpr  0.15637 0.01273 846 12.284 <.0001
##  Daca - Poco  0.18354 0.01924 844  9.541 <.0001
##  Daca - Popr  0.35015 0.01246 845 28.099 <.0001
##  Daca - Pore  0.14765 0.01493 846  9.886 <.0001
##  Daca - Rusp  0.27376 0.01885 845 14.520 <.0001
##  Daca - Soca -0.12859 0.01243 846 -10.347 <.0001
##  Daca - Sogr -0.02238 0.05045 846 -0.444 1.0000
##  Daca - Sora -0.07178 0.03657 846 -1.963 0.9389
##  Daca - Sypi -0.18833 0.03672 845 -5.128 0.0001
##  Daca - Taof  0.55329 0.02189 846 25.274 <.0001
##  Daca - Trsp  0.21862 0.01677 843 13.034 <.0001
##  Dagl - Elre -0.12384 0.00900 846 -13.758 <.0001
##  Dagl - Eugr -0.37774 0.01214 846 -31.108 <.0001
##  Dagl - Hisp  0.08322 0.00884 846  9.416 <.0001
##  Dagl - Hype -0.05373 0.01664 842 -3.229 0.1520
##  Dagl - Phpr -0.10853 0.00784 839 -13.846 <.0001
##  Dagl - Poco -0.08136 0.01632 844 -4.985 0.0002
##  Dagl - Popr  0.08524 0.00744 846 11.463 <.0001
##  Dagl - Pore -0.11726 0.01121 841 -10.465 <.0001
##  Dagl - Rusp  0.00885 0.01602 830  0.553 1.0000
##  Dagl - Soca -0.39349 0.00736 846 -53.485 <.0001
##  Dagl - Sogr -0.28729 0.04944 846 -5.811 <.0001
##  Dagl - Sora -0.33668 0.03524 846 -9.553 <.0001
##  Dagl - Sypi -0.45324 0.03530 846 -12.840 <.0001
##  Dagl - Taof  0.28838 0.01954 846 14.758 <.0001
##  Dagl - Trsp -0.04628 0.01358 838 -3.407 0.0923
##  Elre - Eugr -0.25390 0.01265 846 -20.077 <.0001
##  Elre - Hisp  0.20707 0.00952 846 21.749 <.0001
##  Elre - Hype  0.07011 0.01699 836  4.128 0.0075
##  Elre - Phpr  0.01531 0.00855 846  1.790 0.9763
##  Elre - Poco  0.04248 0.01678 846  2.531 0.6078
##  Elre - Popr  0.20909 0.00820 844 25.505 <.0001
##  Elre - Pore  0.00658 0.01184 843  0.556 1.0000
##  Elre - Rusp  0.13269 0.01648 831  8.051 <.0001
##  Elre - Soca -0.26965 0.00810 843 -33.294 <.0001
##  Elre - Sogr -0.16344 0.04955 846 -3.299 0.1258
##  Elre - Sora -0.21284 0.03543 846 -6.008 <.0001
##  Elre - Sypi -0.32939 0.03552 846 -9.274 <.0001
##  Elre - Taof  0.41223 0.01990 846 20.720 <.0001
##  Elre - Trsp  0.07756 0.01406 841  5.516 <.0001
##  Eugr - Hisp  0.46097 0.01248 846 36.940 <.0001
##  Eugr - Hype  0.32401 0.01872 844 17.307 <.0001
##  Eugr - Phpr  0.26921 0.01188 844 22.667 <.0001
##  Eugr - Poco  0.29638 0.01864 845 15.902 <.0001
##  Eugr - Popr  0.46299 0.01164 846 39.764 <.0001
##  Eugr - Pore  0.26048 0.01424 846 18.298 <.0001
##  Eugr - Rusp  0.38659 0.01824 844 21.189 <.0001
##  Eugr - Soca -0.01575 0.01153 846 -1.366 0.9993
##  Eugr - Sogr  0.09046 0.05030 846  1.798 0.9751
##  Eugr - Sora  0.04106 0.03643 846  1.127 1.0000
##  Eugr - Sypi -0.07549 0.03626 846 -2.082 0.8961

```

```

##  Eogr - Taof  0.66613 0.02138 845  31.152 <.0001
##  Eogr - Trsp  0.33146 0.01604 845  20.669 <.0001
##  Hisp - Hype -0.13695 0.01687 842  -8.118 <.0001
##  Hisp - Phpr -0.19176 0.00846 843 -22.660 <.0001
##  Hisp - Poco -0.16459 0.01669 846  -9.864 <.0001
##  Hisp - Popr  0.00202 0.00803 845   0.252 1.0000
##  Hisp - Pore -0.20048 0.01149 844 -17.444 <.0001
##  Hisp - Rusp -0.07437 0.01629 818  -4.565 0.0012
##  Hisp - Soca -0.47672 0.00797 844 -59.780 <.0001
##  Hisp - Sogr -0.37051 0.04954 846  -7.479 <.0001
##  Hisp - Sora -0.41991 0.03533 846 -11.886 <.0001
##  Hisp - Sypi -0.53646 0.03541 846 -15.151 <.0001
##  Hisp - Taof  0.20516 0.01971 845  10.410 <.0001
##  Hisp - Trsp -0.12951 0.01382 840  -9.374 <.0001
##  Hype - Phpr -0.05480 0.01644 845 -3.333 0.1144
##  Hype - Poco -0.02763 0.02187 843  -1.263 0.9998
##  Hype - Popr  0.13898 0.01627 842  8.543 <.0001
##  Hype - Pore -0.06353 0.01827 838  -3.477 0.0749
##  Hype - Rusp  0.06258 0.02155 837  2.905 0.3255
##  Hype - Soca -0.33976 0.01618 843 -21.003 <.0001
##  Hype - Sogr -0.23356 0.05159 846  -4.527 0.0014
##  Hype - Sora -0.28295 0.03801 845  -7.444 <.0001
##  Hype - Sypi -0.39950 0.03811 846 -10.482 <.0001
##  Hype - Taof  0.34211 0.02424 843  14.114 <.0001
##  Hype - Trsp  0.00745 0.01972 843  0.378 1.0000
##  Phpr - Poco  0.02717 0.01613 845  1.684 0.9882
##  Phpr - Popr  0.19378 0.00694 845  27.927 <.0001
##  Phpr - Pore -0.00873 0.01093 844  -0.799 1.0000
##  Phpr - Rusp  0.11738 0.01585 818  7.407 <.0001
##  Phpr - Soca -0.28496 0.00686 846 -41.513 <.0001
##  Phpr - Sogr -0.17876 0.04935 846  -3.622 0.0472
##  Phpr - Sora -0.22815 0.03517 846  -6.487 <.0001
##  Phpr - Sypi -0.34470 0.03522 846  -9.787 <.0001
##  Phpr - Taof  0.39692 0.01938 846  20.477 <.0001
##  Phpr - Trsp  0.06225 0.01333 839  4.669 0.0007
##  Poco - Popr  0.16661 0.01588 846  10.494 <.0001
##  Poco - Pore -0.03590 0.01806 843  -1.988 0.9313
##  Poco - Rusp  0.09021 0.02104 845  4.288 0.0039
##  Poco - Soca -0.31213 0.01590 846 -19.628 <.0001
##  Poco - Sogr -0.20593 0.05147 846  -4.001 0.0123
##  Poco - Sora -0.25532 0.03801 846  -6.716 <.0001
##  Poco - Sypi -0.37187 0.03796 845  -9.797 <.0001
##  Poco - Taof  0.36975 0.02405 846  15.376 <.0001
##  Poco - Trsp  0.03508 0.01970 840  1.781 0.9776
##  Popr - Pore -0.20250 0.01059 845 -19.127 <.0001
##  Popr - Rusp -0.07639 0.01561 803  -4.895 0.0003
##  Popr - Soca -0.47874 0.00638 830 -74.980 <.0001
##  Popr - Sogr -0.37253 0.04927 846  -7.561 <.0001
##  Popr - Sora -0.42193 0.03506 846 -12.035 <.0001
##  Popr - Sypi -0.53848 0.03512 846 -15.333 <.0001
##  Popr - Taof  0.20314 0.01918 846  10.590 <.0001
##  Popr - Trsp -0.13153 0.01309 841 -10.046 <.0001
##  Pore - Rusp  0.12611 0.01766 829   7.143 <.0001
##  Pore - Soca -0.27623 0.01058 845 -26.104 <.0001

```

```

## Pore - Sogr -0.17003 0.05000 846 -3.400 0.0941
## Pore - Sora -0.21942 0.03603 846 -6.090 <.0001
## Pore - Sypi -0.33598 0.03602 846 -9.328 <.0001
## Pore - Taof 0.40564 0.02084 846 19.468 <.0001
## Pore - Trsp 0.07098 0.01535 838 4.624 0.0009
## Rusp - Soca -0.40234 0.01559 817 -25.802 <.0001
## Rusp - Sogr -0.29614 0.05135 846 -5.768 <.0001
## Rusp - Sora -0.34553 0.03783 846 -9.134 <.0001
## Rusp - Sypi -0.46209 0.03781 842 -12.221 <.0001
## Rusp - Taof 0.27953 0.02373 846 11.778 <.0001
## Rusp - Trsp -0.05513 0.01923 845 -2.867 0.3513
## Soca - Sogr 0.10621 0.04931 846 2.154 0.8627
## Soca - Sora 0.05681 0.03504 846 1.621 0.9926
## Soca - Sypi -0.05974 0.03510 846 -1.702 0.9866
## Soca - Taof 0.68188 0.01917 846 35.562 <.0001
## Soca - Trsp 0.34721 0.01305 839 26.605 <.0001
## Sogr - Sora -0.04939 0.06011 837 -0.822 1.0000
## Sogr - Sypi -0.16595 0.06019 846 -2.757 0.4309
## Sogr - Taof 0.57567 0.05249 842 10.967 <.0001
## Sogr - Trsp 0.24100 0.05064 846 4.759 0.0005
## Sora - Sypi -0.11655 0.04926 846 -2.366 0.7327
## Sora - Taof 0.62507 0.03941 843 15.859 <.0001
## Sora - Trsp 0.29040 0.03693 846 7.864 <.0001
## Sypi - Taof 0.74162 0.03952 846 18.766 <.0001
## Sypi - Trsp 0.40695 0.03697 846 11.007 <.0001
## Taof - Trsp -0.33467 0.02198 844 -15.224 <.0001
##
## Results are averaged over the levels of: state, year_factor
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 22 estimates

```

```
emmeans(mod7a, list(pairwise ~ insecticide), adjust = "tukey")
```

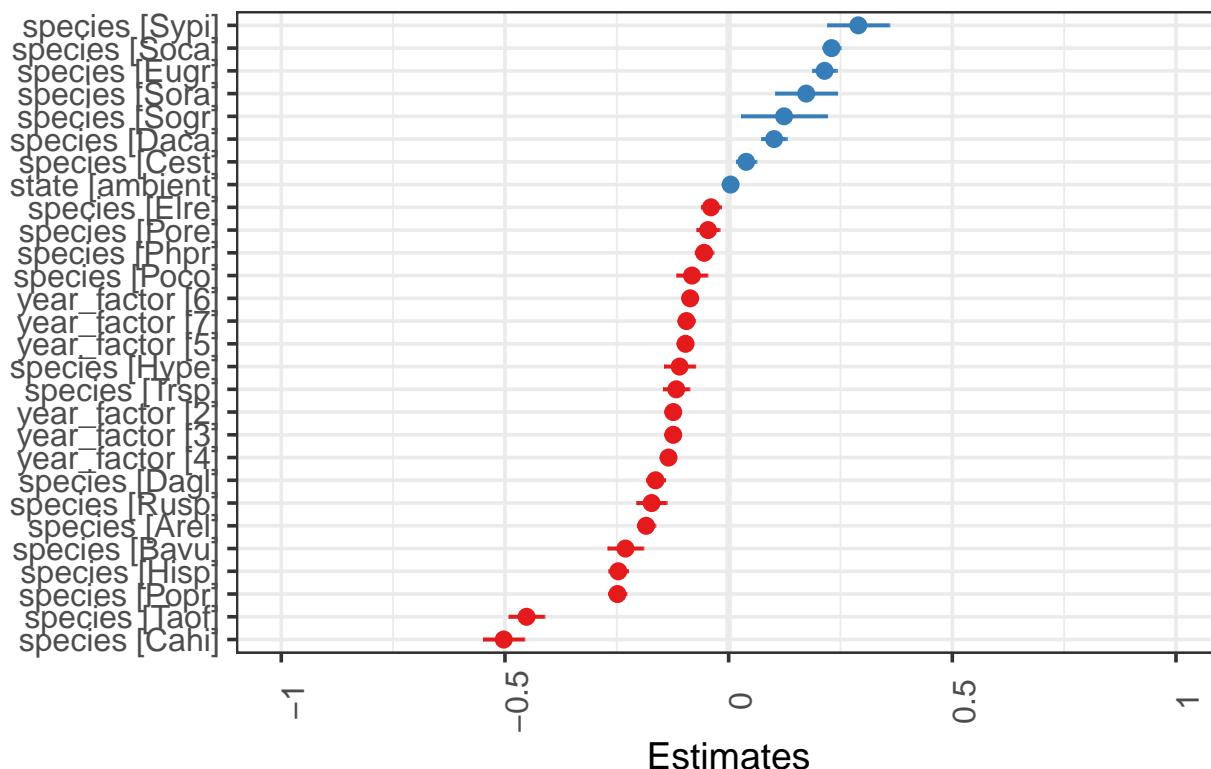
```
## boundary (singular) fit: see ?isSingular
```

```
## Error in emmeans(object = object, specs = specs[[i]], name = contr.name, : No variable named insectici
```

*# Take a look at the estimates for each fixed effect. These are the estimates
from summary(mod7a). You'll see that species vary a lot - and many of them are
different from zero (meaning their half cover date is significantly different
from zero).*

```
plot_model(mod7a, sort.est = TRUE)
```

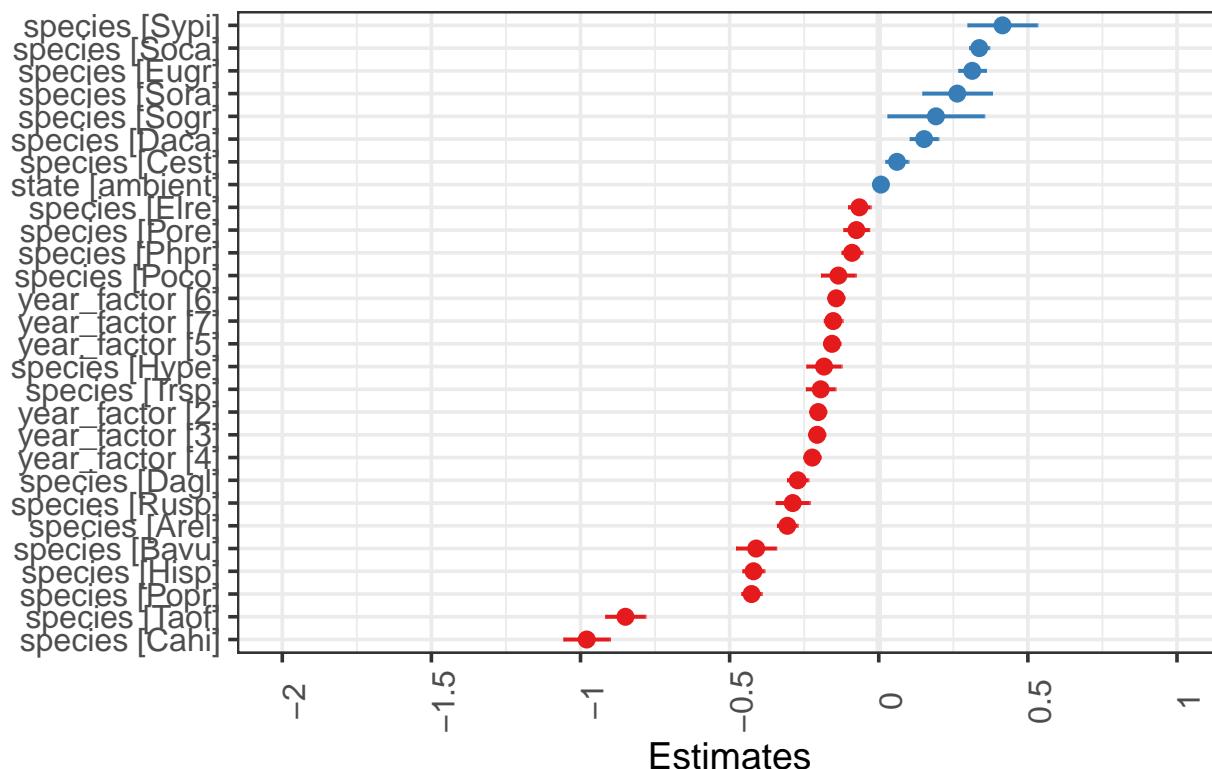
log(julian min)



```
# if you want to standardize the estimates:  
plot_model(mod7a, sort.est = TRUE, type = "std")
```

```
## Formula contains log- or sqrt-terms. See help("standardize") for how such terms are standardized.  
## boundary (singular) fit: see ?isSingular
```

log(julian min)

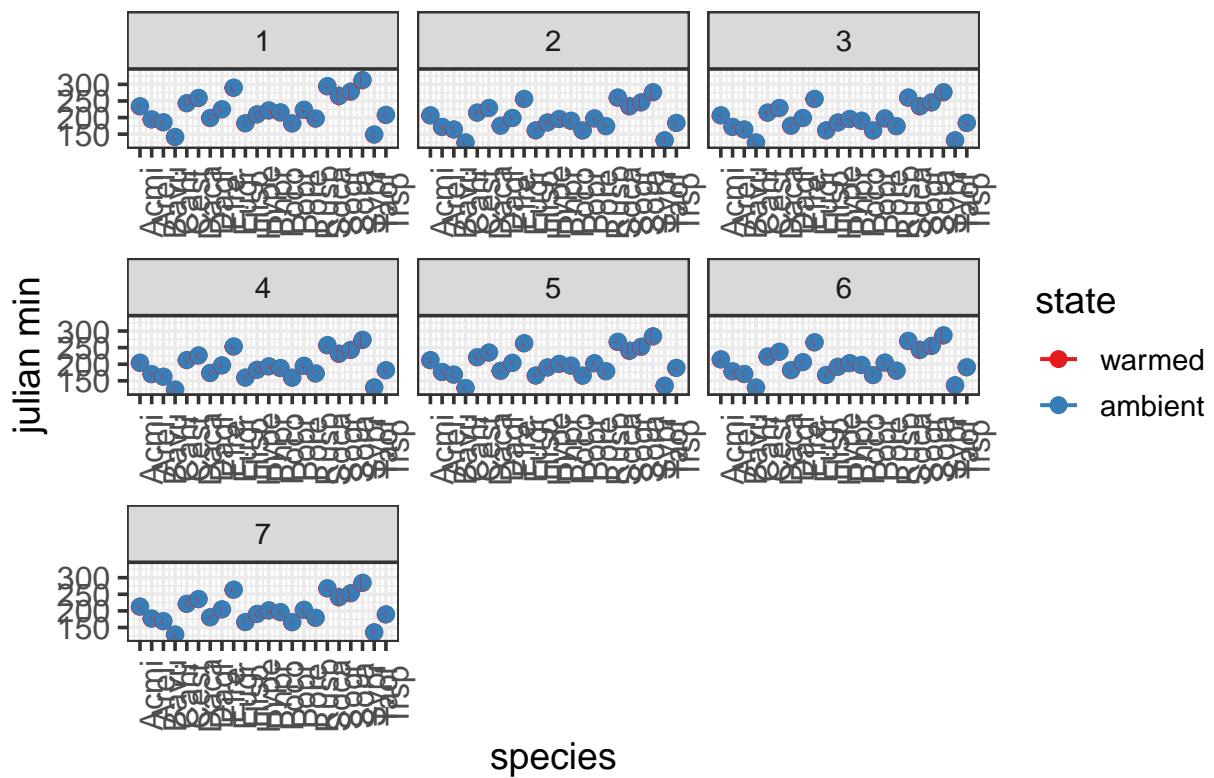


these are the fixed predicted values: - note this is a new plot

```
plot_model(mod7a, type = "pred", terms = c("species", "state", "year_factor"))
```

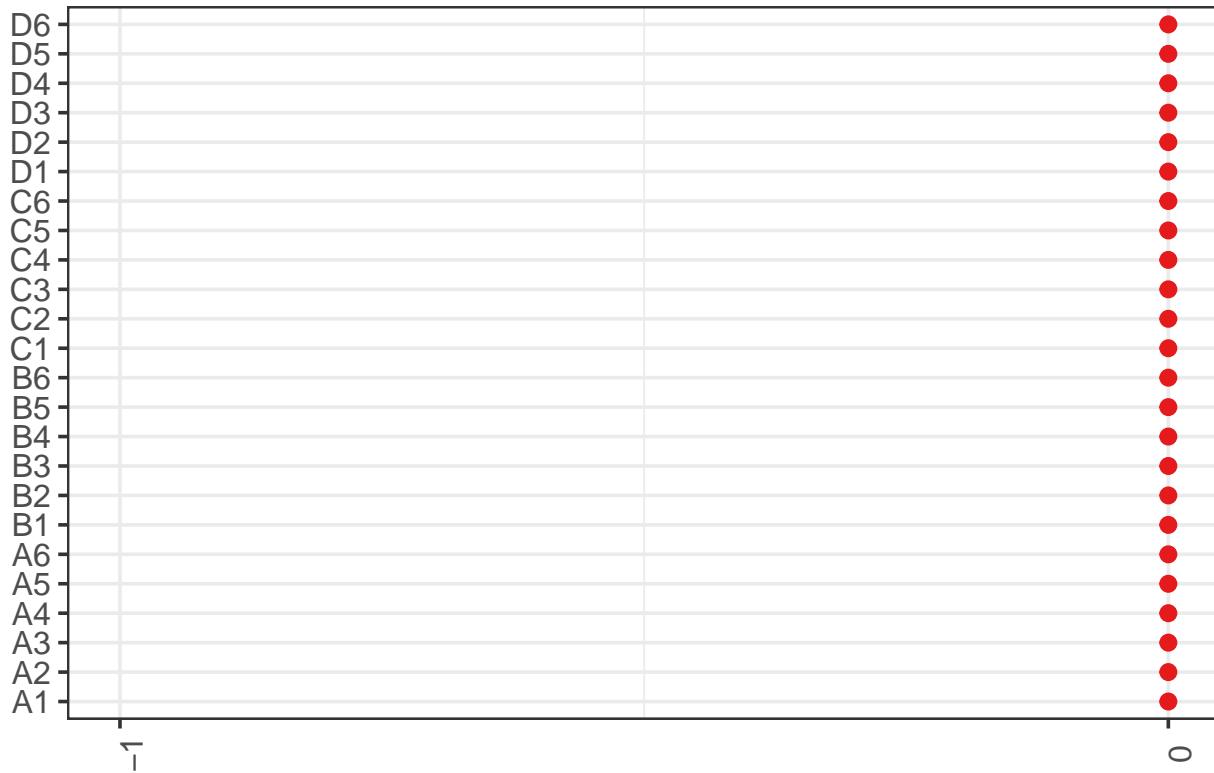
Model has log-transformed response. Back-transforming predictions to original response scale. Standardized residuals are plotted below.

Predicted values of julian min



```
# these are the random effects estimates  
plot_model(mod7a, type = "re")
```

Random effects



```
# including native vs. exotic
kbs_sd_spp <- within(kbs_sd_spp, origin <- relevel(factor(origin), ref = "Native")) # releveling so native is reference level
mod8 <- lmer(log(julian_min) ~ state * origin + (1 + year_factor | plot), kbs_sd_spp,
             REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod9 <- lmer(log(julian_min) ~ state + origin + (1 + year_factor | plot), kbs_sd_spp,
              REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod9a <- lmer(log(julian_min) ~ state + origin + year_factor + (1 | plot), kbs_sd_spp,
               REML = FALSE)

## boundary (singular) fit: see ?isSingular

anova(mod8, mod9) # model 8 is a better fit to data

## Data: kbs_sd_spp
## Models:
## mod9: log(julian_min) ~ state + origin + (1 + year_factor | plot)
## mod8: log(julian_min) ~ state * origin + (1 + year_factor | plot)
##      npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## mod9   33 -1154.2 -999.03 610.08   -1220.2
## mod8   35 -1151.1 -986.56 610.54   -1221.1 0.9268  2     0.6291
```

```

anova(mod8, mod9a) # mod 8

## Data: kbs_sd_spp
## Models:
## mod9a: log(julian_min) ~ state + origin + year_factor + (1 | plot)
## mod8: log(julian_min) ~ state * origin + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod9a    12 -1237.6 -1181.16 630.78   -1261.6
## mod8     35 -1151.1  -986.56 610.54   -1221.1      0 23           1

summary(mod8)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log(julian_min) ~ state * origin + (1 + year_factor | plot)
## Data: kbs_sd_spp
##
##      AIC      BIC      logLik deviance df.resid
## -1151.1  -986.6    610.5   -1221.1      778
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.0296 -0.5743  0.0208  0.6870  3.3227
##
## Random effects:
## Groups   Name        Variance Std.Dev. Corr
## plot     (Intercept) 0.005363 0.07324
##         year_factor2 0.006936 0.08328 -1.00
##         year_factor3 0.011825 0.10874 -1.00  1.00
##         year_factor4 0.009032 0.09504 -1.00  1.00  1.00
##         year_factor5 0.003570 0.05975 -1.00  1.00  1.00  1.00
##         year_factor6 0.002650 0.05148 -1.00  1.00  1.00  1.00  1.00
##         year_factor7 0.007075 0.08411 -1.00  1.00  1.00  1.00  1.00
## Residual          0.012630 0.11238
## Number of obs: 813, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)  5.55011   0.01202 787.41127 461.566 <2e-16 ***
## stateambient -0.01382   0.01717 783.92132 -0.805   0.421    
## originBoth   -0.41492   0.02452 794.37409 -16.919 <2e-16 ***
## originExotic -0.33135   0.01410 793.37366 -23.506 <2e-16 ***
## stateambient:originBoth  0.01734   0.03125 793.27064   0.555   0.579    
## stateambient:originExotic 0.01890   0.01971 793.86699   0.958   0.338    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttmbn orgnBt orgnEx sttm:B
## stateambient -0.700
## originBoth   -0.492  0.344
## originExotic -0.855  0.599  0.424

```

```

## sttmbnt:rgB  0.386 -0.547 -0.785 -0.333
## sttmbnt:rgE  0.611 -0.870 -0.303 -0.715  0.479
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

anova(mod8)

## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state      0.0003  0.0003     1 466.06  0.0254 0.8735
## origin     15.2258  7.6129     2 794.07 602.7549 <2e-16 ***
## state:origin 0.0118  0.0059     2 794.07  0.4653 0.6281
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

emmeans(mod8, list(pairwise ~ state * origin), adjust = "tukey")

## boundary (singular) fit: see ?isSingular

## $`emmeans of state, origin`
##   state origin emmean       SE    df lower.CL upper.CL
##   warmed Native  5.550 0.012960 282.83    5.525  5.576
##   ambient Native  5.536 0.013434 275.92    5.510  5.563
##   warmed Both    5.135 0.022426 574.17    5.091  5.179
##   ambient Both   5.139 0.016618 377.46    5.106  5.171
##   warmed Exotic  5.219 0.008588  57.14    5.202  5.236
##   ambient Exotic 5.224 0.008141  33.56    5.207  5.240
##
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $`pairwise differences of state, origin`
##   1           estimate       SE  df t.ratio p.value
##   warmed Native - ambient Native  0.01382 0.0187 291  0.741  0.9767
##   warmed Native - warmed Both    0.41492 0.0253 745 16.404 <.0001
##   warmed Native - ambient Both  0.41141 0.0211 356 19.522 <.0001
##   warmed Native - warmed Exotic 0.33135 0.0145 742 22.790 <.0001
##   warmed Native - ambient Exotic 0.32628 0.0153 152 21.318 <.0001
##   ambient Native - warmed Both  0.40110 0.0261 495 15.343 <.0001
##   ambient Native - ambient Both 0.39758 0.0199 728 19.952 <.0001
##   ambient Native - warmed Exotic 0.31752 0.0159 176 19.915 <.0001
##   ambient Native - ambient Exotic 0.31245 0.0142 740 21.965 <.0001
##   warmed Both - ambient Both   -0.00352 0.0279 508 -0.126  1.0000
##   warmed Both - warmed Exotic  -0.08358 0.0233 747 -3.582  0.0049
##   warmed Both - ambient Exotic -0.08865 0.0239 449 -3.716  0.0031
##   ambient Both - warmed Exotic -0.08006 0.0187 260 -4.280  0.0004
##   ambient Both - ambient Exotic -0.08513 0.0171 741 -4.989 <.0001
##   warmed Exotic - ambient Exotic -0.00507 0.0118  45 -0.429  0.9980
##
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 6 estimates

```

```

# including growth form - first with interaction term
kbs_flwr_spp <- within(kbs_flwr_spp, growth_habit <- relevel(factor(growth_habit),
  ref = "Forb")) # releveling so forb is the reference

## Error in within(kbs_flwr_spp, growth_habit <- relevel(factor(growth_habit), :
object 'kbs_flwr_spp' r

mod10 <- lmer(log(julian_min) ~ state * growth_habit + (1 + year_factor | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod11 <- lmer(log(julian_min) ~ state + growth_habit + (1 + year_factor | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod11a <- lmer(log(julian_min) ~ state + growth_habit + year_factor + (1 | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

anova(mod10, mod11) # model 10 is a better fit to data

## Data: kbs_sd_spp
## Models:
## mod11: log(julian_min) ~ state + growth_habit + (1 + year_factor | plot)
## mod10: log(julian_min) ~ state * growth_habit + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod11   33 -936.63 -781.50 501.31  -1002.6
## mod10   35 -933.15 -768.62 501.57  -1003.1 0.5171  2      0.7722

anova(mod10, mod11a) # model 10 is still a better fit to data

## Data: kbs_sd_spp
## Models:
## mod11a: log(julian_min) ~ state + growth_habit + year_factor + (1 | plot)
## mod10: log(julian_min) ~ state * growth_habit + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod11a   12 -1031.46 -975.05 527.73  -1055.5
## mod10    35  -933.15 -768.62 501.57  -1003.1     0 23           1

summary(mod10)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log(julian_min) ~ state * growth_habit + (1 + year_factor | plot)
## Data: kbs_sd_spp
##

```

```

##      AIC      BIC  logLik deviance df.resid
## -933.1   -768.6    501.6   -1003.1      778
##
## Scaled residuals:
##      Min      1Q  Median      3Q     Max
## -5.2893 -0.5947 -0.0447  0.7358  3.3213
##
## Random effects:
## Groups   Name        Variance Std.Dev. Corr
## plot     (Intercept) 0.014273 0.11947
##          year_factor2 0.022080 0.14859 -1.00
##          year_factor3 0.029840 0.17274 -1.00  1.00
##          year_factor4 0.018346 0.13545 -1.00  1.00  1.00
##          year_factor5 0.008682 0.09318 -1.00  1.00  1.00  1.00
##          year_factor6 0.006837 0.08269 -1.00  1.00  1.00  1.00  1.00
##          year_factor7 0.013294 0.11530 -1.00  1.00  1.00  1.00  1.00
## Residual           0.016274 0.12757
## Number of obs: 813, groups: plot, 24
##
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                5.14396   0.02429 729.29347 211.819
## stateambient               0.00150   0.02976 717.33145  0.050
## growth_habitForb           0.31371   0.02672 790.82274 11.742
## growth_habitGraminoid      0.06554   0.02595 790.55480  2.525
## stateambient:growth_habitForb -0.02088   0.03325 790.78340 -0.628
## stateambient:growth_habitGraminoid -0.02313   0.03229 792.13634 -0.716
##                               Pr(>|t|)
## (Intercept)                <2e-16 ***
## stateambient                 0.9598
## growth_habitForb            <2e-16 ***
## growth_habitGraminoid       0.0118 *
## stateambient:growth_habitForb 0.5304
## stateambient:growth_habitGraminoid 0.4741
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttmbn grwt_F grwt_G stt:_F
## stateambient -0.816
## grwth_hbtFr -0.905  0.739
## grwth_hbtGr -0.937  0.765  0.848
## sttmbnt:g_F  0.727 -0.888 -0.803 -0.681
## sttmbnt:g_G  0.753 -0.924 -0.681 -0.804  0.820
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

anova(mod10)

```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##                   Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state              0.0199  0.0199     1  448.68  1.2206 0.2698
## growth_habit      12.0940  6.0470     2  792.10 371.5790 <2e-16 ***
## state:growth_habit 0.0084  0.0042     2  792.10  0.2590 0.7719

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

emmeans(mod10, list(pairwise ~ factor(year_factor) * growth_habit), adjust = "tukey")

## boundary (singular) fit: see ?isSingular

## Error in emmeans(object = object, specs = specs[[i]], name = contr.name, : No variable named year_fa

# New version of our model incorporating interaction term and species within year
# so that there is a separate intercept and slope for each species. The issue
# here is that there are some species that are not found each year. Easiest to
# remove those from another version of this dataframe before running below.
# Otherwise, it's not a balanced design. updated mod4
mod12 <- lmer(log(julian_min) ~ state * factor(year_factor) + (1 + factor(year_factor) |
  species), kbs_sd_spp)

## boundary (singular) fit: see ?isSingular

# So another version of this model would include the interaction but not include
# the nesting (and thus would assume that species aren't observed ea yr) updated
# mod5
mod13 <- lmer(log(julian_min) ~ state * factor(year_factor) + (1 | species), kbs_sd_spp)

# All the models ran:
mod1 <- lmer(log(julian_min) ~ state * year_factor + insecticide * year_factor +
  (1 | species) + (1 | plot), kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod2 <- lmer(log(julian_min) ~ state * year_factor + insecticide * year_factor +
  (1 | species), kbs_sd_spp, REML = FALSE)
mod3 <- lmer(log(julian_min) ~ state * year_factor + (1 | species), kbs_sd_spp, REML = FALSE)
mod4 <- lmer(log(julian_min) ~ state * year_factor + insecticide + (1 | species) +
  (1 | plot), kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod5 <- lmer(log(julian_min) ~ state + year_factor + (1 | species), kbs_flwr_spp,
  REML = FALSE)

## Error: bad 'data': object 'kbs_flwr_spp' not found

mod6 <- lmer(log(julian_min) ~ state + year_factor + (1 | species) + (1 + year |
  plot), kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 1 negative eigenvalue: -2.5e-01

```

```

mod7 <- lmer(log(julian_min) ~ state + species + (1 + factor(year_factor) | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 2 negative eigenvalues: -3.4e-02 -3.2e+00

mod7a <- lmer(log(julian_min) ~ state + species + factor(year_factor) + (1 | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod7b <- lmer(log(julian_min) ~ state * factor(year_factor) + species + (1 | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod7c <- lmer(log(julian_min) ~ state + species + factor(year_factor) + insecticide +
  (1 | plot), kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod8 <- lmer(log(julian_min) ~ state * origin + (1 + factor(year_factor) | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod9 <- lmer(log(julian_min) ~ state + origin + (1 + factor(year_factor) | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod9a <- lmer(log(julian_min) ~ state + origin + factor(year_factor) + (1 | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod10 <- lmer(log(julian_min) ~ state * growth_habit + (1 + factor(year_factor) | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod11 <- lmer(log(julian_min) ~ state + growth_habit + (1 + factor(year_factor) | plot),
  kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

```

```

mod11a <- lmer(log(julian_min) ~ state + growth_habit + factor(year_factor) + (1 |
  plot), kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod12 <- lmer(log(julian_min) ~ state * factor(year_factor) + (1 + factor(year_factor) |
  species), kbs_sd_spp)

## boundary (singular) fit: see ?isSingular

mod13 <- lmer(log(julian_min) ~ state * factor(year_factor) + (1 | species), kbs_sd_spp)
AICctab(mod1, mod2, mod3, mod5, mod6, mod7, mod7a, mod7b, mod7c, mod8, mod9, mod9a,
  mod10, mod11, mod11a, mod13, weights = T) #mod7a is the best fitting model (took out mod12 bc it w

## Error in ICtab(..., mnames = mnames, type = "AICc"): nobs different: must have identical data for all

summary(mod7a)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + species + factor(year_factor) + (1 |
##   plot)
## Data: kbs_sd_spp
##
##      AIC      BIC      logLik deviance df.resid
## -2581.8 -2436.0    1321.9   -2643.8      785
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.5921 -0.5306 -0.0049  0.4909  7.6708
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.0000000 0.00000
## Residual           0.002293 0.04789
## Number of obs: 816, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 5.452875  0.011990 816.000000 454.793 < 2e-16 ***
## stateambient 0.004370  0.003507 816.000000  1.246 0.213101
## speciesArel -0.184443  0.010465 816.000000 -17.624 < 2e-16 ***
## speciesBavu -0.230672  0.020305 816.000000 -11.360 < 2e-16 ***
## speciesCahi -0.502774  0.023467 816.000000 -21.425 < 2e-16 ***
## speciesCest  0.039215  0.011618 816.000000  3.375 0.000772 ***
## speciesDaca  0.101541  0.014495 816.000000  7.005 5.16e-12 ***
## speciesDagl -0.163366  0.010831 816.000000 -15.083 < 2e-16 ***
## speciesElre -0.039521  0.011372 816.000000 -3.475 0.000537 ***
## speciesEugr  0.214379  0.013900 816.000000 15.422 < 2e-16 ***
## speciesHisp -0.246588  0.011127 816.000000 -22.161 < 2e-16 ***
## speciesHype -0.109633  0.017723 816.000000 -6.186 9.75e-10 ***

```

```

## speciesPhpr      -0.054832  0.010547 816.000000 -5.199 2.54e-07 ***
## speciesPoco     -0.082002  0.017606 816.000000 -4.658 3.73e-06 ***
## speciesPopr     -0.248610  0.010203 816.000000 -24.368 < 2e-16 ***
## speciesPore     -0.046105  0.013012 816.000000 -3.543 0.000418 ***
## speciesRusp     -0.172216  0.017181 816.000000 -10.024 < 2e-16 ***
## speciesSoca      0.230129  0.010169 816.000000 22.630 < 2e-16 ***
## speciesSogr      0.123923  0.048964 816.000000  2.531 0.011564 *
## speciesSora      0.173318  0.035168 816.000000  4.928 1.00e-06 ***
## speciesSypi      0.289872  0.035383 816.000000  8.192 9.86e-16 ***
## speciesTaof      -0.451748  0.020315 816.000000 -22.237 < 2e-16 ***
## speciesTrsp      -0.117081  0.014949 816.000000 -7.832 1.49e-14 ***
## factor(year_factor)2 -0.123848  0.008143 816.000000 -15.210 < 2e-16 ***
## factor(year_factor)3 -0.123874  0.008310 816.000000 -14.906 < 2e-16 ***
## factor(year_factor)4 -0.134431  0.008450 816.000000 -15.909 < 2e-16 ***
## factor(year_factor)5 -0.096438  0.008651 816.000000 -11.148 < 2e-16 ***
## factor(year_factor)6 -0.086017  0.008330 816.000000 -10.327 < 2e-16 ***
## factor(year_factor)7 -0.094345  0.009449 816.000000 -9.985 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 29 > 12.
## Use print(x, correlation=TRUE)  or
##      vcov(x)      if you need it

## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

```

```
anova(mod7a)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##                                     Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state                      0.0036 0.00356     1    816   1.5527 0.2131
## species                    23.6339 1.12542    21    816 490.7854 <2e-16 ***
## factor(year_factor)  0.7373 0.12289     6    816  53.5895 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

UMBS Species-level Mixed Effects Models

```
# Do we need to include plot as a random effect with the UMBS models?
mod1u <- lmer(log(julian_min) ~ state * year_factor + insecticide * year_factor +
  (1 | species) + (1 | plot), umbs_sd_spp, REML = FALSE)
```

```
## boundary (singular) fit: see ?isSingular
```

```
mod2u <- lmer(log(julian_min) ~ state * year_factor + insecticide * year_factor +
  (1 | species), umbs_sd_spp, REML = FALSE)
# Run analysis of variance on each model (see this for more explanation on how
# anova on a linear mixed effects model is similar to an anova on a regular
# linear model: https://m-clark.github.io/docs/mixedModels/anovamixed.html)
anova(mod1u)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##                               Sum Sq  Mean Sq NumDF DenDF F value    Pr(>F)
## state                      0.002554 0.002554     1 557.25  0.7920  0.373883
## year_factor                 0.226092 0.045218     5 557.09 14.0246 6.245e-13 ***
## insecticide                  0.000371 0.000371     1 557.62  0.1152  0.734404
## state:year_factor           0.049279 0.009856     5 556.99  3.0568  0.009867 **
## year_factor:insecticide     0.013714 0.002743     5 557.06  0.8507  0.514212
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
anova(mod2u)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##                               Sum Sq  Mean Sq NumDF DenDF F value    Pr(>F)
## state                      0.002554 0.002554     1 557.25  0.7920  0.373883
## year_factor                 0.226092 0.045218     5 557.09 14.0246 6.245e-13 ***
## insecticide                  0.000371 0.000371     1 557.62  0.1152  0.734404
## state:year_factor           0.049279 0.009856     5 556.99  3.0568  0.009867 **
## year_factor:insecticide     0.013714 0.002743     5 557.06  0.8507  0.514212
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# Run an ANOVA to test the 2 models to test whether the more complex model is
# significantly better at capturing the data than the simpler model. If the
# resulting p-value is sufficiently low (usually less than 0.05), we conclude
# that the more complex model is significantly better than the simpler model, and
# thus favor the more complex model. If the p-value is not sufficiently low
# (usually greater than 0.05), we should favor the simpler model.
# https://bookdown.org/ndphillips/YaRrrr/comparing-regression-models-with-anova.html
anova(mod1u, mod2u) # models are not different from each other so go with simpler model = model 2

```

```

## Data: umbs_sd_spp
## Models:
## mod2u: log(julian_min) ~ state * year_factor + insecticide * year_factor +
## mod2u:      (1 | species)
## mod1u: log(julian_min) ~ state * year_factor + insecticide * year_factor +
## mod1u:      (1 | species) + (1 | plot)
##      npar   AIC     BIC logLik deviance Chisq Df Pr(>Chisq)
## mod2u   20 -1548 -1461.1    794    -1588
## mod1u   21 -1546 -1454.8    794    -1588     0   1          1

```

```
summary(mod1u)
```

```

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state * year_factor + insecticide * year_factor +
##           (1 | species) + (1 | plot)
## Data: umbs_sd_spp
##
##      AIC      BIC logLik deviance df.resid
## -1546.0 -1454.7    794.0   -1588.0      549
## 

```

```

## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -4.0398 -0.5475 -0.0745  0.4461  6.7673
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   plot     (Intercept) 0.000000 0.00000
##   species (Intercept) 0.020955 0.14476
##   Residual           0.003224 0.05678
## Number of obs: 570, groups: plot, 24; species, 13
##
## Fixed effects:
##                                     Estimate Std. Error      df t value
##   (Intercept)                  5.254347  0.041860 14.657384 125.521
##   stateambient                -0.019576  0.012326 556.846881 -1.588
##   year_factor3                 -0.025558  0.014600 556.854464 -1.751
##   year_factor4                 -0.047509  0.014355 556.928652 -3.309
##   year_factor5                 -0.015195  0.014466 556.928727 -1.050
##   year_factor6                 -0.017244  0.014279 556.942336 -1.208
##   year_factor7                 0.066555  0.021906 556.885156  3.038
##   insecticideno_insects       -0.005063  0.012486 557.134035 -0.406
##   stateambient:year_factor3   0.016808  0.016708 556.838316  1.006
##   stateambient:year_factor4   0.035202  0.016472 557.042223  2.137
##   stateambient:year_factor5   0.043369  0.016303 556.945035  2.660
##   stateambient:year_factor6   0.022944  0.016173 557.023529  1.419
##   stateambient:year_factor7   -0.029702  0.023951 556.976099 -1.240
##   year_factor3:insecticideno_insects 0.013227  0.016733 556.858067  0.790
##   year_factor4:insecticideno_insects 0.007581  0.016517 557.082863  0.459
##   year_factor5:insecticideno_insects 0.009038  0.016399 557.113757  0.551
##   year_factor6:insecticideno_insects -0.011582  0.016277 557.247922 -0.712
##   year_factor7:insecticideno_insects  0.023105  0.023214 557.017457  0.995
##
##                                     Pr(>|t|)
##   (Intercept) < 2e-16 ***
##   stateambient 0.112821
##   year_factor3 0.080574 .
##   year_factor4 0.000995 ***
##   year_factor5 0.293990
##   year_factor6 0.227696
##   year_factor7 0.002491 **
##   insecticideno_insects 0.685259
##   stateambient:year_factor3 0.314837
##   stateambient:year_factor4 0.033026 *
##   stateambient:year_factor5 0.008032 **
##   stateambient:year_factor6 0.156556
##   stateambient:year_factor7 0.215467
##   year_factor3:insecticideno_insects 0.429602
##   year_factor4:insecticideno_insects 0.646403
##   year_factor5:insecticideno_insects 0.581780
##   year_factor6:insecticideno_insects 0.477042
##   year_factor7:insecticideno_insects 0.320026
##   ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```

## Correlation matrix not shown by default, as p = 18 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)      if you need it

## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

summary(mod2u)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state * year_factor + insecticide * year_factor +
##           (1 | species)
## Data: umbs_sd_spp
##
##          AIC      BIC  logLik deviance df.resid
## -1548.0 -1461.1    794.0   -1588.0      550
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.0398 -0.5475 -0.0745  0.4461  6.7673
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## species (Intercept) 0.020955 0.14476
## Residual            0.003224 0.05678
## Number of obs: 570, groups: species, 13
##
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                5.254347  0.041860 14.657430 125.521
## stateambient              -0.019576  0.012326 556.846868 -1.588
## year_factor3               -0.025558  0.014600 556.854451 -1.751
## year_factor4               -0.047509  0.014355 556.928640 -3.309
## year_factor5               -0.015195  0.014466 556.928714 -1.050
## year_factor6               -0.017244  0.014279 556.942323 -1.208
## year_factor7                0.066555  0.021906 556.885144  3.038
## insecticideno_insects     -0.005063  0.012486 557.134023 -0.406
## stateambient:year_factor3  0.016808  0.016708 556.838303  1.006
## stateambient:year_factor4  0.035202  0.016472 557.042211  2.137
## stateambient:year_factor5  0.043369  0.016303 556.945023  2.660
## stateambient:year_factor6  0.022944  0.016173 557.023516  1.419
## stateambient:year_factor7  -0.029702  0.023951 556.976086 -1.240
## year_factor3:insecticideno_insects  0.013227  0.016733 556.858055  0.790
## year_factor4:insecticideno_insects  0.007581  0.016517 557.082851  0.459
## year_factor5:insecticideno_insects  0.009038  0.016399 557.113745  0.551
## year_factor6:insecticideno_insects -0.011582  0.016277 557.247910 -0.712
## year_factor7:insecticideno_insects  0.023105  0.023214 557.017445  0.995
##                               Pr(>|t|)
## (Intercept)                < 2e-16 ***
## stateambient               0.112821
## year_factor3                0.080574 .
## year_factor4                0.000995 ***

```

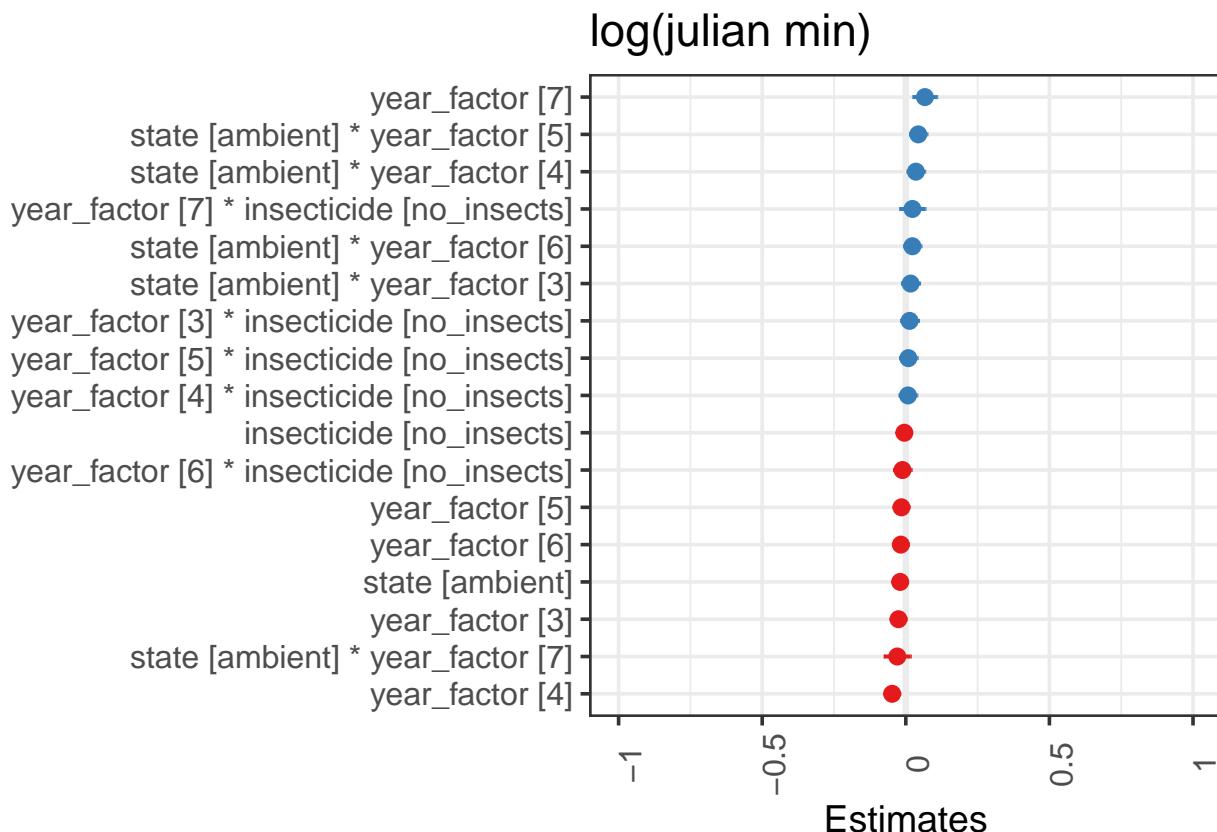
```

## year_factor5          0.293990
## year_factor6          0.227696
## year_factor7          0.002491 **
## insecticideno_insects 0.685259
## stateambient:year_factor3 0.314837
## stateambient:year_factor4 0.033026 *
## stateambient:year_factor5 0.008032 **
## stateambient:year_factor6 0.156556
## stateambient:year_factor7 0.215467
## year_factor3:insecticideno_insects 0.429602
## year_factor4:insecticideno_insects 0.646403
## year_factor5:insecticideno_insects 0.581780
## year_factor6:insecticideno_insects 0.477042
## year_factor7:insecticideno_insects 0.320026
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 18 > 12.
## Use print(x, correlation=TRUE)  or
##      vcov(x)      if you need it

# Next, plot the model. There are multiple variables but here's one way to do it
# based on this package sjPlot:
# https://strengejacke.github.io/sjPlot/articles/plot\_model\_estimates.html
# Annoyingly, this package somehow overwrites the factor order in its plotting so
# we will have to modify the code to get warmed = red. I haven't figured this out
# yet. It does seem to work on some of the plots. hmm. ?plot_model Plot the fixed
# effects estimates for different models these are the fixed effects estimates
plot_model(mod2u, sort.est = TRUE)

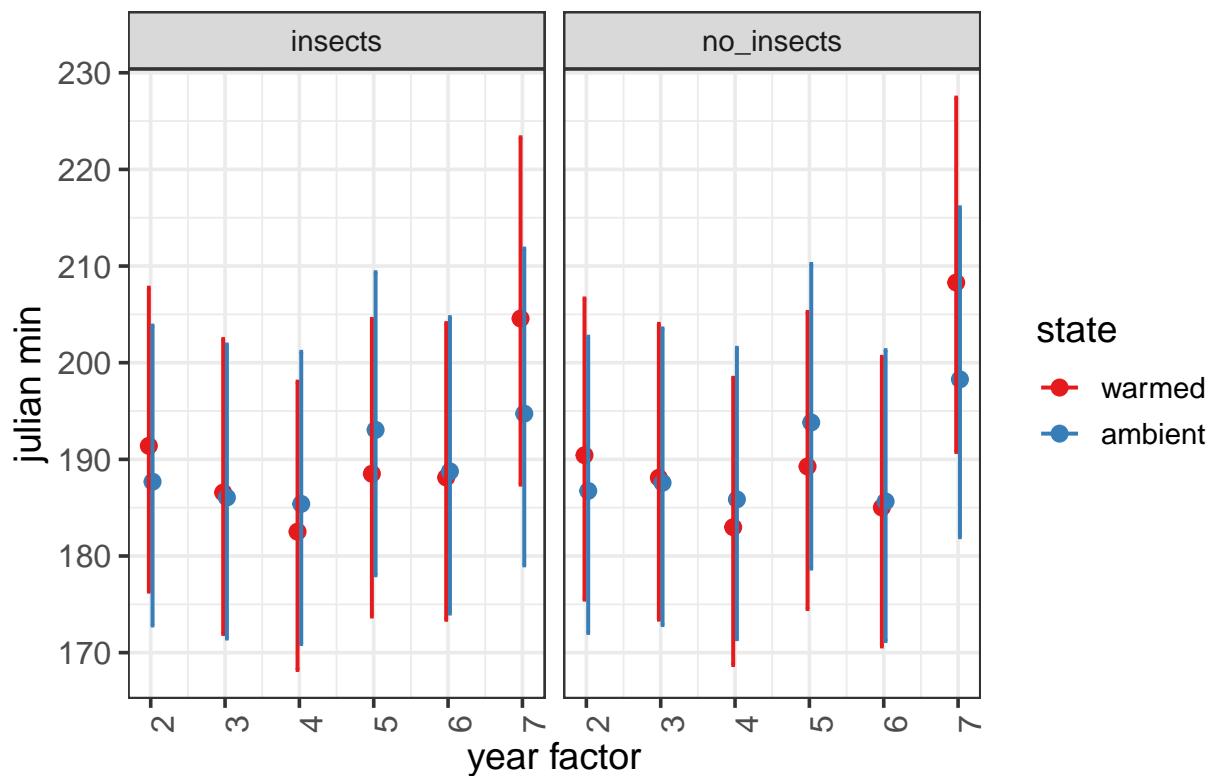
```



```
# these are the fixed predicted values:
plot_model(mod2u, type = "pred", terms = c("year_factor", "state", "insecticide"))
```

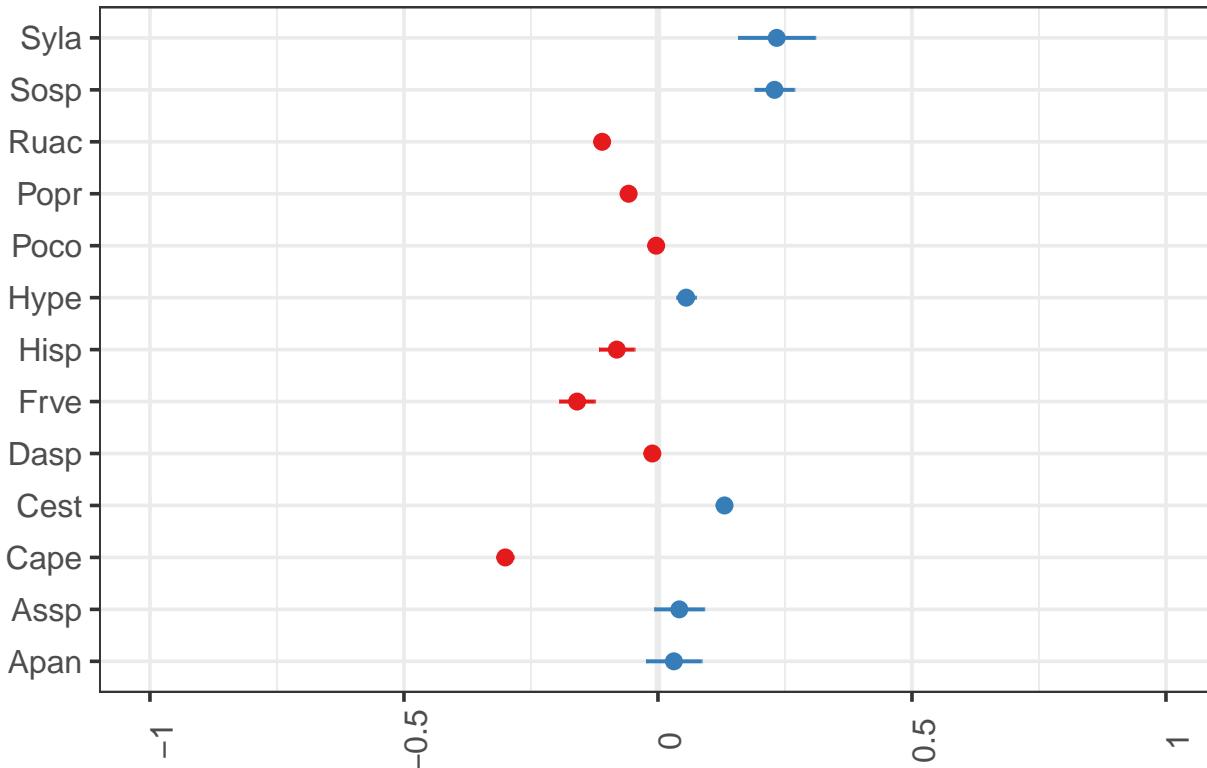
```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standardized
```

Predicted values of julian min



```
# these are the random effects estimates
plot_model(mod2u, type = "re", terms = c("species"))
```

Random effects



```
# Do we need to include insecticide?
mod3u <- lmer(log(julian_min) ~ state * year_factor + (1 | species), umbs_sd_spp,
    REML = FALSE)
anova(mod1u, mod3u) # Don't need insecticide, continue with mod3
```

```
## Data: umbs_sd_spp
## Models:
## mod3u: log(julian_min) ~ state * year_factor + (1 | species)
## mod1u: log(julian_min) ~ state * year_factor + insecticide * year_factor +
## mod1u: (1 | species) + (1 | plot)
##      npar     AIC     BIC logLik deviance Chisq Df Pr(>Chisq)
## mod3u   14 -1555.8 -1494.9 791.88 -1583.8
## mod1u   21 -1546.0 -1454.8 794.00 -1588.0 4.2401  7     0.7517
```

```
# Does year need to be interactive with insecticide? - already removed
# insecticide mod4 <- lmer(log(julian_min) ~ state*year_factor + insecticide +
# (1/species) + (1/plot), umbs_sd_spp, REML=FALSE)
```

```
# Does year need to be interactive with state?
mod5u <- lmer(log(julian_min) ~ state + year_factor + (1 | species) + (1 | plot),
    umbs_sd_spp, REML = FALSE)
```

```
## boundary (singular) fit: see ?isSingular
```

```

anova(mod3u, mod5u) # more complex model is better so go with mod3u

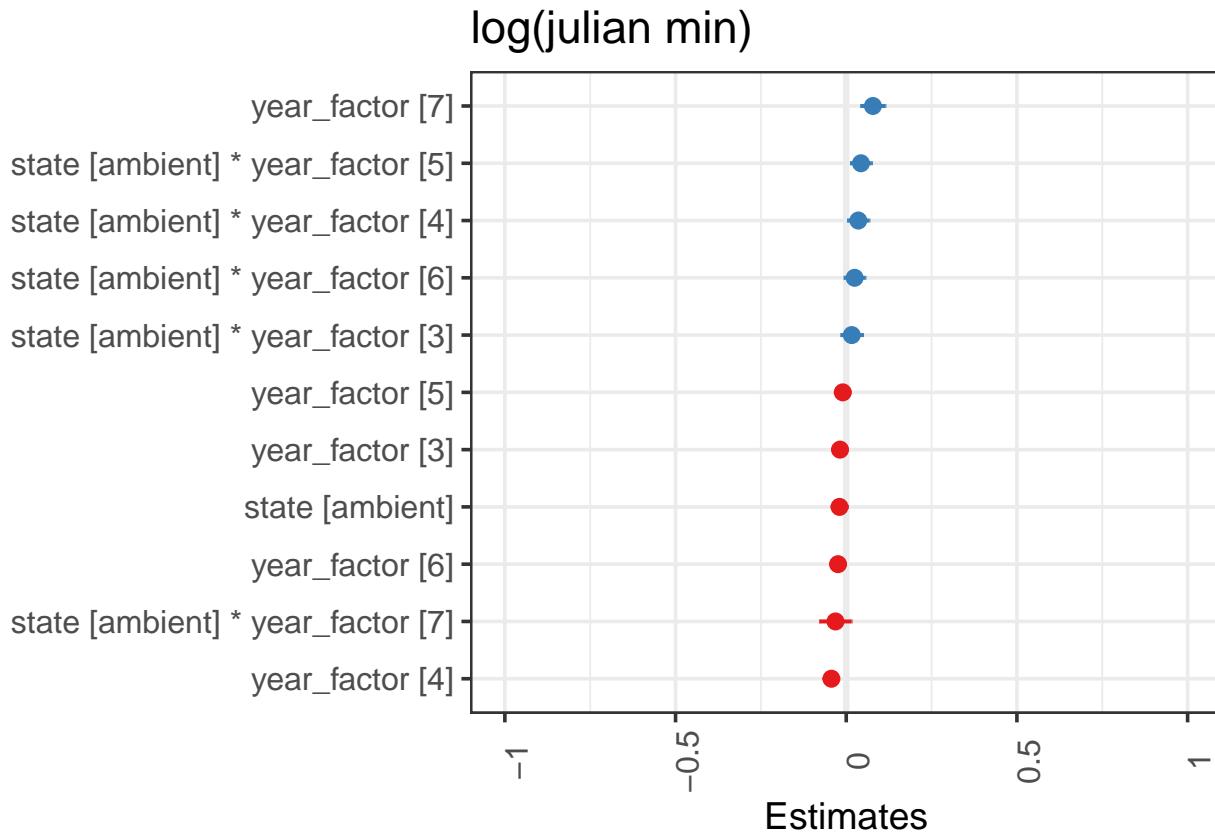
## Data: umbs_sd_spp
## Models:
## mod5u: log(julian_min) ~ state + year_factor + (1 | species) + (1 | 
## mod5u:     plot)
## mod3u: log(julian_min) ~ state * year_factor + (1 | species)
##      npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## mod5u   10 -1548.2 -1504.7 784.09  -1568.2
## mod3u   14 -1555.8 -1494.9 791.88  -1583.8 15.59  4  0.003622 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

AICctab(mod3u, mod5u, weights = T) # go with mod3u

##          dAICc df weight
## mod3u    0.0   14  0.974
## mod5u   7.2   10  0.026

# Plot the fixed effects estimates for different models these are the fixed
# effects estimates from summary(mod5)
plot_model(mod3u, sort.est = TRUE)

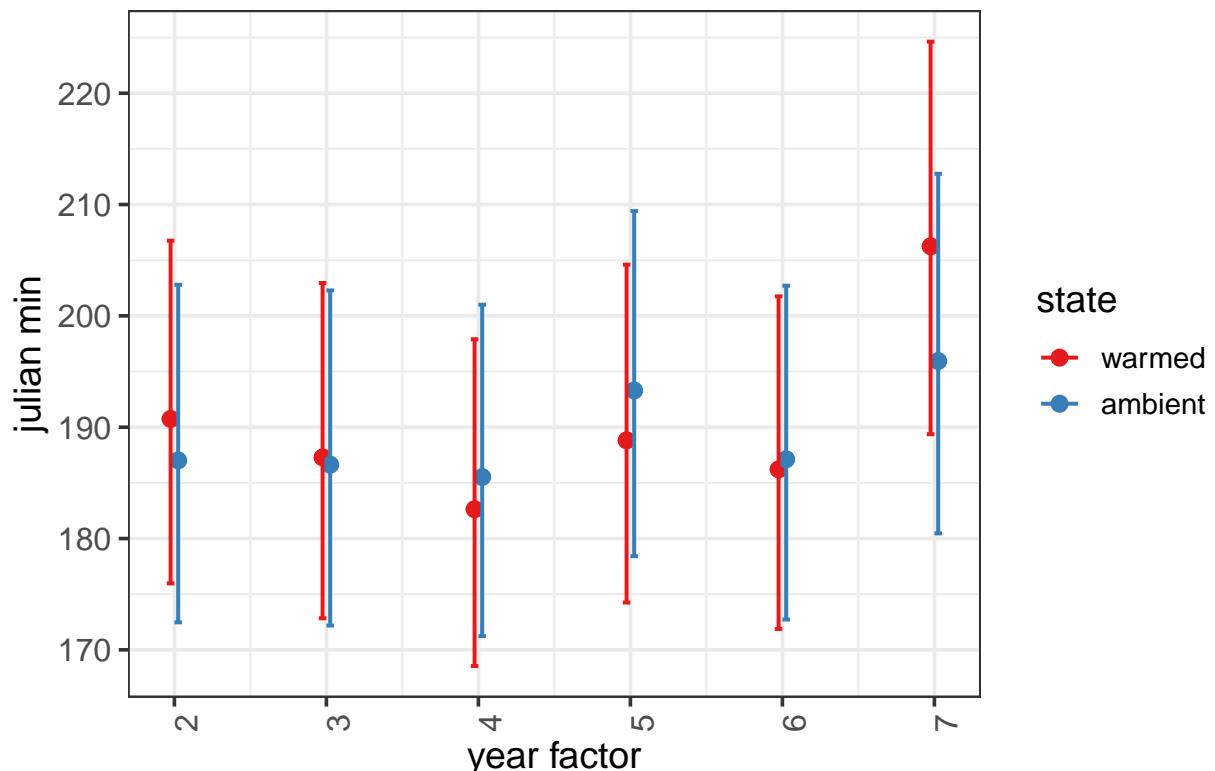
```



```
# these are the fixed predicted values:  
plot_model(mod3u, type = "pred", terms = c("year_factor", "state"))
```

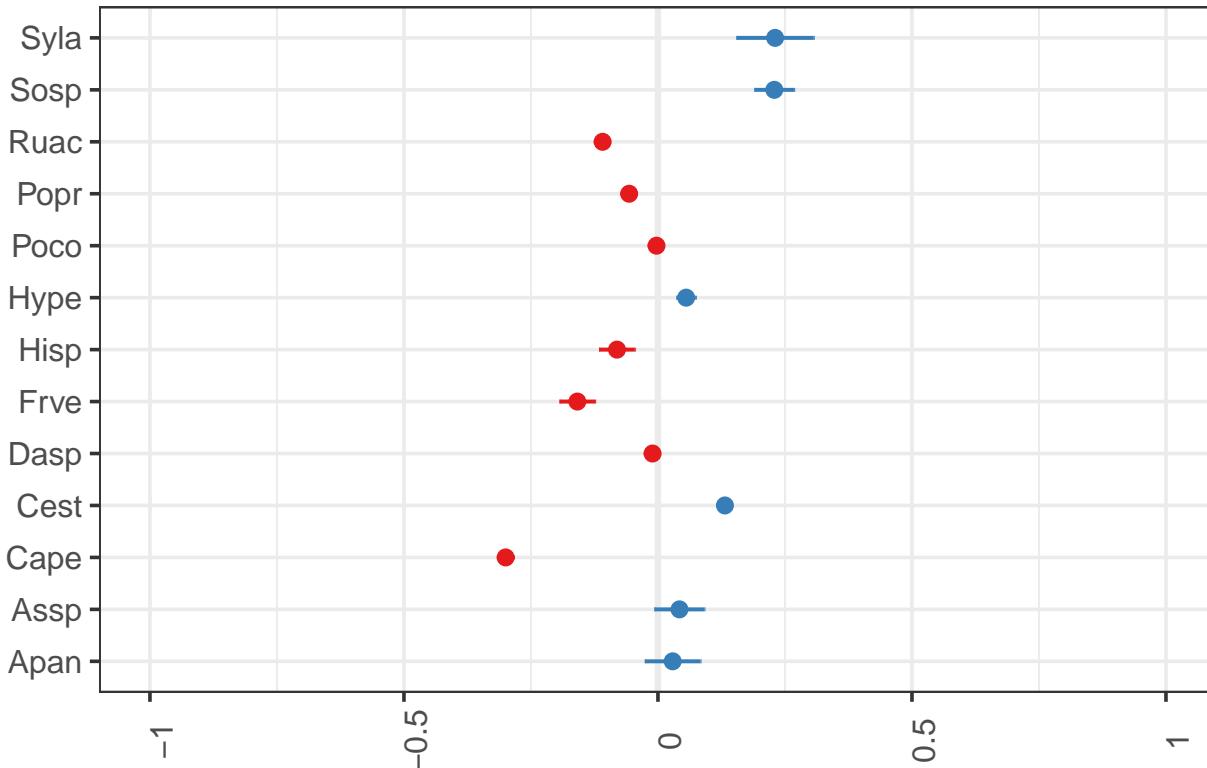
Model has log-transformed response. Back-transforming predictions to original response scale. Standard errors are back-transformed and may not be accurate.

Predicted values of julian min



```
# these are the random effects estimates  
plot_model(mod3u, type = "re", terms = c("species"))
```

Random effects



```
# If we wanted to include plots nested within year it would look like this:
mod6u <- lmer(log(julian_min) ~ state + year_factor + (1 | species) + (1 + year |
  plot), umbs_sd_spp, REML = FALSE)
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -2.5e-02
```

```
# model failed to converge
anova(mod3u, mod6u) #stick with mod3
```

```
## Data: umbs_sd_spp
## Models:
## mod6u: log(julian_min) ~ state + year_factor + (1 | species) + (1 +
## mod6u:      year | plot)
## mod3u: log(julian_min) ~ state * year_factor + (1 | species)
##          npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## mod6u   12 -1543.7 -1491.6 783.86   -1567.7
## mod3u   14 -1555.8 -1494.9 791.88   -1583.8 16.048  2  0.0003274 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod3u)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
```

```

##                                     Sum Sq  Mean Sq NumDF DenDF F value    Pr(>F)
## state                      0.002882 0.002882     1 557.30  0.8867  0.346768
## year_factor                 0.223871 0.044774     5 557.12 13.7785 1.054e-12 ***
## state:year_factor           0.051365 0.010273     5 557.01  3.1613  0.007989 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# We could consider an alternative model that's simpler to understand and also
# one that provides more insight about the species. That would be something like
# this:
mod7u <- lmer(log(julian_min) ~ state + species + (1 + year_factor | plot), umbs_sd_spp,
  REML = FALSE)

## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 1 negative eigenvalue: -1.1e-01

mod7au <- lmer(log(julian_min) ~ state + species + year_factor + (1 | plot), umbs_sd_spp,
  REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod7bu <- lmer(log(julian_min) ~ state * year_factor + species + (1 | plot), umbs_sd_spp,
  REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod7cu <- lmer(log(julian_min) ~ state + species + year_factor + insecticide + (1 |
  plot), umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

anova(mod3u, mod7u)  # model 3u is a better fit to data

## Data: umbs_sd_spp
## Models:
## mod3u: log(julian_min) ~ state * year_factor + (1 | species)
## mod7u: log(julian_min) ~ state + species + (1 + year_factor | plot)
##        npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod3u   14 -1555.8 -1494.9 791.88  -1583.8
## mod7u   36 -1541.2 -1384.8 806.62  -1613.2 29.468 22       0.132

anova(mod3u, mod7au)  #mod 7au

## Data: umbs_sd_spp
## Models:
## mod3u: log(julian_min) ~ state * year_factor + (1 | species)
## mod7au: log(julian_min) ~ state + species + year_factor + (1 | plot)
##        npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod3u   14 -1555.8 -1494.9 791.88  -1583.8
## mod7au   21 -1603.5 -1512.2 822.75  -1645.5 61.74  7  6.777e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
anova(mod7au, mod7bu) #mod7bu
```

```
## Data: umbs_sd_spp
## Models:
## mod7au: log(julian_min) ~ state + species + year_factor + (1 | plot)
## mod7bu: log(julian_min) ~ state * year_factor + species + (1 | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod7au    21 -1603.5 -1512.2 822.75   -1645.5
## mod7bu    26 -1609.3 -1496.3 830.66   -1661.3 15.817  5   0.007387 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod7au, mod7cu) #mod7bu
```

```
## Data: umbs_sd_spp
## Models:
## mod7au: log(julian_min) ~ state + species + year_factor + (1 | plot)
## mod7cu: log(julian_min) ~ state + species + year_factor + insecticide +
## mod7cu:      (1 | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod7au    21 -1603.5 -1512.2 822.75   -1645.5
## mod7cu    22 -1601.5 -1505.9 822.77   -1645.5 0.0341  1     0.8535
```

```
summary(mod7bu)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log(julian_min) ~ state * year_factor + species + (1 | plot)
## Data: umbs_sd_spp
##
##      AIC      BIC      logLik deviance df.resid
## -1609.3 -1496.3     830.7   -1661.3      544
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.1578 -0.5522 -0.0742  0.4366  6.8562
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.000000 0.00000
## Residual           0.003175 0.05635
## Number of obs: 570, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)  5.28120   0.02963 570.00000 178.209 < 2e-16 ***
## stateambient -0.01965   0.01222 570.00000 -1.608 0.108486  
## year_factor3 -0.01824   0.01128 570.00000 -1.617 0.106498  
## year_factor4 -0.04340   0.01126 570.00000 -3.856 0.000128 ***
## year_factor5 -0.01014   0.01124 570.00000 -0.902 0.367467  
## year_factor6 -0.02397   0.01110 570.00000 -2.159 0.031247 *
## year_factor7  0.07843   0.01841 570.00000  4.261 2.38e-05 ***
```

```

## speciesAssp          0.01351   0.03794 570.00000   0.356 0.721859
## speciesCape         -0.33113   0.02952 570.00000  -11.218 < 2e-16 ***
## speciesCest          0.10177   0.02881 570.00000   3.533 0.000445 ***
## speciesDasp          -0.04092   0.02900 570.00000  -1.411 0.158813
## speciesFrve          -0.19135   0.03344 570.00000  -5.722 1.71e-08 ***
## speciesHisp          -0.11219   0.03363 570.00000  -3.336 0.000907 ***
## speciesHype          0.02553   0.02989 570.00000   0.854 0.393327
## speciesPoco          -0.03314   0.02922 570.00000  -1.134 0.257238
## speciesPopr          -0.08703   0.02879 570.00000  -3.023 0.002618 **
## speciesRuac          -0.13950   0.02931 570.00000  -4.760 2.46e-06 ***
## speciesSosp          0.20329   0.03461 570.00000   5.874 7.25e-09 ***
## speciesSyla          0.21872   0.04894 570.00000   4.469 9.48e-06 ***
## stateambient:year_factor3 0.01624   0.01655 570.00000   0.981 0.326769
## stateambient:year_factor4 0.03501   0.01634 570.00000   2.143 0.032564 *
## stateambient:year_factor5 0.04303   0.01613 570.00000   2.667 0.007867 **
## stateambient:year_factor6 0.02413   0.01602 570.00000   1.506 0.132542
## stateambient:year_factor7 -0.03163   0.02369 570.00000  -1.335 0.182358
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 24 > 12.
## Use print(x, correlation=TRUE)  or
##      vcov(x)      if you need it

## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

anova(mod7bu) # investigates whether at least one of the levels within each factor is significantly different from the others.

## Type III Analysis of Variance Table with Satterthwaite's method
##                               Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## state                      0.0030 0.00299     1    570  0.9432  0.33186
## year_factor                 0.2255 0.04510     5    570 14.2063 4.076e-13 ***
## species                     7.8481 0.65401    12    570 205.9914 < 2.2e-16 ***
## state:year_factor           0.0509 0.01018     5    570   3.2077  0.00726 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Yes, at least one of the species is different (they do not all have the median
# first flwr dates).
emmeans(mod7bu, list(pairwise ~ state + year_factor), adjust = "tukey")

## boundary (singular) fit: see ?isSingular

## $`emmeans of state, year_factor`
##   state  year_factor emmean       SE   df lower.CL upper.CL
##   warmed 2            5.253 0.009880 443.7   5.233   5.272
##   ambient 2           5.233 0.010709 421.1   5.212   5.254
##   warmed 3           5.234 0.009290 367.0   5.216   5.253
##   ambient 3           5.231 0.009827 415.3   5.212   5.250
##   warmed 4           5.209 0.009188 366.4   5.191   5.227

```

```

## ambient 4      5.225 0.008862 310.6    5.207  5.242
## warmed 5     5.242 0.009204 350.2    5.224  5.261
## ambient 5      5.266 0.008805 313.9    5.248  5.283
## warmed 6      5.229 0.008831 327.3    5.211  5.246
## ambient 6      5.233 0.008631 272.5    5.216  5.250
## warmed 7      5.331 0.017593 574.7    5.296  5.366
## ambient 7      5.280 0.013202 525.3    5.254  5.306
##
## Results are averaged over the levels of: species
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $'pairwise differences of state, year_factor'
##   1           estimate      SE  df t.ratio p.value
## warmed 2 - ambient 2  0.019652 0.0125 417  1.571  0.9185
## warmed 2 - warmed 3  0.018244 0.0115 579  1.581  0.9154
## warmed 2 - ambient 3  0.021652 0.0119 403  1.821  0.8058
## warmed 2 - warmed 4  0.043402 0.0115 577  3.772  0.0097
## warmed 2 - ambient 4  0.028044 0.0117 375  2.407  0.4042
## warmed 2 - warmed 5  0.010140 0.0115 578  0.882  0.9993
## warmed 2 - ambient 5 -0.013239 0.0115 369 -1.156  0.9917
## warmed 2 - warmed 6  0.023974 0.0114 579  2.112  0.6145
## warmed 2 - ambient 6  0.019500 0.0115 358  1.700  0.8673
## warmed 2 - warmed 7 -0.078435 0.0189 592 -4.161  0.0021
## warmed 2 - ambient 7 -0.027158 0.0149 506 -1.822  0.8053
## ambient 2 - warmed 3 -0.001409 0.0122 396 -0.115  1.0000
## ambient 2 - ambient 3  0.002000 0.0125 579  0.160  1.0000
## ambient 2 - warmed 4  0.023750 0.0122 398  1.951  0.7259
## ambient 2 - ambient 4  0.008392 0.0123 586  0.682  0.9999
## ambient 2 - warmed 5 -0.009512 0.0121 393 -0.783  0.9998
## ambient 2 - ambient 5 -0.032891 0.0121 586 -2.715  0.2215
## ambient 2 - warmed 6  0.004322 0.0120 386  0.359  1.0000
## ambient 2 - ambient 6 -0.000152 0.0121 587 -0.013  1.0000
## ambient 2 - warmed 7 -0.098087 0.0193 558 -5.090 <.0001
## ambient 2 - ambient 7 -0.046810 0.0155 587 -3.027  0.1034
## warmed 3 - ambient 3  0.003408 0.0114 369  0.298  1.0000
## warmed 3 - warmed 4  0.025158 0.0110 577  2.297  0.4800
## warmed 3 - ambient 4  0.009800 0.0112 337  0.879  0.9993
## warmed 3 - warmed 5 -0.008103 0.0109 577 -0.741  0.9999
## warmed 3 - ambient 5 -0.031483 0.0109 331 -2.888  0.1499
## warmed 3 - warmed 6  0.005730 0.0108 577  0.533  1.0000
## warmed 3 - ambient 6  0.001257 0.0109 323  0.115  1.0000
## warmed 3 - warmed 7 -0.096678 0.0186 594 -5.197 <.0001
## warmed 3 - ambient 7 -0.045401 0.0146 494 -3.115  0.0818
## ambient 3 - warmed 4  0.021750 0.0114 373  1.912  0.7512
## ambient 3 - ambient 4  0.006392 0.0115 579  0.556  1.0000
## ambient 3 - warmed 5 -0.011512 0.0114 366 -1.012  0.9974
## ambient 3 - ambient 5 -0.034891 0.0113 579 -3.082  0.0891
## ambient 3 - warmed 6  0.002322 0.0112 359  0.207  1.0000
## ambient 3 - ambient 6 -0.002152 0.0113 581 -0.190  1.0000
## ambient 3 - warmed 7 -0.100087 0.0188 558 -5.328 <.0001
## ambient 3 - ambient 7 -0.048810 0.0148 583 -3.288  0.0490
## warmed 4 - ambient 4 -0.015358 0.0111 339 -1.386  0.9656

```

```

##  warmed 4 - warmed 5  -0.033262 0.0109 577 -3.064  0.0937
##  warmed 4 - ambient 5 -0.056641 0.0108 330 -5.239 <.0001
##  warmed 4 - warmed 6  -0.019428 0.0107 577 -1.819  0.8071
##  warmed 4 - ambient 6 -0.023901 0.0108 321 -2.207  0.5461
##  warmed 4 - warmed 7  -0.121836 0.0186 591 -6.566 <.0001
##  warmed 4 - ambient 7 -0.070559 0.0145 496 -4.854  0.0001
##  ambient 4 - warmed 5 -0.017904 0.0111 335 -1.615  0.9027
##  ambient 4 - ambient 5 -0.041283 0.0110 584 -3.768  0.0098
##  ambient 4 - warmed 6  -0.004070 0.0109 325 -0.374  1.0000
##  ambient 4 - ambient 6 -0.008544 0.0109 582 -0.782  0.9998
##  ambient 4 - warmed 7  -0.106479 0.0186 550 -5.716 <.0001
##  ambient 4 - ambient 7 -0.055201 0.0146 589 -3.776  0.0095
##  warmed 5 - ambient 5  -0.023379 0.0108 323 -2.173  0.5705
##  warmed 5 - warmed 6   0.013834 0.0106 575  1.301  0.9789
##  warmed 5 - ambient 6  0.009360 0.0108 314  0.868  0.9994
##  warmed 5 - warmed 7  -0.088575 0.0185 592 -4.780  0.0001
##  warmed 5 - ambient 7 -0.037298 0.0145 492 -2.566  0.3022
##  ambient 5 - warmed 6  0.037213 0.0106 311  3.521  0.0245
##  ambient 5 - ambient 6 0.032740 0.0107 577  3.074  0.0912
##  ambient 5 - warmed 7  -0.065195 0.0185 549 -3.521  0.0234
##  ambient 5 - ambient 7 -0.013918 0.0144 578 -0.967  0.9983
##  warmed 6 - ambient 6  -0.004474 0.0106 304 -0.422  1.0000
##  warmed 6 - warmed 7  -0.102409 0.0184 590 -5.552 <.0001
##  warmed 6 - ambient 7 -0.051131 0.0144 490 -3.554  0.0211
##  ambient 6 - warmed 7  -0.097935 0.0185 545 -5.292 <.0001
##  ambient 6 - ambient 7 -0.046658 0.0145 582 -3.228  0.0587
##  warmed 7 - ambient 7  0.051277 0.0208 571  2.470  0.3608
##
## Results are averaged over the levels of: species
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 12 estimates

```

```
emmeans(mod7bu, list(pairwise ~ year_factor), adjust = "tukey")
```

```

## boundary (singular) fit: see ?isSingular

## NOTE: Results may be misleading due to involvement in interactions

## $`emmeans of year_factor`
##  year_factor emmean      SE    df lower.CL upper.CL
##  2          5.243 0.008187 440.3   5.227   5.259
##  3          5.233 0.007670 405.5   5.218   5.248
##  4          5.217 0.007127 338.4   5.203   5.231
##  5          5.254 0.007223 337.9   5.240   5.268
##  6          5.231 0.006941 296.8   5.217   5.244
##  7          5.305 0.011584 549.1   5.283   5.328
##
## Results are averaged over the levels of: state, species
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
```

```

## $`pairwise differences of year_factor`
##   1      estimate      SE  df t.ratio p.value
## 2 - 3  0.01012  0.00856 579  1.183  0.8452
## 2 - 4  0.02590  0.00850 582  3.048  0.0289
## 2 - 5 -0.01138  0.00845 583 -1.346  0.7588
## 2 - 6  0.01191  0.00842 585  1.415  0.7178
## 2 - 7 -0.06262  0.01226 590 -5.110 <.0001
## 3 - 4  0.01578  0.00795 578  1.983  0.3531
## 3 - 5 -0.02150  0.00790 578 -2.721  0.0727
## 3 - 6  0.00179  0.00786 580  0.228  0.9999
## 3 - 7 -0.07274  0.01194 591 -6.091 <.0001
## 4 - 5 -0.03727  0.00773 581 -4.820 <.0001
## 4 - 6 -0.01399  0.00765 580 -1.828  0.4484
## 4 - 7 -0.08852  0.01186 590 -7.463 <.0001
## 5 - 6  0.02329  0.00752 576  3.095  0.0251
## 5 - 7 -0.05125  0.01181 588 -4.339  0.0002
## 6 - 7 -0.07453  0.01179 588 -6.324 <.0001
##
## Results are averaged over the levels of: state, species
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 6 estimates

```

```
emmeans(mod7bu, list(pairwise ~ species), adjust = "tukey")
```

```
## boundary (singular) fit: see ?isSingular
```

```

## $`emmeans of species`
##   species emmean      SE  df lower.CL upper.CL
##  Apan     5.276 0.029277 531.1    5.218   5.333
##  Assp     5.289 0.026227 574.2    5.238   5.341
##  Cape     4.945 0.008721 456.9    4.927   4.962
##  Cest     5.377 0.005665 371.9    5.366   5.389
##  Dasp     5.235 0.006171 378.2    5.223   5.247
##  Frve     5.084 0.018578 441.6    5.048   5.121
##  Hisp     5.164 0.018615 511.6    5.127   5.200
##  Hype     5.301 0.010214 495.0    5.281   5.321
##  Poco     5.243 0.007708 471.6    5.227   5.258
##  Popr     5.189 0.005178 300.0    5.178   5.199
##  Ruac     5.136 0.007729 412.6    5.121   5.151
##  Sosp     5.479 0.020798 388.2    5.438   5.520
##  Syla     5.494 0.041413 576.5    5.413   5.576
##
## Results are averaged over the levels of: state, year_factor
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $`pairwise differences of species`
##   1      estimate      SE  df t.ratio p.value
##  Apan - Assp -0.01351  0.03915 545  -0.345  1.0000
##  Apan - Cape  0.33113  0.03042 554  10.886 <.0001
##  Apan - Cest -0.10177  0.02970 548  -3.427  0.0367

```

```

##  Apan - Dasp  0.04092 0.02991 543   1.368 0.9785
##  Apan - Frve  0.19135 0.03457 507   5.536 <.0001
##  Apan - Hisp  0.11219 0.03469 554   3.234 0.0661
##  Apan - Hype -0.02553 0.03081 553  -0.829 0.9998
##  Apan - Poco  0.03314 0.03016 530   1.099 0.9969
##  Apan - Popr  0.08703 0.02969 548   2.932 0.1500
##  Apan - Ruac  0.13950 0.03021 555   4.618 0.0004
##  Apan - Sosp -0.20329 0.03581 485  -5.678 <.0001
##  Apan - Syla -0.21872 0.05051 558  -4.330 0.0012
##  Assp - Cape  0.34464 0.02750 584  12.532 <.0001
##  Assp - Cest -0.08826 0.02680 587  -3.293 0.0553
##  Assp - Dasp  0.05443 0.02694 591   2.021 0.7188
##  Assp - Frve  0.20486 0.03210 554   6.383 <.0001
##  Assp - Hisp  0.12570 0.03224 583   3.898 0.0070
##  Assp - Hype -0.01202 0.02804 586  -0.429 1.0000
##  Assp - Poco  0.04665 0.02731 580   1.708 0.8909
##  Assp - Popr  0.10055 0.02670 586   3.766 0.0115
##  Assp - Ruac  0.15301 0.02726 591   5.612 <.0001
##  Assp - Sosp -0.18978 0.03338 550  -5.686 <.0001
##  Assp - Syla -0.20521 0.04879 589  -4.206 0.0021
##  Cape - Cest -0.43290 0.01022 593  -42.352 <.0001
##  Cape - Dasp -0.29021 0.01061 594  -27.342 <.0001
##  Cape - Frve -0.13978 0.02049 520  -6.821 <.0001
##  Cape - Hisp -0.21894 0.02055 552  -10.656 <.0001
##  Cape - Hype -0.35666 0.01322 571  -26.978 <.0001
##  Cape - Poco -0.29799 0.01139 595  -26.172 <.0001
##  Cape - Popr -0.24409 0.01005 590  -24.288 <.0001
##  Cape - Ruac -0.19163 0.01157 543  -16.557 <.0001
##  Cape - Sosp -0.53442 0.02242 468  -23.834 <.0001
##  Cape - Syla -0.54985 0.04229 577  -13.001 <.0001
##  Cest - Dasp  0.14270 0.00817 587  17.469 <.0001
##  Cest - Frve  0.29312 0.01937 501  15.134 <.0001
##  Cest - Hisp  0.21396 0.01931 552  11.083 <.0001
##  Cest - Hype  0.07624 0.01139 593  6.696 <.0001
##  Cest - Poco  0.13492 0.00925 594  14.588 <.0001
##  Cest - Popr  0.18881 0.00753 577  25.062 <.0001
##  Cest - Ruac  0.24127 0.00939 594  25.699 <.0001
##  Cest - Sosp -0.10152 0.02141 429  -4.742 0.0002
##  Cest - Syla -0.11695 0.04172 581  -2.803 0.2034
##  Dasp - Frve  0.15043 0.01960 502  7.675 <.0001
##  Dasp - Hisp  0.07126 0.01945 570  3.663 0.0166
##  Dasp - Hype -0.06646 0.01179 594  -5.635 <.0001
##  Dasp - Poco -0.00778 0.00979 595  -0.795 0.9999
##  Dasp - Popr  0.04611 0.00799 593  5.768 <.0001
##  Dasp - Ruac  0.09857 0.00979 595  10.068 <.0001
##  Dasp - Sosp -0.24422 0.02163 433  -11.290 <.0001
##  Dasp - Syla -0.25964 0.04185 584  -6.204 <.0001
##  Frve - Hisp -0.07917 0.02621 589  -3.021 0.1193
##  Frve - Hype -0.21688 0.02109 513  -10.286 <.0001
##  Frve - Poco -0.15821 0.02000 526  -7.912 <.0001
##  Frve - Popr -0.10432 0.01928 497  -5.411 <.0001
##  Frve - Ruac -0.05186 0.02014 466  -2.575 0.3289
##  Frve - Sosp -0.39464 0.02781 465  -14.193 <.0001
##  Frve - Syla -0.41007 0.04511 587  -9.090 <.0001

```

```

##  Hisp - Hype -0.13772 0.02118 557  -6.503 <.0001
##  Hisp - Poco -0.07904 0.02011 550  -3.931 0.0062
##  Hisp - Popr -0.02515 0.01928 548  -1.305 0.9855
##  Hisp - Ruac  0.02731 0.02001 571   1.365 0.9789
##  Hisp - Sosp -0.31548 0.02776 521  -11.365 <.0001
##  Hisp - Syla -0.33091 0.04521 595  -7.319 <.0001
##  Hype - Poco  0.05868 0.01251 562   4.691 0.0003
##  Hype - Popr  0.11257 0.01135 592   9.922 <.0001
##  Hype - Ruac  0.16503 0.01271 594  12.987 <.0001
##  Hype - Sosp -0.17776 0.02303 402  -7.718 <.0001
##  Hype - Syla -0.19319 0.04256 573  -4.539 0.0005
##  Poco - Popr  0.05389 0.00917 595  5.876 <.0001
##  Poco - Ruac  0.10635 0.01082 566  9.831 <.0001
##  Poco - Sosp -0.23644 0.02200 453  -10.748 <.0001
##  Poco - Syla -0.25186 0.04206 581  -5.988 <.0001
##  Popr - Ruac  0.05246 0.00922 593  5.690 <.0001
##  Popr - Sosp -0.29033 0.02137 437  -13.587 <.0001
##  Popr - Syla -0.30575 0.04171 581  -7.331 <.0001
##  Ruac - Sosp -0.34279 0.02210 434  -15.509 <.0001
##  Ruac - Syla -0.35822 0.04201 587  -8.527 <.0001
##  Sosp - Syla -0.01543 0.04600 594  -0.335 1.0000
##
## Results are averaged over the levels of: state, year_factor
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 13 estimates

```

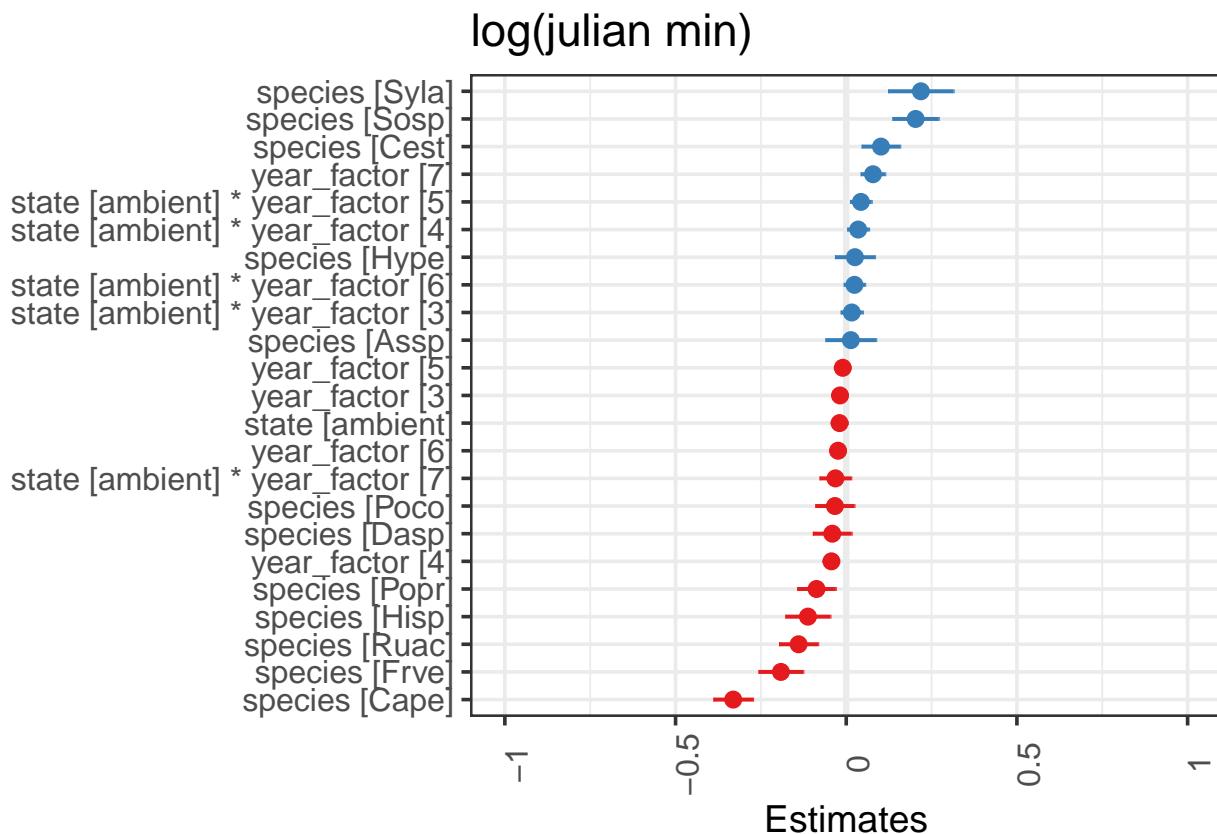
```
emmeans(mod7bu, list(pairwise ~ insecticide), adjust = "tukey")
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Error in emmeans(object = object, specs = specs[[i]], name = contr.name, : No variable named insectici
```

```
# Take a look at the estimates for each fixed effect. These are the estimates
# from summary(mod7a). You'll see that species vary a lot - and many of them are
# different from zero (meaning their half cover date is significantly different
# from zero).
```

```
plot_model(mod7bu, sort.est = TRUE)
```

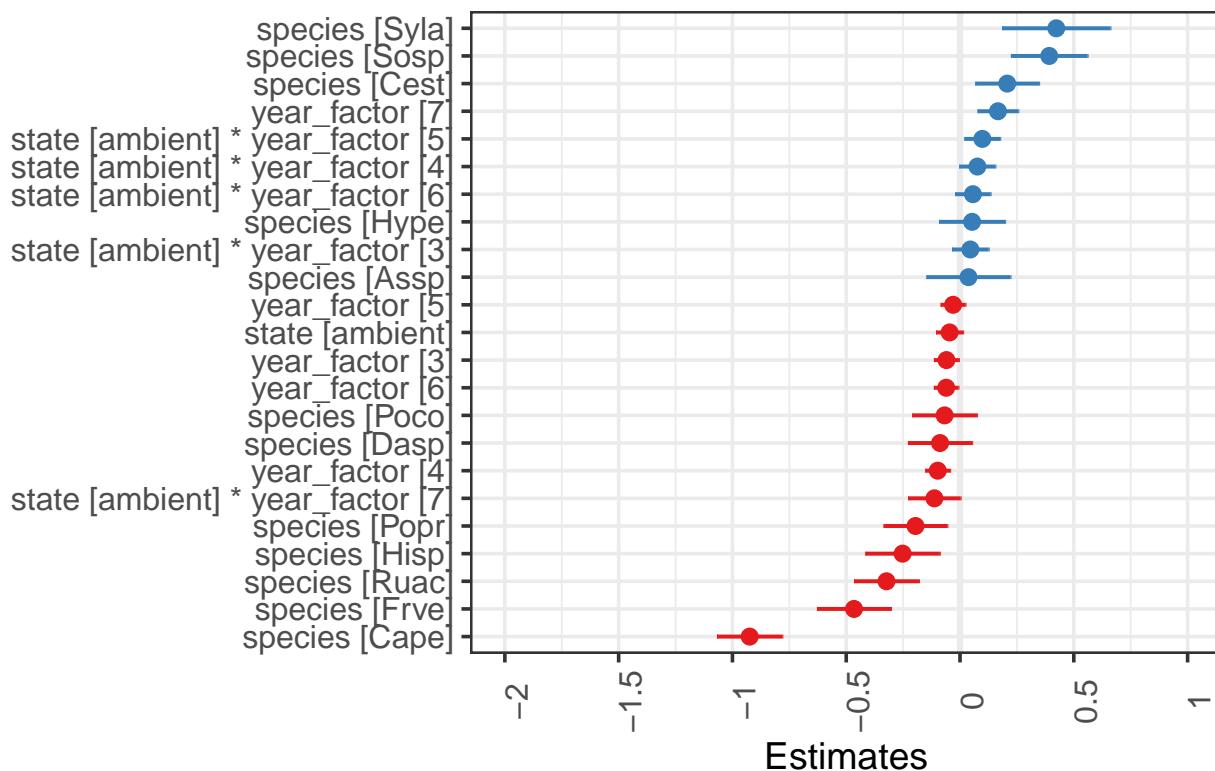


```
# if you want to standardize the estimates:
plot_model(mod7bu, sort.est = TRUE, type = "std")
```

```
## Formula contains log- or sqrt-terms. See help("standardize") for how such terms are standardized.

## boundary (singular) fit: see ?isSingular
```

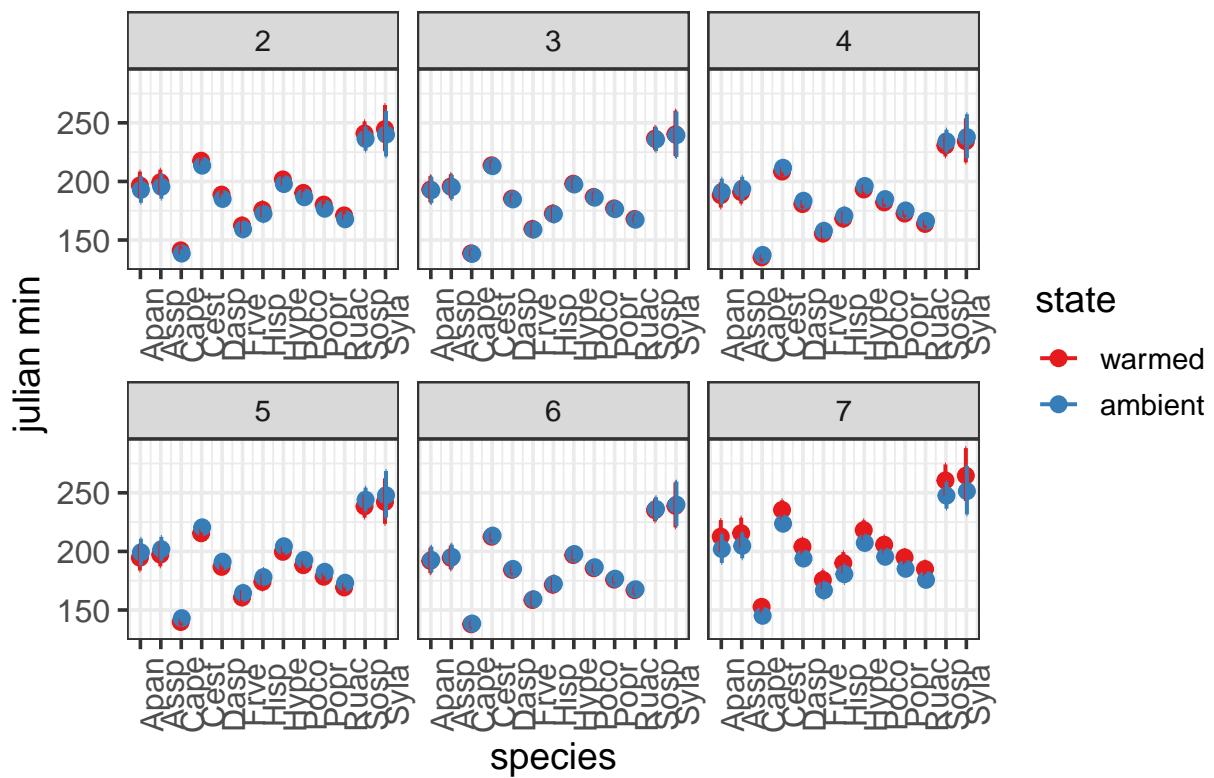
log(julian min)



```
# these are the fixed predicted values: - note this is a new plot
plot_model(mod7bu, type = "pred", terms = c("species", "state", "year_factor"))
```

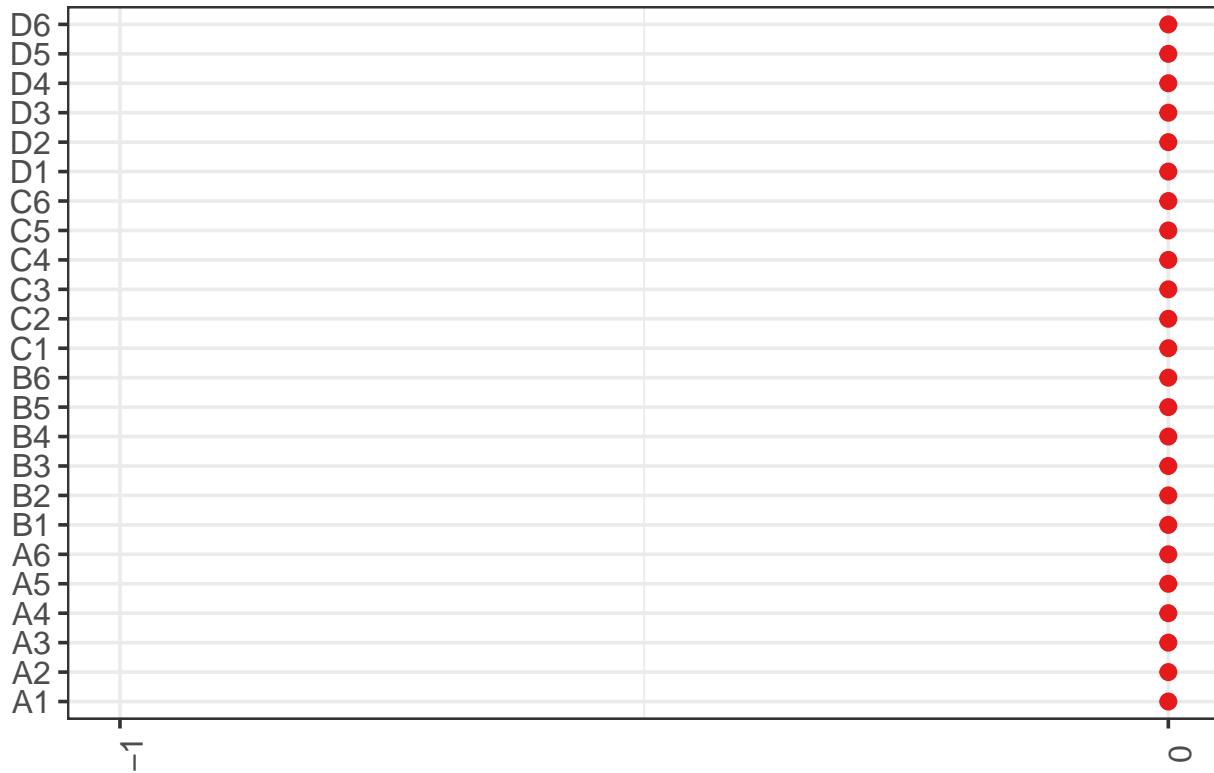
```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standardized
```

Predicted values of julian min



```
# these are the random effects estimates
plot_model(mod7bu, type = "re")
```

Random effects



```
# including native vs. exotic
umbss_sd_spp <- within(kbs_sd_spp, origin <- relevel(factor(origin), ref = "Native")) # releveling so no NA's
mod8u <- lmer(log(julian_min) ~ state * origin + (1 + year_factor | plot), umbss_sd_spp,
               REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod9u <- lmer(log(julian_min) ~ state + origin + (1 + year_factor | plot), umbss_sd_spp,
               REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod9au <- lmer(log(julian_min) ~ state + origin + year_factor + (1 | plot), umbss_sd_spp,
                REML = FALSE)

## boundary (singular) fit: see ?isSingular

anova(mod8u, mod9u) # model 9u is a better fit to data

## Data: umbss_sd_spp
## Models:
## mod9u: log(julian_min) ~ state + origin + (1 + year_factor | plot)
## mod8u: log(julian_min) ~ state * origin + (1 + year_factor | plot)
##      npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## mod9u   33 -1154.2 -999.03 610.08 -1220.2
## mod8u   35 -1151.1 -986.56 610.54 -1221.1 0.9268  2     0.6291
```

```

anova(mod9u, mod9au) # mod 9au

## Data: umbs_sd_spp
## Models:
## mod9au: log(julian_min) ~ state + origin + year_factor + (1 | plot)
## mod9u: log(julian_min) ~ state + origin + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod9au    12 -1237.6 -1181.16 630.78 -1261.6
## mod9u     33 -1154.2 -999.03 610.08 -1220.2      0 21          1

summary(mod9au)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + origin + year_factor + (1 | plot)
##   Data: umbs_sd_spp
##
##      AIC      BIC      logLik deviance df.resid
## -1237.6 -1181.2     630.8 -1261.6      801
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.1280 -0.6127  0.0255  0.6215  3.4904
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   plot     (Intercept) 0.00000  0.0000
##   Residual           0.01241  0.1114
## Number of obs: 813, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 5.636e+00 1.848e-02 8.130e+02 304.969 < 2e-16 ***
## stateambient 3.164e-05 7.921e-03 8.130e+02  0.004 0.996814
## originBoth   -4.029e-01 1.497e-02 8.130e+02 -26.905 < 2e-16 ***
## originExotic -3.207e-01 9.782e-03 8.130e+02 -32.786 < 2e-16 ***
## year_factor2 -1.036e-01 1.836e-02 8.130e+02 -5.640 2.35e-08 ***
## year_factor3 -1.281e-01 1.883e-02 8.130e+02 -6.803 1.99e-11 ***
## year_factor4 -1.178e-01 1.940e-02 8.130e+02 -6.070 1.96e-09 ***
## year_factor5 -7.814e-02 1.994e-02 8.130e+02 -3.919 9.63e-05 ***
## year_factor6 -6.981e-02 1.906e-02 8.130e+02 -3.662 0.000266 ***
## year_factor7 -1.135e-01 2.149e-02 8.130e+02 -5.284 1.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) sttmbn orgnBt orgnEx yr_fc2 yr_fc3 yr_fc4 yr_fc5 yr_fc6
## stateambient -0.198
## originBoth   -0.183 -0.092
## originExotic -0.401 -0.061  0.491
## year_factr2 -0.803 -0.016 -0.093  0.029
## year_factr3 -0.784 -0.004 -0.060  0.023  0.791

```

```

## year_factr4 -0.740 -0.017 -0.039 -0.020  0.761  0.741
## year_factr5 -0.722 -0.032 -0.041 -0.006  0.743  0.722  0.699
## year_factr6 -0.780 -0.013 -0.040  0.042  0.780  0.758  0.731  0.713
## year_factr7 -0.671  0.021 -0.048 -0.030  0.687  0.669  0.648  0.631  0.660
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

anova(mod9au)

## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## state      0.0000  0.0000     1   813    0.00    0.9968
## origin     15.2484  7.6242     2   813  614.58 < 2.2e-16 ***
## year_factor 0.7868  0.1311     6   813   10.57 2.546e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

emmeans(mod9au, list(pairwise ~ state + origin), adjust = "tukey")

## boundary (singular) fit: see ?isSingular

## $`emmeans of state, origin`
##   state  origin emmean       SE   df lower.CL upper.CL
##   warmed Native  5.549 0.009663 172.72    5.530    5.568
##   ambient Native  5.549 0.009794 163.67    5.529    5.568
##   warmed Both     5.146 0.013849 433.30    5.119    5.173
##   ambient Both    5.146 0.013141 336.34    5.120    5.172
##   warmed Exotic   5.228 0.006725  46.92    5.214    5.241
##   ambient Exotic   5.228 0.006168  30.62    5.215    5.241
##
## Results are averaged over the levels of: year_factor
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $`pairwise differences of state, origin`
##   1                  estimate       SE   df t.ratio p.value
##   warmed Native - ambient Native -3.16e-05 0.00800 22.9 -0.004  1.0000
##   warmed Native - warmed Both    4.03e-01 0.01510 821.8 26.689 <.0001
##   warmed Native - ambient Both   4.03e-01 0.01644 273.4 24.507 <.0001
##   warmed Native - warmed Exotic 3.21e-01 0.00986 822.6 32.518 <.0001
##   warmed Native - ambient Exotic 3.21e-01 0.01231 116.9 26.051 <.0001
##   ambient Native - warmed Both   4.03e-01 0.01771 362.6 22.756 <.0001
##   ambient Native - ambient Both   4.03e-01 0.01510 821.8 26.689 <.0001
##   ambient Native - warmed Exotic 3.21e-01 0.01308 140.0 24.530 <.0001
##   ambient Native - ambient Exotic 3.21e-01 0.00986 822.6 32.518 <.0001
##   warmed Both - ambient Both    -3.16e-05 0.00800 22.9 -0.004  1.0000
##   warmed Both - warmed Exotic   -8.22e-02 0.01339 823.1 -6.134 <.0001
##   warmed Both - ambient Exotic   -8.22e-02 0.01598 274.6 -5.144 <.0001
##   ambient Both - warmed Exotic   -8.21e-02 0.01521 209.6 -5.398 <.0001
##   ambient Both - ambient Exotic   -8.22e-02 0.01339 823.1 -6.134 <.0001
##   warmed Exotic - ambient Exotic -3.16e-05 0.00800 22.9 -0.004  1.0000

```

```

##  

## Results are averaged over the levels of: year_factor  

## Degrees-of-freedom method: kenward-roger  

## Results are given on the log (not the response) scale.  

## P value adjustment: tukey method for comparing a family of 6 estimates

# including growth form - first with interaction term
umbs_sd_spp <- within(umbs_sd_spp, growth_habit <- relevel(factor(growth_habit),
  ref = "Forb")) # releveling so forb is the reference
mod10u <- lmer(log(julian_min) ~ state * growth_habit + (1 + year_factor | plot),
  umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod11u <- lmer(log(julian_min) ~ state + growth_habit + (1 + year_factor | plot),
  umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod11au <- lmer(log(julian_min) ~ state + growth_habit + year_factor + (1 | plot),
  umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

anova(mod10u, mod11u) # model 11u is a better fit to data

## Data: umbs_sd_spp
## Models:
## mod11u: log(julian_min) ~ state + growth_habit + (1 + year_factor | plot)
## mod10u: log(julian_min) ~ state * growth_habit + (1 + year_factor | plot)
##          npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod11u   33 -936.63 -781.50 501.31   -1002.6
## mod10u   35 -933.15 -768.62 501.57   -1003.1 0.5171  2      0.7722

anova(mod11u, mod11au) # model 11u is still a better fit to data

## Data: umbs_sd_spp
## Models:
## mod11au: log(julian_min) ~ state + growth_habit + year_factor + (1 | plot)
## mod11u: log(julian_min) ~ state + growth_habit + (1 + year_factor | plot)
##          npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod11au   12 -1031.46 -975.05 527.73   -1055.5
## mod11u    33 -936.63 -781.50 501.31   -1002.6      0 21           1

summary(mod11u)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + growth_habit + (1 + year_factor | plot)

```

```

##      Data: umbs_sd_spp
##
##      AIC      BIC  logLik deviance df.resid
## -936.6   -781.5   501.3   -1002.6      780
##
## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -5.2851 -0.5989 -0.0475  0.7416  3.3066
##
## Random effects:
## Groups   Name        Variance Std.Dev. Corr
## plot     (Intercept) 0.014242 0.11934
##          year_factor2 0.022091 0.14863 -1.00
##          year_factor3 0.029831 0.17272 -1.00  1.00
##          year_factor4 0.018320 0.13535 -1.00  1.00  1.00
##          year_factor5 0.008661 0.09306 -1.00  1.00  1.00  1.00
##          year_factor6 0.006836 0.08268 -1.00  1.00  1.00  1.00  1.00
##          year_factor7 0.013196 0.11487 -1.00  1.00  1.00  1.00  1.00
## Residual           0.016284 0.12761
## Number of obs: 813, groups: plot, 24
##
## Fixed effects:
##                   Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)      5.457329  0.009114 606.673254 598.794 <2e-16 ***
## stateambient    -0.018677  0.009135 586.364698 -2.045  0.0413 *
## growth_habit     -0.299861  0.015877 790.434459 -18.887 <2e-16 ***
## growth_habitGraminoid -0.249369  0.009817 792.730576 -25.402 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) sttmbn grwth_
## stateambient -0.553
## growth_habit -0.351 -0.062
## grwth_hbtGr -0.656  0.014  0.355
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

```

```
anova(mod11u)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##                   Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state             0.0681  0.0681     1 586.36   4.1801 0.04135 *
## growth_habit    12.3186  6.1593     2 789.42 378.2403 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
emmeans(mod11u, list(pairwise ~ growth_habit), adjust = "tukey")
```

```
## boundary (singular) fit: see ?isSingular
```

```
## $`emmeans of growth_habit`
```

```

##  growth_habit emmean      SE   df lower.CL upper.CL
##  Forb          5.448 0.008641 133.39    5.431    5.465
##              5.148 0.015138 493.51    5.118    5.178
##  Graminoid     5.199 0.007251  65.85    5.184    5.213
##
## Results are averaged over the levels of: state
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $'pairwise differences of growth_habit'
##  1           estimate      SE  df t.ratio p.value
##  Forb -            0.2999 0.0163 736 18.365 <.0001
##  Forb - Graminoid  0.2494 0.0102 749 24.424 <.0001
##  - Graminoid     -0.0505 0.0161 751 -3.143  0.0049
##
## Results are averaged over the levels of: state
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 3 estimates

# New version of our model incorporating interaction term and species within year
# so that there is a separate intercept and slope for each species. The issue
# here is that there are some species that are not found each year. Easiest to
# remove those from another version of this dataframe before running below.
# Otherwise, it's not a balanced design. updated mod4
mod12u <- lmer(log(julian_min) ~ state * factor(year_factor) + (1 + factor(year_factor) | species), umbs_sd_spp)

## boundary (singular) fit: see ?isSingular

# So another version of this model would include the interaction but not include
# the nesting (and thus would assume that species aren't observed ea yr) updated
# mod5
mod13u <- lmer(log(julian_min) ~ state * factor(year_factor) + (1 | species), umbs_sd_spp)

# All the models ran:
mod1u <- lmer(log(julian_min) ~ state * year_factor + insecticide * year_factor +
  (1 | species) + (1 | plot), umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod2u <- lmer(log(julian_min) ~ state * year_factor + insecticide * year_factor +
  (1 | species), umbs_sd_spp, REML = FALSE)
mod3u <- lmer(log(julian_min) ~ state * year_factor + (1 | species), umbs_sd_spp,
  REML = FALSE)
mod4u <- lmer(log(julian_min) ~ state * year_factor + insecticide + (1 | species) +
  (1 | plot), umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

```

```

mod5u <- lmer(log(julian_min) ~ state + year_factor + (1 | species), umbs_sd_spp,
  REML = FALSE)
mod6u <- lmer(log(julian_min) ~ state + year_factor + (1 | species) + (1 + year |
  plot), umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 1 negative eigenvalue: -2.5e-01

mod7u <- lmer(log(julian_min) ~ state + species + (1 + factor(year_factor) | plot),
  umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 2 negative eigenvalues: -3.4e-02 -3.2e+00

mod7au <- lmer(log(julian_min) ~ state + species + factor(year_factor) + (1 | plot),
  umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod7bu <- lmer(log(julian_min) ~ state * factor(year_factor) + species + (1 | plot),
  umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod7cu <- lmer(log(julian_min) ~ state + species + factor(year_factor) + insecticide +
  (1 | plot), umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod8u <- lmer(log(julian_min) ~ state * origin + (1 + factor(year_factor) | plot),
  umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod9u <- lmer(log(julian_min) ~ state + origin + (1 + factor(year_factor) | plot),
  umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod9au <- lmer(log(julian_min) ~ state + origin + factor(year_factor) + (1 | plot),
  umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

```

```

mod10u <- lmer(log(julian_min) ~ state * growth_habit + (1 + factor(year_factor) |
  plot), umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod11u <- lmer(log(julian_min) ~ state + growth_habit + (1 + factor(year_factor) |
  plot), umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod11au <- lmer(log(julian_min) ~ state + growth_habit + factor(year_factor) + (1 |
  plot), umbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod12u <- lmer(log(julian_min) ~ state * factor(year_factor) + (1 + factor(year_factor) |
  species), umbs_sd_spp)

## boundary (singular) fit: see ?isSingular

mod13u <- lmer(log(julian_min) ~ state * factor(year_factor) + (1 | species), umbs_sd_spp)
AICctab(mod1u, mod2u, mod3u, mod5u, mod6u, mod7u, mod7au, mod7bu, mod7cu, mod8u,
  mod9u, mod9au, mod10u, mod11u, mod11au, mod13u, weights = T) # idk why this wont run

## Error in ICtab(..., mnames = mnames, type = "AICc"): nobs different: must have identical data for all

AICctab(mod3u, mod9au, mod11u, weights = T)

## Error in ICtab(..., mnames = mnames, type = "AICc"): nobs different: must have identical data for all

summary(mod7au)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + species + factor(year_factor) + (1 |
##   plot)
## Data: umbs_sd_spp
##
##       AIC      BIC    logLik deviance df.resid
## -2581.8 -2436.0  1321.9   -2643.8      785
##
## Scaled residuals:
##       Min     1Q Median     3Q    Max
## -4.5921 -0.5306 -0.0049  0.4909  7.6708
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.000000 0.000000

```

```

## Residual           0.002293 0.04789
## Number of obs: 816, groups: plot, 24
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                5.452875  0.011990 816.000000 454.793 < 2e-16 ***
## stateambient               0.004370  0.003507 816.000000   1.246 0.213101
## speciesArel                -0.184443  0.010465 816.000000 -17.624 < 2e-16 ***
## speciesBavu                 -0.230672  0.020305 816.000000 -11.360 < 2e-16 ***
## speciesCahi                 -0.502774  0.023467 816.000000 -21.425 < 2e-16 ***
## speciesCest                 0.039215  0.011618 816.000000   3.375 0.000772 ***
## speciesDaca                 0.101541  0.014495 816.000000   7.005 5.16e-12 ***
## speciesDagl                 -0.163366  0.010831 816.000000 -15.083 < 2e-16 ***
## speciesElre                 -0.039521  0.011372 816.000000 -3.475 0.000537 ***
## speciesEugr                 0.214379  0.013900 816.000000  15.422 < 2e-16 ***
## speciesHisp                 -0.246588  0.011127 816.000000 -22.161 < 2e-16 ***
## speciesHype                 -0.109633  0.017723 816.000000 -6.186 9.75e-10 ***
## speciesPhpr                 -0.054832  0.010547 816.000000 -5.199 2.54e-07 ***
## speciesPoco                 -0.082002  0.017606 816.000000 -4.658 3.73e-06 ***
## speciesPopr                 -0.248610  0.010203 816.000000 -24.368 < 2e-16 ***
## speciesPore                 -0.046105  0.013012 816.000000 -3.543 0.000418 ***
## speciesRusp                 -0.172216  0.017181 816.000000 -10.024 < 2e-16 ***
## speciesSoca                 0.230129  0.010169 816.000000  22.630 < 2e-16 ***
## speciesSogr                 0.123923  0.048964 816.000000   2.531 0.011564 *
## speciesSora                 0.173318  0.035168 816.000000   4.928 1.00e-06 ***
## speciesSypi                 0.289872  0.035383 816.000000   8.192 9.86e-16 ***
## speciesTaof                 -0.451748  0.020315 816.000000 -22.237 < 2e-16 ***
## speciesTrsp                 -0.117081  0.014949 816.000000 -7.832 1.49e-14 ***
## factor(year_factor)2        -0.123848  0.008143 816.000000 -15.210 < 2e-16 ***
## factor(year_factor)3        -0.123874  0.008310 816.000000 -14.906 < 2e-16 ***
## factor(year_factor)4        -0.134431  0.008450 816.000000 -15.909 < 2e-16 ***
## factor(year_factor)5        -0.096438  0.008651 816.000000 -11.148 < 2e-16 ***
## factor(year_factor)6        -0.086017  0.008330 816.000000 -10.327 < 2e-16 ***
## factor(year_factor)7        -0.094345  0.009449 816.000000 -9.985 < 2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 29 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)      if you need it

## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

anova(mod7au)

## Type III Analysis of Variance Table with Satterthwaite's method
##                               Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state                      0.0036  0.00356     1    816   1.5527 0.2131
## species                     23.6339  1.12542    21    816 490.7854 <2e-16 ***
## factor(year_factor)       0.7373  0.12289     6    816  53.5895 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```