

warmXtrophic Project: Seed Set Phenology Analyses

Moriah Young

February 10, 2022

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)
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 (Ariel 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
sd_plot_origin <- read.csv(file.path(L2_dir, "phenology/final_sd_plot_origin_L2.csv")) # plot level da
sd_plot_growthhabit <- read.csv(file.path(L2_dir, "phenology/final_sd_plot_growthhabit_L2.csv")) # plo

# get rid of 'X' column that shows up
sd_species$X <- NULL
sd_plot$X <- NULL
sd_plot_origin$X <- NULL
sd_plot_growthhabit$X <- NULL

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
kbs_sd_plot_origin <- subset(sd_plot_origin, site == "kbs")
kbs_sd_plot_growthhabit <- subset(sd_plot_growthhabit, site == "kbs")
umbs_sd_plot_origin <- subset(sd_plot_origin, site == "umbs")
umbs_sd_plot_growthhabit <- subset(sd_plot_growthhabit, site == "umbs")

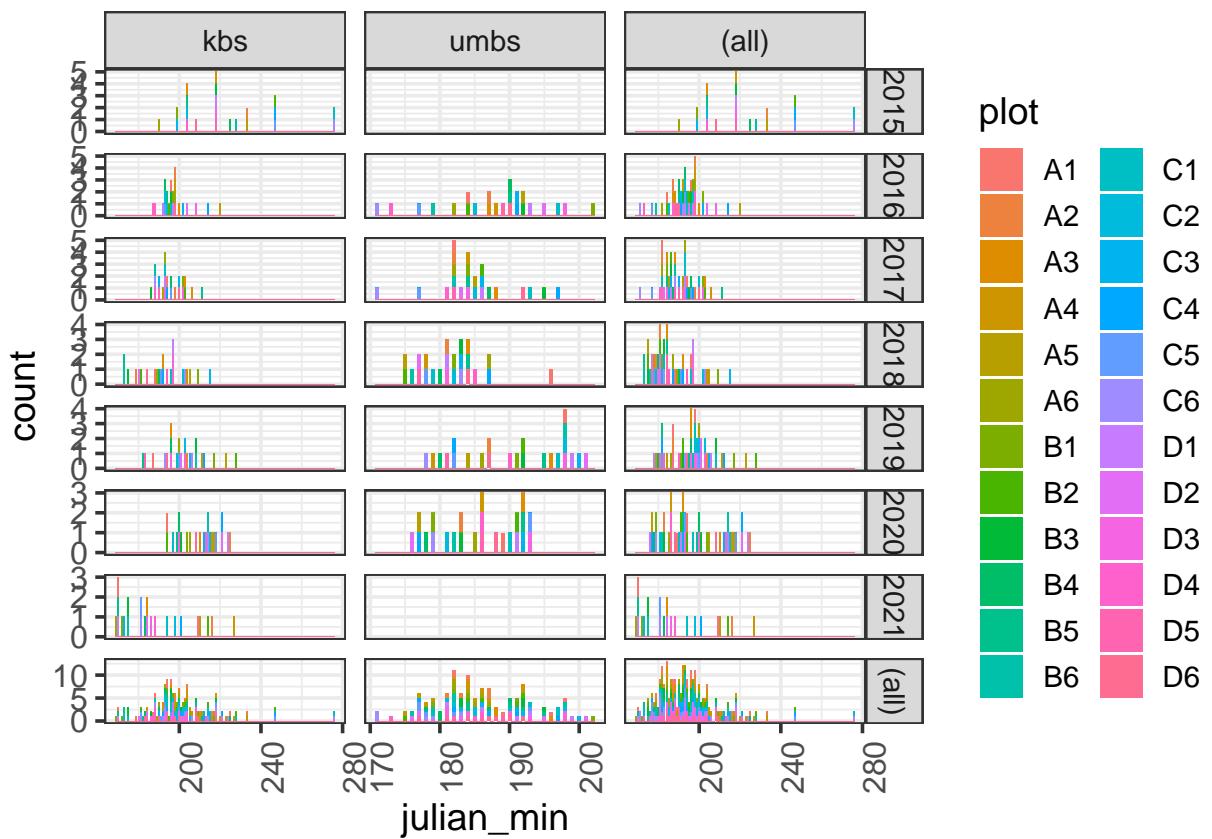
```

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

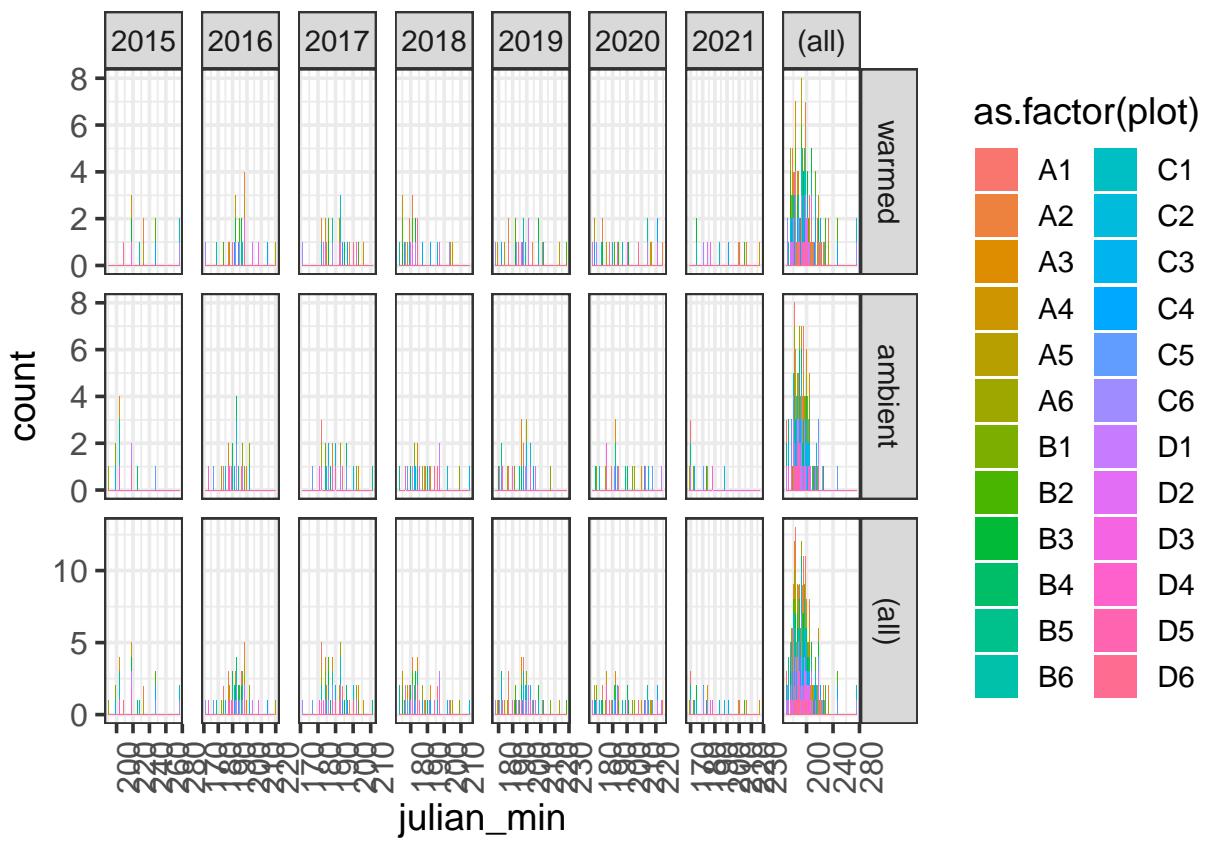
```

# 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")

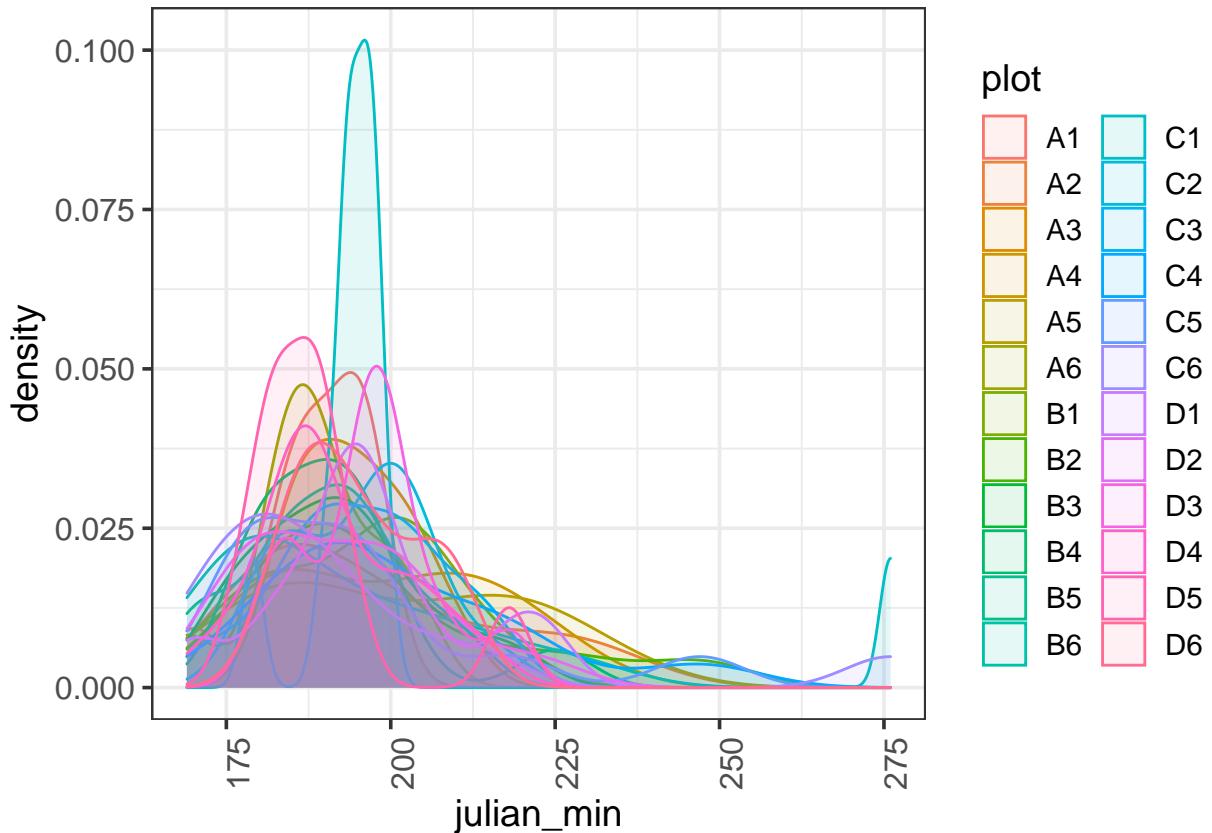
```



```
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: 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: 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
```

```
## 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
```

```
## 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

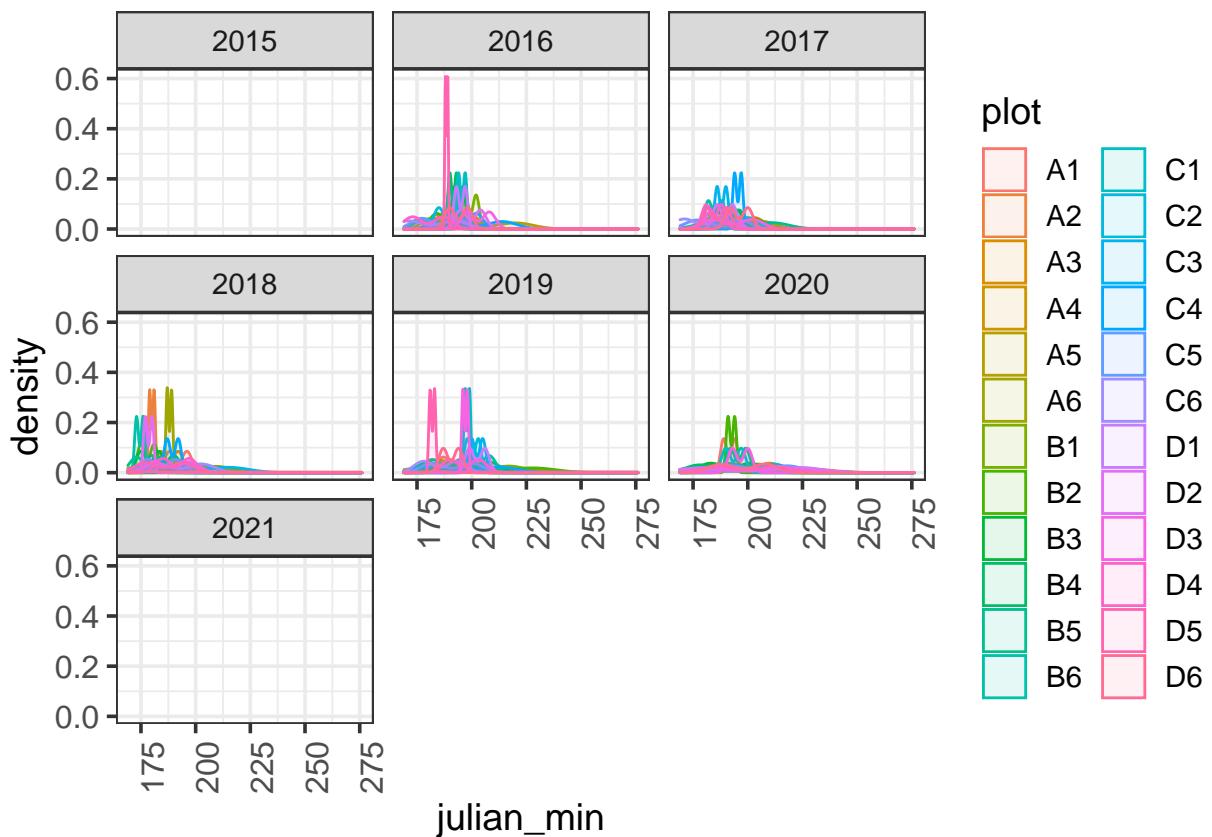
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## 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: 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: 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

## 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

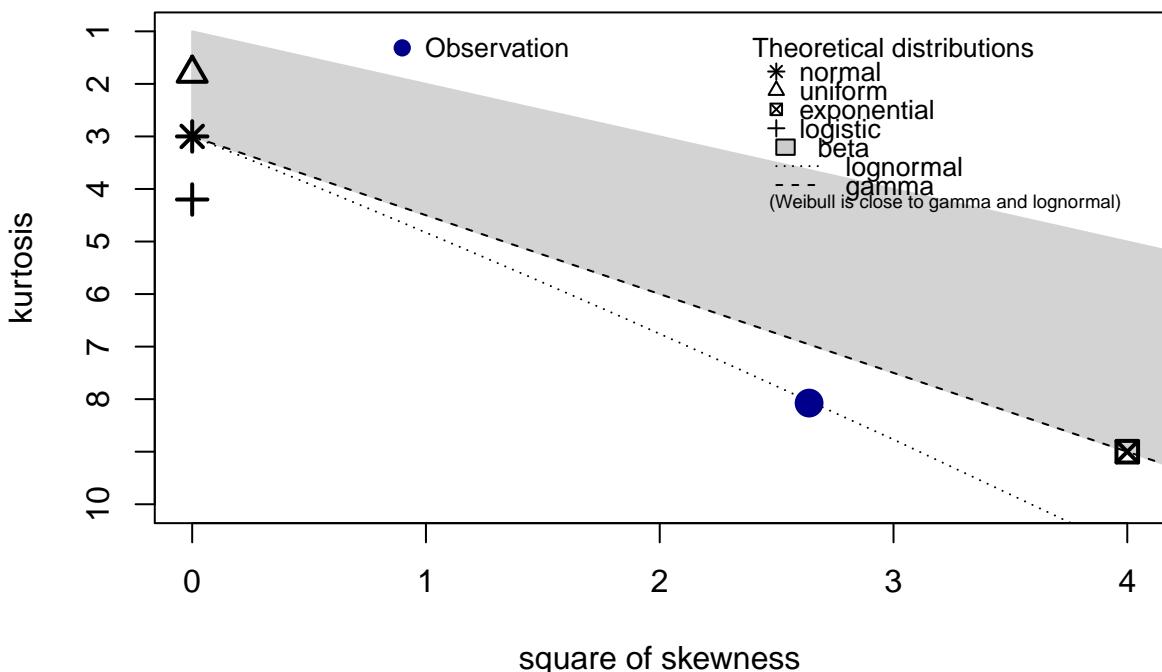
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## 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
```



Cullen and Frey graph

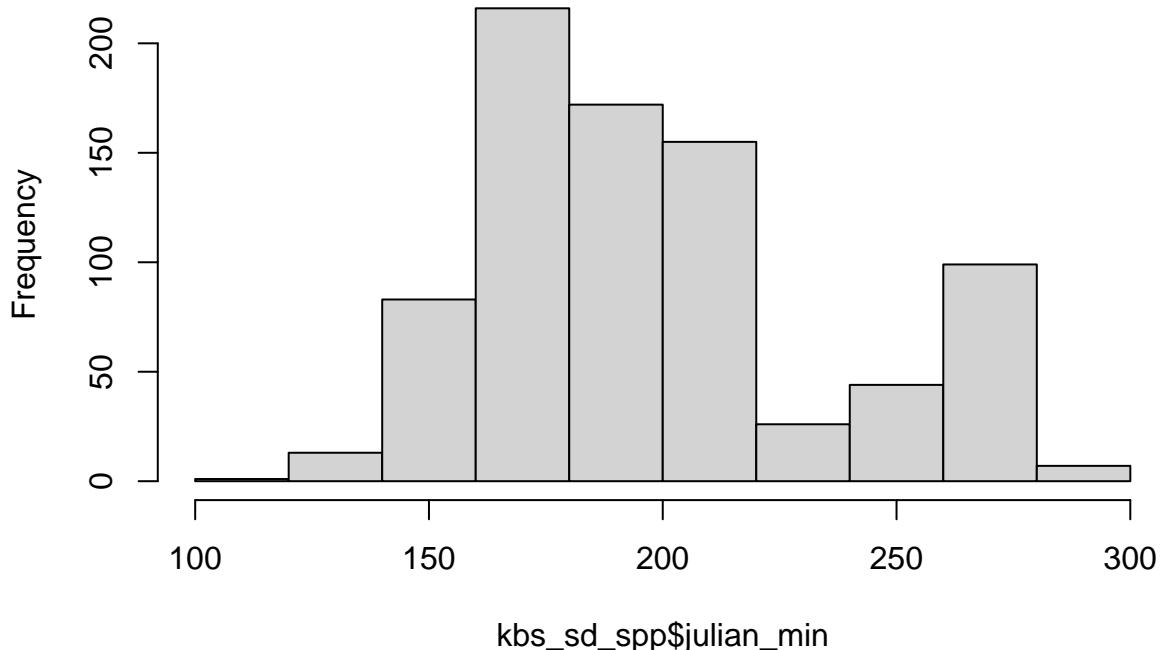


```
## summary statistics
## -----
## min: 169   max: 276
## median: 193
## mean: 194.7183
## estimated sd: 15.60958
## estimated skewness: 1.624461
## estimated kurtosis: 8.074794
```

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

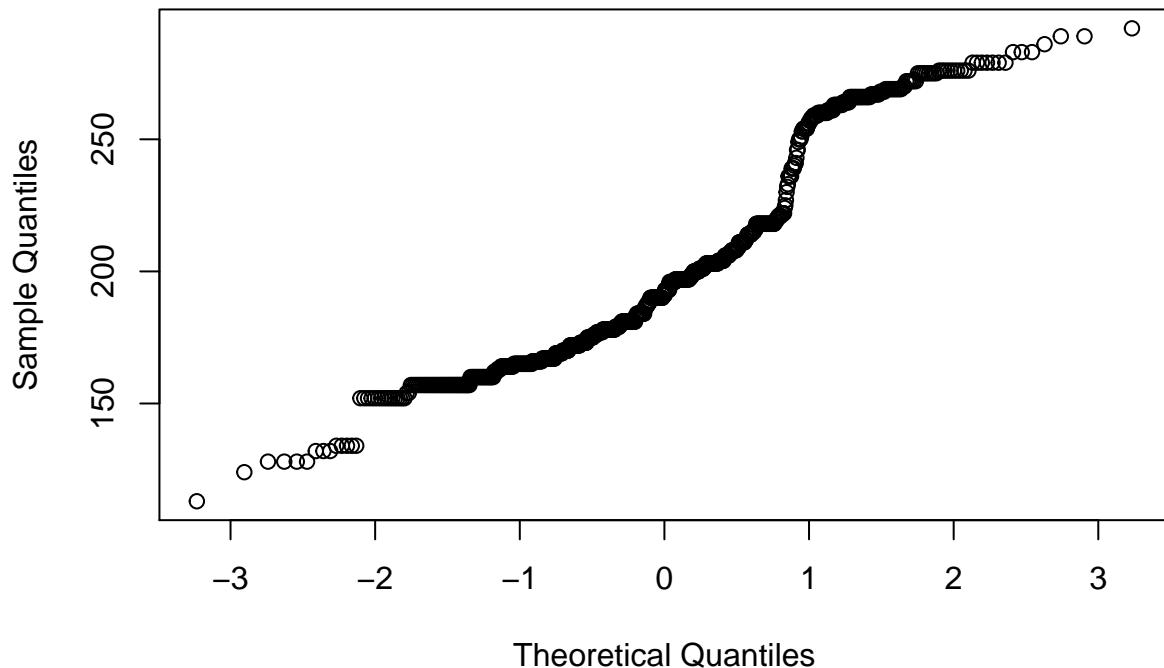
```
### KBS ####
hist(kbs_sd_spp$julian_min)
```

Histogram of kbs_sd_spp\$julian_min



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

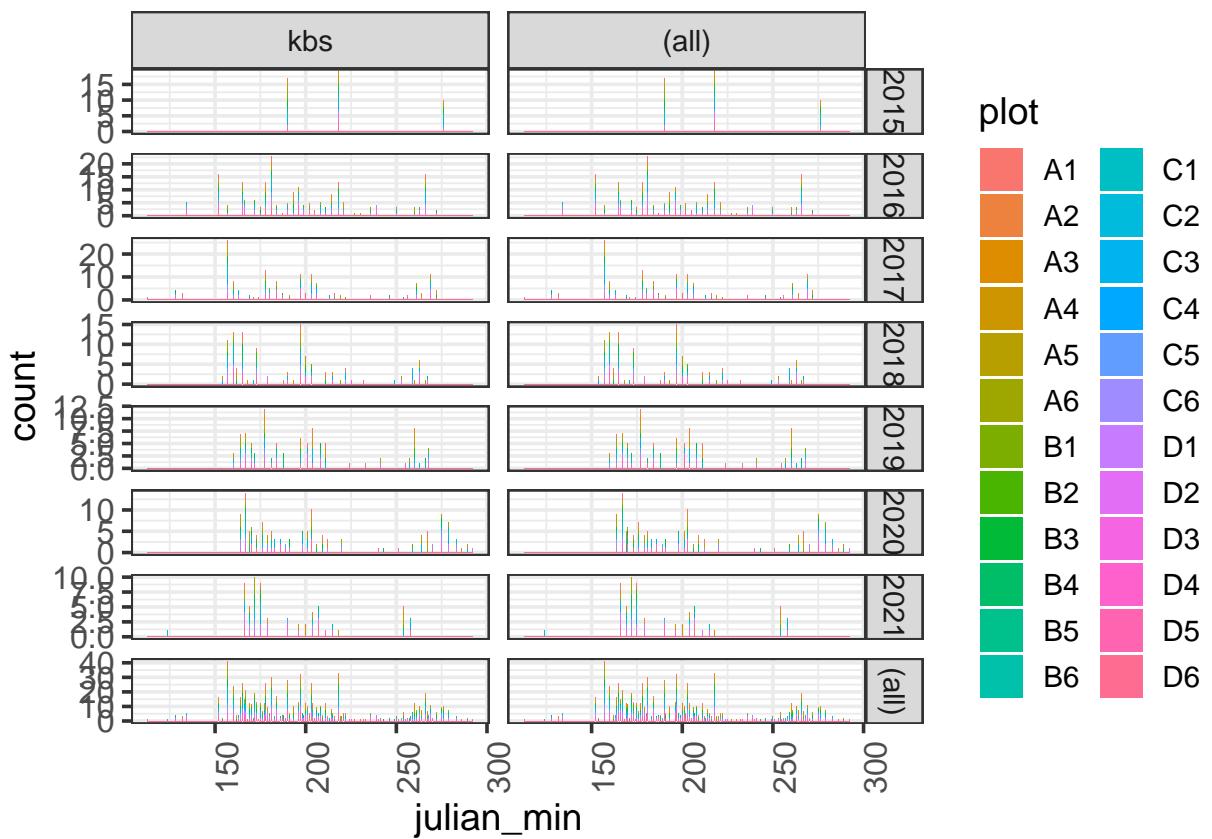
Normal Q-Q Plot



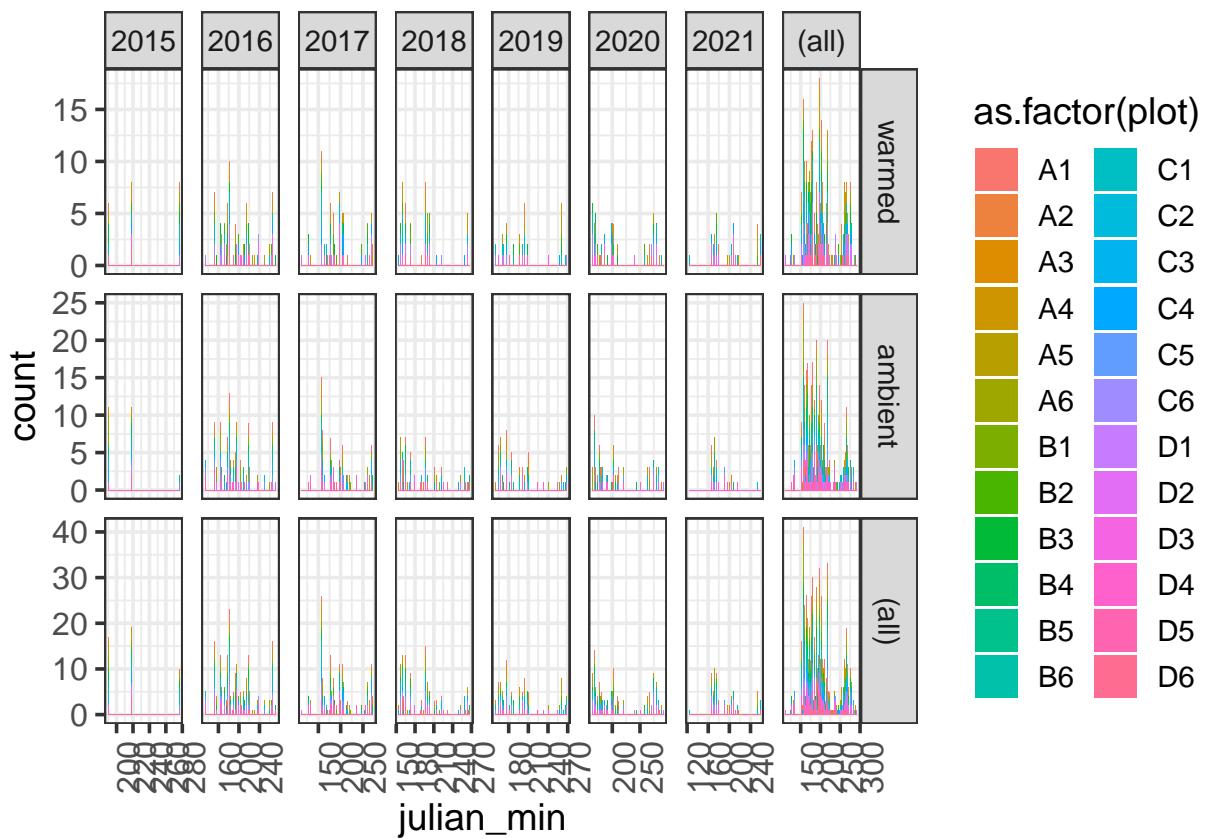
```
shapiro.test(kbs_sd_spp$julian_min) # pvalue is < 0.05 so we reject the null hypothesis that the data

##
## Shapiro-Wilk normality test
##
## data: kbs_sd_spp$julian_min
## W = 0.92015, p-value < 2.2e-16

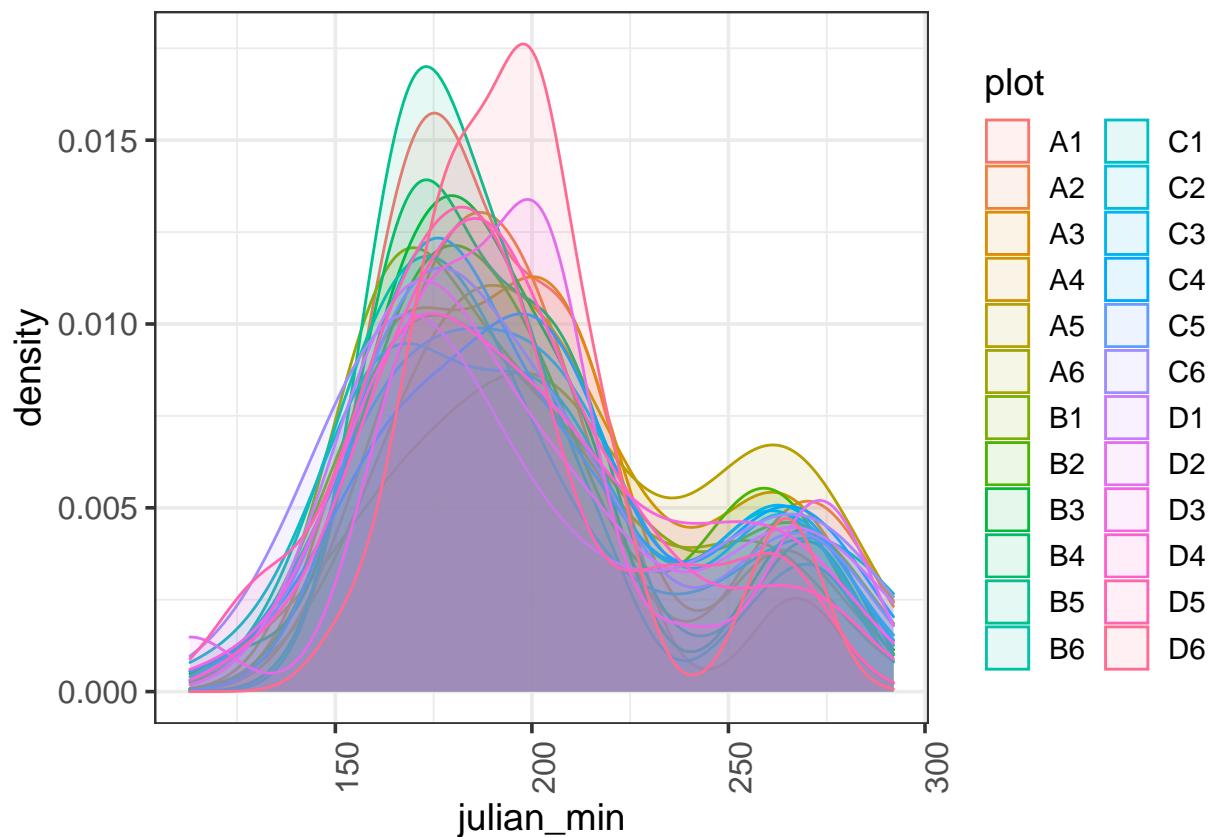
# Visualizing first Julian date for kbs at the SPECIES LEVEL
ggplot(kbs_sd_spp, aes(julian_min, fill = plot)) + geom_histogram(binwidth = 0.5) +
  facet_grid(year ~ site, margins = TRUE, scales = "free")
```



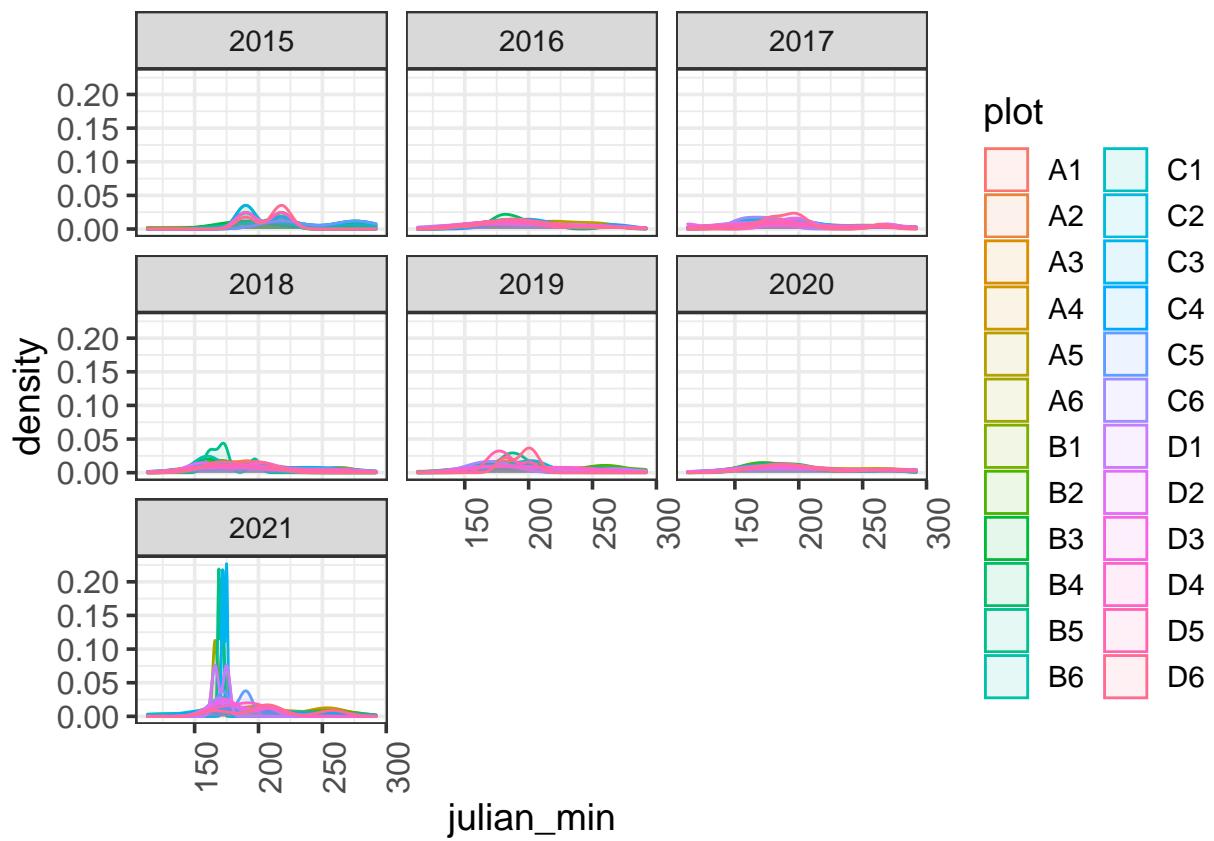
```
ggplot(kbs_sd_spp, aes(julian_min, fill = as.factor(plot))) + geom_histogram(binwidth = 0.5) +
  facet_grid(state ~ year, margins = TRUE, scales = "free")
```



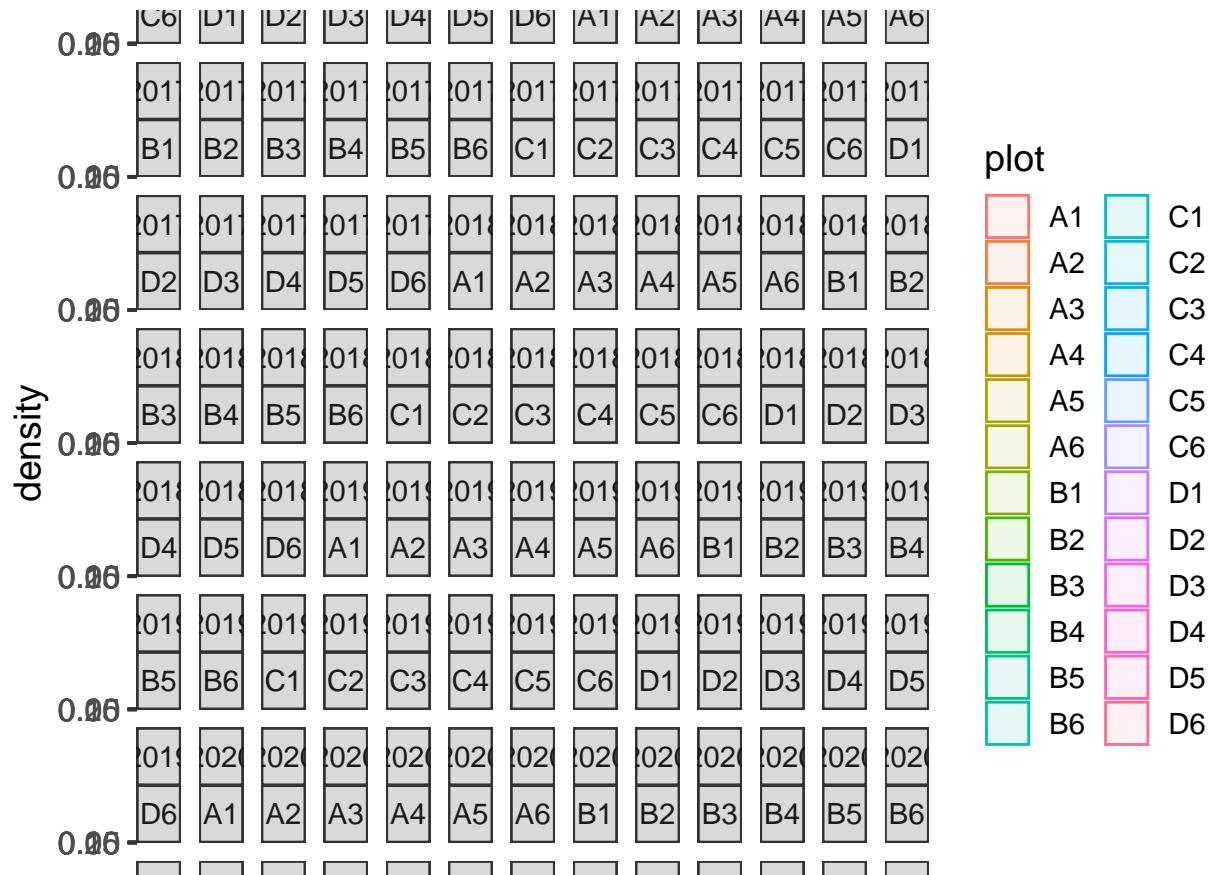
```
ggplot(kbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

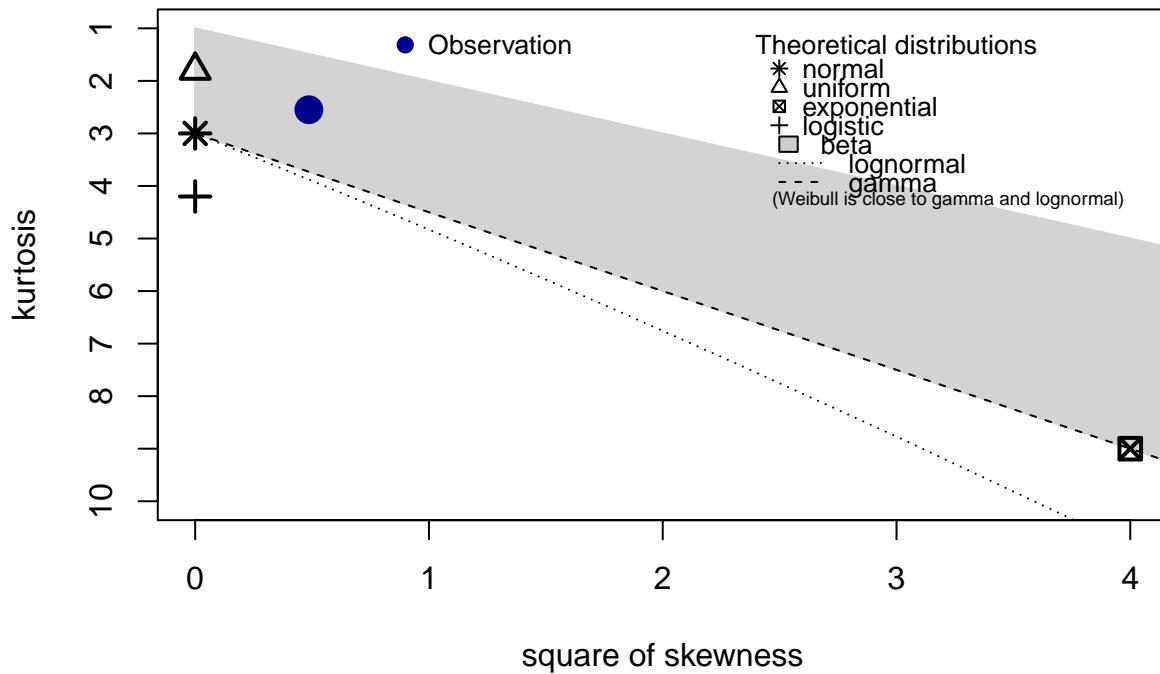


```
ggplot(kbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



```
# Exploring distributions
descdist(kbs_sd_spp$julian_min, discrete = FALSE)
```

Cullen and Frey graph

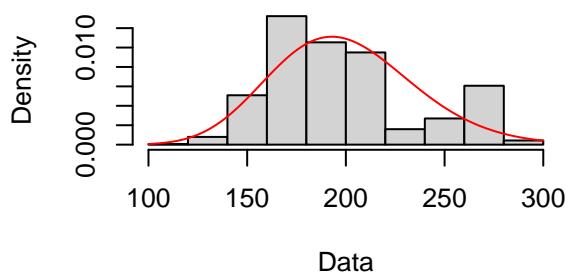
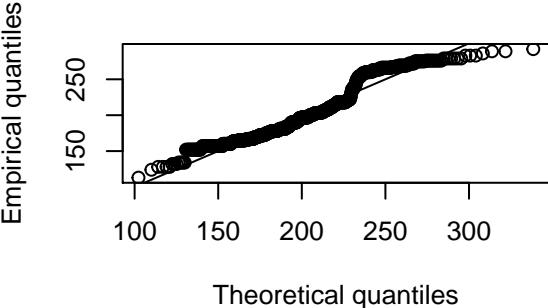
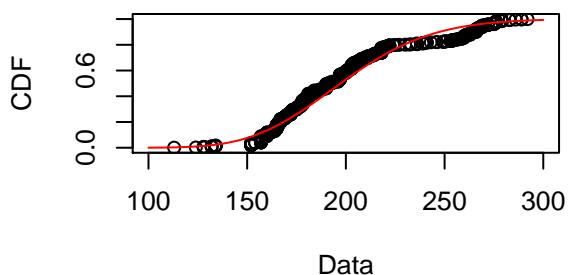
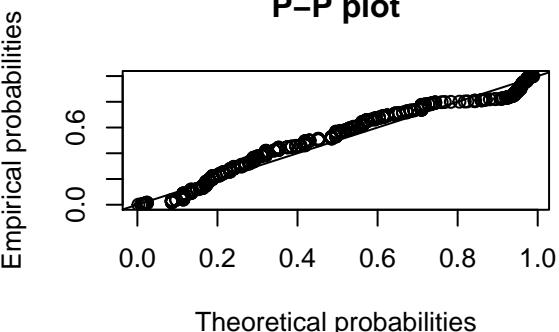


```

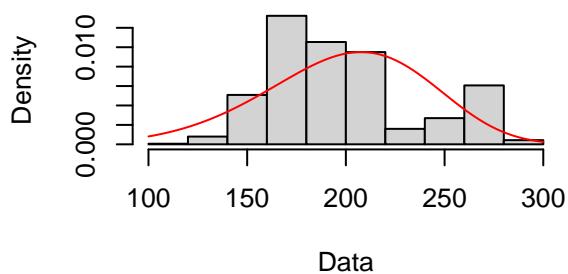
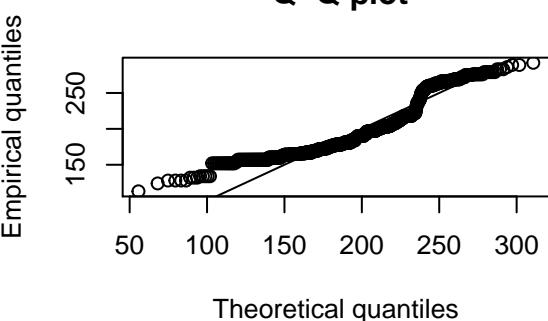
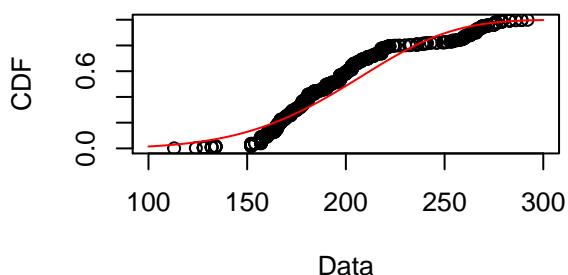
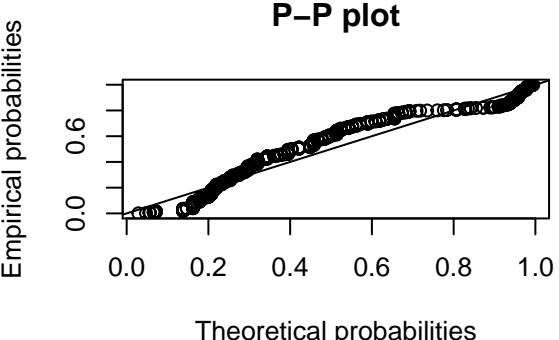
## summary statistics
## -----
## min: 113  max: 292
## median: 191
## mean: 199.5539
## estimated sd: 37.41916
## estimated skewness: 0.697435
## estimated kurtosis: 2.549031

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

```

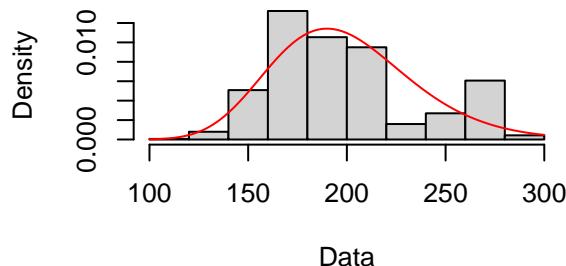
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

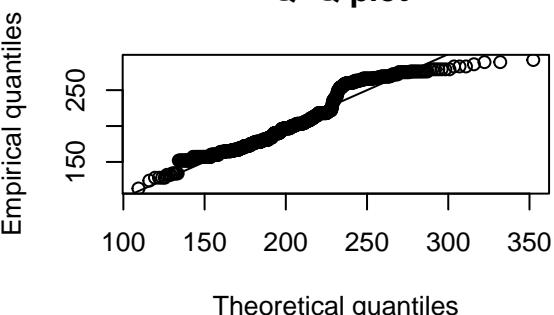
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

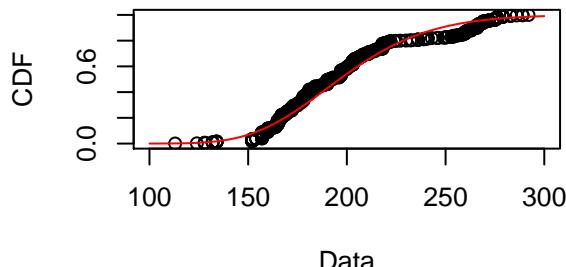
Empirical and theoretical dens.



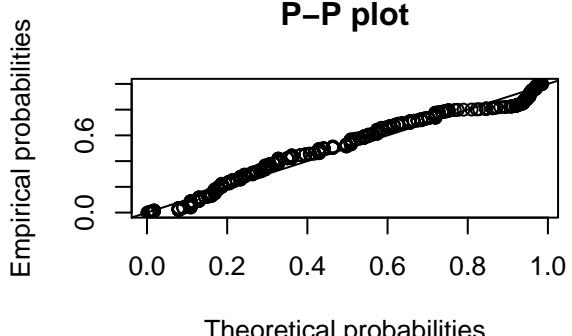
Q–Q plot



Empirical and theoretical CDFs

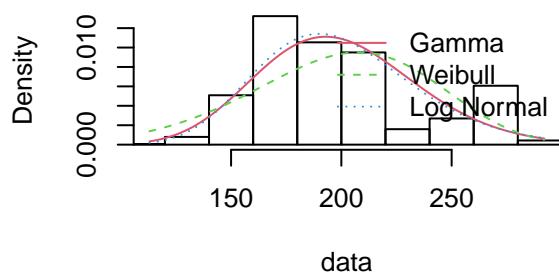


P–P plot

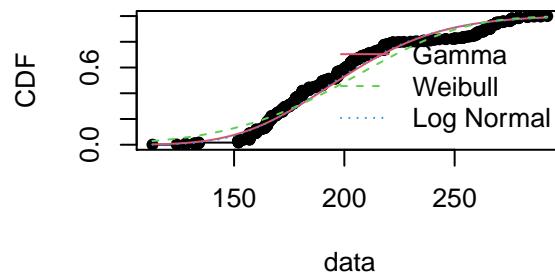


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

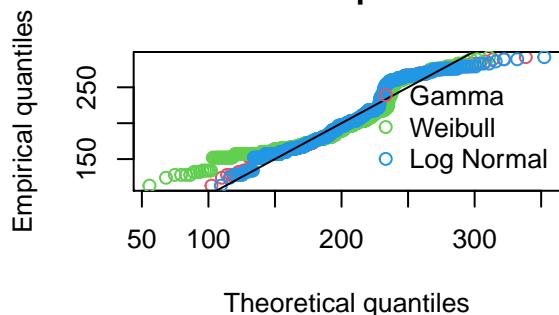
Histogram and theoretical densities



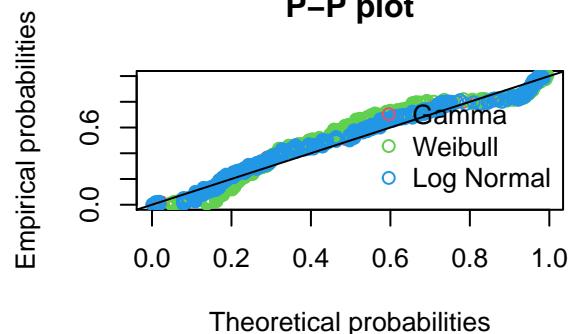
Empirical and theoretical CDFs



Q-Q plot



P-P plot



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

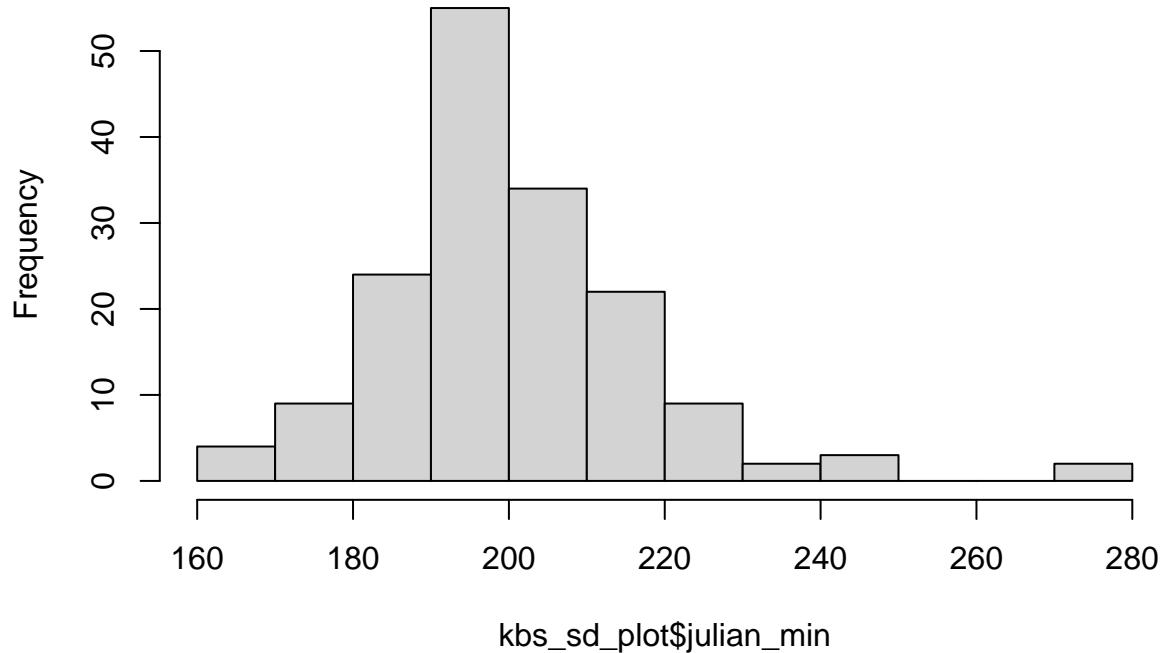
```
## Goodness-of-fit statistics
##                                     Gamma   Weibull  Log Normal
## Kolmogorov-Smirnov statistic 0.09938489 0.1359955 0.09279707
## Cramer-von Mises statistic   2.39387460 4.9973652 1.87659322
## Anderson-Darling statistic  16.96958844 30.7903177 13.94184575
##
## Goodness-of-fit criteria
##                                     Gamma   Weibull  Log Normal
## Akaike's Information Criterion 8167.610 8315.031 8146.322
## Bayesian Information Criterion 8177.019 8324.440 8155.731
```

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

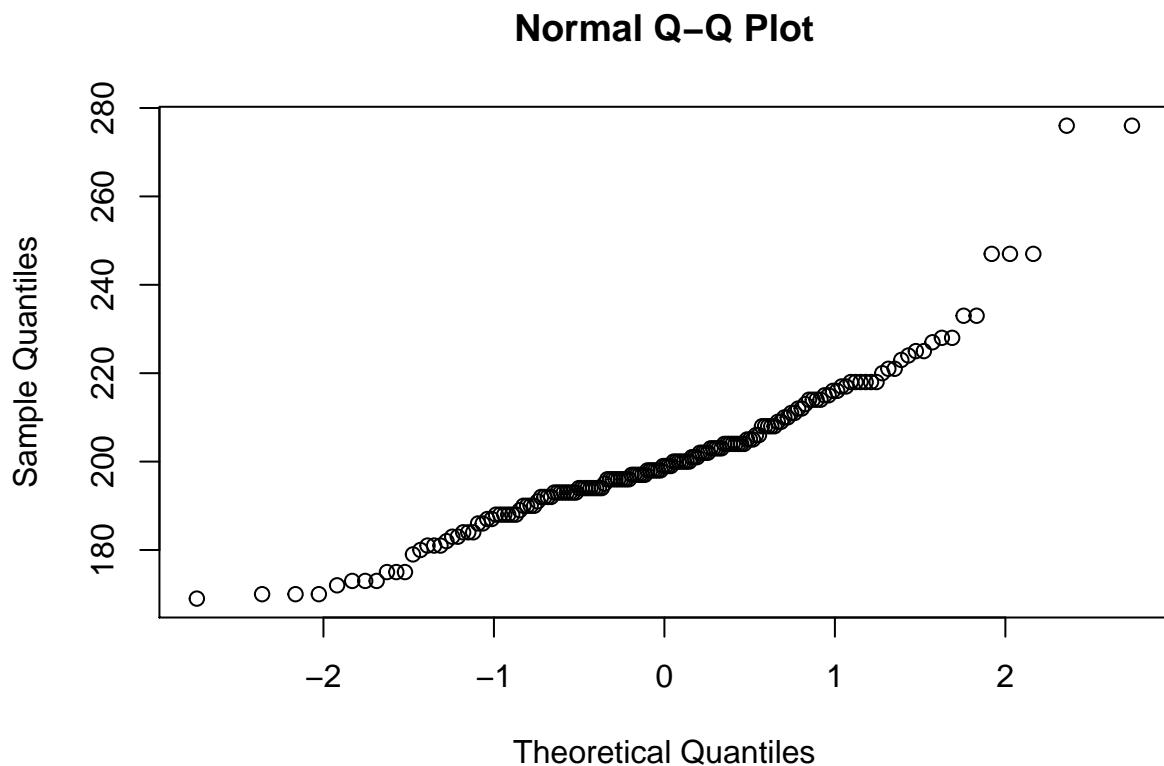
KBS PLOT LEVEL - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

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

Histogram of kbs_sd_plot\$julian_min



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



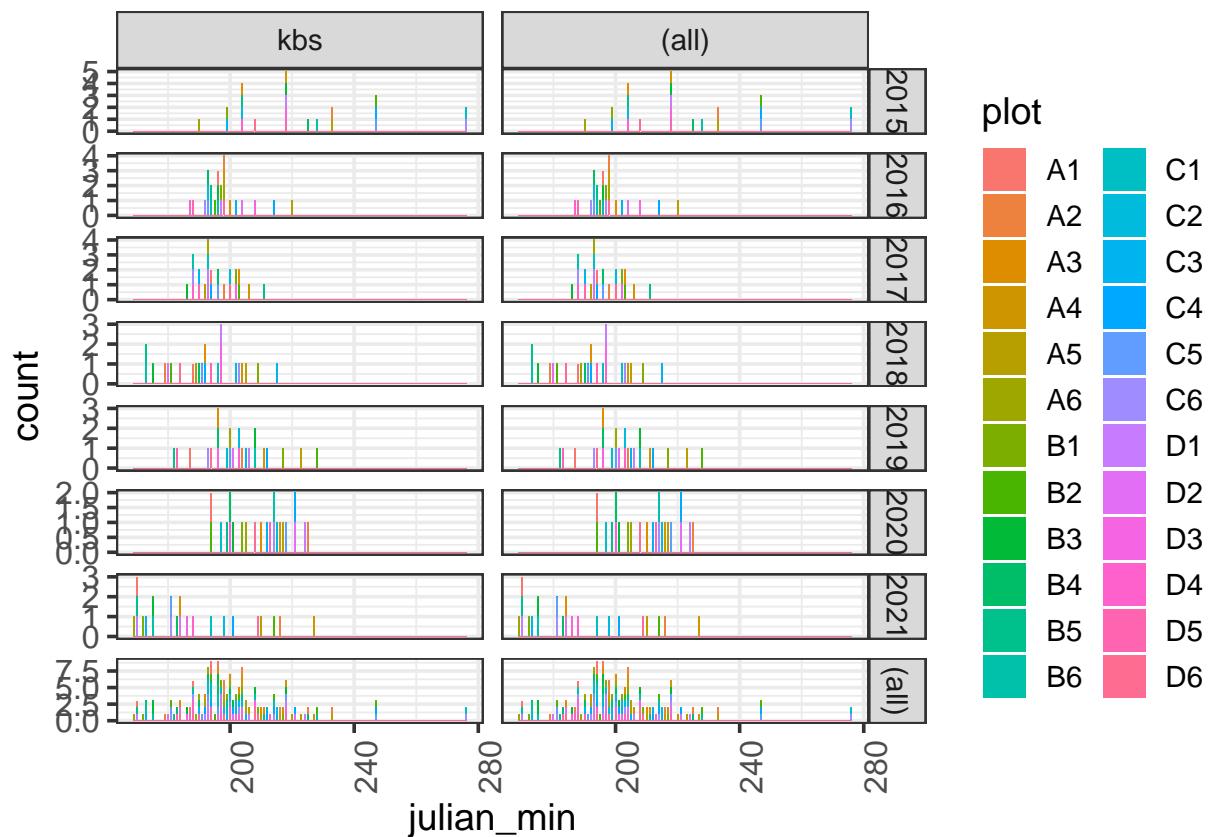
```

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 = 0.92262, p-value = 1.11e-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")

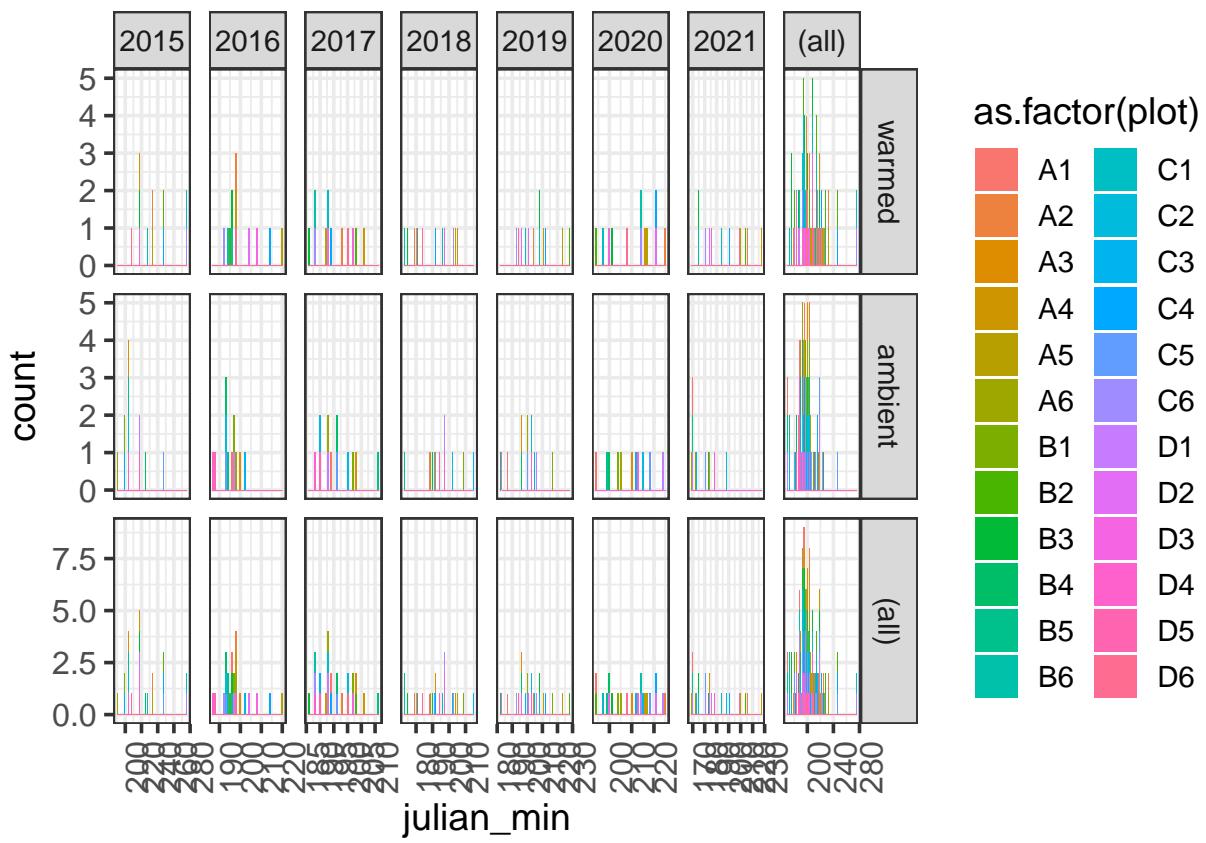
```



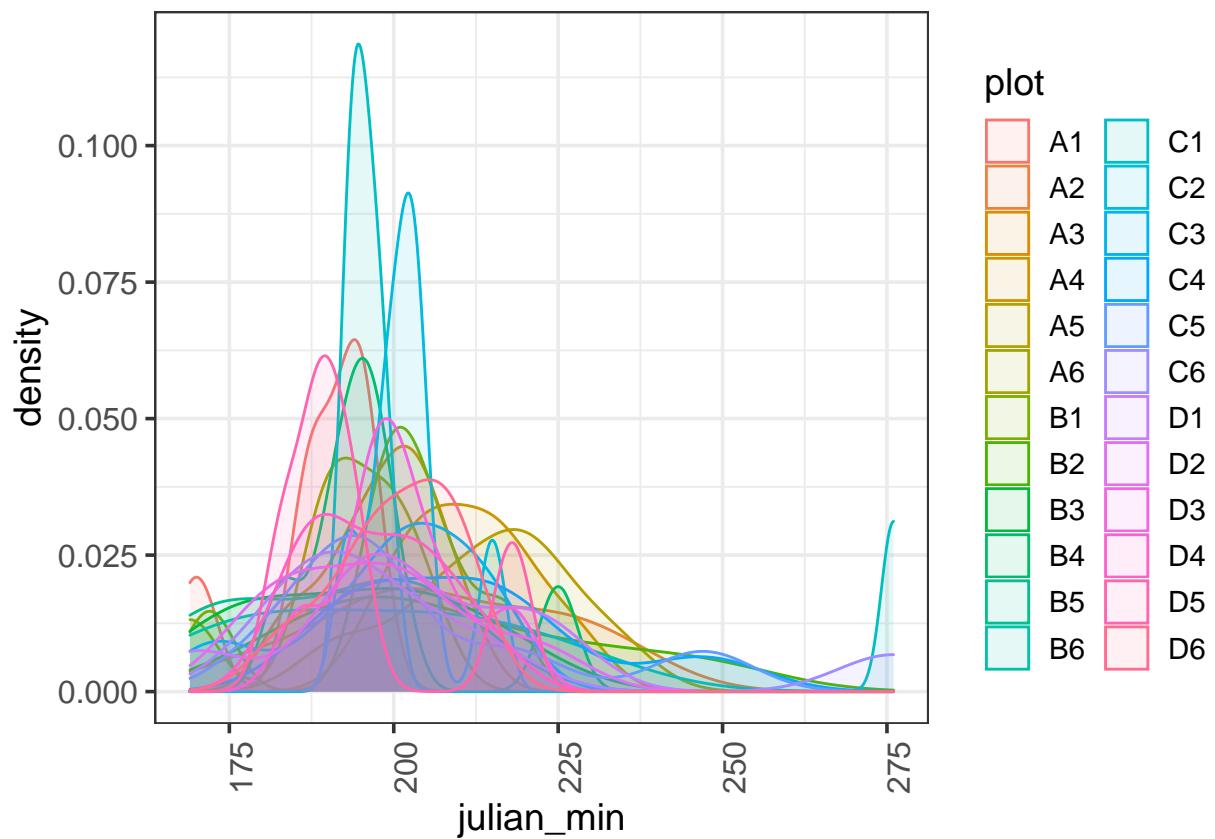
```

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

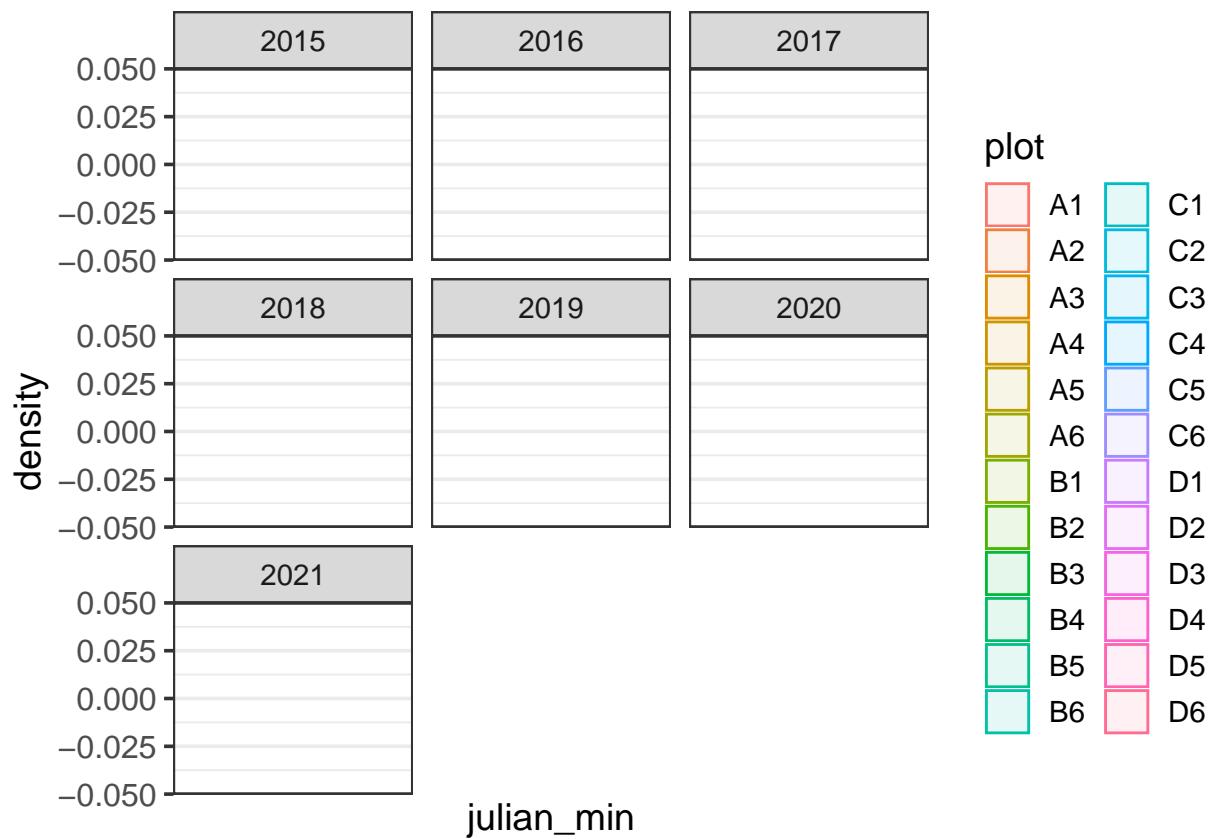
```



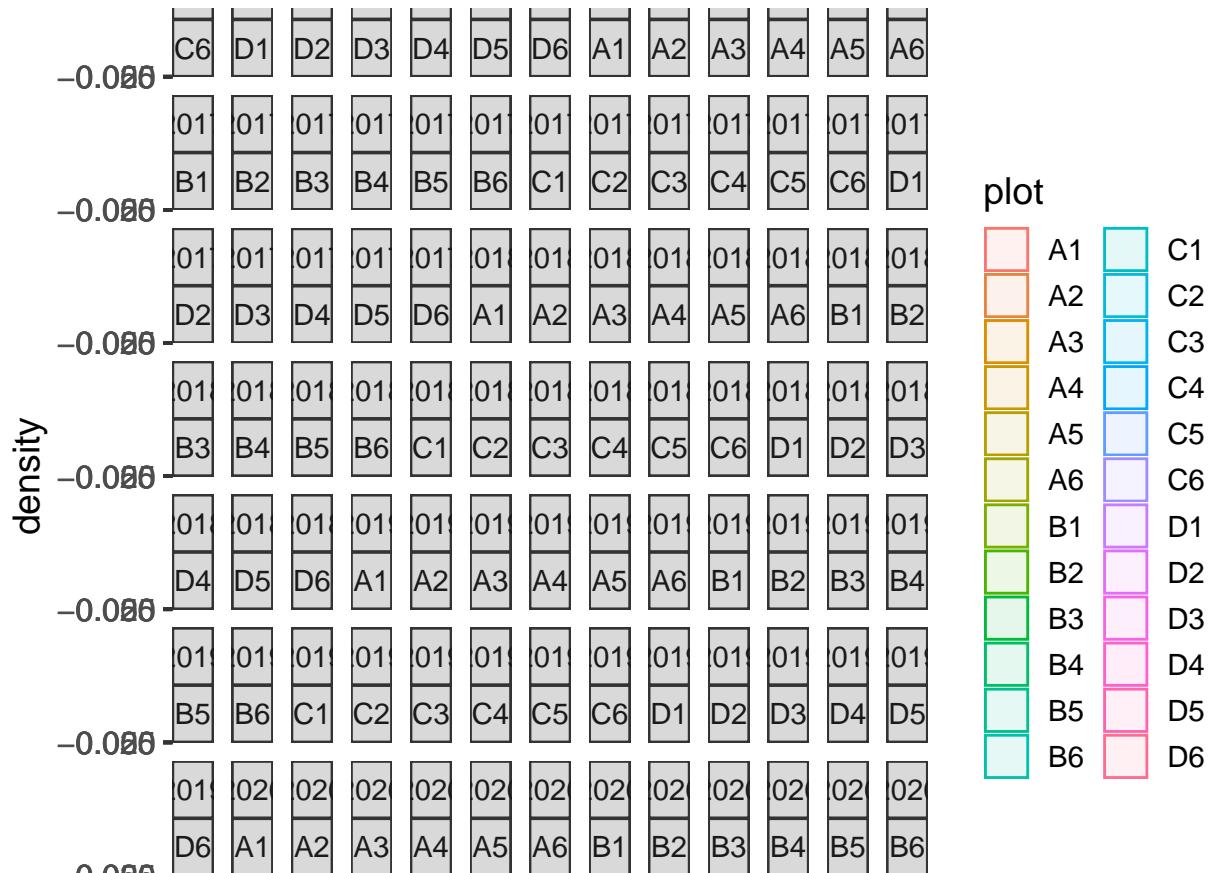
```
ggplot(kbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

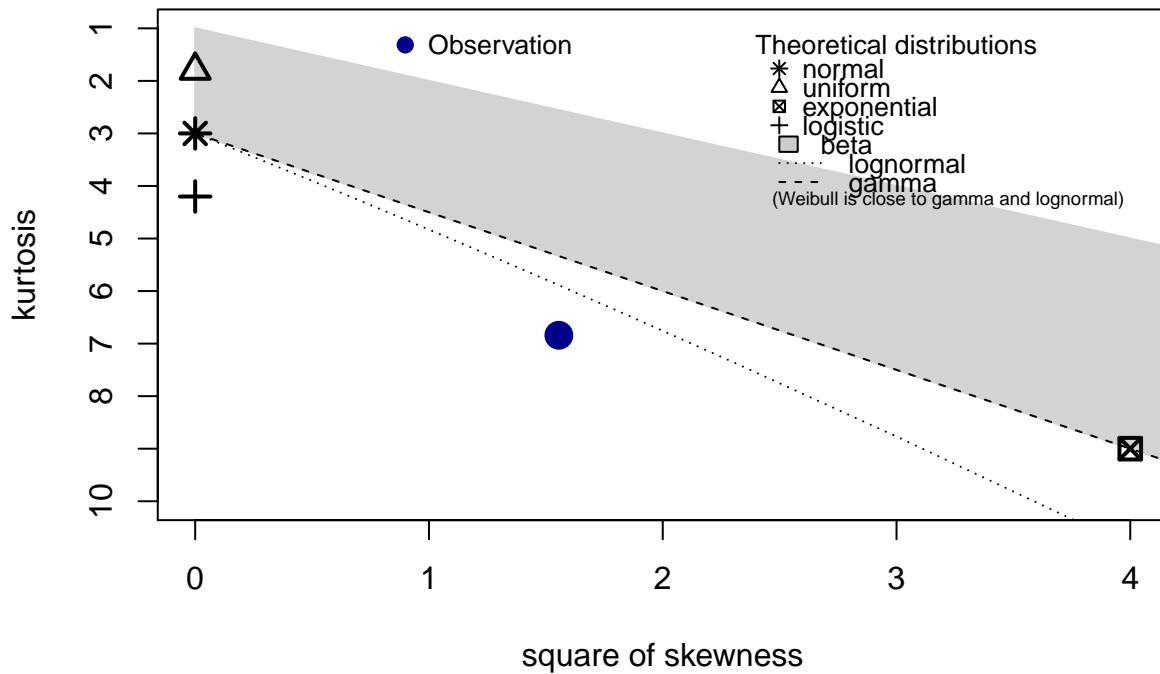


```
ggplot(kbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



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

Cullen and Frey graph

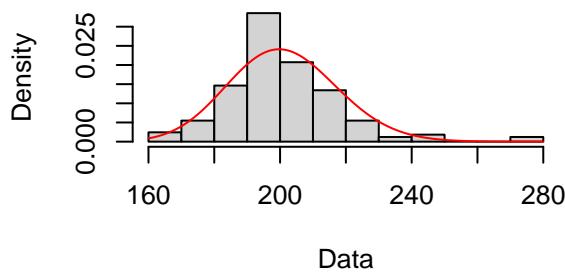
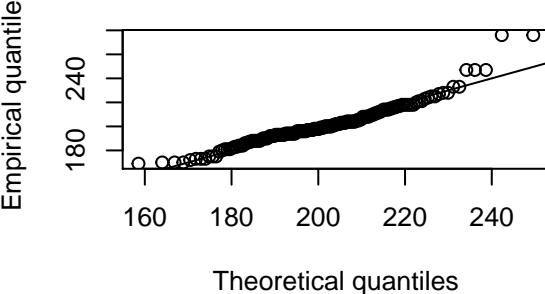
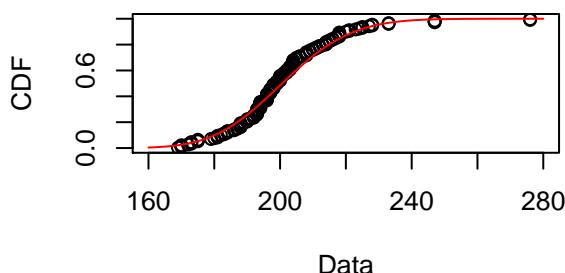
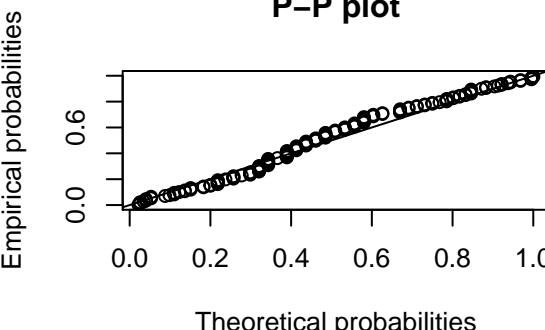


```

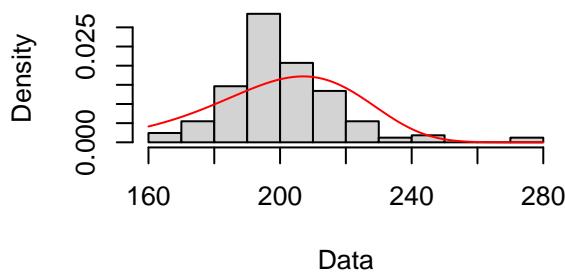
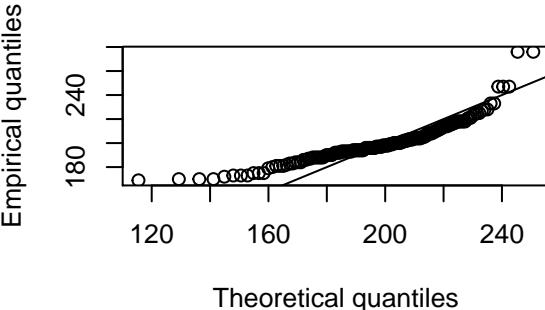
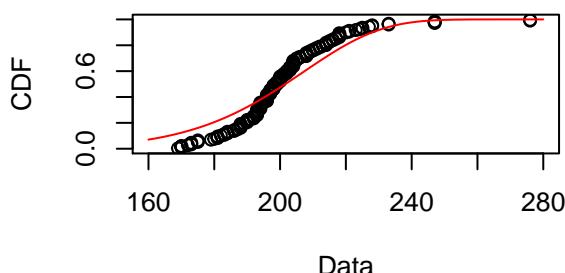
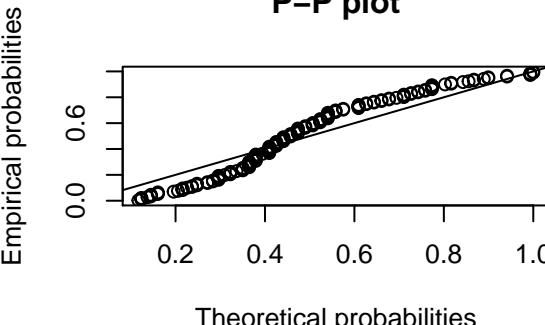
## summary statistics
## -----
## min: 169   max: 276
## median: 199
## mean: 201.0793
## estimated sd: 17.09016
## estimated skewness: 1.247272
## estimated kurtosis: 6.841134

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

```

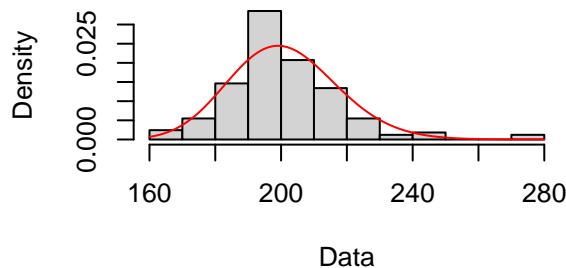
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

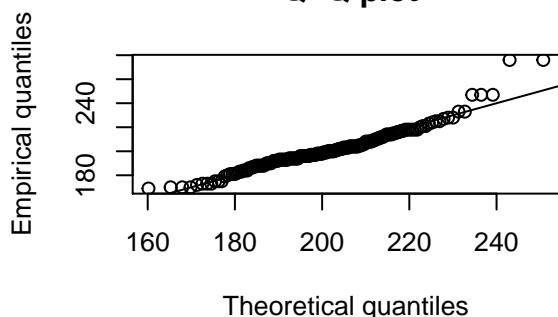
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

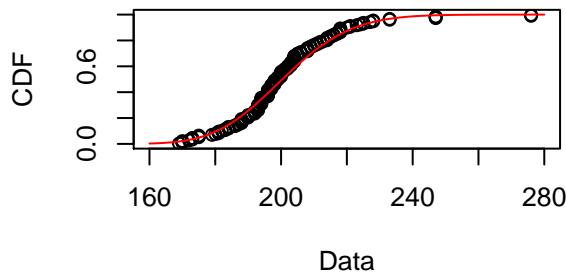
Empirical and theoretical dens.



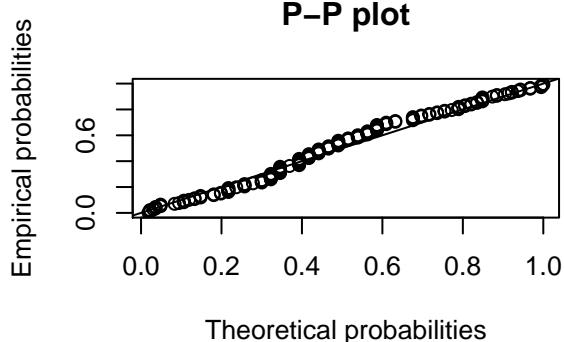
Q-Q plot



Empirical and theoretical CDFs

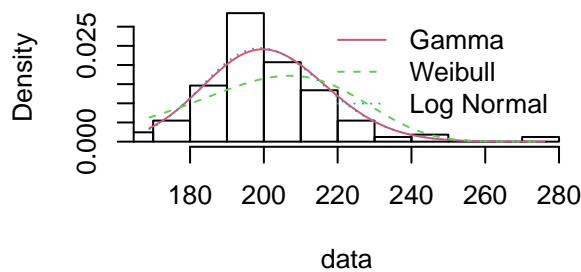


P-P plot

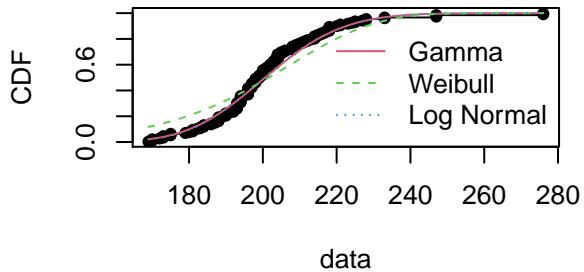


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

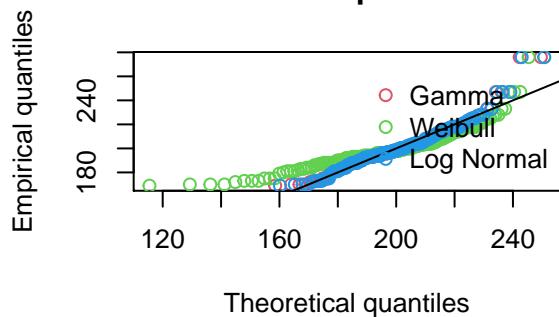
Histogram and theoretical densities



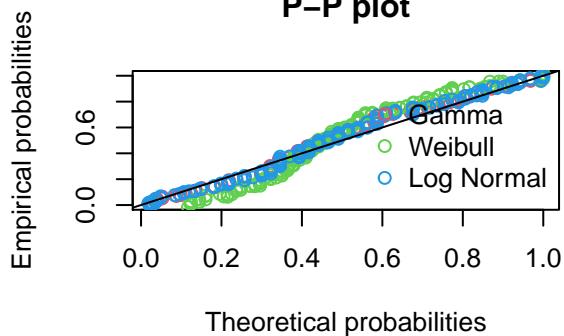
Empirical and theoretical CDFs



Q-Q plot



P-P plot



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

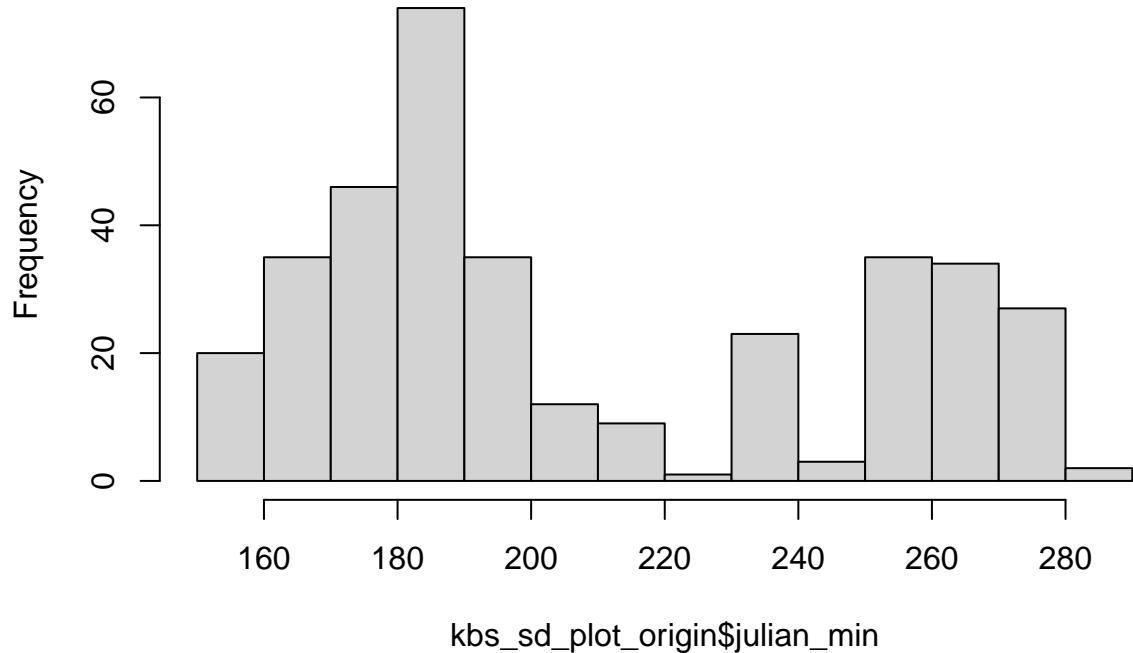
```
## Goodness-of-fit statistics
##                               Gamma   Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.1026051 0.1434858 0.09652636
## Cramer-von Mises statistic   0.3021108 1.4167964 0.26201156
## Anderson-Darling statistic   1.7308323 8.6247264 1.50269118
##
## Goodness-of-fit criteria
##                               Gamma   Weibull Log Normal
## Akaike's Information Criterion 1390.111 1463.355 1386.305
## Bayesian Information Criterion 1396.311 1469.555 1392.505
```

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

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

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

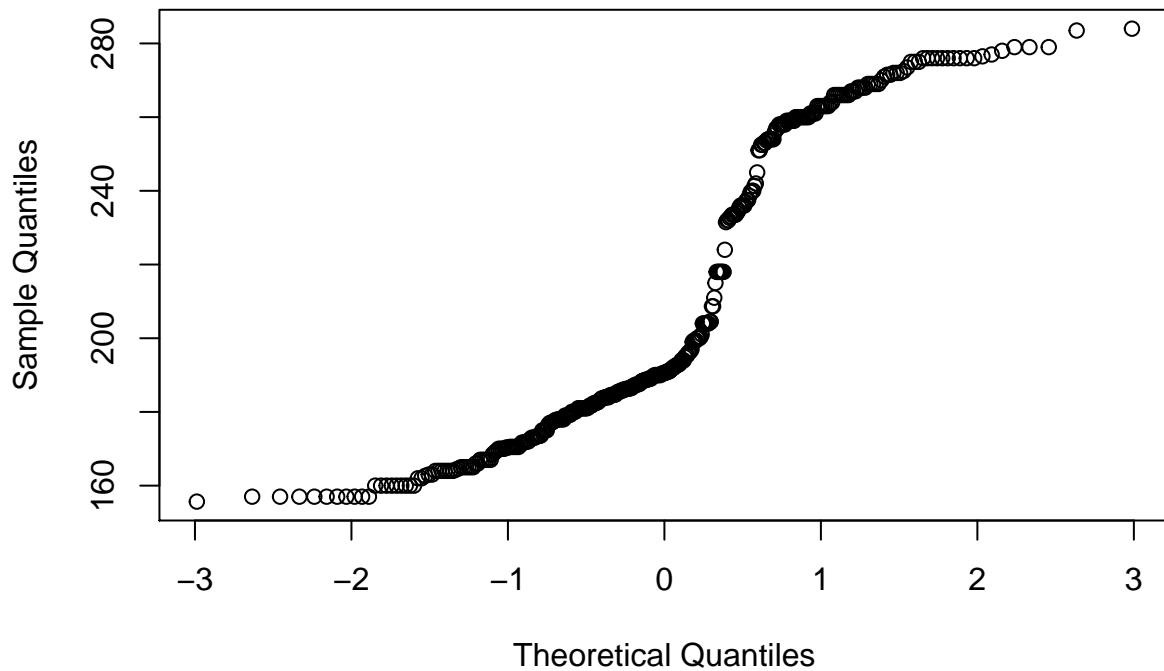
Histogram of kbs_sd_plot_origin\$julian_min



```
kbs_sd_plot_origin$julian_min
```

```
qqnorm(kbs_sd_plot_origin$julian_min)
```

Normal Q-Q Plot



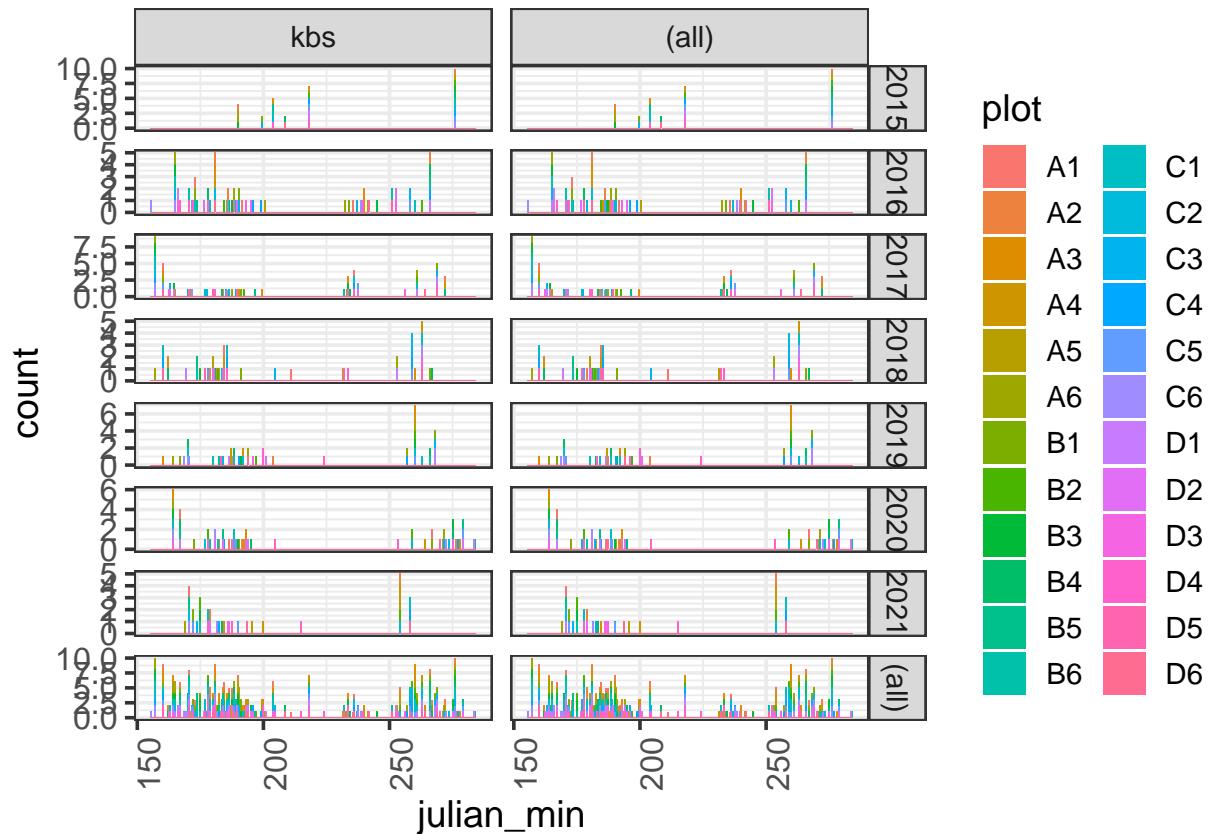
```

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

## 
## Shapiro-Wilk normality test
## 
## data: kbs_sd_plot_origin$julian_min
## W = 0.87102, p-value < 2.2e-16

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

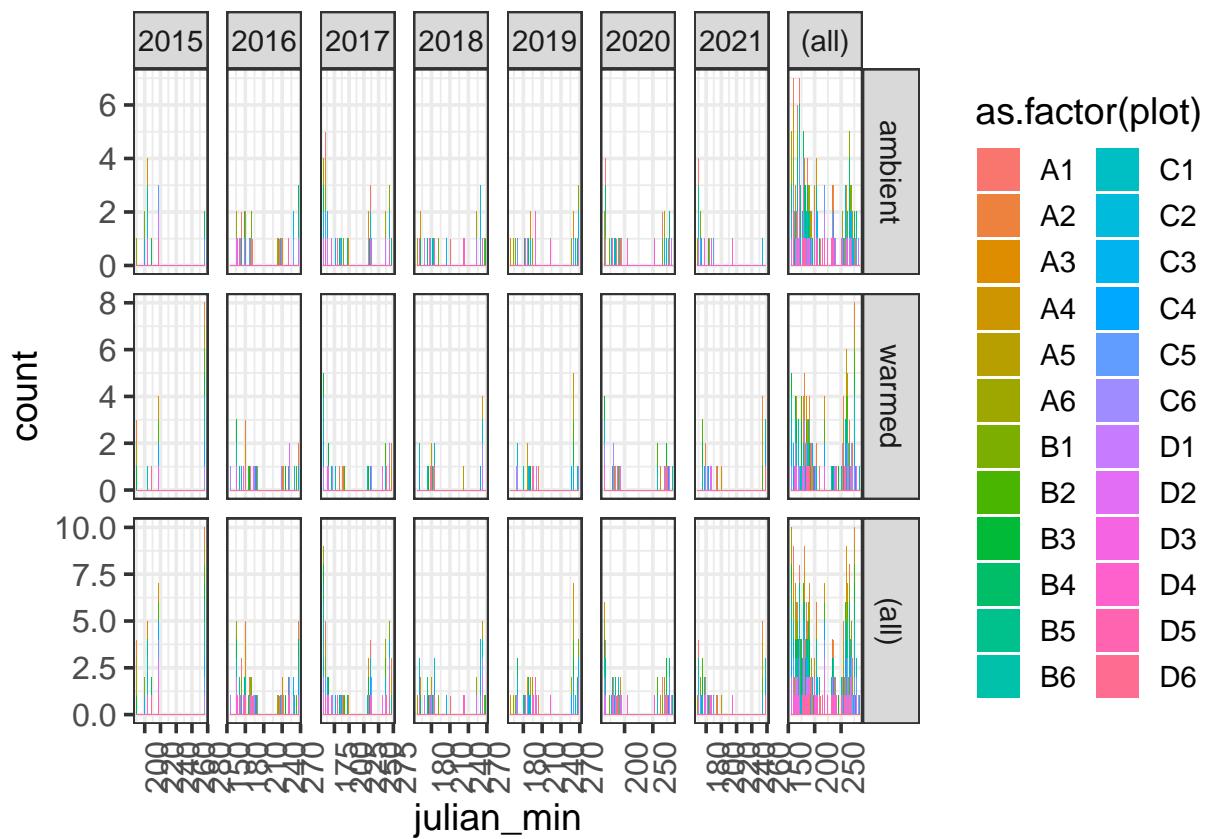
```



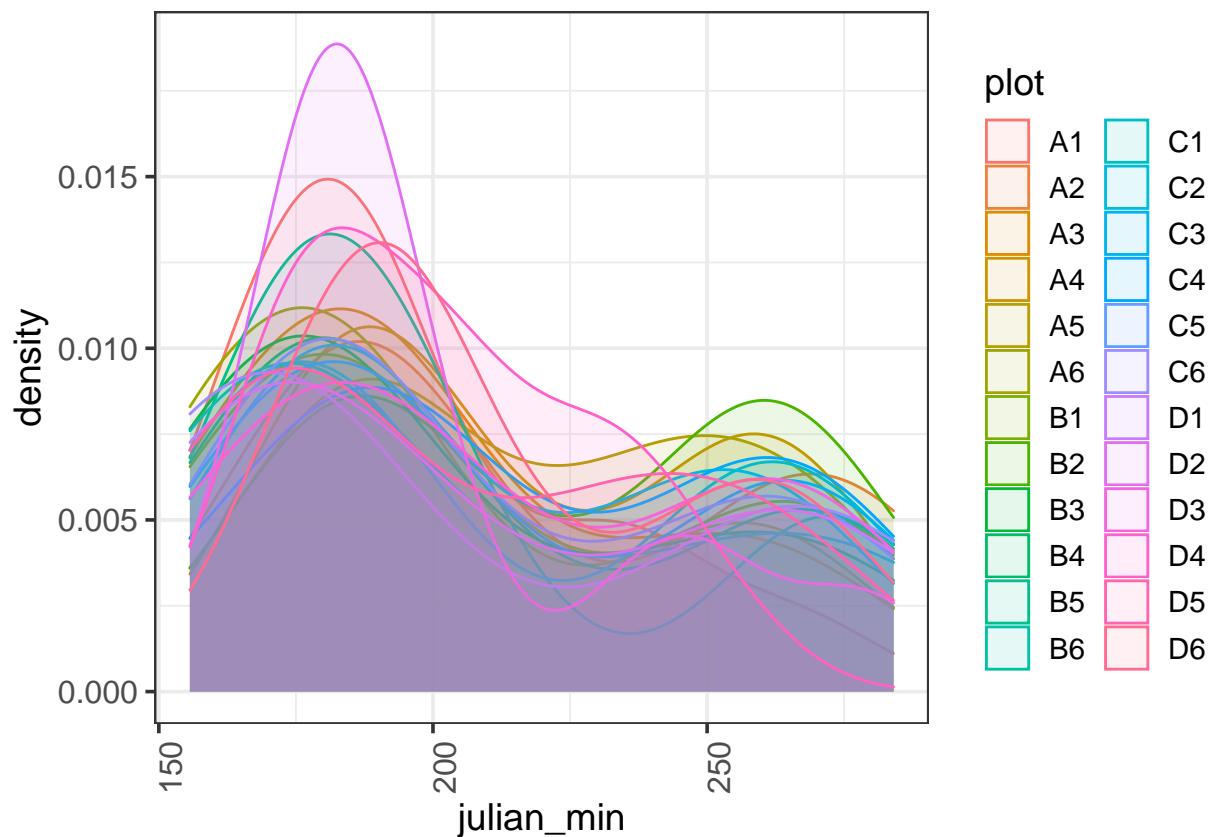
```

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

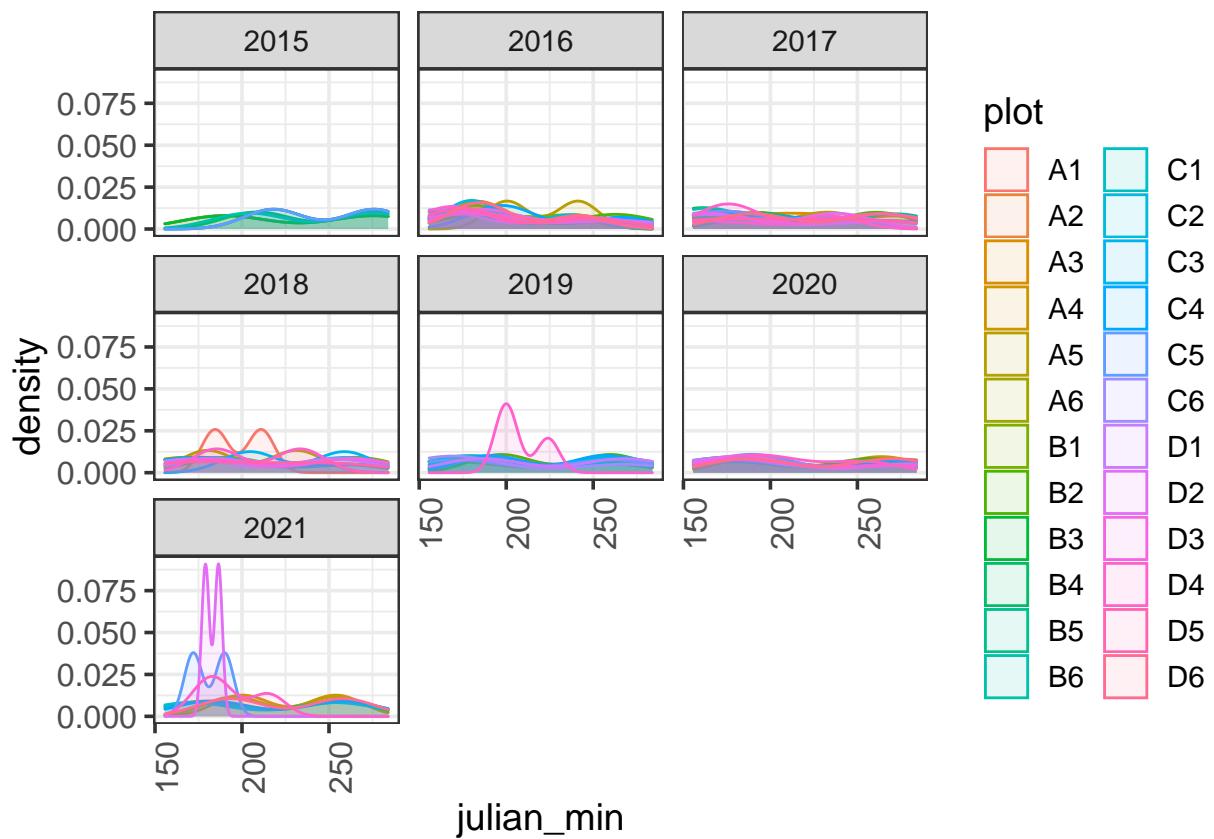
```



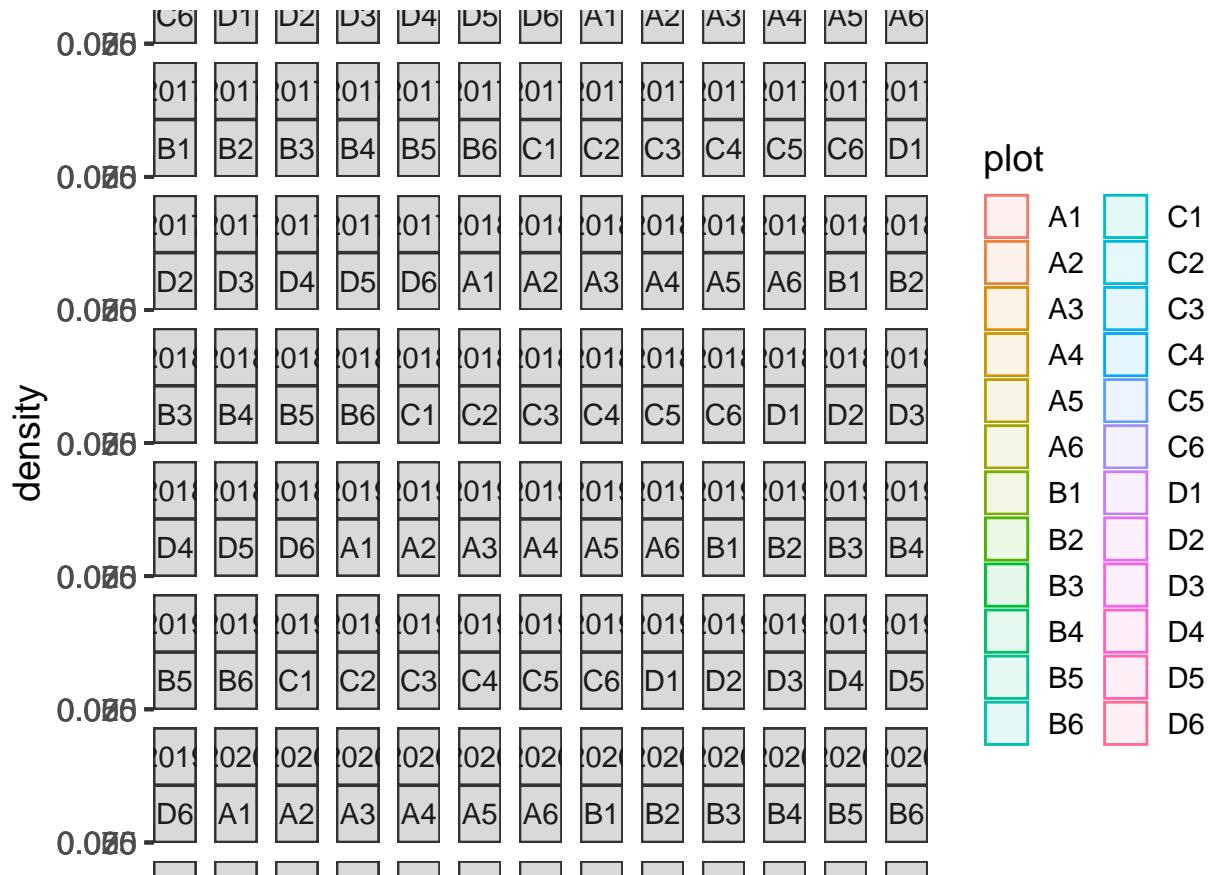
```
ggplot(kbs_sd_plot_origin, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

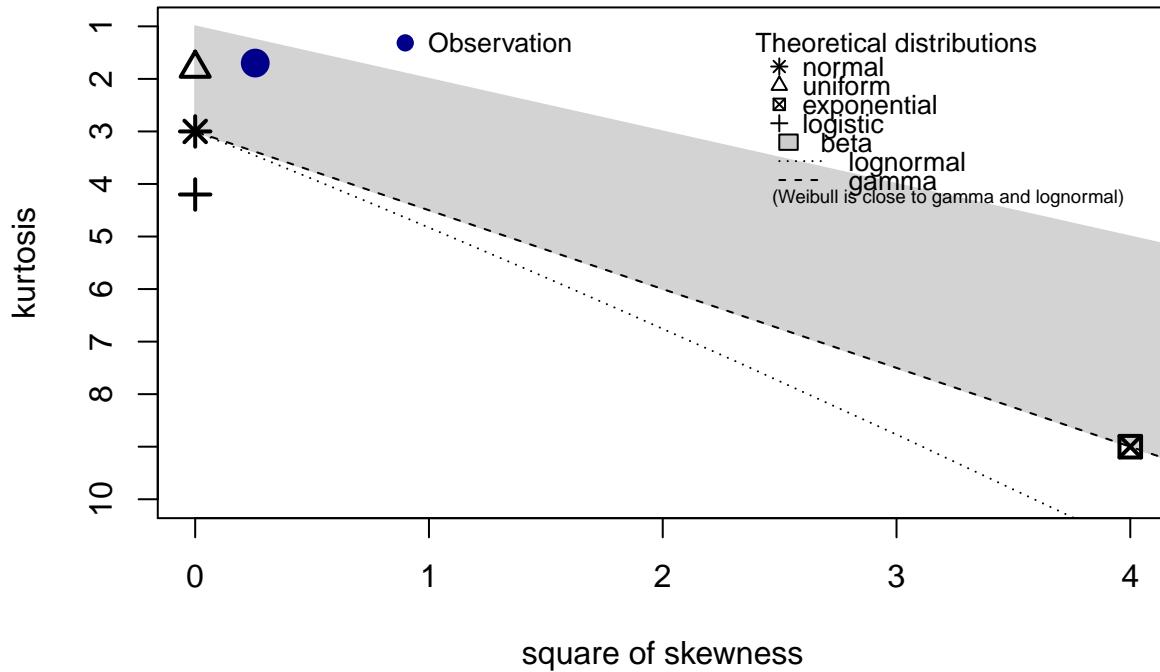


```
ggplot(kbs_sd_plot_origin, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



```
# Exploring distributions for these data:  
descdist(kbs_sd_plot_origin$julian_min, discrete = FALSE)
```

Cullen and Frey graph

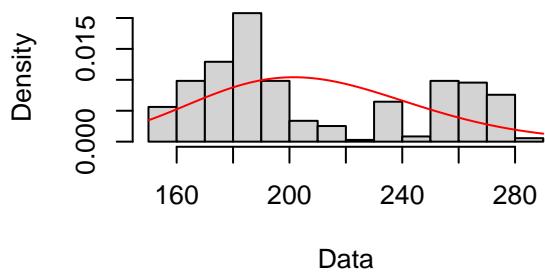
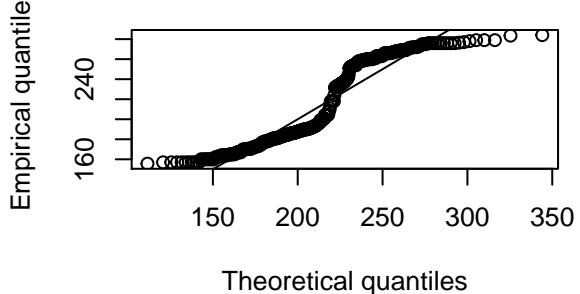
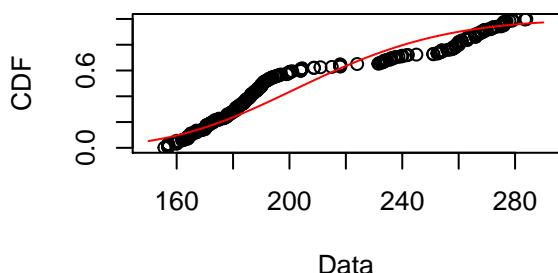
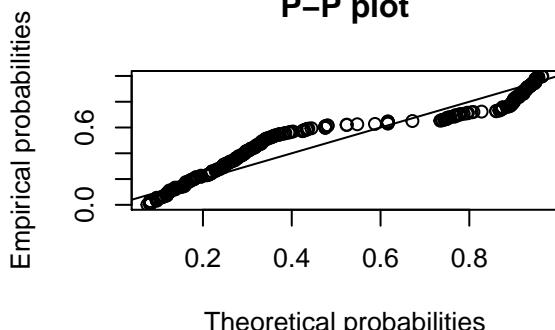


```

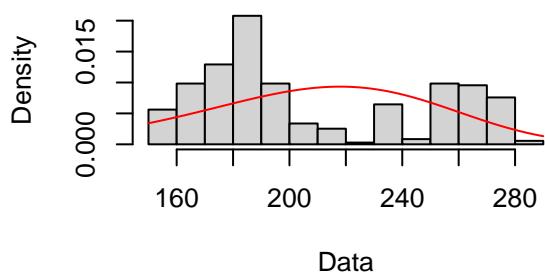
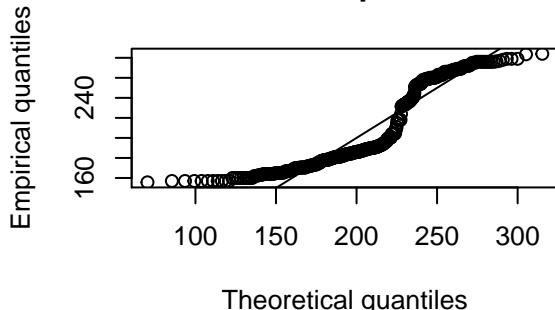
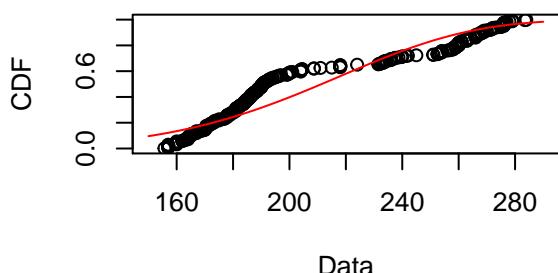
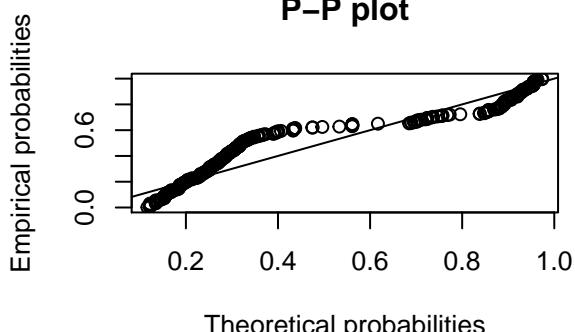
## summary statistics
## -----
## min: 155.6667  max: 284
## median: 190.5833
## mean: 208.7543
## estimated sd: 39.74692
## estimated skewness: 0.5068793
## estimated kurtosis: 1.698634

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

```

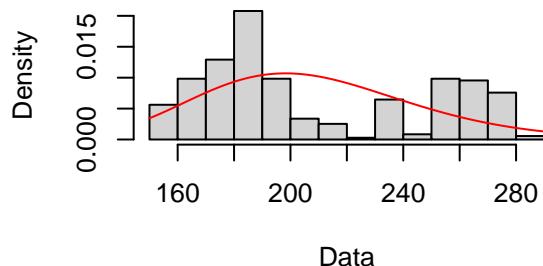
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

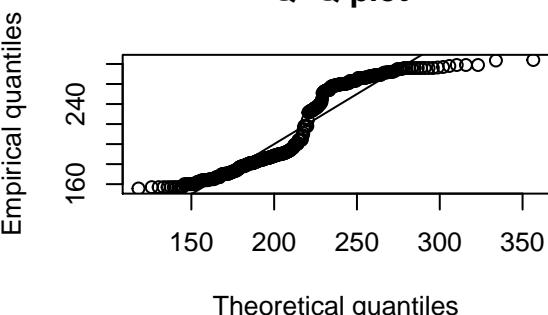
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

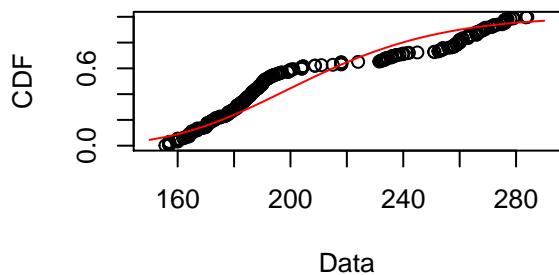
Empirical and theoretical dens.



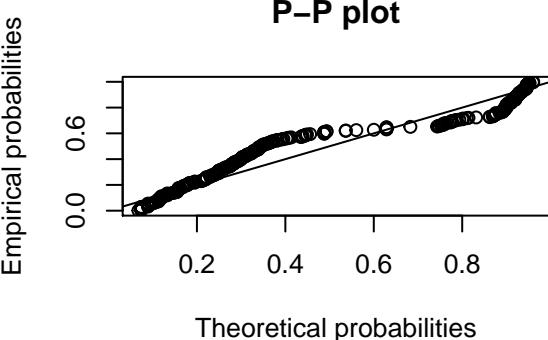
Q–Q plot



Empirical and theoretical CDFs

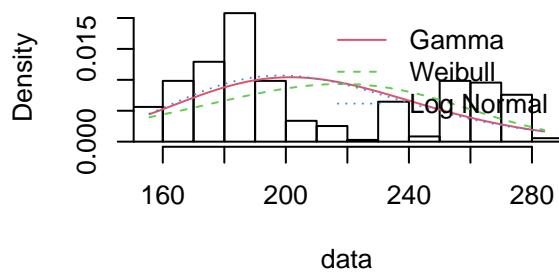


P–P plot

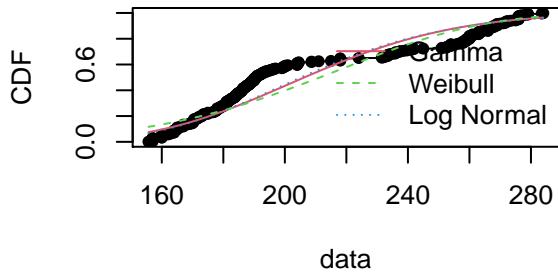


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

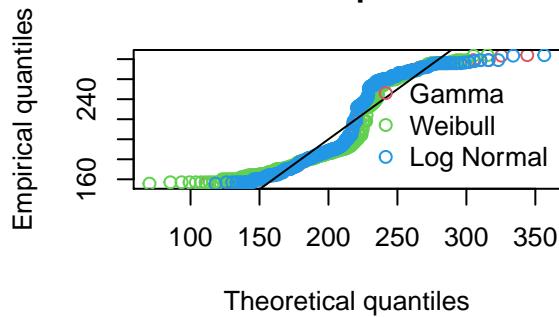
Histogram and theoretical densities



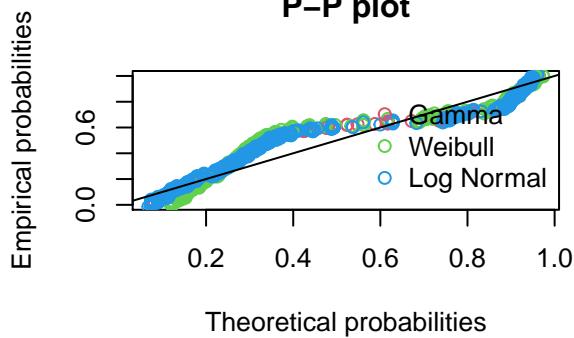
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```
# Goodness of fit comparisons across fits (can't include the sqrt normal bc it
# becomes diff response values)
gofstat(list(fit.gamma, fit.weibull, fit.ln), fitnames = c("Gamma", "Weibull", "Log Normal"))
```

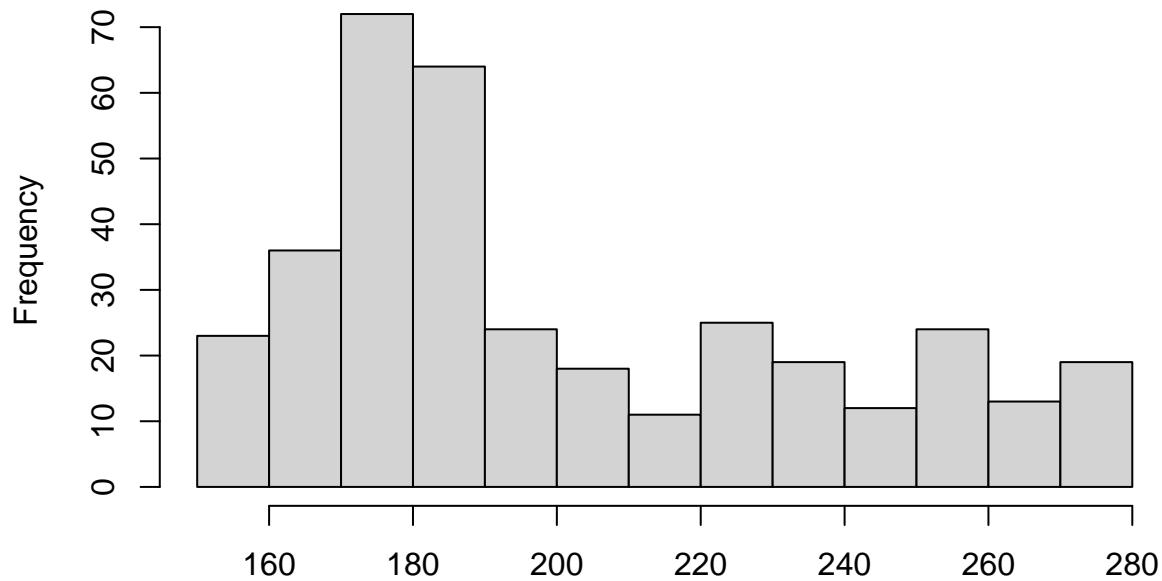
```
## Goodness-of-fit statistics
##                                     Gamma   Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.1785836 0.2027496 0.1693063
## Cramer-von Mises statistic  2.8663238 3.4425909 2.6474478
## Anderson-Darling statistic 16.3276941 19.0800536 15.2375314
##
## Goodness-of-fit criteria
##                                     Gamma   Weibull Log Normal
## Akaike's Information Criterion 3611.752 3658.094 3603.051
## Bayesian Information Criterion 3619.502 3665.844 3610.801
```

log normal looks like it's the best fit

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

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

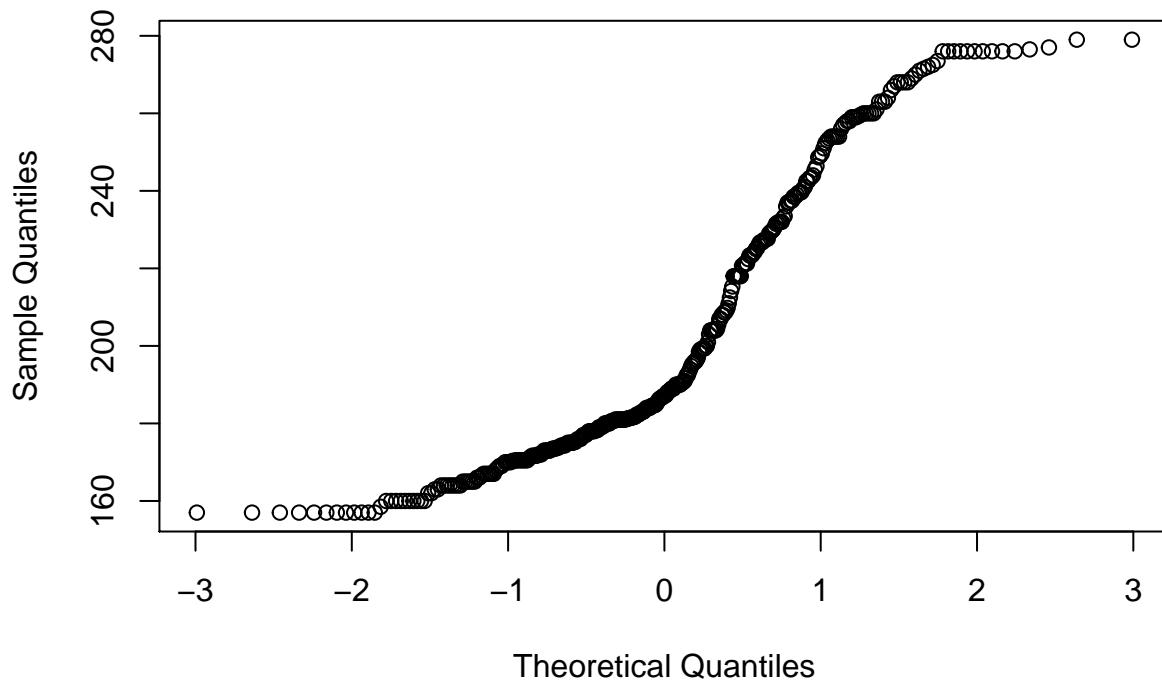
Histogram of kbs_sd_plot_growthhabit\$julian_min



kbs_sd_plot_growthhabit\$julian_min

```
qqnorm(kbs_sd_plot_growthhabit$julian_min)
```

Normal Q-Q Plot



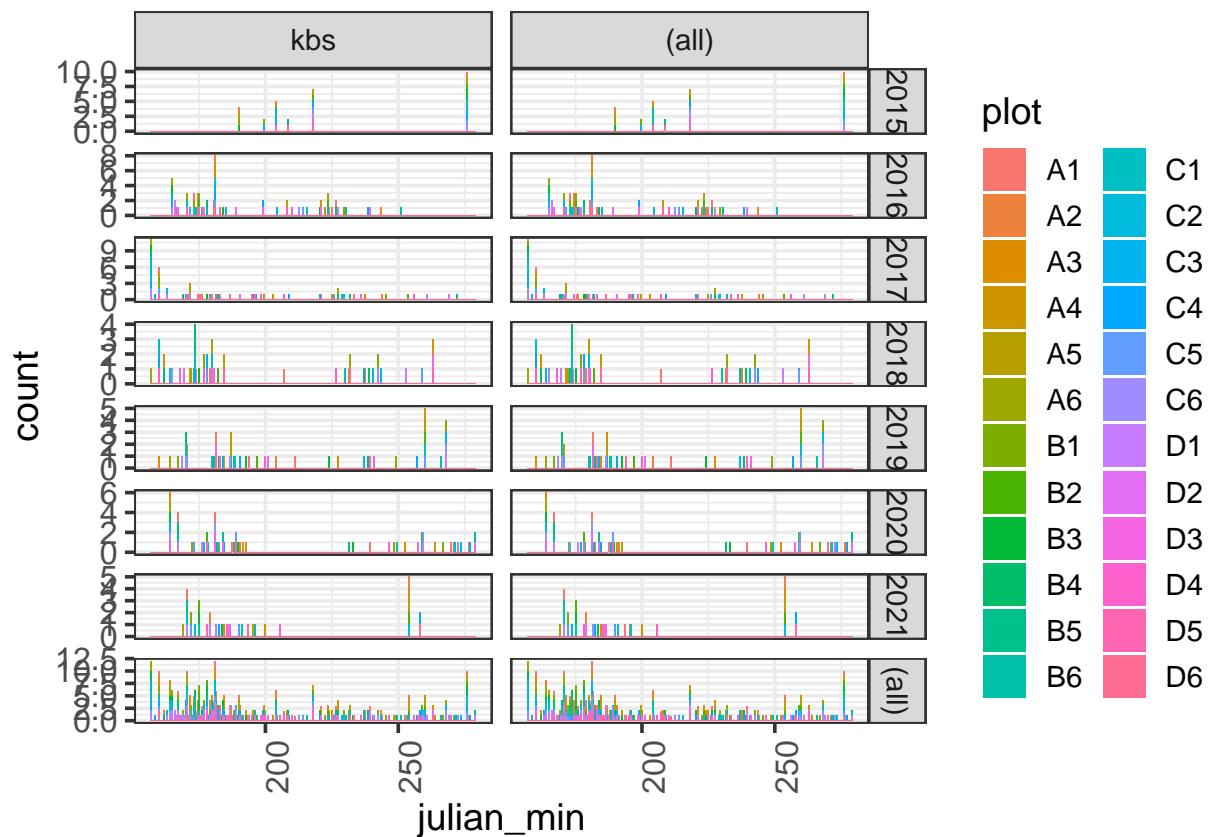
```

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

## 
## Shapiro-Wilk normality test
## 
## data: kbs_sd_plot_growthhabit$julian_min
## W = 0.88978, p-value = 1.962e-15

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

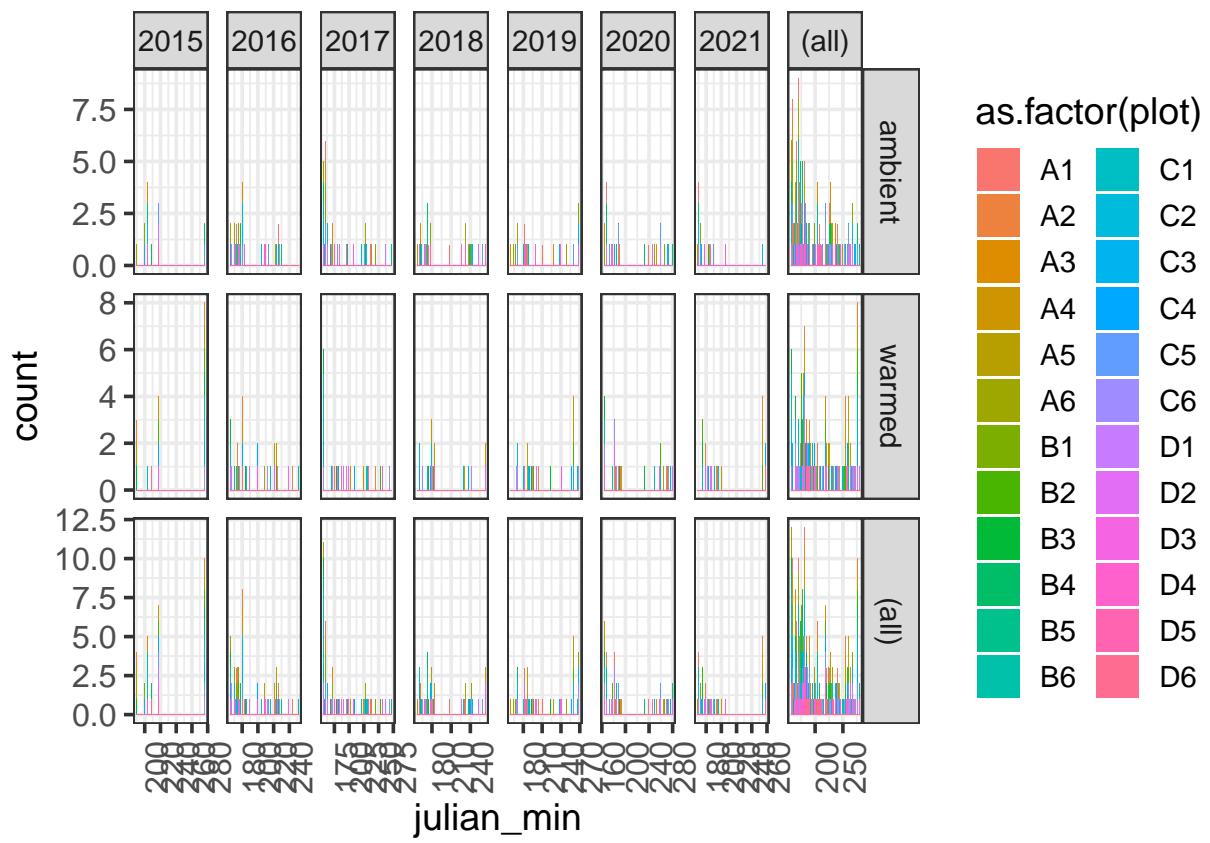
```



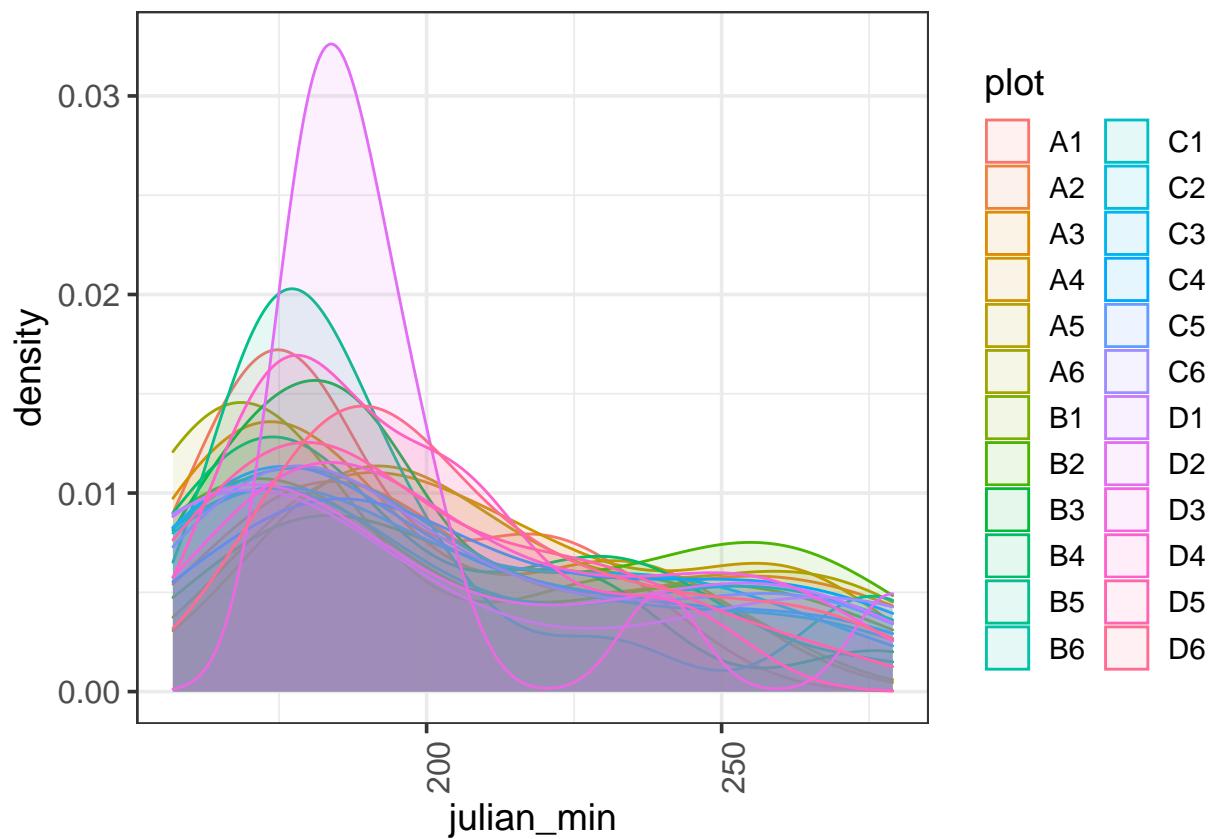
```

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

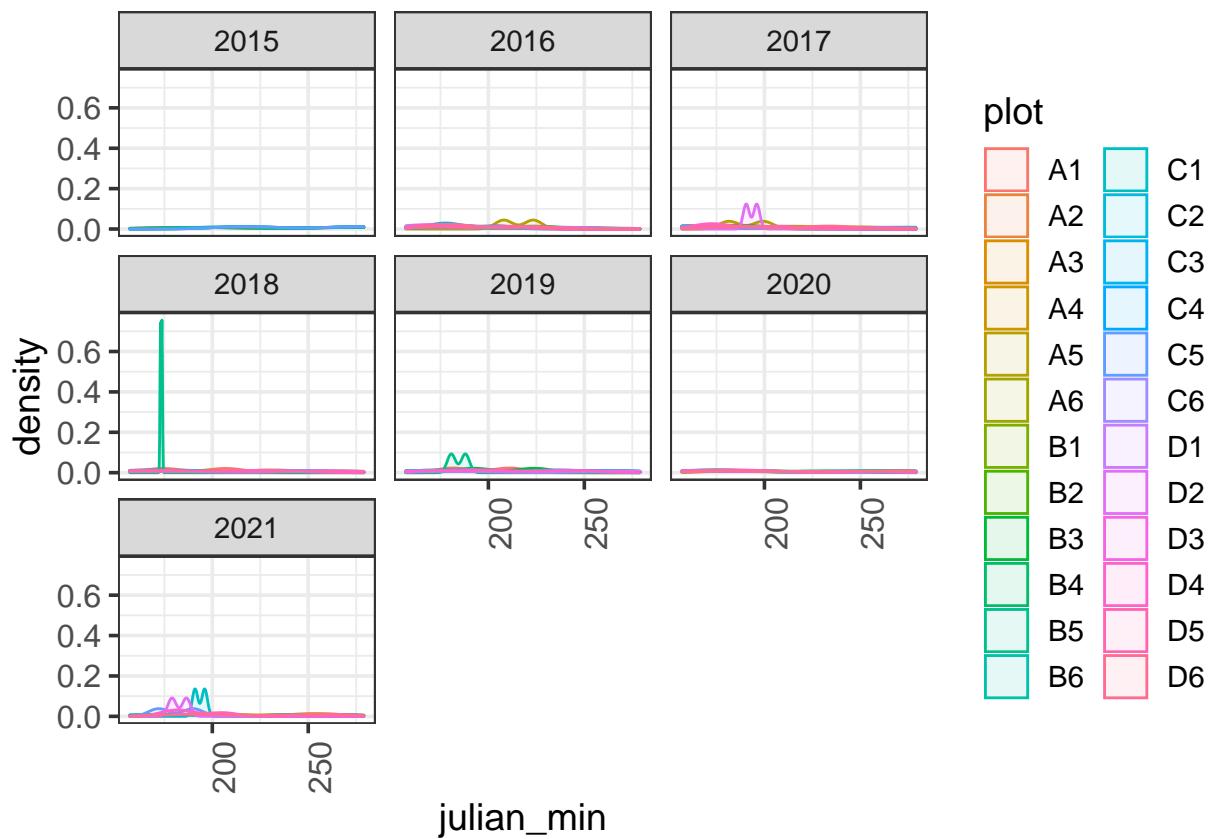
```



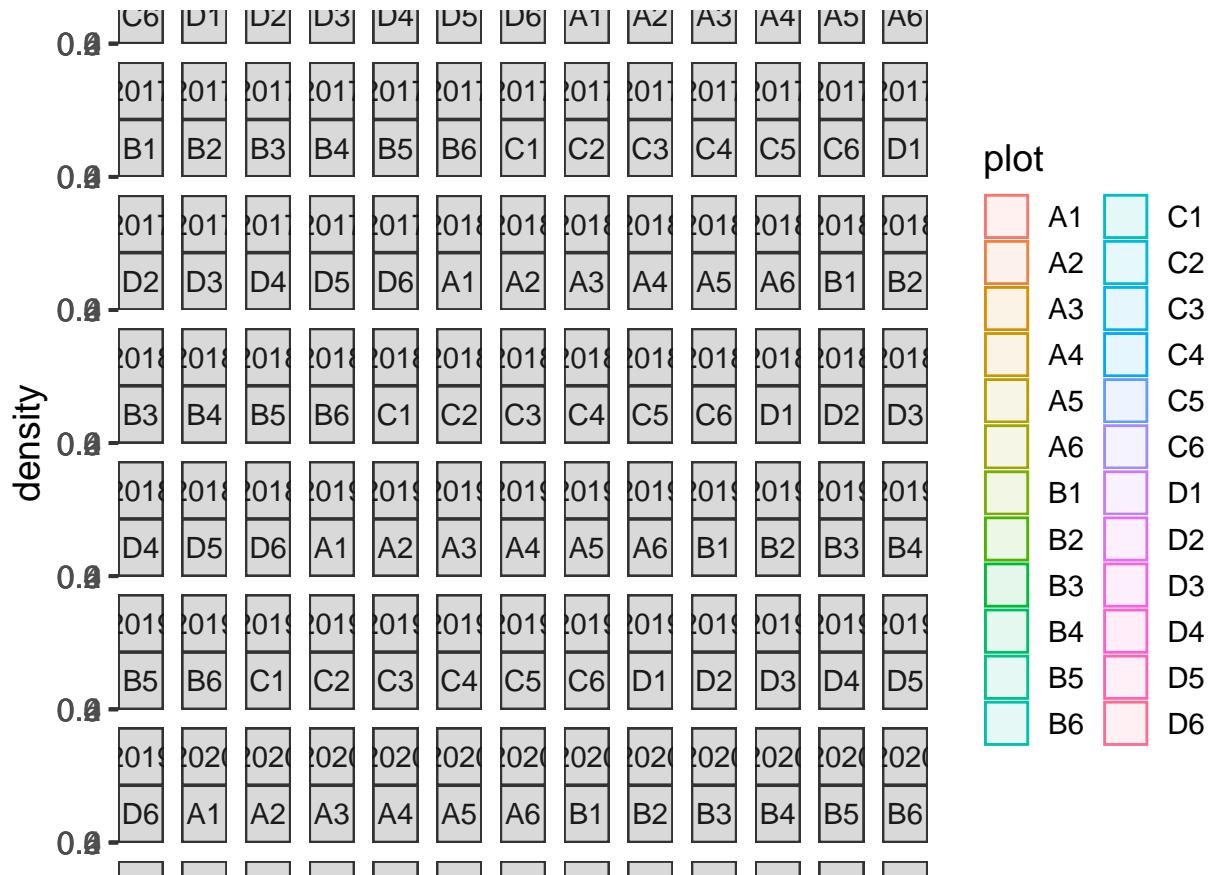
```
ggplot(kbs_sd_plot_growthhabit, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

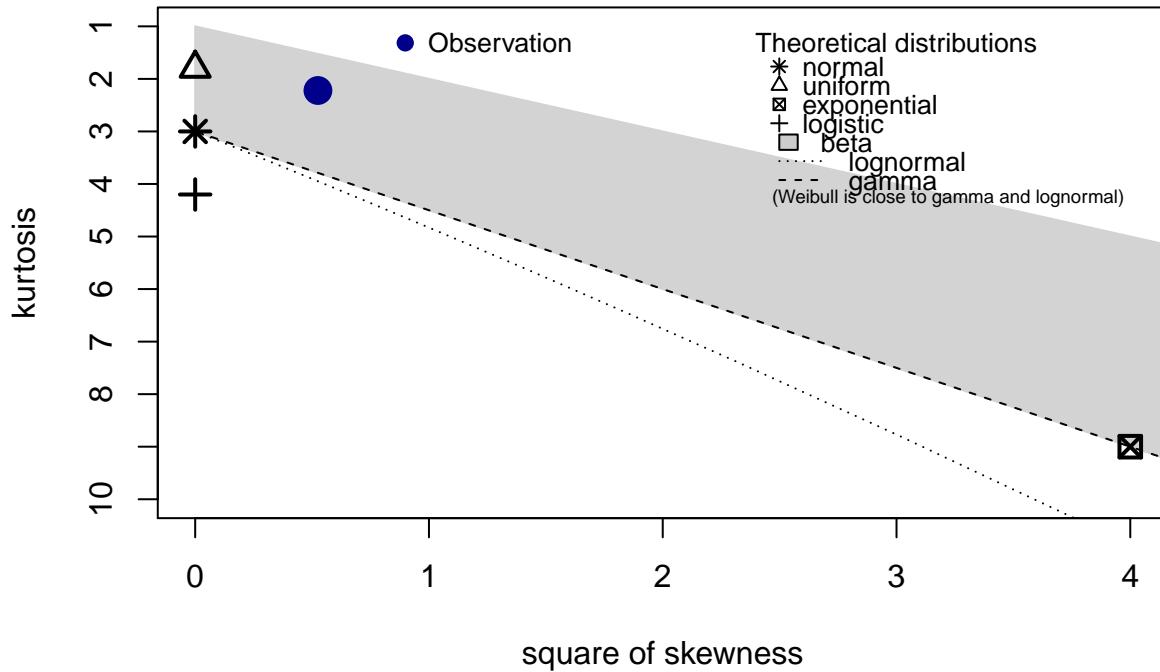


```
ggplot(kbs_sd_plot_growthhabit, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
  facet_wrap(~year + plot)
```



```
# Exploring distributions for these data:  
descdist(kbs_sd_plot_growthhabit$julian_min, discrete = FALSE)
```

Cullen and Frey graph

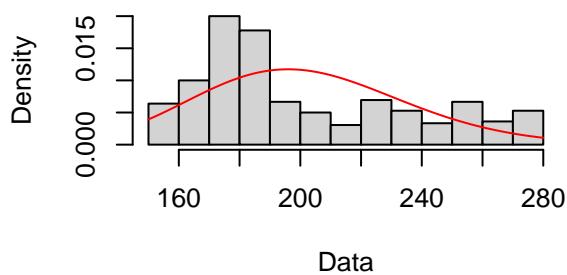
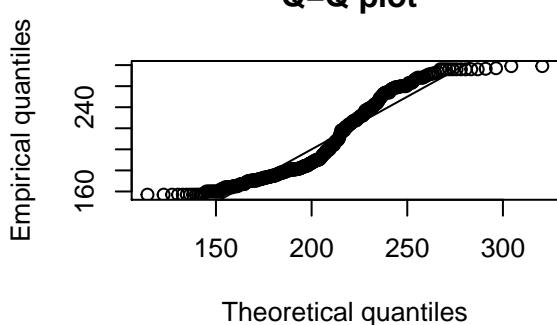
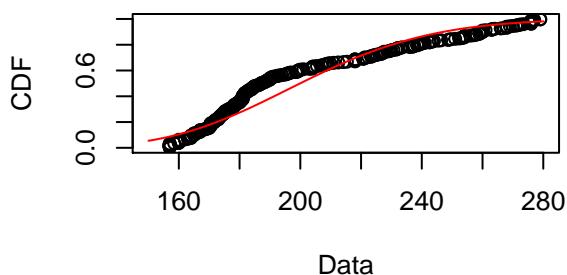
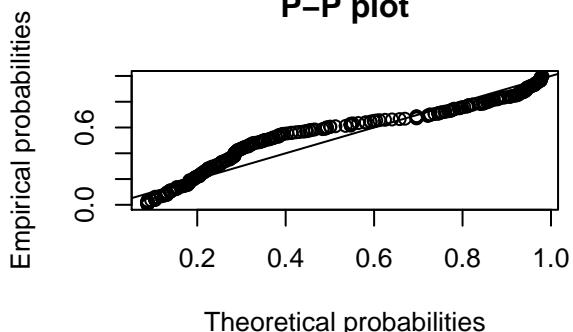


```

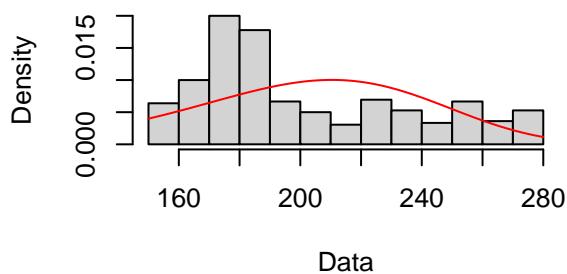
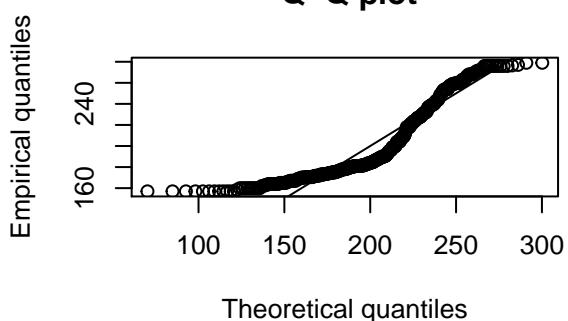
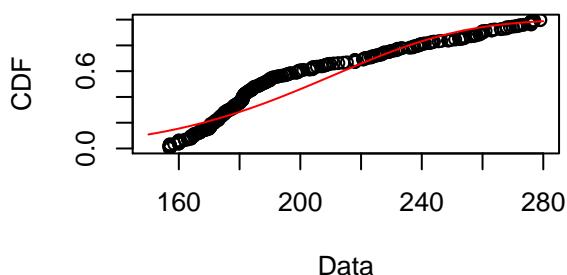
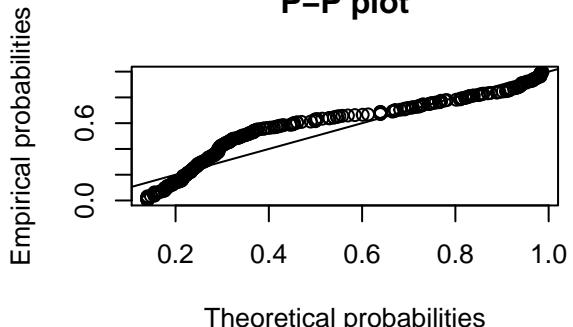
## summary statistics
## -----
## min: 157   max: 279
## median: 187.125
## mean: 201.8628
## estimated sd: 35.57648
## estimated skewness: 0.7246106
## estimated kurtosis: 2.221082

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

```

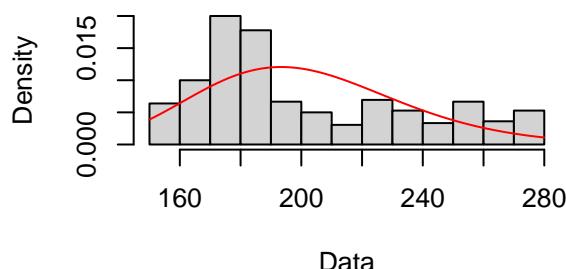
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

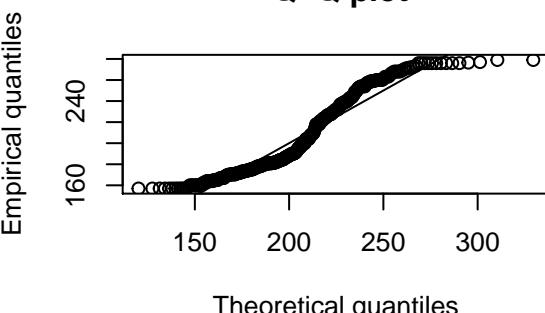
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

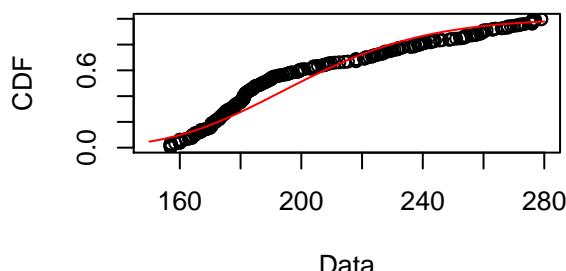
Empirical and theoretical dens.



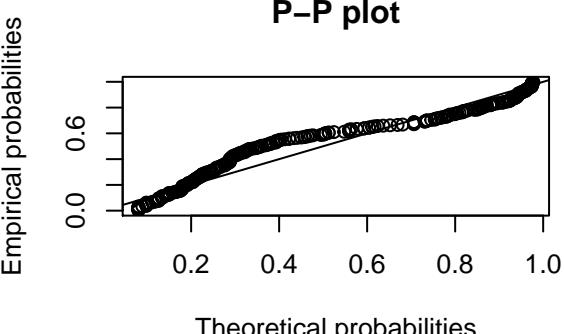
Q-Q plot



Empirical and theoretical CDFs

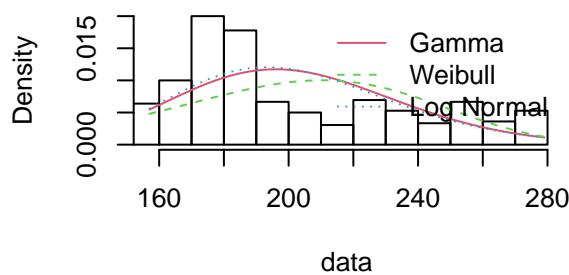


P-P plot

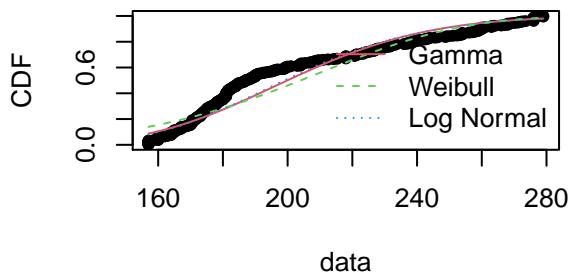


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

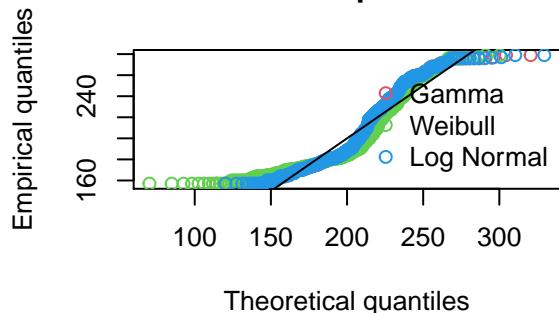
Histogram and theoretical densities



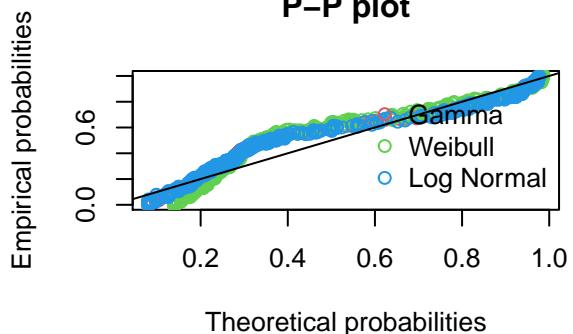
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```
# Goodness of fit comparisons across fits (can't include the sqrt normal bc it
# becomes diff response values)
gofstat(list(fit.gamma, fit.weibull, fit.ln), fitnames = c("Gamma", "Weibull", "Log Normal"))
```

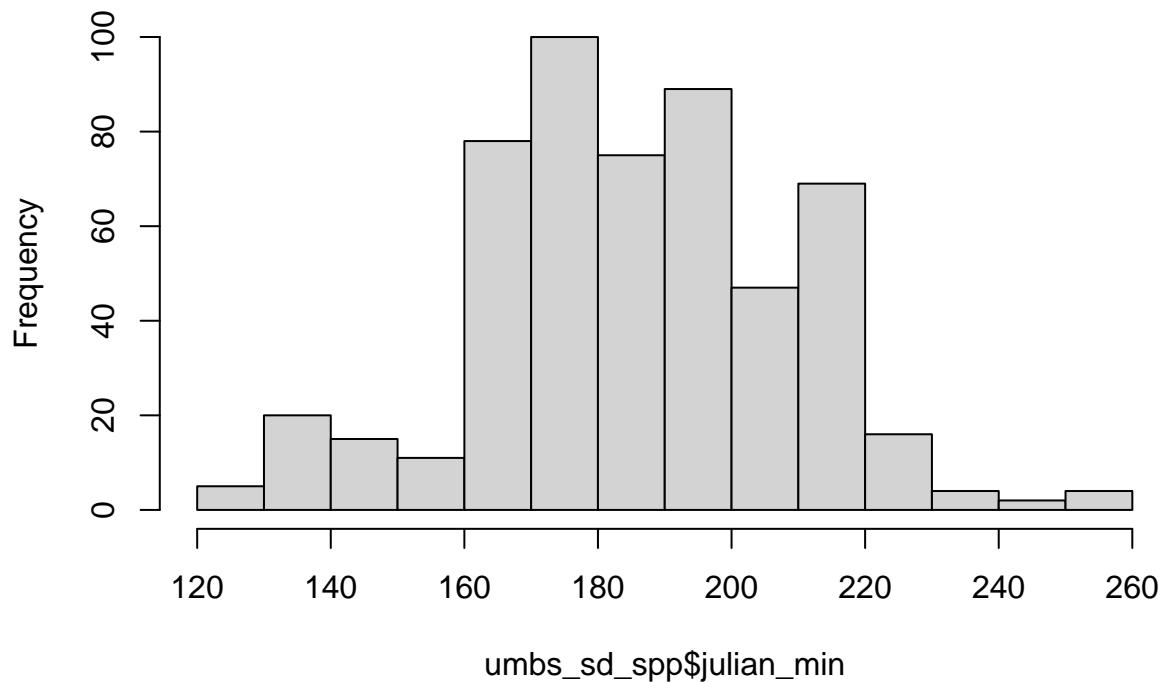
```
## Goodness-of-fit statistics
##                                     Gamma    Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.1575768 0.1766799 0.1489642
## Cramer-von Mises statistic   2.1866107 2.8100431 1.9883597
## Anderson-Darling statistic  12.3851380 16.0068291 11.3025751
##
## Goodness-of-fit criteria
##                                     Gamma    Weibull Log Normal
## Akaike's Information Criterion 3566.956 3636.177 3555.237
## Bayesian Information Criterion 3574.728 3643.949 3563.009
```

log normal looks like it's the best fit

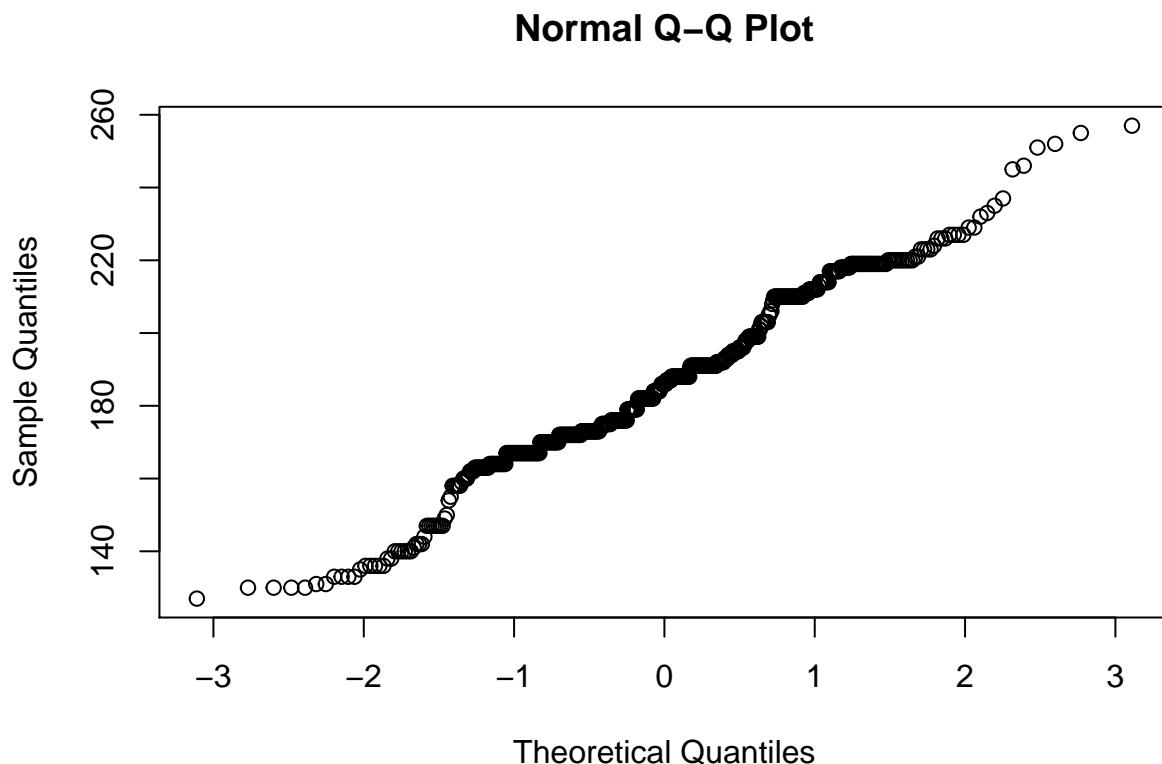
UMBS SPECIES LEVEL - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

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

Histogram of umbs_sd_spp\$julian_min



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



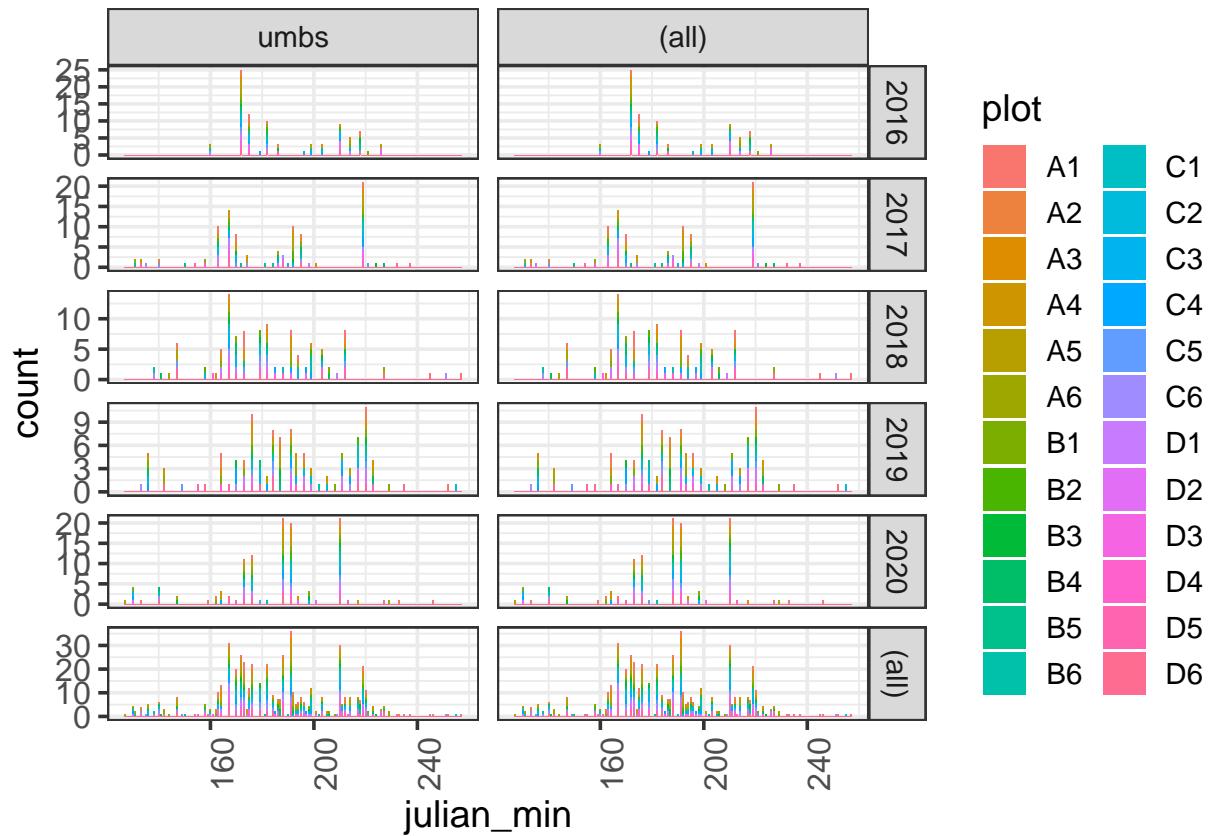
```

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 = 0.98284, p-value = 6.092e-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")

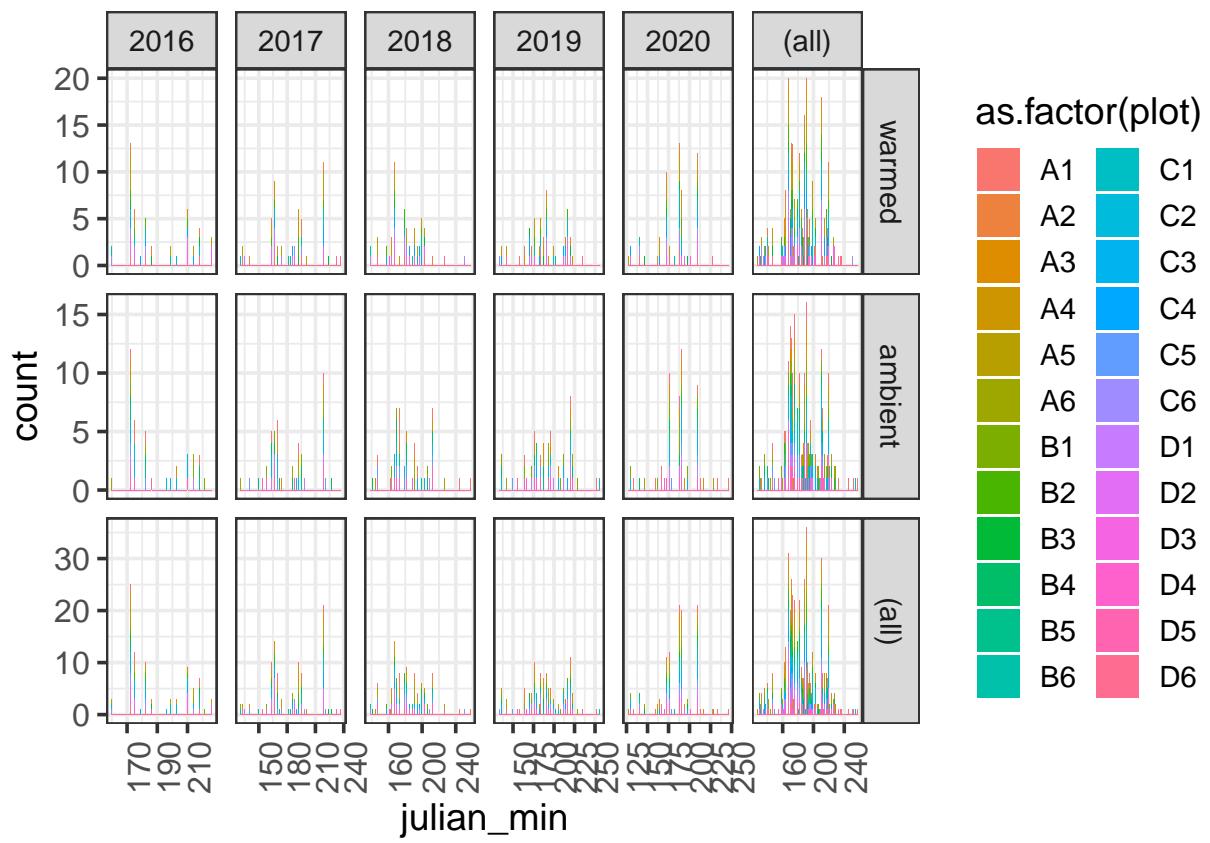
```



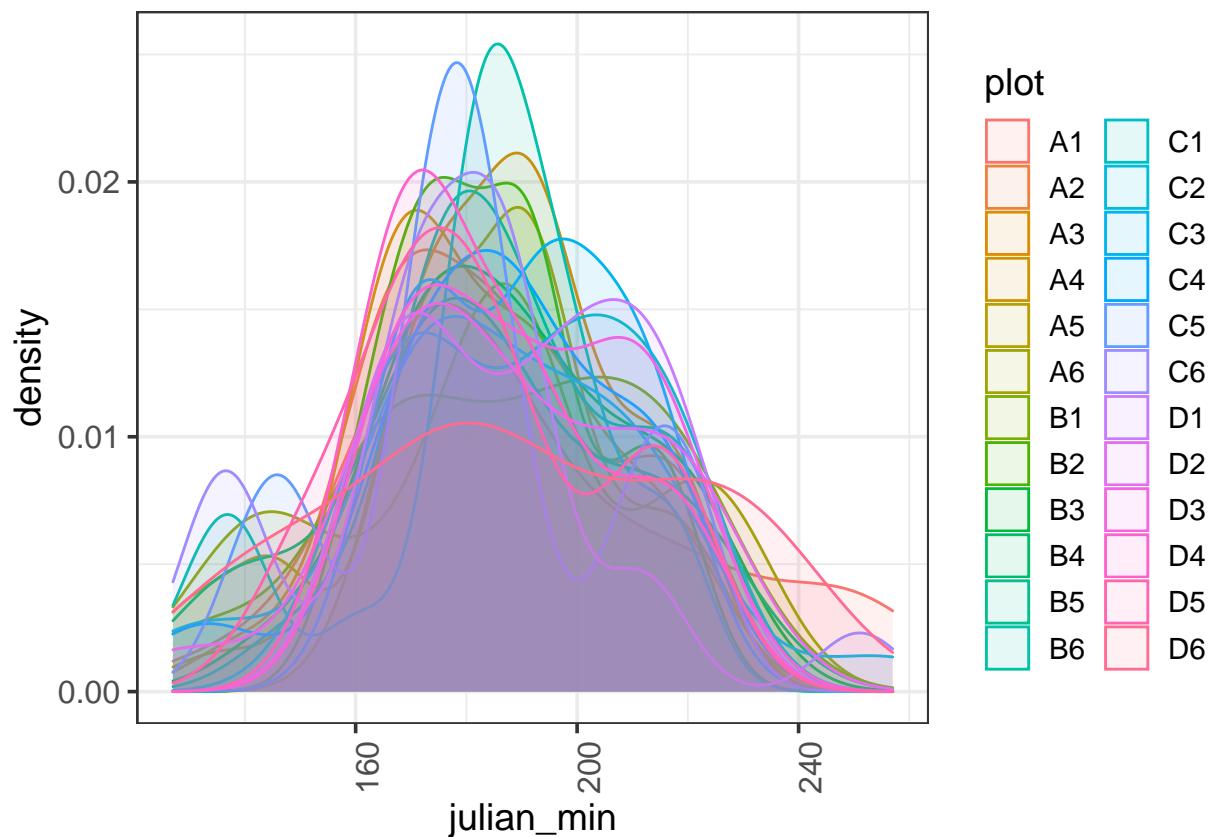
```

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

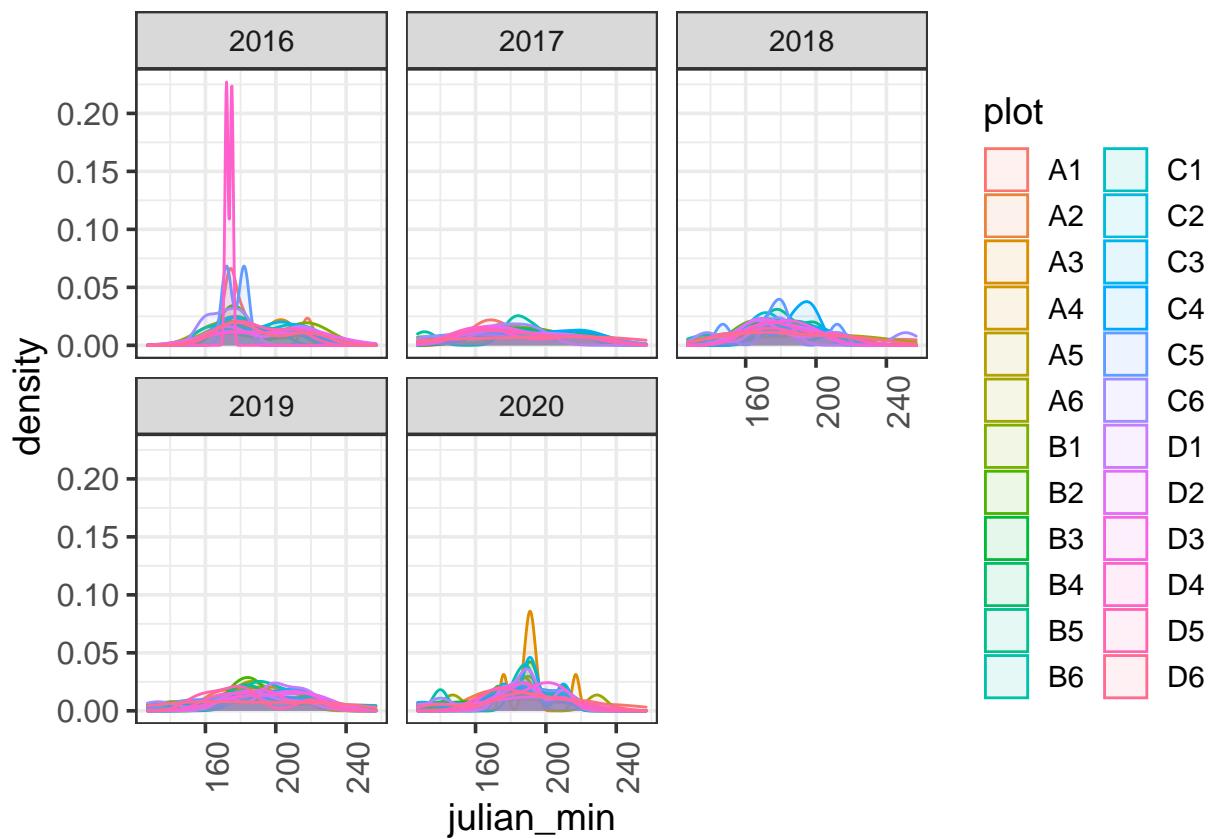
```



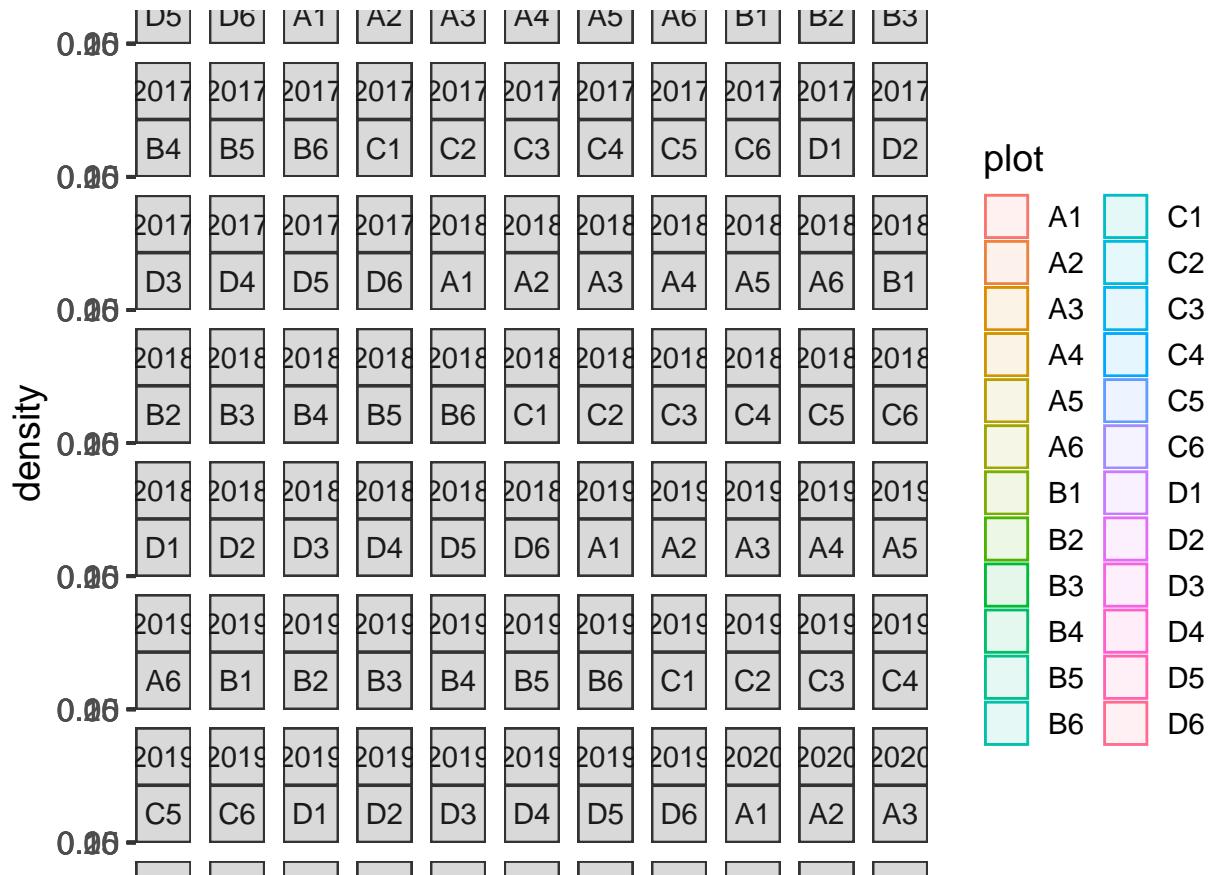
```
ggplot(umbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

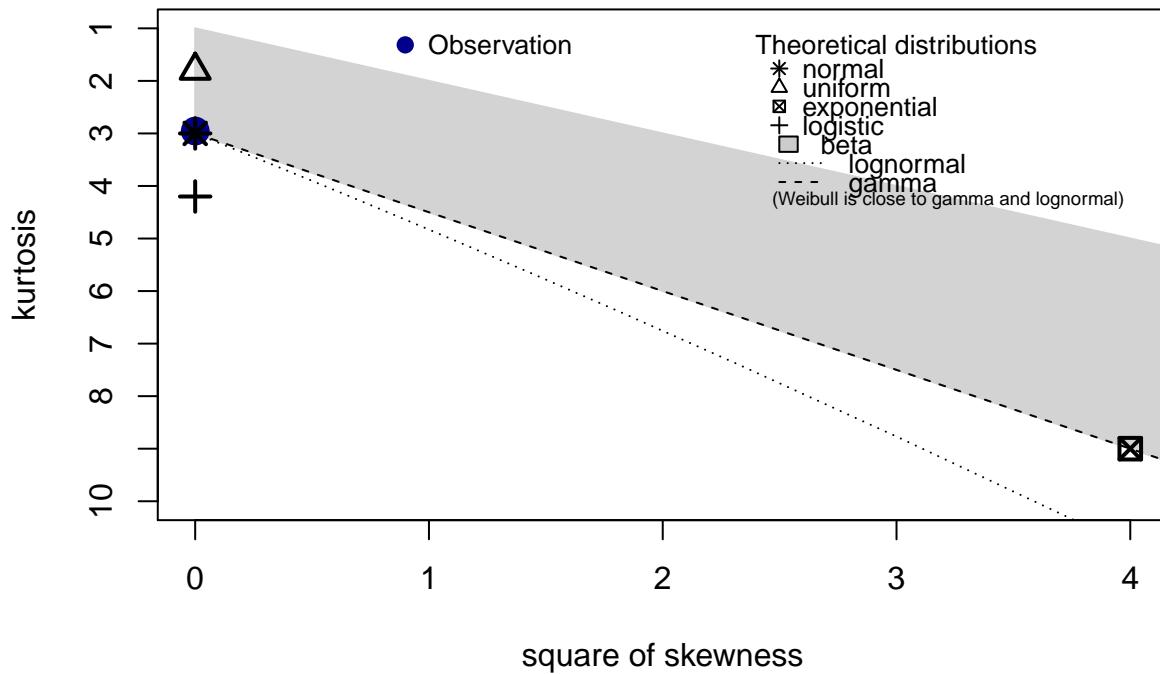


```
ggplot(umbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



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

Cullen and Frey graph

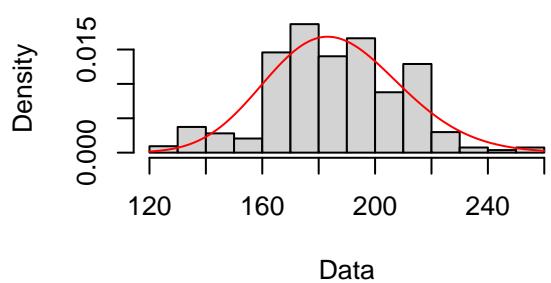
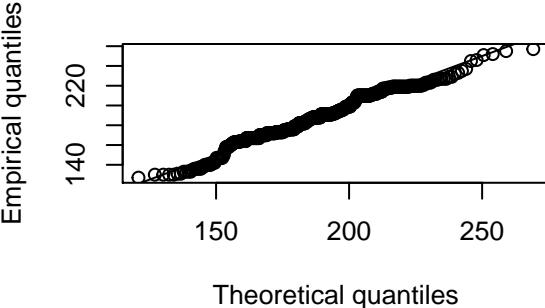
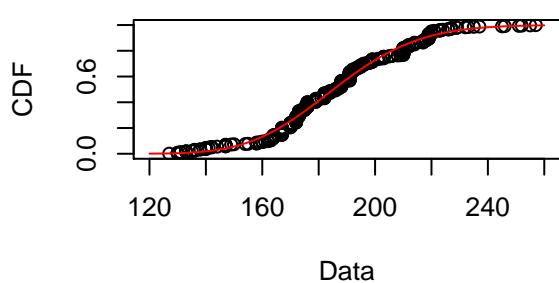
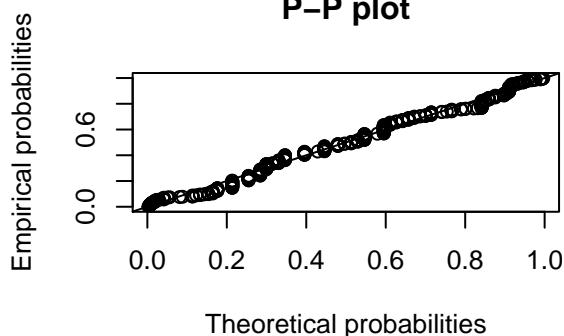


```

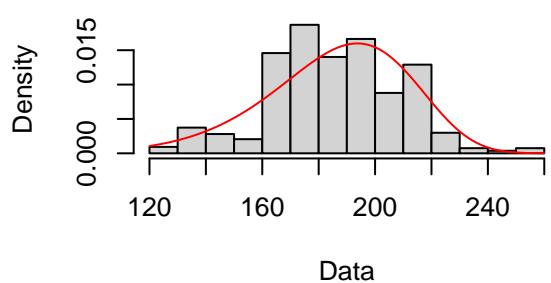
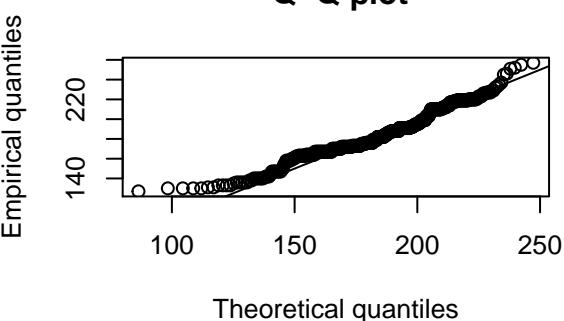
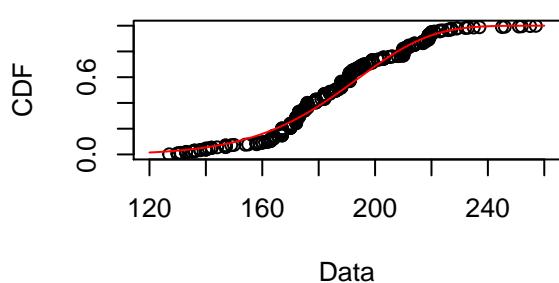
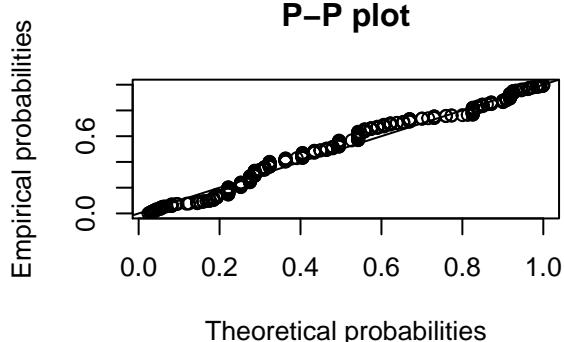
## summary statistics
## -----
## min: 127   max: 257
## median: 186
## mean: 186.2374
## estimated sd: 23.60302
## estimated skewness: 0.01509724
## estimated kurtosis: 2.954713

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

```

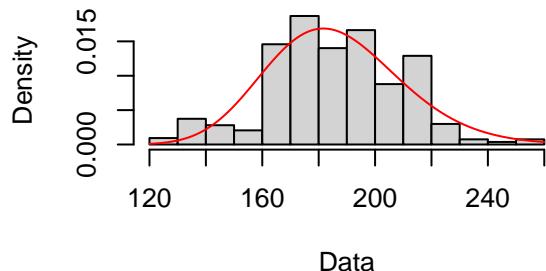
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

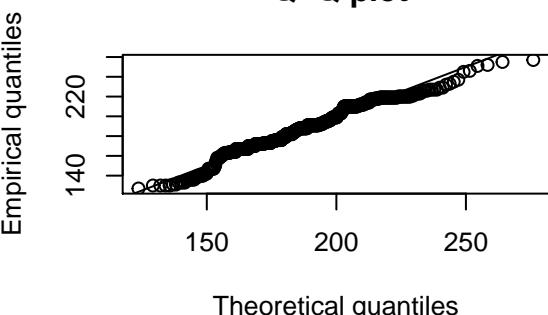
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

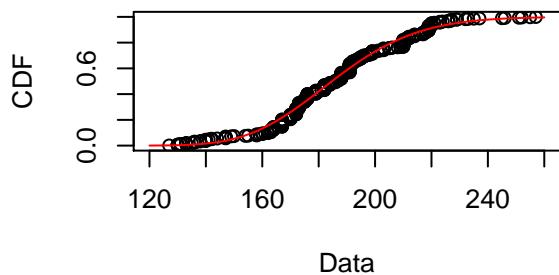
Empirical and theoretical dens.



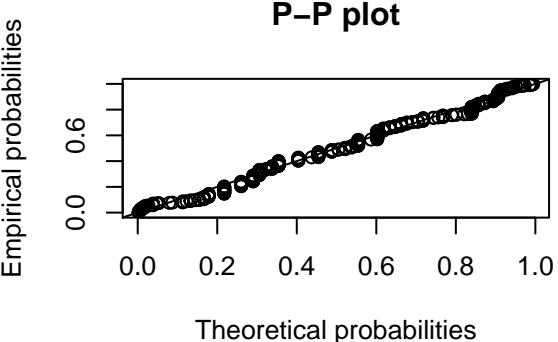
Q-Q plot



Empirical and theoretical CDFs

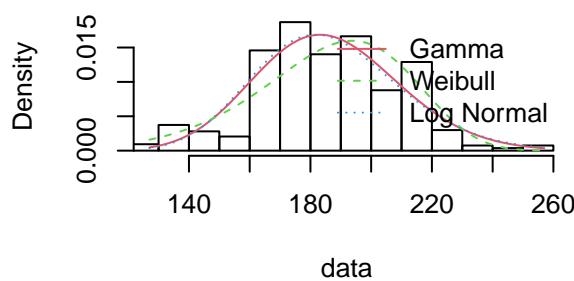


P-P plot

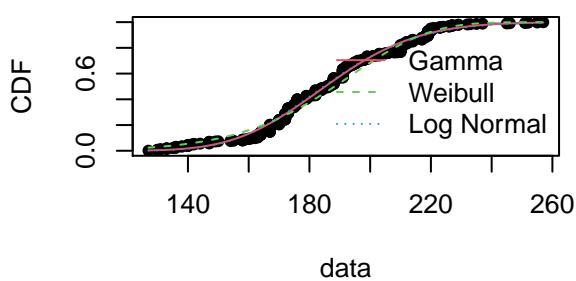


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

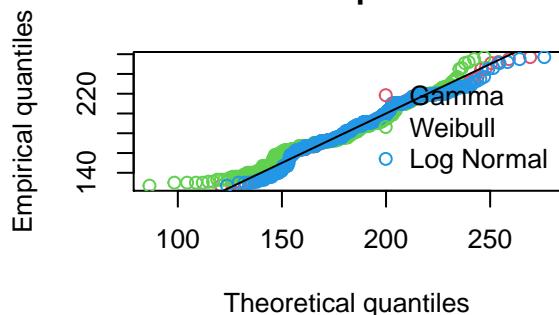
Histogram and theoretical densities



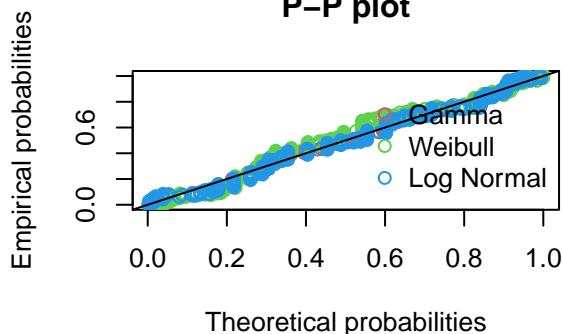
Empirical and theoretical CDFs



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 0.07501038 0.09584775 0.07344273
## Cramer-von Mises statistic   0.42029137 1.07967124 0.42831359
## Anderson-Darling statistic   3.34302569 6.08029776 3.67116016
##
## Goodness-of-fit criteria
##                                     Gamma   Weibull Log Normal
## Akaike's Information Criterion 4909.027 4937.708   4916.032
## Bayesian Information Criterion 4917.592 4946.273   4924.596

# Lognormal is best - going with a lognormal transformation for umbs seed set at
# the species level

```

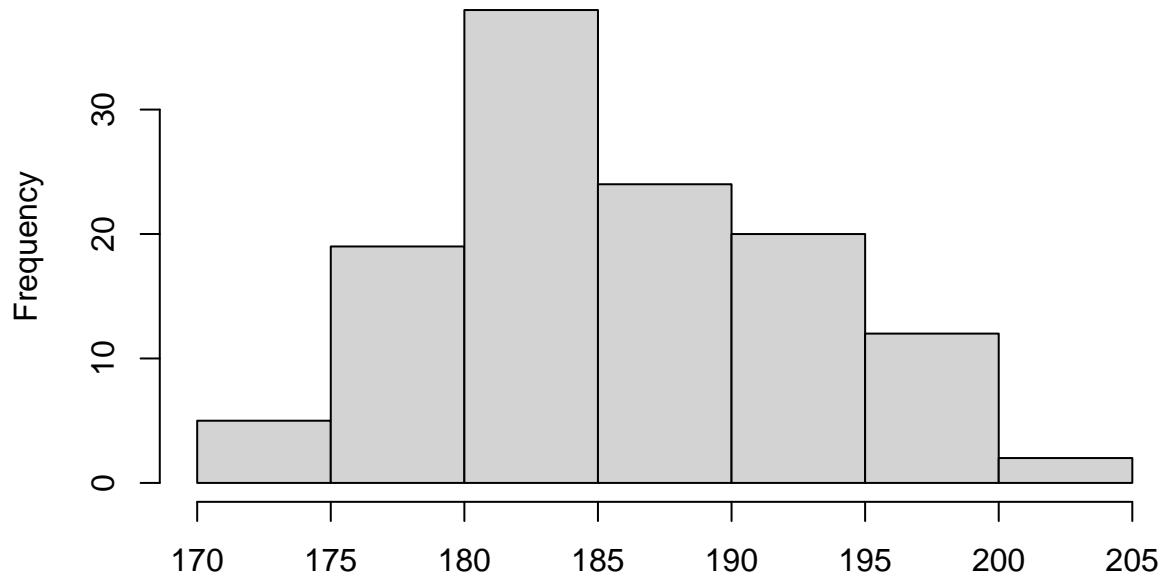
UMBS PLOT LEVEL - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

```

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

```

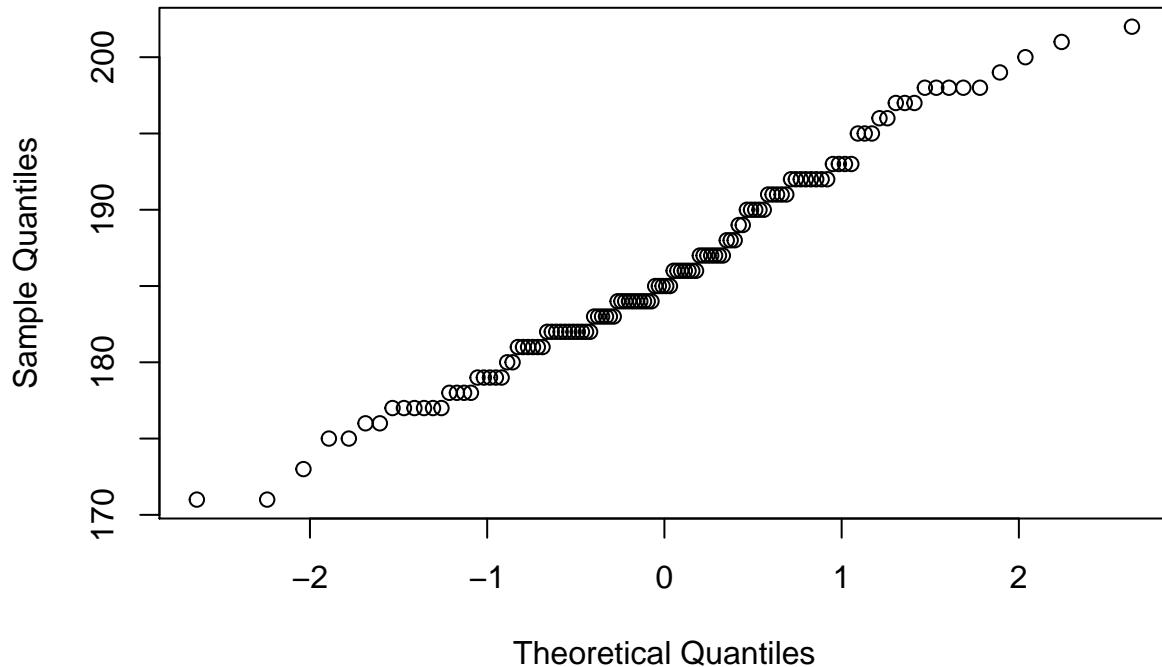
Histogram of umbs_sd_plot\$julian_min



`umbs_sd_plot$julian_min`

```
qqnorm(umbss_sd_plot$julian_min)
```

Normal Q-Q Plot



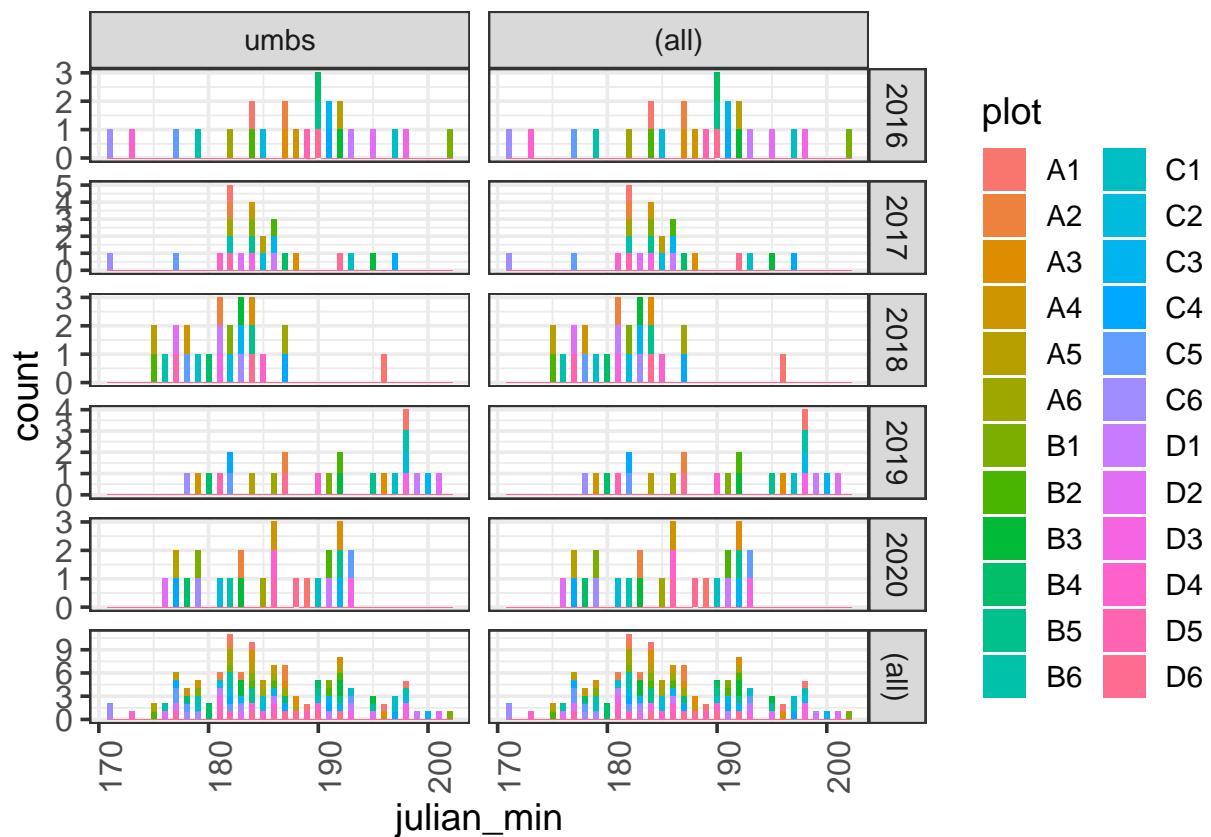
```

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

##
## Shapiro-Wilk normality test
##
## data: umbs_sd_plot$julian_min
## W = 0.98106, p-value = 0.08941

# 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")

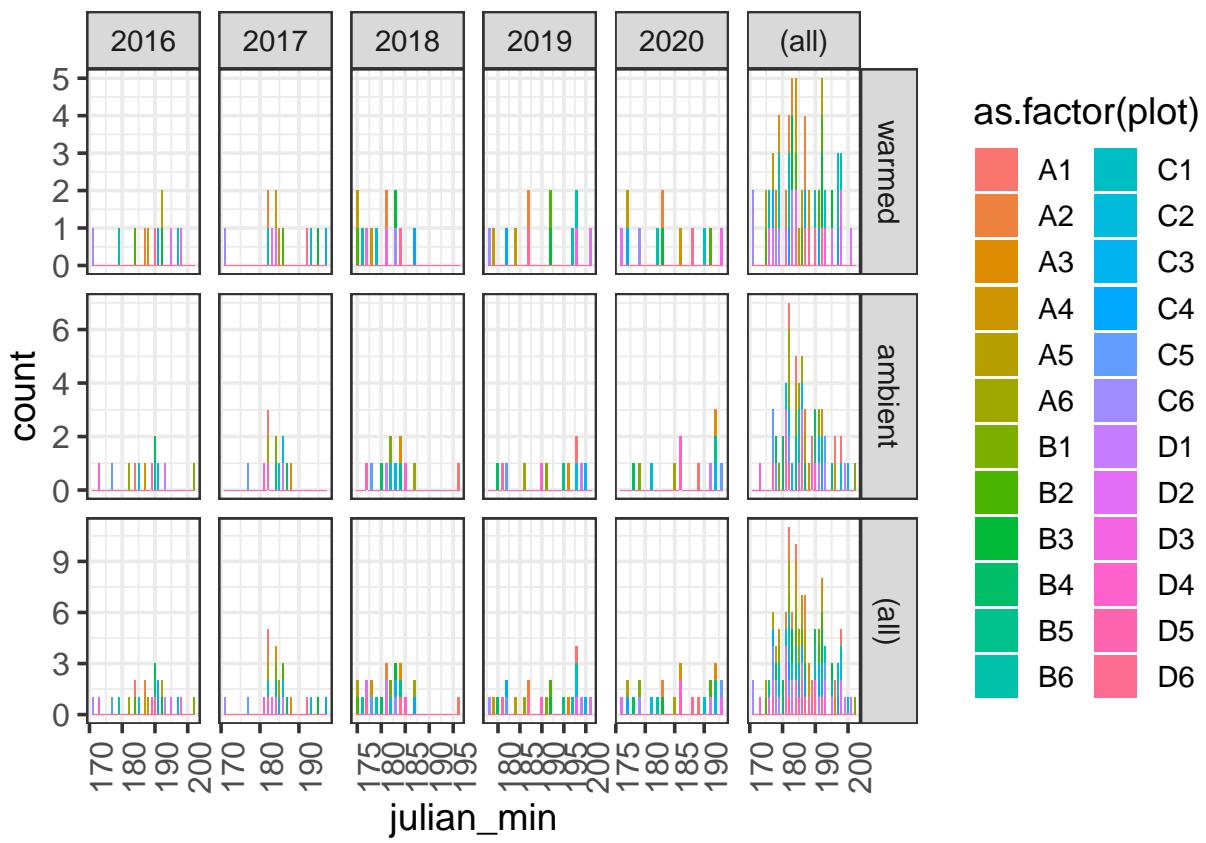
```



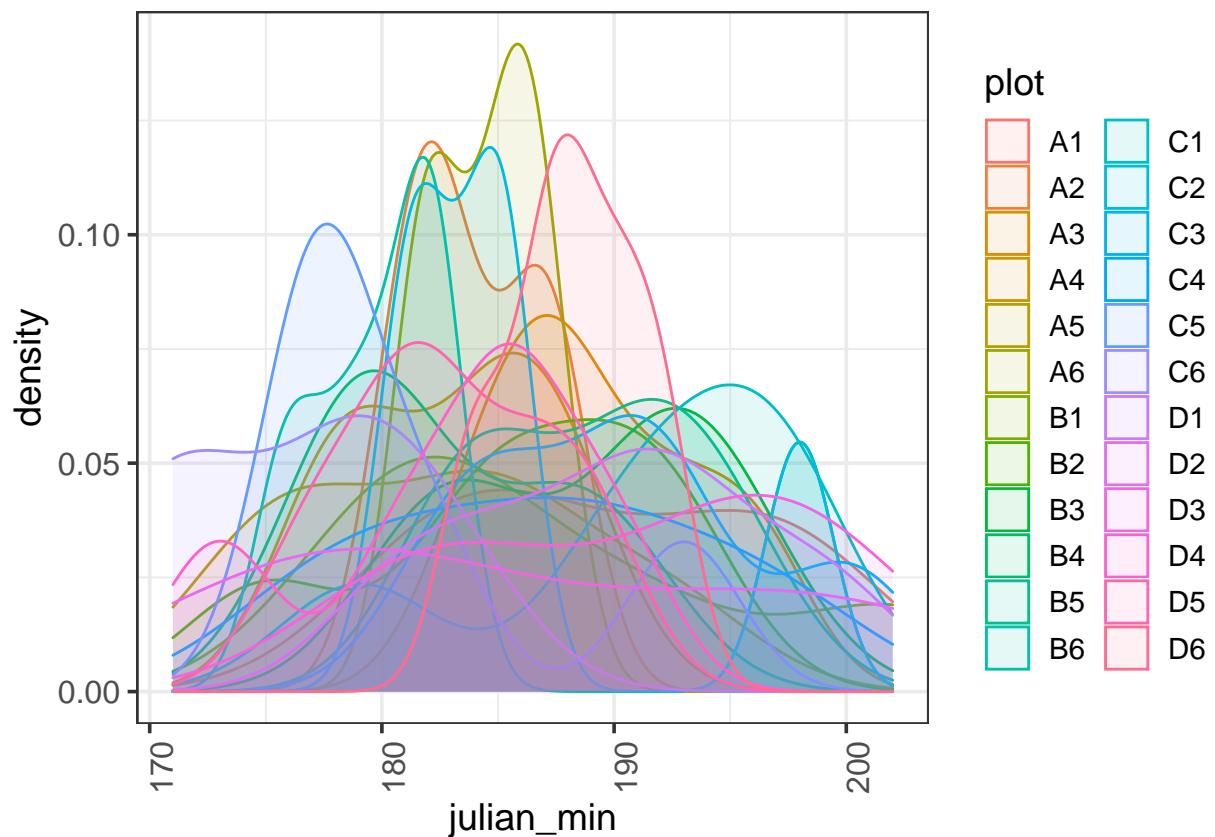
```

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

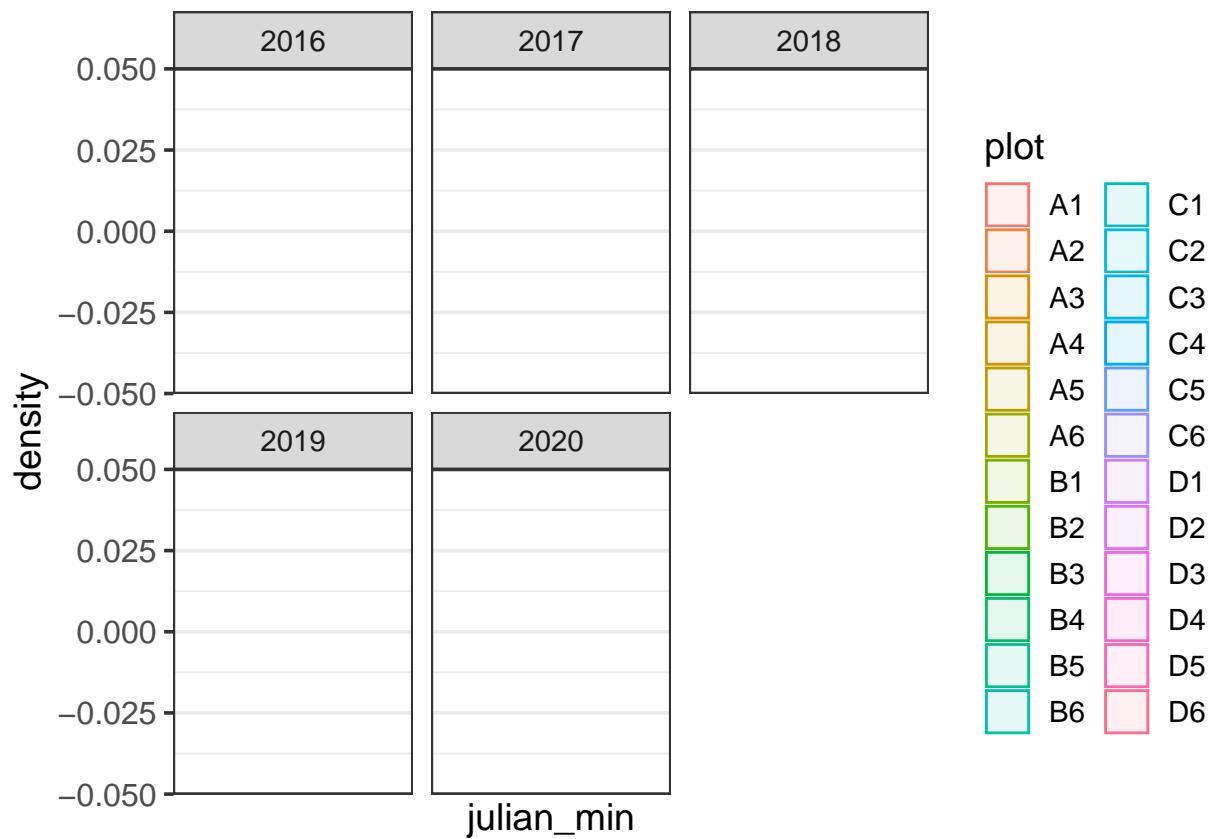
```



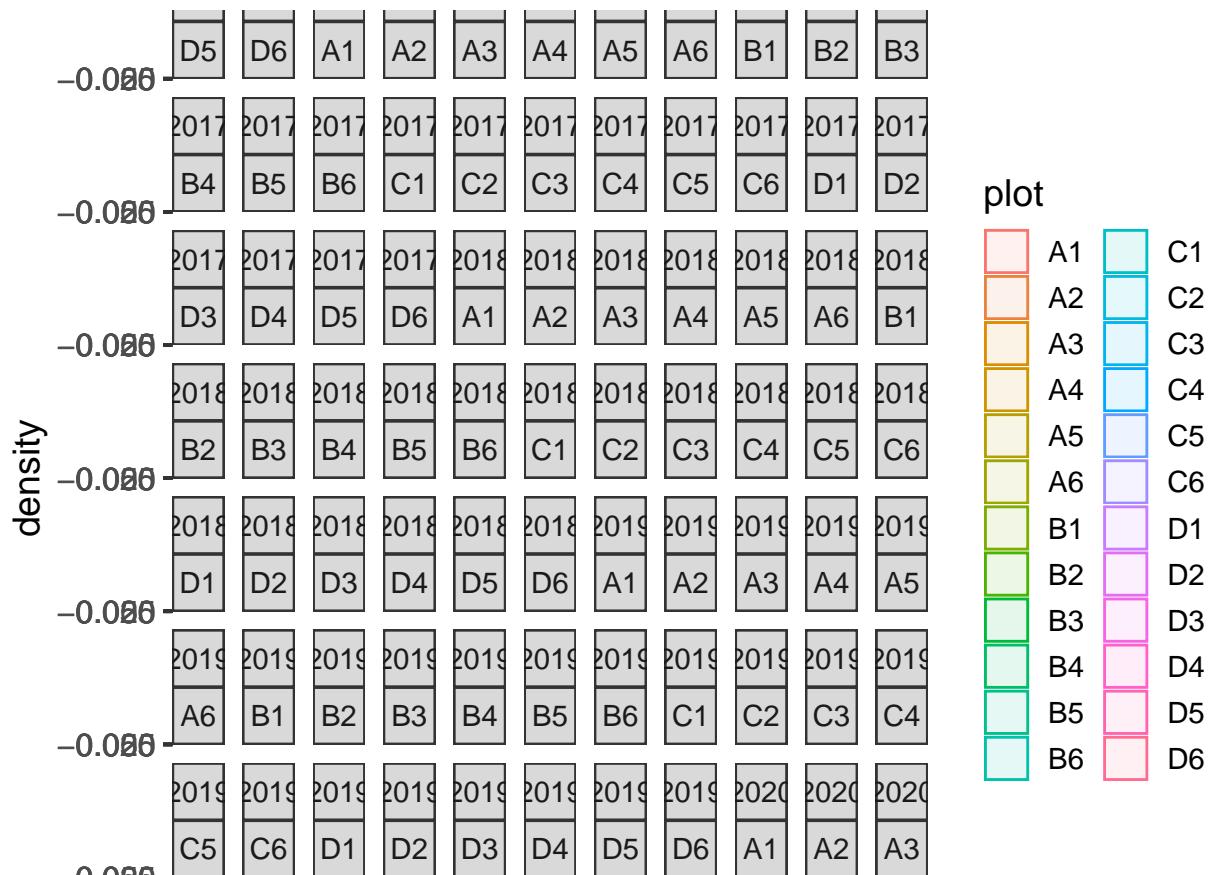
```
ggplot(umbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

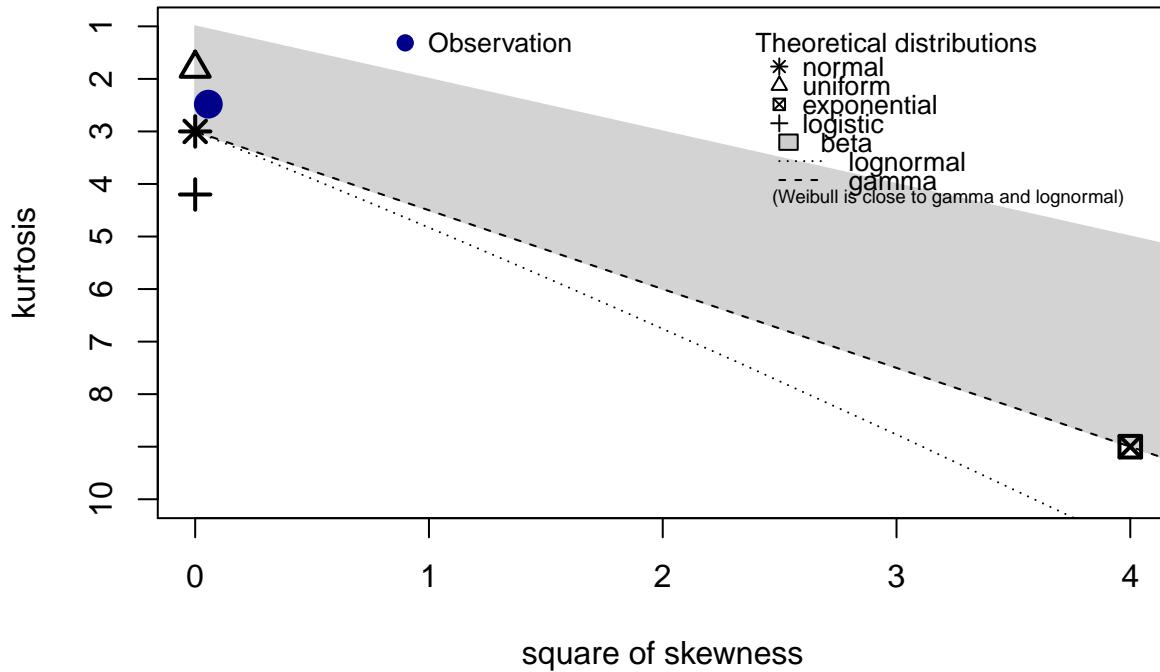


```
ggplot(umbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



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

Cullen and Frey graph



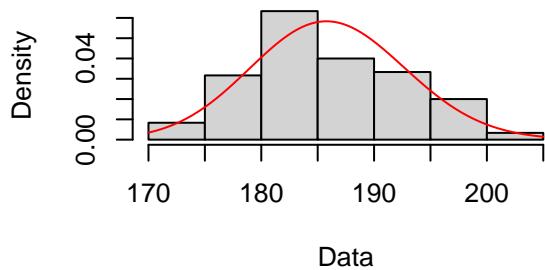
```

## summary statistics
## -----
## min: 171   max: 202
## median: 185
## mean: 186.025
## estimated sd: 6.886185
## estimated skewness: 0.2371013
## estimated kurtosis: 2.480561

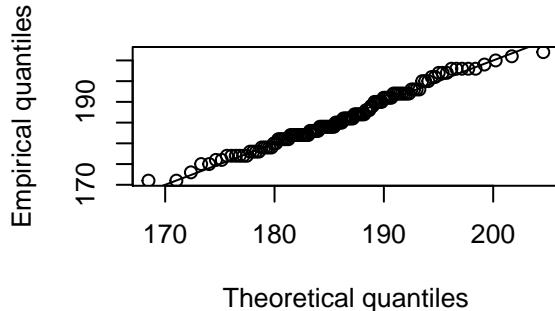
# Gamma distribution
fit.gamma <- fitdist(umbs_sd_plot$julian_min, "gamma")
plot(fit.gamma)

```

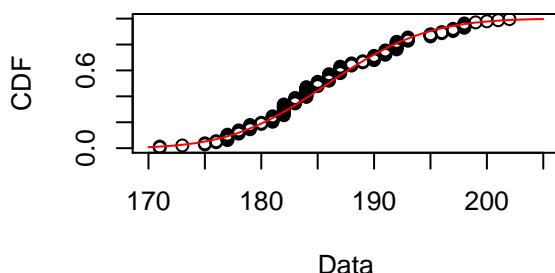
Empirical and theoretical dens.



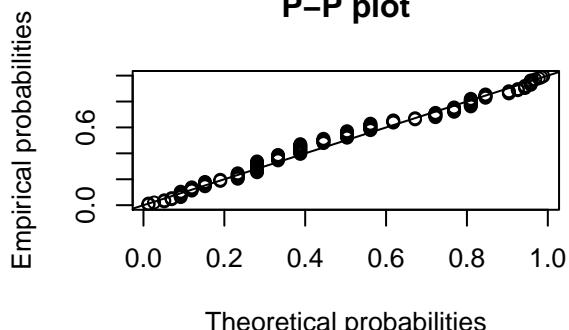
Q-Q plot



Empirical and theoretical CDFs

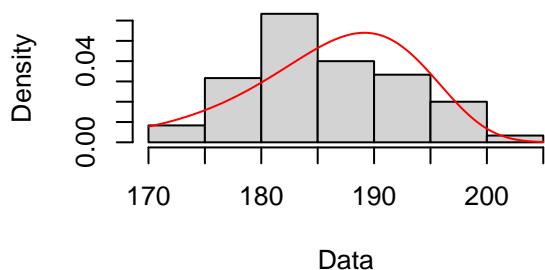


P-P plot

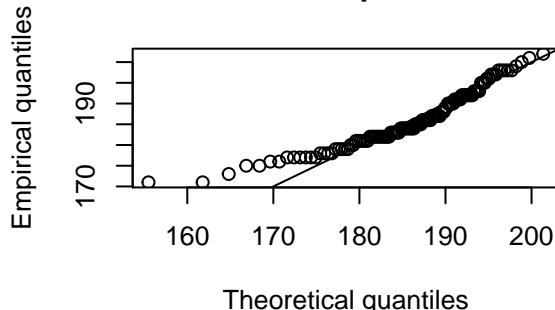


```
# Weibull distribution
fit.weibull <- fitdist(umbs_sd_plot$julian_min, "weibull")
plot(fit.weibull)
```

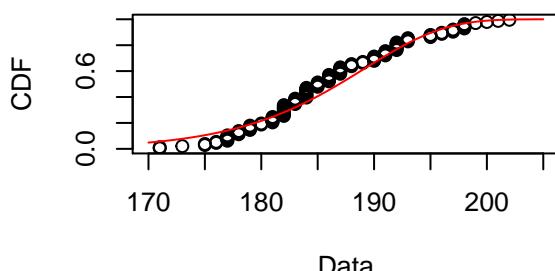
Empirical and theoretical dens.



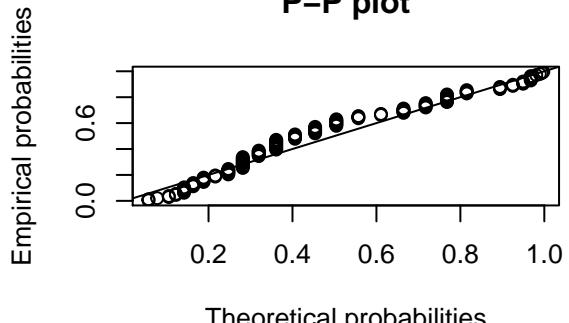
Q-Q plot



Empirical and theoretical CDFs

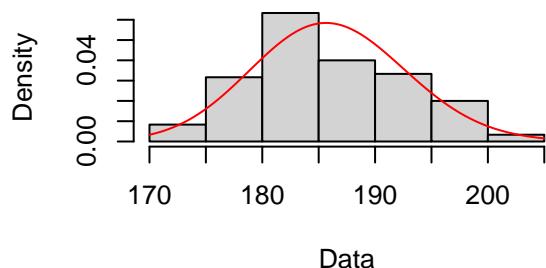


P-P plot

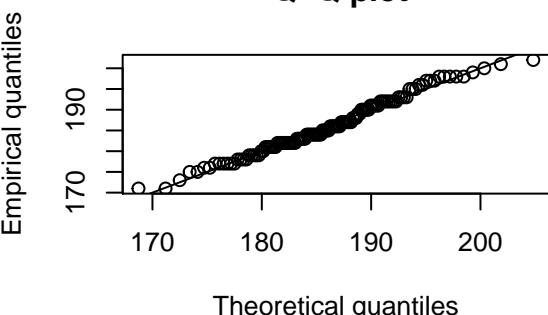


```
# Lognormal distribution
fit.ln <- fitdist(umbss_sd_plot$julian_min, "lnorm")
plot(fit.ln)
```

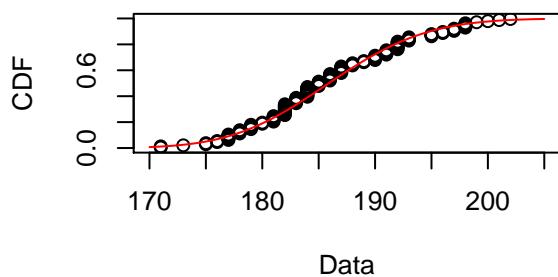
Empirical and theoretical dens.



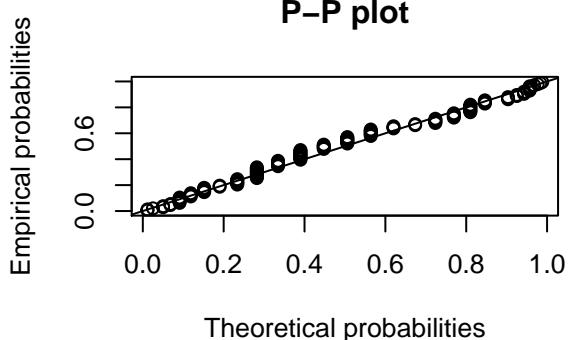
Q-Q plot



Empirical and theoretical CDFs

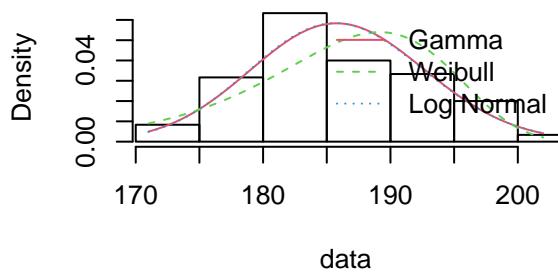


P-P plot

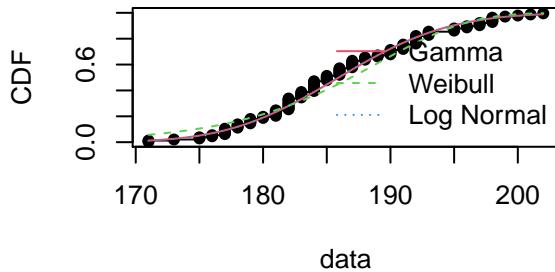


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

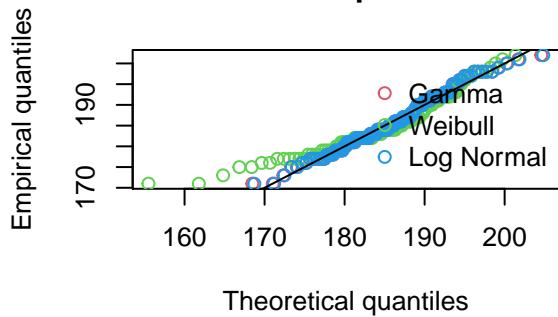
Histogram and theoretical densities



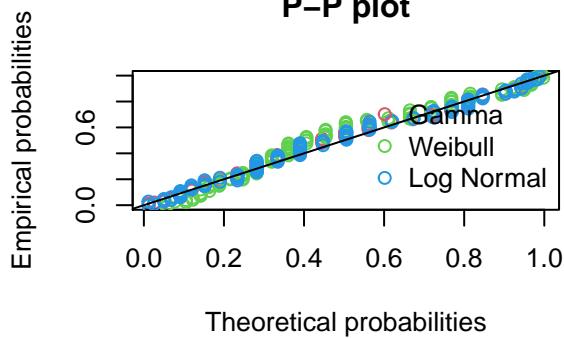
Empirical and theoretical CDFs



Q-Q plot



P-P plot



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

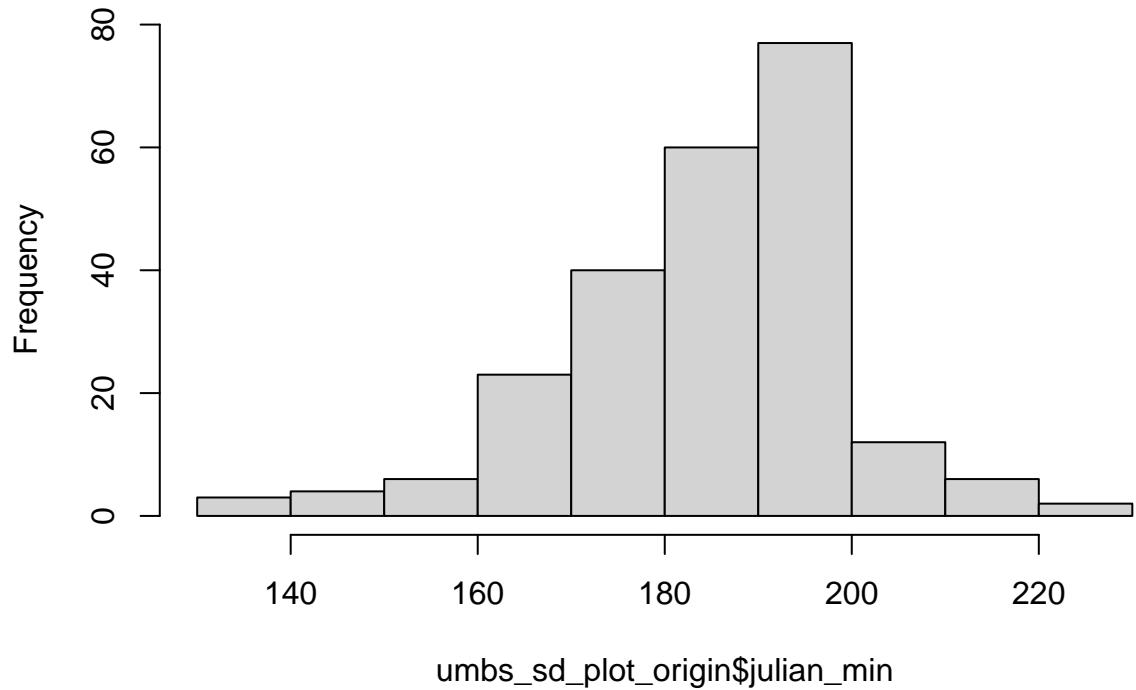
```
## Goodness-of-fit statistics
##                               Gamma   Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.08708919 0.1288883 0.08502908
## Cramer-von Mises statistic   0.12292058 0.3832521 0.11613520
## Anderson-Darling statistic   0.70871844 2.3005633 0.67294119
##
## Goodness-of-fit criteria
##                               Gamma   Weibull Log Normal
## Akaike's Information Criterion 805.9965 824.5308 805.7474
## Bayesian Information Criterion 811.5715 830.1058 811.3224
```

```
# log normal is slightly better here but going to go with a normal transformation
# for umbs seed set at the plot level based on graphs and shapiro test above
```

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

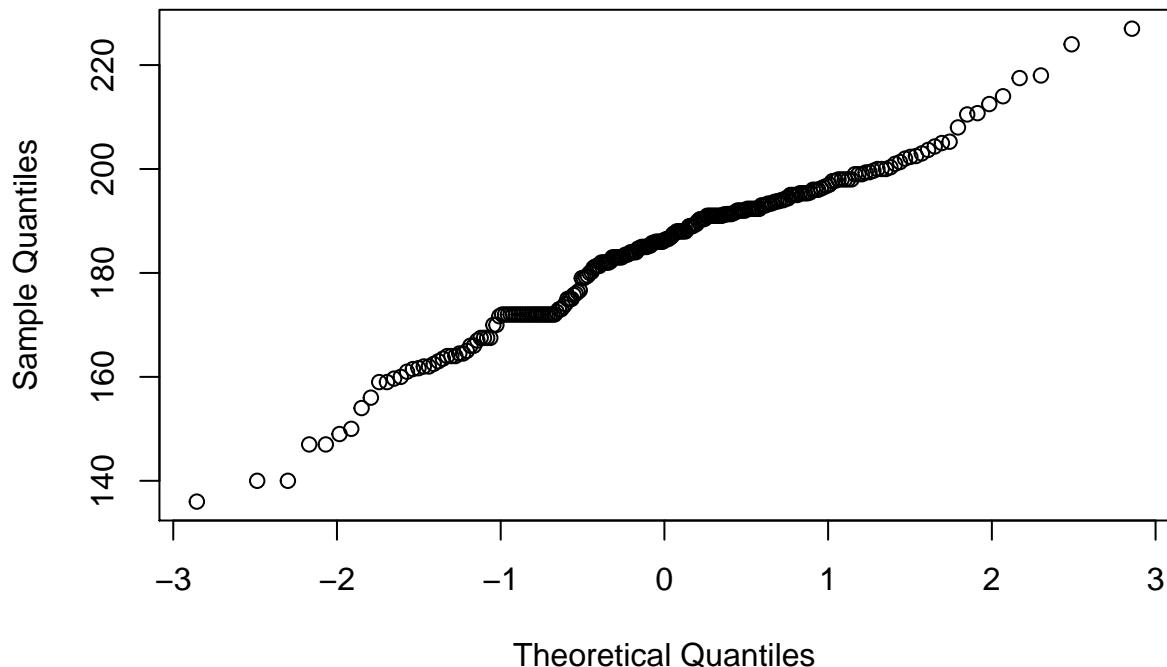
```
### UMBS ####
hist(umbs_sd_plot_origin$julian_min)
```

Histogram of umbs_sd_plot_origin\$julian_min



```
qqnorm(umbs_sd_plot_origin$julian_min)
```

Normal Q-Q Plot



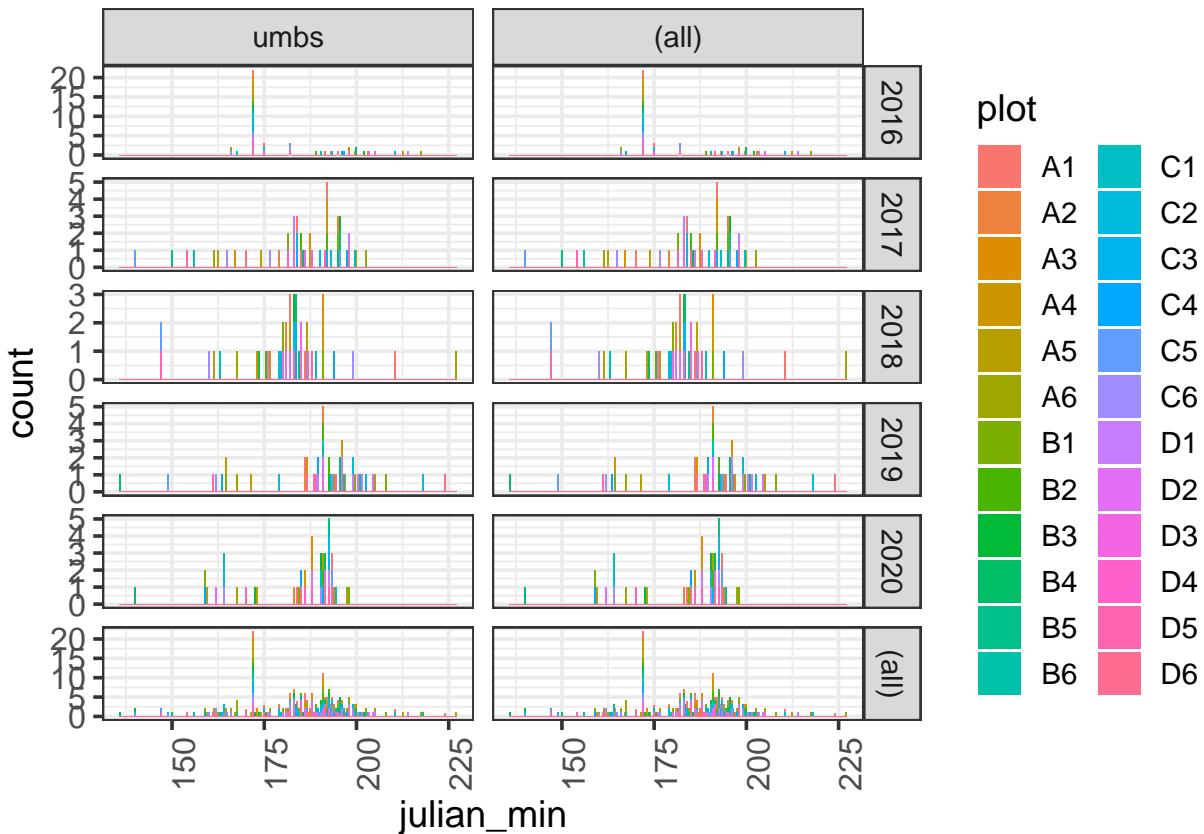
```

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

##
## Shapiro-Wilk normality test
##
## data: umbs_sd_plot_origin$julian_min
## W = 0.97337, p-value = 0.000226

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

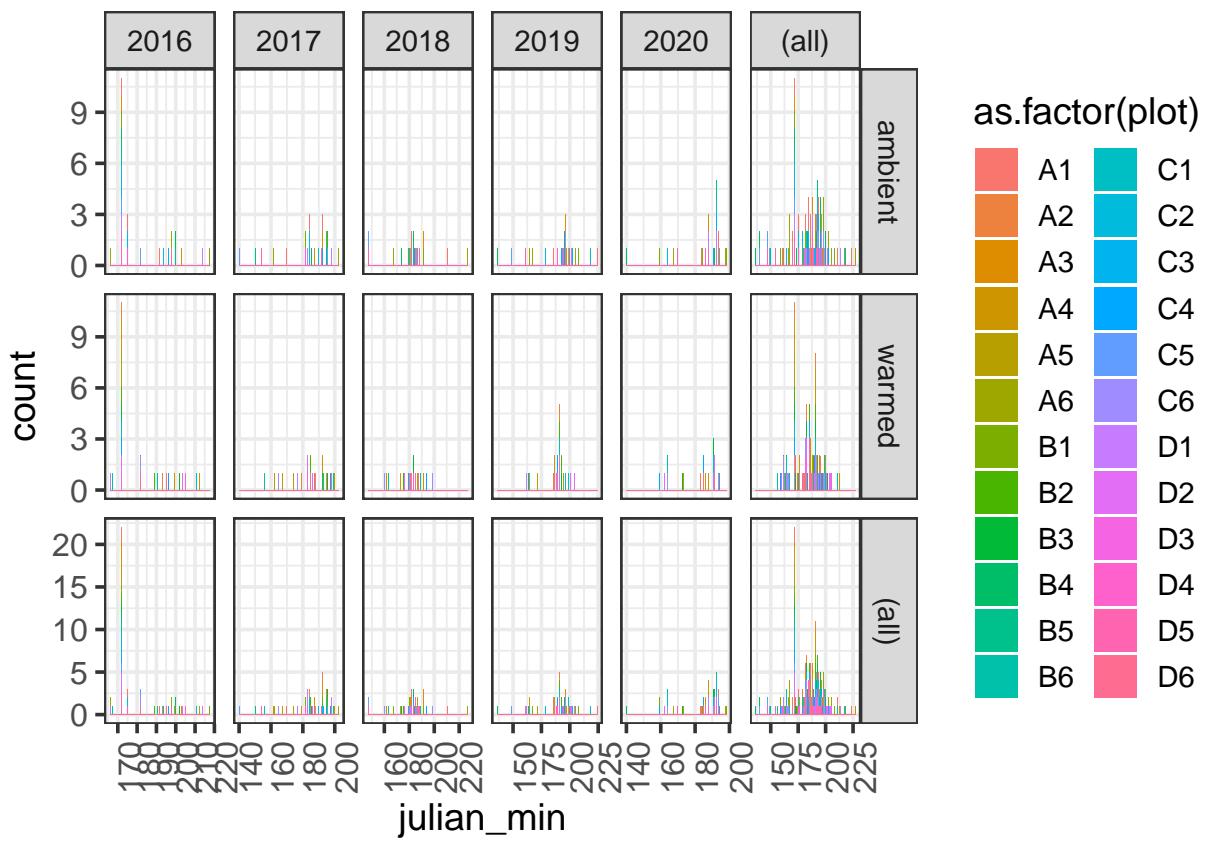
```



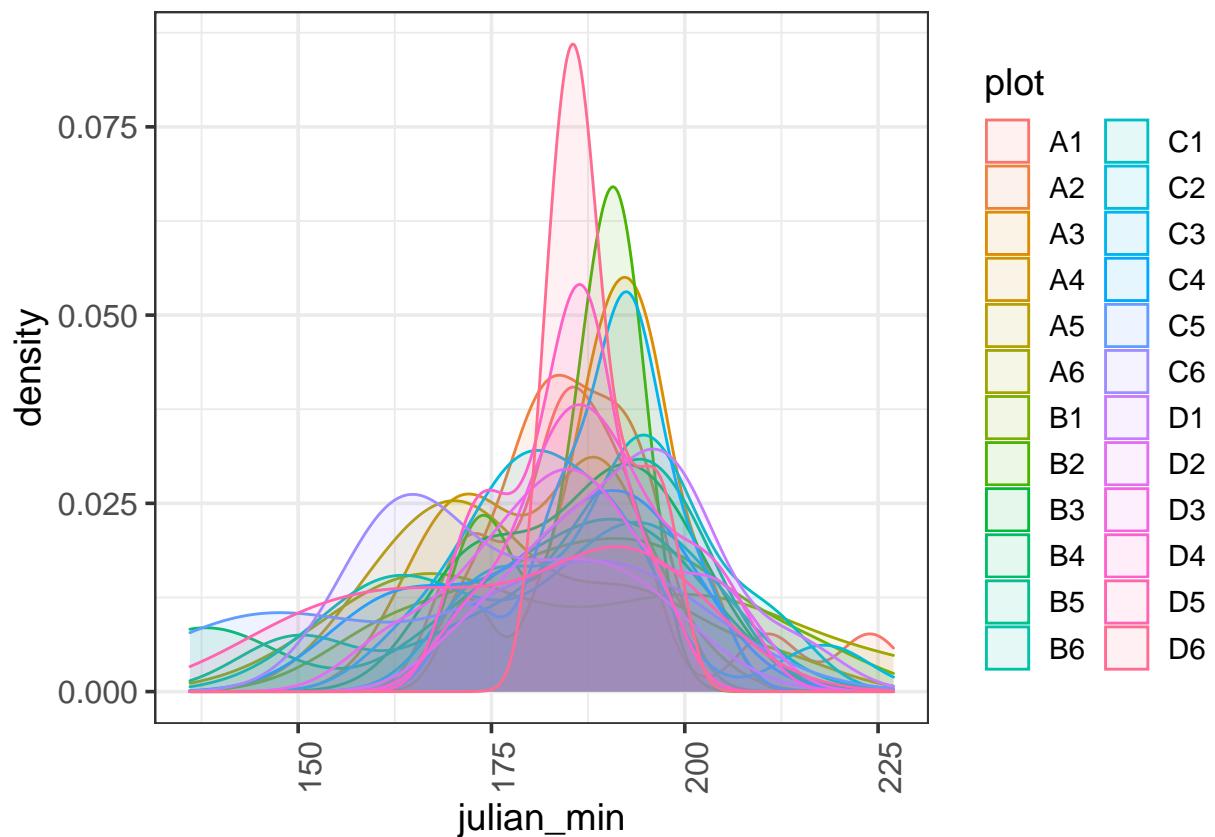
```

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

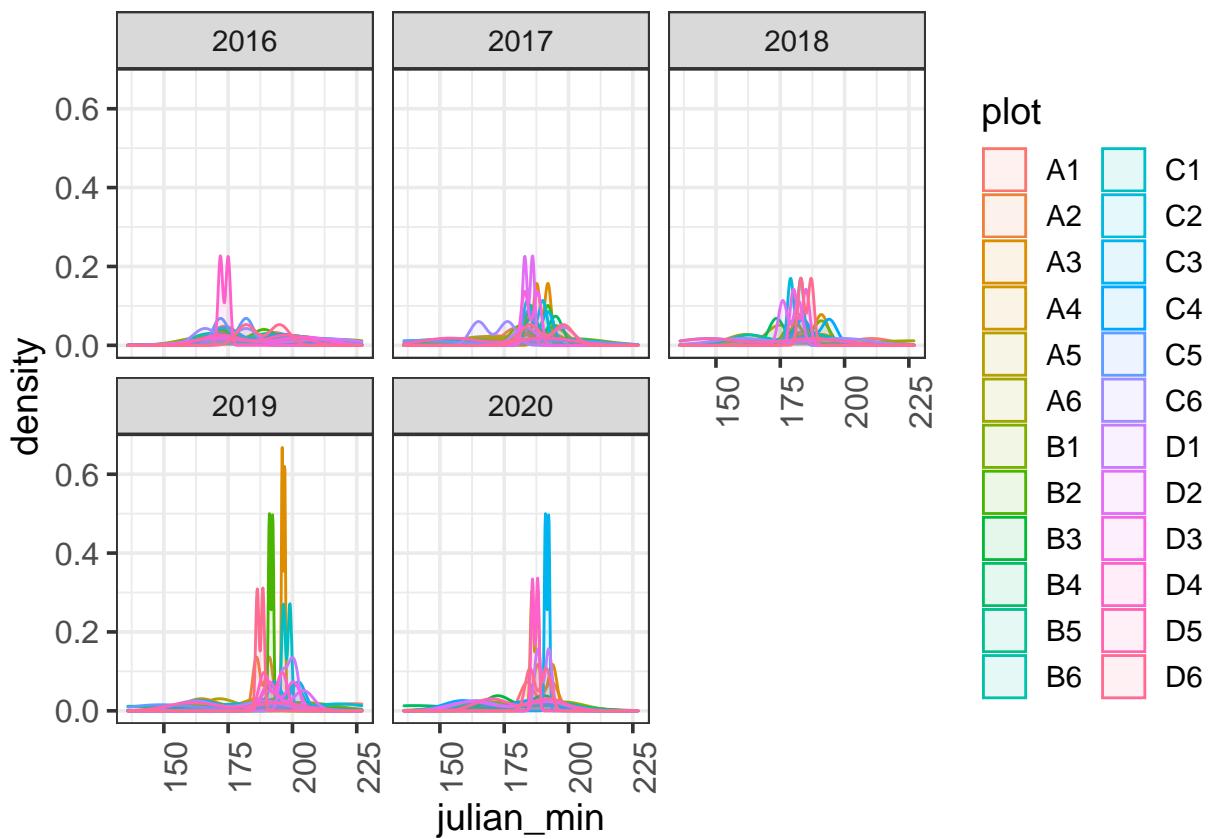
```



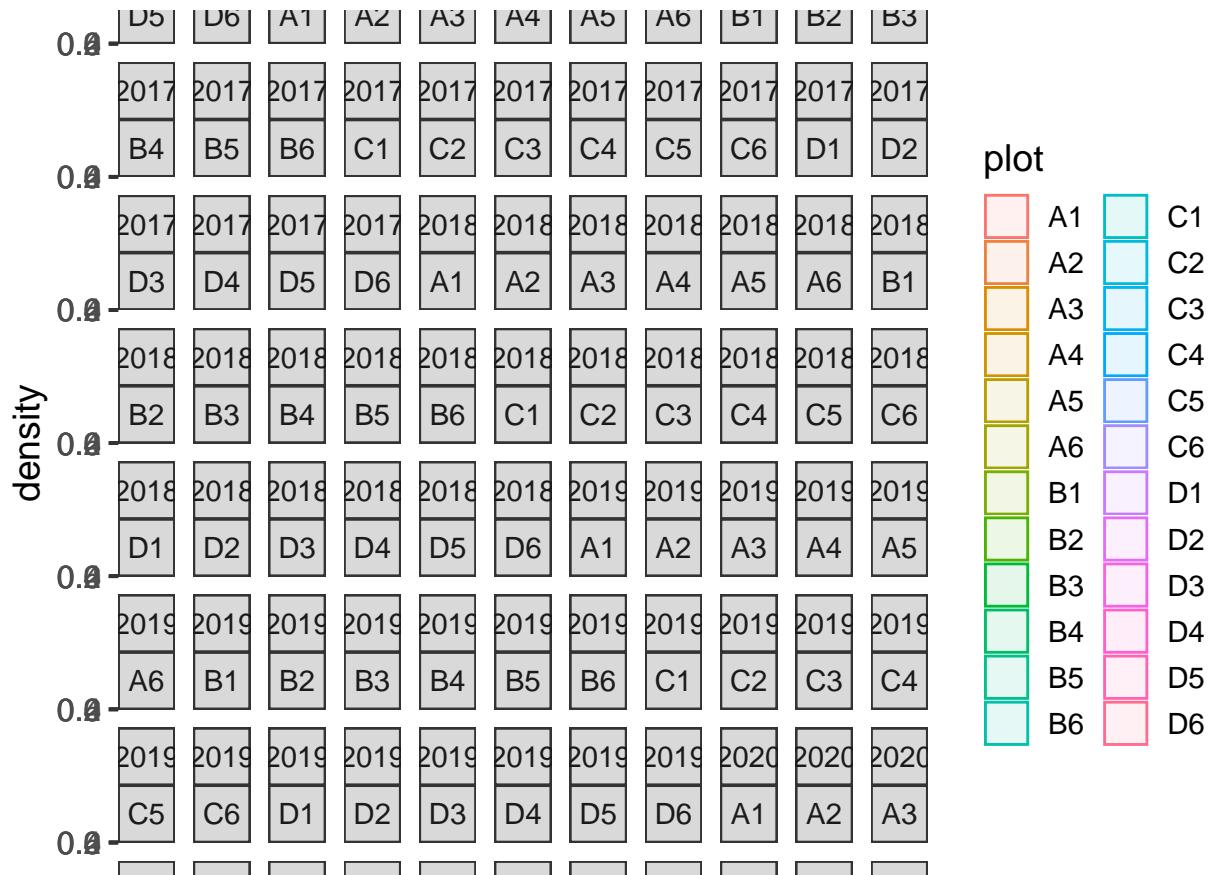
```
ggplot(umbss_sd_plot_origin, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

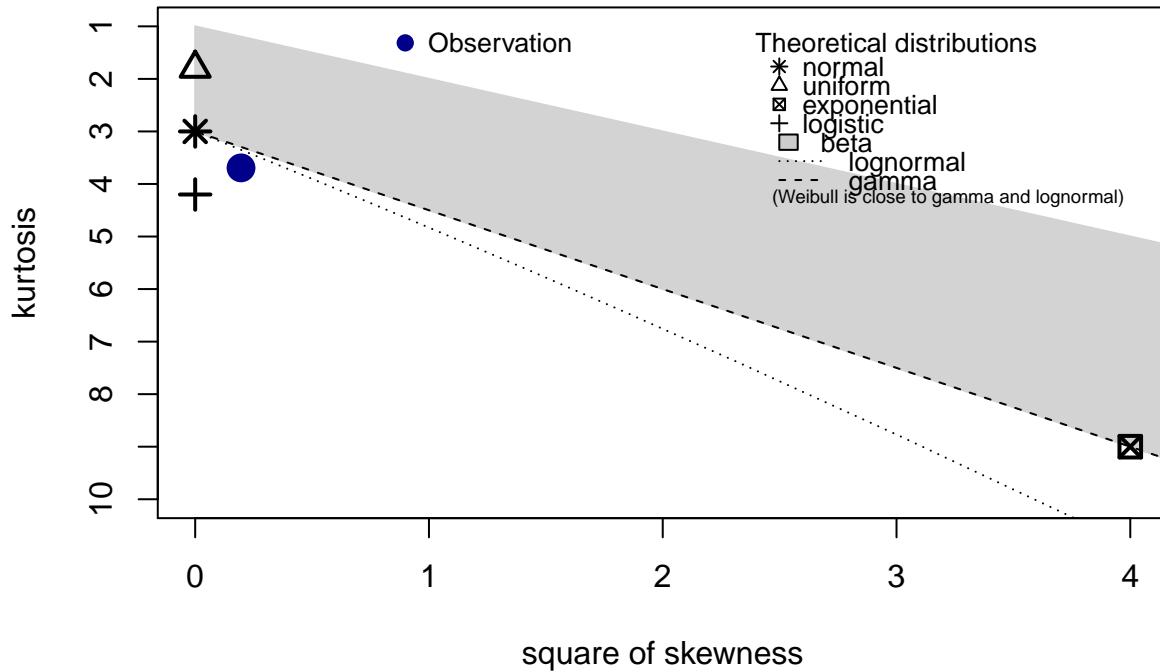


```
ggplot(umbs_sd_plot_origin, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



```
# Exploring distributions for these data:  
descdist(umbs_sd_plot_origin$julian_min, discrete = FALSE)
```

Cullen and Frey graph

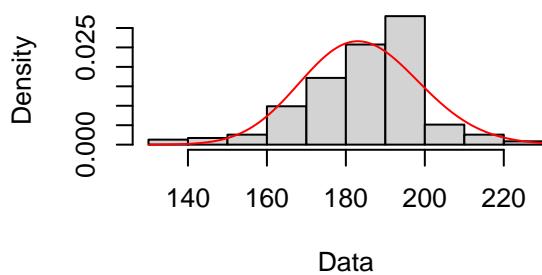
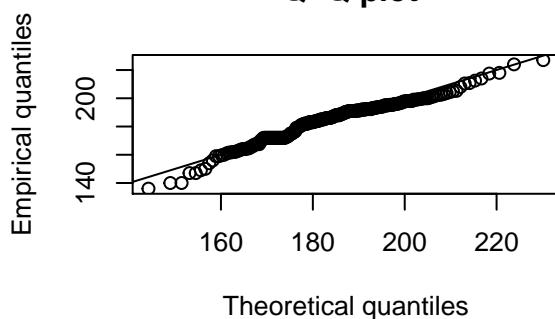
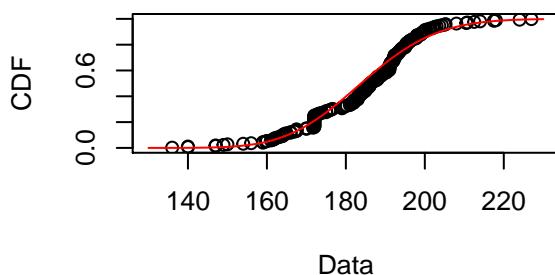
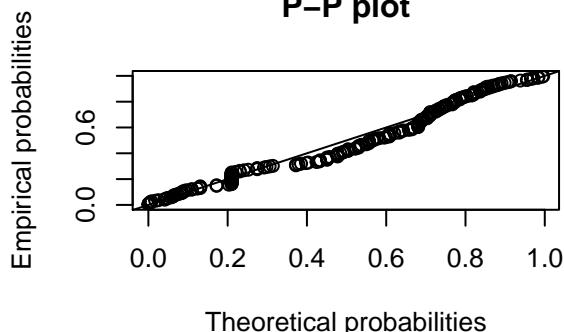


```

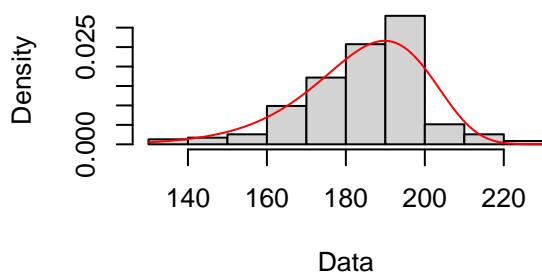
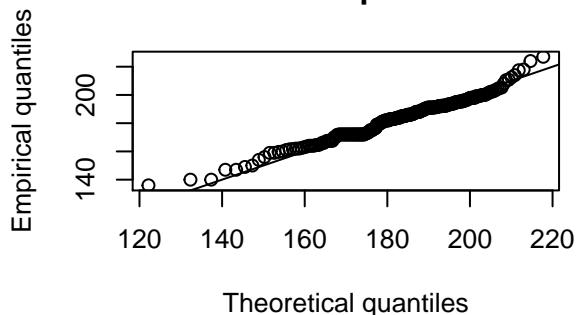
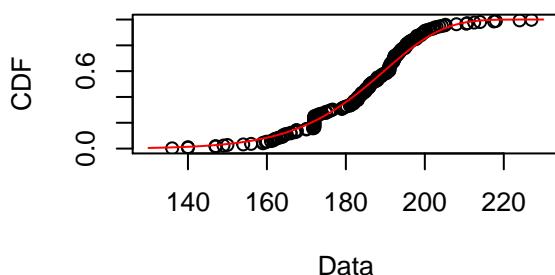
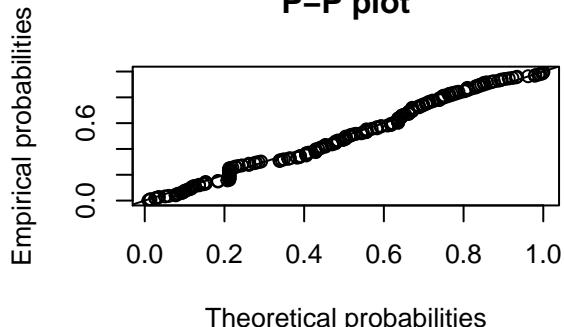
## summary statistics
## -----
## min: 136   max: 227
## median: 186.3333
## mean: 184.2609
## estimated sd: 14.82065
## estimated skewness: -0.4428369
## estimated kurtosis: 3.693731

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

```

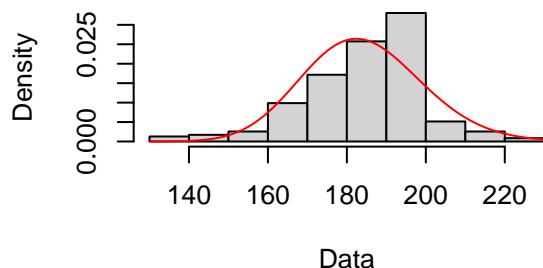
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

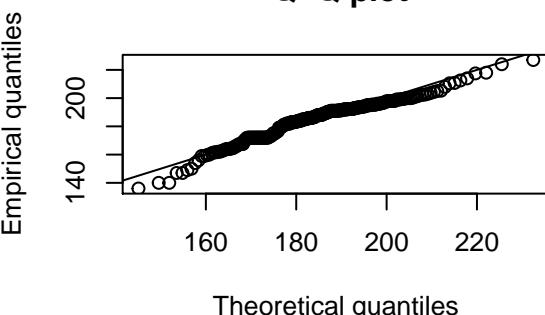
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

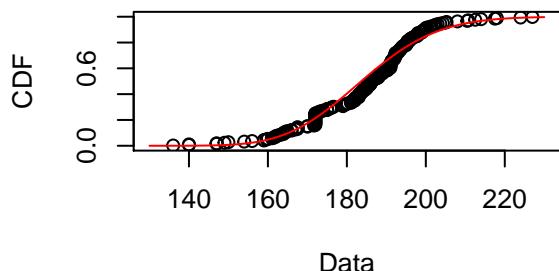
Empirical and theoretical dens.



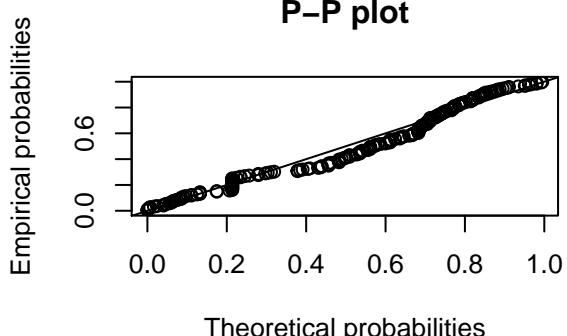
Q-Q plot



Empirical and theoretical CDFs

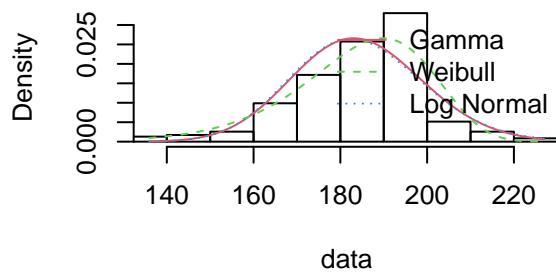


P-P plot

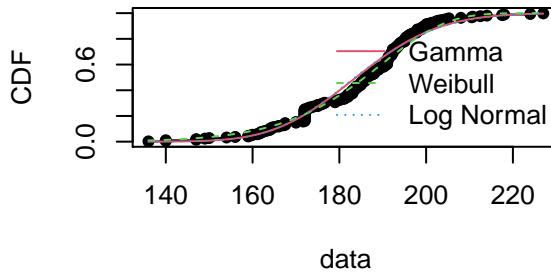


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

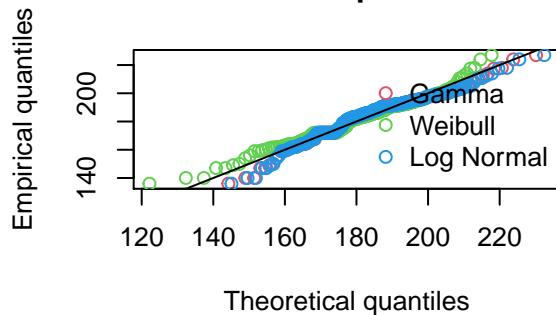
Histogram and theoretical densities



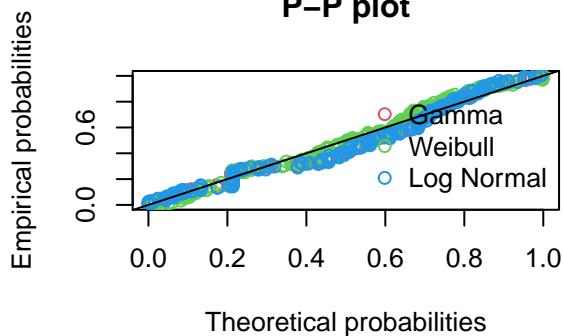
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```
# Goodness of fit comparisons across fits (can't include the sqrt normal bc it
# becomes diff response values)
gofstat(list(fit.gamma, fit.weibull, fit.ln), fitnames = c("Gamma", "Weibull", "Log Normal"))
```

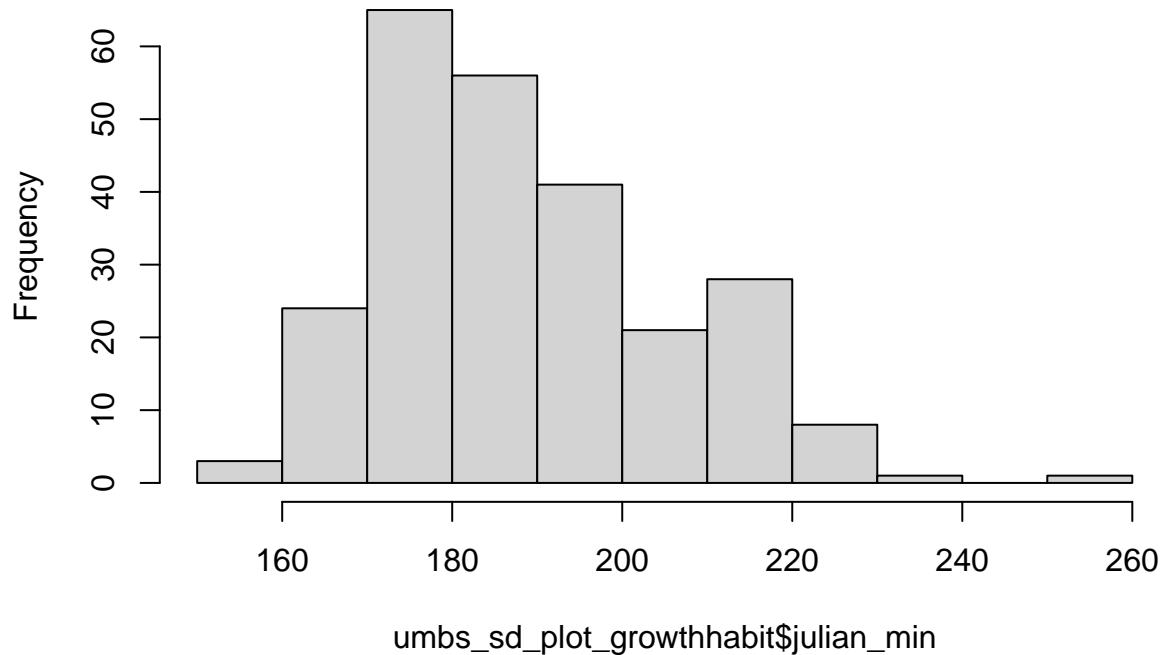
```
## Goodness-of-fit statistics
##                                     Gamma   Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.1039092 0.06597959 0.1095598
## Cramer-von Mises statistic  0.5724530 0.26609180 0.6386879
## Anderson-Darling statistic  3.0359342 1.87536361 3.3972035
##
## Goodness-of-fit criteria
##                                     Gamma   Weibull Log Normal
## Akaike's Information Criterion 1927.599 1924.254 1932.034
## Bayesian Information Criterion 1934.501 1931.156 1938.937
```

log normal looks like it's the best fit

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

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

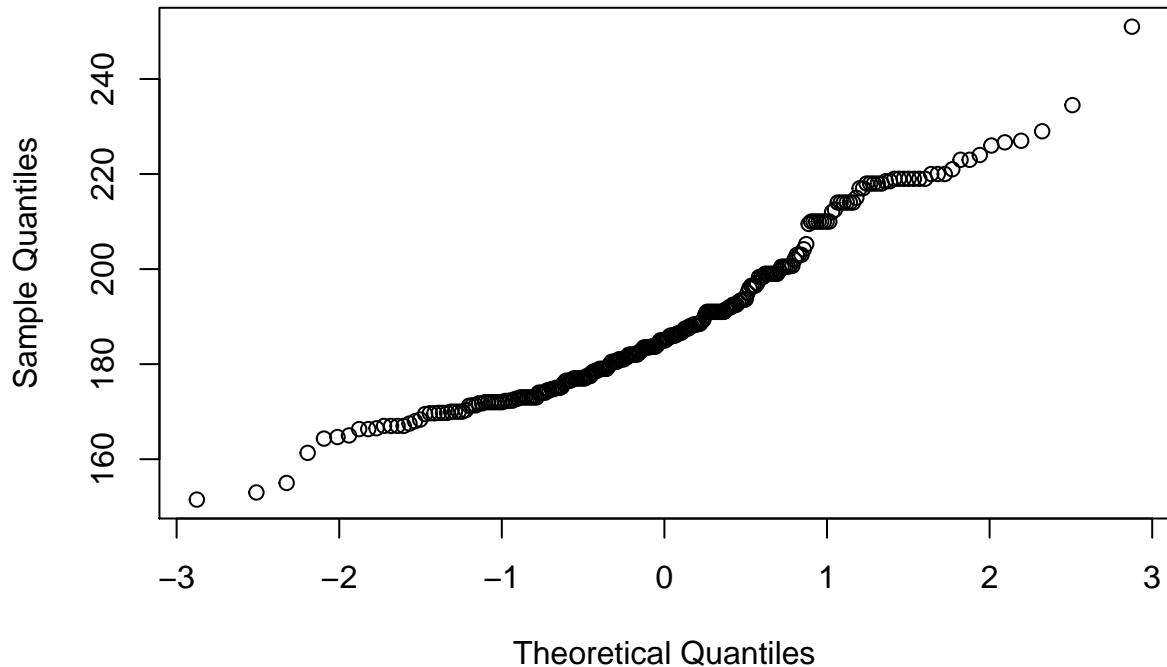
Histogram of umbs_sd_plot_growthhabit\$julian_min



umbs_sd_plot_growthhabit\$julian_min

```
qqnorm(umbss_sd_plot_growthhabit$julian_min)
```

Normal Q-Q Plot



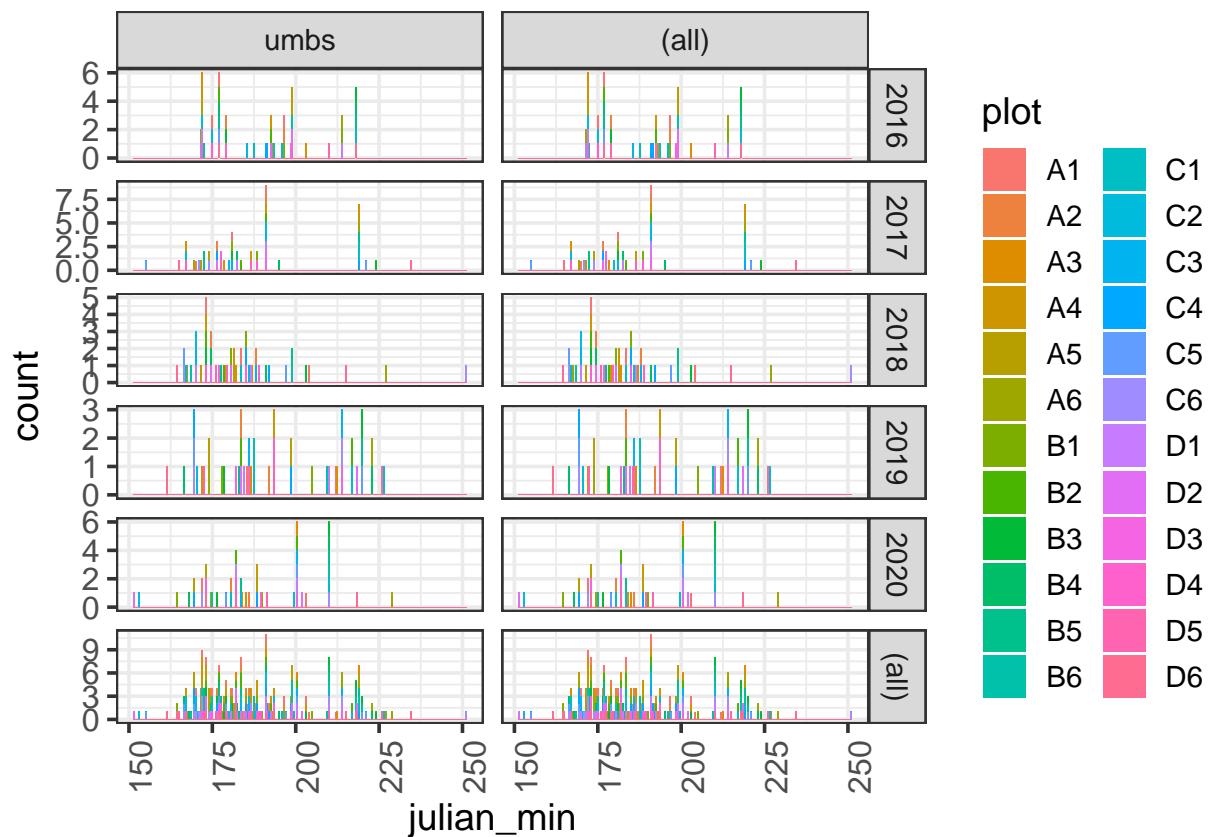
```

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

## 
## Shapiro-Wilk normality test
## 
## data: umbs_sd_plot_growthhabit$julian_min
## W = 0.94586, p-value = 5.916e-08

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

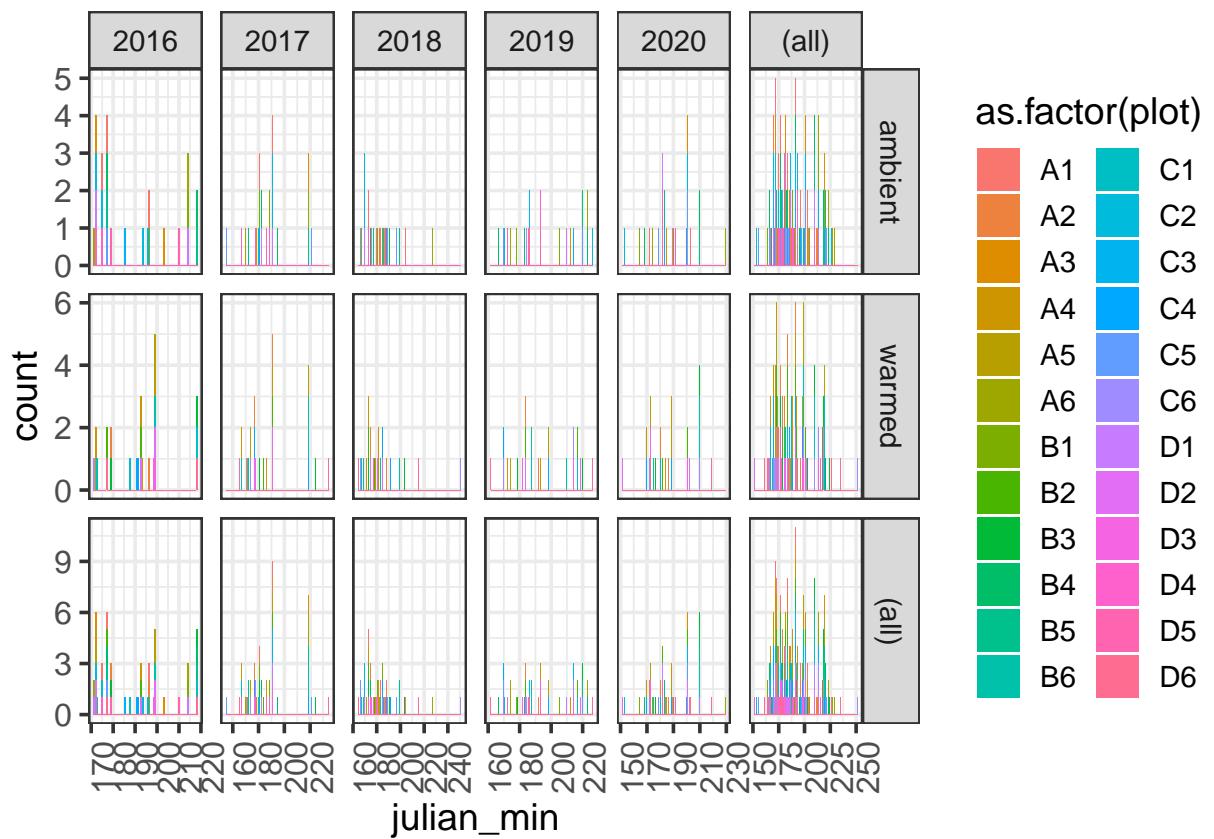
```



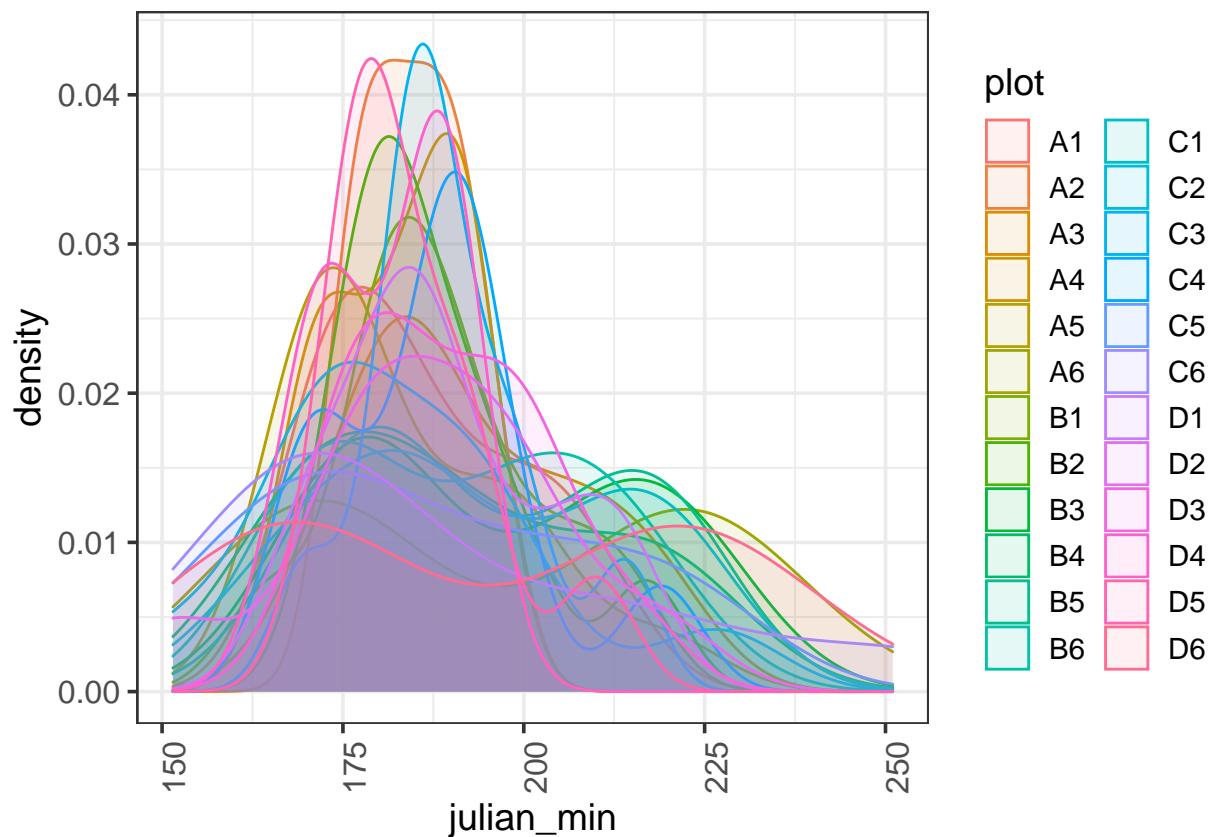
```

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

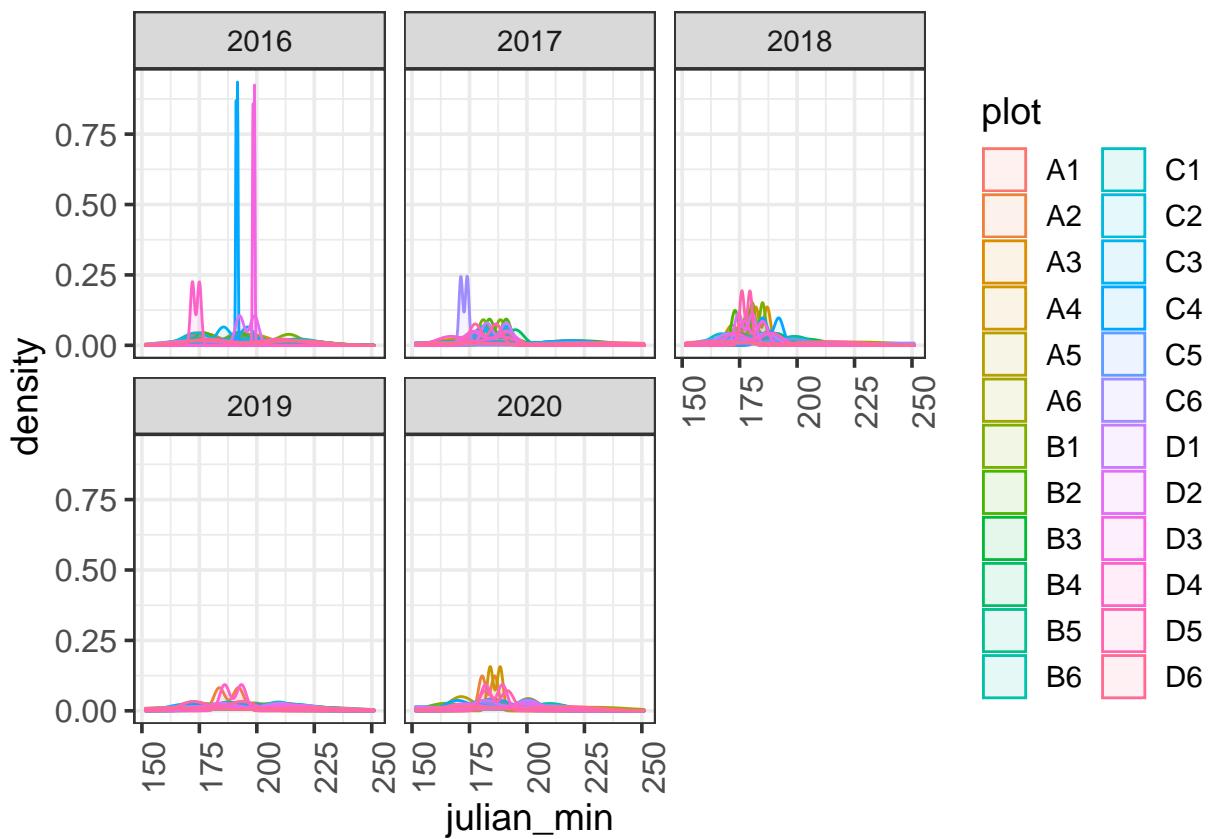
```



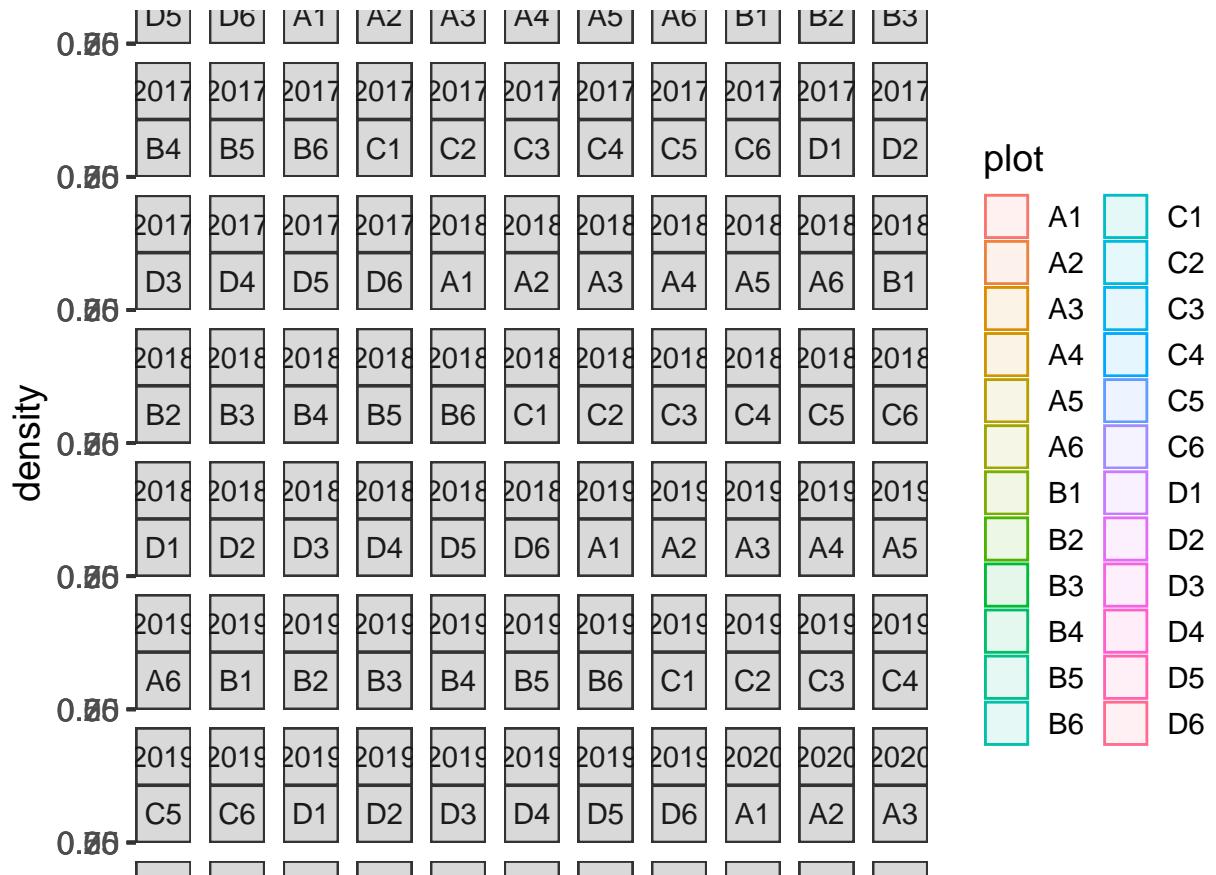
```
ggplot(umbs_sd_plot_growthhabit, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

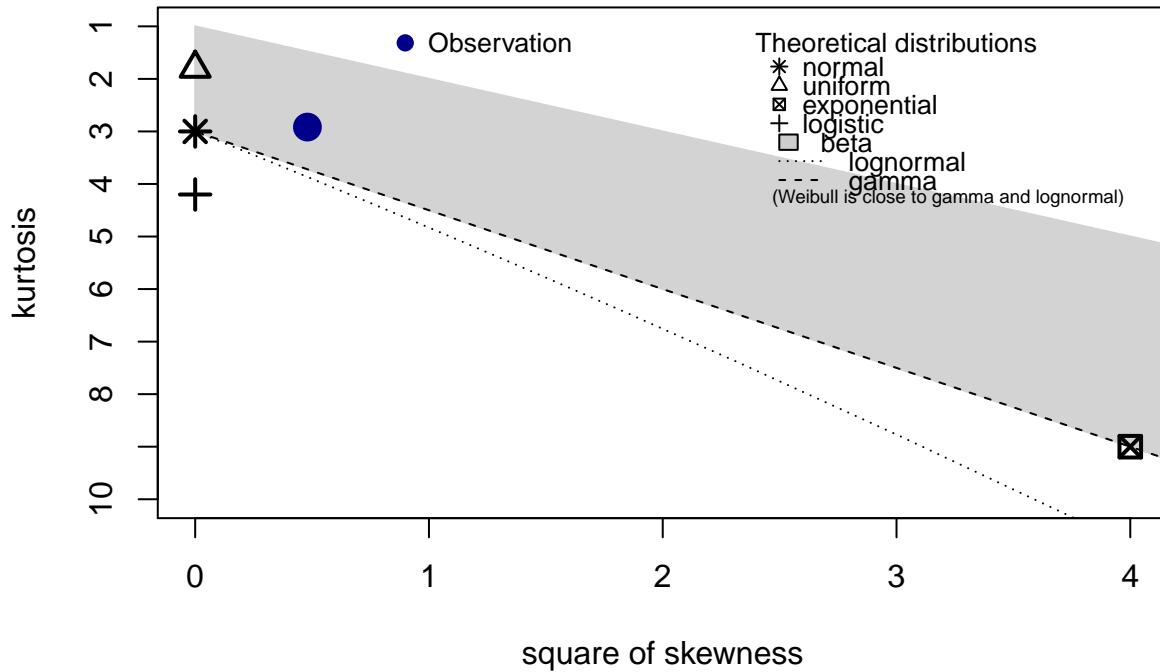


```
ggplot(umbs_sd_plot_growthhabit, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
  facet_wrap(~year + plot)
```



```
# Exploring distributions for these data:  
descdist(umbs_sd_plot_growthhabit$julian_min, discrete = FALSE)
```

Cullen and Frey graph

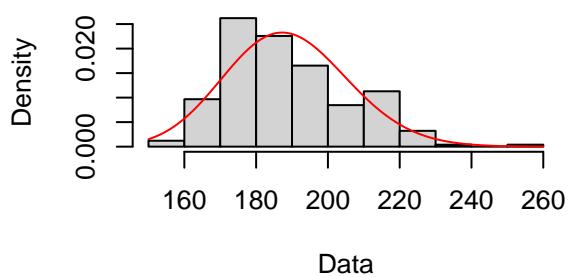
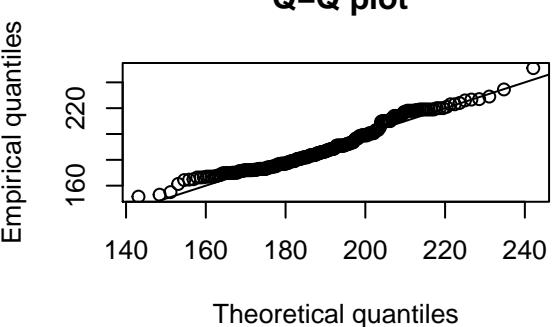
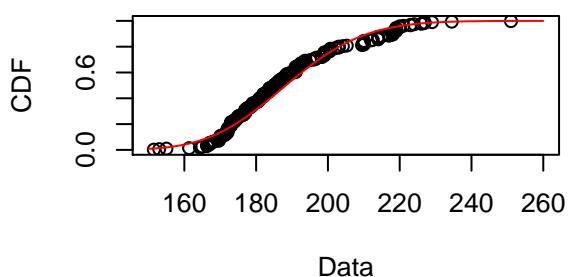
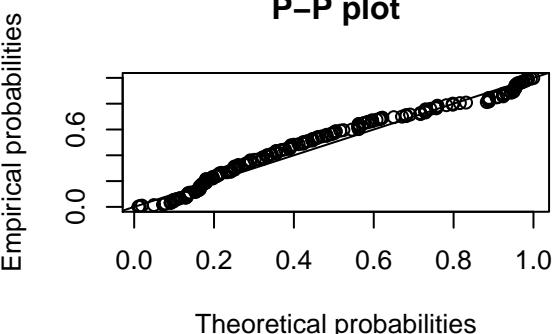


```

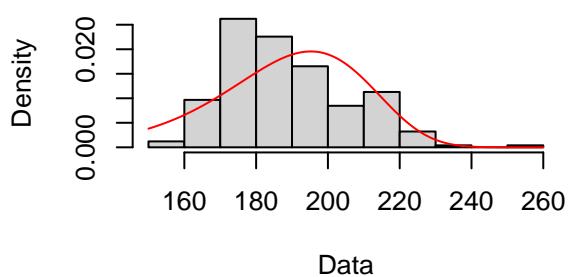
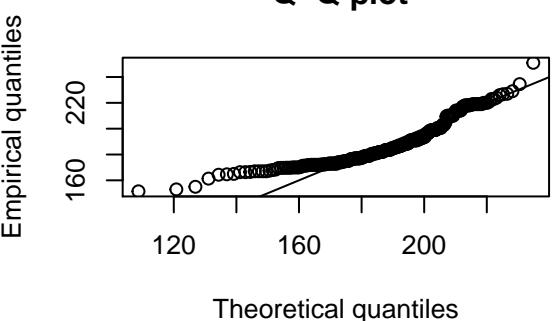
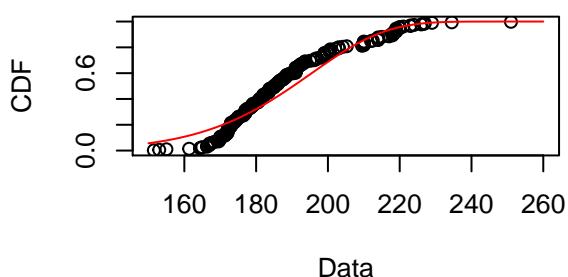
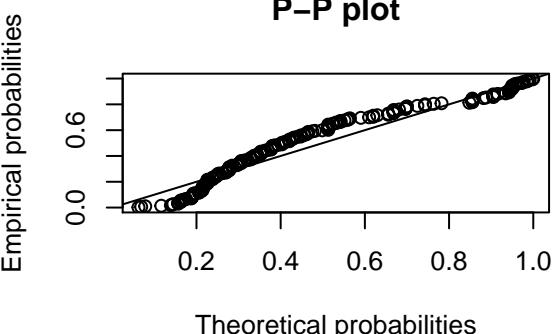
## summary statistics
## -----
## min: 151.5   max: 251
## median: 185
## mean: 188.8222
## estimated sd: 17.53223
## estimated skewness: 0.6929402
## estimated kurtosis: 2.917053

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

```

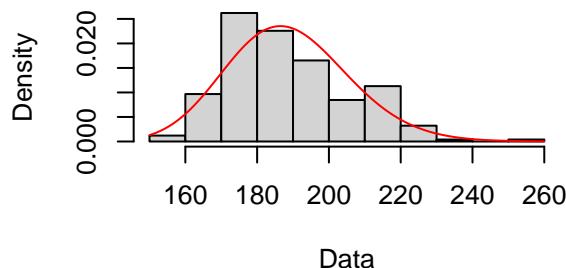
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

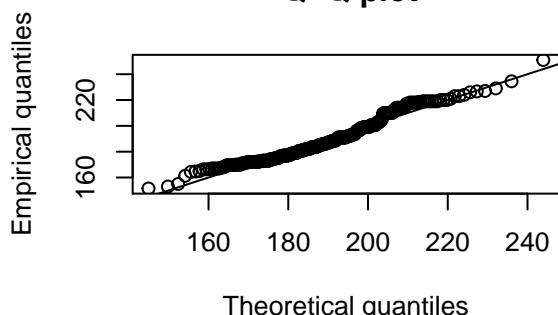
Empirical and theoretical dens.**Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

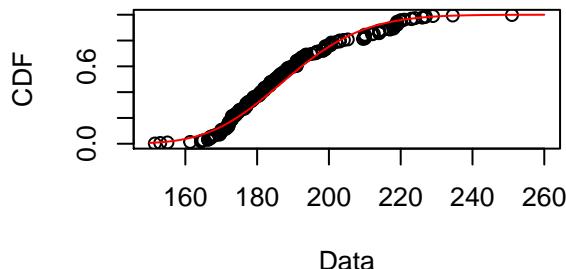
Empirical and theoretical dens.



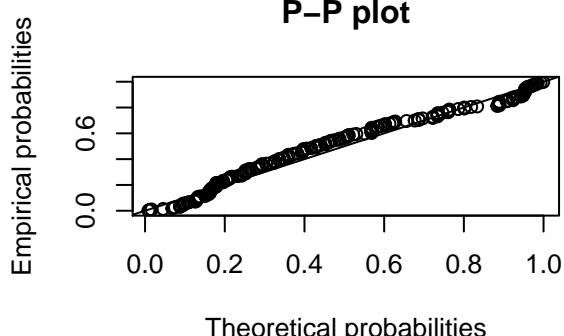
Q–Q plot



Empirical and theoretical CDFs

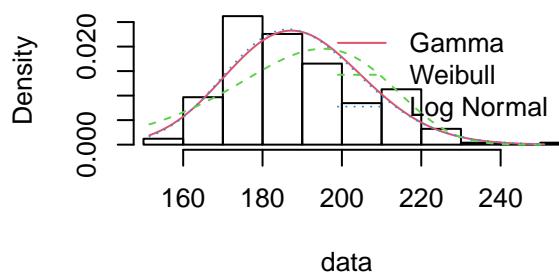


P–P plot

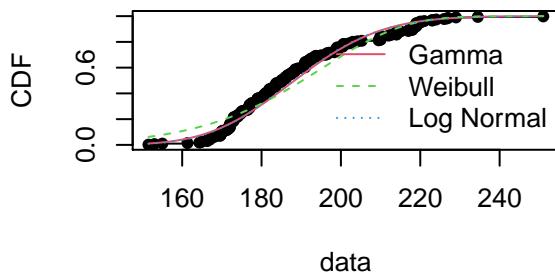


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

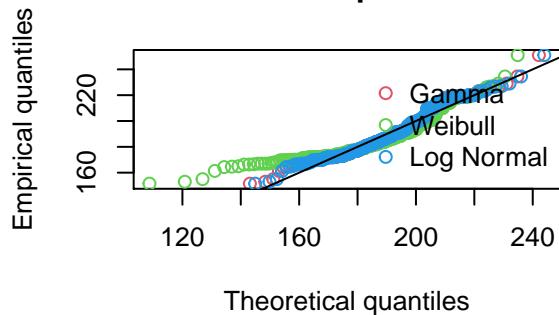
Histogram and theoretical densities



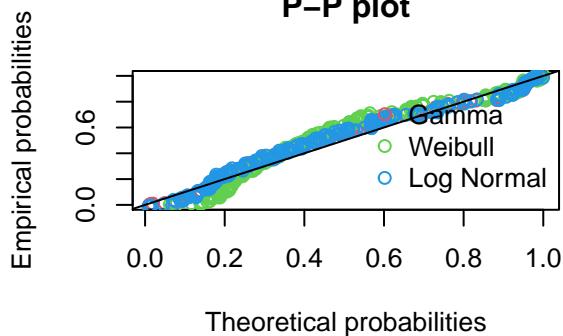
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```
# Goodness of fit comparisons across fits (can't include the sqrt normal bc it
# becomes diff response values)
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 0.08690531 0.132212 0.08234449
## Cramer-von Mises statistic   0.64637436 1.414216 0.57418930
## Anderson-Darling statistic   4.07325427 8.351621 3.67378523
##
## Goodness-of-fit criteria
##                                     Gamma  Weibull Log Normal
## Akaike's Information Criterion 2117.757 2178.627 2113.772
## Bayesian Information Criterion 2124.784 2185.654 2120.798
```

Log normal looks like it's the best fit

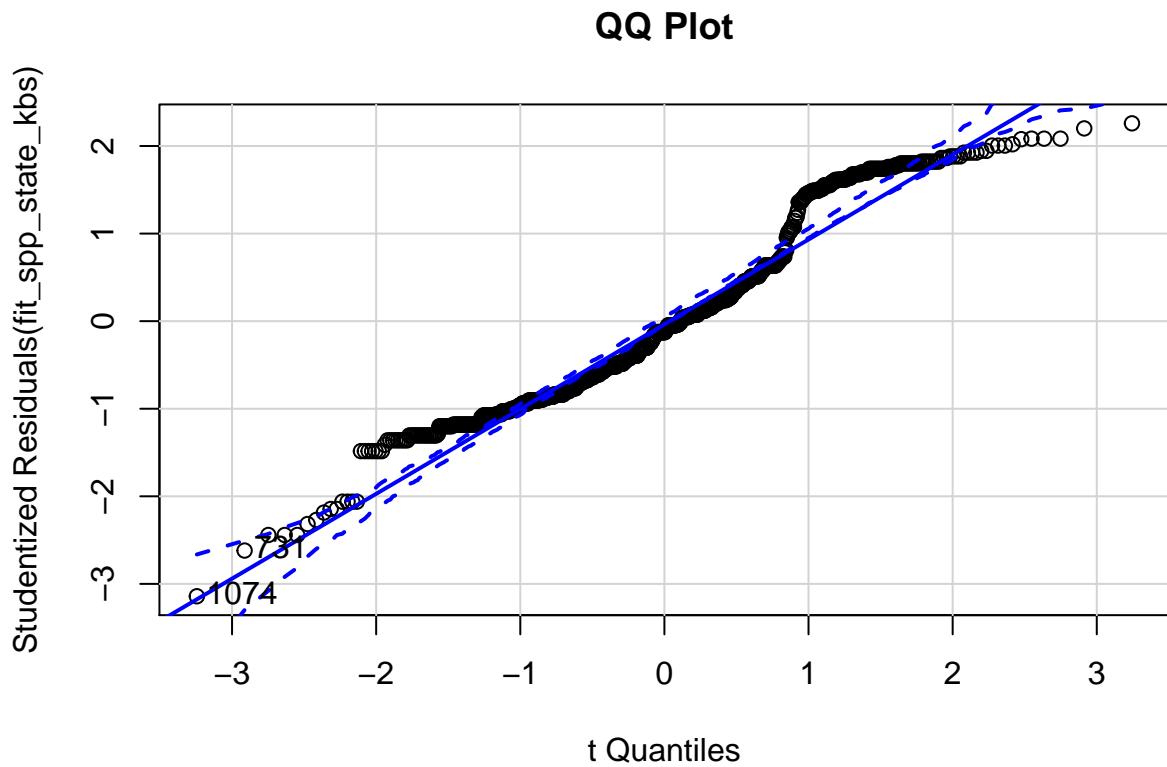
Leverage plots and detecting Outliers. <https://www.statmethods.net/stats/rdiagnostics.html>

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
## 1074 -3.141211      0.0017434        NA
```

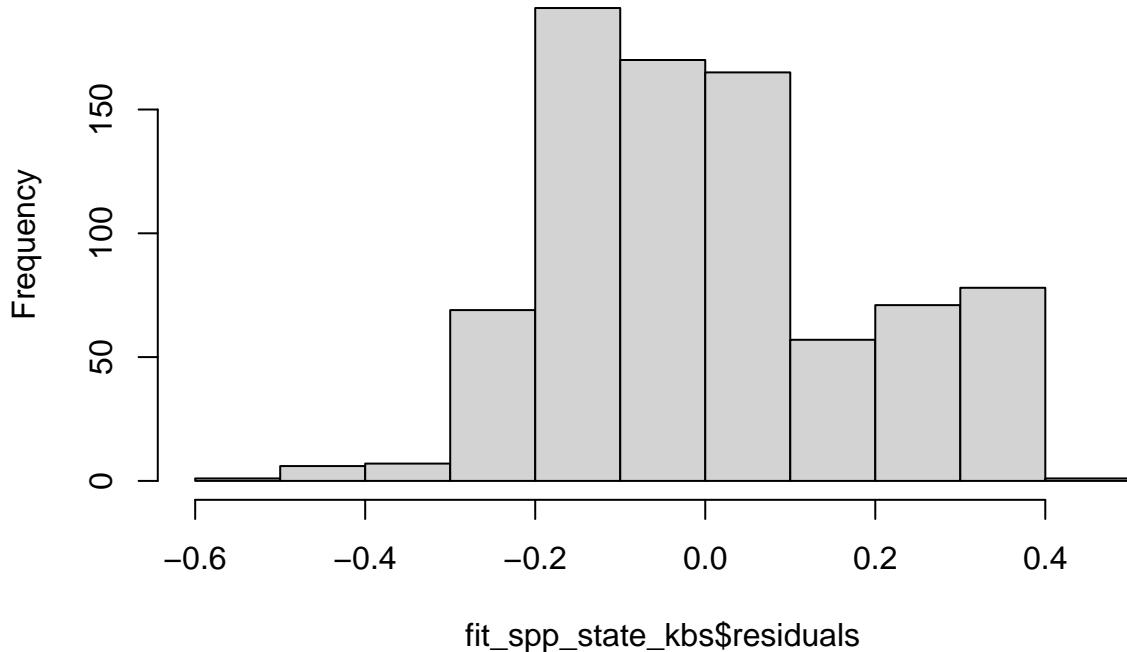
```
qqPlot(fit_spp_state_kbs, main = "QQ Plot")
```



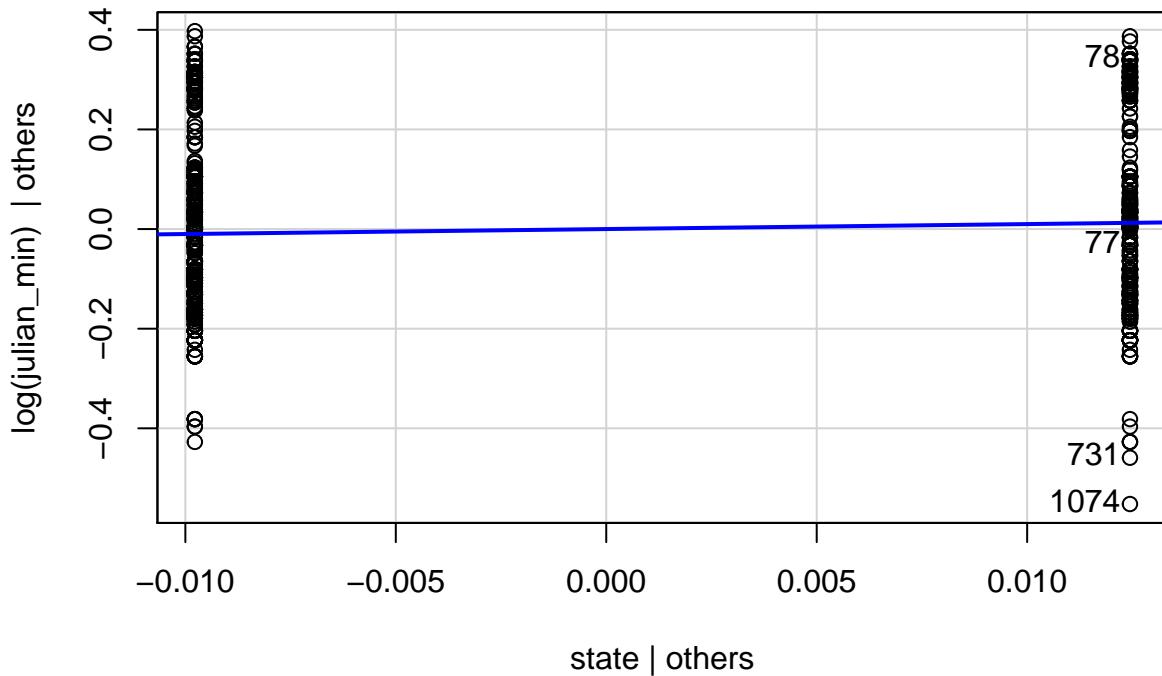
```
##   731 1074
##   433 641
```

```
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)
```

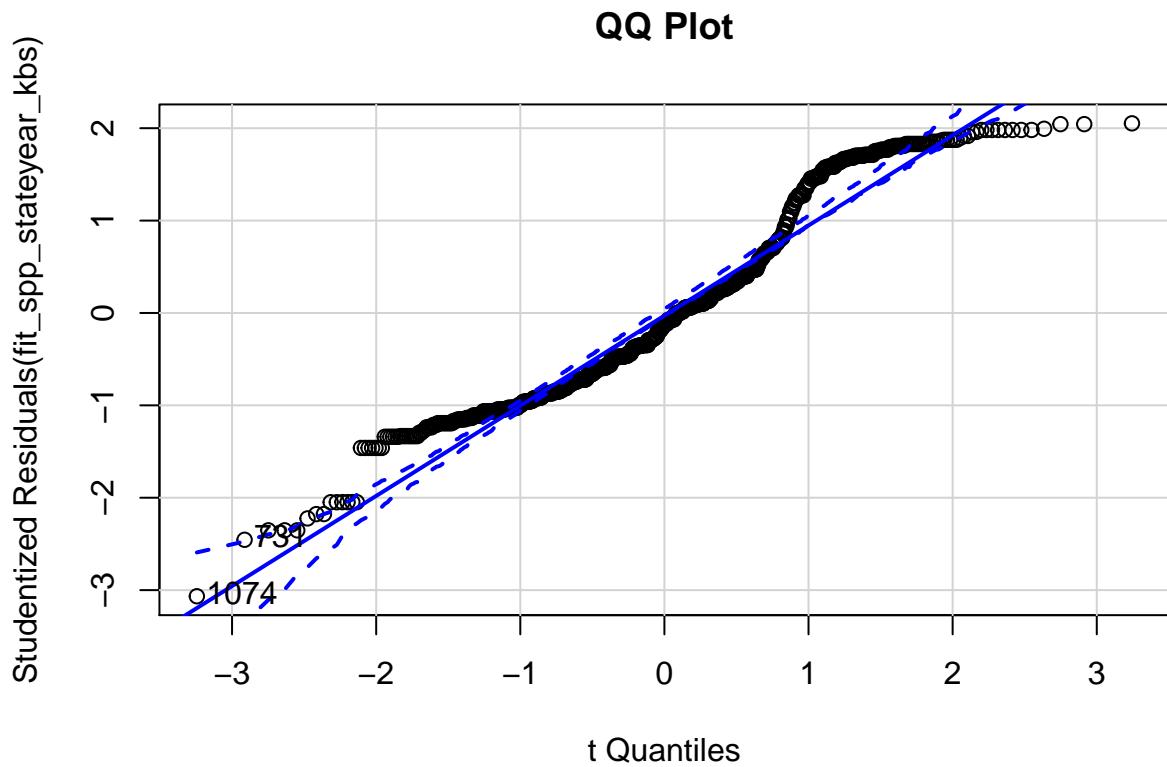
```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
## -----
##      Test          Statistic       pvalue
## -----
## Shapiro-Wilk      0.9545       0.0000
## Kolmogorov-Smirnov 0.0899       0.0000
## Cramer-von Mises 183.0872      0.0000
## Anderson-Darling   13.138        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
## 1074 -3.066401            0.0022388        NA
```

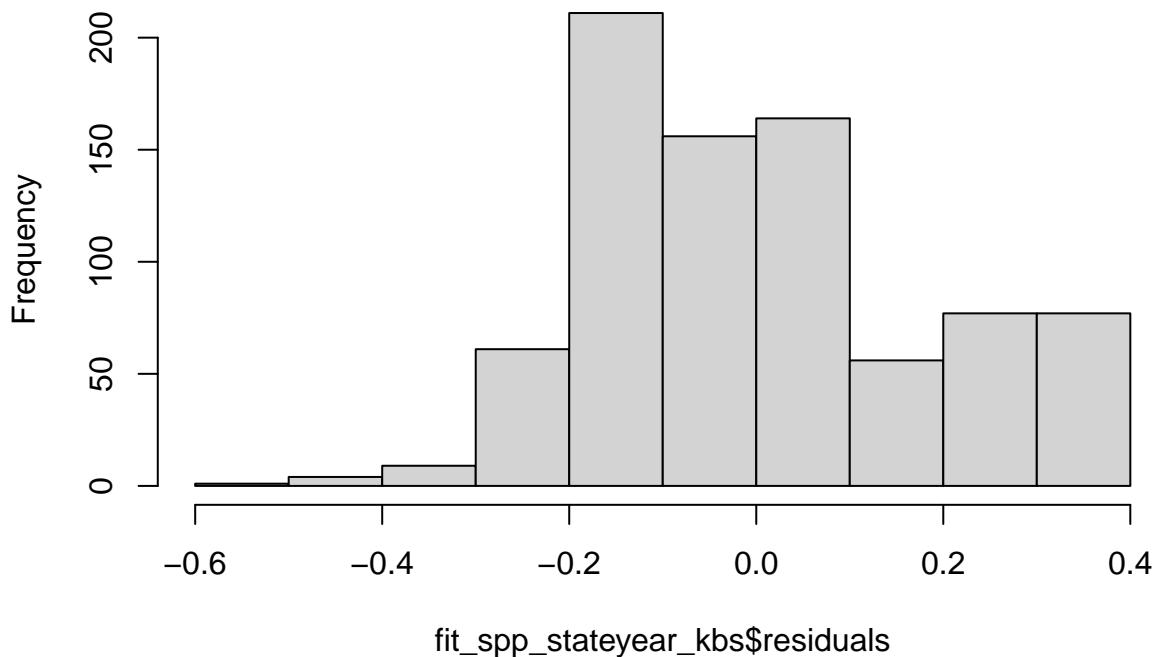
```
qqPlot(fit_spp_stateyear_kbs, main = "QQ Plot")
```



```
##   731 1074
##   433 641
```

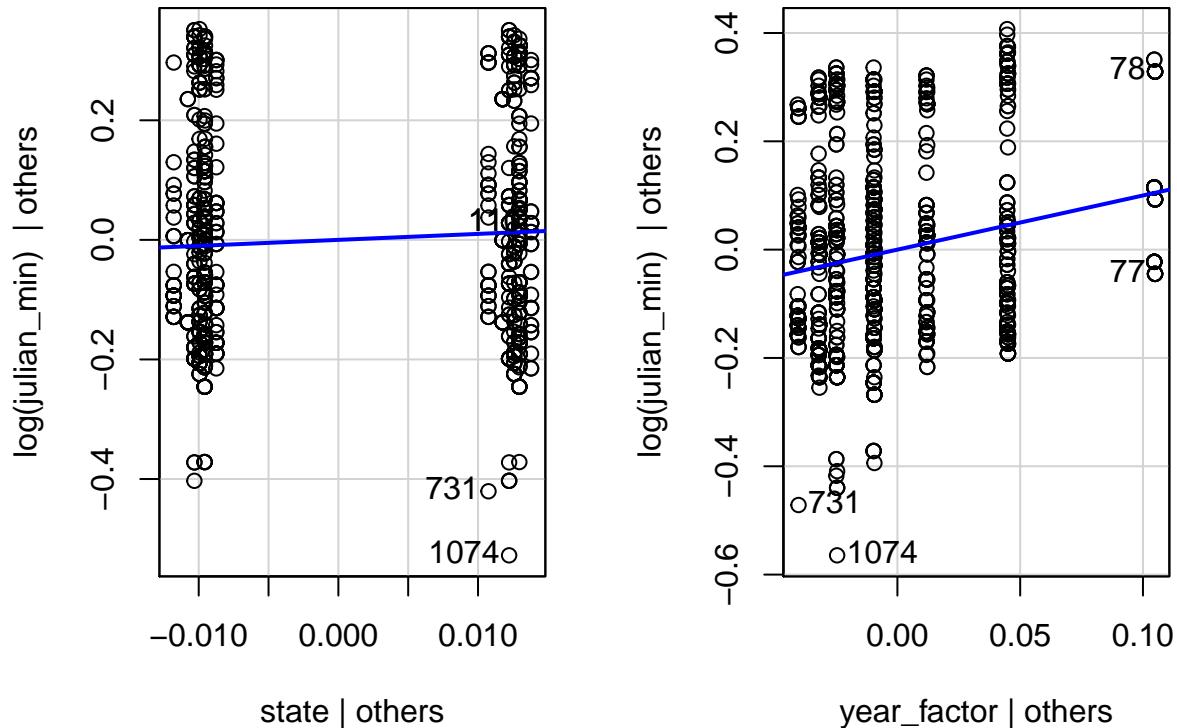
```
hist(fit_spp_stateyear_kbs$residuals)
```

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 0.0904    0.0000
## Cramer-von Mises   184.0342   0.0000
## Anderson-Darling    14.1737   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.

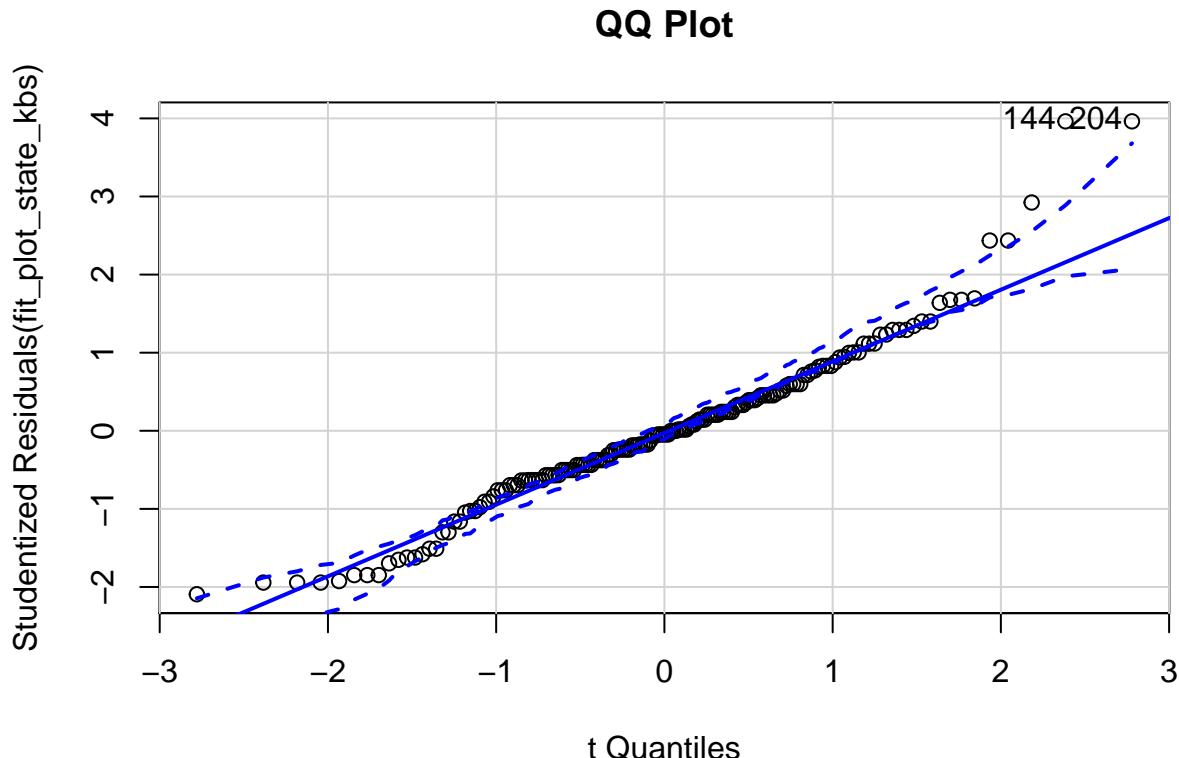
## Error in interact_plot(fit4, pred = year_factor, modx = state, mod2 = species): Focal predictor ("pr

# 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
## 144 3.962233          0.00011133    0.018258
## 204 3.962233          0.00011133    0.018258

qqPlot(fit_plot_state_kbs, main = "QQ Plot")

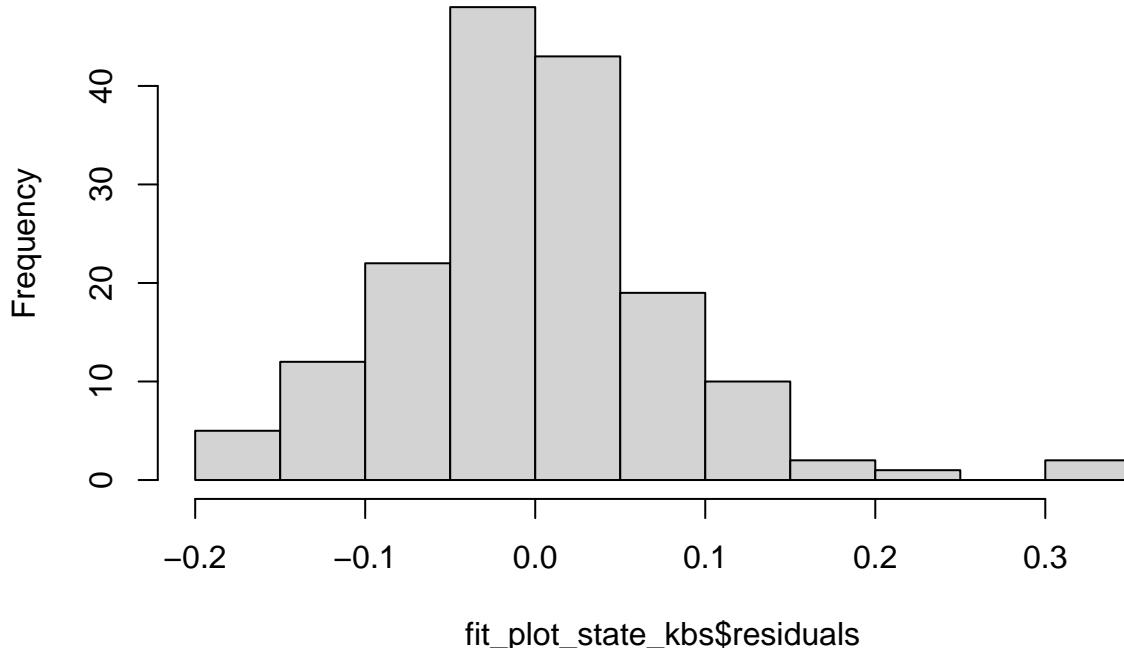
```



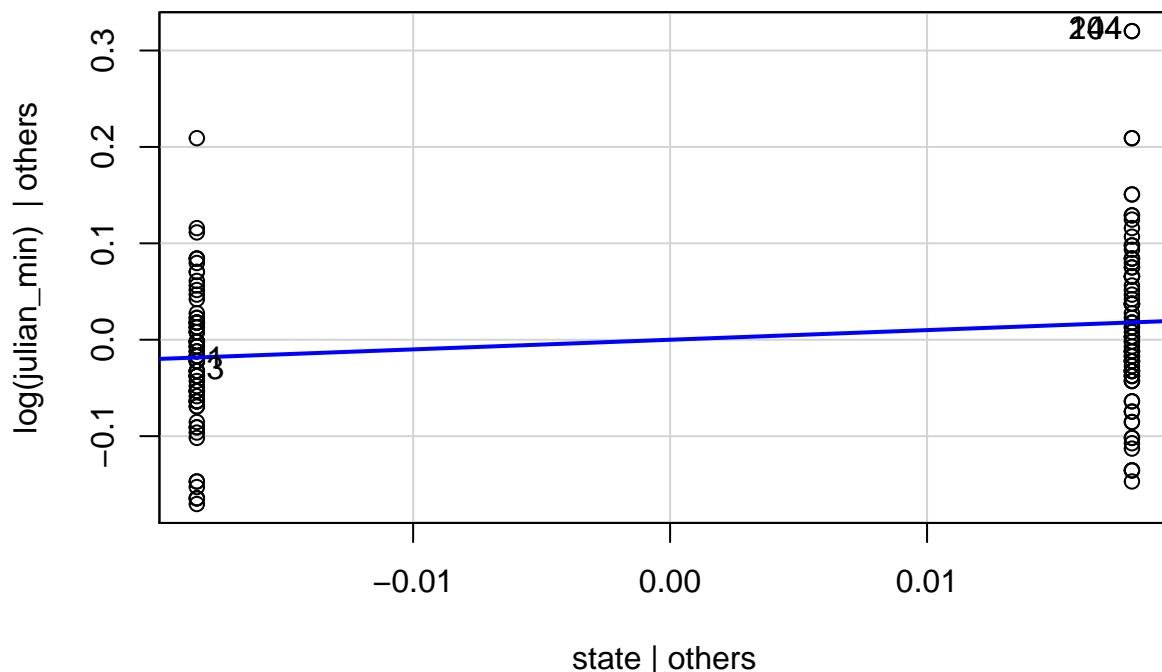
```
## 144 204  
## 84 119
```

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

Histogram of 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      0.9606    1e-04
## Kolmogorov-Smirnov   0.0689    0.4178
## Cramer-von Mises     46.7579    0.0000
## Anderson-Darling      1.1426    0.0053
## -----
# 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

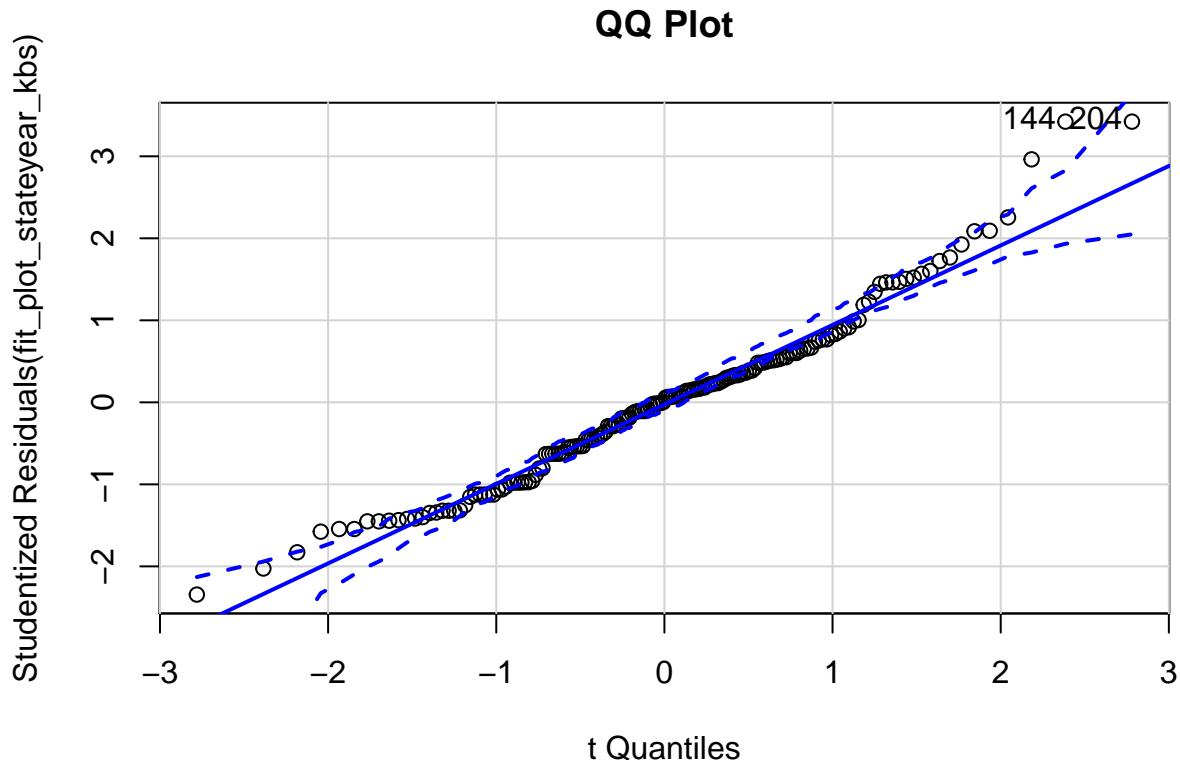
```

```

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##          rstudent unadjusted p-value Bonferroni p
## 1074 -3.066401           0.0022388       NA

```

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



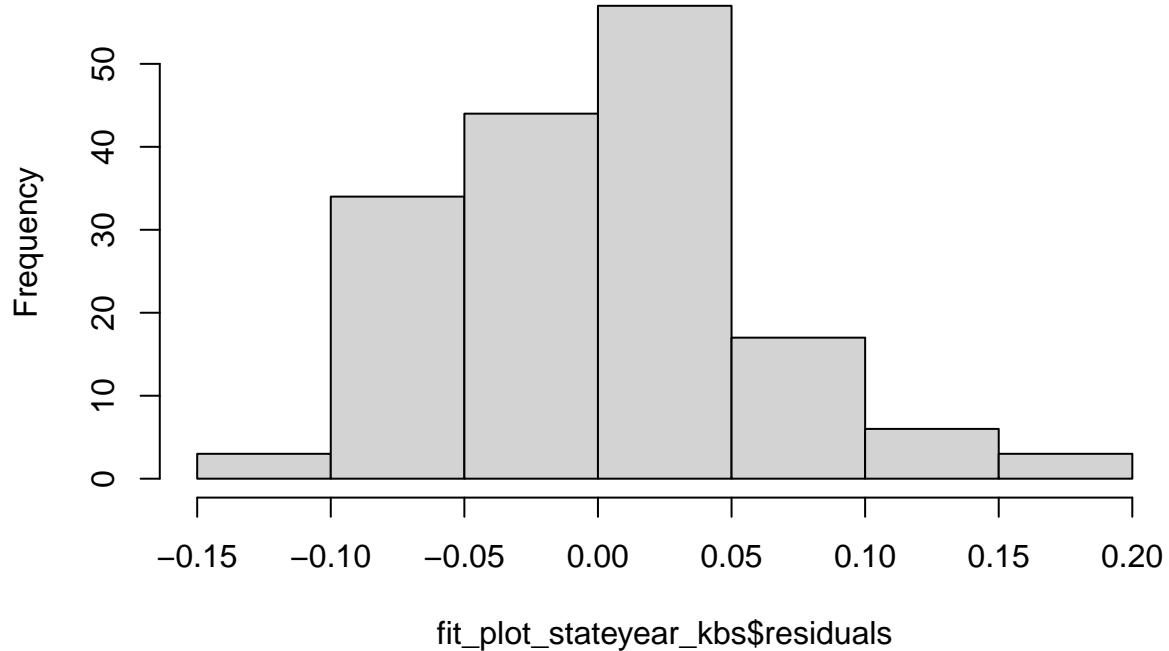
```

## 144 204
## 84 119

```

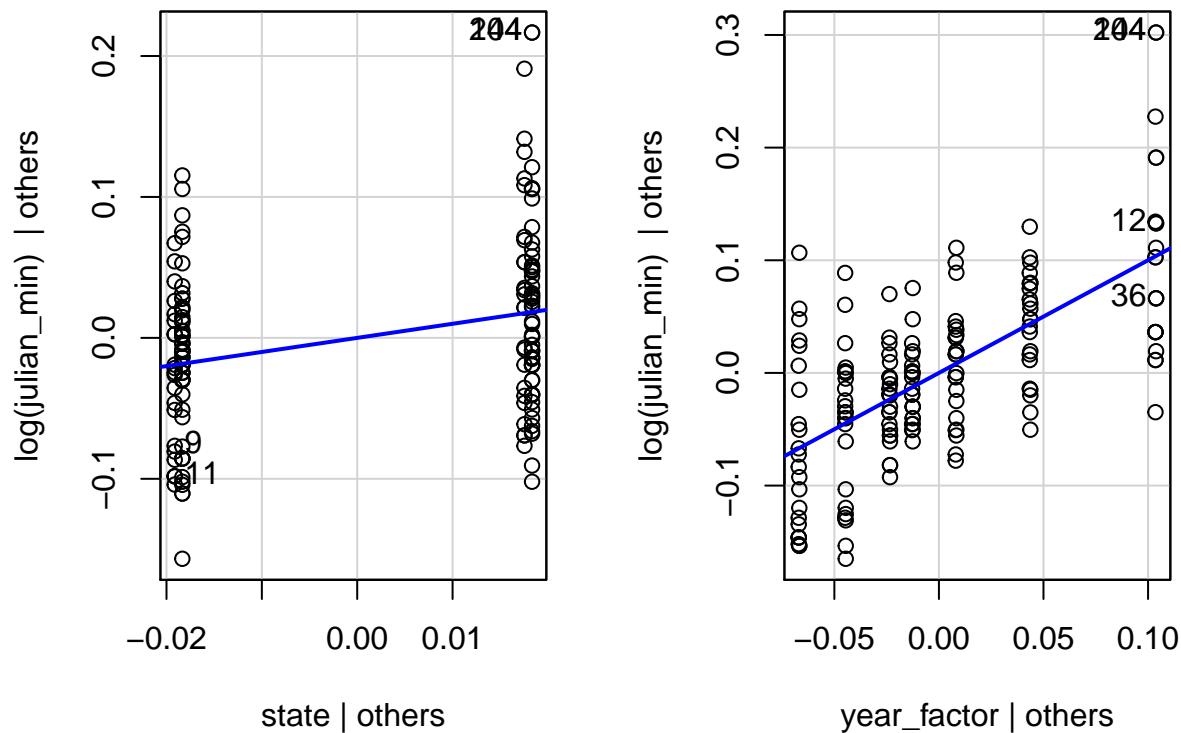
```
hist(fit_plot_stateyear_kbs$residuals)
```

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      0.9746    0.0042
## Kolmogorov-Smirnov   0.0635    0.5238
## Cramer-von Mises   48.3973   0.0000
## Anderson-Darling    0.8627    0.0262
## -----
```

```
# 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
```

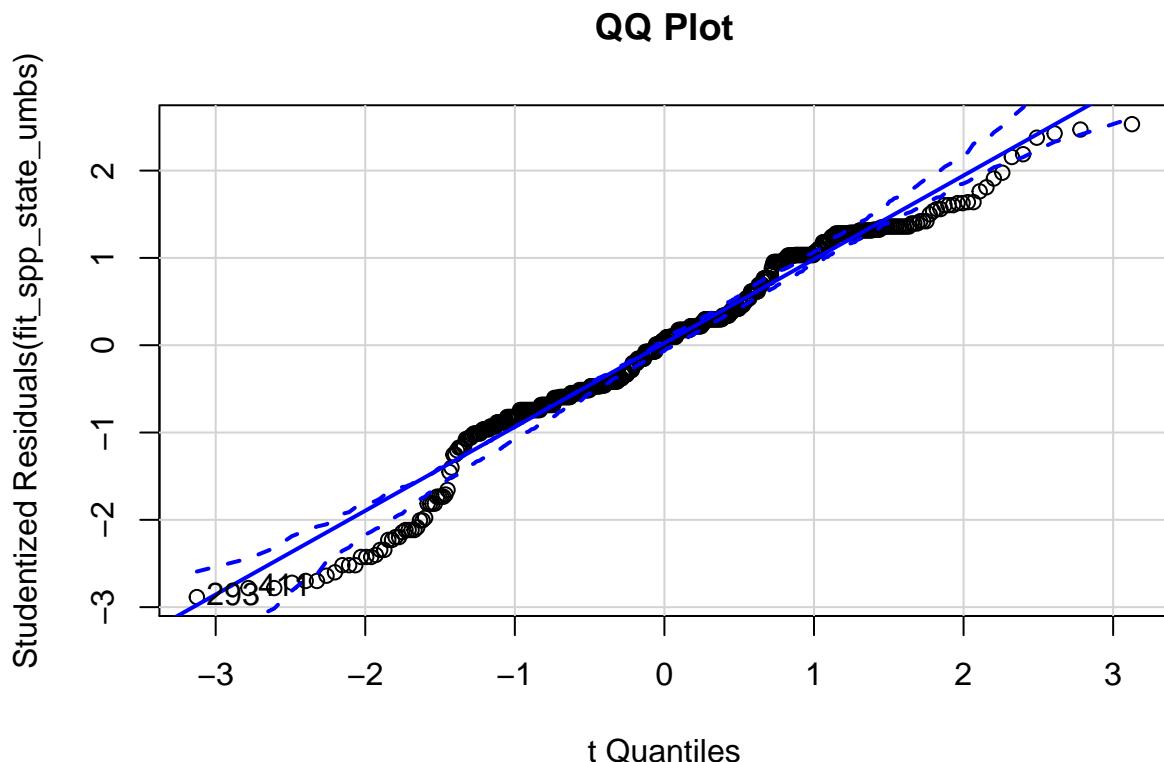
```
## Using data kbs_sd_plot from global environment. This could cause incorrect
## results if kbs_sd_plot has been altered since the model was fit. You can
## manually provide the data to the "data =" argument.
```

```
## 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
## 293 -2.8848          0.0040754        NA
```

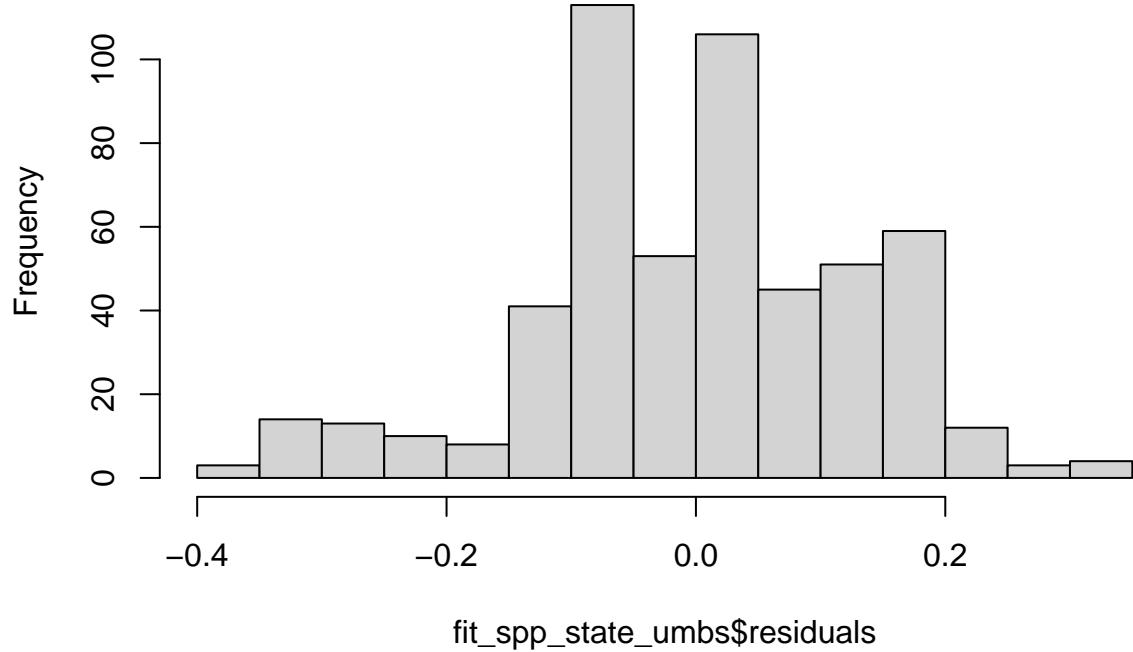
```
qqPlot(fit_spp_state_umbs, main = "QQ Plot")
```



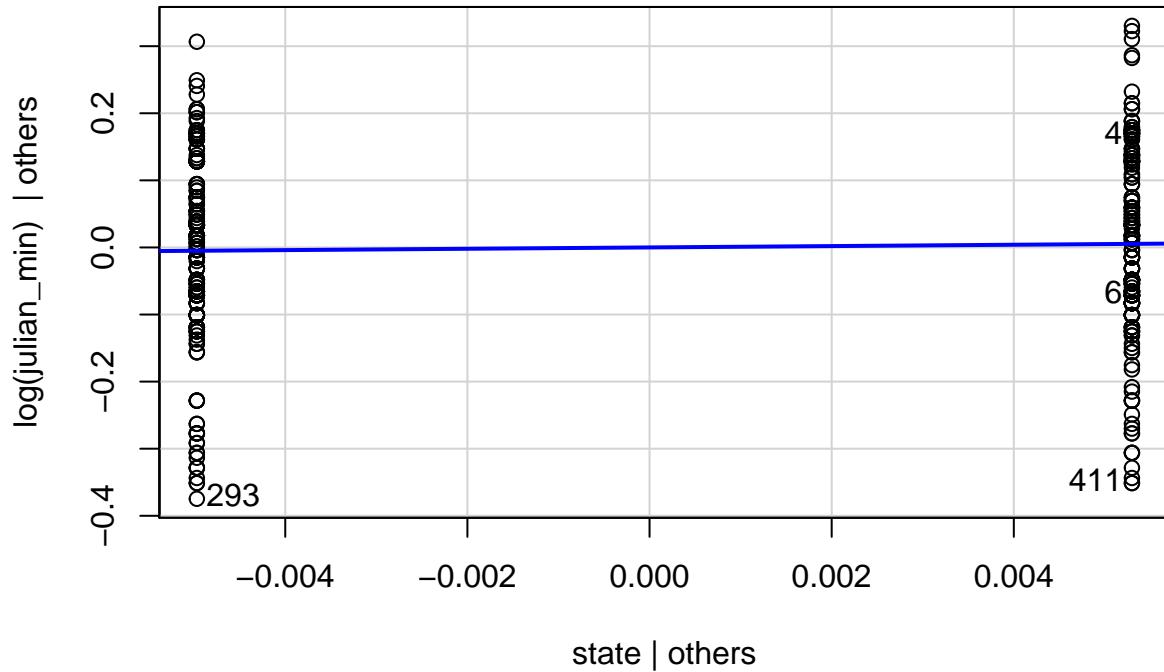
```
## 293 411
## 132 180
```

```
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      0.9754    0.0000
## Kolmogorov-Smirnov 0.0632    0.0278
## Cramer-von Mises   136.2309   0.0000
## Anderson-Darling    3.4524    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

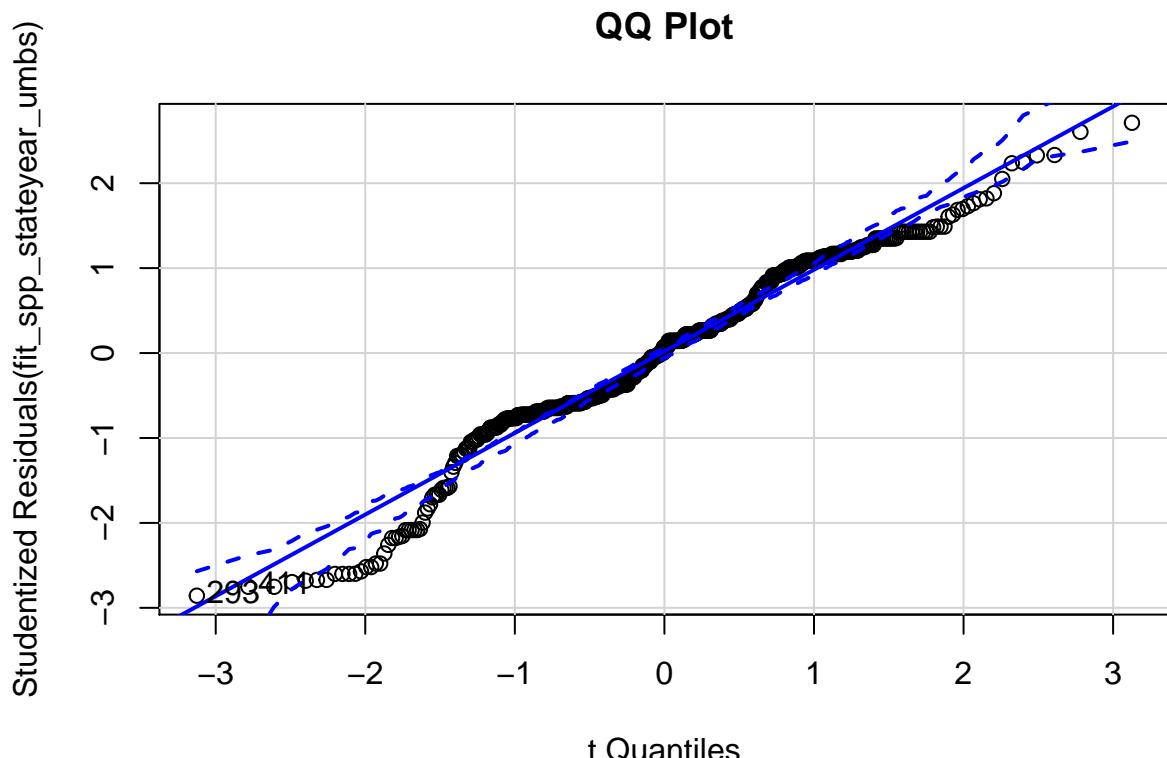
```

```

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##           rstudent unadjusted p-value Bonferroni p
## 293 -2.856926          0.0044463          NA

```

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



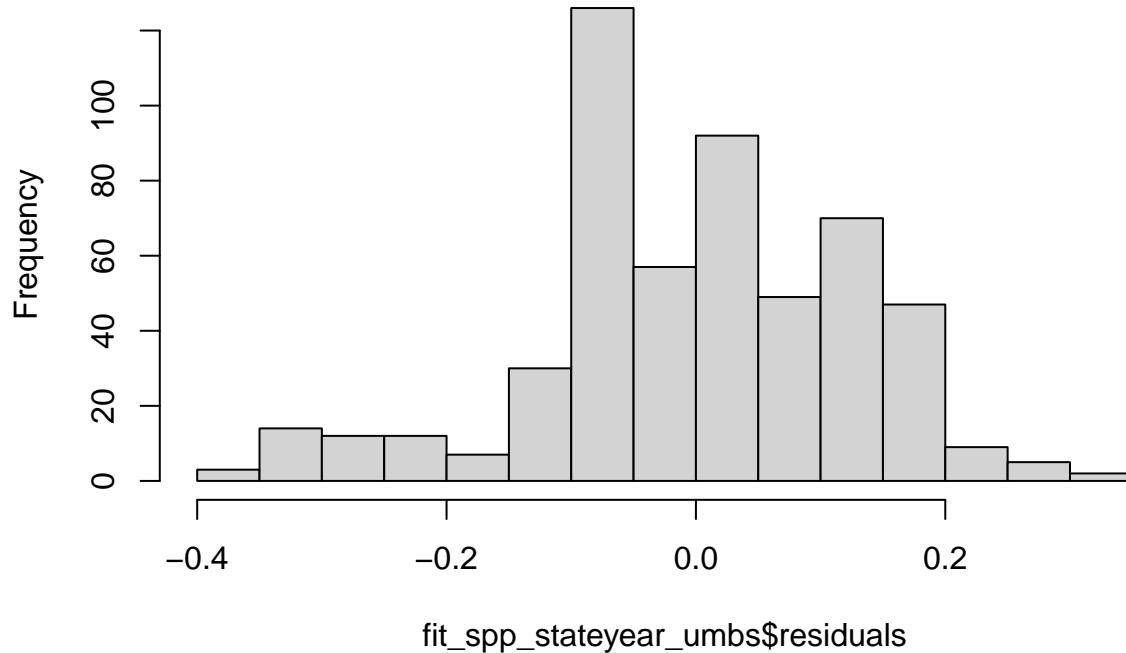
```

## 293 411
## 132 180

```

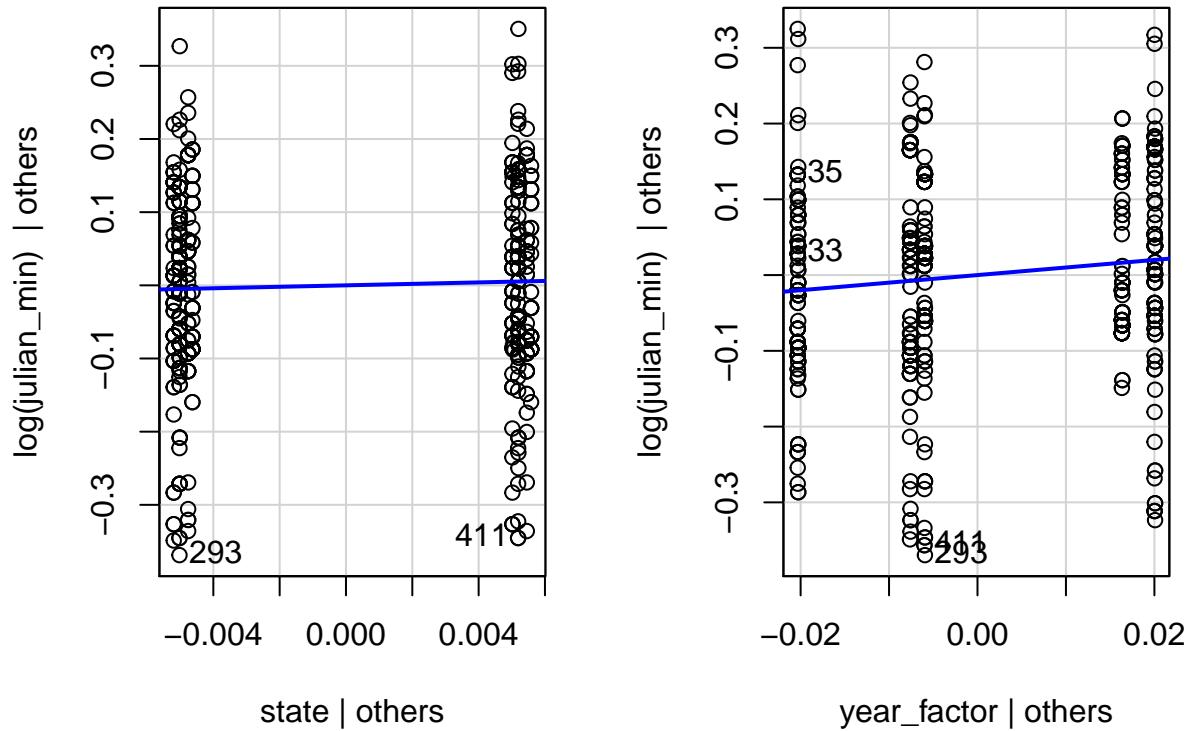
```
hist(fit_spp_stateyear_umbs$residuals)
```

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      0.9747    0.0000
## Kolmogorov-Smirnov   0.0754    0.0046
## Cramer-von Mises   136.3144   0.0000
## Anderson-Darling    3.7057    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.

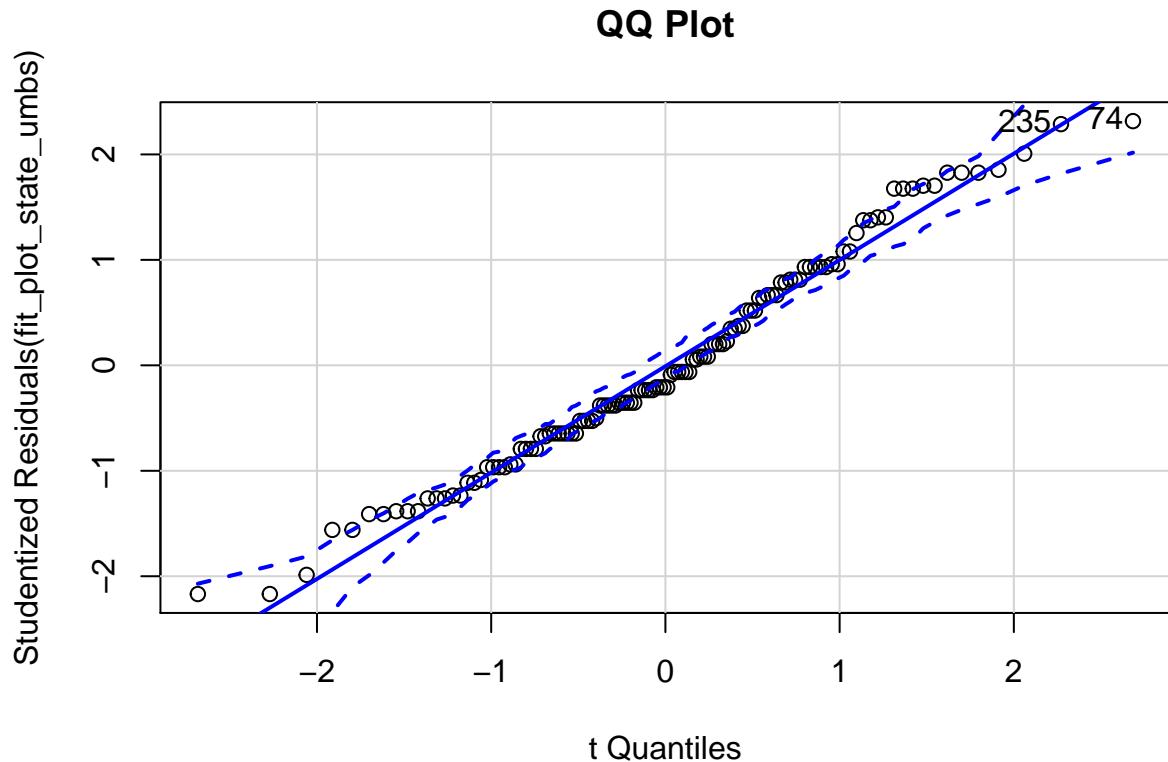
## Error in interact_plot(fit4, pred = year_factor, modx = state, mod2 = species): Focal predictor ("pr

# Plot level data UMBS State-only model
fit_plot_state_umbs <- lm(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
## 74  2.315735        0.022315       NA

```

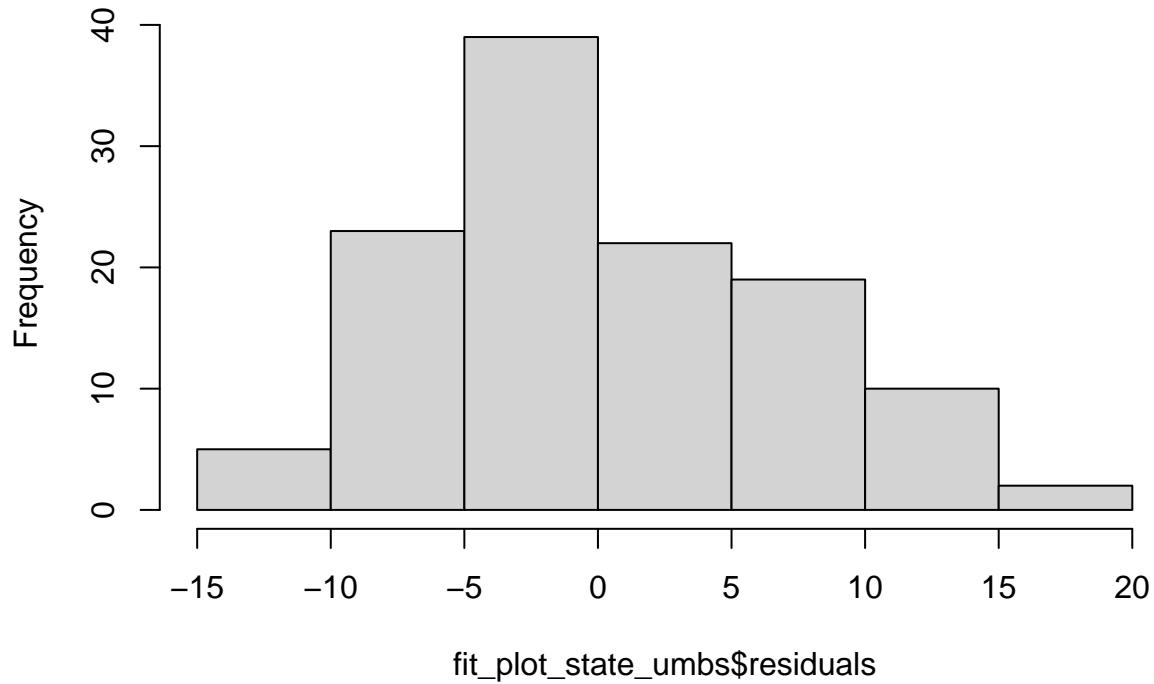
```
qqPlot(fit_plot_state_umbs, main = "QQ Plot")
```



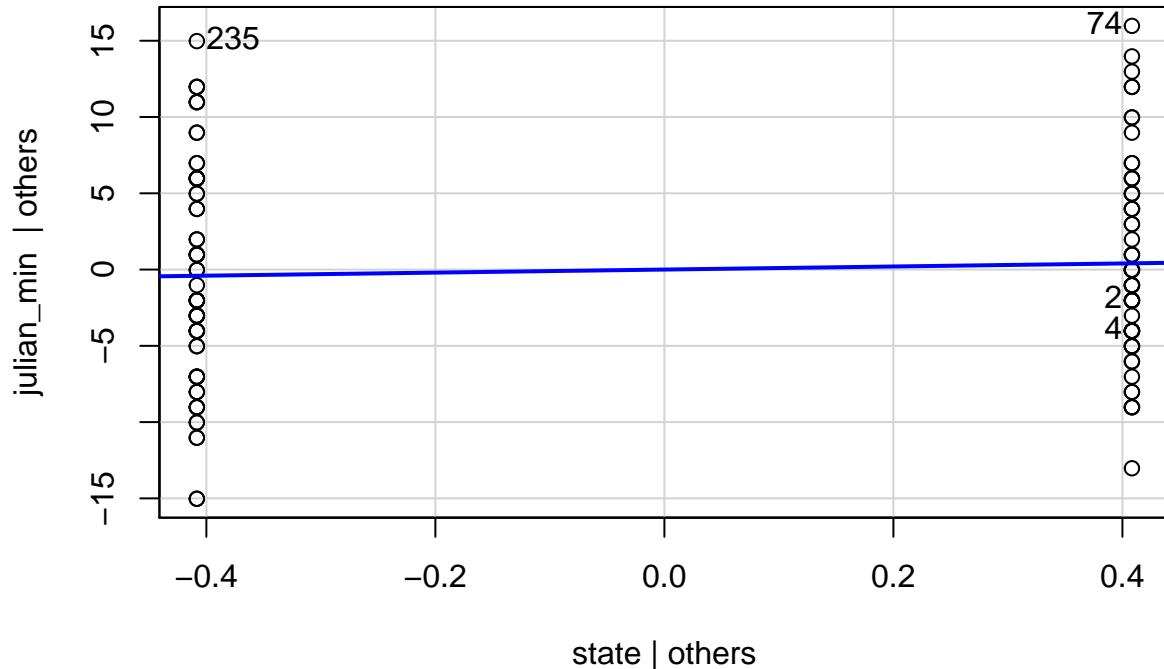
```
## 74 235  
## 31 99
```

```
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      0.9794    0.0623
## Kolmogorov-Smirnov 0.0909    0.2744
## Cramer-von Mises   9.5908    0.0000
## Anderson-Darling    0.7846    0.0406
## -----
# UMBS State and year model
fit_plot_stateyear_umbs <- lm(julian_min ~ state + year_factor, data = umbs_sd_plot)
outlierTest(fit_plot_stateyear_umbs)

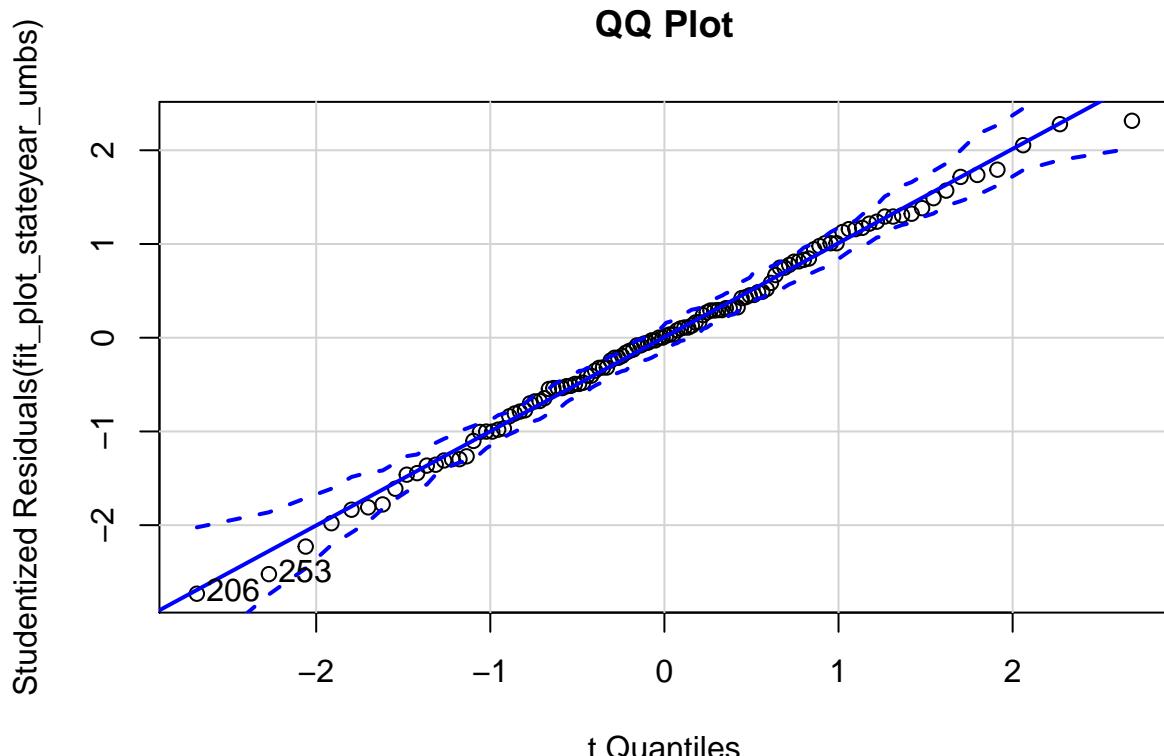
```

```

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##           rstudent unadjusted p-value Bonferroni p
## 206 -2.731017        0.0073274       0.87928

```

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



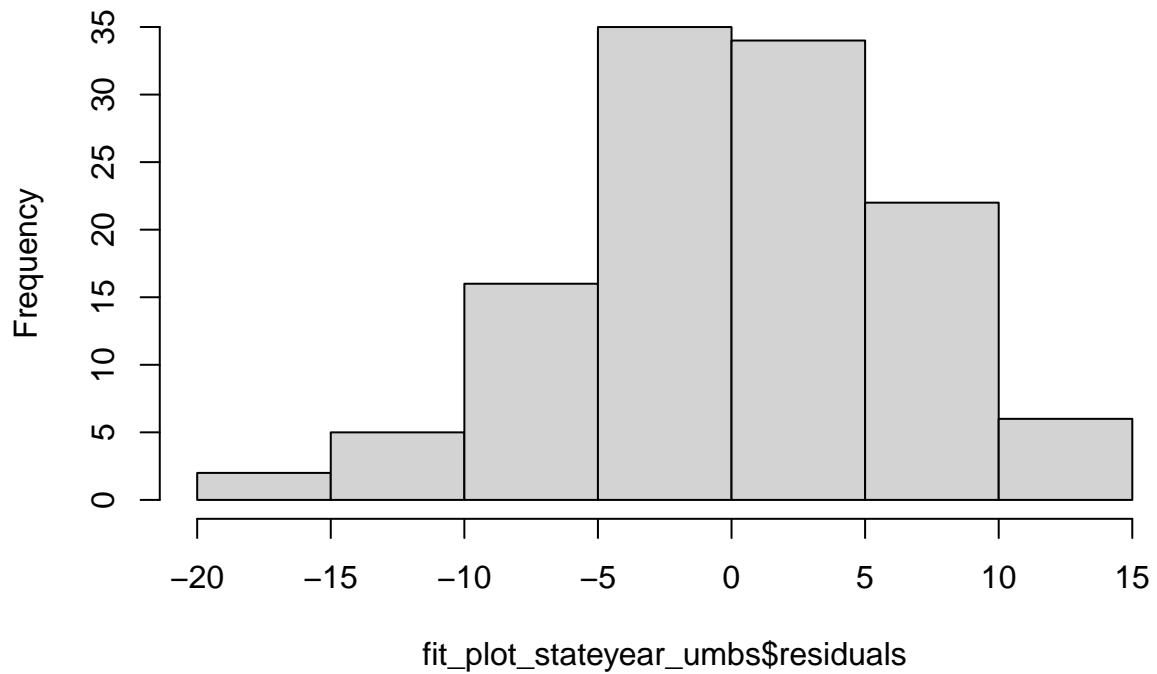
```

## 206 253
## 86 106

```

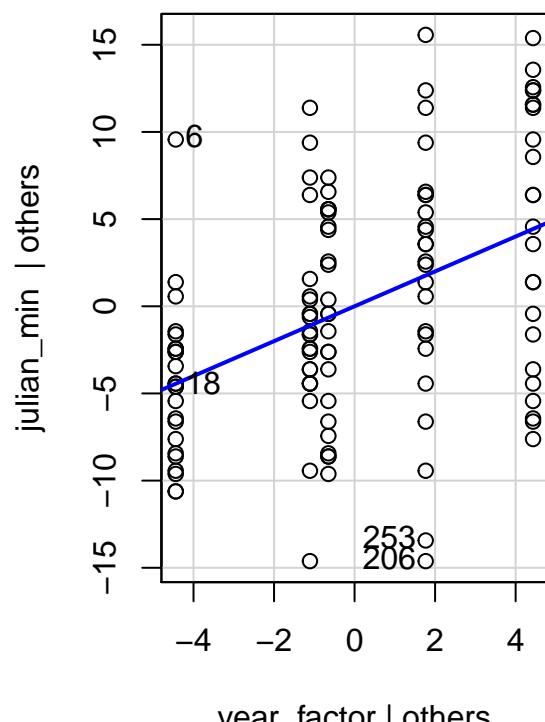
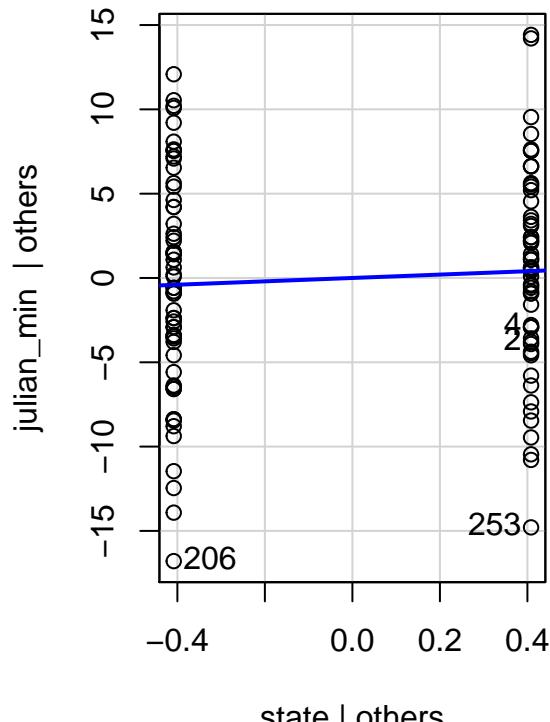
```
hist(fit_plot_stateyear_umbs$residuals)
```

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      0.9939    0.8870
## Kolmogorov-Smirnov   0.0424    0.9822
## Cramer-von Mises     8.6515    0.0000
## Anderson-Darling     0.1808    0.9128
## -----
```

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

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 KBS 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        0.00687 0.006873     1 794.06  2.9468 0.08644 .
## year_factor  0.72580 0.120966     6 794.41 51.8647 < 2e-16 ***
## insecticide  0.00461 0.004613     1 794.09  1.9779 0.16000
```

```

## state:year_factor      0.00936  0.001560      6 794.03  0.6690  0.67480
## year_factor:insecticide 0.01230  0.002050      6 794.06  0.8791  0.50962
## ---
## 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                      0.00687 0.006873      1 794.06  2.9468 0.08644 .
## year_factor                 0.72580 0.120966      6 794.41 51.8647 < 2e-16 ***
## insecticide                  0.00461 0.004613      1 794.09  1.9779 0.16000
## state:year_factor           0.00936 0.001560      6 794.03  0.6690  0.67480
## year_factor:insecticide    0.01230 0.002050      6 794.06  0.8791  0.50962
## ---
## 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/YaRrr/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 +
##       (1 | species)
## mod1: log(julian_min) ~ state * year_factor + insecticide * year_factor +
##       (1 | species) + (1 | plot)
##       npar   AIC     BIC logLik deviance Chisq Df Pr(>Chisq)
## mod2   23 -2457 -2348.8 1251.5    -2503
## mod1   24 -2455 -2342.1 1251.5    -2503      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
##   -2455.0 -2342.1 1251.5 -2503.0      792
##
## Scaled residuals:
##       Min     1Q Median     3Q    Max
## -4.5724 -0.5024 -0.0163  0.4570  7.6152
##
## Random effects:

```

```

## Groups      Name          Variance Std.Dev.
## plot        (Intercept) 0.0000000 0.00000
## species     (Intercept) 0.040349 0.20087
## Residual    0.002332 0.04829
## Number of obs: 816, groups: plot, 24; species, 22
##
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                5.36222   0.04483 25.74721 119.602
## stateambient               0.02064   0.01434 793.79795  1.439
## year_factor2              -0.09614   0.01422 793.86846 -6.763
## year_factor3              -0.10097   0.01467 794.21208 -6.881
## year_factor4              -0.10740   0.01516 793.88729 -7.083
## year_factor5              -0.07736   0.01544 793.82647 -5.011
## year_factor6              -0.06531   0.01462 793.84631 -4.467
## year_factor7              -0.07803   0.01662 793.88812 -4.696
## insecticideno_insects     0.03018   0.01428 793.79075  2.113
## stateambient:year_factor2 -0.02184   0.01595 793.83421 -1.369
## stateambient:year_factor3 -0.01226   0.01639 793.87732 -0.748
## stateambient:year_factor4 -0.02686   0.01701 793.95448 -1.579
## stateambient:year_factor5 -0.01148   0.01752 793.81377 -0.655
## stateambient:year_factor6 -0.01405   0.01667 793.96220 -0.843
## stateambient:year_factor7 -0.01099   0.01873 793.82692 -0.587
## year_factor2:insecticideno_insects -0.03335 0.01585 793.79845 -2.103
## year_factor3:insecticideno_insects -0.03444 0.01635 793.99174 -2.106
## year_factor4:insecticideno_insects -0.02605 0.01687 794.01584 -1.544
## year_factor5:insecticideno_insects -0.02813 0.01735 793.80126 -1.621
## year_factor6:insecticideno_insects -0.02783 0.01657 793.81802 -1.679
## year_factor7:insecticideno_insects -0.02363 0.01888 793.87290 -1.252
##                               Pr(>|t|)
## (Intercept)                < 2e-16 ***
## stateambient               0.1506
## year_factor2              2.62e-11 ***
## year_factor3              1.20e-11 ***
## year_factor4              3.11e-12 ***
## year_factor5              6.68e-07 ***
## year_factor6              9.10e-06 ***
## year_factor7              3.13e-06 ***
## insecticideno_insects     0.0349 *
## stateambient:year_factor2 0.1713
## stateambient:year_factor3 0.4549
## stateambient:year_factor4 0.1147
## stateambient:year_factor5 0.5124
## stateambient:year_factor6 0.3998
## stateambient:year_factor7 0.5574
## year_factor2:insecticideno_insects 0.0357 *
## year_factor3:insecticideno_insects 0.0355 *
## year_factor4:insecticideno_insects 0.1229
## year_factor5:insecticideno_insects 0.1053
## year_factor6:insecticideno_insects 0.0936 .
## year_factor7:insecticideno_insects 0.2111
## ---
## 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
## -2457.0 -2348.8  1251.5 -2503.0     793
## 
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.5724 -0.5024 -0.0163  0.4570  7.6152
## 
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   species (Intercept) 0.040349 0.20087
##   Residual            0.002332 0.04829
## Number of obs: 816, groups: species, 22
## 
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                5.36222  0.04483 25.74721 119.602
## stateambient               0.02064  0.01434 793.79795  1.439
## year_factor2              -0.09614  0.01422 793.86846 -6.763
## year_factor3              -0.10097  0.01467 794.21208 -6.881
## year_factor4              -0.10740  0.01516 793.88728 -7.083
## year_factor5              -0.07736  0.01544 793.82647 -5.011
## year_factor6              -0.06531  0.01462 793.84631 -4.467
## year_factor7              -0.07803  0.01662 793.88812 -4.696
## insecticideno_insects     0.03018  0.01428 793.79075  2.113
## stateambient:year_factor2 -0.02184  0.01595 793.83421 -1.369
## stateambient:year_factor3 -0.01226  0.01639 793.87732 -0.748
## stateambient:year_factor4 -0.02686  0.01701 793.95447 -1.579
## stateambient:year_factor5 -0.01148  0.01752 793.81377 -0.655
## stateambient:year_factor6 -0.01405  0.01667 793.96220 -0.843
## stateambient:year_factor7 -0.01099  0.01873 793.82692 -0.587
## year_factor2:insecticideno_insects -0.03335  0.01585 793.79844 -2.103
## year_factor3:insecticideno_insects -0.03444  0.01635 793.99174 -2.106
## year_factor4:insecticideno_insects -0.02605  0.01687 794.01584 -1.544
## year_factor5:insecticideno_insects -0.02813  0.01735 793.80126 -1.621
## year_factor6:insecticideno_insects -0.02783  0.01657 793.81802 -1.679
## year_factor7:insecticideno_insects -0.02363  0.01888 793.87290 -1.252
## Pr(>|t|)
```

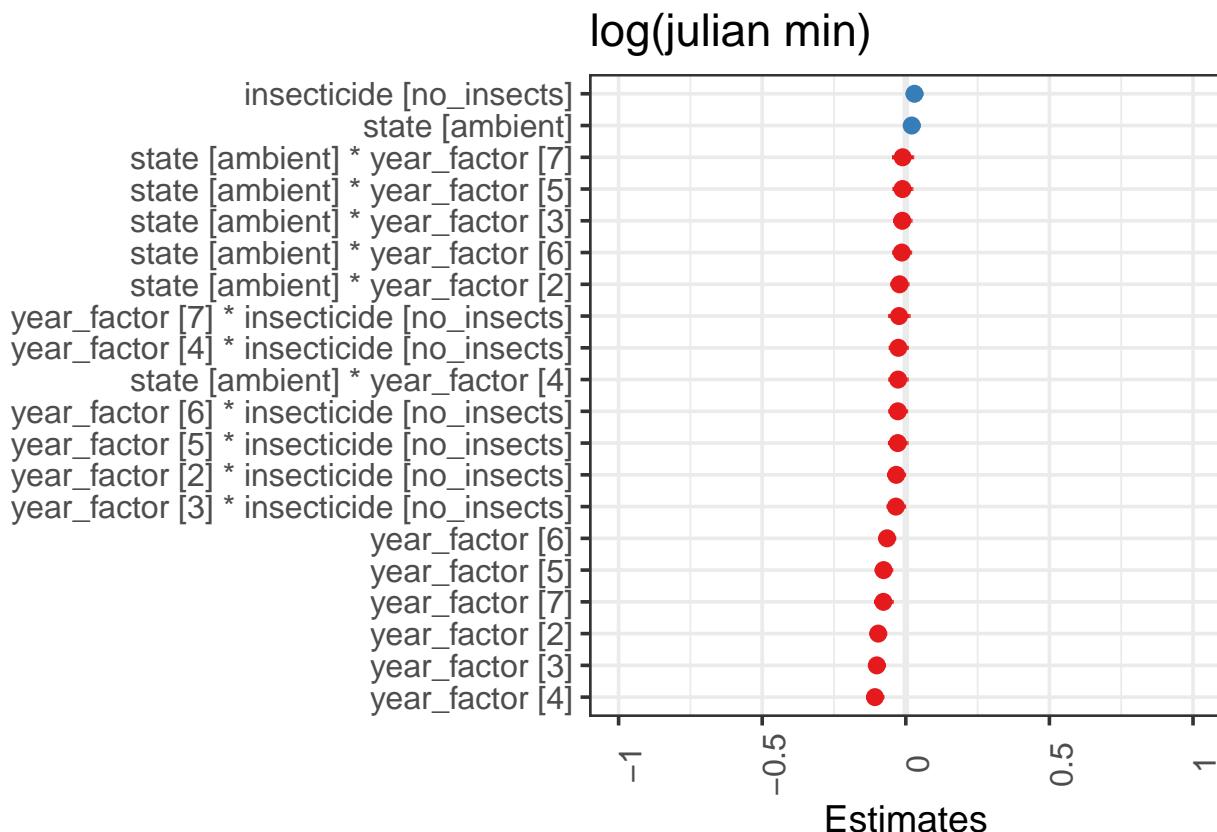
```

## (Intercept) < 2e-16 ***
## stateambient 0.1506
## year_factor2 2.62e-11 ***
## year_factor3 1.20e-11 ***
## year_factor4 3.11e-12 ***
## year_factor5 6.68e-07 ***
## year_factor6 9.10e-06 ***
## year_factor7 3.13e-06 ***
## insecticideno_insects 0.0349 *
## stateambient:year_factor2 0.1713
## stateambient:year_factor3 0.4549
## stateambient:year_factor4 0.1147
## stateambient:year_factor5 0.5124
## stateambient:year_factor6 0.3998
## stateambient:year_factor7 0.5574
## year_factor2:insecticideno_insects 0.0357 *
## year_factor3:insecticideno_insects 0.0355 *
## year_factor4:insecticideno_insects 0.1229
## year_factor5:insecticideno_insects 0.1053
## year_factor6:insecticideno_insects 0.0936 .
## year_factor7:insecticideno_insects 0.2111
## ---
## 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)

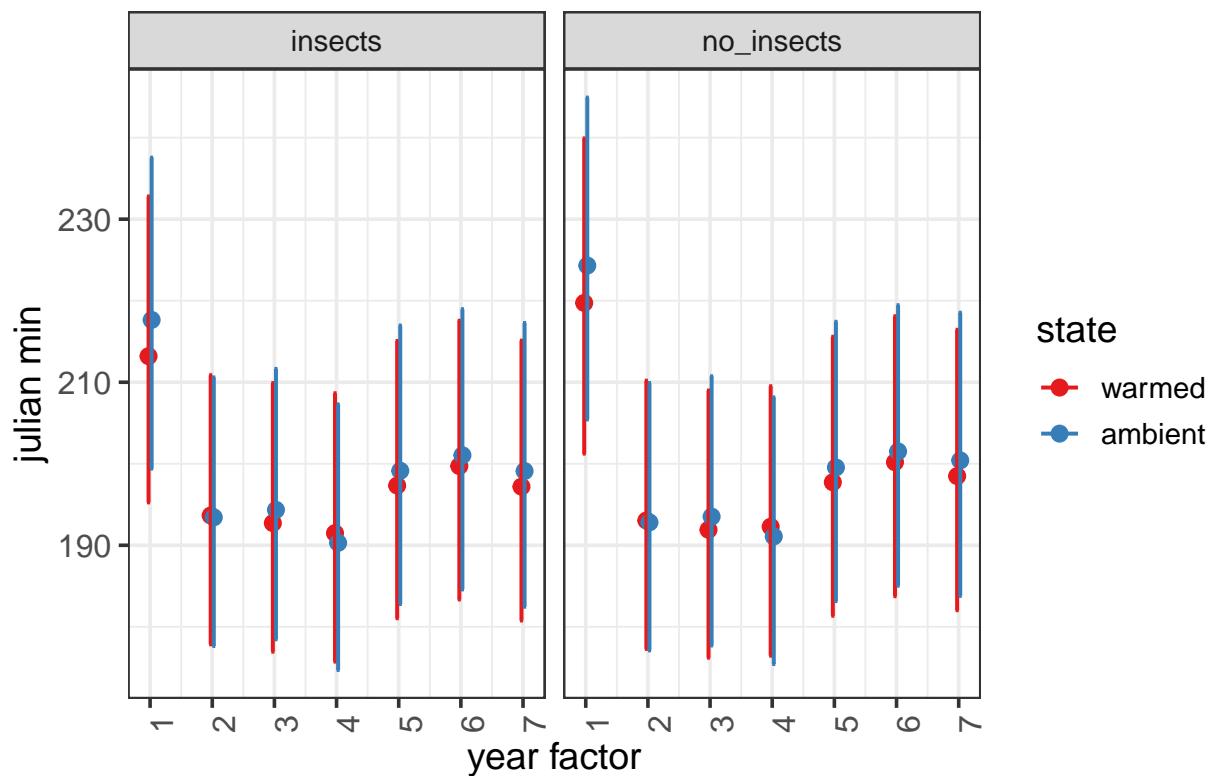
```



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

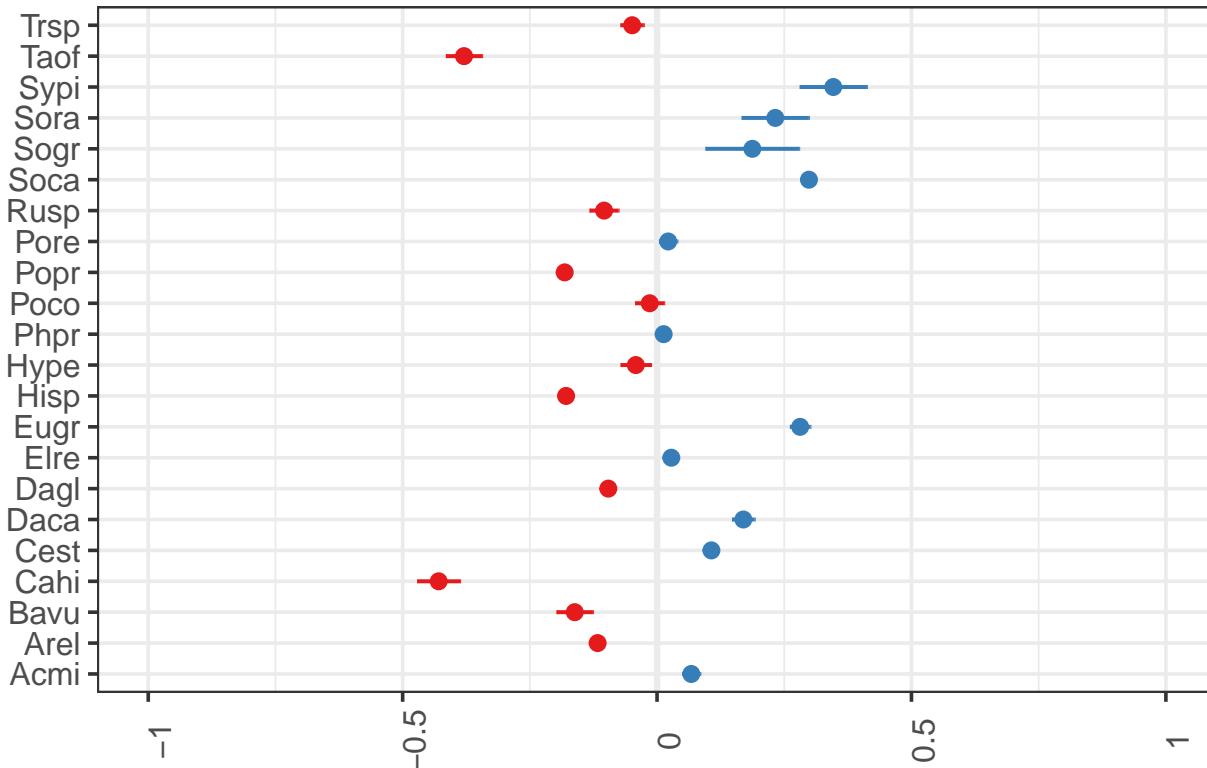
```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standardized
```

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(mod2, mod3) # Don't need insecticide, continue with mod3
```

```
## Data: kbs_sd_spp
## Models:
## mod3: log(julian_min) ~ state * year_factor + (1 | species)
## mod2: log(julian_min) ~ state * year_factor + insecticide * year_factor +
## mod2: (1 | species)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod3   16 -2465.4 -2390.2 1248.7  -2497.4
## mod2   23 -2457.0 -2348.8 1251.5  -2503.0 5.5715 7     0.5906
```

```
# 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 -2471.5 -2419.8 1246.8   -2493.5
## mod3   16 -2465.4 -2390.2 1248.7   -2497.4 3.8809  5      0.5667

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

```

```

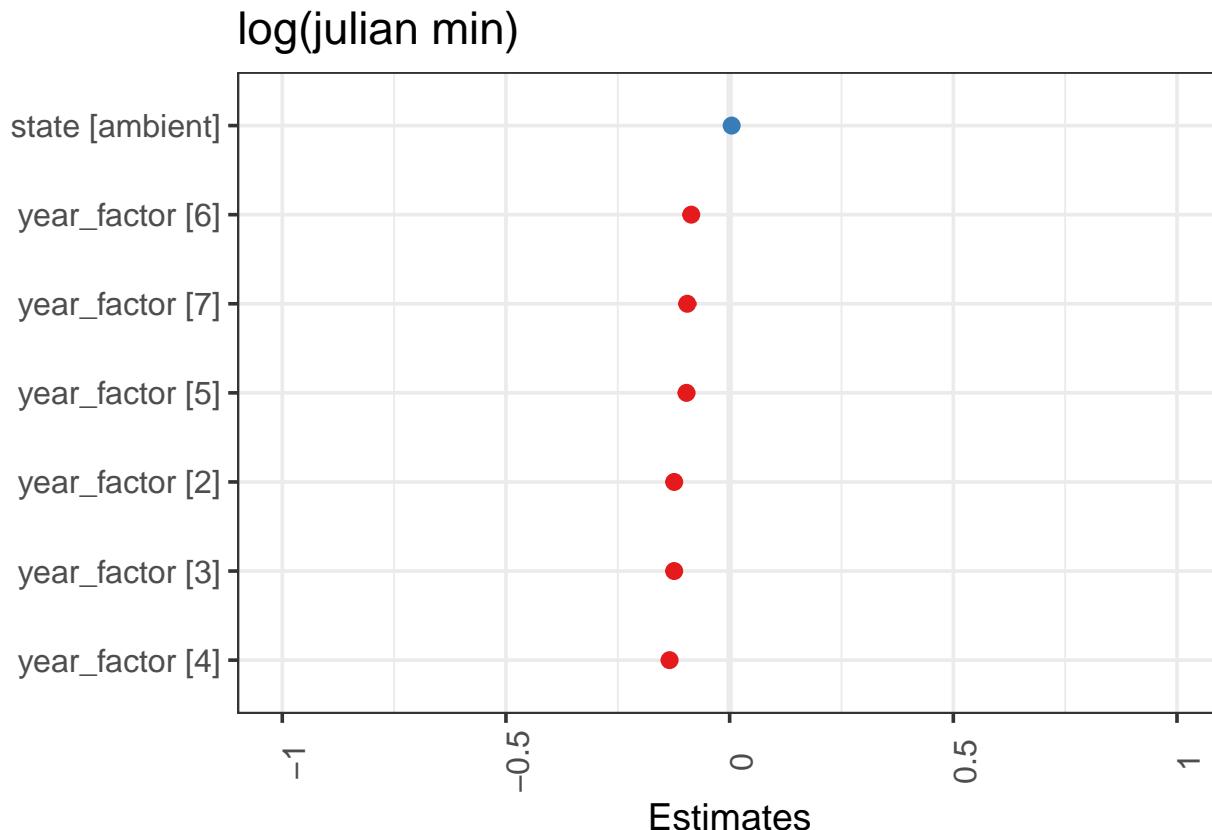
##      dAICc df weight
## mod5  0.0  11  0.962
## mod3  6.5  16  0.038

```

```

# 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"))

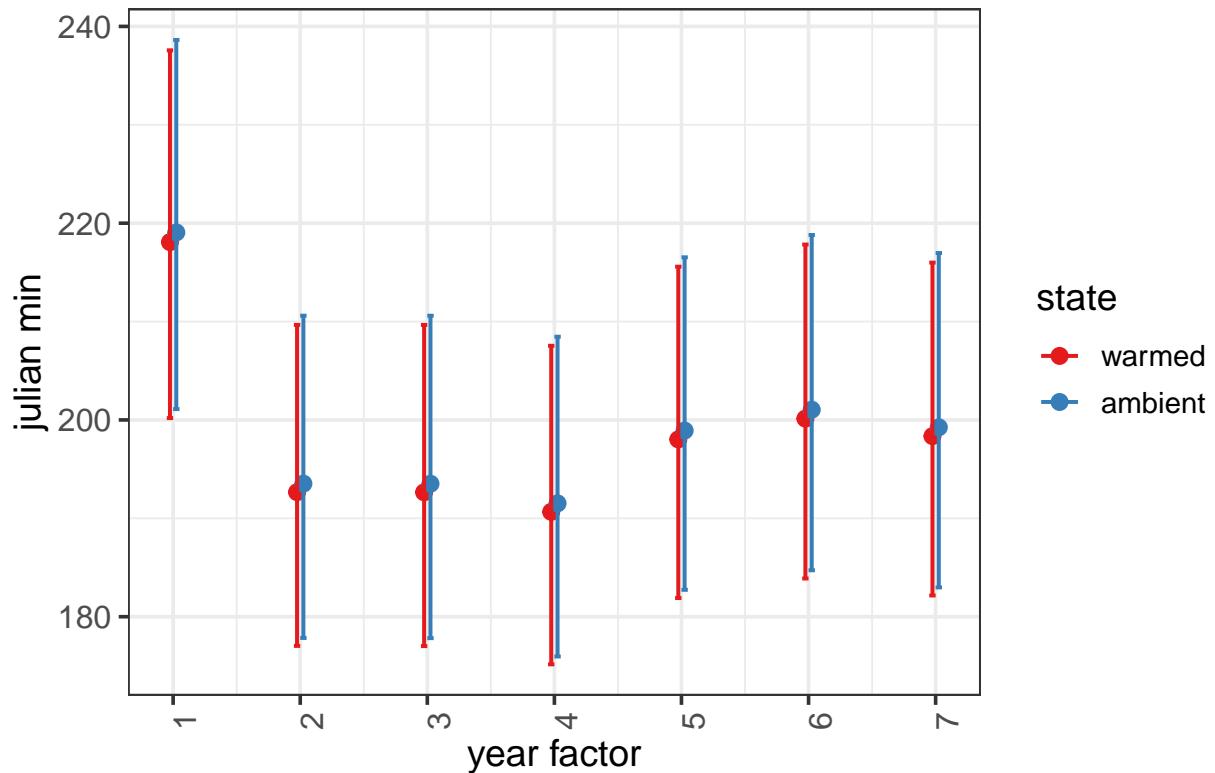
```

```

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

```

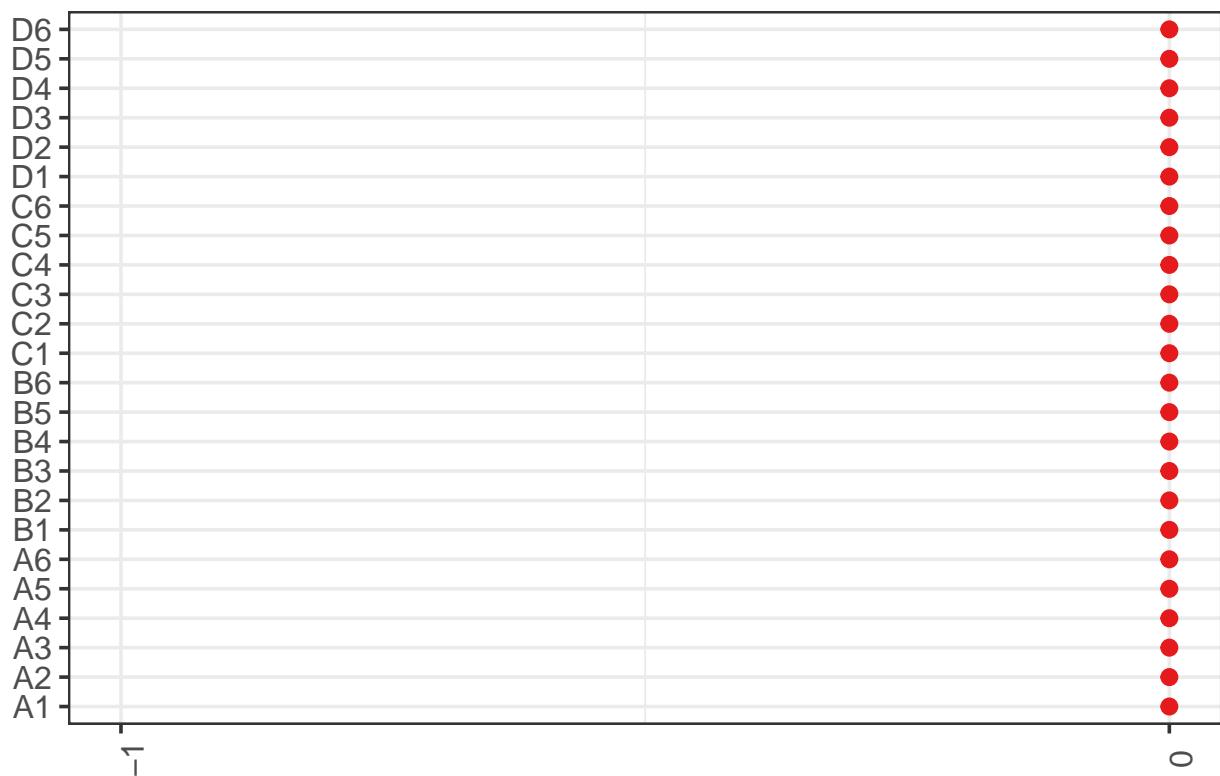
Predicted values of julian min



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

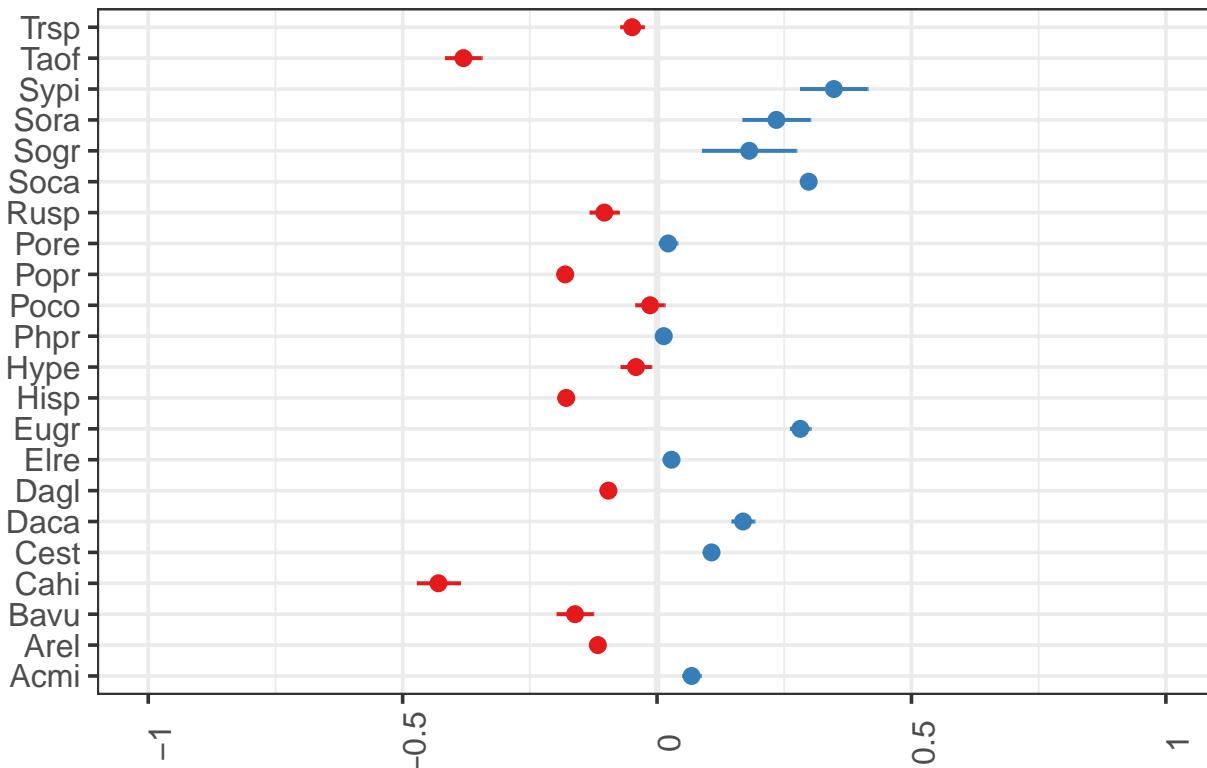
```
## [[1]]
```

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
```

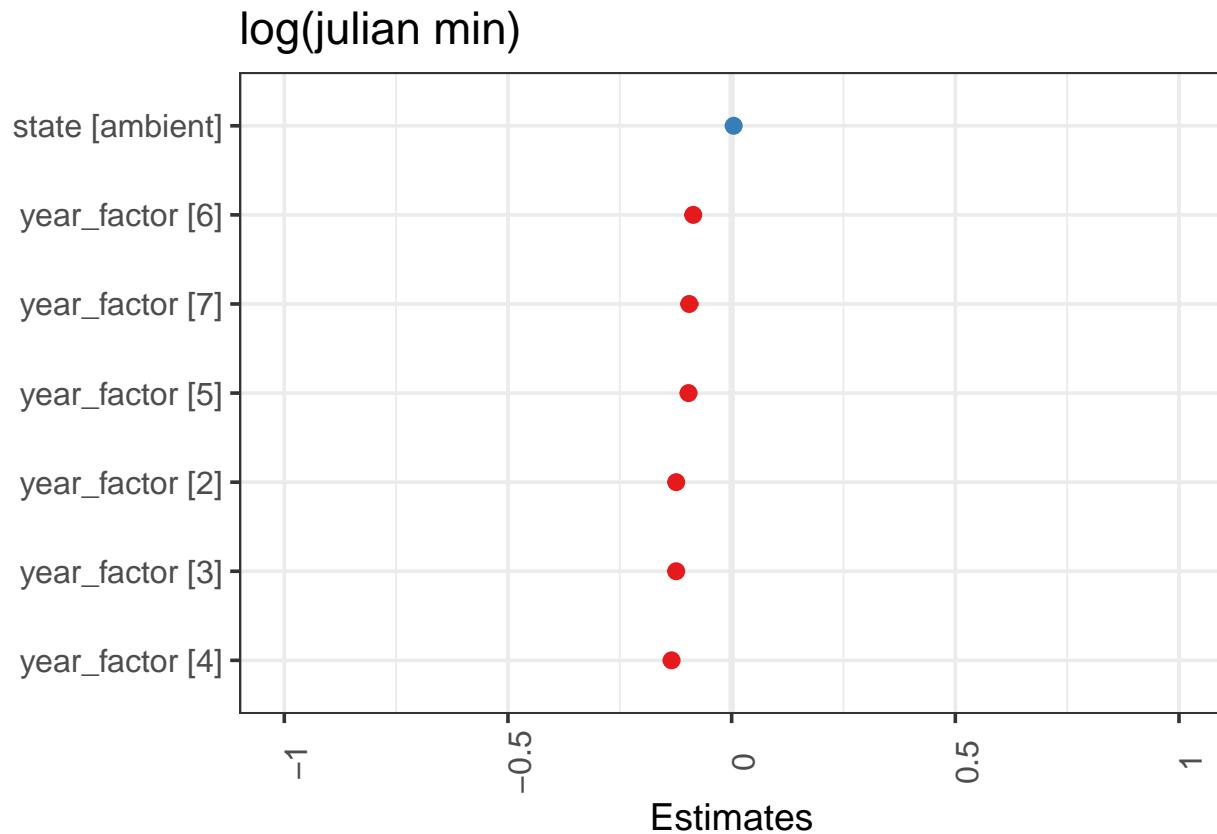
```
anova(mod5, mod6) #stick with mod5
```

```
## 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 -2471.5 -2419.8 1246.8   -2493.5
## mod6     13 -2378.3 -2317.1 1202.1   -2404.3      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       0.00377 0.003773     1 794.25 1.5984 0.2065
## year_factor 0.73837 0.123062     6 794.44 52.1384 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

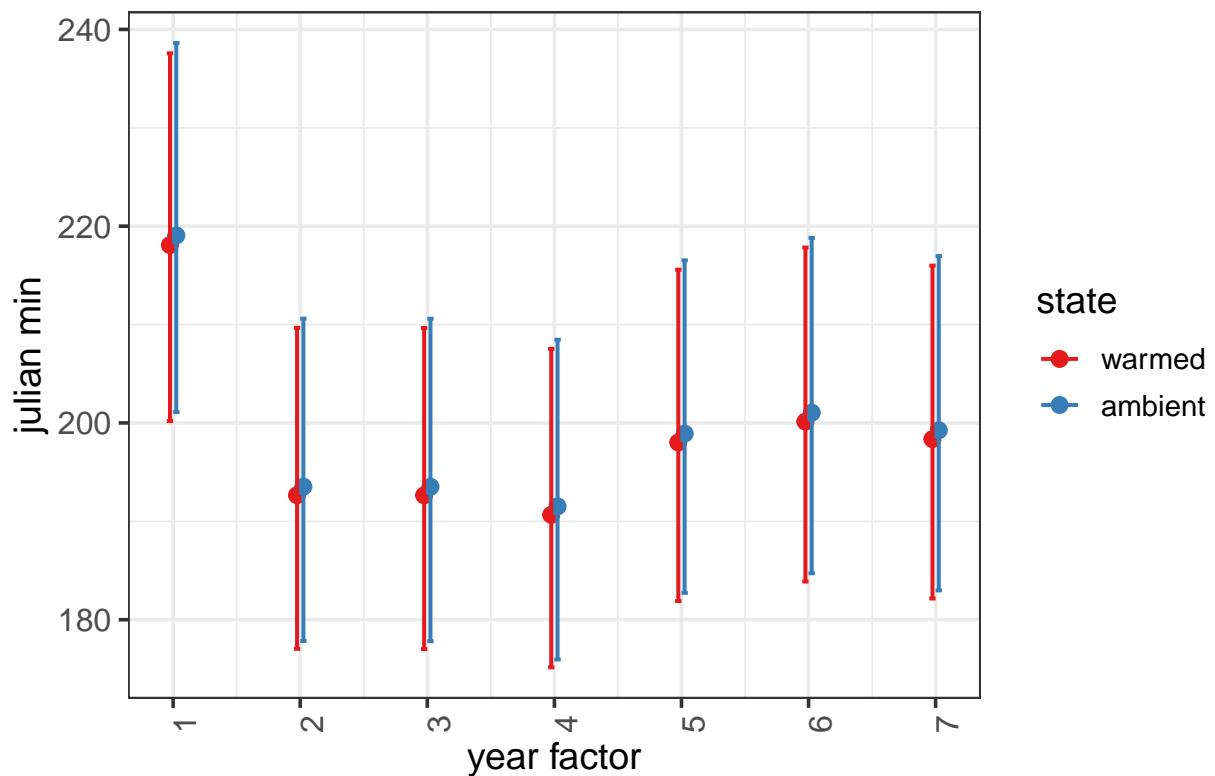
```
plot_model(mod5, sort.est = TRUE)
```



```
# 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. Standard
```

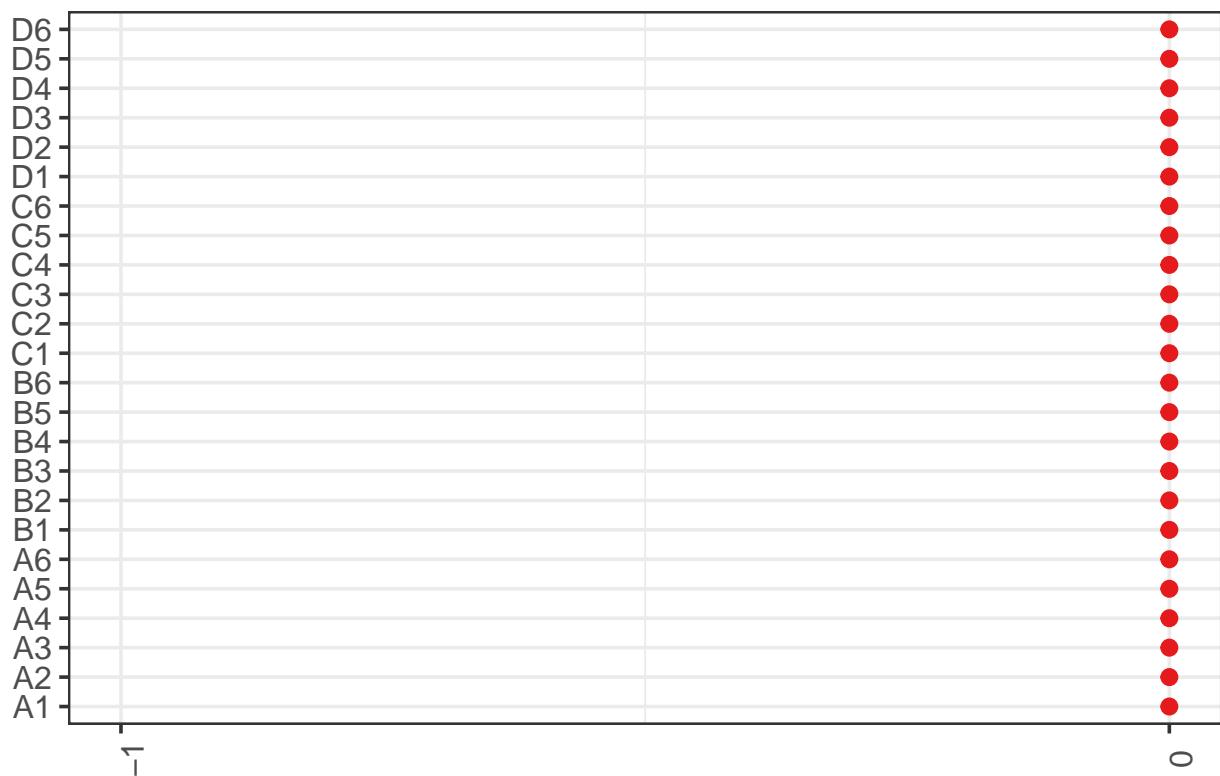
Predicted values of julian min



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

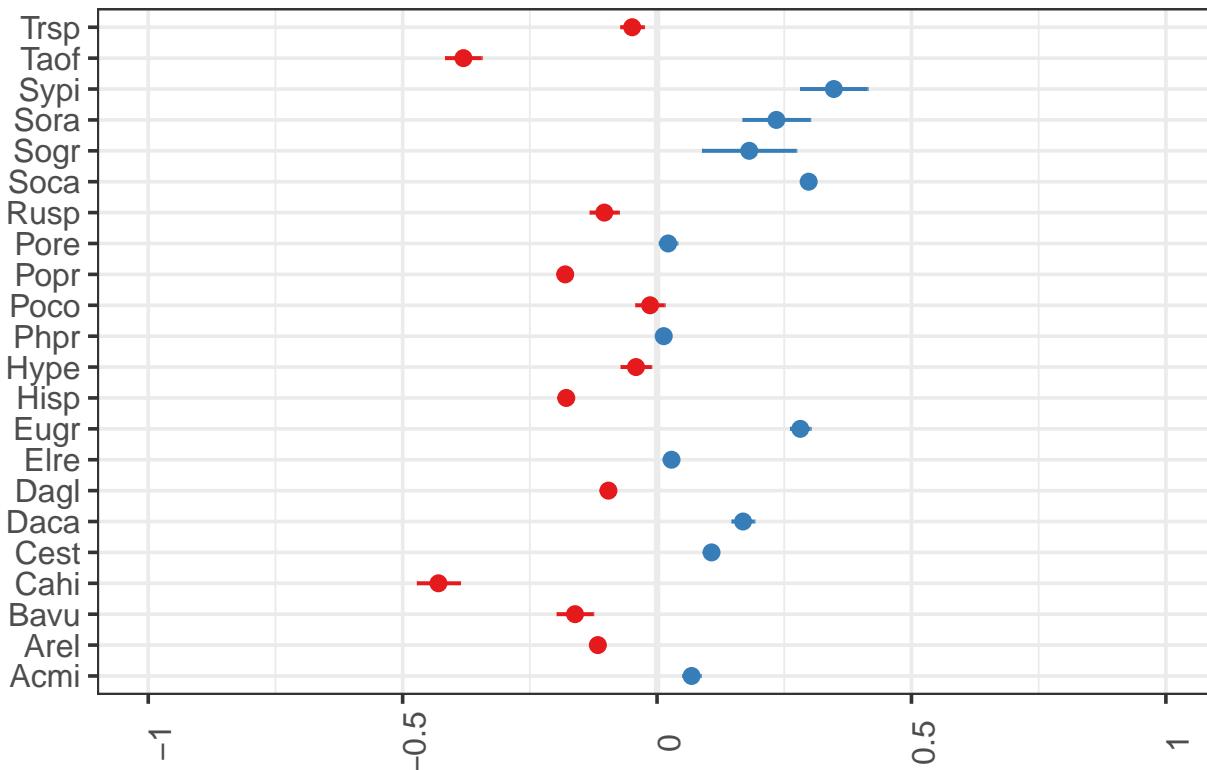
```
## [[1]]
```

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 1 negative eigenvalue: -8.6e-02

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 -2471.5 -2419.8 1246.8   -2493.5
## mod7    52 -2484.1 -2239.4 1294.0   -2588.1 94.521 41  4.132e-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 -2580.7 -2434.8 1321.3   -2642.7
## mod7    52 -2484.1 -2239.4 1294.0   -2588.1      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 -2580.7 -2434.8 1321.3   -2642.7
## mod7b   37 -2572.7 -2398.6 1323.3   -2646.7 3.9907  6     0.6779

```

```
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 -2580.7 -2434.8 1321.3   -2642.7
## mod7c   32 -2579.0 -2428.4 1321.5   -2643.0 0.3283  1     0.5667

```

```
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
## -2580.7 -2434.8  1321.3 -2642.7      785
##
## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -4.5905 -0.5311 -0.0057  0.4910  7.6698
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   plot     (Intercept) 0.0000000 0.00000
##   Residual           0.002296  0.04792
## Number of obs: 816, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)  5.452873  0.011998 816.00000 454.472 < 2e-16 ***
## stateambient 0.004415  0.003510 816.00000  1.258 0.208799  
## speciesArel -0.184402  0.010473 816.00000 -17.608 < 2e-16 ***
## speciesBavu -0.230624  0.020319 816.00000 -11.350 < 2e-16 ***
## speciesCahi -0.502768  0.023483 816.00000 -21.409 < 2e-16 ***
## speciesCest  0.039203  0.011627 816.00000  3.372 0.000782 ***
## speciesDaca  0.101529  0.014505 816.00000  6.999 5.37e-12 ***
## speciesDagl -0.163609  0.010838 816.00000 -15.095 < 2e-16 ***
## speciesElre -0.039505  0.011380 816.00000 -3.471 0.000545 ***
## speciesEugr  0.214376  0.013910 816.00000 15.411 < 2e-16 ***
## speciesHisp -0.246595  0.011135 816.00000 -22.146 < 2e-16 ***
## speciesHype -0.109631  0.017735 816.00000 -6.182 1.00e-09 ***
## speciesPhpr -0.054816  0.010555 816.00000 -5.193 2.61e-07 ***
## speciesPoco -0.081816  0.017619 816.00000 -4.644 3.99e-06 ***
## speciesPopr -0.248585  0.010210 816.00000 -24.348 < 2e-16 ***
## speciesPore -0.046129  0.013021 816.00000 -3.543 0.000419 ***
## speciesRusp -0.172149  0.017193 816.00000 -10.013 < 2e-16 ***
## speciesSoca  0.230145  0.010176 816.00000 22.616 < 2e-16 ***
## speciesSogr  0.123898  0.048999 816.00000  2.529 0.011640 *  
## speciesSora  0.173310  0.035193 816.00000  4.925 1.02e-06 ***
## speciesSypi  0.289844  0.035408 816.00000  8.186 1.04e-15 ***
## speciesTaof -0.451720  0.020330 816.00000 -22.220 < 2e-16 ***
## speciesTrsp -0.117093  0.014959 816.00000 -7.828 1.54e-14 ***
## year_factor2 -0.123863  0.008148 816.00000 -15.201 < 2e-16 ***
## year_factor3 -0.123886  0.008316 816.00000 -14.897 < 2e-16 ***
## year_factor4 -0.134448  0.008456 816.00000 -15.900 < 2e-16 ***
## year_factor5 -0.096464  0.008657 816.00000 -11.143 < 2e-16 ***
## year_factor6 -0.086031  0.008335 816.00000 -10.321 < 2e-16 ***
## year_factor7 -0.094665  0.009456 816.00000 -10.012 < 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 0.00363     1    816  1.5822 0.2088
## species    23.6359 1.12552    21    816 490.1339 <2e-16 ***
## year_factor 0.7372 0.12286     6    816  53.5021 <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 min
# first seed 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 0.008644 731.1   5.369   5.403
##   ambient 1         5.390 0.008533 717.0   5.373   5.407
##   warmed 2          5.262 0.005438 323.6   5.251   5.273
##   ambient 2         5.266 0.005233 282.4   5.256   5.277
##   warmed 3          5.262 0.005634 346.9   5.251   5.273
##   ambient 3         5.266 0.005457 302.0   5.256   5.277
##   warmed 4          5.251 0.006299 469.0   5.239   5.264
##   ambient 4         5.256 0.006067 404.7   5.244   5.268
##   warmed 5          5.289 0.006757 535.0   5.276   5.303
##   ambient 5         5.294 0.006451 460.0   5.281   5.306
##   warmed 6          5.300 0.005921 451.5   5.288   5.311
##   ambient 6         5.304 0.005719 378.3   5.293   5.315
##   warmed 7          5.291 0.007532 606.6   5.276   5.306
##   ambient 7         5.295 0.007548 571.4   5.281   5.310
##
## 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 -4.41e-03 0.00359 25.9 -1.231 0.9907
##   warmed 1 - warmed 2  1.24e-01 0.00831 840.6 14.908 <.0001
##   warmed 1 - ambient 2  1.19e-01 0.00903 509.1 13.223 <.0001
##   warmed 1 - warmed 3  1.24e-01 0.00848 841.6 14.609 <.0001
##   warmed 1 - ambient 3  1.19e-01 0.00920 517.6 12.981 <.0001
##   warmed 1 - warmed 4  1.34e-01 0.00862 839.0 15.597 <.0001
##   warmed 1 - ambient 4  1.30e-01 0.00928 528.5 14.005 <.0001
##   warmed 1 - warmed 5  9.65e-02 0.00882 838.2 10.932 <.0001
##   warmed 1 - ambient 5  9.20e-02 0.00941 540.7  9.780 <.0001
##   warmed 1 - warmed 6  8.60e-02 0.00850 838.6 10.125 <.0001

```

```

## warmed 1 - ambient 6 8.16e-02 0.00920 516.5 8.873 <.0001
## warmed 1 - warmed 7 9.47e-02 0.00964 843.5 9.815 <.0001
## warmed 1 - ambient 7 9.03e-02 0.01039 597.9 8.683 <.0001
## ambient 1 - warmed 2 1.28e-01 0.00906 509.1 14.152 <.0001
## ambient 1 - ambient 2 1.24e-01 0.00831 840.6 14.908 <.0001
## ambient 1 - warmed 3 1.28e-01 0.00921 524.6 13.930 <.0001
## ambient 1 - ambient 3 1.24e-01 0.00848 841.6 14.609 <.0001
## ambient 1 - warmed 4 1.39e-01 0.00939 555.2 14.792 <.0001
## ambient 1 - ambient 4 1.34e-01 0.00862 839.0 15.597 <.0001
## ambient 1 - warmed 5 1.01e-01 0.00964 581.7 10.469 <.0001
## ambient 1 - ambient 5 9.65e-02 0.00882 838.2 10.932 <.0001
## ambient 1 - warmed 6 9.04e-02 0.00925 547.9 9.782 <.0001
## ambient 1 - ambient 6 8.60e-02 0.00850 838.6 10.125 <.0001
## ambient 1 - warmed 7 9.91e-02 0.01018 610.2 9.729 <.0001
## ambient 1 - ambient 7 9.47e-02 0.00964 843.5 9.815 <.0001
## warmed 2 - ambient 2 -4.41e-03 0.00359 25.9 -1.231 0.9907
## warmed 2 - warmed 3 2.34e-05 0.00547 841.2 0.004 1.0000
## warmed 2 - ambient 3 -4.39e-03 0.00656 228.8 -0.669 1.0000
## warmed 2 - warmed 4 1.06e-02 0.00591 839.9 1.791 0.8801
## warmed 2 - ambient 4 6.17e-03 0.00686 258.7 0.899 0.9998
## warmed 2 - warmed 5 -2.74e-02 0.00625 844.9 -4.386 0.0011
## warmed 2 - ambient 5 -3.18e-02 0.00707 269.7 -4.499 0.0008
## warmed 2 - warmed 6 -3.78e-02 0.00568 846.0 -6.659 <.0001
## warmed 2 - ambient 6 -4.22e-02 0.00671 225.3 -6.300 <.0001
## warmed 2 - warmed 7 -2.92e-02 0.00750 845.3 -3.893 0.0080
## warmed 2 - ambient 7 -3.36e-02 0.00846 392.3 -3.974 0.0064
## ambient 2 - warmed 3 4.44e-03 0.00653 232.8 0.680 1.0000
## ambient 2 - ambient 3 2.34e-05 0.00547 841.2 0.004 1.0000
## ambient 2 - warmed 4 1.50e-02 0.00696 288.4 2.154 0.6627
## ambient 2 - ambient 4 1.06e-02 0.00591 839.9 1.791 0.8801
## ambient 2 - warmed 5 -2.30e-02 0.00733 321.8 -3.135 0.1009
## ambient 2 - ambient 5 -2.74e-02 0.00625 844.9 -4.386 0.0011
## ambient 2 - warmed 6 -3.34e-02 0.00673 252.2 -4.965 0.0001
## ambient 2 - ambient 6 -3.78e-02 0.00568 846.0 -6.659 <.0001
## ambient 2 - warmed 7 -2.48e-02 0.00816 394.8 -3.035 0.1303
## ambient 2 - ambient 7 -2.92e-02 0.00750 845.3 -3.893 0.0080
## warmed 3 - ambient 3 -4.41e-03 0.00359 25.9 -1.231 0.9907
## warmed 3 - warmed 4 1.06e-02 0.00616 831.5 1.716 0.9102
## warmed 3 - ambient 4 6.15e-03 0.00706 295.2 0.871 0.9998
## warmed 3 - warmed 5 -2.74e-02 0.00649 843.2 -4.223 0.0022
## warmed 3 - ambient 5 -3.18e-02 0.00728 301.0 -4.376 0.0014
## warmed 3 - warmed 6 -3.79e-02 0.00600 845.5 -6.304 <.0001
## warmed 3 - ambient 6 -4.23e-02 0.00697 258.5 -6.068 <.0001
## warmed 3 - warmed 7 -2.92e-02 0.00767 846.0 -3.808 0.0110
## warmed 3 - ambient 7 -3.36e-02 0.00860 424.8 -3.911 0.0080
## ambient 3 - warmed 4 1.50e-02 0.00719 320.8 2.084 0.7119
## ambient 3 - ambient 4 1.06e-02 0.00616 831.5 1.716 0.9102
## ambient 3 - warmed 5 -2.30e-02 0.00756 348.2 -3.045 0.1278
## ambient 3 - ambient 5 -2.74e-02 0.00649 843.2 -4.223 0.0022
## ambient 3 - warmed 6 -3.34e-02 0.00702 282.6 -4.763 0.0003
## ambient 3 - ambient 6 -3.79e-02 0.00600 845.5 -6.304 <.0001
## ambient 3 - warmed 7 -2.48e-02 0.00834 423.0 -2.975 0.1515
## ambient 3 - ambient 7 -2.92e-02 0.00767 846.0 -3.808 0.0110
## warmed 4 - ambient 4 -4.41e-03 0.00359 25.9 -1.231 0.9907

```

```

##  warmed 4 - warmed 5 -3.80e-02 0.00673 836.3 -5.648 <.0001
##  warmed 4 - ambient 5 -4.24e-02 0.00754 350.2 -5.620 <.0001
##  warmed 4 - warmed 6 -4.84e-02 0.00635 837.9 -7.628 <.0001
##  warmed 4 - ambient 6 -5.28e-02 0.00733 319.3 -7.212 <.0001
##  warmed 4 - warmed 7 -3.98e-02 0.00790 841.4 -5.036 0.0001
##  warmed 4 - ambient 7 -4.42e-02 0.00885 479.0 -4.992 0.0001
##  ambient 4 - warmed 5 -3.36e-02 0.00770 372.8 -4.360 0.0014
##  ambient 4 - ambient 5 -3.80e-02 0.00673 836.3 -5.648 <.0001
##  ambient 4 - warmed 6 -4.40e-02 0.00725 320.3 -6.066 <.0001
##  ambient 4 - ambient 6 -4.84e-02 0.00635 837.9 -7.628 <.0001
##  ambient 4 - warmed 7 -3.54e-02 0.00849 457.5 -4.164 0.0030
##  ambient 4 - ambient 7 -3.98e-02 0.00790 841.4 -5.036 0.0001
##  warmed 5 - ambient 5 -4.41e-03 0.00359 25.9 -1.231 0.9907
##  warmed 5 - warmed 6 -1.04e-02 0.00660 835.6 -1.580 0.9509
##  warmed 5 - ambient 6 -1.48e-02 0.00762 360.1 -1.947 0.7988
##  warmed 5 - warmed 7 -1.80e-03 0.00812 837.5 -0.221 1.0000
##  warmed 5 - ambient 7 -6.21e-03 0.00912 518.6 -0.682 1.0000
##  ambient 5 - warmed 6 -6.02e-03 0.00740 338.9 -0.813 0.9999
##  ambient 5 - ambient 6 -1.04e-02 0.00660 835.6 -1.580 0.9509
##  ambient 5 - warmed 7 2.62e-03 0.00863 481.4 0.303 1.0000
##  ambient 5 - ambient 7 -1.80e-03 0.00812 837.5 -0.221 1.0000
##  warmed 6 - ambient 6 -4.41e-03 0.00359 25.9 -1.231 0.9907
##  warmed 6 - warmed 7 8.63e-03 0.00767 839.7 1.126 0.9977
##  warmed 6 - ambient 7 4.22e-03 0.00862 464.4 0.490 1.0000
##  ambient 6 - warmed 7 1.30e-02 0.00831 437.8 1.570 0.9530
##  ambient 6 - ambient 7 8.63e-03 0.00767 839.7 1.126 0.9977
##  warmed 7 - ambient 7 -4.41e-03 0.00359 25.9 -1.231 0.9907
##
## 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 0.008400 821.4    5.371   5.404
##  2            5.264 0.005027 519.7    5.254   5.274
##  3            5.264 0.005248 537.0    5.254   5.274
##  4            5.253 0.005918 655.2    5.242   5.265
##  5            5.291 0.006358 701.1    5.279   5.304
##  6            5.302 0.005538 656.2    5.291   5.313
##  7            5.293 0.007324 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 0.00831 841 14.908 <.0001
## 1 - 3  1.24e-01 0.00848 842 14.609 <.0001
## 1 - 4  1.34e-01 0.00862 839 15.597 <.0001
## 1 - 5  9.65e-02 0.00882 838 10.932 <.0001
## 1 - 6  8.60e-02 0.00850 839 10.125 <.0001
## 1 - 7  9.47e-02 0.00964 843  9.815 <.0001
## 2 - 3  2.34e-05 0.00547 841  0.004 1.0000
## 2 - 4  1.06e-02 0.00591 840  1.791 0.5547
## 2 - 5 -2.74e-02 0.00625 845 -4.386 0.0003
## 2 - 6 -3.78e-02 0.00568 846 -6.659 <.0001
## 2 - 7 -2.92e-02 0.00750 845 -3.893 0.0020
## 3 - 4  1.06e-02 0.00616 832  1.716 0.6056
## 3 - 5 -2.74e-02 0.00649 843 -4.223 0.0005
## 3 - 6 -3.79e-02 0.00600 846 -6.304 <.0001
## 3 - 7 -2.92e-02 0.00767 846 -3.808 0.0029
## 4 - 5 -3.80e-02 0.00673 836 -5.648 <.0001
## 4 - 6 -4.84e-02 0.00635 838 -7.628 <.0001
## 4 - 7 -3.98e-02 0.00790 841 -5.036 <.0001
## 5 - 6 -1.04e-02 0.00660 836 -1.580 0.6950
## 5 - 7 -1.80e-03 0.00812 838 -0.221 1.0000
## 6 - 7  8.63e-03 0.00767 840  1.126 0.9202
##
## 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 0.009510 766.2    5.342    5.380
##   Arel    5.176 0.004919 533.2    5.167    5.186
##   Bavu    5.130 0.018608 821.7    5.094    5.167
##   Cahí    4.858 0.022248 845.8    4.814    4.902
##   Cest    5.400 0.007671 754.1    5.385    5.415
##   Daca    5.462 0.011690 820.5    5.439    5.485
##   Dagl    5.197 0.005892 667.2    5.186    5.209
##   Elre    5.321 0.006857 758.1    5.308    5.335
##   Eugr    5.575 0.010756 820.3    5.554    5.596
##   Hisp    5.114 0.006759 722.6    5.101    5.128
##   Hype    5.251 0.015654 829.6    5.221    5.282
##   Phpr    5.306 0.005264 567.8    5.296    5.316
##   Poco    5.279 0.015184 839.7    5.249    5.309
##   Popr    5.112 0.004618 543.9    5.103    5.121
##   Pore    5.315 0.009691 827.1    5.296    5.334
##   Rusp    5.189 0.014956 781.0    5.159    5.218
##   Soca    5.591 0.004510 525.4    5.582    5.600
##   Sogr    5.485 0.049135 846.1    5.388    5.581
##   Sora    5.534 0.034813 846.0    5.466    5.603

```

```

##   Sypi      5.651 0.034868 846.0      5.582      5.719
##   Taof      4.909 0.018696 844.5      4.872      4.946
##   Trsp      5.244 0.012369 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  0.18440 0.01071 840  17.212 <.0001
##   Acmi - Bavu  0.23062 0.02073 844  11.127 <.0001
##   Acmi - Cahи  0.50277 0.02397 846  20.978 <.0001
##   Acmi - Cest -0.03920 0.01189 843  -3.297 0.1265
##   Acmi - Daca -0.10153 0.01481 846  -6.857 <.0001
##   Acmi - Dagl  0.16361 0.01107 846  14.779 <.0001
##   Acmi - Elre  0.03950 0.01164 839   3.394 0.0959
##   Acmi - Eugr -0.21438 0.01420 846  -15.094 <.0001
##   Acmi - Hisp  0.24659 0.01138 845  21.672 <.0001
##   Acmi - Hype  0.10963 0.01813 838   6.046 <.0001
##   Acmi - Phpr  0.05482 0.01080 839   5.078 0.0001
##   Acmi - Poco  0.08182 0.01798 845   4.551 0.0013
##   Acmi - Popr  0.24859 0.01045 838  23.789 <.0001
##   Acmi - Pore  0.04613 0.01329 846   3.470 0.0763
##   Acmi - Rusp  0.17215 0.01757 843   9.798 <.0001
##   Acmi - Soca -0.23014 0.01041 843  -22.113 <.0001
##   Acmi - Sogr -0.12390 0.05004 846  -2.476 0.6511
##   Acmi - Sora -0.17331 0.03594 845  -4.822 0.0004
##   Acmi - Sypi -0.28984 0.03615 846  -8.019 <.0001
##   Acmi - Taof  0.45172 0.02076 846  21.762 <.0001
##   Acmi - Trsp  0.11709 0.01525 841   7.676 <.0001
##   Arel - Bavu  0.04622 0.01926 835   2.400 0.7082
##   Arel - Cahи  0.31837 0.02284 844  13.942 <.0001
##   Arel - Cest -0.22360 0.00911 842  -24.538 <.0001
##   Arel - Daca -0.28593 0.01267 846  -22.561 <.0001
##   Arel - Dagl -0.02079 0.00767 846  -2.709 0.4675
##   Arel - Elre -0.14490 0.00848 846  -17.091 <.0001
##   Arel - Eugr -0.39878 0.01182 845  -33.750 <.0001
##   Arel - Hisp  0.06219 0.00834 846   7.457 <.0001
##   Arel - Hype -0.07477 0.01643 844  -4.551 0.0013
##   Arel - Phpr -0.12959 0.00721 840  -17.974 <.0001
##   Arel - Poco -0.10259 0.01590 846  -6.454 <.0001
##   Arel - Popr  0.06418 0.00672 845   9.546 <.0001
##   Arel - Pore -0.13827 0.01081 843  -12.789 <.0001
##   Arel - Rusp -0.01225 0.01570 812  -0.780 1.0000
##   Arel - Soca -0.41455 0.00669 846  -61.963 <.0001
##   Arel - Sogr -0.30830 0.04937 846  -6.245 <.0001
##   Arel - Sora -0.35771 0.03518 846  -10.167 <.0001
##   Arel - Sypi -0.47425 0.03518 846  -13.481 <.0001
##   Arel - Taof  0.26732 0.01932 846  13.838 <.0001
##   Arel - Trsp -0.06731 0.01330 844  -5.061 0.0001
##   Bavu - Cahи  0.27214 0.02885 846   9.434 <.0001
##   Bavu - Cest -0.26983 0.02007 836  -13.446 <.0001

```

```

##  Bavu - Daca -0.33215 0.02186 845 -15.192 <.0001
##  Bavu - Dagl -0.06702 0.01949 839 -3.438 0.0841
##  Bavu - Elre -0.19112 0.01976 838 -9.670 <.0001
##  Bavu - Eugr -0.44500 0.02138 845 -20.819 <.0001
##  Bavu - Hisp 0.01597 0.01970 843 0.811 1.0000
##  Bavu - Hype -0.12099 0.02418 842 -5.004 0.0001
##  Bavu - Phpr -0.17581 0.01932 836 -9.100 <.0001
##  Bavu - Poco -0.14881 0.02397 842 -6.207 <.0001
##  Bavu - Popr 0.01796 0.01915 835 0.938 1.0000
##  Bavu - Pore -0.18449 0.02093 841 -8.815 <.0001
##  Bavu - Rusp -0.05848 0.02375 844 -2.462 0.6620
##  Bavu - Soca -0.46077 0.01910 839 -24.130 <.0001
##  Bavu - Sogr -0.35452 0.05262 846 -6.738 <.0001
##  Bavu - Sora -0.40393 0.03940 845 -10.253 <.0001
##  Bavu - Sypi -0.52047 0.03955 845 -13.161 <.0001
##  Bavu - Taof 0.22110 0.02627 846 8.415 <.0001
##  Bavu - Trsp -0.11353 0.02221 845 -5.111 0.0001
##  Cahи - Cest -0.54197 0.02342 845 -23.139 <.0001
##  Cahи - Daca -0.60430 0.02496 845 -24.206 <.0001
##  Cahи - Dagl -0.33916 0.02297 844 -14.768 <.0001
##  Cahи - Elre -0.46326 0.02320 846 -19.972 <.0001
##  Cahи - Eugr -0.71714 0.02473 845 -28.998 <.0001
##  Cahи - Hisp -0.25617 0.02310 843 -11.089 <.0001
##  Cahи - Hype -0.39314 0.02699 844 -14.563 <.0001
##  Cahи - Phpr -0.44795 0.02285 843 -19.607 <.0001
##  Cahи - Poco -0.42095 0.02702 845 -15.580 <.0001
##  Cahи - Popr -0.25418 0.02268 845 -11.205 <.0001
##  Cahи - Pore -0.45664 0.02421 844 -18.863 <.0001
##  Cahи - Rusp -0.33062 0.02678 846 -12.345 <.0001
##  Cahи - Soca -0.73291 0.02262 845 -32.396 <.0001
##  Cahи - Sogr -0.62667 0.05396 843 -11.613 <.0001
##  Cahи - Sora -0.67608 0.04088 839 -16.538 <.0001
##  Cahи - Sypi -0.79261 0.04144 846 -19.128 <.0001
##  Cahи - Taof -0.05105 0.02902 844 -1.759 0.9805
##  Cahи - Trsp -0.38568 0.02548 843 -15.138 <.0001
##  Cest - Daca -0.06233 0.01376 842 -4.530 0.0014
##  Cest - Dagl 0.20281 0.00956 844 21.206 <.0001
##  Cest - Elre 0.07871 0.01017 844 7.740 <.0001
##  Cest - Eugr -0.17517 0.01306 843 -13.411 <.0001
##  Cest - Hisp 0.28580 0.00992 845 28.809 <.0001
##  Cest - Hype 0.14883 0.01729 840 8.610 <.0001
##  Cest - Phpr 0.09402 0.00918 846 10.240 <.0001
##  Cest - Poco 0.12102 0.01717 846 7.049 <.0001
##  Cest - Popr 0.28779 0.00877 844 32.804 <.0001
##  Cest - Pore 0.08533 0.01206 846 7.076 <.0001
##  Cest - Rusp 0.21135 0.01677 787 12.601 <.0001
##  Cest - Soca -0.19094 0.00877 844 -21.782 <.0001
##  Cest - Sogr -0.08469 0.04966 846 -1.706 0.9863
##  Cest - Sora -0.13411 0.03557 846 -3.770 0.0286
##  Cest - Sypi -0.25064 0.03570 846 -7.021 <.0001
##  Cest - Taof 0.49092 0.02001 845 24.537 <.0001
##  Cest - Trsp 0.15630 0.01416 840 11.040 <.0001
##  Daca - Dagl 0.26514 0.01299 844 20.403 <.0001
##  Daca - Elre 0.14103 0.01346 846 10.478 <.0001

```

```

## Daca - Eugr -0.11285 0.01575 846 -7.165 <.0001
## Daca - Hisp 0.34812 0.01327 846 26.229 <.0001
## Daca - Hype 0.21116 0.01940 844 10.886 <.0001
## Daca - Phpr 0.15635 0.01274 846 12.273 <.0001
## Daca - Poco 0.18334 0.01925 844 9.524 <.0001
## Daca - Popr 0.35011 0.01247 845 28.076 <.0001
## Daca - Pore 0.14766 0.01495 846 9.880 <.0001
## Daca - Rusp 0.27368 0.01887 845 14.506 <.0001
## Daca - Soca -0.12862 0.01244 846 -10.342 <.0001
## Daca - Sogr -0.02237 0.05048 846 -0.443 1.0000
## Daca - Sora -0.07178 0.03659 846 -1.962 0.9393
## Daca - Sypi -0.18832 0.03675 845 -5.124 0.0001
## Daca - Taof 0.55325 0.02191 846 25.254 <.0001
## Daca - Trsp 0.21862 0.01678 843 13.025 <.0001
## Dagl - Elre -0.12410 0.00901 846 -13.777 <.0001
## Dagl - Eugr -0.37798 0.01215 846 -31.105 <.0001
## Dagl - Hisp 0.08299 0.00884 846 9.382 <.0001
## Dagl - Hype -0.05398 0.01665 842 -3.241 0.1471
## Dagl - Phpr -0.10879 0.00784 839 -13.869 <.0001
## Dagl - Poco -0.08179 0.01633 844 -5.008 0.0001
## Dagl - Popr 0.08498 0.00744 846 11.419 <.0001
## Dagl - Pore -0.11748 0.01121 841 -10.477 <.0001
## Dagl - Rusp 0.00854 0.01603 830 0.533 1.0000
## Dagl - Soca -0.39375 0.00736 846 -53.483 <.0001
## Dagl - Sogr -0.28751 0.04947 846 -5.811 <.0001
## Dagl - Sora -0.33692 0.03527 846 -9.553 <.0001
## Dagl - Sypi -0.45345 0.03532 846 -12.837 <.0001
## Dagl - Taof 0.28811 0.01955 846 14.734 <.0001
## Dagl - Trsp -0.04652 0.01359 838 -3.422 0.0884
## Elre - Eugr -0.25388 0.01266 846 -20.061 <.0001
## Elre - Hisp 0.20709 0.00953 846 21.736 <.0001
## Elre - Hype 0.07013 0.01700 836 4.125 0.0076
## Elre - Phpr 0.01531 0.00856 846 1.789 0.9765
## Elre - Poco 0.04231 0.01679 846 2.519 0.6172
## Elre - Popr 0.20908 0.00820 844 25.486 <.0001
## Elre - Pore 0.00662 0.01184 843 0.559 1.0000
## Elre - Rusp 0.13264 0.01649 831 8.042 <.0001
## Elre - Soca -0.26965 0.00810 843 -33.270 <.0001
## Elre - Sogr -0.16340 0.04958 846 -3.295 0.1269
## Elre - Sora -0.21281 0.03545 846 -6.003 <.0001
## Elre - Sypi -0.32935 0.03554 846 -9.266 <.0001
## Elre - Taof 0.41222 0.01991 846 20.704 <.0001
## Elre - Trsp 0.07759 0.01407 841 5.514 <.0001
## Eugr - Hisp 0.46097 0.01249 846 36.915 <.0001
## Eugr - Hype 0.32401 0.01873 844 17.294 <.0001
## Eugr - Phpr 0.26919 0.01189 844 22.650 <.0001
## Eugr - Poco 0.29619 0.01865 845 15.880 <.0001
## Eugr - Popr 0.46296 0.01165 846 39.733 <.0001
## Eugr - Pore 0.26050 0.01425 846 18.286 <.0001
## Eugr - Rusp 0.38652 0.01826 844 21.171 <.0001
## Eugr - Soca -0.01577 0.01153 846 -1.367 0.9993
## Eugr - Sogr 0.09048 0.05034 846 1.797 0.9752
## Eugr - Sora 0.04107 0.03646 846 1.126 1.0000
## Eugr - Sypi -0.07547 0.03629 846 -2.080 0.8971

```

```

##  Eogr - Taof  0.66610 0.02140 845  31.128 <.0001
##  Eogr - Trsp  0.33147 0.01605 845  20.655 <.0001
##  Hisp - Hype -0.13696 0.01688 842  -8.113 <.0001
##  Hisp - Phpr -0.19178 0.00847 843 -22.647 <.0001
##  Hisp - Poco -0.16478 0.01670 846  -9.869 <.0001
##  Hisp - Popr  0.00199 0.00803 845   0.248 1.0000
##  Hisp - Pore -0.20047 0.01150 844 -17.430 <.0001
##  Hisp - Rusp -0.07445 0.01630 818  -4.566 0.0012
##  Hisp - Soca -0.47674 0.00798 844 -59.740 <.0001
##  Hisp - Sogr -0.37049 0.04957 846  -7.474 <.0001
##  Hisp - Sora -0.41990 0.03535 846 -11.878 <.0001
##  Hisp - Sypi -0.53644 0.03543 846 -15.139 <.0001
##  Hisp - Taof  0.20513 0.01972 845  10.401 <.0001
##  Hisp - Trsp -0.12950 0.01383 840  -9.367 <.0001
##  Hype - Phpr -0.05481 0.01646 845  -3.331 0.1149
##  Hype - Poco -0.02782 0.02188 843  -1.271 0.9998
##  Hype - Popr  0.13895 0.01628 842  8.535 <.0001
##  Hype - Pore -0.06350 0.01828 838  -3.473 0.0757
##  Hype - Rusp  0.06252 0.02156 837  2.900 0.3289
##  Hype - Soca -0.33978 0.01619 843 -20.989 <.0001
##  Hype - Sogr -0.23353 0.05163 846  -4.523 0.0014
##  Hype - Sora -0.28294 0.03804 845  -7.439 <.0001
##  Hype - Sypi -0.39947 0.03814 846 -10.473 <.0001
##  Hype - Taof  0.34209 0.02426 843  14.103 <.0001
##  Hype - Trsp  0.00746 0.01973 843  0.378 1.0000
##  Phpr - Poco  0.02700 0.01614 845  1.673 0.9891
##  Phpr - Popr  0.19377 0.00694 845  27.906 <.0001
##  Phpr - Pore -0.00869 0.01094 844  -0.794 1.0000
##  Phpr - Rusp  0.11733 0.01586 818  7.399 <.0001
##  Phpr - Soca -0.28496 0.00687 846 -41.484 <.0001
##  Phpr - Sogr -0.17871 0.04938 846  -3.619 0.0477
##  Phpr - Sora -0.22813 0.03519 846  -6.482 <.0001
##  Phpr - Sypi -0.34466 0.03524 846  -9.779 <.0001
##  Phpr - Taof  0.39690 0.01940 846  20.462 <.0001
##  Phpr - Trsp  0.06228 0.01334 839  4.667 0.0007
##  Poco - Popr  0.16677 0.01589 846  10.497 <.0001
##  Poco - Pore -0.03569 0.01807 843  -1.975 0.9354
##  Poco - Rusp  0.09033 0.02105 845  4.291 0.0039
##  Poco - Soca -0.31196 0.01591 846 -19.604 <.0001
##  Poco - Sogr -0.20571 0.05150 846  -3.994 0.0126
##  Poco - Sora -0.25513 0.03804 846  -6.706 <.0001
##  Poco - Sypi -0.37166 0.03798 845  -9.785 <.0001
##  Poco - Taof  0.36990 0.02406 846  15.372 <.0001
##  Poco - Trsp  0.03528 0.01971 840  1.790 0.9763
##  Popr - Pore -0.20246 0.01059 845 -19.109 <.0001
##  Popr - Rusp -0.07644 0.01562 803  -4.894 0.0003
##  Popr - Soca -0.47873 0.00639 830 -74.925 <.0001
##  Popr - Sogr -0.37248 0.04931 846  -7.555 <.0001
##  Popr - Sora -0.42190 0.03508 846 -12.025 <.0001
##  Popr - Sypi -0.53843 0.03514 846 -15.320 <.0001
##  Popr - Taof  0.20313 0.01920 846  10.582 <.0001
##  Popr - Trsp -0.13149 0.01310 841 -10.036 <.0001
##  Pore - Rusp  0.12602 0.01767 829   7.133 <.0001
##  Pore - Soca -0.27627 0.01059 845 -26.089 <.0001

```

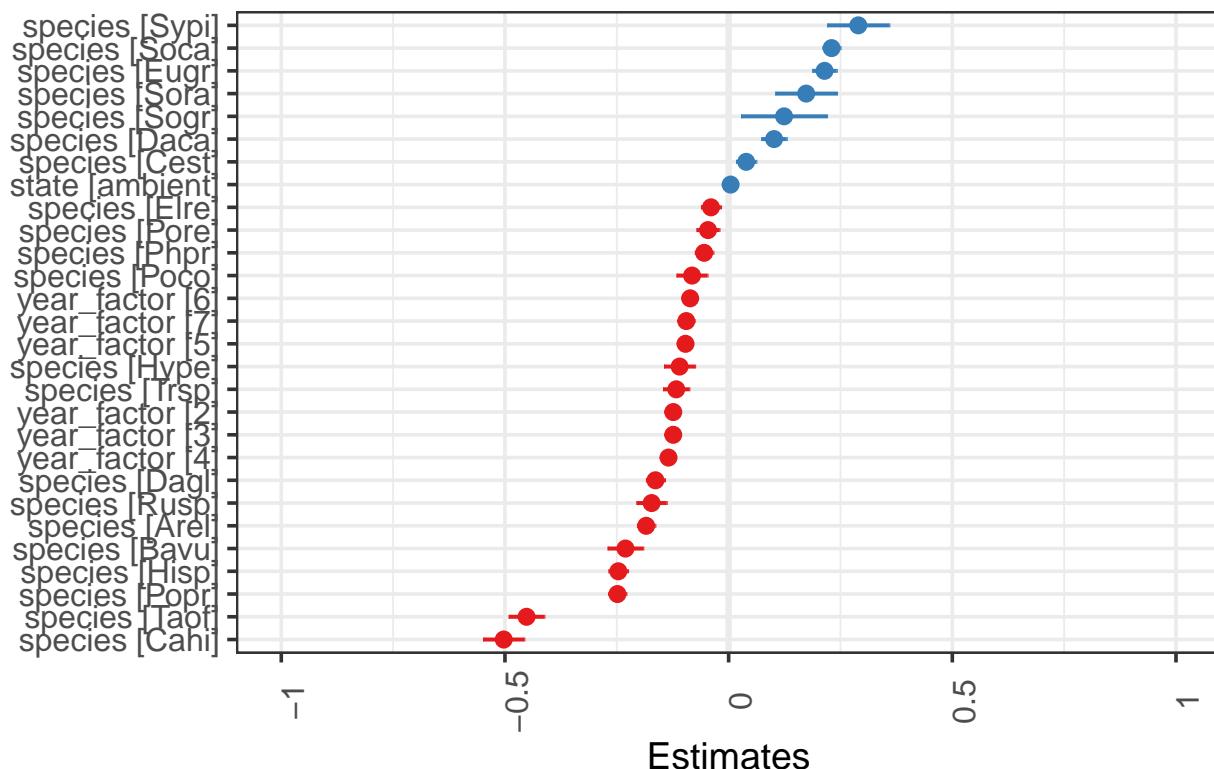
```

## Pore - Sogr -0.17003 0.05004 846 -3.398 0.0948
## Pore - Sora -0.21944 0.03606 846 -6.086 <.0001
## Pore - Sypi -0.33597 0.03604 846 -9.321 <.0001
## Pore - Taof 0.40559 0.02085 846 19.452 <.0001
## Pore - Trsp 0.07096 0.01536 838 4.620 0.0009
## Rusp - Soca -0.40229 0.01560 817 -25.781 <.0001
## Rusp - Sogr -0.29605 0.05138 846 -5.762 <.0001
## Rusp - Sora -0.34546 0.03785 846 -9.126 <.0001
## Rusp - Sypi -0.46199 0.03784 842 -12.210 <.0001
## Rusp - Taof 0.27957 0.02375 846 11.771 <.0001
## Rusp - Trsp -0.05506 0.01925 845 -2.861 0.3555
## Soca - Sogr 0.10625 0.04934 846 2.153 0.8630
## Soca - Sora 0.05683 0.03506 846 1.621 0.9926
## Soca - Sypi -0.05970 0.03512 846 -1.700 0.9869
## Soca - Taof 0.68186 0.01919 846 35.536 <.0001
## Soca - Trsp 0.34724 0.01306 839 26.589 <.0001
## Sogr - Sora -0.04941 0.06015 837 -0.821 1.0000
## Sogr - Sypi -0.16595 0.06023 846 -2.755 0.4324
## Sogr - Taof 0.57562 0.05253 842 10.958 <.0001
## Sogr - Trsp 0.24099 0.05068 846 4.755 0.0005
## Sora - Sypi -0.11653 0.04930 846 -2.364 0.7342
## Sora - Taof 0.62503 0.03944 843 15.847 <.0001
## Sora - Trsp 0.29040 0.03695 846 7.859 <.0001
## Sypi - Taof 0.74156 0.03955 846 18.751 <.0001
## Sypi - Trsp 0.40694 0.03700 846 10.999 <.0001
## Taof - Trsp -0.33463 0.02200 844 -15.211 <.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

# 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)

```

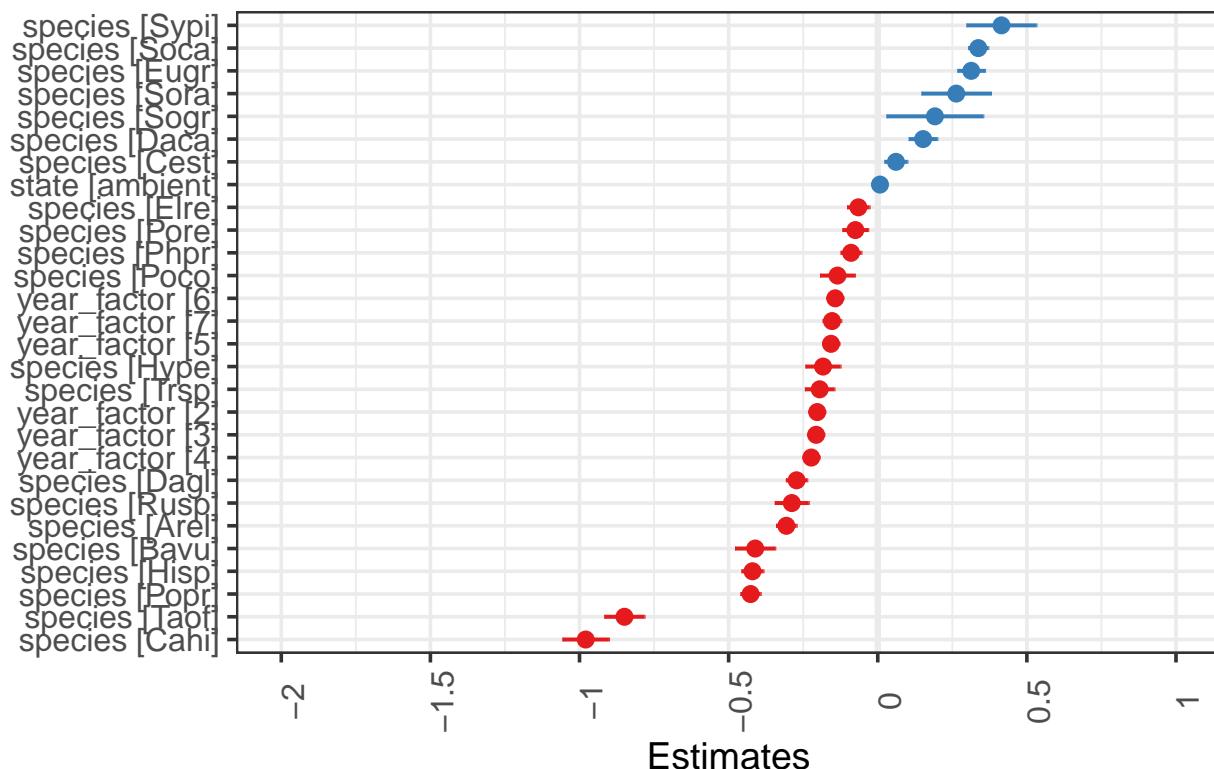
log(julian min)



```
# 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
```

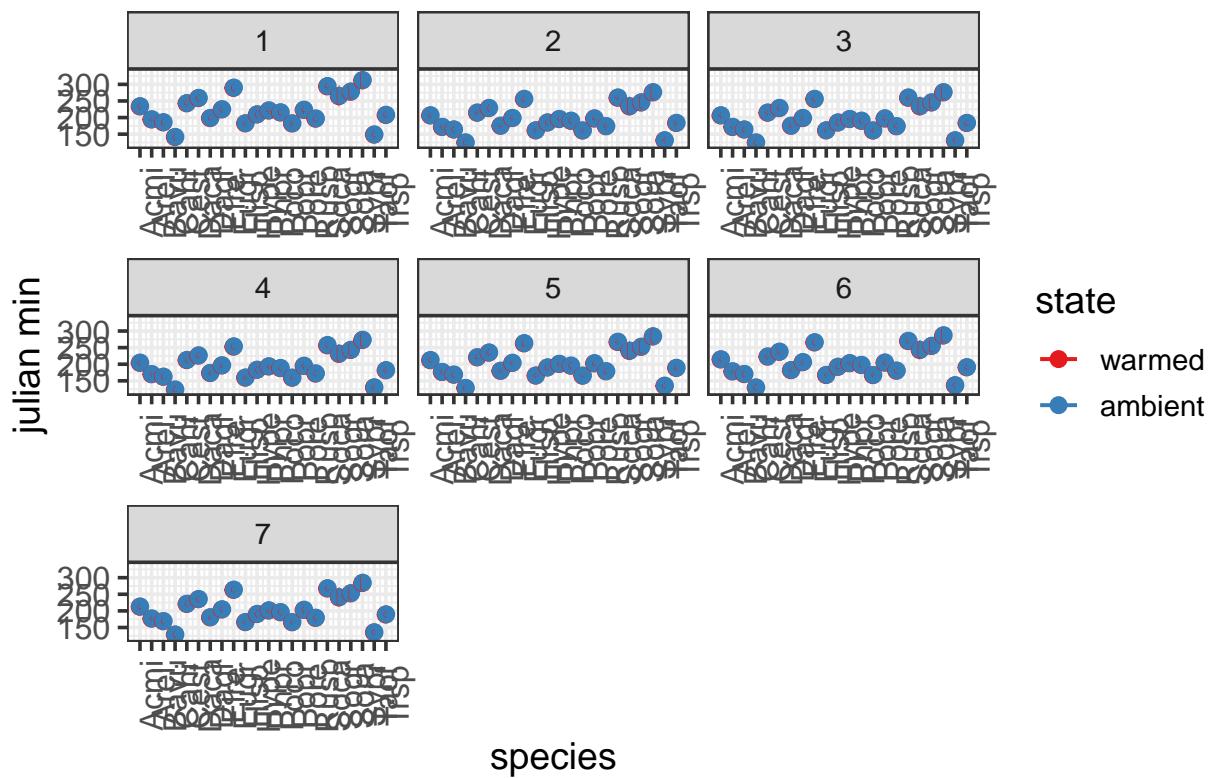
log(julian min)



```
# these are the fixed predicted values: - note this is a new plot
plot_model(mod7a, type = "pred", terms = c("species", "state", "year_factor"))
```

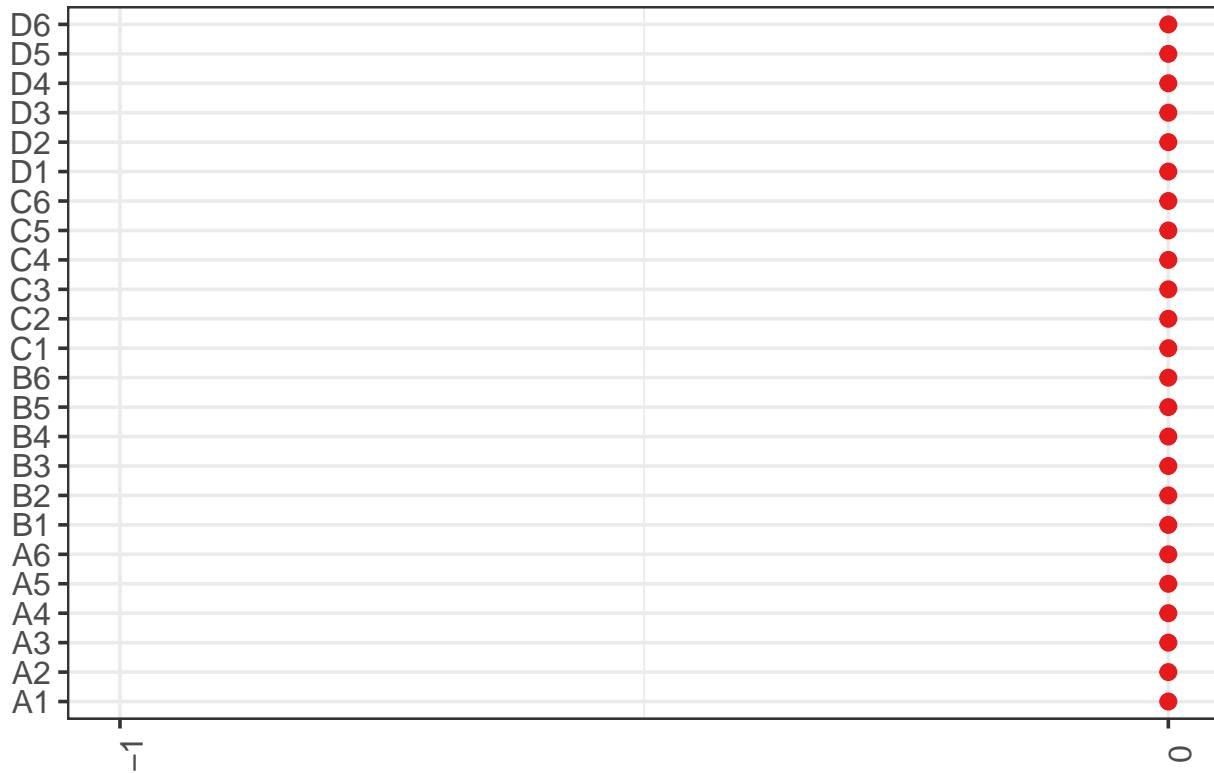
```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standard
```

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) # go with model 9

## 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 -1153.9 -998.79 609.95   -1219.9
## mod8   35 -1150.8 -986.32 610.42   -1220.8 0.9329   2     0.6272
```

```

anova(mod9, mod9a) # mod 9a

## Data: kbs_sd_spp
## Models:
## mod9a: log(julian_min) ~ state + origin + year_factor + (1 | plot)
## mod9: log(julian_min) ~ state + origin + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod9a    12 -1237.3 -1180.92 630.67   -1261.3
## mod9     33 -1153.9  -998.79 609.95   -1219.9      0 21           1

summary(mod9a)

## 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: kbs_sd_spp
##
##      AIC      BIC      logLik deviance df.resid
## -1237.3  -1180.9    630.7   -1261.3      801
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -4.1271 -0.6128  0.0252  0.6213  3.4898
##
## 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+02 304.925 < 2e-16 ***
## stateambient 7.524e-05 7.922e-03 8.130e+02   0.009 0.992425
## originBoth   -4.029e-01 1.498e-02 8.130e+02 -26.902 < 2e-16 ***
## originExotic -3.207e-01 9.783e-03 8.130e+02 -32.784 < 2e-16 ***
## year_factor2 -1.036e-01 1.837e-02 8.130e+02  -5.639 2.36e-08 ***
## year_factor3 -1.281e-01 1.883e-02 8.130e+02  -6.802 2.00e-11 ***
## year_factor4 -1.178e-01 1.940e-02 8.130e+02  -6.069 1.97e-09 ***
## year_factor5 -7.815e-02 1.994e-02 8.130e+02  -3.919 9.65e-05 ***
## year_factor6 -6.982e-02 1.907e-02 8.130e+02  -3.662 0.000266 ***
## year_factor7 -1.138e-01 2.149e-02 8.130e+02  -5.296 1.53e-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
## stateambint -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(mod9a)
```

```

## 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 0.0001 0.9924
## origin     15.2502 7.6251     2     813 614.4807 < 2.2e-16 ***
## year_factor 0.7874 0.1312     6     813 10.5753 2.511e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
emmeans(mod9a, list(pairwise ~ state), adjust = "tukey")
```

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

```

## $`emmeans of state`
##   state emmean       SE   df lower.CL upper.CL
##   warmed  5.307 0.007355 65.76    5.293   5.322
##   ambient  5.308 0.006795 41.47    5.294   5.321
##
## Results are averaged over the levels of: origin, 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`
##   1             estimate       SE   df t.ratio p.value
##   warmed - ambient -7.52e-05 0.008 22.9 -0.009  0.9926
##
## Results are averaged over the levels of: origin, year_factor
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.

```

```
emmeans(mod9a, list(pairwise ~ origin), adjust = "tukey")
```

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

```

## $`emmeans of origin`
##   origin emmean       SE   df lower.CL upper.CL
##   Native  5.549 0.008870 345.58    5.531   5.566
##   Both    5.146 0.012896 603.76    5.120   5.171
##   Exotic  5.228 0.005064  56.82    5.218   5.238
##
## 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 origin'
##   1           estimate      SE  df t.ratio p.value
## Native - Both    0.4029 0.01510 822 26.686 <.0001
## Native - Exotic  0.3207 0.00986 823 32.516 <.0001
## Both - Exotic   -0.0821 0.01340 823 -6.132 <.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 3 estimates

emmeans(mod9a, list(pairwise ~ state + origin), adjust = "tukey")

## boundary (singular) fit: see ?isSingular

## $'emmmeans of state, origin'
##   state  origin emmean      SE  df lower.CL upper.CL
## warmed  Native  5.549 0.009665 172.72    5.530    5.568
## ambient Native  5.549 0.009795 163.67    5.529    5.568
## warmed  Both    5.146 0.013851 433.30    5.119    5.173
## ambient Both    5.146 0.013143 336.34    5.120    5.172
## warmed  Exotic  5.228 0.006726  46.92    5.214    5.241
## ambient Exotic  5.228 0.006169  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 -7.52e-05 0.00800 22.9 -0.009  1.0000
## warmed Native - warmed Both    4.03e-01 0.01510 821.8 26.686 <.0001
## warmed Native - ambient Both   4.03e-01 0.01644 273.4 24.501 <.0001
## warmed Native - warmed Exotic 3.21e-01 0.00986 822.6 32.516 <.0001
## warmed Native - ambient Exotic 3.21e-01 0.01231 116.9 26.046 <.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 26.686 <.0001
## ambient Native - warmed Exotic 3.21e-01 0.01308 140.0 24.531 <.0001
## ambient Native - ambient Exotic 3.21e-01 0.00986 822.6 32.516 <.0001
## warmed Both - ambient Both     -7.52e-05 0.00800 22.9 -0.009  1.0000
## warmed Both - warmed Exotic   -8.21e-02 0.01340 823.1 -6.132 <.0001
## warmed Both - ambient Exotic  -8.22e-02 0.01598 274.6 -5.145 <.0001
## ambient Both - warmed Exotic -8.21e-02 0.01522 209.6 -5.394 <.0001
## ambient Both - ambient Exotic -8.21e-02 0.01340 823.1 -6.132 <.0001
## warmed Exotic - ambient Exotic -7.52e-05 0.00800 22.9 -0.009  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
kbs_sd_spp <- within(kbs_sd_spp, growth_habit <- relevel(factor(growth_habit), ref = "Forb")) # relevance
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

## Warning: Model failed to converge with 1 negative eigenvalue: -6.6e+00

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 11

## 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 -936.43 -781.30 501.21   -1002.4
## mod10    35 -932.94 -768.42 501.47   -1002.9 0.5143  2     0.7733

anova(mod11, mod11a) # model 11a

## Data: kbs_sd_spp
## Models:
## mod11a: log(julian_min) ~ state + growth_habit + year_factor + (1 | plot)
## mod11: log(julian_min) ~ state + growth_habit + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod11a   12 -1031.26 -974.85 527.63   -1055.3
## mod11    33 -936.43 -781.30 501.21   -1002.4      0 21           1

summary(mod11a)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + growth_habit + year_factor + (1 | plot)
##   Data: kbs_sd_spp
##
##      AIC      BIC logLik deviance df.resid
## -1031.3   -974.9    527.6   -1055.3      801
##

```

```

## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -5.3655 -0.6249 -0.0294  0.7470  3.2410
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   plot      (Intercept) 0.000000 0.0000
##   Residual           0.01599  0.1264
## Number of obs: 813, groups: plot, 24
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)
##   (Intercept)             5.591809  0.020737 813.000000 269.653 < 2e-16 ***
##   stateambient          -0.018564  0.008977 813.000000 -2.068   0.039 *
##   growth_habit          -0.298653  0.015750 813.000000 -18.962 < 2e-16 ***
##   growth_habitGraminoid -0.251913  0.009865 813.000000 -25.536 < 2e-16 ***
##   year_factor2          -0.164930  0.021066 813.000000 -7.829 1.53e-14 ***
##   year_factor3          -0.185967  0.021555 813.000000 -8.628 < 2e-16 ***
##   year_factor4          -0.154719  0.022043 813.000000 -7.019 4.71e-12 ***
##   year_factor5          -0.107017  0.022658 813.000000 -4.723 2.73e-06 ***
##   year_factor6          -0.093088  0.021709 813.000000 -4.288 2.02e-05 ***
##   year_factor7          -0.130013  0.024385 813.000000 -5.332 1.26e-07 ***
##   ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) sttmbn grwth_G yr_fc2 yr_fc3 yr_fc4 yr_fc5 yr_fc6
## stateambint -0.230
## growth_habit -0.112 -0.063
## grwth_hbtGr -0.375  0.012  0.340
## year_factr2 -0.847 -0.012 -0.065  0.146
## year_factr3 -0.826 -0.001 -0.032  0.131  0.795
## year_factr4 -0.772 -0.018 -0.016  0.043  0.759  0.740
## year_factr5 -0.749 -0.032 -0.027  0.044  0.741  0.721  0.700
## year_factr6 -0.803 -0.009 -0.035  0.089  0.782  0.760  0.733  0.714
## year_factr7 -0.689  0.019 -0.039 -0.008  0.680  0.663  0.647  0.630  0.658
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

```

```
anova(mod11a)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##                   Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state            0.0684  0.0684     1    813  4.2768 0.03895 *
## growth_habit  12.3397  6.1699     2    813 385.8837 < 2e-16 ***
## year_factor    1.7126  0.2854     6    813 17.8519 < 2e-16 ***
##   ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
emmeans(mod11a, list(pairwise ~ factor(year_factor) * growth_habit), adjust = "tukey")
```

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

```

## $`$`emmeans of year_factor, growth_habit`
##   year_factor growth_habit emmean      SE    df lower.CL upper.CL
## 1           Forb      5.583 0.02034 784.2    5.543    5.622
## 2           Forb      5.418 0.01049 418.8    5.397    5.438
## 3           Forb      5.397 0.01162 479.4    5.374    5.419
## 4           Forb      5.428 0.01365 620.2    5.401    5.455
## 5           Forb      5.476 0.01463 638.1    5.447    5.504
## 6           Forb      5.489 0.01250 513.1    5.465    5.514
## 7           Forb      5.453 0.01778 687.2    5.418    5.487
## 1           Forb      5.284 0.02413 807.5    5.237    5.331
## 2           Forb      5.119 0.01534 665.6    5.089    5.149
## 3           Forb      5.098 0.01680 696.1    5.065    5.131
## 4           Forb      5.129 0.01854 740.8    5.093    5.166
## 5           Forb      5.177 0.01907 741.9    5.139    5.214
## 6           Forb      5.191 0.01736 694.8    5.157    5.225
## 7           Forb      5.154 0.02134 727.3    5.112    5.196
## 1 Graminoid     5.331 0.01891 771.6    5.293    5.368
## 2 Graminoid     5.166 0.01077 409.9    5.145    5.187
## 3 Graminoid     5.145 0.01166 471.9    5.122    5.168
## 4 Graminoid     5.176 0.01221 552.6    5.152    5.200
## 5 Graminoid     5.224 0.01332 611.7    5.197    5.250
## 6 Graminoid     5.238 0.01179 496.7    5.214    5.261
## 7 Graminoid     5.201 0.01597 684.5    5.169    5.232
##
## 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 year_factor, growth_habit`
##   1          estimate      SE    df t.ratio p.value
## 1 Forb - 2 Forb 0.164930 0.02123 818 7.770 <.0001
## 1 Forb - 3 Forb 0.185967 0.02172 819 8.561 <.0001
## 1 Forb - 4 Forb 0.154719 0.02220 816 6.968 <.0001
## 1 Forb - 5 Forb 0.107017 0.02282 816 4.689 0.0006
## 1 Forb - 6 Forb 0.093088 0.02187 817 4.257 0.0041
## 1 Forb - 7 Forb 0.130013 0.02459 822 5.288 <.0001
## 1 Forb - 1      0.298653 0.01588 821 18.812 <.0001
## 1 Forb - 2      0.463583 0.02568 821 18.054 <.0001
## 1 Forb - 3      0.484619 0.02650 822 18.287 <.0001
## 1 Forb - 4      0.453371 0.02709 821 16.734 <.0001
## 1 Forb - 5      0.405669 0.02746 820 14.774 <.0001
## 1 Forb - 6      0.391741 0.02658 821 14.736 <.0001
## 1 Forb - 7      0.428665 0.02876 823 14.903 <.0001
## 1 Forb - 1 Graminoid 0.251913 0.00995 823 25.315 <.0001
## 1 Forb - 2 Graminoid 0.416843 0.02473 822 16.855 <.0001
## 1 Forb - 3 Graminoid 0.437879 0.02506 821 17.477 <.0001
## 1 Forb - 4 Graminoid 0.406632 0.02472 818 16.451 <.0001
## 1 Forb - 5 Graminoid 0.358930 0.02529 816 14.194 <.0001
## 1 Forb - 6 Graminoid 0.345001 0.02482 818 13.901 <.0001
## 1 Forb - 7 Graminoid 0.381926 0.02643 819 14.448 <.0001
## 2 Forb - 3 Forb 0.021036 0.01376 817 1.528 0.9953
## 2 Forb - 4 Forb -0.010212 0.01509 816 -0.677 1.0000
## 2 Forb - 5 Forb -0.057914 0.01594 821 -3.633 0.0421

```

## 2 Forb - 6 Forb	-0.071842	0.01426	823	-5.038	0.0001
## 2 Forb - 7 Forb	-0.034918	0.01862	823	-1.875	0.9534
## 2 Forb - 1	0.133722	0.02731	817	4.897	0.0002
## 2 Forb - 2	0.298653	0.01588	821	18.812	<.0001
## 2 Forb - 3	0.319689	0.02152	820	14.855	<.0001
## 2 Forb - 4	0.288441	0.02263	820	12.747	<.0001
## 2 Forb - 5	0.240739	0.02303	821	10.454	<.0001
## 2 Forb - 6	0.226811	0.02180	822	10.406	<.0001
## 2 Forb - 7	0.263735	0.02475	823	10.654	<.0001
## 2 Forb - 1 Graminoid	0.086983	0.02208	817	3.940	0.0143
## 2 Forb - 2 Graminoid	0.251913	0.00995	823	25.315	<.0001
## 2 Forb - 3 Graminoid	0.272949	0.01683	818	16.217	<.0001
## 2 Forb - 4 Graminoid	0.241701	0.01684	814	14.350	<.0001
## 2 Forb - 5 Graminoid	0.193999	0.01762	815	11.009	<.0001
## 2 Forb - 6 Graminoid	0.180071	0.01670	818	10.781	<.0001
## 2 Forb - 7 Graminoid	0.216995	0.01946	820	11.149	<.0001
## 3 Forb - 4 Forb	-0.031248	0.01583	810	-1.974	0.9247
## 3 Forb - 5 Forb	-0.078950	0.01666	820	-4.738	0.0005
## 3 Forb - 6 Forb	-0.092878	0.01510	822	-6.151	<.0001
## 3 Forb - 7 Forb	-0.055954	0.01919	823	-2.915	0.3008
## 3 Forb - 1	0.112686	0.02730	818	4.127	0.0069
## 3 Forb - 2	0.277616	0.02049	819	13.550	<.0001
## 3 Forb - 3	0.298653	0.01588	821	18.812	<.0001
## 3 Forb - 4	0.267405	0.02265	818	11.806	<.0001
## 3 Forb - 5	0.219703	0.02307	820	9.523	<.0001
## 3 Forb - 6	0.205774	0.02186	822	9.412	<.0001
## 3 Forb - 7	0.242699	0.02476	823	9.804	<.0001
## 3 Forb - 1 Graminoid	0.065946	0.02267	820	2.909	0.3047
## 3 Forb - 2 Graminoid	0.230877	0.01713	822	13.474	<.0001
## 3 Forb - 3 Graminoid	0.251913	0.00995	823	25.315	<.0001
## 3 Forb - 4 Graminoid	0.220665	0.01765	812	12.500	<.0001
## 3 Forb - 5 Graminoid	0.172963	0.01842	817	9.390	<.0001
## 3 Forb - 6 Graminoid	0.159035	0.01757	820	9.050	<.0001
## 3 Forb - 7 Graminoid	0.195959	0.02014	818	9.729	<.0001
## 4 Forb - 5 Forb	-0.047702	0.01744	813	-2.735	0.4261
## 4 Forb - 6 Forb	-0.061630	0.01611	814	-3.826	0.0217
## 4 Forb - 7 Forb	-0.024706	0.01977	819	-1.250	0.9997
## 4 Forb - 1	0.143934	0.02750	815	5.234	<.0001
## 4 Forb - 2	0.308864	0.02116	818	14.598	<.0001
## 4 Forb - 3	0.329901	0.02218	816	14.871	<.0001
## 4 Forb - 4	0.298653	0.01588	821	18.812	<.0001
## 4 Forb - 5	0.250951	0.02342	816	10.717	<.0001
## 4 Forb - 6	0.237022	0.02234	818	10.611	<.0001
## 4 Forb - 7	0.273947	0.02500	822	10.960	<.0001
## 4 Forb - 1 Graminoid	0.097194	0.02394	819	4.060	0.0090
## 4 Forb - 2 Graminoid	0.262125	0.01923	822	13.628	<.0001
## 4 Forb - 3 Graminoid	0.283161	0.01969	819	14.383	<.0001
## 4 Forb - 4 Graminoid	0.251913	0.00995	823	25.315	<.0001
## 4 Forb - 5 Graminoid	0.204211	0.02010	814	10.162	<.0001
## 4 Forb - 6 Graminoid	0.190283	0.01945	817	9.784	<.0001
## 4 Forb - 7 Graminoid	0.227207	0.02159	813	10.524	<.0001
## 5 Forb - 6 Forb	-0.013928	0.01692	814	-0.823	1.0000
## 5 Forb - 7 Forb	0.022996	0.02045	817	1.124	0.9999
## 5 Forb - 1	0.191636	0.02814	816	6.810	<.0001

## 5 Forb - 2	0.356566	0.02195	821	16.243	<.0001
## 5 Forb - 3	0.377603	0.02296	821	16.446	<.0001
## 5 Forb - 4	0.346355	0.02375	818	14.582	<.0001
## 5 Forb - 5	0.298653	0.01588	821	18.812	<.0001
## 5 Forb - 6	0.284724	0.02310	818	12.326	<.0001
## 5 Forb - 7	0.321649	0.02570	821	12.518	<.0001
## 5 Forb - 1 Graminoid	0.144896	0.02450	820	5.913	<.0001
## 5 Forb - 2 Graminoid	0.309827	0.01989	823	15.576	<.0001
## 5 Forb - 3 Graminoid	0.330863	0.02035	823	16.259	<.0001
## 5 Forb - 4 Graminoid	0.299615	0.02007	820	14.930	<.0001
## 5 Forb - 5 Graminoid	0.251913	0.00995	823	25.315	<.0001
## 5 Forb - 6 Graminoid	0.237985	0.02011	819	11.835	<.0001
## 5 Forb - 7 Graminoid	0.274909	0.02221	815	12.380	<.0001
## 6 Forb - 7 Forb	0.036924	0.01936	819	1.907	0.9451
## 6 Forb - 1	0.205564	0.02746	816	7.487	<.0001
## 6 Forb - 2	0.370495	0.02087	821	17.750	<.0001
## 6 Forb - 3	0.391531	0.02196	821	17.833	<.0001
## 6 Forb - 4	0.360283	0.02289	818	15.740	<.0001
## 6 Forb - 5	0.312581	0.02330	817	13.418	<.0001
## 6 Forb - 6	0.298653	0.01588	821	18.812	<.0001
## 6 Forb - 7	0.335577	0.02492	822	13.464	<.0001
## 6 Forb - 1 Graminoid	0.158825	0.02321	820	6.844	<.0001
## 6 Forb - 2 Graminoid	0.323755	0.01805	822	17.939	<.0001
## 6 Forb - 3 Graminoid	0.344791	0.01858	823	18.556	<.0001
## 6 Forb - 4 Graminoid	0.313543	0.01840	820	17.036	<.0001
## 6 Forb - 5 Graminoid	0.265841	0.01913	817	13.896	<.0001
## 6 Forb - 6 Graminoid	0.251913	0.00995	823	25.315	<.0001
## 6 Forb - 7 Graminoid	0.288837	0.02074	814	13.924	<.0001
## 7 Forb - 1	0.168640	0.02976	818	5.666	<.0001
## 7 Forb - 2	0.333570	0.02418	822	13.796	<.0001
## 7 Forb - 3	0.354606	0.02506	821	14.150	<.0001
## 7 Forb - 4	0.323359	0.02571	817	12.578	<.0001
## 7 Forb - 5	0.275657	0.02609	815	10.567	<.0001
## 7 Forb - 6	0.261728	0.02515	816	10.408	<.0001
## 7 Forb - 7	0.298653	0.01588	821	18.812	<.0001
## 7 Forb - 1 Graminoid	0.121900	0.02661	823	4.580	0.0010
## 7 Forb - 2 Graminoid	0.286830	0.02264	817	12.670	<.0001
## 7 Forb - 3 Graminoid	0.307867	0.02300	821	13.384	<.0001
## 7 Forb - 4 Graminoid	0.276619	0.02266	823	12.206	<.0001
## 7 Forb - 5 Graminoid	0.228917	0.02327	822	9.836	<.0001
## 7 Forb - 6 Graminoid	0.214988	0.02274	823	9.453	<.0001
## 7 Forb - 7 Graminoid	0.251913	0.00995	823	25.315	<.0001
## 1 - 2	0.164930	0.02123	818	7.770	<.0001
## 1 - 3	0.185967	0.02172	819	8.561	<.0001
## 1 - 4	0.154719	0.02220	816	6.968	<.0001
## 1 - 5	0.107017	0.02282	816	4.689	0.0006
## 1 - 6	0.093088	0.02187	817	4.257	0.0041
## 1 - 7	0.130013	0.02459	822	5.288	<.0001
## 1 - 1 Graminoid	-0.046740	0.01563	822	-2.990	0.2557
## 1 - 2 Graminoid	0.118191	0.02829	821	4.178	0.0057
## 1 - 3 Graminoid	0.139227	0.02819	821	4.939	0.0002
## 1 - 4 Graminoid	0.107979	0.02770	818	3.898	0.0167
## 1 - 5 Graminoid	0.060277	0.02835	817	2.126	0.8604
## 1 - 6 Graminoid	0.046349	0.02802	818	1.654	0.9878

##	1	-	7	Graminoid	0.083273	0.02955	817	2.818	0.3658
##	2	-	3		0.021036	0.01376	817	1.528	0.9953
##	2	-	4		-0.010212	0.01509	816	-0.677	1.0000
##	2	-	5		-0.057914	0.01594	821	-3.633	0.0421
##	2	-	6		-0.071842	0.01426	823	-5.038	0.0001
##	2	-	7		-0.034918	0.01862	823	-1.875	0.9534
##	2	-	1	Graminoid	-0.211670	0.02428	822	-8.717	<.0001
##	2	-	2	Graminoid	-0.046740	0.01563	822	-2.990	0.2557
##	2	-	3	Graminoid	-0.025704	0.02017	822	-1.274	0.9996
##	2	-	4	Graminoid	-0.056952	0.01992	820	-2.859	0.3378
##	2	-	5	Graminoid	-0.104653	0.02078	821	-5.037	0.0001
##	2	-	6	Graminoid	-0.118582	0.02012	822	-5.895	<.0001
##	2	-	7	Graminoid	-0.081657	0.02258	819	-3.616	0.0447
##	3	-	4		-0.031248	0.01583	810	-1.974	0.9247
##	3	-	5		-0.078950	0.01666	820	-4.738	0.0005
##	3	-	6		-0.092878	0.01510	822	-6.151	<.0001
##	3	-	7		-0.055954	0.01919	823	-2.915	0.3008
##	3	-	1	Graminoid	-0.232706	0.02525	823	-9.215	<.0001
##	3	-	2	Graminoid	-0.067776	0.02146	823	-3.158	0.1708
##	3	-	3	Graminoid	-0.046740	0.01563	822	-2.990	0.2557
##	3	-	4	Graminoid	-0.077988	0.02113	819	-3.691	0.0348
##	3	-	5	Graminoid	-0.125690	0.02196	821	-5.724	<.0001
##	3	-	6	Graminoid	-0.139618	0.02136	822	-6.537	<.0001
##	3	-	7	Graminoid	-0.102694	0.02364	818	-4.345	0.0028
##	4	-	5		-0.047702	0.01744	813	-2.735	0.4261
##	4	-	6		-0.061630	0.01611	814	-3.826	0.0217
##	4	-	7		-0.024706	0.01977	819	-1.250	0.9997
##	4	-	1	Graminoid	-0.201459	0.02660	823	-7.574	<.0001
##	4	-	2	Graminoid	-0.036528	0.02340	823	-1.561	0.9939
##	4	-	3	Graminoid	-0.015492	0.02331	822	-0.665	1.0000
##	4	-	4	Graminoid	-0.046740	0.01563	822	-2.990	0.2557
##	4	-	5	Graminoid	-0.094442	0.02360	820	-4.001	0.0113
##	4	-	6	Graminoid	-0.108370	0.02315	821	-4.681	0.0006
##	4	-	7	Graminoid	-0.071446	0.02509	816	-2.847	0.3454
##	5	-	6		-0.013928	0.01692	814	-0.823	1.0000
##	5	-	7		0.022996	0.02045	817	1.124	0.9999
##	5	-	1	Graminoid	-0.153756	0.02696	823	-5.703	<.0001
##	5	-	2	Graminoid	0.011174	0.02378	823	0.470	1.0000
##	5	-	3	Graminoid	0.032210	0.02371	823	1.359	0.9990
##	5	-	4	Graminoid	0.000962	0.02324	822	0.041	1.0000
##	5	-	5	Graminoid	-0.046740	0.01563	822	-2.990	0.2557
##	5	-	6	Graminoid	-0.060668	0.02354	822	-2.577	0.5484
##	5	-	7	Graminoid	-0.023744	0.02547	815	-0.932	1.0000
##	6	-	7		0.036924	0.01936	819	1.907	0.9451
##	6	-	1	Graminoid	-0.139828	0.02570	823	-5.441	<.0001
##	6	-	2	Graminoid	0.025102	0.02215	822	1.133	0.9999
##	6	-	3	Graminoid	0.046138	0.02210	823	2.087	0.8792
##	6	-	4	Graminoid	0.014890	0.02172	822	0.686	1.0000
##	6	-	5	Graminoid	-0.032812	0.02251	822	-1.458	0.9974
##	6	-	6	Graminoid	-0.046740	0.01563	822	-2.990	0.2557
##	6	-	7	Graminoid	-0.009815	0.02411	816	-0.407	1.0000
##	7	-	1	Graminoid	-0.176753	0.02872	821	-6.155	<.0001
##	7	-	2	Graminoid	-0.011822	0.02592	815	-0.456	1.0000
##	7	-	3	Graminoid	0.009214	0.02582	819	0.357	1.0000

```

## 7 - 4 Graminoid      -0.022034 0.02532 822 -0.870  1.0000
## 7 - 5 Graminoid      -0.069736 0.02602 823 -2.680  0.4675
## 7 - 6 Graminoid      -0.083664 0.02563 823 -3.264  0.1290
## 7 - 7 Graminoid      -0.046740 0.01563 822 -2.990  0.2557
## 1 Graminoid - 2 Graminoid  0.164930 0.02123 818  7.770 <.0001
## 1 Graminoid - 3 Graminoid  0.185967 0.02172 819  8.561 <.0001
## 1 Graminoid - 4 Graminoid  0.154719 0.02220 816  6.968 <.0001
## 1 Graminoid - 5 Graminoid  0.107017 0.02282 816  4.689  0.0006
## 1 Graminoid - 6 Graminoid  0.093088 0.02187 817  4.257  0.0041
## 1 Graminoid - 7 Graminoid  0.130013 0.02459 822  5.288 <.0001
## 2 Graminoid - 3 Graminoid  0.021036 0.01376 817  1.528  0.9953
## 2 Graminoid - 4 Graminoid -0.010212 0.01509 816 -0.677  1.0000
## 2 Graminoid - 5 Graminoid -0.057914 0.01594 821 -3.633  0.0421
## 2 Graminoid - 6 Graminoid -0.071842 0.01426 823 -5.038  0.0001
## 2 Graminoid - 7 Graminoid -0.034918 0.01862 823 -1.875  0.9534
## 3 Graminoid - 4 Graminoid -0.031248 0.01583 810 -1.974  0.9247
## 3 Graminoid - 5 Graminoid -0.078950 0.01666 820 -4.738  0.0005
## 3 Graminoid - 6 Graminoid -0.092878 0.01510 822 -6.151 <.0001
## 3 Graminoid - 7 Graminoid -0.055954 0.01919 823 -2.915  0.3008
## 4 Graminoid - 5 Graminoid -0.047702 0.01744 813 -2.735  0.4261
## 4 Graminoid - 6 Graminoid -0.061630 0.01611 814 -3.826  0.0217
## 4 Graminoid - 7 Graminoid -0.024706 0.01977 819 -1.250  0.9997
## 5 Graminoid - 6 Graminoid -0.013928 0.01692 814 -0.823  1.0000
## 5 Graminoid - 7 Graminoid  0.022996 0.02045 817  1.124  0.9999
## 6 Graminoid - 7 Graminoid  0.036924 0.01936 819  1.907  0.9451
##
## 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 21 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
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_sd_spp, REML = FALSE)
mod6 <- lmer(log(julian_min) ~ state + year_factor + (1 | species) + (1 + year |
  plot), kbs_sd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

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 1 negative eigenvalue: -8.6e-02

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

## Warning: Model failed to converge with 1 negative eigenvalue: -6.6e+00

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
## -2580.7 -2434.8    1321.3   -2642.7      785
##
## Scaled residuals:
##      Min      1Q Median      3Q     Max
## -4.5905 -0.5311 -0.0057  0.4910  7.6698
##

```

```

## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.0000000 0.00000
## Residual          0.002296 0.04792
## Number of obs: 816, groups: plot, 24
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                5.452873  0.011998 816.000000 454.472 < 2e-16 ***
## stateambient               0.004415  0.003510 816.000000  1.258 0.208799
## speciesArel                -0.184402  0.010473 816.000000 -17.608 < 2e-16 ***
## speciesBavu                 -0.230624  0.020319 816.000000 -11.350 < 2e-16 ***
## speciesCahi                 -0.502768  0.023483 816.000000 -21.409 < 2e-16 ***
## speciesCest                 0.039203  0.011627 816.000000  3.372 0.000782 ***
## speciesDaca                 0.101529  0.014505 816.000000  6.999 5.37e-12 ***
## speciesDagl                 -0.163609  0.010838 816.000000 -15.095 < 2e-16 ***
## speciesElre                 -0.039505  0.011380 816.000000 -3.471 0.000545 ***
## speciesEugr                  0.214376  0.013910 816.000000 15.411 < 2e-16 ***
## speciesHisp                 -0.246595  0.011135 816.000000 -22.146 < 2e-16 ***
## speciesHype                 -0.109631  0.017735 816.000000 -6.182 1.00e-09 ***
## speciesPhpr                 -0.054816  0.010555 816.000000 -5.193 2.61e-07 ***
## speciesPoco                 -0.081816  0.017619 816.000000 -4.644 3.99e-06 ***
## speciesPopr                 -0.248585  0.010210 816.000000 -24.348 < 2e-16 ***
## speciesPore                 -0.046129  0.013021 816.000000 -3.543 0.000419 ***
## speciesRusp                 -0.172149  0.017193 816.000000 -10.013 < 2e-16 ***
## speciesSoca                 0.230145  0.010176 816.000000 22.616 < 2e-16 ***
## speciesSogr                 0.123898  0.048999 816.000000  2.529 0.011640 *
## speciesSora                 0.173310  0.035193 816.000000  4.925 1.02e-06 ***
## speciesSypi                 0.289844  0.035408 816.000000  8.186 1.04e-15 ***
## speciesTaof                 -0.451720  0.020330 816.000000 -22.220 < 2e-16 ***
## speciesTrsp                 -0.117093  0.014959 816.000000 -7.828 1.54e-14 ***
## factor(year_factor)2        -0.123863  0.008148 816.000000 -15.201 < 2e-16 ***
## factor(year_factor)3        -0.123886  0.008316 816.000000 -14.897 < 2e-16 ***
## factor(year_factor)4        -0.134448  0.008456 816.000000 -15.900 < 2e-16 ***
## factor(year_factor)5        -0.096464  0.008657 816.000000 -11.143 < 2e-16 ***
## factor(year_factor)6        -0.086031  0.008335 816.000000 -10.321 < 2e-16 ***
## factor(year_factor)7        -0.094665  0.009456 816.000000 -10.012 < 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  0.00363     1    816   1.5822 0.2088

```

```

## species          23.6359 1.12552    21    816 490.1339 <2e-16 ***
## factor(year_factor) 0.7372 0.12286     6    816  53.5021 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

KBS Plot-level Mixed Effects Models

```
mod1p <- lmer(log(julian_min) ~ state + (1 | plot), kbs_sd_plot, REML = FALSE)
```

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

```
mod2p <- lmer(log(julian_min) ~ insecticide + (1 | plot), kbs_sd_plot, REML = FALSE)
```

```
mod3p <- lmer(log(julian_min) ~ insecticide + state + (1 | plot), kbs_sd_plot, REML = FALSE)
```

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

```
mod4p <- lmer(log(julian_min) ~ insecticide * state + (1 | plot), kbs_sd_plot, REML = FALSE)
```

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

```
mod5p <- lmer(log(julian_min) ~ state + year_factor + (1 | plot), kbs_sd_plot, REML = FALSE)
```

```
mod6p <- lmer(log(julian_min) ~ state + year_factor + insecticide + (1 | plot), kbs_sd_plot,
REML = FALSE)
```

```
mod7p <- lmer(log(julian_min) ~ state * year_factor + (1 | plot), kbs_sd_plot, REML = FALSE)
```

```
mod8p <- lmer(log(julian_min) ~ state * year_factor + insecticide + (1 | plot), kbs_sd_plot,
REML = FALSE)
```

```
mod9p <- lmer(log(julian_min) ~ state * insecticide + year_factor + (1 | plot), kbs_sd_plot,
REML = FALSE)
```

```
mod10p <- lmer(log(julian_min) ~ state + insecticide * year_factor + (1 | plot),
kbs_sd_plot, REML = FALSE)
```

```
mod11p <- lmer(log(julian_min) ~ state * year_factor * insecticide + (1 | plot),
kbs_sd_plot, REML = FALSE)
```

```
AICctab(mod1p, mod2p, mod3p, mod4p, mod5p, mod6p, mod7p, mod8p, mod9p, mod10p, mod11p,
weights = T) # model 11p and 10p the same
```

```

##           dAICc df weight
## mod7p      0.0 16  0.51
## mod8p      0.1 17  0.49
## mod6p     19.1 11 <0.001
## mod5p     19.2 10 <0.001
## mod9p     19.2 12 <0.001
## mod11p    20.0 30 <0.001
## mod10p    28.1 17 <0.001
## mod1p    99.7  4 <0.001
## mod3p   100.7  5 <0.001
## mod4p   101.5  6 <0.001
## mod2p   107.0  4 <0.001

```

```
anova(mod5p, mod6p) #6p just barely better, going with 5p because it's simpler though
```

```

## Data: kbs_sd_plot
## Models:
## mod5p: log(julian_min) ~ state + year_factor + (1 | plot)
## mod6p: log(julian_min) ~ state + year_factor + insecticide + (1 | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod5p    10 -438.28 -407.28 229.14   -458.28
## mod6p    11 -438.65 -404.55 230.32   -460.65 2.3669  1     0.1239

summary(mod5p)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + year_factor + (1 | plot)
## Data: kbs_sd_plot
##
##      AIC      BIC logLik deviance df.resid
## -438.3   -407.3   229.1   -458.3      154
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.2389 -0.6907  0.0560  0.5003  3.3775
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.0001583 0.01258
## Residual           0.0034400 0.05865
## Number of obs: 164, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)  5.42166  0.01382 138.90560 392.241 < 2e-16 ***
## stateambient -0.03679  0.01051  24.16707 -3.501 0.001827 **  
## year_factor2 -0.11587  0.01733 141.70701 -6.688 4.85e-10 ***
## year_factor3 -0.12672  0.01733 141.70701 -7.314 1.75e-11 ***
## year_factor4 -0.14790  0.01733 141.70701 -8.536 1.94e-14 ***
## year_factor5 -0.09501  0.01733 141.70701 -5.484 1.86e-07 ***
## year_factor6 -0.05977  0.01751 142.50003 -3.413 0.000837 *** 
## year_factor7 -0.17025  0.01751 142.50003 -9.721 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttmbn yr_fc2 yr_fc3 yr_fc4 yr_fc5 yr_fc6
## stateambient -0.380
## year_factr2 -0.655  0.000
## year_factr3 -0.655  0.000  0.522
## year_factr4 -0.655  0.000  0.522  0.522
## year_factr5 -0.655  0.000  0.522  0.522  0.522
## year_factr6 -0.652  0.011  0.517  0.517  0.517  0.517
## year_factr7 -0.652  0.011  0.517  0.517  0.517  0.517  0.512

```

```
anova(mod5p)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq  Mean Sq NumDF   DenDF F value    Pr(>F)
## state       0.04215 0.042152     1  24.167 12.254 0.001827 **
## year_factor 0.44701 0.074502     6 141.026 21.658 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

emmeans(mod5p, list(pairwise ~ state + year_factor), adjust = "tukey")

## $`emmeans of state, year_factor`
##   state  year_factor emmean      SE  df lower.CL upper.CL
##   warmed 1          5.42 0.0142 148    5.39    5.45
##   ambient 1         5.38 0.0142 148    5.36    5.41
##   warmed 2          5.31 0.0137 144    5.28    5.33
##   ambient 2         5.27 0.0137 144    5.24    5.30
##   warmed 3          5.29 0.0137 144    5.27    5.32
##   ambient 3         5.26 0.0137 144    5.23    5.29
##   warmed 4          5.27 0.0137 144    5.25    5.30
##   ambient 4         5.24 0.0137 144    5.21    5.26
##   warmed 5          5.33 0.0137 144    5.30    5.35
##   ambient 5         5.29 0.0137 144    5.26    5.32
##   warmed 6          5.36 0.0139 146    5.33    5.39
##   ambient 6         5.33 0.0140 146    5.30    5.35
##   warmed 7          5.25 0.0139 146    5.22    5.28
##   ambient 7         5.21 0.0140 146    5.19    5.24
##
## 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  0.03679 0.0110 26.1  3.350  0.1018
##   warmed 1 - warmed 2  0.11587 0.0177 147.8  6.539 <.0001
##   warmed 1 - ambient 2  0.15265 0.0208 154.7  7.322 <.0001
##   warmed 1 - warmed 3  0.12672 0.0177 147.8  7.152 <.0001
##   warmed 1 - ambient 3  0.16351 0.0208 154.7  7.843 <.0001
##   warmed 1 - warmed 4  0.14790 0.0177 147.8  8.347 <.0001
##   warmed 1 - ambient 4  0.18468 0.0208 154.7  8.859 <.0001
##   warmed 1 - warmed 5  0.09501 0.0177 147.8  5.362 <.0001
##   warmed 1 - ambient 5  0.13180 0.0208 154.7  6.322 <.0001
##   warmed 1 - warmed 6  0.05977 0.0179 148.7  3.336  0.0619
##   warmed 1 - ambient 6  0.09656 0.0211 155.4  4.573  0.0008
##   warmed 1 - warmed 7  0.17025 0.0179 148.7  9.502 <.0001
##   warmed 1 - ambient 7  0.20703 0.0211 155.4  9.804 <.0001
##   ambient 1 - warmed 2  0.07908 0.0208 154.8  3.794  0.0147
##   ambient 1 - ambient 2  0.11587 0.0177 147.8  6.539 <.0001
##   ambient 1 - warmed 3  0.08994 0.0208 154.8  4.315  0.0022
##   ambient 1 - ambient 3  0.12672 0.0177 147.8  7.152 <.0001
##   ambient 1 - warmed 4  0.11111 0.0208 154.8  5.330 <.0001
##   ambient 1 - ambient 4  0.14790 0.0177 147.8  8.347 <.0001
##   ambient 1 - warmed 5  0.05822 0.0208 154.8  2.793  0.2392
##   ambient 1 - ambient 5  0.09501 0.0177 147.8  5.362 <.0001
##   ambient 1 - warmed 6  0.02299 0.0209 155.1  1.099  0.9980

```

```

## ambient 1 - ambient 6  0.05977 0.0179 148.7  3.336  0.0619
## ambient 1 - warmed 7  0.13346 0.0209 155.1  6.382 <.0001
## ambient 1 - ambient 7  0.17025 0.0179 148.7  9.502 <.0001
## warmed 2 - ambient 2  0.03679 0.0110 26.1   3.350  0.1018
## warmed 2 - warmed 3   0.01086 0.0173 146.3  0.627  1.0000
## warmed 2 - ambient 3  0.04765 0.0205 153.3  2.325  0.5397
## warmed 2 - warmed 4   0.03203 0.0173 146.3  1.851  0.8487
## warmed 2 - ambient 4  0.06882 0.0205 153.3  3.358  0.0580
## warmed 2 - warmed 5   -0.02086 0.0173 146.3 -1.205  0.9950
## warmed 2 - ambient 5  0.01593 0.0205 153.3  0.777  1.0000
## warmed 2 - warmed 6   -0.05609 0.0175 147.1 -3.204  0.0893
## warmed 2 - ambient 6  -0.01931 0.0208 154.2 -0.930  0.9997
## warmed 2 - warmed 7   0.05438 0.0175 147.1  3.106  0.1154
## warmed 2 - ambient 7  0.09117 0.0208 154.2  4.390  0.0017
## ambient 2 - warmed 3  -0.02593 0.0205 153.3 -1.265  0.9921
## ambient 2 - ambient 3  0.01086 0.0173 146.3  0.627  1.0000
## ambient 2 - warmed 4  -0.00476 0.0205 153.3 -0.232  1.0000
## ambient 2 - ambient 4  0.03203 0.0173 146.3  1.851  0.8487
## ambient 2 - warmed 5  -0.05765 0.0205 153.3 -2.812  0.2296
## ambient 2 - ambient 5  -0.02086 0.0173 146.3 -1.205  0.9950
## ambient 2 - warmed 6  -0.09288 0.0206 153.8 -4.517  0.0010
## ambient 2 - ambient 6  -0.05609 0.0175 147.1 -3.204  0.0893
## ambient 2 - warmed 7  0.01759 0.0206 153.8  0.856  0.9999
## ambient 2 - ambient 7  0.05438 0.0175 147.1  3.106  0.1154
## warmed 3 - ambient 3  0.03679 0.0110 26.1   3.350  0.1018
## warmed 3 - warmed 4   0.02117 0.0173 146.3  1.223  0.9942
## warmed 3 - ambient 4  0.05796 0.0205 153.3  2.828  0.2222
## warmed 3 - warmed 5   -0.03172 0.0173 146.3 -1.833  0.8575
## warmed 3 - ambient 5  0.00507 0.0205 153.3  0.247  1.0000
## warmed 3 - warmed 6   -0.06695 0.0175 147.1 -3.825  0.0135
## warmed 3 - ambient 6  -0.03016 0.0208 154.2 -1.452  0.9737
## warmed 3 - warmed 7   0.04352 0.0175 147.1  2.486  0.4249
## warmed 3 - ambient 7  0.08031 0.0208 154.2  3.867  0.0115
## ambient 3 - warmed 4  -0.01562 0.0205 153.3 -0.762  1.0000
## ambient 3 - ambient 4  0.02117 0.0173 146.3  1.223  0.9942
## ambient 3 - warmed 5  -0.06850 0.0205 153.3 -3.342  0.0606
## ambient 3 - ambient 5  -0.03172 0.0173 146.3 -1.833  0.8575
## ambient 3 - warmed 6  -0.10374 0.0206 153.8 -5.045  0.0001
## ambient 3 - ambient 6  -0.06695 0.0175 147.1 -3.825  0.0135
## ambient 3 - warmed 7  0.00673 0.0206 153.8  0.328  1.0000
## ambient 3 - ambient 7  0.04352 0.0175 147.1  2.486  0.4249
## warmed 4 - ambient 4  0.03679 0.0110 26.1   3.350  0.1018
## warmed 4 - warmed 5   -0.05289 0.0173 146.3 -3.056  0.1311
## warmed 4 - ambient 5  -0.01610 0.0205 153.3 -0.786  0.9999
## warmed 4 - warmed 6   -0.08812 0.0175 147.1 -5.034  0.0001
## warmed 4 - ambient 6  -0.05134 0.0208 154.2 -2.472  0.4344
## warmed 4 - warmed 7   0.02235 0.0175 147.1  1.277  0.9914
## warmed 4 - ambient 7  0.05914 0.0208 154.2  2.847  0.2128
## ambient 4 - warmed 5  -0.08968 0.0205 153.3 -4.375  0.0018
## ambient 4 - ambient 5  -0.05289 0.0173 146.3 -3.056  0.1311
## ambient 4 - warmed 6  -0.12491 0.0206 153.8 -6.075 <.0001
## ambient 4 - ambient 6  -0.08812 0.0175 147.1 -5.034  0.0001
## ambient 4 - warmed 7  -0.01444 0.0206 153.8 -0.702  1.0000
## ambient 4 - ambient 7  0.02235 0.0175 147.1  1.277  0.9914

```

```

##  warmed 5 - ambient 5   0.03679 0.0110  26.1  3.350  0.1018
##  warmed 5 - warmed 6  -0.03524 0.0175 147.1 -2.013  0.7572
##  warmed 5 - ambient 6   0.00155 0.0208 154.2  0.075  1.0000
##  warmed 5 - warmed 7   0.07524 0.0175 147.1  4.298  0.0024
##  warmed 5 - ambient 7   0.11202 0.0208 154.2  5.394 <.0001
##  ambient 5 - warmed 6  -0.07202 0.0206 153.8 -3.503  0.0376
##  ambient 5 - ambient 6  -0.03524 0.0175 147.1 -2.013  0.7572
##  ambient 5 - warmed 7   0.03845 0.0206 153.8  1.870  0.8392
##  ambient 5 - ambient 7   0.07524 0.0175 147.1  4.298  0.0024
##  warmed 6 - ambient 6   0.03679 0.0110  26.1  3.350  0.1018
##  warmed 6 - warmed 7   0.11047 0.0177 146.3  6.249 <.0001
##  warmed 6 - ambient 7   0.14726 0.0208 155.5  7.076 <.0001
##  ambient 6 - warmed 7   0.07369 0.0208 155.5  3.541  0.0334
##  ambient 6 - ambient 7   0.11047 0.0177 146.3  6.249 <.0001
##  warmed 7 - ambient 7   0.03679 0.0110  26.1  3.350  0.1018
##
## 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

# including native vs. exotic
kbs_sd_plot_origin <- within(kbs_sd_plot_origin, origin <- relevel(factor(origin),
  ref = "Native")) # releveling so native is the reference
mod12p <- lmer(log(julian_min) ~ state * origin + (1 + year_factor | plot), kbs_sd_plot_origin,
  REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod13p <- lmer(log(julian_min) ~ state + origin + (1 + year_factor | plot), kbs_sd_plot_origin,
  REML = FALSE)

## boundary (singular) fit: see ?isSingular

mod14p <- lmer(log(julian_min) ~ state + origin + year_factor + (1 | plot), kbs_sd_plot_origin,
  REML = FALSE)
anova(mod12p, mod13p) # go with model 13pu

## Data: kbs_sd_plot_origin
## Models:
## mod13p: log(julian_min) ~ state + origin + (1 + year_factor | plot)
## mod12p: log(julian_min) ~ state * origin + (1 + year_factor | plot)
##      npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## mod13p     8 -953.82 -922.89 484.91 -969.82
## mod12p    10 -951.70 -913.03 485.85 -971.70 1.8758  2      0.3914

anova(mod13p, mod14p) # mod 14pu

## Data: kbs_sd_plot_origin
## Models:
## mod14p: log(julian_min) ~ state + origin + year_factor + (1 | plot)
## mod13p: log(julian_min) ~ state + origin + (1 + year_factor | plot)

```

```

##          npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod14p     7 -959.54 -932.47 486.77   -973.54
## mod13p     8 -953.82 -922.89 484.91   -969.82      0   1           1

summary(mod14p)

## 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: kbs_sd_plot_origin
##
##          AIC      BIC      logLik deviance df.resid
## -959.5   -932.5    486.8    -973.5      346
##
## Scaled residuals:
##    Min      1Q Median      3Q      Max
## -3.2407 -0.5652  0.0970  0.6413  2.9848
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 1.392e-05 0.003731
## Residual            3.700e-03 0.060827
## Number of obs: 353, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 5.566519  0.009446 223.277786 589.306 <2e-16 ***
## statewarmed 0.006255  0.006688  23.293127  0.935  0.3592
## originBoth   -0.429290  0.009279 344.471133 -46.264 <2e-16 ***
## originExotic -0.326590  0.007265 333.991632 -44.951 <2e-16 ***
## year_factor  -0.004030  0.001753 338.199437 -2.299  0.0221 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttwrm orgnBt orgnEx
## statewarmed -0.367
## originBoth   -0.403  0.093
## originExotic -0.418  0.021  0.443
## year_factor  -0.722 -0.005  0.038 -0.034

anova(mod14p)

## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state       0.0032  0.0032     1  23.29   0.8748 0.35924
## origin      10.6691  5.3345     2 341.60 1441.7922 < 2e-16 ***
## year_factor  0.0196  0.0196     1 338.20   5.2874 0.02209 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

emmeans(mod14p, list(pairwise ~ state + origin + year_factor), adjust = "tukey")

## $`emmeans of state, origin, year_factor`
##   state    origin year_factor emmean      SE     df lower.CL upper.CL
##   ambient   Native          3.929  5.551 0.006722  88.49    5.537    5.564
##   warmed   Native          3.929  5.557 0.006564  80.52    5.544    5.570
##   ambient   Both            3.929  5.121 0.008137 148.28    5.105    5.137
##   warmed   Both            3.929  5.128 0.008710 174.50    5.110    5.145
##   ambient   Exotic          3.929  5.224 0.006082  62.24    5.212    5.236
##   warmed   Exotic          3.929  5.230 0.006077  62.32    5.218    5.242
##
## 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, year_factor`
##   1                           estimate
##   ambient Native 3.92917847025496 - warmed Native 3.92917847025496 -0.00626
##   ambient Native 3.92917847025496 - ambient Both 3.92917847025496  0.42929
##   ambient Native 3.92917847025496 - warmed Both 3.92917847025496  0.42303
##   ambient Native 3.92917847025496 - ambient Exotic 3.92917847025496  0.32659
##   ambient Native 3.92917847025496 - warmed Exotic 3.92917847025496  0.32033
##   warmed Native 3.92917847025496 - ambient Both 3.92917847025496  0.43555
##   warmed Native 3.92917847025496 - warmed Both 3.92917847025496  0.42929
##   warmed Native 3.92917847025496 - ambient Exotic 3.92917847025496  0.33284
##   warmed Native 3.92917847025496 - warmed Exotic 3.92917847025496  0.32659
##   ambient Both 3.92917847025496 - warmed Both 3.92917847025496 -0.00626
##   ambient Both 3.92917847025496 - ambient Exotic 3.92917847025496 -0.10270
##   ambient Both 3.92917847025496 - warmed Exotic 3.92917847025496 -0.10896
##   warmed Both 3.92917847025496 - ambient Exotic 3.92917847025496 -0.09645
##   warmed Both 3.92917847025496 - warmed Exotic 3.92917847025496 -0.10270
##   ambient Exotic 3.92917847025496 - warmed Exotic 3.92917847025496 -0.00626
##
##   SE     df t.ratio p.value
##   0.00702 26.4 -0.892 0.9452
##   0.00935 348.1 45.894 <.0001
##   0.01218 179.2 34.719 <.0001
##   0.00731 337.4 44.706 <.0001
##   0.01023 115.3 31.319 <.0001
##   0.01118 144.1 38.964 <.0001
##   0.00935 348.1 45.894 <.0001
##   0.01003 106.8 33.195 <.0001
##   0.00731 337.4 44.706 <.0001
##   0.00702 26.4 -0.892 0.9452
##   0.00897 350.6 -11.445 <.0001
##   0.01095 134.6 -9.946 <.0001
##   0.01181 162.5 -8.167 <.0001
##   0.00897 350.6 -11.445 <.0001
##   0.00702 26.4 -0.892 0.9452
##
## 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(mod14p, list(pairwise ~ state), adjust = "tukey")
```

```
## $'emmeans of state'  
##   state    emmean       SE     df lower.CL upper.CL  
##   ambient   5.299 0.004986 26.77    5.288   5.309  
##   warmed    5.305 0.005231 32.34    5.294   5.316  
##  
## Results are averaged over the levels of: origin  
## 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'  
##   1           estimate       SE   df t.ratio p.value  
##   ambient - warmed -0.00626 0.00702 26.4 -0.892  0.3806  
##  
## Results are averaged over the levels of: origin  
## Degrees-of-freedom method: kenward-roger  
## Results are given on the log (not the response) scale.
```

```
emmeans(mod14p, list(pairwise ~ origin), adjust = "tukey")
```

```
## $'emmeans of origin'  
##   origin    emmean       SE     df lower.CL upper.CL  
##   Native    5.554 0.005642 152.8    5.543   5.565  
##   Both      5.125 0.007664 258.8    5.109   5.140  
##   Exotic    5.227 0.004965 107.3    5.217   5.237  
##  
## 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 origin'  
##   1           estimate       SE   df t.ratio p.value  
##   Native - Both     0.429 0.00935 348  45.894 <.0001  
##   Native - Exotic   0.327 0.00731 337  44.706 <.0001  
##   Both - Exotic    -0.103 0.00897 351 -11.445 <.0001  
##  
## 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
```

```
# including growth form - first with interaction term  
kbs_sd_plot_growthhabit <- within(kbs_sd_plot_growthhabit, growth_habit <- relevel(factor(growth_habit)  
  ref = "Forb")) # releveling so forb is the reference  
mod15p <- lmer(log(julian_min) ~ state * growth_habit + (1 + year_factor | plot),  
  kbs_sd_plot_growthhabit, REML = FALSE)
```

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

```
mod16p <- lmer(log(julian_min) ~ state + growth_habit + (1 + year_factor | plot),  
  kbs_sd_plot_growthhabit, REML = FALSE)
```

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

```
mod17p <- lmer(log(julian_min) ~ state + growth_habit + year_factor + (1 | plot),  
  kbs_sd_plot_growthhabit, REML = FALSE)
```

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

```
anova(mod15p, mod16p) # go with model 16p
```

```
## Data: kbs_sd_plot_growthhabit  
## Models:  
## mod16p: log(julian_min) ~ state + growth_habit + (1 + year_factor | plot)  
## mod15p: log(julian_min) ~ state * growth_habit + (1 + year_factor | plot)  
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)  
## mod16p     8 -773.44 -742.42 394.72   -789.44  
## mod15p    10 -771.73 -732.95 395.86   -791.73 2.2811  2      0.3196
```

```
anova(mod16p, mod17p) # mod 17p
```

```
## Data: kbs_sd_plot_growthhabit  
## Models:  
## mod17p: log(julian_min) ~ state + growth_habit + year_factor + (1 | plot)  
## mod16p: log(julian_min) ~ state + growth_habit + (1 + year_factor | plot)  
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)  
## mod17p     7 -775.48 -748.33 394.74   -789.48  
## mod16p     8 -773.44 -742.42 394.72   -789.44      0  1          1
```

```
summary(mod17p)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's  
##   method [lmerModLmerTest]  
## Formula: log(julian_min) ~ state + growth_habit + year_factor + (1 | plot)  
##   Data: kbs_sd_plot_growthhabit  
##  
##      AIC      BIC logLik deviance df.resid  
## -775.5   -748.3   394.7   -789.5      350  
##  
## Scaled residuals:  
##    Min     1Q Median     3Q    Max  
## -3.9517 -0.5667 -0.0714  0.6529  2.3694  
##  
## Random effects:  
##   Groups   Name        Variance Std.Dev.  
##   plot      (Intercept) 0.000000 0.00000  
##   Residual           0.006414 0.08009  
## Number of obs: 357, groups:  plot, 24  
##
```

```

## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)           5.471e+00  1.229e-02 3.570e+02 445.105 < 2e-16 ***
## statewarmed          2.839e-02  8.519e-03 3.570e+02   3.332 0.000951 ***
## growth_habit         -3.562e-01 1.214e-02 3.570e+02 -29.353 < 2e-16 ***
## growth_habitGraminoid -2.762e-01 9.476e-03 3.570e+02 -29.153 < 2e-16 ***
## year_factor          -4.021e-04 2.303e-03 3.570e+02  -0.175 0.861497
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) sttwrm grwth_ grwt_G
## statewarmed -0.355
## growth_habit -0.400  0.089
## grwth_hbtGr -0.415  0.014  0.436
## year_factor  -0.735 -0.003  0.043 -0.028
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

```

```
anova(mod17p)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## state       0.0712  0.0712     1    357 11.1046 0.0009513 ***
## growth_habit 7.6457  3.8229     2    357 596.0300 < 2.2e-16 ***
## year_factor  0.0002  0.0002     1    357   0.0305 0.8614969
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
emmeans(mod17p, list(pairwise ~ state + growth_habit + year_factor), adjust = "tukey")
```

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

```

## $`emmeans of state, growth_habit, year_factor`
##   state   growth_habit year_factor emmean      SE      df lower.CL upper.CL
##   ambient   Forb          3.938  5.470 0.008399  76.86   5.453  5.487
##   warmed   Forb          3.938  5.498 0.008263  71.94   5.482  5.515
##   ambient          Forb          3.938  5.114 0.010456 135.47   5.093  5.134
##   warmed          Forb          3.938  5.142 0.011223 160.99   5.120  5.164
##   ambient Graminoid        3.938  5.194 0.007663  55.71   5.178  5.209
##   warmed Graminoid        3.938  5.222 0.007659  56.87   5.207  5.237
##
## 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, growth_habit, year_factor`
##   1
##   ambient Forb 3.93837535014006 - warmed Forb 3.93837535014006
##   ambient Forb 3.93837535014006 - ambient   3.93837535014006
##   ambient Forb 3.93837535014006 - warmed   3.93837535014006
##   ambient Forb 3.93837535014006 - ambient Graminoid 3.93837535014006

```

```

## ambient Forb 3.93837535014006 - warmed Graminoid 3.93837535014006
## warmed Forb 3.93837535014006 - ambient 3.93837535014006
## warmed Forb 3.93837535014006 - warmed 3.93837535014006
## warmed Forb 3.93837535014006 - ambient Graminoid 3.93837535014006
## warmed Forb 3.93837535014006 - warmed Graminoid 3.93837535014006
## ambient 3.93837535014006 - warmed 3.93837535014006
## ambient 3.93837535014006 - ambient Graminoid 3.93837535014006
## ambient 3.93837535014006 - warmed Graminoid 3.93837535014006
## warmed 3.93837535014006 - ambient Graminoid 3.93837535014006
## warmed 3.93837535014006 - warmed Graminoid 3.93837535014006
## ambient Graminoid 3.93837535014006 - warmed Graminoid 3.93837535014006
## estimate SE df t.ratio p.value
## -0.0284 0.00859 22.5 -3.304 0.0329
## 0.3562 0.01226 356.2 29.064 <.0001
## 0.3279 0.01559 164.4 21.034 <.0001
## 0.2762 0.00955 343.8 28.932 <.0001
## 0.2479 0.01293 104.4 19.168 <.0001
## 0.3846 0.01432 132.6 26.851 <.0001
## 0.3562 0.01226 356.2 29.064 <.0001
## 0.3046 0.01276 97.3 23.875 <.0001
## 0.2762 0.00955 343.8 28.932 <.0001
## -0.0284 0.00859 22.5 -3.304 0.0329
## -0.0800 0.01182 358.1 -6.767 <.0001
## -0.1084 0.01403 124.6 -7.724 <.0001
## -0.0516 0.01517 149.8 -3.401 0.0109
## -0.0800 0.01182 358.1 -6.767 <.0001
## -0.0284 0.00859 22.5 -3.304 0.0329
##
## 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(mod17p, 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        5.484 0.007138 138.4    5.470    5.498
##           5.128 0.009959 243.9    5.108    5.147
## Graminoid   5.208 0.006343 100.0    5.195    5.220
##
## 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 -           0.356 0.01226 356 29.064 <.0001
## Forb - Graminoid 0.276 0.00955 344 28.932 <.0001
## - Graminoid     -0.080 0.01182 358 -6.767 <.0001
##
```

```

## 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

```

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                      0.002117 0.0021175     1  522.35  0.7065  0.40099
## year_factor                 0.086462 0.0216156     4  522.10  7.2121 1.172e-05 ***
## insecticide                  0.000282 0.0002819     1  522.72  0.0941  0.75920
## state:year_factor           0.026659 0.0066647     4  521.99  2.2237  0.06531 .
## year_factor:insecticide     0.010509 0.0026273     4  522.09  0.8766  0.47767
## ---
## 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                      0.002117 0.0021175     1  522.35  0.7065  0.40099
## year_factor                 0.086462 0.0216156     4  522.10  7.2121 1.172e-05 ***
## insecticide                  0.000282 0.0002819     1  522.72  0.0941  0.75920
## state:year_factor           0.026659 0.0066647     4  521.99  2.2237  0.06531 .
## year_factor:insecticide     0.010509 0.0026273     4  522.09  0.8766  0.47767
## ---
## 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/YaRrr/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    17 -1492 -1419.2 762.99     -1526
## mod1u    18 -1490 -1412.9 762.99     -1526      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
## -1490.0 -1412.9  763.0  -1526.0      517
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.0351 -0.4932 -0.0655  0.4447  6.9115
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.000000 0.00000
## species  (Intercept) 0.020042 0.14157
## Residual            0.002997 0.05475
## Number of obs: 535, groups: plot, 24; species, 13
##
## Fixed effects:
##                               Estimate Std. Error       df t value
## (Intercept)                  5.257377  0.040910 14.608317 128.511
## stateambient                 -0.019454  0.011885 521.845430 -1.637
## year_factor3                 -0.026413  0.014079 521.854144 -1.876
## year_factor4                 -0.049132  0.013845 521.925762 -3.549
## year_factor5                 -0.016187  0.013953 521.944380 -1.160
## year_factor6                 -0.019020  0.013772 521.941571 -1.381
## insecticideno_insects       -0.005013  0.012044 522.112659 -0.416
## stateambient:year_factor3    0.016609  0.016110 521.838764  1.031
## stateambient:year_factor4    0.035078  0.015885 522.035413  2.208
## stateambient:year_factor5    0.042685  0.015724 521.967300  2.715
## stateambient:year_factor6    0.023213  0.015595 522.015769  1.488
## year_factor3:insecticideno_insects 0.012964  0.016137 521.859176  0.803
## year_factor4:insecticideno_insects 0.007544  0.015930 522.073920  0.474
## year_factor5:insecticideno_insects 0.008673  0.015819 522.132214  0.548
## year_factor6:insecticideno_insects -0.011677 0.015699 522.226885 -0.744
##                               Pr(>|t|) 
## (Intercept) < 2e-16 ***
## stateambient 0.102267
## year_factor3 0.061214 .
## year_factor4 0.000422 ***

```

```

## year_factor5          0.246527
## year_factor6          0.167866
## insecticideno_insects 0.677402
## stateambient:year_factor3 0.303014
## stateambient:year_factor4 0.027664 *
## stateambient:year_factor5 0.006853 **
## stateambient:year_factor6 0.137245
## year_factor3:insecticideno_insects 0.422133
## year_factor4:insecticideno_insects 0.635988
## year_factor5:insecticideno_insects 0.583751
## year_factor6:insecticideno_insects 0.457340
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 15 > 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
## -1492.0  -1419.2     763.0    -1526.0      518
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -4.0351 -0.4932 -0.0655  0.4447  6.9115
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   species (Intercept) 0.020042 0.14157
##   Residual           0.002997 0.05475
## Number of obs: 535, groups: species, 13
##
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                5.257377  0.040910 14.608310 128.511
## stateambient              -0.019454  0.011885 521.845433 -1.637
## year_factor3              -0.026413  0.014079 521.854146 -1.876
## year_factor4              -0.049132  0.013845 521.925764 -3.549
## year_factor5              -0.016187  0.013953 521.944382 -1.160
## year_factor6              -0.019020  0.013772 521.941574 -1.381
## insecticideno_insects     -0.005013  0.012044 522.112661 -0.416
## stateambient:year_factor3  0.016609  0.016110 521.838767  1.031

```

```

## stateambient:year_factor4      0.035078  0.015885 522.035416  2.208
## stateambient:year_factor5      0.042685  0.015724 521.967303  2.715
## stateambient:year_factor6      0.023213  0.015595 522.015772  1.488
## year_factor3:insecticideno_insects 0.012964  0.016137 521.859178  0.803
## year_factor4:insecticideno_insects 0.007544  0.015930 522.073922  0.474
## year_factor5:insecticideno_insects 0.008673  0.015819 522.132216  0.548
## year_factor6:insecticideno_insects -0.011677 0.015699 522.226887 -0.744
##
##                                     Pr(>|t|)

## (Intercept) < 2e-16 ***
## stateambient          0.102267
## year_factor3          0.061214 .
## year_factor4          0.000422 ***
## year_factor5          0.246527
## year_factor6          0.167866
## insecticideno_insects 0.677402
## stateambient:year_factor3 0.303014
## stateambient:year_factor4 0.027664 *
## stateambient:year_factor5 0.006853 **
## stateambient:year_factor6 0.137245
## year_factor3:insecticideno_insects 0.422133
## year_factor4:insecticideno_insects 0.635988
## year_factor5:insecticideno_insects 0.583751
## year_factor6:insecticideno_insects 0.457340
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

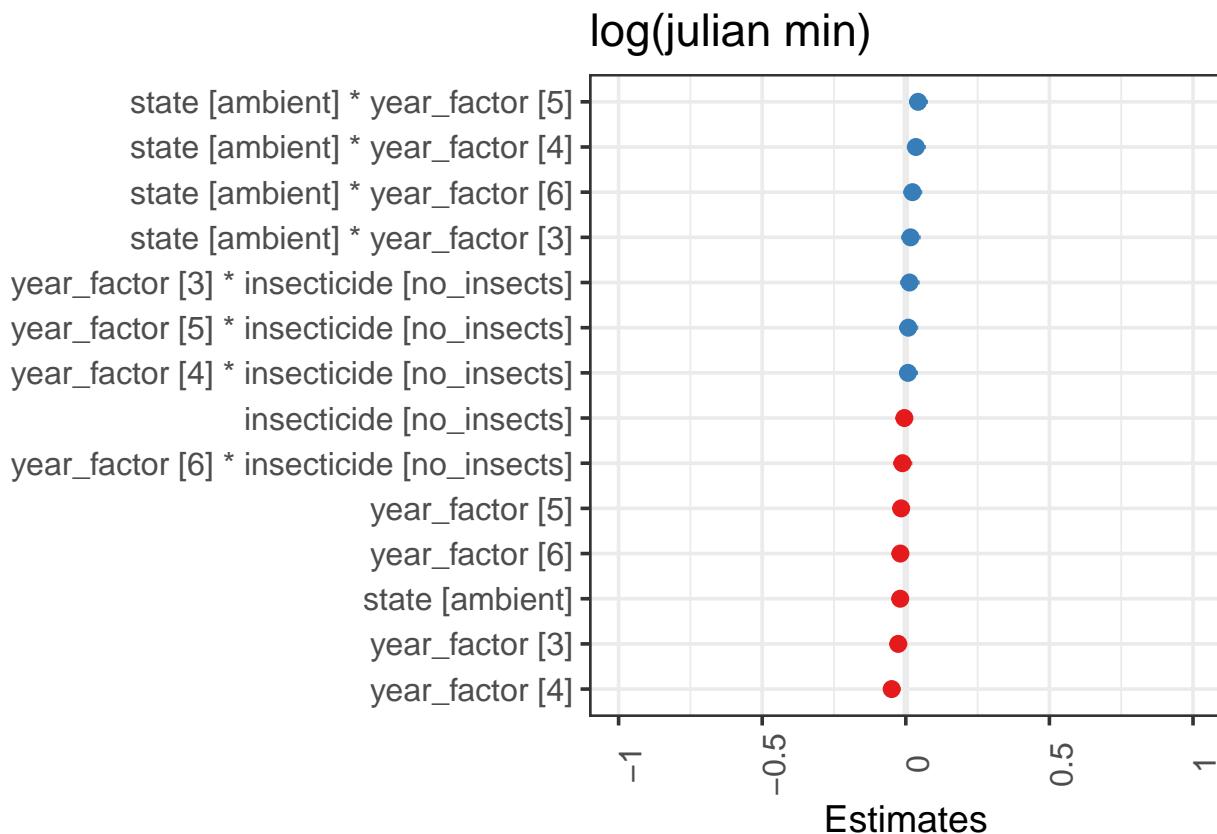
##
## Correlation matrix not shown by default, as p = 15 > 12.
## Use print(x, correlation=TRUE)  or
##      vcov(x)      if you need it

```

```

# 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)

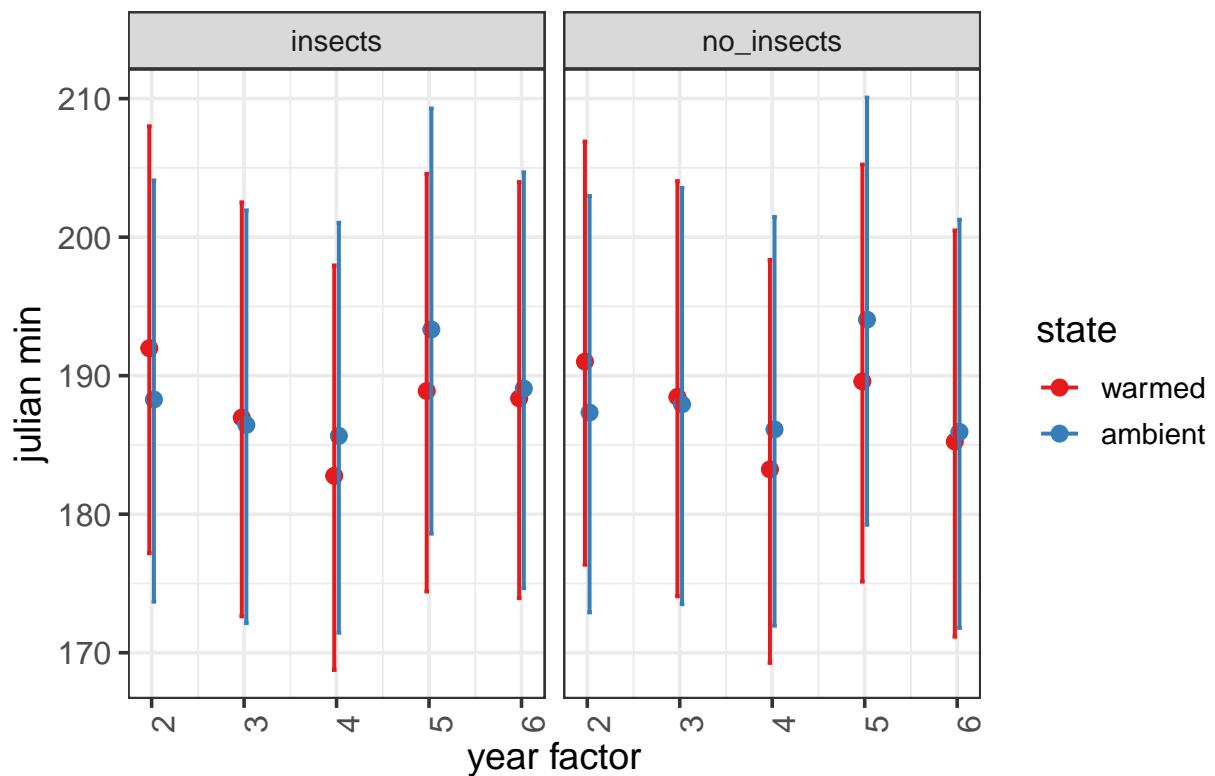
```



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

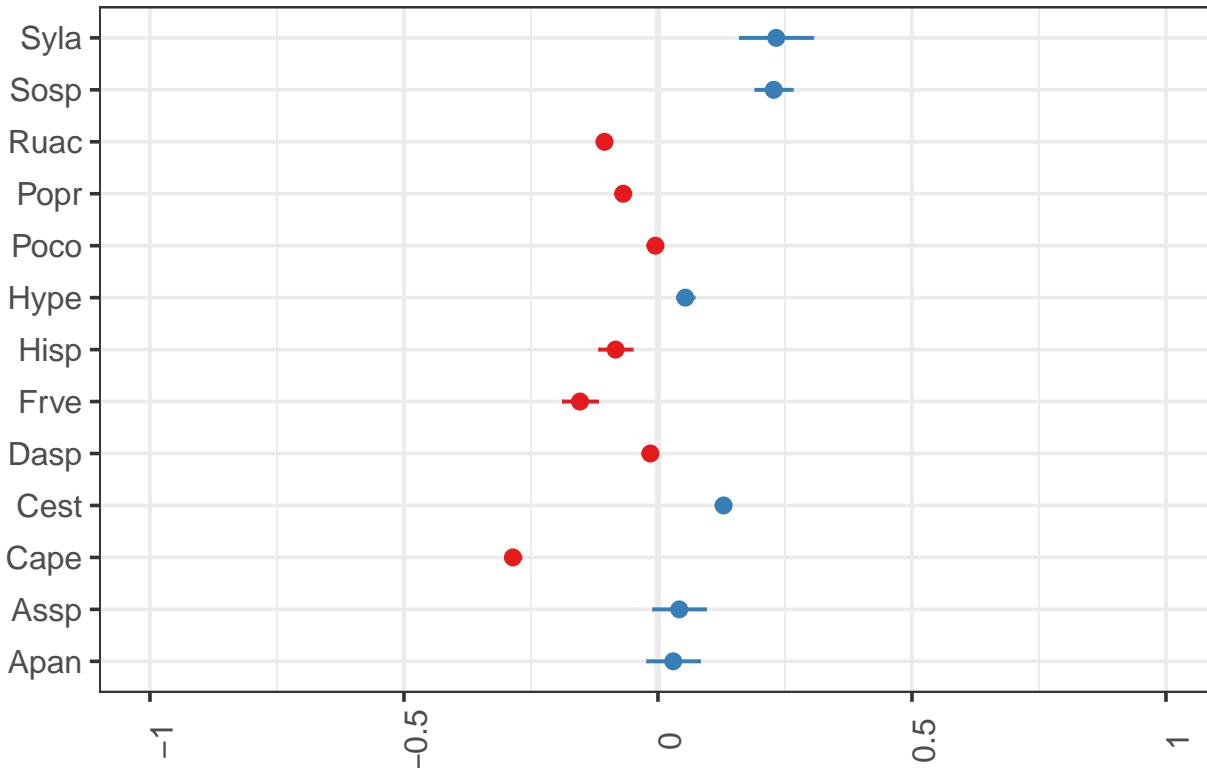
```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standardized
```

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   12 -1498.4 -1447.0 761.19 -1522.4
## mod1u   18 -1490.0 -1412.9 762.99 -1526.0 3.6031  6     0.7302
```

```
# 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
```

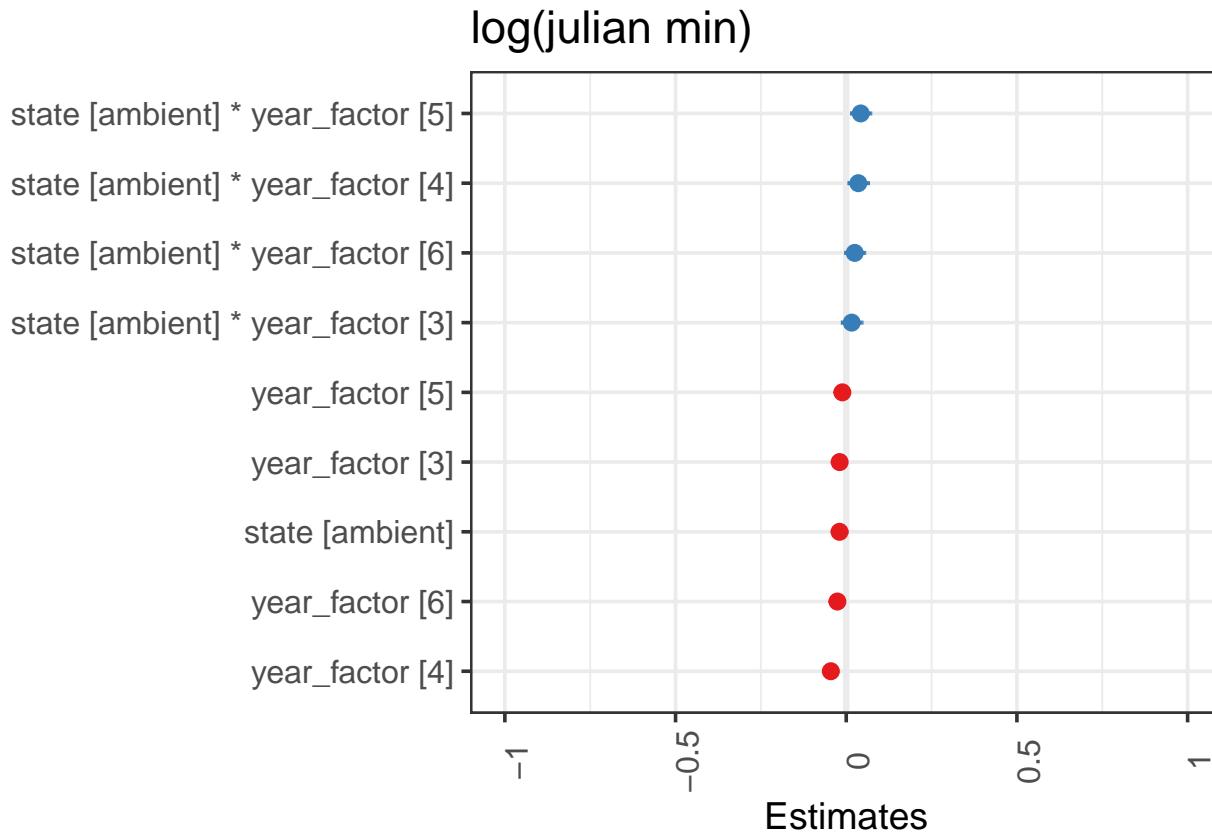
```
anova(mod3u, mod5u) # more complex model is better so go with mod3u
```

```
## 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     9 -1495.6 -1457 756.78  -1513.6
## mod3u    12 -1498.4 -1447 761.19  -1522.4 8.8273  3     0.03168 *
## ---
## 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   12  0.78
## mod5u    2.6    9  0.22
```

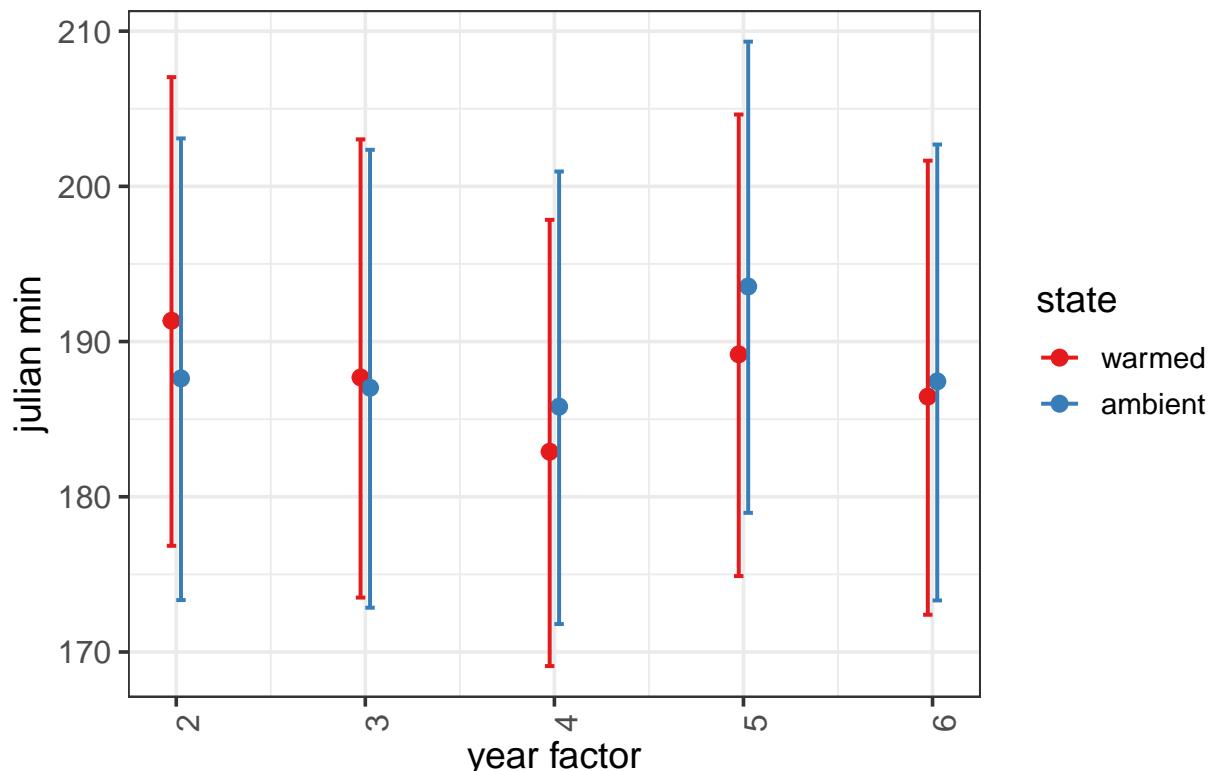
```
# Plot the fixed effects estimates for different models these are the fixed
# effects estimates from summary(mod5)
plot_model(mod3u, sort.est = TRUE)
```



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

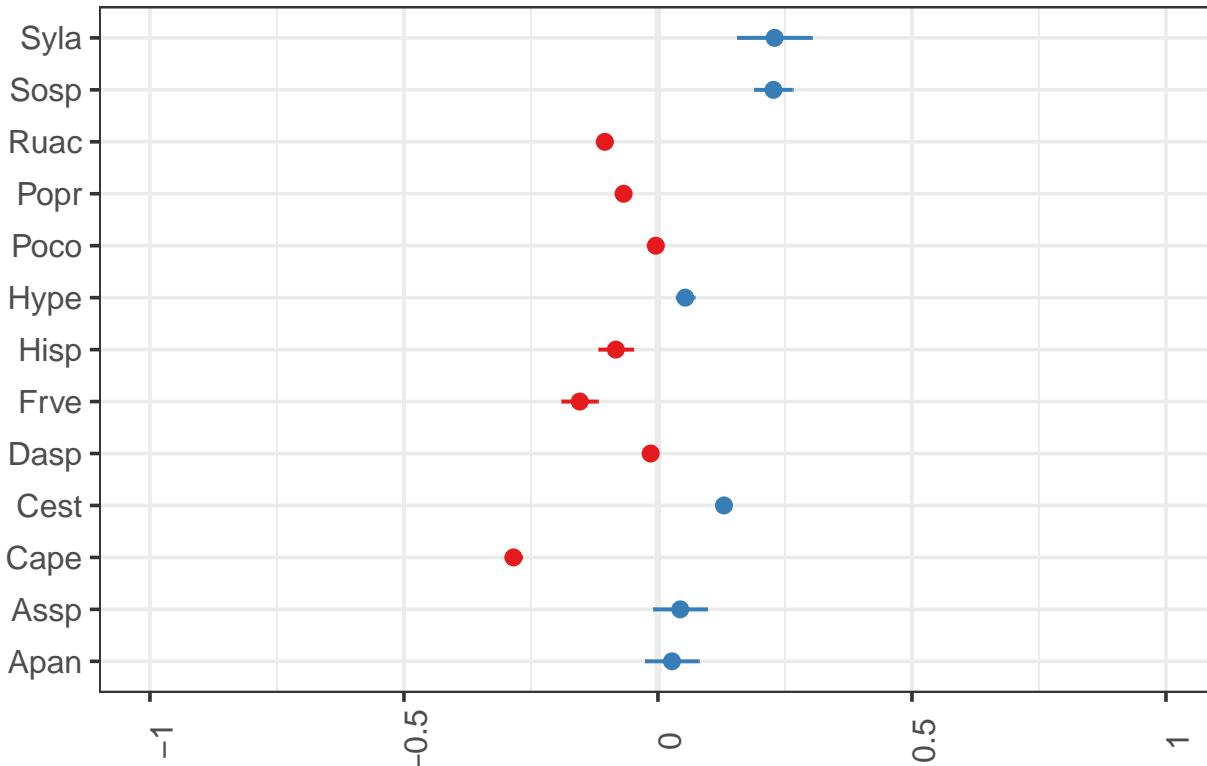
Model has log-transformed response. Back-transforming predictions to original response scale. Standard errors are back-transformed and may not be accurate.

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
```

```
# 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    11 -1490.6 -1443.5 756.30   -1512.6
## mod3u    12 -1498.4 -1447.0 761.19   -1522.4 9.7818  1   0.001763 **
## ---
## 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       0.002204 0.0022043     1 522.37  0.7302  0.39320
```

```

## year_factor      0.086777 0.0216943      4 522.13  7.1871 1.225e-05 ***
## state:year_factor 0.026867 0.0067169      4 522.00  2.2252  0.06515 .
## ---
## 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: -1.2e+00

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 * species + year_factor + (1 | plot), umbs_sd_spp,
                 REML = FALSE)

## fixed-effect model matrix is rank deficient so dropping 2 columns / coefficients
## 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 7u 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   12 -1498.4 -1447.0 761.19 -1522.4
## mod7u   30 -1511.0 -1382.6 785.51 -1571.0 48.642 18  0.0001208 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(mod7u, mod7au) #mod 7au

## Data: umbs_sd_spp
## Models:
## mod7au: log(julian_min) ~ state + species + year_factor + (1 | plot)
## mod7u: log(julian_min) ~ state + species + (1 + year_factor | plot)
##          npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod7au   20 -1550.7 -1465.0 795.33 -1590.7
## mod7u   30 -1511.0 -1382.6 785.51 -1571.0      0 10           1

```

```
anova(mod7au, mod7bu) #mod 7au
```

```
## Data: umbs_sd_spp
## Models:
## mod7au: log(julian_min) ~ state + species + year_factor + (1 | plot)
## mod7bu: log(julian_min) ~ state * species + year_factor + (1 | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod7au    20 -1550.7 -1465.0 795.33   -1590.7
## mod7bu    30 -1539.2 -1410.7 799.59   -1599.2 8.5164 10      0.5785
```

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

```
## Data: umbs_sd_spp
## Models:
## mod7au: log(julian_min) ~ state + species + year_factor + (1 | plot)
## mod7cu: log(julian_min) ~ state + species + year_factor + insecticide +
##          (1 | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod7au    20 -1550.7 -1465 795.33   -1590.7
## mod7cu    21 -1548.9 -1459 795.45   -1590.9 0.2351 1      0.6278
```

```
summary(mod7au)
```

```
## 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: umbs_sd_spp
##
##      AIC      BIC      logLik deviance df.resid
## -1550.7 -1465.0     795.3   -1590.7      515
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -4.0509 -0.5567 -0.0942  0.4244  7.1040
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.000000 0.00000
## Residual           0.002994 0.05472
## Number of obs: 535, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 5.275269  0.028422 535.00000 185.605 < 2e-16 ***
## stateambient 0.005314  0.004791 535.00000   1.109 0.267859
## speciesAssp  0.018990  0.038751 535.00000   0.490 0.624309
## speciesCape -0.317463  0.028741 535.00000  -11.046 < 2e-16 ***
## speciesCest  0.097483  0.027938 535.00000   3.489 0.000524 ***
## speciesDasp -0.047424  0.028165 535.00000  -1.684 0.092798 .
## speciesFrve -0.188684  0.032904 535.00000  -5.734 1.64e-08 ***
## speciesHisp -0.116700  0.032639 535.00000  -3.575 0.000381 ***
## speciesHype  0.021510  0.028978 535.00000   0.742 0.458238
```

```

## speciesPoco -0.036693 0.028360 535.000000 -1.294 0.196288
## speciesPopr -0.099799 0.027963 535.000000 -3.569 0.000391 ***
## speciesRuac -0.136634 0.028511 535.000000 -4.792 2.14e-06 ***
## speciesSosp 0.201367 0.033595 535.000000 5.994 3.77e-09 ***
## speciesSyla 0.217607 0.047440 535.000000 4.587 5.61e-06 ***
## year_factor3 -0.012164 0.008108 535.000000 -1.500 0.134124
## year_factor4 -0.029082 0.008055 535.000000 -3.610 0.000334 ***
## year_factor5 0.008588 0.008016 535.000000 1.071 0.284514
## year_factor6 -0.015002 0.007981 535.000000 -1.880 0.060687 .
## ---
## 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

anova(mod7au) # investigates whether at least one of the levels within each factor is significantly different

## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## state      0.0037 0.00368     1    535   1.2302   0.2679
## species    7.1773 0.59811    12    535 199.7567 < 2.2e-16 ***
## year_factor 0.0906 0.02265     4    535   7.5648 6.218e-06 ***
## ---
## 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 same first seed dates).
emmeans(mod7au, 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           5.246 0.008383 300.1    5.229    5.262
##   ambient 2          5.251 0.008281 265.3    5.235    5.267
##   warmed 3          5.233 0.007952 261.6    5.218    5.249
##   ambient 3          5.239 0.007770 231.4    5.223    5.254
##   warmed 4          5.216 0.007480 217.8    5.202    5.231
##   ambient 4          5.222 0.007264 178.4    5.207    5.236
##   warmed 5          5.254 0.007597 219.7    5.239    5.269
##   ambient 5          5.259 0.007291 180.0    5.245    5.274
##   warmed 6          5.231 0.007324 194.4    5.216    5.245
##   ambient 6          5.236 0.007081 155.0    5.222    5.250
##
## 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 -0.00531 0.00489 21.7 -1.086 0.9811
## warmed 2 - warmed 3  0.01216 0.00825 537.6  1.474 0.9020
## warmed 2 - ambient 3  0.00685 0.00953 240.4  0.718 0.9994
## warmed 2 - warmed 4  0.02908 0.00820 540.5  3.546 0.0154
## warmed 2 - ambient 4  0.02377 0.00947 227.3  2.509 0.2692
## warmed 2 - warmed 5 -0.00859 0.00816 541.9 -1.052 0.9888
## warmed 2 - ambient 5 -0.01390 0.00937 223.5 -1.484 0.8972
## warmed 2 - warmed 6  0.01500 0.00813 542.9  1.846 0.7057
## warmed 2 - ambient 6  0.00969 0.00939 218.9  1.031 0.9900
## ambient 2 - warmed 3  0.01748 0.00966 234.5  1.810 0.7282
## ambient 2 - ambient 3  0.01216 0.00825 537.6  1.474 0.9020
## ambient 2 - warmed 4  0.03440 0.00963 235.9  3.572 0.0153
## ambient 2 - ambient 4  0.02908 0.00820 540.5  3.546 0.0154
## ambient 2 - warmed 5 -0.00327 0.00967 232.3 -0.339 1.0000
## ambient 2 - ambient 5 -0.00859 0.00816 541.9 -1.052 0.9888
## ambient 2 - warmed 6  0.02032 0.00958 231.1  2.120 0.5152
## ambient 2 - ambient 6  0.01500 0.00813 542.9  1.846 0.7057
## warmed 3 - ambient 3 -0.00531 0.00489 21.7 -1.086 0.9811
## warmed 3 - warmed 4  0.01692 0.00768 536.7  2.203 0.4562
## warmed 3 - ambient 4  0.01160 0.00909 199.7  1.277 0.9577
## warmed 3 - warmed 5 -0.02075 0.00764 536.9 -2.718 0.1692
## warmed 3 - ambient 5 -0.02607 0.00898 196.6 -2.904 0.1114
## warmed 3 - warmed 6  0.00284 0.00759 537.9  0.374 1.0000
## warmed 3 - ambient 6 -0.00248 0.00900 192.2 -0.275 1.0000
## ambient 3 - warmed 4  0.02223 0.00913 213.7  2.436 0.3099
## ambient 3 - ambient 4  0.01692 0.00768 536.7  2.203 0.4562
## ambient 3 - warmed 5 -0.01544 0.00916 211.3 -1.685 0.8029
## ambient 3 - ambient 5 -0.02075 0.00764 536.9 -2.718 0.1692
## ambient 3 - warmed 6  0.00815 0.00907 209.6  0.899 0.9964
## ambient 3 - ambient 6  0.00284 0.00759 537.9  0.374 1.0000
## warmed 4 - ambient 4 -0.00531 0.00489 21.7 -1.086 0.9811
## warmed 4 - warmed 5 -0.03767 0.00748 539.6 -5.037 <.0001
## warmed 4 - ambient 5 -0.04298 0.00886 192.7 -4.851 0.0001
## warmed 4 - warmed 6 -0.01408 0.00740 538.2 -1.903 0.6667
## warmed 4 - ambient 6 -0.01940 0.00885 188.3 -2.191 0.4667
## ambient 4 - warmed 5 -0.03236 0.00901 193.9 -3.590 0.0149
## ambient 4 - ambient 5 -0.03767 0.00748 539.6 -5.037 <.0001
## ambient 4 - warmed 6 -0.00877 0.00889 191.9 -0.986 0.9927
## ambient 4 - ambient 6 -0.01408 0.00740 538.2 -1.903 0.6667
## warmed 5 - ambient 5 -0.00531 0.00489 21.7 -1.086 0.9811
## warmed 5 - warmed 6  0.02359 0.00728 535.1  3.242 0.0411
## warmed 5 - ambient 6  0.01828 0.00883 184.8  2.070 0.5511
## ambient 5 - warmed 6  0.02890 0.00871 187.0  3.319 0.0354
## ambient 5 - ambient 6  0.02359 0.00728 535.1  3.242 0.0411
## warmed 6 - ambient 6 -0.00531 0.00489 21.7 -1.086 0.9811
##
## 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 10 estimates

```

```

emmeans(mod7au, list(pairwise ~ year_factor), adjust = "tukey")

## boundary (singular) fit: see ?isSingular

## $`emmeans of year_factor`
##   year_factor emmean      SE  df lower.CL upper.CL
##   2            5.248 0.007965 406.1    5.233    5.264
##   3            5.236 0.007471 370.9    5.221    5.251
##   4            5.219 0.006955 308.9    5.205    5.233
##   5            5.257 0.007032 309.4    5.243    5.271
##   6            5.233 0.006775 272.3    5.220    5.247
##
## 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  0.01216 0.00825 538  1.474  0.5800
##   2 - 4  0.02908 0.00820 540  3.546  0.0039
##   2 - 5 -0.00859 0.00816 542 -1.052  0.8308
##   2 - 6  0.01500 0.00813 543  1.846  0.3485
##   3 - 4  0.01692 0.00768 537  2.203  0.1803
##   3 - 5 -0.02075 0.00764 537 -2.718  0.0527
##   3 - 6  0.00284 0.00759 538  0.374  0.9958
##   4 - 5 -0.03767 0.00748 540 -5.037 <.0001
##   4 - 6 -0.01408 0.00740 538 -1.903  0.3166
##   5 - 6  0.02359 0.00728 535  3.242  0.0110
##
## 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 5 estimates

```

```
emmeans(mod7au, list(pairwise ~ species), adjust = "tukey")
```

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

```
## $`emmeans of species`
##   species emmean      SE  df lower.CL upper.CL
##   Apan     5.268 0.028223 491.0    5.213    5.324
##   Assp     5.287 0.028166 550.5    5.232    5.343
##   Cape     4.951 0.008814 424.4    4.934    4.968
##   Cest     5.366 0.005195 306.0    5.356    5.376
##   Dasp     5.221 0.006122 367.1    5.209    5.233
##   Frve     5.080 0.018871 436.5    5.043    5.117
##   Hisp     5.152 0.017922 465.6    5.116    5.187
##   Hype     5.290 0.009708 454.6    5.271    5.309
##   Poco     5.232 0.007361 422.1    5.217    5.246
##   Popr     5.169 0.005289 313.5    5.158    5.179
##   Ruac     5.132 0.007726 388.7    5.117    5.147
```

```

##   Sosp      5.470 0.020037 350.0      5.430      5.509
##   Syla      5.486 0.039928 532.3      5.408      5.564
##
## 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 -0.01899 0.03976 522  -0.478 1.0000
## Apan - Cape  0.31746 0.02950 509  10.761 <.0001
## Apan - Cest -0.09748 0.02868 508  -3.398 0.0403
## Apan - Dasp  0.04742 0.02892 506   1.640 0.9166
## Apan - Frve  0.18868 0.03386 473   5.573 <.0001
## Apan - Hisp  0.11670 0.03353 514   3.480 0.0310
## Apan - Hype -0.02151 0.02975 513  -0.723 1.0000
## Apan - Poco  0.03669 0.02914 491   1.259 0.9893
## Apan - Popr  0.09980 0.02871 506   3.476 0.0315
## Apan - Ruac  0.13663 0.02926 518   4.670 0.0003
## Apan - Sosp -0.20137 0.03462 445  -5.816 <.0001
## Apan - Syla -0.21761 0.04879 515  -4.460 0.0007
## Assp - Cape  0.33645 0.02948 552  11.414 <.0001
## Assp - Cest -0.07849 0.02863 553  -2.742 0.2335
## Assp - Dasp  0.06641 0.02884 554   2.303 0.5157
## Assp - Frve  0.20767 0.03376 539   6.151 <.0001
## Assp - Hisp  0.13569 0.03349 552   4.051 0.0039
## Assp - Hype -0.00252 0.02972 553  -0.085 1.0000
## Assp - Poco  0.05568 0.02909 551   1.914 0.7868
## Assp - Popr  0.11879 0.02866 553   4.145 0.0027
## Assp - Ruac  0.15562 0.02920 554   5.330 <.0001
## Assp - Sosp -0.18238 0.03452 535  -5.284 <.0001
## Assp - Syla -0.19862 0.04861 552  -4.086 0.0034
## Cape - Cest -0.41495 0.01022 549  -40.605 <.0001
## Cape - Dasp -0.27004 0.01076 552  -25.086 <.0001
## Cape - Frve -0.12878 0.02076 491  -6.204 <.0001
## Cape - Hisp -0.20076 0.02005 510  -10.014 <.0001
## Cape - Hype -0.33897 0.01303 526  -26.018 <.0001
## Cape - Poco -0.28077 0.01134 554  -24.764 <.0001
## Cape - Popr -0.21766 0.01027 547  -21.197 <.0001
## Cape - Ruac -0.18083 0.01178 494  -15.344 <.0001
## Cape - Sosp -0.51883 0.02183 436  -23.769 <.0001
## Cape - Syla -0.53507 0.04088 534  -13.088 <.0001
## Cest - Dasp  0.14491 0.00804 545   18.018 <.0001
## Cest - Frve  0.28617 0.01956 483   14.633 <.0001
## Cest - Hisp  0.21418 0.01867 510   11.471 <.0001
## Cest - Hype  0.07597 0.01100 551   6.909 <.0001
## Cest - Poco  0.13418 0.00898 552  14.935 <.0001
## Cest - Popr  0.19728 0.00741 535  26.620 <.0001
## Cest - Ruac  0.23412 0.00931 553  25.152 <.0001
## Cest - Sosp -0.10388 0.02070 392  -5.019 0.0001
## Cest - Syla -0.12012 0.04026 537  -2.984 0.1315
## Dasp - Frve  0.14126 0.01989 480   7.102 <.0001
## Dasp - Hisp  0.06928 0.01887 526   3.671 0.0163

```

```

##  Dasp - Hype -0.06893 0.01149 553  -5.998 <.0001
##  Dasp - Poco -0.01073 0.00965 554  -1.112 0.9965
##  Dasp - Popr  0.05237 0.00810 548   6.465 <.0001
##  Dasp - Ruac  0.08921 0.00987 554   9.043 <.0001
##  Dasp - Sosp -0.24879 0.02097 396  -11.865 <.0001
##  Dasp - Syla -0.26503 0.04042 541  -6.557 <.0001
##  Frve - Hisp -0.07198 0.02597 546  -2.772 0.2185
##  Frve - Hype -0.21019 0.02114 484  -9.942 <.0001
##  Frve - Poco -0.15199 0.02016 505  -7.538 <.0001
##  Frve - Popr -0.08888 0.01959 486  -4.538 0.0005
##  Frve - Ruac -0.05205 0.02041 453  -2.550 0.3444
##  Frve - Sosp -0.39005 0.02746 446  -14.203 <.0001
##  Frve - Syla -0.40629 0.04391 547  -9.253 <.0001
##  Hisp - Hype -0.13821 0.02047 515  -6.751 <.0001
##  Hisp - Poco -0.08001 0.01947 510  -4.109 0.0032
##  Hisp - Popr -0.01690 0.01869 511  -0.904 0.9995
##  Hisp - Ruac  0.01993 0.01945 529   1.025 0.9984
##  Hisp - Sosp -0.31807 0.02684 480  -11.850 <.0001
##  Hisp - Syla -0.33431 0.04363 554  -7.663 <.0001
##  Hype - Poco  0.05820 0.01212 523   4.804 0.0002
##  Hype - Popr  0.12131 0.01104 551  10.984 <.0001
##  Hype - Ruac  0.15814 0.01245 552  12.706 <.0001
##  Hype - Sosp -0.17986 0.02228 366  -8.074 <.0001
##  Hype - Syla -0.19610 0.04108 530  -4.773 0.0002
##  Poco - Popr  0.06311 0.00904 552  6.982 <.0001
##  Poco - Ruac  0.09994 0.01071 527  9.333 <.0001
##  Poco - Sosp -0.23806 0.02129 417  -11.184 <.0001
##  Poco - Syla -0.25430 0.04059 538  -6.265 <.0001
##  Popr - Ruac  0.03683 0.00936 553  3.935 0.0062
##  Popr - Sosp -0.30117 0.02071 400  -14.540 <.0001
##  Popr - Syla -0.31741 0.04027 538  -7.881 <.0001
##  Ruac - Sosp -0.33800 0.02148 394  -15.732 <.0001
##  Ruac - Syla -0.35424 0.04061 543  -8.724 <.0001
##  Sosp - Syla -0.01624 0.04443 553  -0.366 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

```

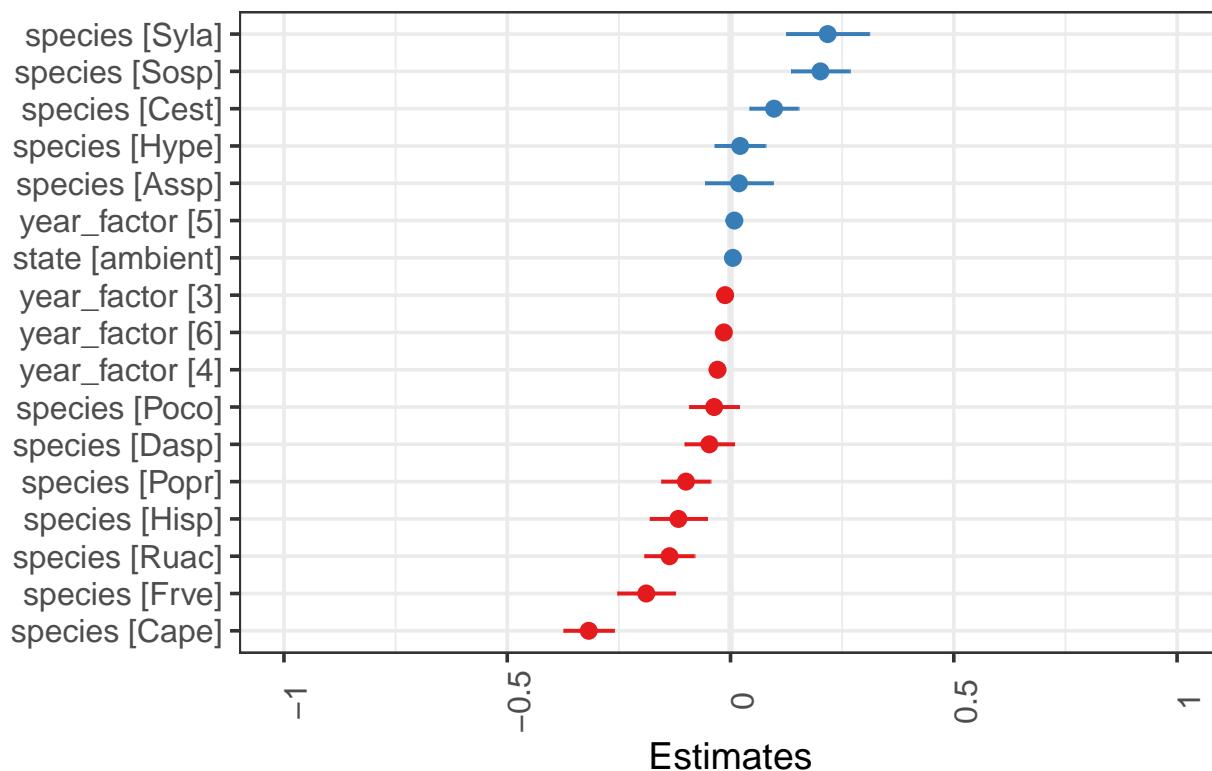
```

# Take a look at the estimates for each fixed effect. These are the estimates
# from summary(mod7au). 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(mod7au, sort.est = TRUE)
```

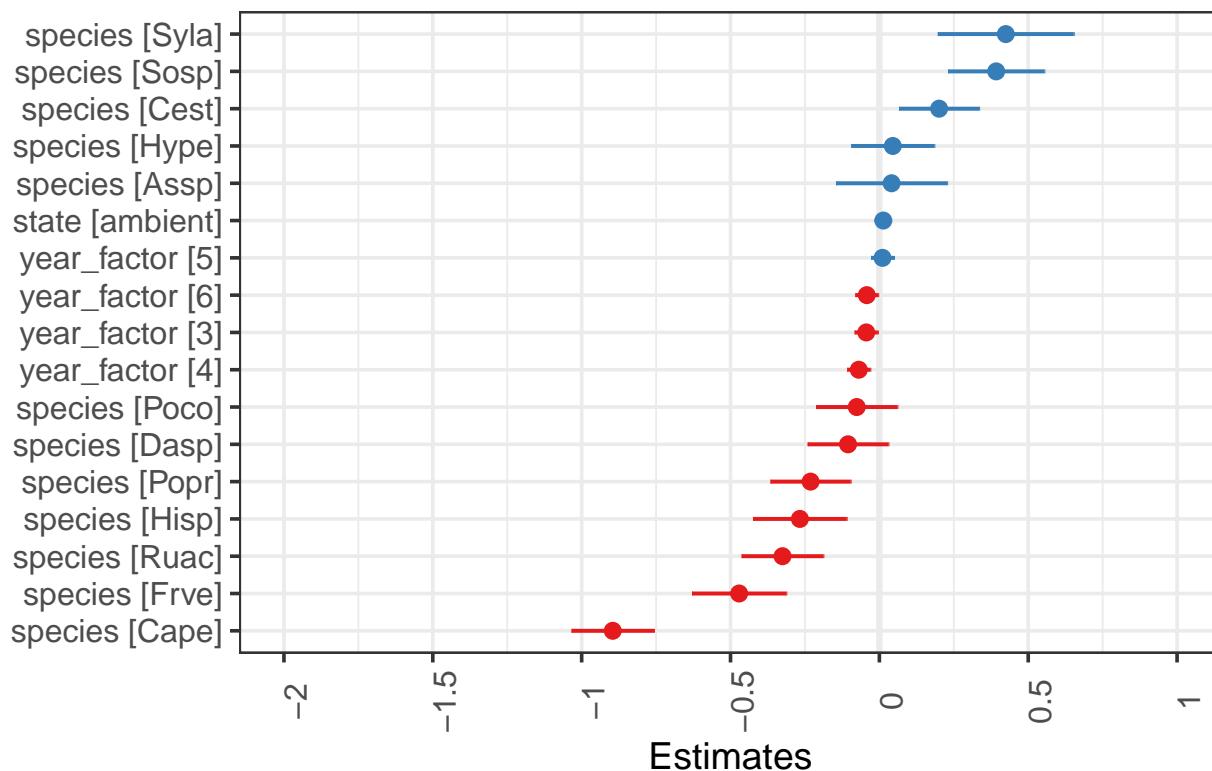
log(julian min)



```
# if you want to standardize the estimates:  
plot_model(mod7au, 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)

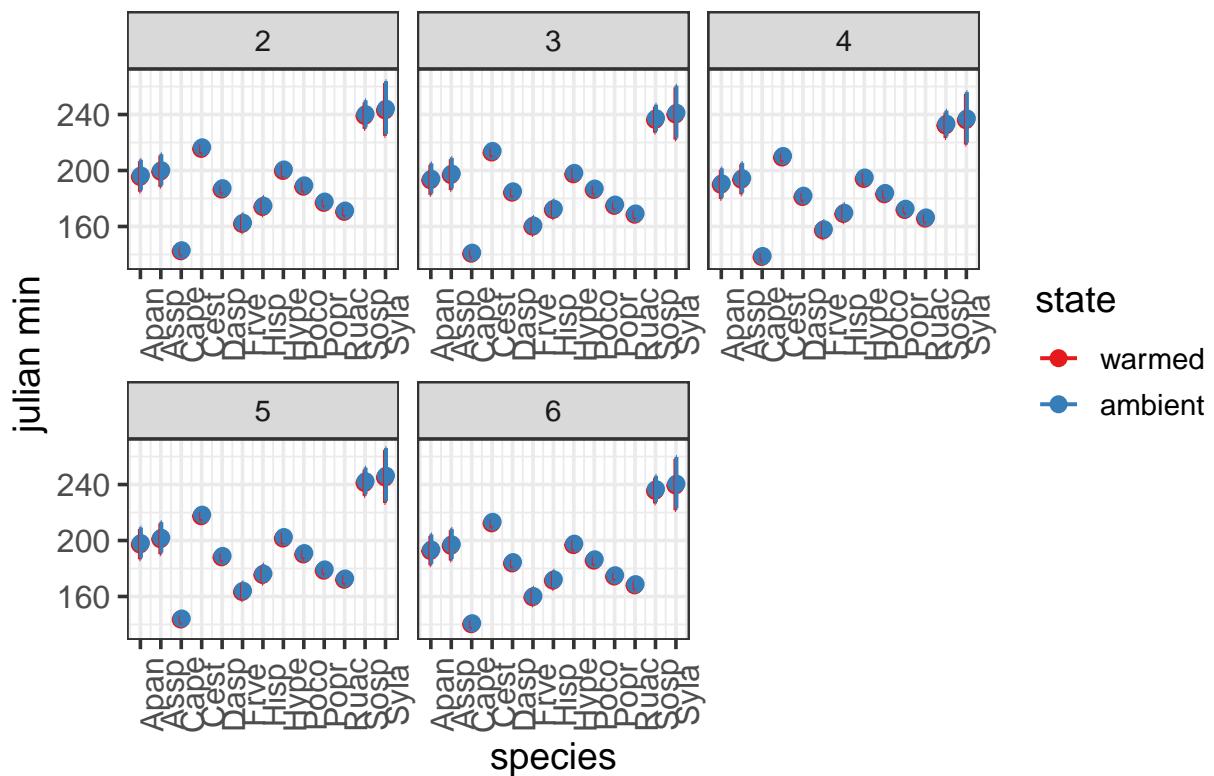


these are the fixed predicted values: - note this is a new plot

```
plot_model(mod7au, type = "pred", terms = c("species", "state", "year_factor"))
```

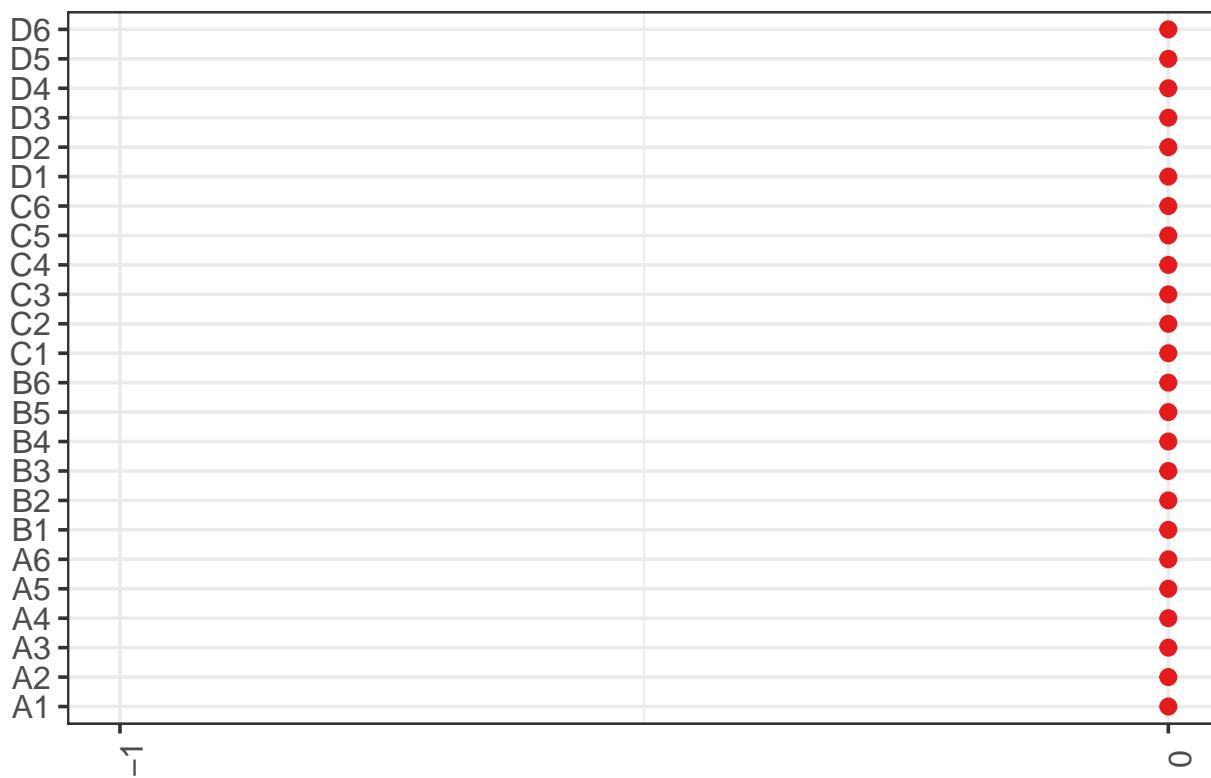
Model has log-transformed response. Back-transforming predictions to original response scale. Standardized

Predicted values of julian min



```
# these are the random effects estimates  
plot_model(mod7au, type = "re")
```

Random effects



```

# including native vs. exotic

umbs_sd_spp <- within(kbs_sd_spp, origin <- relevel(factor(origin), ref = "Native")) # releveling so no
mod8u <- lmer(log(julian_min) ~ state * origin + (1 + year_factor | plot), umbs_sd_spp,
               REML = FALSE)

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## Data: umbs_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 -1153.9 -998.79 609.95   -1219.9
## mod8u    35 -1150.8 -986.32 610.42   -1220.8 0.9329  2     0.6272

```

```

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 -1237.3 -1180.92 630.67   -1261.3
## mod9u     33 -1153.9  -998.79 609.95   -1219.9      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
## -1237.3 -1180.9     630.7   -1261.3      801
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -4.1271 -0.6128  0.0252  0.6213  3.4898
##
## 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+02 304.925 < 2e-16 ***
## stateambient 7.524e-05 7.922e-03 8.130e+02  0.009 0.992425
## originBoth   -4.029e-01 1.498e-02 8.130e+02 -26.902 < 2e-16 ***
## originExotic -3.207e-01 9.783e-03 8.130e+02 -32.784 < 2e-16 ***
## year_factor2 -1.036e-01 1.837e-02 8.130e+02 -5.639 2.36e-08 ***
## year_factor3 -1.281e-01 1.883e-02 8.130e+02 -6.802 2.00e-11 ***
## year_factor4 -1.178e-01 1.940e-02 8.130e+02 -6.069 1.97e-09 ***
## year_factor5 -7.815e-02 1.994e-02 8.130e+02 -3.919 9.65e-05 ***
## year_factor6 -6.982e-02 1.907e-02 8.130e+02 -3.662 0.000266 ***
## year_factor7 -1.138e-01 2.149e-02 8.130e+02 -5.296 1.53e-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
## stateambint -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  0.0001  0.9924
## origin     15.2502 7.6251     2   813 614.4807 < 2.2e-16 ***
## year_factor 0.7874 0.1312     6   813 10.5753 2.511e-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 0.009665 172.72    5.530   5.568
##   ambient Native  5.549 0.009795 163.67    5.529   5.568
##   warmed  Both    5.146 0.013851 433.30    5.119   5.173
##   ambient Both    5.146 0.013143 336.34    5.120   5.172
##   warmed  Exotic  5.228 0.006726  46.92    5.214   5.241
##   ambient Exotic  5.228 0.006169  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 -7.52e-05 0.00800 22.9 -0.009  1.0000
##   warmed Native - warmed Both    4.03e-01 0.01510 821.8 26.686 <.0001
##   warmed Native - ambient Both   4.03e-01 0.01644 273.4 24.501 <.0001
##   warmed Native - warmed Exotic 3.21e-01 0.00986 822.6 32.516 <.0001
##   warmed Native - ambient Exotic 3.21e-01 0.01231 116.9 26.046 <.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 26.686 <.0001
##   ambient Native - warmed Exotic 3.21e-01 0.01308 140.0 24.531 <.0001
##   ambient Native - ambient Exotic 3.21e-01 0.00986 822.6 32.516 <.0001
##   warmed Both - ambient Both    -7.52e-05 0.00800 22.9 -0.009  1.0000
##   warmed Both - warmed Exotic   -8.21e-02 0.01340 823.1 -6.132 <.0001
##   warmed Both - ambient Exotic   -8.22e-02 0.01598 274.6 -5.145 <.0001
##   ambient Both - warmed Exotic   -8.21e-02 0.01522 209.6 -5.394 <.0001
##   ambient Both - ambient Exotic   -8.21e-02 0.01340 823.1 -6.132 <.0001
##   warmed Exotic - ambient Exotic -7.52e-05 0.00800 22.9 -0.009  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

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

## boundary (singular) fit: see ?isSingular

## $`emmeans of origin`
##   origin emmean      SE    df lower.CL upper.CL
##   Native  5.549 0.008870 345.58    5.531    5.566
##   Both    5.146 0.012896 603.76    5.120    5.171
##   Exotic  5.228 0.005064  56.82    5.218    5.238
##
## 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 origin`
##   1           estimate      SE   df t.ratio p.value
##   Native - Both     0.4029 0.01510 822 26.686 <.0001
##   Native - Exotic   0.3207 0.00986 823 32.516 <.0001
##   Both - Exotic    -0.0821 0.01340 823 -6.132 <.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 3 estimates

# including growth form - first with interaction term
umbssd_spp <- within(umbssd_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),
  umbssd_spp, REML = FALSE)

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 1 negative eigenvalue: -6.6e+00

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

## boundary (singular) fit: see ?isSingular

```

```

anova(mod10u, mod11u) # model 11u

## 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 -936.43 -781.30 501.21   -1002.4
## mod10u    35 -932.94 -768.42 501.47   -1002.9 0.5143  2     0.7733

```

```

anova(mod11u, mod11au) # model 11au

```

```

## 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 -1031.26 -974.85 527.63   -1055.3
## mod11u    33 -936.43 -781.30 501.21   -1002.4          0 21           1

```

```

summary(mod11au)

```

```

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + growth_habit + year_factor + (1 | plot)
##   Data: umbs_sd_spp
##
##      AIC      BIC      logLik deviance df.resid
## -1031.3   -974.9    527.6   -1055.3       801
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -5.3655 -0.6249 -0.0294  0.7470  3.2410
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   plot     (Intercept) 0.00000  0.0000
##   Residual            0.01599  0.1264
## Number of obs: 813, groups:  plot, 24
##
## Fixed effects:
##                   Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)      5.591809  0.020737 813.000000 269.653 < 2e-16 ***
## stateambient    -0.018564  0.008977 813.000000 -2.068  0.039 *
## growth_habit    -0.298653  0.015750 813.000000 -18.962 < 2e-16 ***
## growth_habitGraminoid -0.251913  0.009865 813.000000 -25.536 < 2e-16 ***
## year_factor2    -0.164930  0.021066 813.000000 -7.829 1.53e-14 ***
## year_factor3    -0.185967  0.021555 813.000000 -8.628 < 2e-16 ***
## year_factor4    -0.154719  0.022043 813.000000 -7.019 4.71e-12 ***
## year_factor5    -0.107017  0.022658 813.000000 -4.723 2.73e-06 ***
## year_factor6    -0.093088  0.021709 813.000000 -4.288 2.02e-05 ***
## year_factor7    -0.130013  0.024385 813.000000 -5.332 1.26e-07 ***

```

```

## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttmbn grwth_G yr_fc2 yr_fc3 yr_fc4 yr_fc5 yr_fc6
## stateambient -0.230
## growth_habit -0.112 -0.063
## grwth_hbtGr -0.375  0.012  0.340
## year_factr2 -0.847 -0.012 -0.065  0.146
## year_factr3 -0.826 -0.001 -0.032  0.131  0.795
## year_factr4 -0.772 -0.018 -0.016  0.043  0.759  0.740
## year_factr5 -0.749 -0.032 -0.027  0.044  0.741  0.721  0.700
## year_factr6 -0.803 -0.009 -0.035  0.089  0.782  0.760  0.733  0.714
## year_factr7 -0.689  0.019 -0.039 -0.008  0.680  0.663  0.647  0.630  0.658
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

```

```
anova(mod11au)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state      0.0684  0.0684     1    813  4.2768 0.03895 *
## growth_habit 12.3397  6.1699     2    813 385.8837 < 2e-16 ***
## year_factor   1.7126  0.2854     6    813 17.8519 < 2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
emmeans(mod11au, 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         5.463 0.008231 186.96    5.447    5.479
##             5.164 0.014674 599.35    5.136    5.193
##   Graminoid    5.211 0.006249  79.39    5.199    5.224
##
## 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 growth_habit`
##   1           estimate       SE   df t.ratio p.value
##   Forb -            0.2987 0.01588 821 18.812 <.0001
##   Forb - Graminoid  0.2519 0.00995 823 25.315 <.0001
##   - Graminoid     -0.0467 0.01563 822 -2.990  0.0081
##
## 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 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

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 1 negative eigenvalue: -8.6e-02

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

## Warning: Model failed to converge with 1 negative eigenvalue: -6.6e+00

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
## -2580.7 -2434.8   1321.3   -2642.7     785
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.5905 -0.5311 -0.0057  0.4910  7.6698
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.000000 0.00000
## Residual            0.002296 0.04792
## Number of obs: 816, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept) 5.452873  0.011998 816.000000 454.472 < 2e-16 ***
## stateambient 0.004415  0.003510 816.000000  1.258 0.208799  
## speciesArel -0.184402  0.010473 816.000000 -17.608 < 2e-16 ***
## speciesBavu -0.230624  0.020319 816.000000 -11.350 < 2e-16 ***
## speciesCahi -0.502768  0.023483 816.000000 -21.409 < 2e-16 ***
## speciesCest  0.039203  0.011627 816.000000  3.372 0.000782 ***
## speciesDaca  0.101529  0.014505 816.000000  6.999 5.37e-12 ***
## speciesDagl -0.163609  0.010838 816.000000 -15.095 < 2e-16 ***
## speciesElre -0.039505  0.011380 816.000000 -3.471 0.000545 ***
## speciesEugr  0.214376  0.013910 816.000000 15.411 < 2e-16 ***
## speciesHisp -0.246595  0.011135 816.000000 -22.146 < 2e-16 ***
## speciesHype -0.109631  0.017735 816.000000 -6.182 1.00e-09 ***
## speciesPhpr -0.054816  0.010555 816.000000 -5.193 2.61e-07 ***
## speciesPoco -0.081816  0.017619 816.000000 -4.644 3.99e-06 ***
## speciesPopr -0.248585  0.010210 816.000000 -24.348 < 2e-16 ***
## speciesPore -0.046129  0.013021 816.000000 -3.543 0.000419 ***
## speciesRusp -0.172149  0.017193 816.000000 -10.013 < 2e-16 ***
## speciesSoca  0.230145  0.010176 816.000000 22.616 < 2e-16 ***

```

```

## speciesSogr      0.123898  0.048999 816.000000  2.529 0.011640 *
## speciesSora     0.173310  0.035193 816.000000  4.925 1.02e-06 ***
## speciesSypi     0.289844  0.035408 816.000000  8.186 1.04e-15 ***
## speciesTaof     -0.451720 0.020330 816.000000 -22.220 < 2e-16 ***
## speciesTrsp     -0.117093 0.014959 816.000000 -7.828 1.54e-14 ***
## factor(year_factor)2 -0.123863 0.008148 816.000000 -15.201 < 2e-16 ***
## factor(year_factor)3 -0.123886 0.008316 816.000000 -14.897 < 2e-16 ***
## factor(year_factor)4 -0.134448 0.008456 816.000000 -15.900 < 2e-16 ***
## factor(year_factor)5 -0.096464 0.008657 816.000000 -11.143 < 2e-16 ***
## factor(year_factor)6 -0.086031 0.008335 816.000000 -10.321 < 2e-16 ***
## factor(year_factor)7 -0.094665 0.009456 816.000000 -10.012 < 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 0.00363     1    816  1.5822 0.2088
## species                          23.6359 1.12552    21    816 490.1339 <2e-16 ***
## factor(year_factor)  0.7372 0.12286     6    816  53.5021 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

UMBS Plot-level Mixed Effects Models

```

mod1pu <- lmer(julian_min ~ state + (1 | plot), umbs_sd_plot, REML = FALSE)
mod2pu <- lmer(julian_min ~ insecticide + (1 | plot), umbs_sd_plot, REML = FALSE)
mod3pu <- lmer(julian_min ~ insecticide + state + (1 | plot), umbs_sd_plot, REML = FALSE)
mod4pu <- lmer(julian_min ~ insecticide * state + (1 | plot), umbs_sd_plot, REML = FALSE)
mod5pu <- lmer(julian_min ~ state + year_factor + (1 | plot), umbs_sd_plot, REML = FALSE)
mod6pu <- lmer(julian_min ~ state + year_factor + insecticide + (1 | plot), umbs_sd_plot,
REML = FALSE)
mod7pu <- lmer(julian_min ~ state * year_factor + (1 | plot), umbs_sd_plot, REML = FALSE)
mod8pu <- lmer(julian_min ~ state * year_factor + insecticide + (1 | plot), umbs_sd_plot,
REML = FALSE)
mod9pu <- lmer(julian_min ~ state * insecticide + year_factor + (1 | plot), umbs_sd_plot,
REML = FALSE)
mod10pu <- lmer(julian_min ~ state + insecticide * year_factor + (1 | plot), umbs_sd_plot,
REML = FALSE)
mod11pu <- lmer(julian_min ~ state * year_factor * insecticide + (1 | plot), umbs_sd_plot,
REML = FALSE)
AICcTab(mod1pu, mod2pu, mod3pu, mod4pu, mod5pu, mod6pu, mod7pu, mod8pu, mod9pu, mod10pu,
mod11pu, weights = T) # model 11p and 10p the same

```

```

##          dAICc df weight
## mod5pu    0.0   8  0.6144
## mod6pu    2.3   9  0.1908
## mod7pu    3.6  12  0.1026
## mod9pu    4.7  10  0.0600
## mod8pu    6.1  13  0.0291
## mod10pu   10.9 13  0.0027
## mod11pu   15.1 22 <0.001
## mod1pu    19.9  4 <0.001
## mod2pu    20.2  4 <0.001
## mod3pu    22.1  5 <0.001
## mod4pu    24.2  6 <0.001

anova(mod5pu, mod6pu) #5pu just barely better, going with 5pu because it's simpler though

## Data: umbs_sd_plot
## Models:
## mod5pu: julian_min ~ state + year_factor + (1 | plot)
## mod6pu: julian_min ~ state + year_factor + insecticide + (1 | plot)
##      npar   AIC   BIC logLik deviance Chisq Df Pr(>Chisq)
## mod5pu     8 787.59 809.89 -385.8    771.59
## mod6pu     9 789.59 814.68 -385.8    771.59 1e-04  1      0.9909

summary(mod5pu)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: julian_min ~ state + year_factor + (1 | plot)
## Data: umbs_sd_plot
##
##      AIC      BIC   logLik deviance df.resid
##    787.6    809.9   -385.8    771.6      112
##
## Scaled residuals:
##       Min     1Q   Median     3Q    Max
## -2.39707 -0.61543  0.03155  0.61564  2.35330
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   plot     (Intercept) 6.536    2.557
##   Residual            31.490   5.612
## Number of obs: 120, groups:  plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept) 187.3833   1.4557 75.9400 128.719 < 2e-16 ***
## stateambient  0.8167   1.4625 24.0000  0.558 0.581754  
## year_factor3 -2.8750   1.6199 96.0000 -1.775 0.079108 .
## year_factor4 -6.2083   1.6199 96.0000 -3.832 0.000227 ***
## year_factor5  2.6667   1.6199 96.0000  1.646 0.103003  
## year_factor6 -2.4167   1.6199 96.0000 -1.492 0.139024  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

## 
## Correlation of Fixed Effects:
##           (Intr) sttmbn yr_fc3 yr_fc4 yr_fc5
## stateambint -0.502
## year_factr3 -0.556  0.000
## year_factr4 -0.556  0.000  0.500
## year_factr5 -0.556  0.000  0.500  0.500
## year_factr6 -0.556  0.000  0.500  0.500  0.500

anova(mod5pu)

## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## state        9.82   9.819     1     24  0.3118   0.5818
## year_factor 1059.72 264.929     4     96 8.4130 7.322e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

emmeans(mod5pu, list(pairwise ~ state + year_factor), adjust = "tukey")

## $`emmeans of state, year_factor`
##   state  year_factor emmean   SE  df lower.CL upper.CL
##   warmed 2          187 1.5 82.1    184    190
##   ambient 2          188 1.5 82.1    185    191
##   warmed 3          185 1.5 82.1    182    188
##   ambient 3          185 1.5 82.1    182    188
##   warmed 4          181 1.5 82.1    178    184
##   ambient 4          182 1.5 82.1    179    185
##   warmed 5          190 1.5 82.1    187    193
##   ambient 5          191 1.5 82.1    188    194
##   warmed 6          185 1.5 82.1    182    188
##   ambient 6          186 1.5 82.1    183    189
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $`pairwise differences of state, year_factor`
##   1           estimate   SE  df t.ratio p.value
##   warmed 2 - ambient 2 -0.817 1.53 26.2 -0.535  0.9999
##   warmed 2 - warmed 3  2.875 1.65 100.2  1.737  0.7715
##   warmed 2 - ambient 3  2.058 2.25  93.2  0.914  0.9957
##   warmed 2 - warmed 4  6.208 1.65 100.2  3.752  0.0105
##   warmed 2 - ambient 4  5.392 2.25  93.2  2.394  0.3414
##   warmed 2 - warmed 5 -2.667 1.65 100.2 -1.611  0.8395
##   warmed 2 - ambient 5 -3.483 2.25  93.2 -1.547  0.8694
##   warmed 2 - warmed 6  2.417 1.65 100.2  1.460  0.9042
##   warmed 2 - ambient 6  1.600 2.25  93.2  0.710  0.9994
##   ambient 2 - warmed 3  3.692 2.25  93.2  1.639  0.8253
##   ambient 2 - ambient 3  2.875 1.65 100.2  1.737  0.7715
##   ambient 2 - warmed 4  7.025 2.25  93.2  3.119  0.0695
##   ambient 2 - ambient 4  6.208 1.65 100.2  3.752  0.0105
##   ambient 2 - warmed 5 -1.850 2.25  93.2 -0.821  0.9981
##   ambient 2 - ambient 5 -2.667 1.65 100.2 -1.611  0.8395

```

```

## ambient 2 - warmed 6      3.233 2.25  93.2   1.436  0.9126
## ambient 2 - ambient 6    2.417 1.65 100.2   1.460  0.9042
## warmed 3 - ambient 3    -0.817 1.53  26.2  -0.535  0.9999
## warmed 3 - warmed 4     3.333 1.65 100.2   2.014  0.5912
## warmed 3 - ambient 4    2.517 2.25  93.2   1.117  0.9817
## warmed 3 - warmed 5     -5.542 1.65 100.2  -3.349  0.0363
## warmed 3 - ambient 5    -6.358 2.25  93.2  -2.823  0.1439
## warmed 3 - warmed 6     -0.458 1.65 100.2  -0.277  1.0000
## warmed 3 - ambient 6    -1.275 2.25  93.2  -0.566  0.9999
## ambient 3 - warmed 4    4.150 2.25  93.2   1.843  0.7065
## ambient 3 - ambient 4    3.333 1.65 100.2   2.014  0.5912
## ambient 3 - warmed 5    -4.725 2.25  93.2  -2.098  0.5336
## ambient 3 - ambient 5    -5.542 1.65 100.2  -3.349  0.0363
## ambient 3 - warmed 6     0.358 2.25  93.2   0.159  1.0000
## ambient 3 - ambient 6    -0.458 1.65 100.2  -0.277  1.0000
## warmed 4 - ambient 4    -0.817 1.53  26.2  -0.535  0.9999
## warmed 4 - warmed 5     -8.875 1.65 100.2  -5.363  <.0001
## warmed 4 - ambient 5    -9.692 2.25  93.2  -4.303  0.0016
## warmed 4 - warmed 6     -3.792 1.65 100.2  -2.291  0.4038
## warmed 4 - ambient 6    -4.608 2.25  93.2  -2.046  0.5693
## ambient 4 - warmed 5    -8.058 2.25  93.2  -3.578  0.0187
## ambient 4 - ambient 5    -8.875 1.65 100.2  -5.363  <.0001
## ambient 4 - warmed 6    -2.975 2.25  93.2  -1.321  0.9463
## ambient 4 - ambient 6    -3.792 1.65 100.2  -2.291  0.4038
## warmed 5 - ambient 5    -0.817 1.53  26.2  -0.535  0.9999
## warmed 5 - warmed 6     5.083 1.65 100.2   3.072  0.0777
## warmed 5 - ambient 6    4.267 2.25  93.2   1.895  0.6726
## ambient 5 - warmed 6    5.900 2.25  93.2   2.620  0.2234
## ambient 5 - ambient 6    5.083 1.65 100.2   3.072  0.0777
## warmed 6 - ambient 6    -0.817 1.53  26.2  -0.535  0.9999
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 10 estimates

# including native vs. exotic
umbs_sd_plot_origin <- within(umbs_sd_plot_origin, origin <- relevel(factor(origin),
  ref = "Native")) # releveling so native is the reference
mod12pu <- lmer(log(julian_min) ~ state * origin + (1 + year_factor | plot), umbs_sd_plot_origin,
  REML = FALSE)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00645698 (tol = 0.002, component 1)

mod13pu <- lmer(log(julian_min) ~ state + origin + (1 + year_factor | plot), umbs_sd_plot_origin,
  REML = FALSE)

## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 1 negative eigenvalue: -6.7e-02

mod14pu <- lmer(log(julian_min) ~ state + origin + year_factor + (1 | plot), umbs_sd_plot_origin,
  REML = FALSE)
anova(mod12pu, mod13pu) # go with model 13pu

```

```

## Data: umbs_sd_plot_origin
## Models:
## mod13pu: log(julian_min) ~ state + origin + (1 + year_factor | plot)
## mod12pu: log(julian_min) ~ state * origin + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod13pu     8 -560.91 -533.30 288.45   -576.91
## mod12pu    10 -560.82 -526.31 290.41   -580.82 3.9164  2      0.1411

```

```
anova(mod13pu, mod14pu) # mod 14pu
```

```

## Data: umbs_sd_plot_origin
## Models:
## mod14pu: log(julian_min) ~ state + origin + year_factor + (1 | plot)
## mod13pu: log(julian_min) ~ state + origin + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod14pu     7 -561.62 -537.47 287.81   -575.62
## mod13pu     8 -560.91 -533.30 288.45   -576.91 1.2848  1      0.257

```

```
summary(mod14pu)
```

```

## 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_plot_origin
##
##      AIC      BIC logLik deviance df.resid
## -561.6   -537.5   287.8   -575.6      226
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.4958 -0.6412 -0.0811  0.7023  3.3251
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.0003208 0.01791
## Residual           0.0046986 0.06855
## Number of obs: 233, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept) 5.167399  0.015657 174.152793 330.049 <2e-16 ***
## statewarmed -0.003980  0.011631  23.706415 -0.342   0.735  
## originBoth   0.018679  0.020883 232.702099  0.894   0.372  
## originExotic  0.086444  0.009319 213.578718  9.276 <2e-16 ***
## year_factor  0.000469  0.003149 211.921877  0.149   0.882  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttwrm orgnBt orgnEx
## statewarmed -0.372
## originBoth   -0.179  0.011
## originExotic -0.291 -0.017  0.235
## year_factor  -0.788  0.013  0.044 -0.036

```

```
anova(mod14pu)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq  Mean Sq NumDF DenDF F value Pr(>F)
## state      0.00055 0.000550     1   23.706  0.1171 0.7352
## origin     0.41245 0.206226     2 225.309 43.8910 <2e-16 ***
## year_factor 0.00010 0.000104     1 211.922  0.0222 0.8817
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
emmeans(mod14pu, list(pairwise ~ state + origin), adjust = "tukey")
```

```
## $`emmeans of state, origin`
##    state  origin emmean       SE   df lower.CL upper.CL
## ambient Native  5.169 0.010006 48.02   5.149   5.189
## warmed  Native  5.165 0.010196 50.54   5.145   5.186
## ambient Both    5.188 0.021400 198.82  5.146   5.230
## warmed  Both    5.184 0.021629 197.81  5.141   5.227
## ambient Exotic  5.256 0.009660 42.42   5.236   5.275
## warmed  Exotic  5.252 0.009657 42.08   5.232   5.271
##
## 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
## ambient Native - warmed Native  0.00398 0.0122 25.9  0.326  0.9994
## ambient Native - ambient Both   -0.01868 0.0213 237.3 -0.878  0.9516
## ambient Native - warmed Both   -0.01470 0.0247 170.1 -0.596  0.9912
## ambient Native - ambient Exotic -0.08644 0.0094 216.5 -9.199 <.0001
## ambient Native - warmed Exotic -0.08246 0.0153 64.3 -5.401 <.0001
## warmed Native - ambient Both   -0.02266 0.0244 171.5 -0.928  0.9387
## warmed Native - warmed Both   -0.01868 0.0213 237.3 -0.878  0.9516
## warmed Native - ambient Exotic -0.09042 0.0155 67.3 -5.825 <.0001
## warmed Native - warmed Exotic -0.08644 0.0094 216.5 -9.199 <.0001
## ambient Both - warmed Both    0.00398 0.0122 25.9  0.326  0.9994
## ambient Both - ambient Exotic -0.06777 0.0212 238.1 -3.196  0.0195
## ambient Both - warmed Exotic -0.06379 0.0243 166.2 -2.630  0.0959
## warmed Both - ambient Exotic -0.07175 0.0247 166.0 -2.909  0.0466
## warmed Both - warmed Exotic  -0.06777 0.0212 238.1 -3.196  0.0195
## ambient Exotic - warmed Exotic  0.00398 0.0122 25.9  0.326  0.9994
##
## 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(mod14pu, list(pairwise ~ origin), adjust = "tukey")
```

```
## $`emmeans of origin`
##   origin emmean       SE   df lower.CL upper.CL
```

```

##  Native  5.167 0.008053  76.28    5.151    5.183
##  Both    5.186 0.020632 224.53    5.145    5.227
##  Exotic  5.254 0.007489  61.31    5.239    5.269
##
## 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 origin'
##   1           estimate      SE  df t.ratio p.value
##  Native - Both    -0.0187 0.0213 237 -0.878  0.6550
##  Native - Exotic  -0.0864 0.0094 217 -9.199 <.0001
##  Both - Exotic    -0.0678 0.0212 238 -3.196  0.0045
##
## 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

# including growth form - first with interaction term
umbss_sd_plot_growthhabit <- within(umbss_sd_plot_growthhabit, growth_habit <- relevel(factor(growth_habit,
  ref = "Forb")) # releveling so forb is the reference
mod15pu <- lmer(log(julian_min) ~ state * growth_habit + (1 + year_factor | plot),
  umbss_sd_plot_growthhabit, REML = FALSE)

## boundary (singular) fit: see ?isSingular

## Warning: Model failed to converge with 1 negative eigenvalue: -6.6e-02

mod16pu <- lmer(log(julian_min) ~ state + growth_habit + (1 + year_factor | plot),
  umbss_sd_plot_growthhabit, REML = FALSE)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00743057 (tol = 0.002, component 1)

mod17pu <- lmer(log(julian_min) ~ state + growth_habit + year_factor + (1 | plot),
  umbss_sd_plot_growthhabit, REML = FALSE)
anova(mod15pu, mod16pu) # go with model 16pu

## Data: umbss_sd_plot_growthhabit
## Models:
## mod16pu: log(julian_min) ~ state + growth_habit + (1 + year_factor | plot)
## mod15pu: log(julian_min) ~ state * growth_habit + (1 + year_factor | plot)
##          npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod16pu     8 -682.08 -653.97 349.04  -698.08
## mod15pu    10 -678.22 -643.09 349.11  -698.22 0.1448  2      0.9301

anova(mod16pu, mod17pu) # mod 17pu

```

```

## Data: umbs_sd_plot_growthhabit
## Models:
## mod17pu: log(julian_min) ~ state + growth_habit + year_factor + (1 | plot)
## mod16pu: log(julian_min) ~ state + growth_habit + (1 + year_factor | plot)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod17pu     7 -684.29 -659.70 349.15   -698.29
## mod16pu     8 -682.08 -653.97 349.04   -698.08      0  1          1

summary(mod17pu)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + growth_habit + year_factor + (1 | plot)
## Data: umbs_sd_plot_growthhabit
##
##      AIC      BIC      logLik deviance df.resid
## -684.3    -659.7    349.1    -698.3      241
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.5614 -0.6795 -0.0298  0.6921  3.6643
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 1.489e-05 0.003859
## Residual           3.490e-03 0.059079
## Number of obs: 248, groups: plot, 24
##
## Fixed effects:
##                   Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)      5.313390  0.012688 225.970280 418.774 < 2e-16 ***
## statewarmed      0.001336  0.007674  22.674484   0.174  0.863
## growth_habit     -0.159688  0.019719 221.715677  -8.098 3.69e-14 ***
## growth_habitGraminoid -0.135005  0.007660 223.328344 -17.625 < 2e-16 ***
## year_factor      -0.001417  0.002678 223.984005  -0.529  0.597
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) sttwrm grwth_ grwt_G
## statewarmed -0.296
## growth_habit -0.215 -0.038
## grwth_hbtGr -0.314  0.000  0.197
## year_factor -0.849 -0.008  0.127  0.012

anova(mod17pu)

## Type III Analysis of Variance Table with Satterthwaite's method
##             Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## state       0.00011 0.00011     1  22.674  0.0303 0.8633
## growth_habit 1.16176 0.58088     2 225.720 166.4228 <2e-16 ***
## year_factor  0.00098 0.00098     1 223.984  0.2799 0.5973
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

emmeans(mod17pu, list(pairwise ~ state + growth_habit), adjust = "tukey")

## $`emmeans of state, growth_habit`
##   state   growth_habit emmean      SE     df lower.CL upper.CL
##   ambient   Forb      5.308 0.006947  59.30    5.294    5.322
##   warmed   Forb      5.309 0.006949  59.64    5.295    5.323
##   ambient      Forb      5.148 0.020192 196.17    5.108    5.188
##   warmed      Forb      5.149 0.019907 191.32    5.110    5.189
##   ambient  Graminoid    5.173 0.006910  58.22    5.159    5.187
##   warmed  Graminoid    5.174 0.006910  58.52    5.160    5.188
##
## 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, growth_habit`
##   1                           estimate      SE     df t.ratio p.value
##   ambient Forb - warmed Forb -0.00134 0.00808  26.2 -0.165  1.0000
##   ambient Forb - ambient      0.15969 0.02035 235.2  7.846 <.0001
##   ambient Forb - warmed      0.15835 0.02163 177.1  7.320 <.0001
##   ambient Forb - ambient Graminoid 0.13501 0.00771 226.9 17.516 <.0001
##   ambient Forb - warmed Graminoid 0.13367 0.01116 93.8 11.974 <.0001
##   warmed Forb - ambient      0.16102 0.02216 186.0  7.268 <.0001
##   warmed Forb - warmed      0.15969 0.02035 235.2  7.846 <.0001
##   warmed Forb - ambient Graminoid 0.13634 0.01116 93.9 12.212 <.0001
##   warmed Forb - warmed Graminoid 0.13501 0.00771 226.9 17.516 <.0001
##   ambient - warmed      -0.00134 0.00808  26.2 -0.165  1.0000
##   ambient - ambient Graminoid -0.02468 0.02033 234.7 -1.214  0.8298
##   ambient - warmed Graminoid -0.02602 0.02214 185.3 -1.175  0.8481
##   warmed - ambient Graminoid -0.02335 0.02161 176.3 -1.080  0.8887
##   warmed - warmed Graminoid -0.02468 0.02033 234.7 -1.214  0.8298
##   ambient Graminoid - warmed Graminoid -0.00134 0.00808  26.2 -0.165  1.0000
##
## 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(mod17pu, list(pairwise ~ growth_habit), adjust = "tukey")
```

```

## $`emmeans of growth_habit`
##   growth_habit emmean      SE     df lower.CL upper.CL
##   Forb          5.308 0.005654  98.68    5.297    5.320
##   Graminoid     5.149 0.019639 209.66    5.110    5.187
##   Graminoid     5.173 0.005607  96.44    5.162    5.185
##
## 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 -          0.1597 0.02035 235  7.846 <.0001
##  Forb - Graminoid 0.1350 0.00771 227 17.516 <.0001
##  - Graminoid     -0.0247 0.02033 235 -1.214  0.4461
##
## 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
```