

# warmXtrophic Project: Seed Set Phenology Analyses

Moriah Young

January 27, 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)

## Install package "strengejacke" from GitHub ('devtools::install_github("strengejacke/strengejacke")')

library(effects)

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

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

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

##
## Attaching package: 'GGally'

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

library(bbmle)  # AICtab() function

##
## Loading required package: stats4

##
## Attaching package: 'bbmle'

```

```

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

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

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

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

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

# Read in data
sd_species <- read.csv(file.path(L2_dir, "phenology/final_sd_species_L2.csv")) # species level data
sd_plot <- read.csv(file.path(L2_dir, "phenology/final_sd_plot_L2.csv")) # plot level data
sd_plot_origin <- read.csv(file.path(L2_dir, "phenology/final_sd_plot_origin_L2.csv")) # plot level data
sd_plot_growthhabit <- read.csv(file.path(L2_dir, "phenology/final_sd_plot_growthhabit_L2.csv")) # plot level data

# 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
# take a look at the data to see if looks good
View(sd_species)
View(sd_plot)

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

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

## [1] "ambient" "warmed"

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

## [1] "warmed"   "ambient"

# [1] 'warmed'   'ambient'

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

```

```

## [1] "ambient" "warmed"

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

## [1] "warmed"   "ambient"

# [1] 'warmed'   'ambient'

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

umbssd_plot <- subset(sd_plot, site == "umbss") # pull out umbs only data at plot level
umbssd_plot <- subset(umbssd_plot, year != "2021") # delete 2021 data from umbs dataframe (doesn't matter)
umbssd_spp <- subset(sd_species, site == "umbss") # pull out umbs only data at species level
umbssd_spp <- subset(umbssd_spp, year != "2021") # delete 2021 data from umbs dataframe (doesn't matter)
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")
umbssd_plot_origin <- subset(sd_plot_origin, site == "umbss")
umbssd_plot_origin <- subset(umbssd_plot_origin, year != "2021") # delete 2021 data
umbssd_plot_growthhabit <- subset(sd_plot_growthhabit, site == "umbss")
umbssd_plot_growthhabit <- subset(umbssd_plot_growthhabit, year != "2021") # delete 2021 data

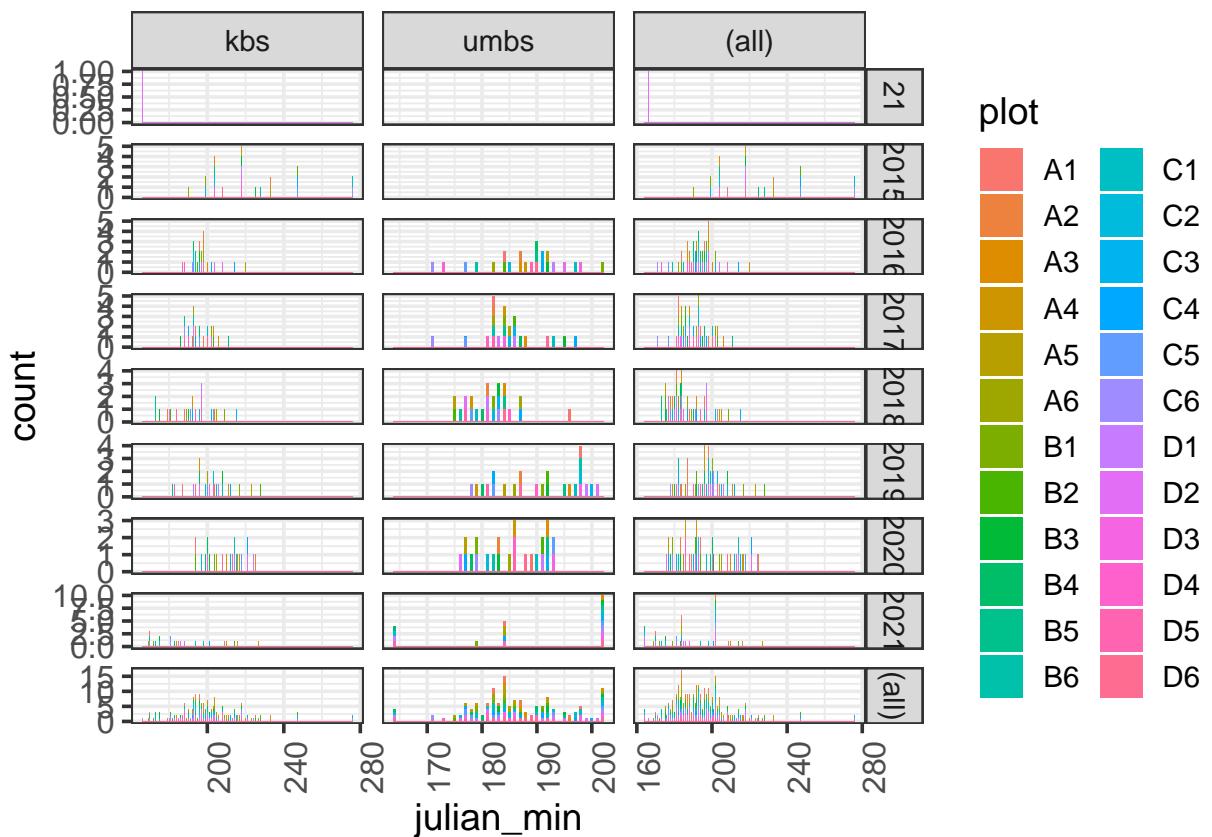
```

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

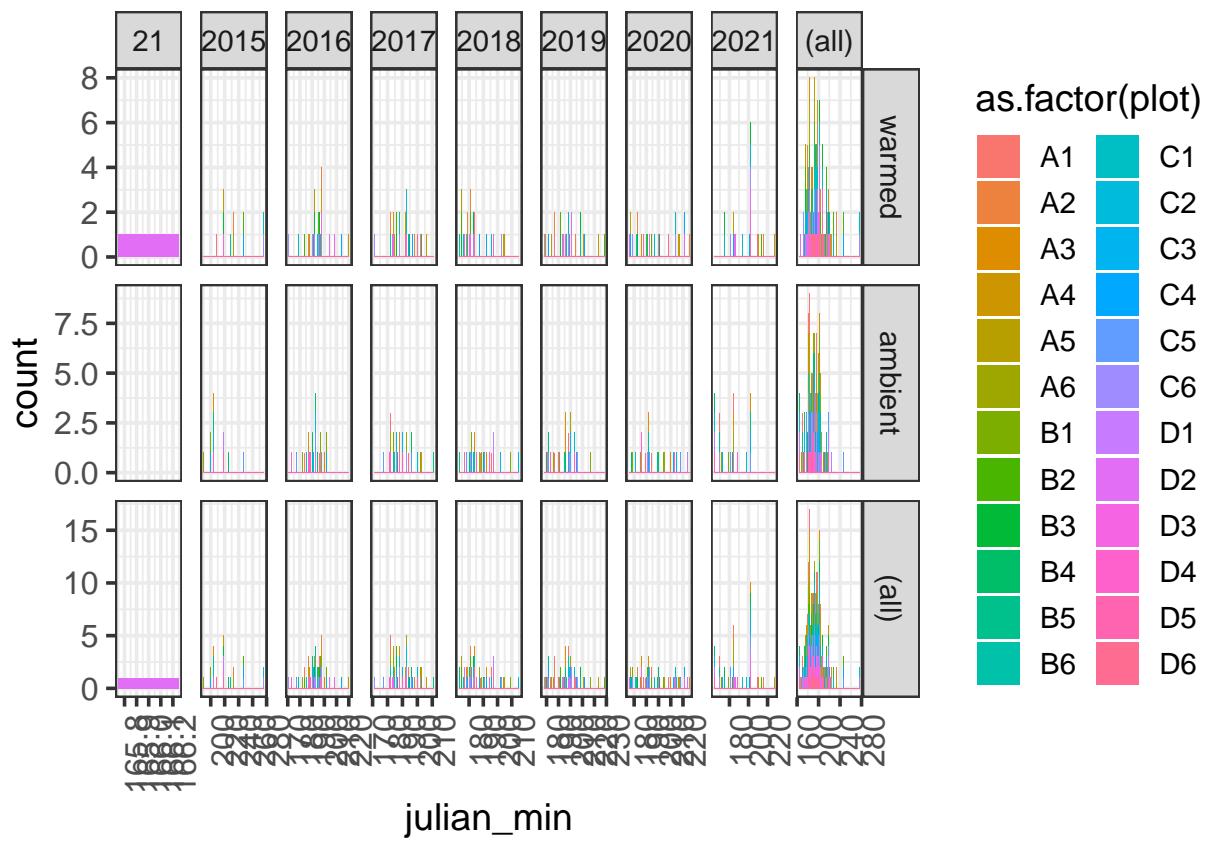
```

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

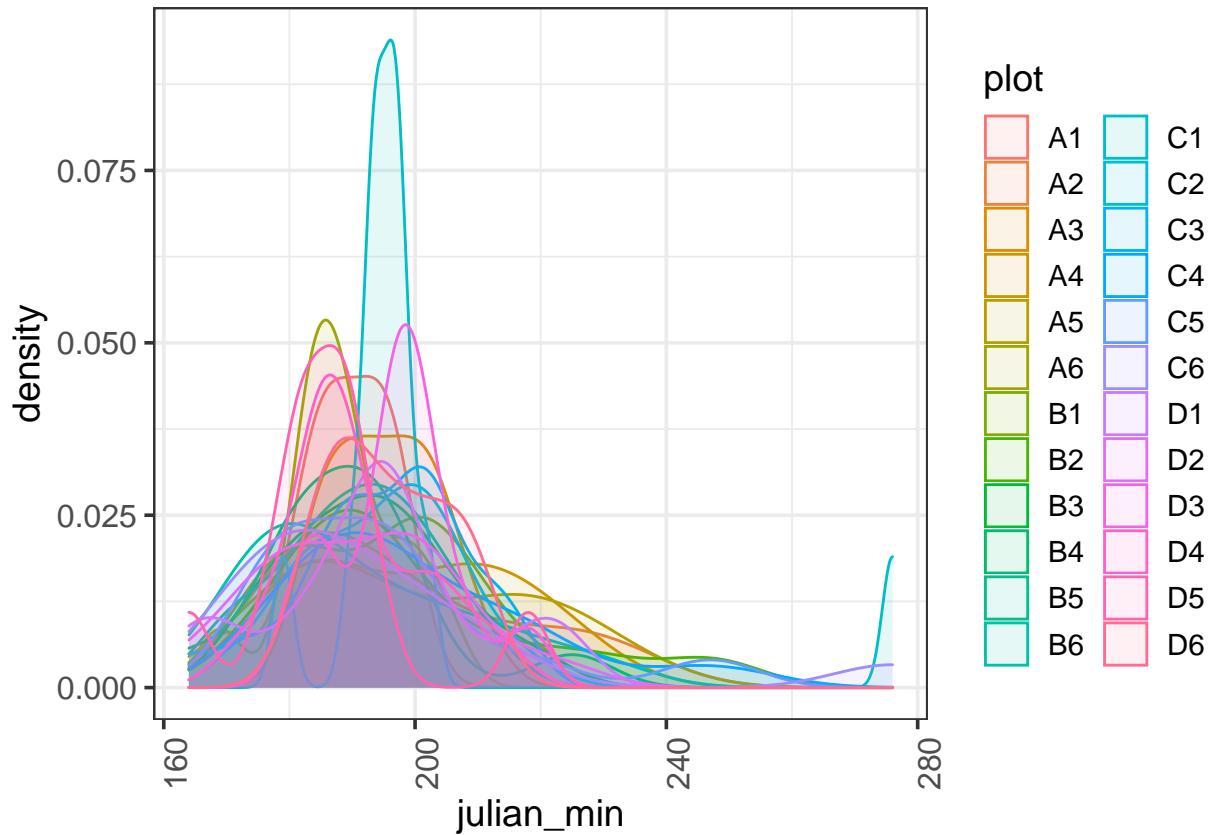
```



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



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



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

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



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

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

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

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

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

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

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

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

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

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

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

```

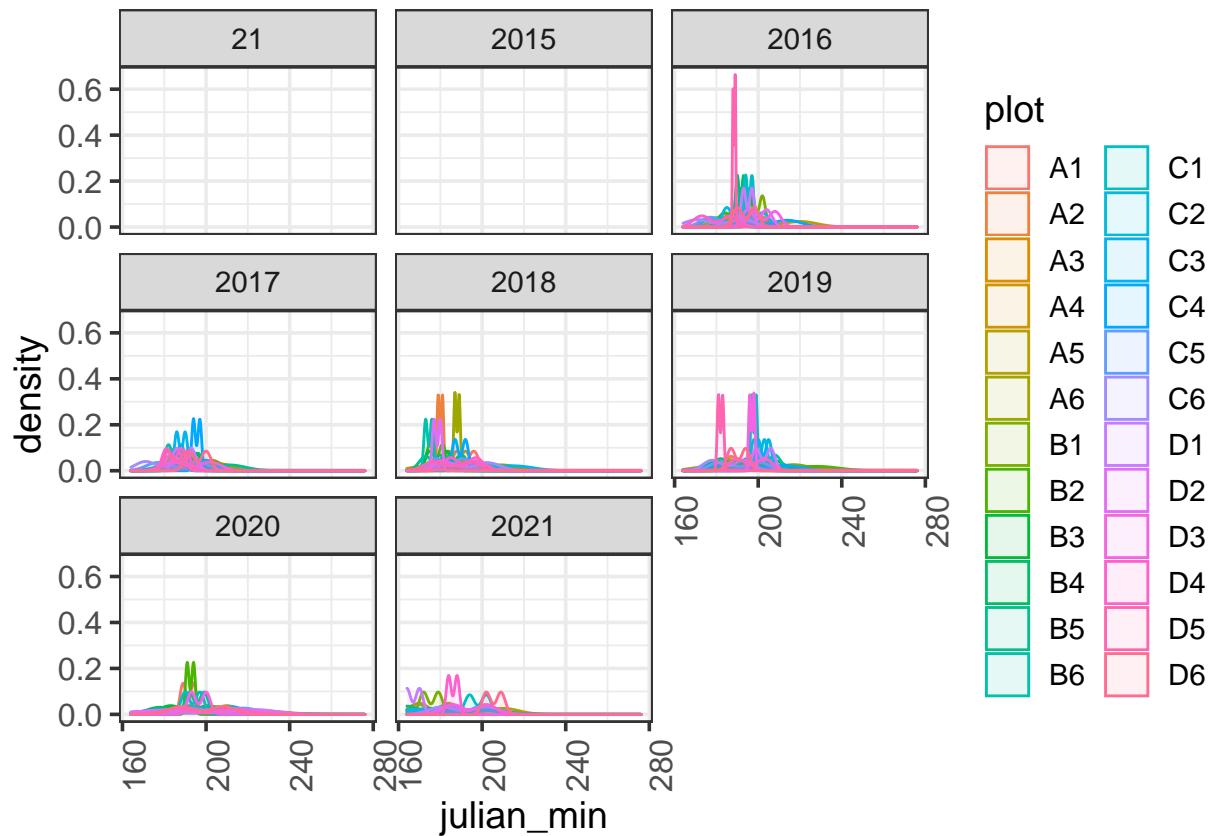
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

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

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

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

```



```

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

```

```

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

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

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

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

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

```



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

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

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

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

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

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

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

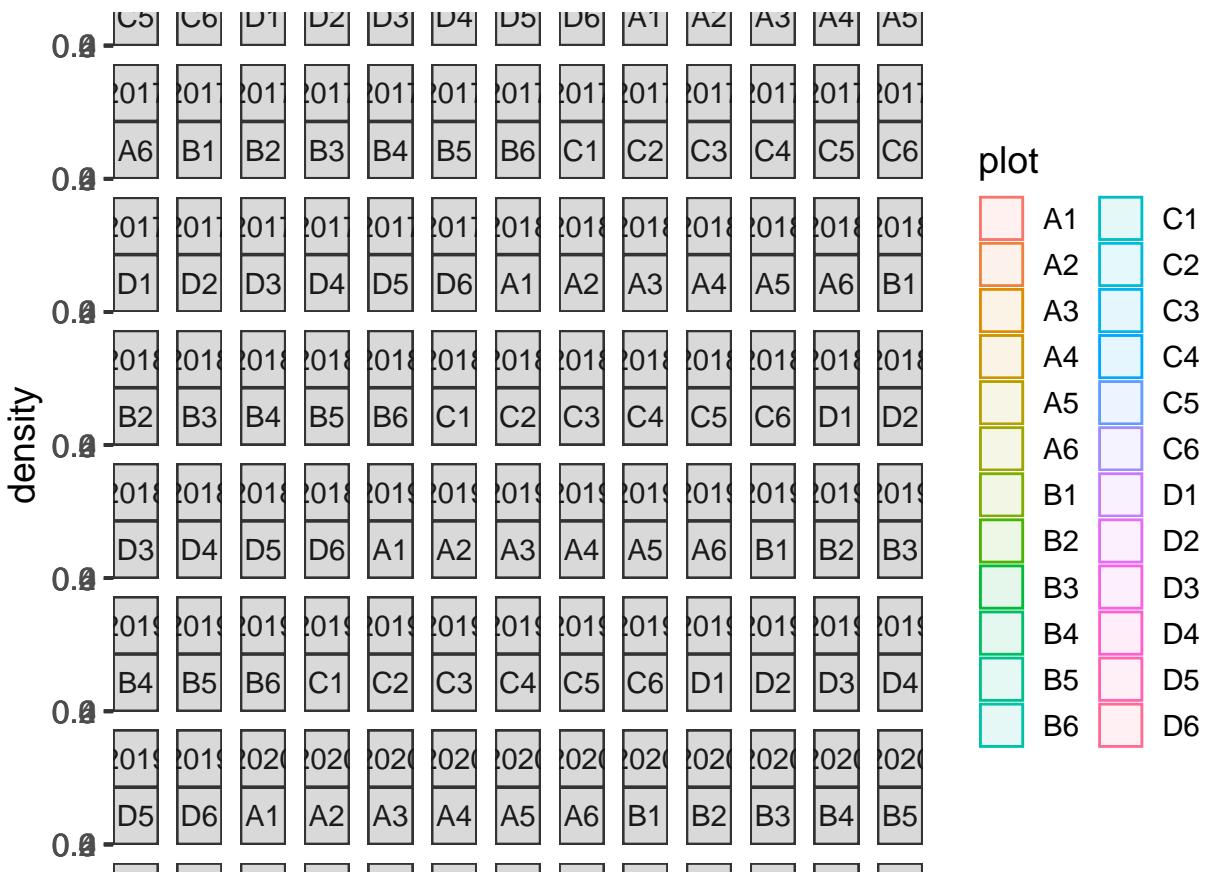
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

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

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

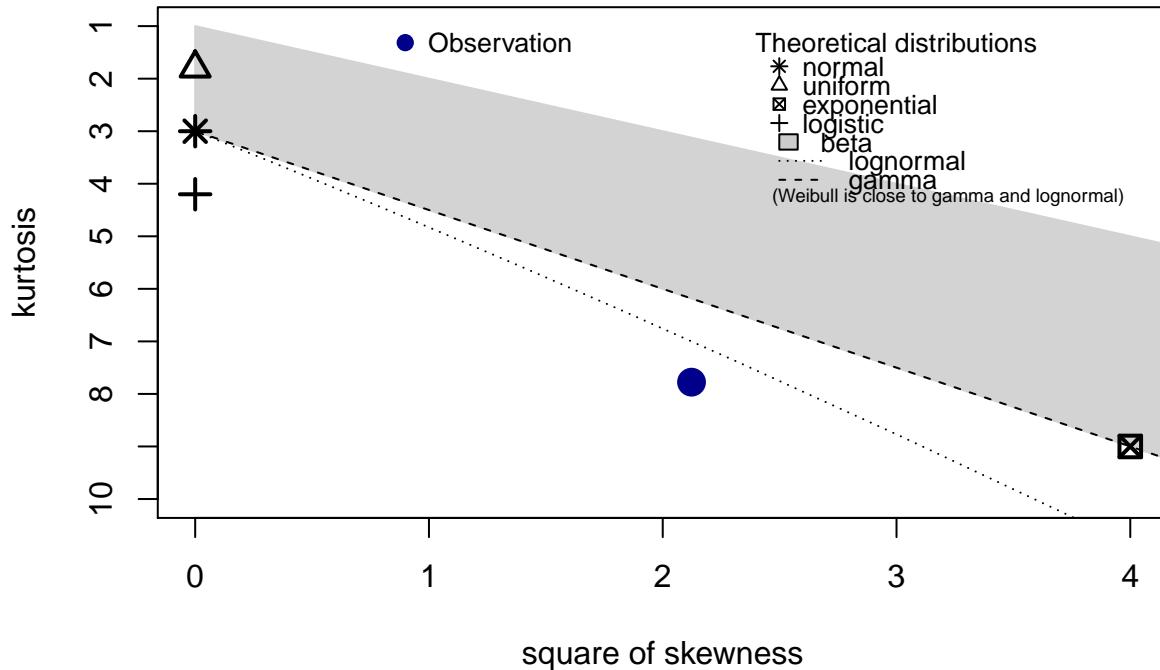
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

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



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

### Cullen and Frey graph

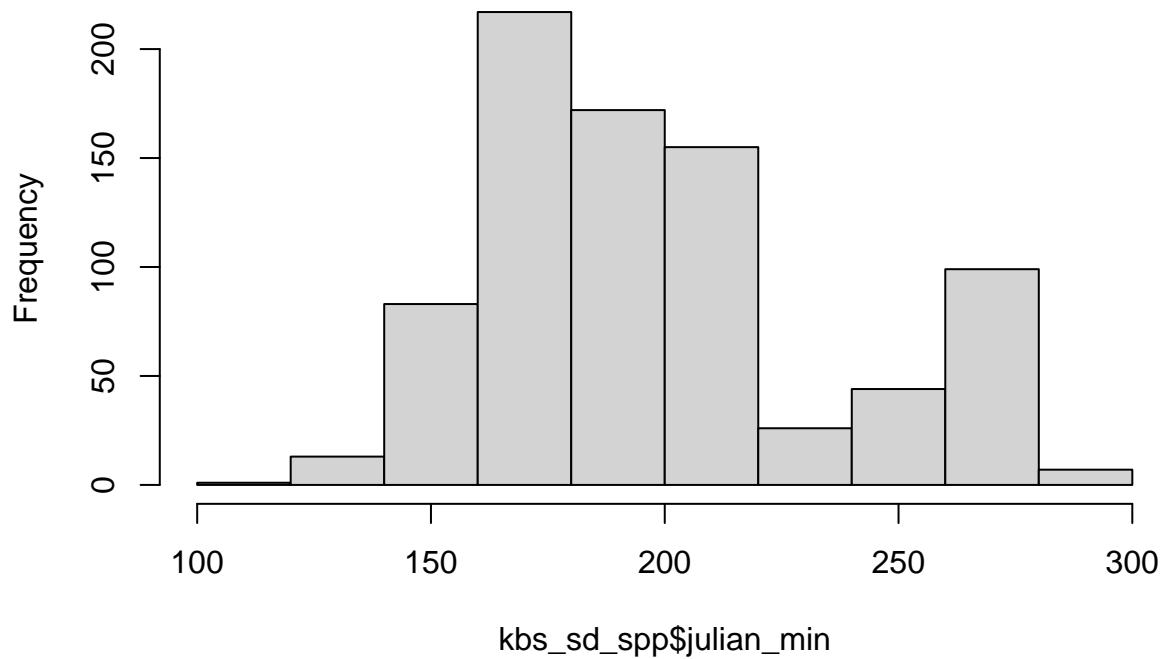


```
## summary statistics
## -----
## min: 164   max: 276
## median: 193
## mean: 194.2361
## estimated sd: 15.68764
## estimated skewness: 1.45713
## estimated kurtosis: 7.776184
```

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

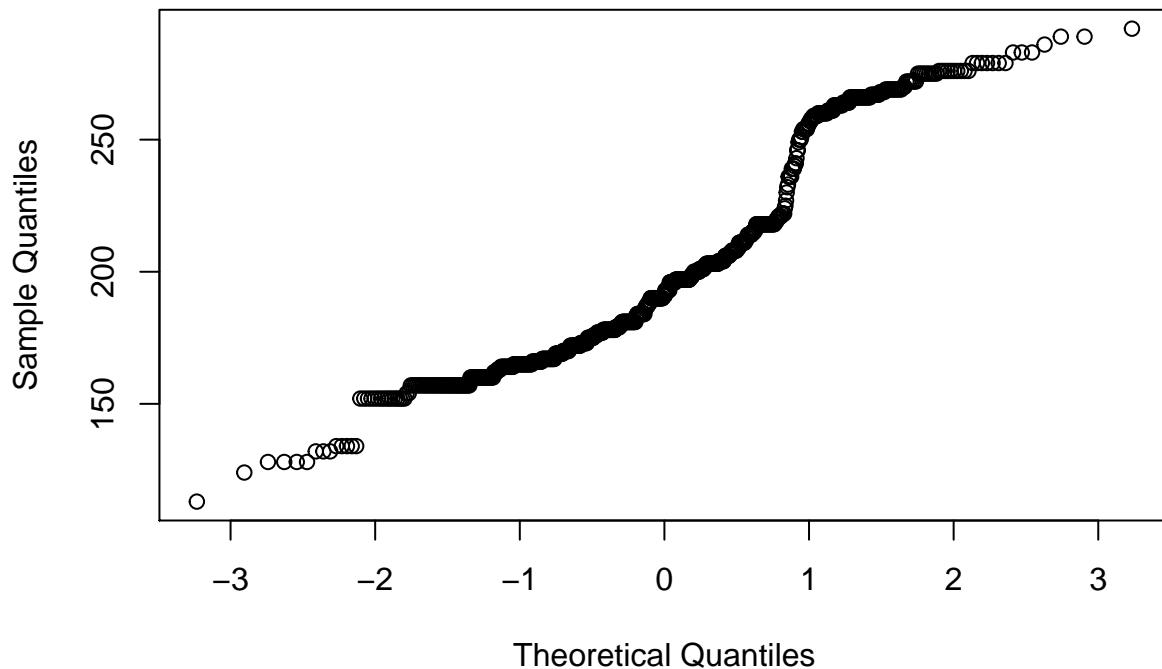
```
### KBS ####
hist(kbs_sd_spp$julian_min)
```

### Histogram of kbs\_sd\_spp\$julian\_min



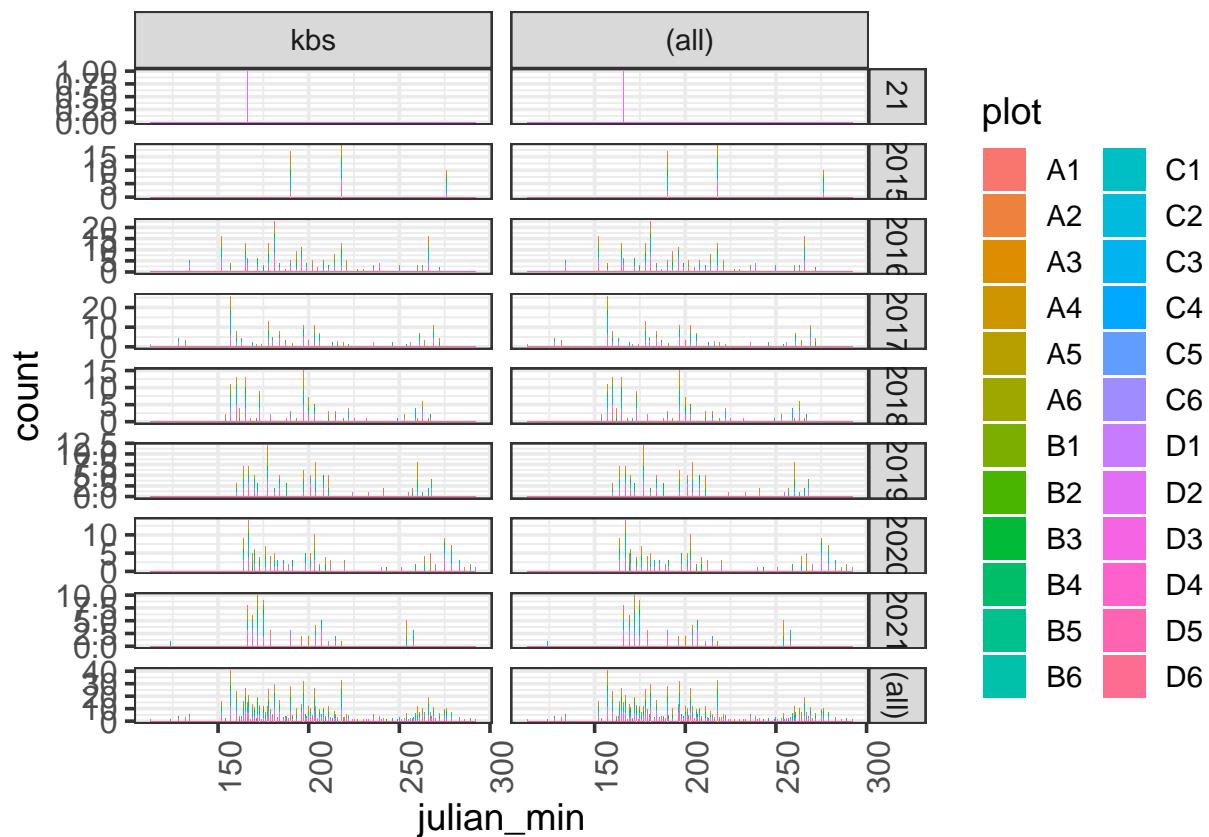
```
qqnorm(kbs_sd_spp$julian_min)
```

### Normal Q-Q Plot

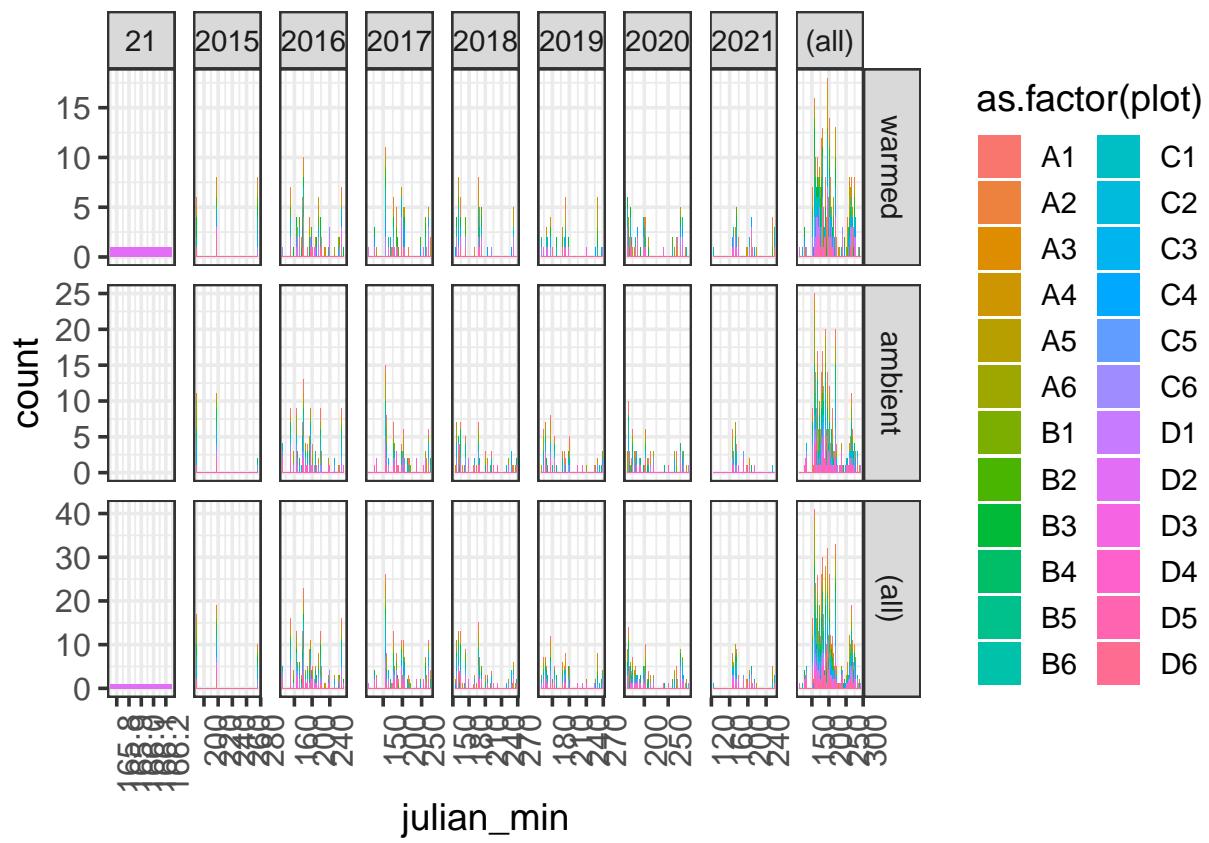


```
shapiro.test(kbs_sd_spp$julian_min) # pvalue is < 0.05 so we reject the null hypothesis that the data
```

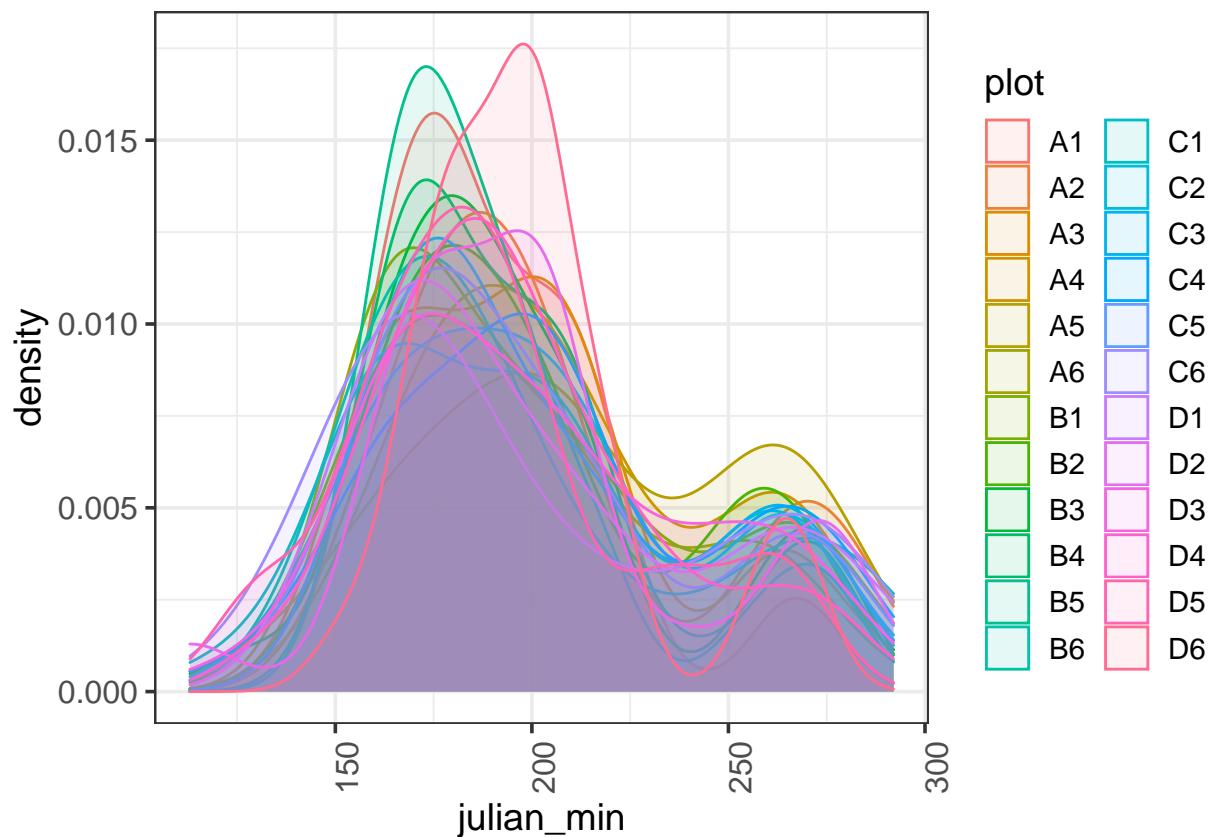
```
##  
## Shapiro-Wilk normality test  
##  
## data: kbs_sd_spp$julian_min  
## W = 0.91991, p-value < 2.2e-16  
  
# Visualizing first Julian date for kbs at the SPECIES LEVEL  
ggplot(kbs_sd_spp, aes(julian_min, fill = plot)) + geom_histogram(binwidth = 0.5) +  
  facet_grid(year ~ site, margins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



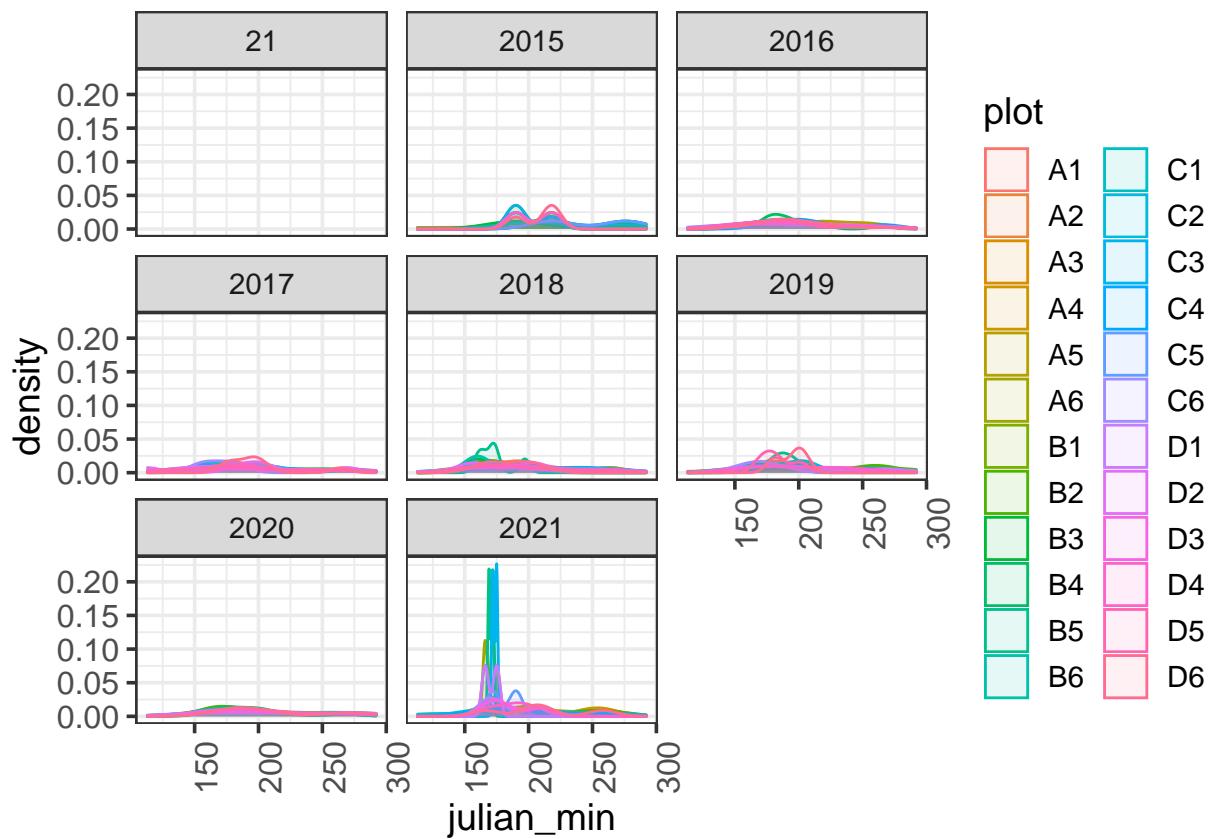
```
ggplot(kbs_sd_spp, aes(julian_min, fill = as.factor(plot))) + geom_histogram(binwidth = 0.5) +  
  facet_grid(state ~ year, margins = TRUE, scales = "free")
```



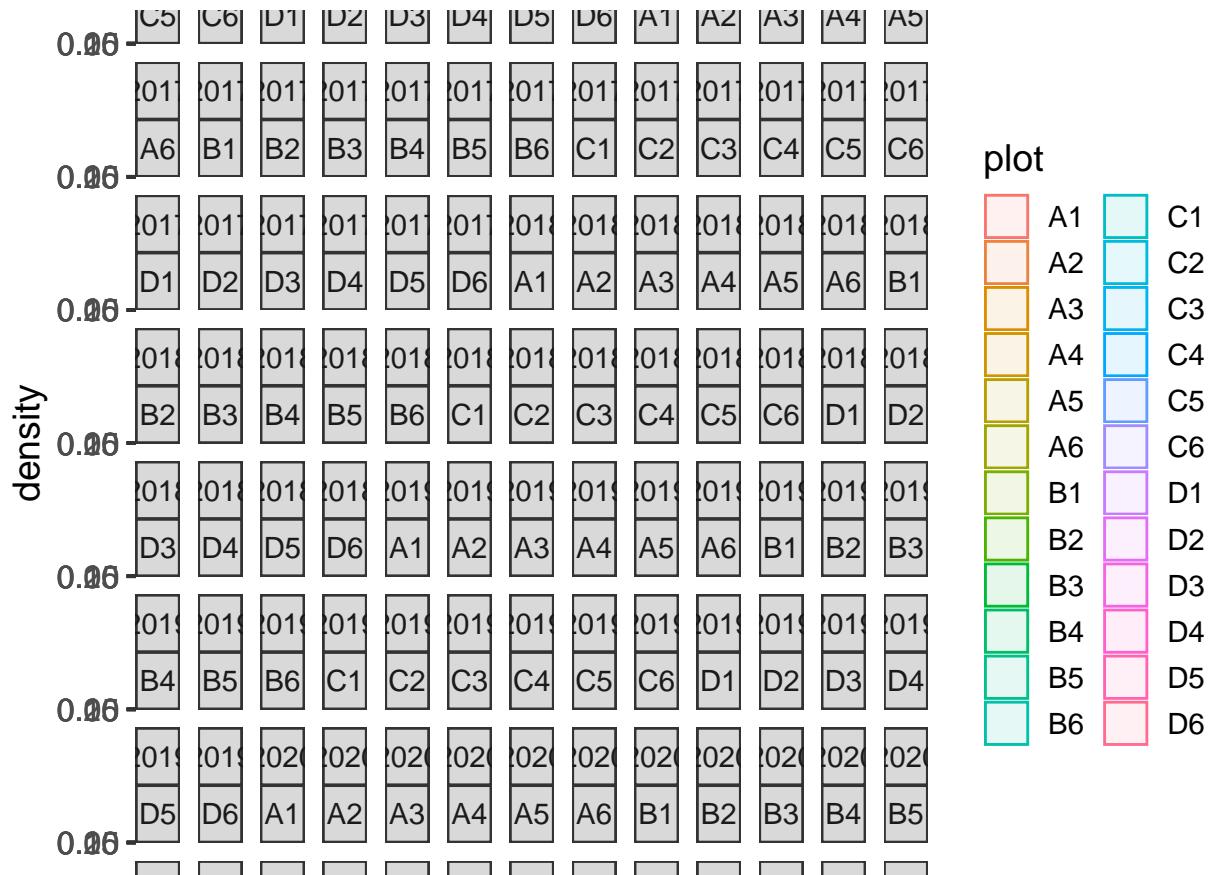
```
ggplot(kbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

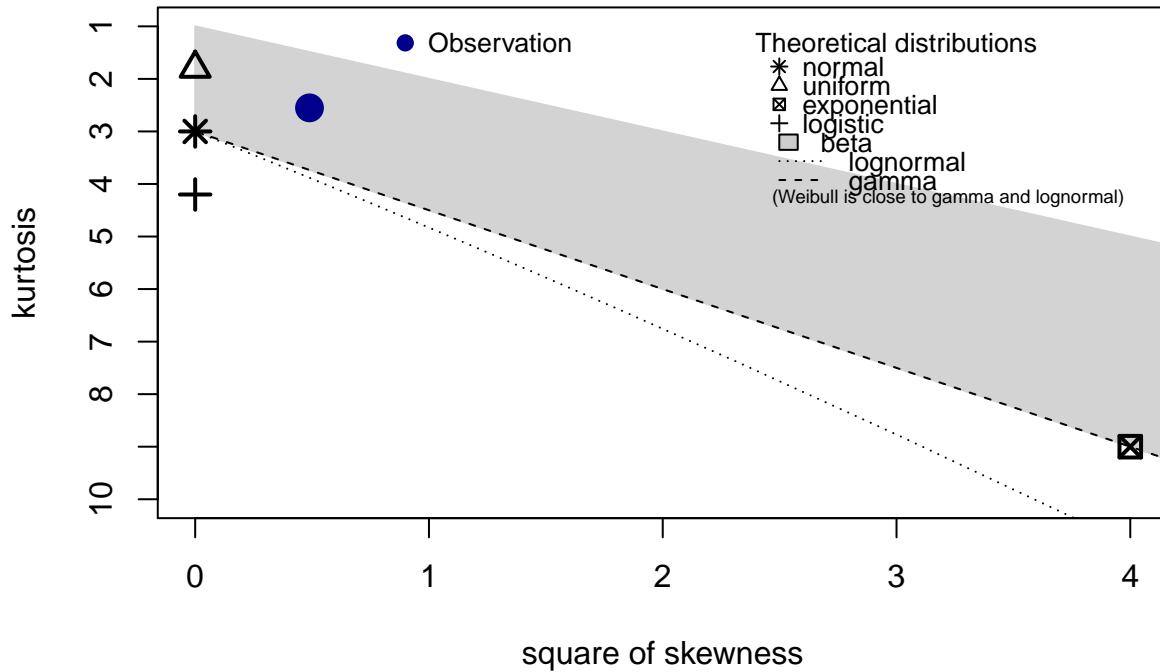


```
ggplot(kbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



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

## Cullen and Frey graph

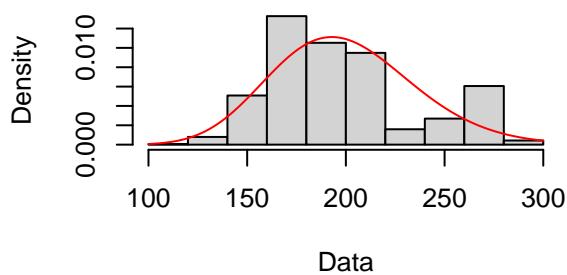
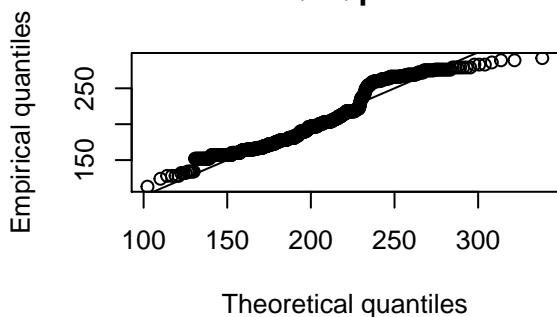
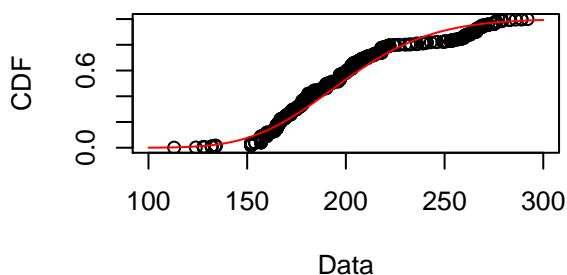
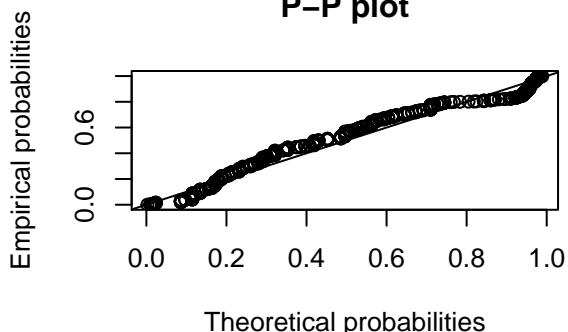


```

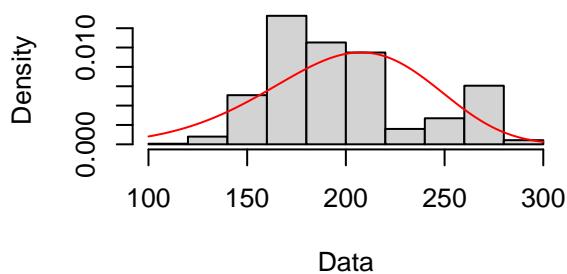
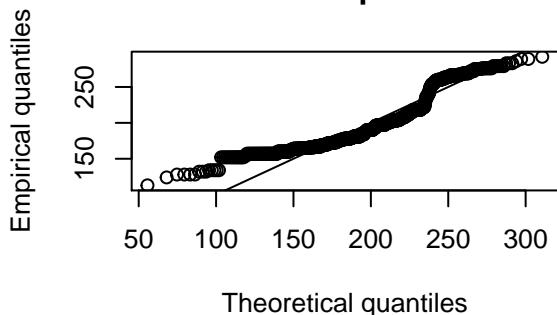
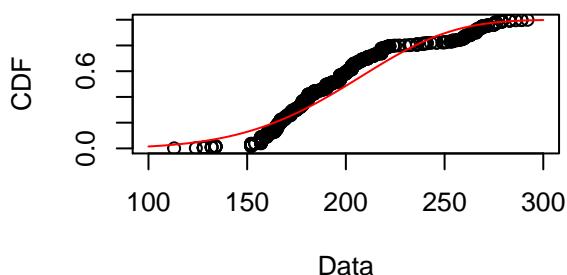
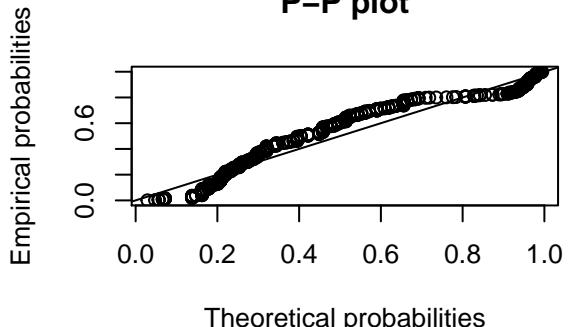
## summary statistics
## -----
## min: 113  max: 292
## median: 191
## mean: 199.5165
## estimated sd: 37.4115
## estimated skewness: 0.6993432
## estimated kurtosis: 2.551332

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

```

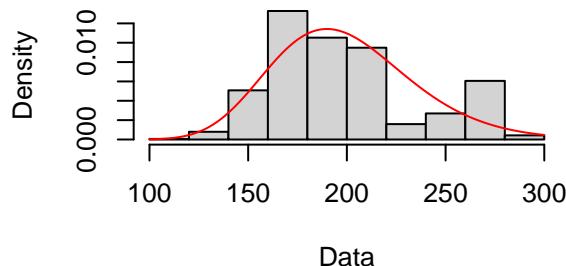
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

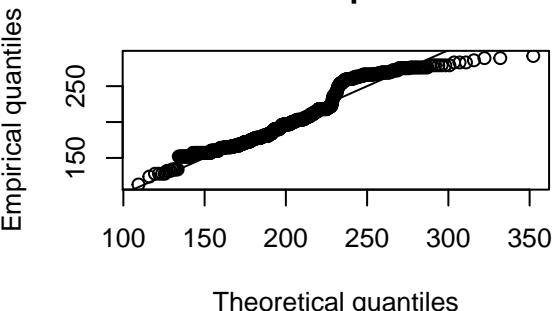
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

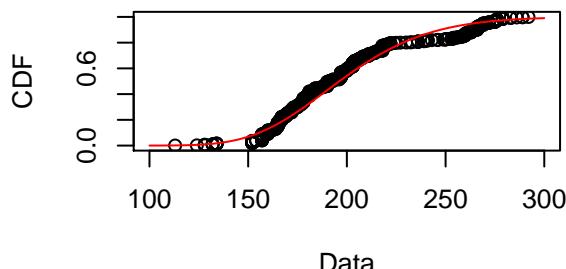
**Empirical and theoretical dens.**



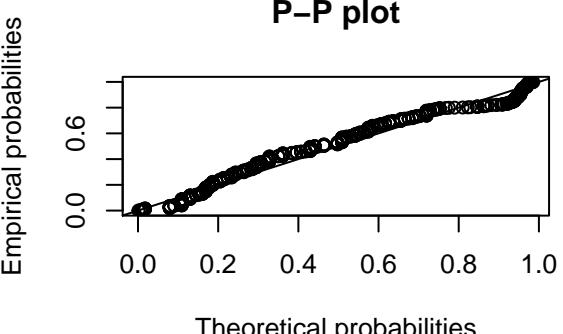
**Q-Q plot**



**Empirical and theoretical CDFs**

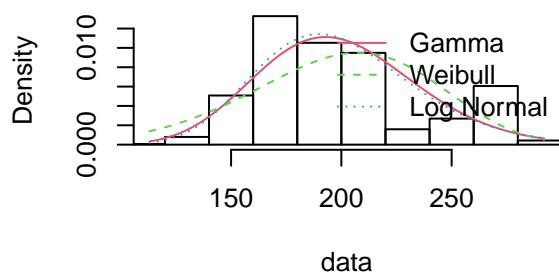


**P-P plot**

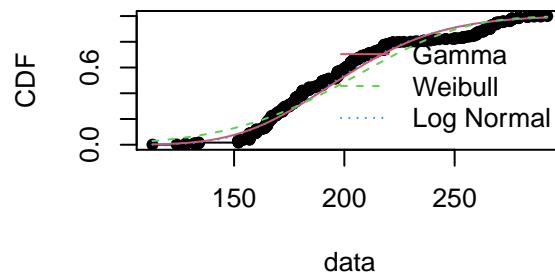


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

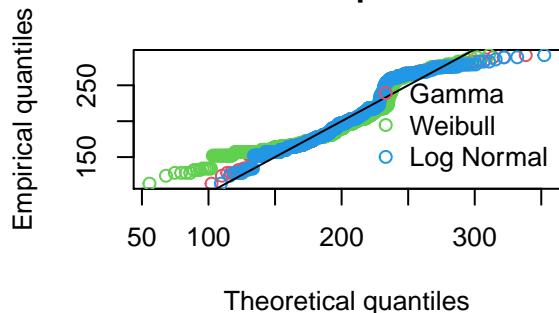
### Histogram and theoretical densities



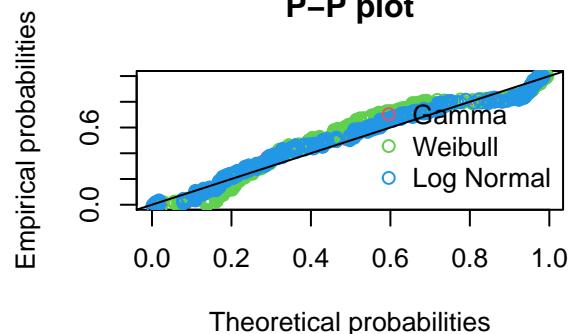
### Empirical and theoretical CDFs



### Q-Q plot



### P-P plot



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

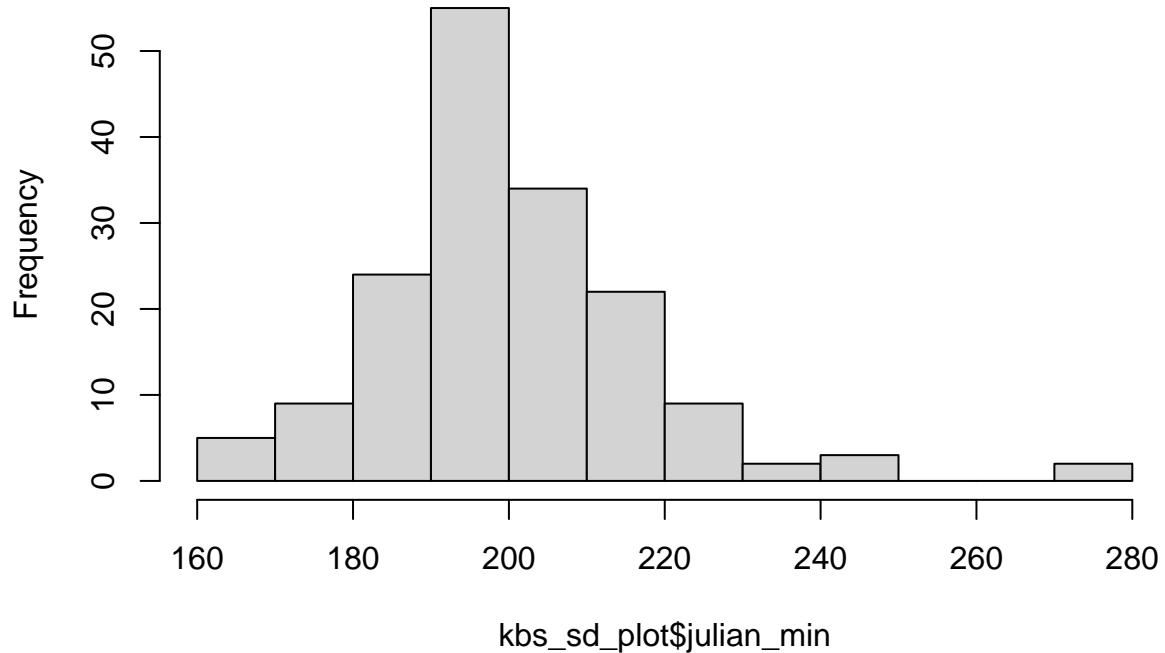
```
## Goodness-of-fit statistics
##                                     Gamma   Weibull  Log Normal
## Kolmogorov-Smirnov statistic 0.09988396 0.1360783 0.09317306
## Cramer-von Mises statistic   2.41253973 5.0192692 1.88913912
## Anderson-Darling statistic  17.07536252 30.9119372 14.02047891
##
##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

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

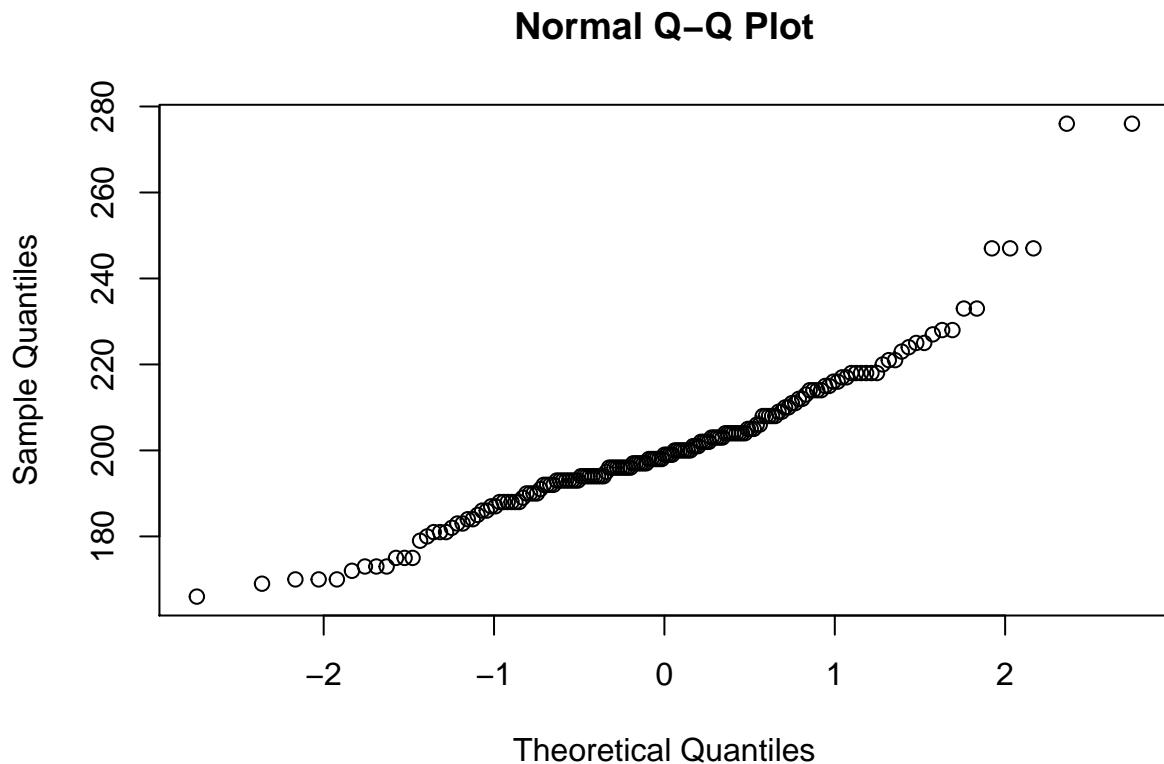
## KBS PLOT LEVEL - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

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

### Histogram of kbs\_sd\_plot\$julian\_min



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



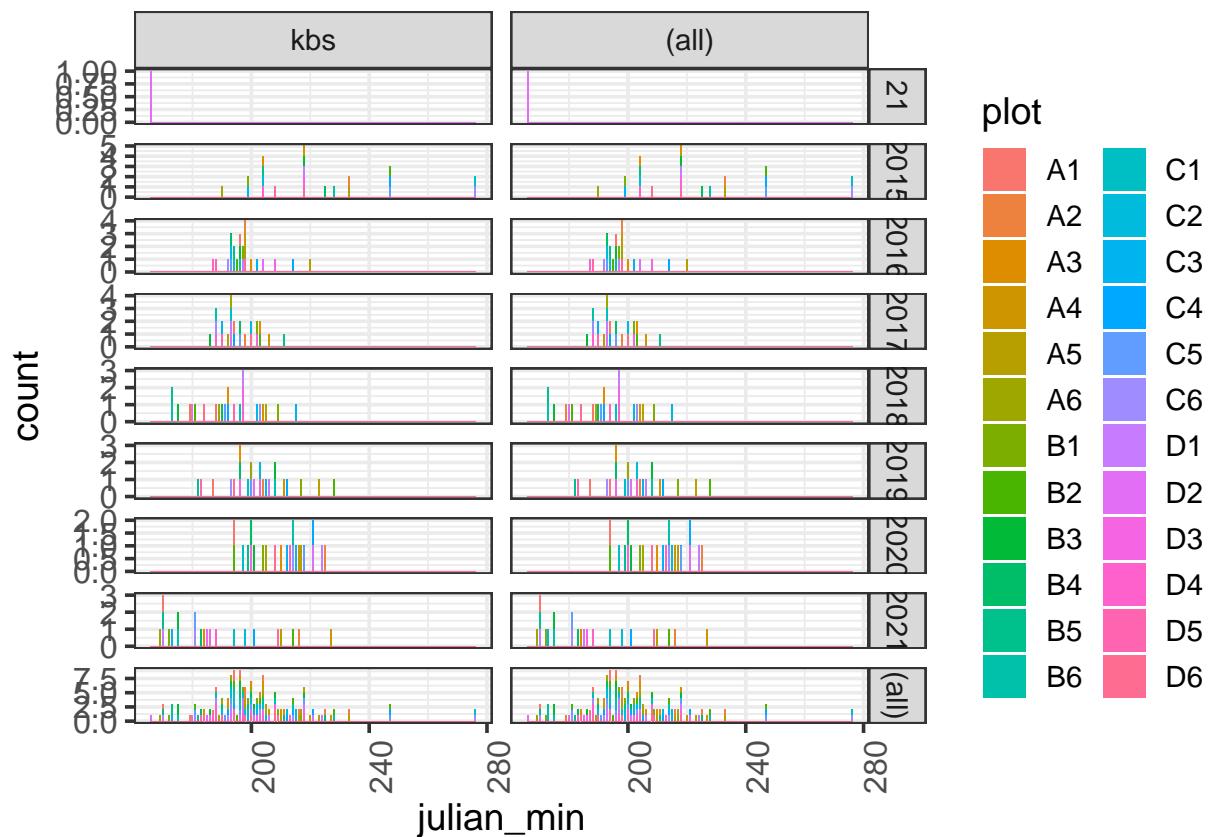
```

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

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

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

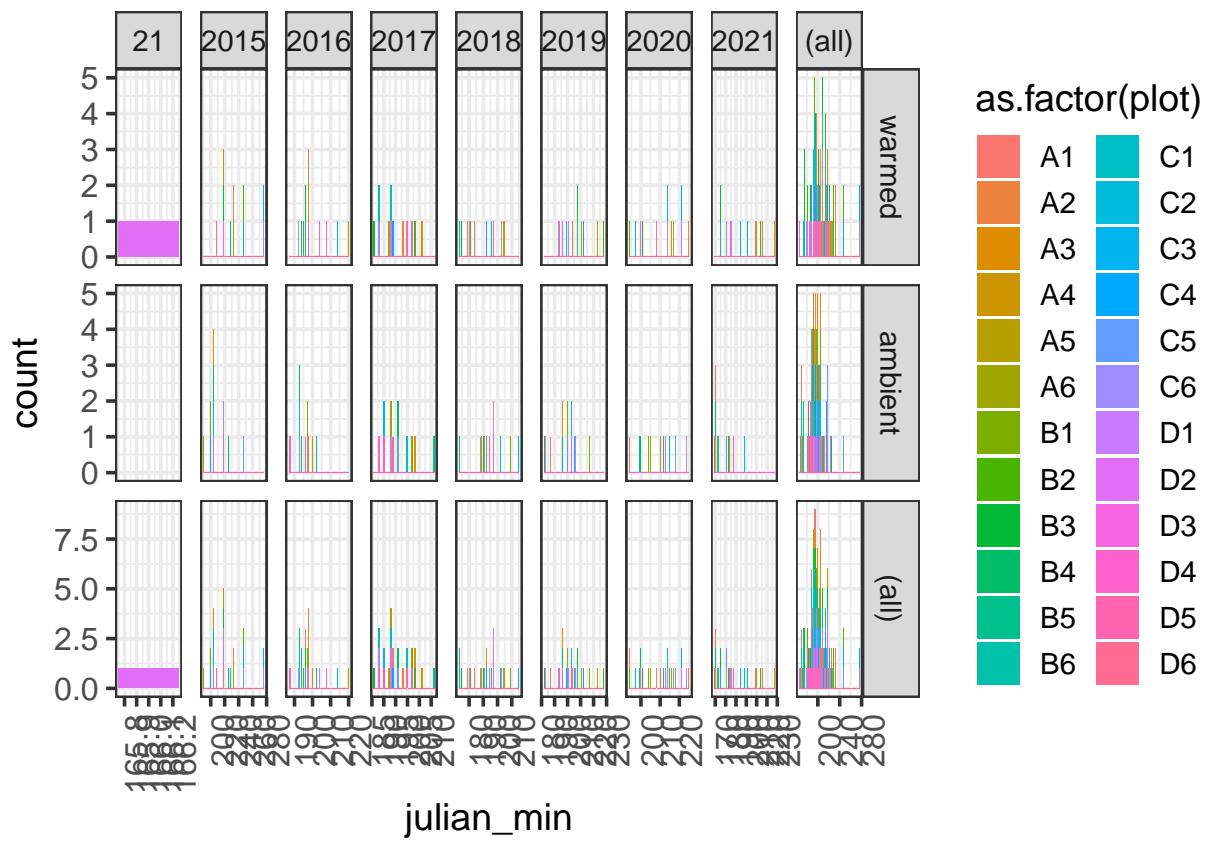
```



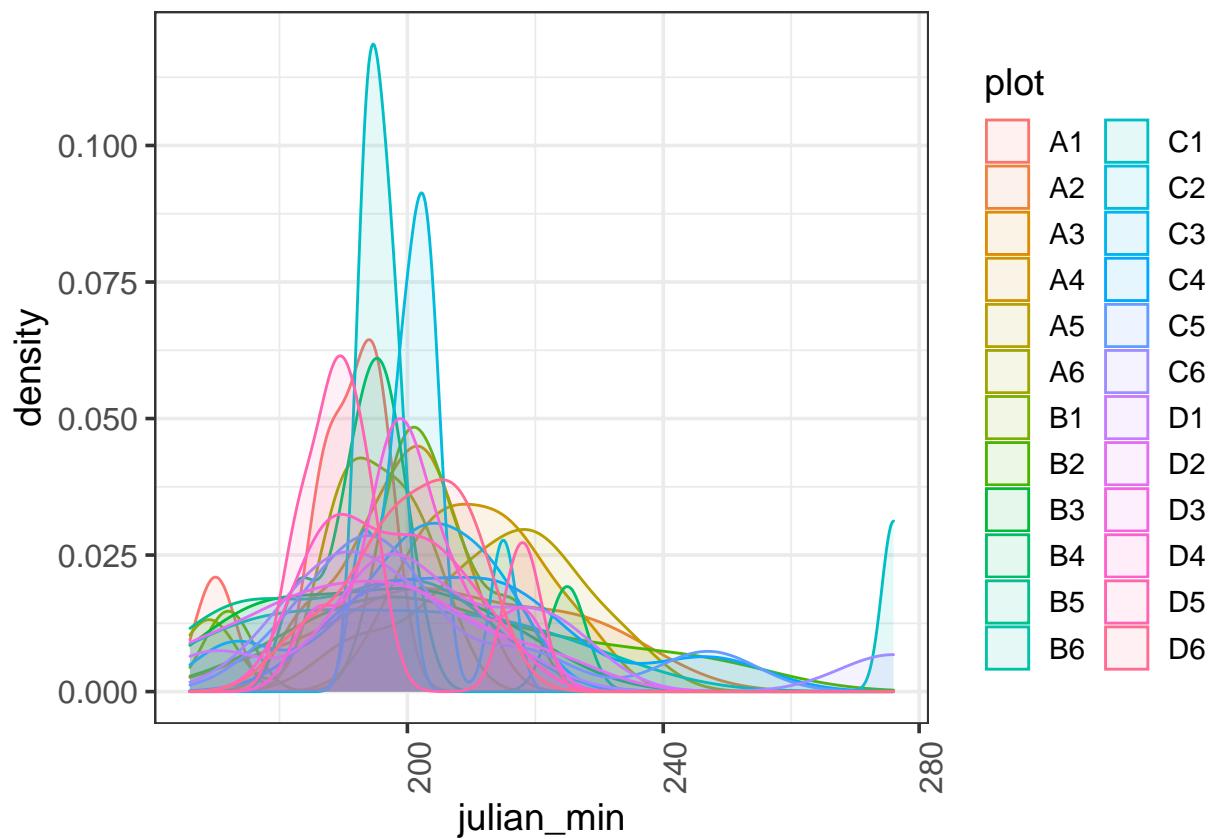
```

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

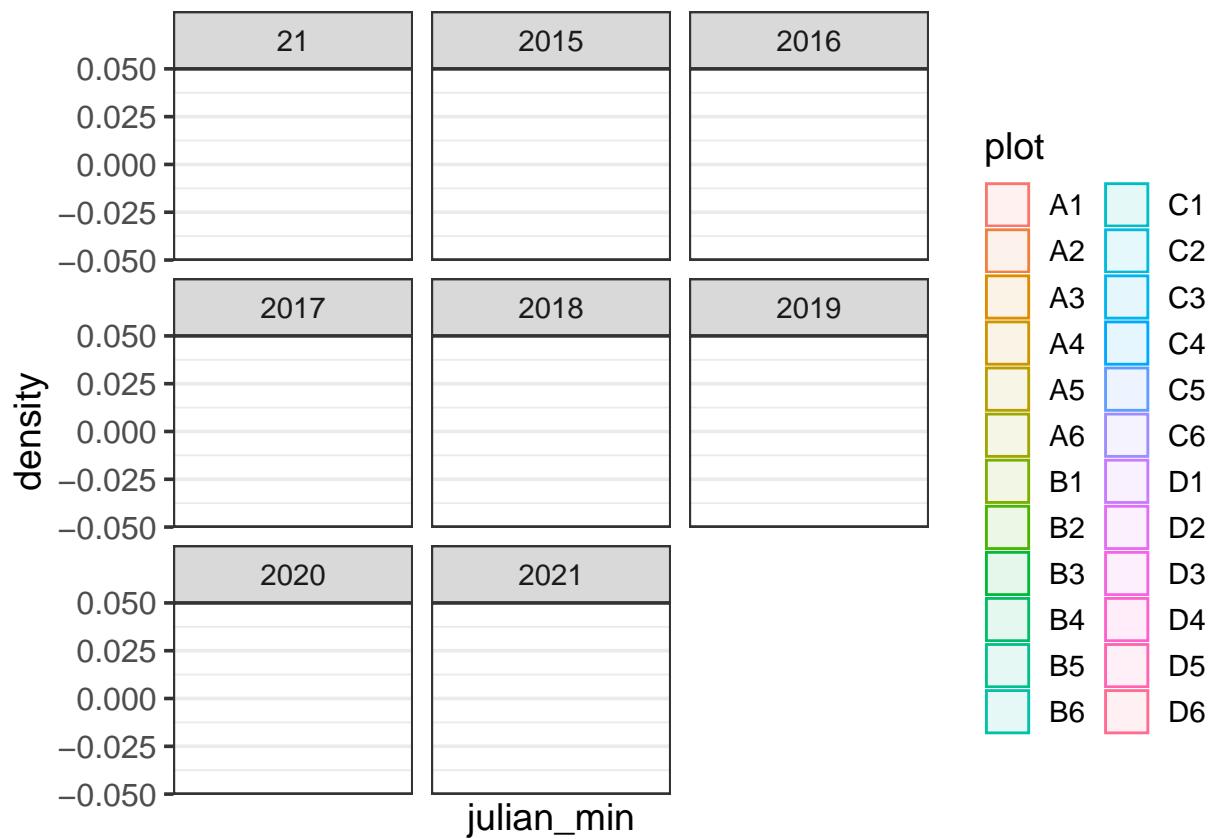
```



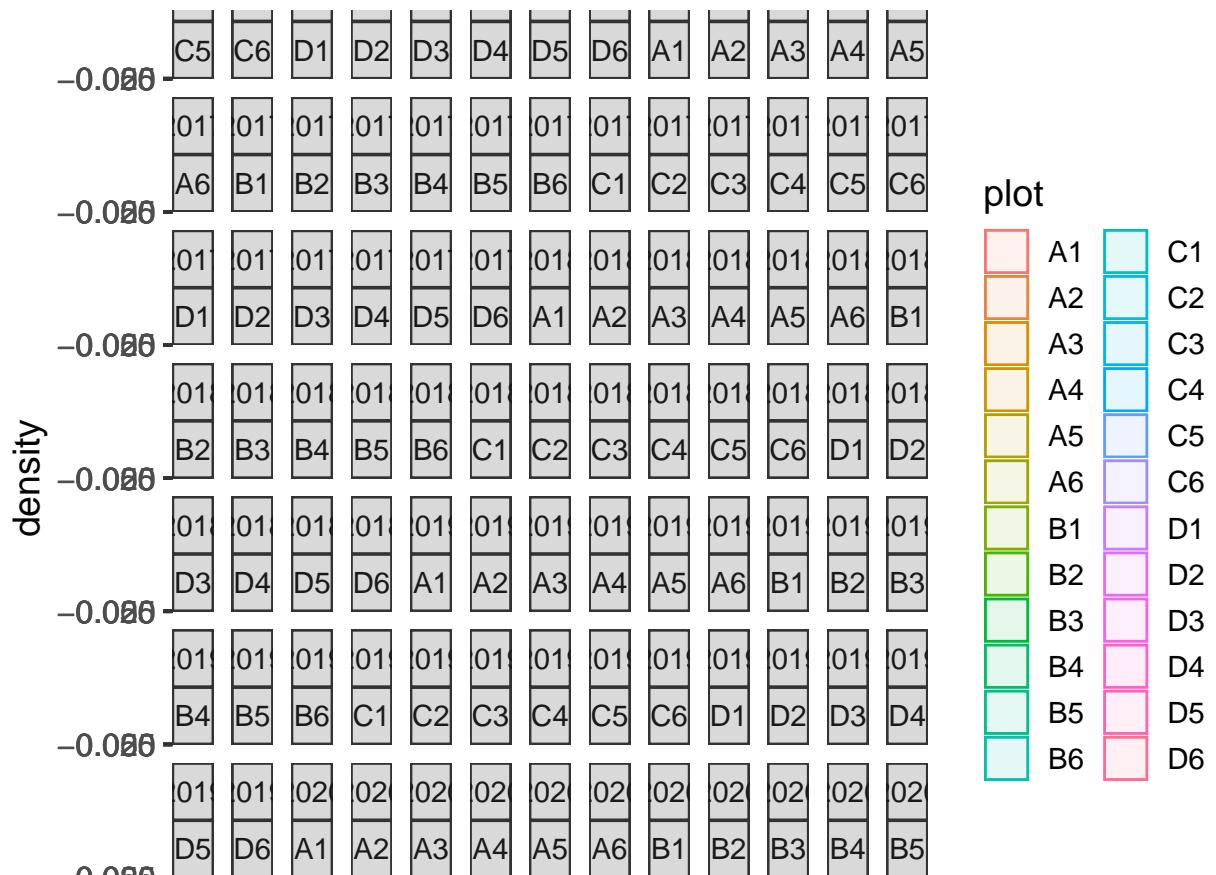
```
ggplot(kbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

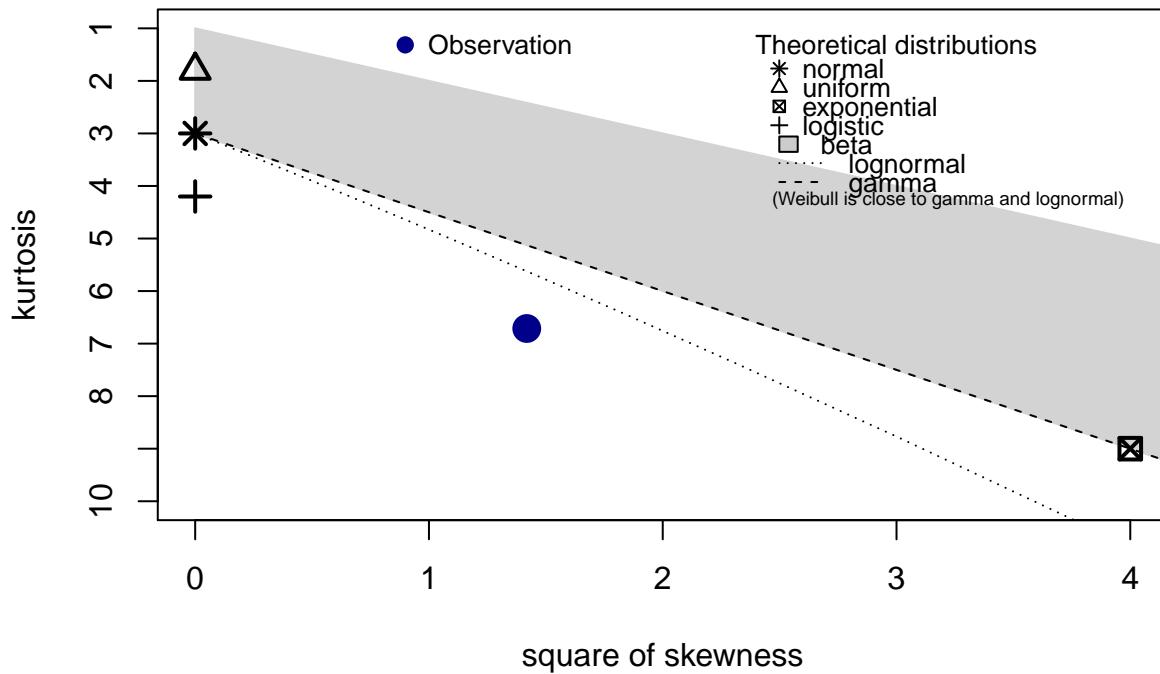


```
ggplot(kbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



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

## Cullen and Frey graph

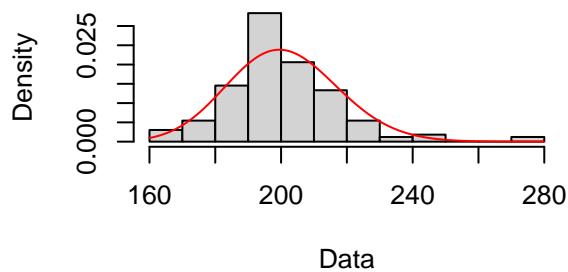
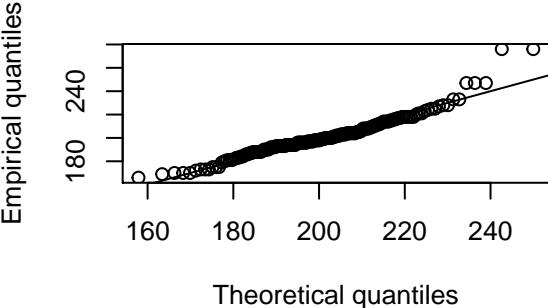
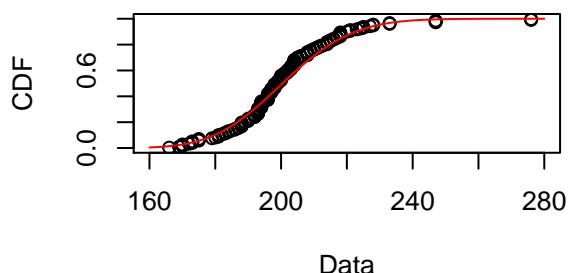
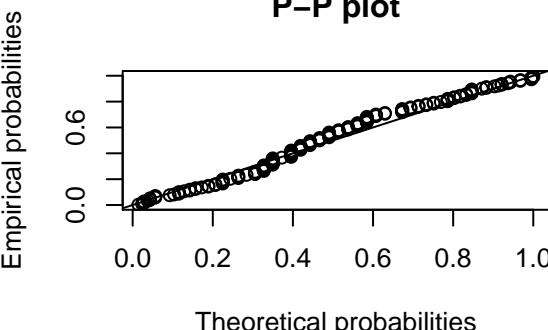


```

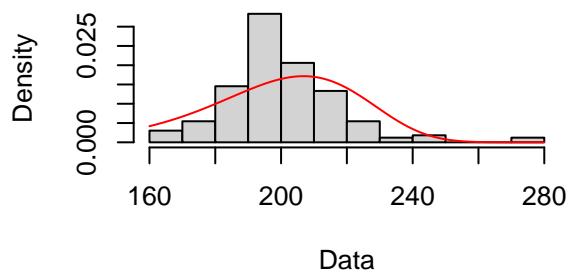
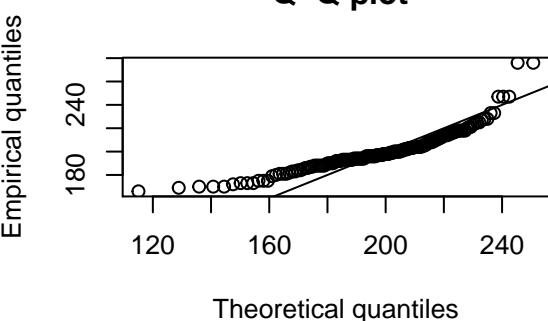
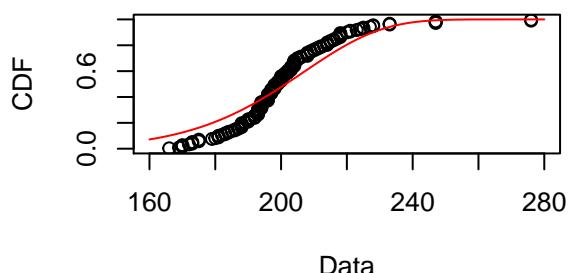
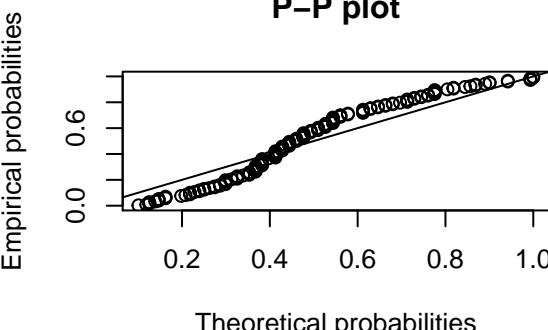
## summary statistics
## -----
## min: 166   max: 276
## median: 199
## mean: 200.8727
## estimated sd: 17.24966
## estimated skewness: 1.190949
## estimated kurtosis: 6.712314

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

```

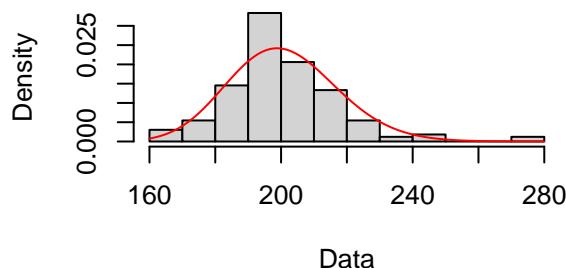
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

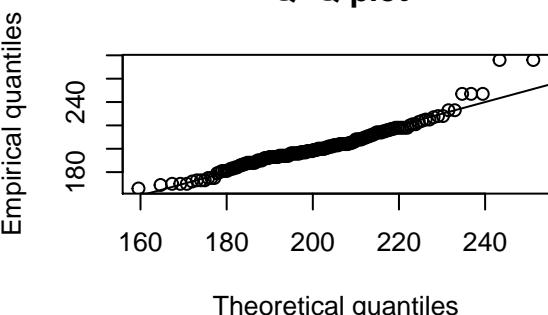
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

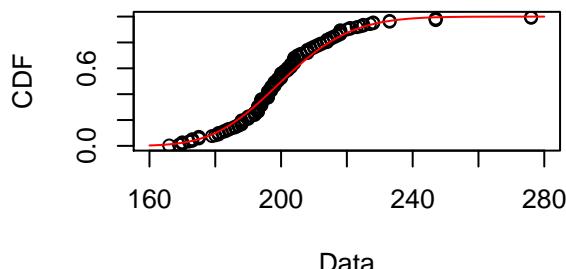
**Empirical and theoretical dens.**



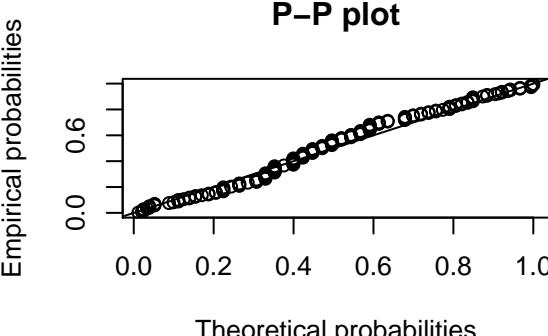
**Q–Q plot**



**Empirical and theoretical CDFs**

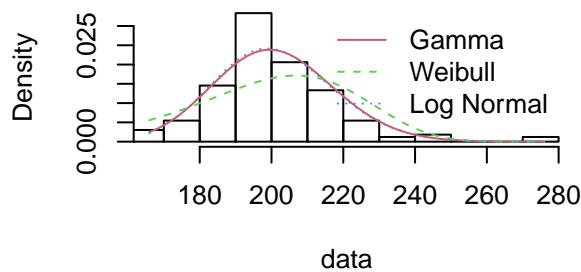


**P–P plot**

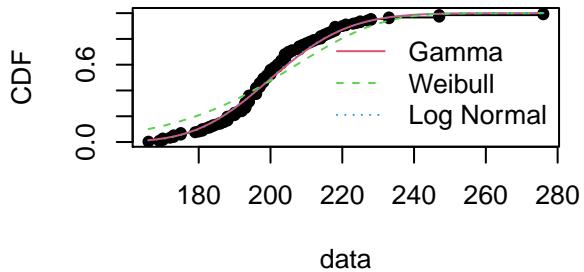


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

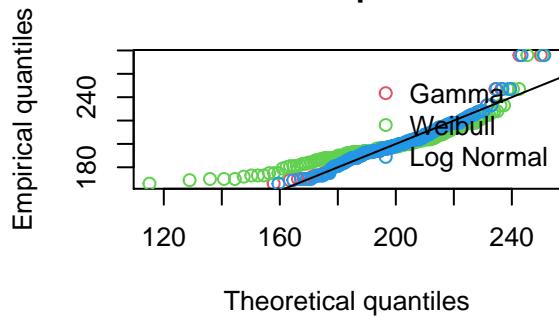
### Histogram and theoretical densities



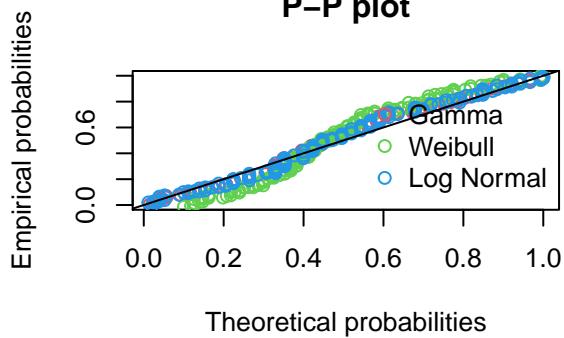
### Empirical and theoretical CDFs



### Q-Q plot



### P-P plot



```

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

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

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

```

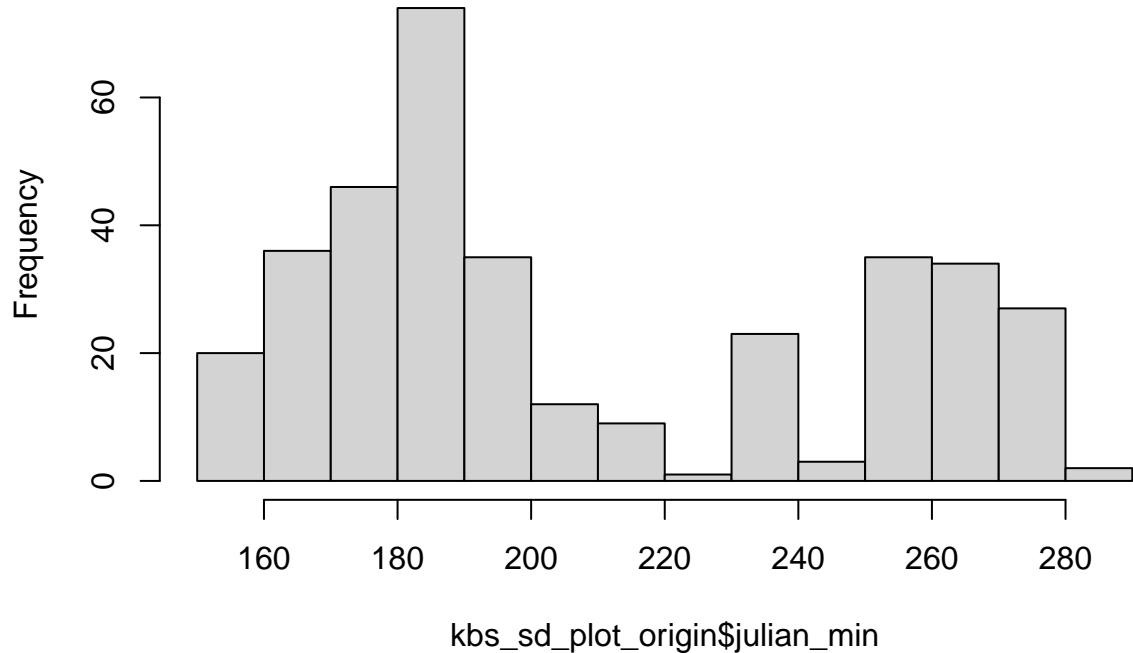
## KBS PLOT LEVEL ORIGIN - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

```

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

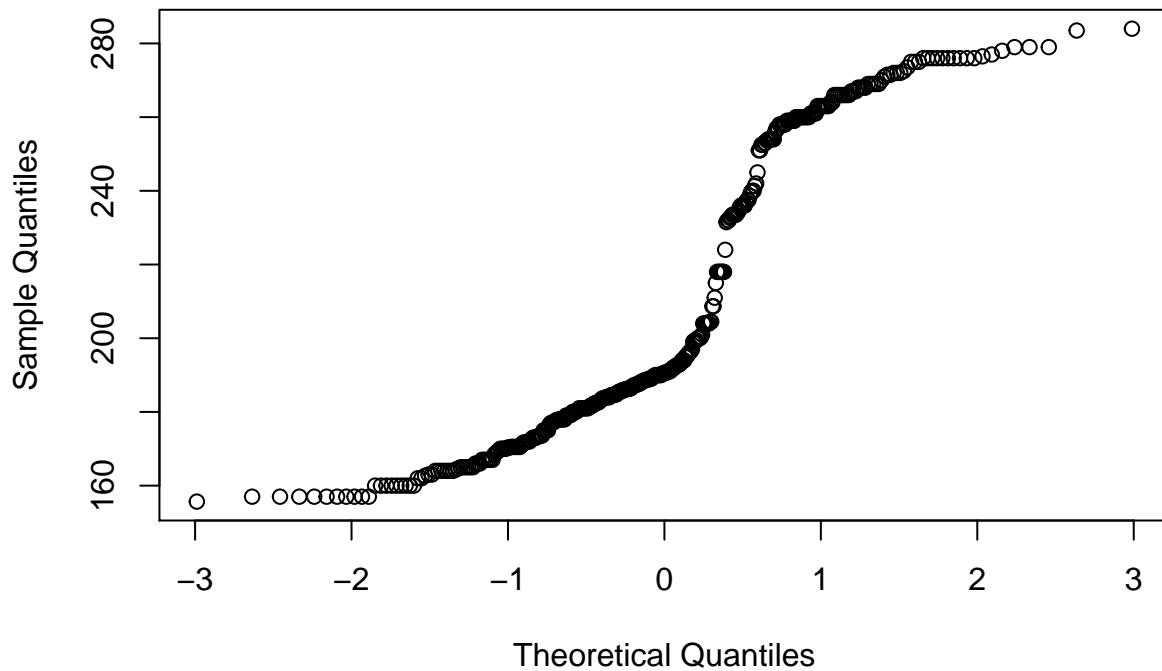
```

### Histogram of kbs\_sd\_plot\_origin\$julian\_min



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

### Normal Q-Q Plot



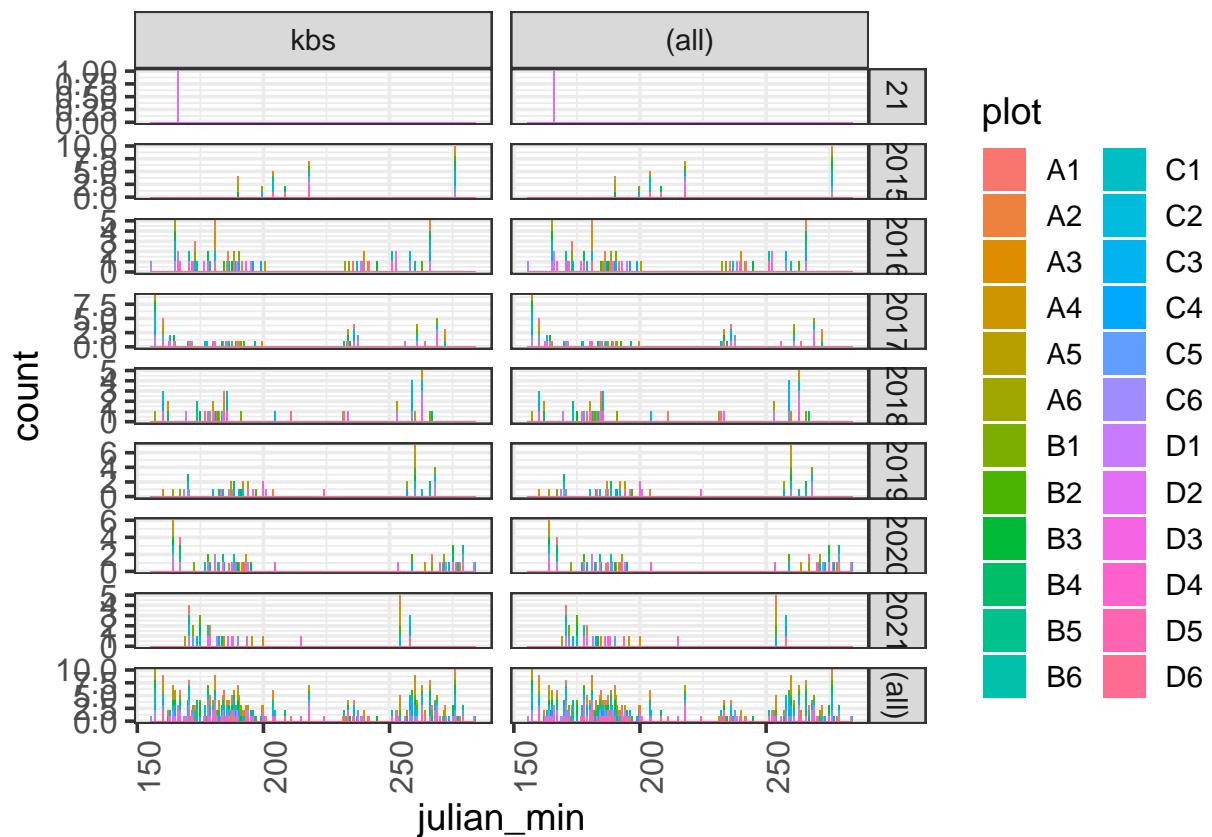
```

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 = 0.87068, 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")

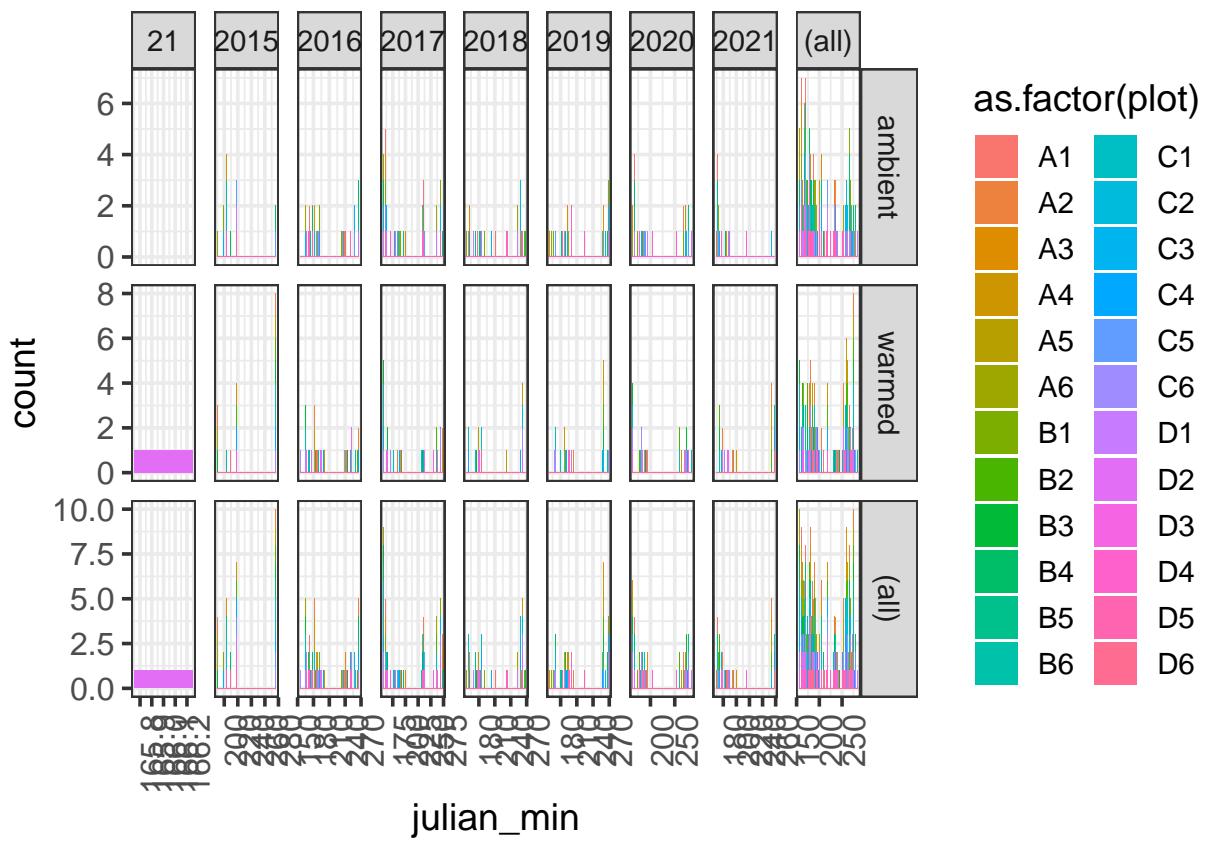
```



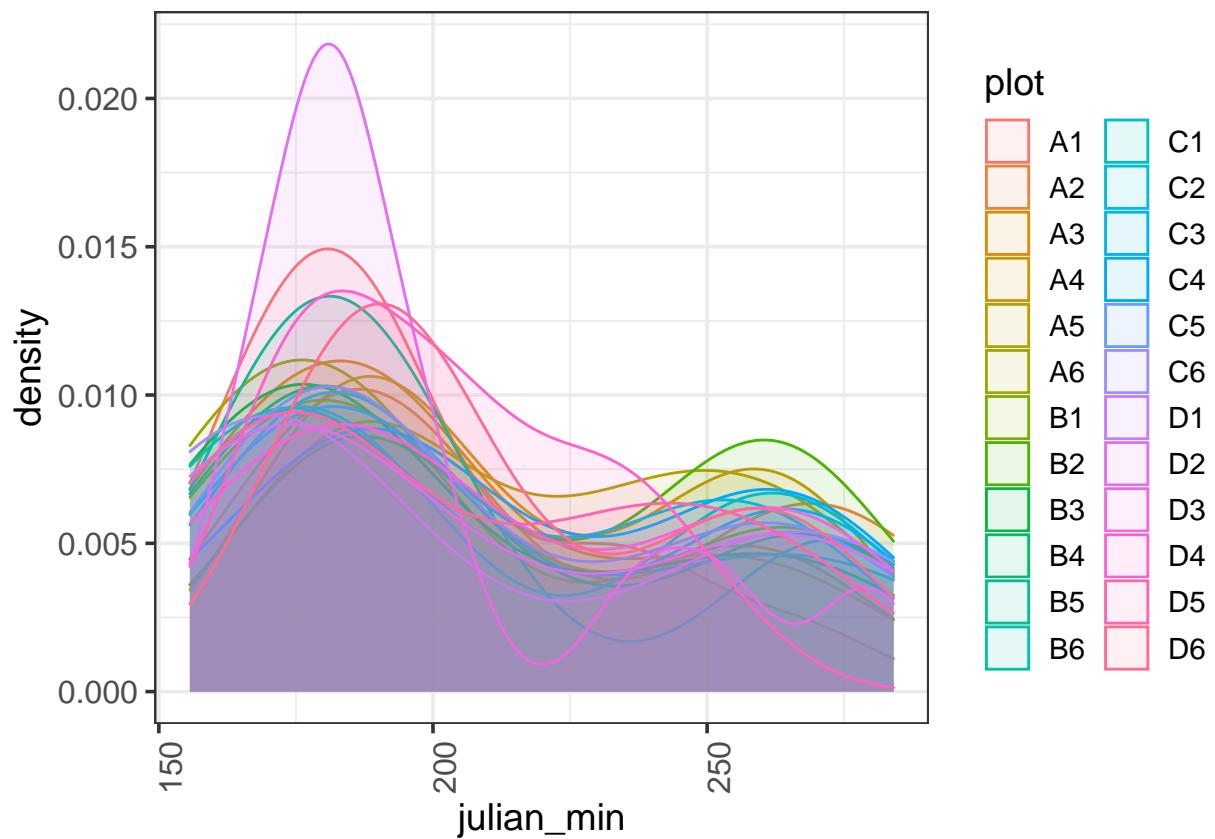
```

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")

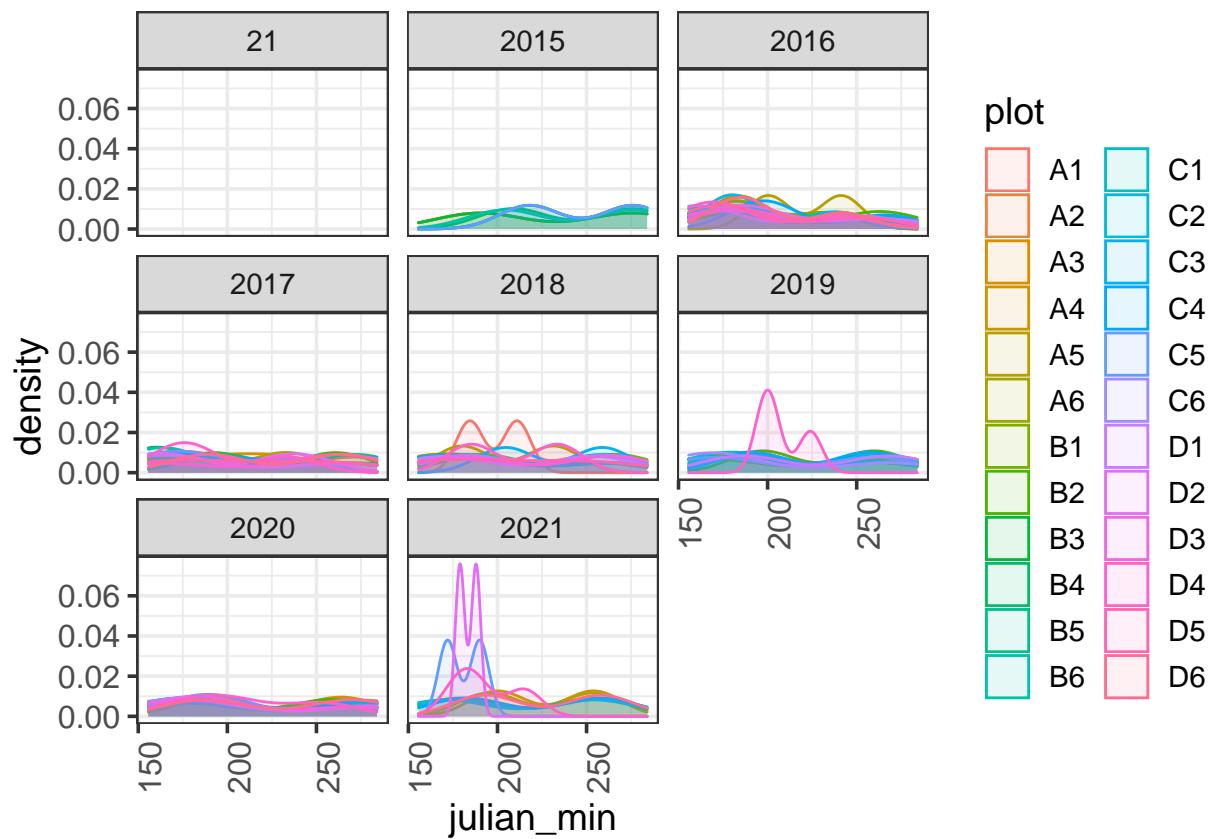
```



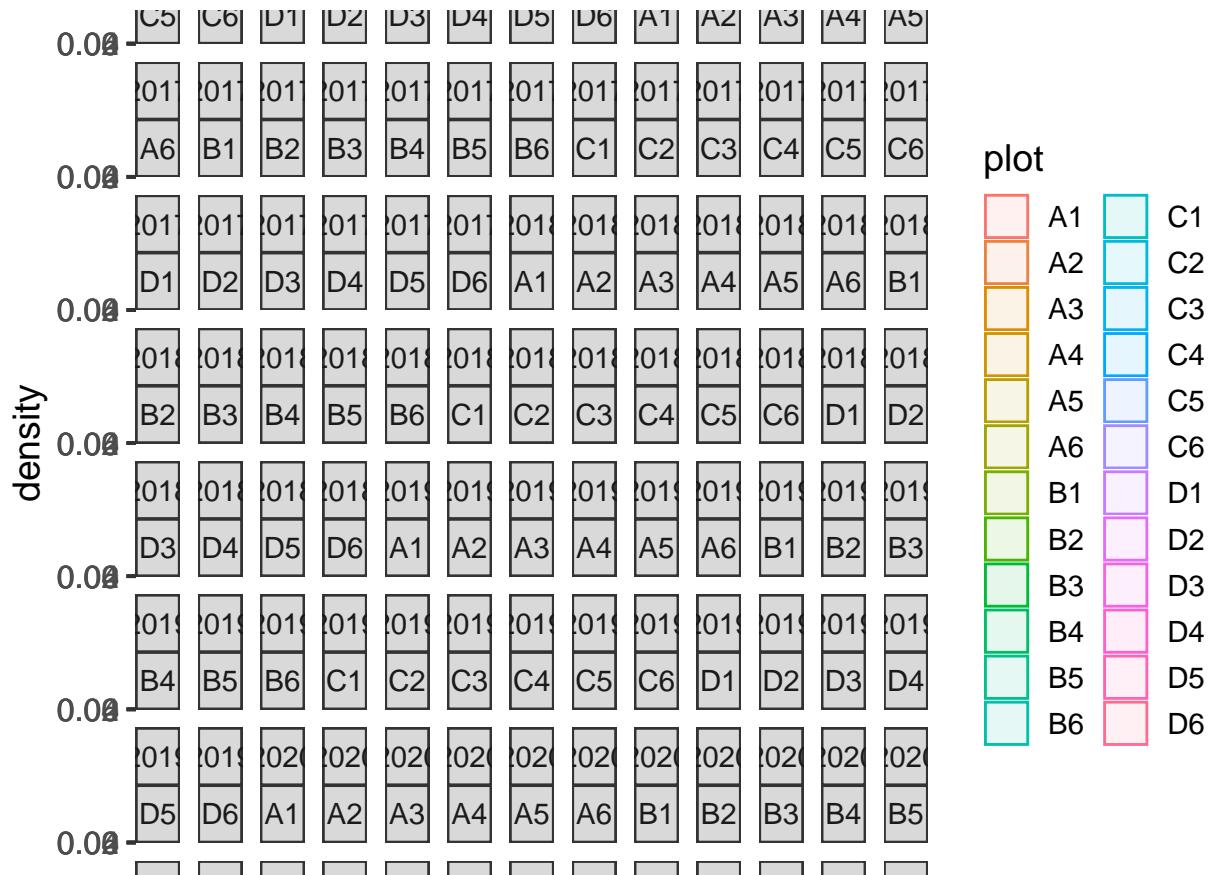
```
ggplot(kbs_sd_plot_origin, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

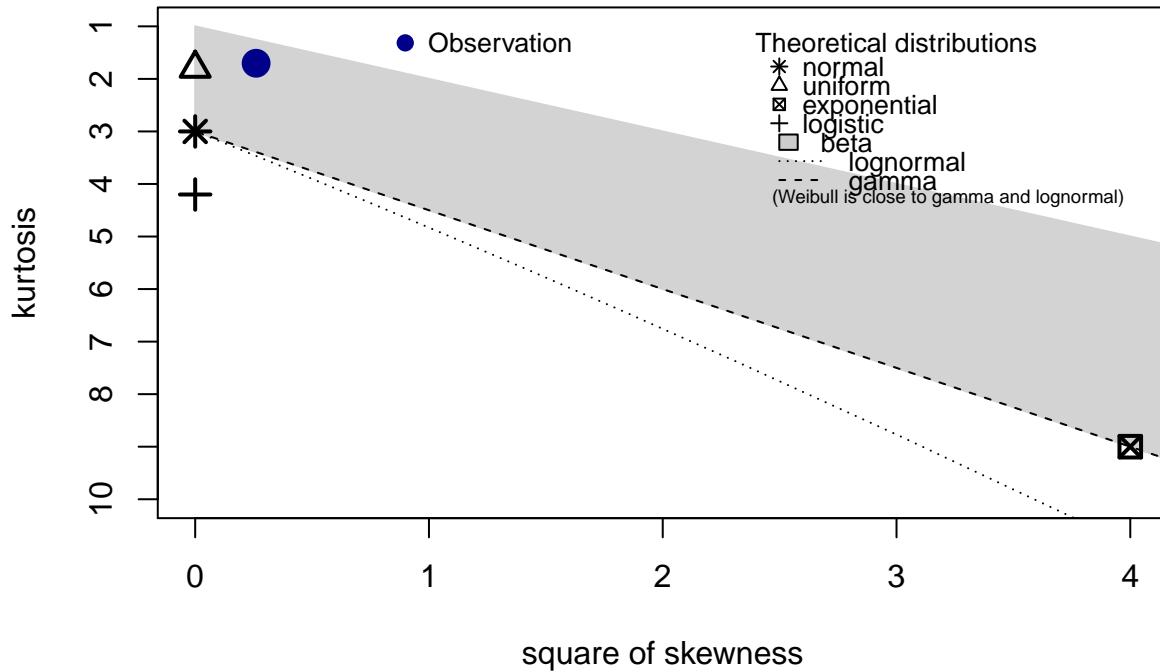


```
ggplot(kbs_sd_plot_origin, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



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

## Cullen and Frey graph

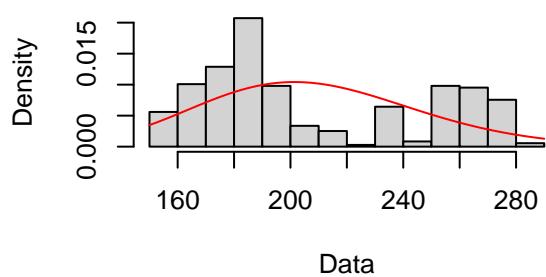
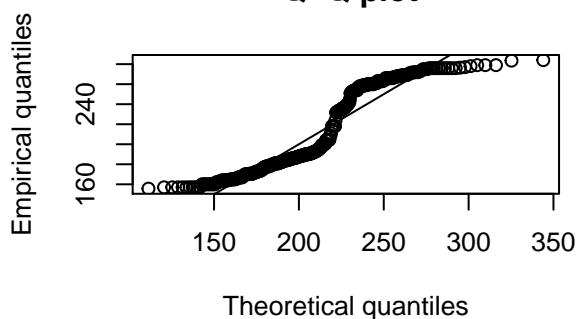
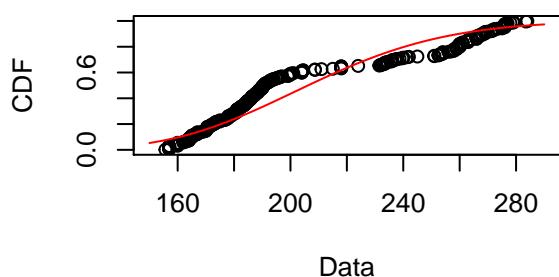
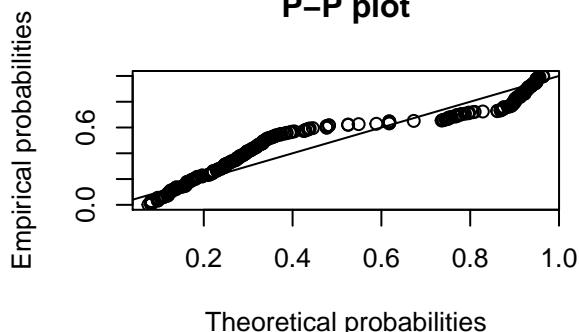


```

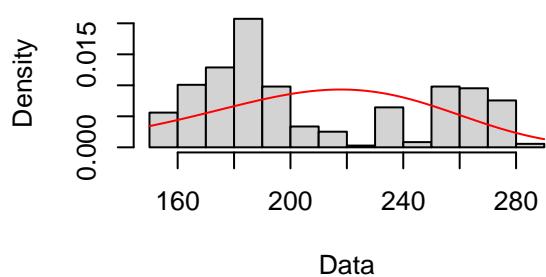
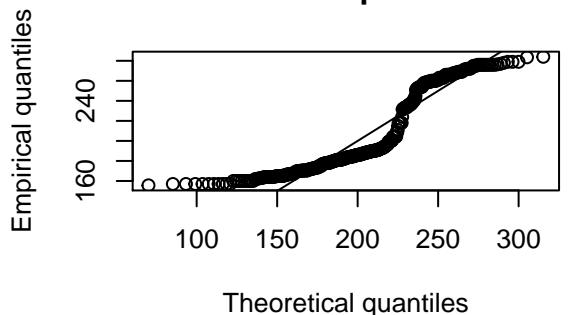
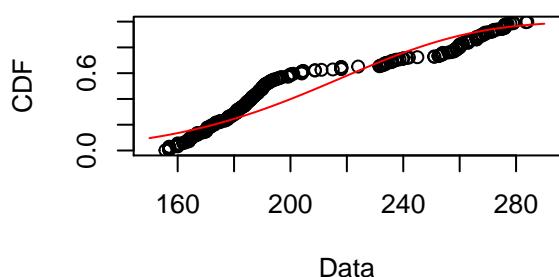
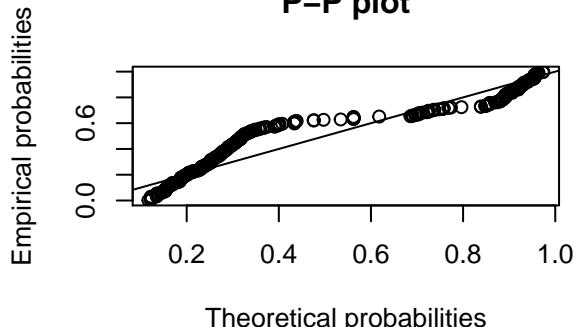
## summary statistics
## -----
## min: 155.6667  max: 284
## median: 190.5
## mean: 208.6387
## estimated sd: 39.75324
## estimated skewness: 0.5105558
## estimated kurtosis: 1.702394

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

```

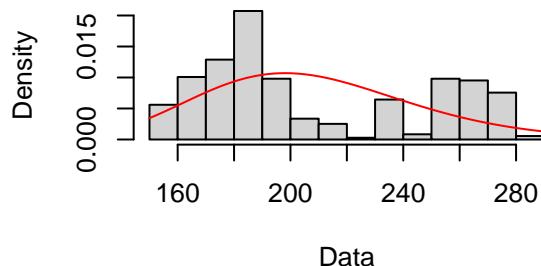
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

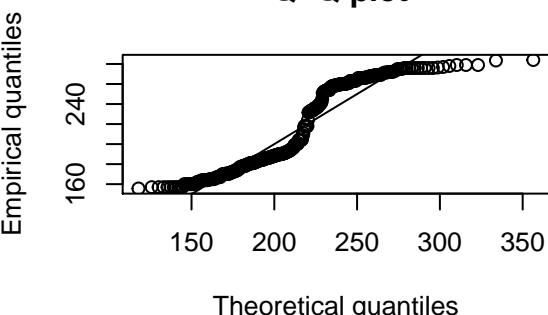
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

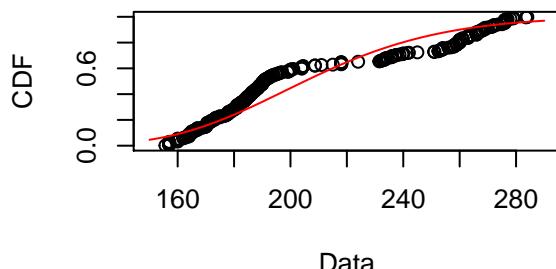
**Empirical and theoretical dens.**



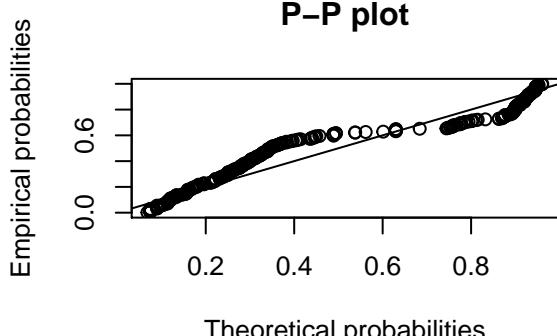
**Q–Q plot**



**Empirical and theoretical CDFs**

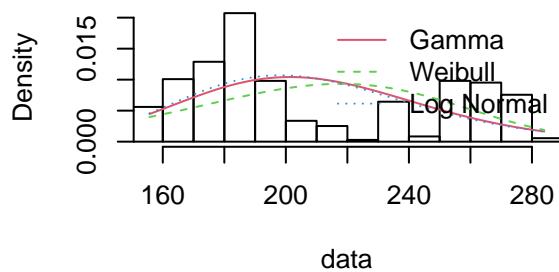


**P–P plot**

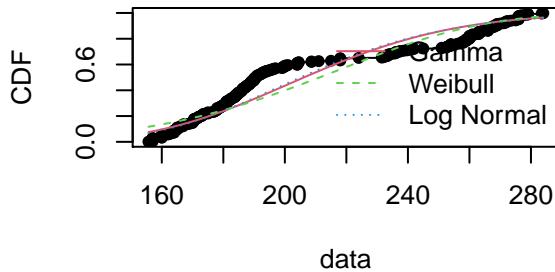


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

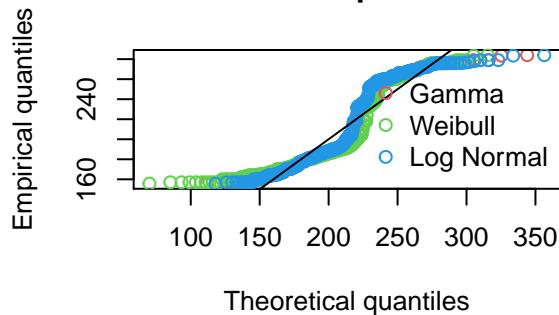
### Histogram and theoretical densities



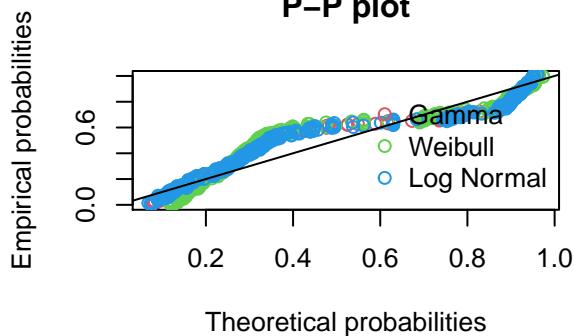
### Empirical and theoretical CDFs



### 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"))
```

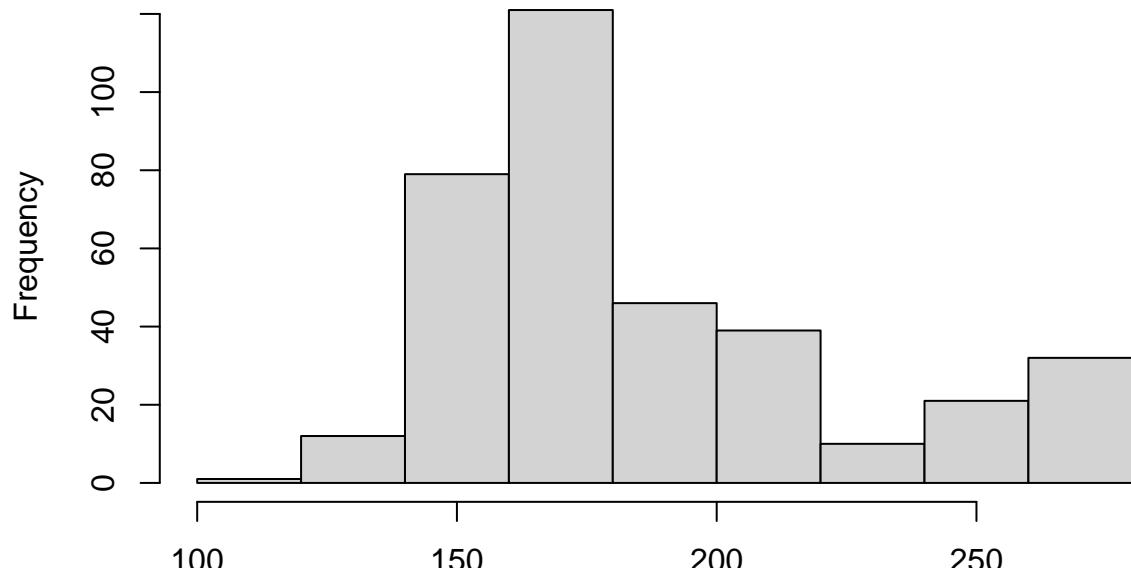
```
## Goodness-of-fit statistics
##                                     Gamma  Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.1787132 0.20307 0.1693461
## Cramer-von Mises statistic   2.8784699 3.46052 2.6551340
## Anderson-Darling statistic  16.4100954 19.17250 15.2972231
##
## Goodness-of-fit criteria
##                                     Gamma  Weibull Log Normal
## Akaike's Information Criterion 3621.828 3668.628 3613.018
## Bayesian Information Criterion 3629.584 3676.384 3620.774
```

*# 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)
```

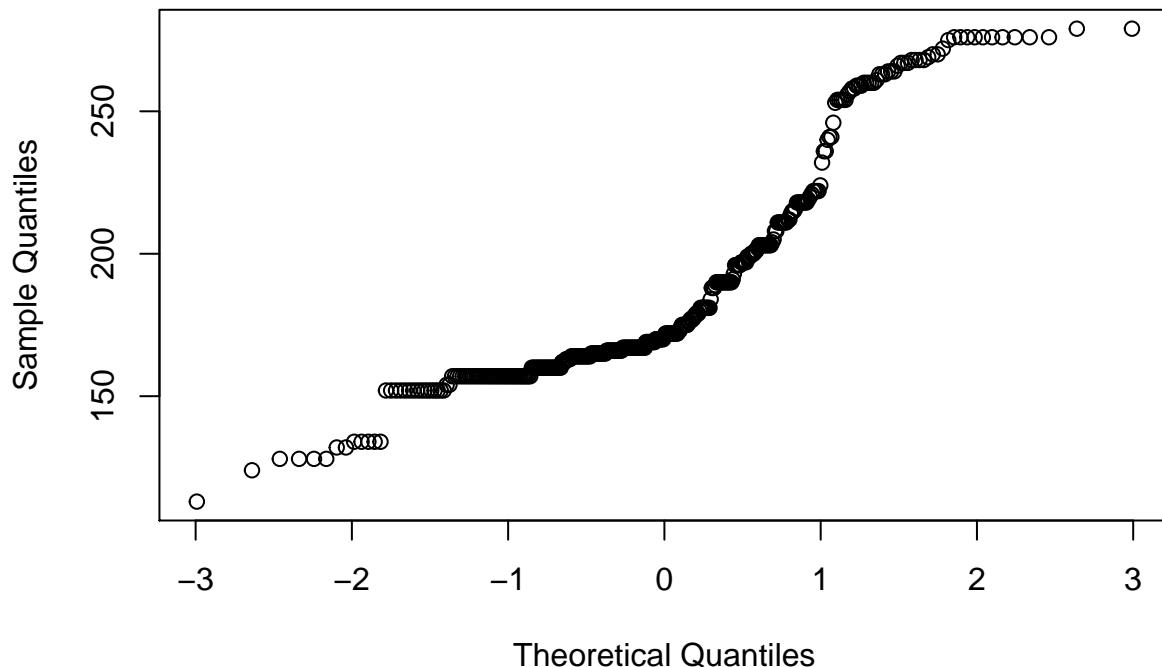
### Histogram of kbs\_sd\_plot\_growthhabit\$julian\_min



`kbs_sd_plot_growthhabit$julian_min`

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

### Normal Q-Q Plot



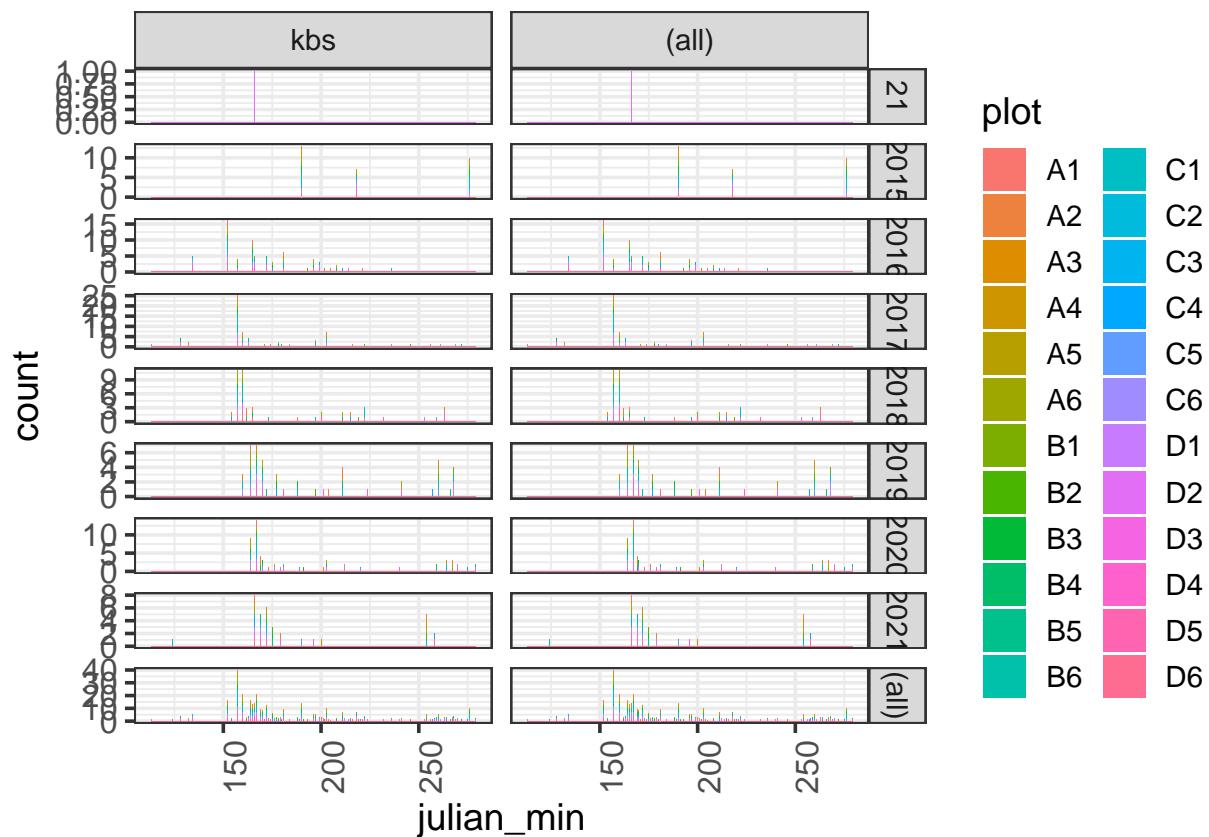
```

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

## 
## Shapiro-Wilk normality test
## 
## data: kbs_sd_plot_growthhabit$julian_min
## W = 0.86086, p-value < 2.2e-16

# 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")

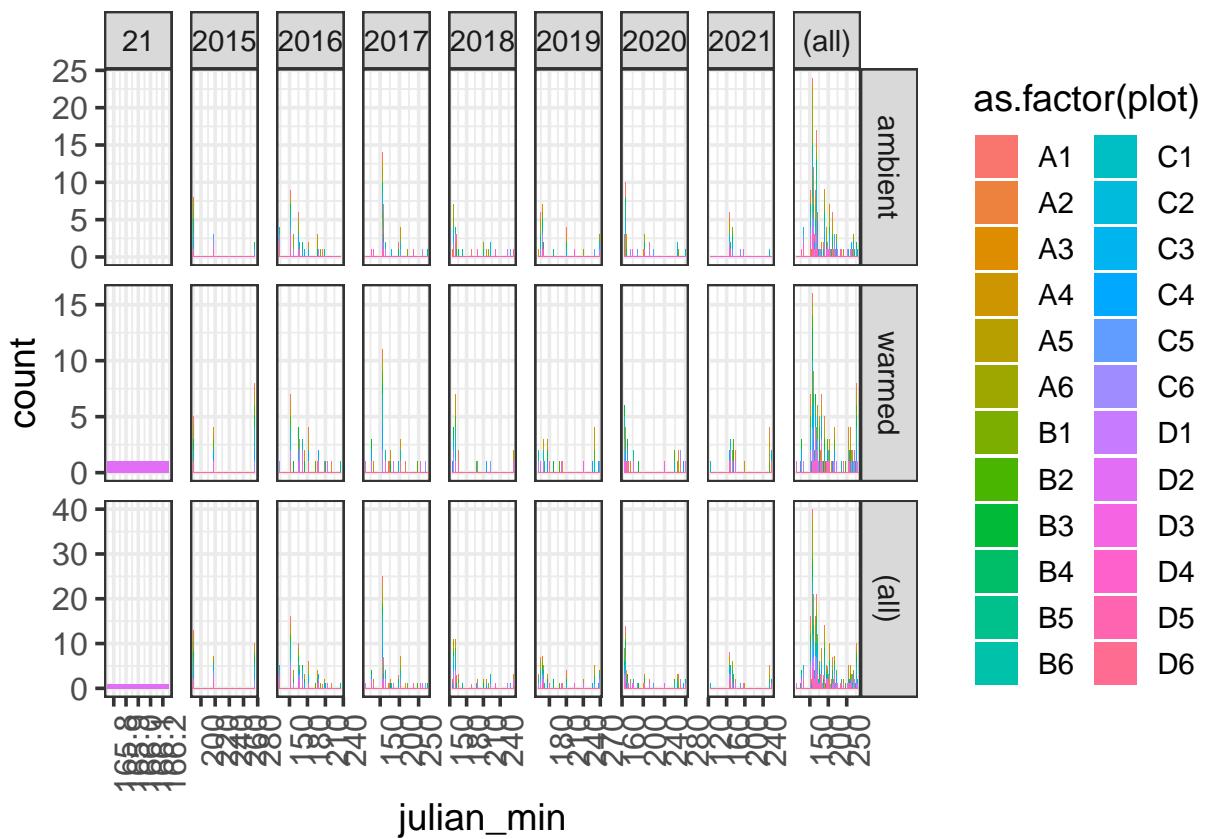
```



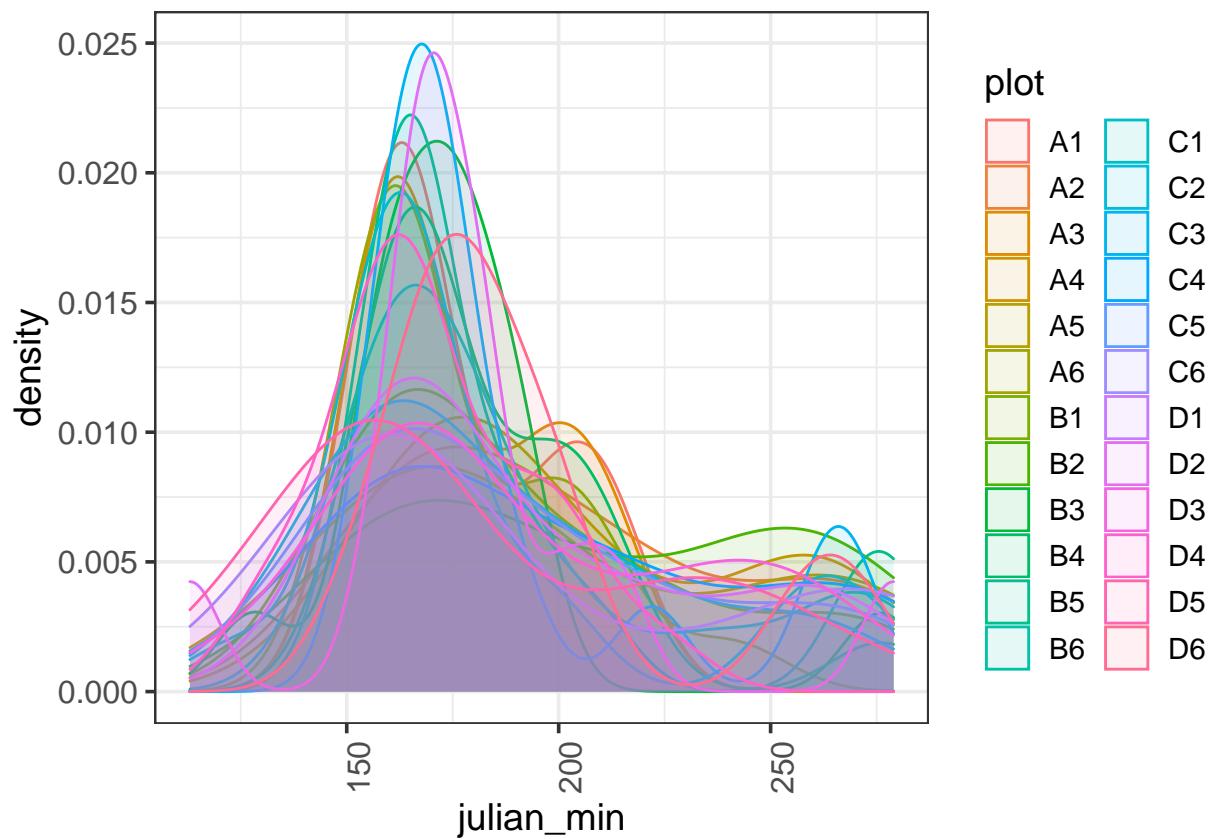
```

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")

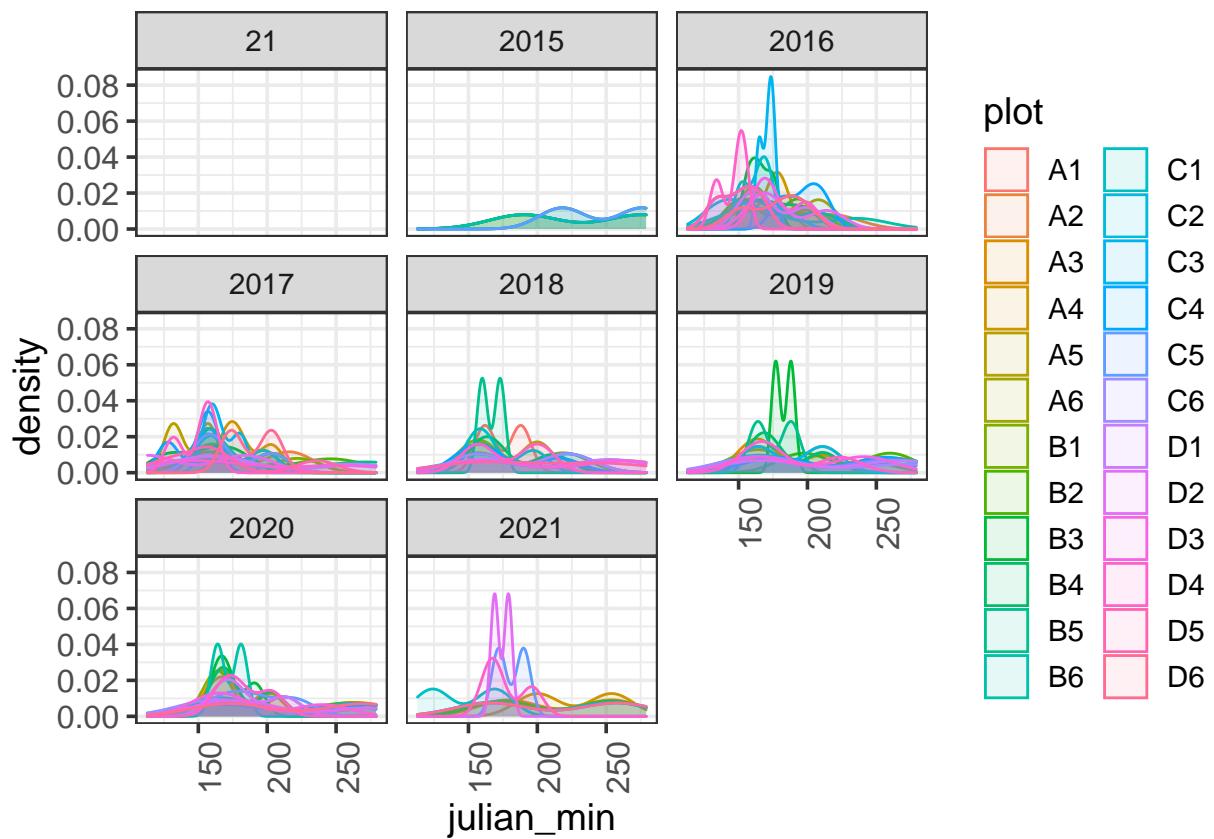
```



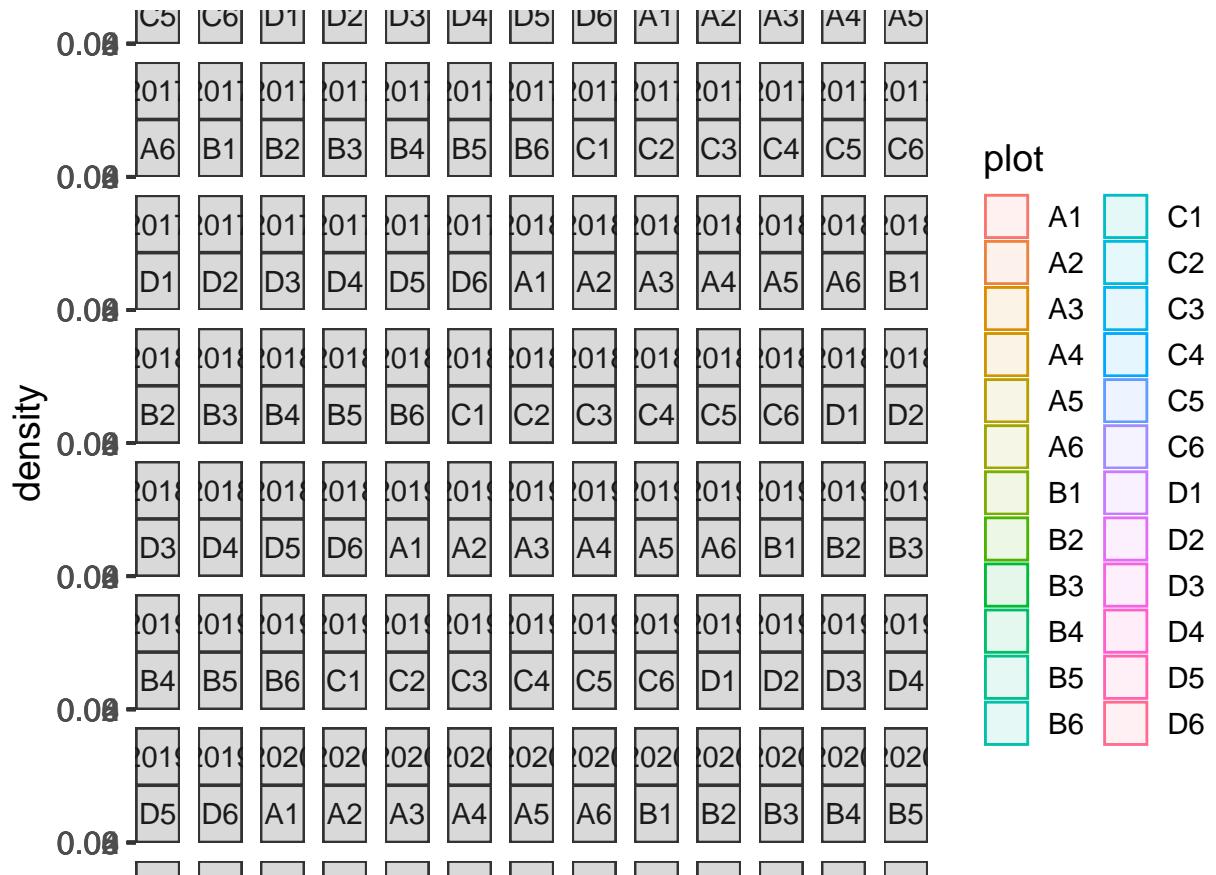
```
ggplot(kbs_sd_plot_growthhabit, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

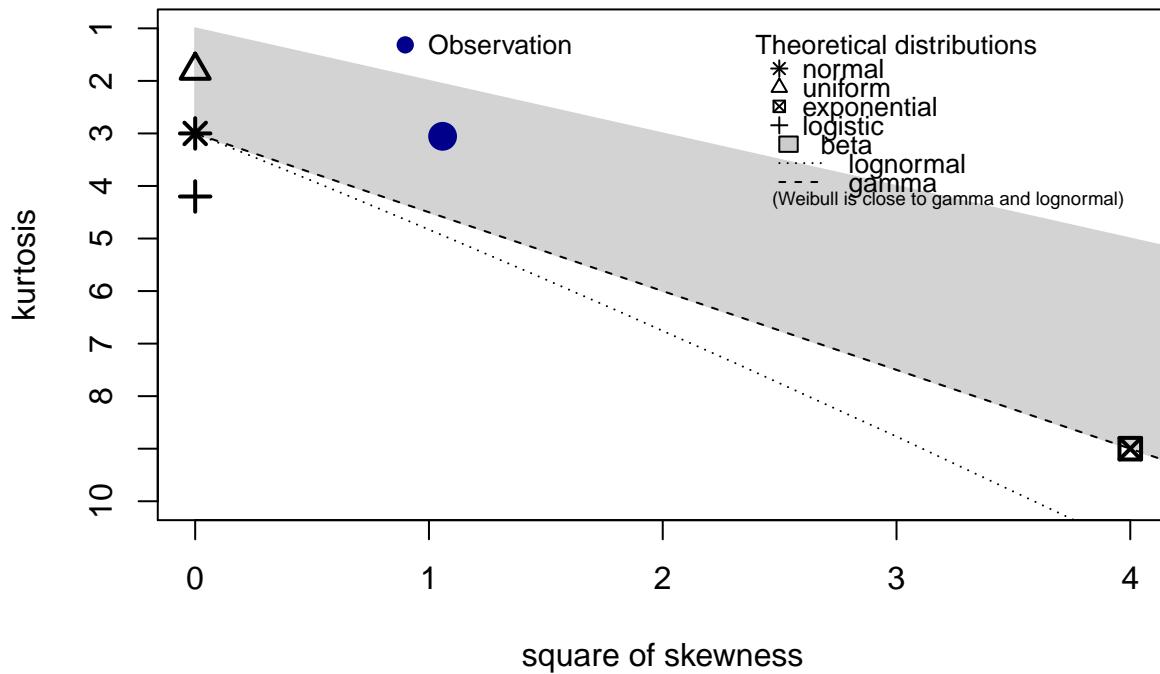


```
ggplot(kbs_sd_plot_growthhabit, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
  facet_wrap(~year + plot)
```



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

## Cullen and Frey graph

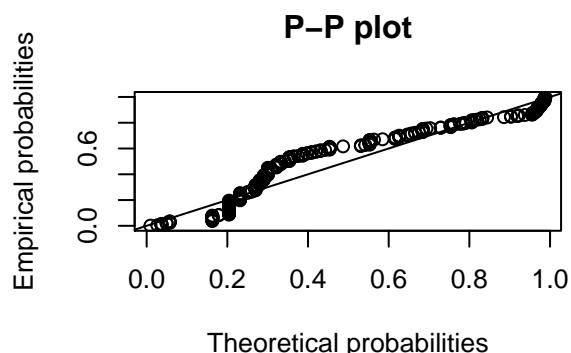
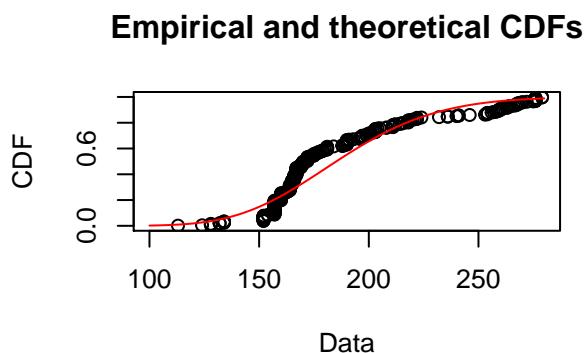
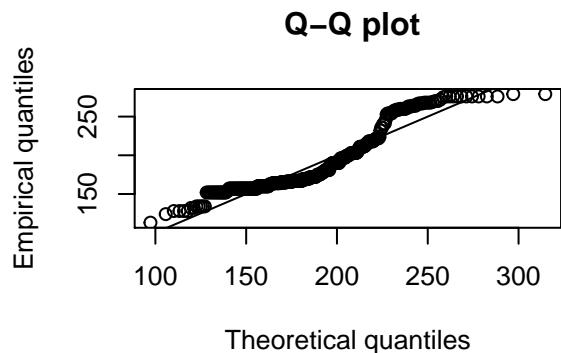
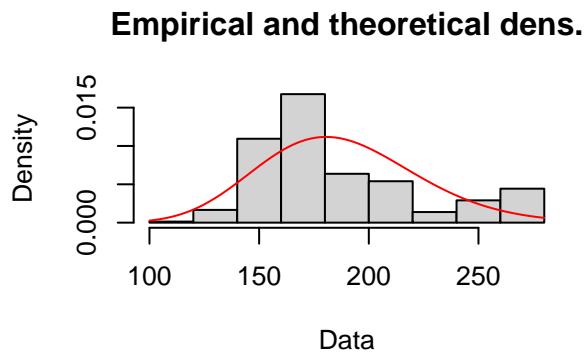


```

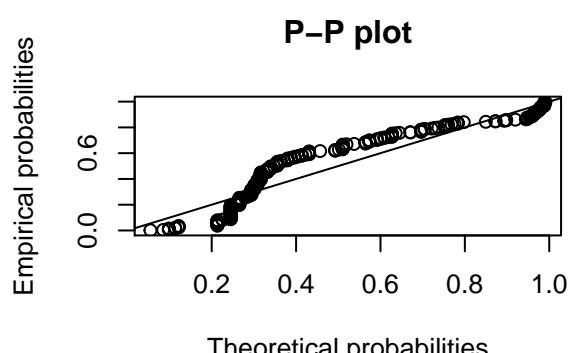
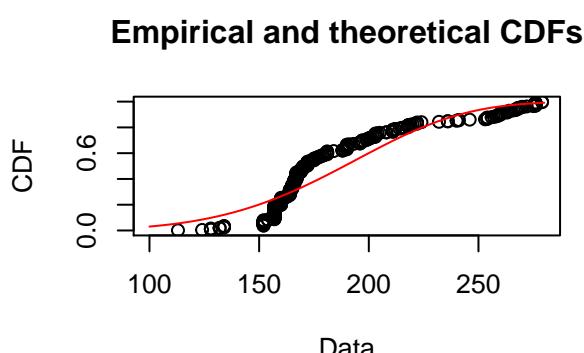
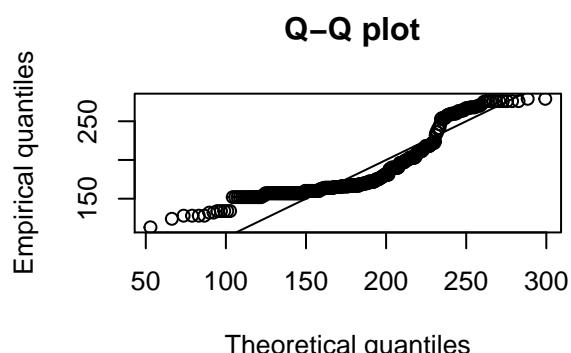
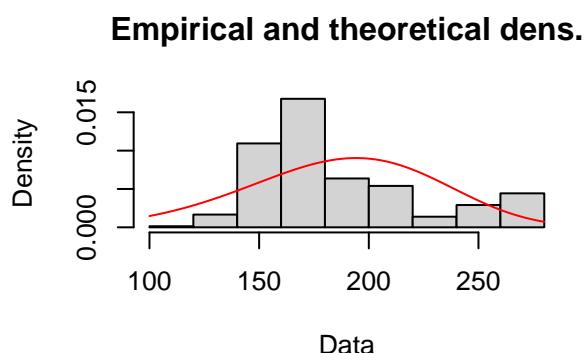
## summary statistics
## -----
## min: 113  max: 279
## median: 171
## mean: 187.5042
## estimated sd: 38.14505
## estimated skewness: 1.028789
## estimated kurtosis: 3.054855

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

```

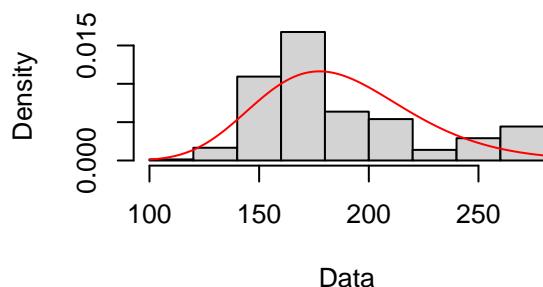


```
# Weibull distribution
fit.weibull <- fitdist(kbs_sd_plot_growthhabit$julian_min, "weibull")
plot(fit.weibull)
```

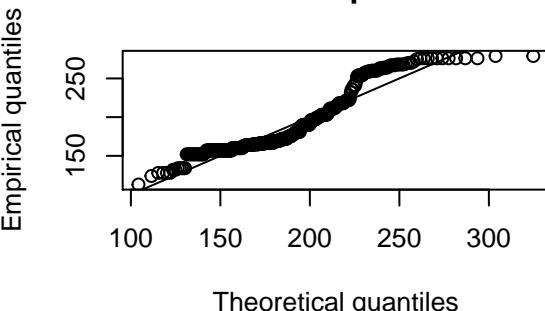


```
# Lognormal distribution
fit.ln <- fitdist(kbs_sd_plot_growthhabit$julian_min, "lnorm")
plot(fit.ln)
```

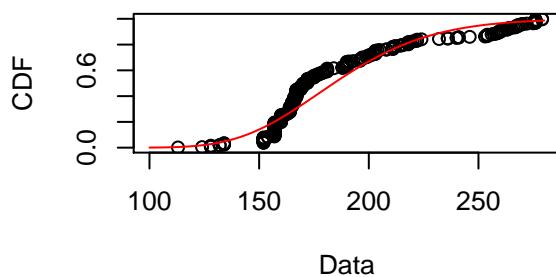
**Empirical and theoretical dens.**



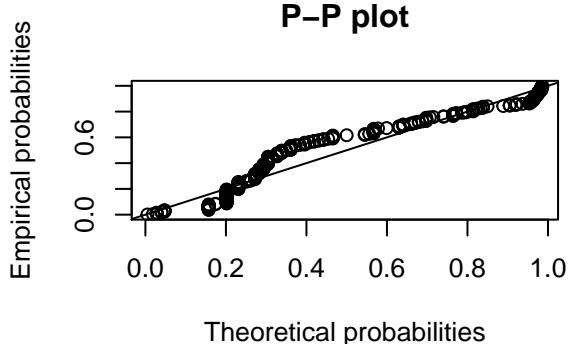
**Q-Q plot**



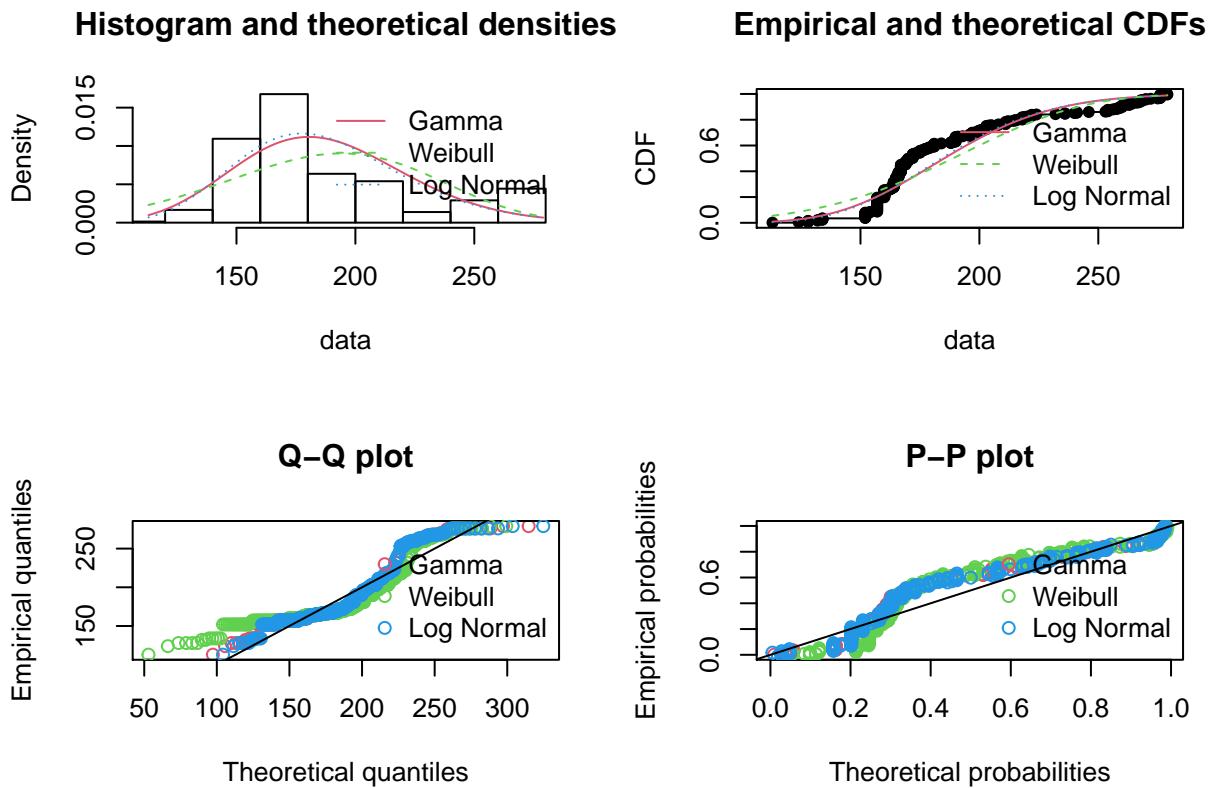
**Empirical and theoretical CDFs**



**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)
```



```
# 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"))

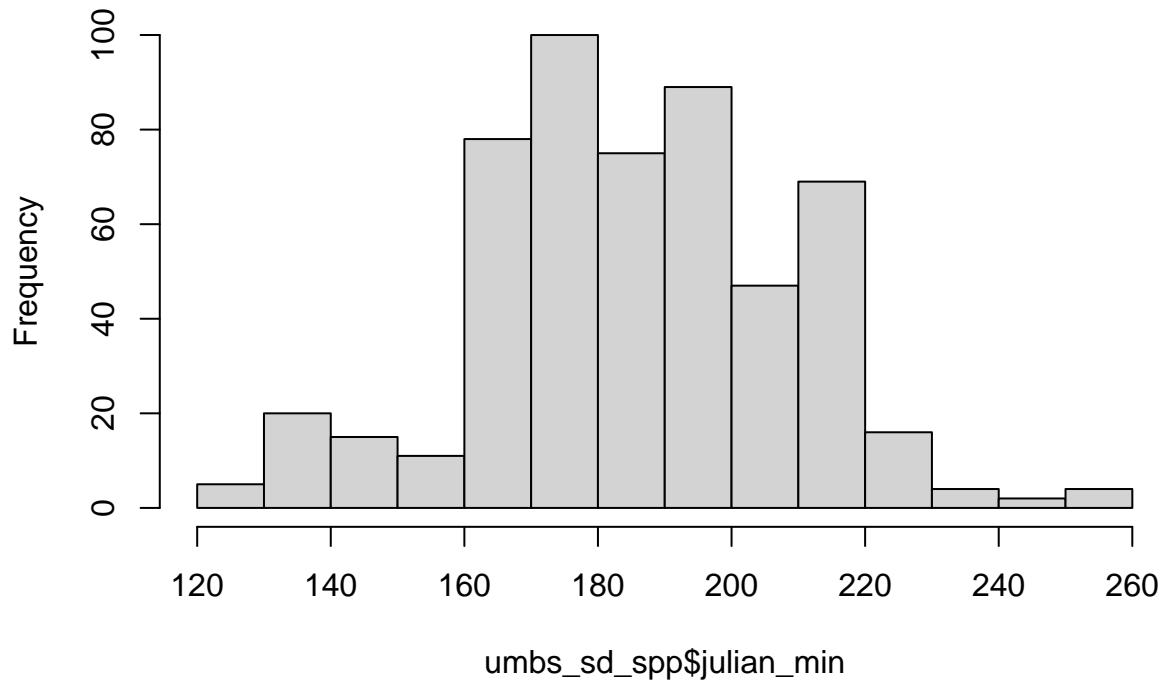
## Goodness-of-fit statistics
##                                     Gamma    Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.180911 0.1841854 0.1734767
## Cramer-von Mises statistic   2.814479 3.9664067 2.4803553
## Anderson-Darling statistic   16.284847 22.5165581 14.3842796
##
## Goodness-of-fit criteria
##                                     Gamma    Weibull Log Normal
## Akaike's Information Criterion 3611.901 3702.044 3594.292
## Bayesian Information Criterion 3619.678 3709.822 3602.070

# log normal looks like it's the best fit
```

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

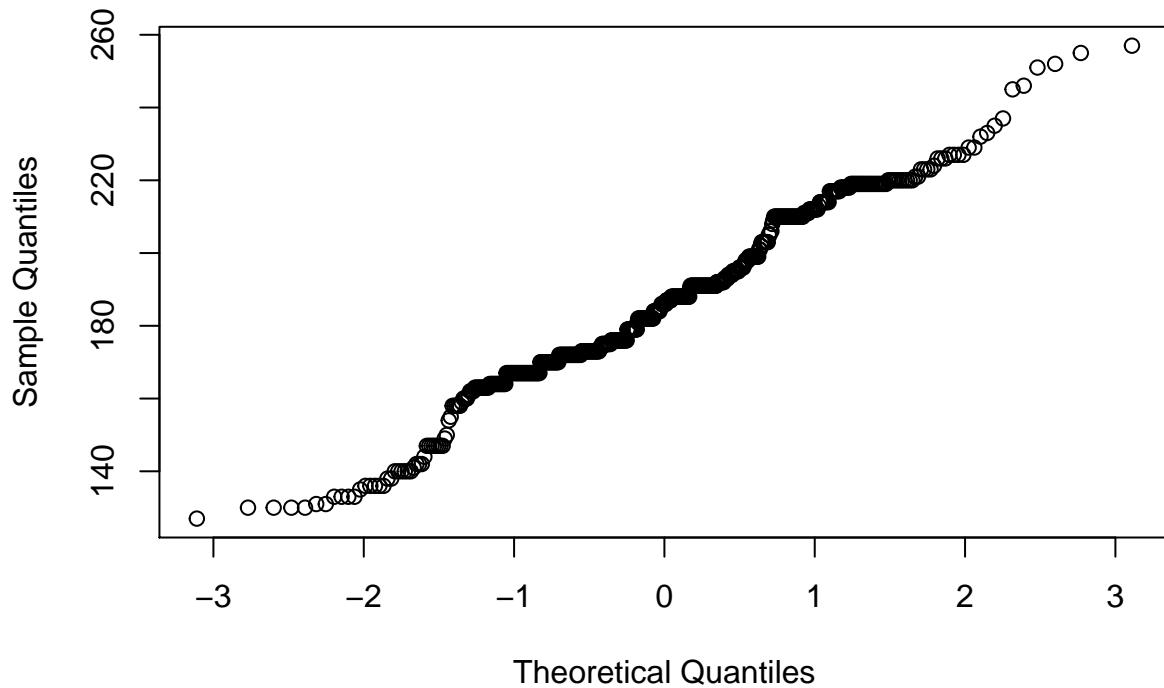
```
### UMBS ####
hist(umbs_sd_spp$julian_min)
```

### Histogram of umbs\_sd\_spp\$julian\_min



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

### Normal Q-Q Plot



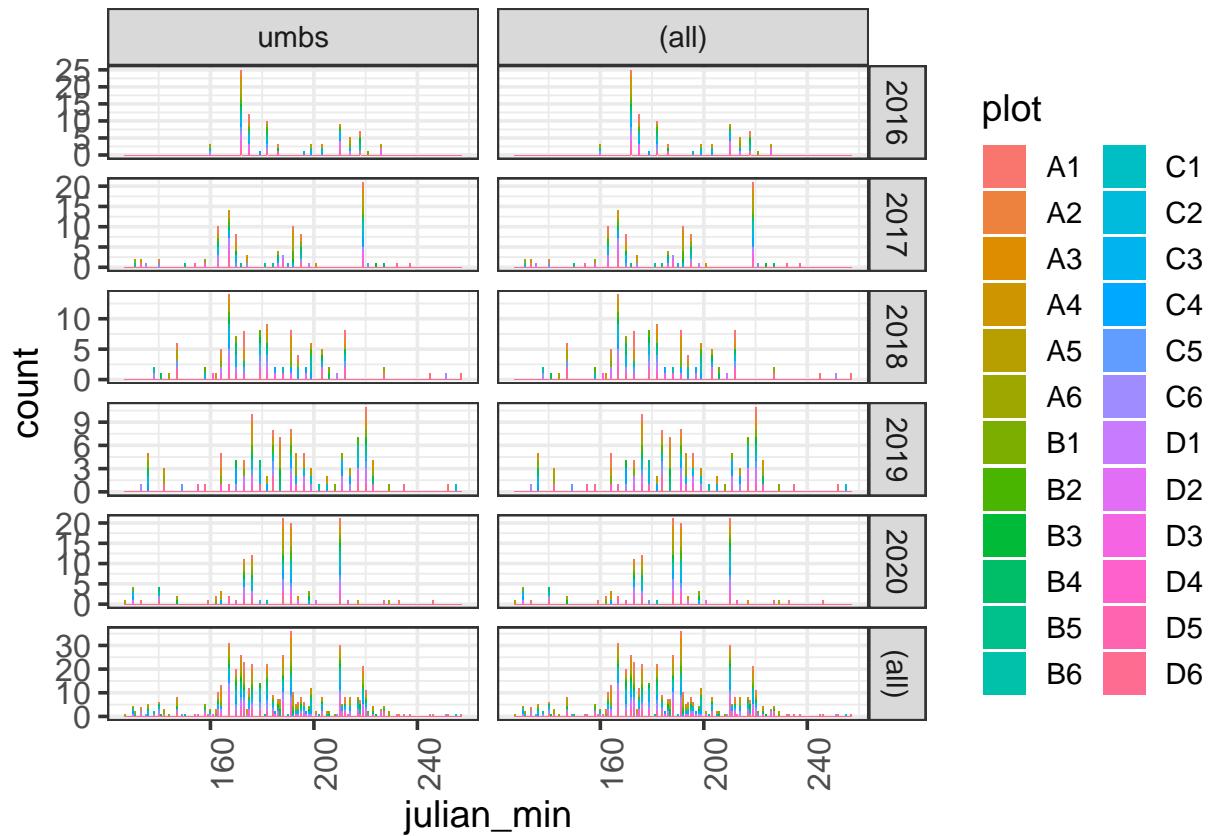
```

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

## 
## Shapiro-Wilk normality test
## 
## data: umbs_sd_spp$julian_min
## W = 0.98284, 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")

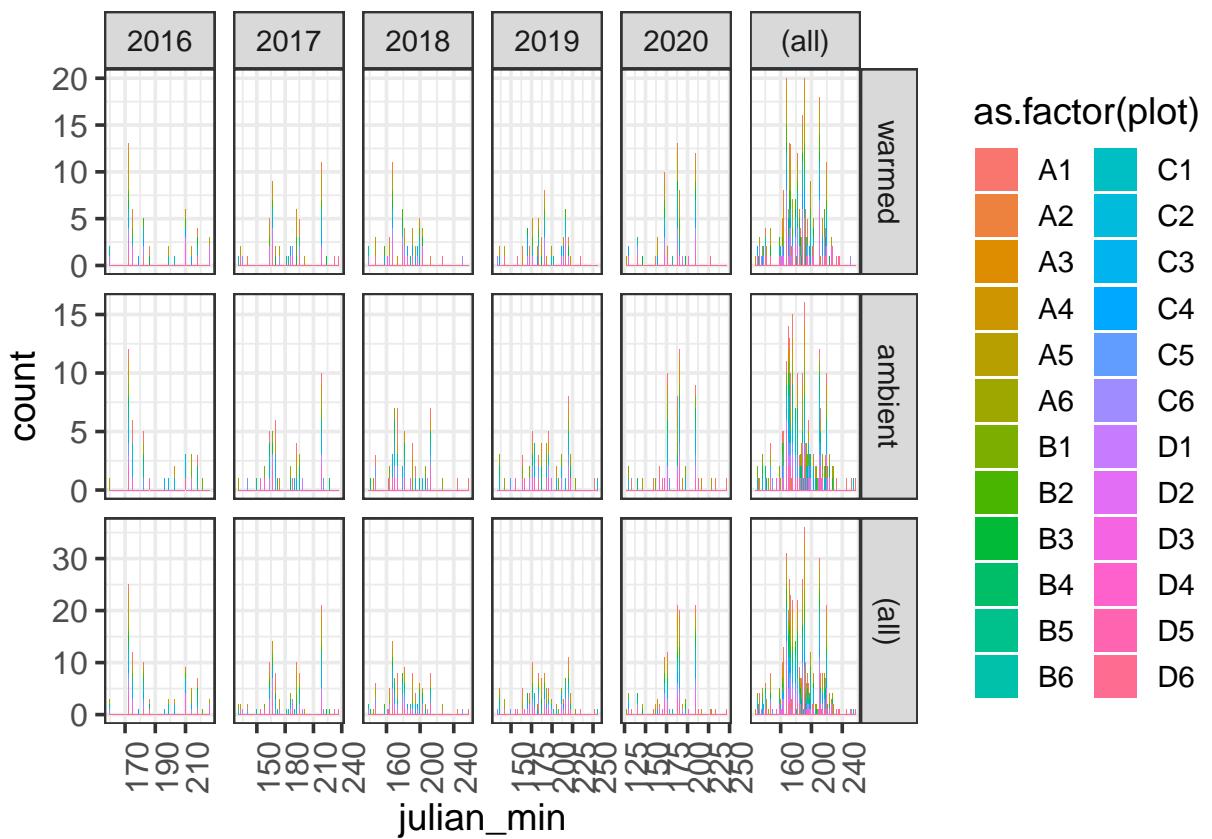
```



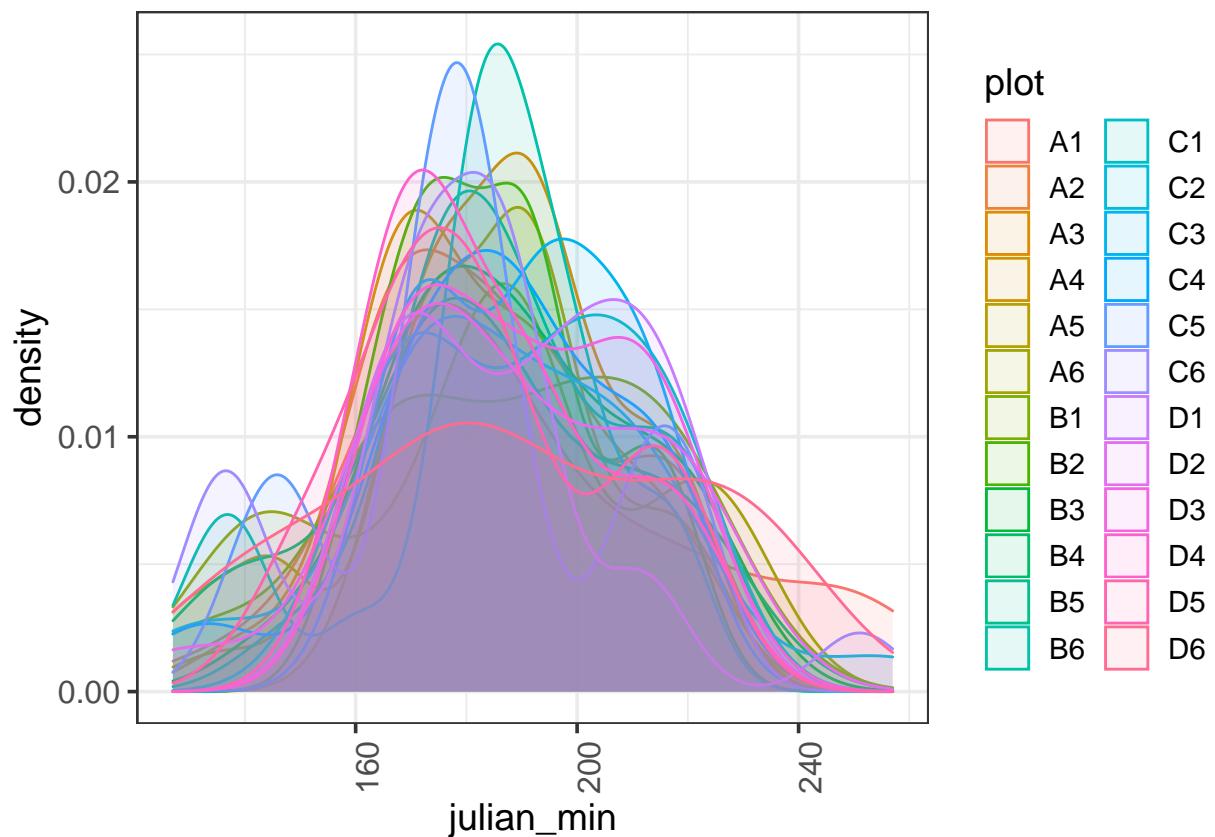
```

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

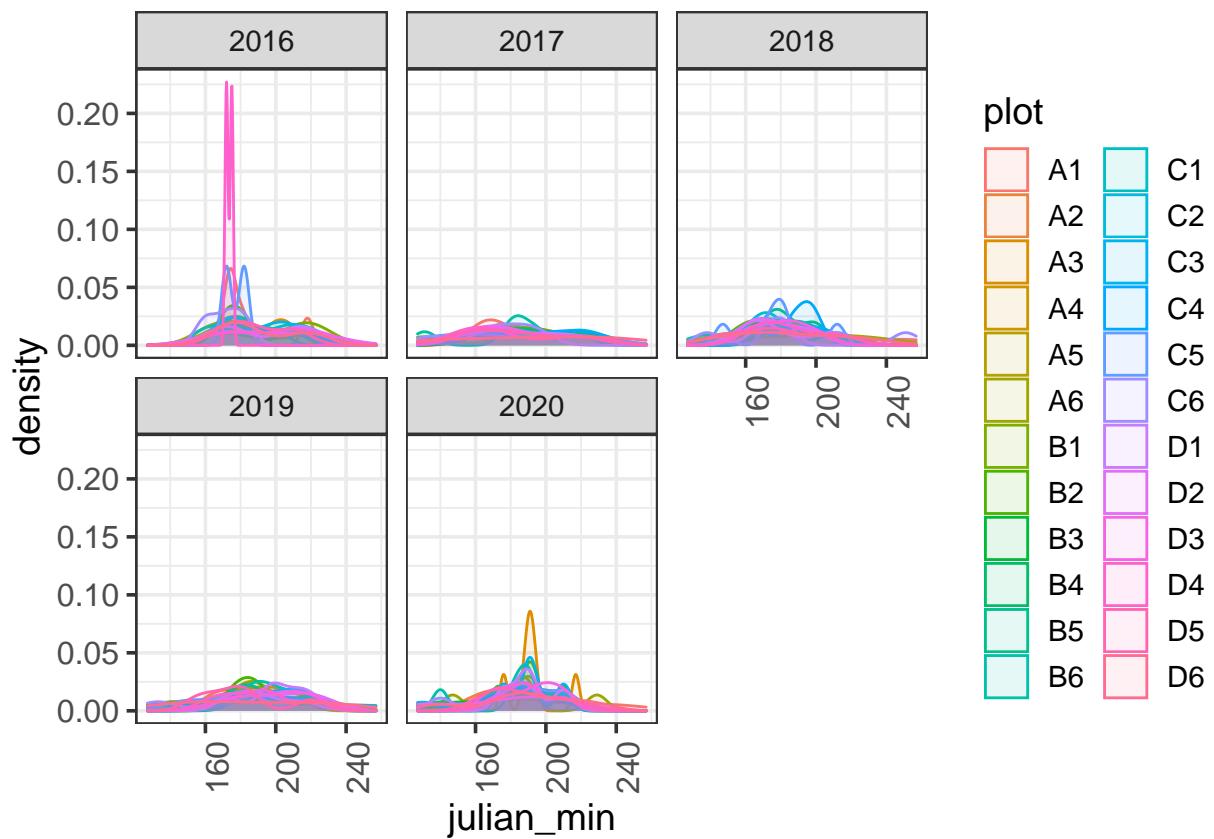
```



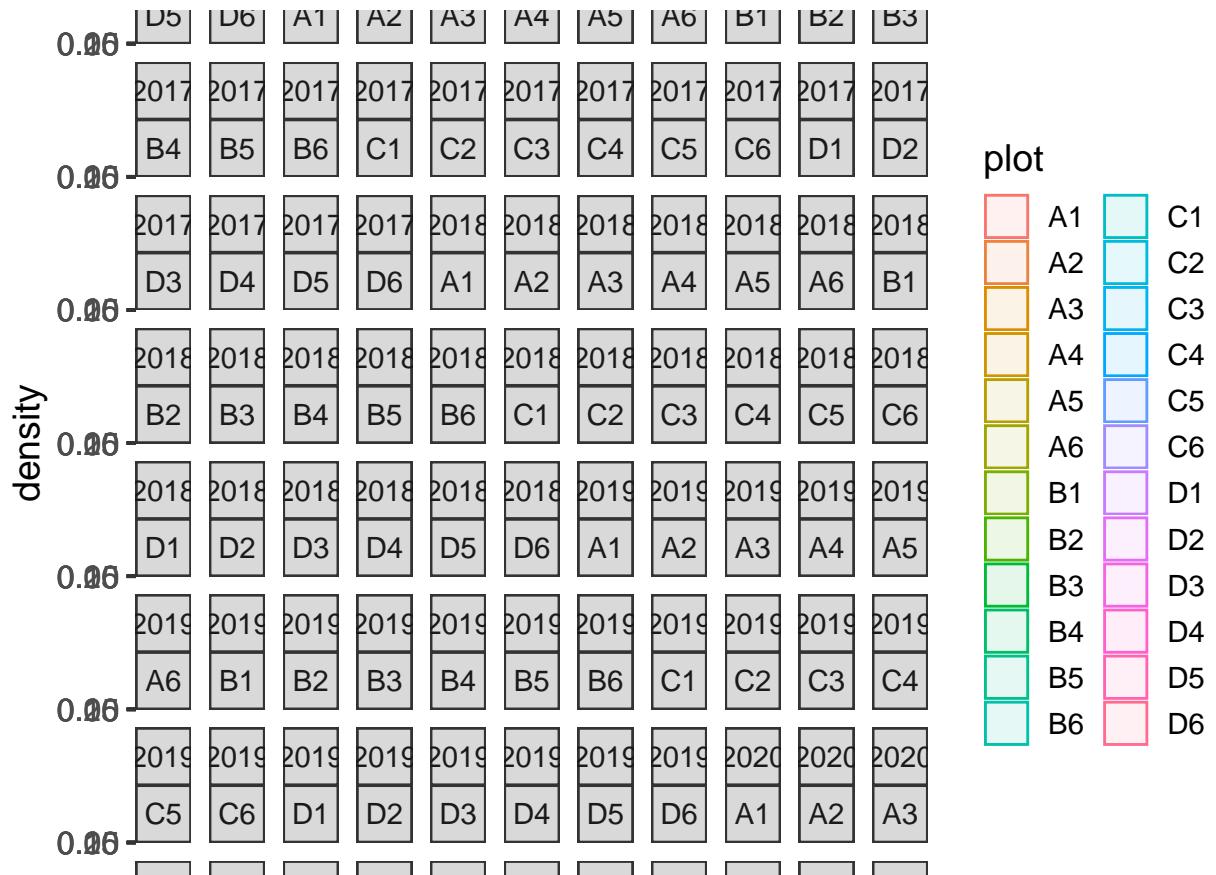
```
ggplot(umbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

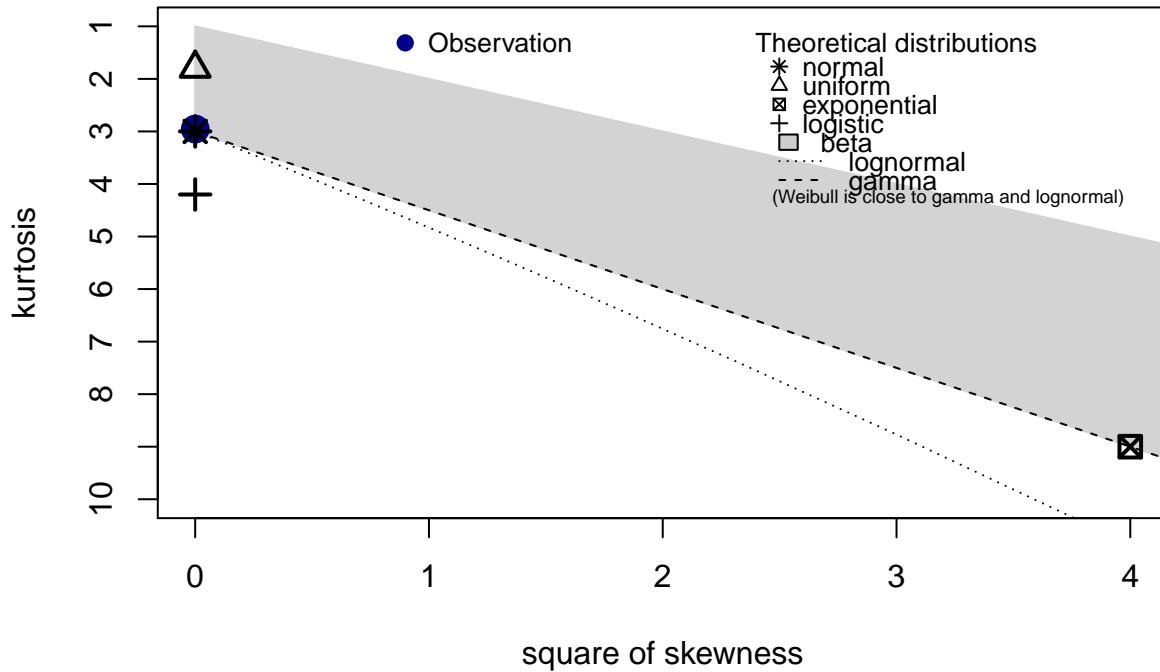


```
ggplot(umbs_sd_spp, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



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

## Cullen and Frey graph

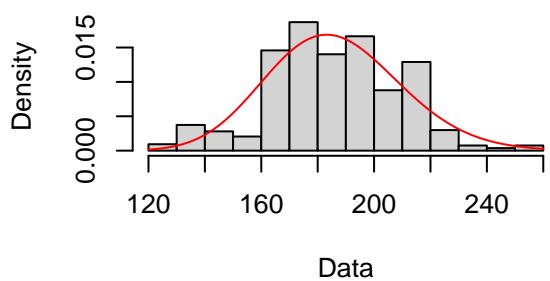
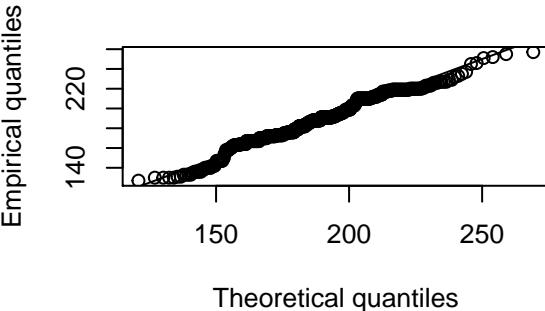
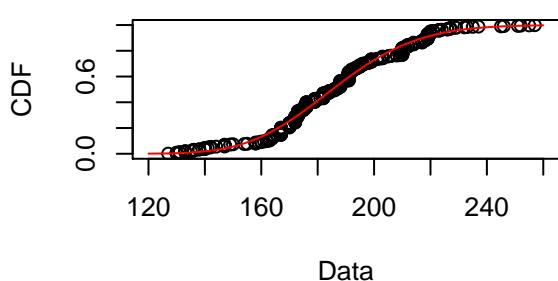
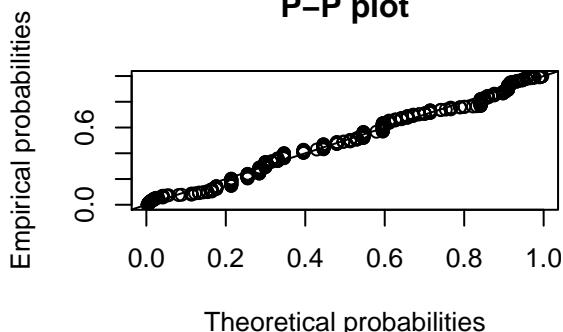


```

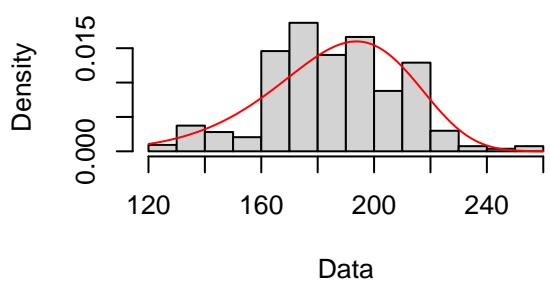
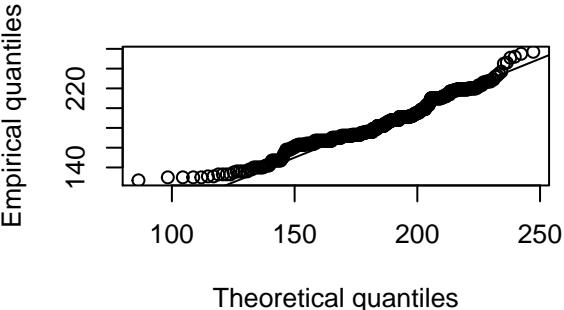
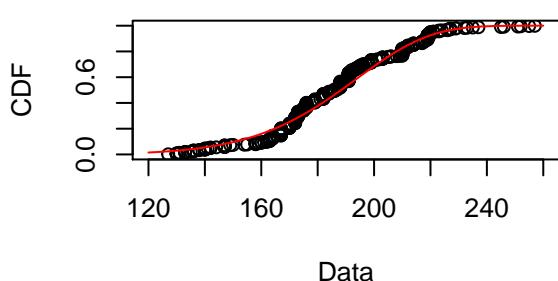
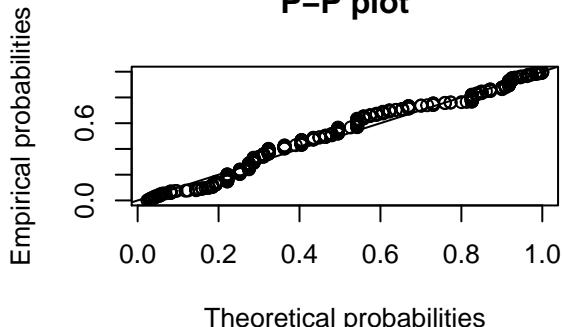
## summary statistics
## -----
## min: 127   max: 257
## median: 186
## mean: 186.2374
## estimated sd: 23.60302
## estimated skewness: 0.01509724
## estimated kurtosis: 2.954713

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

```

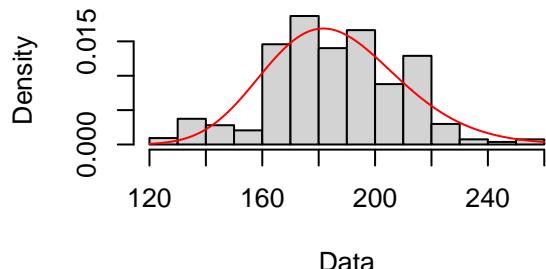
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

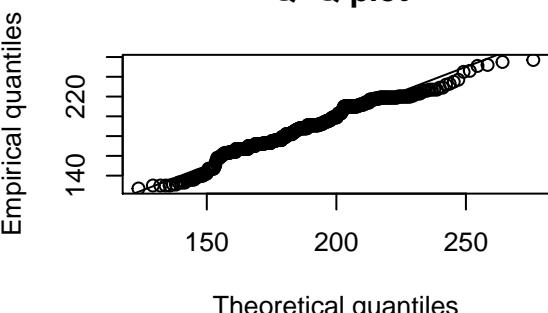
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

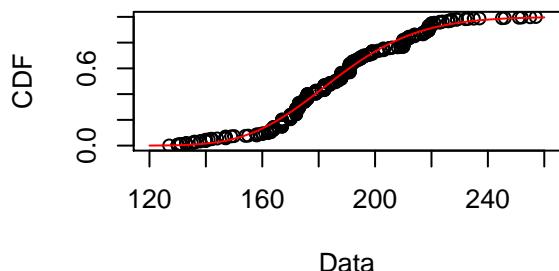
**Empirical and theoretical dens.**



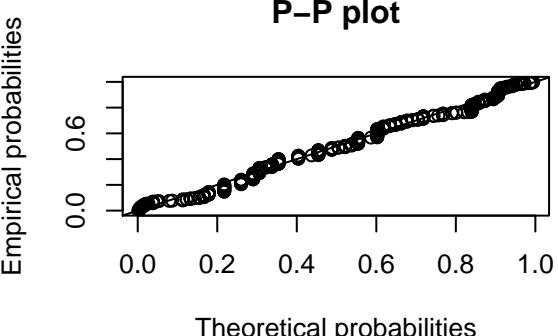
**Q-Q plot**



**Empirical and theoretical CDFs**

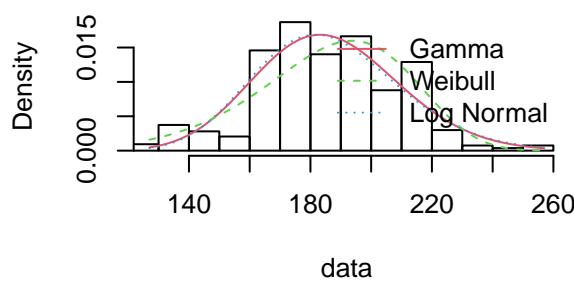


**P-P plot**

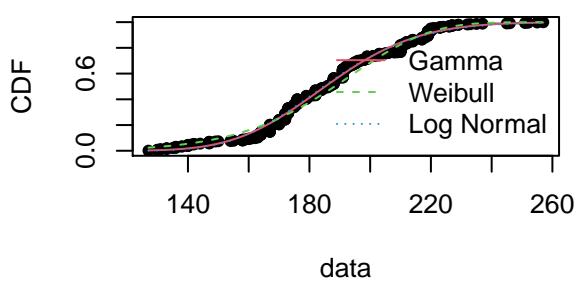


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

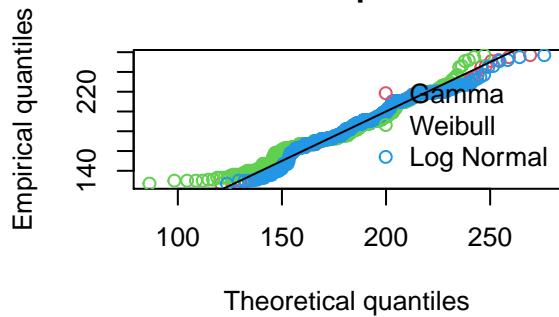
### Histogram and theoretical densities



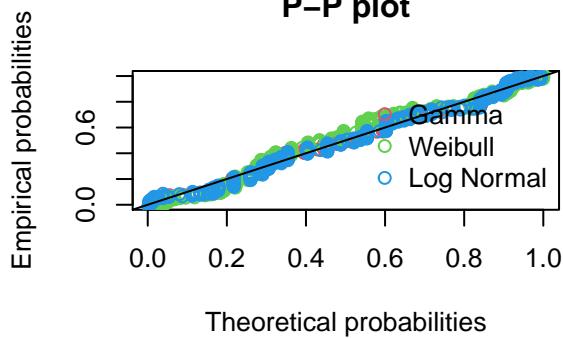
### Empirical and theoretical CDFs



### 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

```

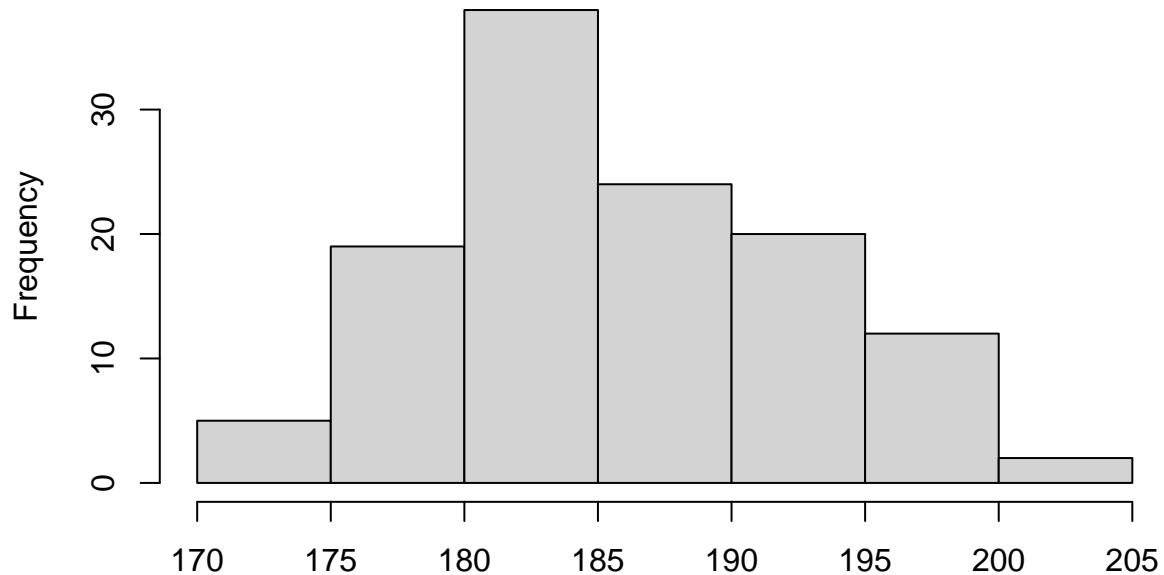
## UMBS PLOT LEVEL - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

```

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

```

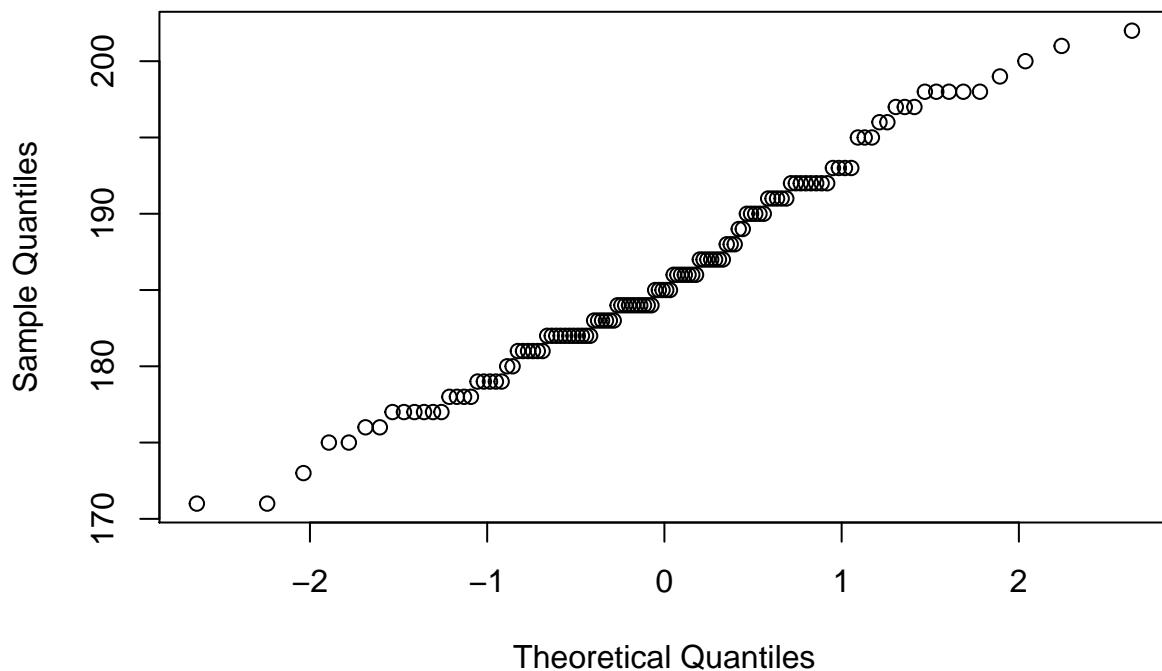
### Histogram of umbs\_sd\_plot\$julian\_min



umbs\_sd\_plot\$julian\_min

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

### Normal Q-Q Plot



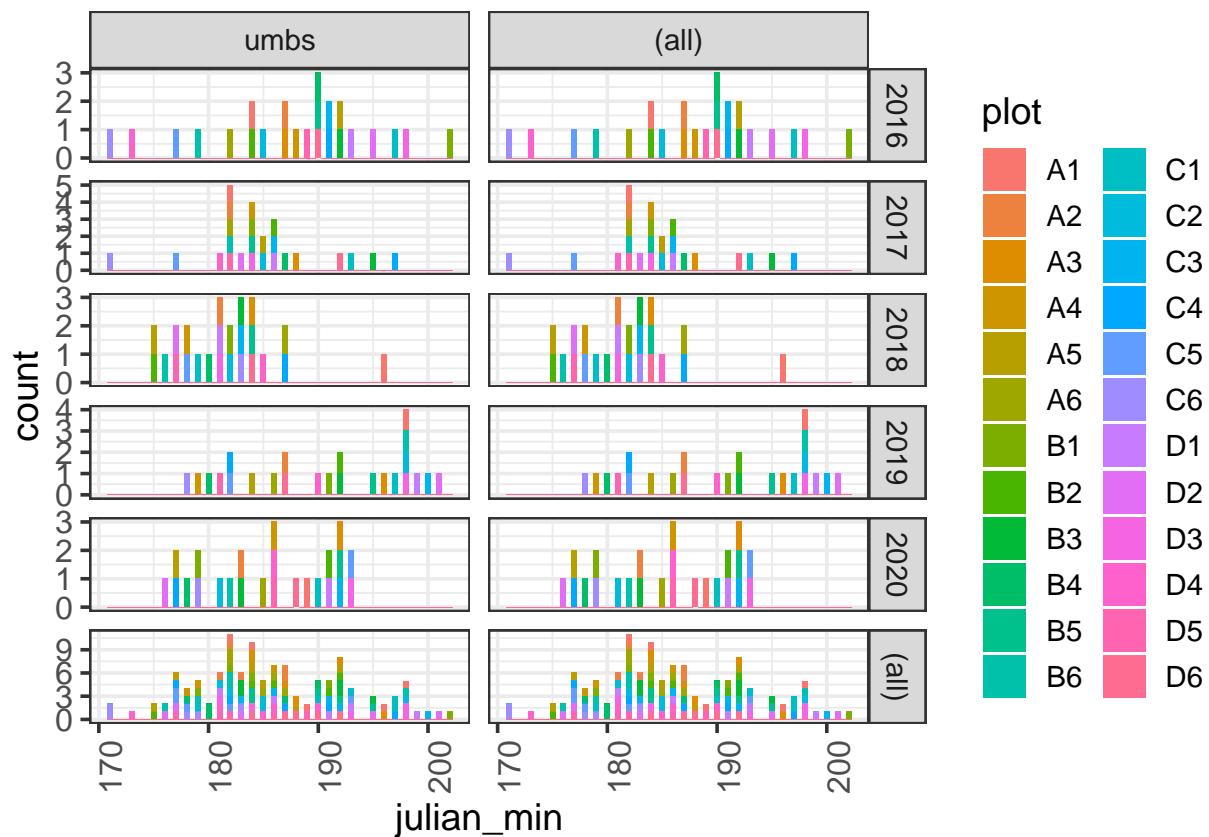
```

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")

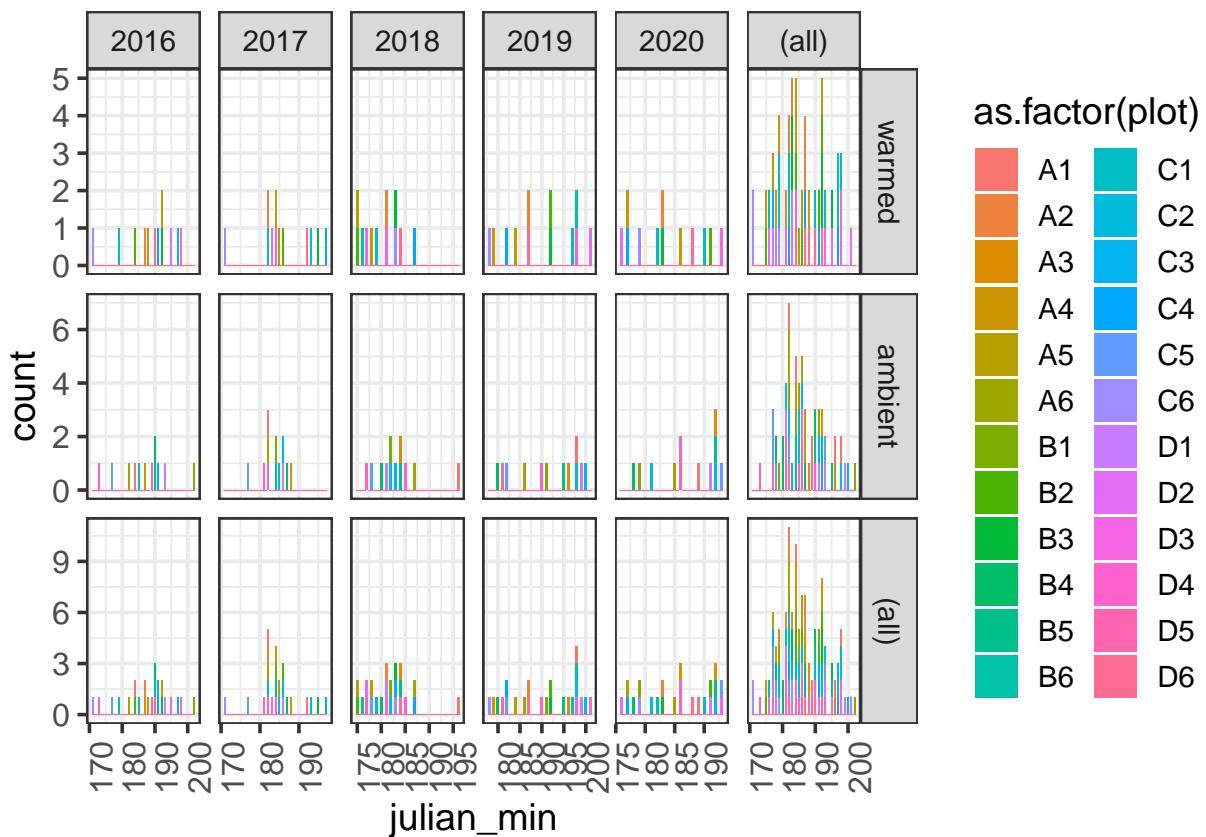
```



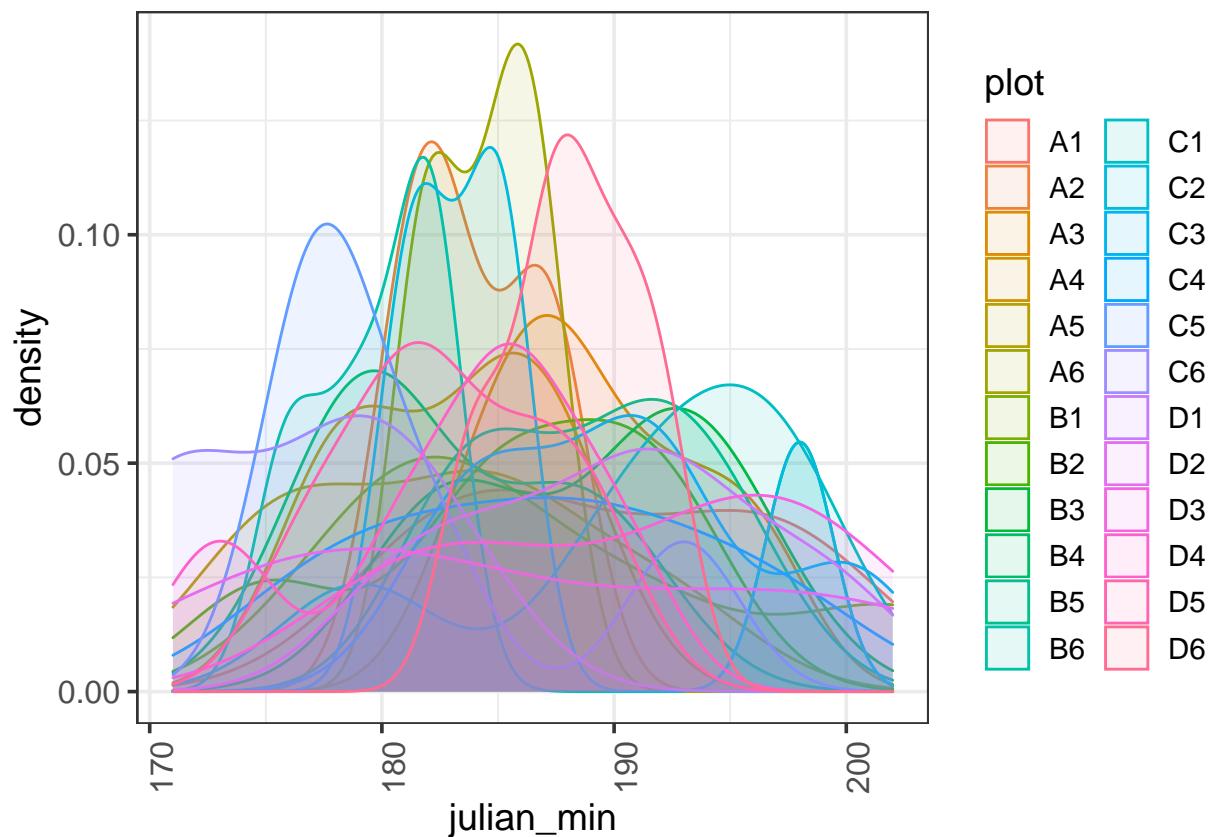
```

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

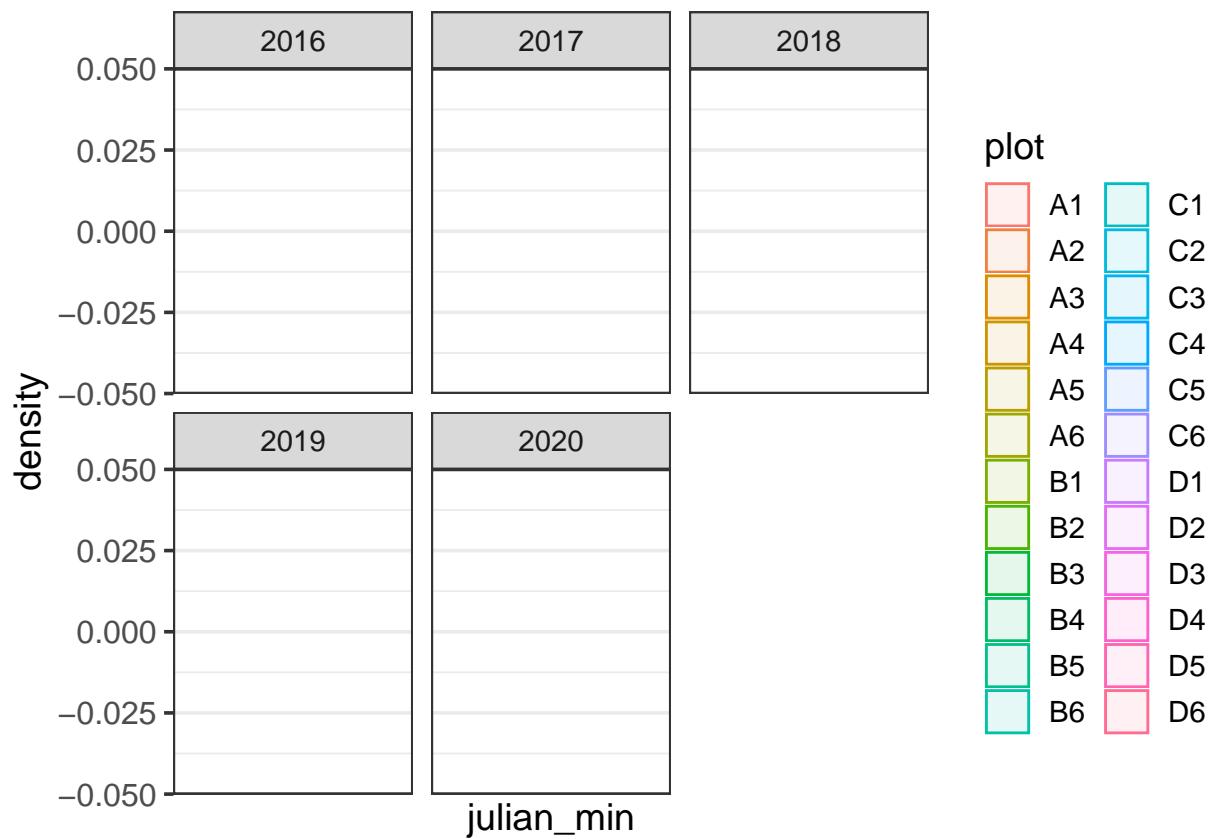
```



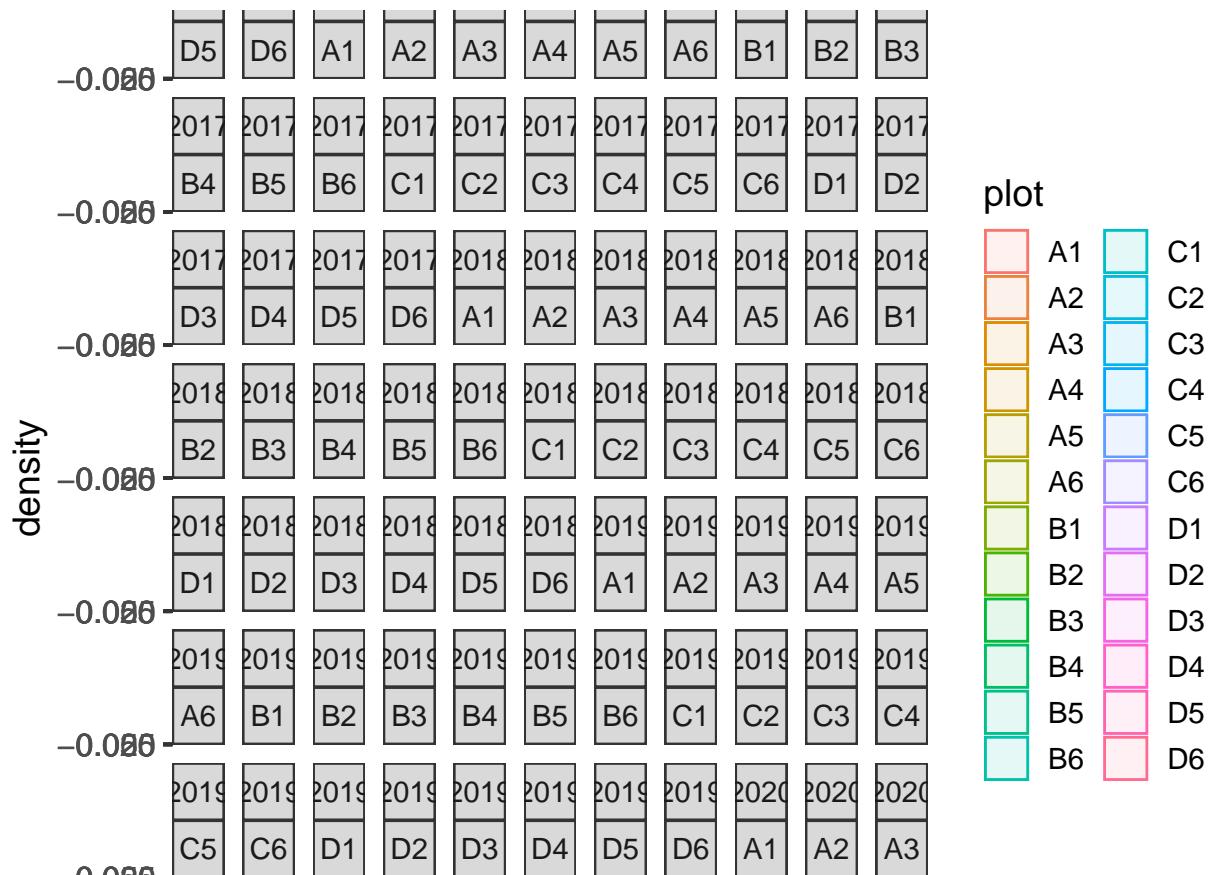
```
ggplot(umbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

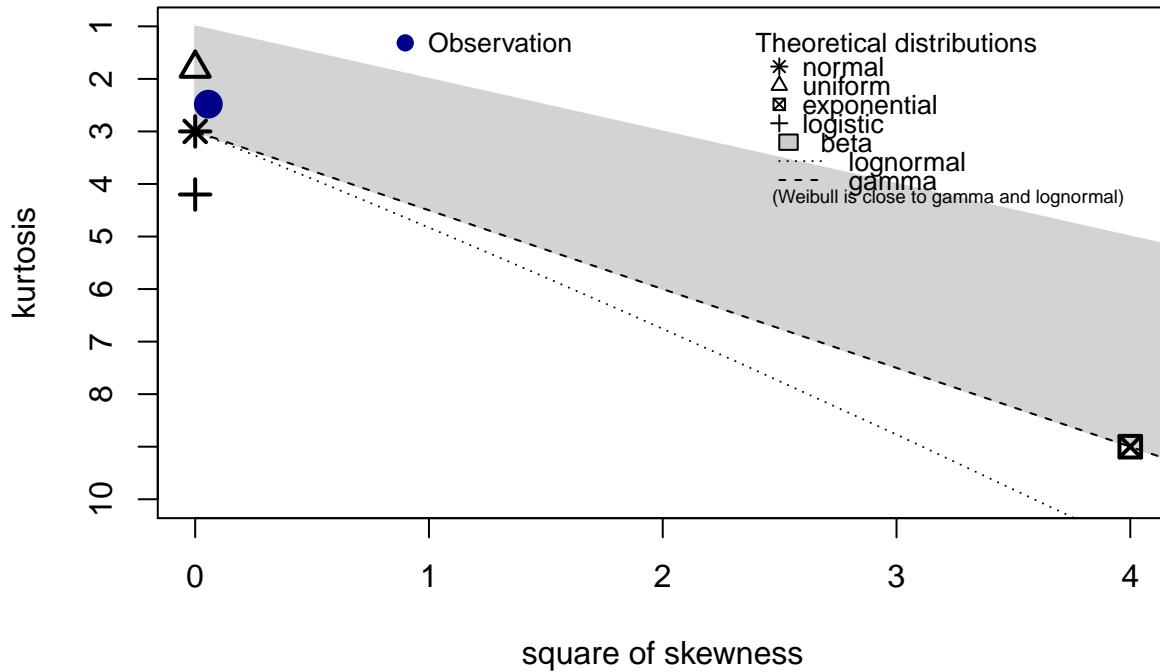


```
ggplot(umbs_sd_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



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

## Cullen and Frey graph



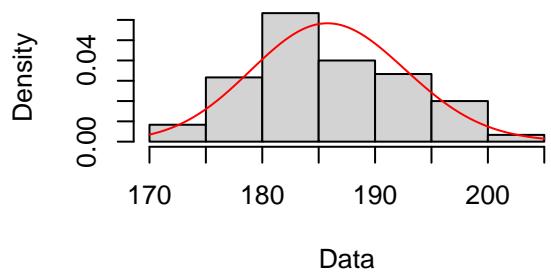
```

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

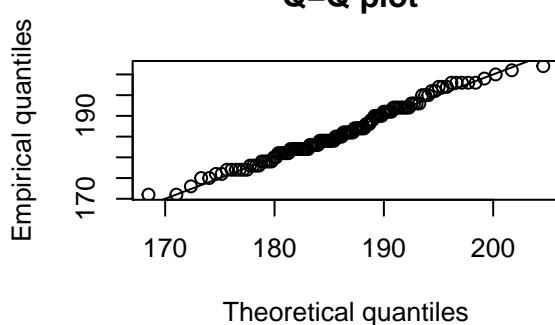
# Gamma distribution
fit.gamma <- fitdist(umbs_sd_plot$julian_min, "gamma")
plot(fit.gamma)

```

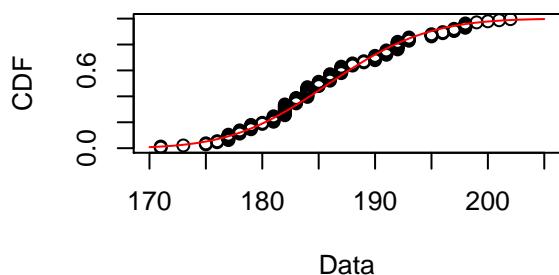
**Empirical and theoretical dens.**



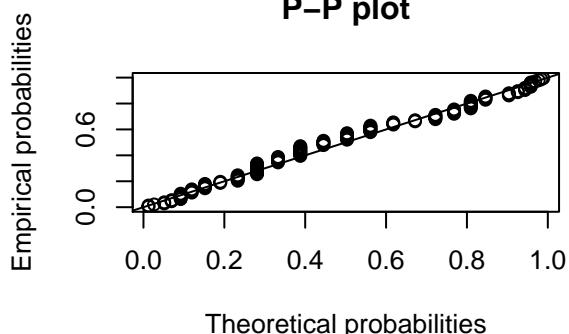
**Q-Q plot**



**Empirical and theoretical CDFs**

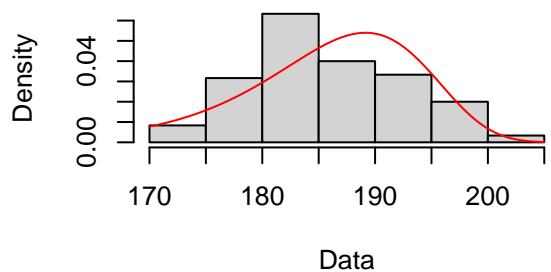


**P-P plot**

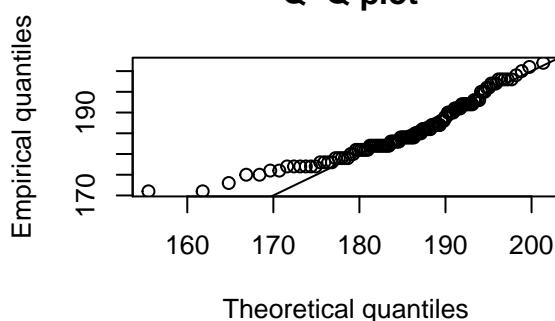


```
# Weibull distribution
fit.weibull <- fitdist(umbs_sd_plot$julian_min, "weibull")
plot(fit.weibull)
```

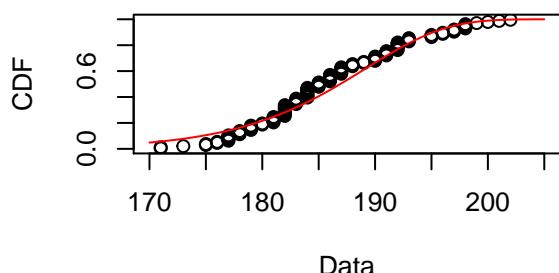
**Empirical and theoretical dens.**



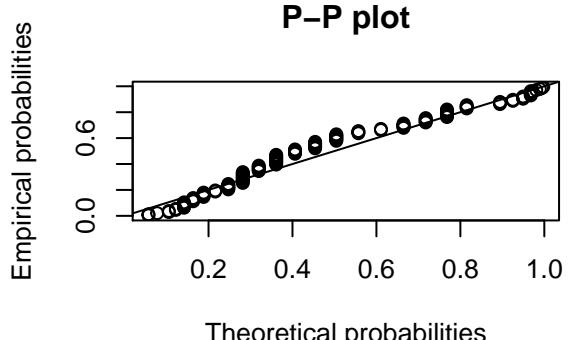
**Q-Q plot**



**Empirical and theoretical CDFs**

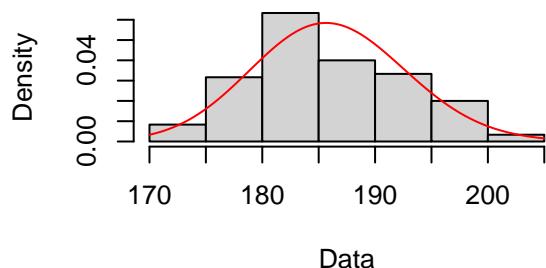


**P-P plot**

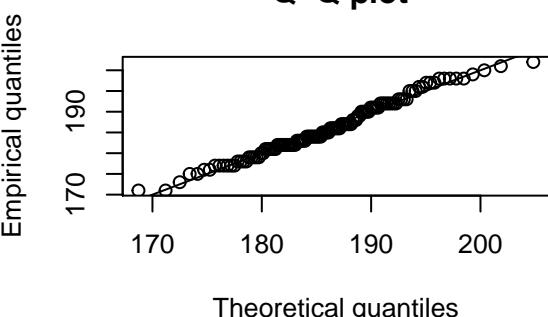


```
# Lognormal distribution
fit.ln <- fitdist(umbss_sd_plot$julian_min, "lnorm")
plot(fit.ln)
```

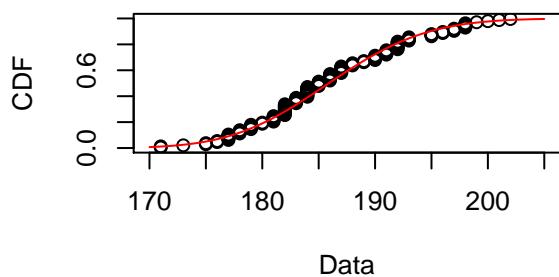
**Empirical and theoretical dens.**



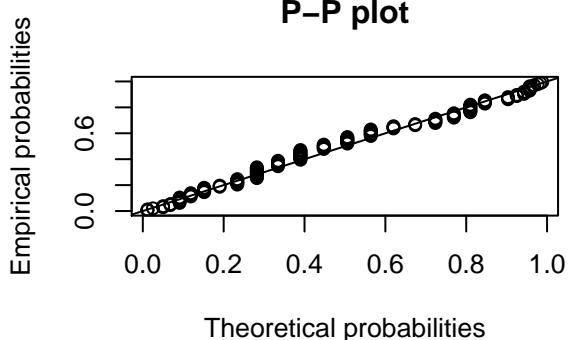
**Q-Q plot**



**Empirical and theoretical CDFs**

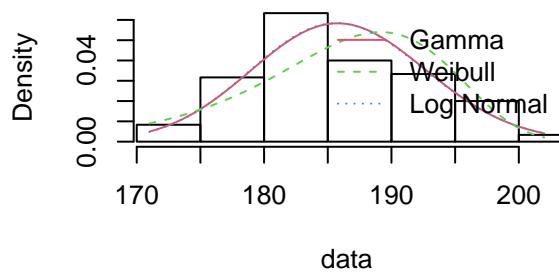


**P-P plot**

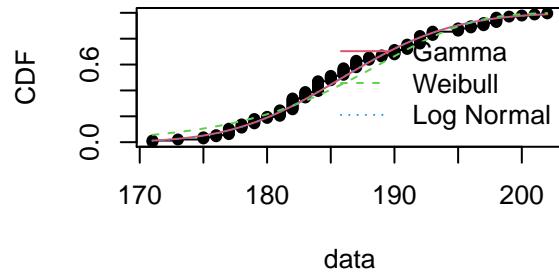


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

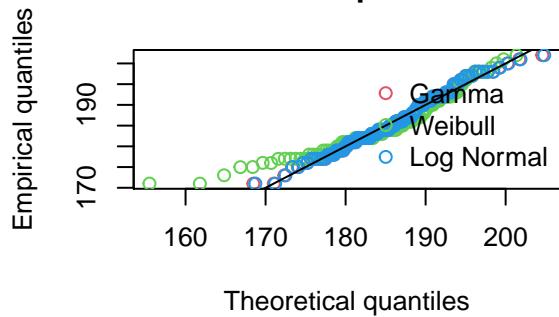
### Histogram and theoretical densities



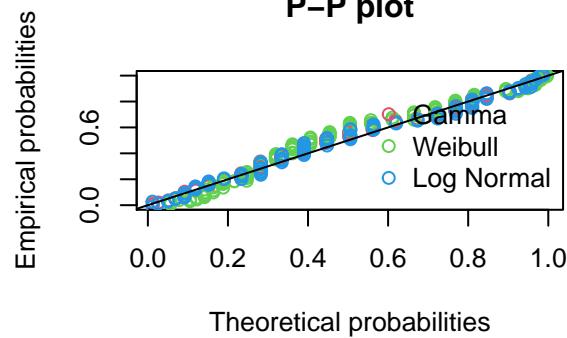
### Empirical and theoretical CDFs



### 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

```

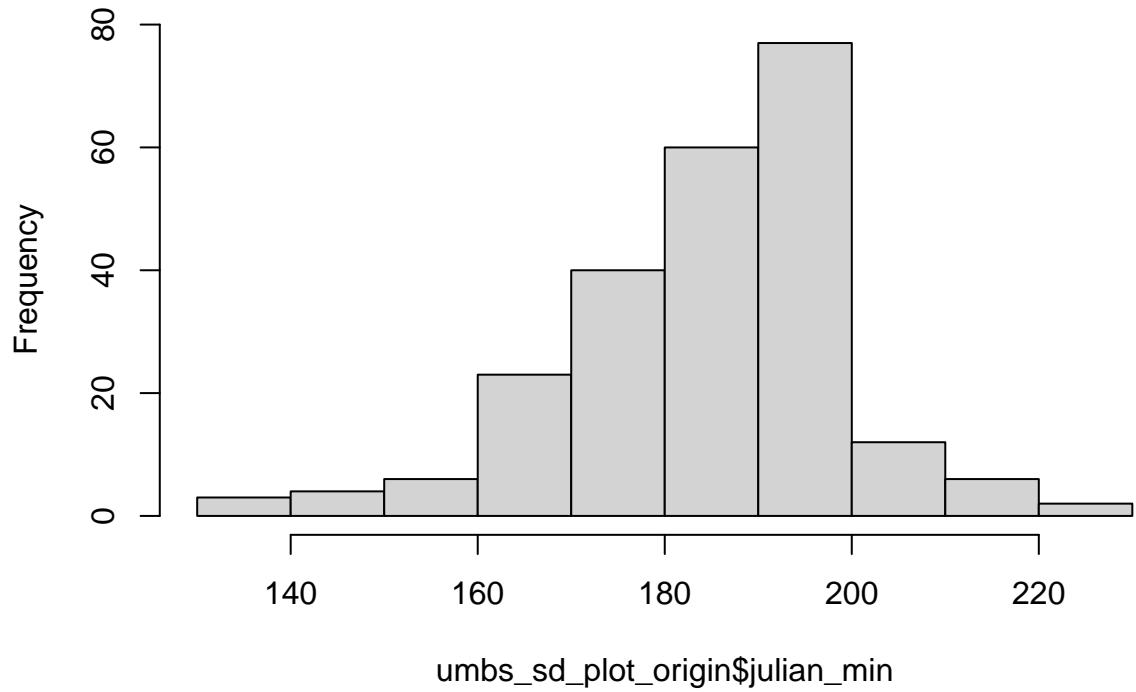
## UMBS PLOT LEVEL ORIGIN - Looking at MINIMUM JULIAN DAY (aka date of observed first seed set)

```

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

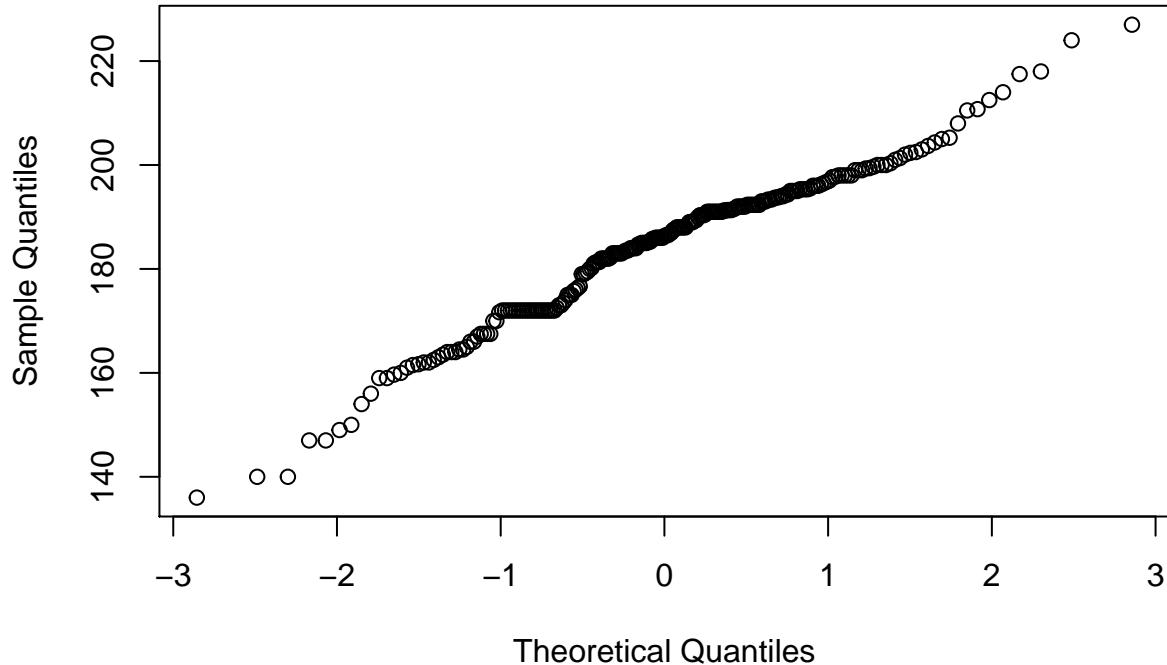
```

### Histogram of umbs\_sd\_plot\_origin\$julian\_min



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

### Normal Q-Q Plot



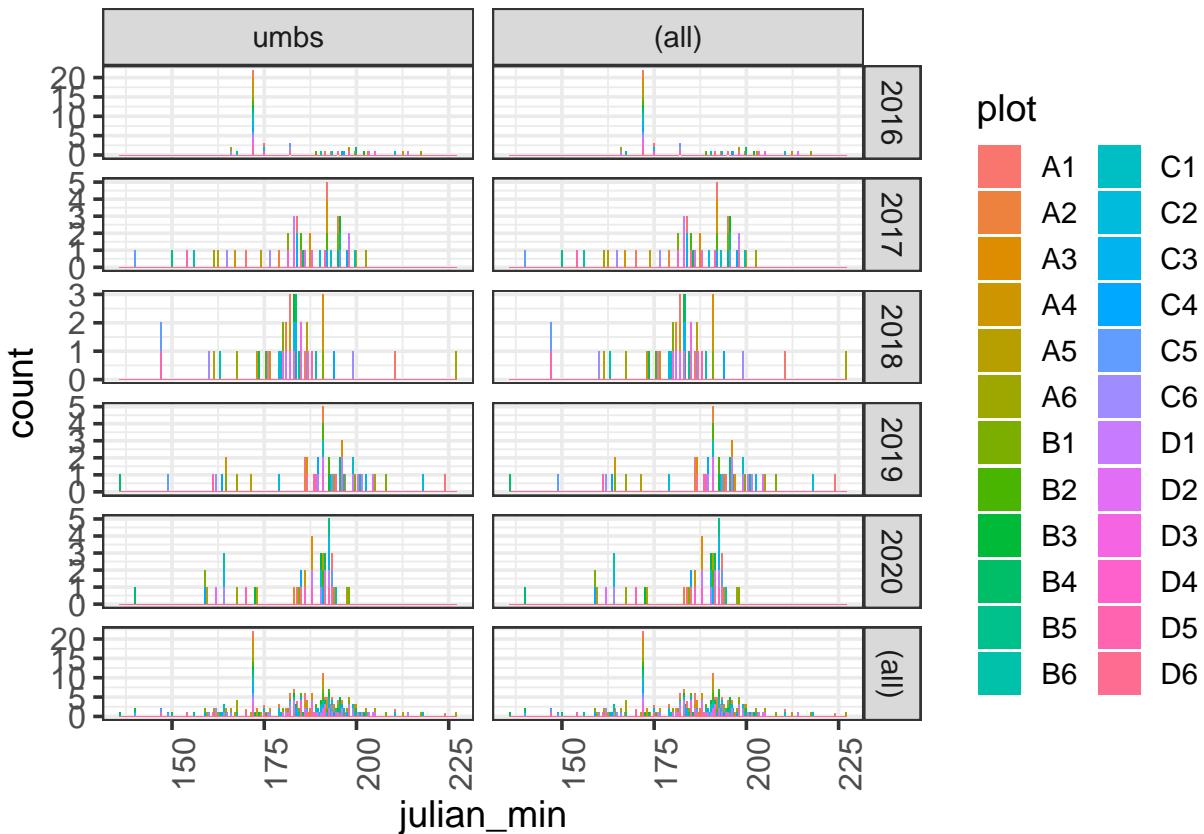
```

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")

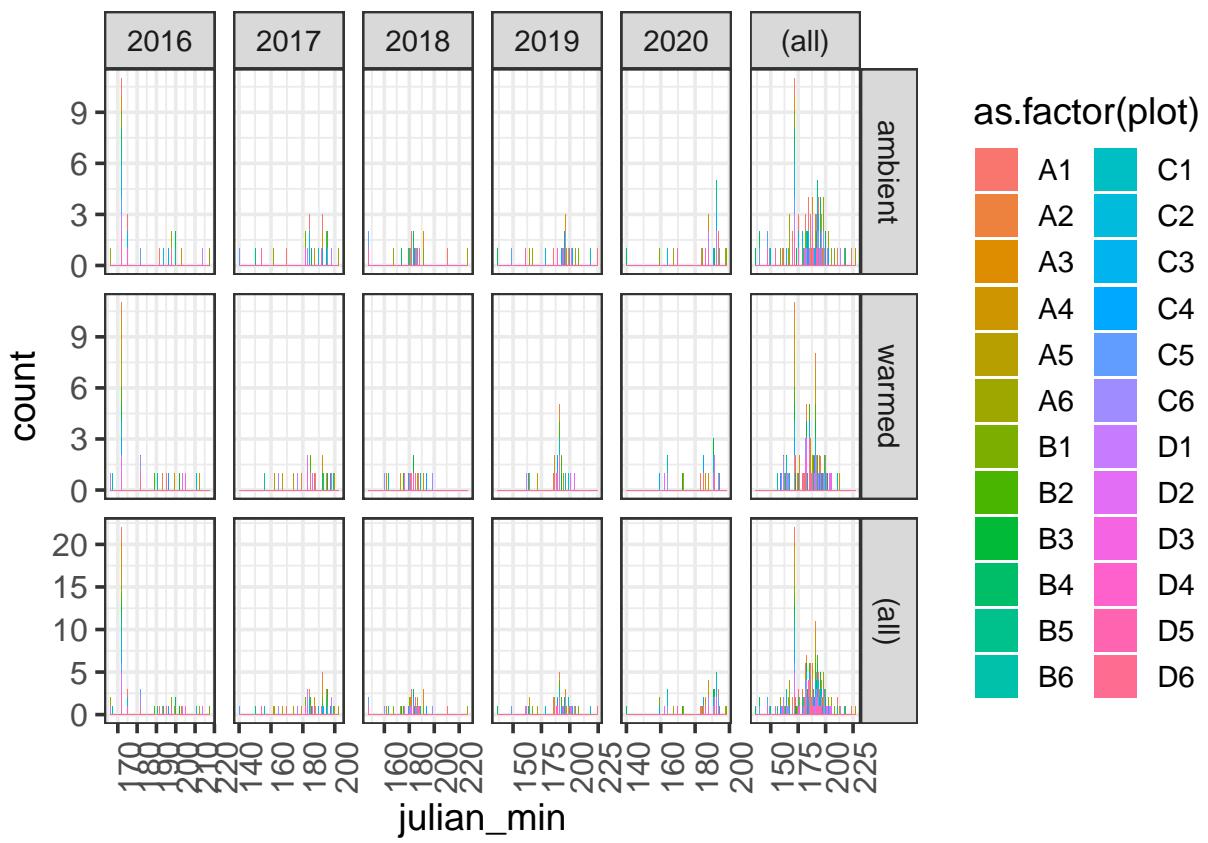
```



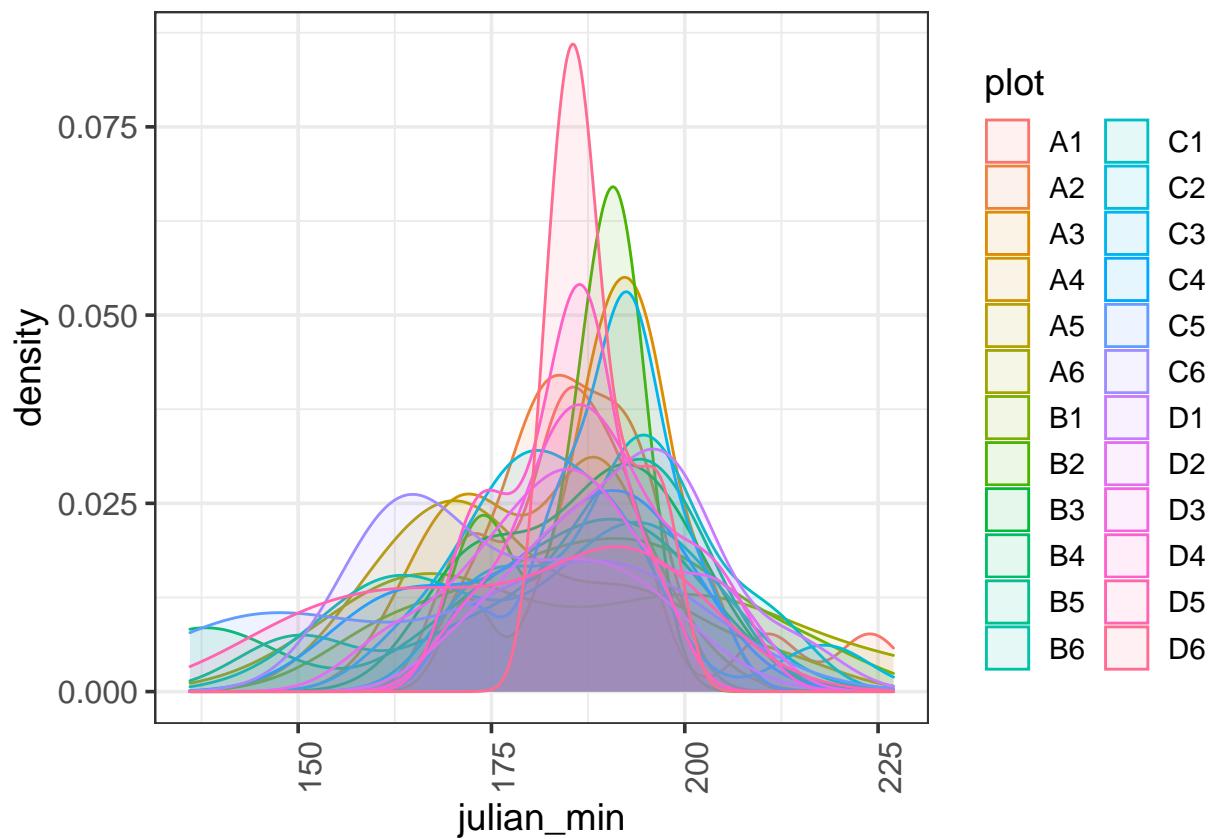
```

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")

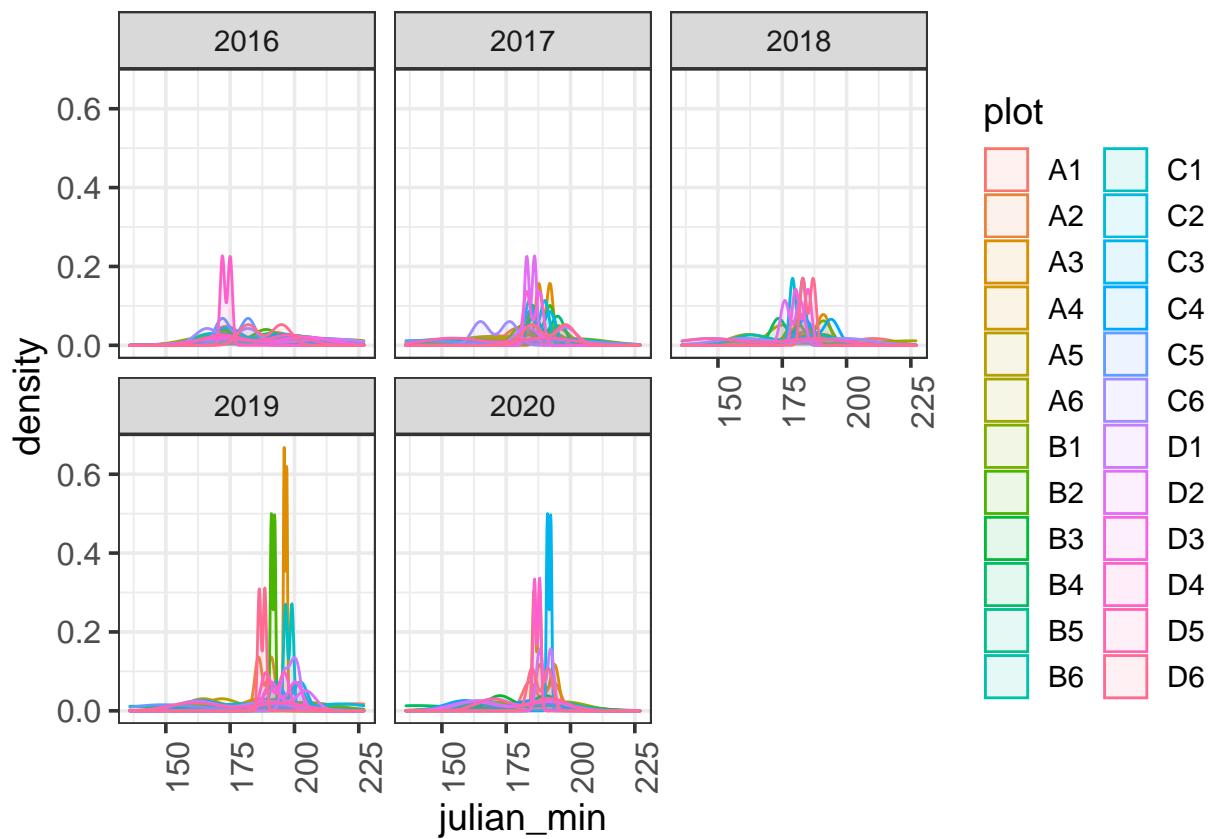
```



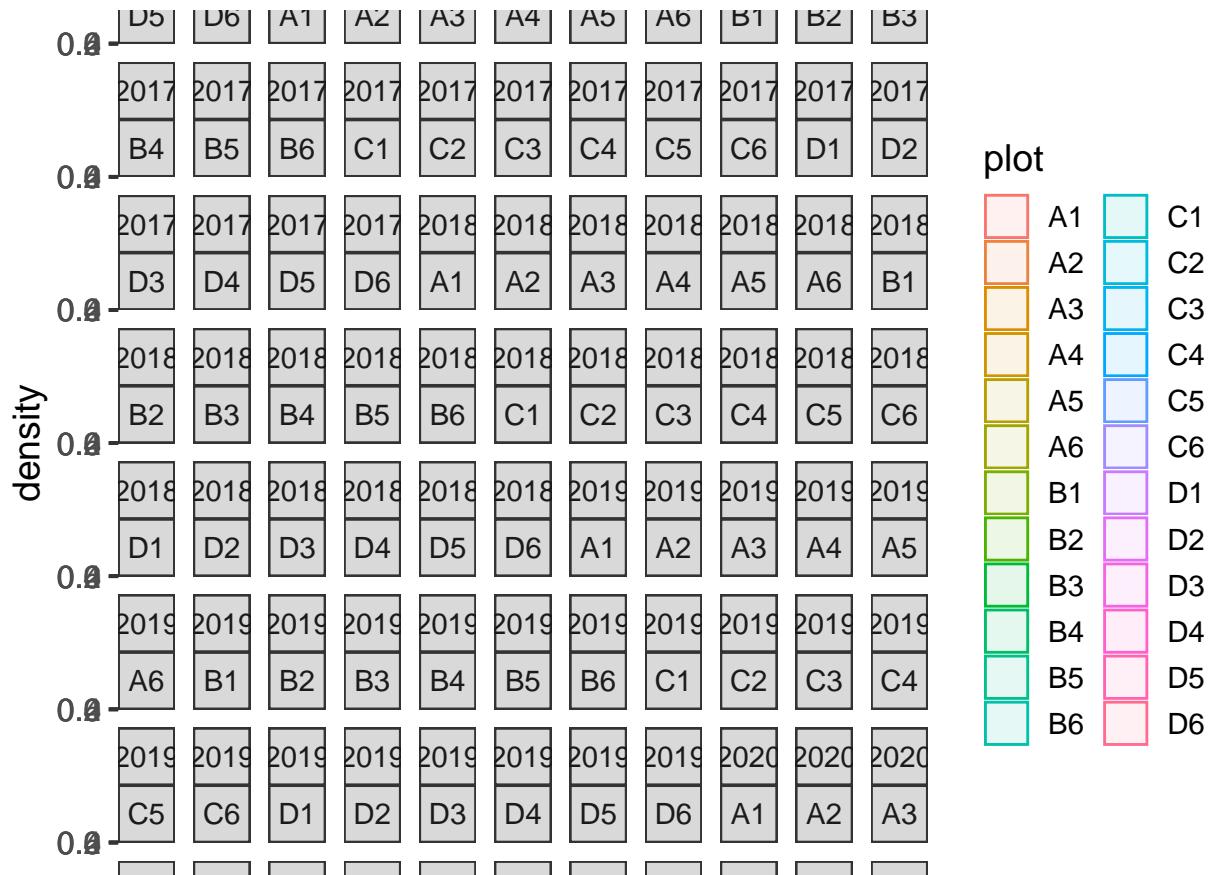
```
ggplot(umbs_sd_plot_origin, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

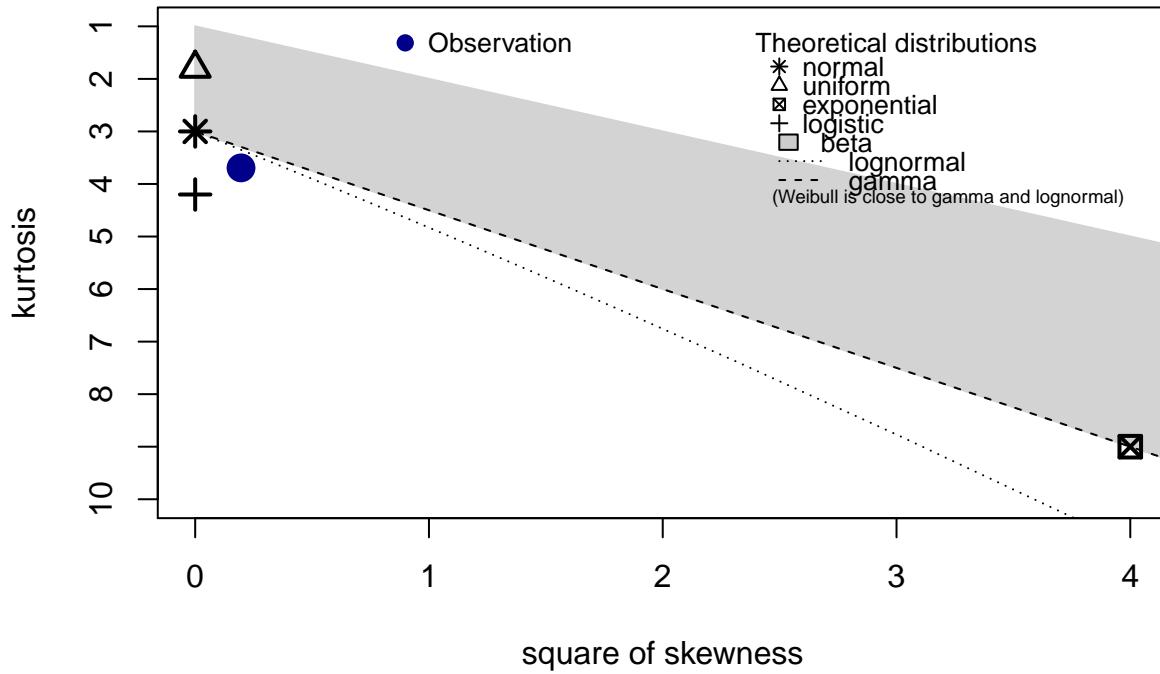


```
ggplot(umbs_sd_plot_origin, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1) +
  facet_wrap(~year + plot)
```



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

## Cullen and Frey graph

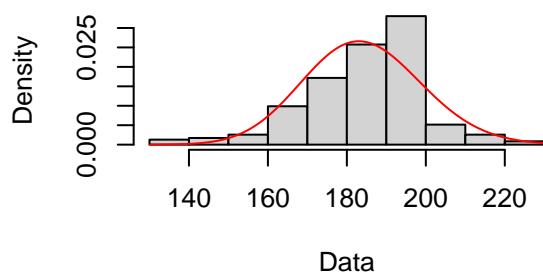
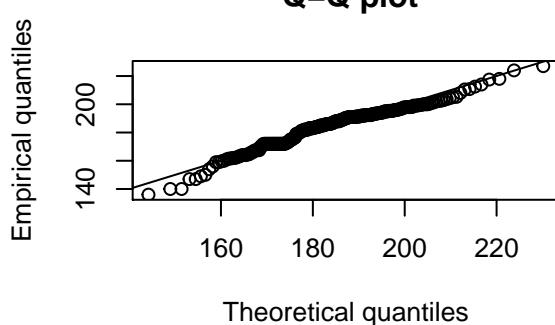
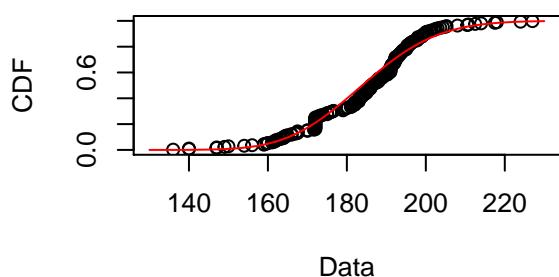
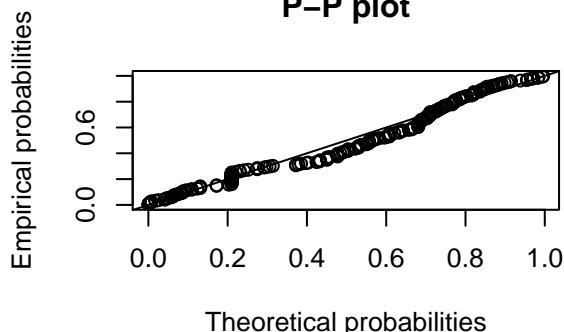


```

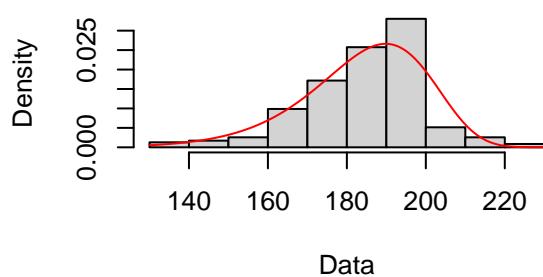
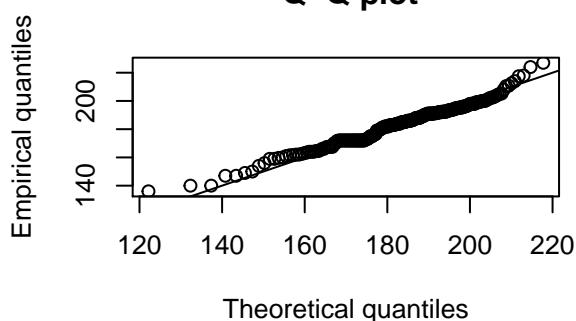
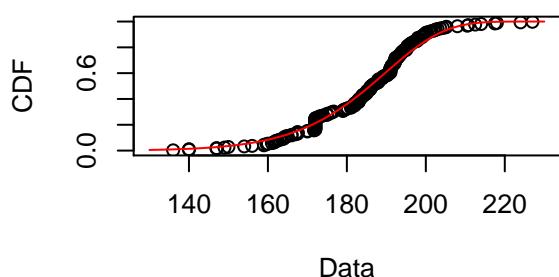
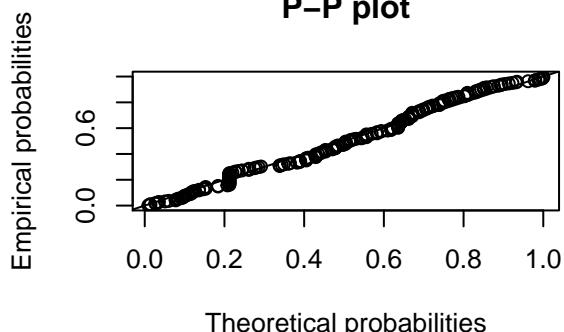
## summary statistics
## -----
## min: 136   max: 227
## median: 186.3333
## mean: 184.2609
## estimated sd: 14.82065
## estimated skewness: -0.4428369
## estimated kurtosis: 3.693731

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

```

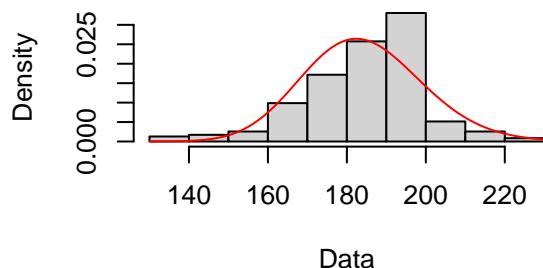
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

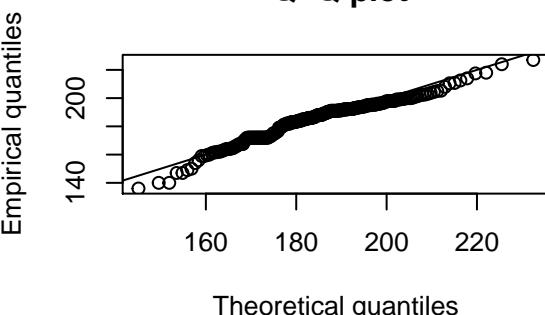
**Empirical and theoretical dens.****Q-Q plot****Empirical and theoretical CDFs****P-P plot**

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

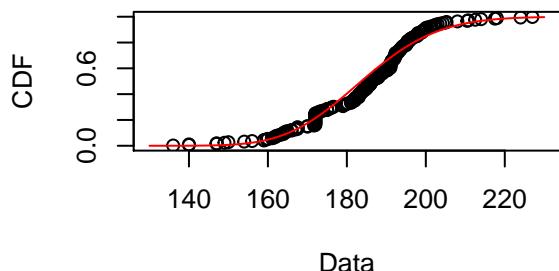
**Empirical and theoretical dens.**



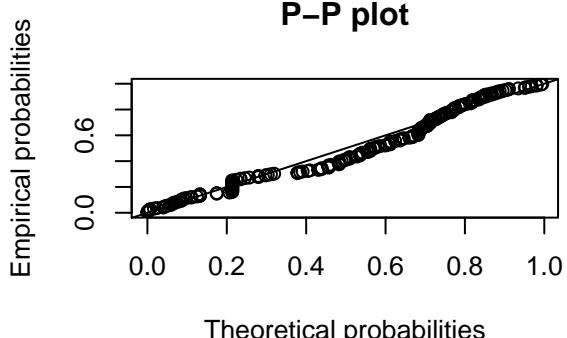
**Q-Q plot**



**Empirical and theoretical CDFs**

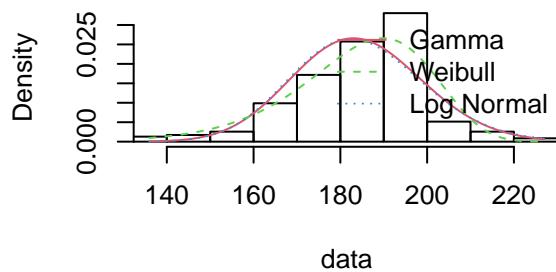


**P-P plot**

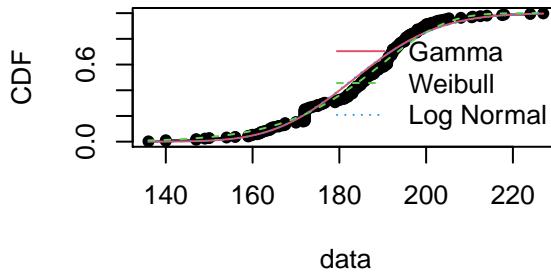


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

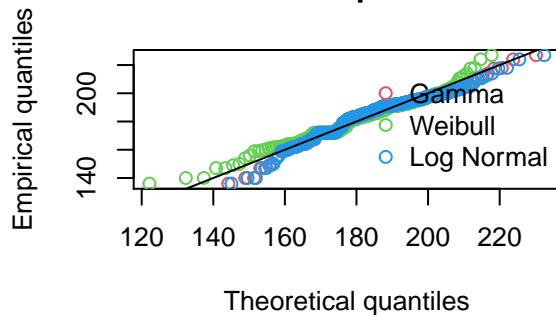
### Histogram and theoretical densities



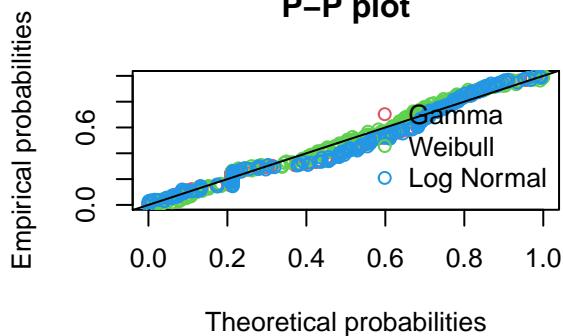
### Empirical and theoretical CDFs



### 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"))

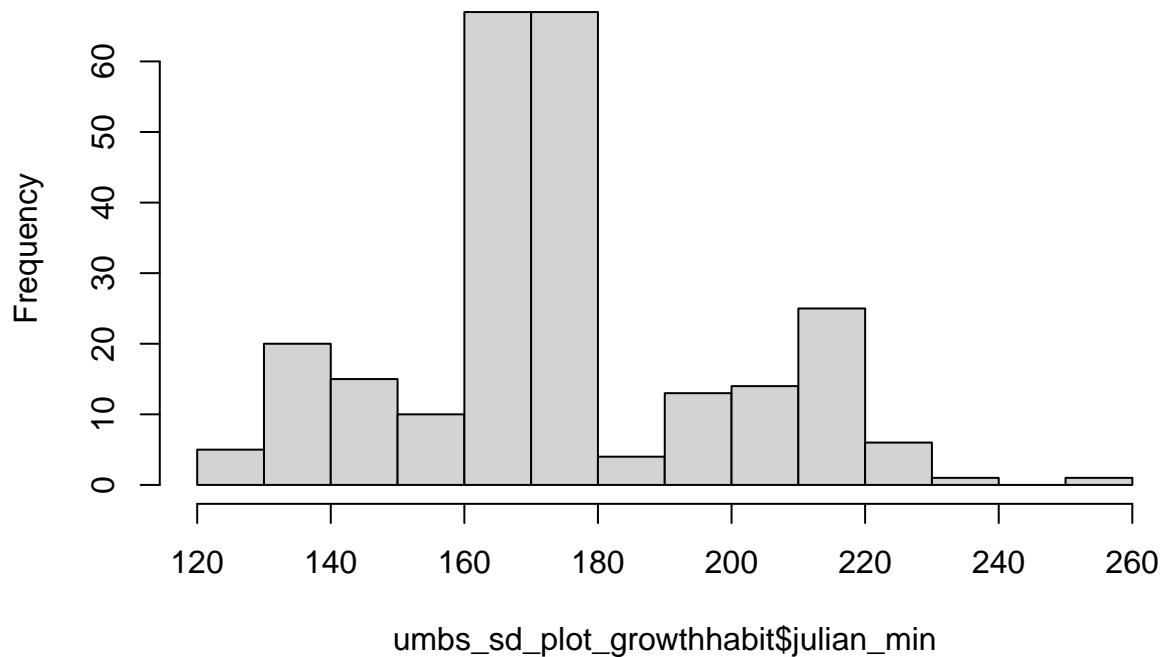
## Goodness-of-fit statistics
##                                     Gamma    Weibull Log Normal
## Kolmogorov-Smirnov statistic 0.1039092 0.06597959 0.1095598
## Cramer-von Mises statistic   0.5724530 0.26609180 0.6386879
## Anderson-Darling statistic   3.0359342 1.87536361 3.3972035
##
## Goodness-of-fit 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)
```

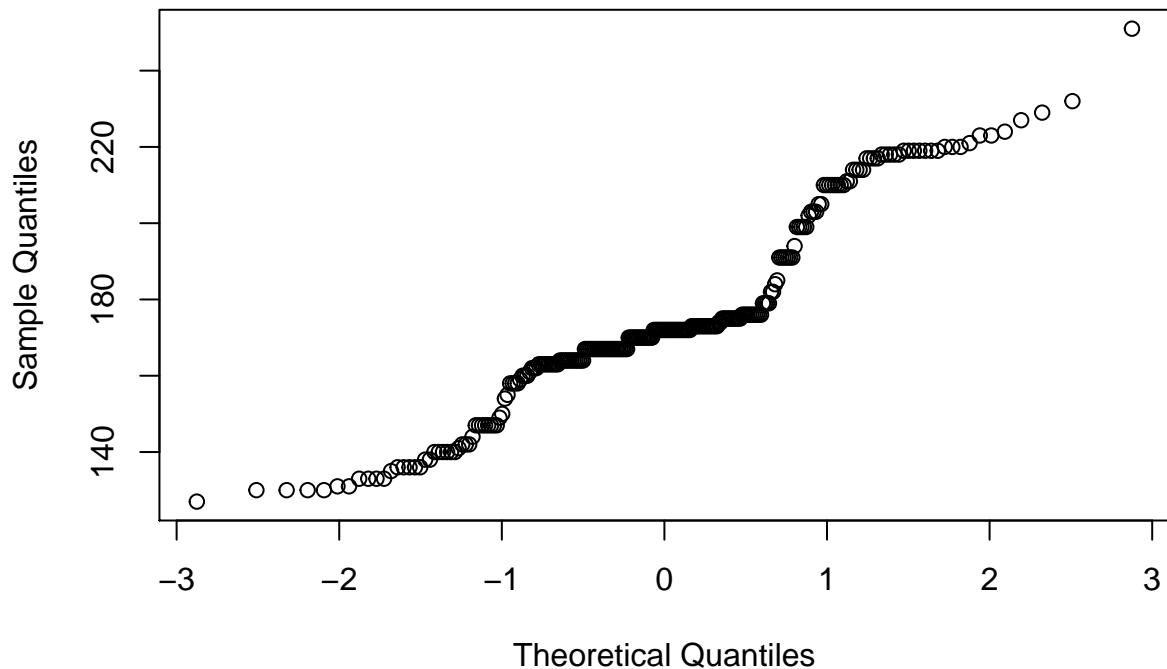
## Histogram of umbs\_sd\_plot\_growthhabit\$julian\_min



umbs\_sd\_plot\_growthhabit\$julian\_min

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

## Normal Q-Q Plot



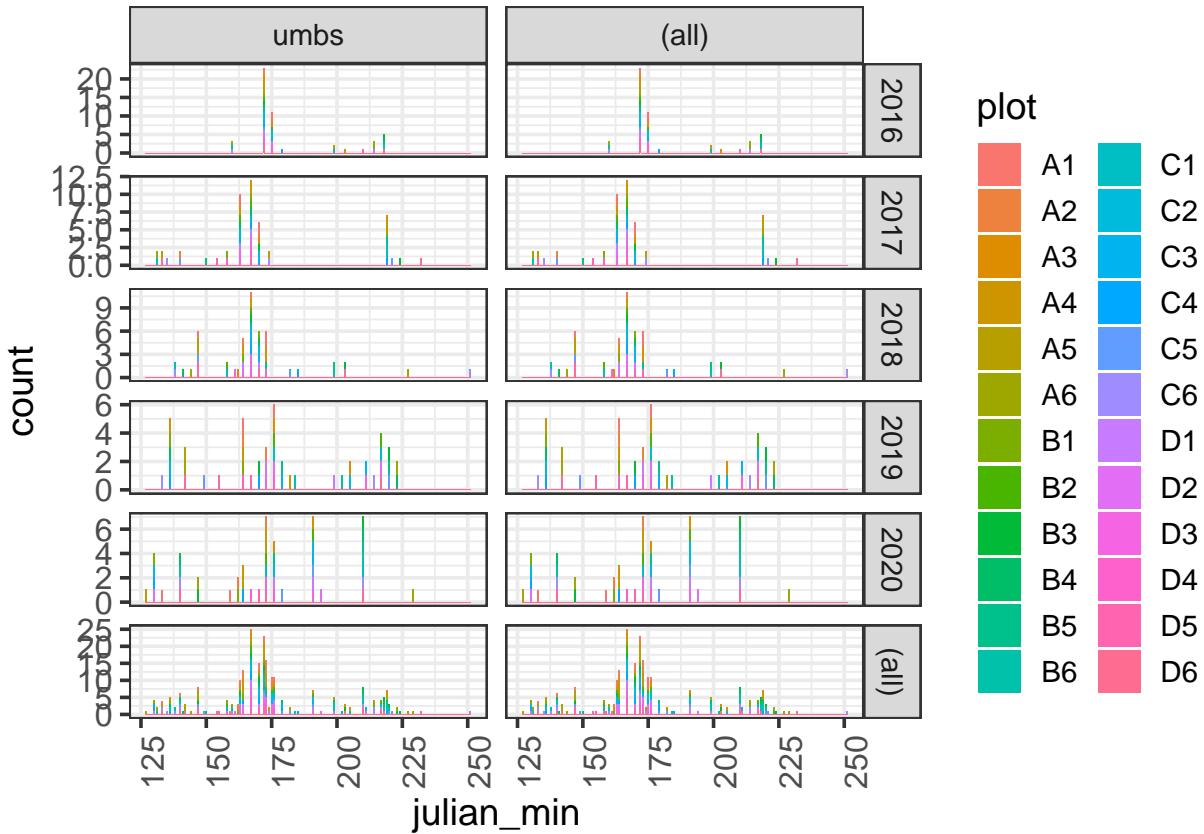
```

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 = 0.9322, p-value = 2.951e-09

# 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")

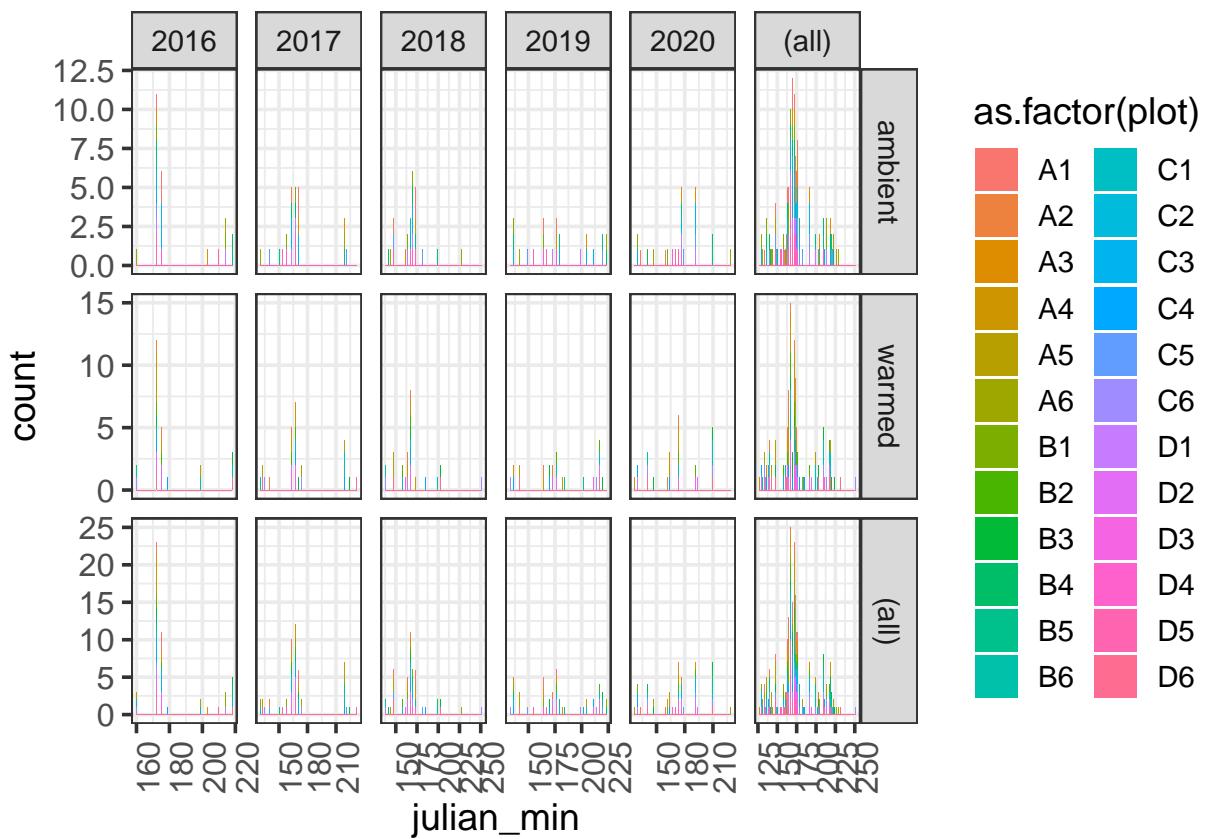
```



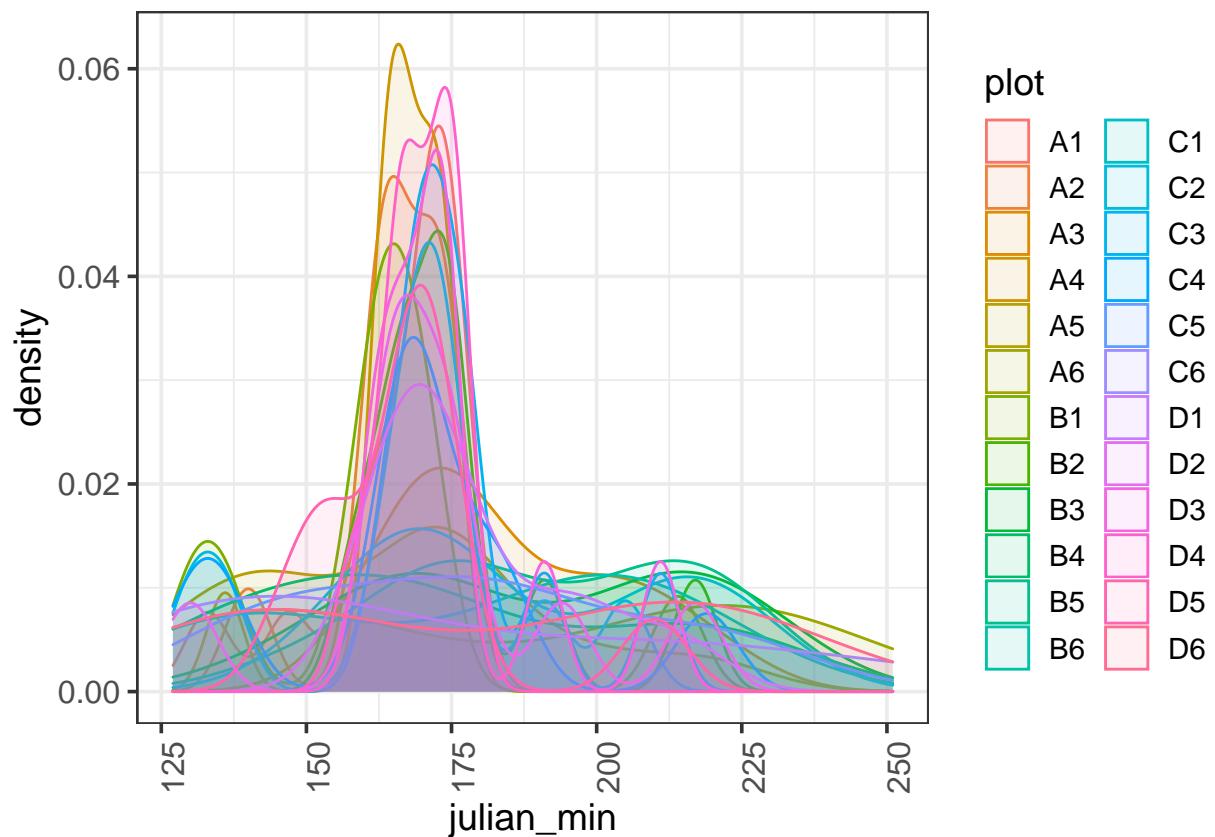
```

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")

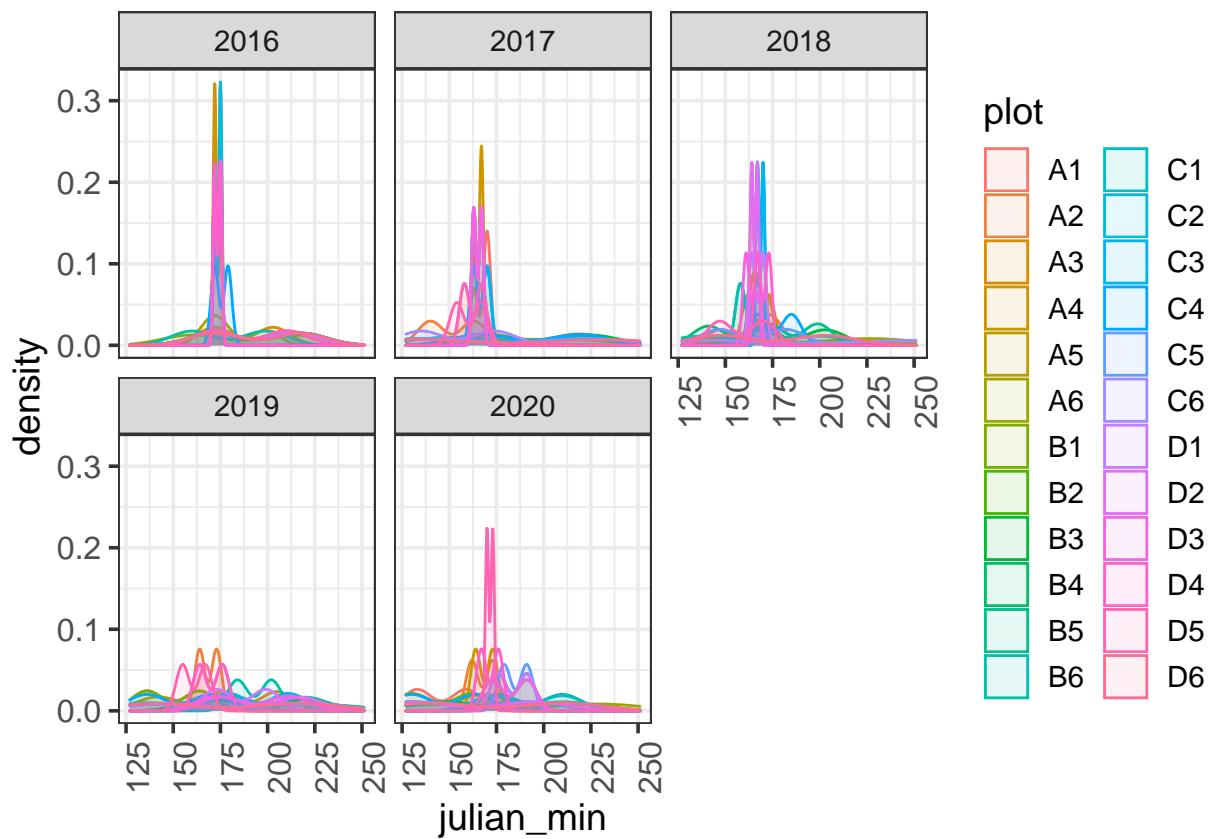
```



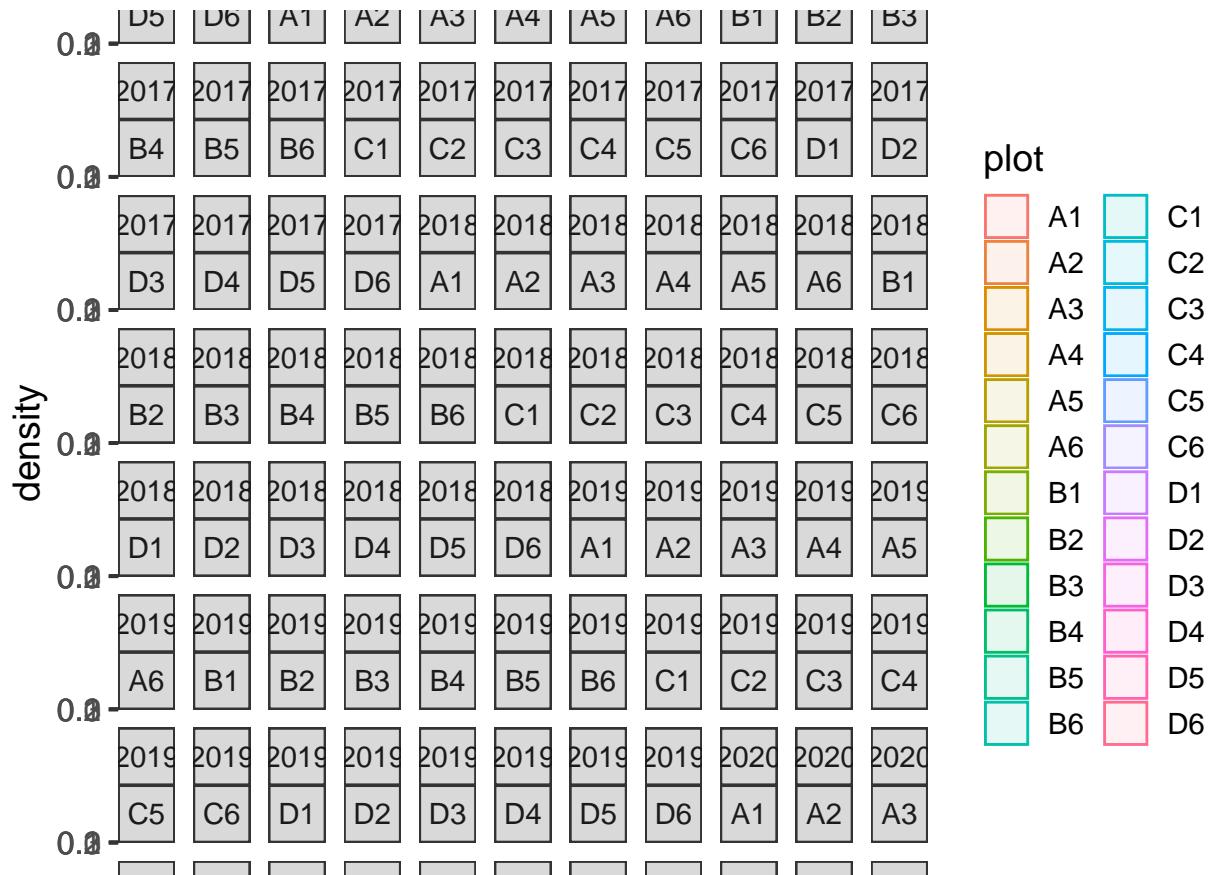
```
ggplot(umbs_sd_plot_growthhabit, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



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

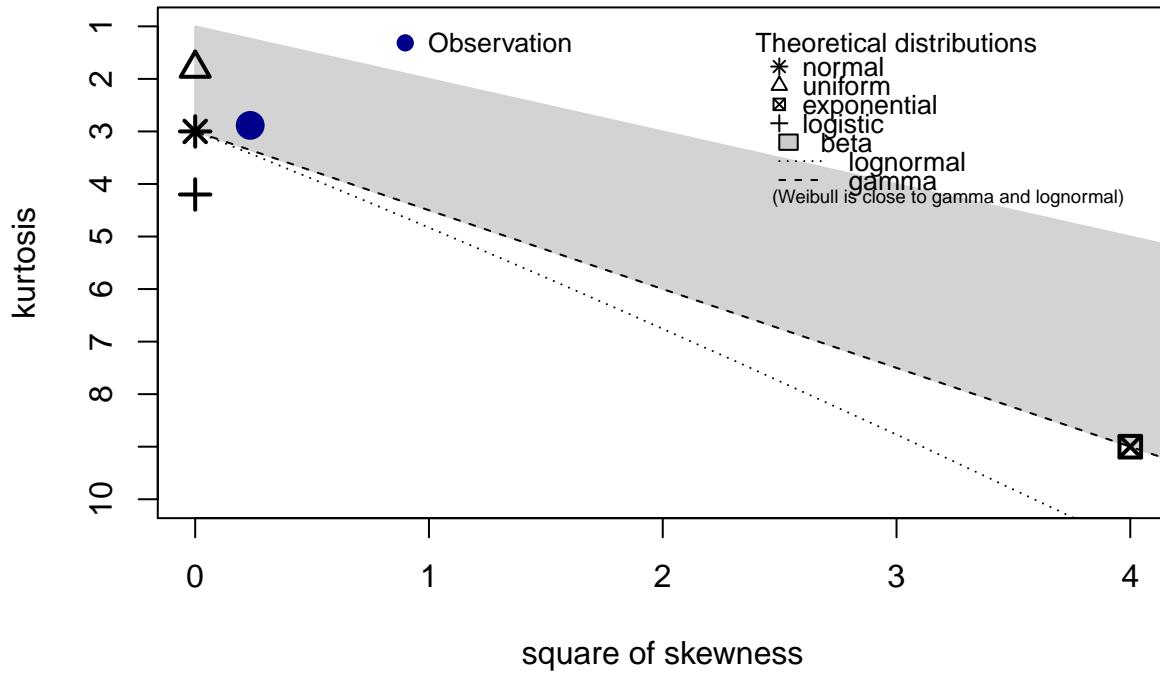


```
ggplot(umbs_sd_plot_growthhabit, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
  facet_wrap(~year + plot)
```



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

## Cullen and Frey graph



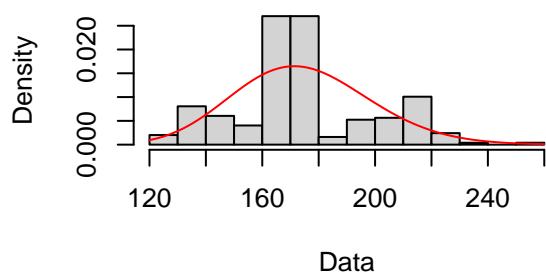
```

## summary statistics
## -----
## min: 127   max: 251
## median: 172
## mean: 174.7581
## estimated sd: 24.71714
## estimated skewness: 0.4851066
## estimated kurtosis: 2.884973

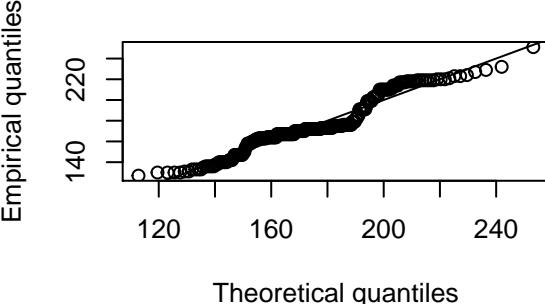
# Gamma distribution
fit.gamma <- fitdist(umbs_sd_plot_growthhabit$julian_min, "gamma")
plot(fit.gamma)

```

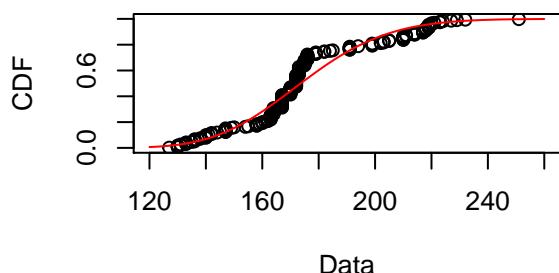
**Empirical and theoretical dens.**



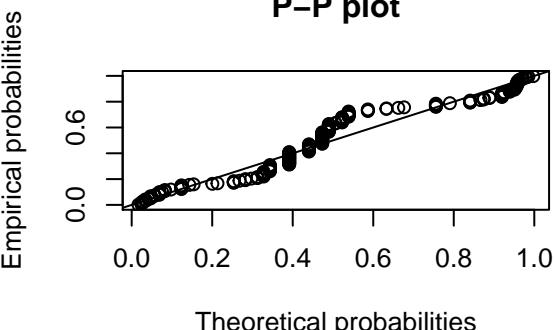
**Q-Q plot**



**Empirical and theoretical CDFs**

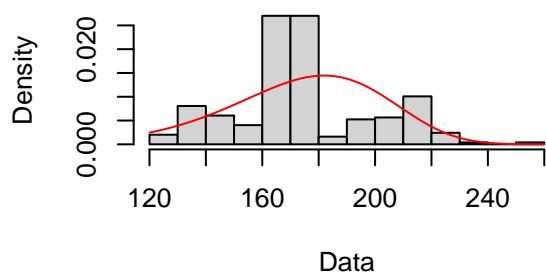


**P-P plot**

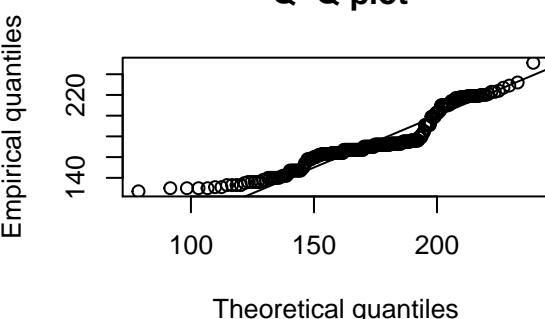


```
# Weibull distribution
fit.weibull <- fitdist(umbs_sd_plot_growthhabit$julian_min, "weibull")
plot(fit.weibull)
```

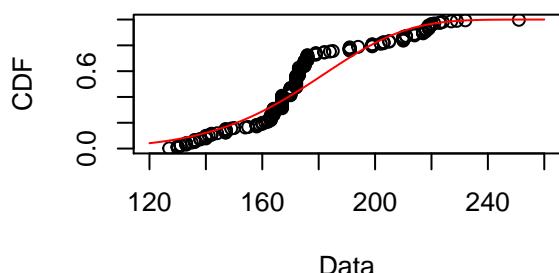
**Empirical and theoretical dens.**



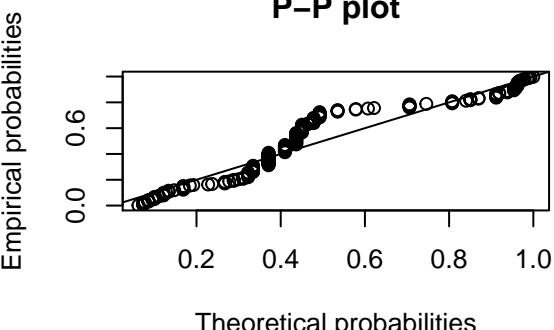
**Q-Q plot**



**Empirical and theoretical CDFs**

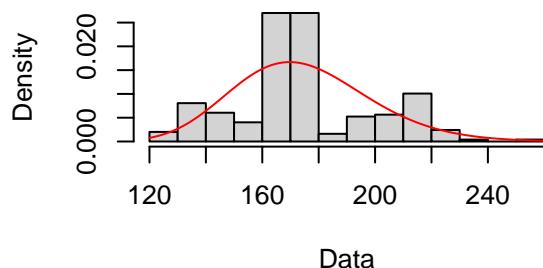


**P-P plot**

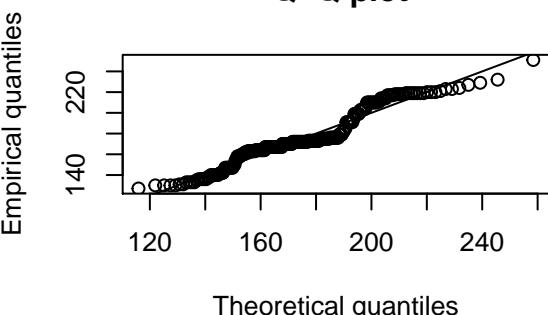


```
# Lognormal distribution
fit.ln <- fitdist(umbs_sd_plot_growthhabit$julian_min, "lnorm")
plot(fit.ln)
```

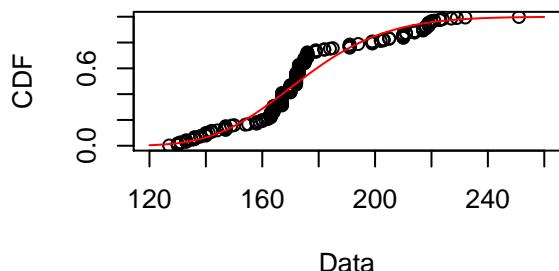
**Empirical and theoretical dens.**



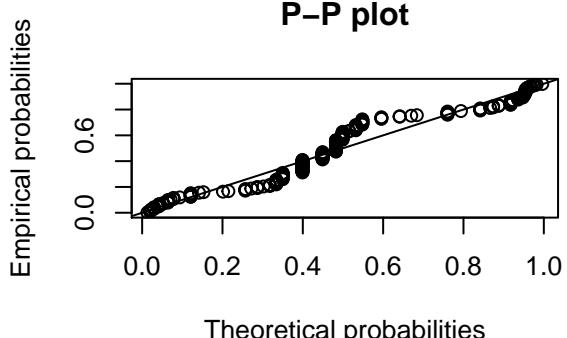
**Q-Q plot**



**Empirical and theoretical CDFs**

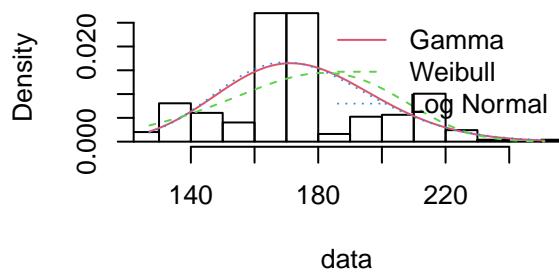


**P-P plot**

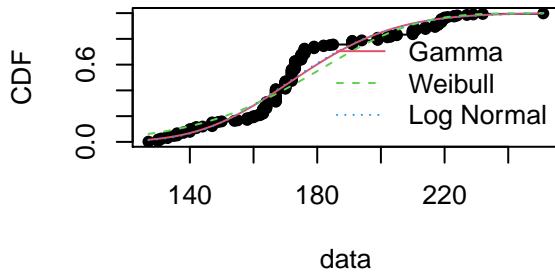


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

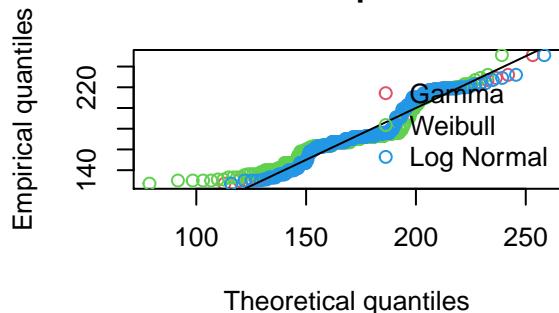
### Histogram and theoretical densities



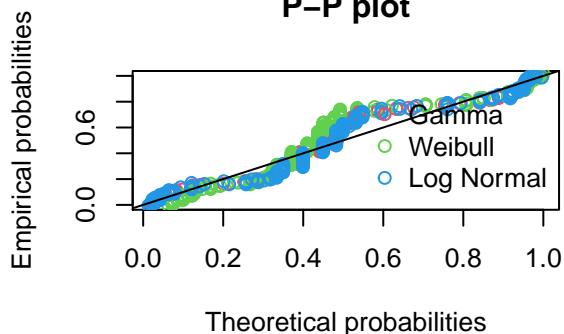
### Empirical and theoretical CDFs



### 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 0.1870698 0.2335321 0.1776961
## Cramer-von Mises statistic   1.3110891 2.1292901 1.2248073
## Anderson-Darling statistic   6.6239300 10.5105885 6.2406073
##
## Goodness-of-fit criteria
##                                     Gamma   Weibull Log Normal
## Akaike's Information Criterion 2289.017 2328.736 2286.855
## Bayesian Information Criterion 2296.044 2335.763 2293.882
```

```
# Log normal looks like it's the best fit
```

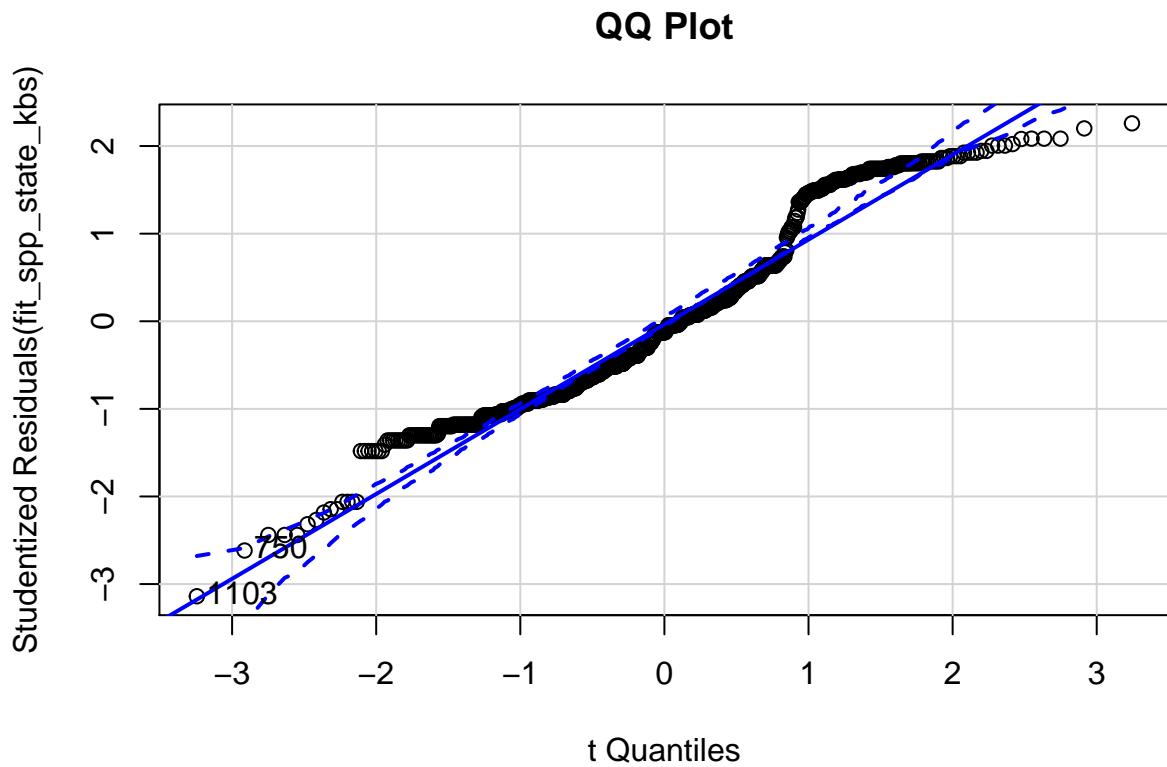
Leverage plots and detecting Outliers. <https://www.statmethods.net/stats/rdiagnostics.html>

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
## 1103 -3.139037      0.0017562        NA
```

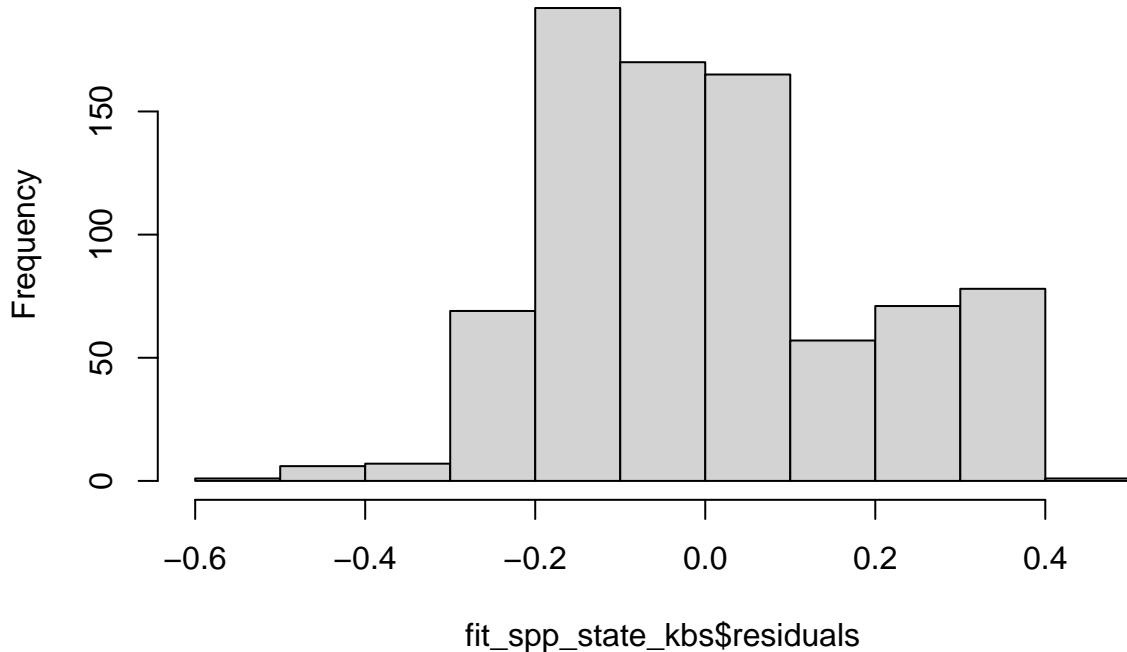
```
qqPlot(fit_spp_state_kbs, main = "QQ Plot")
```



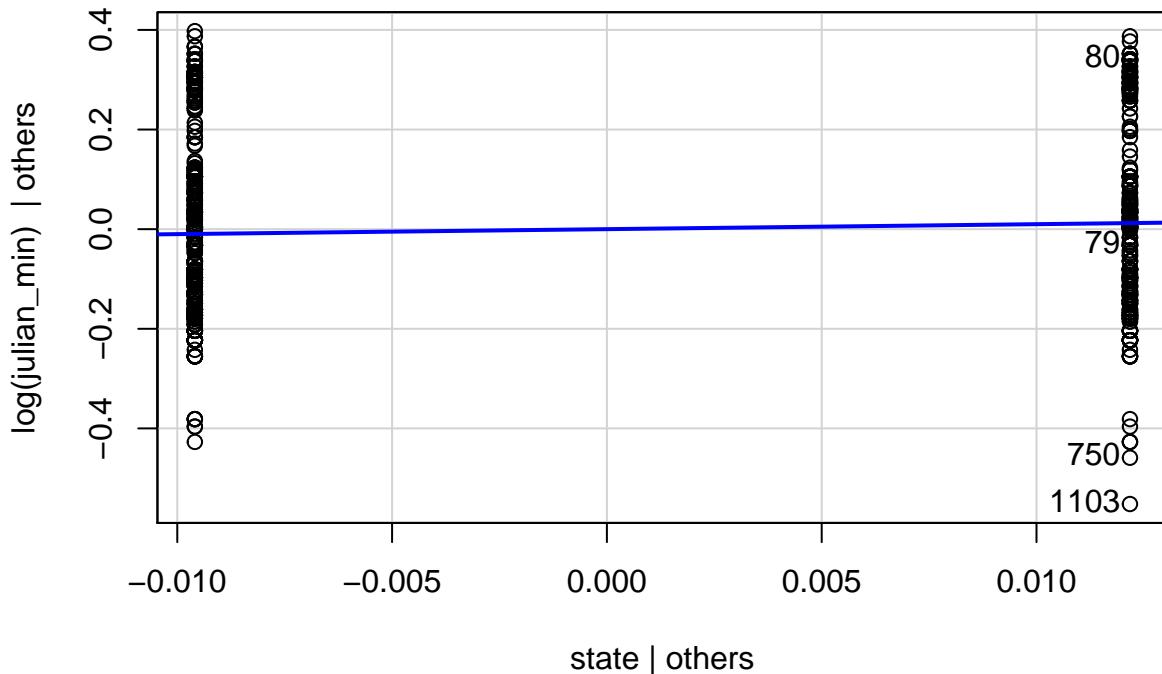
```
##  750 1103
##  433 642
```

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

**Histogram of fit\_spp\_state\_kbs\$residuals**



```
leveragePlots(fit_spp_state_kbs)
```



```
ols_test_normality(fit_spp_state_kbs)
```

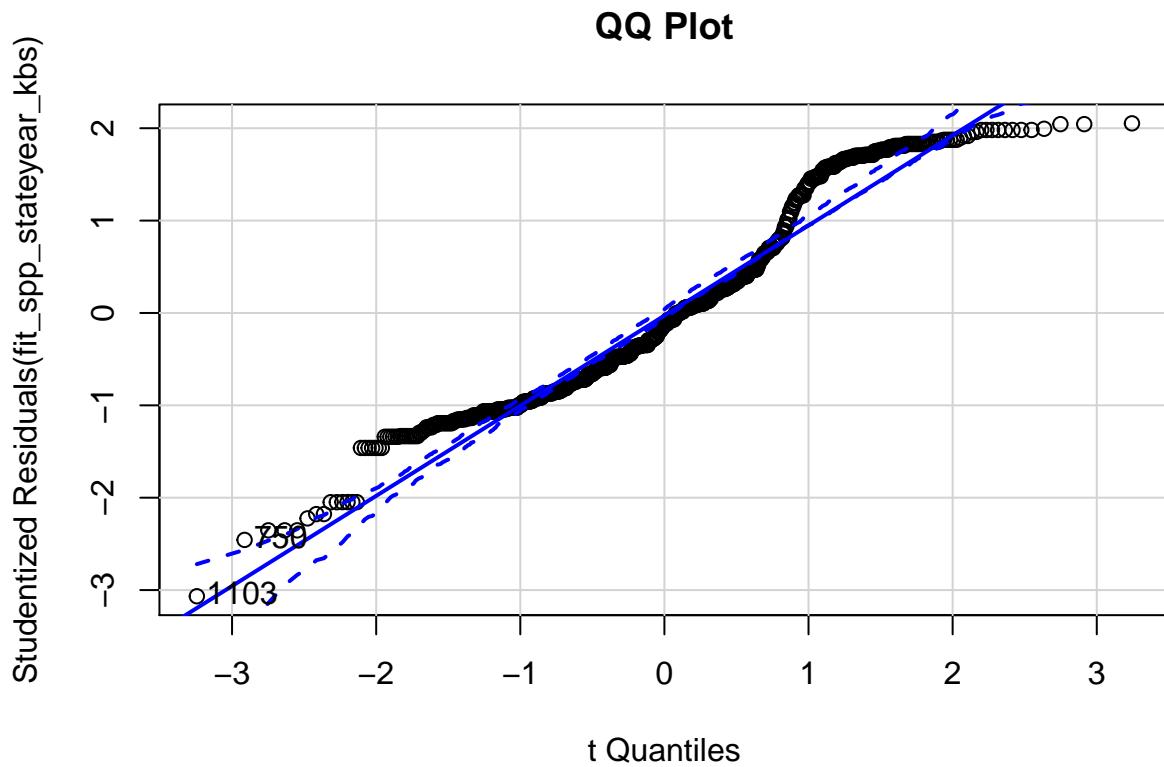
```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present for
## the Kolmogorov-Smirnov test
```

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

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

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

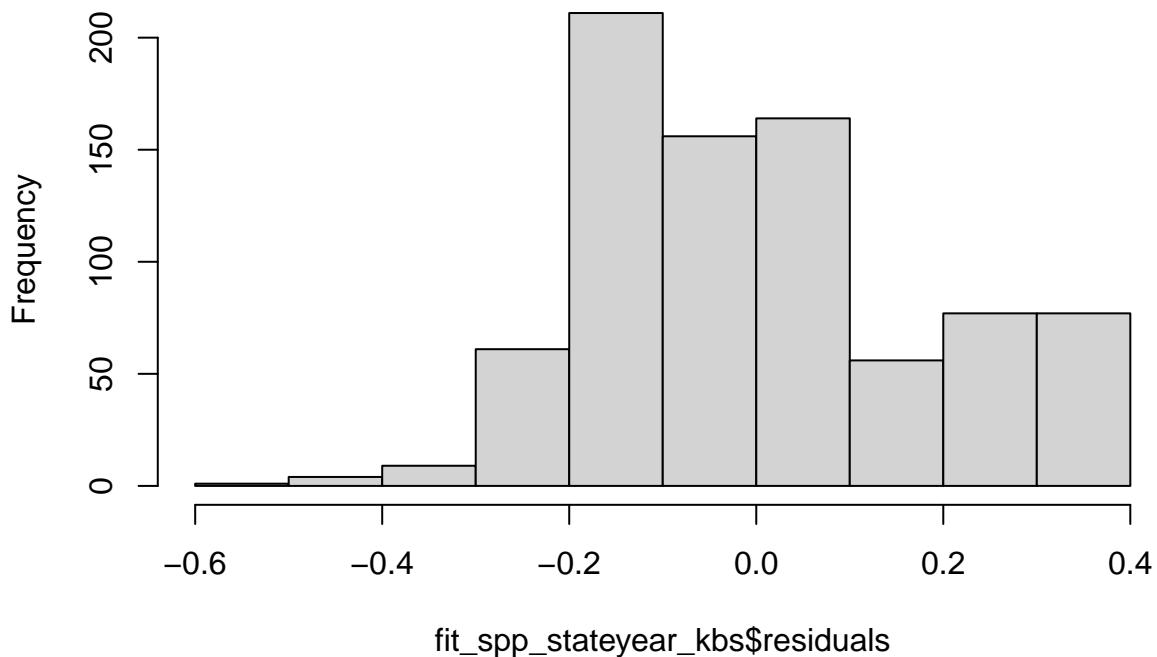
```
qqPlot(fit_spp_stateyear_kbs, main = "QQ Plot")
```



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

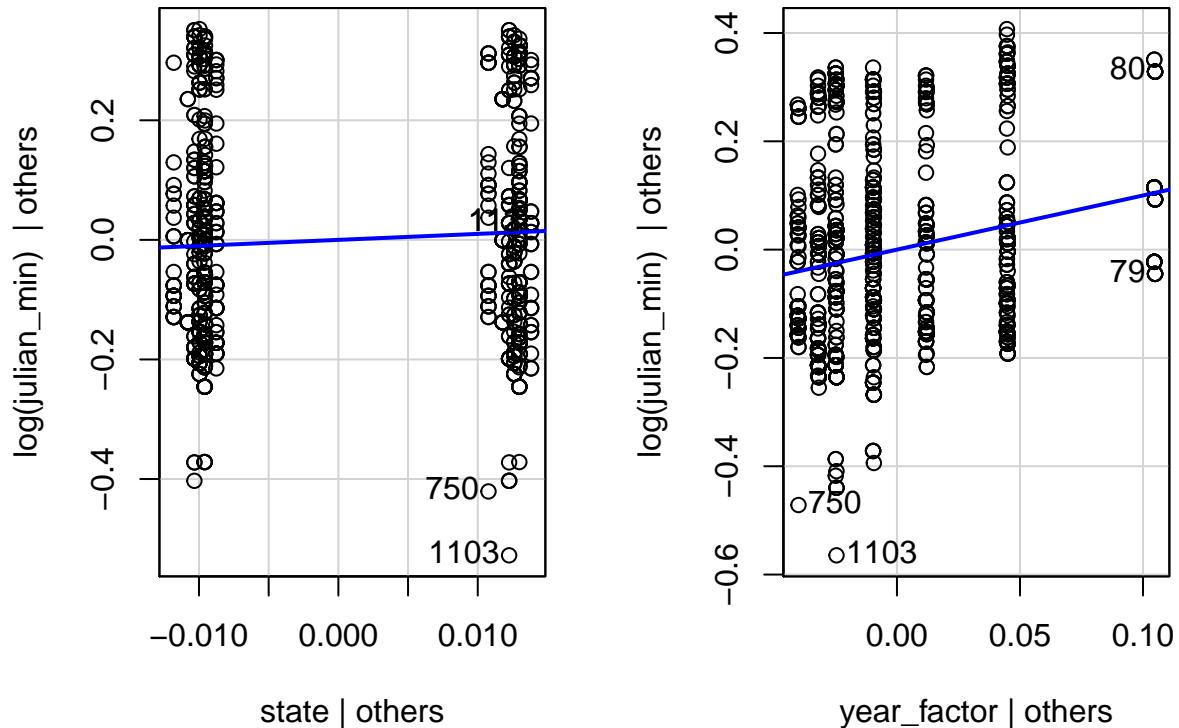
```
hist(fit_spp_stateyear_kbs$residuals)
```

**Histogram of fit\_spp\_stateyear\_kbs\$residuals**



```
leveragePlots(fit_spp_stateyear_kbs)
```

## Leverage Plots



```
ols_test_normality(fit_spp_stateyear_kbs)
```

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

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

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

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

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

```

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

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

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

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

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

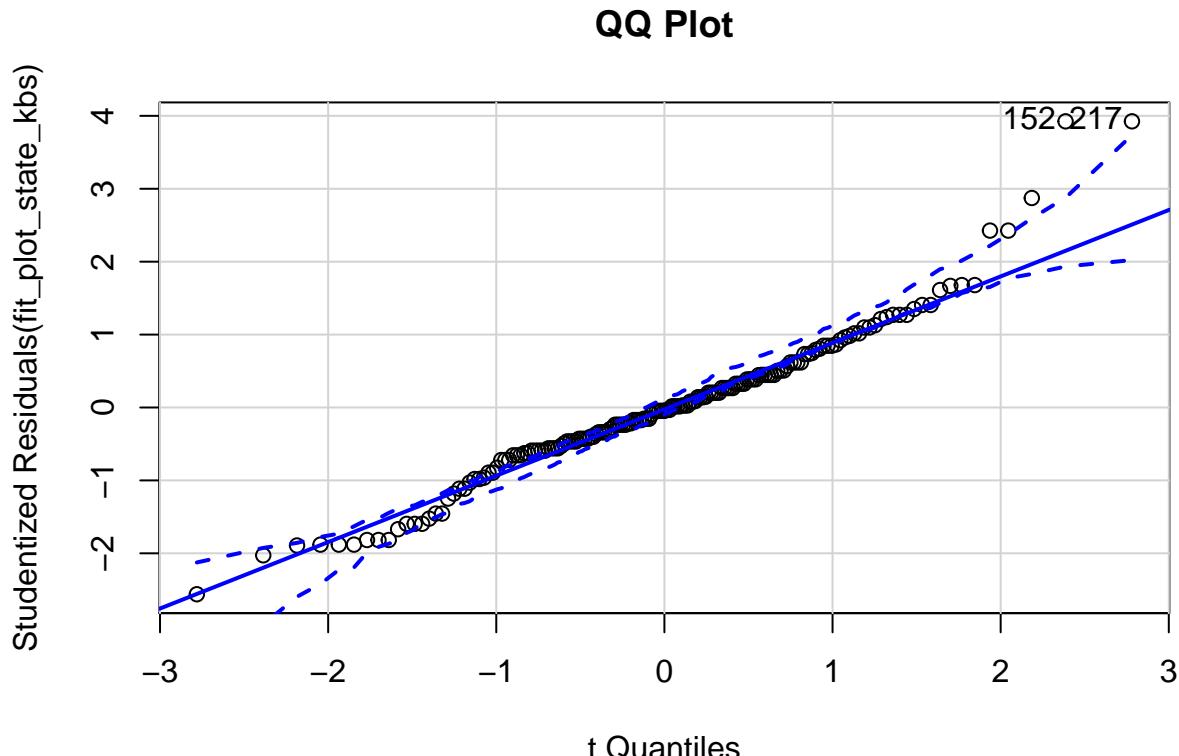
## Error in interact_plot(fit4, pred = year_factor, modx = state, mod2 = species): Focal predictor ("pred")

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

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

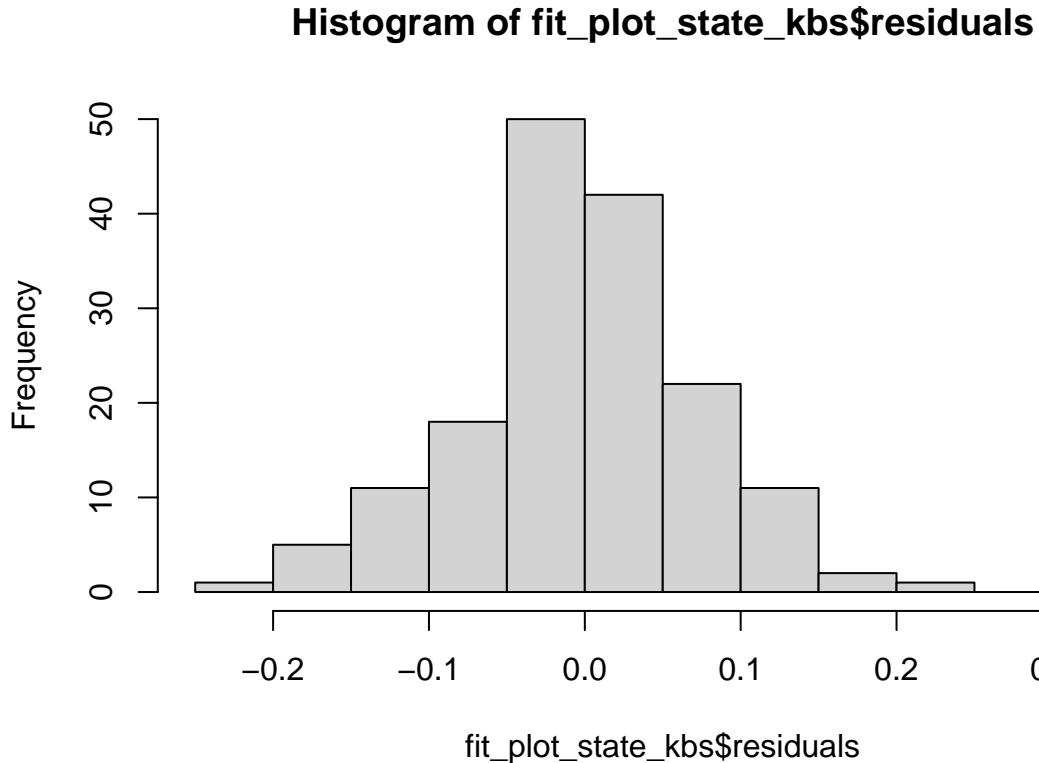
qqPlot(fit_plot_state_kbs, main = "QQ Plot")

```

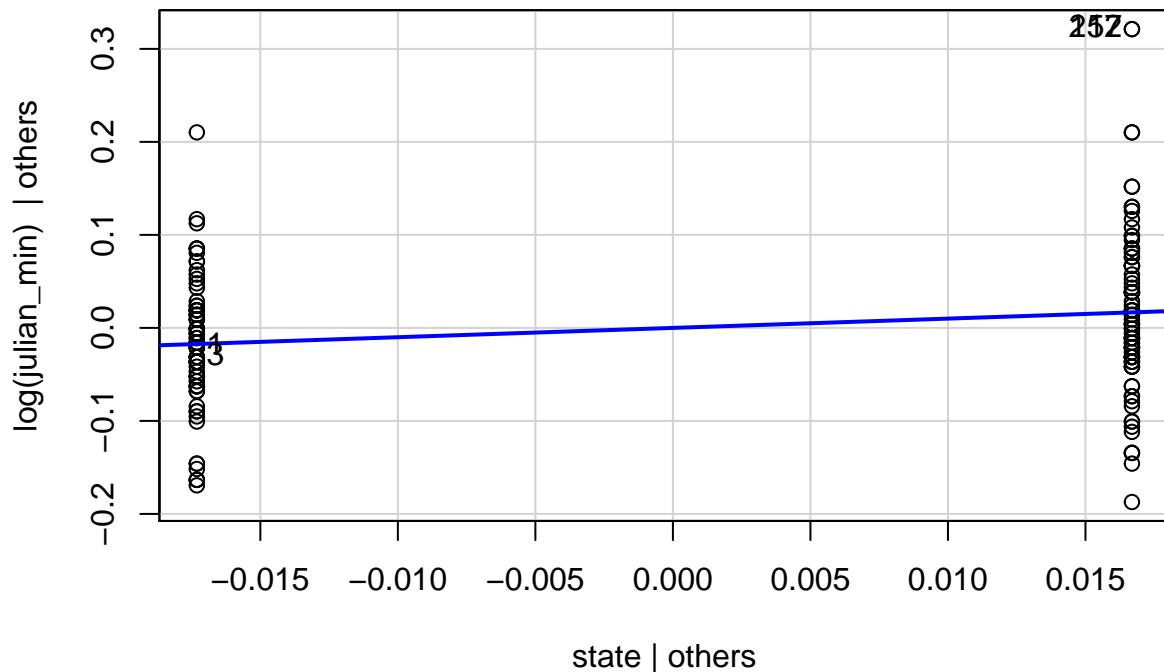


```
## 152 217  
## 84 119
```

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



```
leveragePlots(fit_plot_state_kbs)
```



```

ols_test_normality(fit_plot_state_kbs)

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

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

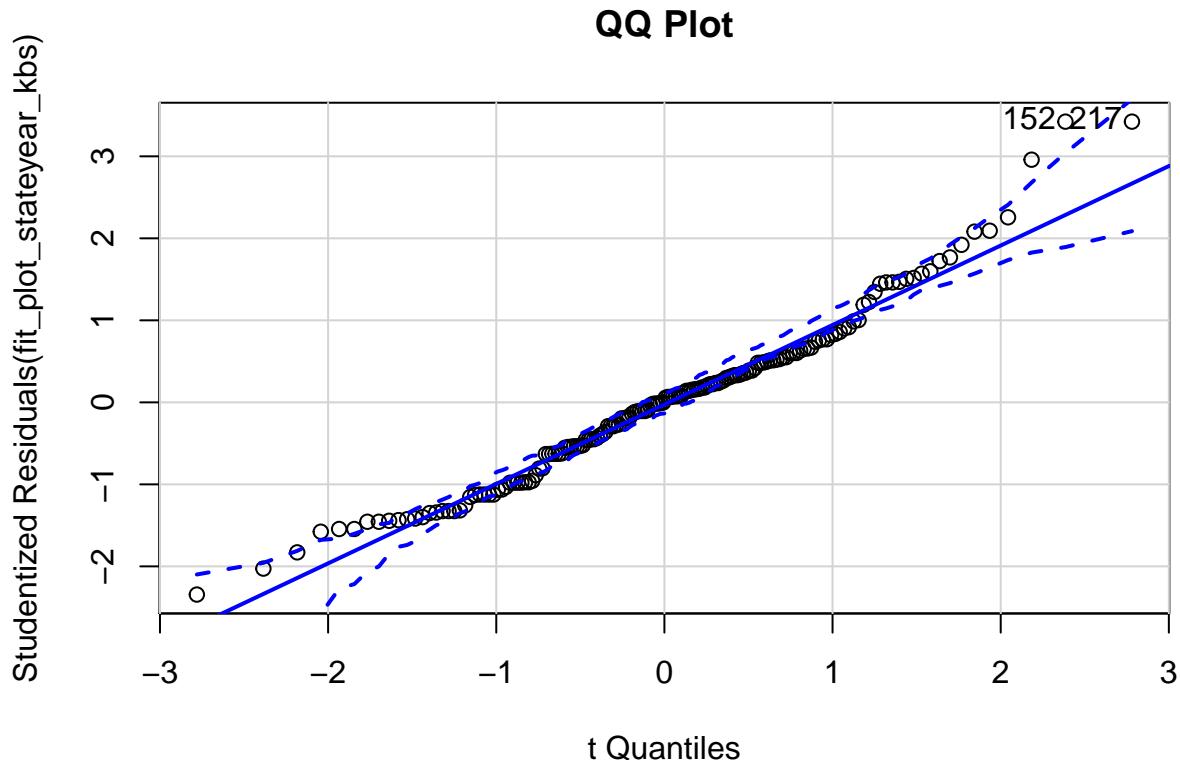
```

```

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

```

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



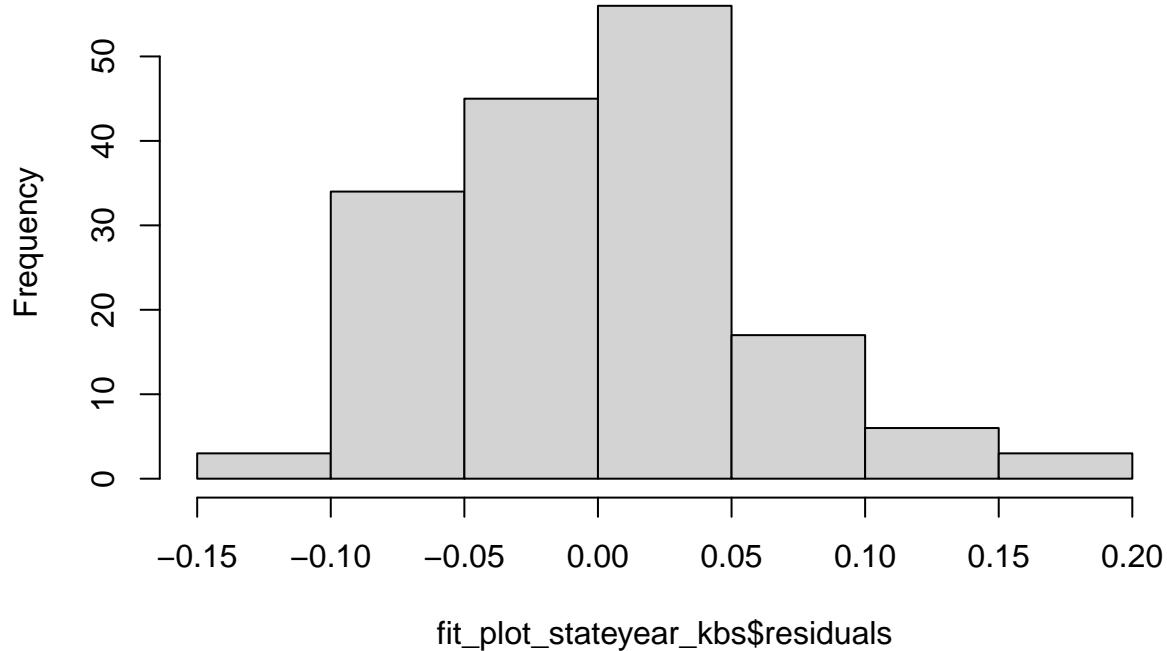
```

## 152 217
## 84 119

```

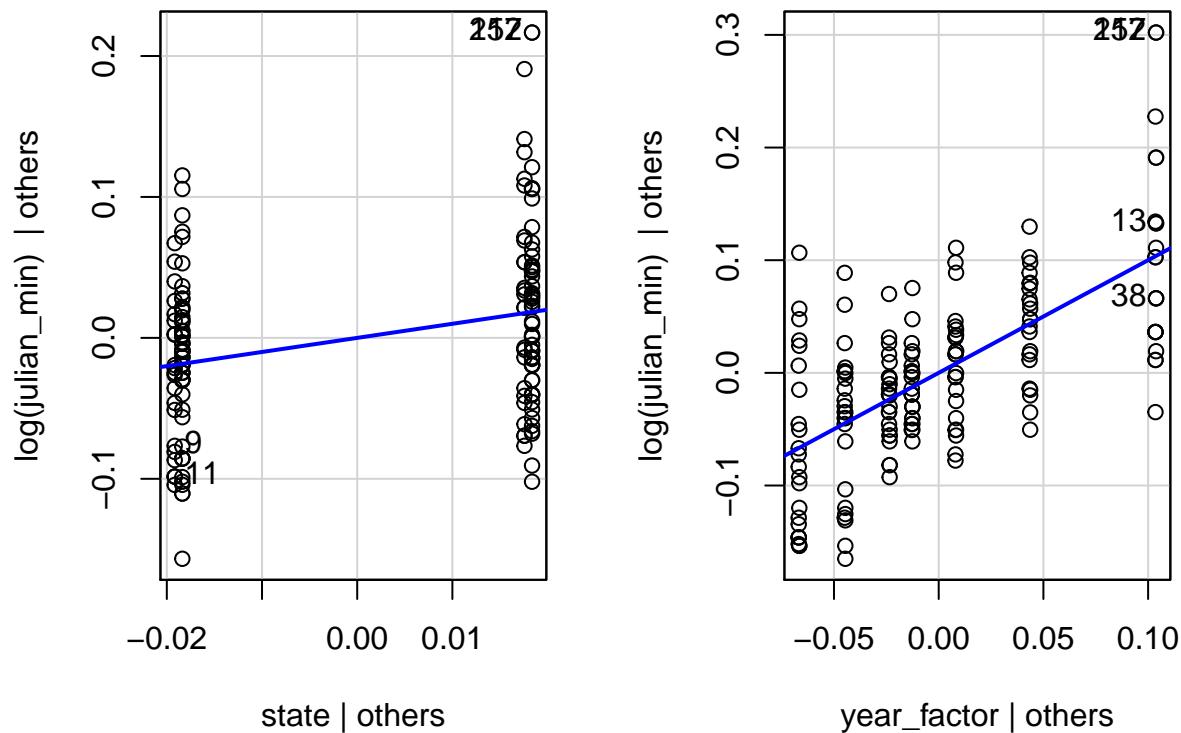
```
hist(fit_plot_stateyear_kbs$residuals)
```

**Histogram of fit\_plot\_stateyear\_kbs\$residuals**



```
leveragePlots(fit_plot_stateyear_kbs)
```

## Leverage Plots



```
ols_test_normality(fit_plot_stateyear_kbs)
```

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

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

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

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

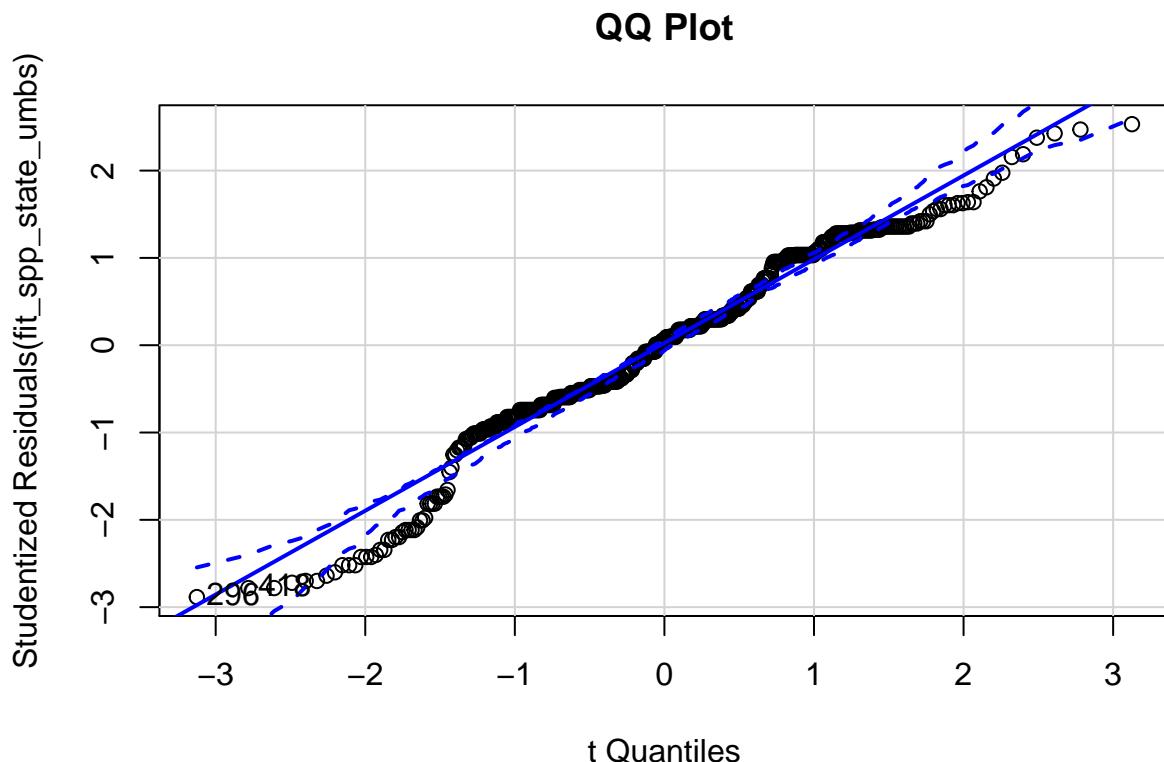
```
## Using data kbs_sd_plot from global environment. This could cause incorrect
## results if kbs_sd_plot has been altered since the model was fit. You can
## manually provide the data to the "data =" argument.
```

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

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

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferroni p
## 296 -2.8848          0.0040754        NA
```

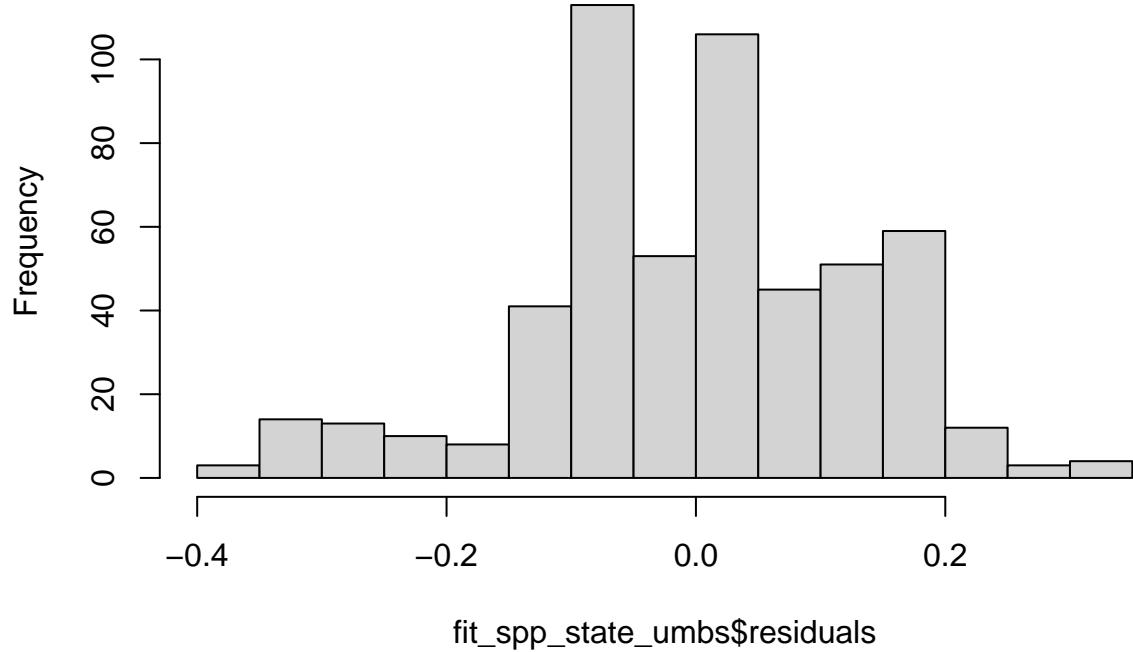
```
qqPlot(fit_spp_state_umbs, main = "QQ Plot")
```



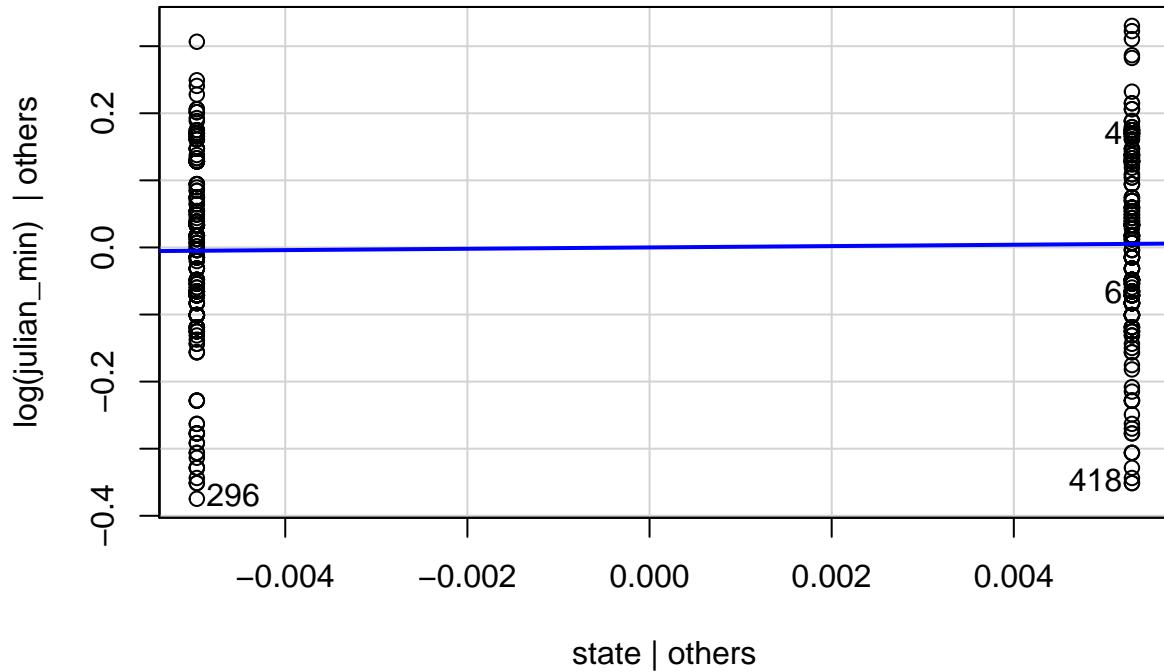
```
## 296 418
## 132 180
```

```
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

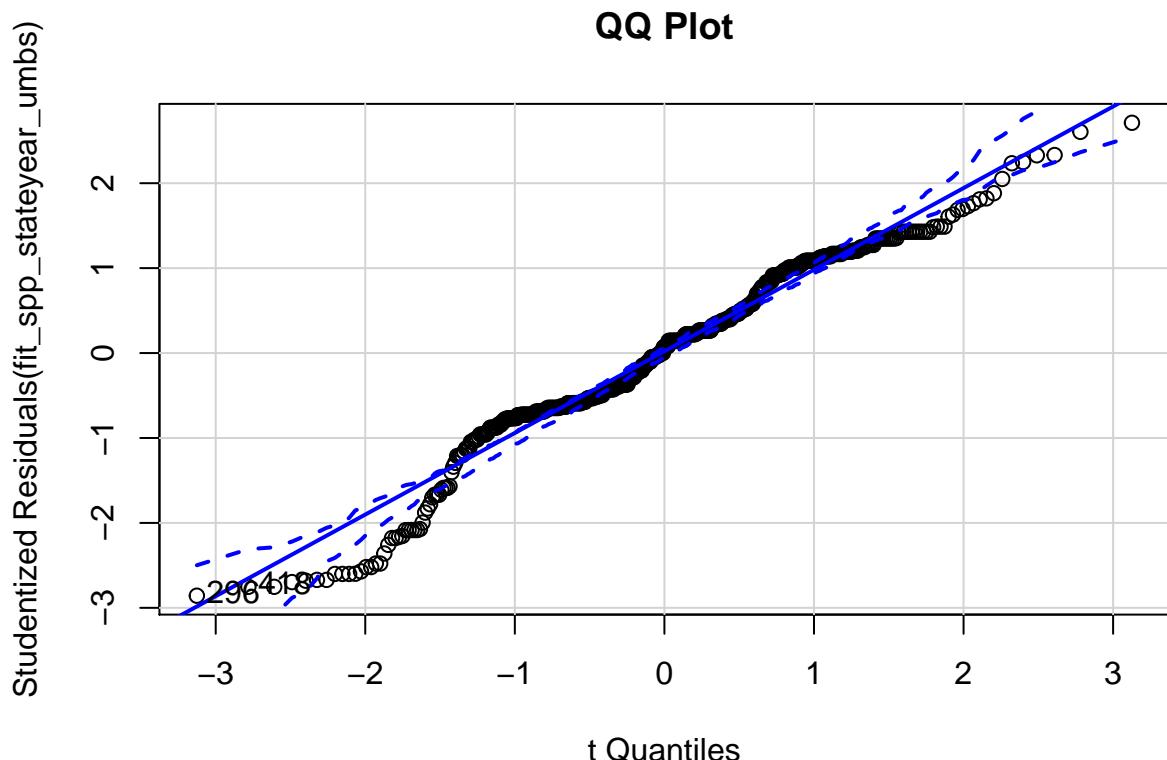
```

```

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##           rstudent unadjusted p-value Bonferroni p
## 296 -2.856926          0.0044463        NA

```

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



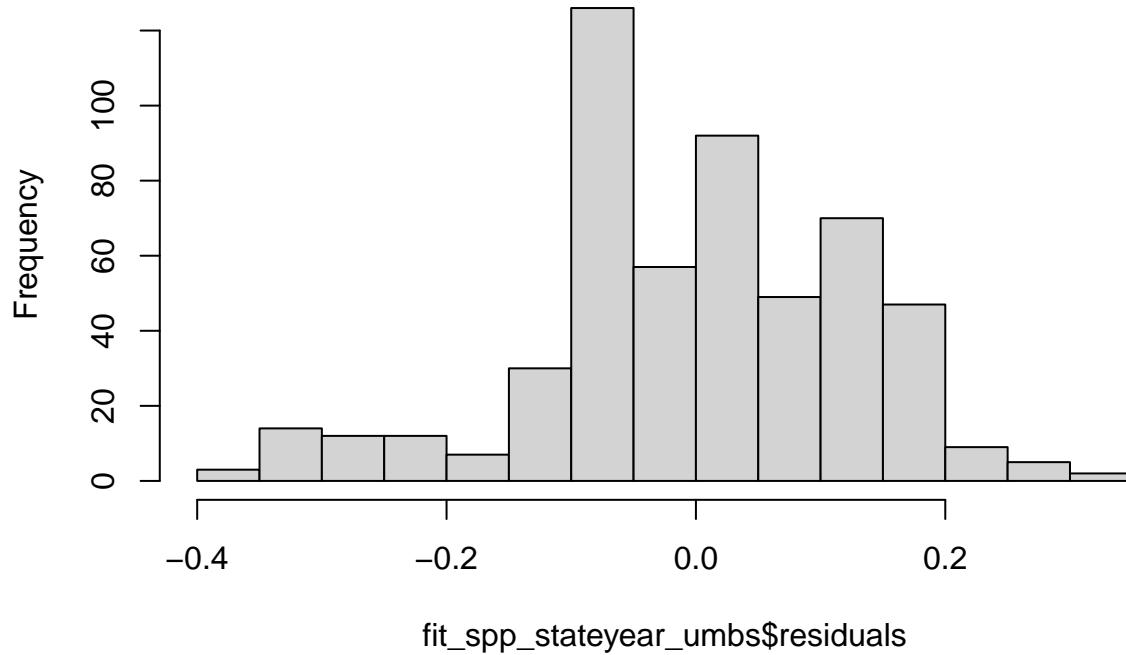
```

## 296 418
## 132 180

```

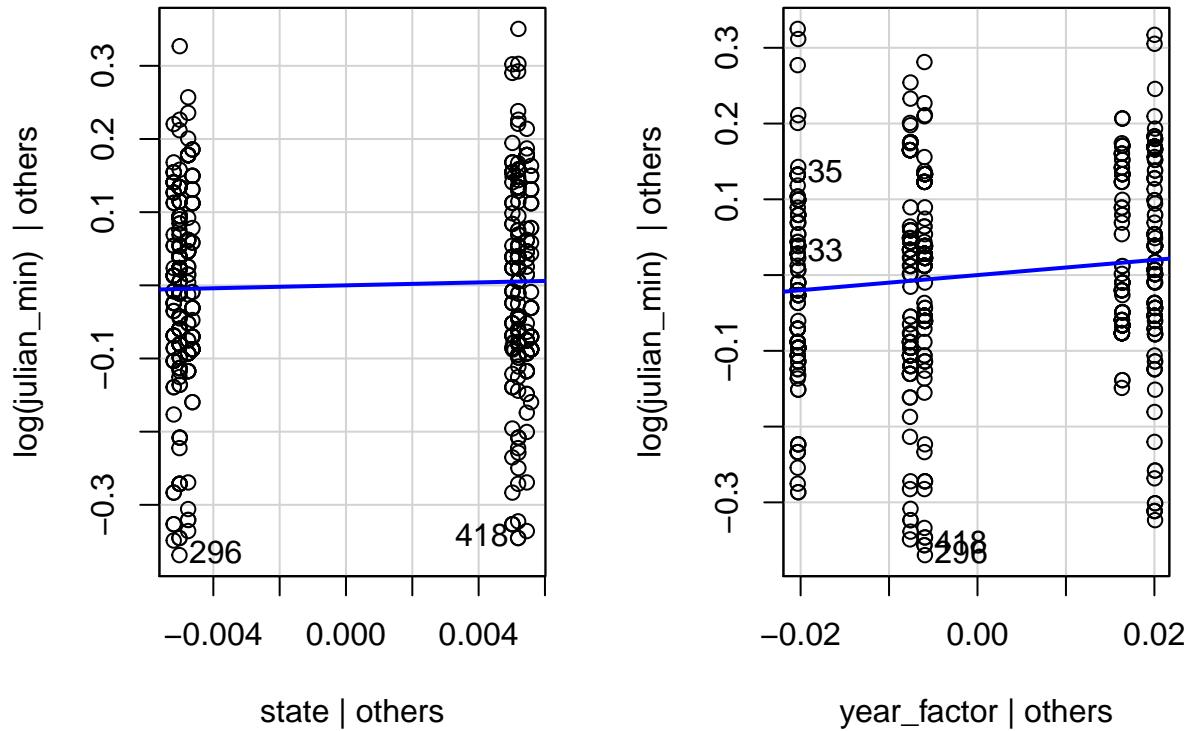
```
hist(fit_spp_stateyear_umbs$residuals)
```

### 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.

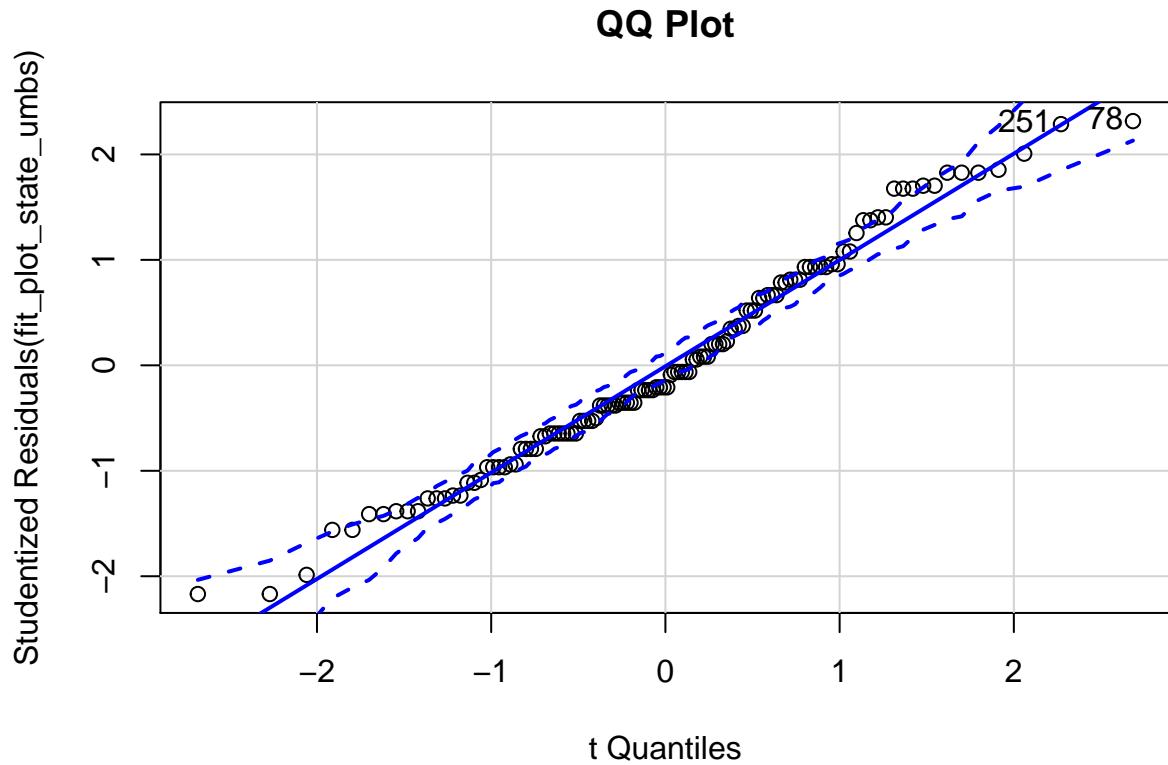
## Error in interact_plot(fit4, pred = year_factor, modx = state, mod2 = species): Focal predictor ("pr

# 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
## 78  2.315735        0.022315       NA

```

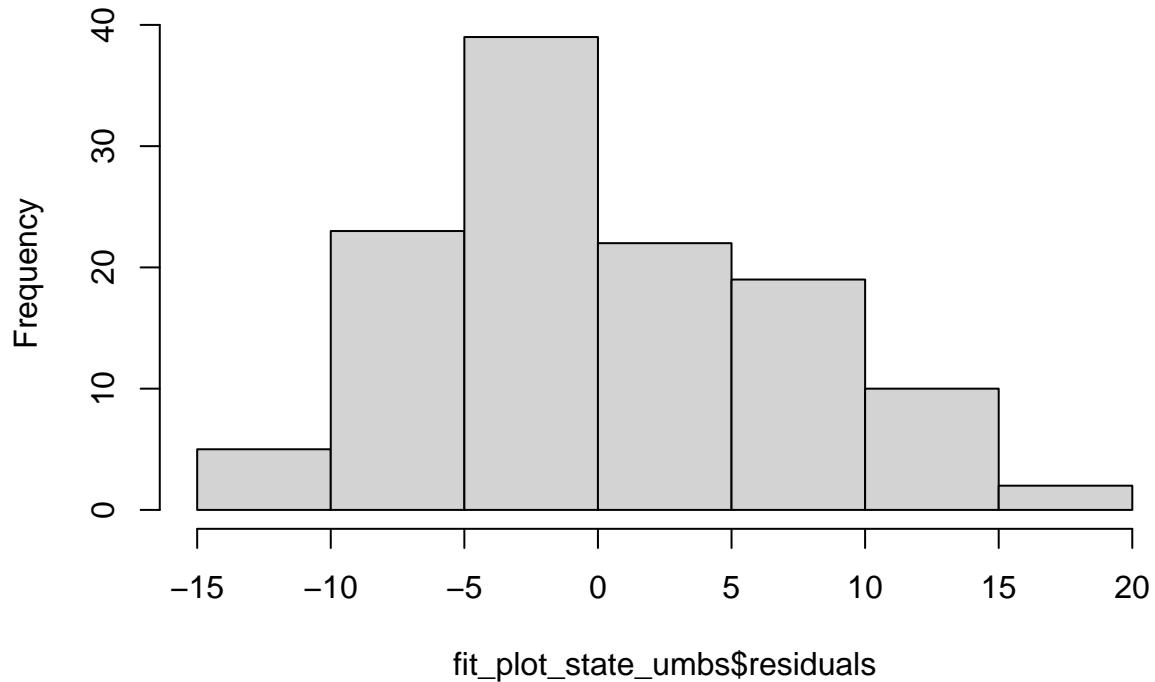
```
qqPlot(fit_plot_state_umbs, main = "QQ Plot")
```



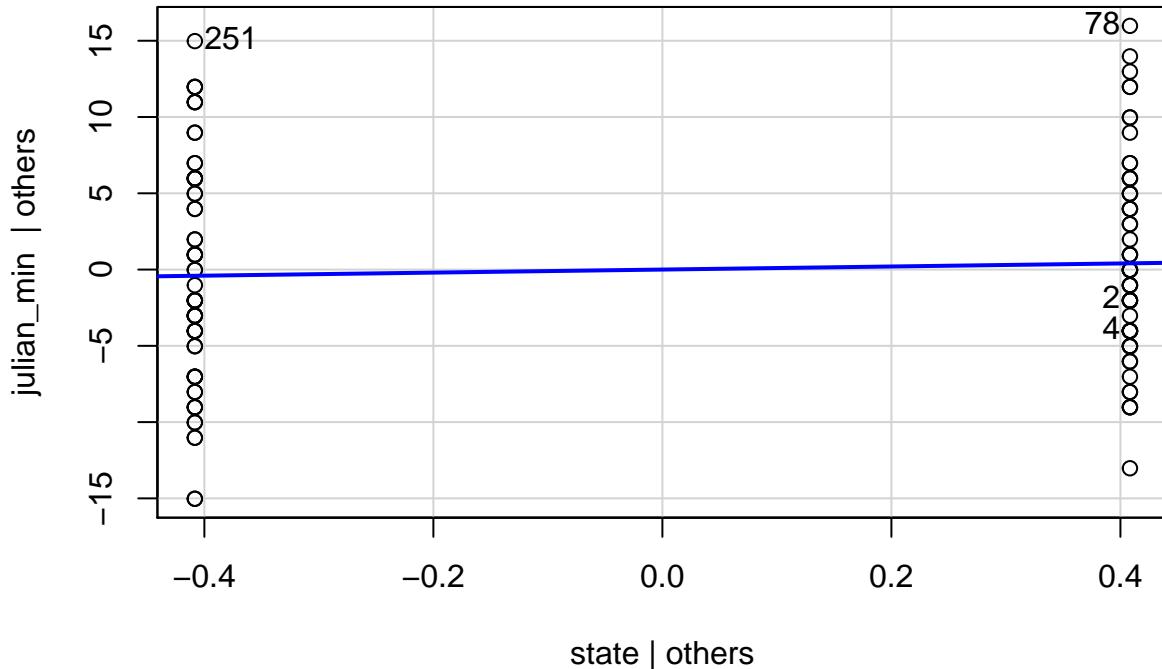
```
## 78 251  
## 31 99
```

```
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)

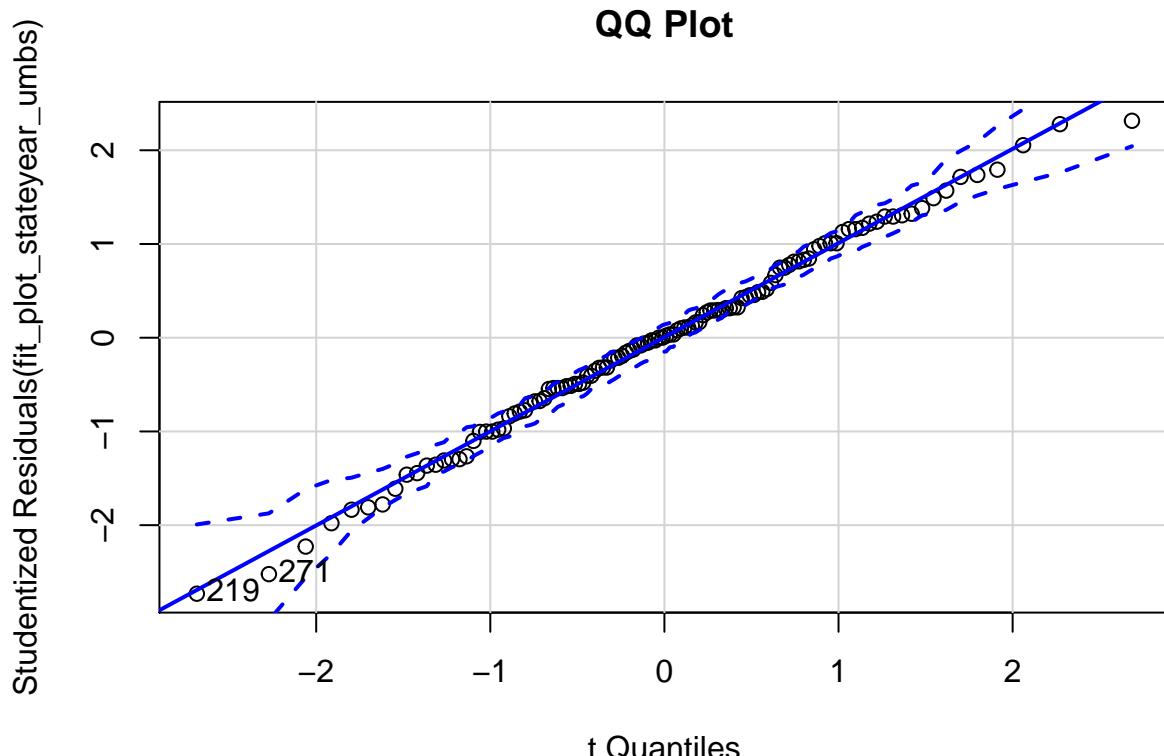
```

```

## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##           rstudent unadjusted p-value Bonferroni p
## 219 -2.731017        0.0073274       0.87928

```

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



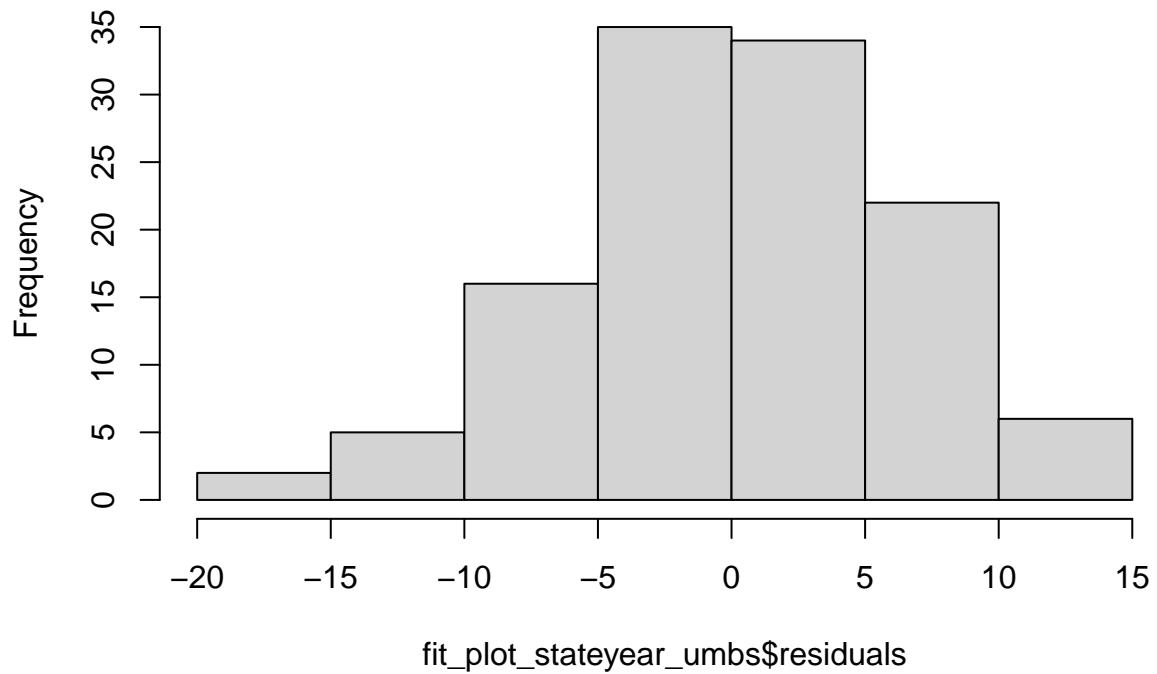
```

## 219 271
## 86 106

```

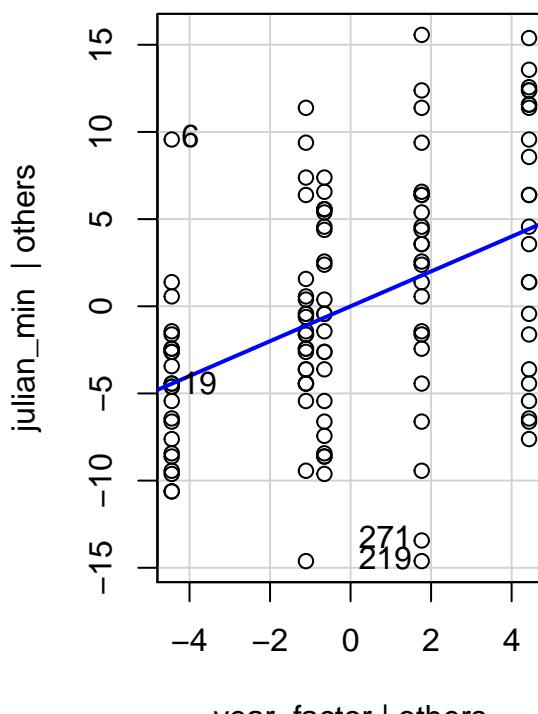
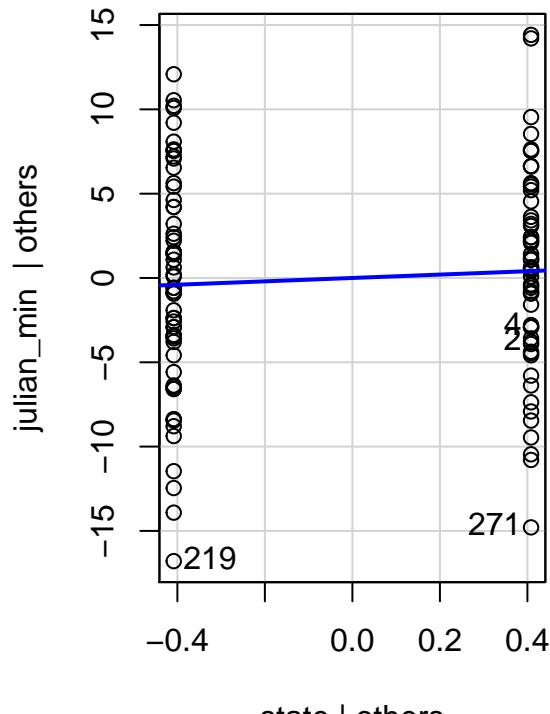
```
hist(fit_plot_stateyear_umbs$residuals)
```

### 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)
```

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
## state	0.00671	0.006706	1	794.06	2.8796	0.0901 .
## year_factor	0.72591	0.120986	6	794.41	51.9480	<2e-16 ***
## insecticide	0.00474	0.004742	1	794.09	2.0359	0.1540

```

## state:year_factor      0.00928 0.001546      6 794.03  0.6640 0.6789
## year_factor:insecticide 0.01238 0.002063      6 794.06  0.8857 0.5048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
anova(mod2)
```

```

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

```

# Run an ANOVA to test the 2 models to test whether the more complex model is significantly better at capturing the data than the simpler model. If the resulting p-value is sufficiently low (usually less than 0.05), we conclude that the more complex model is significantly better than the simpler model, and thus favor the more complex model. If the p-value is not sufficiently low (usually greater than 0.05), we should favor the simpler model.

```
# https://bookdown.org/ndphillips/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 -2458.1 -2349.9 1252.1 -2504.1
## mod1  24 -2456.1 -2343.2 1252.1 -2504.1      0  1          1

```

```
summary(mod1)
```

```

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state * year_factor + insecticide * year_factor +
##           (1 | species) + (1 | plot)
## Data: kbs_sd_spp
##
##       AIC      BIC   logLik deviance df.resid
##   -2456.1  -2343.2   1252.1  -2504.1      792
##
## Scaled residuals:
##       Min     1Q Median     3Q    Max
## -4.5749 -0.5062 -0.0168  0.4582  7.6160
##
## Random effects:

```

```

## Groups      Name          Variance Std.Dev.
## plot        (Intercept) 0.0000000 0.00000
## species     (Intercept) 0.040350 0.20087
## Residual    0.002329 0.04826
## Number of obs: 816, groups: plot, 24; species, 22
##
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                5.36220  0.04483 25.74146 119.607
## stateambient               0.02061  0.01433 793.79819  1.438
## year_factor2              -0.09614  0.01421 793.86859 -6.768
## year_factor3              -0.10098  0.01466 794.21173 -6.887
## year_factor4              -0.10740  0.01515 793.88740 -7.088
## year_factor5              -0.07735  0.01543 793.82667 -5.014
## year_factor6              -0.06531  0.01461 793.84647 -4.470
## year_factor7              -0.07777  0.01661 793.88823 -4.683
## insecticideno_insects     0.03017  0.01427 793.79100  2.114
## stateambient:year_factor2 -0.02182  0.01594 793.83440 -1.369
## stateambient:year_factor3 -0.01223  0.01638 793.87744 -0.747
## stateambient:year_factor4 -0.02685  0.01700 793.95450 -1.580
## stateambient:year_factor5 -0.01148  0.01751 793.81399 -0.656
## stateambient:year_factor6 -0.01401  0.01666 793.96220 -0.841
## stateambient:year_factor7 -0.01151  0.01871 793.82712 -0.615
## year_factor2:insecticideno_insects -0.03334  0.01584 793.79868 -2.105
## year_factor3:insecticideno_insects -0.03443  0.01634 793.99170 -2.107
## year_factor4:insecticideno_insects -0.02605  0.01686 794.01578 -1.545
## year_factor5:insecticideno_insects -0.02812  0.01734 793.80149 -1.622
## year_factor6:insecticideno_insects -0.02782  0.01656 793.81823 -1.680
## year_factor7:insecticideno_insects -0.02310  0.01887 793.87303 -1.224
##                               Pr(>|t|)
## (Intercept)                < 2e-16 ***
## stateambient               0.1507
## year_factor2              2.53e-11 ***
## year_factor3              1.16e-11 ***
## year_factor4              3.01e-12 ***
## year_factor5              6.59e-07 ***
## year_factor6              8.95e-06 ***
## year_factor7              3.32e-06 ***
## insecticideno_insects     0.0348 *
## stateambient:year_factor2 0.1714
## stateambient:year_factor3 0.4555
## stateambient:year_factor4 0.1146
## stateambient:year_factor5 0.5123
## stateambient:year_factor6 0.4006
## stateambient:year_factor7 0.5388
## year_factor2:insecticideno_insects 0.0356 *
## year_factor3:insecticideno_insects 0.0354 *
## year_factor4:insecticideno_insects 0.1228
## year_factor5:insecticideno_insects 0.1053
## year_factor6:insecticideno_insects 0.0934 .
## year_factor7:insecticideno_insects 0.2212
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

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

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

summary(mod2)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log(julian_min) ~ state * year_factor + insecticide * year_factor +
##           (1 | species)
## Data: kbs_sd_spp
##
##      AIC      BIC  logLik deviance df.resid
## -2458.1 -2349.9  1252.1 -2504.1     793
## 
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.5749 -0.5062 -0.0168  0.4582  7.6160
## 
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   species (Intercept) 0.040350 0.20087
##   Residual            0.002329 0.04826
## Number of obs: 816, groups: species, 22
## 
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                5.36220  0.04483 25.74146 119.607
## stateambient               0.02061  0.01433 793.79818   1.438
## year_factor2              -0.09614  0.01421 793.86859  -6.768
## year_factor3              -0.10098  0.01466 794.21173  -6.887
## year_factor4              -0.10740  0.01515 793.88740  -7.088
## year_factor5              -0.07735  0.01543 793.82667  -5.014
## year_factor6              -0.06531  0.01461 793.84647  -4.470
## year_factor7              -0.07777  0.01661 793.88823  -4.683
## insecticideno_insects     0.03017  0.01427 793.79100   2.114
## stateambient:year_factor2 -0.02182  0.01594 793.83440  -1.369
## stateambient:year_factor3 -0.01223  0.01638 793.87744  -0.747
## stateambient:year_factor4 -0.02685  0.01700 793.95450  -1.580
## stateambient:year_factor5 -0.01148  0.01751 793.81399  -0.656
## stateambient:year_factor6 -0.01401  0.01666 793.96220  -0.841
## stateambient:year_factor7 -0.01151  0.01871 793.82712  -0.615
## year_factor2:insecticideno_insects -0.03334  0.01584 793.79868  -2.105
## year_factor3:insecticideno_insects -0.03443  0.01634 793.99170  -2.107
## year_factor4:insecticideno_insects -0.02605  0.01686 794.01578  -1.545
## year_factor5:insecticideno_insects -0.02812  0.01734 793.80149  -1.622
## year_factor6:insecticideno_insects -0.02782  0.01656 793.81823  -1.680
## year_factor7:insecticideno_insects -0.02310  0.01887 793.87303  -1.224
## Pr(>|t|)
```

```

## (Intercept) < 2e-16 ***
## stateambient 0.1507
## year_factor2 2.53e-11 ***
## year_factor3 1.16e-11 ***
## year_factor4 3.01e-12 ***
## year_factor5 6.59e-07 ***
## year_factor6 8.95e-06 ***
## year_factor7 3.32e-06 ***
## insecticideno_insects 0.0348 *
## stateambient:year_factor2 0.1714
## stateambient:year_factor3 0.4555
## stateambient:year_factor4 0.1146
## stateambient:year_factor5 0.5123
## stateambient:year_factor6 0.4006
## stateambient:year_factor7 0.5388
## year_factor2:insecticideno_insects 0.0356 *
## year_factor3:insecticideno_insects 0.0354 *
## year_factor4:insecticideno_insects 0.1228
## year_factor5:insecticideno_insects 0.1053
## year_factor6:insecticideno_insects 0.0934 .
## year_factor7:insecticideno_insects 0.2212
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

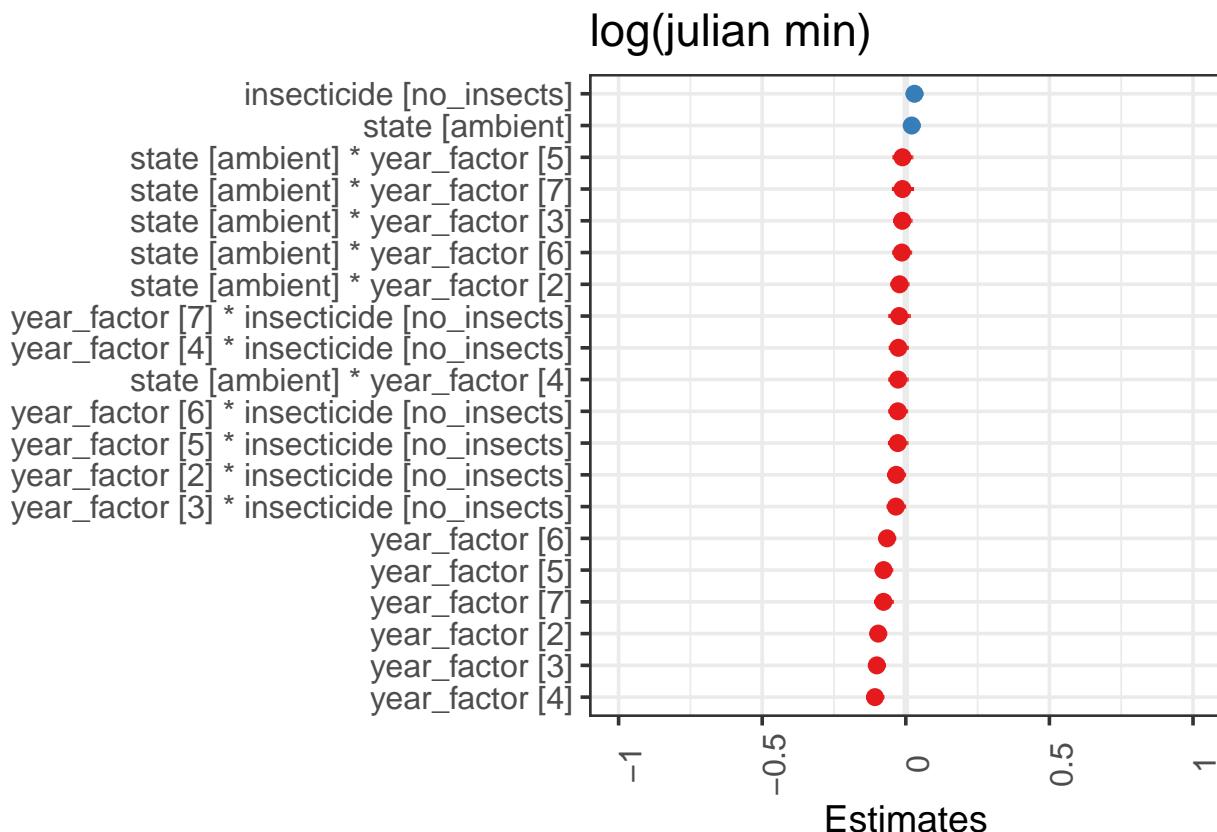
##
## Correlation matrix not shown by default, as p = 21 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)      if you need it

```

```

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

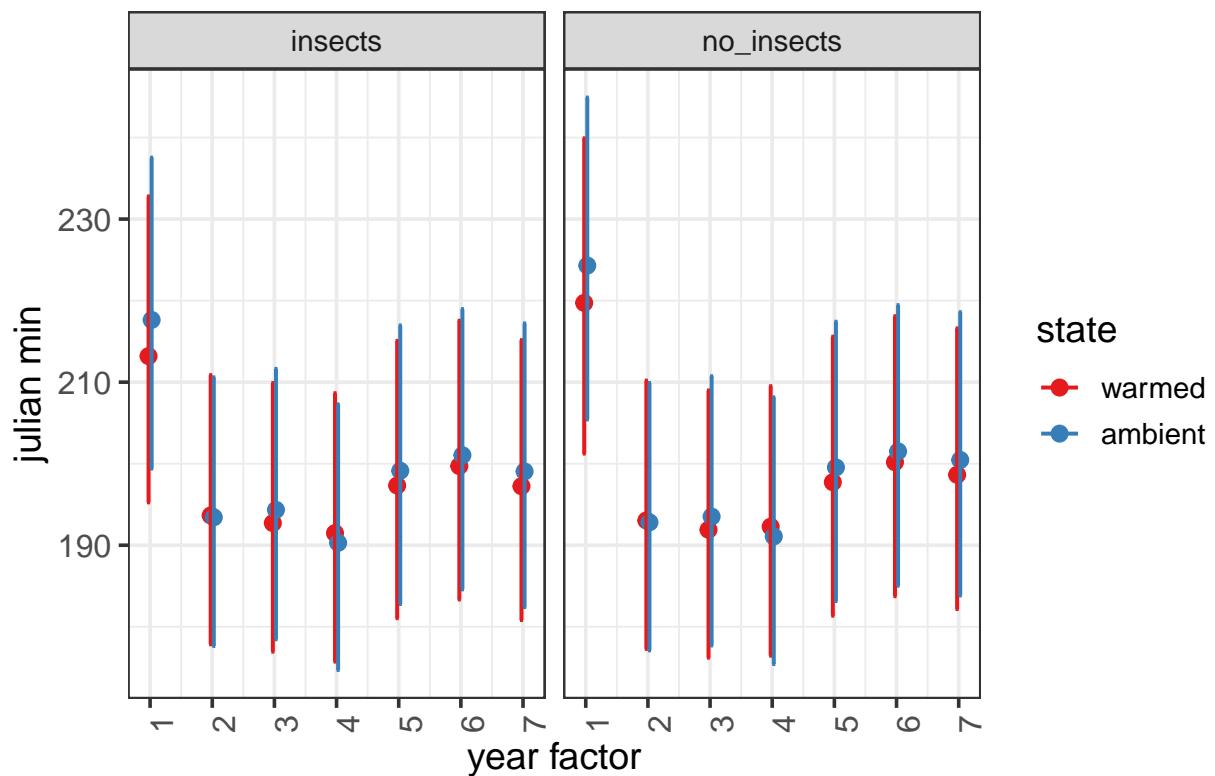
```



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

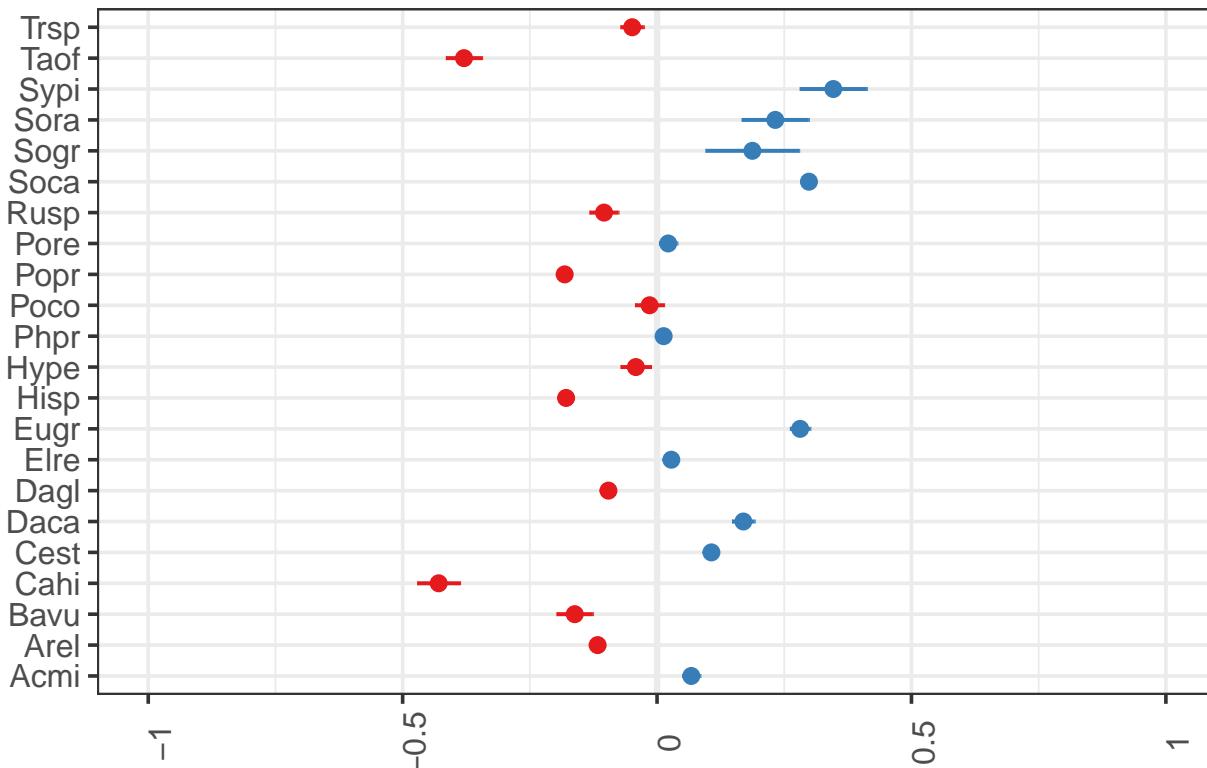
```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standardized
```

## Predicted values of julian min



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

## Random effects



```
# Do we need to include insecticide?
mod3 <- lmer(log(julian_min) ~ state * year_factor + (1 | species), kbs_sd_spp, REML = FALSE)
anova(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 +
##        (1 | species)
##          npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## mod3   16 -2466.5 -2391.2 1249.3   -2498.5
## mod2   23 -2458.1 -2349.9 1252.1   -2504.1 5.6243  7     0.5842
```

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

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

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

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

```

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

```

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

```

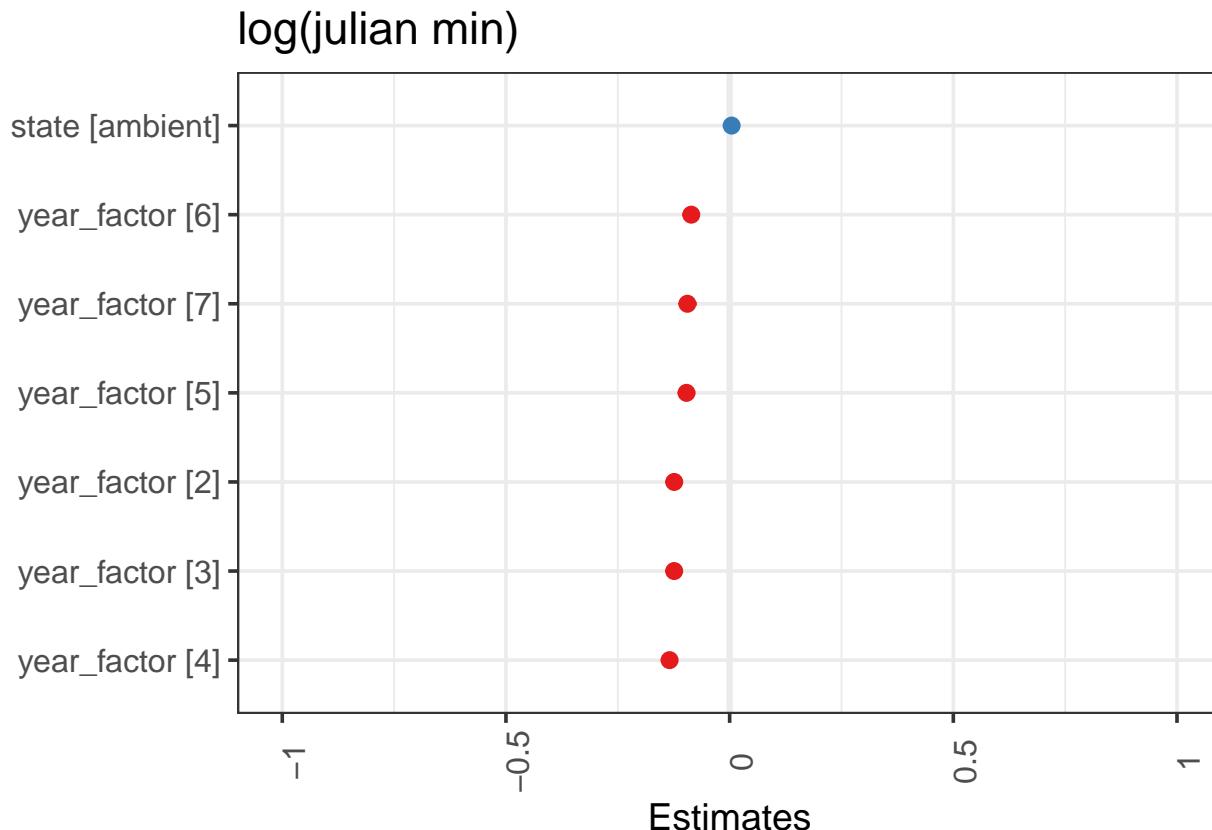
##      dAICc df weight
## mod5  0.0  11  0.963
## mod3  6.5  16  0.037

```

```

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

```



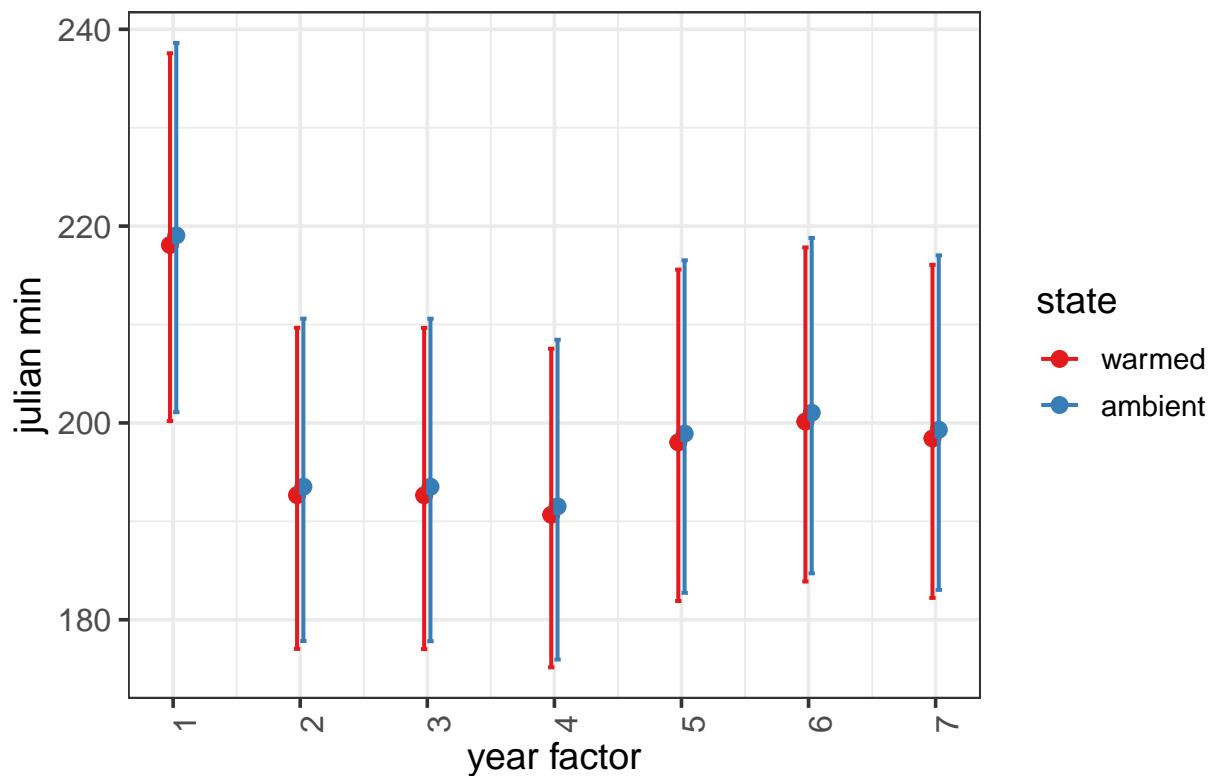
```

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

```

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

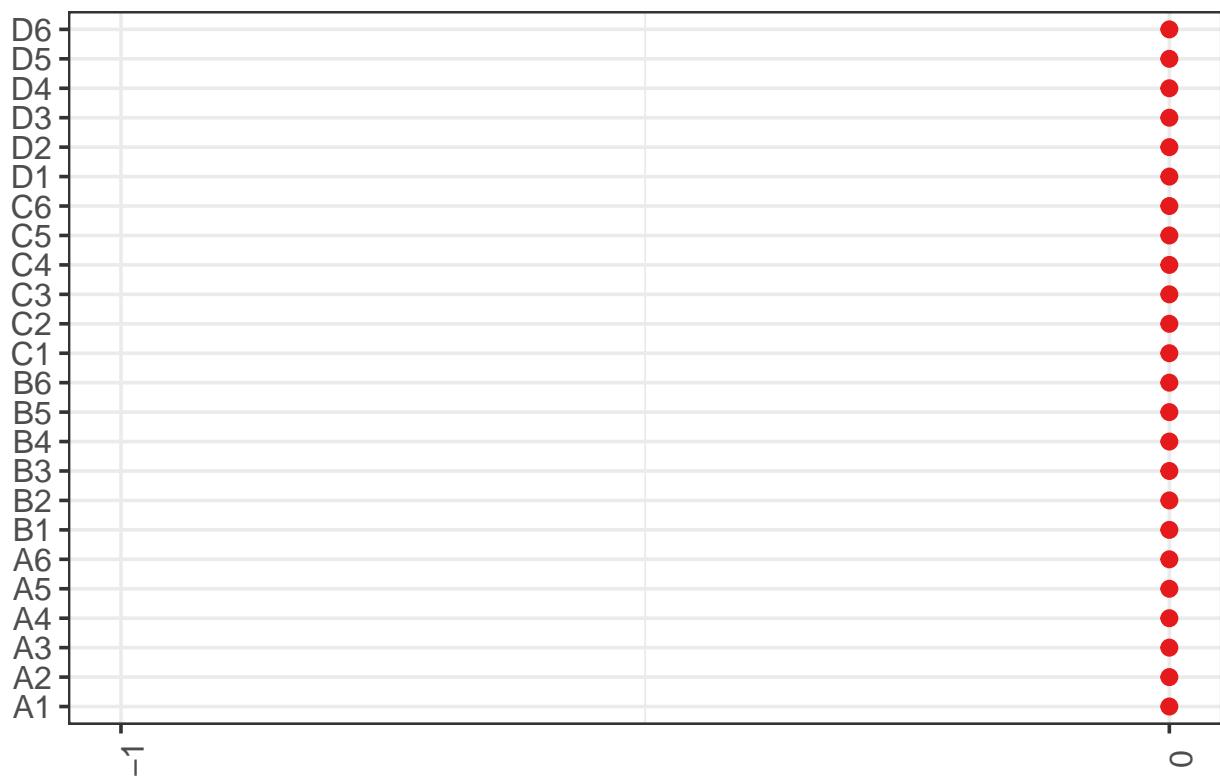
## Predicted values of julian min



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

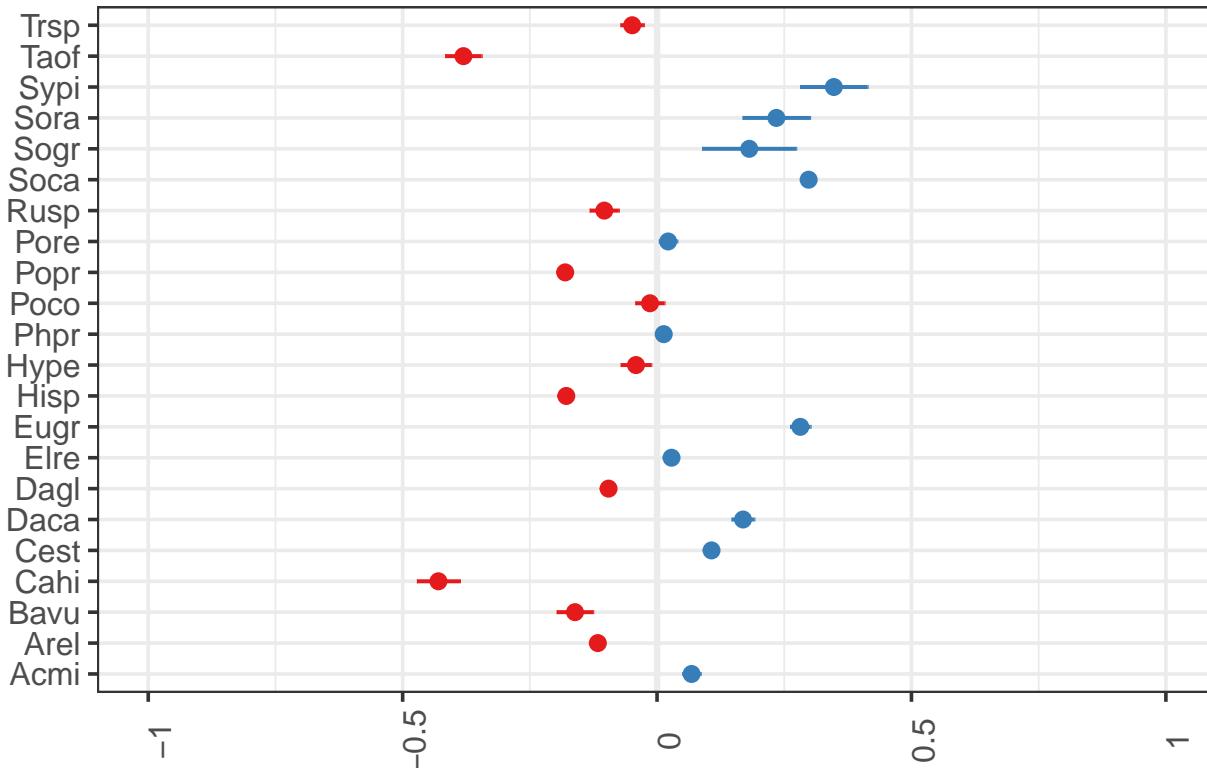
```
## [[1]]
```

## Random effects



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

## Random effects



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

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

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

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

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

```
anova(mod5)
```

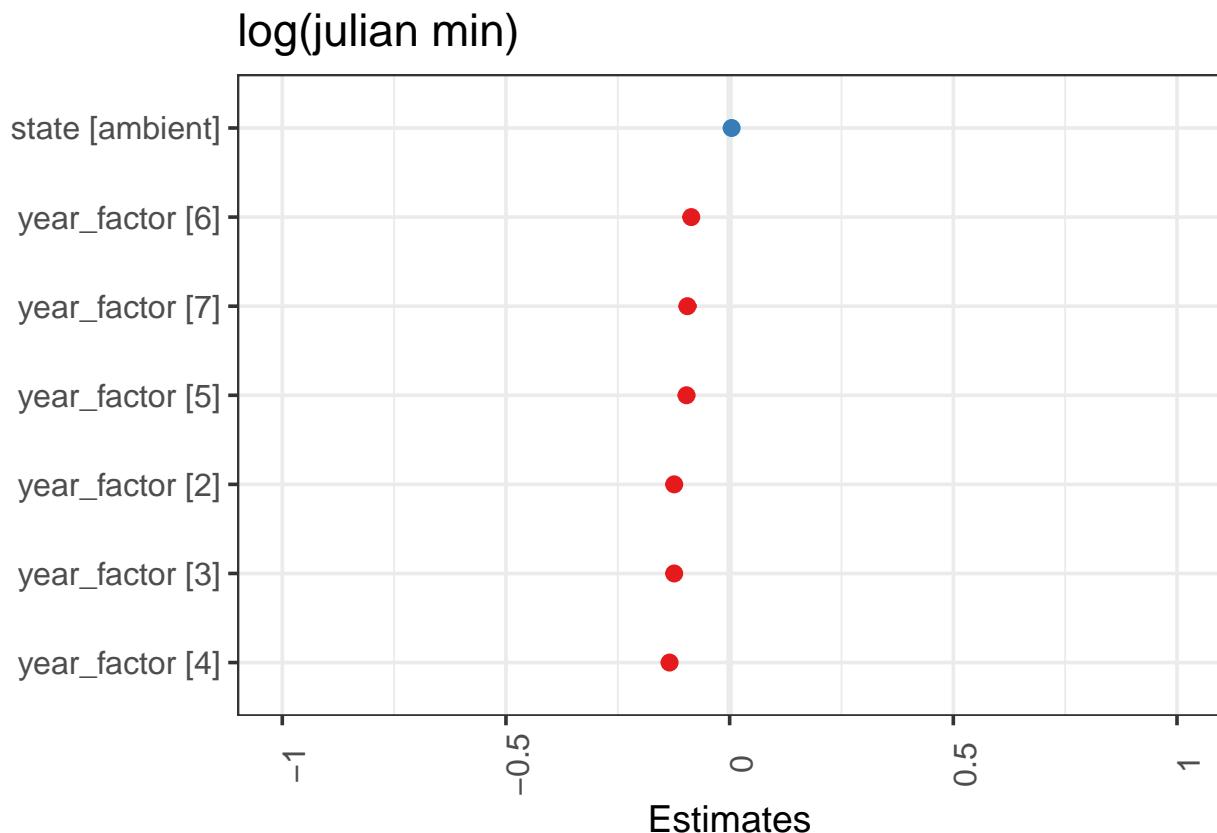
```
## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq  Mean Sq NumDF DenDF F value Pr(>F)
## state      0.00370 0.003698     1 794.25  1.569 0.2107
```

```

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

plot_model(mod5, sort.est = TRUE)

```



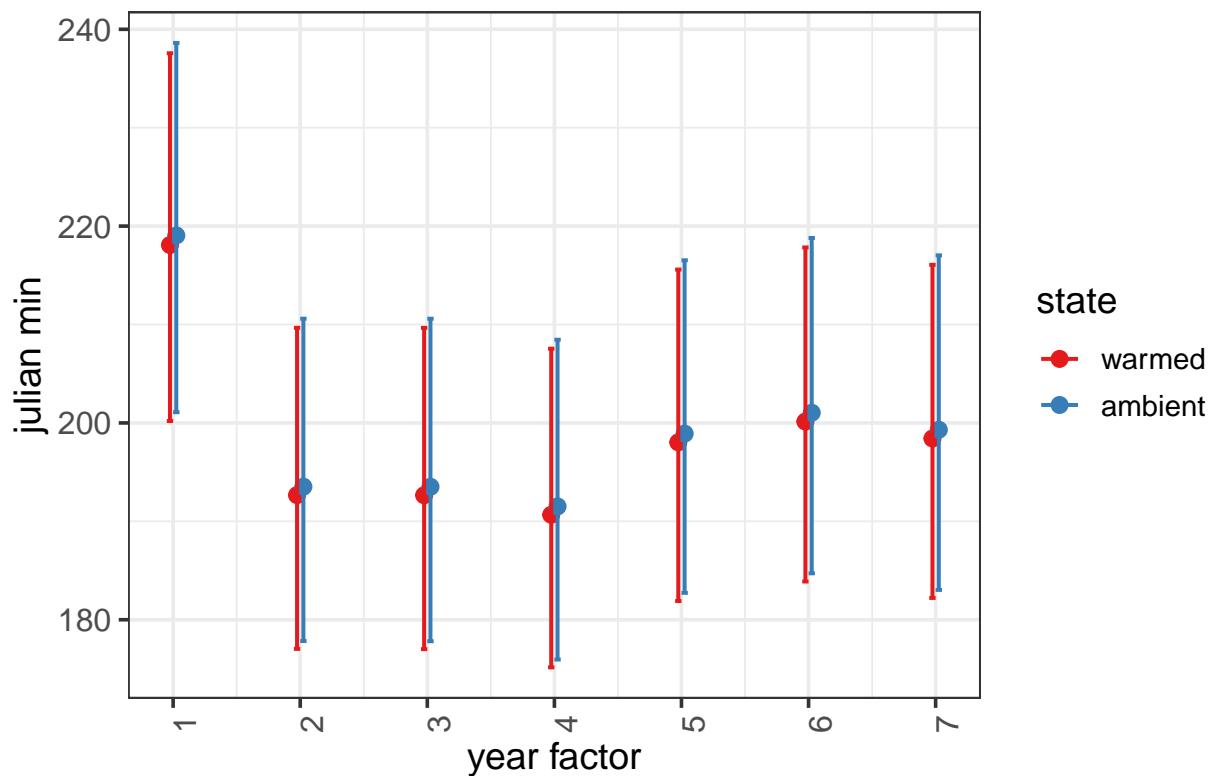
```

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

```

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

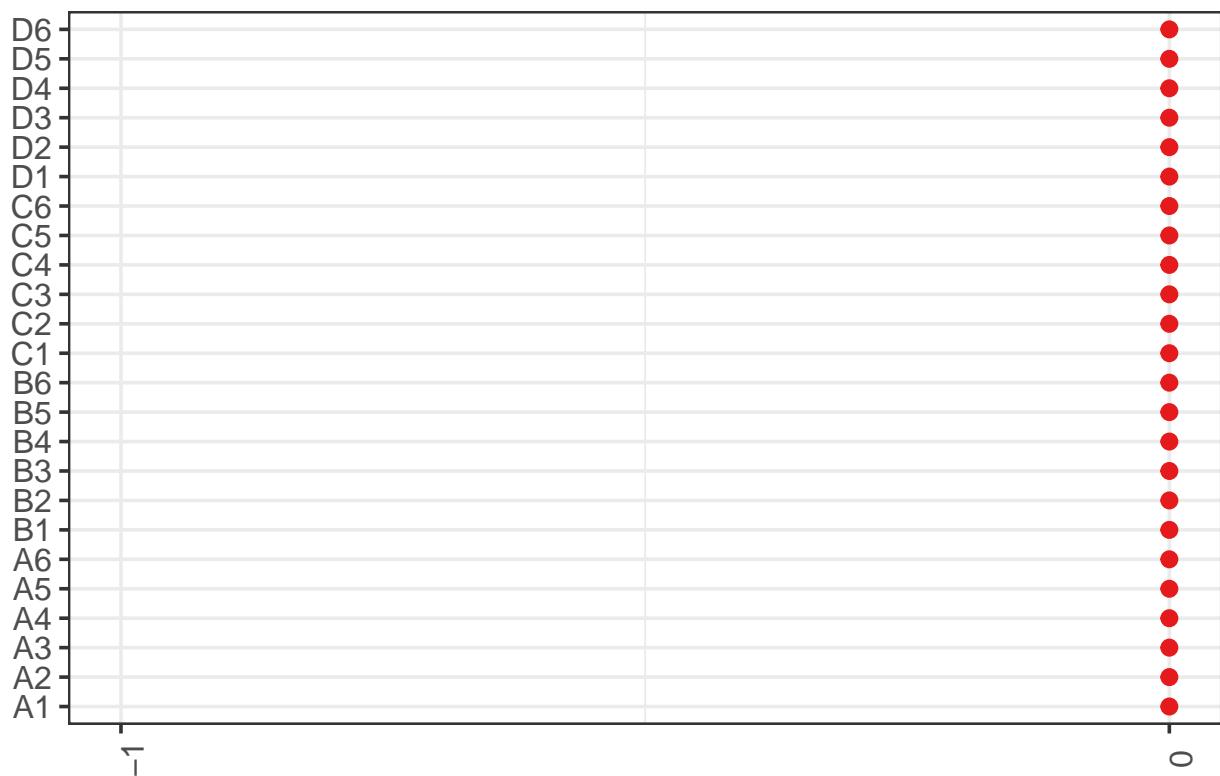
## Predicted values of julian min



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

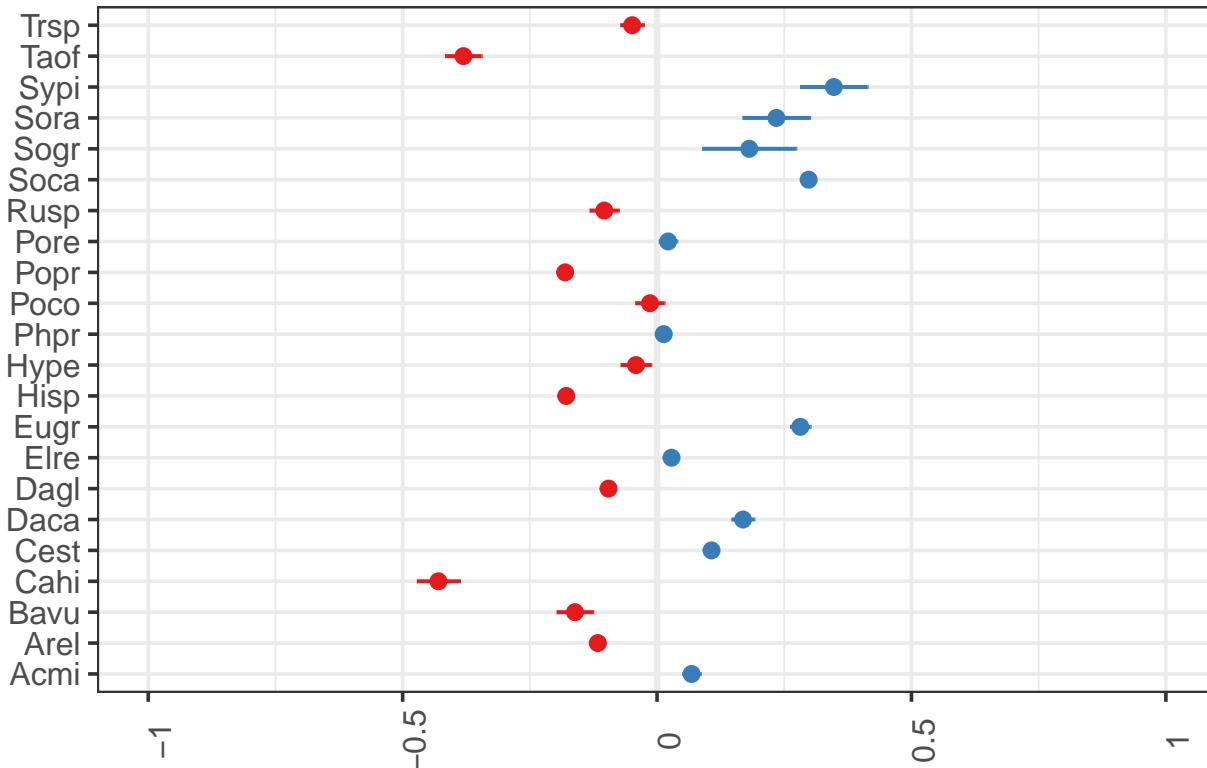
```
## [[1]]
```

## Random effects



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

## Random effects



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

## boundary (singular) fit: see ?isSingular

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

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```
summary(mod7a)
```

```

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

```

```

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

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

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

```

```

## boundary (singular) fit: see ?isSingular

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

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

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

```

```

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

```

```

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

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

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

```

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

```

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

```

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

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

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

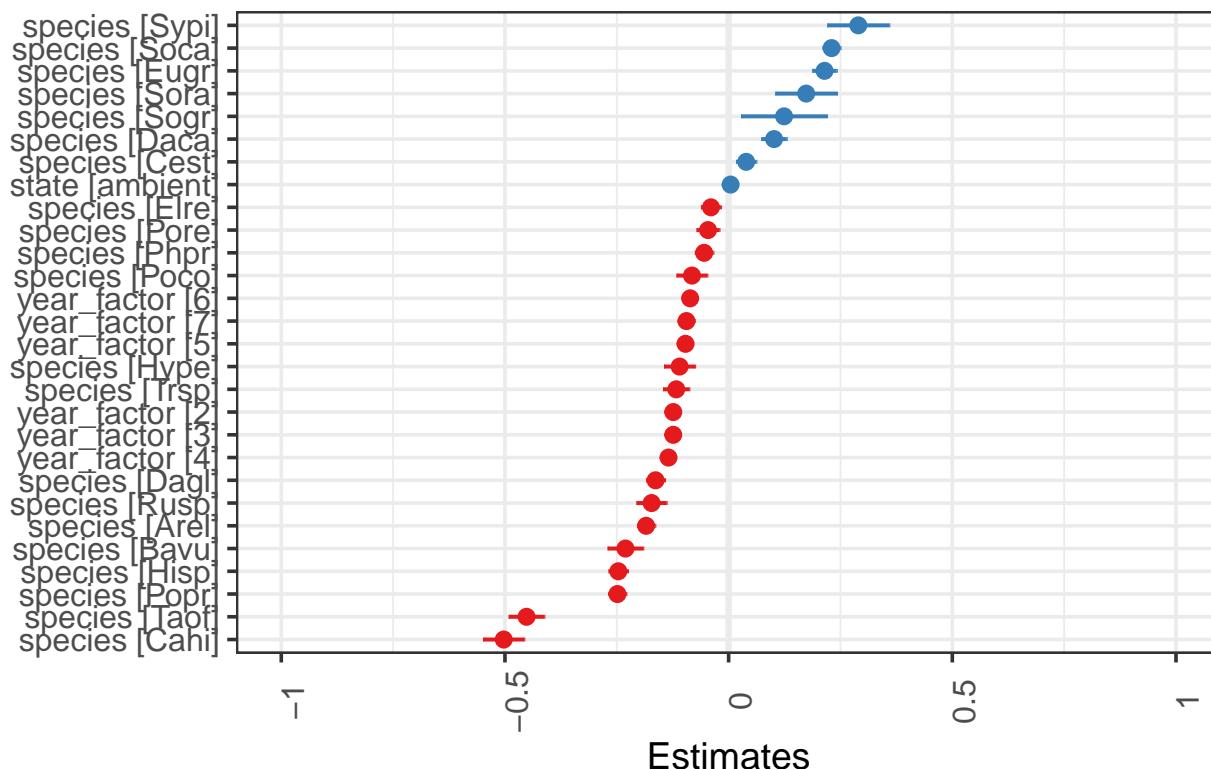
```

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

# 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)

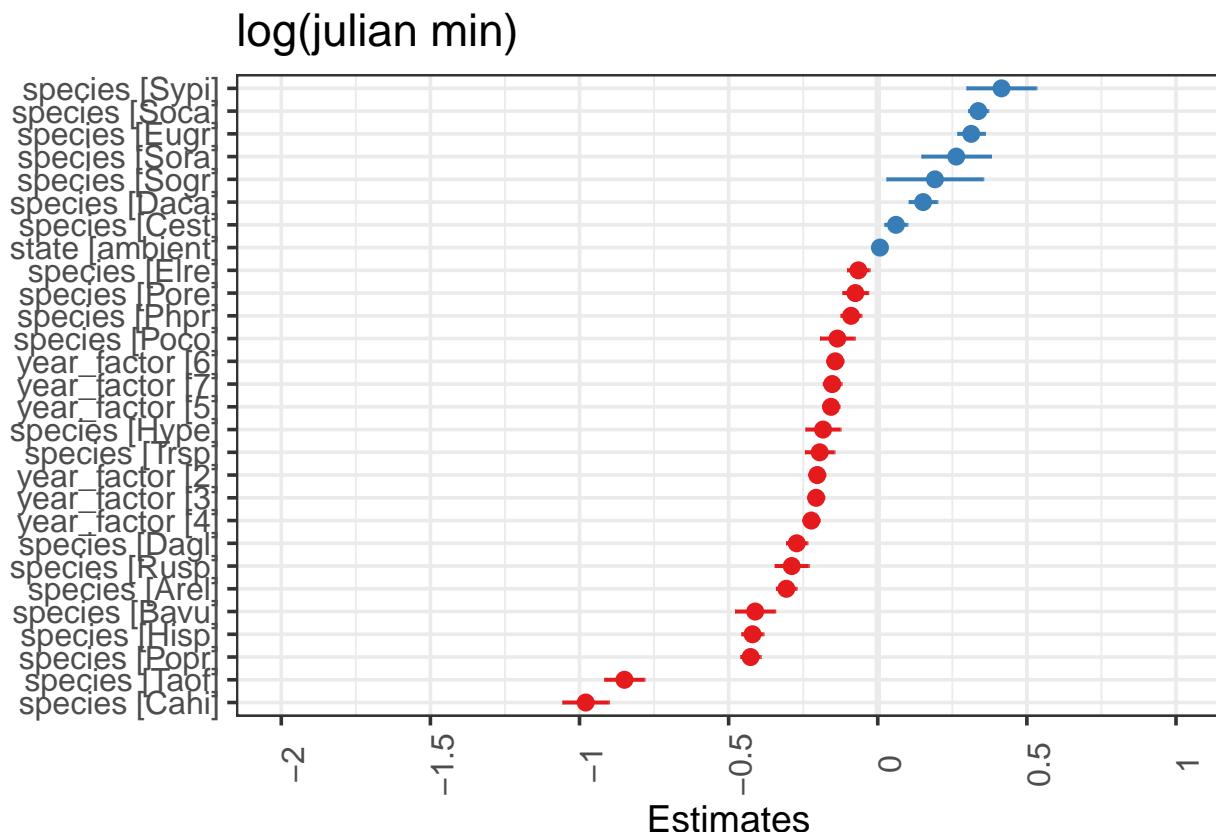
```

## 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
```

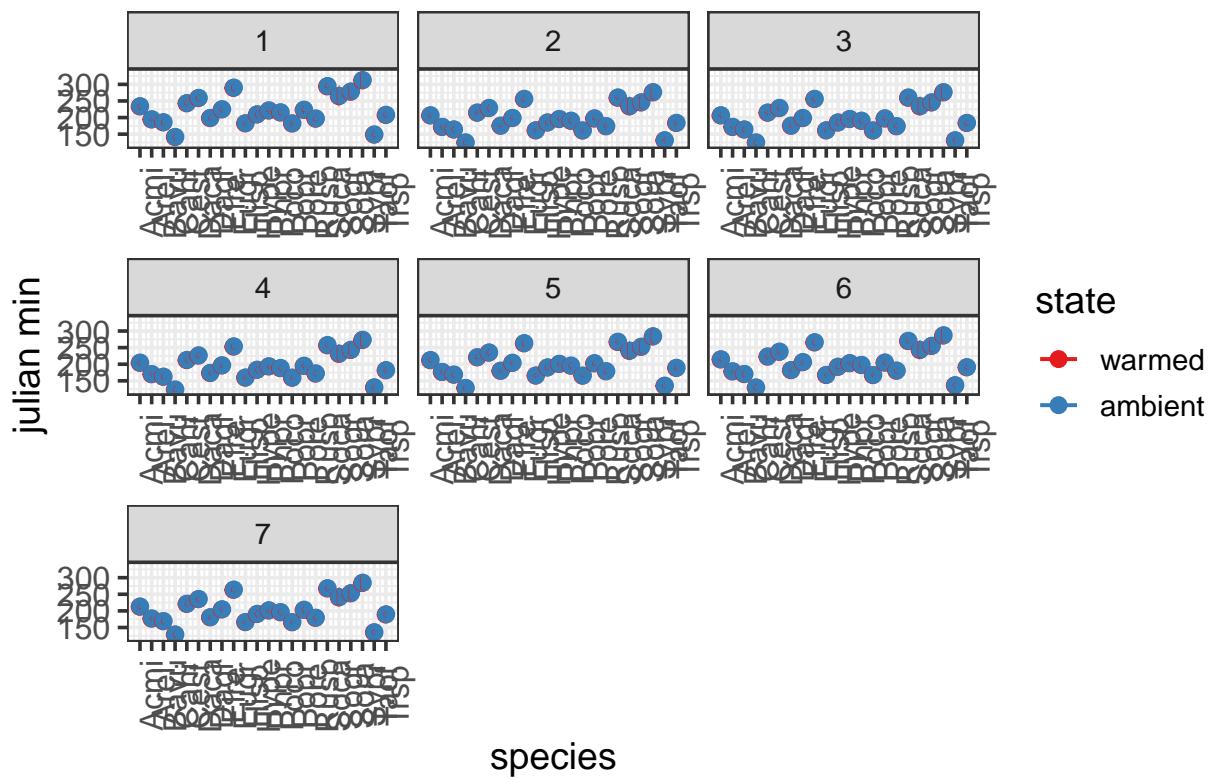


# these are the fixed predicted values: - note this is a new plot

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

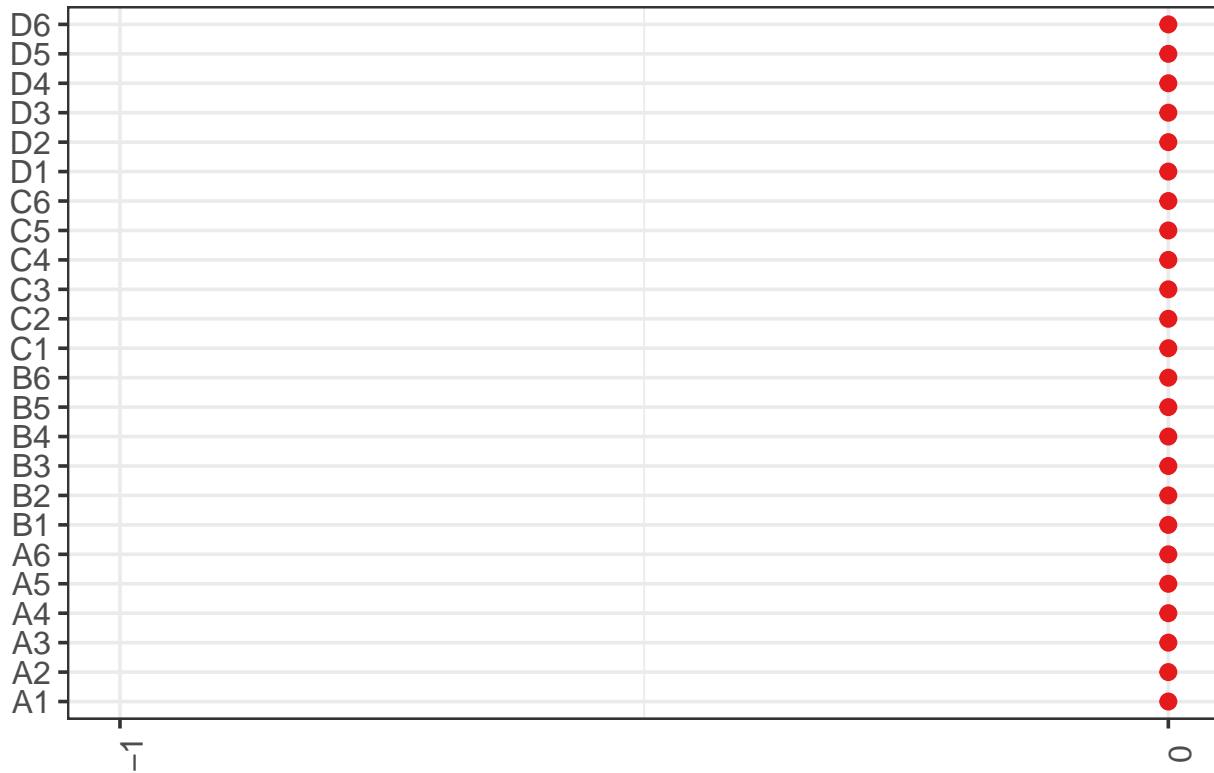
## Model has log-transformed response. Back-transforming predictions to original response scale. Standardized residuals are plotted.

## 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 na
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 -1154.2 -999.03 610.08   -1220.2
## mod8   35 -1151.1 -986.56 610.54   -1221.1 0.9268  2     0.6291
```

```

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 -1237.6 -1181.16 630.78 -1261.6
## mod9     33 -1154.2 -999.03 610.08 -1220.2      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
## -1237.6 -1181.2     630.8   -1261.6      801
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -4.1280 -0.6127  0.0255  0.6215  3.4904
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.00000  0.0000
## Residual           0.01241  0.1114
## Number of obs: 813, groups: plot, 24
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 5.636e+00 1.848e-02 8.130e+02 304.969 < 2e-16 ***
## stateambient 3.164e-05 7.921e-03 8.130e+02   0.004 0.996814
## originBoth   -4.029e-01 1.497e-02 8.130e+02 -26.905 < 2e-16 ***
## originExotic -3.207e-01 9.782e-03 8.130e+02 -32.786 < 2e-16 ***
## year_factor2 -1.036e-01 1.836e-02 8.130e+02 -5.640 2.35e-08 ***
## year_factor3 -1.281e-01 1.883e-02 8.130e+02 -6.803 1.99e-11 ***
## year_factor4 -1.178e-01 1.940e-02 8.130e+02 -6.070 1.96e-09 ***
## year_factor5 -7.814e-02 1.994e-02 8.130e+02 -3.919 9.63e-05 ***
## year_factor6 -6.981e-02 1.906e-02 8.130e+02 -3.662 0.000266 ***
## year_factor7 -1.135e-01 2.149e-02 8.130e+02 -5.284 1.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttmbn orgnBt orgnEx yr_fc2 yr_fc3 yr_fc4 yr_fc5 yr_fc6
## stateambient -0.198
## originBoth   -0.183 -0.092
## originExotic -0.401 -0.061  0.491
## year_factr2 -0.803 -0.016 -0.093  0.029
## year_factr3 -0.784 -0.004 -0.060  0.023  0.791

```

```

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

```

```
anova(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   0.00    0.9968
## origin     15.2484 7.6242     2    813  614.58 < 2.2e-16 ***
## year_factor 0.7868 0.1311     6    813   10.57 2.546e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
emmeans(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 0.007354 65.76    5.293    5.322
##   ambient  5.308 0.006794 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 -3.16e-05 0.008 22.9 -0.004  0.9969
##
## 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 0.008869 345.58    5.531    5.566
##   Both    5.146 0.012894 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 26.689 <.0001
## Native - Exotic  0.3207 0.00986 823 32.518 <.0001
## Both - Exotic   -0.0822 0.01339 823 -6.134 <.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

## $'emmeans of state, origin'
##   state  origin emmean      SE  df lower.CL upper.CL
## warmed  Native  5.549 0.009663 172.72    5.530    5.568
## ambient Native  5.549 0.009794 163.67    5.529    5.568
## warmed  Both    5.146 0.013849 433.30    5.119    5.173
## ambient Both    5.146 0.013141 336.34    5.120    5.172
## warmed  Exotic  5.228 0.006725  46.92    5.214    5.241
## ambient Exotic  5.228 0.006168  30.62    5.215    5.241
##
## Results are averaged over the levels of: year_factor
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $'pairwise differences of state, origin'
##   1           estimate      SE  df t.ratio p.value
## warmed Native - ambient Native -3.16e-05 0.00800 22.9 -0.004  1.0000
## warmed Native - warmed Both    4.03e-01 0.01510 821.8 26.689 <.0001
## warmed Native - ambient Both   4.03e-01 0.01644 273.4 24.507 <.0001
## warmed Native - warmed Exotic 3.21e-01 0.00986 822.6 32.518 <.0001
## warmed Native - ambient Exotic 3.21e-01 0.01231 116.9 26.051 <.0001
## ambient Native - warmed Both   4.03e-01 0.01771 362.6 22.756 <.0001
## ambient Native - ambient Both   4.03e-01 0.01510 821.8 26.689 <.0001
## ambient Native - warmed Exotic 3.21e-01 0.01308 140.0 24.530 <.0001
## ambient Native - ambient Exotic 3.21e-01 0.00986 822.6 32.518 <.0001
## warmed Both - ambient Both     -3.16e-05 0.00800 22.9 -0.004  1.0000
## warmed Both - warmed Exotic   -8.22e-02 0.01339 823.1 -6.134 <.0001
## warmed Both - ambient Exotic  -8.22e-02 0.01598 274.6 -5.144 <.0001
## ambient Both - warmed Exotic -8.21e-02 0.01521 209.6 -5.398 <.0001
## ambient Both - ambient Exotic -8.22e-02 0.01339 823.1 -6.134 <.0001
## warmed Exotic - ambient Exotic -3.16e-05 0.00800 22.9 -0.004  1.0000
##
## Results are averaged over the levels of: year_factor
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 6 estimates

```

```

# including growth form - first with interaction term
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

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 -936.63 -781.50 501.31  -1002.6
## mod10   35 -933.15 -768.62 501.57  -1003.1 0.5171  2     0.7722

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 -1031.46 -975.05 527.73  -1055.5
## mod11    33  -936.63 -781.50 501.31  -1002.6      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
## -1031.5  -975.1   527.7  -1055.5      801
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max

```

```

## -5.3663 -0.6249 -0.0292  0.7473  3.2414
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.00000  0.0000
## Residual           0.01598  0.1264
## Number of obs: 813, groups: plot, 24
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                5.591815  0.020735 813.000000 269.686 < 2e-16 ***
## stateambient              -0.018607  0.008976 813.000000 -2.073  0.0385 *
## growth_habit               -0.298651  0.015748 813.000000 -18.964 < 2e-16 ***
## growth_habitGraminoid     -0.251892  0.009864 813.000000 -25.537 < 2e-16 ***
## year_factor2               -0.164920  0.021064 813.000000 -7.830 1.52e-14 ***
## year_factor3               -0.185958  0.021552 813.000000 -8.628 < 2e-16 ***
## year_factor4               -0.154714  0.022040 813.000000 -7.020 4.69e-12 ***
## year_factor5               -0.107010  0.022655 813.000000 -4.723 2.73e-06 ***
## year_factor6               -0.093082  0.021707 813.000000 -4.288 2.02e-05 ***
## year_factor7               -0.129739  0.024382 813.000000 -5.321 1.34e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttmbn grwth_ grwt_G yr_fc2 yr_fc3 yr_fc4 yr_fc5 yr_fc6
## stateambint -0.230
## growth_habt -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       0.0687  0.0687     1    813  4.2973 0.03849 *
## growth_habit 12.3382  6.1691     2    813 385.9317 < 2e-16 ***
## year_factor  1.7126  0.2854     6    813 17.8559 < 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 0.02033 784.2   5.543  5.622
##  2           Forb      5.418 0.01048 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 0.01777 687.2   5.418  5.488
##  1           Graminoid 5.284 0.02413 807.5   5.237  5.331
##  2           Graminoid 5.119 0.01534 665.6   5.089  5.149
##  3           Graminoid 5.098 0.01680 696.1   5.065  5.131
##  4           Graminoid 5.129 0.01854 740.8   5.093  5.166
##  5           Graminoid 5.177 0.01907 741.9   5.139  5.214
##  6           Graminoid 5.191 0.01736 694.8   5.157  5.225
##  7           Graminoid 5.154 0.02134 727.3   5.112  5.196
##  1           Graminoid 5.331 0.01891 771.6   5.294  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 0.01178 496.7   5.214  5.261
##  7           Graminoid 5.201 0.01597 684.5   5.170  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 0.164920 0.02122 818 7.771 <.0001
##  1 Forb - 3 Forb 0.185958 0.02172 819 8.562 <.0001
##  1 Forb - 4 Forb 0.154714 0.02220 816 6.969 <.0001
##  1 Forb - 5 Forb 0.107010 0.02282 816 4.689 0.0006
##  1 Forb - 6 Forb 0.093082 0.02187 817 4.257 0.0041
##  1 Forb - 7 Forb 0.129739 0.02458 822 5.277 <.0001
##  1 Forb - 1       0.298651 0.01587 821 18.814 <.0001
##  1 Forb - 2       0.463571 0.02567 821 18.056 <.0001
##  1 Forb - 3       0.484610 0.02650 822 18.289 <.0001
##  1 Forb - 4       0.453365 0.02709 821 16.736 <.0001
##  1 Forb - 5       0.405661 0.02745 820 14.776 <.0001
##  1 Forb - 6       0.391733 0.02658 821 14.737 <.0001
##  1 Forb - 7       0.428390 0.02876 823 14.895 <.0001
##  1 Forb - 1 Graminoid 0.251892 0.00995 823 25.316 <.0001
##  1 Forb - 2 Graminoid 0.416812 0.02473 822 16.856 <.0001
##  1 Forb - 3 Graminoid 0.437850 0.02505 821 17.478 <.0001
##  1 Forb - 4 Graminoid 0.406606 0.02472 818 16.452 <.0001
##  1 Forb - 5 Graminoid 0.358902 0.02528 816 14.195 <.0001
##  1 Forb - 6 Graminoid 0.344974 0.02482 818 13.901 <.0001
##  1 Forb - 7 Graminoid 0.381631 0.02643 819 14.439 <.0001
##  2 Forb - 3 Forb 0.021039 0.01376 817 1.529 0.9953
##  2 Forb - 4 Forb -0.010206 0.01509 816 -0.676 1.0000
##  2 Forb - 5 Forb -0.057910 0.01594 821 -3.634 0.0421
##  2 Forb - 6 Forb -0.071838 0.01426 823 -5.039 0.0001

```

## 2 Forb - 7 Forb	-0.035181	0.01862	823	-1.890	0.9498
## 2 Forb - 1	0.133732	0.02731	817	4.898	0.0002
## 2 Forb - 2	0.298651	0.01587	821	18.814	<.0001
## 2 Forb - 3	0.319690	0.02152	820	14.857	<.0001
## 2 Forb - 4	0.288446	0.02262	820	12.749	<.0001
## 2 Forb - 5	0.240742	0.02303	821	10.455	<.0001
## 2 Forb - 6	0.226813	0.02179	822	10.408	<.0001
## 2 Forb - 7	0.263470	0.02475	823	10.645	<.0001
## 2 Forb - 1 Graminoid	0.086972	0.02208	817	3.940	0.0143
## 2 Forb - 2 Graminoid	0.251892	0.00995	823	25.316	<.0001
## 2 Forb - 3 Graminoid	0.272931	0.01683	818	16.218	<.0001
## 2 Forb - 4 Graminoid	0.241686	0.01684	814	14.350	<.0001
## 2 Forb - 5 Graminoid	0.193982	0.01762	815	11.009	<.0001
## 2 Forb - 6 Graminoid	0.180054	0.01670	818	10.781	<.0001
## 2 Forb - 7 Graminoid	0.216711	0.01946	820	11.136	<.0001
## 3 Forb - 4 Forb	-0.031245	0.01583	810	-1.974	0.9247
## 3 Forb - 5 Forb	-0.078948	0.01666	820	-4.739	0.0005
## 3 Forb - 6 Forb	-0.092877	0.01510	822	-6.152	<.0001
## 3 Forb - 7 Forb	-0.056219	0.01919	823	-2.929	0.2918
## 3 Forb - 1	0.112693	0.02730	818	4.128	0.0069
## 3 Forb - 2	0.277613	0.02049	819	13.552	<.0001
## 3 Forb - 3	0.298651	0.01587	821	18.814	<.0001
## 3 Forb - 4	0.267407	0.02265	818	11.807	<.0001
## 3 Forb - 5	0.219703	0.02307	820	9.524	<.0001
## 3 Forb - 6	0.205775	0.02186	822	9.413	<.0001
## 3 Forb - 7	0.242432	0.02475	823	9.794	<.0001
## 3 Forb - 1 Graminoid	0.065934	0.02267	820	2.909	0.3049
## 3 Forb - 2 Graminoid	0.230853	0.01713	822	13.475	<.0001
## 3 Forb - 3 Graminoid	0.251892	0.00995	823	25.316	<.0001
## 3 Forb - 4 Graminoid	0.220647	0.01765	812	12.501	<.0001
## 3 Forb - 5 Graminoid	0.172944	0.01842	817	9.390	<.0001
## 3 Forb - 6 Graminoid	0.159015	0.01757	820	9.050	<.0001
## 3 Forb - 7 Graminoid	0.195672	0.02014	818	9.716	<.0001
## 4 Forb - 5 Forb	-0.047704	0.01744	813	-2.735	0.4258
## 4 Forb - 6 Forb	-0.061632	0.01611	814	-3.827	0.0216
## 4 Forb - 7 Forb	-0.024975	0.01977	819	-1.263	0.9997
## 4 Forb - 1	0.143937	0.02749	815	5.235	<.0001
## 4 Forb - 2	0.308857	0.02116	818	14.599	<.0001
## 4 Forb - 3	0.329896	0.02218	816	14.873	<.0001
## 4 Forb - 4	0.298651	0.01587	821	18.814	<.0001
## 4 Forb - 5	0.250947	0.02341	816	10.718	<.0001
## 4 Forb - 6	0.237019	0.02234	818	10.612	<.0001
## 4 Forb - 7	0.273676	0.02499	822	10.950	<.0001
## 4 Forb - 1 Graminoid	0.097178	0.02394	819	4.060	0.0090
## 4 Forb - 2 Graminoid	0.262098	0.01923	822	13.628	<.0001
## 4 Forb - 3 Graminoid	0.283136	0.01968	819	14.384	<.0001
## 4 Forb - 4 Graminoid	0.251892	0.00995	823	25.316	<.0001
## 4 Forb - 5 Graminoid	0.204188	0.02009	814	10.162	<.0001
## 4 Forb - 6 Graminoid	0.190260	0.01945	817	9.784	<.0001
## 4 Forb - 7 Graminoid	0.226917	0.02159	813	10.512	<.0001
## 5 Forb - 6 Forb	-0.013928	0.01691	814	-0.824	1.0000
## 5 Forb - 7 Forb	0.022729	0.02045	817	1.111	1.0000
## 5 Forb - 1	0.191641	0.02814	816	6.811	<.0001
## 5 Forb - 2	0.356561	0.02195	821	16.245	<.0001

## 5 Forb - 3	0.377600	0.02296	821	16.448	<.0001
## 5 Forb - 4	0.346355	0.02375	818	14.584	<.0001
## 5 Forb - 5	0.298651	0.01587	821	18.814	<.0001
## 5 Forb - 6	0.284723	0.02310	818	12.327	<.0001
## 5 Forb - 7	0.321380	0.02569	821	12.509	<.0001
## 5 Forb - 1 Graminoid	0.144882	0.02450	820	5.913	<.0001
## 5 Forb - 2 Graminoid	0.309802	0.01989	823	15.577	<.0001
## 5 Forb - 3 Graminoid	0.330840	0.02035	823	16.260	<.0001
## 5 Forb - 4 Graminoid	0.299596	0.02006	820	14.931	<.0001
## 5 Forb - 5 Graminoid	0.251892	0.00995	823	25.316	<.0001
## 5 Forb - 6 Graminoid	0.237964	0.02011	819	11.835	<.0001
## 5 Forb - 7 Graminoid	0.274621	0.02220	815	12.369	<.0001
## 6 Forb - 7 Forb	0.036657	0.01936	819	1.894	0.9488
## 6 Forb - 1	0.205570	0.02745	816	7.488	<.0001
## 6 Forb - 2	0.370490	0.02087	821	17.752	<.0001
## 6 Forb - 3	0.391528	0.02195	821	17.835	<.0001
## 6 Forb - 4	0.360284	0.02289	818	15.742	<.0001
## 6 Forb - 5	0.312580	0.02329	817	13.419	<.0001
## 6 Forb - 6	0.298651	0.01587	821	18.814	<.0001
## 6 Forb - 7	0.335309	0.02492	822	13.455	<.0001
## 6 Forb - 1 Graminoid	0.158810	0.02320	820	6.844	<.0001
## 6 Forb - 2 Graminoid	0.323730	0.01805	822	17.940	<.0001
## 6 Forb - 3 Graminoid	0.344769	0.01858	823	18.557	<.0001
## 6 Forb - 4 Graminoid	0.313524	0.01840	820	17.037	<.0001
## 6 Forb - 5 Graminoid	0.265820	0.01913	817	13.897	<.0001
## 6 Forb - 6 Graminoid	0.251892	0.00995	823	25.316	<.0001
## 6 Forb - 7 Graminoid	0.288549	0.02074	814	13.912	<.0001
## 7 Forb - 1	0.168912	0.02976	818	5.676	<.0001
## 7 Forb - 2	0.333832	0.02418	822	13.809	<.0001
## 7 Forb - 3	0.354871	0.02506	821	14.162	<.0001
## 7 Forb - 4	0.323626	0.02571	817	12.590	<.0001
## 7 Forb - 5	0.275923	0.02608	815	10.578	<.0001
## 7 Forb - 6	0.261994	0.02514	816	10.420	<.0001
## 7 Forb - 7	0.298651	0.01587	821	18.814	<.0001
## 7 Forb - 1 Graminoid	0.122153	0.02661	823	4.590	0.0010
## 7 Forb - 2 Graminoid	0.287073	0.02264	817	12.682	<.0001
## 7 Forb - 3 Graminoid	0.308111	0.02300	821	13.396	<.0001
## 7 Forb - 4 Graminoid	0.276867	0.02266	823	12.218	<.0001
## 7 Forb - 5 Graminoid	0.229163	0.02327	822	9.847	<.0001
## 7 Forb - 6 Graminoid	0.215235	0.02274	823	9.465	<.0001
## 7 Forb - 7 Graminoid	0.251892	0.00995	823	25.316	<.0001
## 1 - 2	0.164920	0.02122	818	7.771	<.0001
## 1 - 3	0.185958	0.02172	819	8.562	<.0001
## 1 - 4	0.154714	0.02220	816	6.969	<.0001
## 1 - 5	0.107010	0.02282	816	4.689	0.0006
## 1 - 6	0.093082	0.02187	817	4.257	0.0041
## 1 - 7	0.129739	0.02458	822	5.277	<.0001
## 1 - 1 Graminoid	-0.046759	0.01563	822	-2.991	0.2548
## 1 - 2 Graminoid	0.118160	0.02828	821	4.178	0.0057
## 1 - 3 Graminoid	0.139199	0.02819	821	4.938	0.0002
## 1 - 4 Graminoid	0.107955	0.02770	818	3.897	0.0167
## 1 - 5 Graminoid	0.060251	0.02835	817	2.125	0.8607
## 1 - 6 Graminoid	0.046322	0.02801	818	1.654	0.9878
## 1 - 7 Graminoid	0.082979	0.02955	817	2.808	0.3726

##	2	-	3		0.021039	0.01376	817	1.529	0.9953
##	2	-	4		-0.010206	0.01509	816	-0.676	1.0000
##	2	-	5		-0.057910	0.01594	821	-3.634	0.0421
##	2	-	6		-0.071838	0.01426	823	-5.039	0.0001
##	2	-	7		-0.035181	0.01862	823	-1.890	0.9498
##	2	-	1	Graminoid	-0.211679	0.02428	822	-8.719	<.0001
##	2	-	2	Graminoid	-0.046759	0.01563	822	-2.991	0.2548
##	2	-	3	Graminoid	-0.025721	0.02017	822	-1.275	0.9996
##	2	-	4	Graminoid	-0.056965	0.01992	820	-2.860	0.3371
##	2	-	5	Graminoid	-0.104669	0.02077	821	-5.038	0.0001
##	2	-	6	Graminoid	-0.118598	0.02011	822	-5.896	<.0001
##	2	-	7	Graminoid	-0.081940	0.02258	819	-3.629	0.0428
##	3	-	4		-0.031245	0.01583	810	-1.974	0.9247
##	3	-	5		-0.078948	0.01666	820	-4.739	0.0005
##	3	-	6		-0.092877	0.01510	822	-6.152	<.0001
##	3	-	7		-0.056219	0.01919	823	-2.929	0.2918
##	3	-	1	Graminoid	-0.232718	0.02525	823	-9.216	<.0001
##	3	-	2	Graminoid	-0.067798	0.02146	823	-3.159	0.1702
##	3	-	3	Graminoid	-0.046759	0.01563	822	-2.991	0.2548
##	3	-	4	Graminoid	-0.078004	0.02113	819	-3.692	0.0346
##	3	-	5	Graminoid	-0.125708	0.02195	821	-5.726	<.0001
##	3	-	6	Graminoid	-0.139636	0.02136	822	-6.538	<.0001
##	3	-	7	Graminoid	-0.102979	0.02363	818	-4.357	0.0027
##	4	-	5		-0.047704	0.01744	813	-2.735	0.4258
##	4	-	6		-0.061632	0.01611	814	-3.827	0.0216
##	4	-	7		-0.024975	0.01977	819	-1.263	0.9997
##	4	-	1	Graminoid	-0.201473	0.02659	823	-7.576	<.0001
##	4	-	2	Graminoid	-0.036554	0.02340	823	-1.562	0.9938
##	4	-	3	Graminoid	-0.015515	0.02331	822	-0.666	1.0000
##	4	-	4	Graminoid	-0.046759	0.01563	822	-2.991	0.2548
##	4	-	5	Graminoid	-0.094463	0.02360	820	-4.003	0.0113
##	4	-	6	Graminoid	-0.108392	0.02315	821	-4.682	0.0006
##	4	-	7	Graminoid	-0.071735	0.02509	816	-2.859	0.3373
##	5	-	6		-0.013928	0.01691	814	-0.824	1.0000
##	5	-	7		0.022729	0.02045	817	1.111	1.0000
##	5	-	1	Graminoid	-0.153769	0.02696	823	-5.705	<.0001
##	5	-	2	Graminoid	0.011150	0.02377	823	0.469	1.0000
##	5	-	3	Graminoid	0.032189	0.02370	823	1.358	0.9990
##	5	-	4	Graminoid	0.000945	0.02324	822	0.041	1.0000
##	5	-	5	Graminoid	-0.046759	0.01563	822	-2.991	0.2548
##	5	-	6	Graminoid	-0.060688	0.02354	822	-2.578	0.5475
##	5	-	7	Graminoid	-0.024030	0.02547	815	-0.944	1.0000
##	6	-	7		0.036657	0.01936	819	1.894	0.9488
##	6	-	1	Graminoid	-0.139841	0.02570	823	-5.442	<.0001
##	6	-	2	Graminoid	0.025079	0.02215	822	1.132	0.9999
##	6	-	3	Graminoid	0.046117	0.02210	823	2.087	0.8796
##	6	-	4	Graminoid	0.014873	0.02171	822	0.685	1.0000
##	6	-	5	Graminoid	-0.032831	0.02251	822	-1.459	0.9974
##	6	-	6	Graminoid	-0.046759	0.01563	822	-2.991	0.2548
##	6	-	7	Graminoid	-0.010102	0.02411	816	-0.419	1.0000
##	7	-	1	Graminoid	-0.176498	0.02871	821	-6.147	<.0001
##	7	-	2	Graminoid	-0.011579	0.02592	815	-0.447	1.0000
##	7	-	3	Graminoid	0.009460	0.02582	819	0.366	1.0000
##	7	-	4	Graminoid	-0.021784	0.02531	822	-0.861	1.0000

```

## 7 - 5 Graminoid      -0.069488 0.02601 823 -2.671  0.4746
## 7 - 6 Graminoid      -0.083417 0.02563 823 -3.255  0.1323
## 7 - 7 Graminoid      -0.046759 0.01563 822 -2.991  0.2548
## 1 Graminoid - 2 Graminoid  0.164920 0.02122 818  7.771 <.0001
## 1 Graminoid - 3 Graminoid  0.185958 0.02172 819  8.562 <.0001
## 1 Graminoid - 4 Graminoid  0.154714 0.02220 816  6.969 <.0001
## 1 Graminoid - 5 Graminoid  0.107010 0.02282 816  4.689  0.0006
## 1 Graminoid - 6 Graminoid  0.093082 0.02187 817  4.257  0.0041
## 1 Graminoid - 7 Graminoid  0.129739 0.02458 822  5.277 <.0001
## 2 Graminoid - 3 Graminoid  0.021039 0.01376 817  1.529  0.9953
## 2 Graminoid - 4 Graminoid -0.010206 0.01509 816 -0.676  1.0000
## 2 Graminoid - 5 Graminoid -0.057910 0.01594 821 -3.634  0.0421
## 2 Graminoid - 6 Graminoid -0.071838 0.01426 823 -5.039  0.0001
## 2 Graminoid - 7 Graminoid -0.035181 0.01862 823 -1.890  0.9498
## 3 Graminoid - 4 Graminoid -0.031245 0.01583 810 -1.974  0.9247
## 3 Graminoid - 5 Graminoid -0.078948 0.01666 820 -4.739  0.0005
## 3 Graminoid - 6 Graminoid -0.092877 0.01510 822 -6.152 <.0001
## 3 Graminoid - 7 Graminoid -0.056219 0.01919 823 -2.929  0.2918
## 4 Graminoid - 5 Graminoid -0.047704 0.01744 813 -2.735  0.4258
## 4 Graminoid - 6 Graminoid -0.061632 0.01611 814 -3.827  0.0216
## 4 Graminoid - 7 Graminoid -0.024975 0.01977 819 -1.263  0.9997
## 5 Graminoid - 6 Graminoid -0.013928 0.01691 814 -0.824  1.0000
## 5 Graminoid - 7 Graminoid  0.022729 0.02045 817  1.111  1.0000
## 6 Graminoid - 7 Graminoid  0.036657 0.01936 819  1.894  0.9488
##
## 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

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

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

## boundary (singular) fit: see ?isSingular

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

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

```

```

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

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

summary(mod7a)

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

```

```

##  plot      (Intercept) 0.000000 0.00000
##  Residual          0.002293 0.04789
## Number of obs: 816, groups: plot, 24
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                5.452875  0.011990 816.000000 454.793 < 2e-16 ***
## stateambient               0.004370  0.003507 816.000000  1.246 0.213101
## speciesArel                -0.184443  0.010465 816.000000 -17.624 < 2e-16 ***
## speciesBavu                 -0.230672  0.020305 816.000000 -11.360 < 2e-16 ***
## speciesCahi                 -0.502774  0.023467 816.000000 -21.425 < 2e-16 ***
## speciesCest                  0.039215  0.011618 816.000000  3.375 0.000772 ***
## speciesDaca                  0.101541  0.014495 816.000000  7.005 5.16e-12 ***
## speciesDagl                 -0.163366  0.010831 816.000000 -15.083 < 2e-16 ***
## speciesElre                 -0.039521  0.011372 816.000000 -3.475 0.000537 ***
## speciesEugr                  0.214379  0.013900 816.000000 15.422 < 2e-16 ***
## speciesHisp                 -0.246588  0.011127 816.000000 -22.161 < 2e-16 ***
## speciesHype                 -0.109633  0.017723 816.000000 -6.186 9.75e-10 ***
## speciesPhpr                  0.054832  0.010547 816.000000 -5.199 2.54e-07 ***
## speciesPoco                 -0.082002  0.017606 816.000000 -4.658 3.73e-06 ***
## speciesPopr                 -0.248610  0.010203 816.000000 -24.368 < 2e-16 ***
## speciesPore                 -0.046105  0.013012 816.000000 -3.543 0.000418 ***
## speciesRusp                 -0.172216  0.017181 816.000000 -10.024 < 2e-16 ***
## speciesSoca                 0.230129  0.010169 816.000000 22.630 < 2e-16 ***
## speciesSogr                 0.123923  0.048964 816.000000  2.531 0.011564 *
## speciesSora                 0.173318  0.035168 816.000000  4.928 1.00e-06 ***
## speciesSyti                  0.289872  0.035383 816.000000  8.192 9.86e-16 ***
## speciesTaof                 -0.451748  0.020315 816.000000 -22.237 < 2e-16 ***
## speciesTrsp                 -0.117081  0.014949 816.000000 -7.832 1.49e-14 ***
## factor(year_factor)2        -0.123848  0.008143 816.000000 -15.210 < 2e-16 ***
## factor(year_factor)3        -0.123874  0.008310 816.000000 -14.906 < 2e-16 ***
## factor(year_factor)4        -0.134431  0.008450 816.000000 -15.909 < 2e-16 ***
## factor(year_factor)5        -0.096438  0.008651 816.000000 -11.148 < 2e-16 ***
## factor(year_factor)6        -0.086017  0.008330 816.000000 -10.327 < 2e-16 ***
## factor(year_factor)7        -0.094345  0.009449 816.000000 -9.985 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

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

```

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

```
anova(mod7a)
```

```

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

```

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

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

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

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 -438.38 -407.38  229.19  -458.38
## mod6p   11 -438.77 -404.67  230.38  -460.77  2.3927   1     0.1219

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
## -438.4   -407.4    229.2    -458.4      154
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.2394 -0.6911  0.0559  0.5012  3.3777
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   plot      (Intercept) 0.0001572 0.01254
##   Residual           0.0034388 0.05864
## Number of obs: 164, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  5.42170  0.01382 139.00228 392.403 < 2e-16 ***
## stateambient -0.03685  0.01050  24.17695 -3.510 0.001783 **
## year_factor2 -0.11588  0.01732 141.71838 -6.689 4.80e-10 ***
## year_factor3 -0.12674  0.01732 141.71838 -7.316 1.73e-11 ***
## year_factor4 -0.14791  0.01732 141.71838 -8.538 1.91e-14 ***
## year_factor5 -0.09502  0.01732 141.71838 -5.485 1.84e-07 ***
## year_factor6 -0.05978  0.01751 142.51177 -3.414 0.000833 ***
## year_factor7 -0.17002  0.01751 142.51177 -9.710 < 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
## stateambint -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          0.04237 0.042368     1  24.177 12.320 0.001783 **
## year_factor  0.44634 0.074391     6 141.037 21.632 < 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  0.03685 0.0110 26.1  3.359  0.1000
##  warmed 1 - warmed 2  0.11588 0.0177 147.8  6.541 <.0001
##  warmed 1 - ambient 2  0.15273 0.0208 154.8  7.329 <.0001
##  warmed 1 - warmed 3  0.12674 0.0177 147.8  7.154 <.0001
##  warmed 1 - ambient 3  0.16359 0.0208 154.8  7.850 <.0001
##  warmed 1 - warmed 4  0.14791 0.0177 147.8  8.349 <.0001
##  warmed 1 - ambient 4  0.18476 0.0208 154.8  8.866 <.0001
##  warmed 1 - warmed 5  0.09502 0.0177 147.8  5.364 <.0001
##  warmed 1 - ambient 5  0.13187 0.0208 154.8  6.328 <.0001
##  warmed 1 - warmed 6  0.05978 0.0179 148.7  3.337  0.0617
##  warmed 1 - ambient 6  0.09664 0.0211 155.4  4.578  0.0008
##  warmed 1 - warmed 7  0.17002 0.0179 148.7  9.491 <.0001
##  warmed 1 - ambient 7  0.20687 0.0211 155.4  9.801 <.0001
##  ambient 1 - warmed 2  0.07903 0.0208 154.9  3.793  0.0148
##  ambient 1 - ambient 2  0.11588 0.0177 147.8  6.541 <.0001
##  ambient 1 - warmed 3  0.08988 0.0208 154.9  4.314  0.0022
##  ambient 1 - ambient 3  0.12674 0.0177 147.8  7.154 <.0001
##  ambient 1 - warmed 4  0.11106 0.0208 154.9  5.330 <.0001
##  ambient 1 - ambient 4  0.14791 0.0177 147.8  8.349 <.0001
##  ambient 1 - warmed 5  0.05817 0.0208 154.9  2.792  0.2400
##  ambient 1 - ambient 5  0.09502 0.0177 147.8  5.364 <.0001
##  ambient 1 - warmed 6  0.02293 0.0209 155.2  1.097  0.9980
##  ambient 1 - ambient 6  0.05978 0.0179 148.7  3.337  0.0617
##  ambient 1 - warmed 7  0.13317 0.0209 155.2  6.371 <.0001
##  ambient 1 - ambient 7  0.17002 0.0179 148.7  9.491 <.0001
##  warmed 2 - ambient 2  0.03685 0.0110 26.1  3.359  0.1000
##  warmed 2 - warmed 3  0.01086 0.0173 146.3  0.628  1.0000
##  warmed 2 - ambient 3  0.04771 0.0205 153.4  2.329  0.5367
##  warmed 2 - warmed 4  0.03203 0.0173 146.3  1.851  0.8485
##  warmed 2 - ambient 4  0.06888 0.0205 153.4  3.362  0.0572
##  warmed 2 - warmed 5 -0.02086 0.0173 146.3 -1.205  0.9950
##  warmed 2 - ambient 5  0.01599 0.0205 153.4  0.781  1.0000

```

```

## warmed 2 - warmed 6 -0.05609 0.0175 147.1 -3.205 0.0892
## warmed 2 - ambient 6 -0.01924 0.0208 154.2 -0.927 0.9997
## warmed 2 - warmed 7 0.05414 0.0175 147.1 3.094 0.1193
## warmed 2 - ambient 7 0.09100 0.0208 154.2 4.383 0.0017
## ambient 2 - warmed 3 -0.02599 0.0205 153.4 -1.269 0.9919
## ambient 2 - ambient 3 0.01086 0.0173 146.3 0.628 1.0000
## ambient 2 - warmed 4 -0.00482 0.0205 153.4 -0.235 1.0000
## ambient 2 - ambient 4 0.03203 0.0173 146.3 1.851 0.8485
## ambient 2 - warmed 5 -0.05771 0.0205 153.4 -2.817 0.2275
## ambient 2 - ambient 5 -0.02086 0.0173 146.3 -1.205 0.9950
## ambient 2 - warmed 6 -0.09295 0.0206 153.9 -4.522 0.0010
## ambient 2 - ambient 6 -0.05609 0.0175 147.1 -3.205 0.0892
## ambient 2 - warmed 7 0.01729 0.0206 153.9 0.841 0.9999
## ambient 2 - ambient 7 0.05414 0.0175 147.1 3.094 0.1193
## warmed 3 - ambient 3 0.03685 0.0110 26.1 3.359 0.1000
## warmed 3 - warmed 4 0.02117 0.0173 146.3 1.224 0.9942
## warmed 3 - ambient 4 0.05802 0.0205 153.4 2.832 0.2201
## warmed 3 - warmed 5 -0.03172 0.0173 146.3 -1.833 0.8574
## warmed 3 - ambient 5 0.00514 0.0205 153.4 0.251 1.0000
## warmed 3 - warmed 6 -0.06695 0.0175 147.1 -3.825 0.0134
## warmed 3 - ambient 6 -0.03010 0.0208 154.2 -1.450 0.9741
## warmed 3 - warmed 7 0.04329 0.0175 147.1 2.473 0.4338
## warmed 3 - ambient 7 0.08014 0.0208 154.2 3.860 0.0118
## ambient 3 - warmed 4 -0.01568 0.0205 153.4 -0.765 1.0000
## ambient 3 - ambient 4 0.02117 0.0173 146.3 1.224 0.9942
## ambient 3 - warmed 5 -0.06857 0.0205 153.4 -3.347 0.0598
## ambient 3 - ambient 5 -0.03172 0.0173 146.3 -1.833 0.8574
## ambient 3 - warmed 6 -0.10380 0.0206 153.9 -5.050 0.0001
## ambient 3 - ambient 6 -0.06695 0.0175 147.1 -3.825 0.0134
## ambient 3 - warmed 7 0.00643 0.0206 153.9 0.313 1.0000
## ambient 3 - ambient 7 0.04329 0.0175 147.1 2.473 0.4338
## warmed 4 - ambient 4 0.03685 0.0110 26.1 3.359 0.1000
## warmed 4 - warmed 5 -0.05289 0.0173 146.3 -3.057 0.1309
## warmed 4 - ambient 5 -0.01604 0.0205 153.4 -0.783 0.9999
## warmed 4 - warmed 6 -0.08812 0.0175 147.1 -5.035 0.0001
## warmed 4 - ambient 6 -0.05127 0.0208 154.2 -2.470 0.4359
## warmed 4 - warmed 7 0.02211 0.0175 147.1 1.263 0.9922
## warmed 4 - ambient 7 0.05897 0.0208 154.2 2.840 0.2161
## ambient 4 - warmed 5 -0.08974 0.0205 153.4 -4.380 0.0017
## ambient 4 - ambient 5 -0.05289 0.0173 146.3 -3.057 0.1309
## ambient 4 - warmed 6 -0.12498 0.0206 153.9 -6.080 <.0001
## ambient 4 - ambient 6 -0.08812 0.0175 147.1 -5.035 0.0001
## ambient 4 - warmed 7 -0.01474 0.0206 153.9 -0.717 1.0000
## ambient 4 - ambient 7 0.02211 0.0175 147.1 1.263 0.9922
## warmed 5 - ambient 5 0.03685 0.0110 26.1 3.359 0.1000
## warmed 5 - warmed 6 -0.03524 0.0175 147.1 -2.013 0.7570
## warmed 5 - ambient 6 0.00162 0.0208 154.2 0.078 1.0000
## warmed 5 - warmed 7 0.07500 0.0175 147.1 4.285 0.0026
## warmed 5 - ambient 7 0.11185 0.0208 154.2 5.388 <.0001
## ambient 5 - warmed 6 -0.07209 0.0206 153.9 -3.507 0.0371
## ambient 5 - ambient 6 -0.03524 0.0175 147.1 -2.013 0.7570
## ambient 5 - warmed 7 0.03815 0.0206 153.9 1.856 0.8462
## ambient 5 - ambient 7 0.07500 0.0175 147.1 4.285 0.0026
## warmed 6 - ambient 6 0.03685 0.0110 26.1 3.359 0.1000

```

```

##  warmed 6 - warmed 7   0.11024 0.0177 146.3  6.237 <.0001
##  warmed 6 - ambient 7  0.14709 0.0208 155.6  7.070 <.0001
##  ambient 6 - warmed 7  0.07339 0.0208 155.6  3.528 0.0348
##  ambient 6 - ambient 7  0.11024 0.0177 146.3  6.237 <.0001
##  warmed 7 - ambient 7  0.03685 0.0110  26.1   3.359 0.1000
##
## 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 -953.81 -922.88 484.90  -969.81
## mod12p    10 -951.67 -913.00 485.83  -971.67 1.8597  2      0.3946

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 -959.48 -932.41 486.74  -973.48
## mod13p     8 -953.81 -922.88 484.90  -969.81      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   -932.4    486.7    -973.5      346
##
## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -3.2397 -0.5653  0.0968  0.6404  2.9846
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 1.423e-05 0.003772
## Residual           3.700e-03 0.060830
## Number of obs: 353, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|) 
## (Intercept) 5.566419  0.009448 223.083716 589.180 <2e-16 ***
## statewarmed 0.006302  0.006692  23.289336  0.942  0.3560
## originBoth  -0.429279  0.009280 344.456110 -46.259 <2e-16 ***
## originExotic -0.326543  0.007266 333.981948 -44.942 <2e-16 ***
## year_factor  -0.004011  0.001753 338.187599 -2.288  0.0227 * 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttwrm orgnBt orgnEx
## statewarmed -0.367
## originBoth  -0.402  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       0.0033  0.0033     1   23.29  0.8869 0.35599
## origin      10.6672  5.3336     2 341.59 1441.3991 < 2e-16 ***
## year_factor  0.0194  0.0194     1 338.19   5.2358 0.02274 *
## ---
## 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 0.006725  88.41    5.537  5.564
## warmed  Native      3.929  5.557 0.006567  80.45    5.544  5.570
## ambient Both       3.929  5.121 0.008139 148.17    5.105  5.137
## warmed  Both       3.929  5.128 0.008713 174.38    5.110  5.145
## ambient Exotic    3.929  5.224 0.006084  62.19    5.212  5.236
## warmed  Exotic    3.929  5.230 0.006079  62.28    5.218  5.243
##
```

```

## 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
## ambient Native 3.92917847025496 - warmed Native 3.92917847025496 -0.0063
## ambient Native 3.92917847025496 - ambient Both 3.92917847025496 0.4293
## ambient Native 3.92917847025496 - warmed Both 3.92917847025496 0.4230
## ambient Native 3.92917847025496 - ambient Exotic 3.92917847025496 0.3265
## ambient Native 3.92917847025496 - warmed Exotic 3.92917847025496 0.3202
## warmed Native 3.92917847025496 - ambient Both 3.92917847025496 0.4356
## warmed Native 3.92917847025496 - warmed Both 3.92917847025496 0.4293
## warmed Native 3.92917847025496 - ambient Exotic 3.92917847025496 0.3328
## warmed Native 3.92917847025496 - warmed Exotic 3.92917847025496 0.3265
## ambient Both 3.92917847025496 - warmed Both 3.92917847025496 -0.0063
## ambient Both 3.92917847025496 - ambient Exotic 3.92917847025496 -0.1027
## ambient Both 3.92917847025496 - warmed Exotic 3.92917847025496 -0.1090
## warmed Both 3.92917847025496 - ambient Exotic 3.92917847025496 -0.0964
## warmed Both 3.92917847025496 - warmed Exotic 3.92917847025496 -0.1027
## ambient Exotic 3.92917847025496 - warmed Exotic 3.92917847025496 -0.0063
##      SE   df t.ratio p.value
## 0.00702 26.4 -0.898 0.9436
## 0.00935 348.0 45.891 <.0001
## 0.01219 179.0 34.706 <.0001
## 0.00731 337.4 44.697 <.0001
## 0.01023 115.2 31.300 <.0001
## 0.01118 144.0 38.957 <.0001
## 0.00935 348.0 45.891 <.0001
## 0.01003 106.7 33.184 <.0001
## 0.00731 337.4 44.697 <.0001
## 0.00702 26.4 -0.898 0.9436
## 0.00897 350.5 -11.449 <.0001
## 0.01096 134.5 -9.951 <.0001
## 0.01181 162.3 -8.164 <.0001
## 0.00897 350.5 -11.449 <.0001
## 0.00702 26.4 -0.898 0.9436
##
## 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 0.004989 26.78    5.288    5.309
## warmed  5.305 0.005234 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 -0.0063 0.00702 26.4 -0.898  0.3774
##
## 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 0.005644 152.6    5.543    5.565
## Both    5.125 0.007665 258.7    5.109    5.140
## Exotic  5.227 0.004967 107.2    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  45.891 <.0001
## Native - Exotic   0.327 0.00731 337  44.697 <.0001
## Both - Exotic    -0.103 0.00897 351 -11.449 <.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 -339.15 -308.12 177.57   -355.15
## mod15p    10 -335.65 -296.87 177.82   -355.65 0.5005  2      0.7786
```

```
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 -344.60 -317.46 179.30   -358.60
## mod16p     8 -339.15 -308.12 177.57   -355.15      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
## -344.6   -317.5    179.3    -358.6      350
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.4528 -0.3872 -0.0961  0.5183  2.0152
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.00000  0.0000
## Residual           0.02144  0.1464
## Number of obs: 357, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept) 5.321938  0.022476 357.000000 236.785 <2e-16 ***
## statewarmed 0.033945  0.015576 357.000000   2.179  0.0300 *
## growth_habit -0.251555  0.022191 357.000000 -11.336 <2e-16 ***
## growth_habitGraminoid -0.240385  0.017326 357.000000 -13.875 <2e-16 ***
## year_factor   0.007848  0.004211 357.000000   1.864  0.0632 .
## ---
## 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_habt -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      0.1018 0.10183     1    357  4.7491 0.02997 *
## growth_habit 4.8685 2.43423     2    357 113.5215 < 2e-16 ***
## year_factor  0.0745 0.07447     1    357  3.4729 0.06320 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

## $`emmeans of state, growth_habit, year_factor`
## state   growth_habit year_factor emmean      SE    df lower.CL upper.CL
## ambient Forb            3.94  5.35 0.0155  84.0    5.32    5.38
## warmed  Forb            3.94  5.39 0.0153  78.0    5.36    5.42
## ambient            3.94  5.10 0.0193 146.8    5.06    5.14
## warmed             3.94  5.14 0.0206 172.4    5.09    5.18
## ambient Graminoid       3.94  5.11 0.0142  61.0    5.08    5.14
## warmed  Graminoid       3.94  5.15 0.0142  61.5    5.12    5.17
##
## 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
## -0.0339 0.0161 24.6 -2.115 0.3127
## 0.2516 0.0224 354.6 11.234 <.0001

```

```

##      0.2176 0.0287 175.8  7.590 <.0001
##      0.2404 0.0174 341.8 13.788 <.0001
##      0.2064 0.0239 112.9  8.655 <.0001
##      0.2855 0.0264 142.7 10.822 <.0001
##      0.2516 0.0224 354.6 11.234 <.0001
##      0.2743 0.0235 105.5 11.651 <.0001
##      0.2404 0.0174 341.8 13.788 <.0001
##     -0.0339 0.0161  24.6 -2.115  0.3127
##     -0.0112 0.0216 356.6 -0.517  0.9955
##     -0.0451 0.0259 134.3 -1.745  0.5048
##      0.0228 0.0279 160.8  0.816  0.9643
##     -0.0112 0.0216 356.6 -0.517  0.9955
##     -0.0339 0.0161  24.6 -2.115  0.3127
##
## 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")
```

```

## $`emmeans of growth_habit`
##   growth_habit emmean       SE   df lower.CL upper.CL
##   Forb          5.370 0.01314 148.8    5.344   5.396
##   Graminoid     5.118 0.01828 256.6    5.082   5.154
##   Graminoid     5.129 0.01170 107.9    5.106   5.153
##
## 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 -          0.2516 0.0224 355 11.234 <.0001
##   Forb - Graminoid 0.2404 0.0174 342 13.788 <.0001
##   - Graminoid    -0.0112 0.0216 357 -0.517  0.8630
##
## 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|)                  < 2e-16 ***
## (Intercept)                  0.102267
## stateambient                  0.061214 .
## year_factor4                  0.000422 ***
## year_factor5                  0.246527
## year_factor6                  0.167866
## insecticideno_insects       0.677402

```

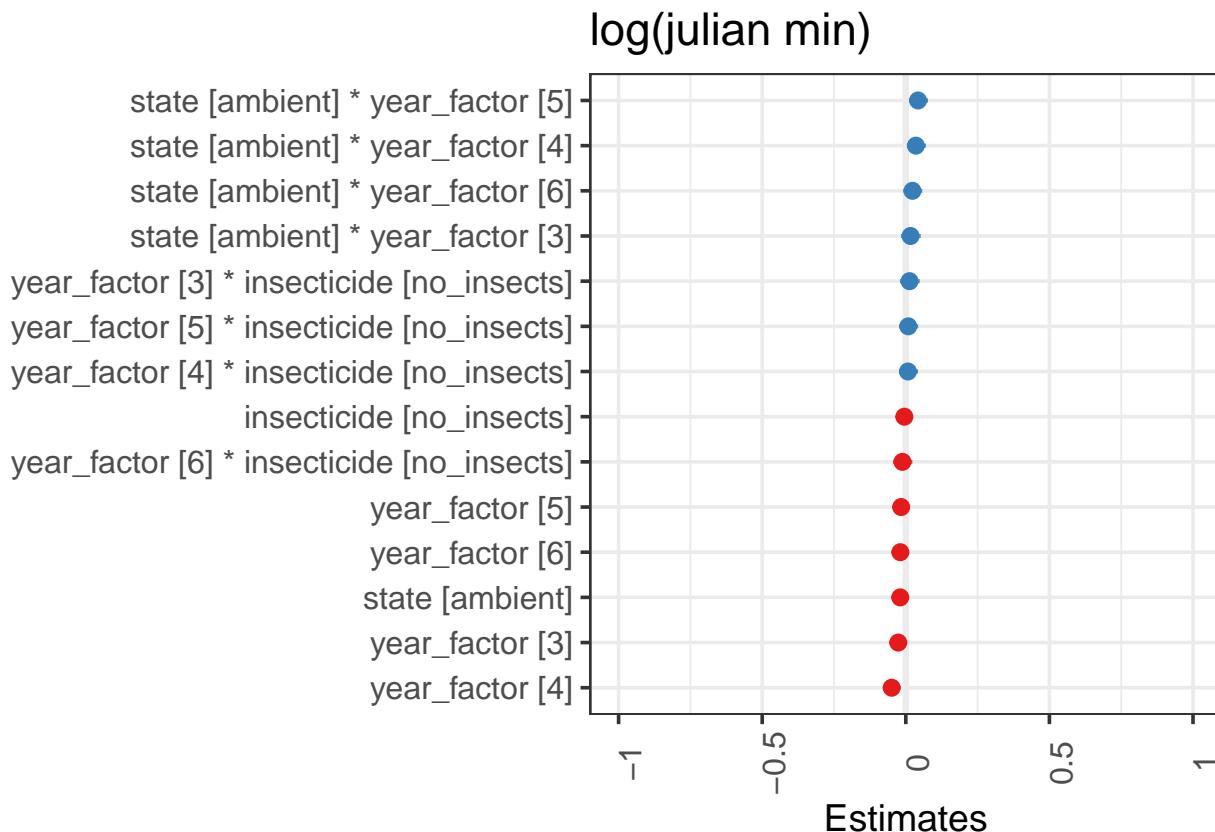
```

## 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)

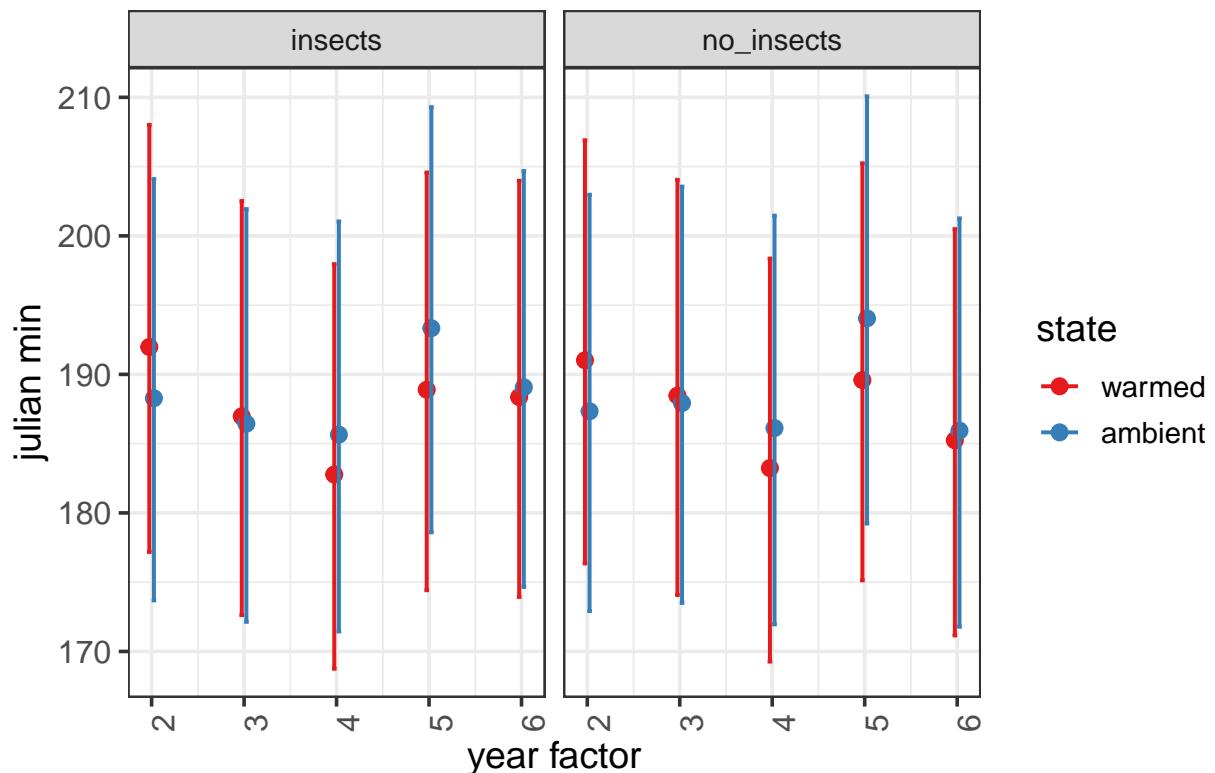
```



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

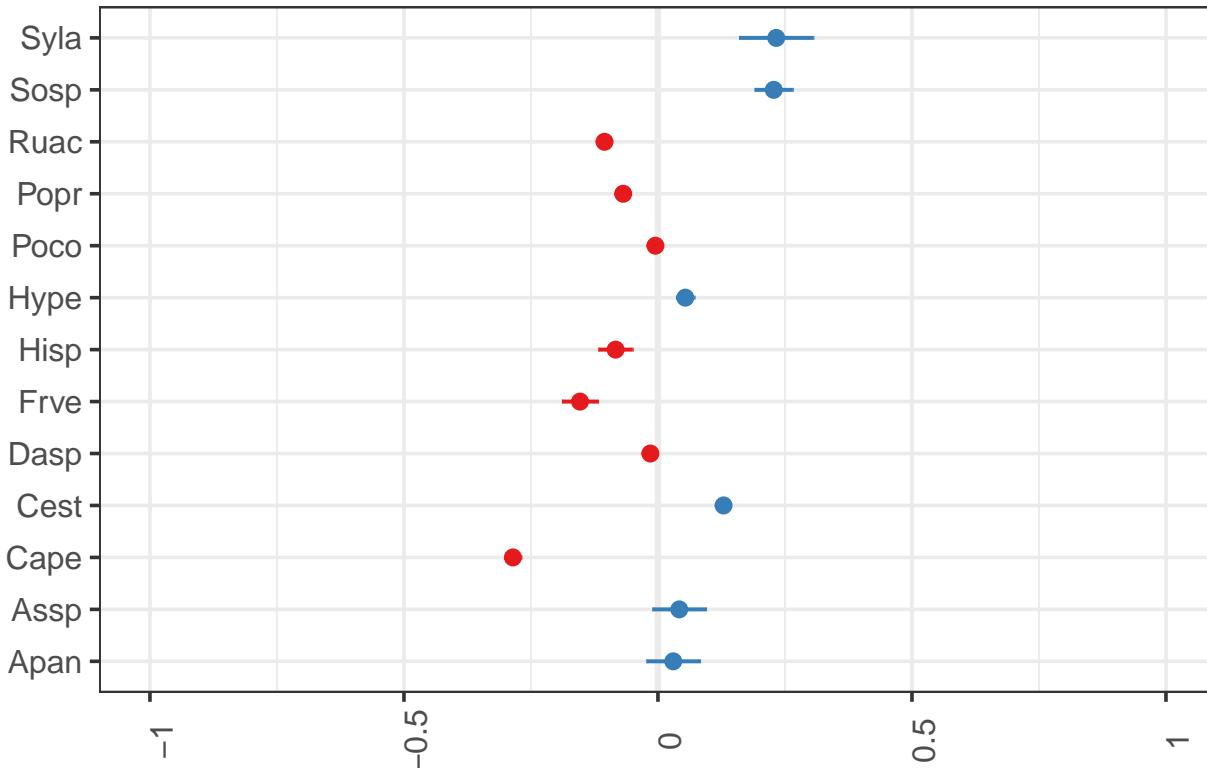
## Model has log-transformed response. Back-transforming predictions to original response scale. Standard errors are back-transformed and may not be accurate.

## 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
```

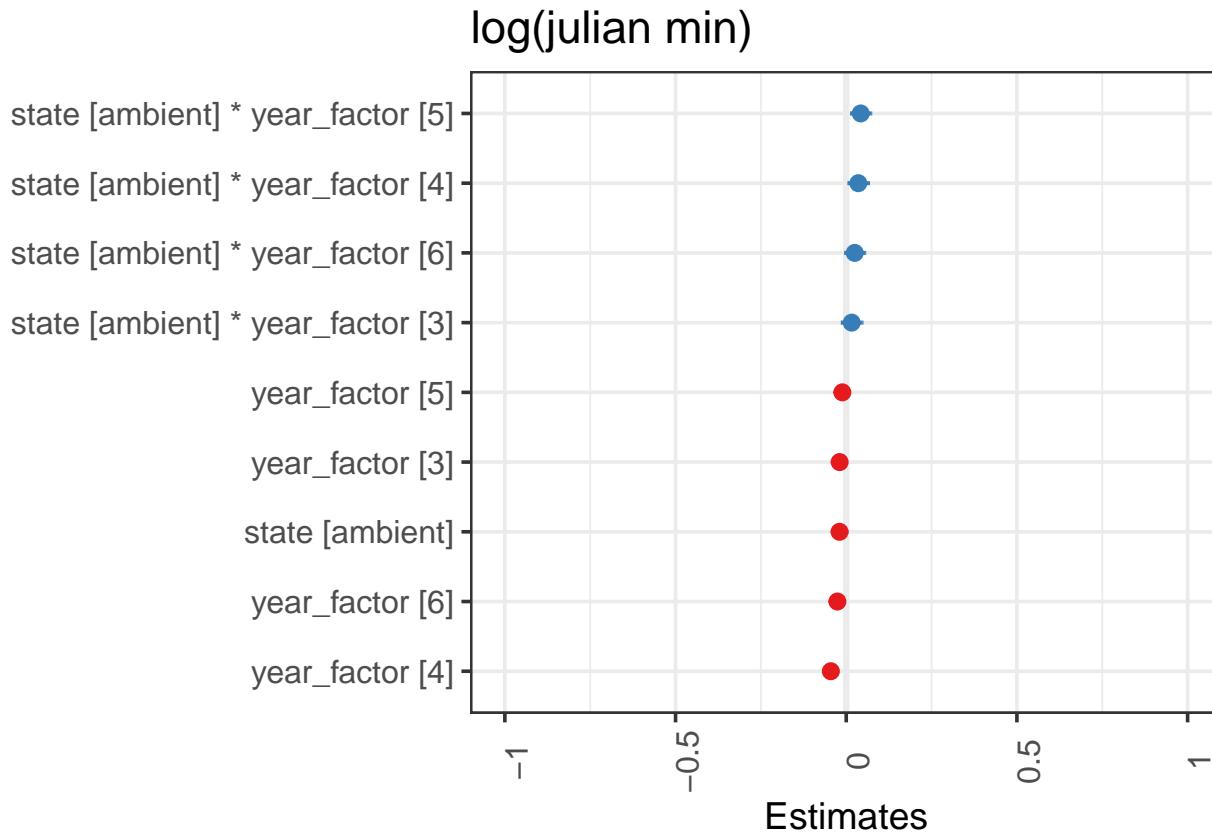
```
anova(mod3u, mod5u) # more complex model is better so go with mod3u
```

```
## 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
```

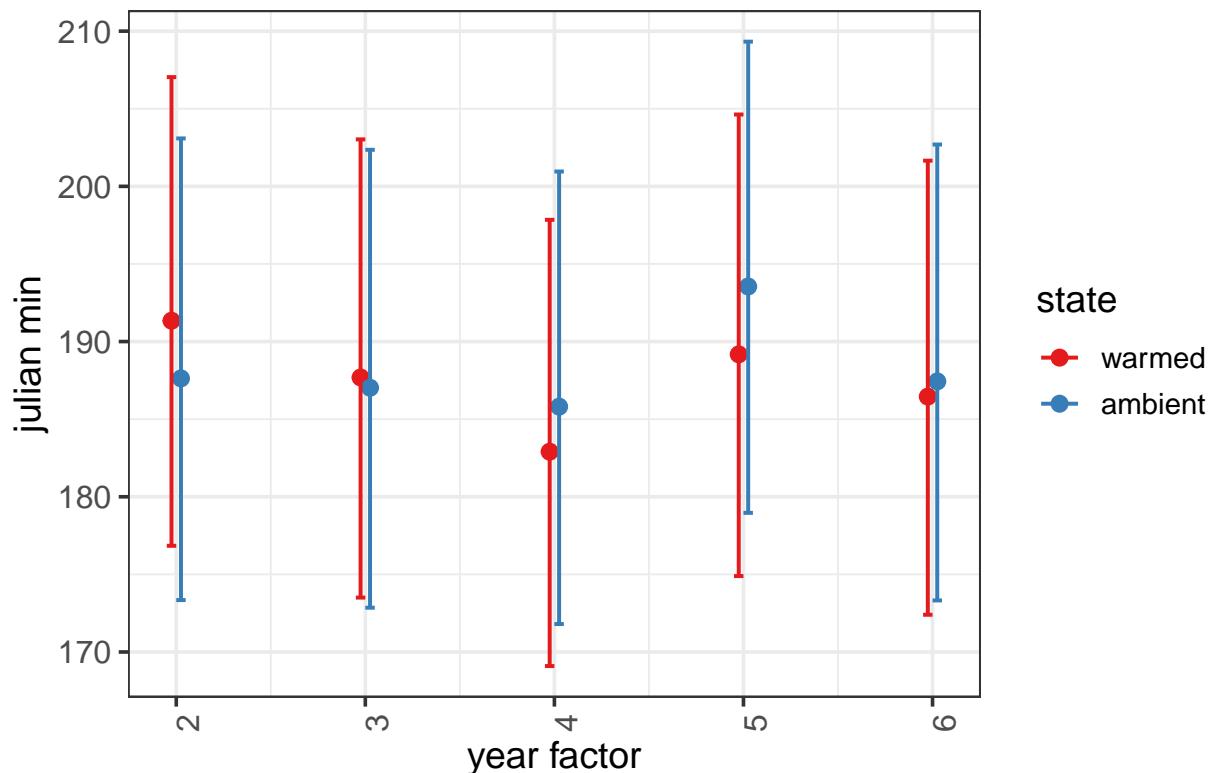
```
# Plot the fixed effects estimates for different models these are the fixed
# effects estimates from summary(mod5)
plot_model(mod3u, sort.est = TRUE)
```



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

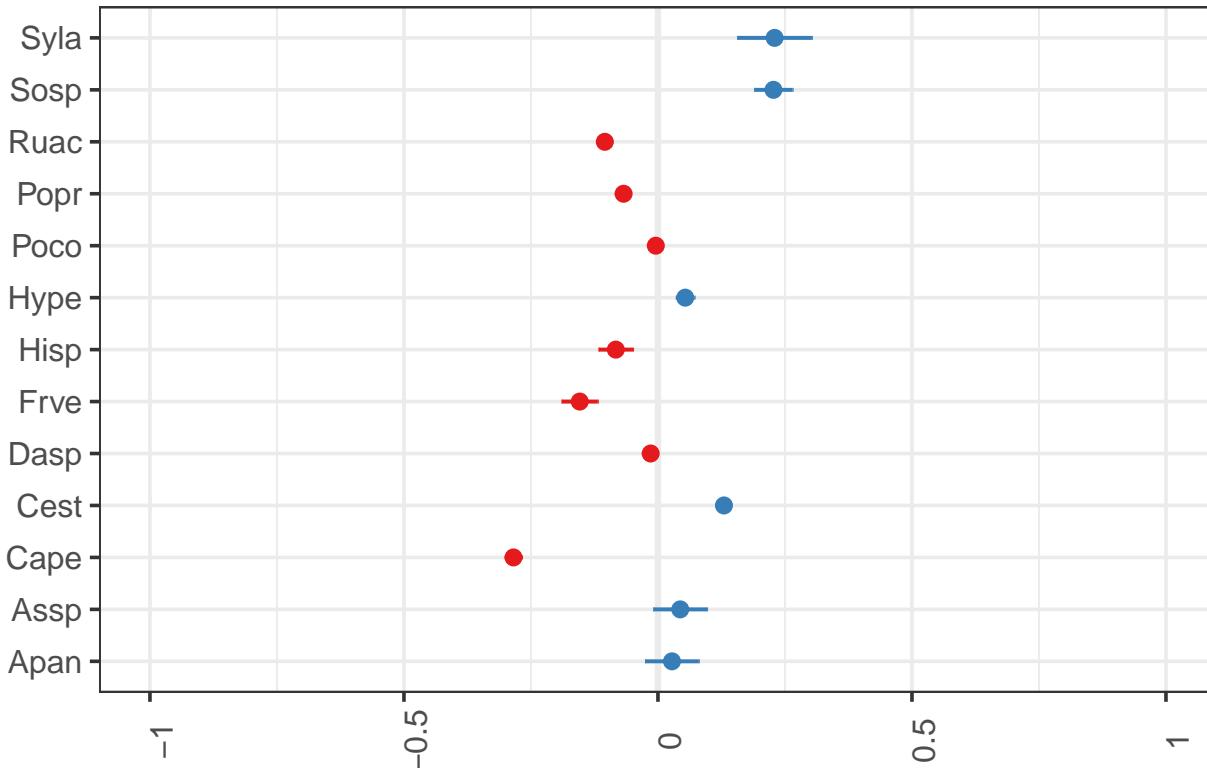
## Model has log-transformed response. Back-transforming predictions to original response scale. Standard errors are back-transformed and may not be accurate.

## 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 from the others.
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## state     0.0037 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 minimum 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

```

```

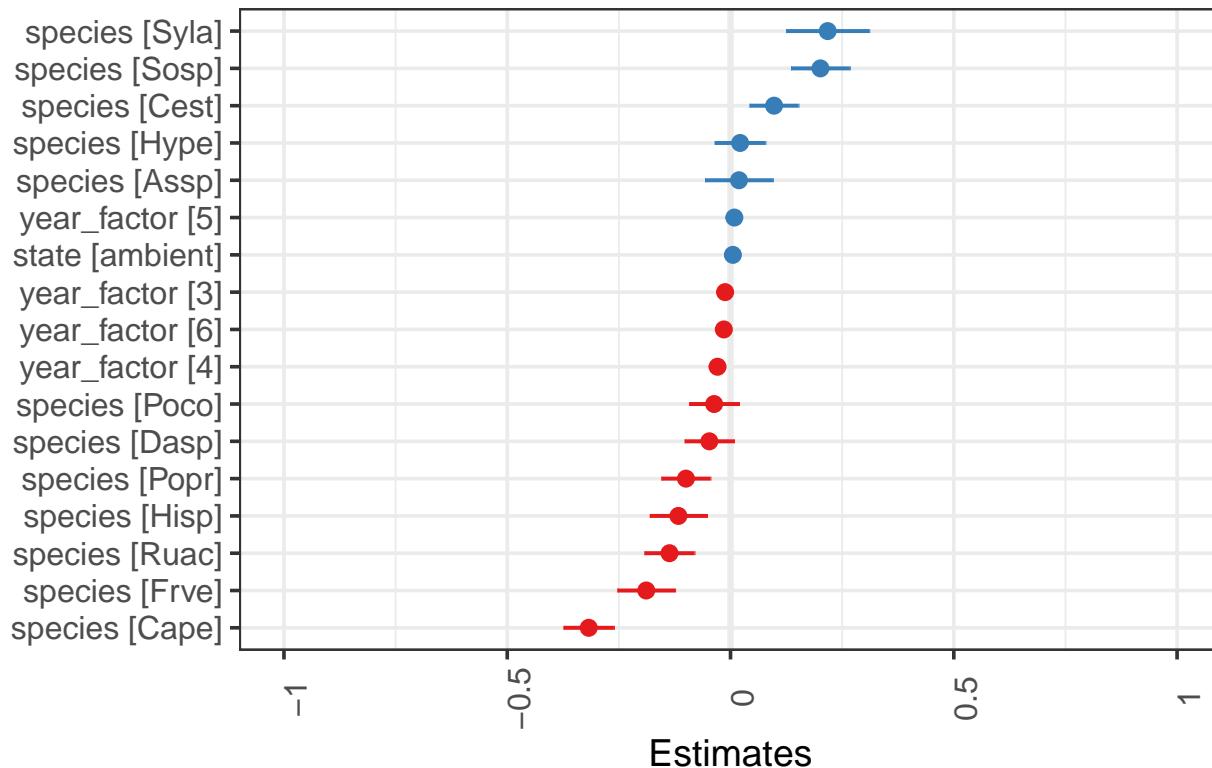
##  Dasp - Hype -0.06893 0.01149 553  -5.998 <.0001
##  Dasp - Poco -0.01073 0.00965 554  -1.112 0.9965
##  Dasp - Popr  0.05237 0.00810 548   6.465 <.0001
##  Dasp - Ruac  0.08921 0.00987 554   9.043 <.0001
##  Dasp - Sosp -0.24879 0.02097 396  -11.865 <.0001
##  Dasp - Syla -0.26503 0.04042 541  -6.557 <.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)
```

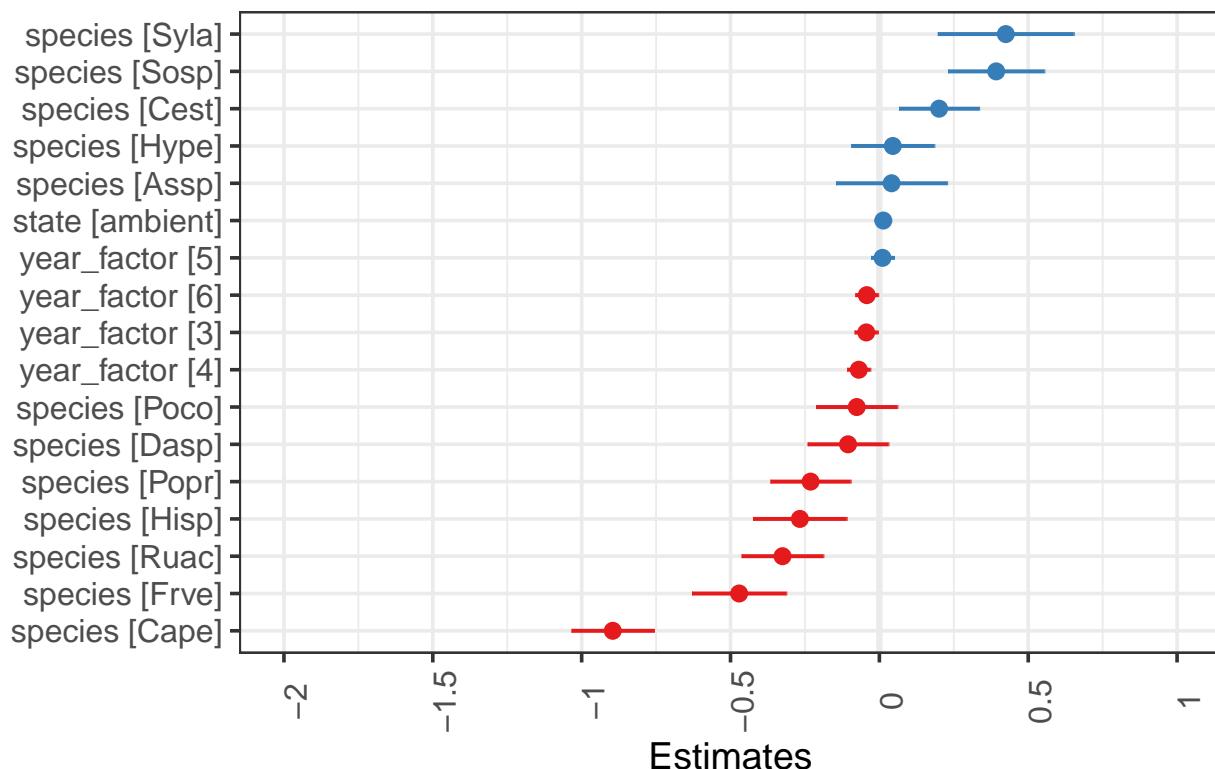
## log(julian min)



```
# 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
```

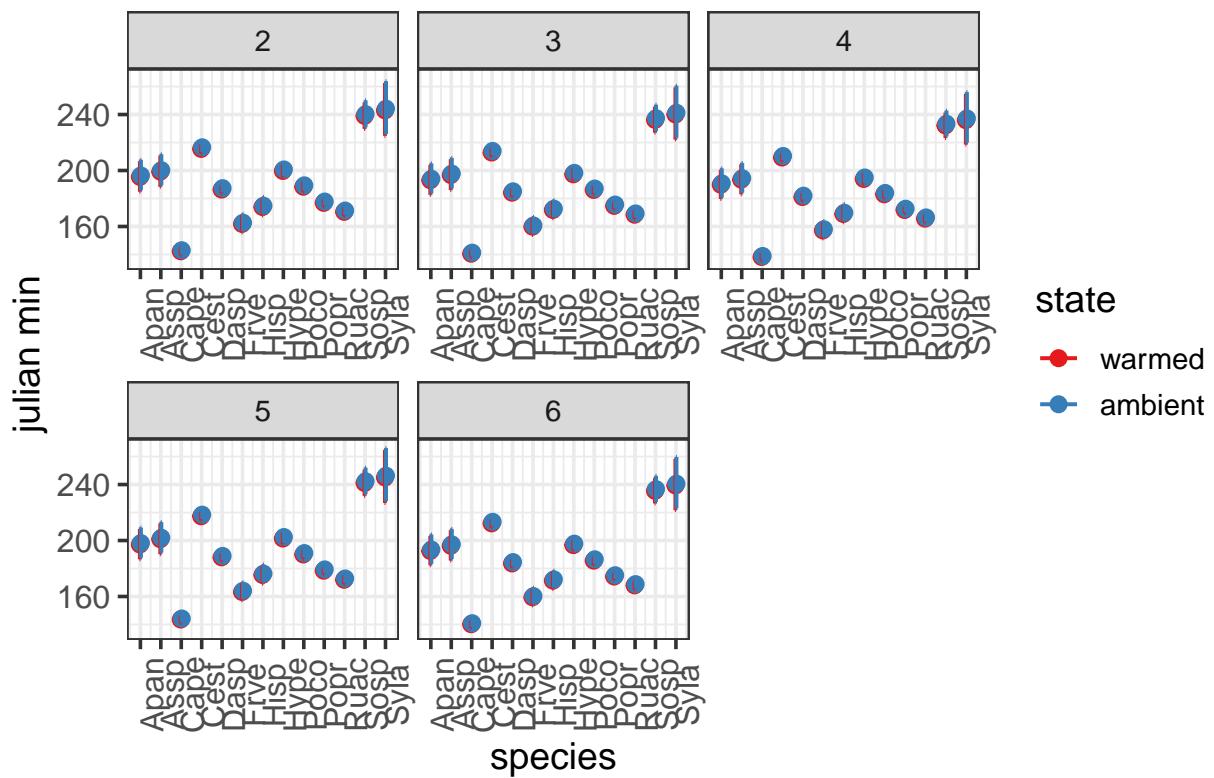
## log(julian min)



```
# these are the fixed predicted values: - note this is a new plot
plot_model(mod7au, type = "pred", terms = c("species", "state", "year_factor"))
```

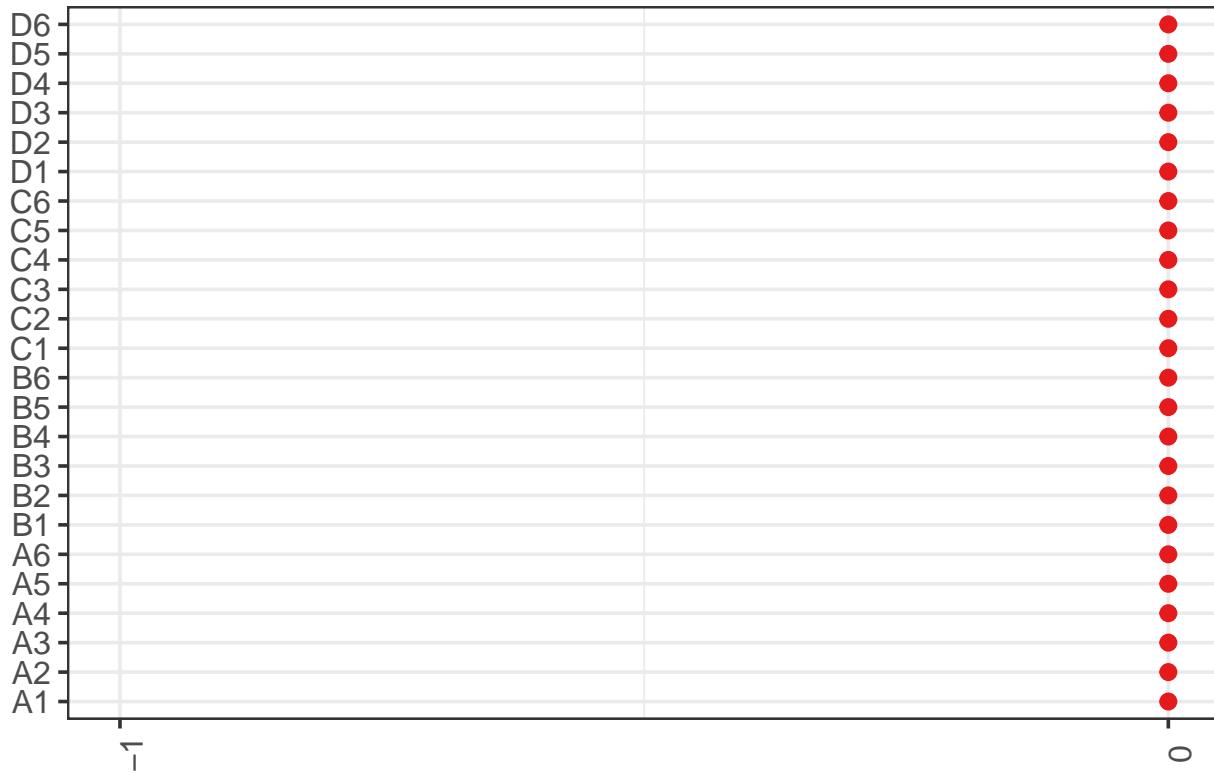
```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standard
```

## Predicted values of julian min



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

## Random effects



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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

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

```

anova(mod9u, mod9au) # mod 9au

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

summary(mod9au)

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

```

```

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

anova(mod9au)

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

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

## boundary (singular) fit: see ?isSingular

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

```

```

## 
## Results are averaged over the levels of: year_factor
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 6 estimates

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 0.008869 345.58    5.531    5.566
##   Both    5.146 0.012894 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 26.689 <.0001
##   Native - Exotic   0.3207 0.00986 823 32.518 <.0001
##   Both - Exotic    -0.0822 0.01339 823 -6.134 <.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

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 -936.63 -781.50 501.31   -1002.6
## mod10u    35 -933.15 -768.62 501.57   -1003.1 0.5171  2     0.7722

```

```

anova(mod11u, mod11au) # model 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 -1031.46 -975.05 527.73   -1055.5
## mod11u    33 -936.63 -781.50 501.31   -1002.6      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
## -1031.5   -975.1   527.7   -1055.5      801
##
## Scaled residuals:
##       Min     1Q Median     3Q    Max
## -5.3663 -0.6249 -0.0292  0.7473  3.2414
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   plot     (Intercept) 0.00000  0.0000
##   Residual            0.01598  0.1264
## Number of obs: 813, groups:  plot, 24
##
## Fixed effects:
##                   Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)      5.591815  0.020735 813.000000 269.686 < 2e-16 ***
## stateambient    -0.018607  0.008976 813.000000 -2.073  0.0385 *
## growth_habit    -0.298651  0.015748 813.000000 -18.964 < 2e-16 ***
## growth_habitGraminoid -0.251892  0.009864 813.000000 -25.537 < 2e-16 ***
## year_factor2    -0.164920  0.021064 813.000000 -7.830 1.52e-14 ***
## year_factor3    -0.185958  0.021552 813.000000 -8.628 < 2e-16 ***
## year_factor4    -0.154714  0.022040 813.000000 -7.020 4.69e-12 ***
## year_factor5    -0.107010  0.022655 813.000000 -4.723 2.73e-06 ***
## year_factor6    -0.093082  0.021707 813.000000 -4.288 2.02e-05 ***
## year_factor7    -0.129739  0.024382 813.000000 -5.321 1.34e-07 ***

```

```

## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttmbn grwth_ grwt_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      0.0687  0.0687     1    813  4.2973 0.03849 *
## growth_habit 12.3382  6.1691     2    813 385.9317 < 2e-16 ***
## year_factor   1.7126  0.2854     6    813 17.8559 < 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 0.008230 186.96    5.447    5.479
##             5.165 0.014672 599.35    5.136    5.193
##   Graminoid    5.211 0.006248  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 0.01587 821 18.814 <.0001
##   Forb - Graminoid  0.2519 0.00995 823 25.316 <.0001
##   - Graminoid     -0.0468 0.01563 822 -2.991  0.0080
##
## 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

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

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

## boundary (singular) fit: see ?isSingular

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

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

## boundary (singular) fit: see ?isSingular

```

```

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

```

```

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

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

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

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

summary(mod7au)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + species + factor(year_factor) + (1 |
##           plot)
## Data: umbs_sd_spp
##
##      AIC      BIC      logLik deviance df.resid
## -2581.8 -2436.0   1321.9   -2643.8      785
##
## Scaled residuals:
##      Min      1Q Median      3Q     Max
## -4.5921 -0.5306 -0.0049  0.4909  7.6708
##
## Random effects:
## Groups   Name      Variance Std.Dev.
## plot     (Intercept) 0.000000 0.00000
## Residual            0.002293 0.04789
## Number of obs: 816, groups: plot, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept) 5.452875  0.011990 816.000000 454.793 < 2e-16 ***
## stateambient 0.004370  0.003507 816.000000  1.246 0.213101    
## speciesArel -0.184443  0.010465 816.000000 -17.624 < 2e-16 ***
## speciesBavu -0.230672  0.020305 816.000000 -11.360 < 2e-16 ***
## speciesCahi -0.502774  0.023467 816.000000 -21.425 < 2e-16 ***
## speciesCest  0.039215  0.011618 816.000000  3.375 0.000772 ***
## speciesDaca  0.101541  0.014495 816.000000  7.005 5.16e-12 ***
## speciesDagl -0.163366  0.010831 816.000000 -15.083 < 2e-16 ***
## speciesElre -0.039521  0.011372 816.000000 -3.475 0.000537 ***
## speciesEugr  0.214379  0.013900 816.000000 15.422 < 2e-16 ***
## speciesHisp -0.246588  0.011127 816.000000 -22.161 < 2e-16 ***
## speciesHype -0.109633  0.017723 816.000000 -6.186 9.75e-10 ***
## speciesPhpr -0.054832  0.010547 816.000000 -5.199 2.54e-07 ***
## speciesPoco -0.082002  0.017606 816.000000 -4.658 3.73e-06 ***
## speciesPopr -0.248610  0.010203 816.000000 -24.368 < 2e-16 ***
## speciesPore -0.046105  0.013012 816.000000 -3.543 0.000418 ***
## speciesRusp -0.172216  0.017181 816.000000 -10.024 < 2e-16 ***
## speciesSoca  0.230129  0.010169 816.000000 22.630 < 2e-16 ***

```

```

## speciesSogr      0.123923  0.048964 816.000000  2.531 0.011564 *
## speciesSora     0.173318  0.035168 816.000000  4.928 1.00e-06 ***
## speciesSypi     0.289872  0.035383 816.000000  8.192 9.86e-16 ***
## speciesTaof     -0.451748  0.020315 816.000000 -22.237 < 2e-16 ***
## speciesTrsp     -0.117081  0.014949 816.000000 -7.832 1.49e-14 ***
## factor(year_factor)2 -0.123848  0.008143 816.000000 -15.210 < 2e-16 ***
## factor(year_factor)3 -0.123874  0.008310 816.000000 -14.906 < 2e-16 ***
## factor(year_factor)4 -0.134431  0.008450 816.000000 -15.909 < 2e-16 ***
## factor(year_factor)5 -0.096438  0.008651 816.000000 -11.148 < 2e-16 ***
## factor(year_factor)6 -0.086017  0.008330 816.000000 -10.327 < 2e-16 ***
## factor(year_factor)7 -0.094345  0.009449 816.000000 -9.985 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

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

```

```
anova(mod7au)
```

```

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

```

## 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

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

## boundary (singular) fit: see ?isSingular

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 -355.85 -327.75 185.93   -371.85
## mod15pu    10 -354.67 -319.53 187.33   -374.67 2.811  2     0.2452

anova(mod16pu, mod17pu) # mod 17pu

## Data: umbss_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 -356.19 -331.60 185.09   -370.19
## mod16pu     8 -355.85 -327.75 185.93   -371.85 1.6653  1     0.1969

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
## -356.2   -331.6   185.1   -370.2     241
##
## Scaled residuals:
##    Min      1Q Median      3Q      Max
## -2.0682 -0.9016  0.2706  0.7563  2.6865
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## plot     (Intercept) 0.0008765 0.02961
## Residual           0.0124843 0.11173
## Number of obs: 248, groups: plot, 24
##
## Fixed effects:
##                   Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)      5.271908  0.025405 181.439771 207.517 <2e-16 ***
## statewarmed     -0.004444  0.018670  23.023443 -0.238  0.8140
## growth_habit    -0.066348  0.038721 247.308775 -1.713  0.0879 .
## growth_habitGraminoid -0.157582  0.014489 223.325999 -10.876 <2e-16 ***
## year_factor     -0.009070  0.005067 223.719202 -1.790  0.0748 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sttwrm grwth_ grwt_G
## statewarmed -0.363
## growth_habit -0.196 -0.028
## grwth_hbtGr -0.298  0.000  0.190
## year_factor -0.803 -0.006  0.123  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       0.00071 0.00071     1   23.023  0.0567 0.81397
## growth_habit 1.47825 0.73913     2  236.227 59.2044 < 2e-16 ***
## year_factor  0.04000 0.04000     1  223.719  3.2040 0.07481 .
## ---
## 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      5.24 0.0157  45.1     5.20    5.27
##   warmed   Forb      5.23 0.0157  45.3     5.20    5.26
##   ambient           5.17 0.0402 221.5     5.09    5.25
##   warmed            5.17 0.0397 218.4     5.09    5.24
##   ambient  Graminoid  5.08 0.0157  44.5     5.05    5.11
##   warmed  Graminoid  5.07 0.0157  44.6     5.04    5.11
##
## 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  0.00444 0.0196  26.3  0.227  0.9999
##   ambient Forb - ambient       0.06635 0.0395 253.2  1.678  0.5476
##   ambient Forb - warmed        0.07079 0.0437 187.7  1.621  0.5856
##   ambient Forb - ambient Graminoid 0.15758 0.0146 226.9 10.806 <.0001
##   ambient Forb - warmed Graminoid 0.16203 0.0244  64.8  6.636 <.0001
##   warmed Forb - ambient        0.06190 0.0446 194.7  1.389  0.7338
##   warmed Forb - warmed         0.06635 0.0395 253.2  1.678  0.5476
##   warmed Forb - ambient Graminoid 0.15314 0.0244  64.8  6.271 <.0001
##   warmed Forb - warmed Graminoid 0.15758 0.0146 226.9 10.806 <.0001
##   ambient   - warmed          0.00444 0.0196  26.3  0.227  0.9999
##   ambient   - ambient Graminoid 0.09123 0.0395 253.2  2.309  0.1941
##   ambient   - warmed Graminoid 0.09568 0.0445 194.3  2.148  0.2674
##   warmed   - ambient Graminoid 0.08679 0.0436 187.2  1.989  0.3526
##   warmed   - warmed Graminoid  0.09123 0.0395 253.2  2.309  0.1941
##   ambient Graminoid - warmed Graminoid 0.00444 0.0196  26.3  0.227  0.9999
##
## 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          5.234 0.01232  67.33    5.209    5.258
##               5.167 0.03872 240.20    5.091    5.244
##   Graminoid    5.076 0.01225  65.86    5.052    5.100
##
## 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 -          0.0663 0.0395 253  1.678  0.2156
##  Forb - Graminoid 0.1576 0.0146 227 10.806 <.0001
##  - Graminoid      0.0912 0.0395 253  2.309  0.0563
##
## 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
```