

warmXtrophic Project: Flowering Phenology Analyses

Moriah Young

July 14, 2021

Load in packages & data

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

# Load packages
library(tidyverse)
library(ggplot2)
library(lme4)
library(lmerTest)
library(emmeans)
library(vegan)
library(car)
library(rstatix)
library(scales)
library(fitdistrplus)
library(moments) # for calculating skewness of data
library(ggpubr)
library(jtools) # summ() function
library(predictmeans)
library(olsrr)
library(car)
library(fitdistrplus)
library(ggpubr)
library(interactions)
library(sjPlot)
library(effects)
library(glmmTMB)
library(GGally) # ggpairs() function
library(bbmle) # AICtab() function

# Set working directory
Sys.setenv("L1DIR")
```

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

```
L1_dir <- Sys.setenv("L1DIR")
L2_dir <- Sys.setenv("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
flwr_species <- read.csv(file.path(L2_dir, "phenology/final_flwr_species_L2.csv"))
flwr_plot <- read.csv(file.path(L2_dir, "phenology/final_flwr_plot_L2.csv"))
flwr_species$X <- NULL # get rid of 'X' column that shows up
flwr_plot$X <- NULL # get rid of 'X' column that shows up
View(flwr_species) # take a look at the data to see if looks good
View(flwr_plot) # take a look at the data to see if looks good

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

```

```
## [1] "ambient" "warmed"
```

```

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

```

```
## [1] "warmed" "ambient"
```

```

# [1] 'warmed' 'ambient'

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

```

```
## [1] "ambient" "warmed"
```

```

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

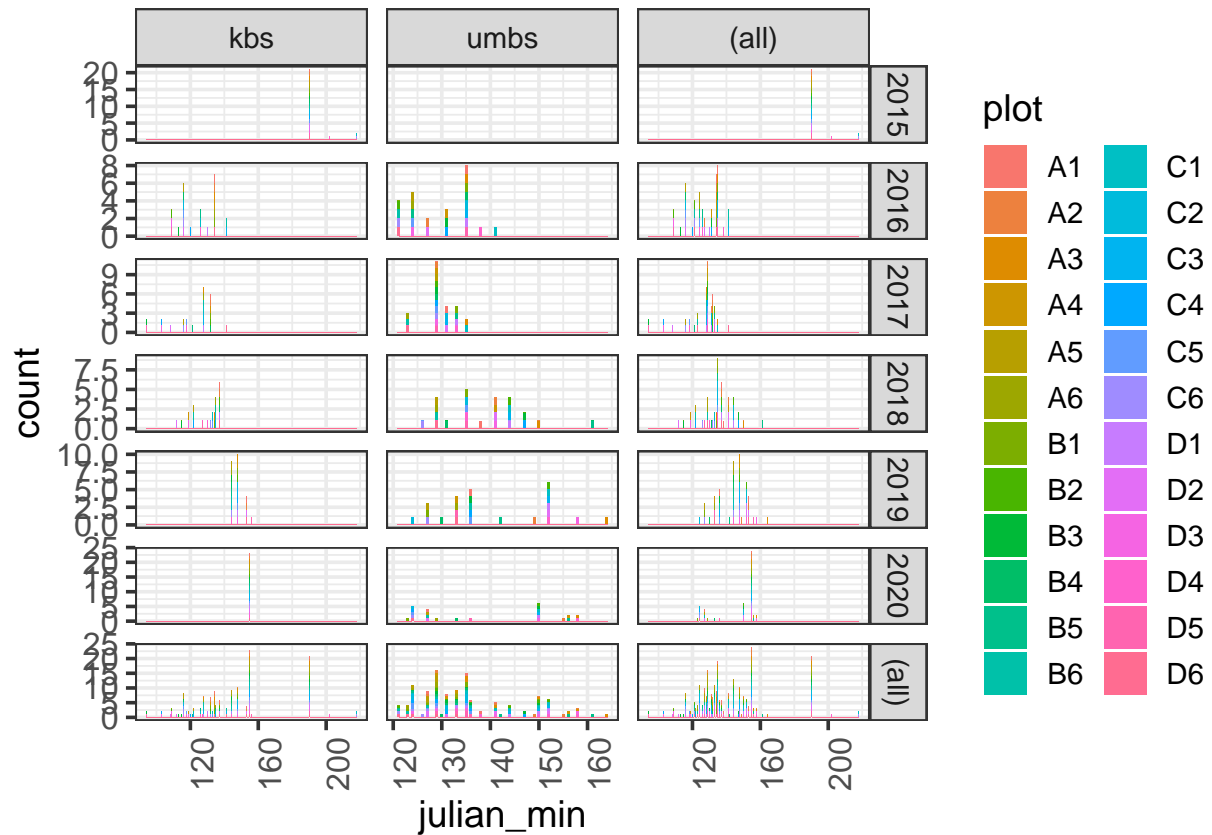
```

```
## [1] "warmed" "ambient"
```

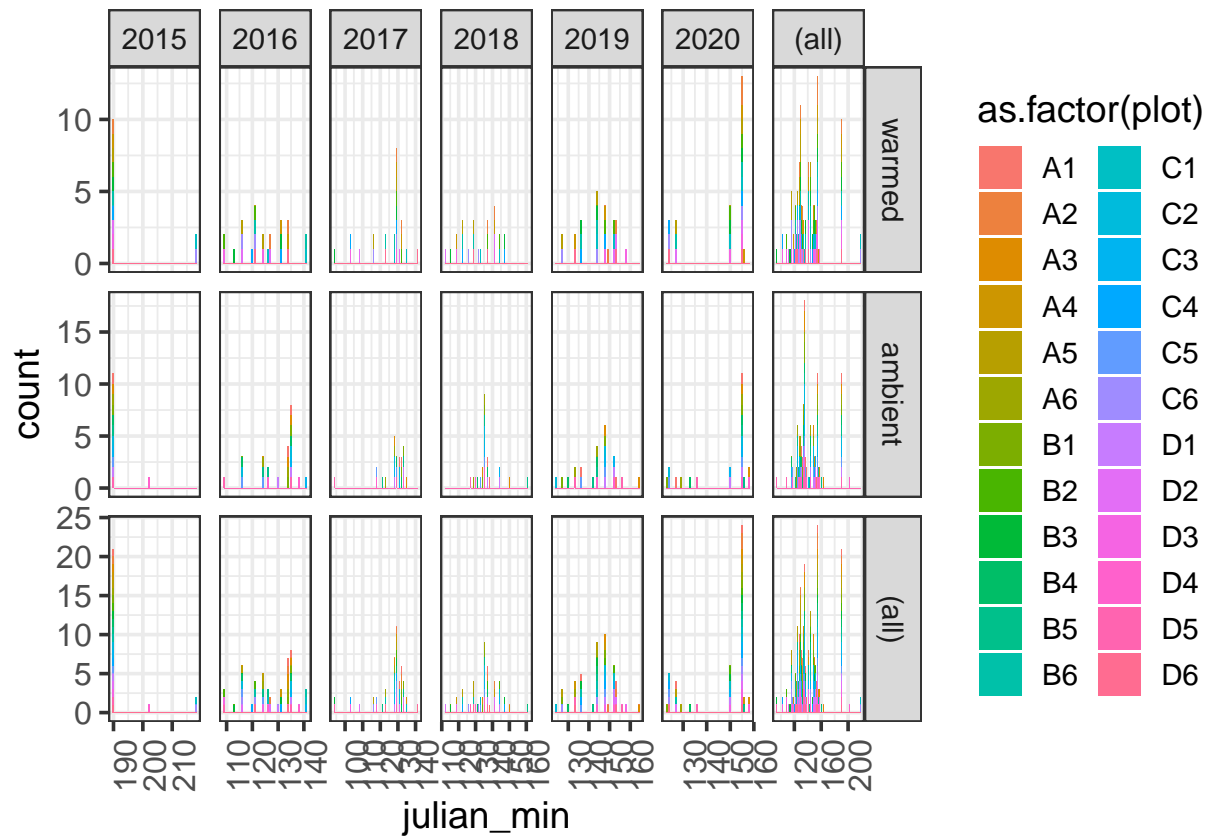
```
# [1] 'warmed' 'ambient'
```

Data exploration for median Julian date of flowering at the PLOT LEVEL

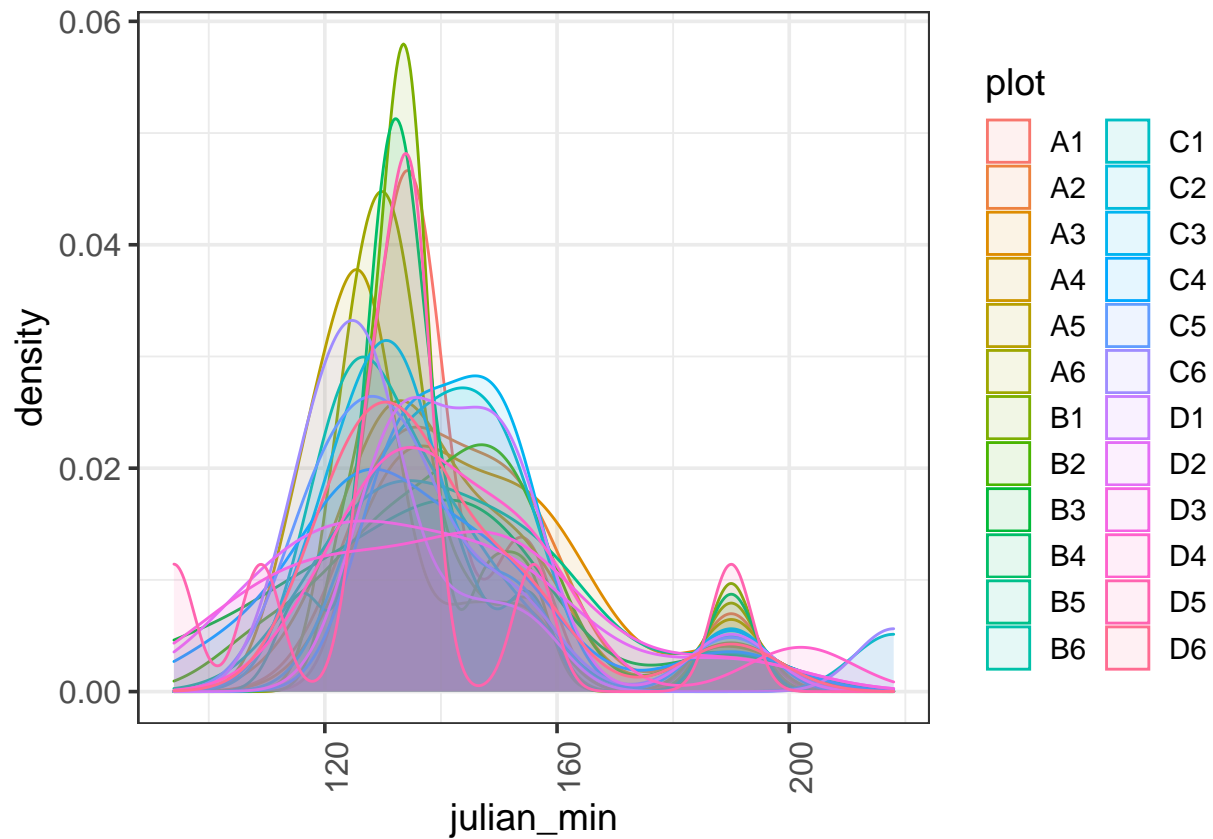
```
# Visualizing avg minimum Julian date for both sites at the PLOT LEVEL
ggplot(flwr_plot, aes(julian_min, fill = plot)) + geom_histogram(binwidth = 0.5) +
  facet_grid(year ~ site, margins = TRUE, scales = "free")
```



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



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



```
ggplot(flwr_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.
```



```
## Inf

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

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

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

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

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

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

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

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

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

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

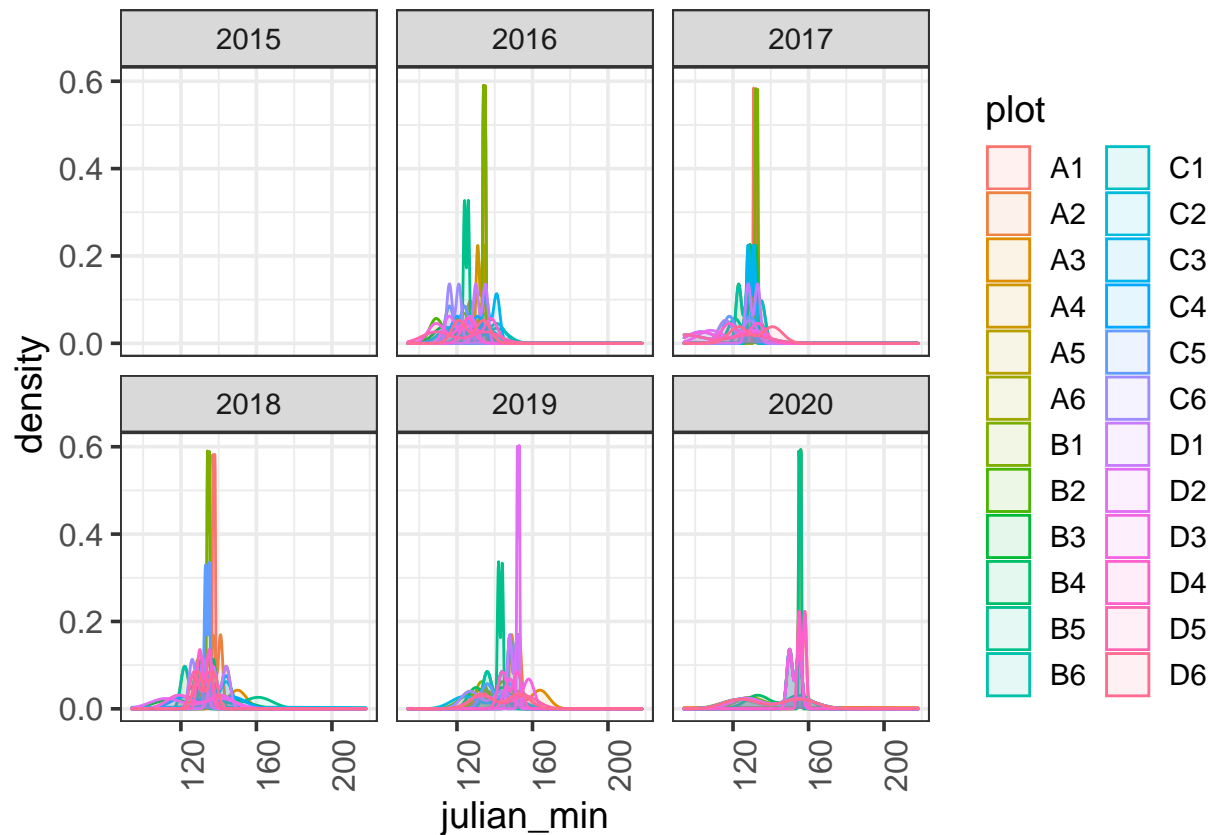
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

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

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



```

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

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

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

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

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

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

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

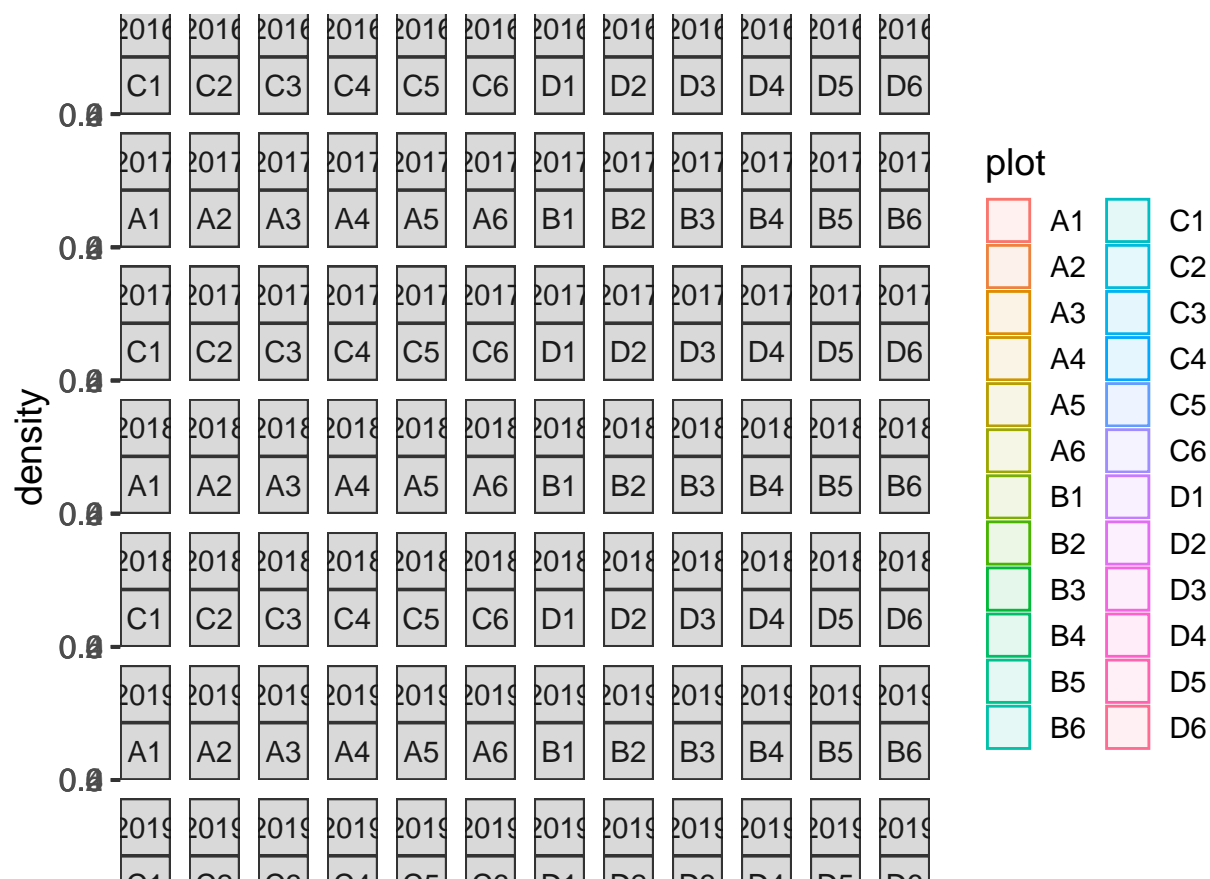
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

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

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

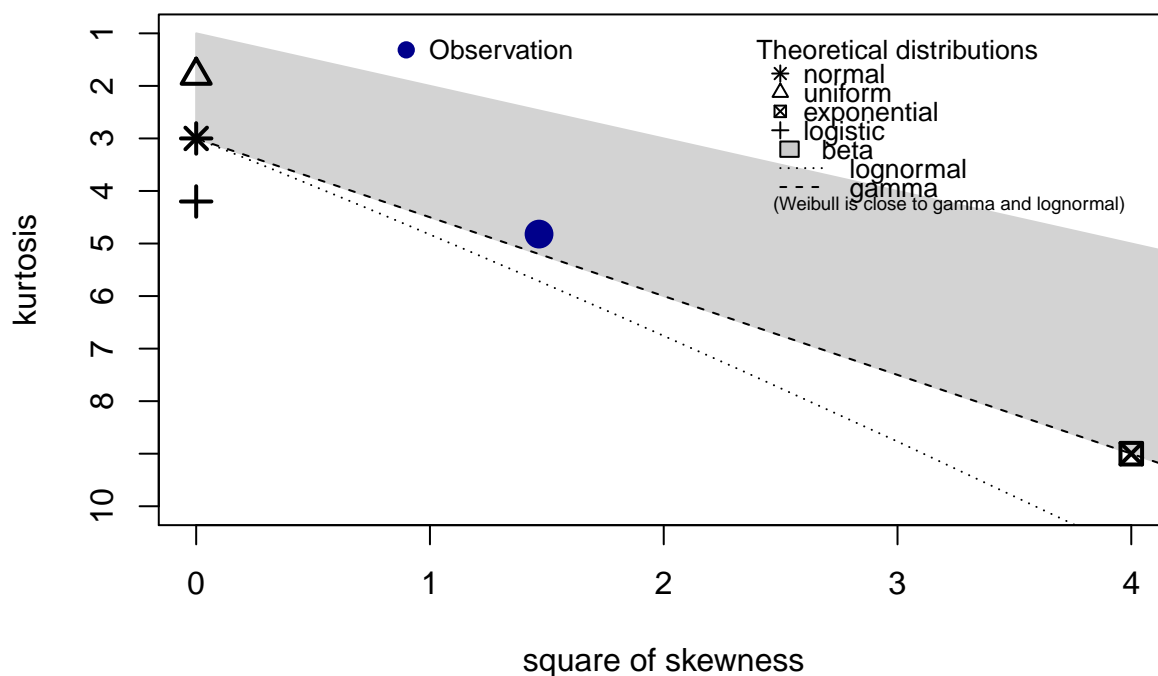
```

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



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

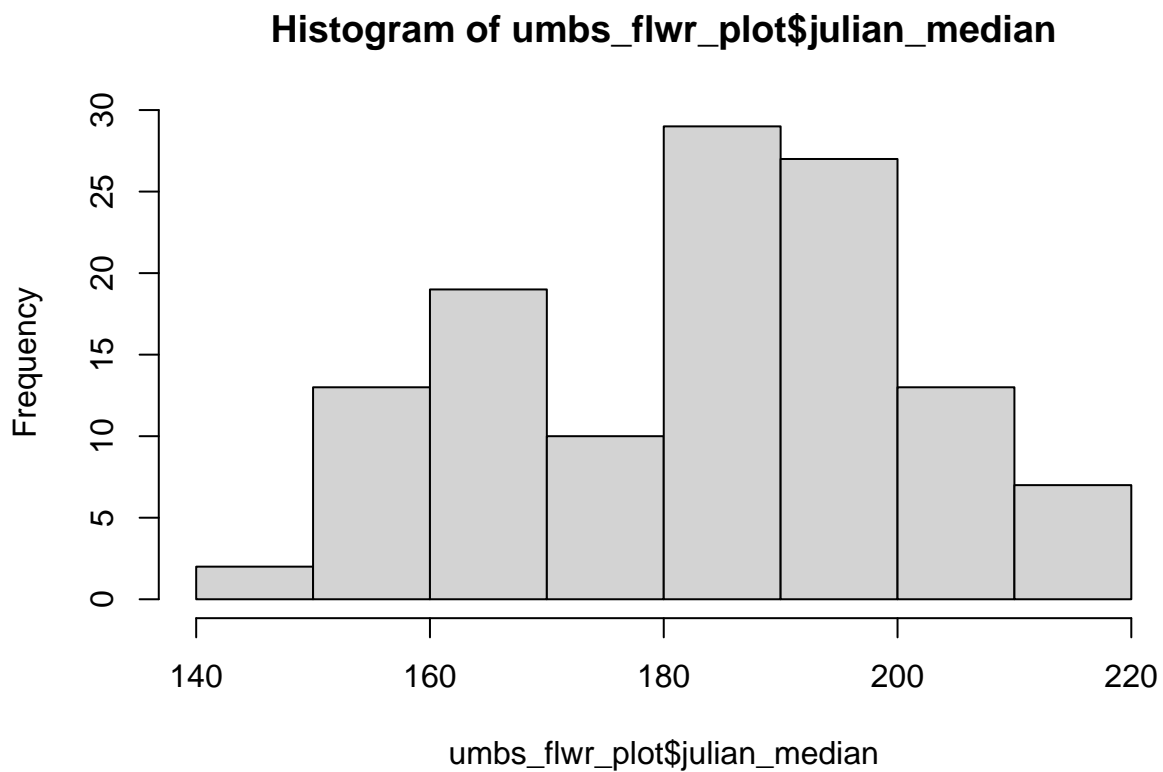
Cullen and Frey graph



```
## summary statistics
## -----
## min: 94    max: 218
## median: 135
## mean: 140.7452
## estimated sd: 20.92644
## estimated skewness: 1.211022
## estimated kurtosis: 4.820602
```

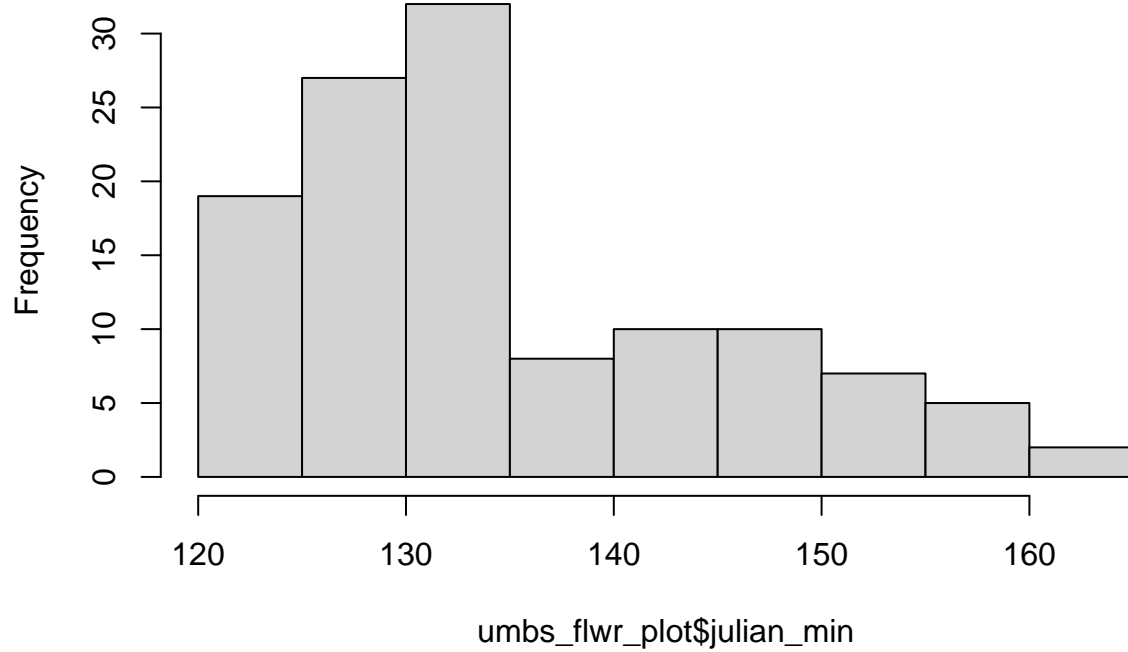
UMBS PLOT LEVEL

```
### UMBS ###
umbs_flwr_plot <- subset(flwr_plot, site == "umbs") # pull out umbs only data at plot level
hist(umbs_flwr_plot$julian_median)
```

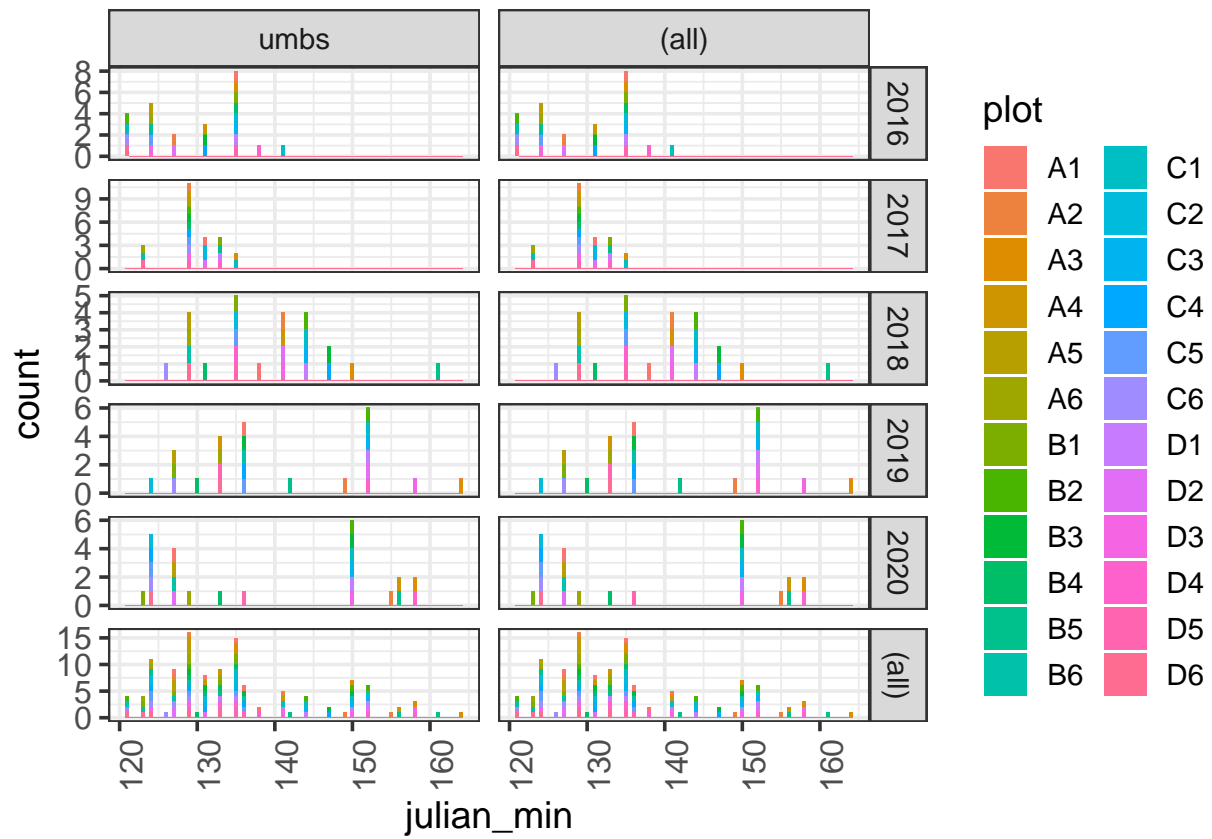


```
hist(umbs_flwr_plot$julian_min)
```

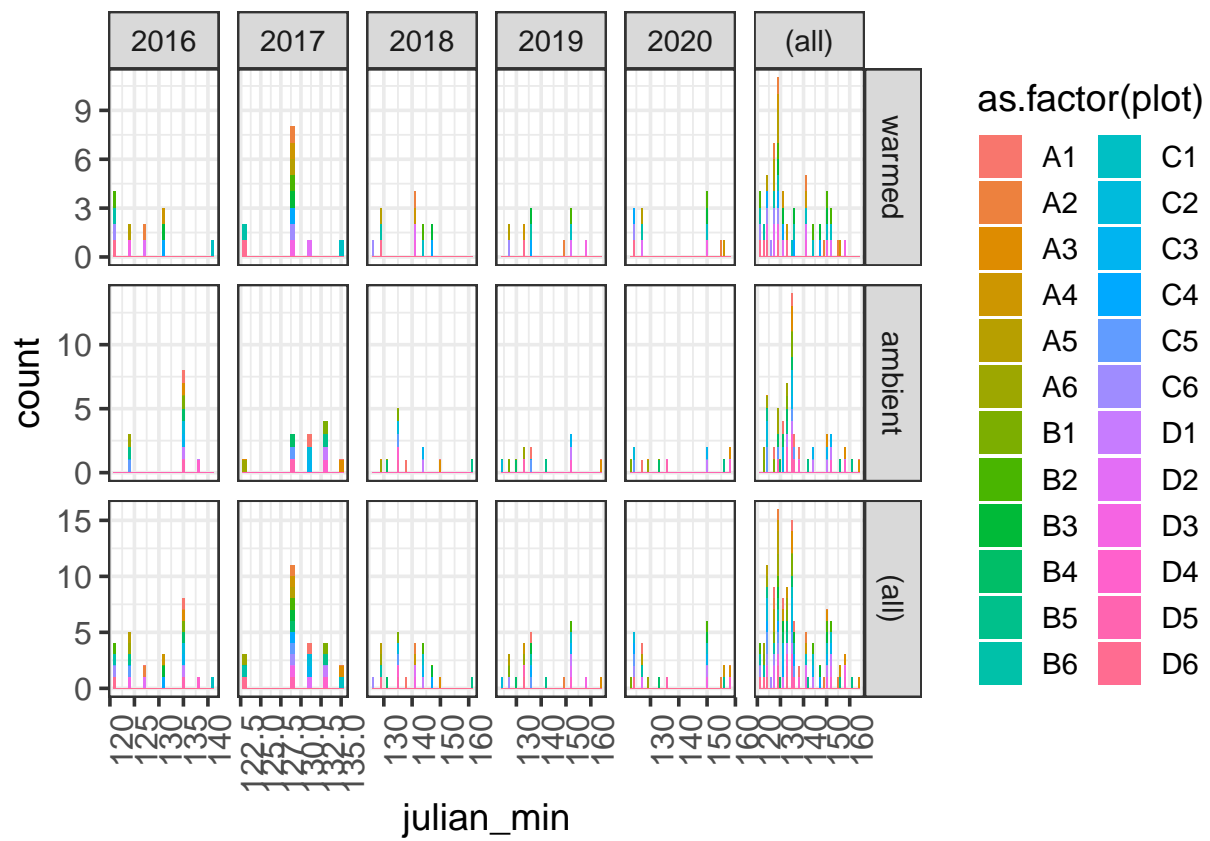
Histogram of umbs_flwr_plot\$julian_min



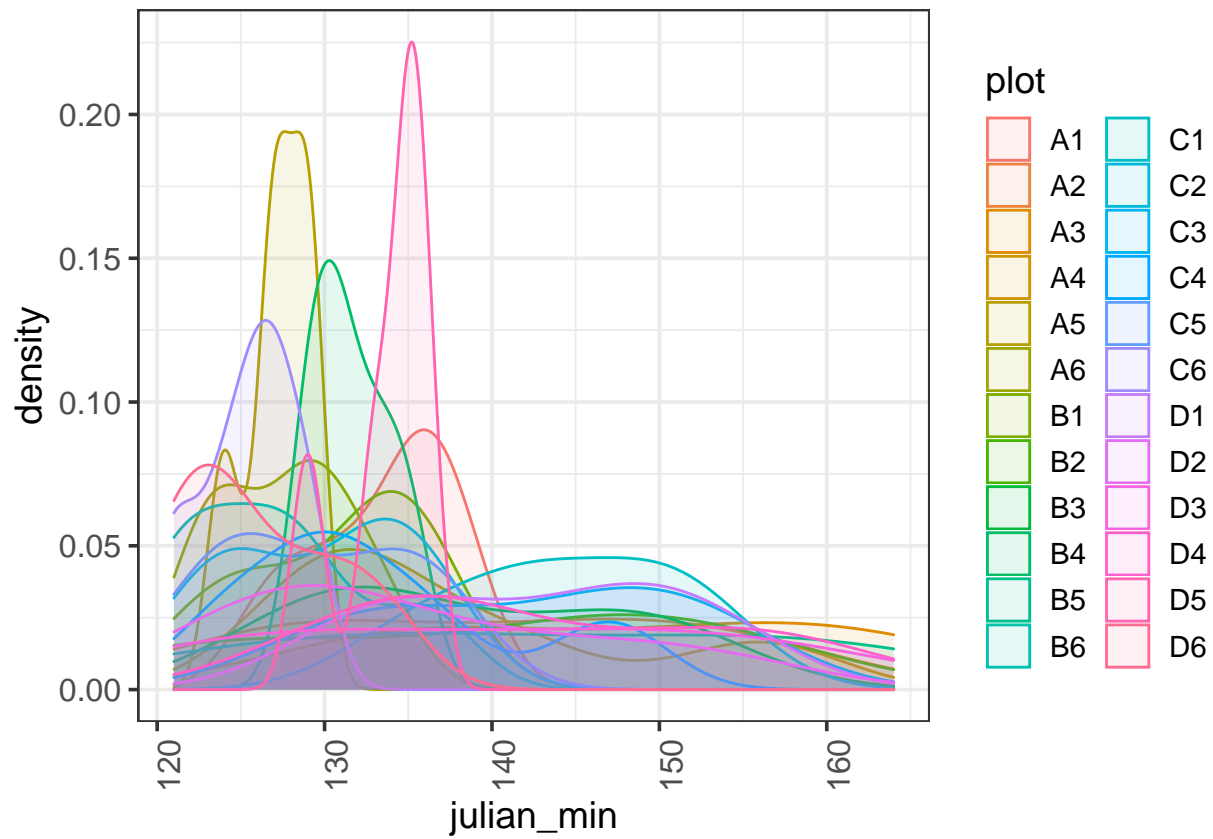
```
# Visualizing avg minimum Julian date for both sites at the PLOT LEVEL
ggplot(umbs_flwr_plot, aes(julian_min, fill = plot)) + geom_histogram(binwidth = 0.5) +
  facet_grid(year ~ site, margins = TRUE, scales = "free")
```



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



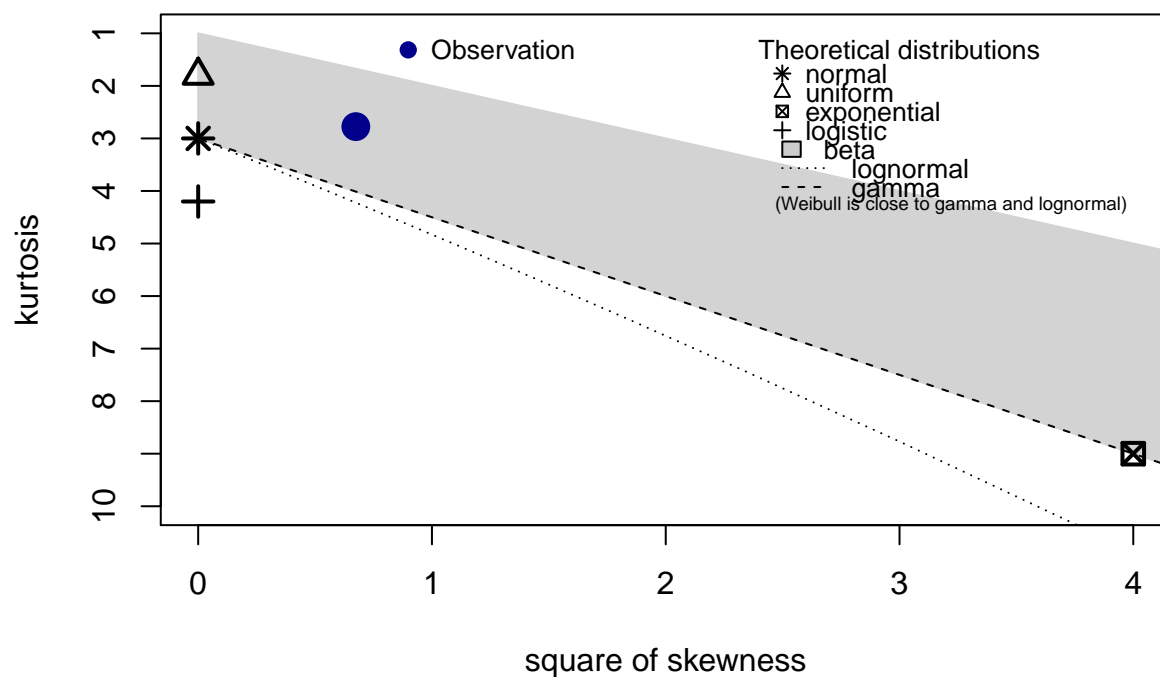
```
ggplot(umbs_flwr_plot, aes(julian_min, fill = plot, color = plot)) + geom_density(alpha = 0.1)
```



```
# ggplot(umbs_flwr_plot, aes(julian_min, fill = plot, color=plot)) +
# geom_density(alpha = 0.1) + facet_wrap(~year) ggplot(umbs_flwr_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_flwr_plot$julian_min, discrete = FALSE)
```

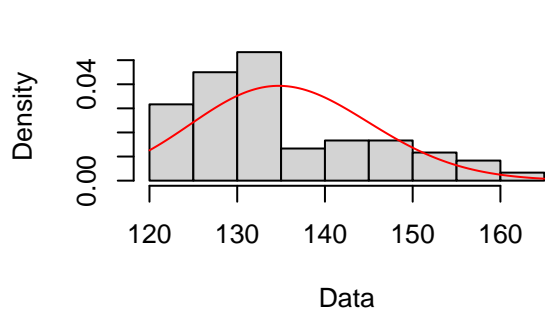

Cullen and Frey graph



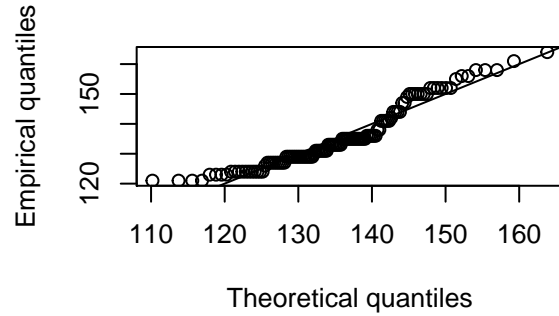
```
## summary statistics
## -----
## min: 121   max: 164
## median: 133
## mean: 135.4917
## estimated sd: 10.38745
## estimated skewness: 0.8212294
## estimated kurtosis: 2.775374
```

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

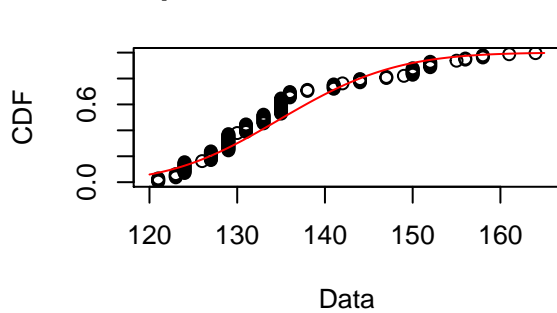
Empirical and theoretical dens.



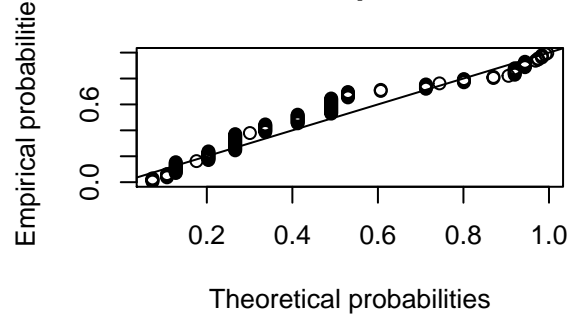
Q-Q plot



Empirical and theoretical CDFs

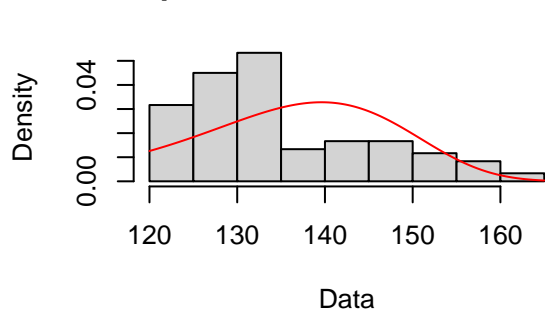


P-P plot

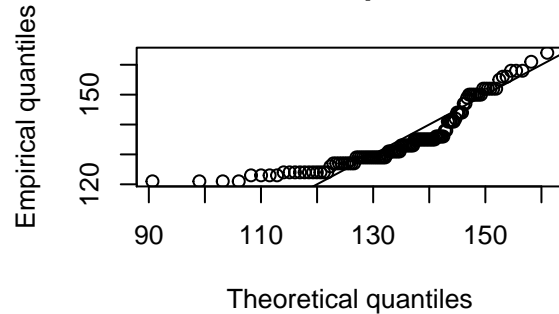


```
# Weibull distribution
fit.weibull <- fitdist(umbs_flwr_plot$julian_min, "weibull")
plot(fit.weibull)
```

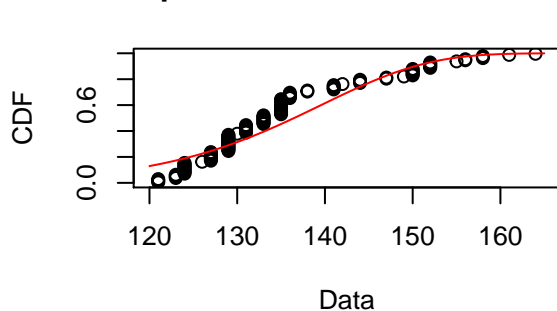
Empirical and theoretical dens.



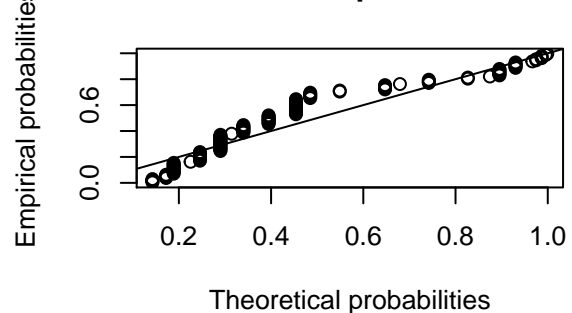
Q-Q plot



Empirical and theoretical CDFs

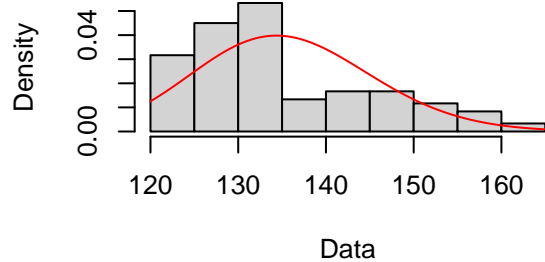


P-P plot

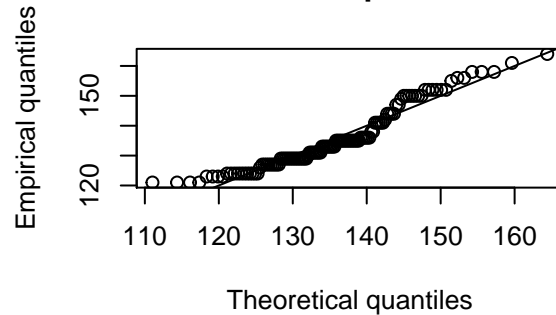


```
# Lognormal distribution
fit.ln <- fitdist(umbs_flwr_plot$julian_min, "lnorm")
plot(fit.ln)
```

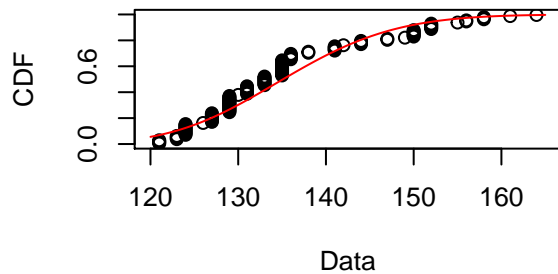
Empirical and theoretical dens.



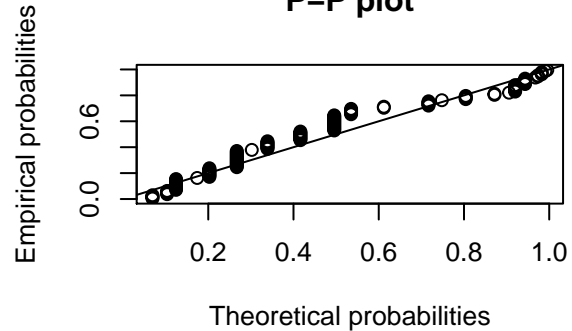
Q-Q plot



Empirical and theoretical CDFs

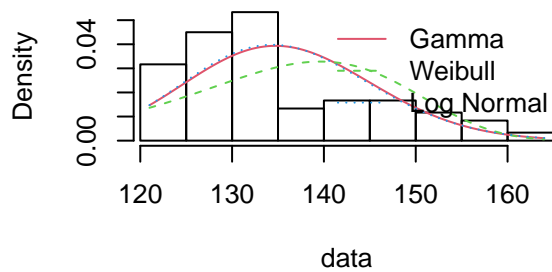


P-P plot

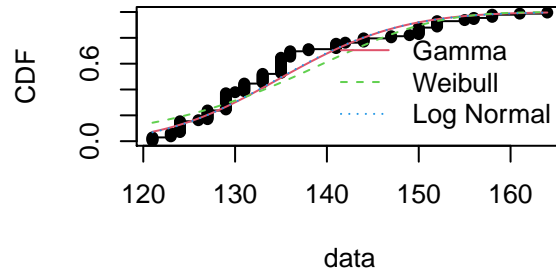


```
par(mfrow = c(2, 2))
plot.legend <- c("Gamma", "Weibull", "Log Normal")
denscomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
cdfcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
qqcomp(list(fit.gamma, fit.weibull, fit.ln), legendtext = plot.legend)
ppcomp(list(fit.gamma,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

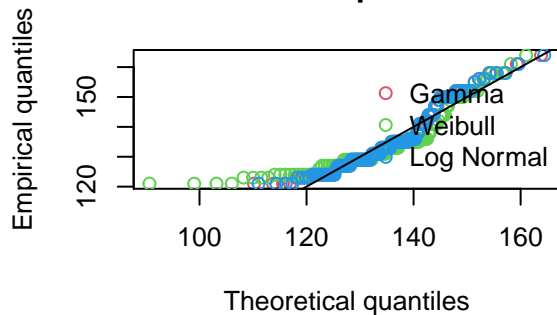
Histogram and theoretical densities



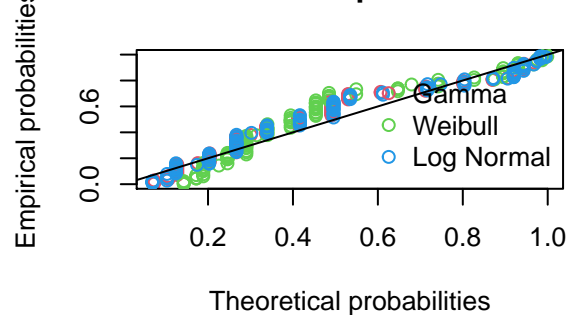
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```
# 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 0.1700250 0.2150911 0.1648747
## Cramer-von Mises statistic 0.5808497 1.0209433 0.5424176
## Anderson-Darling statistic 3.3369537 5.6394902 3.1360083
##
## Goodness-of-fit criteria
##
##      Gamma    Weibull Log Normal
## Akaike's Information Criterion 900.5729 933.5664 898.4475
## Bayesian Information Criterion 906.1479 939.1413 904.0225
```

```
# Lognormal is slightly better than gamma
```

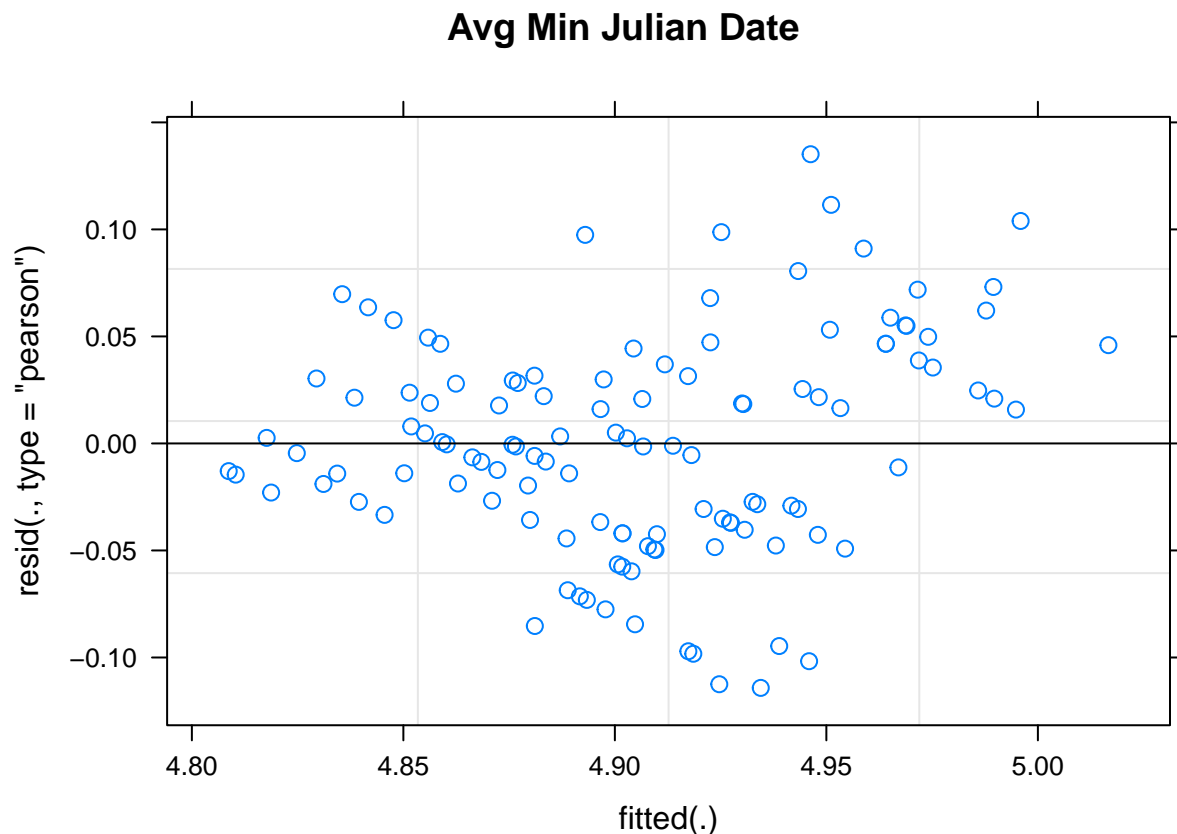
Centering and transforming data

```
umbs_flwr_plot$log_julian_median <- log(umbs_flwr_plot$julian_median)
umbs_flwr_plot$log_julianMedian_centered = umbs_flwr_plot$log_julian_median - mean(umbs_flwr_plot$log_j
hist(umbs_flwr_plot$log_julianMedian_centered)
```

Determining appropriate distribution in mixed effects model

```
# See Ben Bolker's site for details on fitting glms:
# https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html Fixed effects: warming,
# year, insecticide Random effect = plot (bc observations are nested within plot)
# Fit with ML bc we are interested in estimating fixed effects more than random
# effects See
# https://uoftcoders.github.io/rcourse/lec08-linear-mixed-effects-models.html for
# more details
m1 <- lmer(log(julian_min) ~ state + year + insecticide + (1 | plot), data = umbs_flwr_plot,
  REML = FALSE)

# Check Assumptions: (1) Linearity: if covariates are not categorical (year
# isn't) (2) Homogeneity: Need to Check by plotting residuals vs predicted
# values.
par(mfrow = c(1, 2))
plot(m1, main = "Avg Min Julian Date")
```



```
# Homogeneity of variance is ok here (increasing variance in resids is not
# increasing with fitted values) Check for homogeneity of variances (true if
# p>0.05). If the result is not significant, the assumption of equal variances
# (homoscedasticity) is met (no significant difference between the group
# variances).
leveneTest(residuals(m1) ~ umbs_flwr_plot$state)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
```

```
##          Df F value Pr(>F)
## group    1   0.001 0.9744
##          118
```

```
# Assumption met
leveneTest(residuals(m1) ~ umbs_flwr_plot$plot)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group    23  0.7084 0.8265
##          96
```

```
# Assumption met
leveneTest(residuals(m1) ~ umbs_flwr_plot$insecticide)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

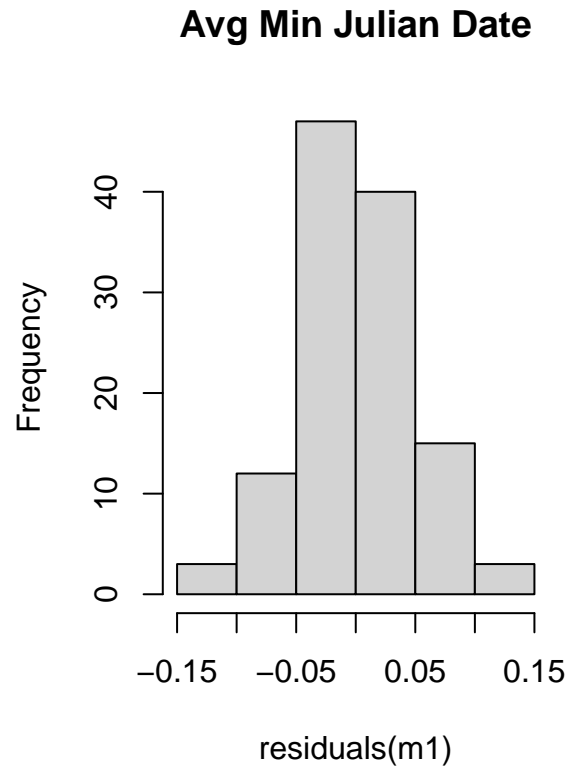
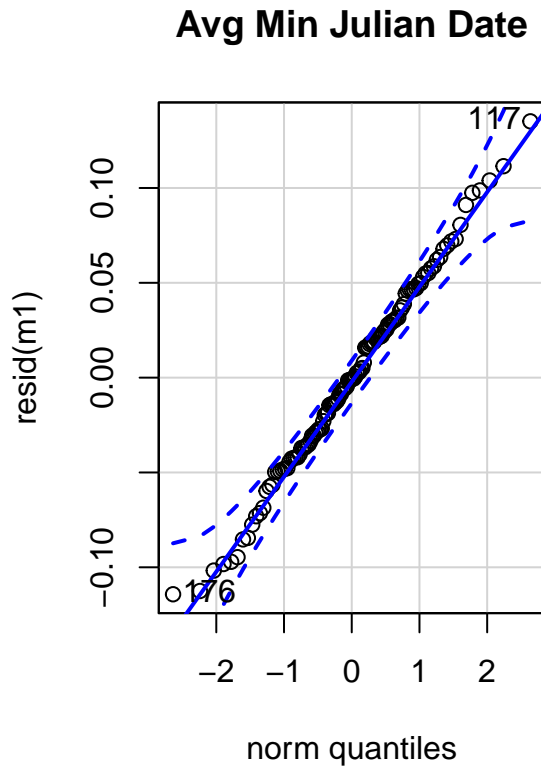
```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group     1  1.5017 0.2228
##          118
```

```
# Assumption met
```

```
# (3) Normality of error term: need to check by histogram, QQplot of residuals,
# could do Kolmogorov-Smirnov test. Check for normal residuals
qqPlot(resid(m1), main = "Avg Min Julian Date")
```

```
## 117 176
##  53  80
```

```
hist(residuals(m1), main = "Avg Min Julian Date")
```



```
shapiro.test(resid(m1)) # p>0.05 so we can assume normality
```

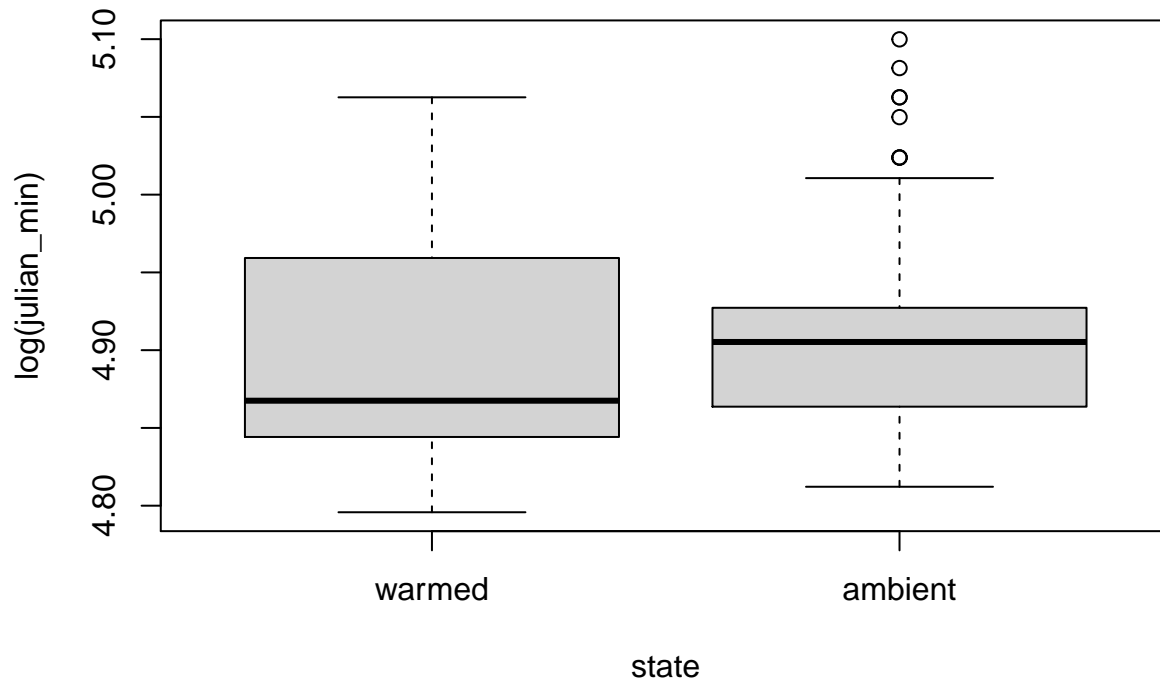
```
##
## Shapiro-Wilk normality test
##
## data: resid(m1)
## W = 0.99448, p-value = 0.922
```

```
# Outlier test - no outliers
outlierTest(m1)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferroni p
## 117 2.691335      0.0081977      0.98372
```

```
# (4) Normality of random effect: Get the estimate of random effect (e.g., random
# intercepts), and check them as you would check the residual. [***need to do for
# final model]
```

```
# Plot key relationships Convert all columns to factor first
umbs_flwr_plot <- as.data.frame(unclass(umbs_flwr_plot), stringsAsFactors = TRUE)
plot(log(julian_min) ~ state, data = umbs_flwr_plot)
```



Fit models

```
summary(m1)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log(julian_min) ~ state + year + insecticide + (1 | plot)
## Data: umbs_flwr_plot
##
##      AIC      BIC   logLik deviance df.resid
##   -313.6   -296.9    162.8   -325.6     114
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.09279 -0.65972 -0.01544  0.57734  2.47616
##
## Random effects:
## Groups   Name                Variance Std.Dev.
## plot     (Intercept)  0.001646  0.04057
## Residual                    0.002978  0.05457
## Number of obs: 120, groups: plot, 24
##
## Fixed effects:
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)   -36.974594   7.108547 95.990277  -5.201 1.12e-06 ***
## stateambient    0.013007   0.019330 24.000000   0.673  0.507
## year            0.020752   0.003523 95.989212   5.891 5.66e-08 ***
## insecticidenoinsects -0.005981   0.019330 24.000000  -0.309  0.760
## ---
```



```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr) sttmbn year
## stateambint -0.001
## year        -1.000  0.000
## insctcdn_ns -0.001  0.000  0.000
```

Compare models

```
# AICtab(m0,m1, weights=TRUE)
```