warmXtrophic Project: Greenup Analyses

Kara Dobson, Phoebe Zarnetske

July 13, 2021

Script Details:

```
script_tbl <- data.frame(Item = c("OVERVIEW", "COLLABORATORS", "REQUIRES", "DATA INPUT",
    "DATA OUTPUT", "NOTES"), Details = c("This script explores and analyses the greenup data from the KB
    "Moriah Young, Mark Hammond, Pat Bills", "Prior to running this script, make sure plant_comp_clean_L
    "Data imported as csv files from shared Google drive 'SpaCE_Lab_warmXtrophic' plant comp folder",
    "... a brief description of the data output from through the script, including what format it's in"
    "Each row in 'greenup' is the date at which spp_half_cover_date was recorded, per species. The 'gree

kbl(script_tbl) %>% kable_paper(full_width = F) %>% column_spec(1, bold = T, border_right = T) %>%
    column_spec(2, width = "30em", background = "lightblue")

metadata_tbl <- data.frame(Variable = c("spp_half_cover_date", "plot_half_cover_date",
    "state"), Definition = c("date at which 50% of a species max cover was reached (per species, per plo
    "the date at which 50% of a plot's max cover was reached (per plot, per year)",
    "describes each treatment: warmed or ambient"))

kbl(metadata_tbl) %>% kable_paper(full_width = F) %>% column_spec(1, bold = T, border_right = T) %>%
    column_spec(2, width = "30em", background = "lightyellow")
```

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

# Load packages
library(tidyverse)
library(ggplot2)
library(lme4)
library(olsrr)
library(predictmeans)
library(car)
library(fitdistrplus)
library(ggpubr)
library(rstatix)
library(vegan)
library(interactions)
library(sjPlot)
library(effects)
library(glmmTMB)
# install.packages('TMB',type='source')
```

```r
# Get data
Sys.getenv("L1DIR")
```

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

```r
L1_dir <- Sys.getenv("L1DIR")
L2_dir <- Sys.getenv("L2DIR")
greenup <- read.csv(file.path(L2_dir, "final_greenup_species_L2.csv"))
greenup <- greenup %>% select(-X)  # get rid of 'X' column that shows up
greenupp <- read.csv(file.path(L2_dir, "final_greenup_plot_L2.csv"))
greenupp <- greenupp %>% select(-X)  # get rid of 'X' column that shows up

# Set ggplot2 plots to bw: see here for more options:
# http://www.sthda.com/english/wiki/ggplot2-themes-and-background-colors-the-3-elements
theme_set(theme_bw(base_size = 14))
# check variable types
str(greenup)
```

```
## 'data.frame':    2026 obs. of  18 variables:
##  $ site              : chr  "kbs" "kbs" "kbs" "kbs" ...
##  $ plot              : chr  "A1" "A1" "A1" "A1" ...
##  $ year              : int  2016 2017 2018 2019 2020 2016 2017 2016 2017 2018 ...
##  $ species           : chr  "Acmi" "Acmi" "Acmi" "Acmi" ...
##  $ spp_half_cover_date: int  197 101 122 120 127 88 108 97 99 127 ...
##  $ min_green_date    : int  81 80 122 120 107 81 108 85 80 127 ...
##  $ treatment_key     : chr  "A0" "A0" "A0" "A0" ...
##  $ state             : chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ insecticide       : chr  "no_insects" "no_insects" "no_insects" "no_insects" ...
##  $ scientific_name   : chr  "Achillea millefolium" "Achillea millefolium" "Achillea millefolium" "Ac
##  $ common_name       : chr  "common yarrow" "common yarrow" "common yarrow" "common yarrow" ...
##  $ USDA_species      : chr  "ACMI2" "ACMI2" "ACMI2" "ACMI2" ...
##  $ LTER_species      : chr  "ACHMI" "ACHMI" "ACHMI" "ACHMI" ...
##  $ origin            : chr  "Native" "Native" "Native" "Native" ...
##  $ group             : chr  "Dicot" "Dicot" "Dicot" "Dicot" ...
##  $ family            : chr  "Fabaceae" "Fabaceae" "Fabaceae" "Fabaceae" ...
##  $ duration          : chr  "Biennial" "Biennial" "Biennial" "Biennial" ...
##  $ growth_habit      : chr  "Forb" "Forb" "Forb" "Forb" ...
```

```r
# 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.
greenup$state <- as.factor(greenup$state)
levels(greenup$state)
```

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

```r
greenup$state <- factor(greenup$state, levels(greenup$state)[c(2, 1)])
levels(greenup$state)
```

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

```r
greenupp$state <- as.factor(greenupp$state)
levels(greenupp$state)
```

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

```r
greenupp$state <- factor(greenupp$state, levels(greenupp$state)[c(2, 1)])
levels(greenupp$state)
```

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

```r
# adding sequential year variable starting at 1: this is because 2016... are
# large numbers compare with other values in the dataset. We can always label
# axes with these real years.
greenup$year_factor[greenup$year == 2016] <- 1
greenup$year_factor[greenup$year == 2017] <- 2
greenup$year_factor[greenup$year == 2018] <- 3
greenup$year_factor[greenup$year == 2019] <- 4
greenup$year_factor[greenup$year == 2020] <- 5

greenupp$year_factor[greenupp$year == 2016] <- 1
greenupp$year_factor[greenupp$year == 2017] <- 2
greenupp$year_factor[greenupp$year == 2018] <- 3
greenupp$year_factor[greenupp$year == 2019] <- 4
greenupp$year_factor[greenupp$year == 2020] <- 5

# create dataframes for kbs and umbs - remember that these contain species within
# plots
green_kbs <- subset(greenup, site == "kbs")
green_umbs <- subset(greenup, site == "umbs")

green_kbsp <- subset(greenupp, site == "kbs")
green_umbsp <- subset(greenupp, site == "umbs")
```

Data Exploration: are there differences between warmed vs. ambient plots when we account for species?

Starting with KBS

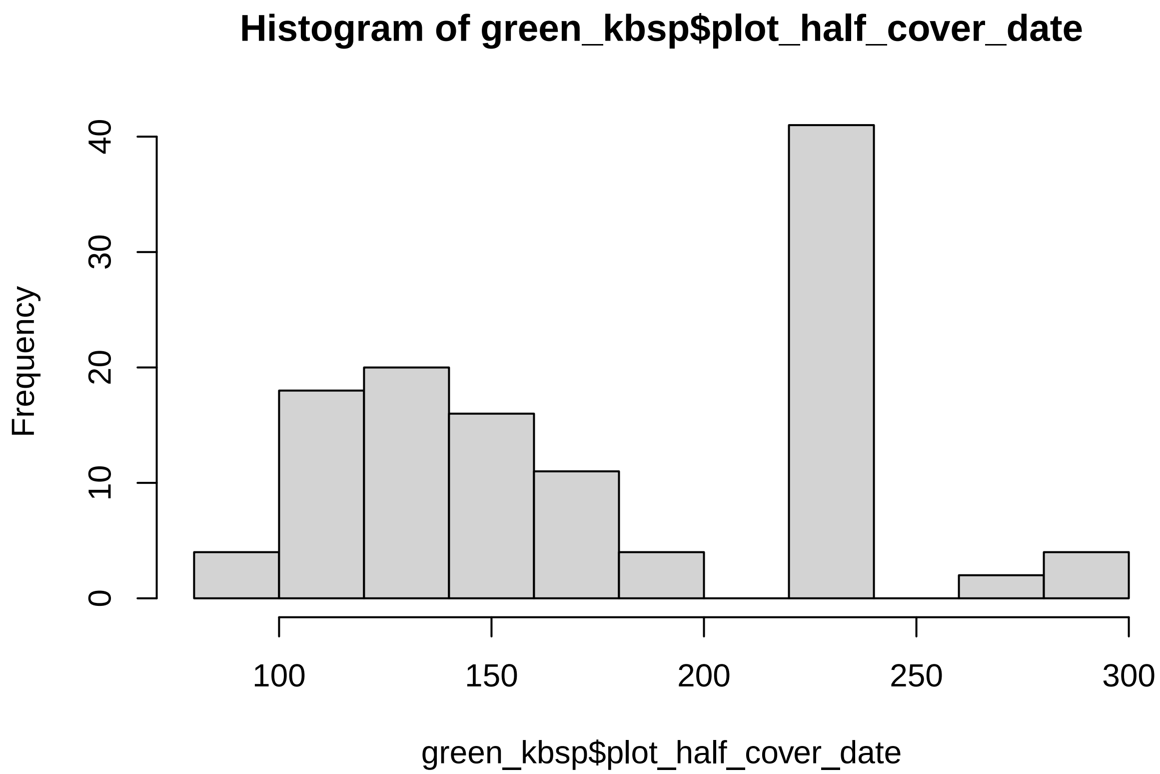
First, checking for normality in raw data. This is a little weird to do because each row in the dataframe is a species-plot combo, so you're missing the influence of species below. Instead, re-run this on plot-level half cover date (re-compute this in plant_comp_clean_L0.R to get that variable first). It's also not going to tell you about normality once you fit a model to these data - that's when you really need to investigate the residuals.

Note that since this change, some of the code below breaks because the variable name changed. Also, there seem to be some errors popping up with qqPlot, suggesting that something else changed when this new variable was created. Check

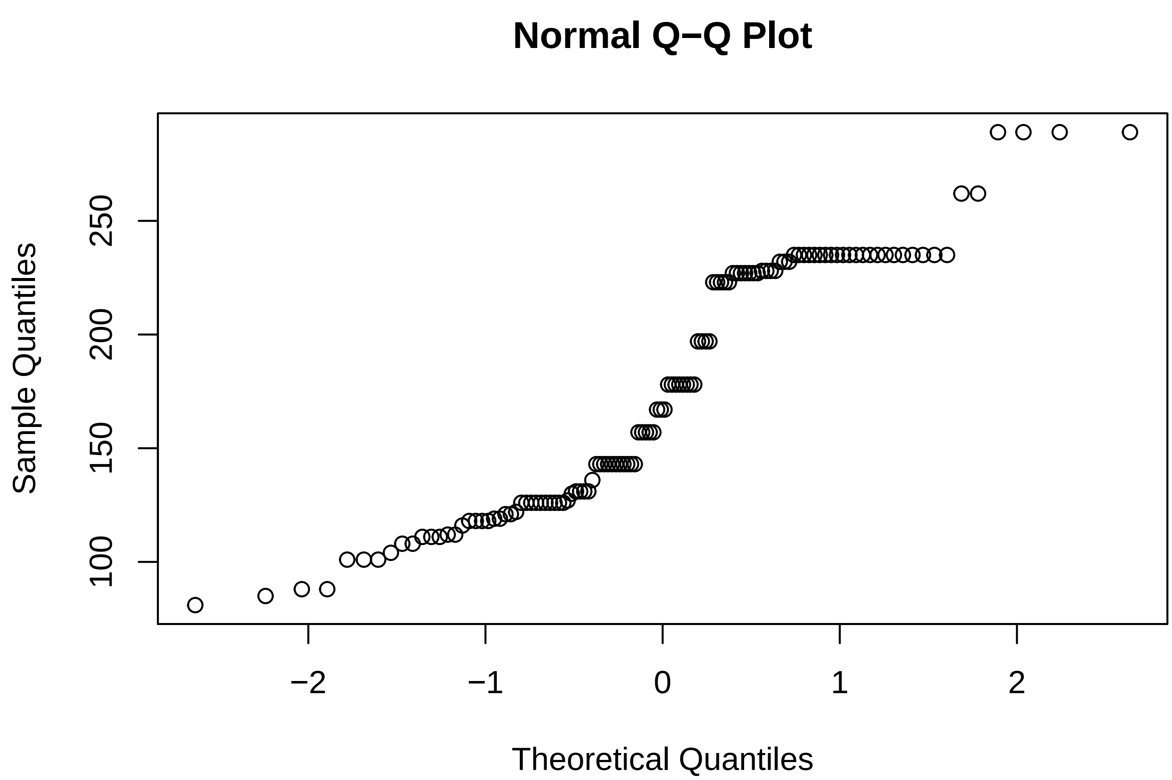
```
# remove species, origin, duration,growth habit,spp_half_cover_date - again, this
# should be done in other script preceding this one
green_kbsp <- subset(green_kbsp, select = c("plot_half_cover_date", "min_green_date",
    "plot", "year", "state", "insecticide", "site", "year_factor"))
head(green_kbsp)
```

```
##    plot_half_cover_date min_green_date plot year   state insecticide site
## 1                   232             81   A1 2016 ambient  no_insects  kbs
## 2                   101             61   A1 2017 ambient  no_insects  kbs
## 3                   235             59   A1 2018 ambient  no_insects  kbs
## 4                   157            106   A1 2019 ambient  no_insects  kbs
## 5                   223             91   A1 2020 ambient  no_insects  kbs
## 11                  126             81   A2 2016  warmed     insects  kbs
##    year_factor
## 1            1
## 2            2
## 3            3
## 4            4
## 5            5
## 11           1
```

```
hist(green_kbsp$plot_half_cover_date)
```



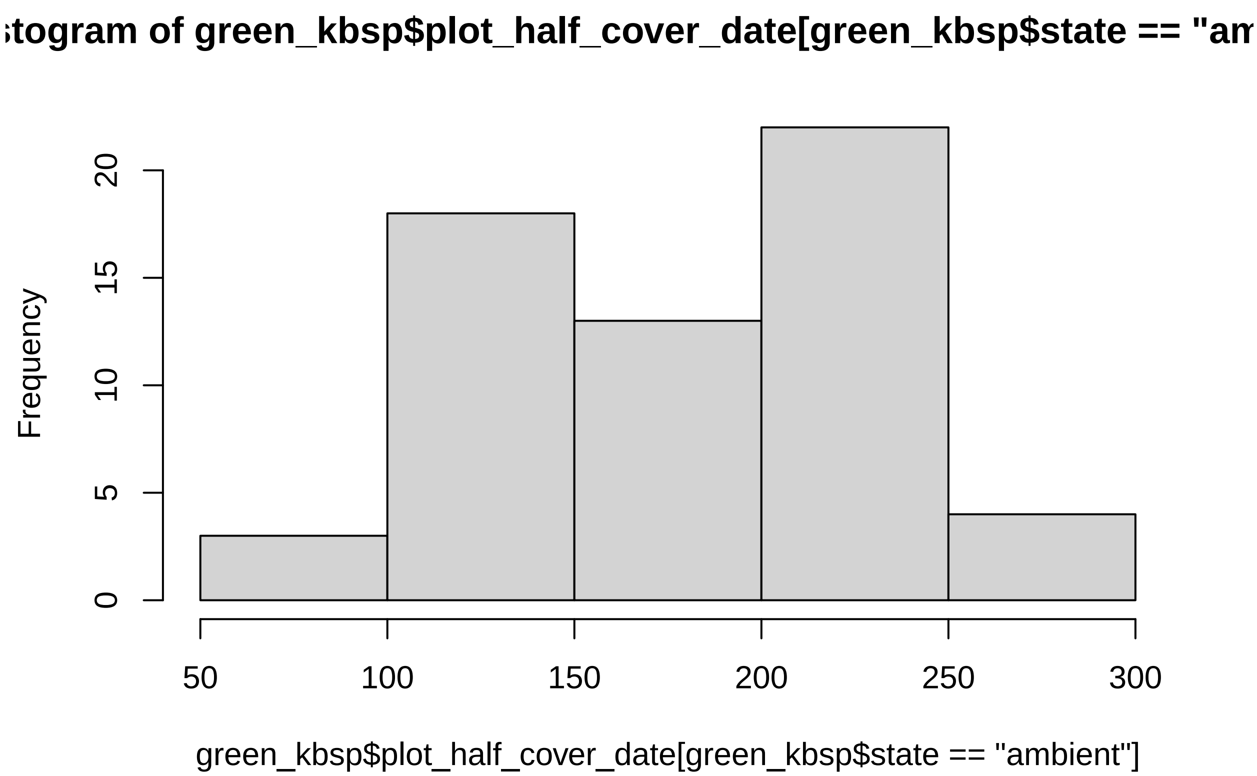

```r
qqnorm(green_kbsp$plot_half_cover_date)
```



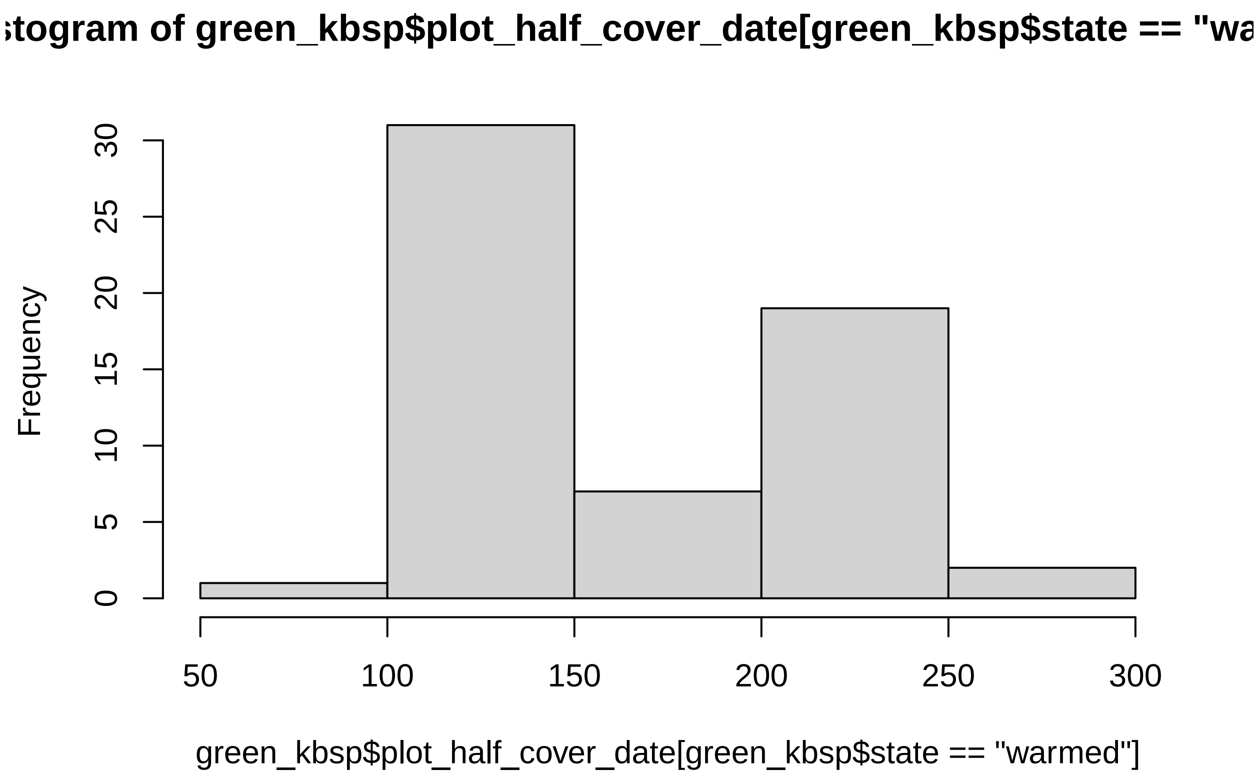
```r
shapiro.test(green_kbsp$plot_half_cover_date)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  green_kbsp$plot_half_cover_date
## W = 0.90721, p-value = 4.673e-07
```

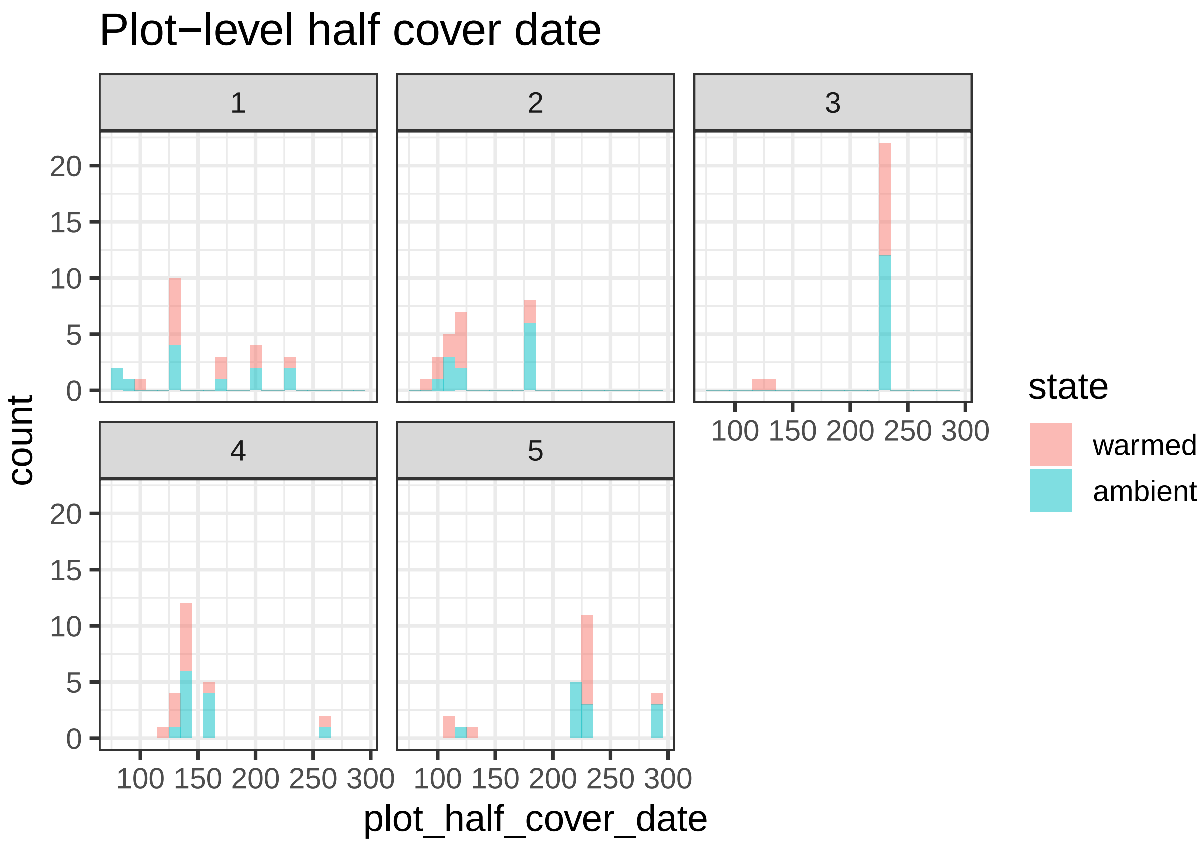
```r
# histograms for each treatment separately
hist(green_kbsp$plot_half_cover_date[green_kbsp$state == "ambient"])
```



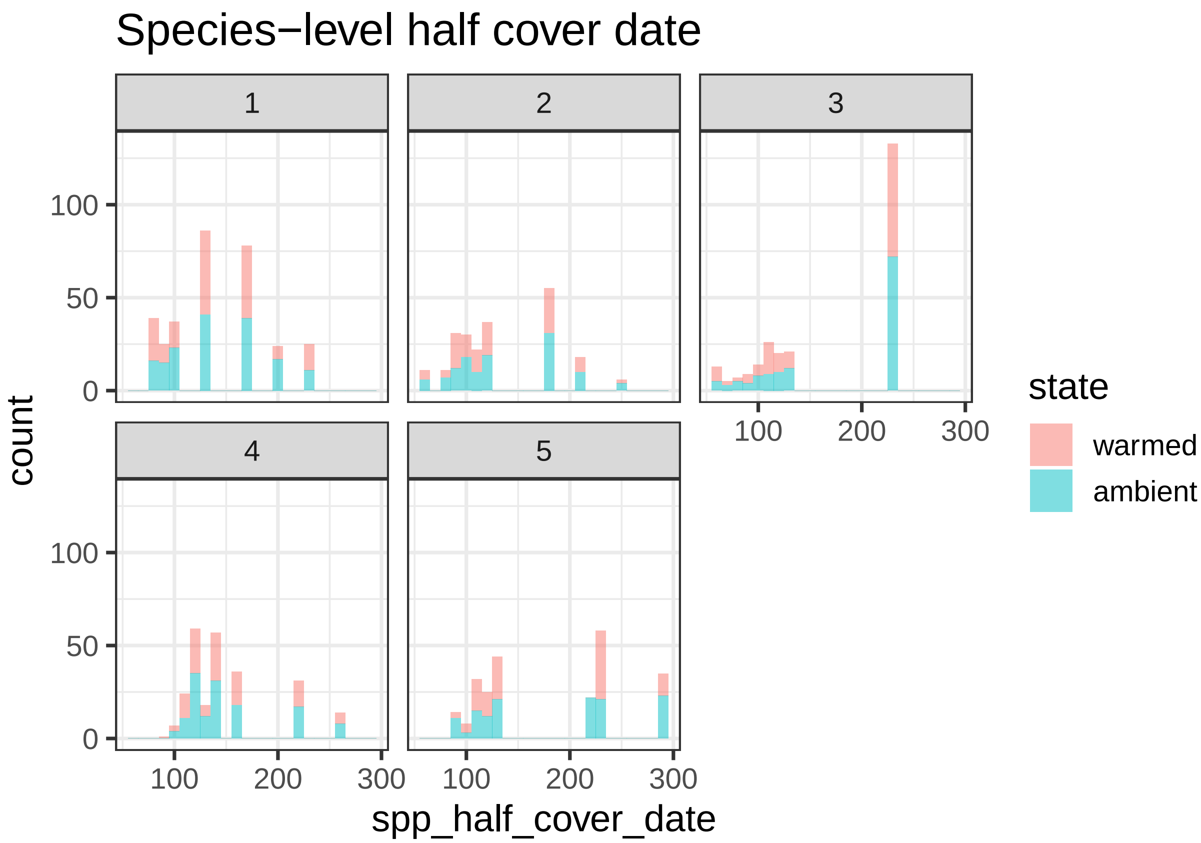

```r
hist(green_kbsp$plot_half_cover_date[green_kbsp$state == "warmed"])
```



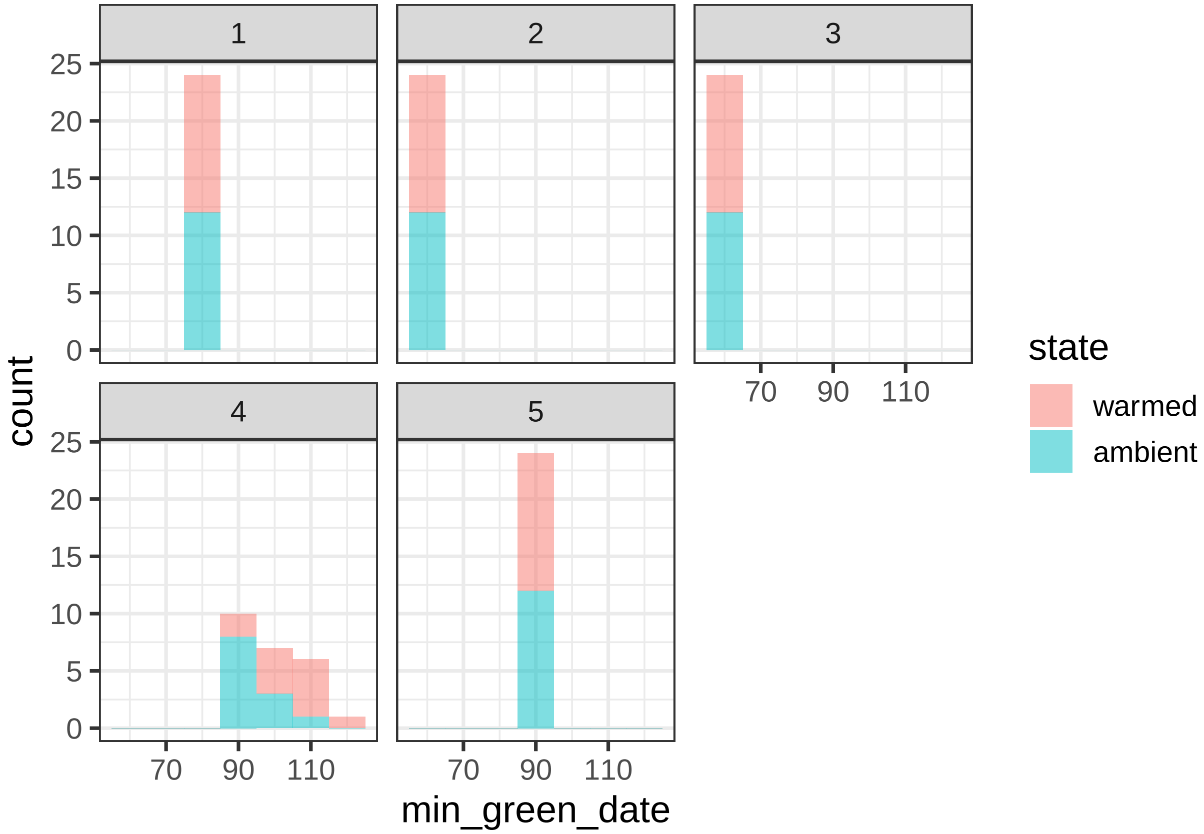

```
# histograms for each year - look at them together:
p1 <- ggplot(data = green_kbsp, aes(x = plot_half_cover_date, fill = state)) + geom_histogram(alpha = 0
    binwidth = 10)
p1 + facet_wrap(~year_factor) + labs(title = "Plot-level half cover date")
```



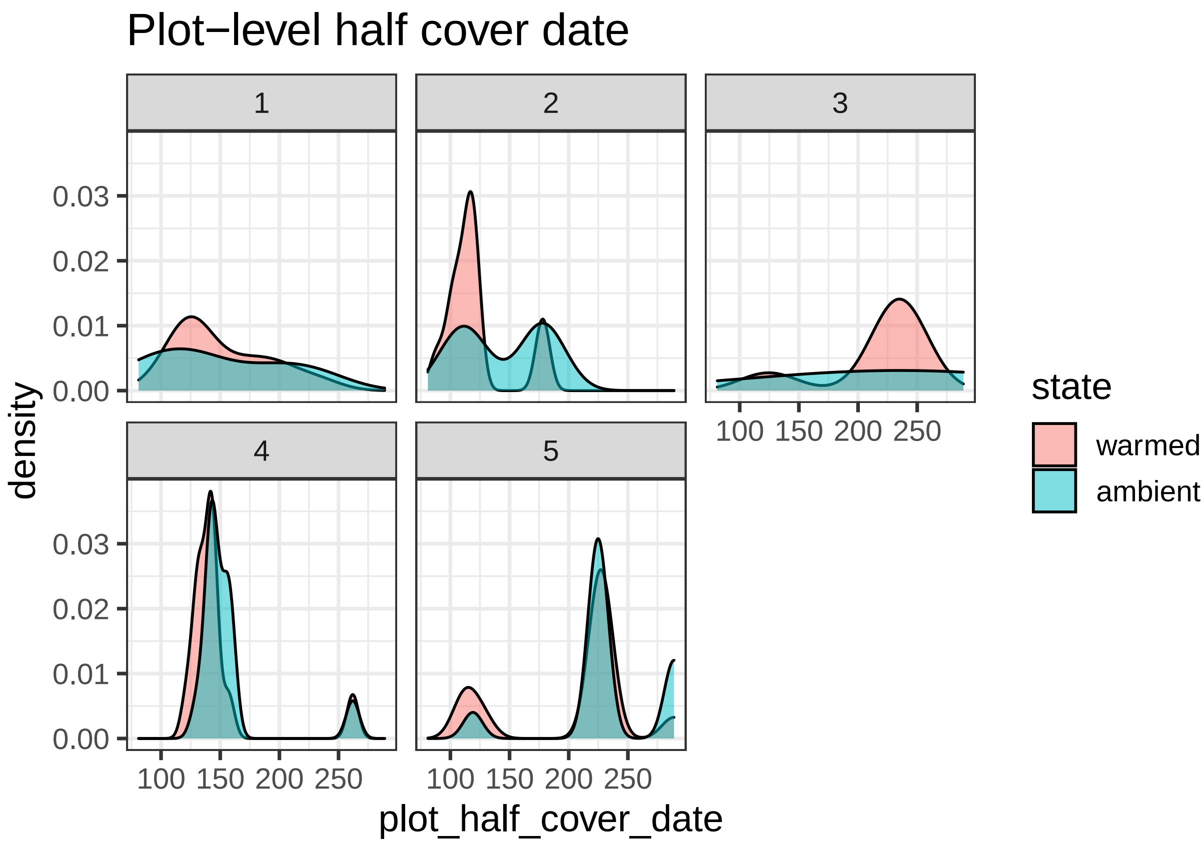

```
p1 <- ggplot(data = green_kbs, aes(x = spp_half_cover_date, fill = state)) + geom_histogram(alpha = 0.5
    binwidth = 10)
p1 + facet_wrap(~year_factor) + labs(title = "Species-level half cover date")
```



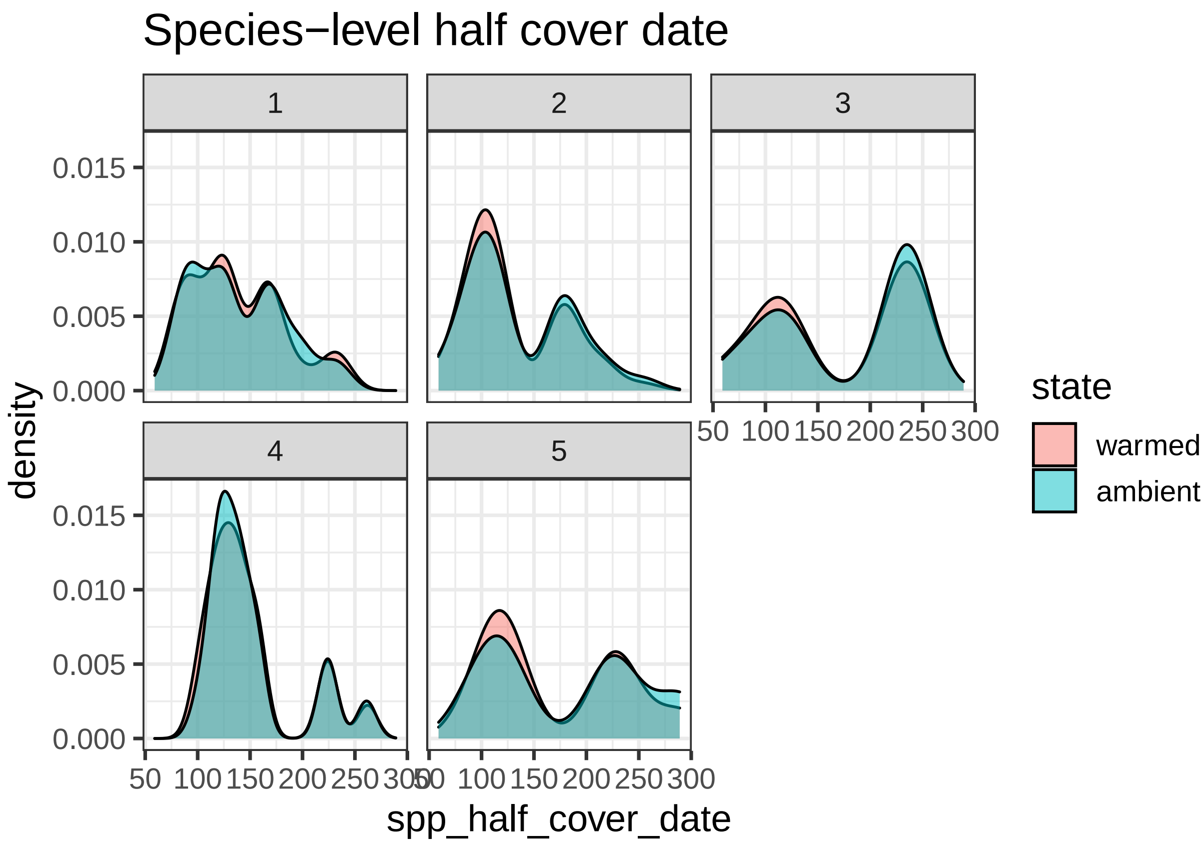

```r
# this will just show sampling date artifact
p2 <- ggplot(data = green_kbsp, aes(x = min_green_date, fill = state)) + geom_histogram(alpha = 0.5,
    binwidth = 10)
p2 + facet_wrap(~year_factor)
```



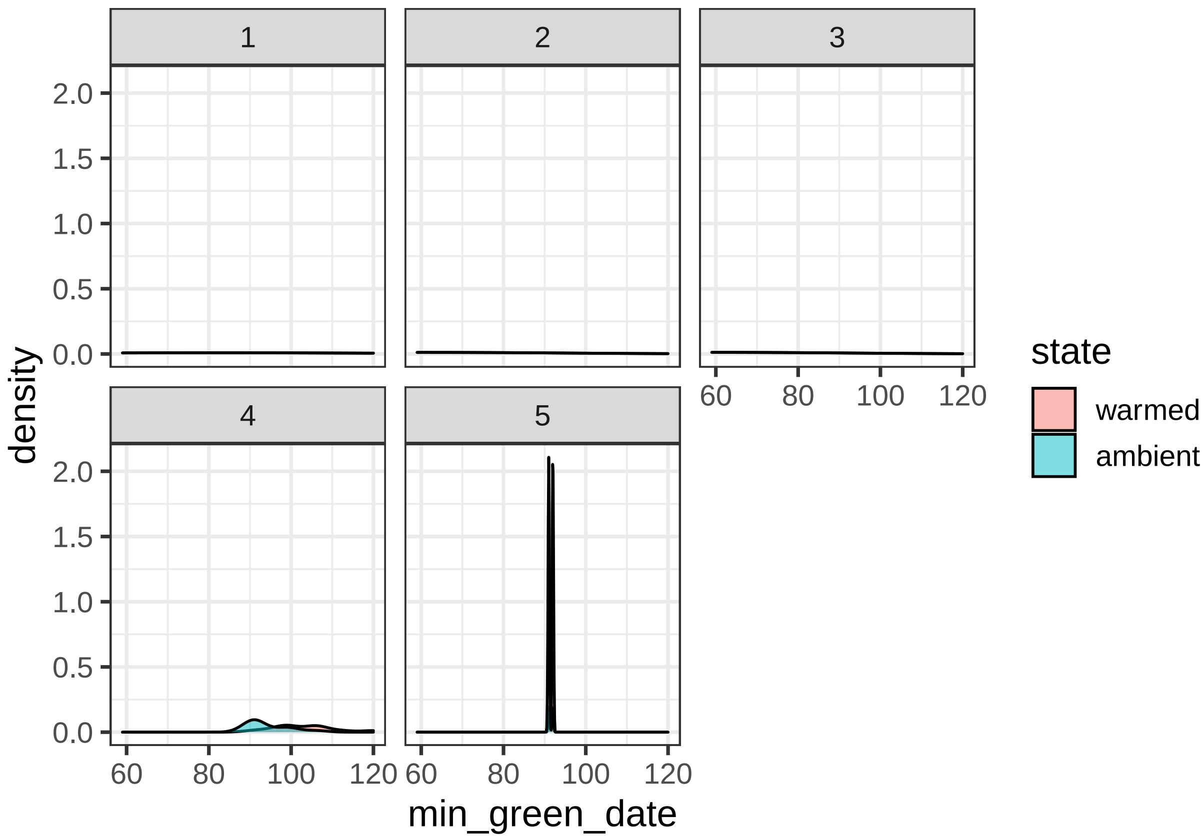

```
# Density plot
p3 <- ggplot(data = green_kbsp, aes(x = plot_half_cover_date, fill = state)) + geom_density(alpha = 0.5)
p3 + facet_wrap(~year_factor) + labs(title = "Plot-level half cover date")
```




```
p3 <- ggplot(data = green_kbs, aes(x = spp_half_cover_date, fill = state)) + geom_density(alpha = 0.5)
p3 + facet_wrap(~year_factor) + labs(title = "Species-level half cover date")
```




```r
# this will just show sampling date artifact
p4 <- ggplot(data = green_kbsp, aes(x = min_green_date, fill = state)) + geom_density(alpha = 0.5)
p4 + facet_wrap(~year_factor)
```




```
# Or try with tidyverse format
green_kbsp.t <- as_tibble(green_kbsp)
# green_kbsp.t %>% gather(state, plot_half_cover_date, year_factor) %>%
# ggplot(aes(plot_half_cover_date, fill = state)) + geom_histogram() +
# facet_wrap(~year_factor)

# looks like the 225 spike is from 2018 and 2020 - what's going on here is that
# you are treating all species-plot records as independent observations, so the
# influence of species differences is likely coming through here.
kbs_2018 <- subset(green_kbs, year == 4)  # many records on 235
kbs_2020 <- subset(green_kbs, year == 6)  # records from 227 & 228
```

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

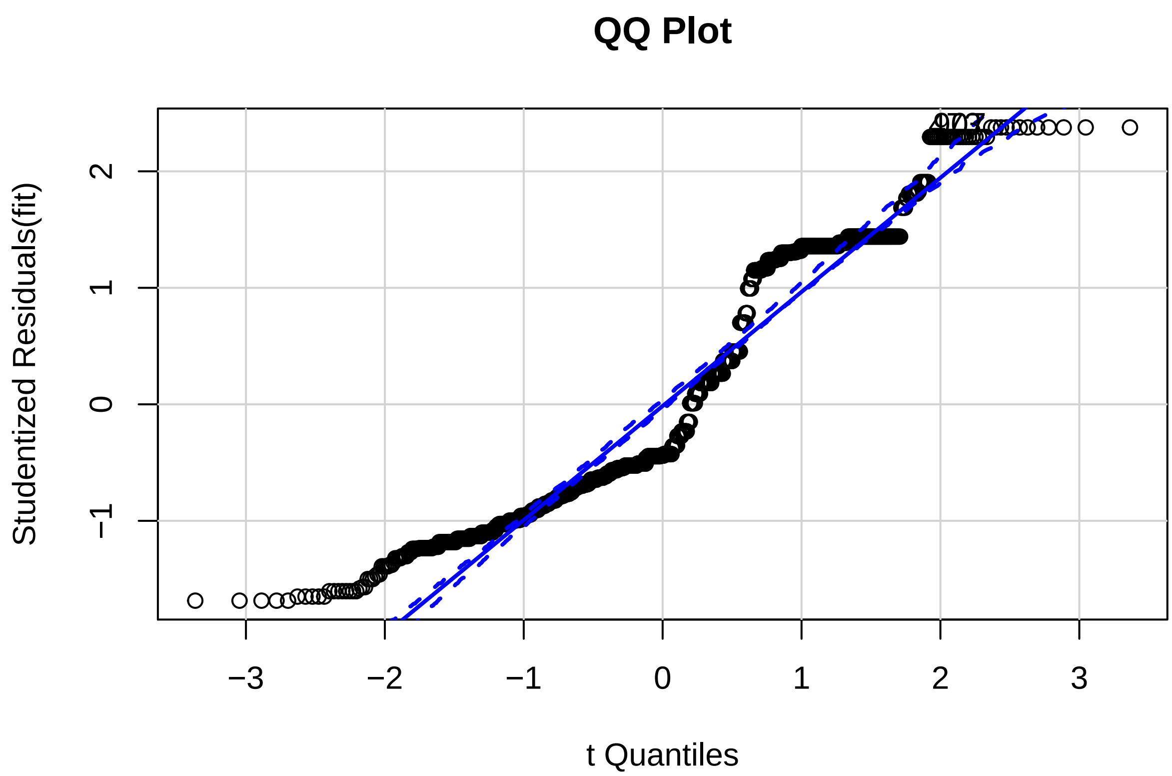
```
# checking fit for date as a function of state and species - bringing in species
# here makes it obvious that that is explaining some of the variation compared
# with the state-only model you had previously.

# State-only model (not really correct bc species differ)
fit <- lm(spp_half_cover_date ~ state, data = green_kbs)
outlierTest(fit)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 473 2.376821           0.017611           NA
```

```
# record 551 is the largest outlier

# State-only model - not really correct bc species should be included
fit <- lm(spp_half_cover_date ~ state, data = green_kbs)
outlierTest(fit)
```

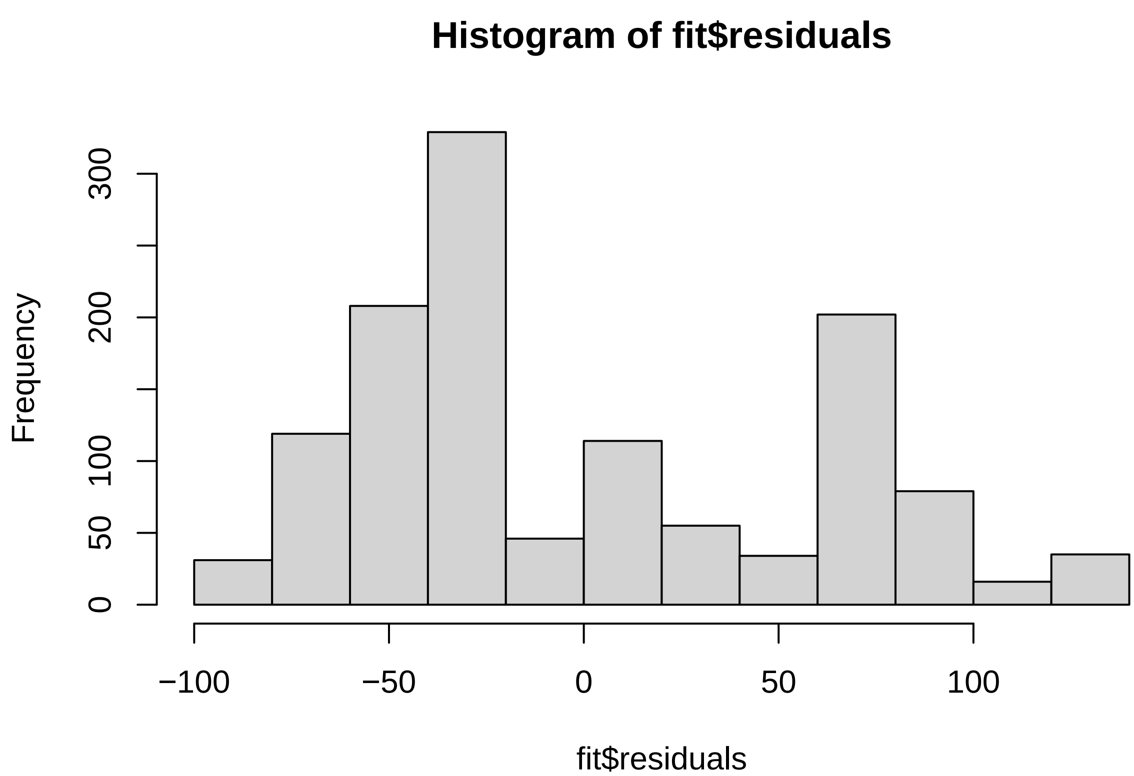
```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 473 2.376821           0.017611           NA
```

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

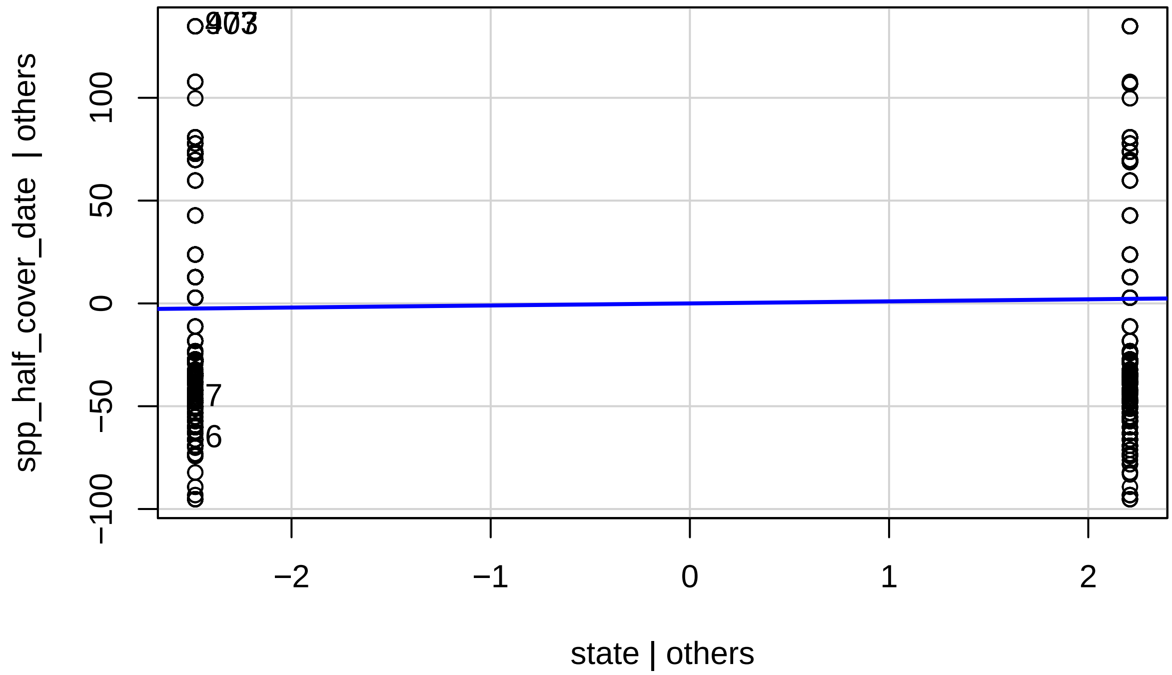



```
## 473 907
## 283 552
```

```
hist(fit$residuals)
```



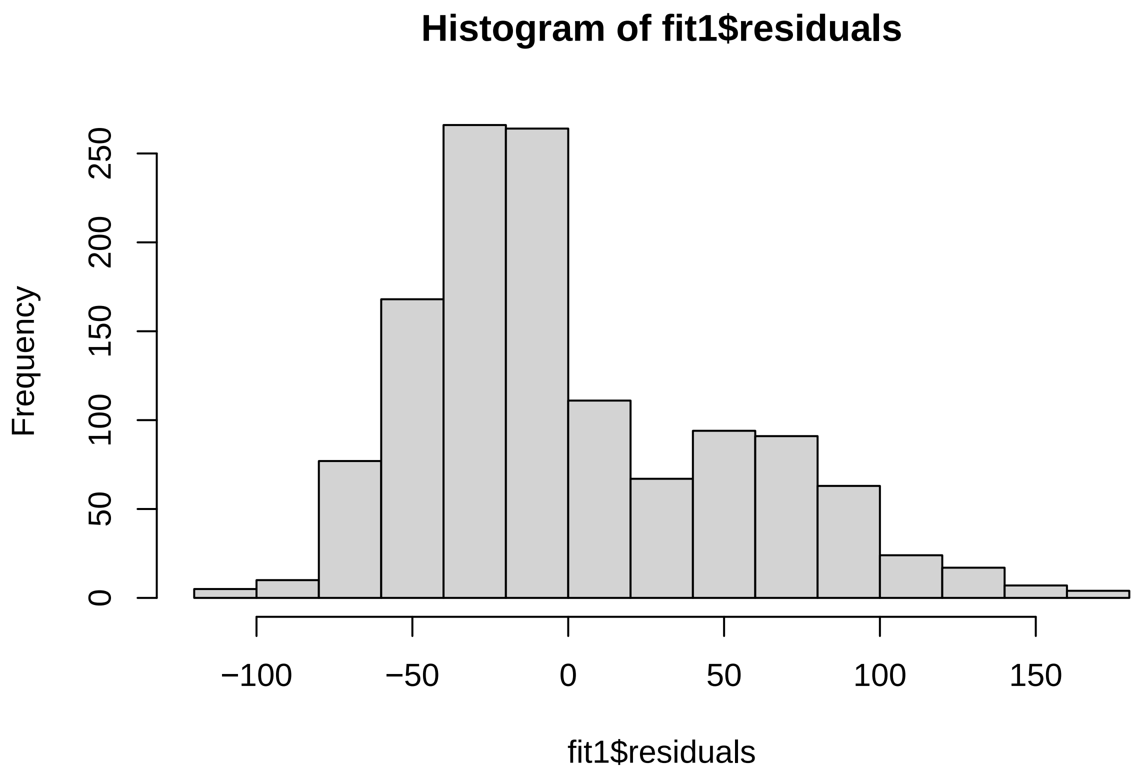
```r
leveragePlots(fit)
```




```r
# State and species model
fit1 <- lm(spp_half_cover_date ~ state + species, data = green_kbs)
outlierTest(fit1)  # no outliers
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferroni p
## 1910 3.455976         0.00056677      0.71866
```

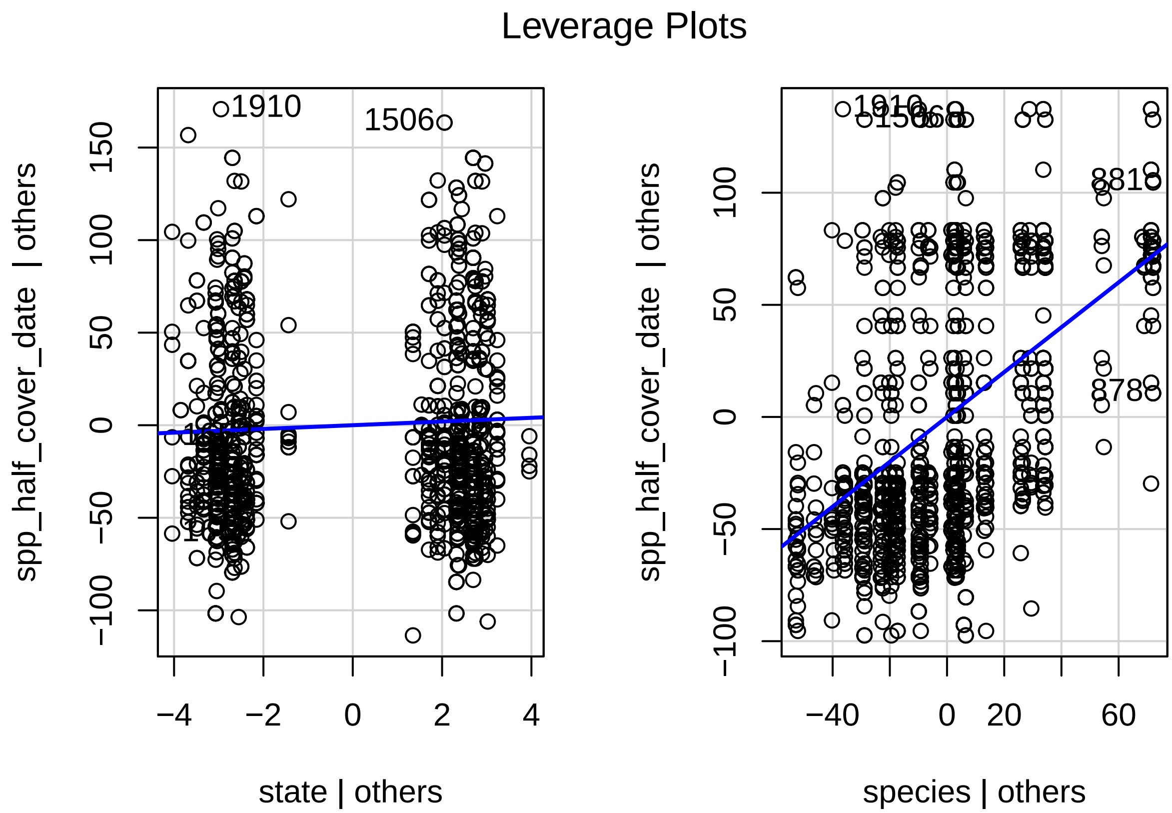
```r
hist(fit1$residuals)
```




```r
qqPlot(fit1, main = "QQ Plot")
```


```
## 1506 1910
##  943 1152
```

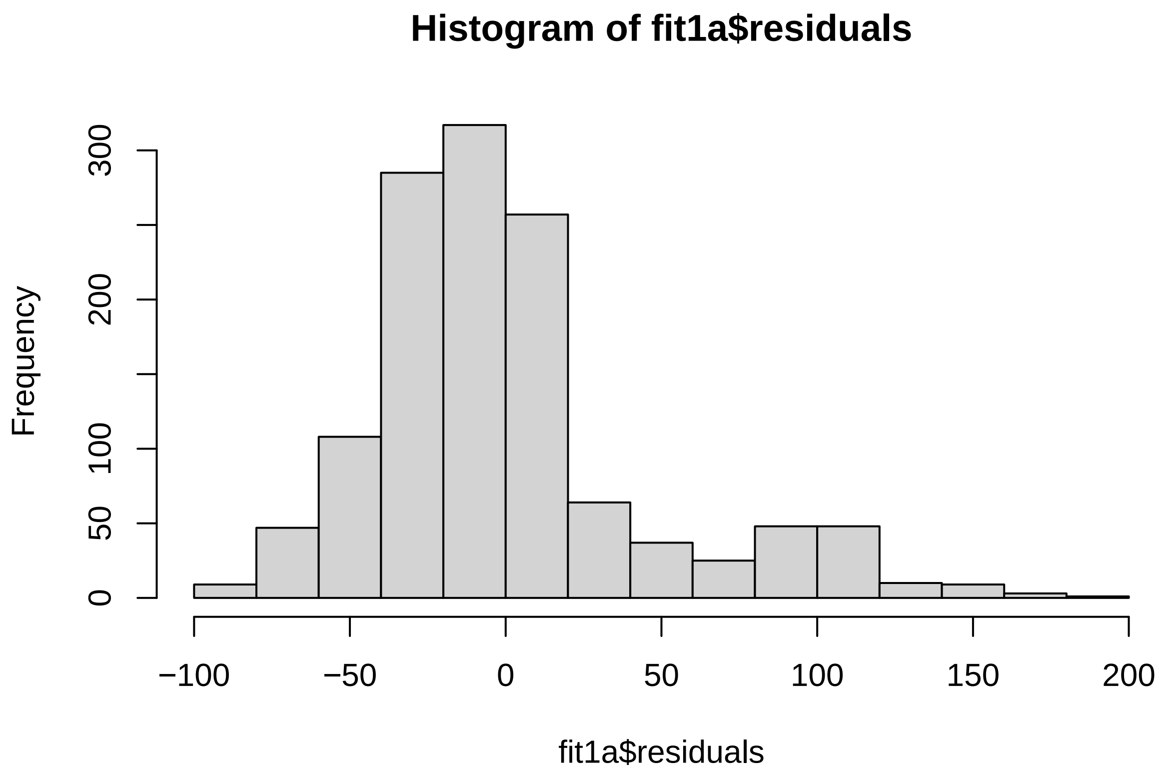
```r
leveragePlots(fit1)
```




```r
fit1a <- lm(min_green_date ~ state + species, data = green_kbs)
outlierTest(fit1a)  # no outliers
```

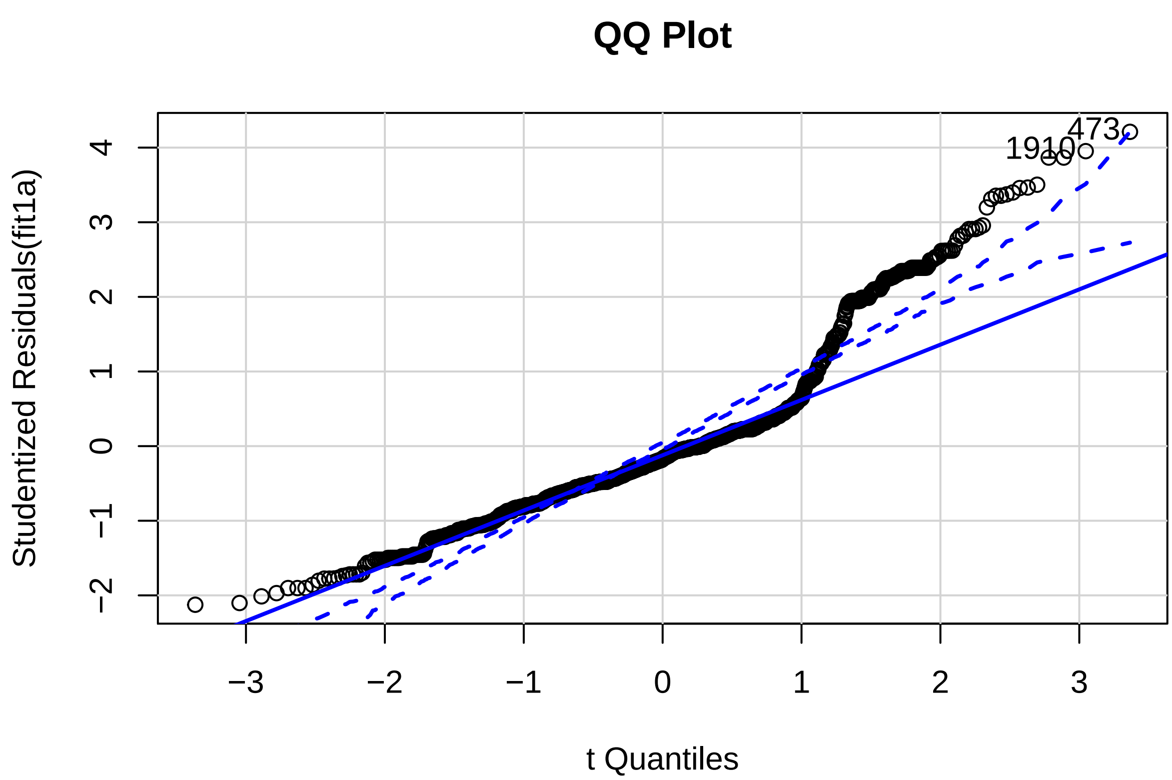
```
##     rstudent unadjusted p-value Bonferroni p
## 473 4.211351         2.7212e-05     0.034505
```

```r
hist(fit1a$residuals)
```



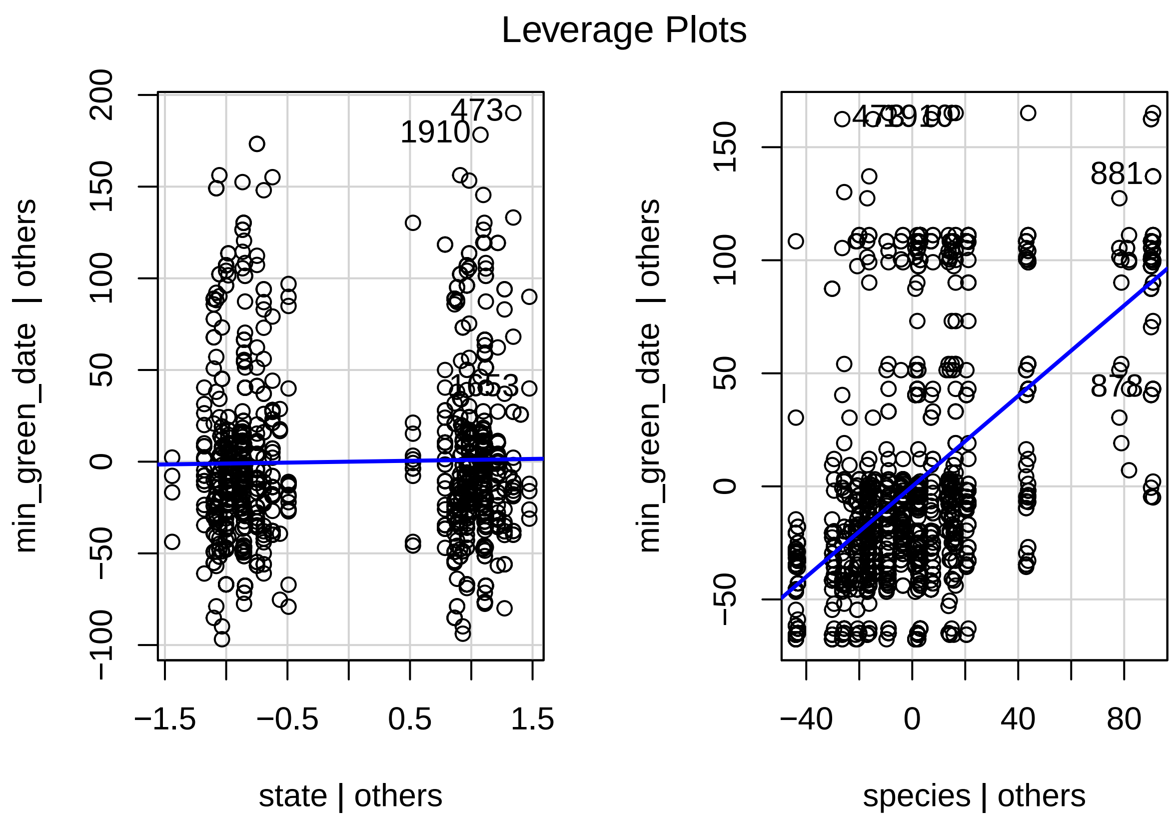

```
qqPlot(fit1a, main = "QQ Plot")
```

```
## Warning in rlm.default(x, y, weights, method = method, wt.method = wt.method, :
## 'rlm' failed to converge in 20 steps
```

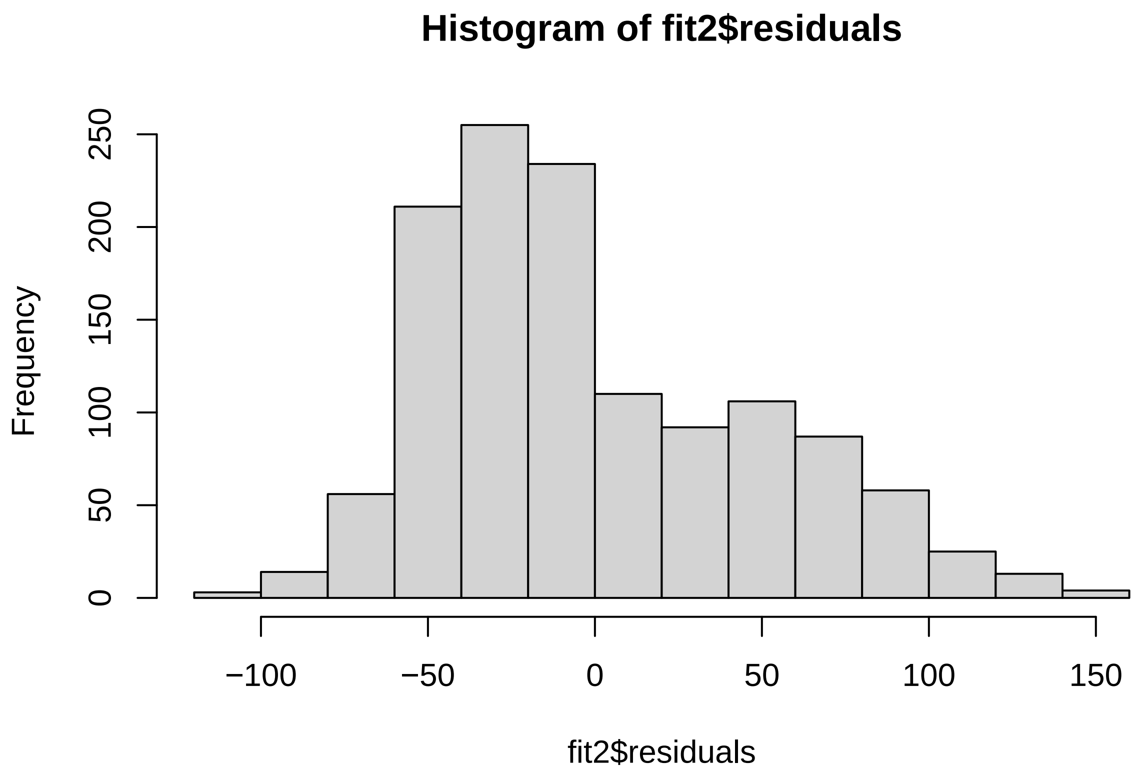


```
## 473 1910
## 283 1152
```

```r
leveragePlots(fit1a)
```



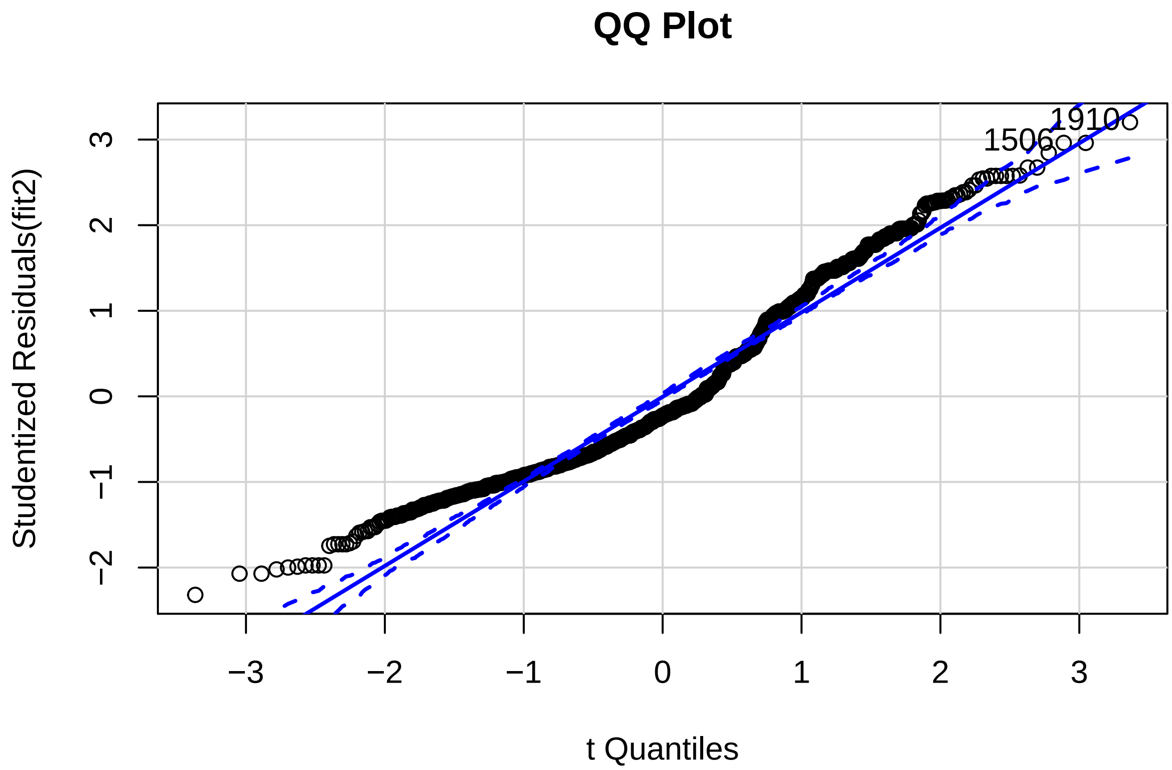

```r
# checking fit for date as a function of state and year
fit2 <- lm(spp_half_cover_date ~ state + species + year, data = green_kbs)
hist(fit2$residuals)
```




```r
outlierTest(fit2)   # no outliers
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferroni p
## 1910 3.202366          0.0013976           NA
```

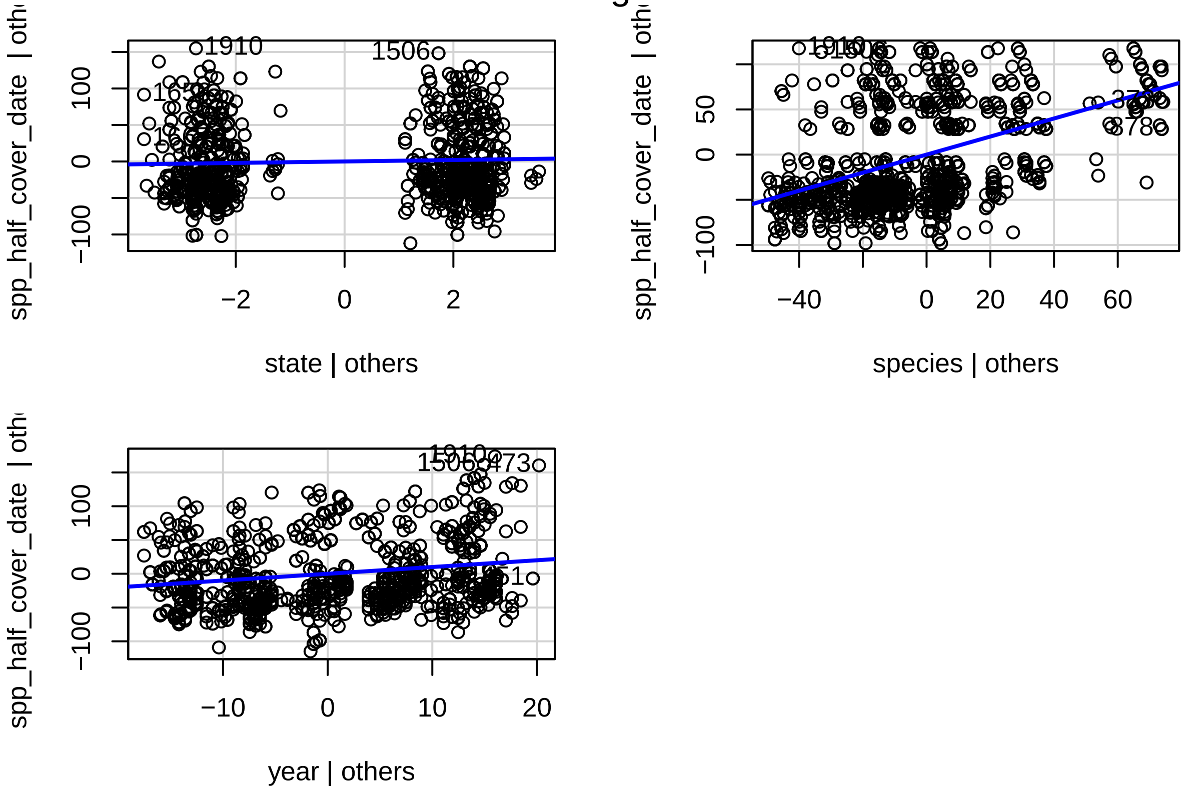
```r
qqPlot(fit2, main = "QQ Plot")
```




```
## 1506 1910
##  943 1152
```

```r
leveragePlots(fit2)
```

Leverage Plots

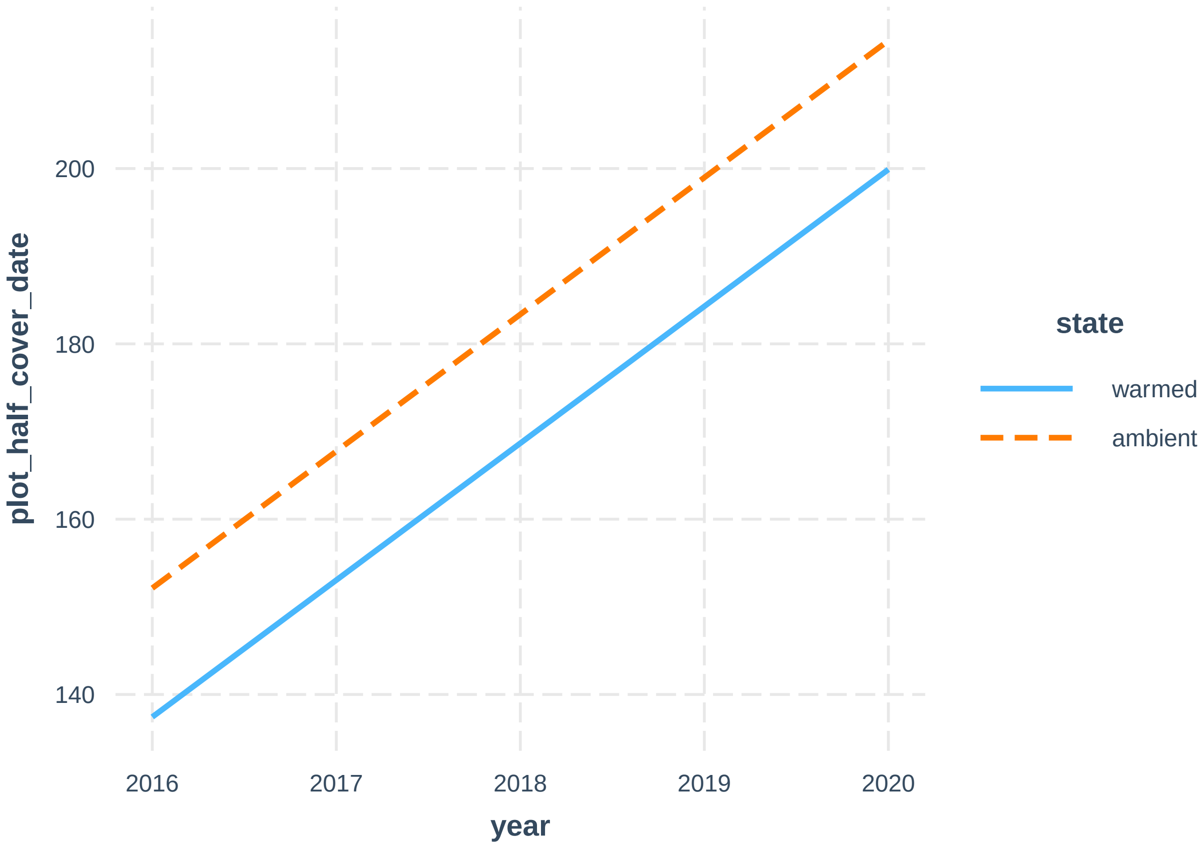


Normal distribution after accounting for species - we will be using species as a random effect to account for their variation. Set up some linear mixed effects models to evaluate. From Ben Bolker: "The traditional view of random effects is as a way to do correct statistical tests when some observations are correlated. ... Random effects are especially useful when we have (1) lots of levels (e.g., many species or blocks), (2) relatively little data on each level (although we need multiple samples from most of the levels), and (3) uneven sampling across levels. People sometimes say that random effects are "factors that you aren't interested in." This is not always true. While it is often the case in ecological experiments (where variation among sites is usually just a nuisance), it is sometimes of great interest." In our case, variation among plots is a nuisance, and not something we're interested in. For some questions, variation among species is also a nuisance for us. It's possible that variation among years is a nuisance if we only care about warm vs. ambient, but I think time is an interesting variable to consider with this study.

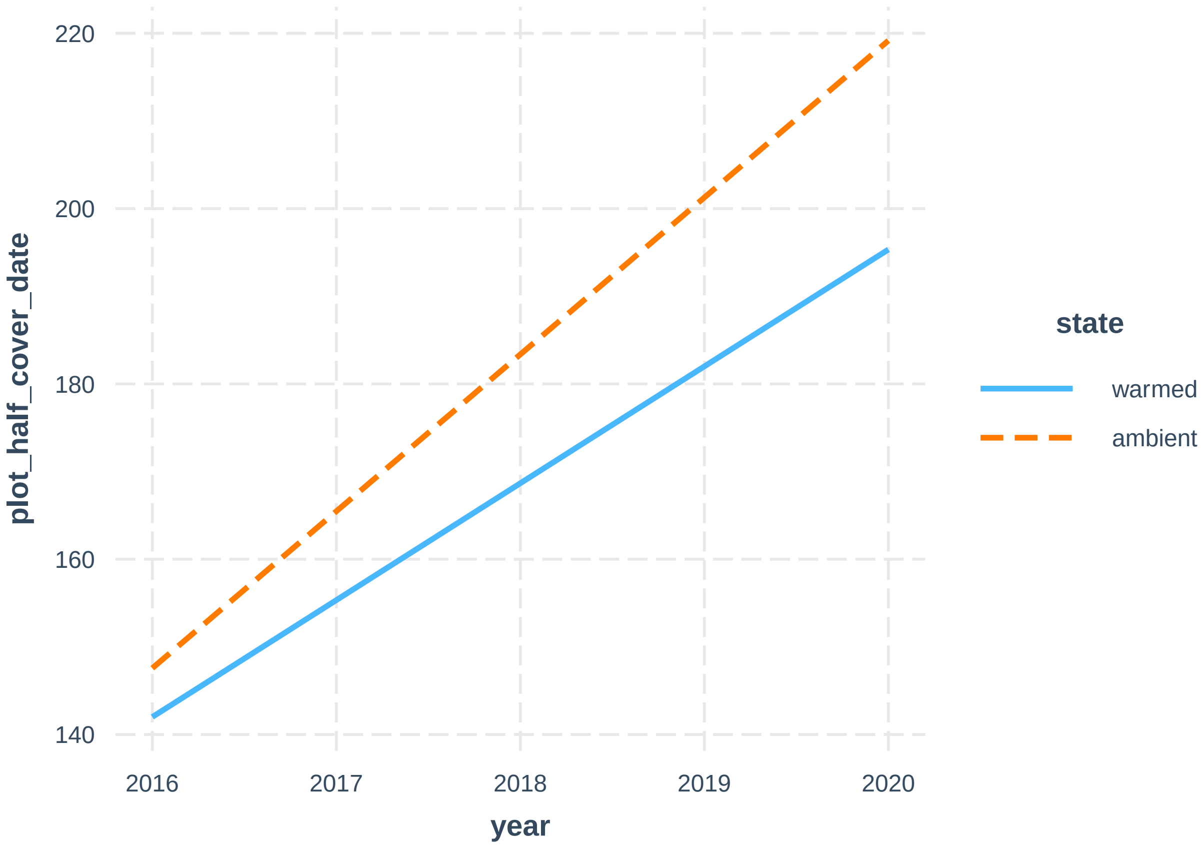
We should also think about how we're treating year. Some of the models have a state * year interaction as a fixed effect, which means that the warming or ambient treatment could affect the half_cover_date differently over time (there would be a different slope for each state in the relationship between half_cover_date (y) and year (x)). If we just had state + year, the states would have the same slope, indicating that they have no interaction in their effect on half_cover_date (but they could still have different intercepts).

```
# 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(plot_half_cover_date ~ state + year, data = green_kbsp)
interact_plot(fit3, pred = year, modx = state)
```

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

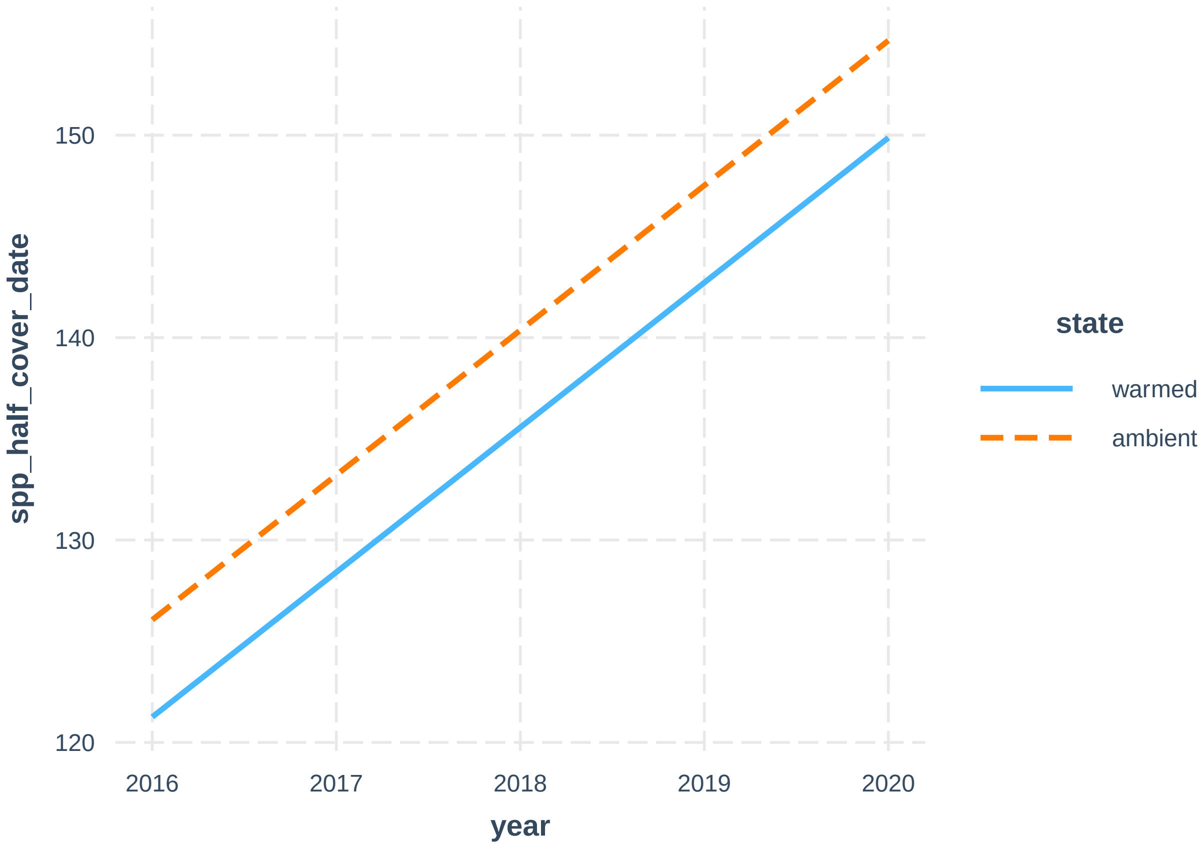



```
fit4 <- lm(plot_half_cover_date ~ state * year, data = green_kbsp)
interact_plot(fit4, pred = year, modx = state)
```



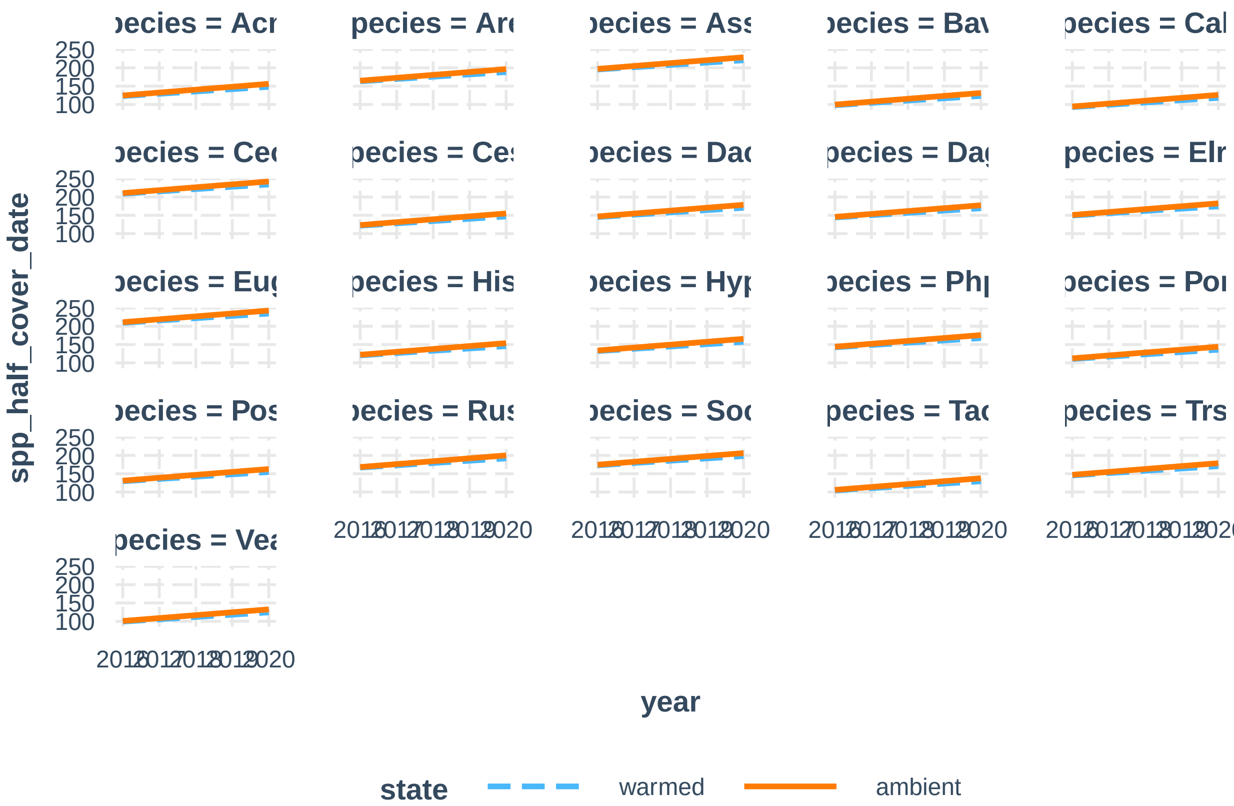

```
fit5 <- lm(spp_half_cover_date ~ state + year + species, data = green_kbs)
interact_plot(fit5, pred = year, modx = state)
```

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

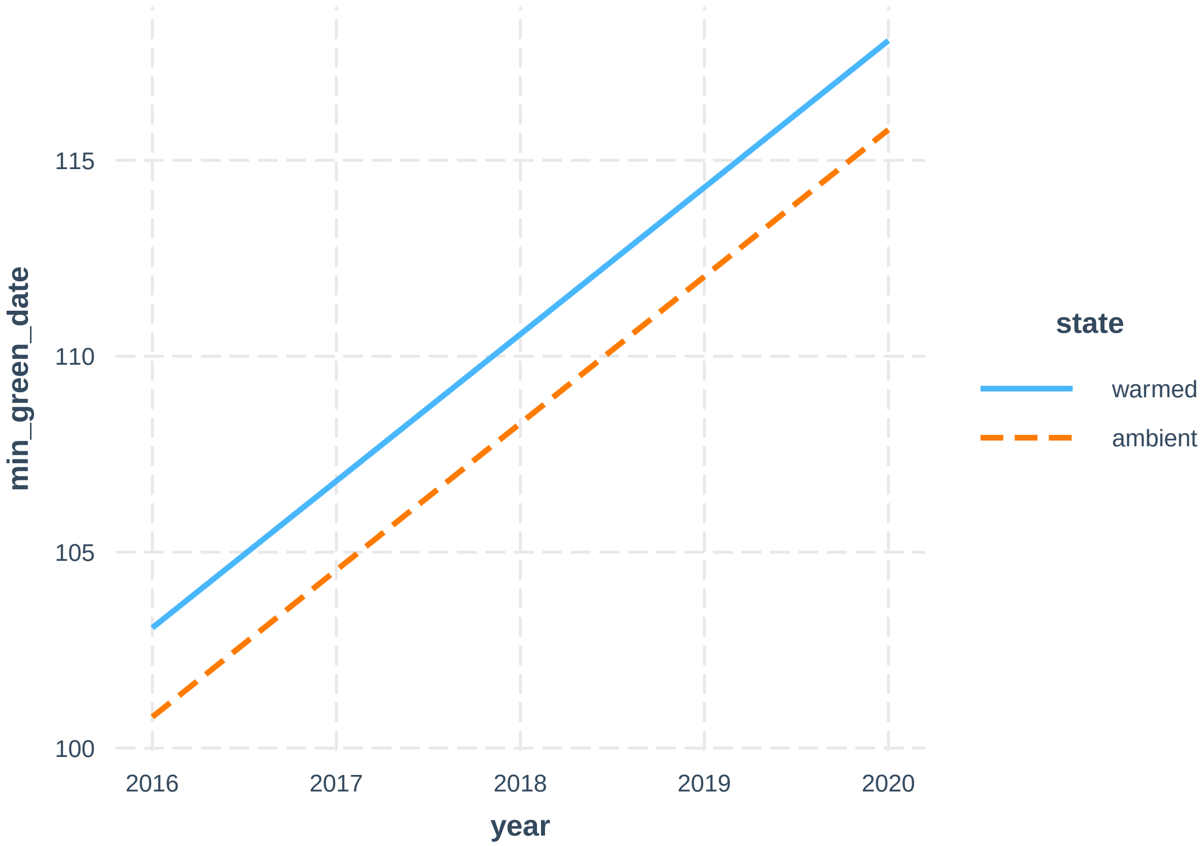



```
fit6 <- lm(spp_half_cover_date ~ state * year + species, data = green_kbs)
interact_plot(fit6, pred = year, modx = state, mod2 = species)
```

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




```
fit7 <- lm(min_green_date ~ state + year + species, data = green_kbs)
interact_plot(fit7, pred = year, modx = state)

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




```r
fit8 <- lm(min_green_date ~ state * year + species, data = green_kbs)
interact_plot(fit8, pred = year, modx = state, mod2 = species)
```

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

Mixed Effects Models:

```
# Start by replicating (almost) what we did in the Oecologia 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(spp_half_cover_date ~ state * year + insecticide * year + (1 | species) +
    (1 | plot), green_kbs, REML = FALSE)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
mod2 <- lmer(spp_half_cover_date ~ state * year + insecticide * year + (1 | species),
    green_kbs, REML = FALSE)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
# 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 anove on a regular
# linear model: https://m-clark.github.io/docs/mixedModels/anovamixed.html)
anova(mod1)
```

```
## Analysis of Variance Table
```

```
##                    npar Sum Sq Mean Sq F value
## state                 1   5504    5504  2.2469
## year                  1 136324  136324 55.6476
## insecticide           1   7561    7561  3.0866
## state:year            1   1563    1563  0.6380
## year:insecticide      1   3596    3596  1.4681
```

```r
anova(mod2)
```

```
## Analysis of Variance Table
##                    npar Sum Sq Mean Sq F value
## state                 1   9394    9394  3.7886
## year                  1 133040  133040 53.6567
## insecticide           1  15357   15357  6.1937
## state:year            1   1618    1618  0.6527
## year:insecticide      1   3961    3961  1.5977
```

```r
# Run an ANOVA to test if 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(mod2, mod1)  # They are different so plot as a random effect should stay in the model (we go with
```

```
## Data: green_kbs
## Models:
## mod2: spp_half_cover_date ~ state * year + insecticide * year + (1 |
## mod2:     species)
## mod1: spp_half_cover_date ~ state * year + insecticide * year + (1 |
## mod1:     species) + (1 | plot)
##      npar   AIC   BIC  logLik deviance Chisq Df Pr(>Chisq)
## mod2    8 13587 13628 -6785.6    13571
## mod1    9 13586 13632 -6784.0    13568 3.374  1    0.06623 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mod1)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: spp_half_cover_date ~ state * year + insecticide * year + (1 |
##     species) + (1 | plot)
##    Data: green_kbs
##
##      AIC      BIC   logLik deviance df.resid
##  13585.9  13632.2  -6784.0  13567.9     1259
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1194 -0.7683 -0.2513  0.6857  3.2607
##
```

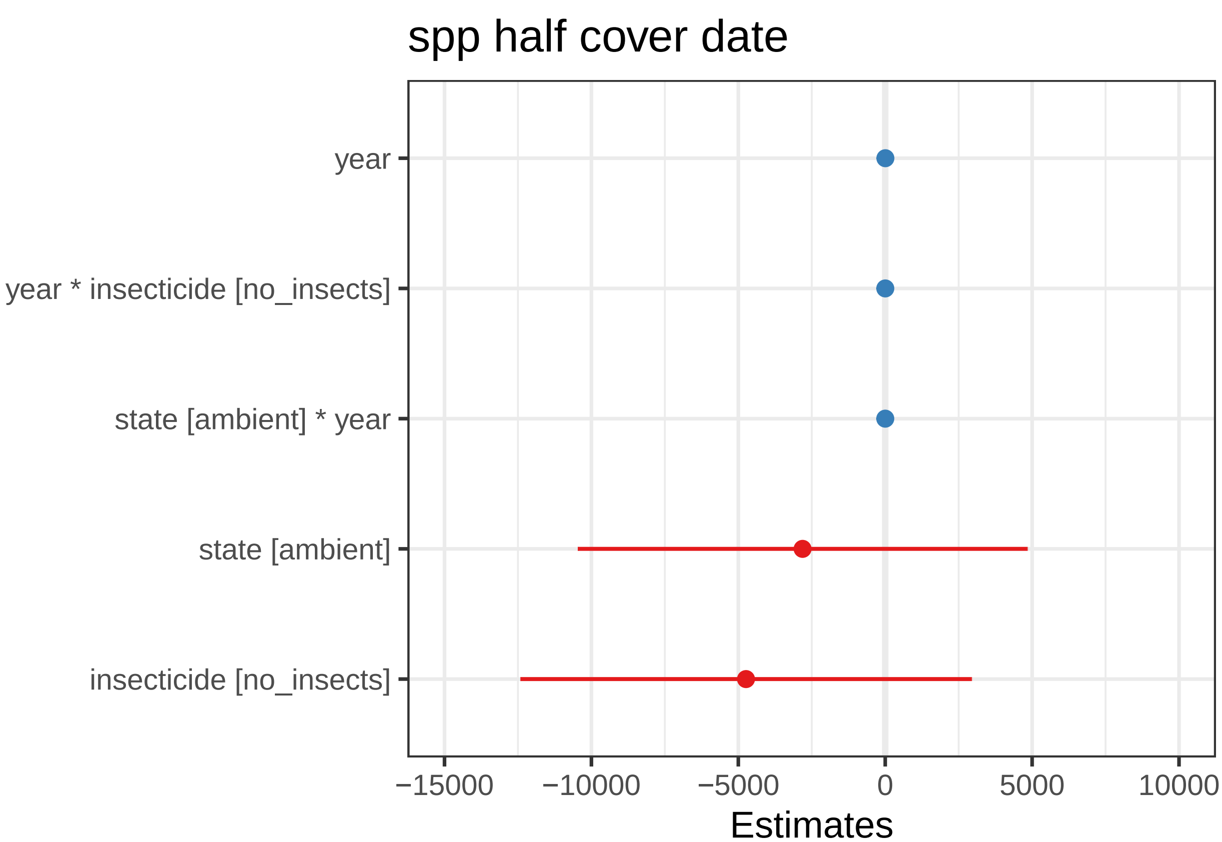
```
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  plot     (Intercept)   32.77   5.725
##  species  (Intercept)  930.34  30.502
##  Residual             2449.77  49.495
## Number of obs: 1268, groups:  plot, 24; species, 21
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)               -11128.065   3367.723  -3.304
## stateambient               -2812.392   3902.785  -0.721
## year                           5.589      1.669   3.349
## insecticideno_insects      -4738.174   3915.775  -1.210
## stateambient:year              1.396      1.934   0.722
## year:insecticideno_insects     2.351      1.941   1.212
##
## Correlation of Fixed Effects:
##             (Intr) sttmbn year   insct_ sttmb:
## stateambint -0.583
## year        -1.000  0.583
## insctcdn_ns -0.519 -0.062  0.519
## statmbnt:yr  0.583 -1.000 -0.583  0.062
## yr:nsctcdn_  0.519  0.062 -0.519 -1.000 -0.062
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

```r
summary(mod2)
```

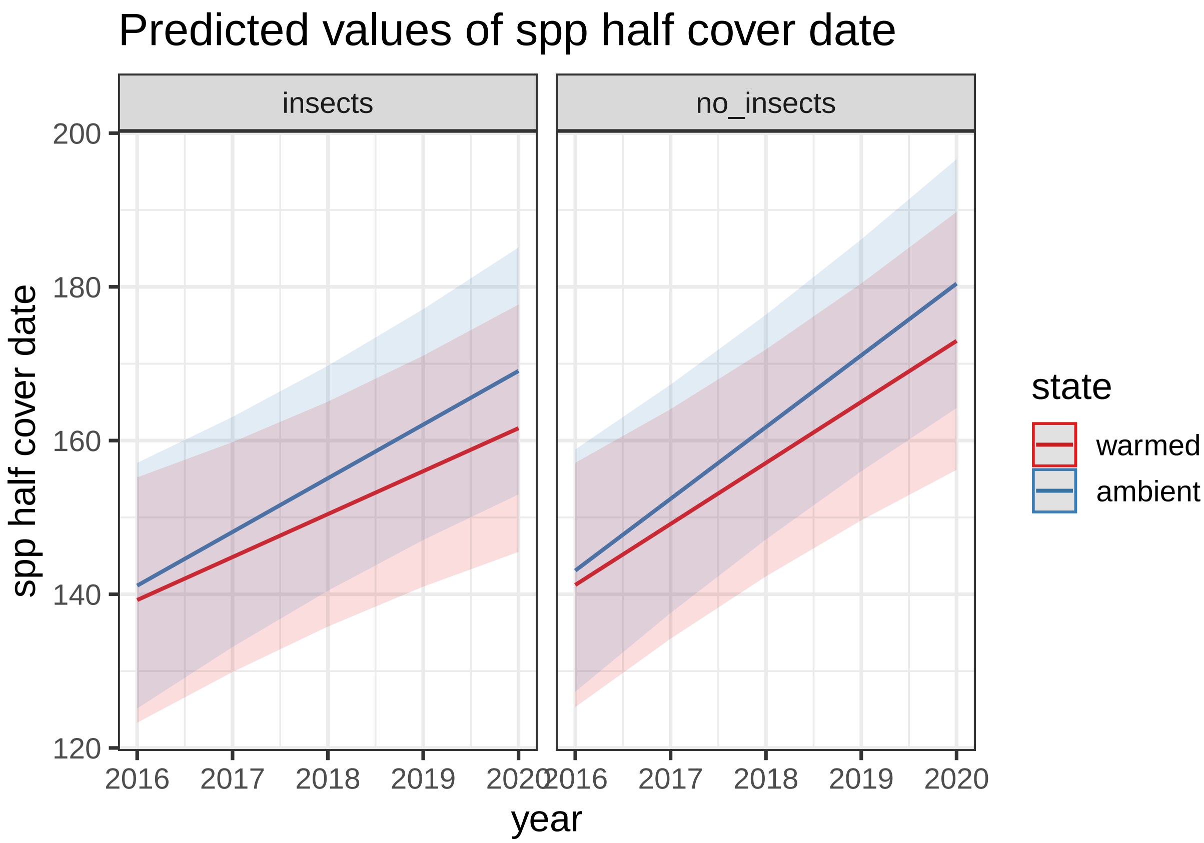
```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: spp_half_cover_date ~ state * year + insecticide * year + (1 |
##     species)
##    Data: green_kbs
##
##      AIC      BIC   logLik deviance df.resid
##  13587.3  13628.4  -6785.6  13571.3     1260
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2600 -0.7665 -0.2441  0.6924  3.2567
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  species  (Intercept)  951.2   30.84
##  Residual             2479.5   49.79
## Number of obs: 1268, groups:  species, 21
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)               -10945.494   3381.228  -3.237
## stateambient               -2850.741   3918.330  -0.728
## year                           5.498      1.676   3.282
## insecticideno_insects      -4965.169   3933.683  -1.262
## stateambient:year              1.415      1.942   0.729
## year:insecticideno_insects     2.464      1.949   1.264
```

```
##
## Correlation of Fixed Effects:
##              (Intr) sttmbn year   insct_ sttmb:
## stateambint -0.583
## year        -1.000  0.583
## insctcdn_ns -0.520 -0.061  0.520
## statmbnt:yr  0.583 -1.000 -0.583  0.061
## yr:nsctcdn_  0.520  0.061 -0.520 -1.000 -0.061
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

```r
# 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 from summary(mod5)
plot_model(mod1, sort.est = TRUE)
```

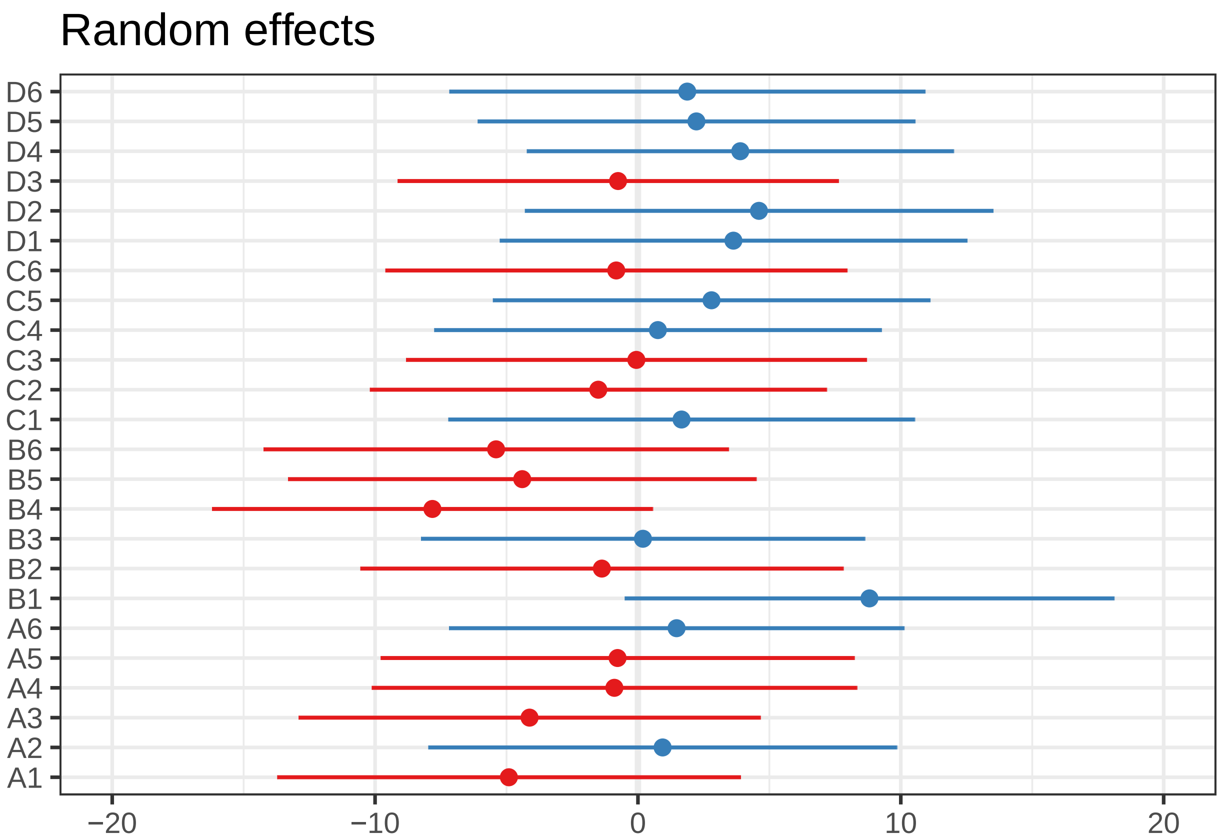



```r
# these are the fixed predicted values:
plot_model(mod1, type = "pred", terms = c("year", "state", "insecticide"))
```

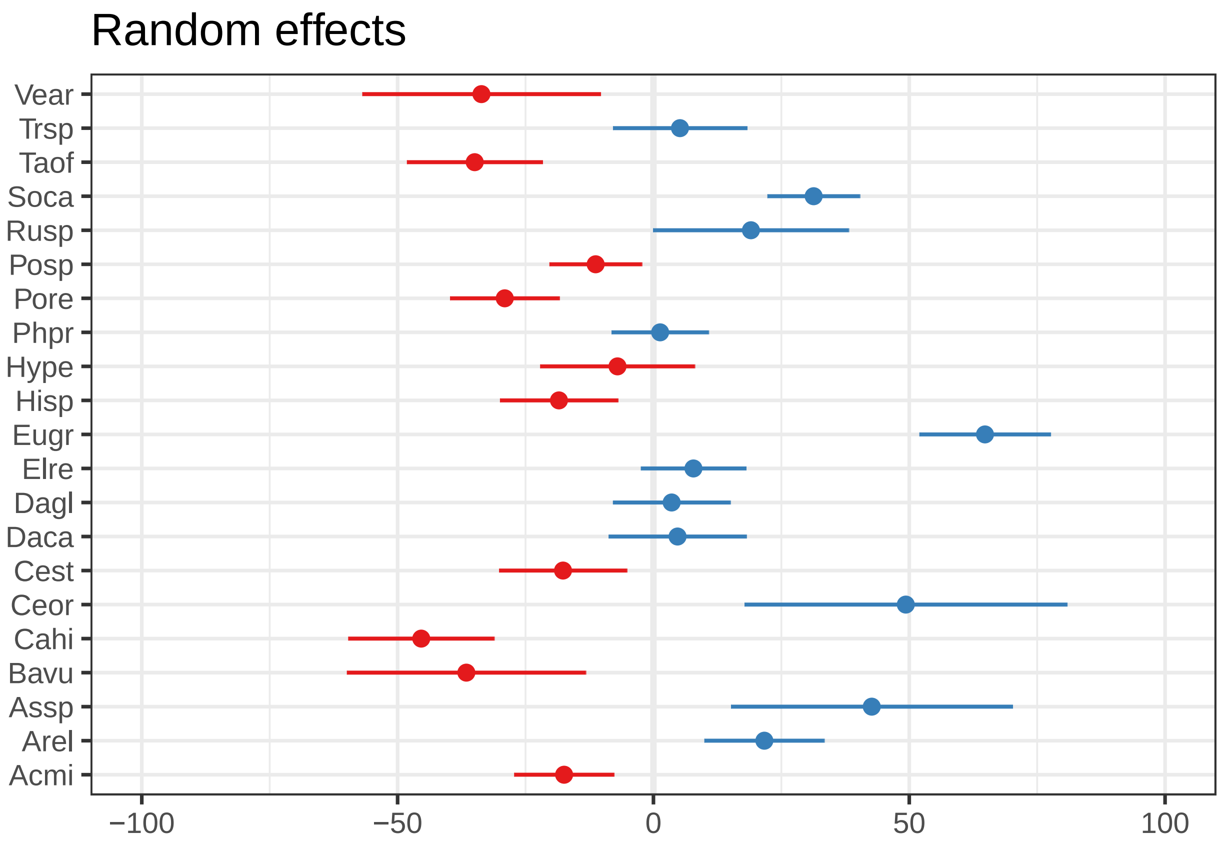



```
# these are the random effects estimates
plot_model(mod1, type = "re", terms = c("species", "plot"))
```

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


Random effects
D6
D5
D4
D3
D2
D1
C6
C5
C4
C3
C2
C1
B6
B5
B4
B3
B2
B1
A6
A5
A4
A3
A2
A1
−20
−10
0
10
20

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




```
# Do we need to include insecticide?
mod3 <- lmer(spp_half_cover_date ~ state * year + (1 | species), green_kbs, REML = FALSE)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
anova(mod1, mod3)
```

```
## Data: green_kbs
## Models:
## mod3: spp_half_cover_date ~ state * year + (1 | species)
## mod1: spp_half_cover_date ~ state * year + insecticide * year + (1 |
## mod1:     species) + (1 | plot)
##      npar   AIC   BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod3    6 13591 13622 -6789.5    13579
## mod1    9 13586 13632 -6784.0    13568 10.994  3    0.01176 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Looks like yes P<0.05, insecticide improves model fit so we will continue to
# include it and stick with mod1

# Does year need to be interactive with insecticide?
mod4 <- lmer(spp_half_cover_date ~ state * year + insecticide + (1 | species) + (1 |
    plot), green_kbs, REML = FALSE)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```r
anova(mod1, mod4)
```

```
## Data: green_kbs
## Models:
## mod4: spp_half_cover_date ~ state * year + insecticide + (1 | species) +
## mod4:     (1 | plot)
## mod1: spp_half_cover_date ~ state * year + insecticide * year + (1 |
## mod1:     species) + (1 | plot)
##      npar   AIC   BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod4    8 13585 13626 -6784.7    13569
## mod1    9 13586 13632 -6784.0    13568 1.4664  1     0.2259
```

```r
# No, P>0.05 so insecticide*year doesn't strongly improve model fit so we will
# shift to mod4
anova(mod3, mod4)
```

```
## Data: green_kbs
## Models:
## mod3: spp_half_cover_date ~ state * year + (1 | species)
## mod4: spp_half_cover_date ~ state * year + insecticide + (1 | species) +
## mod4:     (1 | plot)
##      npar   AIC   BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod3    6 13591 13622 -6789.5    13579
## mod4    8 13585 13626 -6784.7    13569 9.5277  2   0.008533 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Yes, P<0.05 so insecticide still improves model fit so we will stay with mod4

# Does year need to be interactive with state?
mod5 <- lmer(spp_half_cover_date ~ state + year + insecticide + (1 | species) + (1 |
    plot), green_kbs, REML = FALSE)
anova(mod4, mod5)
```

```
## Data: green_kbs
## Models:
## mod5: spp_half_cover_date ~ state + year + insecticide + (1 | species) +
## mod5:     (1 | plot)
## mod4: spp_half_cover_date ~ state * year + insecticide + (1 | species) +
## mod4:     (1 | plot)
##      npar   AIC   BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod5    7 13584 13620 -6785.0    13570
## mod4    8 13585 13626 -6784.7    13569 0.6369  1     0.4249
```

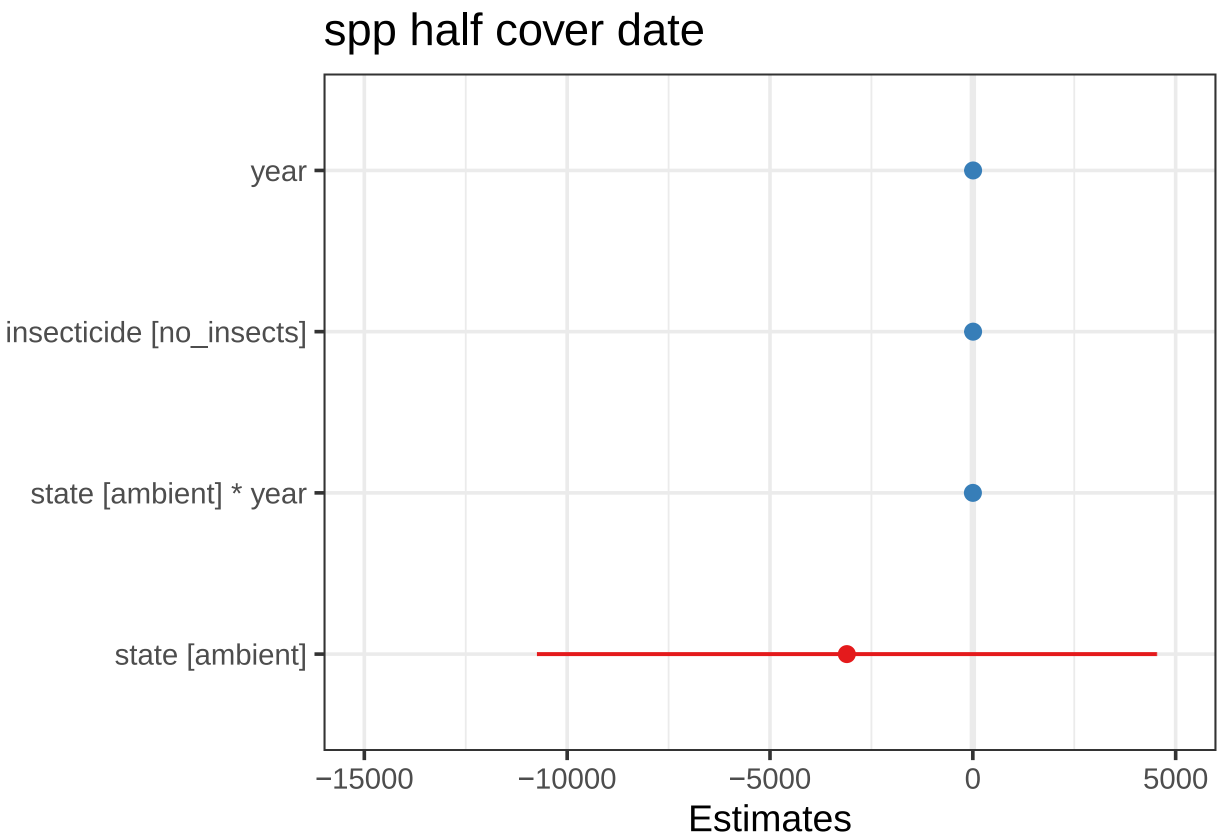
```r
# No, P>0.05 so state*year doesn't improve model fit so we could drop it and go
# with mod5, but note that the AIC values are super close. mod4 makes sense, with
# increased divergence between warmed and ambient.
summary(mod4)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: spp_half_cover_date ~ state * year + insecticide + (1 | species) +
##     (1 | plot)
##    Data: green_kbs
##
##      AIC      BIC   logLik deviance df.resid
##  13585.4  13626.5  -6784.7  13569.4     1260
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1148 -0.7659 -0.2457  0.6789  3.2173
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  plot     (Intercept)   33.48   5.786
##  species  (Intercept)  928.65  30.474
##  Residual             2452.34  49.521
## Number of obs: 1268, groups:  plot, 24; species, 21
##
## Fixed effects:
##                         Estimate Std. Error t value
## (Intercept)           -13246.986   2880.397  -4.599
## stateambient           -3106.128   3897.363  -0.797
## year                       6.639      1.427   4.651
## insecticideno_insects      6.358      3.690   1.723
## stateambient:year          1.542      1.931   0.798
##
## Correlation of Fixed Effects:
##             (Intr) sttmbn year   insct_
## stateambint -0.721
## year        -1.000  0.721
## insctcdn_ns -0.062  0.029  0.061
## statmbnt:yr  0.721 -1.000 -0.721 -0.029
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

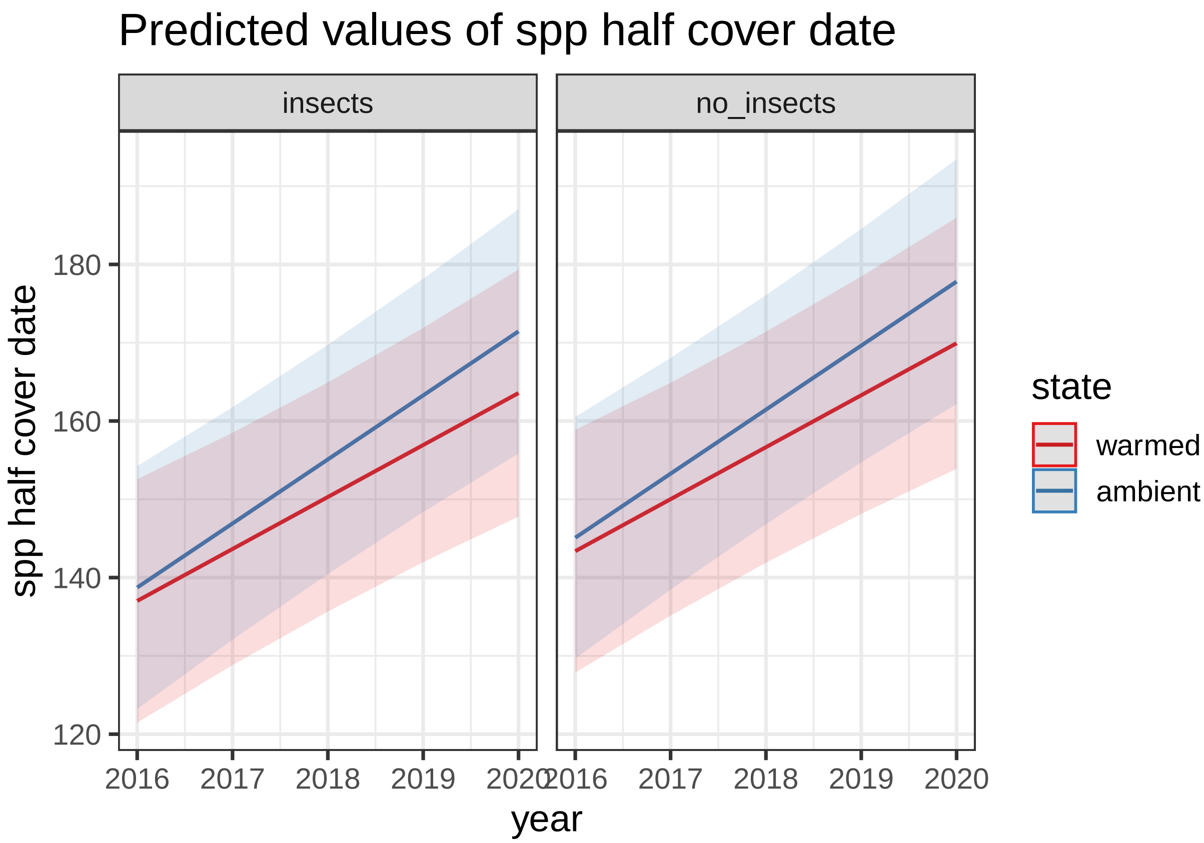
```r
anova(mod4)
```

```
## Analysis of Variance Table
##             npar Sum Sq Mean Sq F value
## state          1   5460    5460  2.2265
## year           1 136373  136373 55.6092
## insecticide    1   7482    7482  3.0510
## state:year     1   1562    1562  0.6371
```

```r
# these are the fixed effects estimates from summary(mod5)
plot_model(mod4, sort.est = TRUE)
```

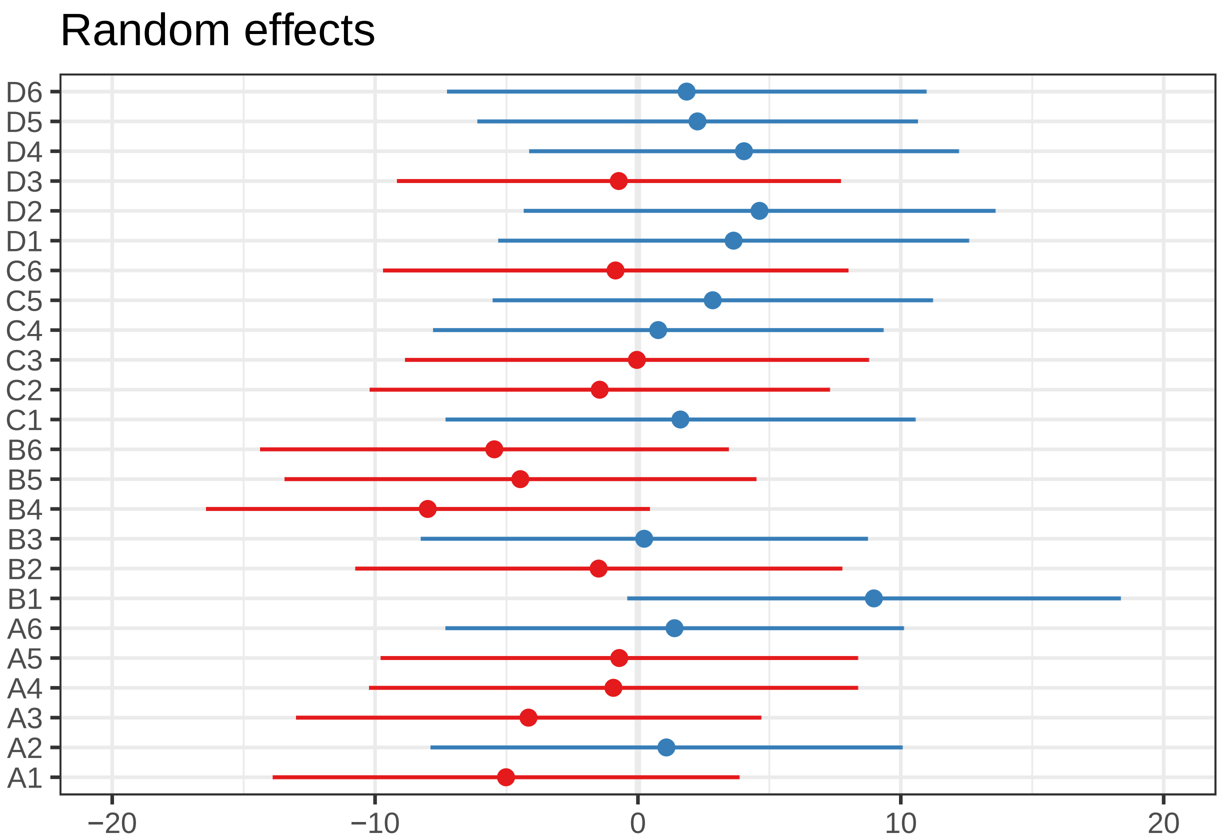



```r
# these are the fixed predicted values:
plot_model(mod4, type = "pred", terms = c("year", "state", "insecticide"))
```



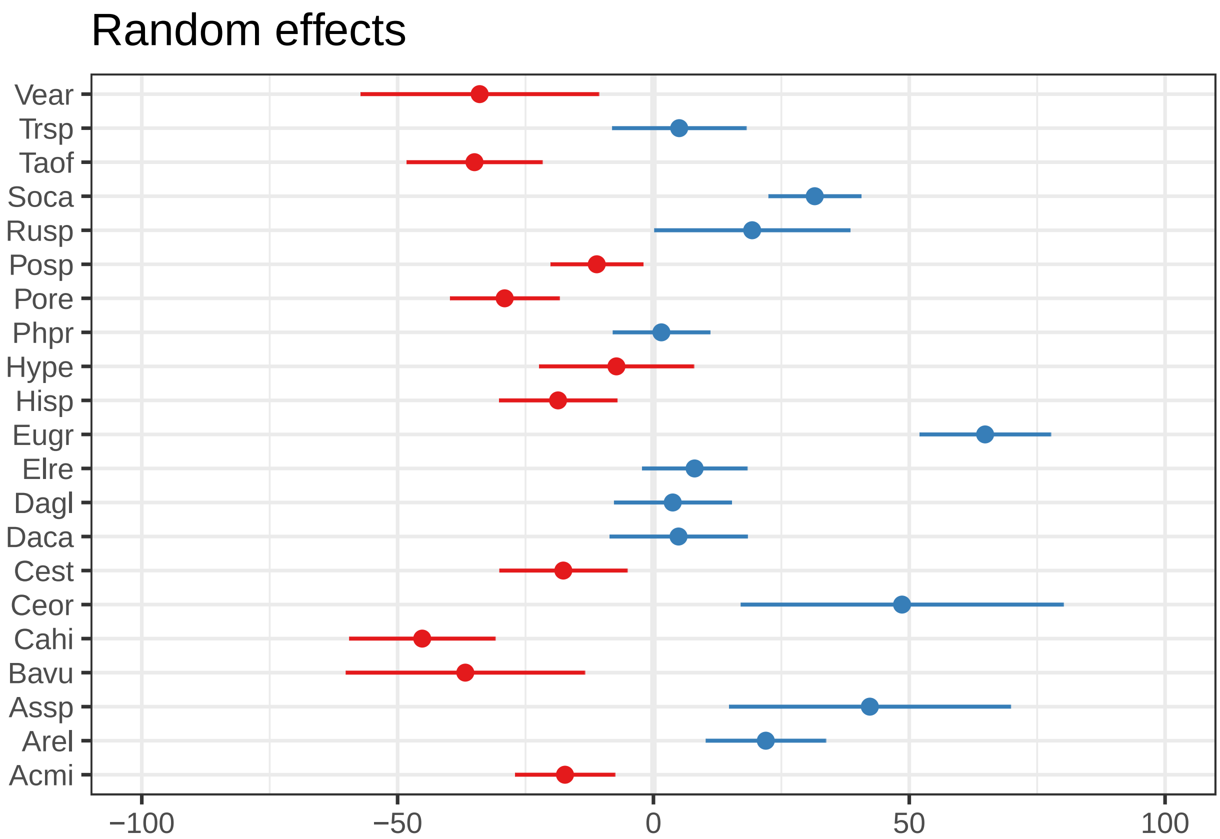

```
# these are the random effects estimates
plot_model(mod4, type = "re", terms = c("species", "plot"))
```

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



Random effects
D6
D5
D4
D3
D2
D1
C6
C5
C4
C3
C2
C1
B6
B5
B4
B3
B2
B1
A6
A5
A4
A3
A2
A1
−20
−10
0
10
20


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




```r
# If we wanted to include plots nested within year it would look like this:
mod7 <- lmer(spp_half_cover_date ~ state * year + insecticide + (1 | species) + (1 +
    year | plot), green_kbs, REML = FALSE)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

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

```r
anova(mod4, mod7)
```

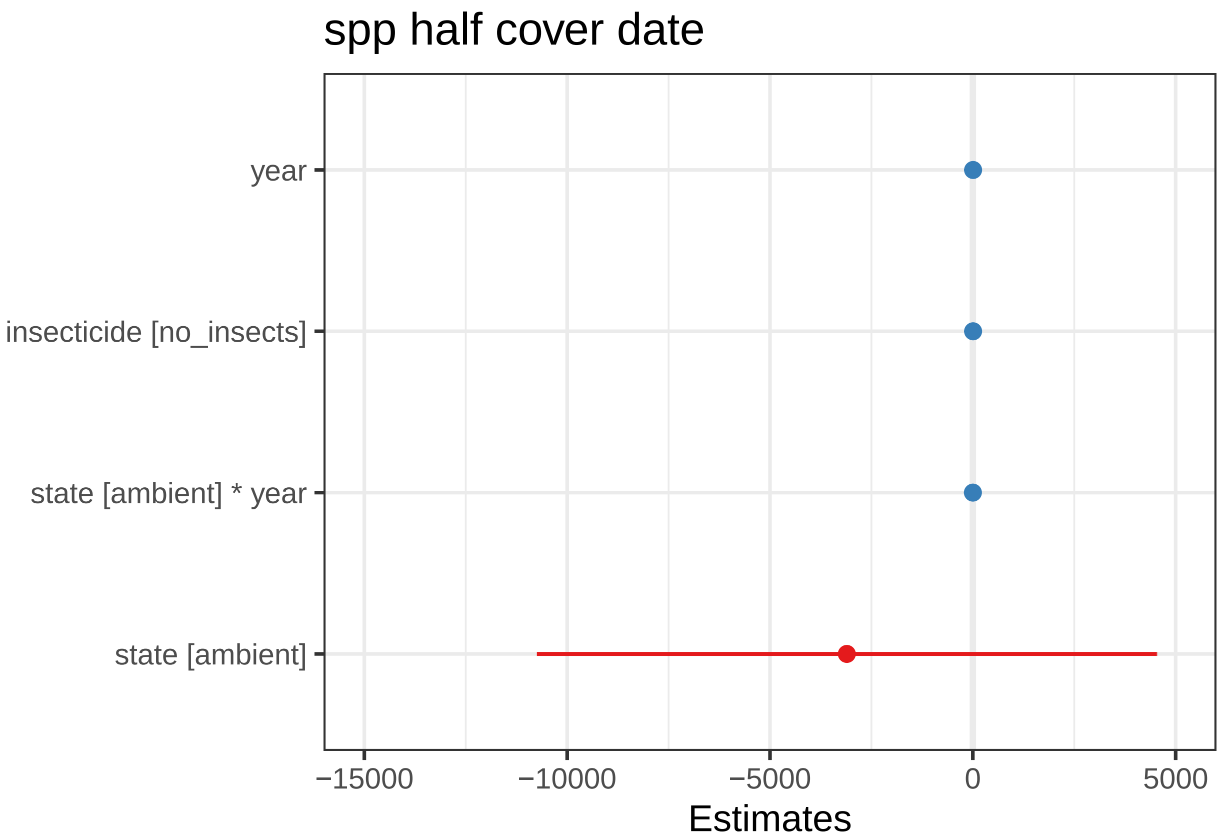
```
## Data: green_kbs
## Models:
## mod4: spp_half_cover_date ~ state * year + insecticide + (1 | species) +
## mod4:     (1 | plot)
## mod7: spp_half_cover_date ~ state * year + insecticide + (1 | species) +
## mod7:     (1 + year | plot)
##      npar   AIC   BIC  logLik deviance Chisq Df Pr(>Chisq)
## mod4    8 13585 13626 -6784.7    13569
## mod7   10 13590 13641 -6784.7    13570     0  2          1
```

```r
anova(mod7)
```

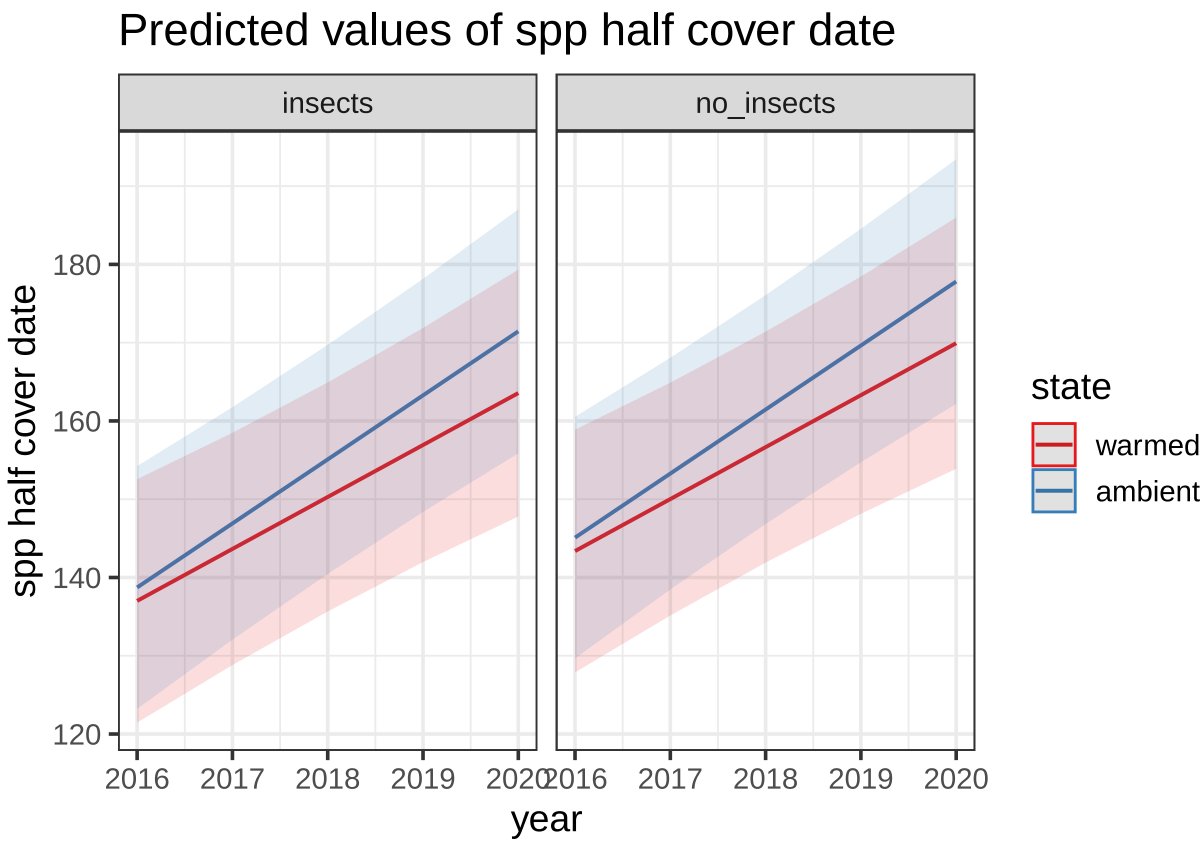
```
## Analysis of Variance Table
##              npar Sum Sq Mean Sq F value
```

```
## state           1   5458    5458  2.2259
## year            1 136384  136384 55.6158
## insecticide     1   7450    7450  3.0380
## state:year      1   1562    1562  0.6371
```

```r
# Yup, seems to matter but it is making this more complex, though not overly so
# because it's on the random effects structure only.
plot_model(mod7, sort.est = TRUE)
```

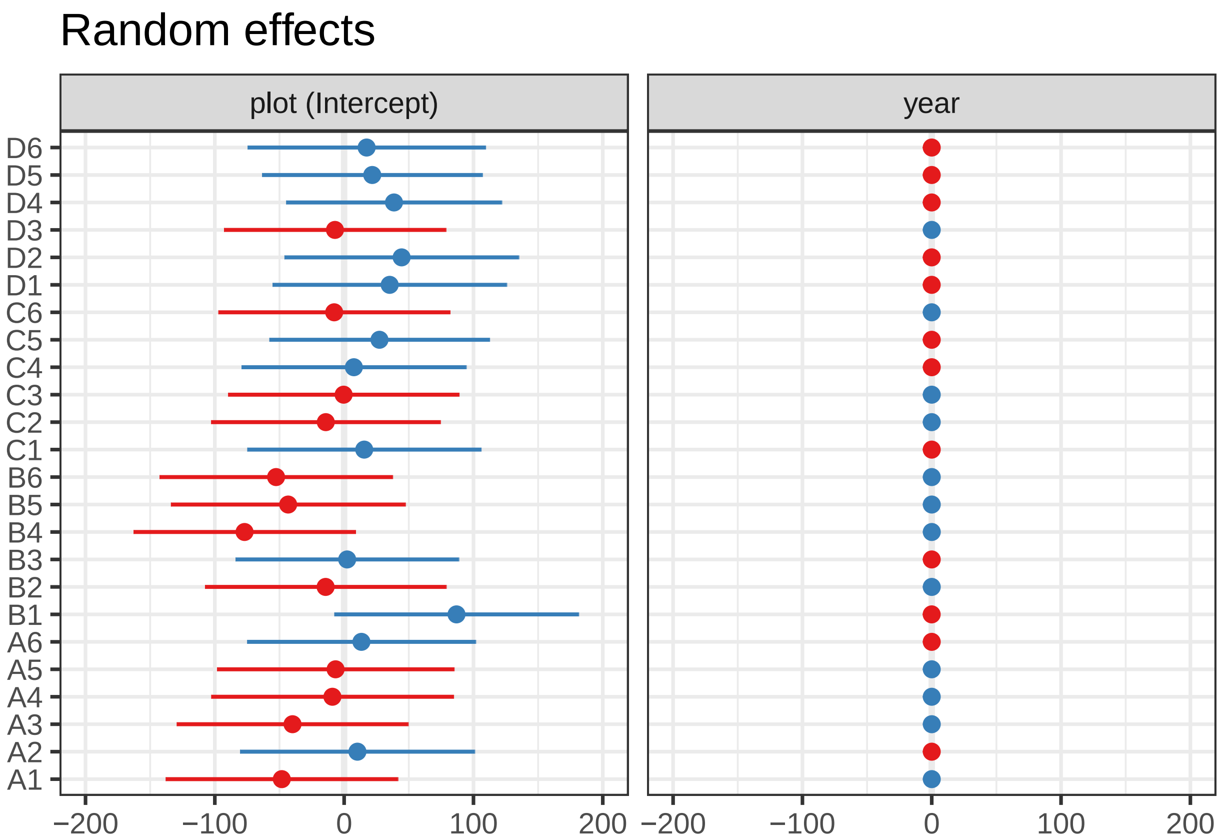



```r
# these are the fixed predicted values:
plot_model(mod7, type = "pred", terms = c("year", "state", "insecticide"))
```

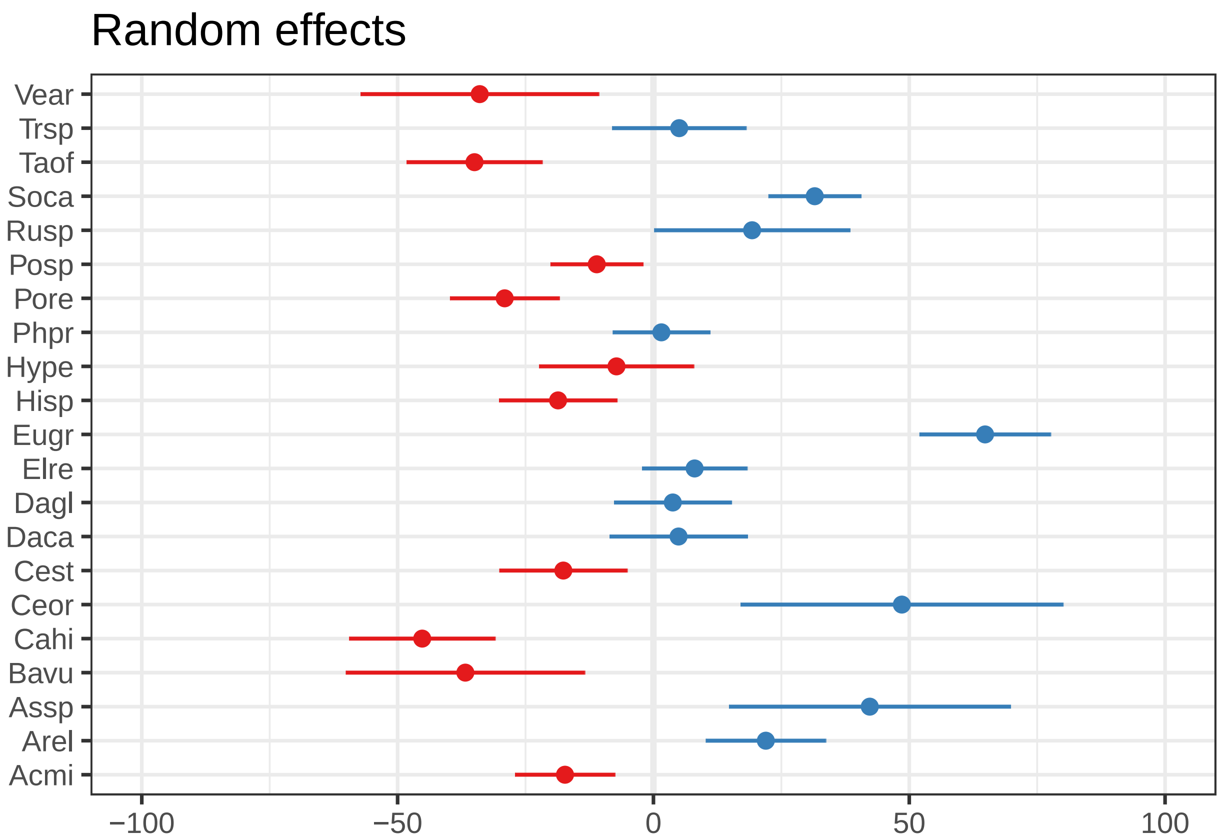



```
# these are the random effects estimates
plot_model(mod7, type = "re", terms = c("species", "plot"))
```

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


Random effects
plot (Intercept)
year
D6
D5
D4
D3
D2
D1
C6
C5
C4
C3
C2
C1
B6
B5
B4
B3
B2
B1
A6
A5
A4
A3
A2
A1
−200
−100
0
100
200
−200
−100
0
100
200

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




```
# mod4 (and mod7) 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: We're going to use lmerTest for
# this example so you can see how to run through post-hoc tests.
library(lmerTest)
```

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

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

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

```
mod8 <- lmer(spp_half_cover_date ~ state + species + (1 + year | plot), green_kbs,
    REML = FALSE)
```

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

## Warning: Model failed to converge with 1 negative eigenvalue: -3.2e-01
```

```r
anova(mod7, mod8)  # model 8 is a better fit to data
```

```
## Data: green_kbs
## Models:
## mod7: spp_half_cover_date ~ state * year + insecticide + (1 | species) +
## mod7:     (1 + year | plot)
## mod8: spp_half_cover_date ~ state + species + (1 + year | plot)
##      npar   AIC   BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod7   10 13590 13641 -6784.7    13570
## mod8   26 13595 13729 -6771.6    13543 26.365 16    0.04911 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mod8)
```

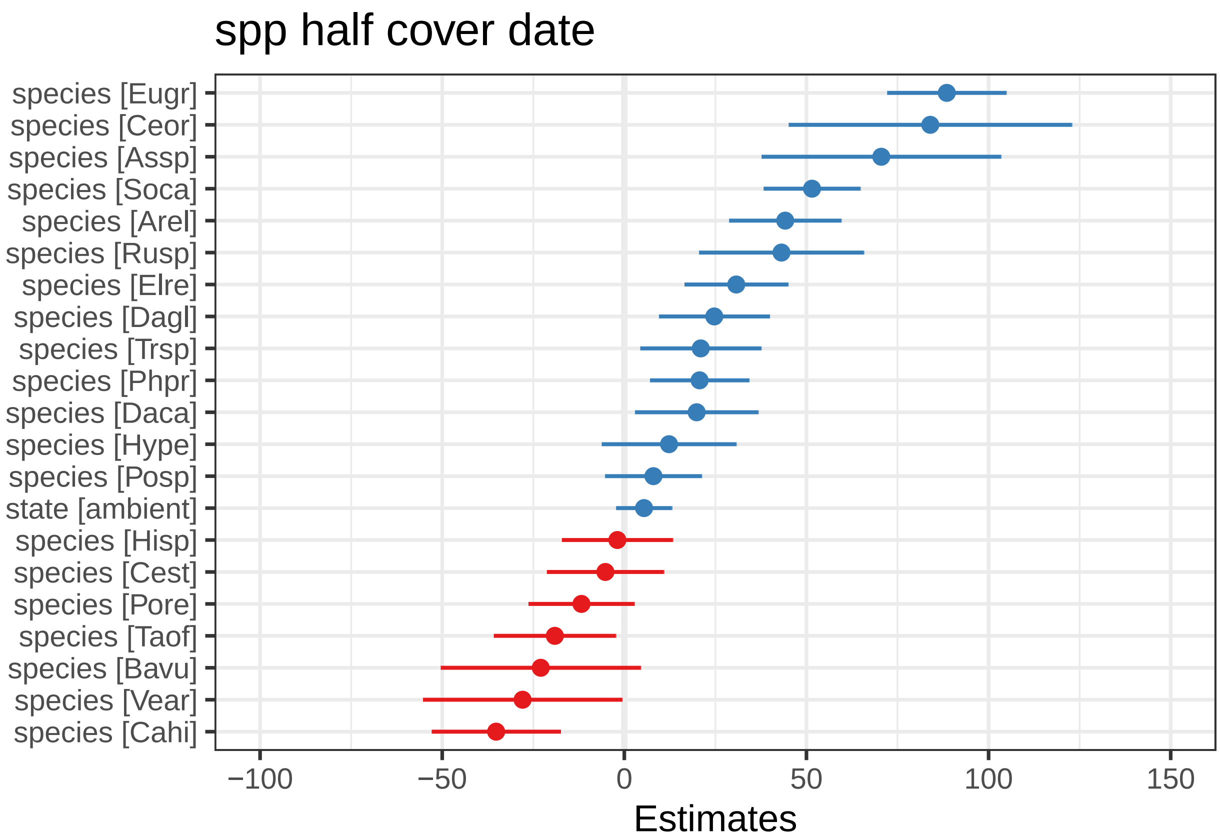
```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: spp_half_cover_date ~ state + species + (1 + year | plot)
##    Data: green_kbs
##
##      AIC      BIC   logLik deviance df.resid
##  13595.1  13728.9  -6771.6  13543.1     1242
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1232 -0.7237 -0.2572  0.7334  3.4828
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  plot     (Intercept) 2.537e+03 50.36370
##           year        4.741e-04  0.02177 -1.00
##  Residual             2.518e+03 50.17603
## Number of obs: 1268, groups:  plot, 24
##
## Fixed effects:
##               Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)    133.901      5.546  302.106  24.143  < 2e-16 ***
## stateambient     5.412      3.897   21.438   1.389 0.179229
## speciesArel     44.169      7.844 1260.160   5.631 2.21e-08 ***
## speciesAssp     70.545     16.757 1267.942   4.210 2.73e-05 ***
## speciesBavu   -22.949     13.996 1267.915  -1.640 0.101318
## speciesCahi   -35.193      9.011 1257.335  -3.906 9.90e-05 ***
## speciesCeor     84.001     19.822 1261.449   4.238 2.42e-05 ***
## speciesCest     -5.209      8.183 1253.784  -0.637 0.524527
## speciesDaca     19.856      8.627 1256.246   2.302 0.021515 *
## speciesDagl     24.693      7.743 1258.750   3.189 0.001462 **
## speciesElre     30.733      7.252 1252.267   4.238 2.42e-05 ***
## speciesEugr     88.527      8.335 1264.020  10.621  < 2e-16 ***
## speciesHisp     -1.916      7.762 1250.982  -0.247 0.805081
## speciesHype     12.271      9.416 1266.314   1.303 0.192754
## speciesPhpr     20.661      6.936 1250.109   2.979 0.002948 **
## speciesPore    -11.775      7.410 1256.420  -1.589 0.112266
```

```
## speciesPosp     7.985      6.762 1247.878   1.181 0.237868
## speciesRusp    43.145     11.525 1262.439   3.744 0.000189 ***
## speciesSoca    51.527      6.762 1247.878   7.620 4.99e-14 ***
## speciesTaof   -19.080      8.535 1265.042  -2.236 0.025558 *
## speciesTrsp    20.983      8.462 1254.243   2.480 0.013279 *
## speciesVear   -27.931     13.943 1264.466  -2.003 0.045361 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 22 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it

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

```r
anova(mod8)  # investigates whether at least one of the levels within each factor is significantly diff
```

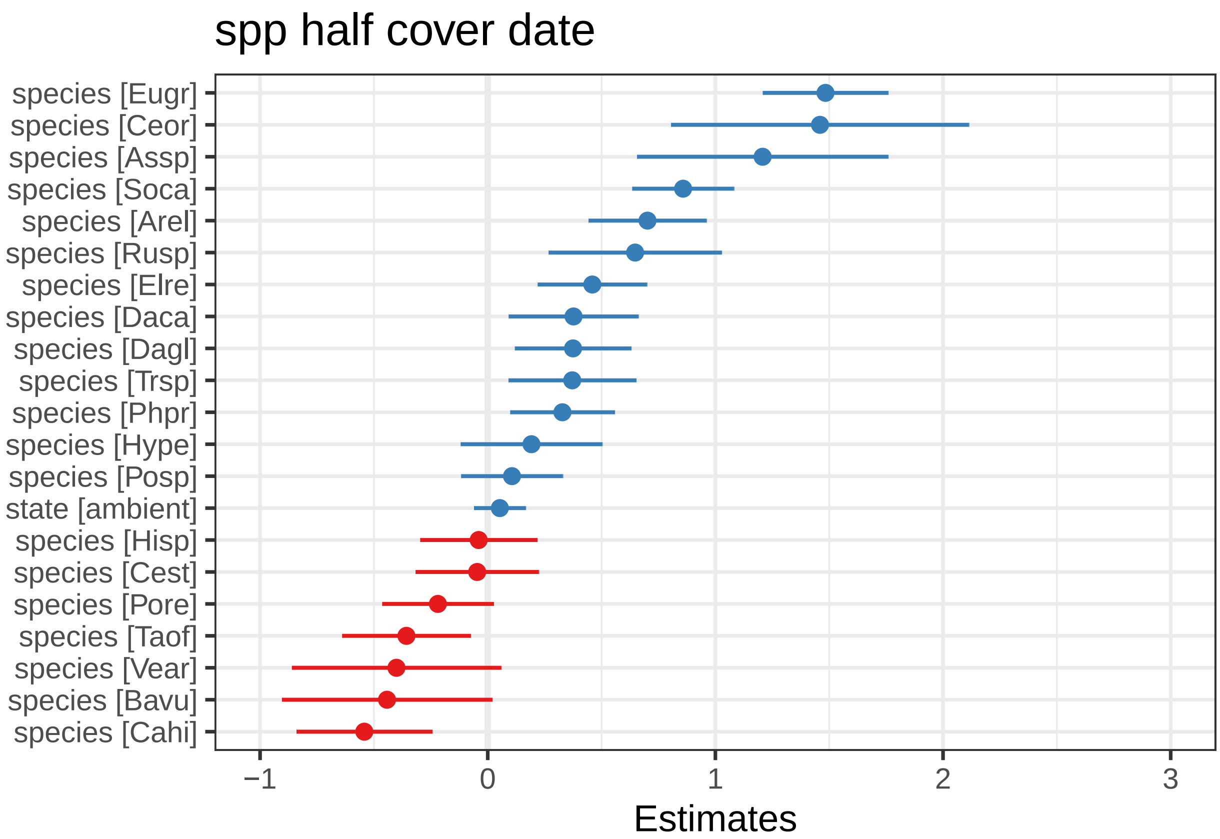
```
## Type III Analysis of Variance Table with Satterthwaite's method
##         Sum Sq Mean Sq NumDF   DenDF F value Pr(>F)
## state     4854    4854     1   21.44  1.9281 0.1792
## species 989680   49484    20 1260.49 19.6550 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Yes, at least one of the species is different (they do not all have the same
# half cover dates).

# Take a look at the estimates for each fixed effect. These are the estimates
# from summary(mod8). 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(mod8, sort.est = TRUE)
```

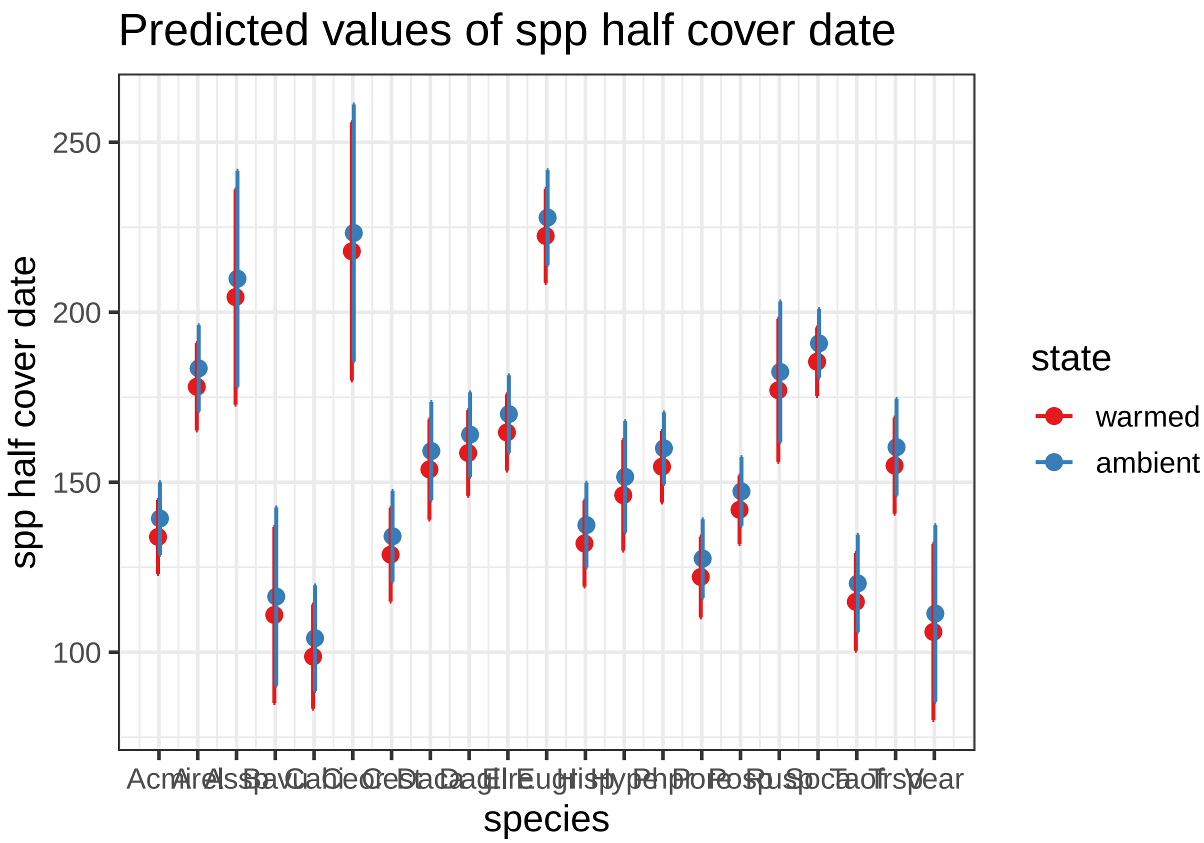



```
# if you want to standardize the estimates:
plot_model(mod8, sort.est = TRUE, type = "std")
```

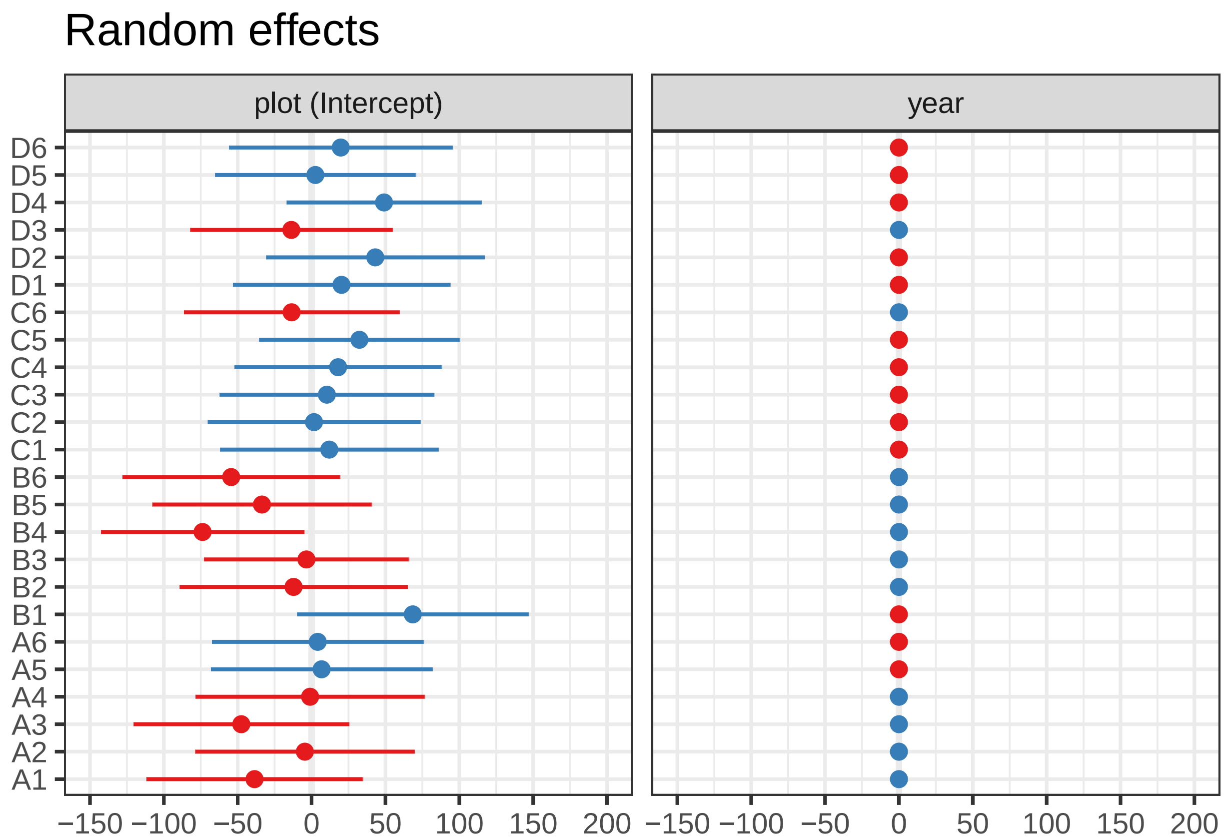
```
## boundary (singular) fit: see ?isSingular
```




```r
# these are the fixed predicted values: - note this is a new plot
plot_model(mod8, type = "pred", terms = c("species", "state"))
```




```
# these are the random effects estimates
plot_model(mod8, type = "re")
```




```
# You could now run some post hoc tests on these (see:
# https://stats.stackexchange.com/questions/169543/output-of-fixed-effects-summary-in-lmertest-in-r-and

# Here are some other options for plotting these plots above:
# https://stackoverflow.com/questions/31075407/plot-mixed-effects-model-in-ggplot

# Here's another approach:
# https://stats.stackexchange.com/questions/98958/plots-to-illustrate-results-of-linear-mixed-effect-mo
# Not quite working yet:
newdat <- expand.grid(state = unique(green_kbs$state), year = c(min(green_kbs$year),
    max(green_kbs$year)), insecticide = unique(green_kbs$insecticide))
p <- ggplot(green_kbs, aes(x = year, y = spp_half_cover_date, colour = state, shape = insecticide)) +
    geom_point(size = 3) + geom_line(aes(y = predict(mod5), group = species, size = "species")) +
    geom_line(data = newdat, aes(y = predict(mod5, level = 0, newdata = newdat),
        size = "Population")) + scale_size_manual(name = "Predictions", values = c(species = 0.5,
    Population = 3)) + # facet_wrap(~insecticide) +
theme_bw(base_size = 22)
# print(p)
```

```
# 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.
mod4 <- lmer(spp_half_cover_date ~ state * year + (1 + year | species), green_kbs)
```

```
## Warning: Some predictor variables are on very different scales: consider
```

```
## rescaling

## boundary (singular) fit: see ?isSingular

## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
# 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)
mod5 <- lmer(spp_half_cover_date ~ state * year + (1 | species), green_kbs)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling

## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
summary(mod5)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: spp_half_cover_date ~ state * year + (1 | species)
##    Data: green_kbs
##
## REML criterion at convergence: 13564.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2175 -0.7629 -0.2477  0.6407  3.1579
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  species  (Intercept) 1013     31.83
##  Residual             2500     50.00
## Number of obs: 1268, groups:  species, 21
##
## Fixed effects:
##                     Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)       -12607.448   2891.752   1246.674  -4.360 1.41e-05 ***
## stateambient       -3486.287   3924.894   1243.930  -0.888    0.375
## year                   6.323      1.433   1246.676   4.412 1.11e-05 ***
## stateambient:year      1.730      1.945   1243.931   0.890    0.374
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) sttmbn year
## stateambint -0.721
## year        -1.000  0.721
## statmbnt:yr  0.721 -1.000 -0.721
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

ORIGINAL CODE BELOW; not edited by Phoebe

Seeing what other distribution could fit

```
descdist(green_kbs$spp_half_cover_date, discrete = FALSE)
```

While uniform looks the closest, I'll try poisson

```
fit <- lm(spp_half_cover_date ~ state, data = green_kbs)
residual <- fit$residuals
hist(residual, main = "Raw residuals")
pois <- glm(spp_half_cover_date ~ state, data = green_kbs, family = "poisson")
hist(pois$residuals, main = "Poisson glm residuals")
```

Below I try a few different generalized linear models with poisson distribution:

An interaction between state and year, plus insecticide as a fixed effect and species and plot as random effects

```
moda <- glmer(spp_half_cover_date ~ state * year + insecticide + (1 | species) +
    (1 | plot), data = green_kbs, family = poisson)
summary(moda)
```

No interaction between state and year, but with state and insecticide as fixed effects and species and plot as random effects

```
modb <- glmer(spp_half_cover_date ~ state + year + insecticide + (1 | species) +
    (1 | plot), data = green_kbs, family = poisson)
summary(modb)
```

State and insecticide as fixed effects & year, species and plot as random effects

```
modc <- glmer(spp_half_cover_date ~ state + insecticide + (1 | year) + (1 | species) +
    (1 | plot), data = green_kbs, family = poisson)
summary(modc)
```

Because no distributions seems to match well, I'll try a Friedman's test

```
# friedman_kbs <- green_kbs %>% friedman_test(spp_half_cover_date ~ state)
```

Error: Must extract column with a single valid subscript. x Subscript `var` can't be `NA`

Can't figure out what this means

If I include the blocks portion of the formula (from the documentation) I get this error

```
# friedman_kbs <- green_kbs %>% friedman_test(spp_half_cover_date ~ state | plot)
```

Error in friedman.test.default(c(141L, 202L, 122L, 101L, 127L, 120L, 197L, : not an unreplicated complete block design

Permanova?

```
per1 <- adonis2(green_kbs$spp_half_cover_date ~ state * year + insecticide, data = green_kbs)
per1
per2 <- adonis(formula = green_kbs$spp_half_cover_date ~ state * year + insecticide,
    strata = green_kbs$plot, data = green_kbs)
per2
```

With per2, when controlling for "plot", there is a difference btwn treatments

UMBS

Checking for normality

```
hist(green_umbs$spp_half_cover_date)
qqnorm(green_umbs$spp_half_cover_date)
shapiro.test(green_umbs$spp_half_cover_date)
hist(green_umbs$spp_half_cover_date[green_kbs$state == "ambient"])
hist(green_umbs$spp_half_cover_date[green_kbs$state == "warmed"])
```

These look pretty good

Trying log transformation

```
green_umbs$date_log <- log(green_umbs$spp_half_cover_date)
hist(green_umbs$date_log)
shapiro.test(green_umbs$date_log)
```

I think this looks good but shapiro-wilk is lower than 0.05

Trying inverse tranformation

```
green_umbs$date_inv <- 1/(green_umbs$spp_half_cover_date)
hist(green_umbs$date_inv)
shapiro.test(green_umbs$date_inv)
```

This also looks good but is also still low for shapiro-wilk