512 Project Part I

Jared Adam

Due Sept 27

**The number of conflicts with dplyr from all of the packages we must download is becoming annoying and breaking code.**

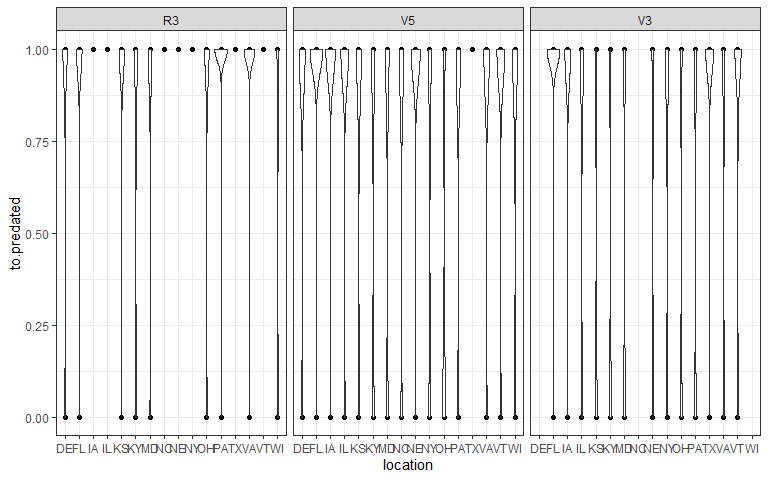
# Part I (512 only, project proposal, 25 pts):

1. Read in your data set and run dim on it:

set.seed(654321)  
s21 <- read\_csv('data/2021 Sentinel Prey Assessment.csv')  
s22 <- read\_csv("data/PSA\_CE2\_SentinelPrey.csv")  
s23 <- read\_csv('data/PSA\_Sent.prey.2023.csv')  
  
# I need to get total predation into a column as a binary. 1 = predation, 0 = not  
  
# 2021 cleaning   
s21

## # A tibble: 5,281 × 17  
## location date n.weather d.weather growth\_stage plot\_id rep.block treatment  
## <chr> <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 PA 6/16/2… 15.3 18.44 V3 101 1 1  
## 2 PA 6/16/2… 15.3 18.44 V3 101 1 1  
## 3 PA 6/16/2… 15.3 18.44 V3 101 1 1  
## 4 PA 6/16/2… 15.3 18.44 V3 101 1 1  
## 5 PA 6/16/2… 15.3 18.44 V3 101 1 1  
## 6 PA 6/16/2… 15.3 18.44 V3 101 1 1  
## 7 PA 6/16/2… 15.3 18.44 V3 102 1 3  
## 8 PA 6/16/2… 15.3 18.44 V3 102 1 3  
## 9 PA 6/16/2… 15.3 18.44 V3 102 1 3  
## 10 PA 6/16/2… 15.3 18.44 V3 102 1 3  
## # ℹ 5,271 more rows  
## # ℹ 9 more variables: row <dbl>, sample <dbl>, n.absent <chr>, n.partial <chr>,  
## # n.predated <chr>, d.absent <chr>, d.partial <chr>, d.predated <chr>,  
## # to.predated <chr>

clean21 <- s21 %>%   
 mutate(year = '2021') %>%   
 dplyr::select(location, year, growth\_stage, plot\_id, rep.block, treatment, to.predated) %>%   
 mutate(to.predated = as.double(to.predated)) %>%   
 dplyr::rename(block = rep.block) %>%   
 group\_by(location, year, growth\_stage, plot\_id, block, treatment) %>%   
 # dplyr::summarise(total = sum(to.predated)) %>%   
 na.omit() %>%   
 mutate(treatment = case\_when(  
 treatment == '33' ~ '3',  
 .default = as.factor(treatment))) %>%   
 dplyr::filter(treatment != '6',  
 treatment != '7',  
 treatment != '8') %>%   
 mutate\_at(vars(1:6), as.factor)   
  
ggplot(clean21, aes(x = location, y = to.predated))+  
 geom\_point()+  
 geom\_violin()+  
 facet\_wrap(~growth\_stage)



# 2022 cleaning   
s22

## # A tibble: 3,246 × 19  
## location date am.weather pm.weather growth\_stage plotid block treatment  
## <chr> <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 PA 6/22/2022 22.7 26.1 V3 101 1 1  
## 2 PA 6/22/2022 22.7 26.1 V3 101 1 1  
## 3 PA 6/22/2022 22.7 26.1 V3 101 1 1  
## 4 PA 6/22/2022 22.7 26.1 V3 101 1 1  
## 5 PA 6/22/2022 22.7 26.1 V3 101 1 1  
## 6 PA 6/22/2022 22.7 26.1 V3 101 1 1  
## 7 PA 6/22/2022 22.7 26.1 V3 102 1 3  
## 8 PA 6/22/2022 22.7 26.1 V3 102 1 3  
## 9 PA 6/22/2022 22.7 26.1 V3 102 1 3  
## 10 PA 6/22/2022 22.7 26.1 V3 102 1 3  
## # ℹ 3,236 more rows  
## # ℹ 11 more variables: row <dbl>, sample <dbl>, am.absent <chr>,  
## # am.partial <chr>, am.predators <chr>, pm.absent <chr>, pm.partial <chr>,  
## # pm.predators <chr>, to.predated <dbl>, n.predated <dbl>, d.predated <dbl>

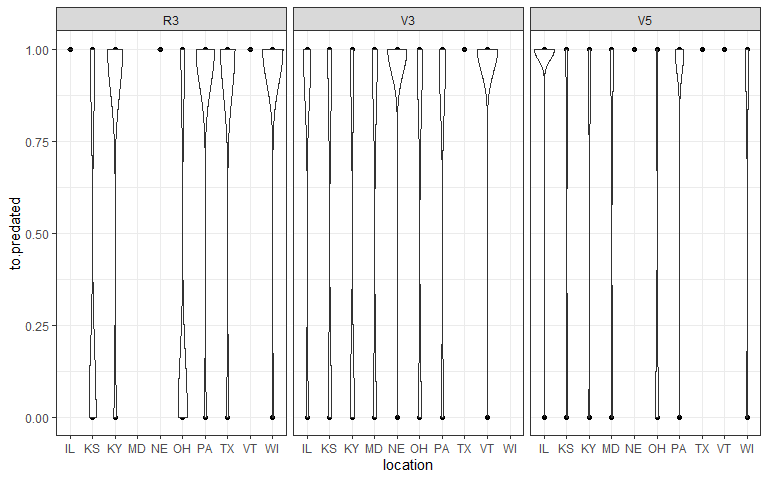
unique(s22$treatment)

## [1] 1 3 2 4

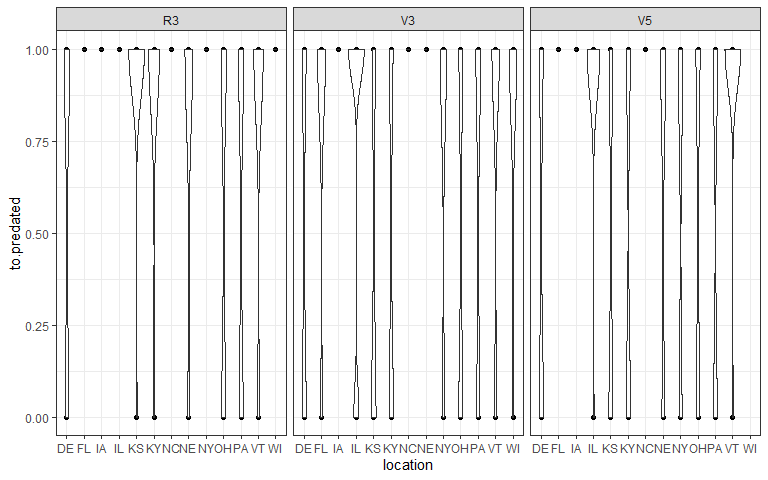
unique(s22$growth\_stage)

## [1] "V3" "V5" "R3" "R2"

clean22 <- s22 %>%   
 mutate(year = '2022') %>%   
 dplyr::select(location, year, growth\_stage, plotid, block, treatment, to.predated) %>%   
 dplyr::rename(plot\_id = plotid) %>%   
 mutate(growth\_stage = case\_when(growth\_stage == 'R2' ~ 'R3',  
 .default = as.character(growth\_stage))) %>%   
 dplyr::group\_by(location, year, growth\_stage, plot\_id, block, treatment) %>%   
 # dplyr::summarise(total = sum(to.predated)) %>%   
 mutate\_at(vars(1:6), as.factor)  
  
ggplot(clean22, aes(x = location, y = to.predated))+  
 geom\_point()+  
 geom\_violin()+  
 facet\_wrap(~growth\_stage)



# 2023 cleaning   
  
clean23 <- s23 %>%  
 mutate(year = '2023') %>%  
 relocate(am.partial, am.absent, pm.partial, pm.absent) %>%   
 mutate\_at(vars(1:4), as.double) %>%   
 mutate(to.predated = if\_else(am.partial | am.absent | pm.partial | pm.absent == 1, 1, 0)) %>%   
 relocate(to.predated)%>%   
 mutate(growth\_stage = case\_when((location == 'NC' & date == '7/20/2023') ~ 'R3',  
 .default = as.character(growth\_stage))) %>%   
 dplyr::select(location, year, growth\_stage, plotid, block, treatmetn, to.predated) %>%   
 dplyr::rename(plot\_id = plotid,   
 treatment = treatmetn) %>%  
 distinct() %>%   
 group\_by(location, year, growth\_stage, plot\_id, block, treatment) %>%   
 na.omit() %>%   
 filter(treatment != 5) %>%   
 mutate\_at(vars(1:6),as.factor)  
  
ggplot(clean23, aes(x = location, y = to.predated))+  
 geom\_point()+  
 geom\_violin()+  
 facet\_wrap(~growth\_stage)



# and in the darkness, bind them   
sent <- rbind(clean21, clean22, clean23)  
as\_tibble(sent)

## # A tibble: 9,227 × 7  
## location year growth\_stage plot\_id block treatment to.predated  
## <fct> <fct> <fct> <fct> <fct> <fct> <dbl>  
## 1 PA 2021 V3 101 1 1 0  
## 2 PA 2021 V3 101 1 1 0  
## 3 PA 2021 V3 101 1 1 0  
## 4 PA 2021 V3 101 1 1 0  
## 5 PA 2021 V3 101 1 1 0  
## 6 PA 2021 V3 101 1 1 0  
## 7 PA 2021 V3 102 1 3 1  
## 8 PA 2021 V3 102 1 3 1  
## 9 PA 2021 V3 102 1 3 0  
## 10 PA 2021 V3 102 1 3 1  
## # ℹ 9,217 more rows

dim(sent)

## [1] 9227 7

1. Prepare a short description of your data set (source if published paper exists), especially providing the study design, sample size, and variables of primary interest. If there is random sampling, note the population sampled from. If there is random assignment, note how and for which variable(s).

Data: Sentinel Prey assessment of arthropod-predator activity in corn fields.

These data come from the Precision Sustainable Agriculture effort through the USDA. I am the lead on the entomology component of this project and responsible for analyzing this three year data set which spans multiple states. This effort began during my Master’s degree, but I only analyzed Pennsylvania data for my thesis.

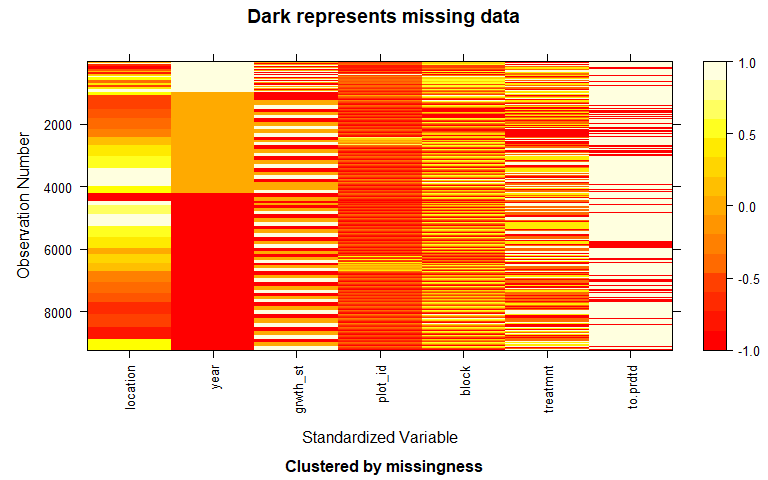
**Study design:**  **Treatments** = 4; No cover crop, early-terminated cover crop, late-terminated cover crop, planting green **Plots** = 20; 5 blocks composed of 4 plots each = 20 plots / study site / year **Years** = 3 (2021,2022,2023) **Locations** = This project comprises 16 states. Not all states collected sentinel prey data every year. **Effort** = Data were collected at three corn growth stages / year (V3,V5,R3). **Sample** = 6 sentinel prey traps were placed in each plot = 120 samples collected / growth stage. Total sample effort per state per season = 360 samples.

**Variables:** **Response** = Total level of predation. This is transformed into a proportion over a constant total. Binomial response **Explanatory** = Crop growth stage (timing, three levels) and treatment (four levels). **Random effects** = Block and location? I am unsure how to use location. I am not interested in seeing how each state differs in the analysis because that is *not* a research question. I plan to go through and conduct each state’s own analysis later. **Repeated measure** = I visit the same trap three times throughout the year. This feels like a term I should identify. If I were to choose, I would select *growth stage*.

Plots were randomly assigned to each block. Field sites were as random as they could be at each respective research station. Sentinel prey traps were placed between pre-determined rows and at specific length intervals within each plot to maintain consistency.

1. Make a missing\_data.frame plot of your data set and explain any missing values indicated:

library(mi)  
# make an object of the missing df and then present the image  
tdf <- missing\_data.frame(data.frame(sent))  
image(tdf)



1. Discuss any other use in classes or theses for the data - either that you have used it for or are currently working on for future submissions.

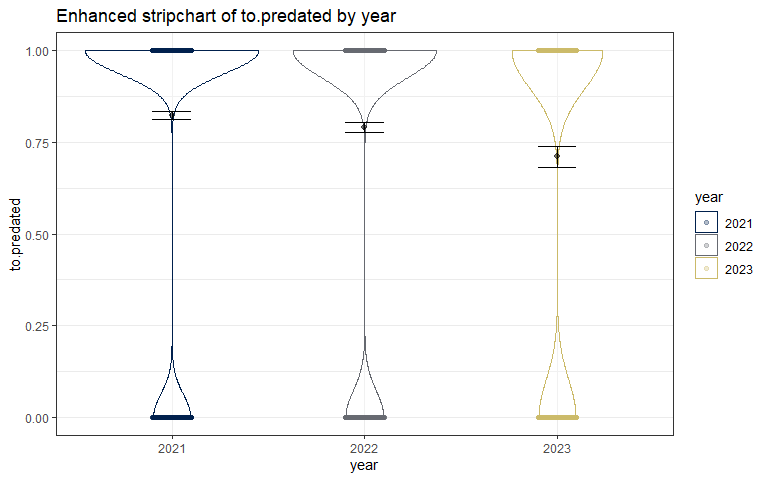
**I am working on this for a publication. The Pennsylvania-exclusive data were analyzed for me master’s thesis. The larger, whole-project data set is for publication.**

1. Provide at least one display of the data, focusing on the response of interest versus a predictor. If you have multiple predictors, try to plot the response versus those too.

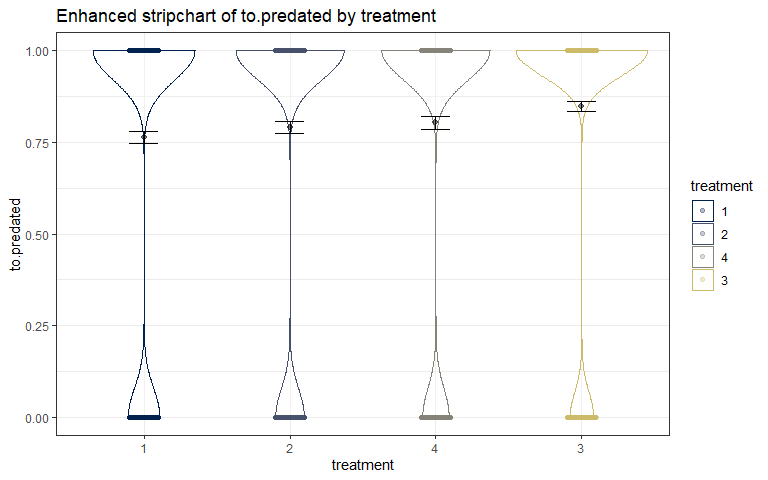
sent

## # A tibble: 9,227 × 7  
## # Groups: location, year, growth\_stage, plot\_id, block, treatment [2,091]  
## location year growth\_stage plot\_id block treatment to.predated  
## <fct> <fct> <fct> <fct> <fct> <fct> <dbl>  
## 1 PA 2021 V3 101 1 1 0  
## 2 PA 2021 V3 101 1 1 0  
## 3 PA 2021 V3 101 1 1 0  
## 4 PA 2021 V3 101 1 1 0  
## 5 PA 2021 V3 101 1 1 0  
## 6 PA 2021 V3 101 1 1 0  
## 7 PA 2021 V3 102 1 3 1  
## 8 PA 2021 V3 102 1 3 1  
## 9 PA 2021 V3 102 1 3 0  
## 10 PA 2021 V3 102 1 3 1  
## # ℹ 9,217 more rows

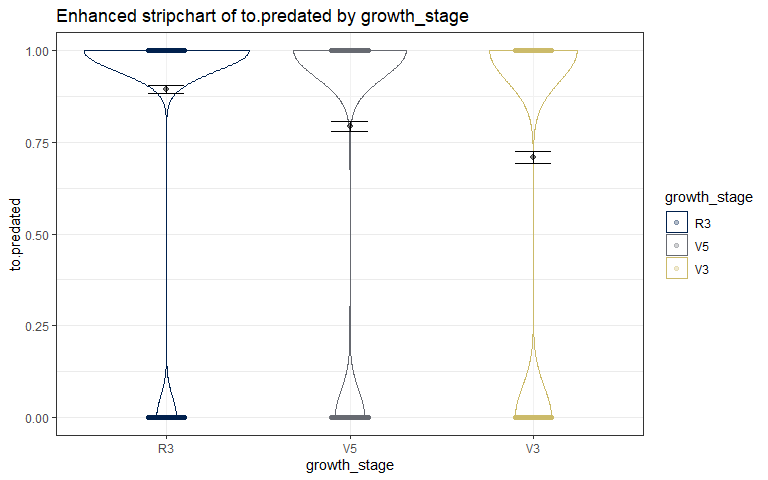
# I do not this plotting function for binary data  
enhanced\_stripchart(to.predated ~ year, data = sent)



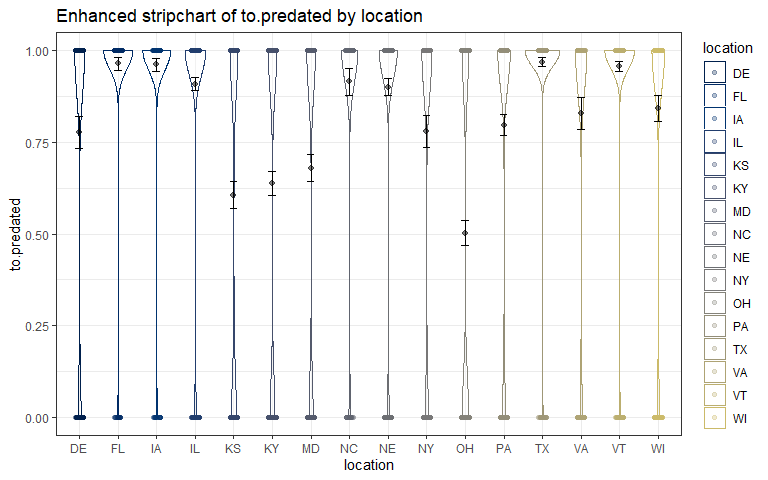
enhanced\_stripchart(to.predated ~ treatment, data = sent) # 3 = planting green



enhanced\_stripchart(to.predated ~ growth\_stage, data = sent) # expect higher predation later in the season



enhanced\_stripchart(to.predated ~ location, data = sent)



1. Provide an initial model you hope to fit (does not need to be fit). If you fit a model, add a model summary and effects plot.

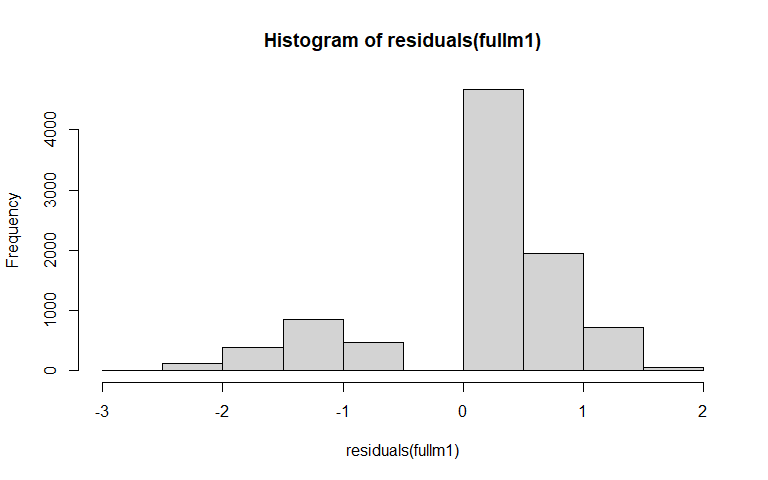
sent

## # A tibble: 9,227 × 7  
## # Groups: location, year, growth\_stage, plot\_id, block, treatment [2,091]  
## location year growth\_stage plot\_id block treatment to.predated  
## <fct> <fct> <fct> <fct> <fct> <fct> <dbl>  
## 1 PA 2021 V3 101 1 1 0  
## 2 PA 2021 V3 101 1 1 0  
## 3 PA 2021 V3 101 1 1 0  
## 4 PA 2021 V3 101 1 1 0  
## 5 PA 2021 V3 101 1 1 0  
## 6 PA 2021 V3 101 1 1 0  
## 7 PA 2021 V3 102 1 3 1  
## 8 PA 2021 V3 102 1 3 1  
## 9 PA 2021 V3 102 1 3 0  
## 10 PA 2021 V3 102 1 3 1  
## # ℹ 9,217 more rows

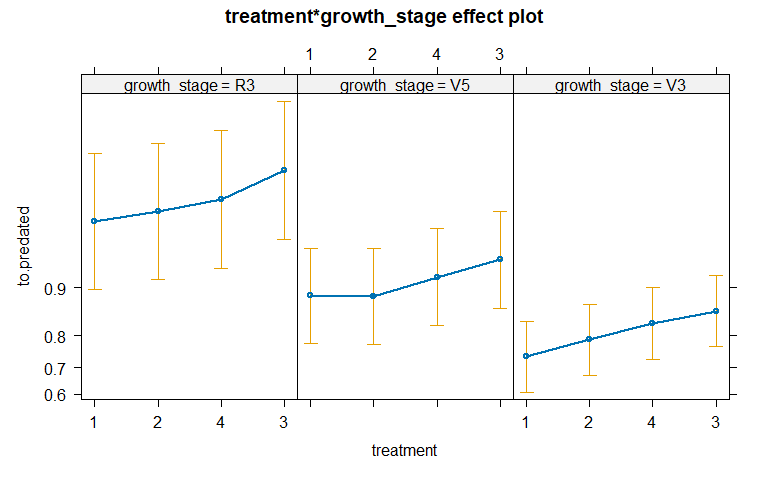
nullm1 <- glmer(to.predated ~ (growth\_stage|year/location/block/plot\_id), family = binomial, data = sent)  
  
  
fullm1 <- glmer(to.predated ~ treatment\*growth\_stage + (growth\_stage|year/location/block/plot\_id), family = binomial, data = sent)   
summary(fullm1)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula:   
## to.predated ~ treatment \* growth\_stage + (growth\_stage | year/location/block/plot\_id)  
## Data: sent  
##   
## AIC BIC logLik deviance df.resid   
## 6811.8 7068.5 -3369.9 6739.8 9191   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -6.6897 0.0490 0.1914 0.4187 2.3997   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## plot\_id:(block:(location:year)) (Intercept) 0.27964 0.5288   
## growth\_stageV5 0.36146 0.6012 -0.72   
## growth\_stageV3 0.53538 0.7317 -0.71 1.00  
## block:(location:year) (Intercept) 0.10357 0.3218   
## growth\_stageV5 0.02733 0.1653 0.99   
## growth\_stageV3 0.02714 0.1647 -0.63 -0.73  
## location:year (Intercept) 6.10710 2.4713   
## growth\_stageV5 2.88527 1.6986 -0.61   
## growth\_stageV3 4.69651 2.1671 -0.81 0.82  
## year (Intercept) 0.31795 0.5639   
## growth\_stageV5 0.80014 0.8945 -1.00   
## growth\_stageV3 0.61540 0.7845 -1.00 1.00  
## Number of obs: 9227, groups:   
## plot\_id:(block:(location:year)), 744; block:(location:year), 191; location:year, 39; year, 3  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 3.30346 0.58113 5.685 1.31e-08  
## treatment2 0.17282 0.20454 0.845 0.398164  
## treatment4 0.36630 0.22329 1.640 0.100916  
## treatment3 0.86350 0.22294 3.873 0.000107  
## growth\_stageV5 -1.24603 0.66472 -1.875 0.060861  
## growth\_stageV3 -2.27113 0.64784 -3.506 0.000455  
## treatment2:growth\_stageV5 -0.18140 0.25131 -0.722 0.470411  
## treatment4:growth\_stageV5 -0.05476 0.27467 -0.199 0.841964  
## treatment3:growth\_stageV5 -0.25759 0.27106 -0.950 0.341952  
## treatment2:growth\_stageV3 0.11873 0.25261 0.470 0.638340  
## treatment4:growth\_stageV3 0.19620 0.27264 0.720 0.471767  
## treatment3:growth\_stageV3 -0.09241 0.27018 -0.342 0.732318  
##   
## Correlation of Fixed Effects:  
## (Intr) trtmn2 trtmn4 trtmn3 grw\_V5 grw\_V3 t2:\_V5 t4:\_V5 t3:\_V5  
## treatment2 -0.165   
## treatment4 -0.143 0.443   
## treatment3 -0.139 0.449 0.413   
## grwth\_stgV5 -0.794 0.142 0.123 0.119   
## grwth\_stgV3 -0.886 0.148 0.127 0.124 0.904   
## trtmnt2:\_V5 0.135 -0.799 -0.355 -0.358 -0.183 -0.130   
## trtmnt4:\_V5 0.116 -0.354 -0.798 -0.329 -0.158 -0.111 0.447   
## trtmnt3:\_V5 0.114 -0.362 -0.335 -0.809 -0.157 -0.111 0.456 0.422   
## trtmnt2:\_V3 0.133 -0.809 -0.358 -0.364 -0.124 -0.182 0.694 0.307 0.315  
## trtmnt4:\_V3 0.116 -0.362 -0.818 -0.337 -0.108 -0.159 0.312 0.701 0.293  
## trtmnt3:\_V3 0.114 -0.368 -0.341 -0.821 -0.106 -0.157 0.315 0.292 0.707  
## t2:\_V3 t4:\_V3  
## treatment2   
## treatment4   
## treatment3   
## grwth\_stgV5   
## grwth\_stgV3   
## trtmnt2:\_V5   
## trtmnt4:\_V5   
## trtmnt3:\_V5   
## trtmnt2:\_V3   
## trtmnt4:\_V3 0.449   
## trtmnt3:\_V3 0.456 0.428  
## optimizer (Nelder\_Mead) convergence code: 4 (failure to converge in 10000 evaluations)  
## Model failed to converge with max|grad| = 0.139343 (tol = 0.002, component 1)  
## failure to converge in 10000 evaluations

hist(residuals(fullm1))



plot(allEffects(fullm1))



#likelihood test  
anova(nullm1, fullm1)

## Data: sent  
## Models:  
## nullm1: to.predated ~ (growth\_stage | year/location/block/plot\_id)  
## fullm1: to.predated ~ treatment \* growth\_stage + (growth\_stage | year/location/block/plot\_id)  
## npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)  
## nullm1 25 6854.2 7032.4 -3402.1 6804.2   
## fullm1 36 6811.8 7068.5 -3369.9 6739.8 64.35 11 1.428e-09

1. Start to work on a Table 1 that summarizes variables of interest, possibly by groups of interest. At a minimum, summarize the response variable, by a grouping variable if one exists.

# table as a proportion   
all\_locs <- sent %>%   
 group\_by(location, treatment, growth\_stage) %>%   
 dplyr::summarise(prop = mean(to.predated),  
 sd = sd(to.predated),  
 n = n(),   
 se = sd/sqrt(n))  
  
by\_trt <- sent %>%   
 group\_by(treatment) %>%   
 dplyr::summarise(prop = mean(to.predated),  
 sd = sd(to.predated),  
 n = n(),   
 se = sd/sqrt(n))  
  
by\_growth <- sent %>%   
 group\_by(growth\_stage) %>%   
 dplyr::summarise(prop = mean(to.predated),  
 sd = sd(to.predated),  
 n = n(),   
 se = sd/sqrt(n))

1. Provide the names of feedback group members and the date, time, and location of your feedback session interaction.

9/25/2024 830 am. Eme Morgan, Kaelin Smith, Rennie Winkelman in the stats class room.

Graded for completion/not but there are points for participation in a feedback session. Note that 412 students get full credit for this.