Primula Clipping Experiment

JL

2/13/2023

To do: - 6/26 update: Met with Jenn, we came up with more ideas: - let’s take out the vegetative plants, bc we think theyre doing weird things - there are more interactions that I didnt try before - we also think that the stepwise selection is wrong, theyre not nested models so it wont work - change clipping trt to ln(seeds) and log(fruits produced) - a better measure of reproductive effort! We will have to treat clipping and abortion the same way, which is maybe questionable… - ^we can compare which one is better - Throw in soil moisture in case that fits better - daily average soil moisutre April-end of May

#read in data, make df's  
# Add in demography data:  
source("../scripts/dode\_allyears\_cleaned.R")

## Warning: Values from `psurvival` are not uniquely identified; output will contain  
## list-cols.  
## • Use `values\_fn = list` to suppress this warning.  
## • Use `values\_fn = {summary\_fun}` to summarise duplicates.  
## • Use the following dplyr code to identify duplicates.  
## {data} %>%  
## dplyr::group\_by(tag, year) %>%  
## dplyr::summarise(n = dplyr::n(), .groups = "drop") %>%  
## dplyr::filter(n > 1L)

## Warning in left\_join(Dodecatheon, dtable, by = c("tag", "year")): Each row in `x` is expected to match at most 1 row in `y`.  
## ℹ Row 78 of `x` matches multiple rows.  
## ℹ If multiple matches are expected, set `multiple = "all"` to silence this  
## warning.

#add in tags in clipping experiment  
clip <- read.csv("../data/2021\_Primula\_Clipping.csv")  
clip <- clip %>%   
 select(1:8, -c(coor, date.of.clipping)) %>%   
 mutate(tag = as.character(tag),  
 plot = as.character(plot),  
 initial.flower = no.flowers.at.start +no.buds.at.start)  
  
#adding in soil moisture data  
sm <- read.csv("C:/Users/Jenna/Dropbox/Williams' Lab/Students/Jenna/loggerdata/cleaned.data/SM\_AprilMayavg\_allyears.csv")  
sm <- sm %>% mutate(plot = as.character(plot),  
 year = as.character(year))  
  
#add in number of seeds made  
seeds21 <- read.csv("C:/Users/Jenna/Dropbox/Williams' Lab/Cowichan IDE/Cowichan\_DemographyData/Dodecatheon/2021\_Dodecatheon\_Seed\_Counts.csv")  
  
#In 2021, which flowers made it to being a capsule?  
#seeds21 <- subset(seeds21, tag != "1556") # not an experiment plant  
#seeds21 <- subset(seeds21, tag != 1998) # not an experiment plant  
#seeds21 <- subset(seeds21, tag != 1813)  
seeds21 <- seeds21 %>%   
 mutate(pseeds = if\_else(no.seeds > 0, 1, 0),  
 tag = as.character(tag),  
 plot = as.character(plot))  
seeds21$pseeds[is.na(seeds21$pseeds)] <- 1 #assuming the plants where we couldn't count capsules separately had all their capsules contribute to seed total  
  
  
cap <- seeds21 %>% group\_by(tag, plot) %>%   
 dplyr::summarize(no.success.cap = sum(pseeds),  
 total.seeds = sum(no.seeds))

## `summarise()` has grouped output by 'tag'. You can override using the `.groups`  
## argument.

cap$total.seeds[cap$tag == 1556] <- 29  
cap$total.seeds[cap$tag == 1988] <- 58  
cap$total.seeds[cap$tag == 1988] <- 58  
cap$total.seeds[cap$tag == 282] <- 50  
cap$total.seeds[cap$tag == 494] <-98  
  
#adding cap to the clip df  
  
clip <- full\_join(clip, cap) %>%   
 mutate(total.seeds = if\_else(no.success.cap == 0, 0, total.seeds))

## Joining with `by = join\_by(plot, tag)`

remove(cap)  
clip$no.success.cap[is.na(clip$no.success.cap)] <- 0  
clip$initial.flower[clip$tag == 1490] <- 6  
clip$initial.flower[clip$tag == 5769] <- 5  
clip$initial.flower[clip$tag == 962] <- 6  
clip$initial.flower[clip$tag == 5746] <- 5  
clip$no.success.cap[clip$tag == 540] <- 4  
clip$no.success.cap[clip$tag == 693] <- 2  
  
  
  
############################################################################  
#add in clipping experiment treatments  
  
clip <- left\_join(clip, Dodecatheon, by = c("tag", "plot")) %>%   
 filter(year == 2021) %>%   
 mutate(fruitset = no.success.cap/initial.flower) %>%   
 mutate(pflower = as.factor(pflower),  
 rgr = log.ros.areaT1-log.ros.area) #calc relative growth rate

## Warning in left\_join(clip, Dodecatheon, by = c("tag", "plot")): Each row in `x` is expected to match at most 1 row in `y`.  
## ℹ Row 1 of `x` matches multiple rows.  
## ℹ If multiple matches are expected, set `multiple = "all"` to silence this  
## warning.

# these are all of the tags in the clipping experiment for 2021 and 2022  
#add soil moisture:  
clip <- left\_join(clip, sm)

## Joining with `by = join\_by(plot, year, trt)`

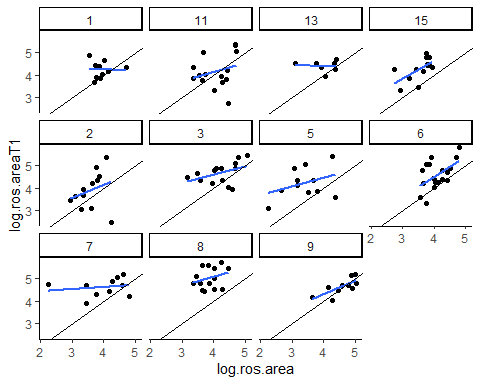
#subsetting - get rid of 12 and 14 (only 3 observations per plot - maybe this will help?)  
clip <- clip %>% filter(plot != "12") %>%   
 filter(plot != "14")  
  
#this is code to get the vegetative plants back in - unchecked for now  
# ###  
# # what sizes are plants in 2021 to 2022  
# Dodecatheon.21 <- Dodecatheon %>%   
# filter(year == "2021")  
# #of the plants that flowered in 2021, what was their min size? it doesnt look like I need to set a max size  
# ggplot(Dodecatheon.21, aes(log.ros.area, log.ros.areaT1, color = pflower)) +  
# geom\_jitter()+  
# facet\_wrap(~trt)  
# ggplot(subset(Dodecatheon.21, !is.na(pflower)), aes(log.ros.area))+  
# geom\_histogram()+  
# facet\_wrap(pflower~trt)  
# min <- Dodecatheon.21 %>%   
# group\_by(pflower, trt) %>%   
# summarize(min.size = min(log.ros.area, na.rm = TRUE))  
# clip2 <- Dodecatheon.21 %>%   
# filter(log.ros.area > 2) %>%   
# mutate(pflower = as.factor(pflower))  
# join <- left\_join(clip2, clip)  
# join <- join %>%   
# #eft\_join(join,sm, by = c("trt", "plot", "year")) %>%   
# mutate(treatment = if\_else(is.na(treatment) & pflower == 0, "veg", if\_else(is.na(treatment) & pflower == 1, "flowernt", treatment), treatment),  
# rgr = log.ros.areaT1-log.ros.area)  
#   
# remove(sm, clip2, Dodecatheon, min, Dodecatheon.21)  
#   
# #non experiment plants that flowered:  
# no.exp <- read.csv("/Users/Jenna/Dropbox/Williams' Lab/Cowichan IDE/Cowichan\_DemographyData/Dodecatheon/2021\_Dodecatheon\_Resurveys.csv", header=T)  
# no.exp <- no.exp %>%   
# filter(!is.na(pseeds)) %>%   
# dplyr::rename(fruitset1 = pseeds) %>%   
# select(c(plot, tag, fruitset1)) %>%   
# mutate(plot = as.character(plot),  
# tag = as.character(tag))  
  
  
#making a decision here that I may need to change: setting pseeds = fruitset (ie assuming that for these plants, if you made seeds, all of your flowers made seeds) - I have the data to check this, but will go check this later  
  
clip <- clip %>%   
 mutate(log.cap = log(no.success.cap + 1), #because there are zeros   
 total.seeds = if\_else(fruitset == 0, 0, total.seeds),  
 pseeds = if\_else(total.seeds > 0, 1, 0),  
 log.total.seeds = log(total.seeds + 1), #because there are zeros   
 psurvivalT1 = if\_else(is.na(psurvivalT1), 0, psurvivalT1))  
clip <- clip %>%  
 select(-c(no.flowers.at.start, no.buds.at.start, tag.pulled, problem.tag, toothpick, lv.length))  
  
  
#clip.sub filtered out the plots with very few observations

ggplot(clip, aes(x = log.ros.area, y = log.ros.areaT1))+  
 geom\_point()+  
 geom\_smooth(method = "lm", se = F)+  
 geom\_abline(slope=1, intercept=0)+  
 facet\_wrap(~plot)

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: Removed 10 rows containing non-finite values (`stat\_smooth()`).

## Warning: Removed 10 rows containing missing values (`geom\_point()`).

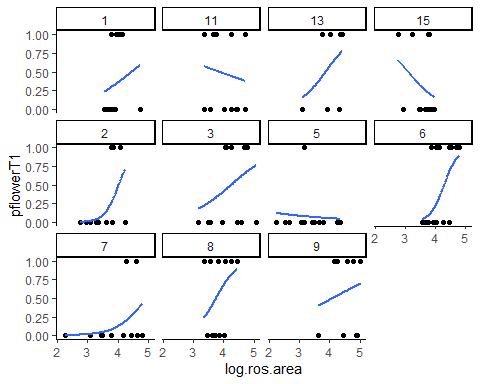


ggplot(clip, aes(x = log.ros.area, y = pflowerT1))+  
 geom\_point()+  
 geom\_smooth(method = "glm", method.args = list(family = binomial), se = F)+   
 geom\_abline(slope=1, intercept=0)+  
 facet\_wrap(~plot)

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (`stat\_smooth()`).

## Warning: Removed 2 rows containing missing values (`geom\_point()`).



#These are the vital rate models we want to build:

growth.01 <- lmer(log.ros.areaT1 ~ log.ros.area +(1|plot), data = clip, REML = F)  
growth.02 <- lmer(log.ros.areaT1 ~ log.ros.area\*trt +(1|plot), data = clip, REML= F)  
growth.03 <- lmer(log.ros.areaT1 ~ log.ros.area + trt + log.total.seeds + (1|plot), data = clip, REML= F)  
growth.04 <- lmer(log.ros.areaT1 ~ log.ros.area + trt + log.total.seeds + trt:log.total.seeds + log.ros.area:trt + (1|plot), data = clip, REML= F)  
growth.05 <- lmer(log.ros.areaT1 ~ log.ros.area + log.total.seeds + trt + log.total.seeds:trt + log.ros.area:log.total.seeds + log.ros.area:trt + log.ros.area:log.total.seeds:trt + (1|plot), data = clip, REML= F)  
  
growth.06 <- lmer(log.ros.areaT1 ~ log.ros.area + trt + log.cap + (1|plot), data = clip, REML= F)  
growth.07 <- lmer(log.ros.areaT1 ~ log.ros.area + trt + log.cap + trt:log.cap + log.ros.area:trt + (1|plot), data = clip, REML= F)  
growth.08 <- lmer(log.ros.areaT1 ~ log.ros.area + log.cap + trt + log.cap:trt + log.ros.area:log.cap + log.ros.area:trt + log.ros.area:log.total.seeds:trt + (1|plot), data = clip, REML= F)  
  
#add in SM data for fun  
growth.09 <- lmer(log.ros.areaT1 ~ log.ros.area\*sm\_avg +(1|plot), data = clip, REML= F)  
growth.10 <- lmer(log.ros.areaT1 ~ log.ros.area + sm\_avg + log.total.seeds + (1|plot), data = clip, REML= F)  
growth.11 <- lmer(log.ros.areaT1 ~ log.ros.area + sm\_avg + log.total.seeds + sm\_avg:log.total.seeds + log.ros.area:sm\_avg + (1|plot), data = clip, REML= F)  
growth.12 <- lmer(log.ros.areaT1 ~ log.ros.area + log.total.seeds + sm\_avg + log.total.seeds:sm\_avg + log.ros.area:log.total.seeds + log.ros.area:sm\_avg + log.ros.area:log.total.seeds:sm\_avg + (1|plot), data = clip, REML= F)  
  
growth.13 <- lmer(log.ros.areaT1 ~ log.ros.area + sm\_avg + log.cap + (1|plot), data = clip, REML= F)  
growth.14 <- lmer(log.ros.areaT1 ~ log.ros.area + sm\_avg + log.cap + sm\_avg:log.cap + log.ros.area:sm\_avg + (1|plot), data = clip, REML= F)  
growth.15 <- lmer(log.ros.areaT1 ~ log.ros.area + log.cap + sm\_avg + log.cap:sm\_avg + log.ros.area:log.cap + log.ros.area:sm\_avg + log.ros.area:log.total.seeds:sm\_avg + (1|plot), data = clip, REML= F)  
growth.16 <- lmer(log.ros.areaT1 ~ log.ros.area + log.total.seeds + (1|plot), data = clip, REML= F)  
growth.17 <- lmer(log.ros.areaT1 ~ log.ros.area\*log.total.seeds + (1|plot), data = clip, REML= F)  
growth.18 <- lmer(log.ros.areaT1 ~ log.ros.area + log.cap + (1|plot), data = clip, REML= F)  
growth.19 <- lmer(log.ros.areaT1 ~ log.ros.area\*log.cap + (1|plot), data = clip, REML= F)  
  
model.names = c("log(size)",   
 "log(size)\*IDE",   
 "log(size) + IDE + log(total seeds)",   
 "log(size) + IDE + log(total seeds) + IDE:log(total seeds) + log(size):IDE",  
 "log(size) + log(total seeds) + IDE + log(total seeds):IDE + log(size):log(total seeds) + log(size):IDE + log(size):log(total seeds):IDE",  
 "log(size) + IDE + log(total capsules)",  
 "log(size) + IDE + log(total capsules) + IDE:log(total capsules) + log(size):IDE",  
 "log(size) + log(total capsules) + IDE + log(total capsules):IDE + log(size):log(total capsules) + log(size):IDE + log(size):log(total seeds):IDE",  
 "log(size)\*soil moisture",  
 "log(size) + soil moisture + log(total seeds)",  
 "log(size) + soil moisture + log(total seeds) + soil moisture:log(total seeds) + log(size):soil moisture",  
 "log(size) + log(total seeds) + soil moisture + log(total seeds):soil moisture + log(size):log(total seeds) + log(size):soil moisture + log(size):log(total seeds):soil moisture",  
 "log(size) + soil moisture + log(total capsules)",  
 "log(size) + soil moisture + log(total capsules) + soil moisture:log(total capsules) + log(size):soil moisture",  
 "log(size) + log(total capsules) + soil moisture + log(total capsules):soil moisture + log(size):log(total capsules) + log(size):soil moisture + log(size):log(total seeds):soil moisture",  
 "log(size) + log(total seeds)",  
 "log(size)\*log(total seeds)",  
 "log(size) + log(total capsules)",  
 "log(size)\*log(total capsules)")  
  
gr.mods <- mget(ls(pattern = "growth")) #make a list of all of those models that start with gm (^ means start with)  
gr\_AICc<-aictab(cand.set = gr.mods, modnames = model.names,second.ord=TRUE,nobs=NULL,sort=TRUE)  
#test<-aictab(cand.set = gr.mods, modnames = NULL,second.ord=TRUE,nobs=NULL,sort=TRUE)  
  
  
summary(growth.16)

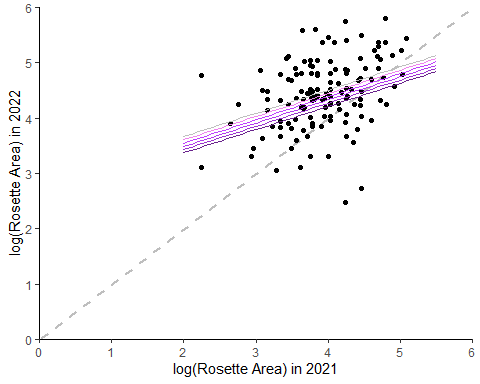
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's  
## method [lmerModLmerTest]  
## Formula: log.ros.areaT1 ~ log.ros.area + log.total.seeds + (1 | plot)  
## Data: clip  
##   
## AIC BIC logLik deviance df.resid   
## 224.1 238.4 -107.0 214.1 126   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.3182 -0.5101 -0.0115 0.5900 2.4006   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.06635 0.2576   
## Residual 0.26775 0.5174   
## Number of obs: 131, groups: plot, 11  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 2.82746 0.38069 124.49497 7.427 1.54e-11 \*\*\*  
## log.ros.area 0.41897 0.09528 130.97842 4.397 2.25e-05 \*\*\*  
## log.total.seeds -0.05886 0.03098 128.77374 -1.900 0.0597 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) lg.rs.  
## log.ros.are -0.968   
## log.ttl.sds 0.102 -0.191

#rm(list = ls()[grepl("growth", ls())]) #removes these models

#create df with all possible x values  
#this is for model without an interaction  
pred.dat <- expand.grid(log.ros.area = seq(2, 5.5, by = .1), log.total.seeds = seq(0, 5.3, by = .1))  
pred.output <- predict(growth.16, newdata = pred.dat, type="response", re.form=~0) #re.form = ~0 tells it to not include random effects  
pred.output <- as.data.frame(pred.output)  
pred.dat.output <- cbind(pred.dat, pred.output) %>%   
 rename(log.ros.areaT1 = pred.output)  
  
#graph it  
ggplot(clip, aes(x = log.ros.area, y = log.ros.areaT1))+  
 geom\_abline(slope=1, intercept = 0, linetype = 2, color= "grey", size = 1)+  
 geom\_point()+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 5), aes(x = log.ros.area, y = log.ros.areaT1), color = "darkorchid4")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 4), aes(x = log.ros.area, y = log.ros.areaT1), color = "darkorchid3")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 3), aes(x = log.ros.area, y = log.ros.areaT1), color = "darkorchid2")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 2), aes(x = log.ros.area, y = log.ros.areaT1), color = "darkorchid1")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 1), aes(x = log.ros.area, y = log.ros.areaT1), color = "violet")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), color = "grey")+  
 scale\_y\_continuous(expand = c(0, 0), limits = c(0, 6))+  
 scale\_x\_continuous(expand = c(0, 0), limits = c(0, 6))+  
 labs(x = "log(Rosette Area) in 2021", y = "log(Rosette Area) in 2022")

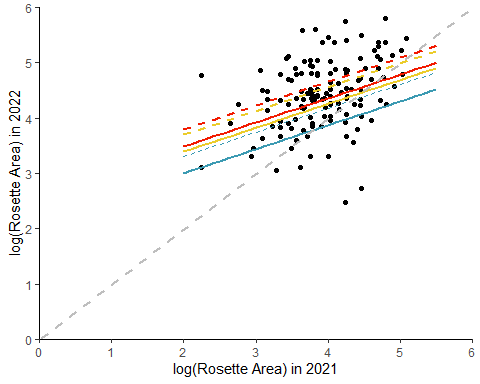
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.

## Warning: Removed 10 rows containing missing values (`geom\_point()`).



#create df with all possible x values  
#this is for model without an interaction  
pred.dat <- expand.grid(log.ros.area = seq(2, 5.5, by = .1), sm\_avg = seq(.1, .25, by = .01), log.total.seeds = seq(0, 5.3, by = .1))  
pred.output <- predict(growth.10, newdata = pred.dat, type="response", re.form=~0) #re.form = ~0 tells it to not include random effects  
pred.output <- as.data.frame(pred.output)  
pred.dat.output <- cbind(pred.dat, pred.output) %>%   
 rename(log.ros.areaT1 = pred.output)  
  
#graph it  
ggplot(clip, aes(x = log.ros.area, y = log.ros.areaT1))+  
 geom\_point()+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .25 & log.total.seeds == 5.3), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[1], size = 1)+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .13 & log.total.seeds == 5.3), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[3], size = 1)+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .1 & log.total.seeds == 5.3), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[5], size = 1)+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .25 & log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), linetype = "dashed", color = Z[1])+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .13 & log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[3], linetype = "dashed", size = 1)+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .1 & log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[5], linetype = "dashed", size = 1)+  
 geom\_abline(slope=1, intercept = 0, linetype = 2, color= "grey", size = 1)+  
 scale\_y\_continuous(expand = c(0, 0), limits = c(0, 6))+  
 scale\_x\_continuous(expand = c(0, 0), limits = c(0, 6))+  
 labs(x = "log(Rosette Area) in 2021", y = "log(Rosette Area) in 2022")

## Warning: Removed 10 rows containing missing values (`geom\_point()`).



#these seem kind of weird - I think the interaction may make more sense

pf.01 <- glmer(pflowerT1 ~ log.ros.area +(1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.02 <- glmer(pflowerT1 ~ log.ros.area\*trt +(1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.03 <- glmer(pflowerT1 ~ log.ros.area + trt + log.total.seeds + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.04 <- glmer(pflowerT1 ~ log.ros.area + trt + log.total.seeds + trt:log.total.seeds + log.ros.area:trt + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.05 <- glmer(pflowerT1 ~ log.ros.area + log.total.seeds + trt + log.total.seeds:trt + log.ros.area:log.total.seeds + log.ros.area:trt + log.ros.area:log.total.seeds:trt + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.06 <- glmer(pflowerT1 ~ log.ros.area + trt + log.cap + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.07 <- glmer(pflowerT1 ~ log.ros.area + trt + log.cap + trt:log.cap + log.ros.area:trt + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.08 <- glmer(pflowerT1 ~ log.ros.area + log.cap + trt + log.cap:trt + log.ros.area:log.cap + log.ros.area:trt + log.ros.area:log.total.seeds:trt + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.09 <- glmer(pflowerT1 ~ log.ros.area\*sm\_avg +(1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.10 <- glmer(pflowerT1 ~ log.ros.area + sm\_avg + log.total.seeds + (1|plot), data = clip, family = "binomial")  
pf.11 <- glmer(pflowerT1 ~ log.ros.area + sm\_avg + log.total.seeds + sm\_avg:log.total.seeds + log.ros.area:sm\_avg + (1|plot), data = clip, family = "binomial")

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable: large eigenvalue ratio  
## - Rescale variables?

pf.12 <- glmer(pflowerT1 ~ log.ros.area + log.total.seeds + sm\_avg + log.total.seeds:sm\_avg + log.ros.area:log.total.seeds + log.ros.area:sm\_avg + log.ros.area:log.total.seeds:sm\_avg + (1|plot), data = clip, family = "binomial")

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable: large eigenvalue ratio  
## - Rescale variables?

pf.13 <- glmer(pflowerT1 ~ log.ros.area + sm\_avg + log.cap + (1|plot), data = clip, family = "binomial")  
pf.14 <- glmer(pflowerT1 ~ log.ros.area + sm\_avg + log.cap + sm\_avg:log.cap + log.ros.area:sm\_avg + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

pf.15 <- glmer(pflowerT1 ~ log.ros.area + log.cap + sm\_avg + log.cap:sm\_avg + log.ros.area:log.cap + log.ros.area:sm\_avg + log.ros.area:log.total.seeds:sm\_avg + (1|plot), data = clip, family = "binomial")

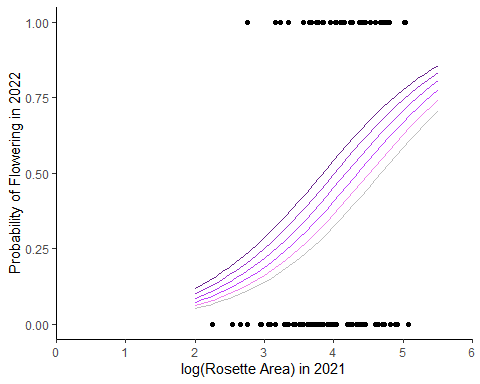
## boundary (singular) fit: see help('isSingular')

pf.16 <- glmer(pflowerT1 ~ log.ros.area + log.total.seeds +(1|plot), data = clip, family = "binomial")  
pf.17 <- glmer(pflowerT1 ~ log.ros.area\*log.total.seeds +(1|plot), data = clip, family = "binomial")  
pf.18 <- glmer(pflowerT1 ~ log.ros.area + log.cap +(1|plot), data = clip, family = "binomial")  
pf.19 <- glmer(pflowerT1 ~ log.ros.area\*log.cap +(1|plot), data = clip, family = "binomial")  
  
pflower <- mget(ls(pattern = "pf")) #make a list of all of those models that start with gm (^ means start with)  
pflower\_AICc<-aictab(cand.set = pflower, modnames = model.names,second.ord=TRUE,nobs=NULL,sort=TRUE)  
  
#rm(list = ls()[grepl("pflower", ls())]) #removes these models  
summary(pf.16)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: pflowerT1 ~ log.ros.area + log.total.seeds + (1 | plot)  
## Data: clip  
##   
## AIC BIC logLik deviance df.resid   
## 174.0 185.7 -83.0 166.0 135   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.5838 -0.7210 -0.4974 0.9565 2.7851   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.0586 0.2421   
## Number of obs: 139, groups: plot, 11  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.0607 1.5530 -3.259 0.00112 \*\*  
## log.ros.area 1.0814 0.3920 2.758 0.00581 \*\*  
## log.total.seeds 0.1812 0.1210 1.497 0.13426   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) lg.rs.  
## log.ros.are -0.987   
## log.ttl.sds 0.151 -0.242

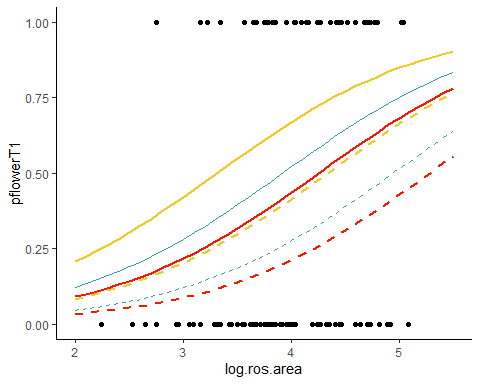
#create df with all possible x values  
#this is for model without an interaction  
pred.dat <- expand.grid(log.ros.area = seq(2, 5.5, by = .1), log.total.seeds = seq(0, 5.3, by = .1))  
pred.output <- predict(pf.16, newdata = pred.dat, type="response", re.form=~0) #re.form = ~0 tells it to not include random effects  
pred.output <- as.data.frame(pred.output)  
pred.dat.output <- cbind(pred.dat, pred.output) %>%   
 rename(log.ros.areaT1 = pred.output)  
  
ggplot(clip, aes(x = log.ros.area, y = pflowerT1))+  
 geom\_point()+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 5), aes(x = log.ros.area, y = log.ros.areaT1), color = "darkorchid4")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 4), aes(x = log.ros.area, y = log.ros.areaT1), color = "darkorchid3")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 3), aes(x = log.ros.area, y = log.ros.areaT1), color = "darkorchid2")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 2), aes(x = log.ros.area, y = log.ros.areaT1), color = "darkorchid1")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 1), aes(x = log.ros.area, y = log.ros.areaT1), color = "violet")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), color = "grey")+  
 scale\_x\_continuous(expand = c(0, 0), limits = c(0, 6))+  
 labs(x = "log(Rosette Area) in 2021", y = "Probability of Flowering in 2022")

## Warning: Removed 2 rows containing missing values (`geom\_point()`).



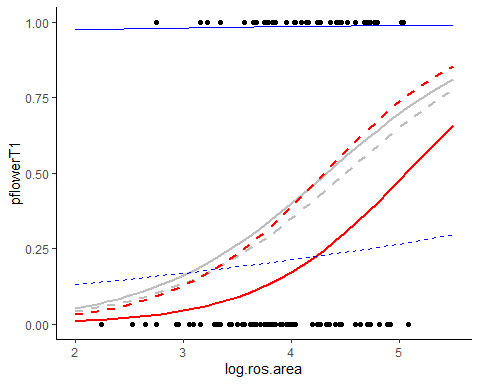
#create df with all possible x values  
#this is for model without an interaction  
pred.dat <- expand.grid(log.ros.area = seq(2, 5.5, by = .1), trt = c("irrigated", "control", "drought"), log.total.seeds = seq(0, 5.3, by = .1))  
pred.output <- predict(pf.03, newdata = pred.dat, type="response", re.form=~0) #re.form = ~0 tells it to not include random effects  
pred.output <- as.data.frame(pred.output)  
pred.dat.output <- cbind(pred.dat, pred.output) %>%   
 rename(log.ros.areaT1 = pred.output)  
  
ggplot(clip, aes(x = log.ros.area, y = pflowerT1))+  
 geom\_point()+  
 geom\_line(data = subset(pred.dat.output, trt == "irrigated" & log.total.seeds == 5.3), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[1])+  
 geom\_line(data = subset(pred.dat.output, trt == "control" & log.total.seeds == 5.3), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[3], size = 1)+  
 geom\_line(data = subset(pred.dat.output, trt == "drought" & log.total.seeds == 5.3), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[5], size = 1)+  
 geom\_line(data = subset(pred.dat.output, trt == "irrigated" & log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), linetype = "dashed", color = Z[1])+  
 geom\_line(data = subset(pred.dat.output, trt == "control" & log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[3], linetype = "dashed", size = 1)+  
 geom\_line(data = subset(pred.dat.output, trt == "drought" & log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), color = Z[5], linetype = "dashed", size = 1)

## Warning: Removed 2 rows containing missing values (`geom\_point()`).



#create df with all possible x values  
#this is for model without an interaction  
pred.dat <- expand.grid(log.ros.area = seq(2, 5.5, by = .1), sm\_avg = seq(.1, .25, by = .01), log.total.seeds = seq(0, 5.3, by = .1))  
pred.output <- predict(pf.11, newdata = pred.dat, type="response", re.form=~0) #re.form = ~0 tells it to not include random effects  
pred.output <- as.data.frame(pred.output)  
pred.dat.output <- cbind(pred.dat, pred.output) %>%   
 rename(log.ros.areaT1 = pred.output)  
  
ggplot(clip, aes(x = log.ros.area, y = pflowerT1))+  
 geom\_point()+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .25 & log.total.seeds == 5.3), aes(x = log.ros.area, y = log.ros.areaT1), color = "blue")+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .13 & log.total.seeds == 5.3), aes(x = log.ros.area, y = log.ros.areaT1), color = "grey", size = 1)+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .1 & log.total.seeds == 5.3), aes(x = log.ros.area, y = log.ros.areaT1), color = "red", size = 1)+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .25 & log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), linetype = "dashed", color = "blue")+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .13 & log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), color = "grey", linetype = "dashed", size = 1)+  
 geom\_line(data = subset(pred.dat.output, sm\_avg == .1 & log.total.seeds == 0), aes(x = log.ros.area, y = log.ros.areaT1), color = "red", linetype = "dashed", size = 1)

## Warning: Removed 2 rows containing missing values (`geom\_point()`).



#why are plants that make more seeds more likely to flower the next year. What to do???

nof.01 <- glmer(no.flowersT1 ~ log.ros.area + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.02 <- glmer(no.flowersT1 ~ log.ros.area\*trt + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.03 <- glmer(no.flowersT1 ~ log.ros.area + trt + log.total.seeds + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.04 <- glmer(no.flowersT1 ~ log.ros.area + trt + log.total.seeds + trt:log.total.seeds + log.ros.area:trt + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.05 <- glmer(no.flowersT1 ~ log.ros.area + log.total.seeds + trt + log.total.seeds:trt + log.ros.area:log.total.seeds + log.ros.area:trt + log.ros.area:log.total.seeds:trt + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.06 <- glmer(no.flowersT1 ~ log.ros.area + trt + log.cap + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.07 <- glmer(no.flowersT1 ~ log.ros.area + trt + log.cap + trt:log.cap + log.ros.area:trt + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.08 <- glmer(no.flowersT1 ~ log.ros.area + log.cap + trt + log.cap:trt + log.ros.area:log.cap + log.ros.area:trt + log.ros.area:log.total.seeds:trt + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.09 <- glmer(no.flowersT1 ~ log.ros.area\*sm\_avg + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.10 <- glmer(no.flowersT1 ~ log.ros.area + sm\_avg + log.total.seeds + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.11 <- glmer(no.flowersT1 ~ log.ros.area + sm\_avg + log.total.seeds + sm\_avg:log.total.seeds + log.ros.area:sm\_avg + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.12 <- glmer(no.flowersT1 ~ log.ros.area + log.total.seeds + sm\_avg + log.total.seeds:sm\_avg + log.ros.area:log.total.seeds + log.ros.area:sm\_avg + log.ros.area:log.total.seeds:sm\_avg + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.13 <- glmer(no.flowersT1 ~ log.ros.area + sm\_avg + log.cap + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.14 <- glmer(no.flowersT1 ~ log.ros.area + sm\_avg + log.cap + sm\_avg:log.cap + log.ros.area:sm\_avg + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.15 <- glmer(no.flowersT1 ~ log.ros.area + log.cap + sm\_avg + log.cap:sm\_avg + log.ros.area:log.cap + log.ros.area:sm\_avg + log.ros.area:log.total.seeds:sm\_avg + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.16 <- glmer(no.flowersT1 ~ log.ros.area + log.total.seeds + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.17 <- glmer(no.flowersT1 ~ log.ros.area\*log.total.seeds + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.18 <- glmer(no.flowersT1 ~ log.ros.area + log.cap + (1|plot), data = clip, family = "poisson")

## boundary (singular) fit: see help('isSingular')

nof.19 <- glmer(no.flowersT1 ~ log.ros.area\*log.cap + (1|plot), data = clip, family = "poisson")

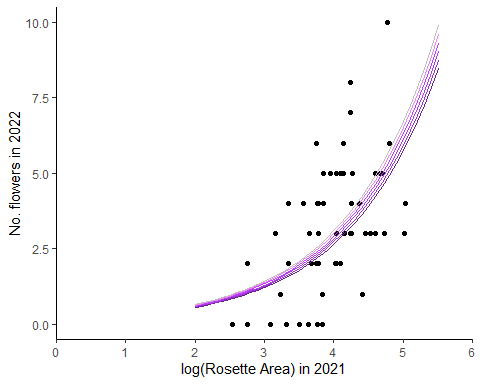
## boundary (singular) fit: see help('isSingular')

flowers <- mget(ls(pattern = "nof.")) #make a list of all of those models that start with gm (^ means start with)  
flowers\_AICc<-aictab(cand.set = flowers, modnames = model.names,second.ord=TRUE,nobs=NULL,sort=TRUE)  
  
#rm(list = ls()[grepl("flowers", ls())]) #removes these models  
summary(nof.17)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: poisson ( log )  
## Formula: no.flowersT1 ~ log.ros.area \* log.total.seeds + (1 | plot)  
## Data: clip  
##   
## AIC BIC logLik deviance df.resid   
## 221.8 231.8 -105.9 211.8 50   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.7083 -0.7617 -0.1230 0.7750 2.3676   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0 0   
## Number of obs: 55, groups: plot, 11  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.65452 0.72944 -3.639 0.000274 \*\*\*  
## log.ros.area 0.93767 0.17403 5.388 7.12e-08 \*\*\*  
## log.total.seeds 0.85465 0.42486 2.012 0.044261 \*   
## log.ros.area:log.total.seeds -0.20491 0.09865 -2.077 0.037794 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) lg.rs. lg.tt.  
## log.ros.are -0.992   
## log.ttl.sds -0.463 0.452   
## lg.rs.r:l.. 0.468 -0.465 -0.995  
## optimizer (Nelder\_Mead) convergence code: 0 (OK)  
## boundary (singular) fit: see help('isSingular')

#create df with all possible x values  
#this is for model without an interaction  
pred.dat <- expand.grid(log.ros.area = seq(2, 5.5, by = .1), log.total.seeds = seq(0, 5.3, by = .1))  
pred.output <- predict(nof.16, newdata = pred.dat, type="response", re.form=~0) #re.form = ~0 tells it to not include random effects  
pred.output <- as.data.frame(pred.output)  
pred.dat.output <- cbind(pred.dat, pred.output) %>%   
 rename(no.flowersT1 = pred.output)  
  
ggplot(clip, aes(x = log.ros.area, y = no.flowersT1))+  
 geom\_point()+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 5), aes(x = log.ros.area, y = no.flowersT1), color = "darkorchid4")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 4), aes(x = log.ros.area, y = no.flowersT1), color = "darkorchid3")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 3), aes(x = log.ros.area, y = no.flowersT1), color = "darkorchid2")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 2), aes(x = log.ros.area, y = no.flowersT1), color = "darkorchid1")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 1), aes(x = log.ros.area, y = no.flowersT1), color = "violet")+  
 geom\_line(data = subset(pred.dat.output, log.total.seeds == 0), aes(x = log.ros.area, y = no.flowersT1), color = "grey")+  
 scale\_x\_continuous(expand = c(0, 0), limits = c(0, 6))+  
 labs(x = "log(Rosette Area) in 2021", y = "No. flowers in 2022")

## Warning: Removed 89 rows containing missing values (`geom\_point()`).



#this seems reasonable...but is a singular fit!

sur.01 <- glmer(psurvivalT1 ~ log.ros.area +(1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.02 <- glmer(psurvivalT1 ~ log.ros.area\*trt +(1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.03 <- glmer(psurvivalT1 ~ log.ros.area + trt + log.total.seeds + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.04 <- glmer(psurvivalT1 ~ log.ros.area + trt + log.total.seeds + trt:log.total.seeds + log.ros.area:trt + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.05 <- glmer(psurvivalT1 ~ log.ros.area + log.total.seeds + trt + log.total.seeds:trt + log.ros.area:log.total.seeds + log.ros.area:trt + log.ros.area:log.total.seeds:trt + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.06 <- glmer(psurvivalT1 ~ log.ros.area + trt + log.cap + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.07 <- glmer(psurvivalT1 ~ log.ros.area + trt + log.cap + trt:log.cap + log.ros.area:trt + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.08 <- glmer(psurvivalT1 ~ log.ros.area + log.cap + trt + log.cap:trt + log.ros.area:log.cap + log.ros.area:trt + log.ros.area:log.total.seeds:trt + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.09 <- glmer(psurvivalT1 ~ log.ros.area\*sm\_avg +(1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.10 <- glmer(psurvivalT1 ~ log.ros.area + sm\_avg + log.total.seeds + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.11 <- glmer(psurvivalT1 ~ log.ros.area + sm\_avg + log.total.seeds + sm\_avg:log.total.seeds + log.ros.area:sm\_avg + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.12 <- glmer(psurvivalT1 ~ log.ros.area + log.total.seeds + sm\_avg + log.total.seeds:sm\_avg + log.ros.area:log.total.seeds + log.ros.area:sm\_avg + log.ros.area:log.total.seeds:sm\_avg + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.13 <- glmer(psurvivalT1 ~ log.ros.area + sm\_avg + log.cap + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.14 <- glmer(psurvivalT1 ~ log.ros.area + sm\_avg + log.cap + sm\_avg:log.cap + log.ros.area:sm\_avg + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.15 <- glmer(psurvivalT1 ~ log.ros.area + log.cap + sm\_avg + log.cap:sm\_avg + log.ros.area:log.cap + log.ros.area:sm\_avg + log.ros.area:log.total.seeds:sm\_avg + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.16 <- glmer(psurvivalT1 ~ log.ros.area + log.total.seeds + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.17 <- glmer(psurvivalT1 ~ log.ros.area\*log.total.seeds + (1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

sur.18 <- glmer(psurvivalT1 ~ log.ros.area + log.cap + (1|plot), data = clip, family = "binomial")

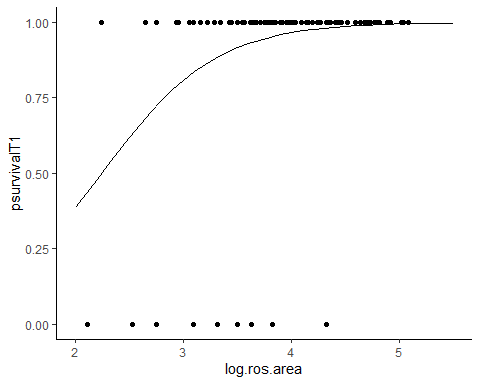
## boundary (singular) fit: see help('isSingular')

sur.19 <- glmer(psurvivalT1 ~ log.ros.area\*log.cap +(1|plot), data = clip, family = "binomial")

## boundary (singular) fit: see help('isSingular')

survival <- mget(ls(pattern = "sur.")) #make a list of all of those models that start with gm (^ means start with)  
survival\_AICc<-aictab(cand.set = survival, modnames = NULL,second.ord=TRUE,nobs=NULL,sort=TRUE)  
  
#rm(list = ls()[grepl("survival", ls())]) #removes these models  
#the "best" 2 models give weird warnigns so going with the null model with just size

#create df with all possible x values  
#this is for model without an interaction  
pred.dat <- expand.grid(log.ros.area = seq(2, 5.5, by = .1))  
pred.output <- predict(sur.01, newdata = pred.dat, type="response", re.form=~0) #re.form = ~0 tells it to not include random effects  
pred.output <- as.data.frame(pred.output)  
pred.dat.output <- cbind(pred.dat, pred.output) %>%   
 rename(log.ros.areaT1 = pred.output)  
  
ggplot(clip, aes(x = log.ros.area, y = psurvivalT1))+  
 geom\_point()+  
 geom\_line(data = pred.dat.output, aes(x = log.ros.area, y = log.ros.areaT1))



Maybe cost is in whether or not you flower, not how much effort you put into flowering?

Growth::

#What to do about differential flowering success → what about 3 models

# fecundity -

#Probability of successfully making seeds ~ treatment\*size (data = plants that tried to flower)

#Seed production ~ treatment\*size (data = plants that made seeds, this is Poisson)

Great, it’s significant - now make a graph