Analyses Cerastium transplants: Local adaptation

Analyses with FFD

Alicia Valdés

25 June, 2024

Table of Contents

# Read clean data from .csv files

data\_transplants <- read\_csv("data/clean/data\_transplants.csv")%>%  
 mutate\_at(c("unique\_id","heat\_zone","group","mother","father","id","repl",  
 "plot","peak","fruit\_coll","poll\_type","crossing","F\_NF\_A",  
 "year"),as.factor)%>%  
 mutate\_at(c("n\_fl\_stems","n\_opened\_fl","n\_closed\_buds","tot\_fl\_bodies",  
 "median\_h","per\_invert\_herb","n\_stems\_grazed","n\_closed\_coll\_fr",  
 "n\_open\_coll\_fr","n\_count\_fr","n\_tot\_fr","n\_seed\_fr1",  
 "n\_seed\_fr2","n\_seed\_fr3","n\_seed\_fr4","rest\_combined",  
 "n\_tot\_seed","tot\_fl\_bodies\_corr"),as.integer)

# Check that temperatures for sires and dams coincide with temperatures in list of sires and dams:

temps\_dams\_sires<-data\_transplants%>%  
 dplyr::select(mother,father,temp\_mother,temp\_father)%>%  
 distinct()%>% # Removes duplicate rows  
 pivot\_longer(cols=starts\_with("temp"),names\_to="type",values\_to="temp")%>%  
 mutate(id = case\_when(type=="temp\_mother"~paste0("D",mother),  
 type=="temp\_father"~paste0("S",father)))%>%  
 dplyr::select(id,temp)  
# Read csv file with temperatures from sire and dam list  
temps\_dams\_sires\_list<-read\_csv("C:/Users/alici/Dropbox/SU/Projects/cerastium\_greenhouse/data/clean/temps\_dams\_sires\_list.csv")  
# Merge the data frames on the "id" column  
merged\_data<-temps\_dams\_sires%>%  
 inner\_join(temps\_dams\_sires\_list,by="id",suffix=c("\_data","\_list"))  
# Check for discrepancies  
discrepancies<-merged\_data%>%filter(temp\_data!=temp\_list)  
# View the discrepancies  
print(discrepancies) # Two discrepancies

## # A tibble: 2 × 3  
## id temp\_data temp\_list  
## <chr> <dbl> <dbl>  
## 1 D24 20 14.9  
## 2 S320 13.6 9

Correct discrepancies:

Temperature for mother 24 should be 14.9, not 20. Temperature for father 320 should be 9.0, not 13.6.

data\_transplants <- data\_transplants %>%  
 mutate(temp\_mother = if\_else(mother == 24, 14.9, temp\_mother),  
 temp\_father = if\_else(father == 320, 9.0, temp\_father))

Check again for discrepancies:

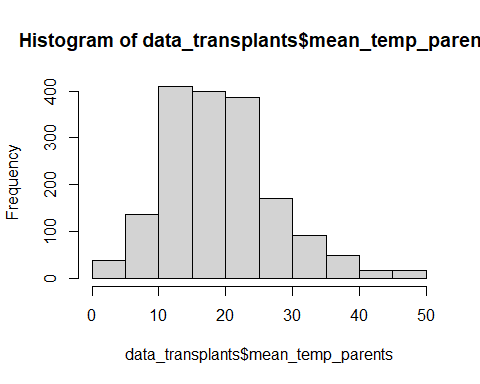
temps\_dams\_sires<-data\_transplants%>%  
 dplyr::select(mother,father,temp\_mother,temp\_father)%>%  
 distinct()%>% # Removes duplicate rows  
 pivot\_longer(cols=starts\_with("temp"),names\_to="type",values\_to="temp")%>%  
 mutate(id = case\_when(type=="temp\_mother"~paste0("D",mother),  
 type=="temp\_father"~paste0("S",father)))%>%  
 dplyr::select(id,temp)  
# Read csv file with temperatures from sire and dam list  
merged\_data<-temps\_dams\_sires%>%  
 inner\_join(temps\_dams\_sires\_list,by="id",suffix=c("\_data","\_list"))  
# Check for discrepancies  
discrepancies<-merged\_data%>%filter(temp\_data!=temp\_list)  
# View the discrepancies  
print(discrepancies) # No discrepancies

## # A tibble: 0 × 3  
## # ℹ 3 variables: id <chr>, temp\_data <dbl>, temp\_list <dbl>

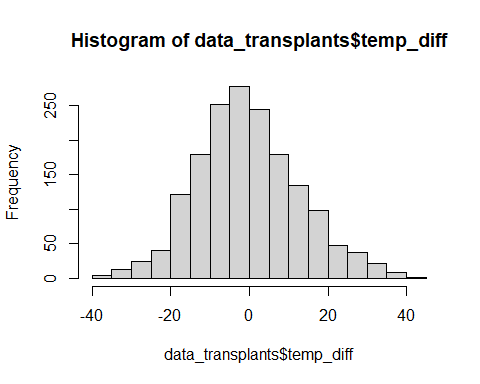
# Calculate temperature difference

Calculate temperature difference (actual value, not absolute!) between experienced temperature (temp) and mid-parental temperature.

data\_transplants<-data\_transplants%>%  
 mutate(mean\_temp\_parents=(temp\_mother+temp\_father)/2,  
 temp\_diff=temp-mean\_temp\_parents)  
hist(data\_transplants$mean\_temp\_parents)



hist(data\_transplants$temp\_diff)



When the temperature difference is positive –> planted in warmer places than their origin When the temperature difference is negative –> planted in colder places than their origin

# Prediction 4 (from old paper 1)

See if we use these analyses!

“Divergent temperature-dependent selection has resulted in small-scale genetic differentiation and local adaptation along gradients of soil temperature, and this local adaptation involves flowering time.”

To test if plants are adapted to their local thermal environment, we regressed fitness on the difference in temperature between the transplantation site and the mean temperature of the parents’ sites of origin.

## Fitness

### Models without phenology

#### 2022

model\_local\_adapt\_22<-glmmTMB(n\_tot\_seed~temp\_diff+(1|plot),  
 subset(data\_transplants,year==2022),family="nbinom2")  
model\_local\_adapt\_22\_zi<-glmmTMB(n\_tot\_seed~temp\_diff+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2022),  
 family="nbinom2")  
summary(model\_local\_adapt\_22)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 7948.3 7967.3 -3970.2 7940.3 845   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.07715 0.2778   
## Number of obs: 849, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 0.314   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.95010 0.12610 31.326 <2e-16 \*\*\*  
## temp\_diff -0.01177 0.00597 -1.972 0.0487 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_22\_zi)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 6896.8 6930.0 -3441.4 6882.8 842   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 5.085e-10 2.255e-05  
## Number of obs: 849, groups: plot, 8  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.9335 0.9662   
## Number of obs: 849, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 3.91   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.452928 0.021556 206.57 <2e-16 \*\*\*  
## temp\_diff 0.001898 0.001740 1.09 0.275   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.517503 0.353406 -1.464 0.143   
## temp\_diff 0.054834 0.008603 6.374 1.84e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_22,model\_local\_adapt\_22\_zi)

## df AIC  
## model\_local\_adapt\_22 4 7948.323  
## model\_local\_adapt\_22\_zi 7 6896.833

Model with zero inflation is best.

Try with quadratic effect of temp\_diff:

model\_local\_adapt\_22\_zi\_sq<-glmmTMB(n\_tot\_seed~temp\_diff+I(temp\_diff^2)+  
 (1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2022),  
 family="nbinom2")  
summary(model\_local\_adapt\_22\_zi\_sq)

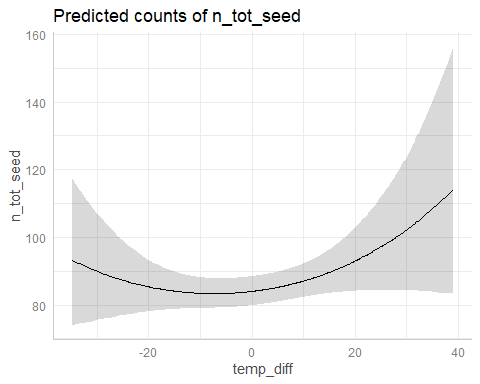
## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 6892.1 6934.8 -3437.0 6874.1 840   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 8.856e-10 2.976e-05  
## Number of obs: 849, groups: plot, 8  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.9674 0.9836   
## Number of obs: 849, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 3.92   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.431e+00 2.603e-02 170.21 <2e-16 \*\*\*  
## temp\_diff 2.126e-03 1.780e-03 1.19 0.232   
## I(temp\_diff^2) 1.462e-04 9.818e-05 1.49 0.137   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.6563984 0.3635522 -1.806 0.07099 .   
## temp\_diff 0.0492371 0.0085897 5.732 9.92e-09 \*\*\*  
## I(temp\_diff^2) 0.0009127 0.0003530 2.586 0.00971 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_22\_zi,model\_local\_adapt\_22\_zi\_sq)

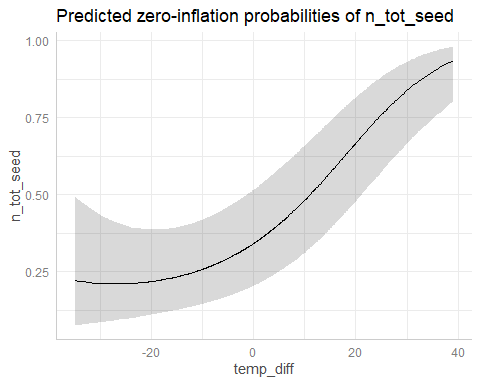
## df AIC  
## model\_local\_adapt\_22\_zi 7 6896.833  
## model\_local\_adapt\_22\_zi\_sq 9 6892.099

Model with quadratic effect is best.

plot(ggpredict(model\_local\_adapt\_22\_zi\_sq,type="count",terms="temp\_diff [all]")) #NS



plot(ggpredict(model\_local\_adapt\_22\_zi\_sq,type="zi\_prob",terms="temp\_diff [all]")) #\*



#### 2023

model\_local\_adapt\_23<-glmmTMB(n\_tot\_seed~temp\_diff+(1|plot),  
 subset(data\_transplants,year==2023),family="nbinom2")  
model\_local\_adapt\_23\_zi<-glmmTMB(n\_tot\_seed~temp\_diff+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2023),  
 family="nbinom2")  
summary(model\_local\_adapt\_23)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + (1 | plot)  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 4841.3 4860.2 -2416.6 4833.3 829   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 6.304 2.511   
## Number of obs: 833, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 0.134   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.073807 0.910654 2.277 0.0228 \*  
## temp\_diff -0.019060 0.009999 -1.906 0.0566 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_23\_zi)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 4119.9 4153.0 -2052.9 4105.9 826   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.02758 0.1661   
## Number of obs: 833, groups: plot, 8  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 2.833 1.683   
## Number of obs: 833, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 3.07   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.044110 0.079090 51.13 <2e-16 \*\*\*  
## temp\_diff -0.002317 0.003132 -0.74 0.459   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.835417 0.617999 1.352 0.176   
## temp\_diff 0.032493 0.007938 4.093 4.25e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_23,model\_local\_adapt\_23\_zi)

## df AIC  
## model\_local\_adapt\_23 4 4841.291  
## model\_local\_adapt\_23\_zi 7 4119.884

Model with zero inflation is best.

Try with quadratic effect of temp\_diff:

model\_local\_adapt\_23\_zi\_sq<-glmmTMB(n\_tot\_seed~temp\_diff+I(temp\_diff^2)+  
 (1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2023),  
 family="nbinom2")  
summary(model\_local\_adapt\_23\_zi\_sq)

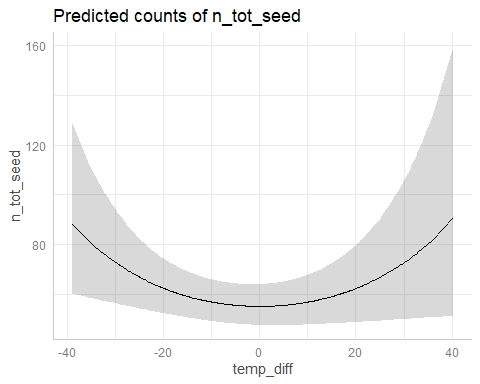
## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 4105.2 4147.8 -2043.6 4087.2 824   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.02276 0.1509   
## Number of obs: 833, groups: plot, 8  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 3.032 1.741   
## Number of obs: 833, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 3.12   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.013e+00 7.514e-02 53.40 <2e-16 \*\*\*  
## temp\_diff -7.153e-05 3.441e-03 -0.02 0.9834   
## I(temp\_diff^2) 3.071e-04 1.264e-04 2.43 0.0151 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.6520985 0.6407205 1.018 0.308793   
## temp\_diff 0.0329361 0.0085315 3.861 0.000113 \*\*\*  
## I(temp\_diff^2) 0.0012801 0.0003757 3.407 0.000656 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_23\_zi,model\_local\_adapt\_23\_zi\_sq)

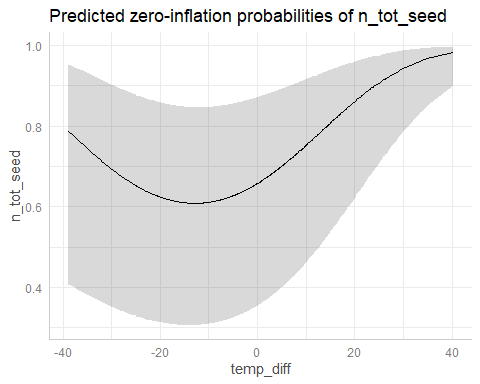
## df AIC  
## model\_local\_adapt\_23\_zi 7 4119.884  
## model\_local\_adapt\_23\_zi\_sq 9 4105.225

Model with quadratic effect is best.

plot(ggpredict(model\_local\_adapt\_23\_zi\_sq,type="count",terms="temp\_diff [all]")) #\*

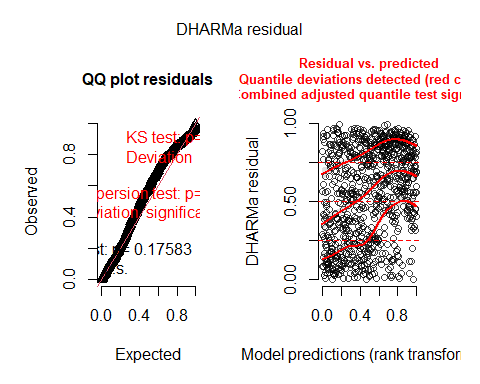


plot(ggpredict(model\_local\_adapt\_23\_zi\_sq,type="zi\_prob",terms="temp\_diff [all]")) #\*

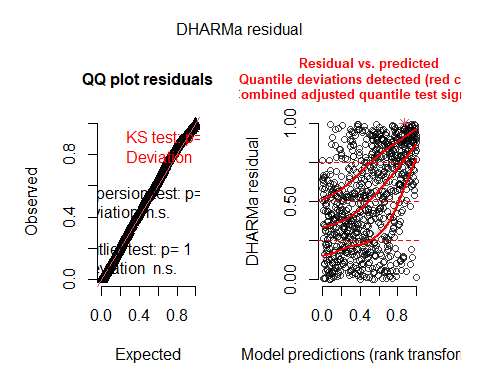


#### Diagnostics

plot(simulateResiduals(model\_local\_adapt\_22\_zi\_sq))



plot(simulateResiduals(model\_local\_adapt\_23\_zi\_sq))



Take as OK??

### Models with phenology

To test to what extent any such evidence of local adaptation to the thermal environment was mediated by differences in flowering time, we ran a model including also expressed flowering time of focal individuals.

If the effect of temperature difference on fitness is not significant when including flowering time, we can conclude that local adaptation is mediated by differences in flowering time?

#### FFD

##### 2022

model\_local\_adapt\_22\_FFD<-glmmTMB(n\_tot\_seed~temp\_diff+FFD\_corr+(1|plot),  
 subset(data\_transplants,year==2022),family="nbinom2")  
model\_local\_adapt\_22\_zi\_FFD<-glmmTMB(n\_tot\_seed~temp\_diff+FFD\_corr,  
 # +(1|plot), #NaNs in summary if plot   
 # included in conditional model  
 # probably because of low variance explained  
 ziformula=~temp\_diff+FFD\_corr+(1|plot),  
 subset(data\_transplants,year==2022),  
 family="nbinom2")  
summary(model\_local\_adapt\_22\_FFD)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + FFD\_corr + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 6386.2 6408.2 -3188.1 6376.2 597   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 1.428e-09 3.778e-05  
## Number of obs: 602, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 2.16   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.684711 0.308769 18.411 < 2e-16 \*\*\*  
## temp\_diff -0.006812 0.002895 -2.353 0.0186 \*   
## FFD\_corr -0.007622 0.001828 -4.170 3.04e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_22\_zi\_FFD)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + FFD\_corr  
## Zero inflation: ~temp\_diff + FFD\_corr + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 6113.0 6148.2 -3048.5 6097.0 594   
##   
## Random effects:  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 1.637e-07 0.0004046  
## Number of obs: 602, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 4.04   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.468817 0.233341 23.437 < 2e-16 \*\*\*  
## temp\_diff -0.003841 0.002176 -1.765 0.0775 .   
## FFD\_corr -0.006087 0.001385 -4.396 1.1e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -9.50111 2.27484 -4.177 2.96e-05 \*\*\*  
## temp\_diff 0.07649 0.02306 3.317 0.000911 \*\*\*  
## FFD\_corr 0.03619 0.01276 2.836 0.004571 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_22\_FFD,model\_local\_adapt\_22\_zi\_FFD)

## df AIC  
## model\_local\_adapt\_22\_FFD 5 6386.196  
## model\_local\_adapt\_22\_zi\_FFD 8 6112.967

Try with quadratic effect of temp\_diff:

model\_local\_adapt\_22\_zi\_FFD\_sq<-glmmTMB(n\_tot\_seed~temp\_diff+I(temp\_diff^2)+  
 FFD\_corr+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2022),  
 family="nbinom2")  
summary(model\_local\_adapt\_22\_zi\_FFD\_sq)

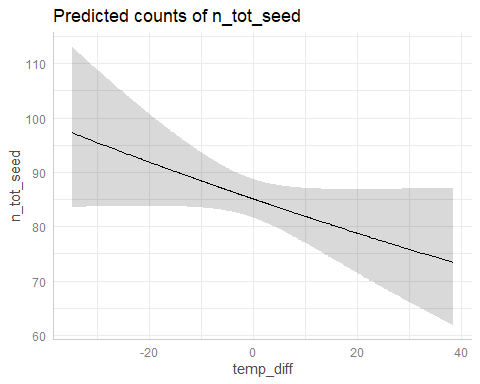
## Family: nbinom2 ( log )  
## Formula:   
## n\_tot\_seed ~ temp\_diff + I(temp\_diff^2) + FFD\_corr + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 6117.5 6165.9 -3047.7 6095.5 591   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 1.312e-12 1.146e-06  
## Number of obs: 602, groups: plot, 8  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 8.325e-08 0.0002885  
## Number of obs: 602, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 4.05   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 5.4307763 NaN NaN NaN  
## temp\_diff -0.0035839 NaN NaN NaN  
## I(temp\_diff^2) 0.0001181 NaN NaN NaN  
## FFD\_corr -0.0059654 NaN NaN NaN  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -9.447507 NaN NaN NaN  
## temp\_diff 0.080972 NaN NaN NaN  
## I(temp\_diff^2) -0.000332 NaN NaN NaN  
## FFD\_corr 0.036081 NaN NaN NaN

AIC(model\_local\_adapt\_22\_zi\_FFD,model\_local\_adapt\_22\_zi\_FFD\_sq)

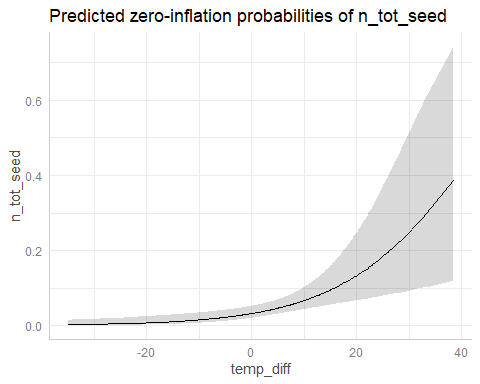
## df AIC  
## model\_local\_adapt\_22\_zi\_FFD 8 6112.967  
## model\_local\_adapt\_22\_zi\_FFD\_sq 11 6117.476

Model with quadratic effect is worst. Use model without quadratic effect.

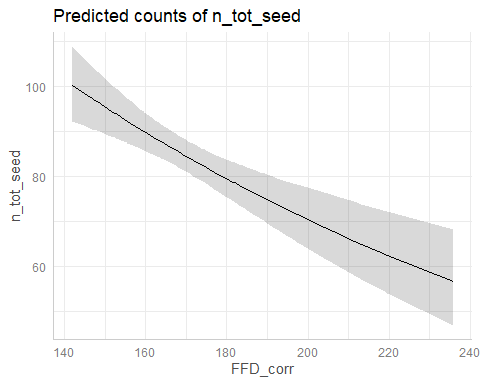
plot(ggpredict(model\_local\_adapt\_22\_zi\_FFD,type="count",terms="temp\_diff [all]")) #NS



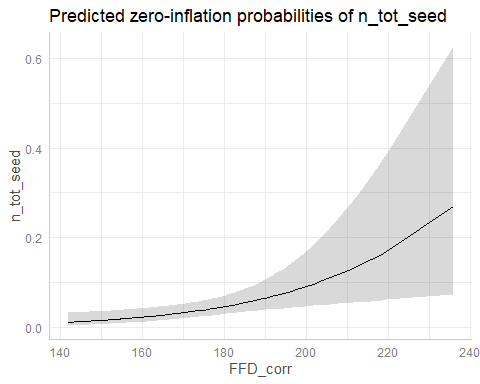
plot(ggpredict(model\_local\_adapt\_22\_zi\_FFD,type="zi\_prob",terms="temp\_diff [all]")) #\*



plot(ggpredict(model\_local\_adapt\_22\_zi\_FFD,type="count",terms="FFD\_corr [all]")) #\*



plot(ggpredict(model\_local\_adapt\_22\_zi\_FFD,type="zi\_prob",terms="FFD\_corr [all]")) #\*



##### 2023

model\_local\_adapt\_23\_FFD<-glmmTMB(n\_tot\_seed~temp\_diff+FFD\_corr+(1|plot),  
 subset(data\_transplants,year==2023),family="nbinom2")  
model\_local\_adapt\_23\_zi\_FFD<-glmmTMB(n\_tot\_seed~temp\_diff+FFD\_corr+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2023),  
 family="nbinom2")  
summary(model\_local\_adapt\_23\_FFD)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + FFD\_corr + (1 | plot)  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 3947.9 3967.9 -1968.9 3937.9 401   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.05842 0.2417   
## Number of obs: 406, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 0.651   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 9.304918 1.650487 5.638 1.72e-08 \*\*\*  
## temp\_diff -0.007258 0.006217 -1.167 0.243031   
## FFD\_corr -0.030886 0.009332 -3.310 0.000934 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_23\_zi\_FFD)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + FFD\_corr + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 3549.4 3585.5 -1765.7 3531.4 397   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.03143 0.1773   
## Number of obs: 406, groups: plot, 7  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.809 1.345   
## Number of obs: 406, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 3.29   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.205603 0.919136 8.928 < 2e-16 \*\*\*  
## temp\_diff -0.005907 0.003234 -1.827 0.0677 .   
## FFD\_corr -0.023694 0.005211 -4.547 5.44e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -9.725060 3.243107 -2.999 0.00271 \*\*  
## temp\_diff -0.003297 0.015626 -0.211 0.83289   
## FFD\_corr 0.041341 0.017541 2.357 0.01843 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_23\_FFD,model\_local\_adapt\_23\_zi\_FFD)

## df AIC  
## model\_local\_adapt\_23\_FFD 5 3947.895  
## model\_local\_adapt\_23\_zi\_FFD 9 3549.404

Try with quadratic effect of temp\_diff:

model\_local\_adapt\_23\_zi\_FFD\_sq<-glmmTMB(n\_tot\_seed~temp\_diff+I(temp\_diff^2)+  
 FFD\_corr+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2023),  
 family="nbinom2")  
summary(model\_local\_adapt\_23\_zi\_FFD\_sq)

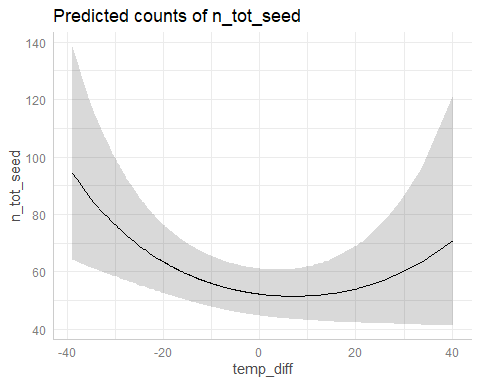
## Family: nbinom2 ( log )  
## Formula:   
## n\_tot\_seed ~ temp\_diff + I(temp\_diff^2) + FFD\_corr + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 3547.8 3591.8 -1762.9 3525.8 395   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.02963 0.1721   
## Number of obs: 406, groups: plot, 7  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.805 1.344   
## Number of obs: 406, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 3.35   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.1089371 0.9177403 8.836 < 2e-16 \*\*\*  
## temp\_diff -0.0040166 0.0035027 -1.147 0.2515   
## I(temp\_diff^2) 0.0002882 0.0001205 2.392 0.0168 \*   
## FFD\_corr -0.0233379 0.0052005 -4.488 7.2e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -9.725e+00 3.241e+00 -3.001 0.0027 \*\*  
## temp\_diff -2.746e-03 1.860e-02 -0.148 0.8826   
## I(temp\_diff^2) 3.428e-05 6.689e-04 0.051 0.9591   
## FFD\_corr 4.133e-02 1.754e-02 2.357 0.0184 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_23\_zi\_FFD,model\_local\_adapt\_23\_zi\_FFD\_sq)

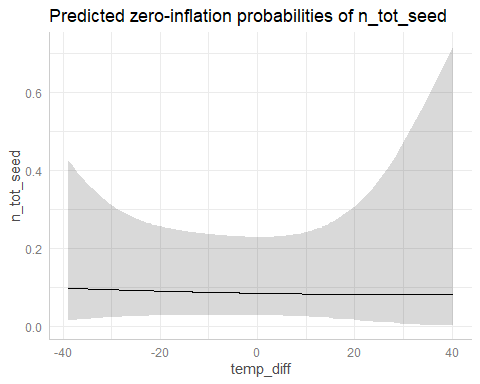
## df AIC  
## model\_local\_adapt\_23\_zi\_FFD 9 3549.404  
## model\_local\_adapt\_23\_zi\_FFD\_sq 11 3547.772

Model with quadratic effect is best.

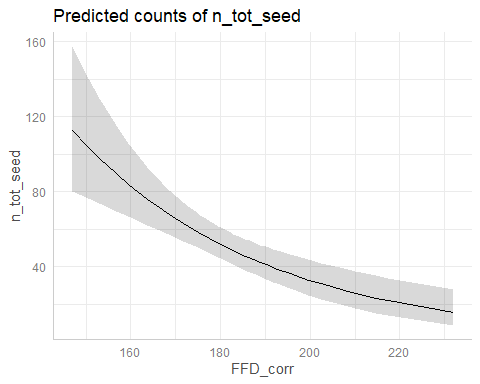
plot(ggpredict(model\_local\_adapt\_23\_zi\_FFD\_sq,type="count",terms="temp\_diff [all]")) #\*



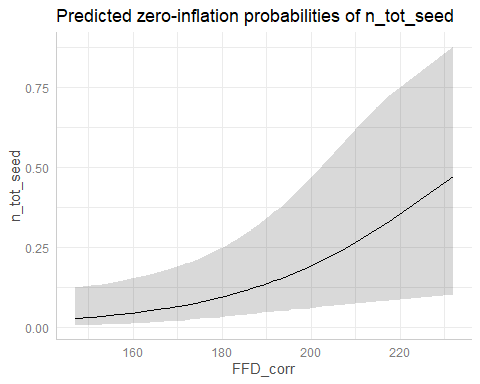
plot(ggpredict(model\_local\_adapt\_23\_zi\_FFD\_sq,type="zi\_prob",terms="temp\_diff [all]")) #NS



plot(ggpredict(model\_local\_adapt\_23\_zi\_FFD\_sq,type="count",terms="FFD\_corr [all]")) #\*



plot(ggpredict(model\_local\_adapt\_23\_zi\_FFD\_sq,type="zi\_prob",terms="FFD\_corr [all]")) #\*



#### MeanFD

##### 2022

model\_local\_adapt\_22\_MeanFD<-glmmTMB(n\_tot\_seed~temp\_diff+MeanFD+(1|plot),  
 subset(data\_transplants,year==2022),family="nbinom2")  
model\_local\_adapt\_22\_zi\_MeanFD<-glmmTMB(n\_tot\_seed~temp\_diff+MeanFD+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2022),  
 family="nbinom2")  
summary(model\_local\_adapt\_22\_MeanFD)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + MeanFD + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 3843.5 3862.9 -1916.7 3833.5 358   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 7.142e-10 2.672e-05  
## Number of obs: 363, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 1.99   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.920160 0.432092 13.701 < 2e-16 \*\*\*  
## temp\_diff -0.011317 0.004103 -2.758 0.005809 \*\*   
## MeanFD -0.008854 0.002446 -3.620 0.000295 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_22\_zi\_MeanFD)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + MeanFD + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 3683.2 3718.2 -1832.6 3665.2 354   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 4.331e-10 2.081e-05  
## Number of obs: 363, groups: plot, 8  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 3.528e-09 5.94e-05  
## Number of obs: 363, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 3.79   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.655573 0.323820 17.465 < 2e-16 \*\*\*  
## temp\_diff -0.008009 0.003052 -2.624 0.008692 \*\*   
## MeanFD -0.007078 0.001838 -3.851 0.000118 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -10.20180 3.07090 -3.322 0.000893 \*\*\*  
## temp\_diff 0.07998 0.03164 2.528 0.011478 \*   
## MeanFD 0.03932 0.01651 2.381 0.017270 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_22\_FFD,model\_local\_adapt\_22\_zi\_MeanFD)

## df AIC  
## model\_local\_adapt\_22\_FFD 5 6386.196  
## model\_local\_adapt\_22\_zi\_MeanFD 9 3683.156

Try with quadratic effect of temp\_diff:

model\_local\_adapt\_22\_zi\_MeanFD\_sq<-glmmTMB(n\_tot\_seed~temp\_diff+I(temp\_diff^2)+  
 MeanFD+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2022),  
 family="nbinom2")  
summary(model\_local\_adapt\_22\_zi\_MeanFD\_sq)

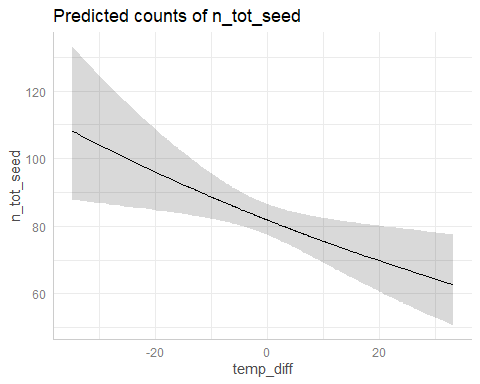
## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + I(temp\_diff^2) + MeanFD + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 3687.1 3729.9 -1832.5 3665.1 352   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 6.348e-11 7.968e-06  
## Number of obs: 363, groups: plot, 8  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 5.703e-08 0.0002388  
## Number of obs: 363, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 3.79   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.658e+00 3.302e-01 17.133 < 2e-16 \*\*\*  
## temp\_diff -8.040e-03 3.181e-03 -2.527 0.011491 \*   
## I(temp\_diff^2) -5.079e-06 1.373e-04 -0.037 0.970481   
## MeanFD -7.087e-03 1.857e-03 -3.816 0.000135 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.006e+01 3.098e+00 -3.247 0.00117 \*\*  
## temp\_diff 8.331e-02 3.522e-02 2.365 0.01802 \*   
## I(temp\_diff^2) -4.215e-04 1.553e-03 -0.271 0.78607   
## MeanFD 3.876e-02 1.655e-02 2.342 0.01918 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_22\_zi\_MeanFD,model\_local\_adapt\_22\_zi\_MeanFD\_sq)

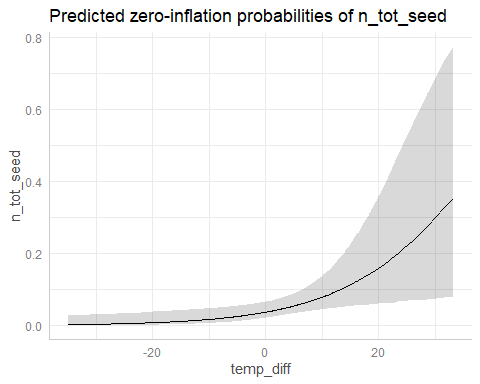
## df AIC  
## model\_local\_adapt\_22\_zi\_MeanFD 9 3683.156  
## model\_local\_adapt\_22\_zi\_MeanFD\_sq 11 3687.081

Model with quadratic effect is worst. Use model without quadratic effect.

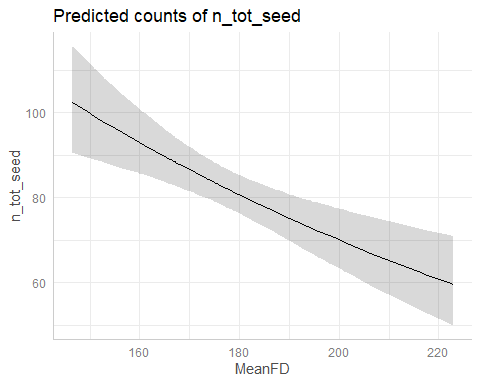
plot(ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,type="count",terms="temp\_diff [all]")) #\*



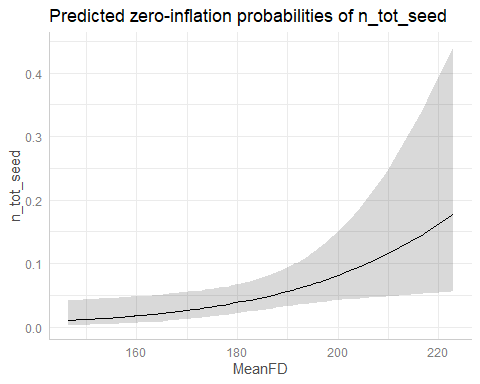
plot(ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,type="zi\_prob",terms="temp\_diff [all]")) #\*



plot(ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,type="count",terms="MeanFD [all]")) #\*



plot(ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,type="zi\_prob",terms="MeanFD [all]")) #\*



##### 2023

model\_local\_adapt\_23\_MeanFD<-glmmTMB(n\_tot\_seed~temp\_diff+MeanFD+(1|plot),  
 subset(data\_transplants,year==2023),family="nbinom2")  
model\_local\_adapt\_23\_zi\_MeanFD<-glmmTMB(n\_tot\_seed~temp\_diff+MeanFD+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2023),  
 family="nbinom2")  
summary(model\_local\_adapt\_23\_MeanFD)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + MeanFD + (1 | plot)  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 2432.5 2450.1 -1211.3 2422.5 244   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.04191 0.2047   
## Number of obs: 249, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 0.674   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 9.562731 2.064849 4.631 3.64e-06 \*\*\*  
## temp\_diff -0.008462 0.008457 -1.001 0.31706   
## MeanFD -0.031545 0.011436 -2.758 0.00581 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_23\_zi\_MeanFD)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + MeanFD + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 2215.1 2246.8 -1098.6 2197.1 240   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.02251 0.15   
## Number of obs: 249, groups: plot, 7  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.745 1.321   
## Number of obs: 249, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 3.04   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.378213 1.233235 6.794 1.09e-11 \*\*\*  
## temp\_diff -0.007402 0.004665 -1.587 0.112563   
## MeanFD -0.024023 0.006856 -3.504 0.000458 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -10.501110 3.984695 -2.635 0.0084 \*\*  
## temp\_diff 0.001202 0.020647 0.058 0.9536   
## MeanFD 0.044689 0.021009 2.127 0.0334 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_23\_MeanFD,model\_local\_adapt\_23\_zi\_MeanFD)

## df AIC  
## model\_local\_adapt\_23\_MeanFD 5 2432.512  
## model\_local\_adapt\_23\_zi\_MeanFD 9 2215.133

Try with quadratic effect of temp\_diff:

model\_local\_adapt\_23\_zi\_MeanFD\_sq<-glmmTMB(n\_tot\_seed~temp\_diff+I(temp\_diff^2)+  
 MeanFD+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2023),  
 family="nbinom2")  
summary(model\_local\_adapt\_23\_zi\_MeanFD\_sq)

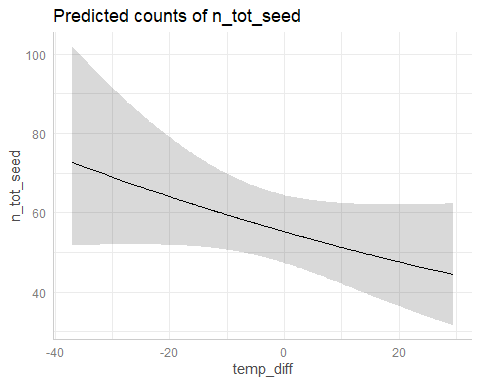
## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp\_diff + I(temp\_diff^2) + MeanFD + (1 | plot)  
## Zero inflation: ~.  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 2218.9 2257.5 -1098.4 2196.9 238   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.02236 0.1495   
## Number of obs: 249, groups: plot, 7  
##   
## Zero-inflation model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.724 1.313   
## Number of obs: 249, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 3.05   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.324e+00 1.248e+00 6.672 2.52e-11 \*\*\*  
## temp\_diff -6.499e-03 5.094e-03 -1.276 0.201998   
## I(temp\_diff^2) 9.634e-05 2.071e-04 0.465 0.641733   
## MeanFD -2.378e-02 6.923e-03 -3.435 0.000593 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.051e+01 3.977e+00 -2.642 0.00825 \*\*  
## temp\_diff 4.667e-03 2.403e-02 0.194 0.84598   
## I(temp\_diff^2) 2.447e-04 9.463e-04 0.259 0.79595   
## MeanFD 4.461e-02 2.098e-02 2.127 0.03345 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_23\_zi\_MeanFD,model\_local\_adapt\_23\_zi\_MeanFD\_sq)

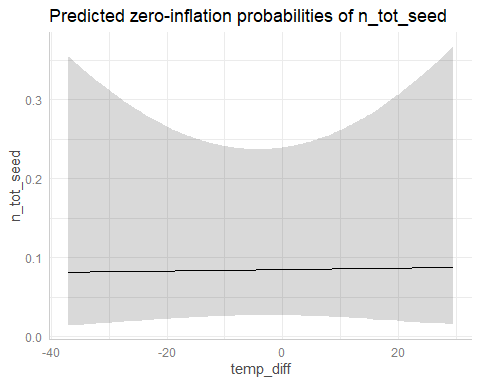
## df AIC  
## model\_local\_adapt\_23\_zi\_MeanFD 9 2215.133  
## model\_local\_adapt\_23\_zi\_MeanFD\_sq 11 2218.856

Model with quadratic effect is worst. Use model without quadratic effect.

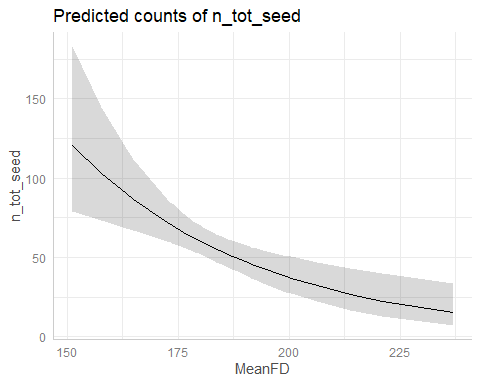
plot(ggpredict(model\_local\_adapt\_23\_zi\_MeanFD,type="count",terms="temp\_diff [all]")) #NS



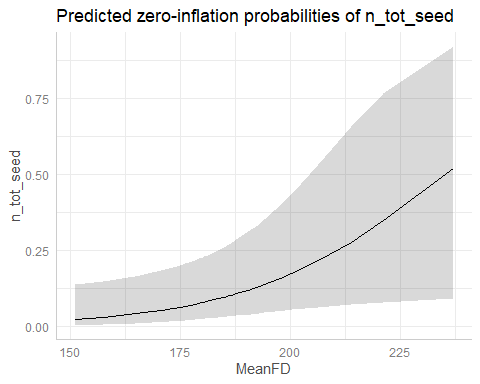
plot(ggpredict(model\_local\_adapt\_23\_zi\_MeanFD,type="zi\_prob",terms="temp\_diff [all]")) #NS



plot(ggpredict(model\_local\_adapt\_23\_zi\_MeanFD,type="count",terms="MeanFD [all]")) #\*



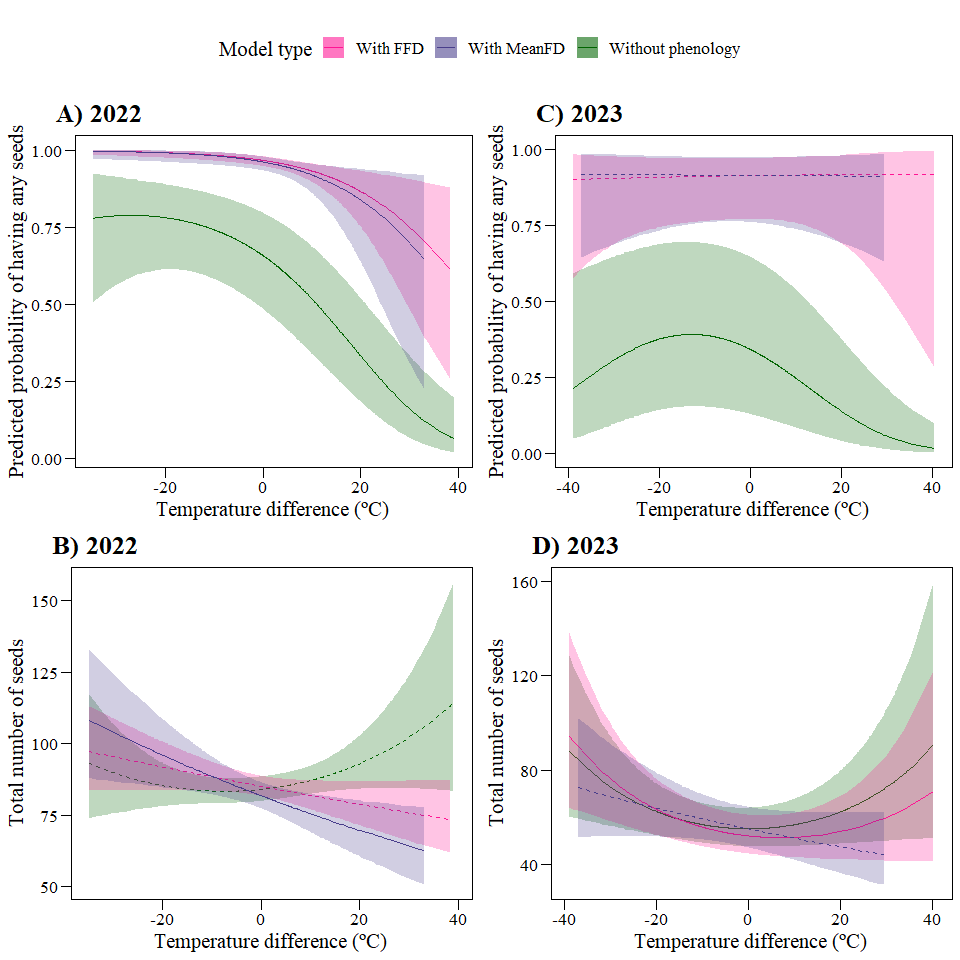
plot(ggpredict(model\_local\_adapt\_23\_zi\_MeanFD,type="zi\_prob",terms="MeanFD [all]")) #\*



### Comparison predictions

legend1<-get\_legend(  
 ggplot()+  
 # Without phenology  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_22\_zi\_sq,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,ymin=1-conf.low,ymax=1-conf.high,  
 fill="Without phenology"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_22\_zi\_sq,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,color="Without phenology"))+  
 # With phenology (FFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_22\_zi\_FFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,ymin=1-conf.low,ymax=1-conf.high,  
 fill="With FFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_22\_zi\_FFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,color="With FFD"))+  
 # With phenology (MeanFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,ymin=1-conf.low,ymax=1-conf.high,  
 fill="With MeanFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,color="With MeanFD"))+  
 scale\_fill\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 scale\_color\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Predicted probability of having any seeds")+  
 theme(legend.position="top")  
)  
prob\_22<-  
 # Probability of having any seeds  
 ggplot()+  
 # Without phenology  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_22\_zi\_sq,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,ymin=1-conf.low,ymax=1-conf.high,  
 fill="Without phenology"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_22\_zi\_sq,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,color="Without phenology"))+  
 # With phenology (FFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_22\_zi\_FFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,ymin=1-conf.low,ymax=1-conf.high,  
 fill="With FFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_22\_zi\_FFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,color="With FFD"))+  
 # With phenology (MeanFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,ymin=1-conf.low,ymax=1-conf.high,  
 fill="With MeanFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,color="With MeanFD"))+  
 scale\_fill\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 scale\_color\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 my\_theme()+xlab("Temperature difference (ºC)")+  
 ylab("Predicted probability of having any seeds")+ggtitle("A) 2022")  
count\_22<-  
 # Total number of seeds  
 ggplot()+  
 # Without phenology  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_22\_zi\_sq,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="Without phenology"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_22\_zi\_sq,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="Without phenology"),linetype="dashed")+  
 # With phenology (FFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_22\_zi\_FFD,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With FFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_22\_zi\_FFD,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With FFD"),linetype="dashed")+  
 # With phenology (MeanFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With MeanFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_22\_zi\_MeanFD,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With MeanFD"))+  
 scale\_fill\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 scale\_color\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 my\_theme()+xlab("Temperature difference (ºC)")+  
 ylab("Total number of seeds")+ggtitle("B) 2022")  
prob\_23<-  
 # Probability of having any seeds  
 ggplot()+  
 # Without phenology  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_23\_zi\_sq,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,ymin=1-conf.low,ymax=1-conf.high,  
 fill="Without phenology"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_23\_zi\_sq,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,color="Without phenology"))+  
 # With phenology (FFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_23\_zi\_FFD\_sq,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,ymin=1-conf.low,ymax=1-conf.high,  
 fill="With FFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_23\_zi\_FFD\_sq,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,color="With FFD"),linetype="dashed")+  
 # With phenology (MeanFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_23\_zi\_MeanFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,ymin=1-conf.low,ymax=1-conf.high,  
 fill="With MeanFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_23\_zi\_MeanFD,  
 type="zi\_prob",terms="temp\_diff [all]"),  
 aes(x=x,y=1-predicted,color="With MeanFD"),linetype="dashed")+  
 scale\_fill\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 scale\_color\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 my\_theme()+xlab("Temperature difference (ºC)")+  
 ylab("Predicted probability of having any seeds")+ggtitle("C) 2023")  
count\_23<-  
 # Total number of seeds  
 ggplot()+  
 # Without phenology  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_23\_zi\_sq,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="Without phenology"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_23\_zi\_sq,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="Without phenology"))+  
 # With phenology (FFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_23\_zi\_FFD\_sq,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With FFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_23\_zi\_FFD\_sq,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With FFD"))+  
 # With phenology (MeanFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_23\_zi\_MeanFD,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With MeanFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_23\_zi\_MeanFD,  
 type="count",terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With MeanFD"),linetype="dashed")+  
 scale\_fill\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 scale\_color\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 my\_theme()+xlab("Temperature difference (ºC)")+  
 ylab("Total number of seeds")+ggtitle("D) 2023")

ggarrange(legend1,  
 grid.arrange(prob\_22,prob\_23,count\_22,count\_23,ncol=2),  
 ncol=1,heights=c(0.1,0.9))

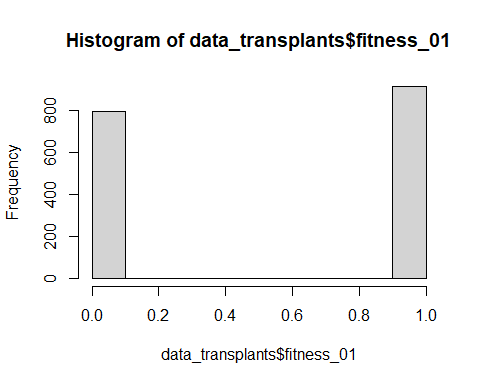


## Fitness 0/1

### Models without phenology

Calculate fitness 0/1:

data\_transplants<-data\_transplants%>%  
 mutate(fitness\_01=ifelse(n\_tot\_seed>0,1,0))  
hist(data\_transplants$fitness\_01)



#### 2022

model\_local\_adapt\_01\_22<-glmmTMB(fitness\_01~temp\_diff+(1|plot),  
 subset(data\_transplants,year==2022),family="binomial")  
model\_local\_adapt\_01\_22\_sq<-glmmTMB(fitness\_01~temp\_diff+I(temp\_diff^2)+  
 (1|plot),  
 subset(data\_transplants,year==2022),  
 family="binomial")  
summary(model\_local\_adapt\_01\_22)

## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 962.3 976.6 -478.2 956.3 846   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.9334 0.9661   
## Number of obs: 849, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.517464 0.353399 1.464 0.143   
## temp\_diff -0.054833 0.008603 -6.374 1.84e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_01\_22\_sq)

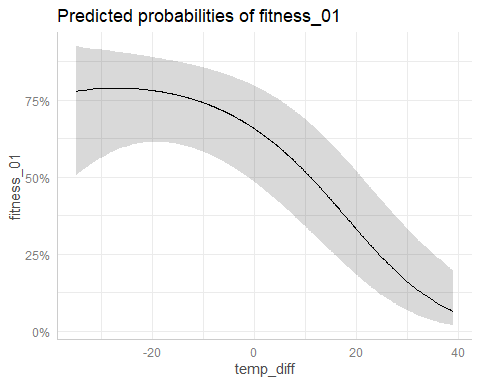
## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 957.7 976.6 -474.8 949.7 845   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.9674 0.9836   
## Number of obs: 849, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.6563801 0.3635482 1.805 0.07100 .   
## temp\_diff -0.0492363 0.0085895 -5.732 9.92e-09 \*\*\*  
## I(temp\_diff^2) -0.0009127 0.0003529 -2.586 0.00971 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_01\_22,model\_local\_adapt\_01\_22\_sq)

## df AIC  
## model\_local\_adapt\_01\_22 3 962.3219  
## model\_local\_adapt\_01\_22\_sq 4 957.6638

Model with quadratic term is best

plot(ggpredict(model\_local\_adapt\_01\_22\_sq,terms="temp\_diff[all]")) #\*



#### 2023

model\_local\_adapt\_01\_23<-glmmTMB(fitness\_01~temp\_diff+(1|plot),  
 subset(data\_transplants,year==2023),family="binomial")  
model\_local\_adapt\_01\_23\_sq<-glmmTMB(fitness\_01~temp\_diff+I(temp\_diff^2)+  
 (1|plot),  
 subset(data\_transplants,year==2023),  
 family="binomial")  
summary(model\_local\_adapt\_01\_23)

## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + (1 | plot)  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 858.4 872.6 -426.2 852.4 830   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 2.832 1.683   
## Number of obs: 833, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.835662 0.617933 -1.352 0.176   
## temp\_diff -0.032489 0.007937 -4.093 4.25e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_01\_23\_sq)

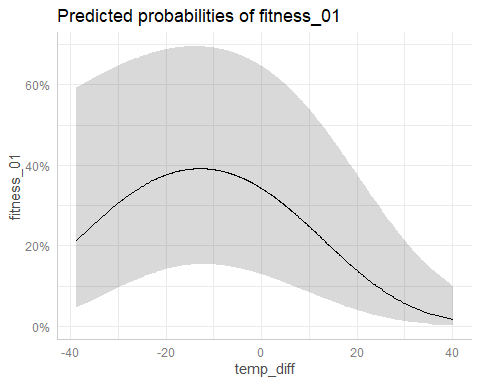
## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 847.6 866.5 -419.8 839.6 829   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 3.032 1.741   
## Number of obs: 833, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.6522428 0.6406610 -1.018 0.308641   
## temp\_diff -0.0329356 0.0085309 -3.861 0.000113 \*\*\*  
## I(temp\_diff^2) -0.0012800 0.0003757 -3.407 0.000656 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_01\_23,model\_local\_adapt\_01\_23\_sq)

## df AIC  
## model\_local\_adapt\_01\_23 3 858.4443  
## model\_local\_adapt\_01\_23\_sq 4 847.6465

Model with quadratic term is best

plot(ggpredict(model\_local\_adapt\_01\_23\_sq,terms="temp\_diff[all]")) #\*



### Models with phenology

#### FFD

##### 2022

model\_local\_adapt\_01\_22\_FFD<-glmmTMB(fitness\_01~temp\_diff+FFD\_corr+(1|plot),  
 subset(data\_transplants,year==2022),family="binomial")  
model\_local\_adapt\_01\_22\_FFD\_sq<-glmmTMB(fitness\_01~temp\_diff+I(temp\_diff^2)+  
 FFD\_corr+(1|plot),  
 subset(data\_transplants,year==2022),  
 family="binomial")  
summary(model\_local\_adapt\_01\_22\_FFD)

## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + FFD\_corr + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 197.1 214.7 -94.6 189.1 598   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 2.83e-09 5.32e-05  
## Number of obs: 602, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 9.50070 2.27456 4.177 2.95e-05 \*\*\*  
## temp\_diff -0.07649 0.02306 -3.317 0.00091 \*\*\*  
## FFD\_corr -0.03619 0.01276 -2.836 0.00457 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_01\_22\_FFD\_sq)

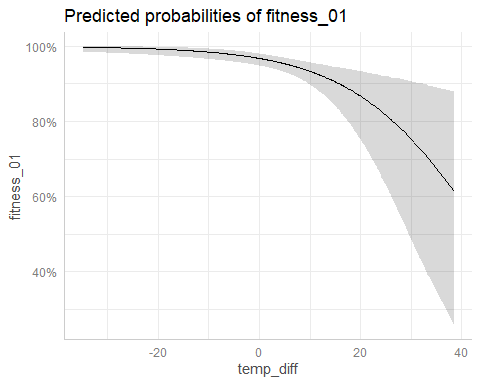
## Family: binomial ( logit )  
## Formula:   
## fitness\_01 ~ temp\_diff + I(temp\_diff^2) + FFD\_corr + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 199.0 221.0 -94.5 189.0 597   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 2.804e-09 5.295e-05  
## Number of obs: 602, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 9.4470395 2.2739985 4.154 3.26e-05 \*\*\*  
## temp\_diff -0.0809677 0.0280778 -2.884 0.00393 \*\*   
## I(temp\_diff^2) 0.0003321 0.0010682 0.311 0.75584   
## FFD\_corr -0.0360793 0.0127235 -2.836 0.00457 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_01\_22\_FFD,model\_local\_adapt\_01\_22\_FFD\_sq)

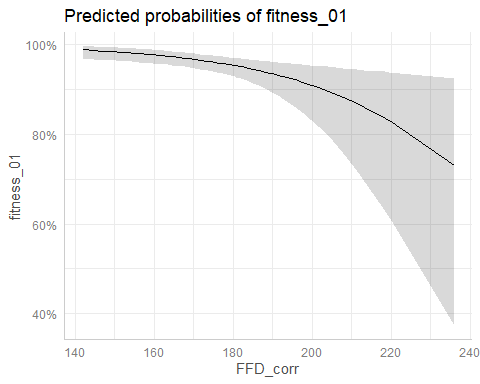
## df AIC  
## model\_local\_adapt\_01\_22\_FFD 4 197.1213  
## model\_local\_adapt\_01\_22\_FFD\_sq 5 199.0253

Model with quadratic term is worst. Use model without quadratic term.

plot(ggpredict(model\_local\_adapt\_01\_22\_FFD,terms="temp\_diff [all]")) #\*



plot(ggpredict(model\_local\_adapt\_01\_22\_FFD,terms="FFD\_corr [all]")) #\*



##### 2023

model\_local\_adapt\_01\_23\_FFD<-glmmTMB(fitness\_01~temp\_diff+FFD\_corr+(1|plot),  
 subset(data\_transplants,year==2023),family="binomial")  
model\_local\_adapt\_01\_23\_FFD\_sq<-glmmTMB(fitness\_01~temp\_diff+I(temp\_diff^2)+  
 FFD\_corr+(1|plot),  
 subset(data\_transplants,year==2023),  
 family="binomial")  
summary(model\_local\_adapt\_01\_23\_FFD)

## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + FFD\_corr + (1 | plot)  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 317.5 333.5 -154.7 309.5 402   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.803 1.343   
## Number of obs: 406, groups: plot, 7  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 9.791296 3.217438 3.043 0.00234 \*\*  
## temp\_diff 0.003224 0.015612 0.206 0.83638   
## FFD\_corr -0.041736 0.017378 -2.402 0.01632 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_01\_23\_FFD\_sq)

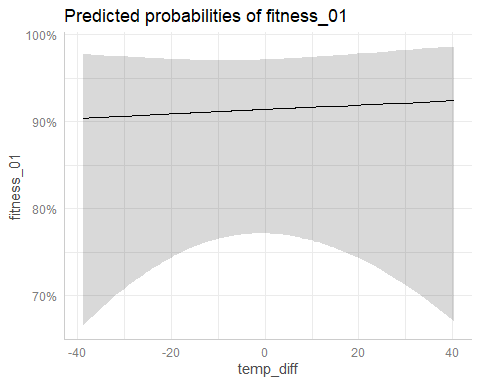
## Family: binomial ( logit )  
## Formula:   
## fitness\_01 ~ temp\_diff + I(temp\_diff^2) + FFD\_corr + (1 | plot)  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 319.5 339.5 -154.7 309.5 401   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.799 1.341   
## Number of obs: 406, groups: plot, 7  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 9.787e+00 3.217e+00 3.042 0.00235 \*\*  
## temp\_diff 2.684e-03 1.859e-02 0.144 0.88518   
## I(temp\_diff^2) -3.425e-05 6.686e-04 -0.051 0.95915   
## FFD\_corr -4.170e-02 1.739e-02 -2.398 0.01646 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_01\_23\_FFD,model\_local\_adapt\_01\_23\_FFD\_sq)

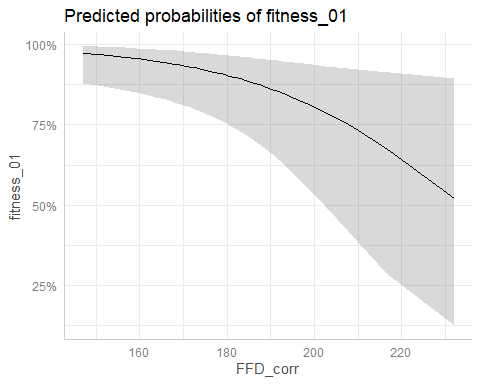
## df AIC  
## model\_local\_adapt\_01\_23\_FFD 4 317.4569  
## model\_local\_adapt\_01\_23\_FFD\_sq 5 319.4542

Model with quadratic term is worst. Use model without quadratic term.

plot(ggpredict(model\_local\_adapt\_01\_23\_FFD,terms="temp\_diff [all]")) #NS



plot(ggpredict(model\_local\_adapt\_01\_23\_FFD,terms="FFD\_corr [all]")) #\*



#### MeanFD

##### 2022

model\_local\_adapt\_01\_22\_MeanFD<-glmmTMB(fitness\_01~temp\_diff+MeanFD+(1|plot),  
 subset(data\_transplants,year==2022),family="binomial")  
model\_local\_adapt\_01\_22\_MeanFD\_sq<-glmmTMB(fitness\_01~temp\_diff+I(temp\_diff^2)+  
 MeanFD+(1|plot),  
 subset(data\_transplants,year==2022),  
 family="binomial")  
summary(model\_local\_adapt\_01\_22\_MeanFD)

## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + MeanFD + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 131.1 146.6 -61.5 123.1 359   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 2.986e-09 5.464e-05  
## Number of obs: 363, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 10.20077 3.07007 3.323 0.000892 \*\*\*  
## temp\_diff -0.07997 0.03163 -2.528 0.011469 \*   
## MeanFD -0.03931 0.01651 -2.381 0.017254 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_01\_22\_MeanFD\_sq)

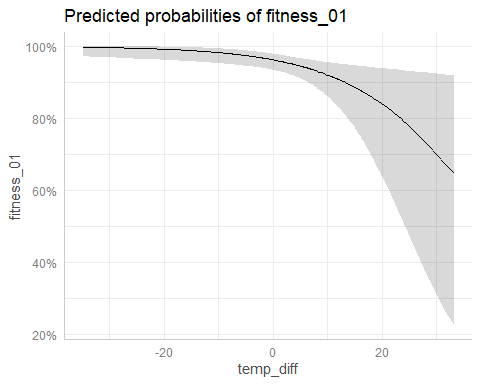
## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + I(temp\_diff^2) + MeanFD + (1 | plot)  
## Data: subset(data\_transplants, year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 133.0 152.5 -61.5 123.0 358   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 1.963e-09 4.431e-05  
## Number of obs: 363, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 10.0581795 3.0974608 3.247 0.00117 \*\*  
## temp\_diff -0.0833042 0.0352149 -2.366 0.01800 \*   
## I(temp\_diff^2) 0.0004216 0.0015529 0.272 0.78599   
## MeanFD -0.0387592 0.0165468 -2.342 0.01916 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_01\_22\_MeanFD,model\_local\_adapt\_01\_22\_MeanFD\_sq)

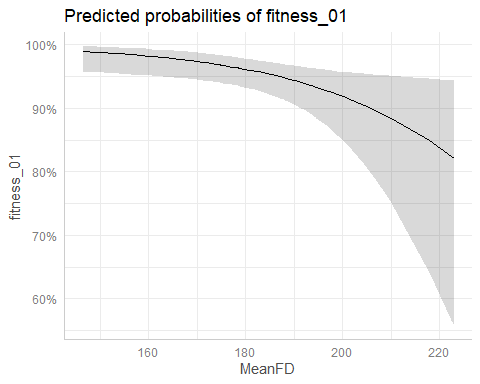
## df AIC  
## model\_local\_adapt\_01\_22\_MeanFD 4 131.0662  
## model\_local\_adapt\_01\_22\_MeanFD\_sq 5 132.9925

Model with quadratic term is worst. Use model without quadratic term.

plot(ggpredict(model\_local\_adapt\_01\_22\_MeanFD,terms="temp\_diff [all]")) #\*



plot(ggpredict(model\_local\_adapt\_01\_22\_MeanFD,terms="MeanFD [all]")) #\*



##### 2023

model\_local\_adapt\_01\_23\_MeanFD<-glmmTMB(fitness\_01~temp\_diff+MeanFD+(1|plot),  
 subset(data\_transplants,year==2023),family="binomial")  
model\_local\_adapt\_01\_23\_MeanFD\_sq<-glmmTMB(fitness\_01~temp\_diff+I(temp\_diff^2)+  
 MeanFD+(1|plot),  
 subset(data\_transplants,year==2023),  
 family="binomial")  
summary(model\_local\_adapt\_01\_23\_MeanFD)

## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + MeanFD + (1 | plot)  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 197.7 211.7 -94.8 189.7 245   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.742 1.32   
## Number of obs: 249, groups: plot, 7  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 10.574097 3.950248 2.677 0.00743 \*\*  
## temp\_diff -0.001296 0.020623 -0.063 0.94988   
## MeanFD -0.045108 0.020800 -2.169 0.03011 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_local\_adapt\_01\_23\_MeanFD\_sq)

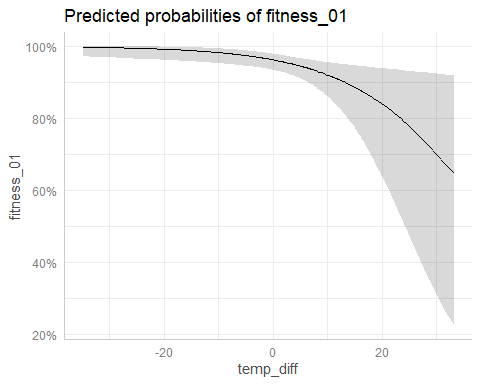
## Family: binomial ( logit )  
## Formula: fitness\_01 ~ temp\_diff + I(temp\_diff^2) + MeanFD + (1 | plot)  
## Data: subset(data\_transplants, year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 199.6 217.2 -94.8 189.6 244   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.721 1.312   
## Number of obs: 249, groups: plot, 7  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 10.5781763 3.9431413 2.683 0.0073 \*\*  
## temp\_diff -0.0047457 0.0239969 -0.198 0.8432   
## I(temp\_diff^2) -0.0002442 0.0009456 -0.258 0.7962   
## MeanFD -0.0450187 0.0207690 -2.168 0.0302 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(model\_local\_adapt\_01\_23\_MeanFD,model\_local\_adapt\_01\_23\_MeanFD\_sq)

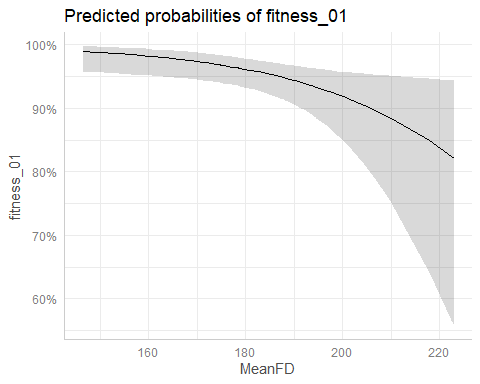
## df AIC  
## model\_local\_adapt\_01\_23\_MeanFD 4 197.6601  
## model\_local\_adapt\_01\_23\_MeanFD\_sq 5 199.5929

Model with quadratic term is worst. Use model without quadratic term.

plot(ggpredict(model\_local\_adapt\_01\_22\_MeanFD,terms="temp\_diff [all]")) #NS



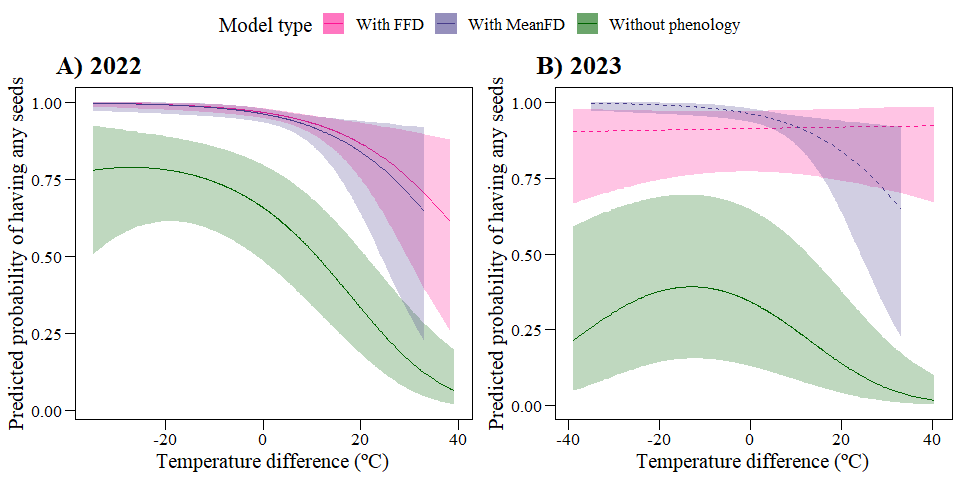
plot(ggpredict(model\_local\_adapt\_01\_22\_MeanFD,terms="MeanFD [all]")) #\*



### Comparison predictions

legend2<-get\_legend(  
 ggplot()+  
 # Without phenology  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_01\_22\_sq,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="Without phenology"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_01\_22\_sq,terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="Without phenology"))+  
 # With phenology (FFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_01\_22\_FFD,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With FFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_01\_22\_FFD,terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With FFD"))+  
 # With phenology (MeanFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_01\_22\_MeanFD,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With MeanFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_01\_22\_MeanFD,terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With MeanFD"))+  
 scale\_fill\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 scale\_color\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Predicted probability of having any seeds")+ggtitle("A) 2022")+  
 theme(legend.position="top")  
)  
prob\_22\_01<-  
 # Probability of having any seeds  
 ggplot()+  
 # Without phenology  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_01\_22\_sq,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="Without phenology"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_01\_22\_sq,terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="Without phenology"))+  
 # With phenology (FFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_01\_22\_FFD,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With FFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_01\_22\_FFD,terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With FFD"))+  
 # With phenology (MeanFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_01\_22\_MeanFD,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With MeanFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_01\_22\_MeanFD,terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With MeanFD"))+  
 scale\_fill\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 scale\_color\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 my\_theme()+xlab("Temperature difference (ºC)")+  
 ylab("Predicted probability of having any seeds")+ggtitle("A) 2022")  
prob\_23\_01<-  
 # Probability of having any seeds  
 ggplot()+  
 # Without phenology  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_01\_23\_sq,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="Without phenology"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_01\_23\_sq,terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="Without phenology"))+  
 # With phenology (FFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_01\_23\_FFD,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With FFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_01\_23\_FFD,terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With FFD"),linetype="dashed")+  
 # With phenology (MeanFD)  
 geom\_ribbon(data=ggpredict(model\_local\_adapt\_01\_22\_MeanFD,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,  
 fill="With MeanFD"),alpha=0.25)+  
 geom\_line(data=ggpredict(model\_local\_adapt\_01\_22\_MeanFD,  
 terms="temp\_diff [all]"),  
 aes(x=x,y=predicted,color="With MeanFD"),linetype="dashed")+  
 scale\_fill\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 scale\_color\_manual(values=c("Without phenology"="darkgreen",  
 "With FFD"="deeppink",  
 "With MeanFD"="darkslateblue"),  
 name="Model type") +  
 my\_theme()+xlab("Temperature difference (ºC)")+  
 ylab("Predicted probability of having any seeds")+ggtitle("B) 2023")

ggarrange(legend2,  
 grid.arrange(prob\_22\_01,prob\_23\_01,ncol=2),  
 ncol=1,heights=c(0.1,0.9))



## Path analysis (REVISE)

[Possibly do a small path-model with direct and indirect, via phenology, effects of origin on fitness?]

### Fitness

#### FFD

data\_transplants$temp\_diff\_square<-(data\_transplants$temp\_diff)^2

path\_local\_adapt\_22\_FFD<-psem(glmmTMB(FFD\_corr~temp\_diff+(1|plot),  
 subset(data\_transplants,year==2022)),  
 glmmTMB(n\_tot\_seed~temp\_diff+temp\_diff\_square+  
 FFD\_corr+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2022),  
 family="nbinom2"))  
path\_local\_adapt\_23\_FFD<-psem(glmmTMB(FFD\_corr~temp\_diff+(1|plot),  
 subset(data\_transplants,year==2023)),  
 glmmTMB(n\_tot\_seed~temp\_diff+temp\_diff\_square+  
 FFD\_corr+(1|plot),  
 ziformula=~.,  
 subset(data\_transplants,year==2023),  
 family="nbinom2"))

coefs(path\_local\_adapt\_22\_FFD)

## Response Predictor Estimate Std.Error DF Crit.Value P.Value  
## 1 FFD\_corr temp\_diff -0.2131 0.0346 603 -6.157 0  
## 2 temp\_diff -0.0036 - 602 - -  
## 3 temp\_diff\_square 0.0001 - 602 - -  
## 4 FFD\_corr -0.0060 - 602 - -  
## Std.Estimate   
## 1 -0.135 \*\*\*  
## 2 - -  
## 3 - -  
## 4 - -

coefs(path\_local\_adapt\_23\_FFD)

## Response Predictor Estimate Std.Error DF Crit.Value P.Value  
## 1 FFD\_corr temp\_diff -0.0263 0.0422 406 -0.6235 0.5329  
## 2 n\_tot\_seed temp\_diff -0.0040 0.0035 406 -1.1467 0.2515  
## 3 n\_tot\_seed temp\_diff\_square 0.0003 0.0001 406 2.3917 0.0168  
## 4 n\_tot\_seed FFD\_corr -0.0233 0.0052 406 -4.4876 0.0000  
## Std.Estimate   
## 1 -0.0261   
## 2 -   
## 3 - \*  
## 4 - \*\*\*

### Fitness 0/1

#### FFD

path\_local\_adapt\_01\_22\_FFD<-psem(glmmTMB(FFD\_corr~temp\_diff+(1|plot),  
 subset(data\_transplants,year==2022)),  
 glmmTMB(fitness\_01~temp\_diff+  
 FFD\_corr+(1|plot),  
 subset(data\_transplants,year==2022),  
 family="binomial"))  
path\_local\_adapt\_01\_23\_FFD<-psem(glmmTMB(FFD\_corr~temp\_diff+(1|plot),  
 subset(data\_transplants,year==2023)),   
 glmmTMB(fitness\_01~temp\_diff+  
 FFD\_corr+(1|plot),  
 subset(data\_transplants,year==2023),  
 family="binomial"))

coefs(path\_local\_adapt\_01\_22\_FFD)

## Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate  
## 1 FFD\_corr temp\_diff -0.2131 0.0346 603 -6.157 0.0000 -0.1350  
## 2 fitness\_01 temp\_diff -0.0765 0.0231 602 -3.317 0.0009 -0.4715  
## 3 fitness\_01 FFD\_corr -0.0362 0.0128 602 -2.836 0.0046 -0.3522  
##   
## 1 \*\*\*  
## 2 \*\*\*  
## 3 \*\*

coefs(path\_local\_adapt\_01\_23\_FFD)

## Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate   
## 1 FFD\_corr temp\_diff -0.0263 0.0422 406 -0.6235 0.5329 -0.0261   
## 2 fitness\_01 temp\_diff 0.0032 0.0156 406 0.2065 0.8364 0.0156   
## 3 fitness\_01 FFD\_corr -0.0417 0.0174 406 -2.4017 0.0163 -0.2034 \*

# Survival

data\_transplants\_22<-data\_transplants%>%  
 filter(year==2022)%>%  
 mutate(surv=ifelse(is.na(F\_NF\_A),0,1))  
not\_surv\_22<-data\_transplants\_22%>%filter(surv==0)  
data\_transplants\_23<-data\_transplants%>%  
 filter(year==2023)%>%  
 mutate(surv=ifelse(is.na(F\_NF\_A),0,1))%>%  
 # If the plant did not survive in 2022, set NA for surv in 2023  
 mutate(surv=ifelse(unique\_id %in% not\_surv\_22$unique\_id,NA,surv))%>%  
 # Correct errors: these below are in not\_surv\_22,  
 # but actually there is data for 2023!  
 mutate(surv=ifelse(unique\_id %in% c("Hot\_10\_a","Hot\_30\_a","Hot\_45\_a",  
 "Hot\_9\_d","Hot\_88\_a","Hot\_125\_b"),1,surv))  
data\_transplants<-rbind(data\_transplants\_22,data\_transplants\_23)  
with(subset(data\_transplants,year==2022),table(surv))

## surv  
## 0 1   
## 50 808

with(subset(data\_transplants,year==2023),table(surv))

## surv  
## 0 1   
## 284 530

Of 858 plants planted in 2021, 808 survived in 2022. Of the 808 that survived in 2022, 530 survived in 2023

## Models

Effect of pollination type?

mod\_surv\_poll\_22<-glmmTMB(surv~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022),  
 family="binomial")  
mod\_surv\_poll\_23<-glmmTMB(surv~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023),  
 family="binomial")  
summary(mod\_surv\_poll\_22)

## Family: binomial ( logit )  
## Formula: surv ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 360.2 374.5 -177.1 354.2 855   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.907 1.381   
## Number of obs: 858, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.61156 0.64667 5.585 2.34e-08 \*\*\*  
## poll\_typeSelfpollinated 0.07603 0.33257 0.229 0.819   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_surv\_poll\_23)

## Family: binomial ( logit )  
## Formula: surv ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 732.5 746.6 -363.2 726.5 811   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 4.149 2.037   
## Number of obs: 814, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.3629 0.7436 1.833 0.0668 .  
## poll\_typeSelfpollinated -0.3468 0.2012 -1.724 0.0847 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

No effect of pollination type, so we do not include it in further models.

mod\_surv\_22<-glmmTMB(surv~temp+temp\_diff+ I(temp\_diff^2)+  
 (1|plot),  
 data\_transplants%>%filter(year==2022),  
 family="binomial")  
mod\_surv\_23<-glmmTMB(surv~temp+temp\_diff+I(temp\_diff^2)+(1|plot),  
 data\_transplants%>%filter(year==2023),  
 family="binomial")  
summary(mod\_surv\_22)

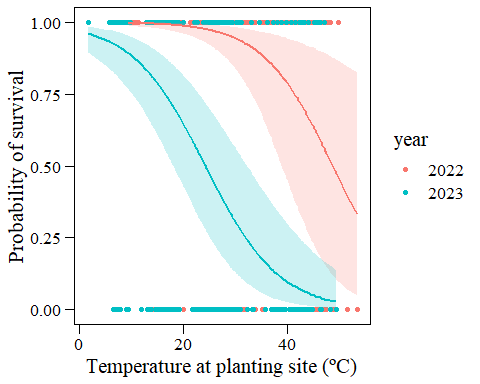
## Family: binomial ( logit )  
## Formula: surv ~ temp + temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 261.6 285.4 -125.8 251.6 847   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.9732 0.9865   
## Number of obs: 852, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 7.3335652 1.0284433 7.131 9.98e-13 \*\*\*  
## temp -0.1503982 0.0349339 -4.305 1.67e-05 \*\*\*  
## temp\_diff 0.0136238 0.0239998 0.568 0.570   
## I(temp\_diff^2) -0.0012299 0.0007631 -1.612 0.107   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_surv\_23)

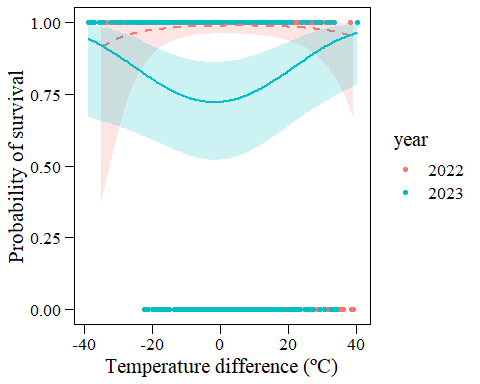
## Family: binomial ( logit )  
## Formula: surv ~ temp + temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 649.9 673.3 -319.9 639.9 793   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.307 1.143   
## Number of obs: 798, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.4560225 0.5750683 6.010 1.86e-09 \*\*\*  
## temp -0.1418874 0.0226769 -6.257 3.93e-10 \*\*\*  
## temp\_diff 0.0045008 0.0126444 0.356 0.7219   
## I(temp\_diff^2) 0.0013426 0.0005791 2.319 0.0204 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Predictions

ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp,y=surv,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_surv\_22,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_surv\_22,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_surv\_23,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_surv\_23,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1)+  
 my\_theme\_legend()+xlab("Temperature at planting site (ºC)")+  
 ylab("Probability of survival")

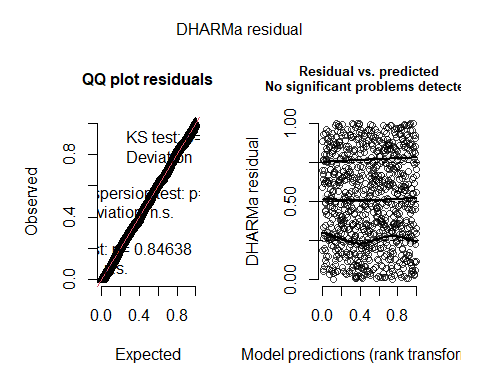


ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp\_diff,y=surv,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_surv\_22,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_surv\_22,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1,linetype="dashed")+  
 geom\_ribbon(data=ggpredict(mod\_surv\_23,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_surv\_23,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1)+  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Probability of survival")

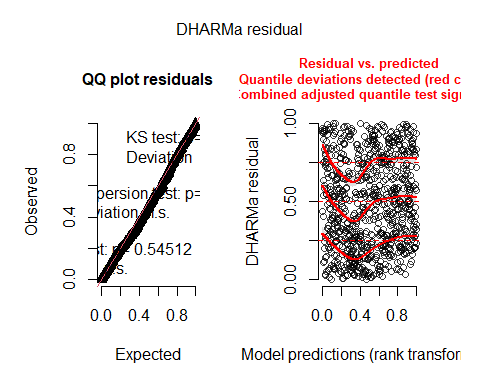


## Diagnostics

plot(simulateResiduals(mod\_surv\_22))

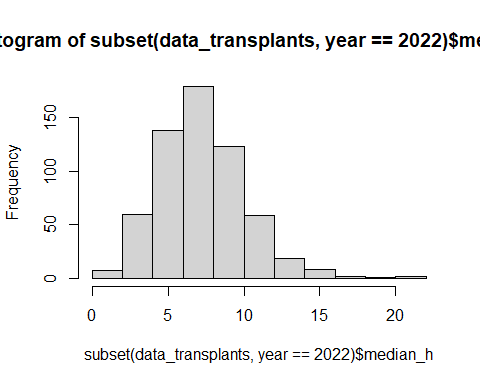


plot(simulateResiduals(mod\_surv\_23))

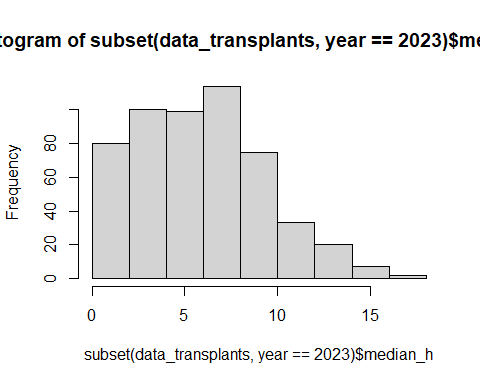


# Median height of flowering stem

hist(subset(data\_transplants,year==2022)$median\_h)



hist(subset(data\_transplants,year==2023)$median\_h)



Effect of pollination type?

mod\_median\_h\_poll\_22<-glmmTMB(median\_h~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022))  
mod\_median\_h\_poll\_23<-glmmTMB(median\_h~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023))  
summary(mod\_median\_h\_poll\_22)

## Family: gaussian ( identity )  
## Formula: median\_h ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 2935.2 2952.8 -1463.6 2927.2 594   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.625 0.7906   
## Residual 7.642 2.7644   
## Number of obs: 598, groups: plot, 8  
##   
## Dispersion estimate for gaussian family (sigma^2): 7.64   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 7.93909 0.32401 24.503 <2e-16 \*\*\*  
## poll\_typeSelfpollinated -0.06672 0.25404 -0.263 0.793   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_median\_h\_poll\_23)

## Family: gaussian ( identity )  
## Formula: median\_h ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 2717.7 2734.8 -1354.8 2709.7 526   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 3.267 1.807   
## Residual 9.306 3.051   
## Number of obs: 530, groups: plot, 8  
##   
## Dispersion estimate for gaussian family (sigma^2): 9.31   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.7618 0.6705 8.593 < 2e-16 \*\*\*  
## poll\_typeSelfpollinated -1.2533 0.2935 -4.269 1.96e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Significant effect of poll\_type in 2023 –> include in further models (so far as fixed effect).

mod\_median\_h\_22<-glmmTMB(median\_h~temp+temp\_diff+I(temp\_diff^2)+   
 (1|plot),data\_transplants%>%filter(year==2022))  
mod\_median\_h\_23\_pollf<-glmmTMB(median\_h~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023))  
summary(mod\_median\_h\_22)

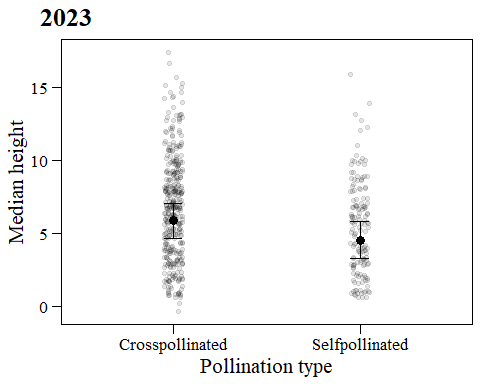
## Family: gaussian ( identity )  
## Formula: median\_h ~ temp + temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 2934.8 2961.1 -1461.4 2922.8 592   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.9667 0.9832   
## Residual 7.5484 2.7474   
## Number of obs: 598, groups: plot, 8  
##   
## Dispersion estimate for gaussian family (sigma^2): 7.55   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 7.2215354 0.6044224 11.948 <2e-16 \*\*\*  
## temp 0.0468034 0.0276514 1.693 0.0905 .   
## temp\_diff -0.0071398 0.0142020 -0.503 0.6152   
## I(temp\_diff^2) -0.0010382 0.0005947 -1.746 0.0808 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_median\_h\_23\_pollf)

## Family: gaussian ( identity )  
## Formula:   
## median\_h ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 2663.4 2693.1 -1324.7 2649.4 514   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 2.628 1.621   
## Residual 9.073 3.012   
## Number of obs: 521, groups: plot, 8  
##   
## Dispersion estimate for gaussian family (sigma^2): 9.07   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 6.694e+00 8.730e-01 7.668 1.75e-14 \*\*\*  
## temp -5.595e-02 4.014e-02 -1.394 0.163   
## temp\_diff 8.926e-03 2.045e-02 0.436 0.662   
## I(temp\_diff^2) -6.979e-05 7.293e-04 -0.096 0.924   
## poll\_typeSelfpollinated -1.342e+00 2.945e-01 -4.558 5.16e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

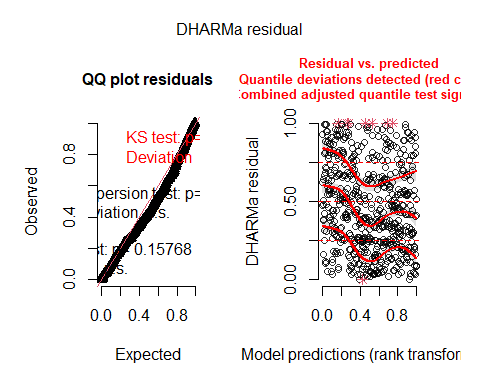
## Predictions

ggplot()+  
 geom\_jitter(data=data\_transplants%>%filter(year==2023),  
 aes(x=poll\_type,y=median\_h),width=0.05,alpha=0.1)+  
 geom\_point(data=ggpredict(mod\_median\_h\_23\_pollf,terms=c("poll\_type")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_median\_h\_23\_pollf,terms=c("poll\_type")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Median height")+ggtitle("2023")

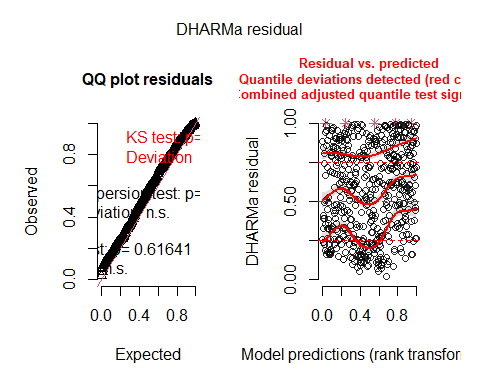


## Diagnostics

plot(simulateResiduals(mod\_median\_h\_22))



plot(simulateResiduals(mod\_median\_h\_23\_pollf))



# Probability of flowering

data\_transplants<-data\_transplants%>%  
 mutate(flowering=ifelse(F\_NF\_A=="F",1,ifelse(F\_NF\_A=="NF",0,NA)))  
with(data\_transplants,table(year,flowering))

## flowering  
## year 0 1  
## 2022 204 603  
## 2023 119 411

Of 808 plants that survived in 2022, 603 flowered and 204 did not flower (and 1 was aborted, not used here). 4 were grazed before they peaked, they are included as flowering here. Of 530 plants that survived in 2023, 411 flowered and 119 did not flower.

Does pollination type have an effect on probability of flowering?

mod\_flowering\_poll\_22<-glmmTMB(flowering~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022),  
 family="binomial")  
mod\_flowering\_poll\_23<-glmmTMB(flowering~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023),  
 family="binomial")  
summary(mod\_flowering\_poll\_22)

## Family: binomial ( logit )  
## Formula: flowering ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 804.1 818.2 -399.1 798.1 804   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.141 1.068   
## Number of obs: 807, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.1197 0.3959 2.828 0.004677 \*\*   
## poll\_typeSelfpollinated -0.6393 0.1856 -3.445 0.000571 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_flowering\_poll\_23)

## Family: binomial ( logit )  
## Formula: flowering ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 484.9 497.7 -239.5 478.9 527   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 2.487 1.577   
## Number of obs: 530, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.8920 0.5914 1.508 0.13148   
## poll\_typeSelfpollinated -0.6496 0.2492 -2.607 0.00914 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

nobs(mod\_flowering\_poll\_22)

## [1] 807

nobs(mod\_flowering\_poll\_23)

## [1] 530

Significant effect of poll\_type in both years –> include in further models (for far as fixed).

mod\_flowering\_22\_pollf<-glmmTMB(flowering~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022),  
 family="binomial")  
mod\_flowering\_23\_pollf<-glmmTMB(flowering~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023),  
 family="binomial")  
summary(mod\_flowering\_22\_pollf)

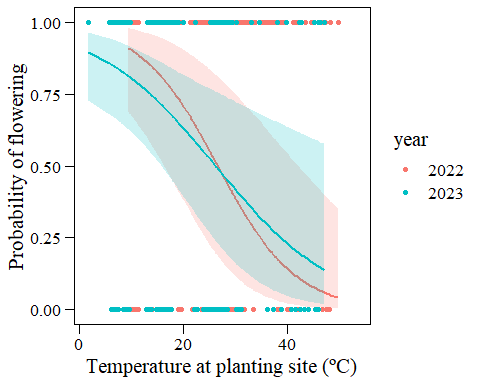
## Family: binomial ( logit )  
## Formula:   
## flowering ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 779.6 807.8 -383.8 767.6 801   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 3.712 1.927   
## Number of obs: 807, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.5767894 0.9589731 3.730 0.000192 \*\*\*  
## temp -0.1342200 0.0346770 -3.871 0.000109 \*\*\*  
## temp\_diff -0.0117860 0.0115125 -1.024 0.305950   
## I(temp\_diff^2) 0.0009116 0.0005099 1.788 0.073782 .   
## poll\_typeSelfpollinated -0.7289754 0.1918542 -3.800 0.000145 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_flowering\_23\_pollf)

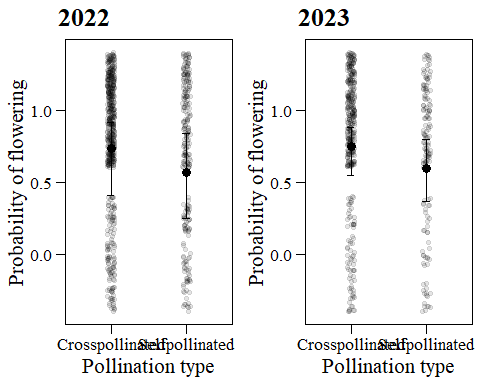
## Family: binomial ( logit )  
## Formula:   
## flowering ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 468.9 494.4 -228.5 456.9 515   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 1.372 1.171   
## Number of obs: 521, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.3734377 0.6655574 3.566 0.000362 \*\*\*  
## temp -0.0874544 0.0300863 -2.907 0.003652 \*\*   
## temp\_diff 0.0146293 0.0160252 0.913 0.361297   
## I(temp\_diff^2) 0.0001930 0.0005922 0.326 0.744552   
## poll\_typeSelfpollinated -0.6950523 0.2572819 -2.702 0.006902 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Predictions

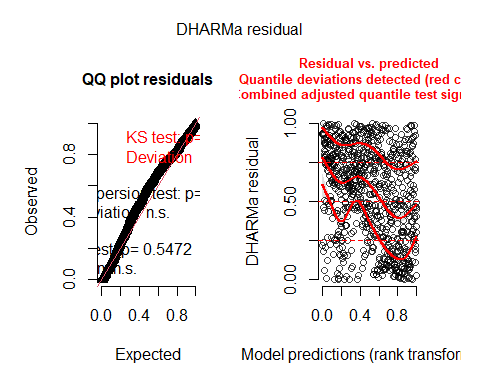
ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp,y=flowering,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_flowering\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_flowering\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_flowering\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_flowering\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1)+  
 my\_theme\_legend()+xlab("Temperature at planting site (ºC)")+  
 ylab("Probability of flowering")



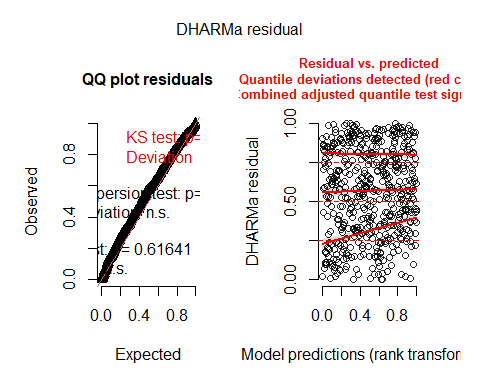
grid.arrange(  
 ggplot()+  
 geom\_jitter(data=data\_transplants%>%filter(year==2022),  
 aes(x=poll\_type,y=flowering),width=0.05,alpha=0.1)+  
 geom\_point(data=ggpredict(mod\_flowering\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_flowering\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Probability of flowering")+  
 ggtitle("2022"),  
 ggplot()+  
 geom\_jitter(data=data\_transplants%>%filter(year==2023),  
 aes(x=poll\_type,y=flowering),width=0.05,alpha=0.1)+  
 geom\_point(data=ggpredict(mod\_flowering\_23\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_flowering\_23\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Probability of flowering")+  
 ggtitle("2023"),  
ncol=2)

 ## Diagnostics

plot(simulateResiduals(mod\_flowering\_22\_pollf))



plot(simulateResiduals(mod\_flowering\_23\_pollf))



# Number of flowers

Correct tot\_fl\_bodies to tot\_fl\_bodies\_corr, to be at least as high as n\_tot\_fr.

nrow(data\_transplants%>%filter(n\_tot\_fr>tot\_fl\_bodies))

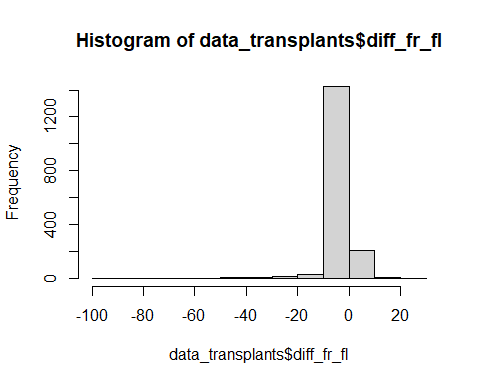
## [1] 221

221 cases where total number of fruits is larger than total number of flowering bodies.

data\_transplants<-data\_transplants%>%  
 mutate(tot\_fl\_bodies\_corr=ifelse(n\_tot\_fr>tot\_fl\_bodies,n\_tot\_fr,  
 tot\_fl\_bodies),  
 diff\_fr\_fl=n\_tot\_fr-tot\_fl\_bodies)

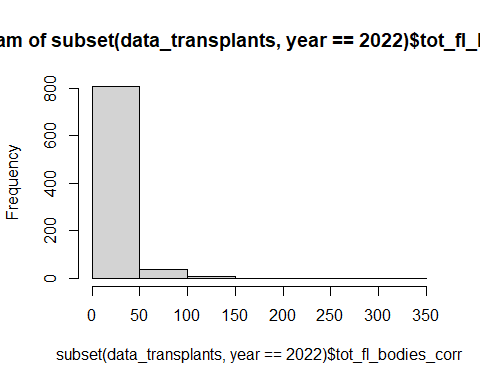
Histogram of differences:

hist(data\_transplants$diff\_fr\_fl)

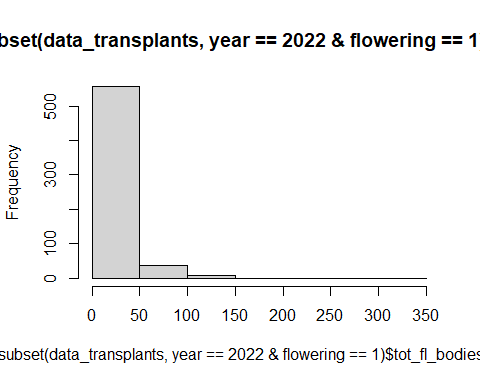


Negative differences: more flowers than fruits - OK Positive differences: more fruits than flowers - not OK –> corrected

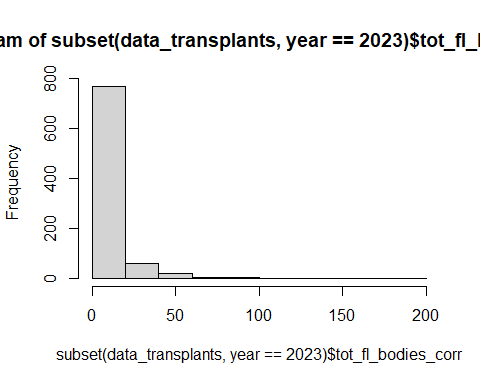
hist(subset(data\_transplants,year==2022)$tot\_fl\_bodies\_corr)



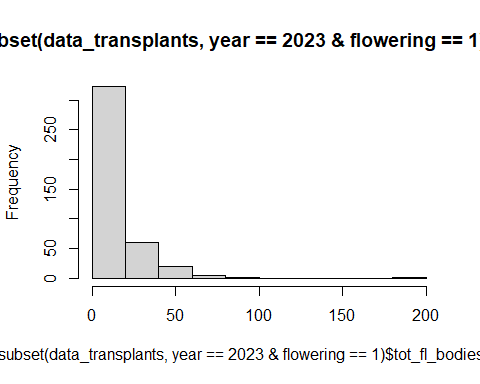
hist(subset(data\_transplants,year==2022&flowering==1)$tot\_fl\_bodies\_corr)



hist(subset(data\_transplants,year==2023)$tot\_fl\_bodies\_corr)



hist(subset(data\_transplants,year==2023&flowering==1)$tot\_fl\_bodies\_corr)



Models for number of flowers in those that flowered (flowering=1).

Does pollination type have an effect on number of flowers?

mod\_nfl\_poll\_22<-glmmTMB(tot\_fl\_bodies\_corr~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&flowering==1),  
 family="nbinom2") # nbinom2 lower AIC than poisson  
mod\_nfl\_poll\_23<-glmmTMB(tot\_fl\_bodies\_corr~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023&flowering==1),  
 family="nbinom2") # nbinom2 lower AIC than poisson  
summary(mod\_nfl\_poll\_22)

## Family: nbinom2 ( log )  
## Formula: tot\_fl\_bodies\_corr ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 4692.3 4709.9 -2342.1 4684.3 598   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.1512 0.3889   
## Number of obs: 602, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 1.35   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.78969 0.14914 18.705 < 2e-16 \*\*\*  
## poll\_typeSelfpollinated -0.28999 0.08259 -3.511 0.000446 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_nfl\_poll\_23)

## Family: nbinom2 ( log )  
## Formula: tot\_fl\_bodies\_corr ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 2942.8 2958.9 -1467.4 2934.8 407   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.4261 0.6528   
## Number of obs: 411, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 1.61   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.37403 0.25631 9.262 < 2e-16 \*\*\*  
## poll\_typeSelfpollinated -0.36809 0.09829 -3.745 0.00018 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

nobs(mod\_nfl\_poll\_22)

## [1] 602

nobs(mod\_nfl\_poll\_23)

## [1] 411

Significant effect of poll\_type in both years –> include in further models (so far as fixed).

### Without median\_h

mod\_nfl\_22\_pollf<-glmmTMB(tot\_fl\_bodies\_corr~temp+temp\_diff+  
 I(temp\_diff^2)+poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&flowering==1),  
 family="nbinom2")  
mod\_nfl\_23\_pollf<-glmmTMB(tot\_fl\_bodies\_corr~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023&flowering==1),  
 family="nbinom2")  
summary(mod\_nfl\_22\_pollf)

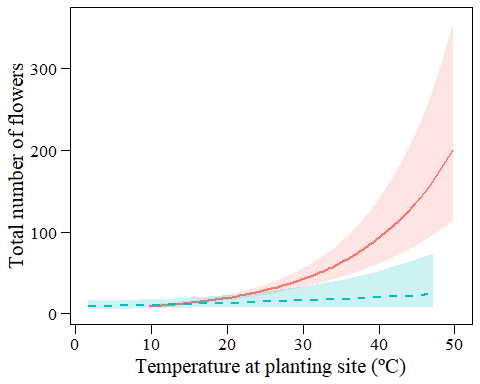
## Family: nbinom2 ( log )  
## Formula:   
## tot\_fl\_bodies\_corr ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type +   
## (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 4598.3 4629.1 -2292.1 4584.3 595   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.04033 0.2008   
## Number of obs: 602, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 1.57   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.3702490 0.1841821 7.440 1.01e-13 \*\*\*  
## temp 0.0787677 0.0087284 9.024 < 2e-16 \*\*\*  
## temp\_diff -0.0197476 0.0047099 -4.193 2.76e-05 \*\*\*  
## I(temp\_diff^2) -0.0004312 0.0001860 -2.319 0.02041 \*   
## poll\_typeSelfpollinated -0.2497364 0.0785961 -3.177 0.00149 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_nfl\_23\_pollf)

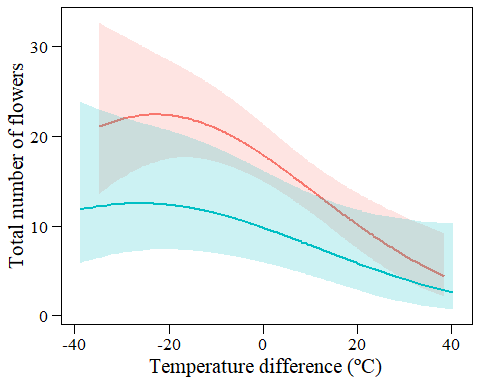
## Family: nbinom2 ( log )  
## Formula:   
## tot\_fl\_bodies\_corr ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type +   
## (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 2878.9 2906.9 -1432.4 2864.9 396   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.4128 0.6425   
## Number of obs: 403, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 1.68   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.0142534 0.3384157 5.952 2.65e-09 \*\*\*  
## temp 0.0219695 0.0152192 1.444 0.148868   
## temp\_diff -0.0186928 0.0071619 -2.610 0.009053 \*\*   
## I(temp\_diff^2) -0.0003552 0.0002398 -1.481 0.138481   
## poll\_typeSelfpollinated -0.3785187 0.1003678 -3.771 0.000162 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### Predictions

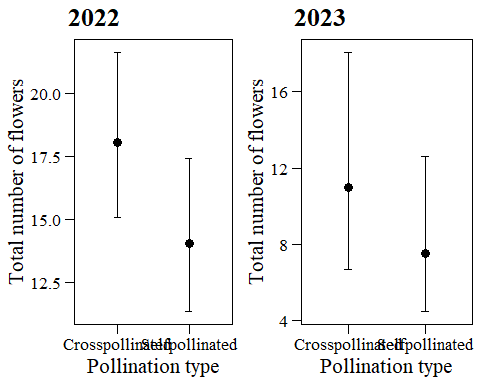
ggplot()+  
 #geom\_point(data=data\_transplants,aes(x=temp,y=tot\_fl\_bodies\_corr,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_nfl\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfl\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_nfl\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfl\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1,linetype="dashed")+  
 my\_theme\_legend()+xlab("Temperature at planting site (ºC)")+  
 ylab("Total number of flowers")



ggplot()+  
 #geom\_point(data=data\_transplants,aes(x=temp\_diff,y=tot\_fl\_bodies\_corr,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_nfl\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfl\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_nfl\_23\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfl\_23\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1)+  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Total number of flowers")

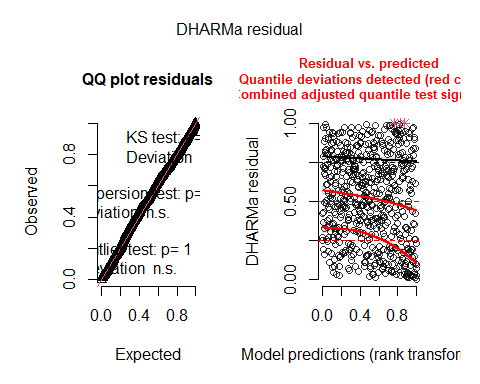


grid.arrange(  
 ggplot()+  
 #geom\_jitter(data=data\_transplants%>%filter(year==2022),  
 # aes(x=poll\_type,y=tot\_fl\_bodies\_corr),width=0.05,alpha=0.1)+  
 geom\_point(data=ggpredict(mod\_nfl\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_nfl\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme\_legend()+xlab("Pollination type")+ylab("Total number of flowers")+  
 ggtitle("2022"),  
 ggplot()+  
 #geom\_jitter(data=data\_transplants%>%filter(year==2023),  
 # aes(x=poll\_type,y=tot\_fl\_bodies\_corr),width=0.05,alpha=0.1)+  
 geom\_point(data=ggpredict(mod\_nfl\_23\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_nfl\_23\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme\_legend()+xlab("Pollination type")+ylab("Total number of flowers")+  
 ggtitle("2023"),  
ncol=2)

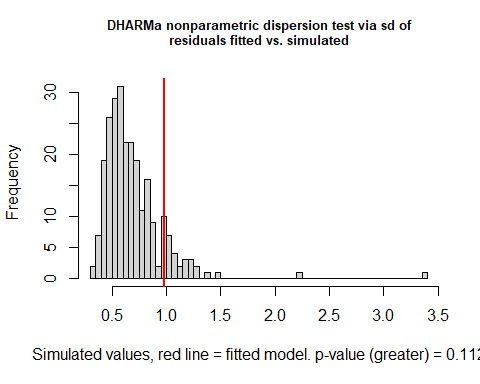


#### Diagnostics

plot(simulateResiduals(mod\_nfl\_22\_pollf))

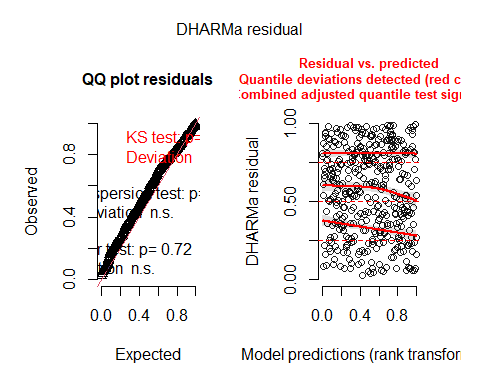


testDispersion(simulateResiduals(mod\_nfl\_22\_pollf),alternative="greater")

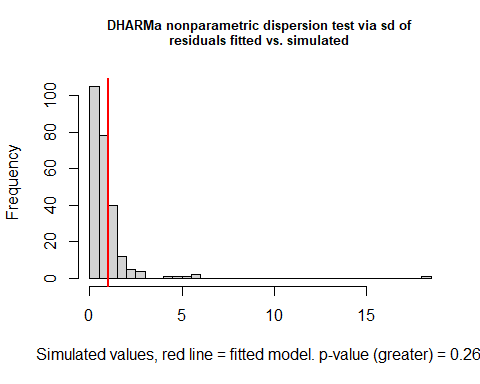


##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 1.4301, p-value = 0.112  
## alternative hypothesis: greater

plot(simulateResiduals(mod\_nfl\_23\_pollf))



testDispersion(simulateResiduals(mod\_nfl\_23\_pollf),alternative="greater")



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 1.1369, p-value = 0.26  
## alternative hypothesis: greater

### With median\_h

mod\_nfl\_22\_mh\_pollf<-glmmTMB(tot\_fl\_bodies\_corr~temp+temp\_diff+I(temp\_diff^2)+   
 median\_h+  
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&flowering==1),  
 family="nbinom2")  
mod\_nfl\_23\_mh\_pollf<-glmmTMB(tot\_fl\_bodies\_corr~temp+temp\_diff+I(temp\_diff^2)+   
 median\_h+  
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023&flowering==1),  
 family="nbinom2")  
summary(mod\_nfl\_22\_mh\_pollf)

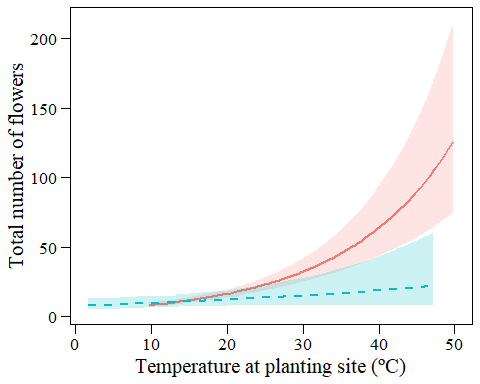
## Family: nbinom2 ( log )  
## Formula:   
## tot\_fl\_bodies\_corr ~ temp + temp\_diff + I(temp\_diff^2) + median\_h +   
## poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 4494.2 4529.4 -2239.1 4478.2 590   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.03781 0.1945   
## Number of obs: 598, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 1.84   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.6027257 0.1834070 3.286 0.001015 \*\*   
## temp 0.0686946 0.0077141 8.905 < 2e-16 \*\*\*  
## temp\_diff -0.0170853 0.0043298 -3.946 7.95e-05 \*\*\*  
## I(temp\_diff^2) -0.0002510 0.0001727 -1.454 0.146083   
## median\_h 0.1140172 0.0130884 8.711 < 2e-16 \*\*\*  
## poll\_typeSelfpollinated -0.2744163 0.0734294 -3.737 0.000186 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_nfl\_23\_mh\_pollf)

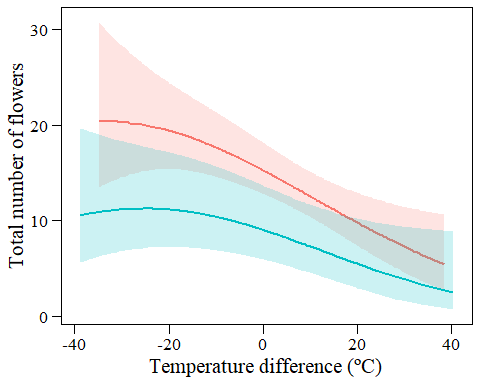
## Family: nbinom2 ( log )  
## Formula:   
## tot\_fl\_bodies\_corr ~ temp + temp\_diff + I(temp\_diff^2) + median\_h +   
## poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 2816.4 2848.4 -1400.2 2800.4 395   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.2739 0.5233   
## Number of obs: 403, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 1.97   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.0263710 0.3148979 3.259 0.00112 \*\*   
## temp 0.0223541 0.0137753 1.623 0.10464   
## temp\_diff -0.0175843 0.0066455 -2.646 0.00814 \*\*   
## I(temp\_diff^2) -0.0003488 0.0002243 -1.555 0.11989   
## median\_h 0.1287413 0.0159247 8.084 6.25e-16 \*\*\*  
## poll\_typeSelfpollinated -0.2601370 0.0955528 -2.722 0.00648 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### Predictions

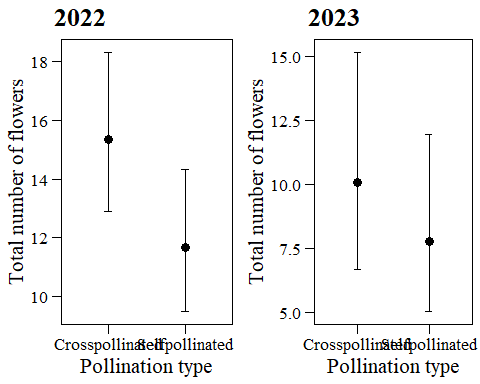
ggplot()+  
 #geom\_point(data=data\_transplants,aes(x=temp,y=tot\_fl\_bodies\_corr,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_nfl\_22\_mh\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfl\_22\_mh\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_nfl\_23\_mh\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfl\_23\_mh\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1,linetype="dashed")+  
 my\_theme\_legend()+xlab("Temperature at planting site (ºC)")+  
 ylab("Total number of flowers")



ggplot()+  
 #geom\_point(data=data\_transplants,aes(x=temp\_diff,y=tot\_fl\_bodies\_corr,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_nfl\_22\_mh\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfl\_22\_mh\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_nfl\_23\_mh\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfl\_23\_mh\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1)+  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Total number of flowers")

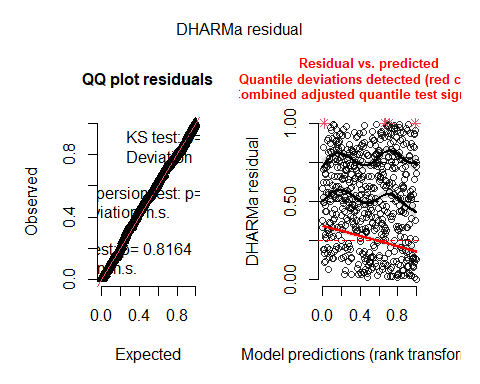


grid.arrange(  
 ggplot()+  
 #geom\_jitter(data=data\_transplants%>%filter(year==2022),  
 # aes(x=poll\_type,y=tot\_fl\_bodies\_corr),width=0.05,alpha=0.1)+  
 geom\_point(data=ggpredict(mod\_nfl\_22\_mh\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_nfl\_22\_mh\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Total number of flowers")+  
 ggtitle("2022"),  
 ggplot()+  
 #geom\_jitter(data=data\_transplants%>%filter(year==2023),  
 # aes(x=poll\_type,y=tot\_fl\_bodies\_corr),width=0.05,alpha=0.1)+  
 geom\_point(data=ggpredict(mod\_nfl\_23\_mh\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_nfl\_23\_mh\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Total number of flowers")+  
 ggtitle("2023"),  
ncol=2)

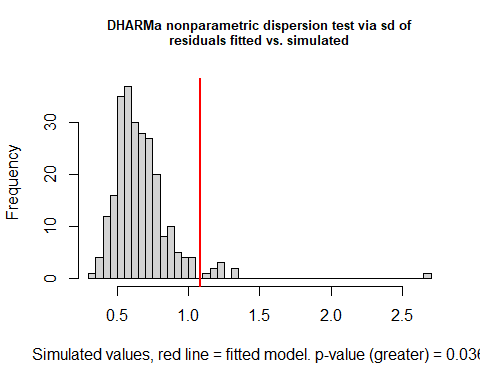


#### Diagnostics

plot(simulateResiduals(mod\_nfl\_22\_mh\_pollf))

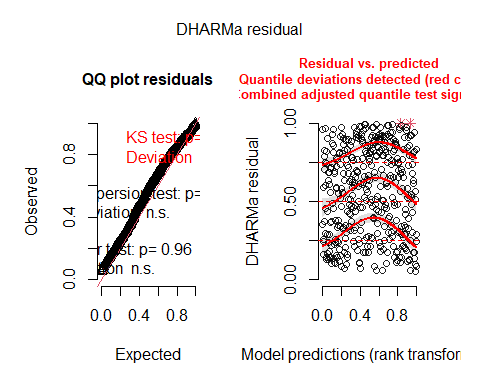


testDispersion(simulateResiduals(mod\_nfl\_22\_mh\_pollf),alternative="greater")

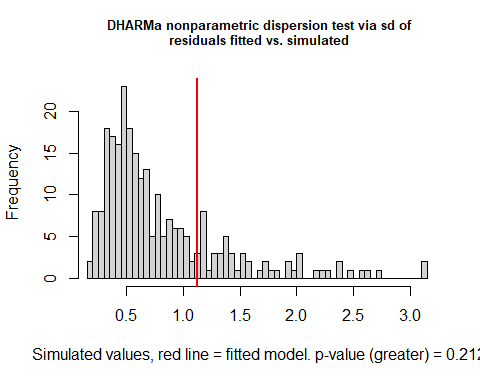


##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 1.62, p-value = 0.036  
## alternative hypothesis: greater

plot(simulateResiduals(mod\_nfl\_23\_mh\_pollf))



testDispersion(simulateResiduals(mod\_nfl\_23\_mh\_pollf),alternative="greater")



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 1.406, p-value = 0.212  
## alternative hypothesis: greater

# TRY: Probability of flowering + n flowers (ZI models)

Convergence problems, NAs in the model output.

# Probability of fruiting

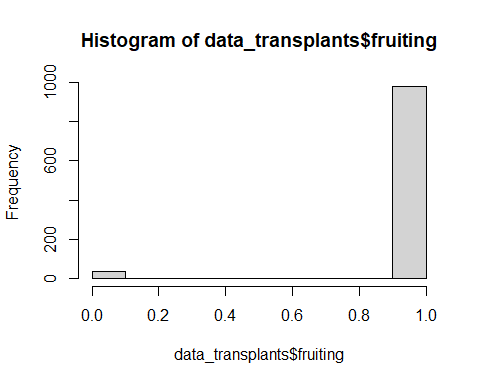
data\_transplants<-data\_transplants%>%  
 mutate(fruiting=ifelse(surv==0|is.na(surv),NA,   
 # NA for plants that did not survive  
 ifelse(F\_NF\_A=="A"|F\_NF\_A=="NF",NA,   
 # NA for plants that aborted or did not flower  
 ifelse(is.na(n\_tot\_fr),NA,  
 # NA for plants where n\_tot\_fr is NA  
 ifelse(n\_tot\_fr>0,1,0)))))   
# 1/0 for plants that produced / did not produce fruits  
table(subset(data\_transplants,year==2022)$fruiting)

##   
## 0 1   
## 21 582

table(subset(data\_transplants,year==2023)$fruiting)

##   
## 0 1   
## 13 398

hist(data\_transplants$fruiting)



Of 603 plants that flowered in 2022, 582 produced fruits and 21 did not produce fruits. Of 411 plants that flowered in 2022, 398 produced fruits and 13 did not produce fruits.

Does pollination type have an effect on probability of fruiting?

mod\_fruiting\_poll\_22<-glmmTMB(fruiting~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022),  
 family="binomial")  
mod\_fruiting\_poll\_23<-glmmTMB(fruiting~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023),  
 family="binomial")  
summary(mod\_fruiting\_poll\_22)

## Family: binomial ( logit )  
## Formula: fruiting ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 187.4 200.6 -90.7 181.4 600   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 4.191e-09 6.474e-05  
## Number of obs: 603, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.2047 0.2474 12.953 <2e-16 \*\*\*  
## poll\_typeSelfpollinated 0.5028 0.5633 0.893 0.372   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_fruiting\_poll\_23)

## Family: binomial ( logit )  
## Formula: fruiting ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 118.4 130.5 -56.2 112.4 408   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 1.154e-08 0.0001074  
## Number of obs: 411, groups: plot, 7  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.1918 0.2945 10.836 <2e-16 \*\*\*  
## poll\_typeSelfpollinated 1.4716 1.0470 1.406 0.16   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

nobs(mod\_fruiting\_poll\_22)

## [1] 603

nobs(mod\_fruiting\_poll\_23)

## [1] 411

No effect of pollination type, so we do not include it in further models.

mod\_fruiting\_22<-glmmTMB(fruiting~temp+temp\_diff+I(temp\_diff^2)+   
 (1|plot),data\_transplants%>%filter(year==2022),  
 family="binomial")  
mod\_fruiting\_23<-glmmTMB(fruiting~temp+temp\_diff+I(temp\_diff^2)+   
 (1|plot),data\_transplants%>%filter(year==2023),  
 family="binomial")  
summary(mod\_fruiting\_22)

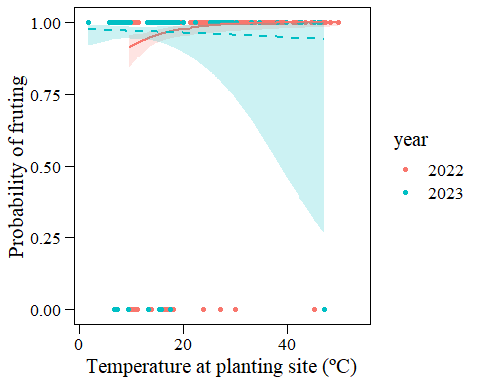
## Family: binomial ( logit )  
## Formula: fruiting ~ temp + temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022)  
##   
## AIC BIC logLik deviance df.resid   
## 179.6 201.6 -84.8 169.6 598   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 1.674e-09 4.091e-05  
## Number of obs: 603, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.9959820 0.7744561 1.286 0.19843   
## temp 0.1334265 0.0502103 2.657 0.00788 \*\*  
## temp\_diff -0.1083797 0.0363732 -2.980 0.00289 \*\*  
## I(temp\_diff^2) 0.0004676 0.0016343 0.286 0.77479   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_fruiting\_23)

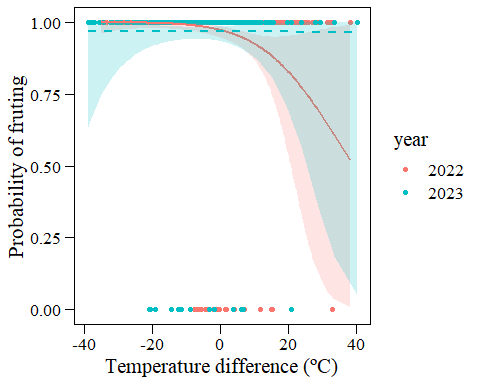
## Family: binomial ( logit )  
## Formula: fruiting ~ temp + temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023)  
##   
## AIC BIC logLik deviance df.resid   
## 124.4 144.4 -57.2 114.4 398   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 4.987e-09 7.062e-05  
## Number of obs: 403, groups: plot, 7  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.631e+00 8.742e-01 4.153 3.28e-05 \*\*\*  
## temp -1.881e-02 5.282e-02 -0.356 0.722   
## temp\_diff -1.849e-03 4.083e-02 -0.045 0.964   
## I(temp\_diff^2) -5.434e-06 1.190e-03 -0.005 0.996   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Predictions

ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp,y=fruiting,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_fruiting\_22,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_fruiting\_22,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_fruiting\_23,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_fruiting\_23,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1,linetype="dashed")+  
 my\_theme\_legend()+xlab("Temperature at planting site (ºC)")+  
 ylab("Probability of fruting")

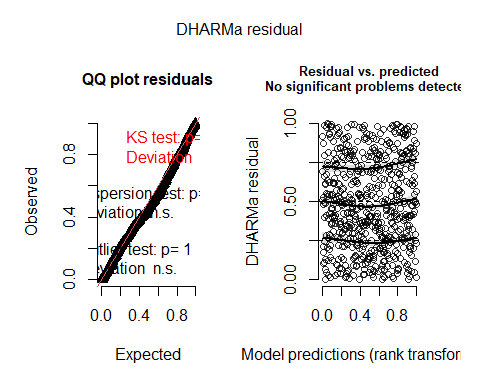


ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp\_diff,y=fruiting,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_fruiting\_22,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_fruiting\_22,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_fruiting\_23,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_fruiting\_23,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1,linetype="dashed")+  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Probability of fruting")

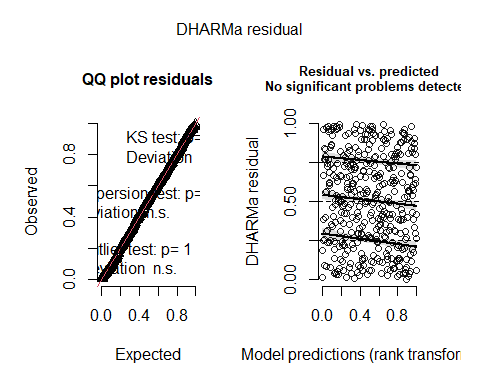


## Diagnostics

plot(simulateResiduals(mod\_fruiting\_22))



plot(simulateResiduals(mod\_fruiting\_23))



# Number of fruits

Models for number of fruits in those that flowered (flowering=1).

Does pollination type have an effect on number of flowers?

mod\_nfr\_poll\_22<-glmmTMB(n\_tot\_fr~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&flowering==1),  
 family="nbinom2") # nbinom2 lower AIC than poisson  
mod\_nfr\_poll\_23<-glmmTMB(n\_tot\_fr~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023&flowering==1),  
 family="nbinom2") # nbinom2 lower AIC than poisson  
summary(mod\_nfl\_poll\_22)

## Family: nbinom2 ( log )  
## Formula: tot\_fl\_bodies\_corr ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 4692.3 4709.9 -2342.1 4684.3 598   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.1512 0.3889   
## Number of obs: 602, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 1.35   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.78969 0.14914 18.705 < 2e-16 \*\*\*  
## poll\_typeSelfpollinated -0.28999 0.08259 -3.511 0.000446 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_nfl\_poll\_23)

## Family: nbinom2 ( log )  
## Formula: tot\_fl\_bodies\_corr ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 2942.8 2958.9 -1467.4 2934.8 407   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.4261 0.6528   
## Number of obs: 411, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 1.61   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.37403 0.25631 9.262 < 2e-16 \*\*\*  
## poll\_typeSelfpollinated -0.36809 0.09829 -3.745 0.00018 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

nobs(mod\_nfl\_poll\_22)

## [1] 602

nobs(mod\_nfl\_poll\_23)

## [1] 411

Significant effect of poll\_type in both years –> include in further models (so far as fixed).

mod\_nfr\_22\_pollf<-glmmTMB(n\_tot\_fr~temp+temp\_diff+I(temp\_diff^2)+  
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&flowering==1),  
 family="nbinom2")  
mod\_nfr\_23\_pollf<-glmmTMB(n\_tot\_fr~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023&flowering==1),  
 family="nbinom2")  
summary(mod\_nfr\_22\_pollf)

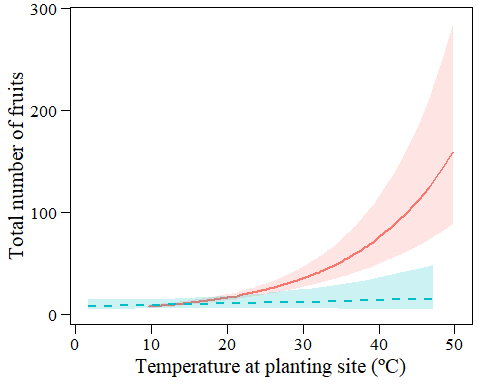
## Family: nbinom2 ( log )  
## Formula:   
## n\_tot\_fr ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 4461.7 4492.5 -2223.8 4447.7 596   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.03875 0.1969   
## Number of obs: 603, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 1.39   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.2802088 0.1834203 6.980 2.96e-12 \*\*\*  
## temp 0.0759337 0.0088389 8.591 < 2e-16 \*\*\*  
## temp\_diff -0.0218949 0.0049469 -4.426 9.60e-06 \*\*\*  
## I(temp\_diff^2) -0.0004903 0.0001940 -2.527 0.01150 \*   
## poll\_typeSelfpollinated -0.2222689 0.0834435 -2.664 0.00773 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_nfr\_23\_pollf)

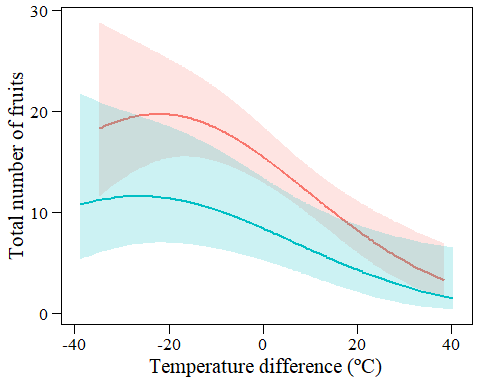
## Family: nbinom2 ( log )  
## Formula:   
## n\_tot\_fr ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 2768.1 2796.1 -1377.1 2754.1 396   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.3562 0.5968   
## Number of obs: 403, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 1.53   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.966535 0.329766 5.963 2.47e-09 \*\*\*  
## temp 0.013190 0.015553 0.848 0.396383   
## temp\_diff -0.024375 0.007774 -3.136 0.001715 \*\*   
## I(temp\_diff^2) -0.000459 0.000261 -1.759 0.078617 .   
## poll\_typeSelfpollinated -0.364677 0.105202 -3.466 0.000527 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### Predictions

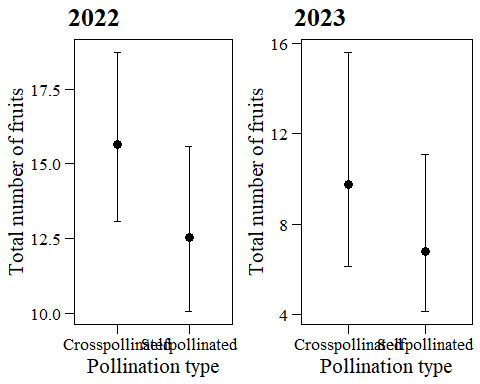
ggplot()+  
 #geom\_point(data=data\_transplants,aes(x=temp,y=n\_tot\_fr,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_nfr\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfr\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_nfr\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfr\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1,linetype="dashed")+  
 my\_theme\_legend()+xlab("Temperature at planting site (ºC)")+  
 ylab("Total number of fruits")



ggplot()+  
 #geom\_point(data=data\_transplants,aes(x=temp\_diff,y=n\_tot\_fr,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_nfr\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfr\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_nfr\_23\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nfr\_23\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1)+  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Total number of fruits")

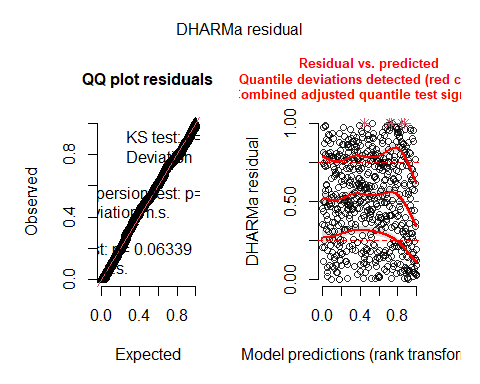


grid.arrange(  
 ggplot()+  
# geom\_jitter(data=data\_transplants%>%filter(year==2022),  
# aes(x=poll\_type,y=n\_tot\_fr),width=0.05,alpha=0.1)+  
 geom\_point(data=ggpredict(mod\_nfr\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_nfr\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Total number of fruits")+  
 ggtitle("2022"),  
 ggplot()+  
# geom\_jitter(data=data\_transplants%>%filter(year==2023),  
# aes(x=poll\_type,y=n\_tot\_fr),width=0.05,alpha=0.1)+  
 geom\_point(data=ggpredict(mod\_nfr\_23\_pollf,terms=c("poll\_type")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_nfr\_23\_pollf,terms=c("poll\_type")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Total number of fruits")+  
 ggtitle("2023"),  
ncol=2)

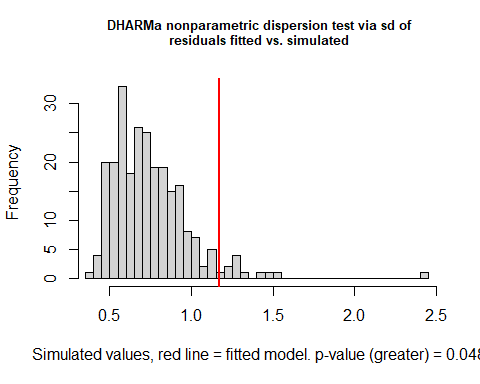


### Diagnostics

plot(simulateResiduals(mod\_nfr\_22\_pollf))

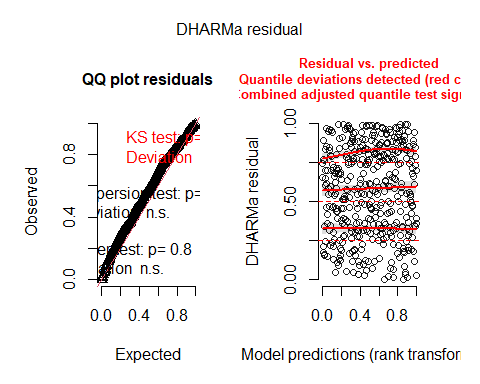


testDispersion(simulateResiduals(mod\_nfr\_22\_pollf),alternative="greater")

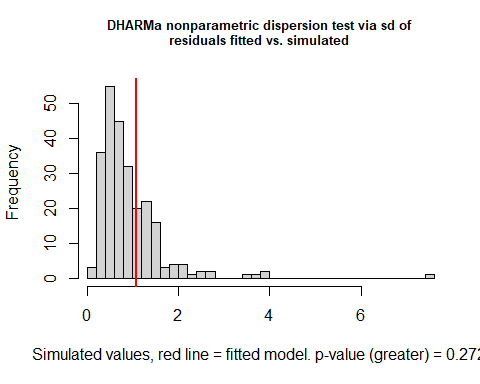


##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 1.5724, p-value = 0.048  
## alternative hypothesis: greater

plot(simulateResiduals(mod\_nfr\_23\_pollf))



testDispersion(simulateResiduals(mod\_nfr\_23\_pollf),alternative="greater")

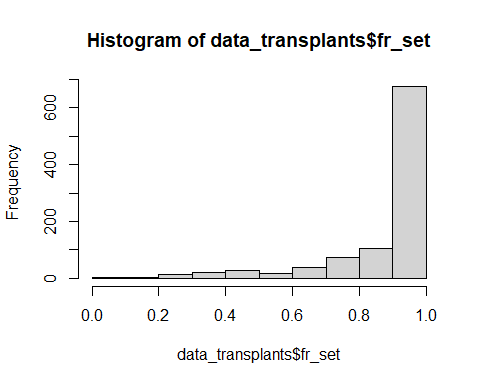


##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 1.176, p-value = 0.272  
## alternative hypothesis: greater

# TRY: Probability of fruiting + n fruits (ZI models)

# Fruit set

data\_transplants<-data\_transplants%>%  
 mutate(fr\_set=ifelse(fruiting==0,NA,n\_tot\_fr/tot\_fl\_bodies\_corr))  
hist(data\_transplants$fr\_set)



range(data\_transplants$fr\_set,na.rm=T)

## [1] 0.04347826 1.00000000

Does pollination type have an effect on fruit set?

mod\_frset\_poll\_22<-glmmTMB(fr\_set~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&flowering==1),  
 weights=tot\_fl\_bodies\_corr,family="binomial")  
mod\_frset\_poll\_23<-glmmTMB(fr\_set~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023&flowering==1),  
 weights=tot\_fl\_bodies\_corr,family="binomial")  
summary(mod\_frset\_poll\_22)

## Family: binomial ( logit )  
## Formula: fr\_set ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & flowering == 1)  
## Weights: tot\_fl\_bodies\_corr  
##   
## AIC BIC logLik deviance df.resid   
## 3896.6 3909.7 -1945.3 3890.6 579   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.6808 0.8251   
## Number of obs: 582, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.1730 0.3099 7.012 2.35e-12 \*\*\*  
## poll\_typeSelfpollinated 0.1669 0.0688 2.426 0.0153 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_frset\_poll\_23)

## Family: binomial ( logit )  
## Formula: fr\_set ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & flowering == 1)  
## Weights: tot\_fl\_bodies\_corr  
##   
## AIC BIC logLik deviance df.resid   
## 2328.5 2340.5 -1161.3 2322.5 395   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.3326 0.5767   
## Number of obs: 398, groups: plot, 7  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.97778 0.24141 8.193 2.56e-16 \*\*\*  
## poll\_typeSelfpollinated 0.02855 0.10551 0.271 0.787   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

nobs(mod\_frset\_poll\_22)

## [1] 582

nobs(mod\_frset\_poll\_23)

## [1] 398

Significant effect of poll\_type in 2022 –> include in further models (so far as fixed).

mod\_frset\_22\_pollf<-glmmTMB(fr\_set~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&flowering==1),  
 weights=tot\_fl\_bodies\_corr,family="binomial")  
mod\_frset\_23<-glmmTMB(fr\_set~temp+temp\_diff+I(temp\_diff^2)+   
 (1|plot),  
 data\_transplants%>%filter(year==2023&flowering==1),  
 weights=tot\_fl\_bodies\_corr,family="binomial")  
summary(mod\_frset\_22\_pollf)

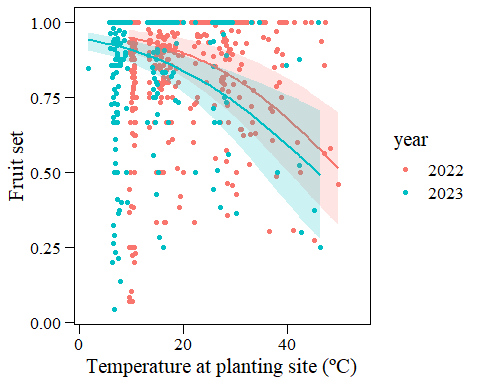
## Family: binomial ( logit )  
## Formula:   
## fr\_set ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & flowering == 1)  
## Weights: tot\_fl\_bodies\_corr  
##   
## AIC BIC logLik deviance df.resid   
## 3527.9 3554.1 -1757.9 3515.9 576   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.8594 0.927   
## Number of obs: 582, groups: plot, 8  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.6192800 0.3697765 9.788 <2e-16 \*\*\*  
## temp -0.0708285 0.0067751 -10.454 <2e-16 \*\*\*  
## temp\_diff -0.0085978 0.0039743 -2.163 0.0305 \*   
## I(temp\_diff^2) -0.0002351 0.0001438 -1.635 0.1020   
## poll\_typeSelfpollinated 0.1694156 0.0701906 2.414 0.0158 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_frset\_23)

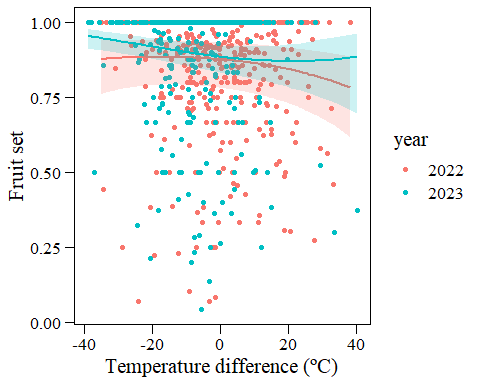
## Family: binomial ( logit )  
## Formula: fr\_set ~ temp + temp\_diff + I(temp\_diff^2) + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & flowering == 1)  
## Weights: tot\_fl\_bodies\_corr  
##   
## AIC BIC logLik deviance df.resid   
## 2230.7 2250.6 -1110.4 2220.7 385   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.2509 0.5009   
## Number of obs: 390, groups: plot, 7  
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.7934537 0.2965712 9.419 < 2e-16 \*\*\*  
## temp -0.0636964 0.0128124 -4.971 6.64e-07 \*\*\*  
## temp\_diff -0.0129536 0.0062786 -2.063 0.0391 \*   
## I(temp\_diff^2) 0.0003265 0.0002262 1.443 0.1490   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### Predictions

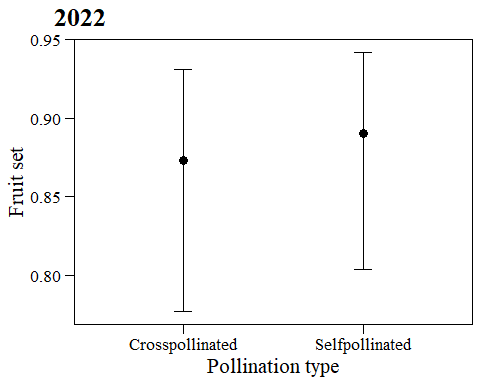
ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp,y=fr\_set,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_frset\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_frset\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_frset\_23,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_frset\_23,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1)+  
 my\_theme\_legend()+xlab("Temperature at planting site (ºC)")+  
 ylab("Fruit set")



ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp\_diff,y=fr\_set,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_frset\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_frset\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_frset\_23,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_frset\_23,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1)+  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Fruit set")

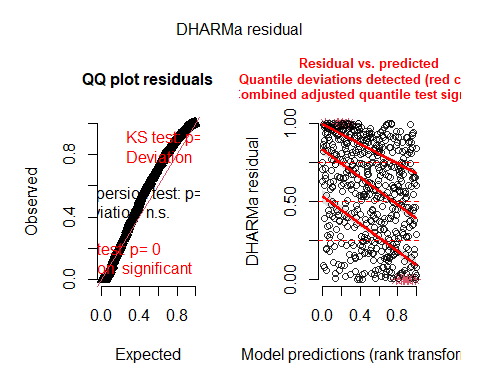


ggplot()+  
 #geom\_jitter(data=data\_transplants%>%filter(year==2022),  
 # aes(x=poll\_type,y=fr\_set),width=0.05,alpha=0.1)+   
 geom\_point(data=ggpredict(mod\_frset\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_frset\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Fruit set")+ggtitle("2022")

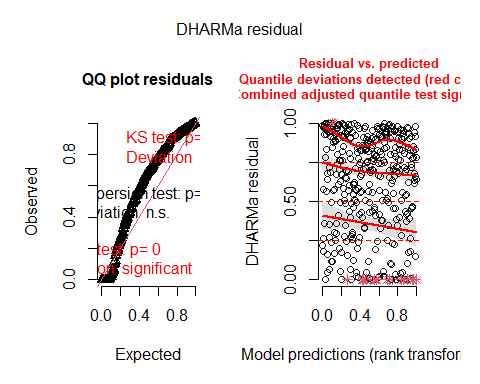


#### Diagnostics

plot(simulateResiduals(mod\_frset\_22\_pollf))



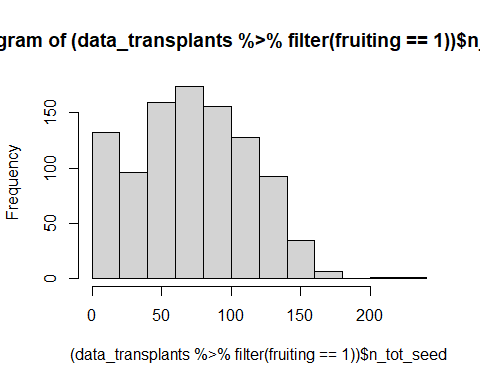
plot(simulateResiduals(mod\_frset\_23))



# Number of seeds

Including only plants that produced fruits.

hist((data\_transplants%>%filter(fruiting==1))$n\_tot\_seed)



Does pollination type have an effect on number of seeds?

mod\_nseed\_poll\_22<-glmmTMB(n\_tot\_seed~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&fruiting==1),  
 family="nbinom2") # nbinom2 lower AIC than poisson  
mod\_nseed\_poll\_23<-glmmTMB(n\_tot\_seed~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023&fruiting==1),  
 family="nbinom2") # nbinom2 lower AIC than poisson  
summary(mod\_nseed\_poll\_22)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & fruiting == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 5999.2 6016.6 -2995.6 5991.2 577   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 6.966e-10 2.639e-05  
## Number of obs: 581, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 3.6   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.40687 0.02632 167.41 < 2e-16 \*\*\*  
## poll\_typeSelfpollinated 0.13519 0.04960 2.73 0.00642 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_nseed\_poll\_23)

## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & fruiting == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 3940.9 3956.8 -1966.4 3932.9 394   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.06378 0.2525   
## Number of obs: 398, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 0.786   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.8457 0.1258 30.568 <2e-16 \*\*\*  
## poll\_typeSelfpollinated 0.3004 0.1307 2.298 0.0215 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

nobs(mod\_nseed\_poll\_22)

## [1] 581

nobs(mod\_nseed\_poll\_23)

## [1] 398

Yes! Self-pollinated have a higher number of seeds than cross-pollinated (WHY??).

Significant effect of poll\_type in both years –> include in further models (so far as fixed).

mod\_nseed\_22\_pollf<-glmmTMB(n\_tot\_seed~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&fruiting==1),  
 family="nbinom2")  
mod\_nseed\_23\_pollf<-glmmTMB(n\_tot\_seed~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023&fruiting==1),  
 family="nbinom2")  
summary(mod\_nseed\_22\_pollf)

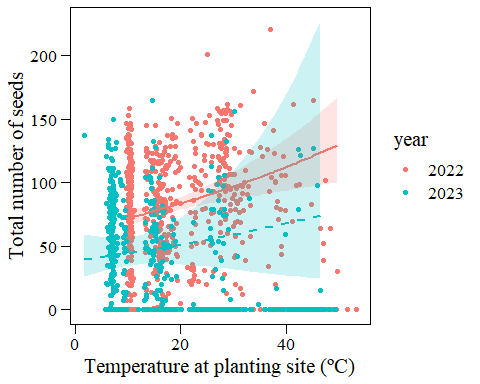
## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type +   
## (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & fruiting == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 5990.8 6021.4 -2988.4 5976.8 574   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## plot (Intercept) 5.894e-10 2.428e-05  
## Number of obs: 581, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 3.7   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.126e+00 7.933e-02 52.01 < 2e-16 \*\*\*  
## temp 1.476e-02 3.998e-03 3.69 0.000222 \*\*\*  
## temp\_diff -6.953e-03 2.827e-03 -2.46 0.013899 \*   
## I(temp\_diff^2) -9.847e-05 1.085e-04 -0.91 0.364009   
## poll\_typeSelfpollinated 1.464e-01 4.994e-02 2.93 0.003373 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_nseed\_23\_pollf)

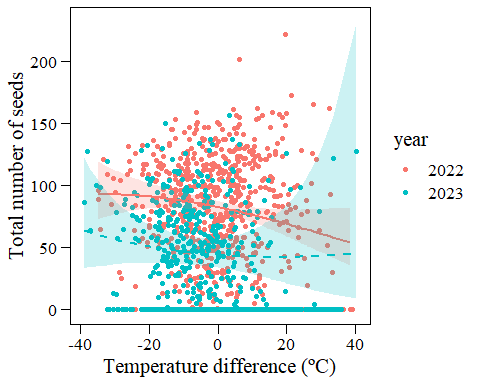
## Family: nbinom2 ( log )  
## Formula: n\_tot\_seed ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type +   
## (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & fruiting == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 3858.1 3885.8 -1922.0 3844.1 383   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.05893 0.2428   
## Number of obs: 390, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 0.773   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.6145010 0.2696506 13.404 <2e-16 \*\*\*  
## temp 0.0141040 0.0160882 0.877 0.3807   
## temp\_diff -0.0044522 0.0097869 -0.455 0.6492   
## I(temp\_diff^2) 0.0001310 0.0002887 0.454 0.6500   
## poll\_typeSelfpollinated 0.2533399 0.1365412 1.855 0.0635 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### Predictions

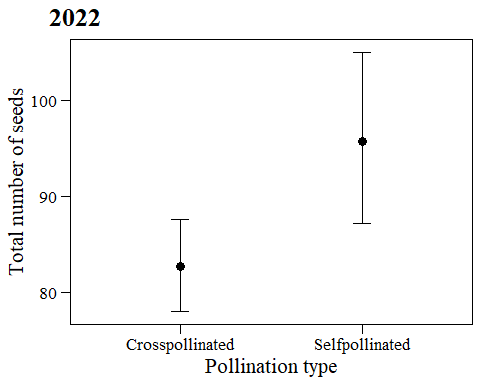
ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp,y=n\_tot\_seed,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_nseed\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nseed\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_nseed\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nseed\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1,linetype="dashed")+  
 my\_theme\_legend()+xlab("Temperature at planting site (ºC)")+  
 ylab("Total number of seeds")



ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp\_diff,y=n\_tot\_seed,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_nseed\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nseed\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_nseed\_23\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_nseed\_23\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1,linetype="dashed")+  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Total number of seeds")

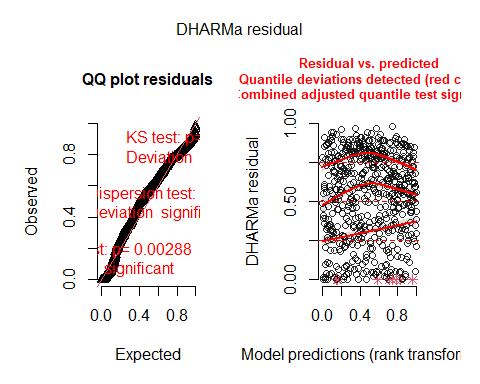


ggplot()+  
 geom\_point(data=ggpredict(mod\_nseed\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_nseed\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),width=0.1)+   
 my\_theme()+xlab("Pollination type")+ylab("Total number of seeds")+  
 ggtitle("2022")

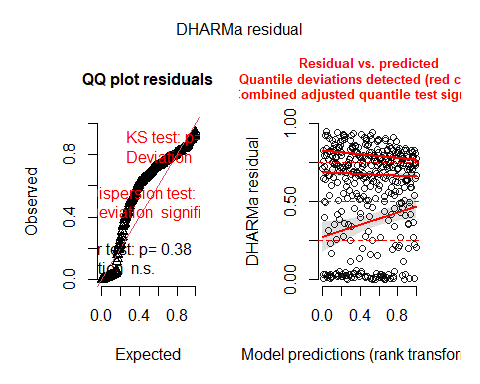


### Diagnostics

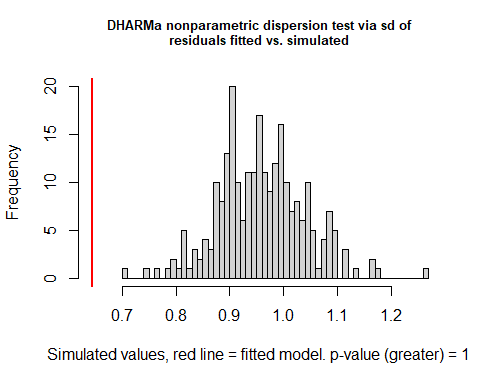
plot(simulateResiduals(mod\_nseed\_22\_pollf))



plot(simulateResiduals(mod\_nseed\_23\_pollf))

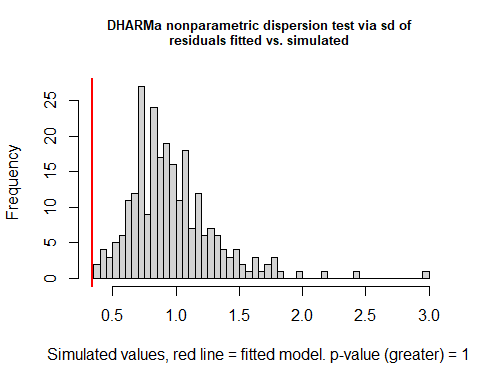


testDispersion(simulateResiduals(mod\_nseed\_22\_pollf),alternative="greater")



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 0.67122, p-value = 1  
## alternative hypothesis: greater

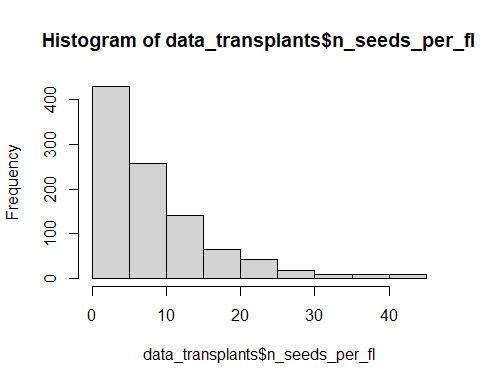
testDispersion(simulateResiduals(mod\_nseed\_23\_pollf),alternative="greater")



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 0.34456, p-value = 1  
## alternative hypothesis: greater

# Seeds per flower

data\_transplants<-data\_transplants%>%  
 mutate(n\_seeds\_per\_fl=ifelse(fruiting==0,NA,  
 n\_tot\_seed/tot\_fl\_bodies\_corr))  
hist(data\_transplants$n\_seeds\_per\_fl)



Does pollination type have an effect on number of seeds per flower?

mod\_seedsperfl\_poll\_22<-glmmTMB(n\_seeds\_per\_fl~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2022&flowering==1),  
 family="nbinom2") # nbinom2 lower AIC than poisson  
mod\_seedsperfl\_poll\_23<-glmmTMB(n\_seeds\_per\_fl~poll\_type+(1|plot),  
 data\_transplants%>%filter(year==2023&flowering==1),  
 family="nbinom2") # nbinom2 lower AIC than poisson  
summary(mod\_seedsperfl\_poll\_22)

## Family: nbinom2 ( log )  
## Formula: n\_seeds\_per\_fl ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2022 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 3704.9 3722.3 -1848.4 3696.9 577   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.03074 0.1753   
## Number of obs: 581, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 1.82   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.11781 0.07882 26.868 < 2e-16 \*\*\*  
## poll\_typeSelfpollinated 0.41911 0.07498 5.589 2.28e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_seedsperfl\_poll\_23)

## Family: nbinom2 ( log )  
## Formula: n\_seeds\_per\_fl ~ poll\_type + (1 | plot)  
## Data: data\_transplants %>% filter(year == 2023 & flowering == 1)  
##   
## AIC BIC logLik deviance df.resid   
## 2366.4 2382.4 -1179.2 2358.4 394   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.1006 0.3171   
## Number of obs: 398, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 1.21   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.8767 0.1409 13.315 < 2e-16 \*\*\*  
## poll\_typeSelfpollinated 0.5742 0.1121 5.121 3.04e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

nobs(mod\_seedsperfl\_poll\_22)

## [1] 581

nobs(mod\_seedsperfl\_poll\_23)

## [1] 398

Yes! Self-pollinated have a higher number of seeds per flower than cross-pollinated (WHY??).

Significant effect of poll\_type in both years –> include in further models (so far as fixed).

mod\_seedsperfl\_22\_pollf<-glmmTMB(n\_seeds\_per\_fl~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%  
 filter(year==2022&flowering==1&tot\_fl\_bodies>0),  
 family="nbinom2")  
mod\_seedsperfl\_23\_pollf<-glmmTMB(n\_seeds\_per\_fl~temp+temp\_diff+I(temp\_diff^2)+   
 poll\_type+(1|plot),  
 data\_transplants%>%  
 filter(year==2023&flowering==1&tot\_fl\_bodies>0),  
 family="nbinom2")  
summary(mod\_seedsperfl\_22\_pollf)

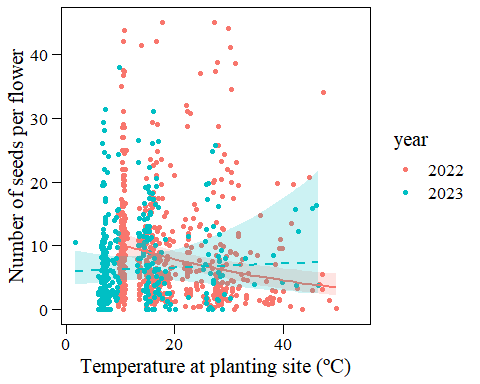
## Family: nbinom2 ( log )  
## Formula:   
## n\_seeds\_per\_fl ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type +   
## (1 | plot)  
## Data:   
## data\_transplants %>% filter(year == 2022 & flowering == 1 & tot\_fl\_bodies >   
## 0)  
##   
## AIC BIC logLik deviance df.resid   
## 3695.3 3725.9 -1840.7 3681.3 574   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.02621 0.1619   
## Number of obs: 581, groups: plot, 8  
##   
## Dispersion parameter for nbinom2 family (): 1.88   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.6070279 0.1460363 17.852 < 2e-16 \*\*\*  
## temp -0.0268520 0.0069900 -3.841 0.000122 \*\*\*  
## temp\_diff 0.0120956 0.0040022 3.022 0.002509 \*\*   
## I(temp\_diff^2) 0.0001869 0.0001810 1.032 0.301938   
## poll\_typeSelfpollinated 0.3731710 0.0761196 4.902 9.47e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(mod\_seedsperfl\_23\_pollf)

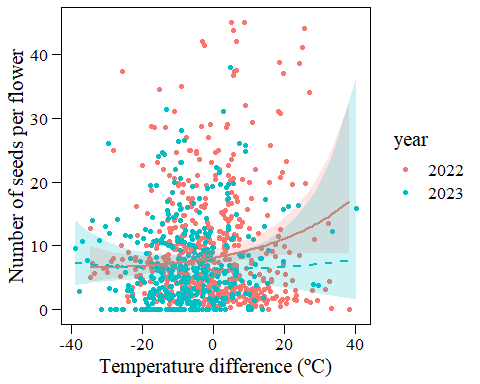
## Family: nbinom2 ( log )  
## Formula:   
## n\_seeds\_per\_fl ~ temp + temp\_diff + I(temp\_diff^2) + poll\_type +   
## (1 | plot)  
## Data:   
## data\_transplants %>% filter(year == 2023 & flowering == 1 & tot\_fl\_bodies >   
## 0)  
##   
## AIC BIC logLik deviance df.resid   
## 2299.7 2327.4 -1142.8 2285.7 382   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## plot (Intercept) 0.09937 0.3152   
## Number of obs: 389, groups: plot, 7  
##   
## Dispersion parameter for nbinom2 family (): 1.23   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.7849373 0.2711533 6.583 4.62e-11 \*\*\*  
## temp 0.0046918 0.0155778 0.301 0.763   
## temp\_diff 0.0005891 0.0086448 0.068 0.946   
## I(temp\_diff^2) 0.0001187 0.0002838 0.418 0.676   
## poll\_typeSelfpollinated 0.5627614 0.1141166 4.931 8.16e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### Predictions

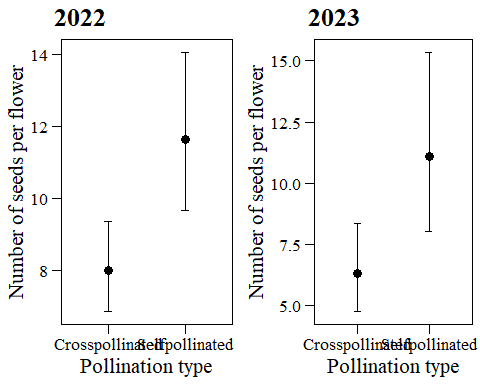
ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp,y=n\_seeds\_per\_fl,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_seedsperfl\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_seedsperfl\_22\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_seedsperfl\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_seedsperfl\_23\_pollf,terms=c("temp[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1,linetype="dashed")+  
 my\_theme\_legend()+xlab("Temperature at planting site (ºC)")+  
 ylab("Number of seeds per flower")



ggplot()+  
 geom\_point(data=data\_transplants,aes(x=temp\_diff,y=n\_seeds\_per\_fl,color=year))+  
 geom\_ribbon(data=ggpredict(mod\_seedsperfl\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#F8766D",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_seedsperfl\_22\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#F8766D",size=1)+  
 geom\_ribbon(data=ggpredict(mod\_seedsperfl\_23\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 fill="#00BFC4",alpha=0.2)+  
 geom\_line(data=ggpredict(mod\_seedsperfl\_23\_pollf,terms=c("temp\_diff[all]")),  
 aes(x=x,y=predicted),color="#00BFC4",size=1,linetype="dashed")+  
 my\_theme\_legend()+xlab("Temperature difference (ºC)")+  
 ylab("Number of seeds per flower")

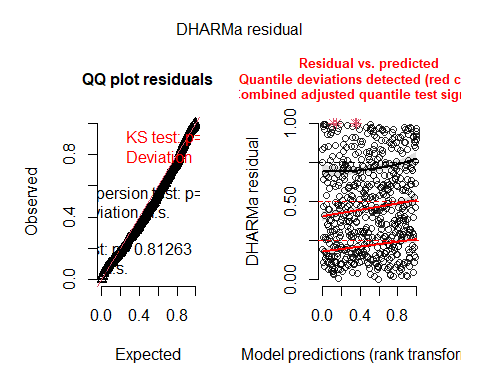


grid.arrange(  
 ggplot()+  
 geom\_point(data=ggpredict(mod\_seedsperfl\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_seedsperfl\_22\_pollf,terms=c("poll\_type[all]")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Number of seeds per flower")+  
 ggtitle("2022"),  
 ggplot()+  
 geom\_point(data=ggpredict(mod\_seedsperfl\_23\_pollf,terms=c("poll\_type")),  
 aes(x=x,y=predicted),size=3)+  
 geom\_errorbar(data=ggpredict(mod\_seedsperfl\_23\_pollf,terms=c("poll\_type")),  
 aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),  
 width=0.1)+  
 my\_theme()+xlab("Pollination type")+ylab("Number of seeds per flower")+  
 ggtitle("2023"),  
ncol=2)

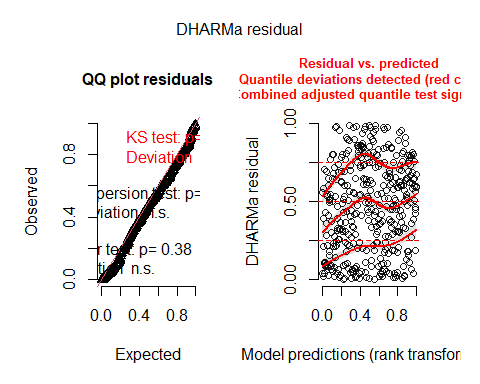


### Diagnostics

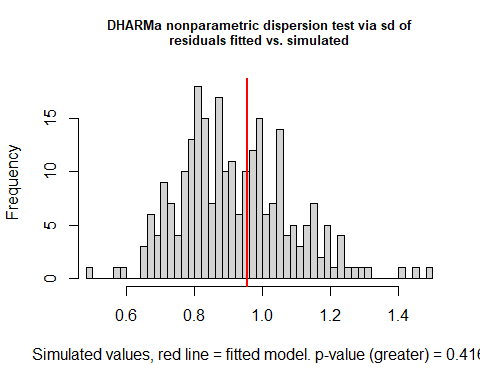
plot(simulateResiduals(mod\_seedsperfl\_22\_pollf))



plot(simulateResiduals(mod\_seedsperfl\_23\_pollf))

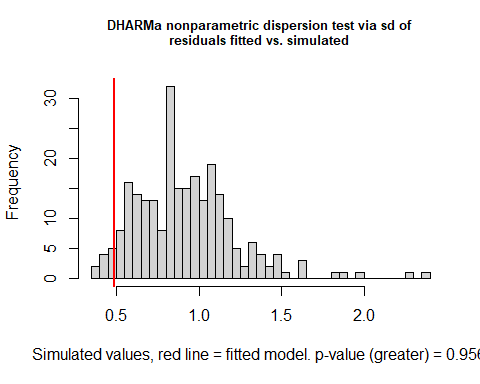


testDispersion(simulateResiduals(mod\_seedsperfl\_22\_pollf),alternative="greater")



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 1.0368, p-value = 0.416  
## alternative hypothesis: greater

testDispersion(simulateResiduals(mod\_seedsperfl\_23\_pollf),alternative="greater")



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 0.52873, p-value = 0.956  
## alternative hypothesis: greater

# Summary

tab\_model(mod\_surv\_22,mod\_median\_h\_22,mod\_flowering\_22\_pollf,mod\_nfl\_22\_pollf,  
 mod\_nfl\_22\_mh\_pollf,mod\_fruiting\_22,mod\_nfr\_22\_pollf,  
 mod\_frset\_22\_pollf,mod\_nseed\_22\_pollf,mod\_seedsperfl\_22\_pollf,  
 transform=NULL,show.intercept=F,show.ci=F,show.se=F,show.stat=F,  
 p.style="stars",  
 show.r2=T,show.icc=F,show.re.var=F,show.ngroups=F,digits=3,  
 dv.labels=c("Survival","Median h","Prob. of flowering",  
 "N flowers (1)","N flowers (2)","Prob. of fruiting",  
 "N fruits","Fruit set","N seeds","N seeds per flower"),  
 file="output/tables/summary\_2022.doc")

Survival

Median h

Prob. of flowering

N flowers (1)

N flowers (2)

Prob. of fruiting

N fruits

Fruit set

N seeds

N seeds per flower

Predictors

Log-Odds

Estimates

Log-Odds

Log-Mean

Log-Mean

Log-Odds

Log-Mean

Log-Odds

Log-Mean

Log-Mean

temp

-0.150 \*\*\*

0.047

-0.134 \*\*\*

0.079 \*\*\*

0.069 \*\*\*

0.133 \*\*

0.076 \*\*\*

-0.071 \*\*\*

0.015 \*\*\*

-0.027 \*\*\*

temp diff

0.014

-0.007

-0.012

-0.020 \*\*\*

-0.017 \*\*\*

-0.108 \*\*

-0.022 \*\*\*

-0.009 \*

-0.007 \*

0.012 \*\*

temp diff^2

-0.001

-0.001

0.001

-0.000 \*

-0.000

0.000

-0.000 \*

-0.000

-0.000

0.000

poll type[Selfpollinated]

-0.729 \*\*\*

-0.250 \*\*

-0.274 \*\*\*

-0.222 \*\*

0.169 \*

0.146 \*\*

0.373 \*\*\*

median h

0.114 \*\*\*

Observations

852

598

807

602

598

603

603

582

581

581

Marginal R2 / Conditional R2

0.351 / 0.499

0.015 / 0.127

0.204 / 0.626

0.333 / 0.380

0.382 / 0.428

0.223 / NA

0.282 / 0.327

0.111 / 0.295

0.043 / NA

0.094 / 0.140

* p<0.05   \*\* p<0.01   \*\*\* p<0.001

tab\_model(mod\_surv\_23,mod\_median\_h\_23\_pollf,mod\_flowering\_23\_pollf,  
 mod\_nfl\_23\_pollf,mod\_nfl\_23\_mh\_pollf,mod\_fruiting\_23,mod\_nfr\_23\_pollf,  
 mod\_frset\_23,mod\_nseed\_23\_pollf,mod\_seedsperfl\_23\_pollf,  
 transform=NULL,show.intercept=F,show.ci=F,show.se=F,show.stat=F,  
 p.style="stars",  
 show.r2=T,show.icc=F,show.re.var=F,show.ngroups=F,digits=3,  
 dv.labels=c("Survival","Median h","Prob. of flowering",  
 "N flowers (1)","N flowers (2)","Prob. of fruiting",  
 "N fruits","Fruit set","N seeds","N seeds per flower"),  
 file="output/tables/summary\_2023.doc")

Survival

Median h

Prob. of flowering

N flowers (1)

N flowers (2)

Prob. of fruiting

N fruits

Fruit set

N seeds

N seeds per flower

Predictors

Log-Odds

Estimates

Log-Odds

Log-Mean

Log-Mean

Log-Odds

Log-Mean

Log-Odds

Log-Mean

Log-Mean

temp

-0.142 \*\*\*

-0.056

-0.087 \*\*

0.022

0.022

-0.019

0.013

-0.064 \*\*\*

0.014

0.005

temp diff

0.005

0.009

0.015

-0.019 \*\*

-0.018 \*\*

-0.002

-0.024 \*\*

-0.013 \*

-0.004

0.001

temp diff^2

0.001 \*

-0.000

0.000

-0.000

-0.000

-0.000

-0.000

0.000

0.000

0.000

poll type[Selfpollinated]

-1.342 \*\*\*

-0.695 \*\*

-0.379 \*\*\*

-0.260 \*\*

-0.365 \*\*\*

0.253

0.563 \*\*\*

median h

0.129 \*\*\*

Observations

798

521

521

403

403

403

403

390

390

389

Marginal R2 / Conditional R2

0.289 / 0.491

0.046 / 0.260

0.101 / 0.366

0.047 / 0.465

0.201 / 0.493

0.008 / NA

0.062 / 0.422

0.098 / 0.162

0.025 / 0.089

0.079 / 0.200

* p<0.05   \*\* p<0.01   \*\*\* p<0.001

# Aster model

# HERE: check survival 22-23

Survival –> Any flowers –> Flowers –> Any Fruits –> Fruits –> Seeds

Survival (did the plant survive?) is Bernouilli.

Any flowers (did the plant produce flowers?) is Bernouilli.

Number of Flowers is zero-truncated negative binomial, because flower number cannot be zero if fruit production = 1.

Any fruits (did the plant produce fruits?) is Bernoulli.

Number of Fruits is zero-truncated negative binomial, because fruit number cannot be zero if fruit production = 1.

Number of seeds is negative binomial (?).

Remove 5 plants that were grazed before they peaked in 2022, 1 plant that was aborted, 3 plants where there is no info about number of seeds, and some plants where temp or temp\_diff (or both) were NA:

data\_transplants\_aster<-data\_transplants%>%  
 filter(comments\_peak!="Grazed before it peaked"|is.na(comments\_peak))%>%  
 filter(F\_NF\_A!="A"|is.na(F\_NF\_A))%>%  
 filter(!is.na(n\_tot\_seed))%>%  
 filter(!is.na(temp\_diff)&!is.na(temp))

# Something weird with surival, check later!  
data\_transplants\_aster$nfl<-with(data\_transplants\_aster,  
 ifelse(is.na(surv),NA,  
 ifelse(surv==0,0,  
 ifelse(flowering==0,0,  
 as.integer(data\_transplants\_aster$  
 tot\_fl\_bodies\_corr)))))  
data\_transplants\_aster$nfr<-with(data\_transplants\_aster,  
 ifelse(surv==0,0,  
 ifelse(flowering==0,0,  
 ifelse(fruiting==0,0,  
 as.integer(data\_transplants\_aster$  
 n\_tot\_fr)))))  
data\_transplants\_aster$nseed<-with(data\_transplants\_aster,  
 ifelse(surv==0,0,  
 ifelse(flowering==0,0,  
 ifelse(fruiting==0,0,  
 as.integer(data\_transplants\_aster$  
 n\_tot\_seed)))))  
data\_transplants\_aster$fl\_01<-with(data\_transplants\_aster,ifelse(nfl>0,1,0))  
data\_transplants\_aster$fr\_01<-with(data\_transplants\_aster,ifelse(nfr>0,1,0))  
  
vars <- c("surv", "fl\_01", "nfl", "fr\_01", "nfr","nseed") # Our fitness variables

Show the graphical model:

pred <- c(0,1,2,3,4,5) # specifies the predecessor structure of the graph  
  
foo <- c("root", vars)  
pvars <- foo[pred + 1]  
bar <- cbind(pvars, vars)  
colnames(bar) <- c("pred", "succ")  
bar

## pred succ   
## [1,] "root" "surv"   
## [2,] "surv" "fl\_01"  
## [3,] "fl\_01" "nfl"   
## [4,] "nfl" "fr\_01"  
## [5,] "fr\_01" "nfr"   
## [6,] "nfr" "nseed"

## 2022

Reshape data:

redata22 <- reshape(data.frame(data\_transplants\_aster%>%filter(year==2022)%>%  
 dplyr::select(plot,poll\_type,temp,temp\_diff,  
 surv,fl\_01,nfl,fr\_01,nfr,nseed)),   
 varying = list(vars),  
 direction = "long",   
 timevar = "varb", # reshape to long format  
 times = as.factor(vars), v.names = "resp")  
redata22 <- data.frame(redata22, root = 1) # This adds a variable root to the df  
# and makes all its values one (including for non-root nodes, but those values  
# are ignored by all aster package functions).  
  
# resp contains all of the data in the variables indicated by the string  
# vars packed into a single vector  
# varb indicates which original variable the corresponding element of  
# resp came from

names(data.frame(data\_transplants\_aster%>%filter(year==2022)%>%  
 dplyr::select(plot,poll\_type,temp,temp\_diff,  
 surv,flowering,nfl,fruiting,nfr,nseed)))

## [1] "plot" "poll\_type" "temp" "temp\_diff" "surv" "flowering"  
## [7] "nfl" "fruiting" "nfr" "nseed"

names(redata22)

## [1] "plot" "poll\_type" "temp" "temp\_diff" "varb" "resp"   
## [7] "id" "root"

nrow(data.frame(data\_transplants\_aster%>%filter(year==2022)%>%  
 dplyr::select(plot,poll\_type,temp,temp\_diff,  
 surv,flowering,nfl,fruiting,nfr,nseed)))

## [1] 843

nrow(redata22)

## [1] 5058

nrow(data.frame(data\_transplants\_aster%>%filter(year==2022)%>%  
 dplyr::select(plot,poll\_type,temp,temp\_diff,  
 surv,flowering,nfl,fruiting,nfr,nseed)))\*  
 length(vars)

## [1] 5058

sapply(redata22, class)

## plot poll\_type temp temp\_diff varb resp id root   
## "factor" "factor" "numeric" "numeric" "factor" "numeric" "integer" "numeric"

levels(redata22$varb)

## [1] "fl\_01" "fr\_01" "nfl" "nfr" "nseed" "surv"

length(unique(redata22$id))

## [1] 843

Look at Poisson distributions for flowers, fruits and seeds:

# For flowers  
nfl.dist\_22 <- (data\_transplants\_aster%>%  
 filter(year==2022))$nfl[(data\_transplants\_aster%>%  
 filter(year==2022))$fl\_01 == 1]  
length(nfl.dist\_22) # 597

## [1] 597

sum(nfl.dist\_22 == 0) # 0

## [1] 0

nfl.parms\_22 <- fitdistr(nfl.dist\_22, "poisson")  
nfl.parms\_22 # lambda = 19.142379

## lambda   
## 19.142379   
## ( 0.179065)

# For fruits  
nfr.dist\_22 <- (data\_transplants\_aster%>%  
 filter(year==2022))$nfr[(data\_transplants\_aster%>%  
 filter(year==2022))$fr\_01 == 1]  
length(nfr.dist\_22) # 581

## [1] 581

sum(nfr.dist\_22 == 0) # 0

## [1] 0

nfr.parms\_22 <- fitdistr(nfr.dist\_22, "poisson")  
nfr.parms\_22 # lambda = 16.8433735

## lambda   
## 16.8433735   
## ( 0.1702654)

# For seeds  
nseed.dist\_22 <- (data\_transplants\_aster%>%  
 filter(year==2022))$nseed[(data\_transplants\_aster%>%  
 filter(year==2022))$nfr > 0]  
length(nseed.dist\_22) # 581

## [1] 581

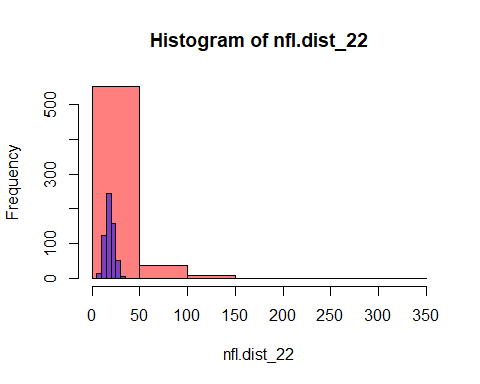
sum(nseed.dist\_22 == 0) # 3 with nfr>0 and nseed=0 - OK?

## [1] 3

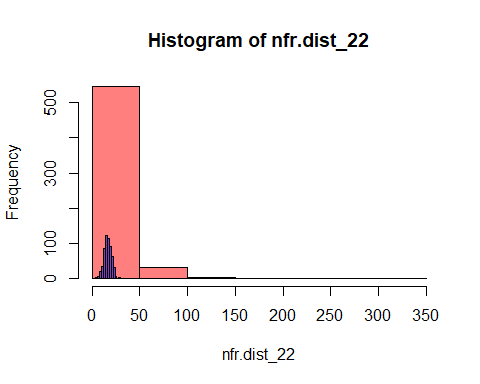
nseed.parms\_22 <- fitdistr(nseed.dist\_22, "poisson")  
nseed.parms\_22 # lambda = 85.342513

## lambda   
## 85.342513   
## ( 0.383261)

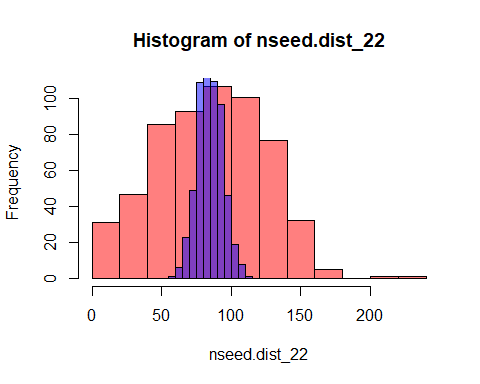
# Compare  
hist(nfl.dist\_22,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rpois(597, lambda = 19.142379),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



hist(nfr.dist\_22,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rpois(581, lambda = 16.8433735),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



hist(nseed.dist\_22,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rpois(581, lambda = 85.342513),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



Look at negative binomial distributions for flowers, fruits and seeds:

# For flowers  
nfl.parms2\_22 <- fitdistr(nfl.dist\_22, "negative binomial")  
nfl.parms2\_22 # size = 1.17467887

## size mu   
## 1.17467887 19.14237854   
## ( 0.06610473) ( 0.74470044)

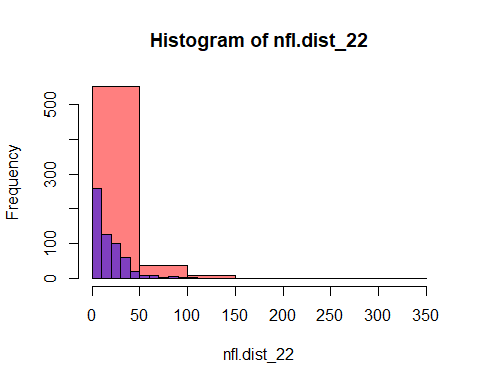
# For fruits  
nfr.parms2\_22 <- fitdistr(nfr.dist\_22, "negative binomial")  
nfr.parms2\_22 # size = 1.21845714

## size mu   
## 1.21845714 16.84337315   
## ( 0.07072351) ( 0.65554456)

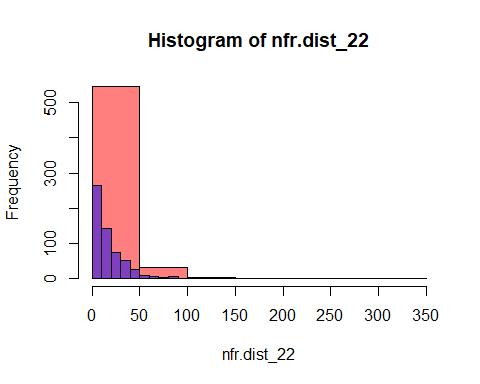
# For seeds  
nseed.parms2\_22 <- fitdistr(nseed.dist\_22, "negative binomial")  
# Warning message: In densfun(x, parm[1], parm[2], ...) : NaNs produced - OK?  
nseed.parms2\_22 # size = 3.555838

## size mu   
## 3.555838 85.342513   
## ( 0.216661) ( 1.916331)

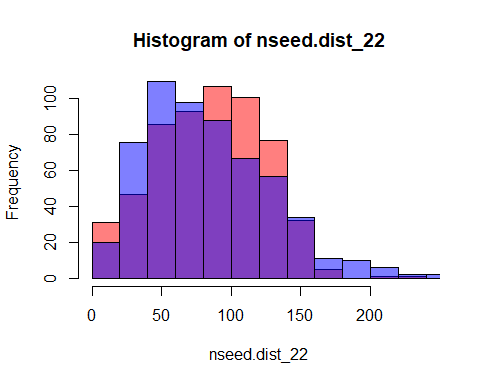
# Compare  
hist(nfl.dist\_22,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rnbinom(597, size = 1.17467887, mu = 19.14237854),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



hist(nfr.dist\_22,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rnbinom(581, size = 1.21845714, mu = 16.84337315),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



hist(nseed.dist\_22,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rnbinom(581, size = 3.555838, mu = 85.342513),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



Specifying families:

fam <- c(1,2,3,4,5,6) # specifies the one-parameter exponential families for the nodes  
famlist <- list(fam.bernoulli(), # surv  
 fam.bernoulli(), # fl\_01  
 fam.truncated.negative.binomial(size=1.17467887,  
 truncation = 0), # nfl  
 fam.bernoulli(),# fr\_01  
 fam.truncated.negative.binomial(size=1.21845714,  
 truncation = 0),# nfr  
 fam.negative.binomial(size=3.555838)# nseed  
 )  
vars[fam == 1]

## [1] "surv"

vars[fam == 2]

## [1] "fl\_01"

vars[fam == 3]

## [1] "nfl"

vars[fam == 4]

## [1] "fr\_01"

vars[fam == 5]

## [1] "nfr"

vars[fam == 6]

## [1] "nseed"

Specifying families (alternative, in case negative binomial really cannot be used with reaster):

famlist1 <- list(fam.bernoulli(), # surv  
 fam.bernoulli(), # fl\_01  
 fam.truncated.poisson(truncation = 0), # nfl  
 fam.bernoulli(),# fr\_01  
 fam.truncated.poisson(truncation = 0),# nfr  
 fam.poisson()# nseed  
 )  
vars[fam == 1]

## [1] "surv"

vars[fam == 2]

## [1] "fl\_01"

vars[fam == 3]

## [1] "nfl"

vars[fam == 4]

## [1] "fr\_01"

vars[fam == 5]

## [1] "nfr"

vars[fam == 6]

## [1] "nseed"

Creating the fitness variable of interest (and others):

surv01 <- grep("surv", as.character(redata22$varb))  
surv01 <- is.element(seq(along = redata22$varb), surv01)  
redata22 <- data.frame(redata22, surv01 = as.integer(surv01))  
  
fl01 <- grep("fl\_01", as.character(redata22$varb))  
fl01 <- is.element(seq(along = redata22$varb), fl01)  
redata22 <- data.frame(redata22, fl01 = as.integer(fl01))  
  
fls <- grep("nfl", as.character(redata22$varb))  
fls <- is.element(seq(along = redata22$varb), fls)  
redata22 <- data.frame(redata22, fls = as.integer(fls))  
  
fr01 <- grep("fr\_01", as.character(redata22$varb))  
fr01 <- is.element(seq(along = redata22$varb), fr01)  
redata22 <- data.frame(redata22, fr01 = as.integer(fr01))  
  
frs <- grep("nfr", as.character(redata22$varb))  
frs <- is.element(seq(along = redata22$varb), frs)  
redata22 <- data.frame(redata22, frs = as.integer(frs))  
  
seeds <- grep("nseed", as.character(redata22$varb))  
seeds <- is.element(seq(along = redata22$varb), seeds)  
redata22 <- data.frame(redata22, seeds = as.integer(seeds))  
  
names(redata22)

## [1] "plot" "poll\_type" "temp" "temp\_diff" "varb" "resp"   
## [7] "id" "root" "surv01" "fl01" "fls" "fr01"   
## [13] "frs" "seeds"

Center the predictors (I think we do not need to use standardized predictors in Aster models, but centering the predictors helps the model to converge):

redata22$temp\_c<-redata22$temp-mean(redata22$temp)  
redata22$temp\_diff\_c<-redata22$temp\_diff-mean(redata22$temp\_diff)

Scaled predictors (in case we need to use them):

redata22$temp\_s<-scale(redata22$temp)  
redata22$temp\_diff\_s<-scale(redata22$temp\_diff)

### Fit model

#### Check effect of pollination

# aster22\_poll\_nb <- reaster(fixed=resp ~ varb+seeds:(poll\_type),  
# random=list(plot= ~0 + surv01:plot + fl01:plot + fls:plot +   
# fr01:plot + frs:plot + seeds:plot), # OK for random effect?  
# pred, fam, famlist=famlist, varb, id, root,   
# data = redata22)

#summary(aster22\_poll\_nb) Gives error

aster22\_poll\_pois <- reaster(fixed=resp ~ varb+seeds:(poll\_type),  
 random=list(plot= ~0 + surv01:plot + fl01:plot + fls:plot +   
 fr01:plot + frs:plot + seeds:plot), # OK for random effect?  
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata22)

summary(aster22\_poll\_pois)

##   
## Call:  
## reaster.formula(fixed = resp ~ varb + seeds:(poll\_type), random = list(plot = ~0 +   
## surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot +   
## seeds:plot), pred = pred, fam = fam, varvar = varb, idvar = id,   
## root = root, famlist = famlist1, data = redata22)  
##   
##   
## Fixed Effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.457e+01 1.323e+00 -11.007 < 2e-16 \*\*\*  
## varbfr\_01 -1.818e+00 1.870e+00 -0.972 0.331   
## varbnfl 1.787e+01 1.860e+00 9.613 < 2e-16 \*\*\*  
## varbnfr 1.146e+01 1.858e+00 6.168 6.92e-10 \*\*\*  
## varbnseed 1.641e+01 1.858e+00 8.836 < 2e-16 \*\*\*  
## varbsurv 1.818e+01 1.952e+00 9.316 < 2e-16 \*\*\*  
## seeds:poll\_typeCrosspollinated 9.934e-05 9.151e-04 0.109 0.914   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Square Roots of Variance Components (P-values are one-tailed):  
## Estimate Std. Error z value Pr(>|z|)/2   
## plot 3.6870 0.3901 9.452 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

There seems to be no effect - suggest removing from further models. But we could also include it, given the results of previous models for fitness components.

#### Negative binomial for counts

Fitting the full Aster model (with plot as random):

# Full Aster model with seeds as the interaction for all fixed variables  
aster22 <- reaster(fixed=resp ~ varb+seeds:(temp\_s+temp\_diff\_s+I(temp\_diff\_s^2)),   
 random=list(plot= ~0 +   
 #surv01:plot + fl01:plot + fls:plot +   
 #fr01:plot + frs:plot + Including these gives error  
 seeds:plot), # OK for random effect?  
 pred, fam, famlist=famlist, varb, id, root,   
 data = redata22)  
# Include varb in the formula to get a different “intercept”   
# for each node in the graph (i.e. one for fitness\_01 and one for n\_seeds)  
# The categorical variable varb gets turned into as many dummy  
# variables as there are nodes in the graph, one is dropped, and the  
# “intercept” dummy variable (all components = 1) is added; the  
# effect is to provide a different intercept for each node.  
# Include interaction seeds: with all other variables  
# Seehttps://groups.google.com/g/aster-analysis-user-group/c/xwanCX8VOWQ/m/AcTnF3E3BAAJ   
# and code there.  
# According to https://www.stat.umn.edu/geyer/8931aster/slides/s1.pdf#page=55   
# (slides 55-71), we do this to have separate coefficients for the “layer”   
# of the graph consisting of terminal nodes (as in aster models, regression  
# coefficients “for” a node of the graph also influence all “earlier” nodes  
# To estimate the effects of each predictor on lifetime fitness, each predictor  
# was fit at the level of total seed set in the model (Shaw et al. 2008)

summary(aster22,   
 show.graph = TRUE # Table about the graph structure in the printout  
 ) # We can use info.tol = 1e-11 To help the model to converge

##   
## Call:  
## reaster.formula(fixed = resp ~ varb + seeds:(temp\_s + temp\_diff\_s +   
## I(temp\_diff\_s^2)), random = list(plot = ~0 + seeds:plot),   
## pred = pred, fam = fam, varvar = varb, idvar = id, root = root,   
## famlist = famlist, data = redata22)  
##   
##   
## Fixed Effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.5207657 0.0928892 -27.137 <2e-16 \*\*\*  
## varbfr\_01 -3.8930027 0.1120493 -34.744 <2e-16 \*\*\*  
## varbnfl 4.0999782 0.0940638 43.587 <2e-16 \*\*\*  
## varbnfr 1.9293748 0.0940714 20.510 <2e-16 \*\*\*  
## varbnseed 2.9890511 0.0929380 32.162 <2e-16 \*\*\*  
## varbsurv 4.7379909 0.2149585 22.041 <2e-16 \*\*\*  
## seeds:temp\_s 0.0004824 0.0005384 0.896 0.370   
## seeds:temp\_diff\_s -0.0007398 0.0004592 -1.611 0.107   
## seeds:I(temp\_diff\_s^2) -0.0002060 0.0002384 -0.864 0.388   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Square Roots of Variance Components (P-values are one-tailed):  
## Estimate Std. Error z value Pr(>|z|)/2  
## plot 0.0006147 0.0005279 1.164 0.122

# (see https://groups.google.com/g/aster-analysis-user-group/c/xXXROez8aAA/m/BrNE6whiIEQJ)

Fitting the full Aster model (without plot as random):

aster22\_noplot <- aster(resp ~ varb+seeds:(temp\_s+temp\_diff\_s+I(temp\_diff\_s^2)),   
 pred, fam, famlist=famlist, varb, id, root,   
 data = redata22)

Compare:

anova(aster22\_noplot,aster22)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2))  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), ~0 + seeds:plot  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 9 0 -130667   
## 2 9 1 -130667 0 1 0.65545 0.20909

LRT not significant, does this mean that we can discard plot as random effect and use aster instead of reaster?

#### (USE) Poisson for counts

Fitting the full Aster model (with plot as random):

aster22\_pois <- reaster(fixed=resp ~ varb+seeds:(temp\_s+temp\_diff\_s+  
 I(temp\_diff\_s^2)),   
 random=list(plot= ~0 + surv01:plot + fl01:plot + fls:plot +   
 fr01:plot + frs:plot + seeds:plot),   
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata22)

summary(aster22\_pois)

##   
## Call:  
## reaster.formula(fixed = resp ~ varb + seeds:(temp\_s + temp\_diff\_s +   
## I(temp\_diff\_s^2)), random = list(plot = ~0 + surv01:plot +   
## fl01:plot + fls:plot + fr01:plot + frs:plot + seeds:plot),   
## pred = pred, fam = fam, varvar = varb, idvar = id, root = root,   
## famlist = famlist1, data = redata22)  
##   
##   
## Fixed Effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.456e+01 1.323e+00 -11.009 < 2e-16 \*\*\*  
## varbfr\_01 -1.818e+00 1.869e+00 -0.973 0.3307   
## varbnfl 1.787e+01 1.859e+00 9.615 < 2e-16 \*\*\*  
## varbnfr 1.146e+01 1.858e+00 6.169 6.88e-10 \*\*\*  
## varbnseed 1.641e+01 1.857e+00 8.838 < 2e-16 \*\*\*  
## varbsurv 1.818e+01 1.951e+00 9.318 < 2e-16 \*\*\*  
## seeds:temp\_s -8.465e-05 1.006e-03 -0.084 0.9330   
## seeds:temp\_diff\_s -1.419e-03 6.821e-04 -2.080 0.0375 \*   
## seeds:I(temp\_diff\_s^2) -2.579e-04 3.711e-04 -0.695 0.4870   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Square Roots of Variance Components (P-values are one-tailed):  
## Estimate Std. Error z value Pr(>|z|)/2   
## plot 3.6852 0.3899 9.451 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Fitting the full Aster model (without plot as random):

aster22\_noplot\_pois <- aster(resp ~ varb+seeds:(temp\_s+temp\_diff\_s+  
 I(temp\_diff\_s^2)),  
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata22)

summary(aster22\_noplot\_pois)

##   
## Call:  
## aster.formula(formula = resp ~ varb + seeds:(temp\_s + temp\_diff\_s +   
## I(temp\_diff\_s^2)), pred = pred, fam = fam, varvar = varb,   
## idvar = id, root = root, data = redata22, famlist = famlist1)  
##   
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.751e+01 1.967e-01 -89.042 < 2e-16 \*\*\*  
## varbfr\_01 -1.711e+00 2.636e-01 -6.493 8.42e-11 \*\*\*  
## varbnfl 2.111e+01 2.053e-01 102.832 < 2e-16 \*\*\*  
## varbnfr 1.627e+01 1.983e-01 82.068 < 2e-16 \*\*\*  
## varbnseed 1.914e+01 1.967e-01 97.270 < 2e-16 \*\*\*  
## varbsurv 1.973e+01 2.762e-01 71.447 < 2e-16 \*\*\*  
## seeds:temp\_s 1.346e-03 6.922e-04 1.944 0.0519 .   
## seeds:temp\_diff\_s -1.598e-03 6.701e-04 -2.384 0.0171 \*   
## seeds:I(temp\_diff\_s^2) -4.658e-04 3.356e-04 -1.388 0.1651   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Compare:

anova(aster22\_noplot\_pois,aster22\_pois)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2))  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 9 0 136229   
## 2 9 1 144788 0 1 8559 0

LRT significant, I guess this means that we cannot discard plot as random effect and need to use reaster?

### LRTs

The Aster manual advises to not look at significances in the summary, but instead compare models with a likelihood ratio test to evaluate significance of each term. The comparison between two or more models . . . will only be valid if they are (1) fitted to the same dataset, (2) models are nested, (3) models are of the same type (all conditional or all unconditional), (4) have the same dependence graph and exponential families. None of this is currently checked.

Determine the significance of fixed effects by comparing submodels without the term of interest to the full model using LRTs.

Use LRTs comparing submodels to fuller models to test each predictor of interest.

Submodels:

Testing the significance of seeds:temp\_s

aster22\_pois\_1 <- reaster(fixed=resp ~ varb+seeds:(temp\_diff\_s+I(temp\_diff\_s^2)),   
 random=list(plot= ~0 + surv01:plot + fl01:plot +   
 fls:plot + fr01:plot + frs:plot +   
 seeds:plot),   
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata22)

anova(aster22\_pois\_1,aster22\_pois)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_diff\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 8 1 144788   
## 2 9 1 144788 1 0 0.0074316 0.9313

Testing the significance of seeds:temp\_diff\_s

aster22\_pois\_2 <- reaster(fixed=resp ~ varb+seeds:(temp\_s+I(temp\_diff\_s^2)),   
 random=list(plot= ~0 + surv01:plot + fl01:plot + fls:plot +   
 fr01:plot + frs:plot + seeds:plot),   
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata22)

anova(aster22\_pois\_2,aster22\_pois)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 8 1 144783   
## 2 9 1 144788 1 0 4.3064 0.03797

Testing the significance of seeds:I(temp\_diff\_s^2)

aster22\_pois\_3 <- reaster(fixed=resp ~ varb+seeds:(temp\_s+temp\_diff\_s),  
 random=list(plot= ~0 + surv01:plot + fl01:plot +  
 fls:plot + fr01:plot + frs:plot +  
 seeds:plot),  
 pred, fam, famlist=famlist1, varb, id, root,  
 data = redata22)

anova(aster22\_pois\_3,aster22\_pois)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_s + temp\_diff\_s), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 8 1 144787   
## 2 9 1 144788 1 0 0.4931 0.48255

### Predictions

I need to use the aster (not reaster) models for predictions, see <https://groups.google.com/g/aster-analysis-user-group/c/oWj8z6snqnQ/m/hJ5FVjW8GQMJ>

“Values were predicted using a fixed-effect aster model, rather than the random-effect model used to test significance of fixed effects, as parameter estimates from random-effects models are difficult to interpret (Magnoli 2020)”

#### seeds:temp\_s

#quantile(redata22$temp\_s,probs = c(0.05,0.95))  
# Prediction for the range with the highest 5% removed  
# NOT USED SO FAR  
  
# make prediction df  
aster22\_predict1 <- data.frame(  
 expand\_grid(temp\_s = seq(from = range(redata22$temp\_s)[1],  
 to = range(redata22$temp\_s)[2],  
 length.out = 100)),  
 #OK from here to below? Not sure...  
 temp\_diff\_s = 0, # scaled so mean=0  
 surv = 1,  
 fl\_01 = 1,  
 nfl = 1,  
 fr\_01 = 1,  
 nfr = 1,  
 nseed = 0,  
 root = 1  
)  
  
# reshape data to long format  
aster22\_predict1\_long <- reshape(as.data.frame(aster22\_predict1),   
 varying = list(vars), direction = "long",   
 timevar = "varb", times = as.factor(vars),   
 v.names = "resp")  
  
# add artifice  
aster22\_predict1\_long$seeds <- as.integer(ifelse(  
 aster22\_predict1\_long$varb == "nseed", 1, 0))  
  
aster22\_predict1.df <- aster22\_predict1\_long  
  
# fixed model predictions  
aster22.p1 <- predict(aster22\_noplot\_pois, # Use aster model and not reaster!  
 newdata = aster22\_predict1\_long,  
 varvar = varb,idvar = id, root = root,  
 #info.tol = 1e-11,   
 se.fit = T)  
  
aster22\_predict1.df$fit <- aster22.p1$fit  
aster22\_predict1.df$se <- aster22.p1$se.fit  
  
aster22\_predict1.df.seeds <- aster22\_predict1.df %>%  
 filter(varb == "nseed")  
  
aster22\_predict1.df.seeds<-aster22\_predict1.df.seeds%>%  
 mutate(ymin=fit-se,ymax=fit+se)  
  
original\_mean\_temp\_22<-as.numeric(data\_transplants\_aster%>%  
 filter(year==2022)%>%  
 summarise(mean\_temp=mean(temp)))  
original\_sd\_temp\_22<-as.numeric(data\_transplants\_aster%>%  
 filter(year==2022)%>%  
 summarise(sd\_temp=sd(temp)))  
  
aster22\_predict1.df.seeds<-aster22\_predict1.df.seeds%>%  
 mutate(temp=temp\_s\*original\_sd\_temp\_22+original\_mean\_temp\_22)

#### seeds:temp\_diff\_s

#quantile(redata22$temp\_diff\_s,probs = c(0.05,0.95))  
# Prediction for the range with the highest 5% removed  
# NOT USED SO FAR  
  
# make prediction df  
aster22\_predict2 <- data.frame(  
 expand\_grid(temp\_diff\_s = seq(from = range(redata22$temp\_diff\_s)[1],  
 to = range(redata22$temp\_diff\_s)[2],  
 length.out = 100)),  
 #OK from here to below? Not sure...  
 temp\_s = 0, # scaled so mean=0  
 surv = 1,  
 fl\_01 = 1,  
 nfl = 1,  
 fr\_01 = 1,  
 nfr = 1,  
 nseed = 0,  
 root = 1  
)  
  
# reshape data to long format  
aster22\_predict2\_long <- reshape(as.data.frame(aster22\_predict2),   
 varying = list(vars), direction = "long",   
 timevar = "varb", times = as.factor(vars),   
 v.names = "resp")  
  
# add artifice  
aster22\_predict2\_long$seeds <- as.integer(ifelse(  
 aster22\_predict2\_long$varb == "nseed", 1, 0))  
  
aster22\_predict2.df <- aster22\_predict2\_long  
  
# fixed model predictions  
aster22.p2 <- predict(aster22\_noplot\_pois, # Use aster model and not reaster!  
 newdata = aster22\_predict2\_long,  
 varvar = varb,idvar = id, root = root,  
 #info.tol = 1e-11,   
 se.fit = T)  
  
aster22\_predict2.df$fit <- aster22.p2$fit  
aster22\_predict2.df$se <- aster22.p2$se.fit  
  
aster22\_predict2.df.seeds <- aster22\_predict2.df %>%  
 filter(varb == "nseed")  
  
aster22\_predict2.df.seeds<-aster22\_predict2.df.seeds%>%  
 mutate(ymin=fit-se,ymax=fit+se)  
  
original\_mean\_temp\_diff\_22<-as.numeric(data\_transplants\_aster%>%  
 filter(year==2022)%>%  
 summarise(mean\_temp\_diff=mean(temp\_diff)))  
original\_sd\_temp\_diff\_22<-as.numeric(data\_transplants\_aster%>%  
 filter(year==2022)%>%  
 summarise(sd\_temp\_diff=sd(temp\_diff)))  
  
aster22\_predict2.df.seeds<-aster22\_predict2.df.seeds%>%  
 mutate(temp\_diff=temp\_diff\_s\*original\_sd\_temp\_diff\_22+  
 original\_mean\_temp\_diff\_22)

## 2023

## HERE: check survival 22-23

Reshape data:

redata23 <- reshape(data.frame(data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv))%>%  
 # CHECK: plants that we know if they survived or not in 2023  
 dplyr::select(plot,poll\_type,temp,temp\_diff,  
 surv,fl\_01,nfl,fr\_01,nfr,nseed)),   
 varying = list(vars),  
 direction = "long",   
 timevar = "varb",   
 times = as.factor(vars), v.names = "resp")  
redata23 <- data.frame(redata23, root = 1)

names(data.frame(data\_transplants\_aster%>%filter(year==2023&!is.na(surv))%>%  
 dplyr::select(plot,poll\_type,temp,temp\_diff,  
 surv,flowering,nfl,fruiting,nfr,nseed)))

## [1] "plot" "poll\_type" "temp" "temp\_diff" "surv" "flowering"  
## [7] "nfl" "fruiting" "nfr" "nseed"

names(redata23)

## [1] "plot" "poll\_type" "temp" "temp\_diff" "varb" "resp"   
## [7] "id" "root"

nrow(data.frame(data\_transplants\_aster%>%filter(year==2023&!is.na(surv))%>%  
 dplyr::select(plot,poll\_type,temp,temp\_diff,  
 surv,flowering,nfl,fruiting,nfr,nseed)))

## [1] 798

nrow(redata23)

## [1] 4788

nrow(data.frame(data\_transplants\_aster%>%filter(year==2023&!is.na(surv))%>%  
 dplyr::select(plot,poll\_type,temp,temp\_diff,  
 surv,flowering,nfl,fruiting,nfr,nseed)))\*  
 length(vars)

## [1] 4788

sapply(redata23, class)

## plot poll\_type temp temp\_diff varb resp id root   
## "factor" "factor" "numeric" "numeric" "factor" "numeric" "integer" "numeric"

levels(redata23$varb)

## [1] "fl\_01" "fr\_01" "nfl" "nfr" "nseed" "surv"

length(unique(redata23$id))

## [1] 798

Look at Poisson distributions for flowers, fruits and seeds:

# For flowers  
nfl.dist\_23 <- (data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv)))$nfl[(data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv)))$fl\_01 == 1]  
length(nfl.dist\_23) # 403

## [1] 403

sum(nfl.dist\_23 == 0) # 0

## [1] 0

nfl.parms\_23 <- fitdistr(nfl.dist\_23, "poisson")  
nfl.parms\_23 # lambda = 14.3176179

## lambda   
## 14.3176179   
## ( 0.1884876)

# For fruits  
nfr.dist\_23 <- (data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv)))$nfr[(data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv)))$fr\_01 == 1]  
length(nfr.dist\_23) # 390

## [1] 390

sum(nfr.dist\_23 == 0) # 0

## [1] 0

nfr.parms\_23 <- fitdistr(nfr.dist\_23, "poisson")  
nfr.parms\_23 # lambda = 12.4282051

## lambda   
## 12.4282051   
## ( 0.1785138)

# For seeds  
nseed.dist\_23 <- (data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv)))$nseed[(data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv)))$nfr > 0]  
length(nseed.dist\_23) # 390

## [1] 390

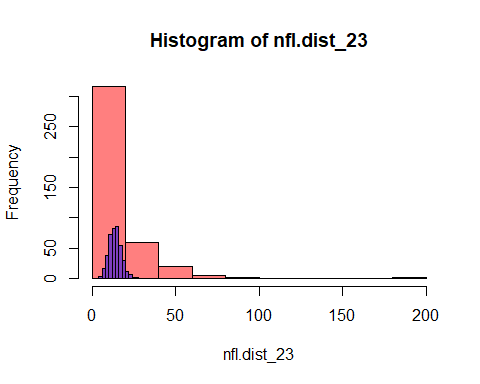
sum(nseed.dist\_23 == 0) # 61 with nfr>0 and nseed=0 - OK?

## [1] 61

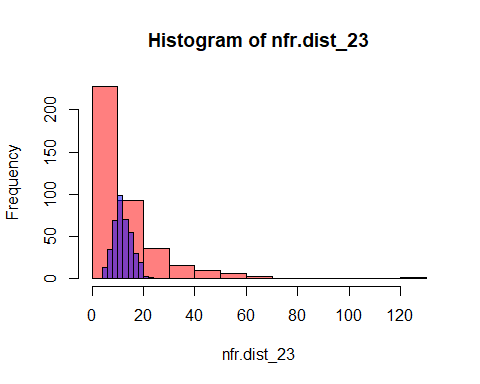
nseed.parms\_23 <- fitdistr(nseed.dist\_23, "poisson")  
nseed.parms\_23 # lambda = 53.0179487

## lambda   
## 53.0179487   
## ( 0.3687051)

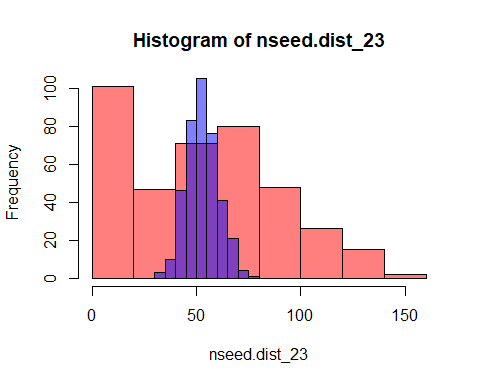
# Compare  
hist(nfl.dist\_23,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rpois(403, lambda = 14.3176179),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



hist(nfr.dist\_23,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rpois(390, lambda = 12.4282051),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



hist(nseed.dist\_23,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rpois(390, lambda = 53.0179487),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



Look at negative binomial distributions for flowers, fruits and seeds:

# For flowers  
nfl.parms2\_23 <- fitdistr(nfl.dist\_23, "negative binomial")  
nfl.parms2\_23 # size = 1.23310598

## size mu   
## 1.23310598 14.31761810   
## ( 0.08801545) ( 0.66935724)

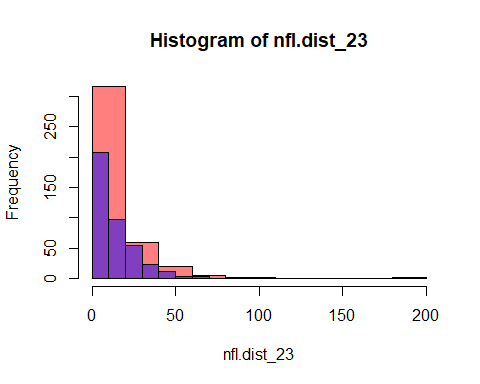
# For fruits  
nfr.parms2\_23 <- fitdistr(nfr.dist\_23, "negative binomial")  
nfr.parms2\_23 # size = 1.21845714

## size mu   
## 1.32845749 12.42820377   
## ( 0.09912671) ( 0.57445302)

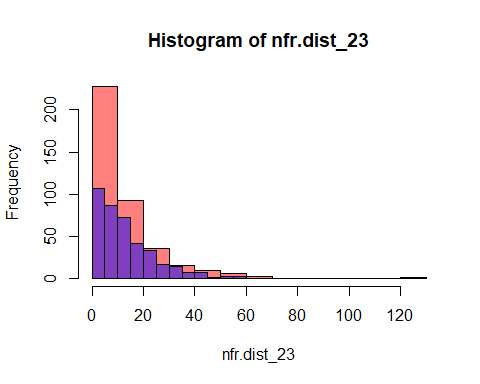
# For seeds  
nseed.parms2\_23 <- fitdistr(nseed.dist\_23, "negative binomial")  
# Warning message: In densfun(x, parm[1], parm[2], ...) : NaNs produced - OK?  
nseed.parms2\_23 # size = 3.555838

## size mu   
## 0.7177442 53.0179487   
## ( 0.0534137) ( 3.1902573)

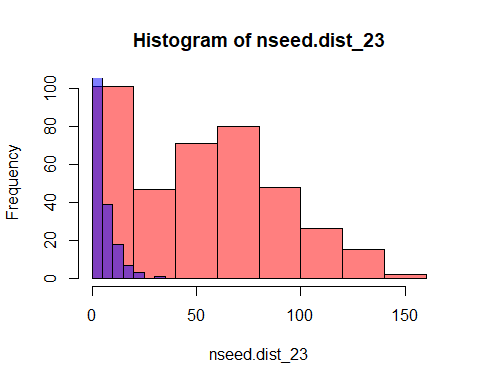
# Compare  
hist(nfl.dist\_23,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rnbinom(403, size = 1.23310598, mu = 14.31761810),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



hist(nfr.dist\_23,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rnbinom(390, size = 1.32845749, mu = 12.42820377),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



hist(nseed.dist\_23,col = rgb(1, 0, 0, 0.5),breaks=10)  
hist(rnbinom(390, size = 0.7177442, mu = 3.1902573),col = rgb(0, 0, 1, 0.5),  
 add=T,breaks=10)



Specifying families:

fam <- c(1,2,3,4,5,6) # specifies the one-parameter exponential families for the nodes  
famlist <- list(fam.bernoulli(), # surv  
 fam.bernoulli(), # fl\_01  
 fam.truncated.negative.binomial(size=1.23310598,  
 truncation = 0), # nfl  
 fam.bernoulli(),# fr\_01  
 fam.truncated.negative.binomial(size=1.32845749,  
 truncation = 0),# nfr  
 fam.negative.binomial(size=0.7177442)# nseed  
 )  
vars[fam == 1]

## [1] "surv"

vars[fam == 2]

## [1] "fl\_01"

vars[fam == 3]

## [1] "nfl"

vars[fam == 4]

## [1] "fr\_01"

vars[fam == 5]

## [1] "nfr"

vars[fam == 6]

## [1] "nseed"

Creating the fitness variable of interest (and others):

surv01 <- grep("surv", as.character(redata23$varb))  
surv01 <- is.element(seq(along = redata23$varb), surv01)  
redata23 <- data.frame(redata23, surv01 = as.integer(surv01))  
  
fl01 <- grep("fl\_01", as.character(redata23$varb))  
fl01 <- is.element(seq(along = redata23$varb), fl01)  
redata23 <- data.frame(redata23, fl01 = as.integer(fl01))  
  
fls <- grep("nfl", as.character(redata23$varb))  
fls <- is.element(seq(along = redata23$varb), fls)  
redata23 <- data.frame(redata23, fls = as.integer(fls))  
  
fr01 <- grep("fr\_01", as.character(redata23$varb))  
fr01 <- is.element(seq(along = redata23$varb), fr01)  
redata23 <- data.frame(redata23, fr01 = as.integer(fr01))  
  
frs <- grep("nfr", as.character(redata23$varb))  
frs <- is.element(seq(along = redata23$varb), frs)  
redata23 <- data.frame(redata23, frs = as.integer(frs))  
  
seeds <- grep("nseed", as.character(redata23$varb))  
seeds <- is.element(seq(along = redata23$varb), seeds)  
redata23 <- data.frame(redata23, seeds = as.integer(seeds))  
  
names(redata23)

## [1] "plot" "poll\_type" "temp" "temp\_diff" "varb" "resp"   
## [7] "id" "root" "surv01" "fl01" "fls" "fr01"   
## [13] "frs" "seeds"

Center the predictors (I think we do not need to use standardized predictors in Aster models, but centering the predictors helps the model to converge):

redata23$temp\_c<-redata23$temp-mean(redata23$temp)  
redata23$temp\_diff\_c<-redata23$temp\_diff-mean(redata23$temp\_diff)

Scaled predictors (in case we need to use them):

redata23$temp\_s<-scale(redata23$temp)  
redata23$temp\_diff\_s<-scale(redata23$temp\_diff)

### Fit model

#### Check effect of pollination

aster23\_poll\_pois <- reaster(fixed=resp ~ varb+seeds:(poll\_type),  
 random=list(plot= ~0 + seeds:plot), # Using more random effects gives error  
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata23)

summary(aster23\_poll\_pois)

##   
## Call:  
## reaster.formula(fixed = resp ~ varb + seeds:(poll\_type), random = list(plot = ~0 +   
## seeds:plot), pred = pred, fam = fam, varvar = varb, idvar = id,   
## root = root, famlist = famlist1, data = redata23)  
##   
##   
## Fixed Effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.245e+01 2.151e-01 -57.868 < 2e-16 \*\*\*  
## varbfr\_01 -1.944e+00 2.829e-01 -6.873 6.27e-12 \*\*\*  
## varbnfl 1.573e+01 2.268e-01 69.379 < 2e-16 \*\*\*  
## varbnfr 1.170e+01 2.176e-01 53.778 < 2e-16 \*\*\*  
## varbnseed 1.389e+01 2.154e-01 64.487 < 2e-16 \*\*\*  
## varbsurv 1.229e+01 2.744e-01 44.779 < 2e-16 \*\*\*  
## seeds:poll\_typeCrosspollinated -6.217e-04 1.668e-03 -0.373 0.709   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Square Roots of Variance Components (P-values are one-tailed):  
## Estimate Std. Error z value Pr(>|z|)/2   
## plot 0.014767 0.005005 2.95 0.00159 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

There seems to be no effect - remove from further models.

#### Negative binomial for counts

Fitting the full Aster model (with plot as random):

aster23 <- reaster(fixed=resp ~ varb+seeds:(temp\_s+temp\_diff\_s+I(temp\_diff\_s^2)),   
 random=list(plot= ~0 +   
 #surv01:plot + fl01:plot + fls:plot +  
 #fr01:plot + frs:plot +  
 seeds:plot), # Including more random effects gives error  
 pred, fam, famlist=famlist, varb, id, root,   
 data = redata23)

summary(aster23,   
 show.graph = TRUE # Table about the graph structure in the printout  
 ) # We can use info.tol = 1e-11 To help the model to converge

##   
## Call:  
## reaster.formula(fixed = resp ~ varb + seeds:(temp\_s + temp\_diff\_s +   
## I(temp\_diff\_s^2)), random = list(plot = ~0 + seeds:plot),   
## pred = pred, fam = fam, varvar = varb, idvar = id, root = root,   
## famlist = famlist, data = redata23)  
##   
##   
## Fixed Effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.0171809 0.1182289 -17.062 < 2e-16 \*\*\*  
## varbfr\_01 -3.7257417 0.1403982 -26.537 < 2e-16 \*\*\*  
## varbnfl 3.5536748 0.1201764 29.570 < 2e-16 \*\*\*  
## varbnfr 1.8493894 0.1188513 15.561 < 2e-16 \*\*\*  
## varbnseed 2.8558305 0.1183761 24.125 < 2e-16 \*\*\*  
## varbsurv 1.8569951 0.2073931 8.954 < 2e-16 \*\*\*  
## seeds:temp\_s -0.0049991 0.0018855 -2.651 0.00802 \*\*   
## seeds:temp\_diff\_s 0.0006496 0.0010905 0.596 0.55135   
## seeds:I(temp\_diff\_s^2) 0.0006323 0.0005142 1.230 0.21883   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Square Roots of Variance Components (P-values are one-tailed):  
## Estimate Std. Error z value Pr(>|z|)/2   
## plot 0.005280 0.002259 2.337 0.00971 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# (see https://groups.google.com/g/aster-analysis-user-group/c/xXXROez8aAA/m/BrNE6whiIEQJ)

Fitting the full Aster model (without plot as random):

aster23\_noplot <- aster(resp ~ varb+seeds:(temp\_s+temp\_diff\_s+I(temp\_diff\_s^2)),   
 pred, fam, famlist=famlist, varb, id, root,   
 data = redata23)

Compare:

anova(aster23\_noplot,aster23)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2))  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), ~0 + seeds:plot  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 9 0 -44953   
## 2 9 1 -31098 0 1 13855 0

LRT not significant, does this mean that we can discard plot as random effect and use aster instead of reaster?

#### (USE) Poisson for counts

Fitting the full Aster model (with plot as random):

aster23\_pois <- reaster(fixed=resp ~ varb+seeds:(temp\_s+temp\_diff\_s+I(temp\_diff\_s^2)),   
 random=list(plot= ~0 + surv01:plot + fl01:plot + fls:plot +   
 fr01:plot + frs:plot + seeds:plot),   
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata23)

summary(aster23\_pois)

##   
## Call:  
## reaster.formula(fixed = resp ~ varb + seeds:(temp\_s + temp\_diff\_s +   
## I(temp\_diff\_s^2)), random = list(plot = ~0 + surv01:plot +   
## fl01:plot + fls:plot + fr01:plot + frs:plot + seeds:plot),   
## pred = pred, fam = fam, varvar = varb, idvar = id, root = root,   
## famlist = famlist1, data = redata23)  
##   
##   
## Fixed Effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.045e+01 1.679e+00 -6.223 4.88e-10 \*\*\*  
## varbfr\_01 -1.373e+00 2.376e+00 -0.578 0.56344   
## varbnfl 1.316e+01 2.335e+00 5.638 1.72e-08 \*\*\*  
## varbnfr 6.949e+00 2.369e+00 2.933 0.00335 \*\*   
## varbnseed 1.220e+01 2.363e+00 5.163 2.42e-07 \*\*\*  
## varbsurv 1.110e+01 2.305e+00 4.816 1.46e-06 \*\*\*  
## seeds:temp\_s -7.268e-03 2.628e-03 -2.766 0.00567 \*\*   
## seeds:temp\_diff\_s 5.772e-04 1.430e-03 0.404 0.68657   
## seeds:I(temp\_diff\_s^2) 1.094e-03 7.412e-04 1.475 0.14015   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Square Roots of Variance Components (P-values are one-tailed):  
## Estimate Std. Error z value Pr(>|z|)/2   
## plot 4.4112 0.4836 9.122 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Fitting the full Aster model (without plot as random):

aster23\_noplot\_pois <- aster(resp ~ varb+seeds:(temp\_s+temp\_diff\_s+I(temp\_diff\_s^2)),  
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata23)

summary(aster23\_noplot\_pois)

##   
## Call:  
## aster.formula(formula = resp ~ varb + seeds:(temp\_s + temp\_diff\_s +   
## I(temp\_diff\_s^2)), pred = pred, fam = fam, varvar = varb,   
## idvar = id, root = root, data = redata23, famlist = famlist1)  
##   
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.249e+01 2.153e-01 -58.003 < 2e-16 \*\*\*  
## varbfr\_01 -1.956e+00 2.837e-01 -6.894 5.44e-12 \*\*\*  
## varbnfl 1.577e+01 2.270e-01 69.500 < 2e-16 \*\*\*  
## varbnfr 1.174e+01 2.178e-01 53.913 < 2e-16 \*\*\*  
## varbnseed 1.393e+01 2.155e-01 64.667 < 2e-16 \*\*\*  
## varbsurv 1.233e+01 2.746e-01 44.900 < 2e-16 \*\*\*  
## seeds:temp\_s -9.383e-03 1.685e-03 -5.569 2.57e-08 \*\*\*  
## seeds:temp\_diff\_s 1.309e-03 1.489e-03 0.879 0.3792   
## seeds:I(temp\_diff\_s^2) 1.497e-03 6.641e-04 2.253 0.0242 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Compare:

anova(aster23\_noplot\_pois,aster23\_pois)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2))  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 9 0 48182   
## 2 9 1 52249 0 1 4067.6 0

LRT significant, I guess this means that we cannot discard plot as random effect and need to use reaster?

### LRTs

The Aster manual advises to not look at significances in the summary, but instead compare models with a likelihood ratio test to evaluate significance of each term. The comparison between two or more models . . . will only be valid if they are (1) fitted to the same dataset, (2) models are nested, (3) models are of the same type (all conditional or all unconditional), (4) have the same dependence graph and exponential families. None of this is currently checked.

Determine the significance of fixed effects by comparing submodels without the term of interest to the full model using LRTs.

Use LRTs comparing submodels to fuller models to test each predictor of interest.

Submodels:

Testing the significance of seeds:temp\_s

aster23\_pois\_modif <- reaster(fixed=resp ~ varb+seeds:(temp\_s+temp\_diff\_s+I(temp\_diff\_s^2)),   
 random=list(plot= ~0 +   
 #surv01:plot + fl01:plot + fls:plot +   
 #fr01:plot + frs:plot +   
 seeds:plot),   
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata23)  
aster23\_pois\_1 <- reaster(fixed=resp ~ varb+seeds:(temp\_diff\_s+I(temp\_diff\_s^2)),   
 random=list(plot= ~0 +   
 #surv01:plot + fl01:plot +   
 #fls:plot + fr01:plot + frs:plot +   
 seeds:plot),   
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata23)  
# aster23\_pois\_1 gives error with all random factors,  
# so we need a modified version of aster23\_pois  
# with a similar model matrix for random effects

anova(aster23\_pois\_1,aster23\_pois\_modif)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_diff\_s + I(temp\_diff\_s^2)), ~0 + seeds:plot  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), ~0 + seeds:plot  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 8 1 48228   
## 2 9 1 48239 1 0 11.499 0.00069644

Testing the significance of seeds:temp\_diff\_s

aster23\_pois\_2 <- reaster(fixed=resp ~ varb+seeds:(temp\_s+I(temp\_diff\_s^2)),   
 random=list(plot= ~0 + surv01:plot + fl01:plot + fls:plot +   
 fr01:plot + frs:plot + seeds:plot),   
 pred, fam, famlist=famlist1, varb, id, root,   
 data = redata23)

anova(aster23\_pois\_2,aster23\_pois)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 8 1 52249   
## 2 9 1 52249 1 0 0.15979 0.68935

Testing the significance of seeds:I(temp\_diff\_s^2)

aster23\_pois\_3 <- reaster(fixed=resp ~ varb+seeds:(temp\_s+temp\_diff\_s),  
 random=list(plot= ~0 + surv01:plot + fl01:plot +  
 fls:plot + fr01:plot + frs:plot +  
 seeds:plot),  
 pred, fam, famlist=famlist1, varb, id, root,  
 data = redata23)

anova(aster23\_pois\_3,aster23\_pois)

## Analysis of Deviance Table  
##   
## Model 1: resp ~ varb + seeds:(temp\_s + temp\_diff\_s), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Model 2: resp ~ varb + seeds:(temp\_s + temp\_diff\_s + I(temp\_diff\_s^2)), c("~0 + surv01:plot + fl01:plot + fls:plot + fr01:plot + frs:plot + ", " seeds:plot")  
## Mod Df Fix Mod Df Rand Mod Dev Df Fix Df Rand Deviance P-value  
## 1 8 1 52247   
## 2 9 1 52249 1 0 2.07 0.15022

### Predictions

#### seeds:temp\_s

#quantile(redata23$temp\_s,probs = c(0.05,0.95))  
# Prediction for the range with the highest 5% removed  
# NOT USED SO FAR  
  
# make prediction df  
aster23\_predict1 <- data.frame(  
 expand\_grid(temp\_s = seq(from = range(redata23$temp\_s)[1],  
 to = range(redata23$temp\_s)[2],  
 length.out = 100)),  
 #OK from here to below? Not sure...  
 temp\_diff\_s = 0, # scaled so mean=0  
 surv = 1,  
 fl\_01 = 1,  
 nfl = 1,  
 fr\_01 = 1,  
 nfr = 1,  
 nseed = 0,  
 root = 1  
)  
  
# reshape data to long format  
aster23\_predict1\_long <- reshape(as.data.frame(aster23\_predict1),   
 varying = list(vars), direction = "long",   
 timevar = "varb", times = as.factor(vars),   
 v.names = "resp")  
  
# add artifice  
aster23\_predict1\_long$seeds <- as.integer(ifelse(  
 aster23\_predict1\_long$varb == "nseed", 1, 0))  
  
aster23\_predict1.df <- aster23\_predict1\_long  
  
# fixed model predictions  
aster23.p1 <- predict(aster23\_noplot\_pois, # Use aster model and not reaster!  
 newdata = aster23\_predict1\_long,  
 varvar = varb,idvar = id, root = root,  
 #info.tol = 1e-11,   
 se.fit = T)  
  
aster23\_predict1.df$fit <- aster23.p1$fit  
aster23\_predict1.df$se <- aster23.p1$se.fit  
  
aster23\_predict1.df.seeds <- aster23\_predict1.df %>%  
 filter(varb == "nseed")  
  
aster23\_predict1.df.seeds<-aster23\_predict1.df.seeds%>%  
 mutate(ymin=fit-se,ymax=fit+se)  
  
original\_mean\_23<-as.numeric(data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv))%>%  
 summarise(mean\_temp=mean(temp)))  
original\_sd\_23<-as.numeric(data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv))%>%  
 summarise(sd\_temp=sd(temp)))  
  
aster23\_predict1.df.seeds<-aster23\_predict1.df.seeds%>%  
 mutate(temp=temp\_s\*original\_sd\_23+original\_mean\_23)

#### seeds:temp\_diff\_s

#quantile(redata23$temp\_diff\_s,probs = c(0.05,0.95))  
# Prediction for the range with the highest 5% removed  
# NOT USED SO FAR  
  
# make prediction df  
aster23\_predict2 <- data.frame(  
 expand\_grid(temp\_diff\_s = seq(from = range(redata23$temp\_diff\_s)[1],  
 to = range(redata23$temp\_diff\_s)[2],  
 length.out = 100)),  
 #OK from here to below? Not sure...  
 temp\_s = 0, # scaled so mean=0  
 surv = 1,  
 fl\_01 = 1,  
 nfl = 1,  
 fr\_01 = 1,  
 nfr = 1,  
 nseed = 0,  
 root = 1  
)  
  
# reshape data to long format  
aster23\_predict2\_long <- reshape(as.data.frame(aster23\_predict2),   
 varying = list(vars), direction = "long",   
 timevar = "varb", times = as.factor(vars),   
 v.names = "resp")  
  
# add artifice  
aster23\_predict2\_long$seeds <- as.integer(ifelse(  
 aster23\_predict2\_long$varb == "nseed", 1, 0))  
  
aster23\_predict2.df <- aster23\_predict2\_long  
  
# fixed model predictions  
aster23.p2 <- predict(aster23\_noplot\_pois, # Use aster model and not reaster!  
 newdata = aster23\_predict2\_long,  
 varvar = varb,idvar = id, root = root,  
 #info.tol = 1e-11,   
 se.fit = T)  
  
aster23\_predict2.df$fit <- aster23.p2$fit  
aster23\_predict2.df$se <- aster23.p2$se.fit  
  
aster23\_predict2.df.seeds <- aster23\_predict2.df %>%  
 filter(varb == "nseed")  
  
aster23\_predict2.df.seeds<-aster23\_predict2.df.seeds%>%  
 mutate(ymin=fit-se,ymax=fit+se)  
  
original\_mean\_temp\_diff\_23<-as.numeric(data\_transplants\_aster%>%  
 filter(year==2023)%>%  
 summarise(mean\_temp\_diff=mean(temp\_diff)))  
original\_sd\_temp\_diff\_23<-as.numeric(data\_transplants\_aster%>%  
 filter(year==2023)%>%  
 summarise(sd\_temp\_diff=sd(temp\_diff)))  
  
aster23\_predict2.df.seeds<-aster23\_predict2.df.seeds%>%  
 mutate(temp\_diff=temp\_diff\_s\*original\_sd\_temp\_diff\_23+  
 original\_mean\_temp\_diff\_23)

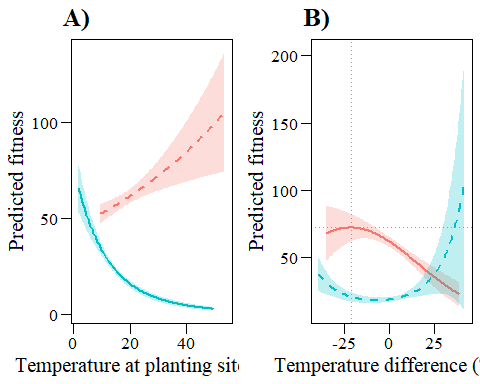
## Combined aster plots

### All effects

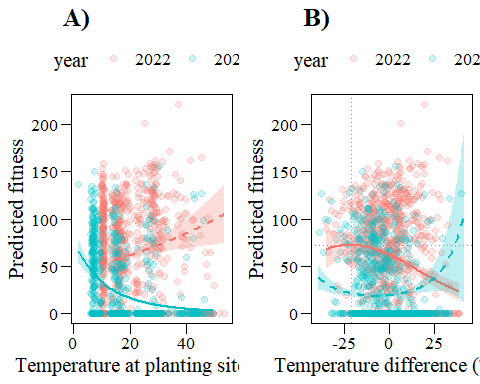
plot\_temp\_nopoints<-ggplot()+  
 geom\_ribbon(data=aster22\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25,fill="#F8766D")+  
 geom\_line(data=aster22\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb),  
 size=1,linetype="dashed",color="#F8766D")+  
 geom\_ribbon(data=aster23\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25,fill="#00BFC4")+  
 geom\_line(data=aster23\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb),  
 size=1,color="#00BFC4")+  
 xlab("Temperature at planting site (ºC)")+ylab("Predicted fitness")+  
 my\_theme\_legend()+theme(legend.position="top")  
plot\_temp\_points<-ggplot()+  
 geom\_point(data=data\_transplants\_aster%>%filter(!is.na(surv)),  
 aes(x=temp,y=nseed,color=year),size=2,alpha=0.2)+  
 geom\_ribbon(data=aster22\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25,fill="#F8766D")+  
 geom\_line(data=aster22\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb),  
 size=1,linetype="dashed",color="#F8766D")+  
 geom\_ribbon(data=aster23\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25,fill="#00BFC4")+  
 geom\_line(data=aster23\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb),  
 size=1,color="#00BFC4")+  
 xlab("Temperature at planting site (ºC)")+ylab("Predicted fitness")+  
 my\_theme\_legend()+theme(legend.position="top")

plot\_temp\_diff\_nopoints<-ggplot()+  
 geom\_ribbon(data=aster22\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25,fill="#F8766D")+  
 geom\_line(data=aster22\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb),  
 size=1,color="#F8766D")+  
 geom\_ribbon(data=aster23\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25,fill="#00BFC4")+  
 geom\_line(data=aster23\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb),  
 size=1,linetype="dashed",color="#00BFC4")+  
 geom\_hline(yintercept=72.23646,linetype="dotted",color="#F8766D")+  
 geom\_vline(xintercept=-21.39296,linetype="dotted",color="#F8766D")+  
 xlab("Temperature difference (ºC)")+ylab("Predicted fitness")+  
 my\_theme\_legend()+theme(legend.position="top")  
plot\_temp\_diff\_points<-ggplot()+  
 geom\_point(data=data\_transplants\_aster%>%filter(!is.na(surv)),  
 aes(x=temp\_diff,y=nseed,color=year),size=2,alpha=0.2)+  
 geom\_ribbon(data=aster22\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25,fill="#F8766D")+  
 geom\_line(data=aster22\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb),  
 size=1,color="#F8766D")+  
 geom\_ribbon(data=aster23\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25,fill="#00BFC4")+  
 geom\_line(data=aster23\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb),  
 size=1,linetype="dashed",color="#00BFC4")+  
 geom\_hline(yintercept=72.23646,linetype="dotted",color="#F8766D")+  
 geom\_vline(xintercept=-21.39296,linetype="dotted",color="#F8766D")+  
 xlab("Temperature difference (ºC)")+ylab("Predicted fitness")+  
 my\_theme\_legend()+theme(legend.position="top")

aster\_plot\_nopoints<-ggarrange(plot\_temp\_nopoints+ggtitle("A)"),  
 plot\_temp\_diff\_nopoints+ggtitle("B)"))  
aster\_plot\_points<-ggarrange(plot\_temp\_points+ggtitle("A)"),  
 plot\_temp\_diff\_points+ggtitle("B)"))  
aster\_plot\_nopoints



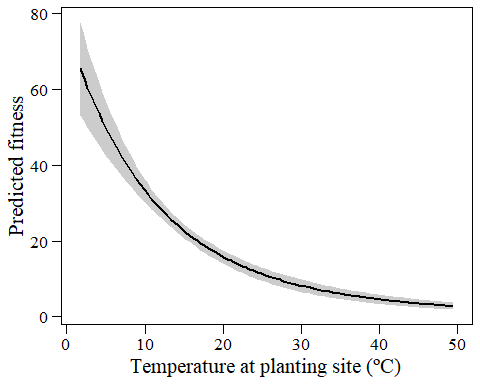
aster\_plot\_points



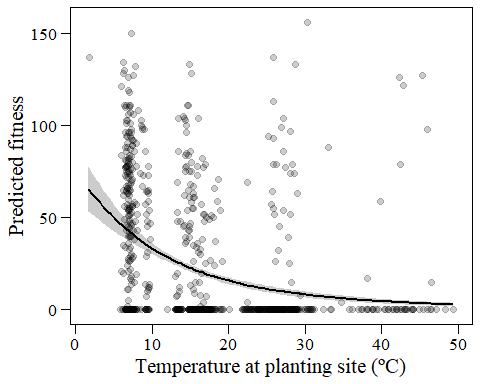
ggsave(filename="output/figures/aster\_plot\_nopoints.tiff",  
 plot=aster\_plot\_nopoints,device="tiff",width=20,height=10,  
 units="cm",dpi=300)  
ggsave(filename="output/figures/aster\_plot\_points.tiff",  
 plot=aster\_plot\_points,device="tiff",width=20,height=10,  
 units="cm",dpi=300)

### (USE) significant effects

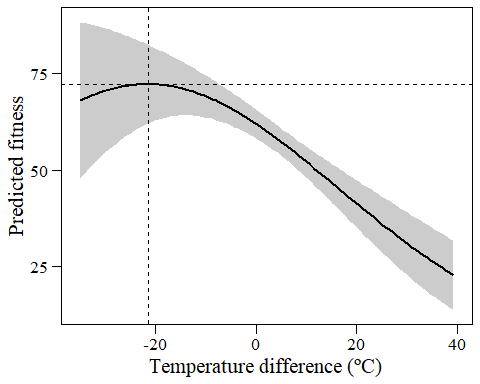
plot\_temp\_23\_nopoints<- ggplot()+  
 geom\_ribbon(data=aster23\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25)+  
 geom\_line(data=aster23\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb),  
 size=1)+  
 xlab("Temperature at planting site (ºC)")+ylab("Predicted fitness")+  
 my\_theme()  
plot\_temp\_23\_points<-ggplot()+  
 geom\_point(data=data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv)),  
 aes(x=temp,y=nseed),size=2,alpha=0.2)+  
 geom\_ribbon(data=aster23\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25)+  
 geom\_line(data=aster23\_predict1.df.seeds,  
 aes(x=temp,y=fit,group=varb),  
 size=1)+  
 xlab("Temperature at planting site (ºC)")+ylab("Predicted fitness")+  
 my\_theme()  
plot\_temp\_23\_nopoints



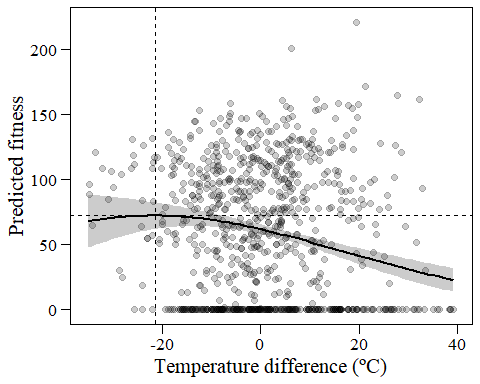
plot\_temp\_23\_points



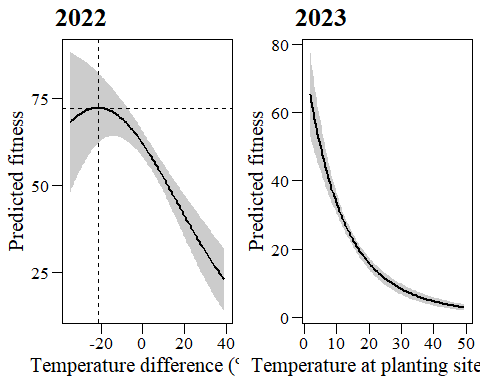
plot\_temp\_diff\_22\_nopoints<-ggplot()+  
 geom\_ribbon(data=aster22\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25)+  
 geom\_line(data=aster22\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb),  
 size=1)+  
 geom\_hline(yintercept=72.23646,linetype="dashed")+  
 geom\_vline(xintercept=-21.39296,linetype="dashed")+  
 xlab("Temperature difference (ºC)")+ylab("Predicted fitness")+  
 my\_theme()  
plot\_temp\_diff\_22\_points<-ggplot()+  
 geom\_point(data=data\_transplants\_aster%>%  
 filter(year==2022),  
 aes(x=temp\_diff,y=nseed),size=2,alpha=0.2)+  
 geom\_ribbon(data=aster22\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb,ymin=ymin,ymax=ymax),  
 alpha=0.25)+  
 geom\_line(data=aster22\_predict2.df.seeds,  
 aes(x=temp\_diff,y=fit,group=varb),  
 size=1)+  
 geom\_hline(yintercept=72.23646,linetype="dashed")+  
 geom\_vline(xintercept=-21.39296,linetype="dashed")+  
 xlab("Temperature difference (ºC)")+ylab("Predicted fitness")+  
 my\_theme()  
plot\_temp\_diff\_22\_nopoints



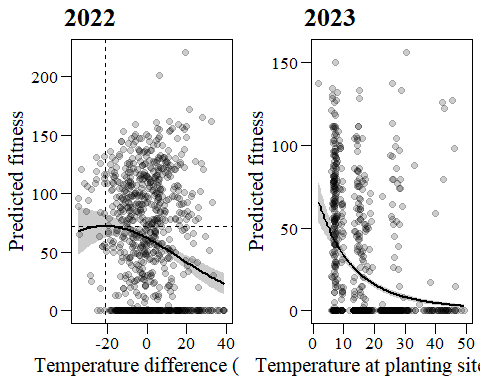
plot\_temp\_diff\_22\_points



aster\_plot\_nopoints\_sig<-ggarrange(plot\_temp\_diff\_22\_nopoints+ggtitle("2022"),  
 plot\_temp\_23\_nopoints+ggtitle("2023"))  
aster\_plot\_points\_sig<-ggarrange(plot\_temp\_diff\_22\_points+ggtitle("2022"),  
 plot\_temp\_23\_points+ggtitle("2023"))  
aster\_plot\_nopoints\_sig



aster\_plot\_points\_sig



ggsave(filename="output/figures/aster\_plot\_nopoints\_sig.tiff",  
 plot=aster\_plot\_nopoints\_sig,device="tiff",width=20,height=10,  
 units="cm",dpi=300)  
ggsave(filename="output/figures/aster\_plot\_points\_sig.tiff",  
 plot=aster\_plot\_points\_sig,device="tiff",width=20,height=10,  
 units="cm",dpi=300)

# HERE

Two cases with no fruits but seeds (n\_tot\_seed >0)! I should probably revise all the data row by row and rerun all analyses.

data\_transplants\_aster%>%  
 filter(year==2023&!is.na(surv))%>%  
 filter(n\_tot\_seed>0&nseed==0)

## # A tibble: 2 × 73  
## unique\_id heat\_zone group mother father id repl plot x y peak   
## <fct> <fct> <fct> <fct> <fct> <fct> <fct> <fct> <dbl> <dbl> <fct>  
## 1 Hot\_119\_b Hot 3 38 360 119 b H05 8 3.4 <NA>   
## 2 Cold\_103\_a Cold 3 33 350 103 a HC1 5 6.8 1   
## fruit\_coll comments1 comments2 poll\_type date\_planted temp  
## <fct> <chr> <chr> <fct> <dttm> <dbl>  
## 1 0 <NA> <NA> Crosspollinated 2021-09-15 00:00:00 30.7  
## 2 0 <NA> <NA> Crosspollinated 2021-09-30 00:00:00 7.1  
## FFD\_corr LFD\_corr MeanFD temp\_mother temp\_father crossing F\_NF\_A n\_fl\_stems  
## <dbl> <dbl> <dbl> <dbl> <dbl> <fct> <fct> <int>  
## 1 NA NA NA 39 22.4 38\_360 <NA> NA  
## 2 181. NA NA 16.3 20.6 33\_350 F 1  
## n\_opened\_fl n\_closed\_buds tot\_fl\_bodies median\_h per\_invert\_herb  
## <int> <int> <int> <int> <int>  
## 1 NA NA 0 NA NA  
## 2 3 2 5 15 1  
## n\_stems\_grazed comments\_peak date\_first\_ripe\_fruit n\_closed\_coll\_fr  
## <int> <chr> <dttm> <int>  
## 1 NA <NA> NA NA  
## 2 1 <NA> NA NA  
## n\_open\_coll\_fr date\_count\_fr n\_count\_fr n\_tot\_fr n\_seed\_fr1 n\_seed\_fr2  
## <int> <dttm> <int> <int> <int> <int>  
## 1 NA NA NA 0 NA NA  
## 2 NA NA NA 0 29 16  
## n\_seed\_fr3 n\_seed\_fr4 rest\_combined n\_tot\_seed mean\_seed\_per\_fr comments\_seeds  
## <int> <int> <int> <int> <dbl> <lgl>   
## 1 NA NA 42 42 NA NA   
## 2 39 NA NA 84 84 NA   
## date\_peak fitness\_rel FFD\_std LFD\_std MeanFD\_std median\_h\_std  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 <NA> 1.68 NA NA NA NA   
## 2 4626386-12-06T00:00:00Z 3.37 0.251 NA NA 2.39  
## tot\_fl\_bodies\_corr tot\_fl\_bodies\_corr\_log n\_fl\_std year comments stem\_width  
## <int> <dbl> <dbl> <fct> <chr> <dbl>  
## 1 0 0 0 2023 <NA> NA   
## 2 5 1.61 0.449 2023 <NA> 1.17  
## mean\_temp\_parents temp\_diff fitness\_01 temp\_diff\_square surv flowering  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 30.7 3.55e-15 1 1.26e-29 0 NA  
## 2 18.5 -1.14e+ 1 1 1.29e+ 2 1 1  
## diff\_fr\_fl fruiting fr\_set n\_seeds\_per\_fl nfl nfr nseed fl\_01 fr\_01  
## <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0 NA NA NA 0 0 0 0 0  
## 2 -5 0 NA NA 5 0 0 1 0

# Session info

sessionInfo()

## R version 4.3.2 (2023-10-31 ucrt)  
## Platform: x86\_64-w64-mingw32/x64 (64-bit)  
## Running under: Windows 11 x64 (build 22631)  
##   
## Matrix products: default  
##   
##   
## locale:  
## [1] LC\_COLLATE=English\_United States.utf8   
## [2] LC\_CTYPE=English\_United States.utf8   
## [3] LC\_MONETARY=English\_United States.utf8  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United States.utf8   
##   
## time zone: Europe/Madrid  
## tzcode source: internal  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] piecewiseSEM\_2.3.0 ggpubr\_0.6.0 aster\_1.1-3 viridis\_0.6.4   
## [5] viridisLite\_0.4.2 gridExtra\_2.3 MASS\_7.3-60 DHARMa\_0.4.6   
## [9] performance\_0.10.8 ggeffects\_1.5.1 sjPlot\_2.8.15 car\_3.1-2   
## [13] carData\_3.0-5 lmerTest\_3.1-3 lme4\_1.1-35.1 Matrix\_1.6-4   
## [17] glmmTMB\_1.1.8 ggthemes\_5.1.0 knitr\_1.45 RColorBrewer\_1.1-3  
## [21] readxl\_1.4.3 lubridate\_1.9.3 forcats\_1.0.0 stringr\_1.5.1   
## [25] dplyr\_1.1.4 purrr\_1.0.2 readr\_2.1.4 tidyr\_1.3.0   
## [29] tibble\_3.2.1 ggplot2\_3.4.4 tidyverse\_2.0.0   
##   
## loaded via a namespace (and not attached):  
## [1] rstudioapi\_0.15.0 jsonlite\_1.8.8 datawizard\_0.9.1   
## [4] magrittr\_2.0.3 TH.data\_1.1-2 estimability\_1.4.1   
## [7] farver\_2.1.1 nloptr\_2.0.3 rmarkdown\_2.25   
## [10] ragg\_1.2.7 vctrs\_0.6.5 minqa\_1.2.6   
## [13] effectsize\_0.8.6 rstatix\_0.7.2 htmltools\_0.5.7   
## [16] haven\_2.5.4 broom\_1.0.5 cellranger\_1.1.0   
## [19] sjmisc\_2.8.9 htmlwidgets\_1.6.4 plyr\_1.8.9   
## [22] sandwich\_3.1-0 emmeans\_1.9.0 zoo\_1.8-12   
## [25] TMB\_1.9.10 mime\_0.12 lifecycle\_1.0.4   
## [28] iterators\_1.0.14 pkgconfig\_2.0.3 gap\_1.5-3   
## [31] sjlabelled\_1.2.0 R6\_2.5.1 fastmap\_1.1.1   
## [34] rbibutils\_2.2.16 shiny\_1.8.0 snakecase\_0.11.1   
## [37] digest\_0.6.33 numDeriv\_2016.8-1.1 colorspace\_2.1-0   
## [40] textshaping\_0.3.7 qgam\_1.3.4 labeling\_0.4.3   
## [43] fansi\_1.0.6 timechange\_0.2.0 abind\_1.4-5   
## [46] mgcv\_1.9-0 compiler\_4.3.2 bit64\_4.0.5   
## [49] withr\_2.5.2 doParallel\_1.0.17 backports\_1.4.1   
## [52] highr\_0.10 ggsignif\_0.6.4 sjstats\_0.18.2   
## [55] tools\_4.3.2 trust\_0.1-8 httpuv\_1.6.13   
## [58] glue\_1.6.2 DiagrammeR\_1.0.10 promises\_1.2.1   
## [61] nlme\_3.1-164 grid\_4.3.2 generics\_0.1.3   
## [64] gtable\_0.3.4 tzdb\_0.4.0 hms\_1.1.3   
## [67] utf8\_1.2.4 foreach\_1.5.2 pillar\_1.9.0   
## [70] vroom\_1.6.5 later\_1.3.2 splines\_4.3.2   
## [73] lattice\_0.22-5 survival\_3.5-7 bit\_4.0.5   
## [76] tidyselect\_1.2.0 stats4\_4.3.2 xfun\_0.41   
## [79] MuMIn\_1.47.5 visNetwork\_2.1.2 stringi\_1.8.3   
## [82] yaml\_2.3.8 boot\_1.3-28.1 evaluate\_0.23   
## [85] codetools\_0.2-19 cli\_3.6.2 systemfonts\_1.0.5   
## [88] parameters\_0.21.3 xtable\_1.8-4 Rdpack\_2.6   
## [91] munsell\_0.5.0 modelr\_0.1.11 Rcpp\_1.0.11   
## [94] coda\_0.19-4 parallel\_4.3.2 ellipsis\_0.3.2   
## [97] bayestestR\_0.13.1 gap.datasets\_0.0.6 mvtnorm\_1.2-4   
## [100] scales\_1.3.0 insight\_0.19.10 crayon\_1.5.2   
## [103] rlang\_1.1.2 cowplot\_1.1.2 multcomp\_1.4-25