Maladaptive plastic responses of flowering time to geothermal heating

Code for analyses in the paper

Alicia Valdés

Contents

Read data

The location of these files would need to be changed.

```
data_plants<-read_csv("data/clean/data_plants.csv")
logger_data<-read_csv("data/clean/logger_data.csv")
logger_data_pairs<-read_csv("data/clean/logger_data_pairs.csv")
```

Correlation between instant measures of soil temperature and mean soil temperature during the period April 1st – June 5th recorded by loggers

For each logger_nr, get mean temperature during April-June and compare with temp_term (which was measured with a thermometer at 10 cm depth on May 2017):

```
with(logger_data%>%
       mutate(month = month(datetime)) %>%
       filter(month==4|month==5|month==6)%>%
       filter(above_below=="B")%>%
       mutate(date=date(datetime))%>%
       filter(!is.na(date))%>% # remove records with no info on date
       filter(datetime<"2018-06-06")%>% # keep only data until June 5
       group_by(logger_nr) %>%
       summarize(mean_logger=mean(temp),temp_term=mean(temp_term)),
     cor.test(mean_logger,temp_term))
```

```
##
##  Pearson's product-moment correlation
##
## data:  mean_logger and temp_term
## t = 21.901, df = 139, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8370572 0.9129479
## sample estimates:
##       cor
## 0.8805259
```

Is soil temperature more weakly correlated with air temperature in warmer soils?

For each date and logger pair, calculate mean, max and min of air and soil temperature (from, respectively, the above and belowground logger). Then, calculate the correlation coefficient for air and soil temperatures over the period May or April-May-June. Finally, regress these correlation coefficients on mean soil temperature (from the belowground logger) for the same period (April-May-June).

May

```
data_corr<-(logger_data_pairs%>%
     mutate(month = month(datetime),date=date(datetime))%>%
              # new variables "month" and "date"
     filter(month==5)%>% # keep data from may
     group_by(date,pair,above_below)%>%
     summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
   #calculate mean, max and min of air and soil temperature
     pivot_wider(names_from="above_below",values_from=c("mean","max","min"))%>%
     group_by(pair)%>%
     summarise(corr_airsoil_mean=cor(mean_A,mean_B,use="pairwise.complete.obs"),
               corr_airsoil_max=cor(max_A,max_B,use="pairwise.complete.obs"),
               corr_airsoil_min=cor(min_A,min_B,use="pairwise.complete.obs"))%>%
   # Calculate correlations air-soil temperatures
     pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                  names_to="measure",values_to="corr")%>%
  left_join(logger_data_pairs%>%
                 mutate(month = month(datetime))%>%
                 filter(month==5)%>%
                 filter(above_below=="B")%>%
                 group_by(pair)%>%
              summarise(meansoiltemp=mean(temp))))
  # calculate mean soil temperature for may
```

```
model_mean<-lm(corr~meansoiltemp,
               data = subset(data_corr,measure=="corr_airsoil_mean"))
model_max<-lm(corr~meansoiltemp,
               data = subset(data_corr,measure=="corr_airsoil_max"))
model_min<-lm(corr~meansoiltemp,
               data = subset(data_corr,measure=="corr_airsoil_min"))
```

Predictions of correlations for minimum and maximum temperatures:

```
ggpredict(model_mean,terms="meansoiltemp[minmax]")
```

```
## # Predicted values of corr
##
## meansoiltemp | Predicted |       95% CI
## ---------------------------------------
##         6.14 |      0.83 | [0.80, 0.86]
##        30.55 |      0.61 | [0.55, 0.68]
```

```
ggpredict(model_max,terms="meansoiltemp[minmax]")
```

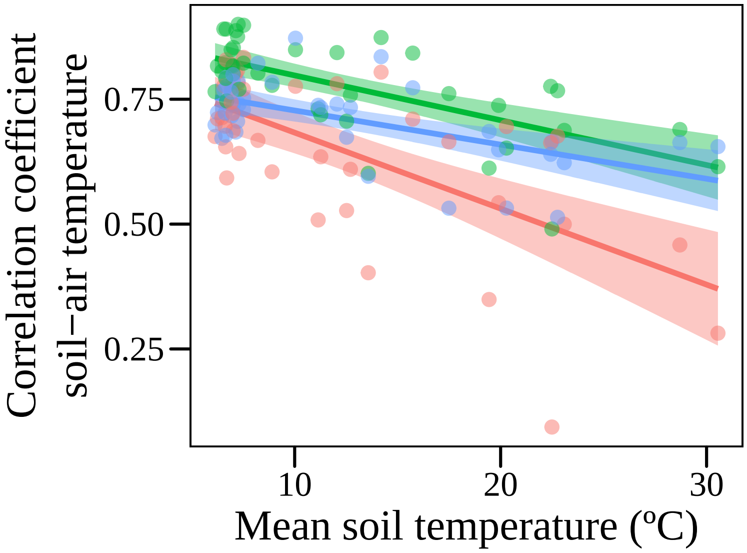
```
## # Predicted values of corr
##
## meansoiltemp | Predicted |       95% CI
## ---------------------------------------
##         6.14 |      0.74 | [0.69, 0.79]
##        30.55 |      0.37 | [0.26, 0.48]
```

```
ggpredict(model_min,terms="meansoiltemp[minmax]")
```

```
## # Predicted values of corr
##
## meansoiltemp | Predicted |       95% CI
## ---------------------------------------
##         6.14 |      0.75 | [0.73, 0.78]
##        30.55 |      0.59 | [0.53, 0.65]
```

Figure 2: Correlations soil-air temperature vs soil temperature

```
fig2<-(logger_data_pairs%>%
          mutate(month = month(datetime),date=date(datetime))%>%
         # new variables "month" and "date"
          filter(month==5)%>% # keep data from may
          group_by(date,pair,above_below)%>%
          summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
          #calculate mean, max and min of air and soil temperature
          pivot_wider(names_from="above_below",
                      values_from=c("mean","max","min"))%>%
          group_by(pair)%>%
          summarise(corr_airsoil_mean=cor(mean_A,mean_B,
                                          use="pairwise.complete.obs"),
                    corr_airsoil_max=cor(max_A,max_B,
                                         use="pairwise.complete.obs"),
                    corr_airsoil_min=cor(min_A,min_B,
                                         use="pairwise.complete.obs"))%>%
          # Calculate correlations air-soil temperatures
          pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                       names_to="measure",values_to="corr")%>%
          left_join(logger_data_pairs%>%
                      mutate(month = month(datetime))%>%
                      filter(month==5)%>%
                      filter(above_below=="B")%>%
                      group_by(pair)%>%
                      summarise(meansoiltemp=mean(temp))))%>%
  # calculate mean soil temperature for may
  ggplot(.,aes(x=meansoiltemp,y=corr,color=measure,fill=measure))+
  geom_smooth(method="lm",size=1)+geom_point(size=2,alpha=0.5)+
  xlab("Mean soil temperature (ºC)")+
  ylab("Correlation coefficient\nsoil-air temperature")+
  my_theme()+
  geom_text_repel(data=. %>% filter(corr<0),aes(label=pair))
fig2
```



```
ggsave(filename="output/figures/fig2.tiff",plot=fig2,
       width=12,height=10,units="cm",dpi=300)
```

Appendix S3 (part 1)

Linear models testing the effect of soil temperature on correlations between soil and air temperature (part of table in Appendix S3):

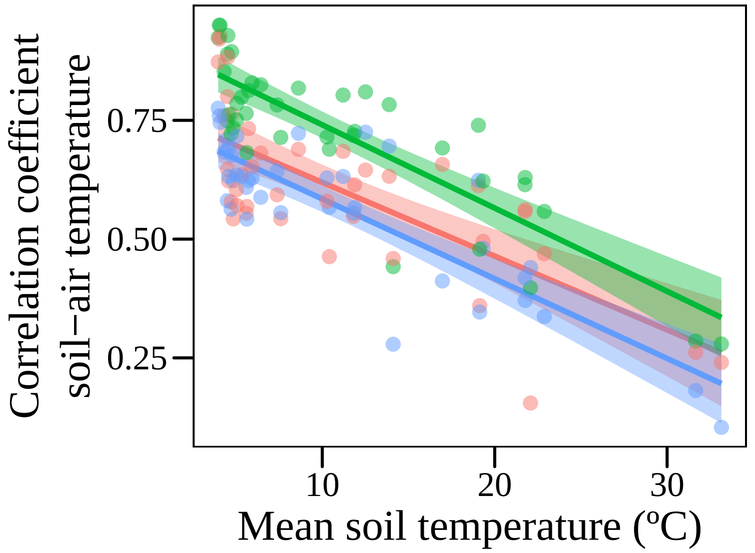
```
(logger_data_pairs%>%
     mutate(month = month(datetime),date=date(datetime))%>%
         # new variables "month" and "date"
     filter(month==5)%>% # keep data from may
     group_by(date,pair,above_below)%>%
     summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
   #calculate mean, max and min of air and soil temperature
     pivot_wider(names_from="above_below",values_from=c("mean","max","min"))%>%
     group_by(pair)%>%
     summarise(corr_airsoil_mean=cor(mean_A,mean_B,use="pairwise.complete.obs"),
               corr_airsoil_max=cor(max_A,max_B,use="pairwise.complete.obs"),
               corr_airsoil_min=cor(min_A,min_B,use="pairwise.complete.obs"))%>%
   # Calculate correlations air-soil temperatures
     pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                  names_to="measure",values_to="corr")%>%
  left_join(logger_data_pairs%>%
                 mutate(month = month(datetime))%>%
                 filter(month==5)%>%
                 filter(above_below=="B")%>%
                 group_by(pair)%>%
              summarise(meansoiltemp=mean(temp))))%>%
  # calculate mean soil temperature for may
  group_by(measure)%>%
  do(fitcorr = tidy(lm(corr~meansoiltemp, data = .))) %>%
  unnest(fitcorr)%>%
  kable(digits=5)
```

measure	term	estimate	std.error	statistic	p.value
corr_airsoil_max	(Intercept)	0.83528	0.04012	20.81870	0e+00
corr_airsoil_max	meansoiltemp	-0.01521	0.00287	-5.29894	1e-05
corr_airsoil_mean	(Intercept)	0.88734	0.02275	39.00495	0e+00
corr_airsoil_mean	meansoiltemp	-0.00897	0.00163	-5.50910	0e+00
corr_airsoil_min	(Intercept)	0.79583	0.02147	37.06454	0e+00
corr_airsoil_min	meansoiltemp	-0.00684	0.00154	-4.44851	7e-05

April-May_june

Appendix S2: Correlations soil-air temperature vs soil temperature (April-June)

```
AppS2<-(logger_data_pairs%>%
          mutate(month = month(datetime),date=date(datetime))%>%
          # new variables "month" and "date"
          filter(month==4|month==5|month==6)%>% # keep data from april-june
          group_by(date,pair,above_below)%>%
          summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
          #calculate mean, max and min of air and soil temperature
          pivot_wider(names_from="above_below",
                       values_from=c("mean","max","min"))%>%
          group_by(pair)%>%
          summarise(corr_airsoil_mean=cor(mean_A,mean_B,
                                           use="pairwise.complete.obs"),
                     corr_airsoil_max=cor(max_A,max_B,
                                          use="pairwise.complete.obs"),
                     corr_airsoil_min=cor(min_A,min_B,
                                          use="pairwise.complete.obs"))%>%
          # Calculate correlations air-soil temperatures
          pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                        names_to="measure",values_to="corr")%>%
          left_join(logger_data_pairs%>%
                       mutate(month = month(datetime))%>%
                       filter(month==4|month==5|month==6)%>%
                       filter(above_below=="B")%>%
                       group_by(pair)%>%
                       summarise(meansoiltemp=mean(temp))))%>%
  # calculate mean soil temperature for april-june
  ggplot(.,aes(x=meansoiltemp,y=corr,color=measure,fill=measure))+
  geom_smooth(method="lm",size=1)+geom_point(size=2,alpha=0.5)+
  xlab("Mean soil temperature (ºC)")+
  ylab("Correlation coefficient\nsoil-air temperature")+
  my_theme()+
  geom_text_repel(data=. %>% filter(corr<0),aes(label=pair))
AppS2
```



```
ggsave(filename="output/figures/AppS2.tiff",plot=AppS2,
       width=12,height=10,units="cm",dpi=300)
```

Appendix S3 (part 2)

Linear models testing the effect of soil temperature on correlations between soil and air temperature (part of table in Appendix S3):

```
(logger_data_pairs%>%
     mutate(month = month(datetime),date=date(datetime))%>%
          # new variables "month" and "date"
     filter(month==4|month==5|month==6)%>% # keep data from april-june
     group_by(date,pair,above_below)%>%
     summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
   #calculate mean, max and min of air and soil temperature
     pivot_wider(names_from="above_below",values_from=c("mean","max","min"))%>%
     group_by(pair)%>%
     summarise(corr_airsoil_mean=cor(mean_A,mean_B,use="pairwise.complete.obs"),
               corr_airsoil_max=cor(max_A,max_B,use="pairwise.complete.obs"),
               corr_airsoil_min=cor(min_A,min_B,use="pairwise.complete.obs"))%>%
   # Calculate correlations air-soil temperatures
     pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                  names_to="measure",values_to="corr")%>%
  left_join(logger_data_pairs%>%
                 mutate(month = month(datetime))%>%
                 filter(month==4|month==5|month==6)%>%
                 filter(above_below=="B")%>%
                 group_by(pair)%>%
               summarise(meansoiltemp=mean(temp))))%>%
  # calculate mean soil temperature for april-june
  group_by(measure)%>%
  do(fitcorr = tidy(lm(corr~meansoiltemp, data = .))) %>%
  unnest(fitcorr)%>%
  kable(digits=5)
```

measure	term	estimate	std.error	statistic	p.value
corr_airsoil_max	(Intercept)	0.77449	0.03184	24.32387	0
corr_airsoil_max	meansoiltemp	-0.01552	0.00236	-6.57111	0
corr_airsoil_mean	(Intercept)	0.91574	0.02391	38.29900	0
corr_airsoil_mean	meansoiltemp	-0.01752	0.00177	-9.88060	0
corr_airsoil_min	(Intercept)	0.75246	0.02361	31.87482	0
corr_airsoil_min	meansoiltemp	-0.01679	0.00175	-9.59056	0

Hypothesis 1: Effect of temperature on FFD

Models including quadratic effects of ffd.

```
FFD_2017_1<-lm(ffd~temp+I(temp^2),subset(data_plants,year==2017))
summ(FFD_2017_1,vif=T)
```

Observations	245
Dependent variable	ffd
Type	OLS linear regression

F(2,242)	17.016
R^2	0.123
Adj. R^2	0.116

	Est.	S.E.	t val.	p	VIF
(Intercept)	183.736	1.761	104.350	0.000	NA
temp	-0.779	0.218	-3.572	0.000	12.715
I(temp^2)	0.012	0.006	2.135	0.034	12.715

Standard errors: OLS

```
FFD_2018_1<-lm(ffd~temp+I(temp^2),subset(data_plants,year==2018))
summ(FFD_2018_1,vif=T)
```

Observations	104
Dependent variable	ffd
Type	OLS linear regression

F(2,101)	25.765
R^2	0.338
Adj. R^2	0.325

Quadratic term of ffd significant in 2017 but not in 2018. Refit model for 2018 withouth quadratic term of ffd.

```
FFD_2018_2<-lm(ffd~temp,subset(data_plants,year==2018))
summ(FFD_2018_2)
```

	Est.	S.E.	t val.	p	VIF
(Intercept)	201.752	4.633	43.546	0.000	NA
temp	-2.161	0.688	-3.142	0.002	20.214
I(temp^2)	0.033	0.021	1.628	0.107	20.214

Standard errors: OLS

Observations	104
Dependent variable	ffd
Type	OLS linear regression

F(1,102)	48.102
R^2	0.320
Adj. R^2	0.314

	Est.	S.E.	t val.	p
(Intercept)	195.104	2.206	88.424	0.000
temp	-1.070	0.154	-6.936	0.000

Standard errors: OLS

Predictions of ffd for minimum and maximum temperatures:

```
ggpredict(FFD_2017_1,terms="temp[minmax]")
```

```
## # Predicted values of ffd
##
##  temp | Predicted |           95% CI
## ------------------------------------
##  4.10 |    180.75 | [178.70, 182.80]
## 45.50 |    173.63 | [165.24, 182.02]
```

```
# 180.75-173.63=8 days earlier on warmer soils
ggpredict(FFD_2018_2,terms="temp[minmax]")
```

```
## # Predicted values of ffd
##
##  temp | Predicted |           95% CI
## ------------------------------------
##  3.50 |    191.36 | [187.91, 194.81]
## 34.00 |    158.74 | [151.85, 165.63]
```

```
# 191.36-158.74=33 days earlier on warmer soils
```

Hypothesis 2: Effect of temperature on fitness

GLMs with negative binomial distribution

```
fitness_2017_4<-glm.nb(n_seed_round~temp+I(temp^2)+log(nfl),
                       subset(data_plants,year==2017))
summ(fitness_2017_4,vif=T)
```

Observations	245
Dependent variable	n_seed_round
Type	Generalized linear model
Family	Negative Binomial(2.1088)
Link	log

χ^2()	0.723	0.088	3273.867	3291.374
Pseudo-R² (Cragg-Uhler)	0.723	0.088	3273.867	3291.374
Pseudo-R² (McFadden)	0.723	0.088	3273.867	3291.374
AIC	0.723	0.088	3273.867	3291.374
BIC	0.723	0.088	3273.867	3291.374

	Est.	S.E.	z val.	p	VIF
(Intercept)	4.221	0.169	24.960	0.000	NA
temp	-0.052	0.021	-2.515	0.012	13.378
I(temp^2)	0.001	0.001	1.078	0.281	13.002
log(nfl)	0.987	0.042	23.320	0.000	1.116

Standard errors: MLE

```
fitness_2018_4<-glm.nb(n_seed_round~temp+I(temp^2)+log(nfl),
                       subset(data_plants,year==2018))
summ(fitness_2018_4,vif=T)
```

Observations	104
Dependent variable	n_seed_round
Type	Generalized linear model
Family	Negative Binomial(1.9935)
Link	log

χ^2()	0.694	0.094	1202.008	1215.230
Pseudo-R² (Cragg-Uhler)	0.694	0.094	1202.008	1215.230
Pseudo-R² (McFadden)	0.694	0.094	1202.008	1215.230
AIC	0.694	0.094	1202.008	1215.230
BIC	0.694	0.094	1202.008	1215.230

	Est.	S.E.	z val.	p	VIF
(Intercept)	3.085	0.318	9.706	0.000	NA
temp	0.036	0.044	0.825	0.409	20.150
I(temp^2)	-0.002	0.001	-1.846	0.065	20.136
log(nfl)	0.973	0.068	14.304	0.000	1.003

Standard errors: MLE

Quadratic terms of ffd not significant. Refit models withouth quadratic terms of ffd.

```r
fitness_2017_5<-glm.nb(n_seed_round~temp+log(nfl),subset(data_plants,year==2017))
summ(fitness_2017_5,vif=T)
```

Observations	245
Dependent variable	n_seed_round
Type	Generalized linear model
Family	Negative Binomial(2.0993)
Link	log

χ^2()	0.722	0.088	3273.106	3287.112
Pseudo-R^2 (Cragg-Uhler)	0.722	0.088	3273.106	3287.112
Pseudo-R^2 (McFadden)	0.722	0.088	3273.106	3287.112
AIC	0.722	0.088	3273.106	3287.112
BIC	0.722	0.088	3273.106	3287.112

	Est.	S.E.	z val.	p	VIF
(Intercept)	4.079	0.113	36.029	0.000	NA
temp	-0.030	0.006	-5.117	0.000	1.094
log(nfl)	0.982	0.042	23.377	0.000	1.094

Standard errors: MLE

```r
fitness_2018_5<-glm.nb(n_seed_round~temp+log(nfl),subset(data_plants,year==2018))
summ(fitness_2018_5,vif=T)
```

Observations	104
Dependent variable	n_seed_round
Type	Generalized linear model
Family	Negative Binomial(1.9313)
Link	log

χ^2()	0.684	0.091	1203.369	1213.947
Pseudo-R^2 (Cragg-Uhler)	0.684	0.091	1203.369	1213.947
Pseudo-R^2 (McFadden)	0.684	0.091	1203.369	1213.947
AIC	0.684	0.091	1203.369	1213.947
BIC	0.684	0.091	1203.369	1213.947

	Est.	S.E.	z val.	p	VIF
(Intercept)	3.543	0.188	18.878	0.000	NA
temp	-0.042	0.010	-4.166	0.000	1.003
log(nfl)	0.981	0.069	14.202	0.000	1.003

Standard errors: MLE

Predictions of fitness for minimum and maximum temperatures:

```
ggpredict(fitness_2017_5,terms="temp[minmax]")
```

```
## # Predicted counts of n_seed_round
##
##  temp | Predicted |            95% CI
## -------------------------------------
##  4.10 |    849.90 | [719.89, 1003.40]
## 45.50 |    243.83 | [169.94,  349.84]
##
## Adjusted for:
## * nfl = 17.13
```

```
ggpredict(fitness_2018_5,terms="temp[minmax]")
```

```
## # Predicted counts of n_seed_round
##
##  temp | Predicted |           95% CI
## ------------------------------------
##  3.50 |    301.27 | [238.31, 380.87]
## 34.00 |     84.77 | [ 54.15, 132.71]
##
## Adjusted for:
## * nfl = 10.54
```

Figure 3: Effects of temperature on ffd, and fitness in 2018

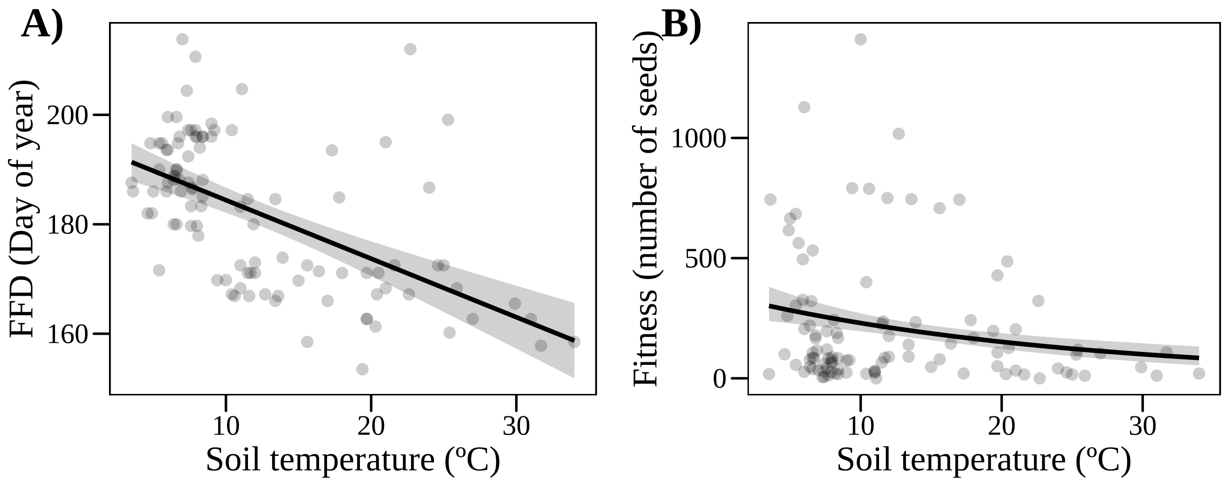
Model prediction ffd : based on model FFD_2018_2 (without quadratic term of ffd)

```
predict_FFD_2018<-ggpredict(FFD_2018_2,terms = "temp [all]")
```

Model prediction fitness : based on model fitness_2018_5.

```
predict_fitness_2018<-ggpredict(fitness_2018_5,terms = "temp [all]")
```

```
fig3<-
  grid.arrange(
    # ffd
    ggplot(subset(data_plants,year==2018),aes(x=temp,y=ffd))+
      xlab("Soil temperature (ºC)")+ylab("FFD (Day of year)")+my_theme()+
      geom_ribbon(data=predict_FFD_2018,
                  aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),
                  fill="grey",alpha=0.7)+
      geom_line(data=predict_FFD_2018,
                aes(x=x,y=predicted),size=1,color="black")+
      geom_point(size=2,alpha=0.2)+
      ggtitle("A)")+theme(plot.title=element_text(hjust=-0.20,vjust=-3))+
      theme(plot.margin = unit(c(-0.6,0.3,0,0.3), "cm")),
    # fitness
    ggplot(subset(data_plants,year==2018),aes(x=temp,y=nseed))+
      xlab("Soil temperature (ºC)")+ylab("Fitness (number of seeds)")+
      my_theme()+
      geom_ribbon(data=predict_fitness_2018,
                  aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),
                  fill="grey",alpha=0.7)+
      geom_line(data=predict_fitness_2018,
                aes(x=x,y=predicted),size=1,color="black")+
      geom_point(size=2,alpha=0.2)+
      ggtitle("B)")+theme(plot.title=element_text(hjust=-0.20,vjust=-3))+
      theme(plot.margin = unit(c(-0.6,0.3,0,0.3), "cm")),
    ncol=2)
```



```
ggsave(filename="output/figures/fig3.tiff",plot=fig3,
       width=20,height=8,units="cm",dpi=300)
```

Figure S1: Effects of temperature on ffd and fitness in 2017

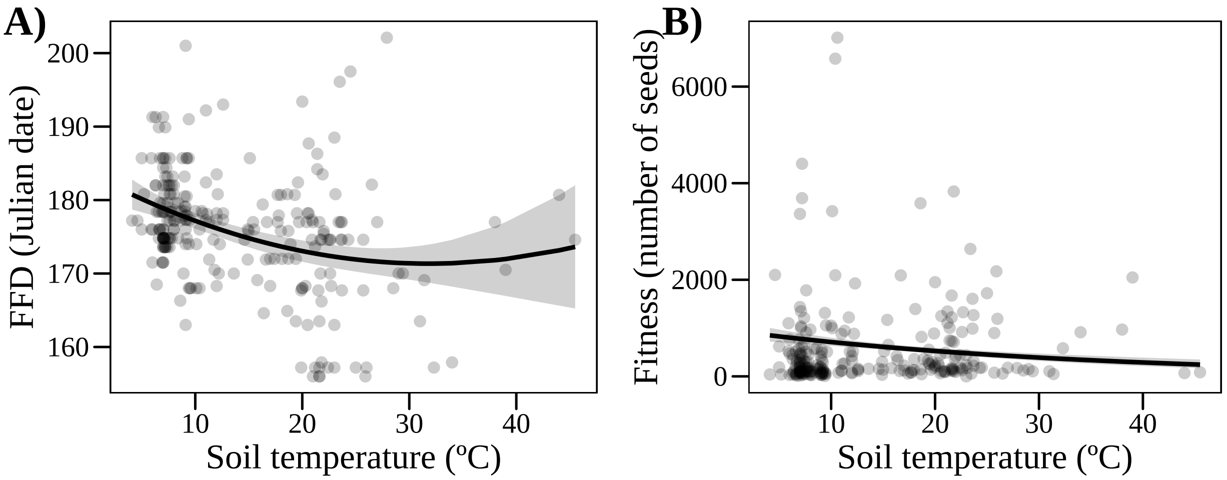
Model prediction ffd : based on model FFD_2017_1 (with quadratic term of ffd)

```
predict_FFD_2017<-ggpredict(FFD_2017_1,terms = "temp [all]")
```

Model prediction fitness : based on model fitness_2017_5

```
predict_fitness_2017<-ggpredict(fitness_2017_5,terms = "temp [all]")
```

```
figS1<-
  grid.arrange(
    # ffd
    ggplot(subset(data_plants,year==2017),aes(x=temp,y=ffd))+
      xlab("Soil temperature (ºC)")+ylab("FFD (Julian date)")+my_theme()+
      geom_ribbon(data=predict_FFD_2017,
                  aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),
                  fill="grey",alpha=0.7)+
      geom_line(data=predict_FFD_2017,
                aes(x=x,y=predicted),size=1,color="black")+
      geom_point(size=2,alpha=0.2)+
      ggtitle("A)")+theme(plot.title=element_text(hjust=-0.24,vjust=-3))+
      theme(plot.margin = unit(c(-0.6,0.3,0,0.3), "cm")),
    # fitness
    ggplot(subset(data_plants,year==2017),aes(x=temp,y=nseed))+
      xlab("Soil temperature (ºC)")+ylab("Fitness (number of seeds)")+
      my_theme()+
      geom_ribbon(data=predict_fitness_2017,
                  aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),
                  fill="grey",alpha=0.7)+
      geom_line(data=predict_fitness_2017,
                aes(x=x,y=predicted),size=1,color="black")+
      geom_point(size=2,alpha=0.2)+
      ggtitle("B)")+theme(plot.title=element_text(hjust=-0.20,vjust=-3))+
      theme(plot.margin = unit(c(-0.6,0.3,0,0.3), "cm")),
    ncol=2)
```



```
ggsave(filename="output/figures/figS1.tiff",plot=figS1,
       width=20,height=8,units="cm",dpi=300)
```

Hypothesis 3: Effect of temperature on selection on FFD

```
selection_2017_1<-lm(nseed_rel~ffd_std*temp+ffd_std*I(temp^2)+nfl_std,
                     subset(data_plants,year==2017))
selection_2018_1<-lm(nseed_rel~ffd_std*temp+ffd_std*I(temp^2)+nfl_std,
                     subset(data_plants,year==2018))
summ(selection_2017_1)
```

Observations	245
Dependent variable	nseed_rel
Type	OLS linear regression

F(6,238)	34.508
R²	0.465
Adj. R²	0.452

	Est.	S.E.	t val.	p
(Intercept)	1.642	0.299	5.490	0.000
ffd_std	0.504	0.389	1.296	0.196
temp	-0.065	0.036	-1.808	0.072
I(temp^2)	0.001	0.001	1.033	0.303
nfl_std	1.226	0.090	13.667	0.000
ffd_std:temp	-0.035	0.045	-0.770	0.442
ffd_std:I(temp^2)	0.001	0.001	0.627	0.531

Standard errors: OLS

```
summ(selection_2018_1)
```

Observations	104
Dependent variable	nseed_rel
Type	OLS linear regression

F(6,97)	22.809
R²	0.585
Adj. R²	0.560

	Est.	S.E.	t val.	p
(Intercept)	1.607	0.517	3.105	0.002
ffd_std	-0.455	0.499	-0.911	0.365
temp	-0.055	0.083	-0.662	0.510
I(temp^2)	0.001	0.003	0.369	0.713
nfl_std	0.973	0.103	9.437	0.000
ffd_std:temp	0.043	0.071	0.614	0.541
ffd_std:I(temp^2)	-0.000	0.002	-0.146	0.884

Standard errors: OLS

Quadratic terms of ffd not significant. Refit models withouth quadratic terms of ffd.

```
selection_2017_2<-lm(nseed_rel~ffd_std*temp+nfl_std,
                     subset(data_plants,year==2017))
selection_2018_2<-lm(nseed_rel~ffd_std*temp+nfl_std,
                     subset(data_plants,year==2018))
summ(selection_2017_2)
```

Observations	245
Dependent variable	nseed_rel
Type	OLS linear regression

F(4,240)	51.549
R²	0.462
Adj. R²	0.453

	Est.	S.E.	t val.	p
(Intercept)	1.408	0.166	8.460	0.000
ffd_std	0.331	0.206	1.602	0.110
temp	-0.030	0.010	-2.937	0.004
nfl_std	1.213	0.089	13.663	0.000
ffd_std:temp	-0.009	0.010	-0.848	0.397

Standard errors: OLS

```
summ(selection_2018_2)
```

Observations	104
Dependent variable	nseed_rel
Type	OLS linear regression

F(4,99)	34.660
R²	0.583
Adj. R²	0.567

	Est.	S.E.	t val.	p
(Intercept)	1.374	0.201	6.825	0.000
ffd_std	-0.293	0.205	-1.431	0.156
temp	-0.021	0.016	-1.368	0.174
nfl_std	0.976	0.101	9.679	0.000
ffd_std:temp	0.027	0.012	2.241	0.027

Standard errors: OLS

BCa intervals

Used for assessing significance.

2017

```
# ffd
slp <- function(selection_2017_2) coef(selection_2017_2)[2]
b <- car::Boot(selection_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selection_2017_2) coef(selection_2017_2)[3]
b <- car::Boot(selection_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# nfl
slp <- function(selection_2017_2) coef(selection_2017_2)[4]
b <- car::Boot(selection_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# ffd:temp
slp <- function(selection_2017_2) coef(selection_2017_2)[5]
b <- car::Boot(selection_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# Save confidence intervals as a table
BCIs_selection_2017 <- cbind(
  rbind(ffd_ci_17[1,] ,temp_ci_17[1,], nfl_ci_17[1,], ffd_temp_ci_17[1,]),
  rbind(ffd_ci_17[2,] ,temp_ci_17[2,], nfl_ci_17[2,], ffd_temp_ci_17[2,])
)
colnames(BCIs_selection_2017)<-c("lower","upper")
rownames(BCIs_selection_2017) <- c("ffd","temp","nfl","ffd:temp")
save(BCIs_selection_2017,file="output/BCIs_selection_2017.RData")
```

```
BCIs_selection_2017
```

```
##                lower       upper
## ffd       0.02063177  0.78962290
## temp     -0.05525377 -0.01332811
## nfl       0.91215295  1.69581048
## ffd:temp -0.02586447  0.00561899
```

2018

```
# ffd
slp <- function(selection_2018_2) coef(selection_2018_2)[2]
b <- car::Boot(selection_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_18 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selection_2018_2) coef(selection_2018_2)[3]
b <- car::Boot(selection_2018_2,slp, R=10000) # note the capital B
```

```
b1 <- boot::boot.ci(b,type="bca")
temp_ci_18 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# nfl
slp <- function(selection_2018_2) coef(selection_2018_2)[4]
b <- car::Boot(selection_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_18 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# ffd:temp
slp <- function(selection_2018_2) coef(selection_2018_2)[5]
b <- car::Boot(selection_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_18 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# Save confidence intervals as a table
BCIs_selection_2018 <- cbind(
  rbind(ffd_ci_18[1,] ,temp_ci_18[1,], nfl_ci_18[1,], ffd_temp_ci_18[1,]),
  rbind(ffd_ci_18[2,] ,temp_ci_18[2,], nfl_ci_18[2,], ffd_temp_ci_18[2,])
)
colnames(BCIs_selection_2018)<-c("lower","upper")
rownames(BCIs_selection_2018) <- c("ffd","temp","nfl","ffd:temp")
save(BCIs_selection_2018,file="output/BCIs_selection_2018.RData")
```

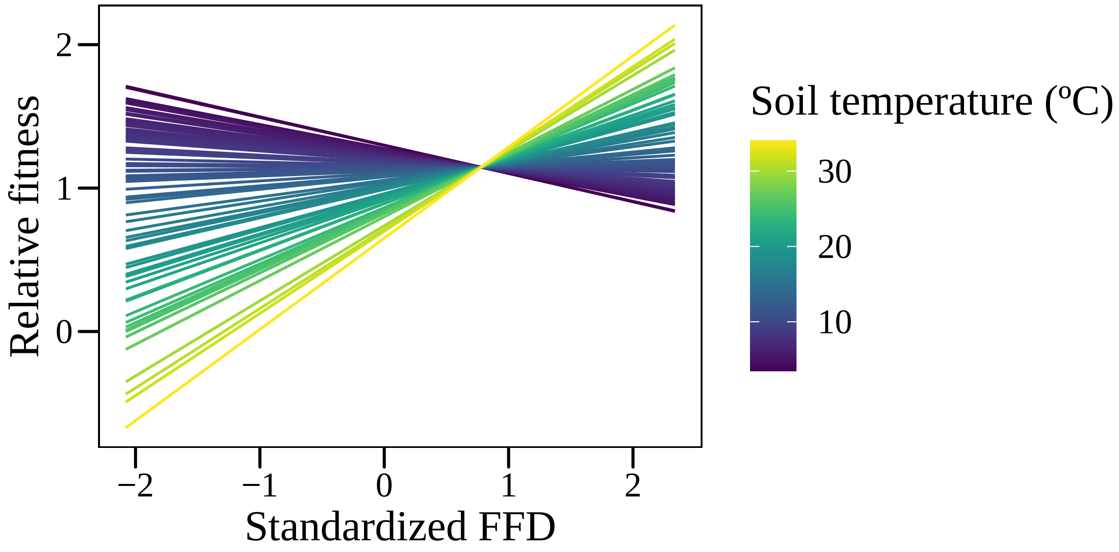
```
BCIs_selection_2018
```

```
##                 lower        upper
## ffd      -0.742240772  0.206171379
## temp     -0.040770890 -0.001494675
## nfl       0.731718203  1.257997557
## ffd:temp  0.003793931  0.052741161
```

Figure 4: Effects of temperature on selection in 2018

```
pred_fitness<-ggpredict(selection_2018_2,
                        terms = c("ffd_std [all]","temp [all]"))
```

```
ggplot(pred_fitness,aes(x,predicted,colour=group,fill=group))+
  geom_line(aes(color=as.numeric(as.character(group))),size=0.5)+my_theme()+
  scale_color_viridis()+
  theme(legend.position="right")+labs(colour="Soil temperature (ºC)")+
  xlab("Standardized FFD")+ylab("Relative fitness")
```

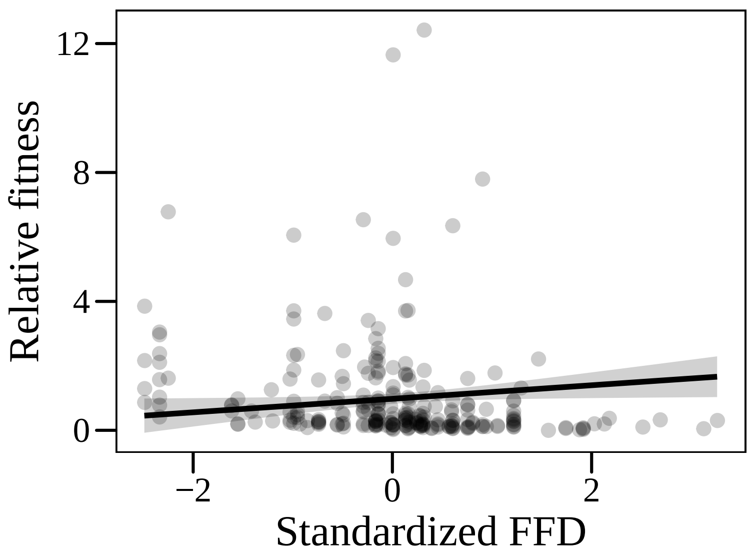


```
ggsave(filename="output/figures/fig4.tiff",width=14,height=8,units="cm",dpi=300)
```

Figure S2: Effects of ffd on relative fitness in 2017

```
pred_fitness_17<-ggpredict(selection_2017_2,
                           terms = c("ffd_std [all]"))
```

```
ggplot(subset(data_plants,year==2017),aes(x=ffd_std,y=nseed_rel))+
  xlab("Standardized FFD")+ylab("Relative fitness")+my_theme()+
  geom_ribbon(data=pred_fitness_17,
              aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),
              fill="grey",alpha=0.7)+
  geom_line(data=pred_fitness_17,aes(x=x,y=predicted),size=1,color="black")+
  geom_point(size=2,alpha=0.2)
```



```
ggsave(filename="output/figures/figS2.tiff",width=10,height=8,
       units="cm",dpi=300)
```

Effect of temperature on the relationship absolute fitness-FFD

```
selectionabs_2017_1<-lm(nseed~ffd*temp+ffd*I(temp^2)+log(nfl),
                        subset(data_plants,year==2017))
selectionabs_2018_1<-lm(nseed~ffd*temp+ffd*I(temp^2)+log(nfl),
                        subset(data_plants,year==2018))
summ(selectionabs_2017_1)
```

Observations	245
Dependent variable	nseed
Type	OLS linear regression

F(6,238)	34.508
R^2	0.465
Adj. R^2	0.452

```
summ(selectionabs_2018_1)
```

Quadratic terms of ffd not significant. Refit models withouth quadratic terms of ffd.

```
selectionabs_2017_2<-lm(nseed~ffd*temp+log(nfl),subset(data_plants,year==2017))
selectionabs_2018_2<-lm(nseed~ffd*temp+log(nfl),subset(data_plants,year==2018))
summ(selectionabs_2017_2)
```

	Est.	S.E.	t val.	p
(Intercept)	-6682.670	4879.792	-1.369	0.172
ffd	35.520	27.405	1.296	0.196
temp	394.423	560.245	0.704	0.482
I(temp^2)	-8.887	15.022	-0.592	0.555
log(nfl)	625.843	45.793	13.667	0.000
ffd:temp	-2.451	3.181	-0.770	0.442
ffd:I(temp^2)	0.054	0.086	0.627	0.531

Standard errors: OLS

Observations	104
Dependent variable	nseed
Type	OLS linear regression

F(6,97)	22.809
R^2	0.585
Adj. R^2	0.560

	Est.	S.E.	t val.	p
(Intercept)	1263.606	1411.947	0.895	0.373
ffd	-7.074	7.767	-0.911	0.365
temp	-134.639	192.293	-0.700	0.485
I(temp^2)	1.104	5.710	0.193	0.847
log(nfl)	197.641	20.943	9.437	0.000
ffd:temp	0.676	1.101	0.614	0.541
ffd:I(temp^2)	-0.005	0.033	-0.146	0.884

Standard errors: OLS

Observations	245
Dependent variable	nseed
Type	OLS linear regression

F(4,240)	51.549
R^2	0.462
Adj. R^2	0.453

	Est.	S.E.	t val.	p
(Intercept)	-4647.341	2606.895	-1.783	0.076
ffd	23.284	14.531	1.602	0.110
temp	88.601	124.771	0.710	0.478
log(nfl)	619.124	45.314	13.663	0.000
ffd:temp	-0.601	0.709	-0.848	0.397

Standard errors: OLS

```
summ(selectionabs_2018_2)
```

Observations	104
Dependent variable	nseed
Type	OLS linear regression

F(4,99)	34.660
R²	0.583
Adj. R²	0.567

	Est.	S.E.	t val.	p
(Intercept)	755.873	607.739	1.244	0.217
ffd	-4.561	3.188	-1.431	0.156
temp	-81.835	33.624	-2.434	0.017
log(nfl)	198.339	20.492	9.679	0.000
ffd:temp	0.425	0.190	2.241	0.027

Standard errors: OLS

BCa intervals

Used for assessing significance.

2017

```
# ffd
slp <- function(selectionabs_2017_2) coef(selectionabs_2017_2)[2]
b <- car::Boot(selectionabs_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_17_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selectionabs_2017_2) coef(selectionabs_2017_2)[3]
b <- car::Boot(selectionabs_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci_17_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# nfl
slp <- function(selectionabs_2017_2) coef(selectionabs_2017_2)[4]
b <- car::Boot(selectionabs_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_17_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# ffd:temp
slp <- function(selectionabs_2017_2) coef(selectionabs_2017_2)[5]
b <- car::Boot(selectionabs_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_17_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# Save confidence intervals as a table
BCIs_selection_2017_abs <- cbind(
  rbind(ffd_ci_17_abs[1,] ,temp_ci_17_abs[1,], nfl_ci_17_abs[1,],
        ffd_temp_ci_17_abs[1,]),
  rbind(ffd_ci_17_abs[2,] ,temp_ci_17_abs[2,], nfl_ci_17_abs[2,],
        ffd_temp_ci_17_abs[2,])
)
colnames(BCIs_selection_2017_abs)<-c("lower","upper")
rownames(BCIs_selection_2017_abs) <- c("ffd","temp","nfl","ffd:temp")
save(BCIs_selection_2017_abs,file="output/BCIs_selection_2017_abs.RData")
```

```
BCIs_selection_2017_abs
```

```
##                lower       upper
## ffd        0.8196429  54.8760193
## temp     -83.1277997 295.4774357
## nfl      462.7282320 870.4565209
## ffd:temp  -1.7818027   0.3868525
```

2018

```
# ffd
slp <- function(selectionabs_2018_2) coef(selectionabs_2018_2)[2]
b <- car::Boot(selectionabs_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_18_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selectionabs_2018_2) coef(selectionabs_2018_2)[3]
b <- car::Boot(selectionabs_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci_18_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# nfl
slp <- function(selectionabs_2018_2) coef(selectionabs_2018_2)[4]
b <- car::Boot(selectionabs_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_18_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# ffd:temp
slp <- function(selectionabs_2018_2) coef(selectionabs_2018_2)[5]
b <- car::Boot(selectionabs_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_18_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# Save confidence intervals as a table
BCIs_selection_2018_abs <- cbind(
  rbind(ffd_ci_18_abs[1,] ,temp_ci_18_abs[1,], nfl_ci_18_abs[1,],
        ffd_temp_ci_18_abs[1,]),
  rbind(ffd_ci_18_abs[2,] ,temp_ci_18_abs[2,], nfl_ci_18_abs[2,],
        ffd_temp_ci_18_abs[2,])
)
colnames(BCIs_selection_2018_abs)<-c("lower","upper")
rownames(BCIs_selection_2018_abs) <- c("ffd","temp","nfl","ffd:temp")
save(BCIs_selection_2018_abs,file="output/BCIs_selection_2018_abs.RData")
```

```
BCIs_selection_2018_abs
```

```
##                  lower       upper
## ffd       -11.64619815   3.3967383
## temp     -151.51624752 -14.4194867
## nfl       149.24336151 255.5197540
## ffd:temp    0.06064084   0.7961516
```

R Session Info

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Sweden.1252  LC_CTYPE=English_Sweden.1252
## [3] LC_MONETARY=English_Sweden.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Sweden.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] ggrepel_0.9.1      viridis_0.6.1      viridisLite_0.4.0  car_3.0-10
##  [5] lmtest_0.9-38      zoo_1.8-9          ggforce_0.3.3      lubridate_1.7.10
##  [9] effects_4.2-0      carData_3.0-4      segmented_1.3-4    MASS_7.3-54
## [13] MuMIn_1.43.17      ggeffects_1.1.0    kableExtra_1.3.4   jtools_2.1.3
## [17] ggpubr_0.4.0       broom_0.7.6        RColorBrewer_1.1-2 DHARMa_0.4.1
## [21] gridExtra_2.3      knitr_1.33         ggthemes_4.2.4     forcats_0.5.1
## [25] stringr_1.4.0      dplyr_1.0.6        purrr_0.3.4        readr_1.4.0
## [29] tidyr_1.1.3        tibble_3.1.2       ggplot2_3.3.5      tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] minqa_1.2.4       colorspace_2.0-1  ggsignif_0.6.1    ellipsis_0.3.2
##  [5] rio_0.5.26        sjlabelled_1.1.8  fs_1.5.0          rstudioapi_0.13  
##  [9] farver_2.1.0      fansi_0.4.2       xml2_1.3.2        codetools_0.2-18 
## [13] splines_4.1.0     polyclip_1.10-0   jsonlite_1.7.2    nloptr_1.2.2.2   
## [17] dbplyr_2.1.1      compiler_4.1.0    httr_1.4.2        backports_1.2.1  
## [21] assertthat_0.2.1  Matrix_1.3-3      survey_4.0        cli_2.5.0        
## [25] tweenr_1.0.2      htmltools_0.5.1.1 tools_4.1.0       gtable_0.3.0     
## [29] glue_1.4.2        Rcpp_1.0.6        cellranger_1.1.0  vctrs_0.3.8      
## [33] svglite_2.0.0     nlme_3.1-152      iterators_1.0.13  insight_0.14.3   
## [37] xfun_0.22         openxlsx_4.2.3    lme4_1.1-27       rvest_1.0.0      
## [41] lifecycle_1.0.0   rstatix_0.7.0     scales_1.1.1      hms_1.1.0        
## [45] yaml_2.2.1        curl_4.3.1        pander_0.6.3      stringi_1.6.1    
## [49] highr_0.9         foreach_1.5.1     boot_1.3-28       zip_2.1.1        
## [53] rlang_0.4.10      pkgconfig_2.0.3   systemfonts_1.0.2 evaluate_0.14    
## [57] lattice_0.20-44   labeling_0.4.2    tidyselect_1.1.1  magrittr_2.0.1   
## [61] R6_2.5.0          generics_0.1.0    DBI_1.1.1         mgcv_1.8-35      
## [65] pillar_1.6.1      haven_2.4.1       foreign_0.8-81    withr_2.4.2      
## [69] survival_3.2-11   abind_1.4-5       nnet_7.3-16       modelr_0.1.8     
## [73] crayon_1.4.1      utf8_1.2.1        rmarkdown_2.8     readxl_1.3.1     
## [77] data.table_1.14.0 reprex_2.0.0      digest_0.6.27     webshot_0.5.2    
## [81] stats4_4.1.0      munsell_0.5.0     mitools_2.4
```