

Increasing crop rotational diversity can enhance cereal yields

Alessio Costa and Monique Smith

2023-02-27

sessionInfo() R version 4.2.2 (2022-10-31) Platform: x86_64-apple-darwin17.0 (64-bit) Running under: macOS Ventura 13.2.1

Matrix products: default LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib

locale: [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages: [1] stats graphics grDevices utils datasets methods base

other attached packages: [1] parameters_0.20.2 ggpubr_0.6.0 emmeans_1.8.4-1 merTools_0.5.2 arm_1.13-1

[6] MASS_7.3-58.1 readr_2.1.4 readxl_1.4.2 ggrepel_0.9.3 gridExtra_2.3

[11] tidyr_1.3.0 plyr_1.8.8 effects_4.2-2 glmmTMB_1.1.5 DHARMA_0.4.6

[16] MuMIn_1.47.1 sjstats_0.18.2 dplyr_1.1.0 reshape2_1.4.4 plotly_4.10.1

[21] ggeffects_1.1.5 car_3.1-1 carData_3.0-5 xlsx_0.6.5 lattice_0.20-45

[26] ggplot2_3.4.1 listviewer_3.0.0 lmerTest_3.1-3 lme4_1.1-31 Matrix_1.5-1

loaded via a namespace (and not attached): [1] minqa_1.2.5 colorspace_2.1-0 ggsignif_0.6.4 ellipsis_0.3.2

[5] sjlabelled_1.2.0 estimability_1.4.1 rstudioapi_0.14 listenv_0.9.0

[9] furrr_0.3.1 fansi_1.0.4 mvtnorm_1.1-3 codetools_0.2-18

[13] splines_4.2.2 knitr_1.42 sjmisc_2.8.9 jsonlite_1.8.4

[17] nloptr_2.0.3 rJava_1.0-6 broom_1.0.3 broom.mixed_0.2.9.4 [21] shiny_1.7.4 compiler_4.2.2 httr_1.4.4 backports_1.4.1

[25] fastmap_1.1.0 lazyeval_0.2.2 survey_4.1-1 cli_3.6.0

[29] later_1.3.0 htmltools_0.5.4 tools_4.2.2 coda_0.19-4

[33] gtable_0.3.1 glue_1.6.2 Rcpp_1.0.10 cellranger_1.1.0

[37] vctrs_0.5.2 nlme_3.1-160 iterators_1.0.14 insight_0.19.0

[41] xfun_0.37 stringr_1.5.0 globals_0.16.2 xlsxjars_0.6.1

[45] mime_0.12 lifecycle_1.0.3 rstatix_0.7.2 future_1.31.0

[49] scales_1.2.1 promises_1.2.0.1 hms_1.1.2 parallel_4.2.2

[53] TMB_1.9.2 yaml_2.3.7 stringi_1.7.12 bayestestR_0.13.0

[57] foreach_1.5.2 blme_1.0-5 boot_1.3-28 rlang_1.0.6

[61] pkgconfig_2.0.3 evaluate_0.20 purrr_1.0.1 htmlwidgets_1.6.1

[65] cowplot_1.1.1 tidyselect_1.2.0 parallelly_1.34.0 magrittr_2.0.3

[69] R6_2.5.1 generics_0.1.3 DBI_1.1.3 pillar_1.8.1

[73] withr_2.5.0 survival_3.4-0 datawizard_0.6.5 abind_1.4-5

[77] nnet_7.3-18 tibble_3.1.8 performance_0.10.2 modelr_0.1.10

[81] utf8_1.2.3 tzdb_0.3.0 rmarkdown_2.20 grid_4.2.2

[85] data.table_1.14.8 forcats_1.0.0 digest_0.6.31 xtable_1.8-4

[89] httpuv_1.6.9 numDeriv_2016.8-1.1 stats4_4.2.2 munsell_0.5.0

[93] viridisLite_0.4.1 mitools_2.4 egg_0.4.5

```
library(listviewer)
library(ggplot2)
library(lattice)
```

```
library(xlsx)
library(car)
```

```
## Loading required package: carData
```

```
library(lme4) #citation(package = "lme4")
```

```
## Loading required package: Matrix
```

```
library(lmerTest)
```

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

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

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

```
library(ggeffects) #citation(package = "ggeffects")
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##   last_plot
```

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

```
## The following object is masked from 'package:graphics':
##
##   layout
```

```
library(reshape2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:car':
##
##   recode
```

```
## The following objects are masked from 'package:stats':  
##  
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(sjstats)  
library(MuMIn)  
library(DHARMA) #citation(package = "DHARMA")
```

```
## This is DHARMA 0.4.6. For overview type '?DHARMA'. For recent changes, type news(package = 'DHARMA')
```

```
library(glmmTMB)
```

```
## Warning in checkDepPackageVersion(dep_pkg = "TMB"): Package version inconsistency detected.  
## glmmTMB was built with TMB version 1.9.1  
## Current TMB version is 1.9.2  
## Please re-install glmmTMB from source or restore original 'TMB' package (see '?reinstalling' for more)
```

```
library(effects)
```

```
## Use the command  
##   lattice::trellis.par.set(effectsTheme())  
## to customize lattice options for effects plots.  
## See ?effectsTheme for details.
```

```
library(plyr)
```

```
## -----  
## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)
```

```
## -----
```

```
##  
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':  
##  
##   arrange, count, desc, failwith, id, mutate, rename, summarise,  
##   summarize
```

```
## The following objects are masked from 'package:plotly':  
##  
##   arrange, mutate, rename, summarise
```

```
library(tidyr)
```

```
##  
## Attaching package: 'tidyr'  
  
## The following object is masked from 'package:reshape2':  
##  
##   smiths  
  
## The following objects are masked from 'package:Matrix':  
##  
##   expand, pack, unpack
```

```
library(gridExtra)
```

```
##  
## Attaching package: 'gridExtra'  
  
## The following object is masked from 'package:dplyr':  
##  
##   combine
```

```
library(ggrepel)  
library(readxl)  
library(readr)  
library(merTools)
```

```
## Loading required package: arm
```

```
## Loading required package: MASS
```

```
##  
## Attaching package: 'MASS'  
  
## The following object is masked from 'package:dplyr':  
##  
##   select
```

```
## The following object is masked from 'package:plotly':  
##  
##   select
```

```
##  
## arm (Version 1.13-1, built: 2022-8-25)
```

```
## Working directory is /Users/meth0003/Library/CloudStorage/GoogleDrive-mono.smith88@gmail.com/My Drive
```

```
##  
## Attaching package: 'arm'
```

```
## The following object is masked from 'package:MuMIn':  
##  
##   coefplot
```

```
## The following object is masked from 'package:car':  
##  
##   logit
```

```
library(emmeans)  
library(ggpubr)
```

```
##  
## Attaching package: 'ggpubr'
```

```
## The following object is masked from 'package:plyr':  
##  
##   mutate
```

```
library(parameters)
```

```
##  
## Attaching package: 'parameters'
```

```
## The following object is masked from 'package:arm':  
##  
##   display
```

```
## The following object is masked from 'package:sjstats':  
##  
##   p_value
```

```
#Data preparation
```

```
#Model selection species diversity
```

```
#detach("package:lmerTest", unload = TRUE) #necessary to look at AICs
```

```
#Maize model
```

```
Spring.df.US$fert=relevel(Spring.df.US$fert, ref="low") #Setting fertiliser reference level as low  
mX<- lmer(s.yield ~ D*time+I(D^2)*time+D*I(time^2)+I(D^2)*I(time^2)+fert+D:fert+I(D^2)*fert + #Full mod  
          (1|site:group) + (1|site:year.f), data=Spring.df.US)
```

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

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

```
drop1(mX, test="Chisq") #all model terms significantly improved the model, no removal necessary
```

```
## Single term deletions using Satterthwaite's method:
```

```
##
```

```
## Model:
```

```
## s.yield ~ D * time + I(D^2) * time + D * I(time^2) + I(D^2) * I(time^2) + fert + D:fert + I(D^2) * f
```

```
##
```

| | Sum Sq | Mean Sq | NumDF | DenDF | F value | Pr(>F) | |
|---------------------|---------|---------|-------|-------|---------|-----------|-----|
| ## D:time | 18.130 | 18.130 | 1 | 11498 | 15.425 | 8.637e-05 | *** |
| ## time:I(D^2) | 18.081 | 18.081 | 1 | 11505 | 15.382 | 8.832e-05 | *** |
| ## D:I(time^2) | 14.904 | 14.904 | 1 | 11486 | 12.679 | 0.0003712 | *** |
| ## I(D^2):I(time^2) | 20.441 | 20.441 | 1 | 11494 | 17.391 | 3.065e-05 | *** |
| ## D:fert | 237.616 | 237.616 | 1 | 11514 | 202.154 | < 2.2e-16 | *** |
| ## I(D^2):fert | 160.016 | 160.016 | 1 | 11512 | 136.135 | < 2.2e-16 | *** |

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(lmerTest)
```

```
#Saving the final model
```

```
mXf= lmer(s.yield ~ D*time+I(D^2)*time+D*I(time^2)+I(D^2)*I(time^2)+fert+D:fert+I(D^2)*fert +  
          (1|site:group) + (1|site:year.f), data=Spring.df.US)
```

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

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

```
#Spring small grain cereal model
```

```
Spring.df.EU$fert=relevel(Spring.df.EU$fert, ref="low")
```

```
detach("package:lmerTest", unload = TRUE)
```

```
sX<- lmer(s.yield ~ D*time+I(D^2)*time+D*I(time^2)+I(D^2)*I(time^2)+D*fert+I(D^2)*fert +  
          (1|site:group) + (1|site:year.f), data=Spring.df.EU)
```

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

```
drop1(sX, test="Chisq") #Removing several terms improved the model, the interaction of the highest order
```

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

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

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

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

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

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

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## s.yield ~ D * time + I(D^2) * time + D * I(time^2) + I(D^2) *
## I(time^2) + D * fert + I(D^2) * fert + (1 | site:group) +
## (1 | site:year.f)
```

```
##          npar   AIC    LRT   Pr(Chi)
## <none>          18249
## D:time          1 18253  6.067  0.01377 *
## time:I(D^2)    1 18250  3.376  0.06616 .
## D:I(time^2)    1 18247  0.154  0.69518
## I(D^2):I(time^2) 1 18247  0.336  0.56231
## D:fert         1 18282 34.830 3.599e-09 ***
## I(D^2):fert    1 18287 40.242 2.244e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
sX1<-update (sX, ~.
             -I(D^2):I(time^2)) #updating the model by removing unnecessary terms
```

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

```
drop1(sX1, test="Chisq") #removing D:I(time^2) improved the model
```

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

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

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

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

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## s.yield ~ D + time + I(D^2) + I(time^2) + fert + (1 | site:group) +
## (1 | site:year.f) + D:time + time:I(D^2) + D:I(time^2) +
## D:fert + I(D^2):fert
```

```
##          npar   AIC    LRT   Pr(Chi)
## <none>          18247
## D:time          1 18301 55.761 8.185e-14 ***
```

```
## time:I(D^2)    1 18309 64.025 1.228e-15 ***
## D:I(time^2)   1 18246  1.109   0.2923
## D:fert        1 18280 34.723 3.802e-09 ***
## I(D^2):fert   1 18286 40.159 2.341e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
sX2<-update (sX1, .~.
             -D:I(time^2)
             )
drop1(sX2, test="Chisq") #removing I(time^2) improved the model
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0031557 (tol = 0.002, component 1)
```

```
## Single term deletions
##
## Model:
## s.yield ~ D + time + I(D^2) + I(time^2) + fert + (1 | site:group) +
##   (1 | site:year.f) + D:time + time:I(D^2) + D:fert + I(D^2):fert
##           npar   AIC    LRT   Pr(Chi)
## <none>                18246
## I(time^2)             1 18244  0.054    0.817
## D:time                 1 18336 91.149 < 2.2e-16 ***
## time:I(D^2)           1 18312 67.742 < 2.2e-16 ***
## D:fert                 1 18280 35.459 2.604e-09 ***
## I(D^2):fert           1 18285 40.871 1.627e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
sX3<-update (sX2, .~.
             -I(time^2)
             )
drop1(sX3, test="Chisq") #no further removal necessary
```

```
## Single term deletions
##
## Model:
## s.yield ~ D + time + I(D^2) + fert + (1 | site:group) + (1 |
##   site:year.f) + D:time + time:I(D^2) + D:fert + I(D^2):fert
##           npar   AIC    LRT   Pr(Chi)
## <none>                18244
## D:time                 1 18334 91.136 < 2.2e-16 ***
## time:I(D^2)           1 18310 67.744 < 2.2e-16 ***
## D:fert                 1 18278 35.465 2.596e-09 ***
## I(D^2):fert           1 18283 40.876 1.622e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
library(lmerTest)
```

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

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

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

```
#Saving the final model
```

```
sX= lmer(s.yield ~ D*time+I(D^2)*time+D*I(time^2)+I(D^2)*I(time^2)+fert+D:fert+I(D^2)*fert +  
        (1|site:group) + (1|site:year.f), data=Spring.df.EU)
```

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

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

```
sXf<-update (sX, .~.  
            -I(D^2):I(time^2)  
            -D:I(time^2)  
            -I(time^2)  
            )
```

```
#Winter small grain cereal model
```

```
Winter.df$fert=relevel(Winter.df$fert, ref="low")  
detach("package:lmerTest", unload = TRUE)  
wX<- lmer(s.yield ~ D*time+I(D^2)*time+D*I(time^2)+I(D^2)*I(time^2)+D*fert+I(D^2)*fert +  
        (1|site:group) + (1|site:year.f), data=Winter.df)
```

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

```
drop1(wX, test="Chisq") #Removing several terms improved the model, the interaction of the highest order
```

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

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

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

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

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

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

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## s.yield ~ D * time + I(D^2) * time + D * I(time^2) + I(D^2) *
## I(time^2) + D * fert + I(D^2) * fert + (1 | site:group) +
## (1 | site:year.f)
```

| | npar | AIC | LRT | Pr(Chi) | |
|------------------|------|-------|---------|----------|----|
| <none> | | 16987 | | | |
| D:time | 1 | 16993 | 7.9281 | 0.004867 | ** |
| time:I(D^2) | 1 | 16986 | 0.9210 | 0.337219 | |
| D:I(time^2) | 1 | 16986 | 1.5015 | 0.220440 | |
| I(D^2):I(time^2) | 1 | 16985 | 0.0017 | 0.967257 | |
| D:fert | 1 | 16996 | 10.6735 | 0.001087 | ** |
| I(D^2):fert | 1 | 16990 | 5.4191 | 0.019918 | * |

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
wX1<-update (wX, .~. #updating the model by removing unnecessary terms
             -I(D^2):I(time^2)
             )
```

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

```
drop1(wX1, test="Chisq") #no further removal necessary
```

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

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

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

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

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## s.yield ~ D + time + I(D^2) + I(time^2) + fert + (1 | site:group) +
## (1 | site:year.f) + D:time + time:I(D^2) + D:I(time^2) +
```

```
##      D:fert + I(D^2):fert
##              npar   AIC   LRT   Pr(Chi)
## <none>                16985
## D:time                1 17049 66.475 3.543e-16 ***
## time:I(D^2)          1 16996 13.711 0.0002132 ***
## D:I(time^2)          1 17055 72.232 < 2.2e-16 ***
## D:fert                1 16994 10.672 0.0010875 **
## I(D^2):fert          1 16988  5.419 0.0199202 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
wXf=wX1 #final model D + time + I(D^2) + I(time^2) + fert + D:time + time:I(D^2) + D:I(time^2) + D:fert
```

```
library(lmerTest)
```

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

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

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

```
#Saving the final model
```

```
wX<- lmer(s.yield ~ D*time+I(D^2)*time+D*I(time^2)+I(D^2)*I(time^2)+D*fert+I(D^2)*fert +
          (1|site:group) + (1|site:year.f), data=Winter.df)
```

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

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

```
wXf<-update (wX, ~. #updating the model by removing unnecessary terms
            -I(D^2):I(time^2)
            )
```

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

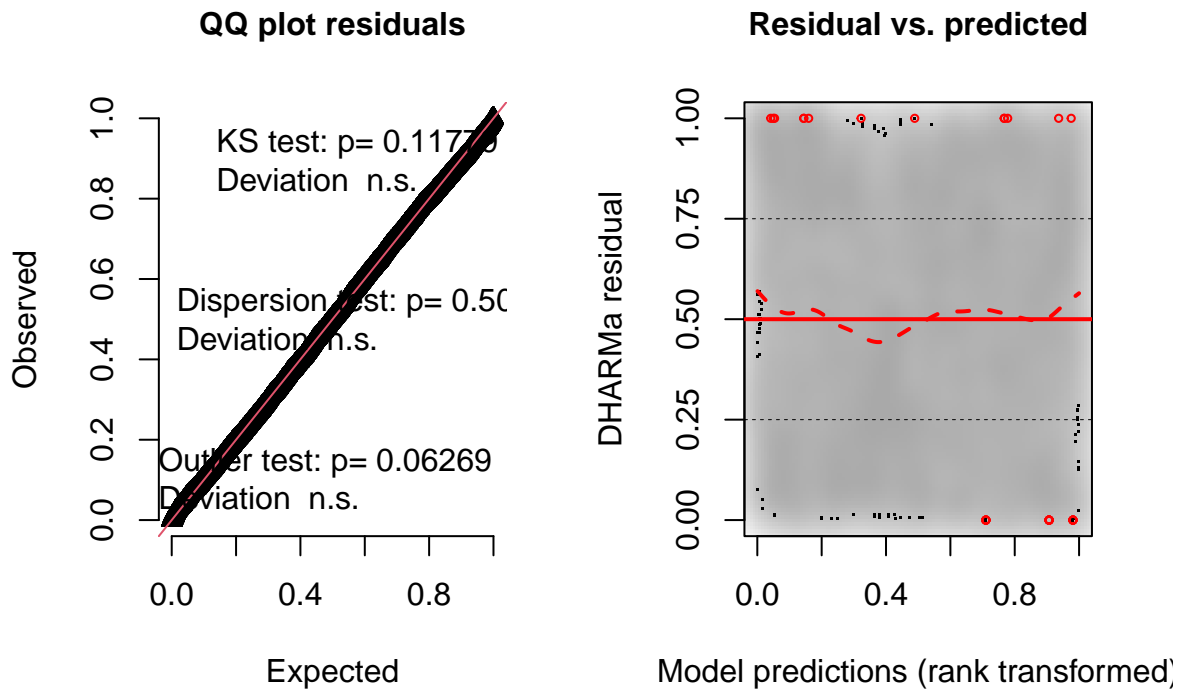
```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
#Model diagnostic species diversity Checking assumptions
```

```
#Maize model
```

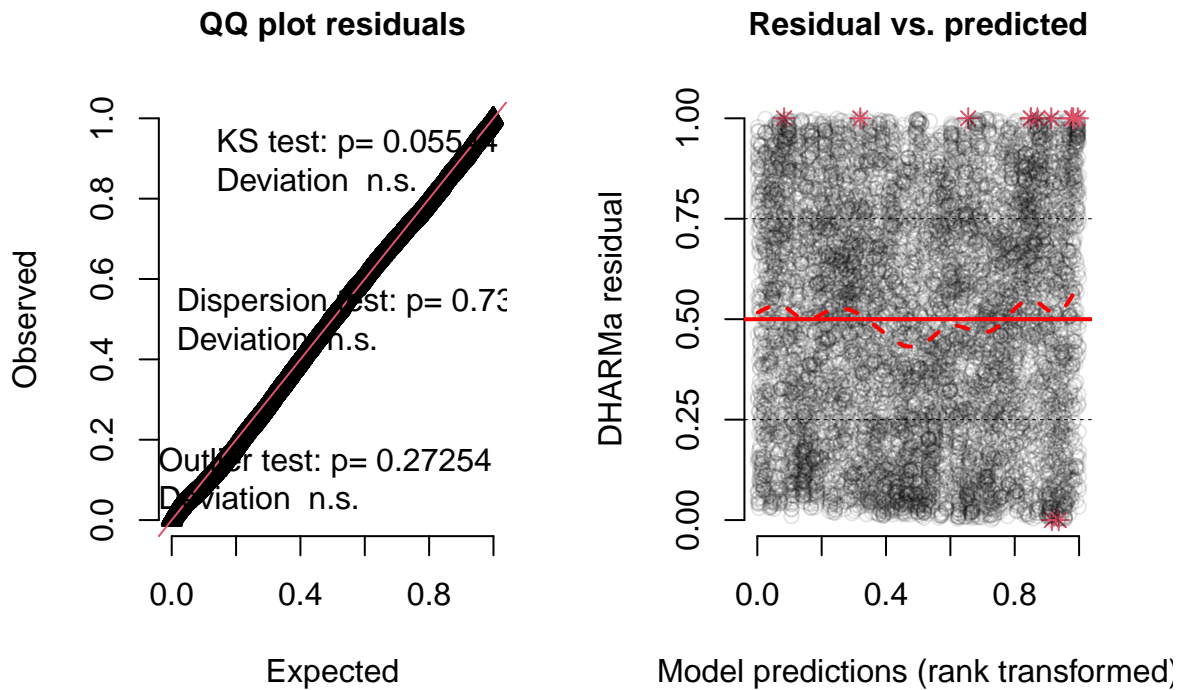
```
simulationOutput <- simulateResiduals(fittedModel = mXf, n=1000)
plot(simulationOutput, asFactor =F)
```

DHARMA residual



```
#Spring small grain cereals model  
simulationOutput <- simulateResiduals(fittedModel = sXf, n=1000)  
plot(simulationOutput, asFactor =F)
```

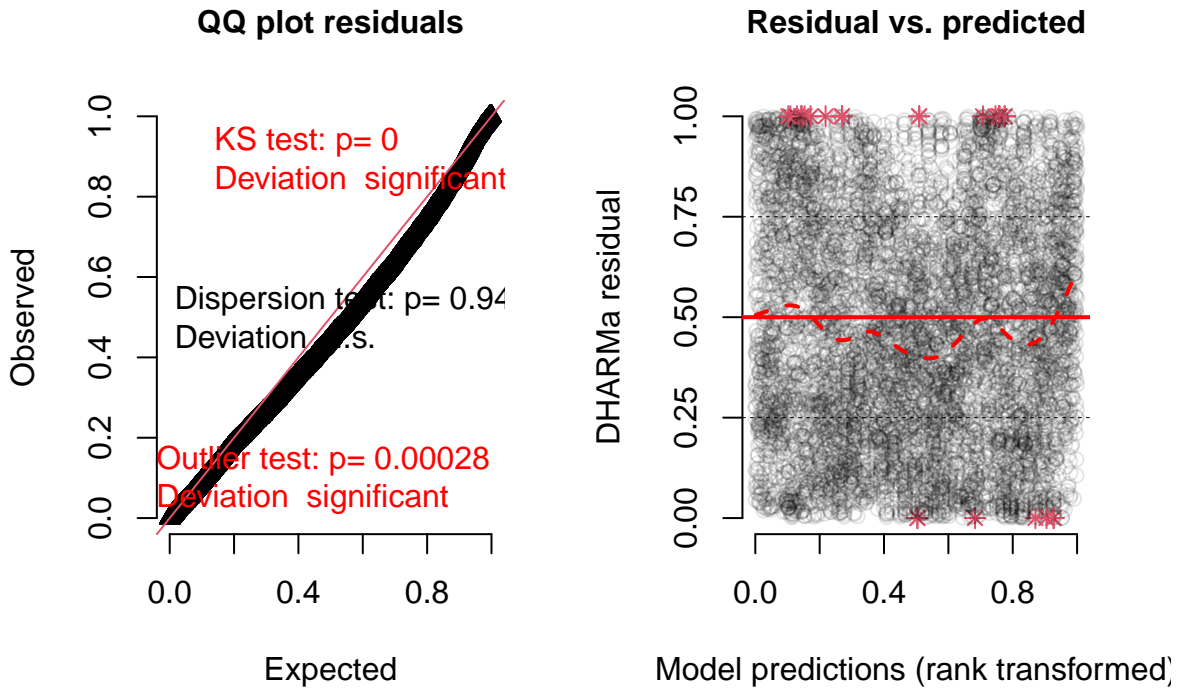
DHARMA residual



Significant deviation from KS test, however the plot suggests that there is a good fit between observed and expected quantiles. Since sample size is large, even slight deviations can be detected as significant. No action taken.

```
#Winter small grain cereals model  
simulationOutput <- simulateResiduals(fittedModel = wXf, n=1000)  
plot(simulationOutput, asFactor =F)
```

DHARMA residual



Significant deviation from KS test, however the plot suggests that there is a good fit between observed and expected quantiles. No action taken.

#Model selection functional richness

```
#Maize model
detach("package:lmerTest", unload = TRUE)
Spring.df.US$fert=relevel(Spring.df.US$fert, ref="low")
mX<- lmer(s.yield ~ F_types*time+F_types*I(time^2)+fert+F_types:fert+
          (1|site:group) + (1|site:year.f), data=Spring.df.US)

drop1(mX, test="Chisq") #no removal necessary
```

```
## Single term deletions
##
## Model:
## s.yield ~ F_types * time + F_types * I(time^2) + fert + F_types:fert +
## (1 | site:group) + (1 | site:year.f)
##          npar  AIC   LRT Pr(Chi)
## <none>          37524
## F_types:time    2 37597  76.85 < 2.2e-16 ***
## F_types:I(time^2) 2 37582  62.07 3.329e-14 ***
## F_types:fert    2 38161 640.68 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(lmerTest)
```

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

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

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

```
#saving the final model
```

```
mXf_f=lmer(s.yield ~ F_types*time+F_types*I(time^2)+fert+F_types:fert+  
           (1|site:group) + (1|site:year.f), data=Spring.df.US)
```

```
#Spring small grain cereal model
```

```
detach("package:lmerTest", unload = TRUE)  
Spring.df.EU$fert=relevel(Spring.df.EU$fert, ref="low")
```

```
sX<- lmer(s.yield ~ F_types*time+F_types*I(time^2)+fert+F_types:fert+  
          (1|site:group) + (1|site:year.f), data=Spring.df.EU)
```

```
drop1(sX, test="Chisq") #removing F_types:I(time^2) improved the model
```

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## s.yield ~ F_types * time + F_types * I(time^2) + fert + F_types:fert +  
## (1 | site:group) + (1 | site:year.f)
```

```
##          npar  AIC    LRT  Pr(Chi)  
## <none>          18200  
## F_types:time    3 18220 25.905 9.983e-06 ***  
## F_types:I(time^2) 3 18199  4.649  0.1994  
## F_types:fert    3 18284 90.441 < 2.2e-16 ***  
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
sX1<-update (sX, .~.  
            -F_types:I(time^2)  
            )
```

```
drop1(sX1, test="Chisq")
```

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## s.yield ~ F_types + time + I(time^2) + fert + (1 | site:group) +  
## (1 | site:year.f) + F_types:time + F_types:fert
```

```
##          npar   AIC     LRT Pr(Chi)
## <none>          18199
## I(time^2)      1 18197   0.001 0.9772
## F_types:time   3 18341 148.649 <2e-16 ***
## F_types:fert   3 18282  89.231 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
sX2<-update (sX1, .~. #removing I(time^2) improved the model
            -I(time^2)
            )
```

```
drop1(sX2, test="Chisq") #no further removal necessary
```

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## s.yield ~ F_types + time + fert + (1 | site:group) + (1 | site:year.f) +
```

```
##   F_types:time + F_types:fert
```

```
##          npar   AIC     LRT   Pr(Chi)
```

```
## <none>          18197
```

```
## F_types:time   3 18339 148.651 < 2.2e-16 ***
```

```
## F_types:fert   3 18280  89.234 < 2.2e-16 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(lmerTest)
```

```
##
```

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

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

```
##
```

```
##   lmer
```

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

```
##
```

```
##   step
```

```
#saving the final model
```

```
sX<- lmer(s.yield ~ F_types*time+F_types*I(time^2)+fert+F_types:fert+
          (1|site:group) + (1|site:year.f), data=Spring.df.EU)
```

```
sXf_f=update (sX, .~.
             -F_types:I(time^2)
             -I(time^2)
             )
```

```
#Winter small grain cereals
```

```
detach("package:lmerTest", unload = TRUE)
```

```
Winter.df$fert=relevel(Winter.df$fert, ref="low")
```

```
wX<- lmer(s.yield ~F_types*time+F_types*I(time^2)+fert+F_types:fert+
```



```

(1|site:group) + (1|site:year.f), data=Winter.df)

drop1(wX, test="Chisq") #no term removal necessary

```

```

## Single term deletions
##
## Model:
## s.yield ~ F_types * time + F_types * I(time^2) + fert + F_types:fert +
## (1 | site:group) + (1 | site:year.f)
##          npar    AIC    LRT   Pr(Chi)
## <none>          17136
## F_types:time    2 17179 46.663 7.365e-11 ***
## F_types:I(time^2) 2 17171 38.621 4.107e-09 ***
## F_types:fert    2 17190 58.459 2.022e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

#saving the final model
library(lmerTest)

```

```

##
## Attaching package: 'lmerTest'

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

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

```

```

wXf_f=lmer(s.yield ~F_types*time+F_types*I(time^2)+fert+F_types:fert+
(1|site:group) + (1|site:year.f), data=Winter.df)

```

```

#Model diagnostic functional richness Checking assumptions

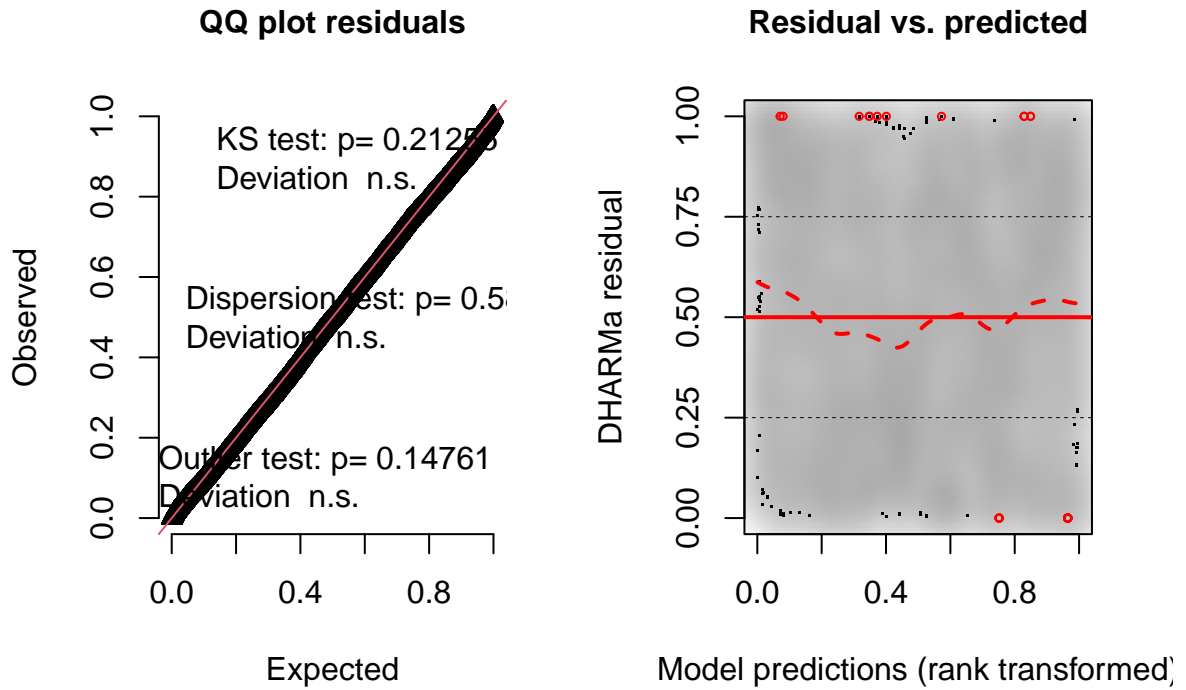
```

```

#Maize model
simulationOutput <- simulateResiduals(fittedModel = mXf_f, n=1000)
plot(simulationOutput, asFactor =F)

```

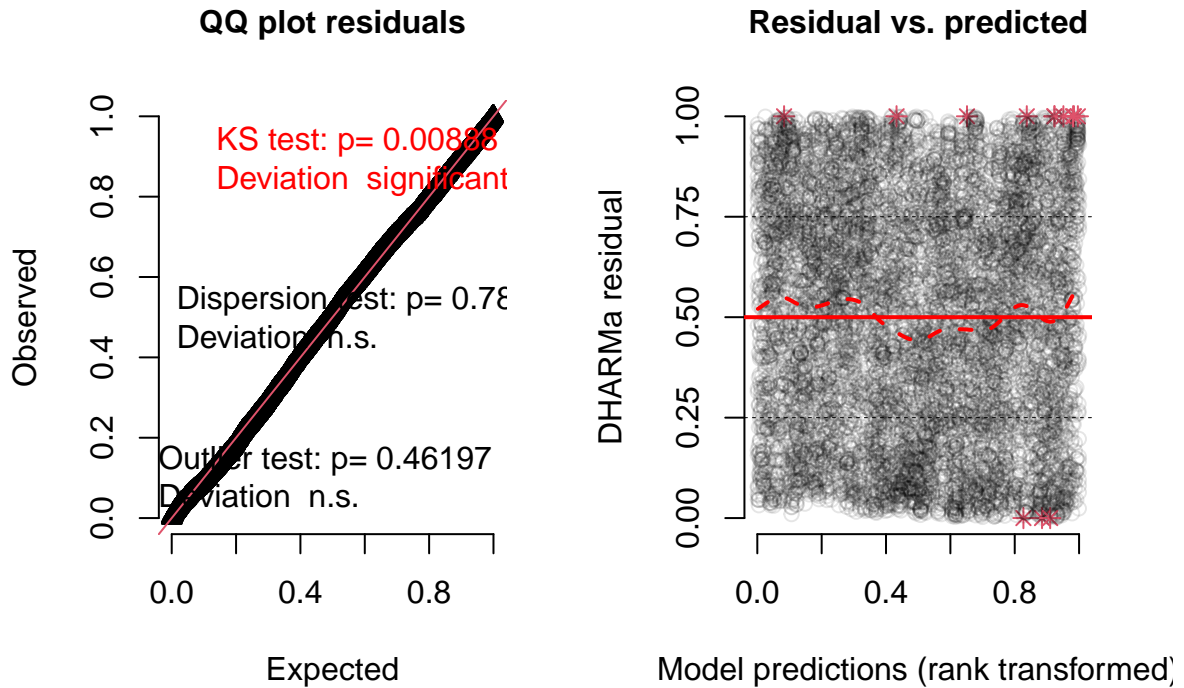
DHARMA residual



No issue detected.

```
#Spring small grain cereals model  
simulationOutput <- simulateResiduals(fittedModel = sXf_f, n=1000)  
plot(simulationOutput, asFactor =F)
```

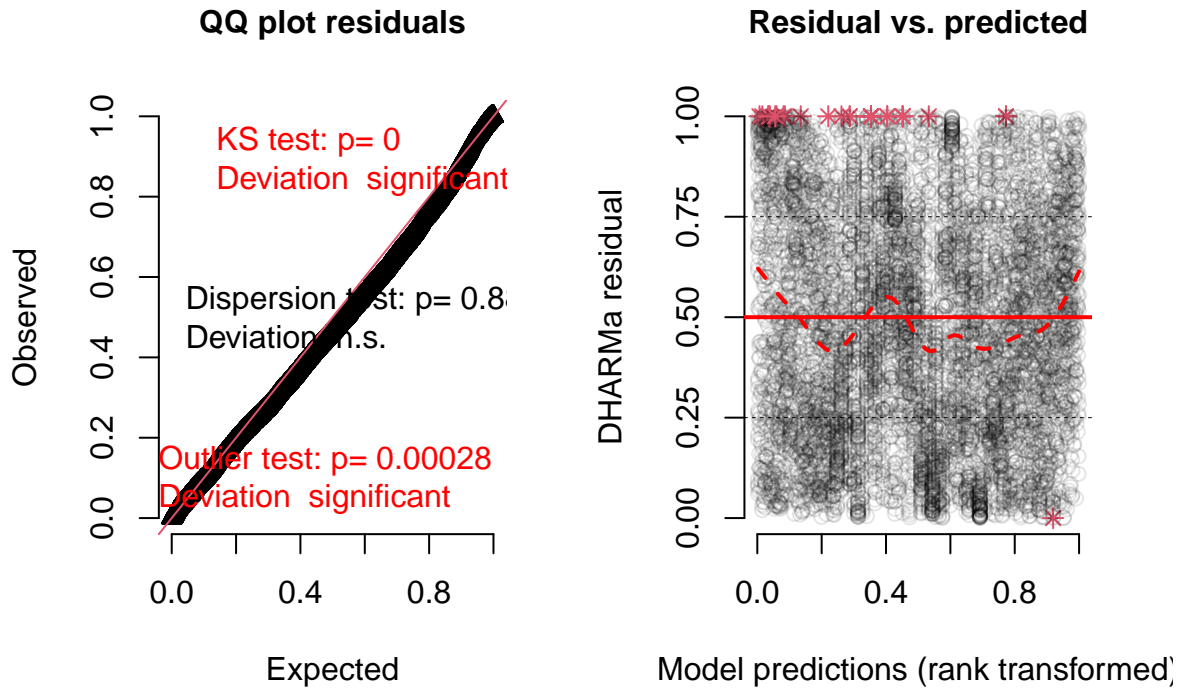
DHARMA residual



Significant deviation from KS test, however the plot suggests that there is a good fit between observed and expected quantiles. Since sample size is large, even slight deviations can be detected as significant. No action taken.

```
#Winter small grain cereals model  
simulationOutput <- simulateResiduals(fittedModel = wXf_f, n=1000)  
plot(simulationOutput, asFactor =F)
```

DHARMA residual



Significant deviation from KS test, however the plot suggests that there is a good fit between observed and expected quantiles. No action taken.

There was a significant deviation from the outlier test.

Identifying observations that produced a standard deviation higher than 2.5.

```
res1=resid(wXf_f, type = "pearson")
Winter.df[which(abs(res1) > 2.5),] #which(abs(res1) > 2.5) returns the row number of the observations t
```

| ## | country | site | lte | time | group | rot | crop | fert | ley | leg | bl | F_types | |
|----|---------|---------|-----------|------|-------|------|------|------|------|-----|----|---------|---|
| ## | 1536 | England | Broadbalk | BB | 4 | 1.90 | BB2 | WW | high | n | n | n | 1 |
| ## | 1539 | England | Broadbalk | BB | 4 | 1.15 | BB2 | WW | high | n | n | n | 1 |
| ## | 1587 | England | Broadbalk | BB | 6 | 1.15 | BB2 | WW | high | n | n | n | 1 |
| ## | 1590 | England | Broadbalk | BB | 6 | 1.16 | BB2 | WW | high | n | n | n | 1 |
| ## | 1730 | England | Broadbalk | BB | 12 | 1.15 | BB2 | WW | high | n | n | n | 1 |
| ## | 1868 | England | Broadbalk | BB | 18 | 1.80 | BB2 | WW | low | n | n | n | 1 |
| ## | 1871 | England | Broadbalk | BB | 18 | 1.90 | BB2 | WW | high | n | n | n | 1 |
| ## | 1874 | England | Broadbalk | BB | 18 | 1.15 | BB2 | WW | high | n | n | n | 1 |
| ## | 1875 | England | Broadbalk | BB | 18 | 1.15 | BB2 | WW | high | n | n | n | 1 |
| ## | 1877 | England | Broadbalk | BB | 18 | 1.16 | BB2 | WW | high | n | n | n | 1 |
| ## | 1878 | England | Broadbalk | BB | 18 | 1.16 | BB2 | WW | high | n | n | n | 1 |
| ## | 1923 | England | Broadbalk | BB | 20 | 1.15 | BB2 | WW | high | n | n | n | 1 |
| ## | 1926 | England | Broadbalk | BB | 20 | 1.16 | BB2 | WW | high | n | n | n | 1 |
| ## | 17190 | Italy | Padova | Pa | 1 | 1.30 | Pa1 | WW | high | n | n | n | 1 |
| ## | 17445 | Italy | Padova | Pa | 0 | 1.10 | Pa2 | WW | low | n | y | y | 3 |
| ## | 17569 | Italy | Padova | Pa | 13 | 2.30 | Pa2 | WW | high | n | y | y | 3 |

```

## 17688 Italy Padova Pa 27 1.10 Pa2 WW low n y y 3
## 18141 Italy Perugia Pe 0 1.00 Pe3 WW high n n n 1
## 18148 Italy Perugia Pe 1 1.00 Pe3 WW high n n n 1
## 18238 Italy Perugia Pe 16 1.00 Pe3 WW high n n n 1
## 18382 Italy Perugia Pe 0 1.00 Pe4 WW high n n n 1
## 21835 Sweden Saby_ley SL 10 1.00 SL2 WW low y y y 3
## 24717 Sweden Stenstugu_ley St 33 1.00 St1 WW low n n y 2
## 24718 Sweden Stenstugu_ley St 33 1.20 St1 WW high n n y 2
## 25563 England Woburn Wo 1 1.10 Wo1 WW low n y n 2
## 25657 England Woburn Wo 18 1.00 Wo1 WW low n y n 2
## 25662 England Woburn Wo 18 1.30 Wo1 WW high n y n 2
## 25665 England Woburn Wo 19 1.00 Wo1 WW low n y n 2
## 25695 England Woburn Wo 1 1.30 Wo2 WW high y y n 2
## F_comb D phenology year.f s.yield
## 1536 C 2.272727 winter 2000 -1.9163073
## 1539 C 2.272727 winter 2000 -1.7378073
## 1587 C 2.272727 winter 2002 4.0506927
## 1590 C 2.272727 winter 2002 4.5606927
## 1730 C 2.272727 winter 2008 4.8836927
## 1868 C 2.272727 winter 2014 4.9516927
## 1871 C 2.272727 winter 2014 5.7761927
## 1874 C 2.272727 winter 2014 6.2691927
## 1875 C 2.272727 winter 2014 5.6146927
## 1877 C 2.272727 winter 2014 6.5836927
## 1878 C 2.272727 winter 2014 6.2521927
## 1923 C 2.272727 winter 2016 4.9686927
## 1926 C 2.272727 winter 2016 5.4531927
## 17190 C 1.000000 winter 1991 2.5651814
## 17445 C,B,LEY 3.600000 winter 1990 1.5733814
## 17569 C,B,LEY 3.600000 winter 2003 3.7421814
## 17688 C,B,LEY 3.600000 winter 2017 -2.5453068
## 18141 C 1.800000 winter 1974 -3.4646052
## 18148 C 1.800000 winter 1975 -3.2123052
## 18238 C 1.800000 winter 1990 -2.5337052
## 18382 C 1.600000 winter 1974 -3.6212052
## 21835 C,B,LEY 4.500000 winter 1979 1.1902210
## 24717 C,B 6.000000 winter 2001 0.8319831
## 24718 C,B 6.000000 winter 2001 -2.7950169
## 25563 C,LEG 5.000000 winter 1999 -3.4058190
## 25657 C,LEG 5.000000 winter 2016 -3.7346926
## 25662 C,LEG 5.000000 winter 2016 4.4535291
## 25665 C,LEG 5.000000 winter 2017 0.8239509
## 25695 C,LEY 2.272727 winter 1999 2.6716810

```

This resulted in 29 outliers from English, Italian, Polish and Swedish sites.

Below you can see the models summary with and without outliers.

```

W.NO=Winter.df[-which(abs(res1) > 2.5),]
WIIIf=lmer(s.yield ~F_types*time+F_types*I(time^2)+fert+F_types:fert+
           (1|site:group) + (1|site:year.f), data=W.NO)
model_parameters(wXf_f, effects="fixed", summary=T)

```

```
## # Fixed Effects
```

```

##
## Parameter          | Coefficient | SE | 95% CI | t(7183) | p
## -----
## (Intercept)        | -1.66 | 0.17 | [-2.00, -1.32] | -9.49 | < .001
## F types [2]        | 0.60 | 0.11 | [ 0.39, 0.81] | 5.70 | < .001
## F types [3]        | 0.76 | 0.08 | [ 0.60, 0.93] | 8.99 | < .001
## time               | -0.02 | 0.01 | [-0.04, 0.00] | -1.83 | 0.068
## time^2             | 7.59e-04 | 2.16e-04 | [ 0.00, 0.00] | 3.51 | < .001
## fert [high]        | 1.82 | 0.17 | [ 1.49, 2.14] | 10.98 | < .001
## F types [2] × time | 0.02 | 8.59e-03 | [ 0.00, 0.03] | 1.81 | 0.071
## F types [3] × time | 0.04 | 5.77e-03 | [ 0.03, 0.05] | 6.83 | < .001
## F types [2] × time^2 | -3.52e-04 | 2.01e-04 | [ 0.00, 0.00] | -1.76 | 0.079
## F types [3] × time^2 | -5.95e-04 | 9.65e-05 | [ 0.00, 0.00] | -6.17 | < .001
## F types [2] × fert [high] | -0.17 | 0.09 | [-0.34, 0.00] | -1.99 | 0.046
## F types [3] × fert [high] | -0.46 | 0.06 | [-0.59, -0.34] | -7.19 | < .001
##
## Model: s.yield ~ F_types * time + F_types * I(time^2) + fert + F_types:fert (7198 Observations)
## Residual standard deviation: 0.722 (df = 7183)
## Conditional R2: 0.776; Marginal R2: 0.303

##
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.

```

```
model_parameters(WIIf, effects="fixed", summary=T)
```

```

## # Fixed Effects
##
## Parameter          | Coefficient | SE | 95% CI | t(7154) | p
## -----
## (Intercept)        | -1.65 | 0.17 | [-1.99, -1.31] | -9.53 | < .001
## F types [2]        | 0.60 | 0.10 | [ 0.40, 0.80] | 5.90 | < .001
## F types [3]        | 0.75 | 0.08 | [ 0.59, 0.91] | 9.17 | < .001
## time               | -0.02 | 0.01 | [-0.04, 0.00] | -1.90 | 0.058
## time^2             | 7.70e-04 | 2.16e-04 | [ 0.00, 0.00] | 3.57 | < .001
## fert [high]        | 1.81 | 0.16 | [ 1.49, 2.13] | 11.05 | < .001
## F types [2] × time | 0.02 | 8.29e-03 | [ 0.00, 0.03] | 1.85 | 0.064
## F types [3] × time | 0.04 | 5.58e-03 | [ 0.03, 0.05] | 7.21 | < .001
## F types [2] × time^2 | -3.45e-04 | 1.93e-04 | [ 0.00, 0.00] | -1.78 | 0.075
## F types [3] × time^2 | -6.04e-04 | 9.31e-05 | [ 0.00, 0.00] | -6.49 | < .001
## F types [2] × fert [high] | -0.17 | 0.08 | [-0.33, -0.01] | -2.05 | 0.041
## F types [3] × fert [high] | -0.47 | 0.06 | [-0.59, -0.34] | -7.50 | < .001
##
## Model: s.yield ~ F_types * time + F_types * I(time^2) + fert + F_types:fert (7169 Observations)
## Residual standard deviation: 0.695 (df = 7154)
## Conditional R2: 0.788; Marginal R2: 0.307

##
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.

```

Removing the outliers does not lead to dramatic changes in the estimates. The outliers will be kept in the analyses.

Main figures

Figure 1 - maps- not included

##Figure 2

```
#Species diversity
Spring.US.effect1 <- ggeffect(mXf, terms = c("D[1:4.6, by=0.1]", "time [5,20,35]", "fert"))
Spring.US.ref1 <- ggeffect(mXf, terms = c("D[1,2]", "time [0,1]", "fert"))
Spring.US.effect1$conf.low <-Spring.US.effect1$conf.low-Spring.US.ref1$predicted[1]
Spring.US.effect1$conf.high <-Spring.US.effect1$conf.high-Spring.US.ref1$predicted[1]
Spring.US.effect1$predicted <-Spring.US.effect1$predicted-Spring.US.ref1$predicted[1]
Winter.effect1 <- ggeffect(wXf, terms = c("D[1:6, by=0.1]", "time [5,20,35]", "fert"))
Winter.ref1 <- ggeffect(wXf, terms = c("D[1,2]", "time [0,1]", "fert"))
Winter.effect1$conf.low <-Winter.effect1$conf.low-Winter.ref1$predicted[1]
Winter.effect1$conf.high <-Winter.effect1$conf.high-Winter.ref1$predicted[1]
Winter.effect1$predicted <-Winter.effect1$predicted-Winter.ref1$predicted[1]
Spring.EU.effect1 <- ggeffect(sXf, terms = c("D[1:6, by=0.1]", "time [5,20,35]", "fert"))
Spring.EU.ref1 <- ggeffect(sXf, terms = c("D[1,2]", "time [0,1]", "fert"))
Spring.EU.effect1$conf.low <-Spring.EU.effect1$conf.low-Spring.EU.ref1$predicted[1]
Spring.EU.effect1$conf.high <-Spring.EU.effect1$conf.high-Spring.EU.ref1$predicted[1]
Spring.EU.effect1$predicted <-Spring.EU.effect1$predicted-Spring.EU.ref1$predicted[1]
#Functional richness

Spring.US.effect2 <- ggeffect(mXf_f, terms = c("F_types", "time [5,20,35]", "fert"))
Spring.US.ref2 <- ggeffect(mXf_f, terms = c("F_types[1,2]", "time [0,1]", "fert"))
Spring.US.effect2$conf.low <-Spring.US.effect2$conf.low-Spring.US.ref2$predicted[1]
Spring.US.effect2$conf.high <-Spring.US.effect2$conf.high-Spring.US.ref2$predicted[1]
Spring.US.effect2$predicted <-Spring.US.effect2$predicted-Spring.US.ref2$predicted[1]
Spring.EU.effect2 <- ggeffect(sXf_f, terms = c("F_types", "time [5,20,35]", "fert"))
Spring.EU.ref2 <- ggeffect(sXf_f, terms = c("F_types[1,2]", "time [0,1]", "fert"))
Spring.EU.effect2$conf.low <-Spring.EU.effect2$conf.low-Spring.EU.ref2$predicted[1]
Spring.EU.effect2$conf.high <-Spring.EU.effect2$conf.high-Spring.EU.ref2$predicted[1]
Spring.EU.effect2$predicted <-Spring.EU.effect2$predicted-Spring.EU.ref2$predicted[1]
Winter.effect2 <- ggeffect(wXf_f, terms = c("F_types", "time [5,20,35]", "fert"))
Winter.ref2 <- ggeffect(wXf_f, terms = c("F_types[1,2]", "time [0,1]", "fert"))
Winter.effect2$conf.low <-Winter.effect2$conf.low-Winter.ref2$predicted[1]
Winter.effect2$conf.high <-Winter.effect2$conf.high-Winter.ref2$predicted[1]
Winter.effect2$predicted <-Winter.effect2$predicted-Winter.ref2$predicted[1]

fert_labeller <- as_labeller(c("low" = "Low fertilisation", #Needed for fertilisation facet label
                             "high" = "High fertilisation"
                             ))

Spring.EU.a=ggplot(Spring.EU.effect1, aes(x = x, y = predicted, colour = group)) +
  geom_line(aes(linetype = group), size = 0.8, show.legend = FALSE) +
  geom_ribbon( aes(ymin = conf.low, ymax = conf.high, fill = group, color = NULL), alpha = .15, show.legend = FALSE) +
  theme_bw() +
  scale_color_manual(values=c('#66c2a5', '#fc8d62', '#8da0cb'))+
  scale_fill_manual(values=c('#66c2a5', '#fc8d62', '#8da0cb'), name="Time (yr)") +
  scale_linetype_manual(values=c("twodash", "dotted", "solid"), name="Time (yr)")+
```

```

labs(colour = "Time (yr)",
      x = "Species Diversity",
      y = "Yield benefit (t/ha)",
      title = "Spring small grain cereals"
)+
ylim(-0.5,3)+
facet_grid(~factor(facet,levels=c("low","high")), labeller = fert_labeller)

```

```

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

```

```

Spring.US.a=ggplot(Spring.US.effect1, aes(x = x, y = predicted, colour = group)) +
  geom_line(aes(linetype = group), size = 0.8, show.legend = FALSE) +
  geom_ribbon( aes(ymin = conf.low, ymax = conf.high, fill = group, color = NULL), alpha = .15, show.legend = FALSE) +
  theme_bw() +
  scale_color_manual(values=c('#66c2a5', '#fc8d62', '#8da0cb'))+
  scale_fill_manual(values=c('#66c2a5', '#fc8d62', '#8da0cb'), name="Time (yr)") +
  scale_linetype_manual(values=c("twodash", "dotted", "solid"), name="Time (yr)")+
  labs(colour = "Time (yr)",
        x = "Species Diversity",
        y = "Yield benefit (t/ha)",
        title = "Maize"
  )+
  xlim(1,6)+
  # ylim(-0.5,4.5)+
  facet_grid(~factor(facet,levels=c("low","high")), labeller = fert_labeller)

```

```

Winter.a=ggplot(Winter.effect1, aes(x = x, y = predicted, colour = group)) +
  geom_line(aes(linetype = group), size = 0.8) +
  geom_ribbon( aes(ymin = conf.low, ymax = conf.high, fill = group, color = NULL), alpha = .15) +
  theme_bw() +
  scale_color_manual(values=c('#66c2a5', '#fc8d62', '#8da0cb'))+
  scale_fill_manual(values=c('#66c2a5', '#fc8d62', '#8da0cb'), name="Time (yr)") +
  scale_linetype_manual(values=c("twodash", "dotted", "solid"), name="Time (yr)")+
  labs(colour = "Time (yr)",
        x = "Species Diversity",
        y = "Yield benefit (t/ha)",
        title = "Winter small grain cereals"
  )+
  ylim(-0.5,3)+
  facet_grid(~factor(facet,levels=c("low","high")), labeller = fert_labeller)

```

```

pd <- position_dodge(width=0.2)
Spring.EU.b=ggplot(Spring.EU.effect2, aes(x = x, y = predicted, colour = group)) +
  geom_point(position=pd, show.legend = FALSE) +
  geom_errorbar(aes(ymin = conf.low, ymax=conf.high, color = group), width=.15,size=.5,position=pd, show.legend = FALSE) +
  theme_bw() +
  scale_color_manual(values=c('#66c2a5', '#fc8d62', '#8da0cb'))+
  scale_fill_manual(values=c('#66c2a5', '#fc8d62', '#8da0cb'), name="Time (yr)") +
  labs(

```



```

    colour = "Time (yr)",
    x = "Functional Richness",
    y = "Yield benefit (t/ha)",
    title = ""
) +
theme(
  plot.title = element_text(vjust=-52,hjust=1.3)# move title to bottom right corner
)+
ylim(-0.4,3)+
facet_grid(~factor(facet,levels=c("low","high")), labeller = fert_labeller)
blank=ggplot(Spring.EU.effect2, aes(x = x, y = predicted, colour = group)) +
  geom_blank()

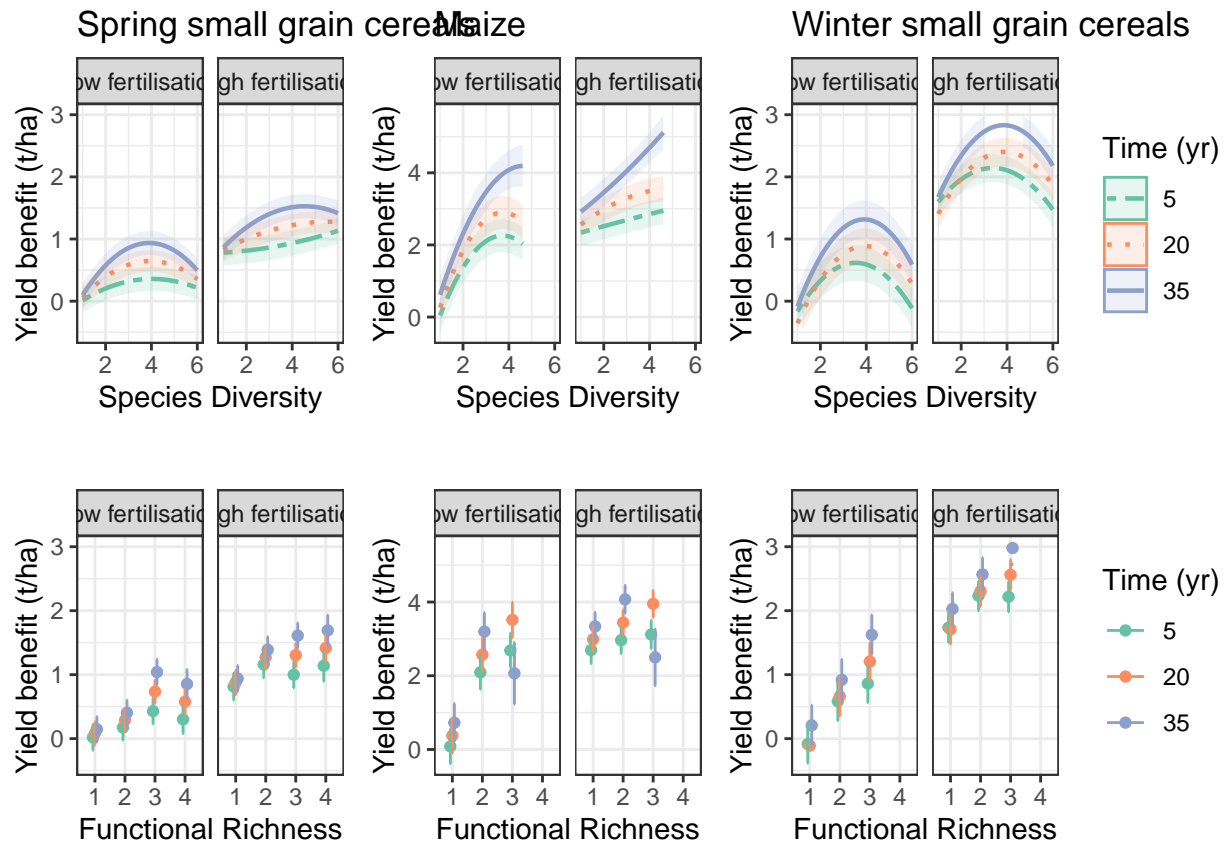
Spring.US.b=blank+
  geom_point(Spring.US.effect2, inherit.aes = TRUE, mapping=aes(x = x, y = predicted, colour = group),position="jitter",
  geom_errorbar(Spring.US.effect2, inherit.aes = TRUE,mapping=aes(ymin = conf.low, ymax=conf.high, color = group),width=0.2),
  theme_bw() +
  scale_color_manual(values=c('#66c2a5','#fc8d62', '#8da0cb'))+
  scale_fill_manual(values=c('#66c2a5','#fc8d62', '#8da0cb'), name="Time (yr)") +
  labs(
    colour = "Time (yr)",
    x = "Functional Richness",
    y = "Yield benefit (t/ha)",
    title = ""
  ) +
  theme(
    plot.title = element_text(vjust=-52,hjust=1.3)# move title to bottom right corner
  )+
  ylim(-0.4,5.5)+
  facet_grid(~factor(facet,levels=c("low","high")), labeller = fert_labeller)

blank=ggplot(Spring.EU.effect2, aes(x = x, y = predicted, colour = group))+
  facet_grid(~factor(facet,levels=c("low","high")), labeller = fert_labeller)+
  geom_blank()

Winter.b=blank+
  geom_point(Winter.effect2, inherit.aes = TRUE, mapping=aes(x = x, y = predicted, colour = group),position="jitter",
  geom_errorbar(Winter.effect2, inherit.aes = TRUE,mapping=aes(ymin = conf.low, ymax=conf.high, color = group),width=0.2),
  theme_bw() +
  scale_color_manual(values=c('#66c2a5','#fc8d62', '#8da0cb'))+
  scale_fill_manual(values=c('#66c2a5','#fc8d62', '#8da0cb'), name="Time (yr)") +
  labs(
    colour = "Time (yr)",
    x = "Functional Richness",
    y = "Yield benefit (t/ha)",
    title = ""
  ) +
  ylim(-0.4,3)+
  theme(
    plot.title = element_text(vjust=-52,hjust=1.3)# move title to bottom right corner
  )

```

```
egg::ggarrange(Spring.EU.a, Spring.US.a, Winter.a, Spring.EU.b, Spring.US.b, Winter.b, nrow=2, ncol=3)
```



##Fig 3

###Preparing required data subset - Species Diversity

```
for (t in c(5,20,35)) #loop code to generate lsmeans estimates at Time snapshot of 5, 20, 35 years
{
  #generating lsmean estimates for every 0.01 point of D - needed to extract yield-maximizing diversity
  print("Spring small grain cereals")
  hsls=as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(3,3.5,by=0.01), time=t,fert=c("hi",
  hsls=rbind(hsls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(3.5,4,by=0.01), time=t,
  hsls=rbind(hsls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(4,4.5,by=0.01), time=t,
  hsls=rbind(hsls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(4.5,5,by=0.01), time=t,
  hsls=rbind(hsls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(5,5.5,by=0.01), time=t,
  hsls=rbind(hsls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(5.5,6,by=0.01), time=t,
  hsmax=hsls[which(hsls$lsmean==max(hsls$lsmean)),] #extracting the lsmean estimate at the yield-maximizing diversity

  print("Maize") #repeating the same procedure for maize
  hmhls=as.data.frame(lsmeans(mXf, pairwise~D*time+D*fert, at=list(D=seq(3,3.5,by=0.01), time=t,fert=c("hi",
  hmhls=rbind(hmhls,as.data.frame(lsmeans(mXf, pairwise~D*time+D*fert, at=list(D=seq(3.5,4,by=0.01), time=t,
  hmhls=rbind(hmhls,as.data.frame(lsmeans(mXf, pairwise~D*time+D*fert, at=list(D=seq(4,4.57,by=0.01), time=t,
  hmhmax=hmhls[which(hmhls$lsmean==max(hmhls$lsmean)),]

  print("Winter small grain cereals") #repeating the same procedure for winter cereals
  hwls=as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(3,3.5,by=0.01), time=t,fert=c("hi",
```

```

hwls=rbind(hwls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(3.5,4,by=0.01), time=t,
hwls=rbind(hwls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(4.,4.5,by=0.01), time=t,
hwls=rbind(hwls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(4.5,5,by=0.01), time=t,
hwls=rbind(hwls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(5,5.5,by=0.01), time=t,
hwls=rbind(hwls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(5.5,6,by=0.01), time=t,
hwmax=hwls[which(hwls$lsmean==max(hwls$lsmean)),]

```

#Repeating the procedures (all 3 indicator crops) under LOW fertilisation

```

print("Spring small grain cereals")
sls=as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(3,3.5,by=0.01), time=t,fert=c("low"),
sls=rbind(sls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(3.5,4,by=0.01), time=t,fert=c("low"),
sls=rbind(sls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(4.,4.5,by=0.01), time=t,fert=c("low"),
sls=rbind(sls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(4.5,5,by=0.01), time=t,fert=c("low"),
sls=rbind(sls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(5,5.5,by=0.01), time=t,fert=c("low"),
sls=rbind(sls,as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=seq(5.5,6,by=0.01), time=t,fert=c("low"),
smax=sls[which(sls$lsmean==max(sls$lsmean)),]

```

```

print("Maize")
mls=as.data.frame(lsmeans(mXf, pairwise~D*time+D*fert, at=list(D=seq(3,3.5,by=0.01), time=t,fert=c("low"),
mls=rbind(mls,as.data.frame(lsmeans(mXf, pairwise~D*time+D*fert, at=list(D=seq(3.5,4,by=0.01), time=t,fert=c("low"),
mls=rbind(mls,as.data.frame(lsmeans(mXf, pairwise~D*time+D*fert, at=list(D=seq(4.,4.57,by=0.01), time=t,fert=c("low"),
mmax=mls[which(mls$lsmean==max(mls$lsmean)),]

```

```

print("Winter small grain cereals")
wls=as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(3,3.5,by=0.01), time=t,fert=c("low"),
wls=rbind(wls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(3.5,4,by=0.01), time=t,fert=c("low"),
wls=rbind(wls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(4.,4.5,by=0.01), time=t,fert=c("low"),
wls=rbind(wls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(4.5,5,by=0.01), time=t,fert=c("low"),
wls=rbind(wls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(5,5.5,by=0.01), time=t,fert=c("low"),
wls=rbind(wls,as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=seq(5.5,6,by=0.01), time=t,fert=c("low"),
wmax=wls[which(wls$lsmean==max(wls$lsmean)),]

```

#extracting lsmeans estimate for the baseline monoculture (sref,mref,wref), and for high-input monoculture

```

sref=as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=1, time=0,fert=c("low")), adjust="TUKI"),
shigh=as.data.frame(lsmeans(sXf, pairwise~D*time+D*fert, at=list(D=1, time=t,fert=c("high")), adjust="TUKI"),

```

```

mref=as.data.frame(lsmeans(mXf, pairwise~D*time+D*fert, at=list(D=1, time=0,fert=c("low")), adjust="TUKI"),
mhigh=as.data.frame(lsmeans(mXf, pairwise~D*time+D*fert, at=list(D=1, time=t,fert=c("high")), adjust="TUKI"),

```

```

wref=as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=1, time=0,fert=c("low")), adjust="TUKI"),
whigh=as.data.frame(lsmeans(wXf, pairwise~D*time+D*fert, at=list(D=1, time=t,fert=c("high")), adjust="TUKI"),

```

#Dataset assemblage####

```

stest=rbind(
  cbind(
    "SSGC",
    smax[,c("D","time","fert")],
    smax[,c("lsmean","asympt.LCL","asympt.UCL")] - sref[, "lsmean"] #Each reported estimate is compared with
  ),
  cbind(
    "SSGC",
    hsmax[,c("D","time","fert")],

```

```

    hsmax[,c("lsmean","asyp.LCL","asyp.UCL")]-sref[, "lsmean"]
  ),
  cbind("SSGC",
    shigh[,c("D","time","fert")],
    shigh[,c("lsmean","asyp.LCL","asyp.UCL")]-sref[, "lsmean"]
  )
)
colnames(stest)[1]="ind"

mtest=rbind(
  cbind(
    "Maize",
    mmax[,c("D","time","fert")],
    mmax[,c("lsmean","asyp.LCL","asyp.UCL")]-mref[, "lsmean"]
  ),
  cbind(
    "Maize",
    hmax[,c("D","time","fert")],
    hmax[,c("lsmean","asyp.LCL","asyp.UCL")]-mref[, "lsmean"]
  ),
  cbind(
    "Maize",
    mhigh[,c("D","time","fert")],
    mhigh[,c("lsmean","asyp.LCL","asyp.UCL")]-mref[, "lsmean"]
  )
)
colnames(mtest)[1]="ind"

wtest=rbind(
  cbind(
    "WSGC",
    wmax[,c("D","time","fert")],
    wmax[,c("lsmean","asyp.LCL","asyp.UCL")]-wref[, "lsmean"]
  ),
  cbind(
    "WSGC",
    hwmax[,c("D","time","fert")],
    hwmax[,c("lsmean","asyp.LCL","asyp.UCL")]-wref[, "lsmean"]
  ),
  cbind(
    "WSGC",
    whigh[,c("D","time","fert")],
    whigh[,c("lsmean","asyp.LCL","asyp.UCL")]-wref[, "lsmean"]
  )
)
colnames(wtest)[1]="ind"

if (t==5)
{
finaltest=rbind(stest,mtest,wtest)
}
else
{finaltest=rbind(finaltest,stest,mtest,wtest)}

```

```
}
```

```
## [1] "Spring small grain cereals"
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)  
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];  
## but be warned that this may result in large computation time and memory use.
```

```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Maize"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Winter small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Spring small grain cereals"

```



```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Maize"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Winter small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

```



```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Spring small grain cereals"

```



```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Maize"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Winter small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Spring small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```

## [1] "Maize"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Winter small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

```



```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Spring small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

```



```

## [1] "Maize"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Winter small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Spring small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Maize"

```

```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Winter small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

```



```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
colnames(finaltest)[2]="CRD_val"
fert_est=cbind(finaltest,"SD",c("CRD", "CRD_Fert","Fertilisation"))
colnames(fert_est)[8]="CRD"
colnames(fert_est)[9]="effect.of"
```

```
###Preparing required data subset - Functional Richness
```

#Repeating the previous procedure for functional richness

```
for (t in c(5,20,35))
{
print("Spring small grain cereals")
sls=as.data.frame(lsmmeans(sXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types=c("1","2","3","4"), time=t, fert=fert),
hsls=as.data.frame(lsmmeans(sXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types=c("1","2","3","4"), time=t, fert=fert),

smax=sls[which(sls$lsmmean==max(sls$lsmmean)),]
hsmax=hsls[which(hsls$lsmmean==max(hsls$lsmmean)),]

print("Maize")
mls=as.data.frame(lsmmeans(mXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types=c("1","2","3"), time=t, fert=fert),
hmls=as.data.frame(lsmmeans(mXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types=c("1","2","3"), time=t, fert=fert),
mmax=mls[which(mls$lsmmean==max(mls$lsmmean)),]
hmmax=hmls[which(hmls$lsmmean==max(hmls$lsmmean)),]

print("Winter small grain cereals")
wls=as.data.frame(lsmmeans(wXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types=c("1","2","3"), time=t, fert=fert),
hwls=as.data.frame(lsmmeans(wXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types=c("1","2","3"), time=t, fert=fert),

wmax=wls[which(wls$lsmmean==max(wls$lsmmean)),]
hwmax=hwls[which(hwls$lsmmean==max(hwls$lsmmean)),]

sref=as.data.frame(lsmmeans(sXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types="1", time=0, fert=fert),
shigh=as.data.frame(lsmmeans(sXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types="1", time=t, fert=fert),

mref=as.data.frame(lsmmeans(mXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types="1", time=0, fert=fert),
mhigh=as.data.frame(lsmmeans(mXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types="1", time=t, fert=fert),

wref=as.data.frame(lsmmeans(wXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types="1", time=0, fert=fert),
whigh=as.data.frame(lsmmeans(wXf_f, pairwise~F_types*time+F_types*fert, at=list(F_types="1", time=t, fert=fert),

stest=rbind(
  cbind(
    "SSGC",
    smax[,c("F_types","time","fert")],
    smax[,c("lsmmean","asyp.LCL","asyp.UCL")]-sref[, "lsmmean"]
  ),
  cbind(
    "SSGC",
    hsmax[,c("F_types","time","fert")],
    hsmax[,c("lsmmean","asyp.LCL","asyp.UCL")]-sref[, "lsmmean"]
  ),
  cbind("SSGC",
    shigh[,c("F_types","time","fert")],
    shigh[,c("lsmmean","asyp.LCL","asyp.UCL")]-sref[, "lsmmean"]
  )
)
colnames(stest)[1]="ind"

mtest=rbind(
  cbind(
```

```

"Maize",
  mmax[,c("F_types","time","fert")],
  mmax[,c("lsmean","asyp.LCL","asyp.UCL")]-mref[,"lsmean"]
),
cbind(
  "Maize",
  hmmax[,c("F_types","time","fert")],
  hmmax[,c("lsmean","asyp.LCL","asyp.UCL")]-mref[,"lsmean"]
),
cbind(
  "Maize",
  mhigh[,c("F_types","time","fert")],
  mhigh[,c("lsmean","asyp.LCL","asyp.UCL")]-mref[,"lsmean"]
)
)
colnames(mtest)[1]="ind"

wtest=rbind(
  cbind(
    "WSGC",
    wmax[,c("F_types","time","fert")],
    wmax[,c("lsmean","asyp.LCL","asyp.UCL")]-wref[,"lsmean"]
  ),
  cbind(
    "WSGC",
    hwmax[,c("F_types","time","fert")],
    hwmax[,c("lsmean","asyp.LCL","asyp.UCL")]-wref[,"lsmean"]
  ),
  cbind(
    "WSGC",
    whigh[,c("F_types","time","fert")],
    whigh[,c("lsmean","asyp.LCL","asyp.UCL")]-wref[,"lsmean"]
  )
)
colnames(wtest)[1]="ind"

if (t==5)
{
finaltest=rbind(stest,mtest,wtest)
}
else
{finaltest=rbind(finaltest,stest,mtest,wtest)}
}

```

```
## [1] "Spring small grain cereals"
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
```

```

## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Maize"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Winter small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Spring small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Maize"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```

## [1] "Winter small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## [1] "Spring small grain cereals"
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.
```

```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.
```



```

## [1] "Maize"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## [1] "Winter small grain cereals"

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 8429' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 8429)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 11833' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 11833)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 7198' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 7198)' or larger];
## but be warned that this may result in large computation time and memory use.

```

```

colnames(finaltest)[2]="CRD_val"

FR_Fert_est=cbind(finaltest,"FR",c("CRD", "CRD_Fert","Fertilisation"))
colnames(FR_Fert_est)[8]="CRD"
colnames(FR_Fert_est)[9]="effect.of"

###Producing the graph

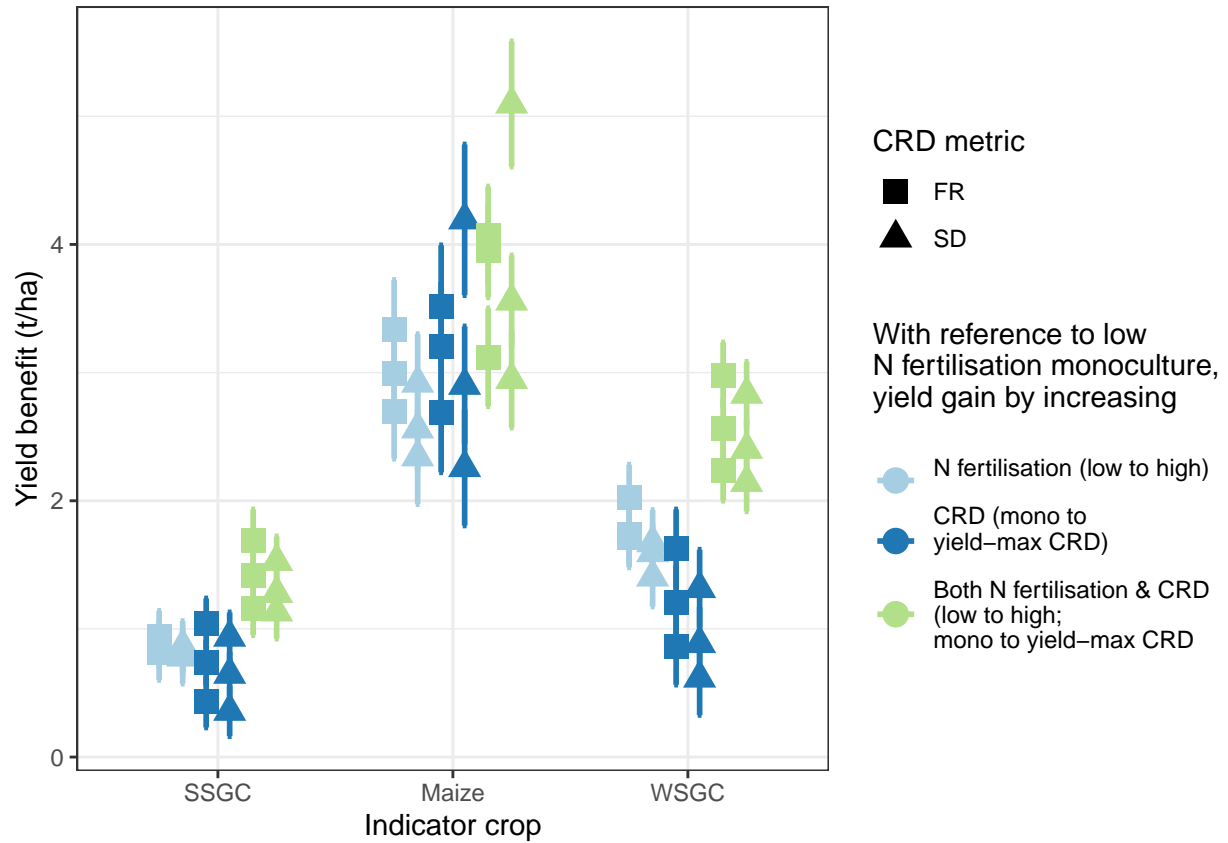
est.df.all=rbind(fert_est,FR_Fert_est) #combining SD and FR subsets
est.df.all$ind <- factor(est.df.all$ind, levels=c("SSGC", "Maize", "WSGC"))
est.df.all$effect.of <- factor(est.df.all$effect.of, levels=c("Fertilisation", "CRD", "CRD_Fert"))

pd <- position_dodge(0.6)

plot.est.all <- ggplot(est.df.all, aes(x=ind, y=lsmean, colour=effect.of, shape = CRD)) +
  geom_errorbar(aes(ymin=asyp.LCL, ymax=asyp.UCL), width=.1, position=pd, size = 1) +
  geom_point(position=pd, size = 4) +
  theme_bw() +
  scale_color_manual(name = "With reference to low\nN fertilisation monoculture,\nyield gain by increas",
                    values=c('#a6cee3','#1f78b4','#b2df8a'),
                    labels=c("N fertilisation (low to high)\n ",
                              "CRD (mono to\nyield-max CRD)\n ",
                              "Both N fertilisation & CRD\n(low to high;\nmono to yield-max CRD"
                              ))+
  scale_shape_manual(name = "CRD metric",
                    values=c(15, 17))+
  labs(x = "Indicator crop",
       y = "Yield benefit (t/ha)")

plot.est.all

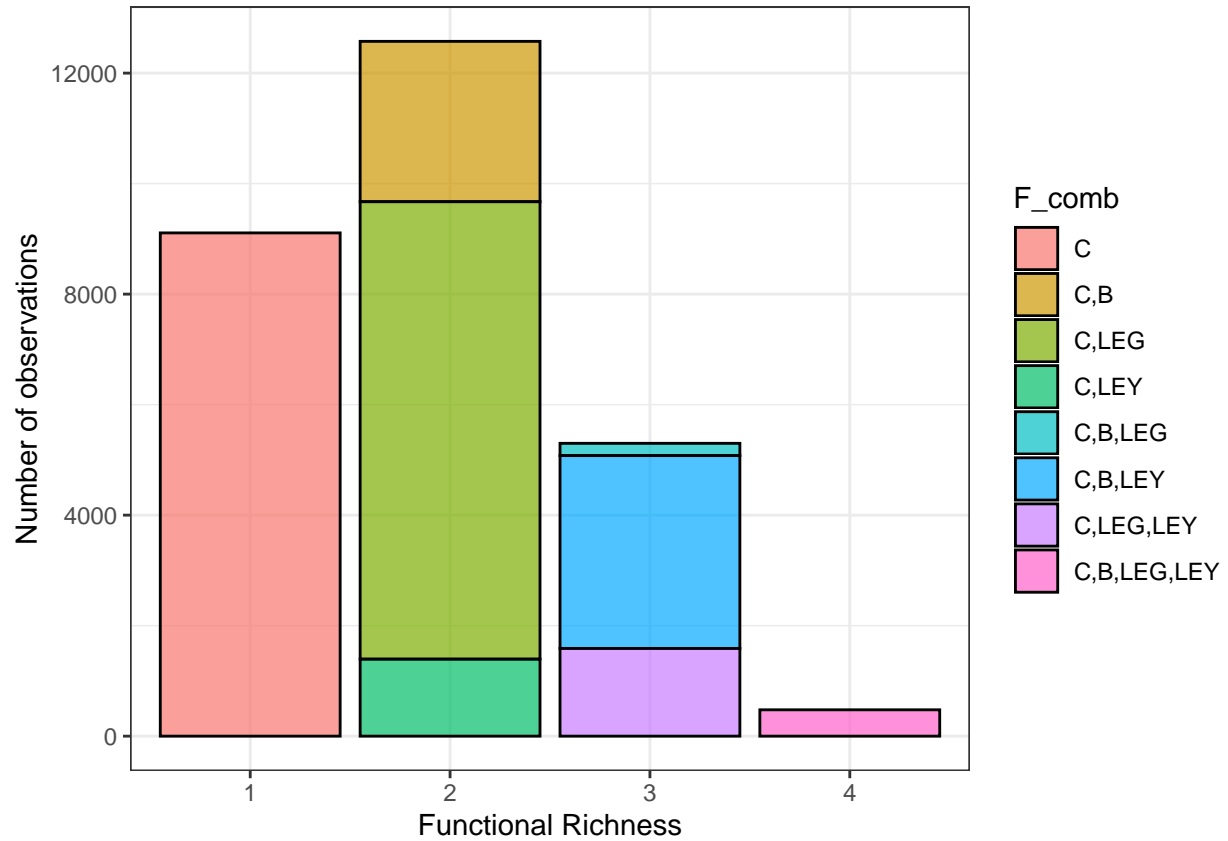
```



Extended data figures

##Fig S1

```
yield.df$F_comb <- factor(yield.df$F_comb, levels = c("C", "C,B", "C,LEG", "C,LEY", "C,B,LEG", "C,B,LEY"))
ggplot(data=yield.df, aes(x = F_types, fill=F_comb))+
  geom_bar(col=I("black"),
           alpha=0.7) +
  labs(x="Functional Richness", y="Number of observations")+
  theme_bw()
```



##Fig S2

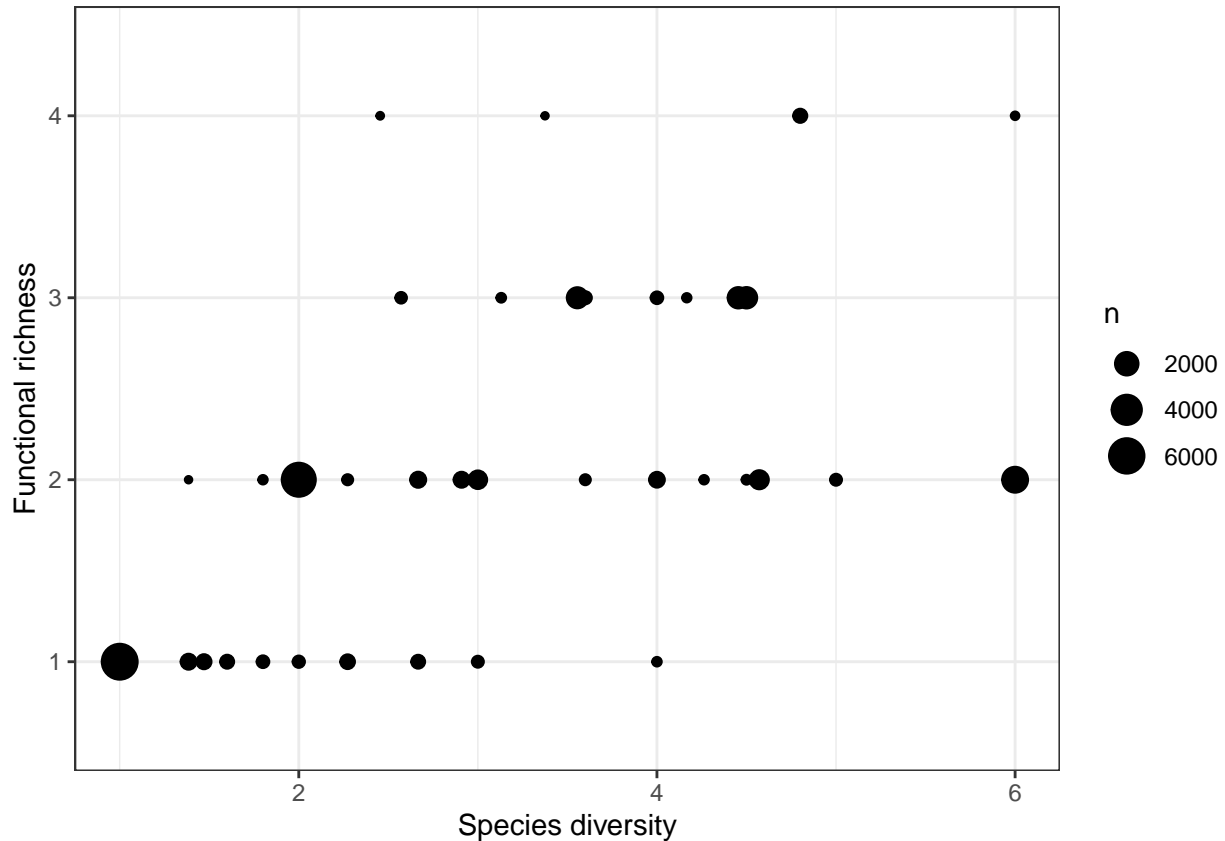
Figure S2 - FR vs SD

```

diversity_summary_plot1=ggplot(yield.df, aes(x = D, y = F_types)) +
  #geom_point()+
  geom_count() +
  theme_bw() +
  labs(x = "Species diversity",
       y = "Functional richness",
       #title = "Species diversity vs functional richness"
  )

```

diversity_summary_plot1



##Fig S3

```
# create the data needed
# winter small grain cereals
Winter.df=Winter.df %>%
  mutate(t_s=case_when( time<=quantile(time,.25) ~ "<=9 yr",
                        time>=quantile(time,.75) ~ ">=28 yr",
                        time<quantile(time,.75)&time>quantile(time,.25) ~ "9 yr<time<28 yr"))
Winter.df$t_s <- ordered(Winter.df$t_s, levels = c("<=9 yr", "9 yr<time<28 yr", ">=28 yr"))

Yield_means_fit_winter <- Winter.df %>%
  group_by(lte, fert, D, t_s) %>%
  dplyr::summarize(meanY=mean(s.yield),
                  sdY = sd(s.yield))
```

'summarise()' has grouped output by 'lte', 'fert', 'D'. You can override using ## the '.groups' argument.

```
Winter.effect1 <- ggeffect(wXf, terms = c("D[1:6, by=0.1]", "time [5,20,35]","fert"))
Winter.effect1$t_s <- ifelse(Winter.effect1$group== '5', "<=9 yr",
                           ifelse(Winter.effect1$group== '20', "9 yr<time<28 yr", ">=28 yr"))
Winter.effect1$t_s <- ordered(Winter.effect1$t_s, levels = c("<=9 yr", "9 yr<time<28 yr", ">=28 yr"))
Winter.effect1$s.yield <- Winter.effect1$predicted
Winter.effect1$D <- Winter.effect1$x
Winter.effect1$fert <- Winter.effect1$facet
```

```

#spring small grain cereals
Spring.df.EU=Spring.df.EU %>%
  mutate(t_s=case_when( time<=quantile(time,.25) ~ "<=9 yr",
                        time>=quantile(time,.75) ~ ">=28 yr",
                        time<quantile(time,.75)&time>quantile(time,.25) ~ "9 yr<time<28 yr"))
Spring.df.EU$t_s <- ordered(Spring.df.EU$t_s, levels = c("<=9 yr", "9 yr<time<28 yr", ">=28 yr"))

Yield_means_fit_spring <- Spring.df.EU %>%
  group_by(lte, fert, D, t_s) %>%
  dplyr::summarize(meanY=mean(s.yield),
                  sdY = sd(s.yield))

```

'summarise()' has grouped output by 'lte', 'fert', 'D'. You can override using
the '.groups' argument.

```

Spring.EU.effect1 <- ggeffect(sXf, terms = c("D[1:6, by=0.1]", "time [5,20,35]", "fert"))
Spring.EU.effect1$t_s <- ifelse(Spring.EU.effect1$group== '5', "<=9 yr",
                              ifelse(Spring.EU.effect1$group== '20', "9 yr<time<28 yr", ">=28 yr"))
Spring.EU.effect1$t_s <- ordered(Spring.EU.effect1$t_s, levels = c("<=9 yr", "9 yr<time<28 yr", ">=28 yr"))
Spring.EU.effect1$s.yield <- Spring.EU.effect1$predicted
Spring.EU.effect1$D <- Spring.EU.effect1$x
Spring.EU.effect1$fert <- Spring.EU.effect1$facet

```

maize

```

Spring.df.US=Spring.df.US %>%
  mutate(t_s=case_when( time<=quantile(time,.25) ~ "<=9 yr",
                        time>=quantile(time,.75) ~ ">=28 yr",
                        time<quantile(time,.75)&time>quantile(time,.25) ~ "9 yr<time<28 yr"))
Spring.df.US$t_s <- ordered(Spring.df.US$t_s, levels = c("<=9 yr", "9 yr<time<28 yr", ">=28 yr"))

Yield_means_fit_maize <- Spring.df.US %>%
  group_by(lte, fert, D, t_s) %>%
  dplyr::summarize(meanY=mean(s.yield),
                  sdY = sd(s.yield))

```

'summarise()' has grouped output by 'lte', 'fert', 'D'. You can override using
the '.groups' argument.

```

Spring.US.effect1 <- ggeffect(mXf, terms = c("D[1:4.6, by=0.1]", "time [5,20,35]", "fert"))
Spring.US.effect1$t_s <- ifelse(Spring.US.effect1$group== '5', "<=9 yr",
                              ifelse(Spring.US.effect1$group== '20', "9 yr<time<28 yr", ">=28 yr"))
Spring.US.effect1$t_s <- ordered(Spring.US.effect1$t_s, levels = c("<=9 yr", "9 yr<time<28 yr", ">=28 yr"))
Spring.US.effect1$s.yield <- Spring.US.effect1$predicted
Spring.US.effect1$D <- Spring.US.effect1$x
Spring.US.effect1$fert <- Spring.US.effect1$facet

```

set up plots

```

fert_time_labeller <- as_labeller(c("low" = "Low fertilisation", #Needed for fertilisation facet label
                                   "high" = "High fertilisation",
                                   "<=9 yr"="<=9 yr",
                                   "9 yr<time<28 yr"="9 yr< time <28 yr",

```

```

    ">=28 yr"=">=28 yr"
  ))

pd <- position_dodge(0.2)

# make plots
winter.sites3 <- ggplot(Yield_means_fit_winter, aes(x = D, y=meanY)) +
  geom_jitter(data=Winter.df,aes(D, s.yield,col=lte,group=lte),alpha = 0.3, size=1,width = 0.05, show.legend = F) +
  geom_errorbar(aes(ymin=meanY-sdY, ymax=meanY+sdY,group=lte), width=.1, position=pd) +
  geom_point(aes(fill=lte), position=pd, size=2, shape=21) +
  geom_line(data = Winter.effect1, aes(x=D, y=s.yield), colour="black", show.legend = F) +
  # geom_ribbon(data = Spring.US.effect1, aes(ymin = conf.low, ymax = conf.high),
  #           show.legend=FALSE, alpha = .5) +
  facet_grid(t_s~factor(fert,levels=c("low","high")), labeller = fert_time_labeller) +
  theme_bw() +
  xlim(1,6)+
  labs(fill = "",
       x = "Species Diversity",
       y = "Standardised yield (t/ha)",
       title = "Winter small grain cereals"
  )+guides(fill = guide_legend(override.aes = list(shape = 21, size=3)))

spring.sites3 <- ggplot(Yield_means_fit_spring, aes(x = D, y=meanY)) +
  geom_jitter(data=Spring.df.EU,aes(D, s.yield,col=lte,group=lte),alpha = 0.3, size=1,width = 0.05, show.legend = F) +
  geom_errorbar(aes(ymin=meanY-sdY, ymax=meanY+sdY,group=lte), width=.1, position=pd) +
  geom_point(aes(fill=lte), position=pd, size=2, shape=21) +
  geom_line(data = Spring.EU.effect1, aes(x=D, y=s.yield), colour="black", show.legend = F) +
  # geom_ribbon(data = Spring.US.effect1, aes(ymin = conf.low, ymax = conf.high),
  #           show.legend=FALSE, alpha = .5) +
  facet_grid(t_s~factor(fert,levels=c("low","high")), labeller = fert_time_labeller) +
  theme_bw() +
  xlim(1,6)+
  labs(fill = "",
       x = "Species Diversity",
       y = "Standardised yield (t/ha)",
       title = "Spring small grain cereals"
  )+guides(fill = guide_legend(override.aes = list(shape = 21, size=3)))

maize.sites4 <- ggplot(Yield_means_fit_maize, aes(x = D, y=meanY)) +
  geom_jitter(data=Spring.df.US,aes(D, s.yield,col=lte,group=lte),alpha = 0.3, size=1,width = 0.05, show.legend = F) +
  geom_errorbar(aes(ymin=meanY-sdY, ymax=meanY+sdY,group=lte), width=.1, position=pd) +
  geom_point(aes(fill=lte), position=pd, size=2, shape=21) +
  geom_line(data = Spring.US.effect1, aes(x=D, y=s.yield), colour="black", show.legend = F) +
  # geom_ribbon(data = Spring.US.effect1, aes(ymin = conf.low, ymax = conf.high),
  #           show.legend=FALSE, alpha = .5) +
  facet_grid(t_s~factor(fert,levels=c("low","high")), labeller = fert_time_labeller) +
  theme_bw() +
  xlim(1,6)+
  labs(fill = "",
       x = "Species Diversity",
       y = "Standardised yield (t/ha)",
       title = "Maize"
  )+guides(fill = guide_legend(override.aes = list(shape = 21, size=3)))

```



```

# combine plots
# ggarrange(spring.sites3, maize.sites4, winter.sites3,
#           ncol = 3, legend = "top")

combined_fig <- ggarrange(spring.sites3 + rremove("ylab") + rremove("xlab"),
                          maize.sites4 + rremove("ylab") + rremove("xlab"),
                          winter.sites3 + rremove("ylab") + rremove("xlab"),
                          labels = NULL,
                          ncol = 3, legend = "top")

```

```

## Warning: 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals

## Warning: Removed 2330 rows containing missing values ('geom_point()').

## Warning: Removed 24 rows containing missing values ('geom_point()').

## Warning: 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals

## Warning: Removed 972 rows containing missing values ('geom_point()').

## Warning: Removed 11 rows containing missing values ('geom_point()').

## Warning: 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals

```

```
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
## 'position_dodge()' requires non-overlapping x intervals
```

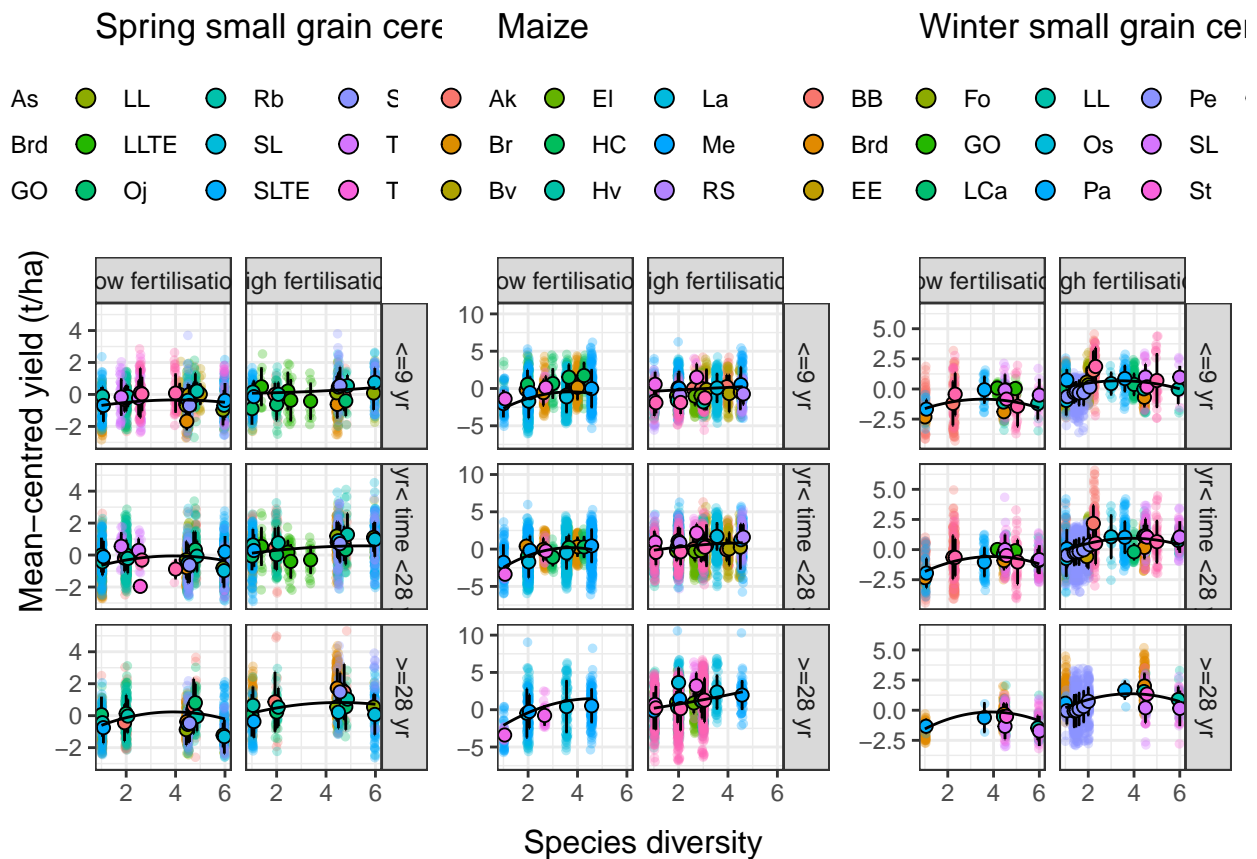
```
## Warning: Removed 1127 rows containing missing values ('geom_point()').
```

```
## Warning: Removed 18 rows containing missing values ('geom_point()').
```

```
require(grid) # for the textGrob() function
```

```
## Loading required package: grid
```

```
annotate_figure(combined_fig, left = textGrob("Mean-centred yield (t/ha)", rot = 90, vjust = 1, gp = gpar(cex = 1)),
                bottom = textGrob("Species diversity", gp = gpar(cex = 1)))
```



Supplementary Tables

```
##Tab S4 and S5
```

```
#Spring small grain cereals, species diversity
model_parameters(sXf, effects="fixed", summary=T)
```

```

## # Fixed Effects
##
## Parameter          | Coefficient |      SE |      95% CI | t(8417) |      p
## -----
## (Intercept)       |      -0.90 |    0.12 | [-1.14, -0.65] |   -7.22 | < .001
## D                  |       0.22 |    0.05 | [ 0.12, 0.32] |    4.40 | < .001
## time              |      -0.01 | 3.45e-03 | [-0.02, 0.00] |   -3.21 | 0.001
## D^2               |      -0.03 | 7.42e-03 | [-0.04, -0.01] |   -3.65 | < .001
## fert [high]       |       1.02 |    0.10 | [ 0.82, 1.22] |   10.11 | < .001
## D × time          |       0.02 | 1.66e-03 | [ 0.01, 0.02] |    9.57 | < .001
## time × D^2        |     -2.08e-03 | 2.52e-04 | [ 0.00, 0.00] |   -8.25 | < .001
## D × fert [high]   |      -0.30 |    0.05 | [-0.40, -0.20] |   -5.96 | < .001
## D^2 × fert [high] |       0.05 | 7.37e-03 | [ 0.03, 0.06] |    6.40 | < .001
##
## Model: s.yield ~ D + time + I(D^2) + fert + D:time + time:I(D^2) + D:fert + I(D^2):fert (8429 Observations)
## Residual standard deviation: 0.649 (df = 8417)
## Conditional R2: 0.719; Marginal R2: 0.154

##
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.

```

```

#Spring small grain cereals, functional richness
model_parameters(sXf_f, effects="fixed", summary=T)

```

```

## # Fixed Effects
##
## Parameter          | Coefficient |      SE |      95% CI | t(8414) |      p
## -----
## (Intercept)       |      -0.75 |    0.11 | [-0.96, -0.53] |   -6.76 | < .001
## F types [2]       |       0.13 |    0.04 | [ 0.06, 0.21] |    3.43 | < .001
## F types [3]       |       0.33 |    0.05 | [ 0.23, 0.42] |    6.74 | < .001
## F types [4]       |       0.21 |    0.07 | [ 0.06, 0.36] |    2.82 | 0.005
## time              |     4.20e-03 | 2.94e-03 | [ 0.00, 0.01] |    1.43 | 0.154
## fert [high]       |       0.79 |    0.08 | [ 0.64, 0.95] |    9.88 | < .001
## F types [2] × time |     3.53e-03 | 1.41e-03 | [ 0.00, 0.01] |    2.51 | 0.012
## F types [3] × time |       0.02 | 1.42e-03 | [ 0.01, 0.02] |   11.41 | < .001
## F types [4] × time |       0.01 | 2.90e-03 | [ 0.01, 0.02] |    4.89 | < .001
## F types [2] × fert [high] |     0.19 |    0.04 | [ 0.12, 0.27] |    5.03 | < .001
## F types [3] × fert [high] |     -0.22 |    0.05 | [-0.32, -0.13] |   -4.67 | < .001
## F types [4] × fert [high] |     0.05 |    0.08 | [-0.11, 0.20] |    0.59 | 0.554
##
## Model: s.yield ~ F_types + time + fert + F_types:time + F_types:fert (8429 Observations)
## Residual standard deviation: 0.647 (df = 8414)
## Conditional R2: 0.730; Marginal R2: 0.173

##
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.

```

```

#Maize, species diversity
model_parameters(mXf, effects="fixed", summary=T)

```

```

## # Fixed Effects
##
## Parameter          | Coefficient | SE | 95% CI | t(11818) | p
## -----
## (Intercept)       | -4.43 | 0.35 | [-5.11, -3.75] | -12.83 | < .001
## D                  | 1.97 | 0.17 | [ 1.64, 2.31] | 11.59 | < .001
## time              | -0.04 | 0.03 | [-0.09, 0.01] | -1.45 | 0.146
## D^2               | -0.25 | 0.03 | [-0.31, -0.19] | -8.26 | < .001
## time^2            | 1.22e-03 | 5.60e-04 | [ 0.00, 0.00] | 2.19 | 0.029
## fert [high]       | 4.05 | 0.29 | [ 3.48, 4.62] | 13.96 | < .001
## D × time          | 0.06 | 0.01 | [ 0.03, 0.09] | 3.93 | < .001
## time × D^2        | -0.01 | 2.87e-03 | [-0.02, -0.01] | -3.92 | < .001
## D × time^2         | -1.22e-03 | 3.43e-04 | [ 0.00, 0.00] | -3.56 | < .001
## D^2 × time^2       | 3.07e-04 | 7.37e-05 | [ 0.00, 0.00] | 4.17 | < .001
## D × fert [high]   | -2.05 | 0.14 | [-2.33, -1.77] | -14.22 | < .001
## D^2 × fert [high] | 0.30 | 0.03 | [ 0.25, 0.35] | 11.67 | < .001
##
## Model: s.yield ~ D * time + I(D^2) * time + D * I(time^2) + I(D^2) * I(time^2) + fert + D:fert + I(D
## Residual standard deviation: 1.084 (df = 11818)
## Conditional R2: 0.771; Marginal R2: 0.128

##
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.

```

```

#Maize, functional richness
model_parameters(mXf_f, effects="fixed", summary=T)

```

```

## # Fixed Effects
##
## Parameter          | Coefficient | SE | 95% CI | t(11818) | p
## -----
## (Intercept)       | -2.98 | 0.29 | [-3.55, -2.42] | -10.35 | < .001
## F types [2]       | 1.97 | 0.08 | [ 1.80, 2.13] | 23.19 | < .001
## F types [3]       | 1.90 | 0.13 | [ 1.64, 2.17] | 14.16 | < .001
## time              | 0.02 | 0.02 | [-0.03, 0.06] | 0.73 | 0.463
## time^2            | 1.26e-04 | 4.43e-04 | [ 0.00, 0.00] | 0.28 | 0.776
## fert [high]       | 2.61 | 0.24 | [ 2.13, 3.09] | 10.69 | < .001
## F types [2] × time | 7.22e-03 | 6.83e-03 | [-0.01, 0.02] | 1.06 | 0.291
## F types [3] × time | 0.17 | 0.02 | [ 0.13, 0.20] | 8.53 | < .001
## F types [2] × time^2 | 2.07e-04 | 1.43e-04 | [ 0.00, 0.00] | 1.45 | 0.148
## F types [3] × time^2 | -5.20e-03 | 7.11e-04 | [-0.01, 0.00] | -7.31 | < .001
## F types [2] × fert [high] | -1.74 | 0.07 | [-1.88, -1.60] | -24.95 | < .001
## F types [3] × fert [high] | -2.18 | 0.10 | [-2.38, -1.98] | -21.50 | < .001
##
## Model: s.yield ~ F_types * time + F_types * I(time^2) + fert + F_types:fert (11833 Observations)
## Residual standard deviation: 1.100 (df = 11818)
## Conditional R2: 0.766; Marginal R2: 0.125

##
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.

```

```
#Winter small grain cereals, species diversity
model_parameters(wXf, effects="fixed", summary=T)
```

```
## # Fixed Effects
```

```
##
```

| ## Parameter | Coefficient | SE | 95% CI | t(7184) | p |
|---------------------------------|-------------|----------|----------------|---------|--------|
| ## (Intercept) | -2.09 | 0.19 | [-2.46, -1.72] | -11.04 | < .001 |
| ## D | 0.74 | 0.07 | [0.61, 0.88] | 10.96 | < .001 |
| ## time | -0.06 | 0.01 | [-0.08, -0.04] | -4.95 | < .001 |
| ## D ² | -0.11 | 0.01 | [-0.14, -0.09] | -10.39 | < .001 |
| ## time ² | 1.26e-03 | 2.28e-04 | [0.00, 0.00] | 5.52 | < .001 |
| ## fert [high] | 1.94 | 0.17 | [1.60, 2.28] | 11.10 | < .001 |
| ## D × time | 0.02 | 2.97e-03 | [0.02, 0.03] | 8.17 | < .001 |
| ## time × D ² | -1.50e-03 | 4.04e-04 | [0.00, 0.00] | -3.71 | < .001 |
| ## D × time ² | -2.40e-04 | 2.81e-05 | [0.00, 0.00] | -8.52 | < .001 |
| ## D × fert [high] | -0.20 | 0.06 | [-0.32, -0.08] | -3.27 | 0.001 |
| ## D ² × fert [high] | 0.02 | 0.01 | [0.00, 0.04] | 2.33 | 0.020 |

```
##
```

```
## Model: s.yield ~ D + time + I(D2) + I(time2) + fert + D:time + time:I(D2) + D:I(time2) + D:fert
```

```
## Residual standard deviation: 0.714 (df = 7184)
```

```
## Conditional R2: 0.775; Marginal R2: 0.295
```

```
##
```

```
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
```

```
## using a Wald t-distribution approximation.
```

```
#Winter small grain cereals, functional richness
```

```
model_parameters(wXf_f, effects="fixed", summary=T)
```

```
## # Fixed Effects
```

```
##
```

| ## Parameter | Coefficient | SE | 95% CI | t(7183) | p |
|------------------------------------|-------------|----------|----------------|---------|--------|
| ## (Intercept) | -1.66 | 0.17 | [-2.00, -1.32] | -9.49 | < .001 |
| ## F types [2] | 0.60 | 0.11 | [0.39, 0.81] | 5.70 | < .001 |
| ## F types [3] | 0.76 | 0.08 | [0.60, 0.93] | 8.99 | < .001 |
| ## time | -0.02 | 0.01 | [-0.04, 0.00] | -1.83 | 0.068 |
| ## time ² | 7.59e-04 | 2.16e-04 | [0.00, 0.00] | 3.51 | < .001 |
| ## fert [high] | 1.82 | 0.17 | [1.49, 2.14] | 10.98 | < .001 |
| ## F types [2] × time | 0.02 | 8.59e-03 | [0.00, 0.03] | 1.81 | 0.071 |
| ## F types [3] × time | 0.04 | 5.77e-03 | [0.03, 0.05] | 6.83 | < .001 |
| ## F types [2] × time ² | -3.52e-04 | 2.01e-04 | [0.00, 0.00] | -1.76 | 0.079 |
| ## F types [3] × time ² | -5.95e-04 | 9.65e-05 | [0.00, 0.00] | -6.17 | < .001 |
| ## F types [2] × fert [high] | -0.17 | 0.09 | [-0.34, 0.00] | -1.99 | 0.046 |
| ## F types [3] × fert [high] | -0.46 | 0.06 | [-0.59, -0.34] | -7.19 | < .001 |

```
##
```

```
## Model: s.yield ~ F_types * time + F_types * I(time2) + fert + F_types:fert (7198 Observations)
```

```
## Residual standard deviation: 0.722 (df = 7183)
```

```
## Conditional R2: 0.776; Marginal R2: 0.303
```

```
##
```

```
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.
```

```
#Tab S5
```

```
mod.winter<- lmer(s.yield ~ ley+leg+bl+
                  (1|site:group) + (1|year.f), data=Winter.df)
mod.spring<- lmer(s.yield ~ ley+leg+bl+
                  (1|site:group) + (1|year.f), data=Spring.df.EU)
mod.maize<- lmer(s.yield ~ ley+leg+bl+
                 (1|site:group) + (1|year.f), data=Spring.df.US)
```

```
model_parameters(mod.spring, effects="fixed", summary=T)
```

```
## # Fixed Effects
```

```
##
## Parameter | Coefficient | SE | 95% CI | t(8422) | p
## -----|-----|-----|-----|-----|-----
## (Intercept) | -0.41 | 0.13 | [-0.66, -0.16] | -3.20 | 0.001
## ley [y] | -0.39 | 0.06 | [-0.51, -0.28] | -6.86 | < .001
## leg [y] | 0.52 | 0.05 | [ 0.42, 0.63] | 9.84 | < .001
## bl [y] | 0.26 | 0.03 | [ 0.21, 0.32] | 9.04 | < .001
##
```

```
## Model: s.yield ~ ley + leg + bl (8429 Observations)
```

```
## Residual standard deviation: 0.913 (df = 8422)
```

```
## Conditional R2: 0.530; Marginal R2: 0.029
```

```
##
```

```
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.
```

```
model_parameters(mod.maize, effects="fixed", summary=T)
```

```
## # Fixed Effects
```

```
##
## Parameter | Coefficient | SE | 95% CI | t(11826) | p
## -----|-----|-----|-----|-----|-----
## (Intercept) | -1.00 | 0.20 | [-1.39, -0.62] | -5.12 | < .001
## ley [y] | 0.43 | 0.07 | [ 0.30, 0.56] | 6.35 | < .001
## leg [y] | 0.74 | 0.04 | [ 0.66, 0.83] | 16.86 | < .001
## bl [y] | 0.10 | 0.12 | [-0.14, 0.33] | 0.81 | 0.417
##
```

```
## Model: s.yield ~ ley + leg + bl (11833 Observations)
```

```
## Residual standard deviation: 1.686 (df = 11826)
```

```
## Conditional R2: 0.469; Marginal R2: 0.023
```

```
##
```

```
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
## using a Wald t-distribution approximation.
```

```
model_parameters(mod.winter, effects="fixed", summary=T)
```

```
## # Fixed Effects
```

```
##
```

```
## Parameter | Coefficient | SE | 95% CI | t(7191) | p
```

```
## -----
```

```
## (Intercept) | -0.73 | 0.14 | [-1.00, -0.46] | -5.22 | < .001
```

```
## ley [y] | 0.10 | 0.06 | [-0.02, 0.23] | 1.63 | 0.103
```

```
## leg [y] | 0.41 | 0.04 | [ 0.33, 0.49] | 9.67 | < .001
```

```
## bl [y] | 0.49 | 0.05 | [ 0.40, 0.59] | 10.28 | < .001
```

```
##
```

```
## Model: s.yield ~ ley + leg + bl (7198 Observations)
```

```
## Residual standard deviation: 1.007 (df = 7191)
```

```
## Conditional R2: 0.599; Marginal R2: 0.057
```

```
##
```

```
## Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed
```

```
## using a Wald t-distribution approximation.
```