multiple stressor

Marcel, Owen, Uriah

02 February, 2022

Contents

Introduction

This markdown contains the full-length 16S rRNA amplicon sequencing analysis of 240 micro-ecosystems, which were treated with 4 drivers in all possible combinations at three different temperatures. The analysis is related to the manuscript "Predicting the effects of multiple global change drivers on microbial communities remains challenging" by Suleiman et al.

Dependencies, data and functions

Dependencies

Check whether all are needed

```
### BIOCONDUCTOR
library(Biostrings)
library(ShortRead)
library(phyloseq)
library(dada2)
library(microbiome)
library(DESeq2)
library(apeglm)

### CRAN
library(pracma)
library(MASS)
library(reshape2)
library(gridExtra)
library(boot)
library(tidyverse)
options(dplyr.summarise.inform = FALSE)
library(lubridate)
library(googlesheets)
library(here) # probably not needed if you open the R-project
library(scales)
library(grid)
library(readxl)
library(ggplot2)
library(dplyr)
library(ggfortify)
library(ggpubr)
library(aod)
```

```r
library(Rcpp)
library(vegan)
library(ggpubr)
library(patchwork)
library(broom)
library(rstatix)
library(plyr)
library(permute)
library(lattice)
library(tm)
library(stringr)
library(viridis)
```

Data

Load phyloseq data, remove chloroplast sequences, remove samples that did not work

```r
ps <- readRDS("phyloseq_multiple_stressor.rds")
taxa_names(ps) <- paste0("Seq", seq(ntaxa(ps)))

ps_new <-  subset_taxa(ps, Order != "Chloroplast")
ps_new_11 <- subset_samples(ps_new,!(Lable %in% c("ori_1","ori_4","B_1_24", "C_1_20","BCD_5_24",
                                                  "CD_1_24","BE_5_24","BCDE_5_24","CD_5_24",
                                                  "BE_5_20", "CE_1_24")))
rm(ps)
```

Functions

```r
# Function for ggplot theme
marcel_theme <- function(){
  theme_bw() %+replace%
    theme(
      panel.grid = element_blank(),
      axis.text=element_text(size=12),
      axis.title=element_text(size=14,face="bold")
    )
}

# Standard error function
stderr <- function(x, na.rm=FALSE) {
  if (na.rm) x <- na.omit(x)
  sqrt(var(x)/length(x))
}

# Function to extract t- anf F-values
extract_test_values <- function(model,variable){
  tidy_tibble_t <- tidy(model)
  names(tidy_tibble_t)[names(tidy_tibble_t) == "statistic"] <- "t_statistic"
  names(tidy_tibble_t)[names(tidy_tibble_t) == "p.value"] <- "t_p.value"
  names(tidy_tibble_t)[names(tidy_tibble_t) == "term"] <- "treatment_combination"
  tidy_tibble_t <- tidy_tibble_t %>%
    mutate(term = removeNumbers(treatment_combination))

  tidy_tibble_f <- tidy(anova(model))
  names(tidy_tibble_f)[names(tidy_tibble_f) == "statistic"] <- "f_statistic"
  names(tidy_tibble_f)[names(tidy_tibble_f) == "p.value"] <- "f_p.value"

  tidy_tibble <- full_join(tidy_tibble_t, tidy_tibble_f, by="term") %>%
    mutate(color = ifelse(f_p.value < 0.05, estimate/max(abs(coef(model))), NA),
           variable=variable) %>%
    filter(term != "(Intercept)", term!="Residuals") %>%
    reorder_levels(treatment_combination,
                   order = c("T24","T28","F","G","M","A","F:T24","G:T24",
                             "M:T24","A:T24","F:T28","G:T28","M:T28","A:T28",
                             "F:G","F:M","G:M","F:A","G:A","M:A","F:G:T24",
                             "F:M:T24","G:M:T24","F:A:T24","G:A:T24","M:A:T24",
                             "F:G:T28","F:M:T28","G:M:T28","F:A:T28","G:A:T28",
                             "M:A:T28","F:G:M","F:G:A","F:M:A","G:M:A",
                             "F:G:M:T24","F:G:A:T24","F:M:A:T24","G:M:A:T24",
                             "F:G:M:T28", "F:G:A:T28","F:M:A:T28","G:M:A:T28",
                             "F:G:M:A","F:G:M:A:T24", "F:G:M:A:T28"))
  # print(c(max(coef(model)),max(abs(coef(model)))))
  return(tidy_tibble)
}
```

Relative abundances

Calculation

The relative abundance is calculated only for sequences that appear > 0.01 %, making dataframes for each column position

```
ps_rel <- transform_sample_counts(ps_new_11, function(x) x / sum(x) )
relab_threshold <- 0.001
ps_relab <- filter_taxa(ps_rel, function(x) !(sum(x < relab_threshold) == length(x)), TRUE)

paste0("The orginal dataset has ",ntaxa(ps_new_11),
       " taxa, while the new one (relative abundances) has ",ntaxa(ps_relab)," taxa")
```

```
## [1] "The orginal dataset has 20319 taxa, while the new one (relative abundances) has 5830 taxa"
```

```
ps_relative <- transform_sample_counts(ps_relab, function(x)  x / sum(x))
```

Figures at family level

Visualizations of relative abundance at family level

Figure: relative abundance as boxplot across treatment combinations and temperatures

```
df_rel <- psmelt(ps_relative)

df_family<- df_rel %>%
  group_by(Family,Lable,Combinations,Temperature_factor) %>%
  dplyr::summarize(Abundance=sum(Abundance))

df_family <- df_family %>%
  reorder_levels(Combinations, order = c("Control","F","G","M","A","FG","FM","FA","GM",
                                          "GA","MA","FGM","FGA","GMA","FMA","FGMA"))

df_family %>%
  filter(Family %in% c("Phormidiaceae","Chromatiaceae"),Combinations != "ori") %>%
  ggplot(aes(x=Combinations, y=Abundance, fill=Family)) +
  geom_boxplot()+
  facet_grid("Temperature_factor") +
  theme_bw()
```

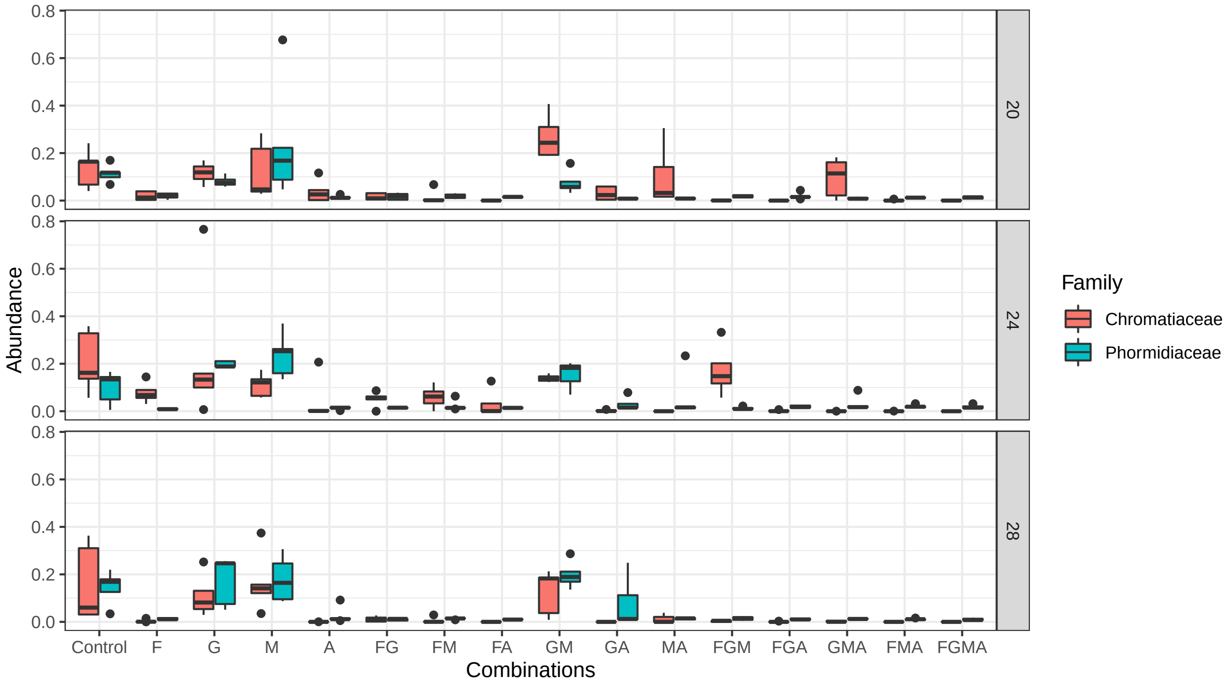


Figure: relative abundance as heatmap across all treatment combinations and temperatures

```
df_family <- df_family %>%
  dplyr::group_by(Family,Combinations,Temperature_factor) %>%
  dplyr::summarize(Abundance_mean = mean(Abundance),
                   Abundance_SE =stderr(Abundance))

index <- which(df_family$Abundance_mean>=0.05)
family_to_keep <- unique(df_family[index,"Family"])
family_to_keep <- unname(unlist(family_to_keep))

df_family$Family_filter <- ifelse(df_family$Family %in% family_to_keep, df_family$Family,"other")

#make other to one
```

```r
df_family<- df_family %>%
  dplyr::group_by(Family_filter,Combinations,Temperature_factor) %>%
  dplyr::summarize(Abundance_sum = sum(Abundance_mean),
                   Abundance_sum_SE =sum(Abundance_SE))

df_family <- df_family %>%
  reorder_levels(Combinations, order =  c("Control","F","G","M","A","FG","FM","FA","GM",
                                          "GA","MA","FGM","FGA","GMA","FMA","FGMA"))

cols <- c("Azospirillaceae"="bisque1", "Bacteroidetes_vadinHA17" = "yellow",
          "Chlorobiaceae"="black","Comamonadaceae" = "green4",
          "Hungateiclostridiaceae" ="maroon1", "Lentimicrobiaceae" ="royalblue1",
          "Marinifilaceae"="lightcyan1","Nocardiaceae" ="thistle1",
          "Phormidiaceae"= "firebrick1", "Prolixibacteraceae" = "magenta4",
          "Pseudomonadaceae" = "grey","Rhodocyclaceae"="green",
          "Rhodospillaceae"="grey38","Rubritaleaceae"="brown",
          "Xanthobacteraceae"="cornflowerblue", "Xanthomonadaceae"="lightskyblue1",
          "Cyanobiaceae"="lavender", "Chromatiaceae" = "lightsalmon")

community_plot <- df_family %>%
  filter(Combinations != "ori", Family_filter !="Mitochondria", Family_filter !="other")%>%
  ggplot(aes(x = Combinations, y = Abundance_sum, fill=Family_filter)) +
  geom_col() +
  facet_wrap("Temperature_factor")+
  theme_bw()+
  theme(axis.text.x = element_text(angle = 50, hjust = 1, size=14),
        axis.title=element_text(size=16,face="bold"),
        legend.text = element_text(size=11)) +
  scale_fill_manual("Family", values = cols) +
  labs(y="rel. abundance family level", x="Driver(s) applied ")

df_family %>%
  filter(Combinations != "ori", Family_filter != "other",
         Family_filter !="Pirellulaceae", Family_filter !="Cyanobiaceae",
         Family_filter !="Mitochondria")%>%
  ggplot(aes(x = Combinations, y = Family_filter, fill=Abundance_sum)) +
  geom_tile() +
  scale_fill_gradient2(low="red", mid="white", high="navy", midpoint=0.15, na.value="lightblue",
                       breaks=c(0.05,0.1,0.15,0.2,0.25), limits=c(0.05,0.25)) +
  facet_wrap("Temperature_factor")+
  theme_bw()+
  theme(axis.text.x = element_text(angle = 50, hjust = 1))
```

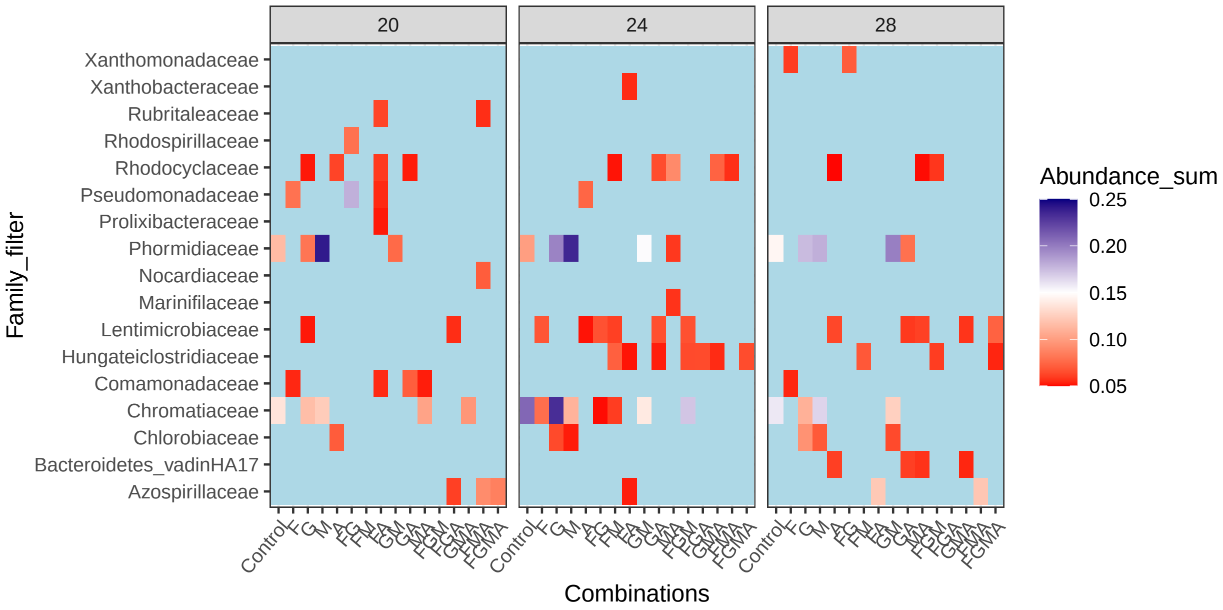


Figure: relative abundance of just Phormidiaceae and chromatiaceae as scatter plot across treatment combinations and temperatures. For supplementary information

```
df_family %>%
  filter(Family_filter %in% c("Phormidiaceae","Chromatiaceae"), Combinations!= "ori") %>%
  ggplot(aes(x=Combinations,y=Abundance_sum,col=Family_filter)) +
  geom_point(aes(fill=Family_filter),position=position_dodge(width = 0.1), size=4)+
  geom_errorbar(aes(ymin=Abundance_sum-Abundance_sum_SE, ymax=Abundance_sum+Abundance_sum_SE),
                width=0.1, alpha=0.4 ,col="black", position=position_dodge(width = .1))+
  facet_wrap("Temperature_factor")+
  theme_bw() +
  theme(axis.text.x = element_text(angle = 50, hjust = 1, size=13)) +
  labs(col = "Family", fill="Family", x="Stressor combination", y="rel. abundance")
```

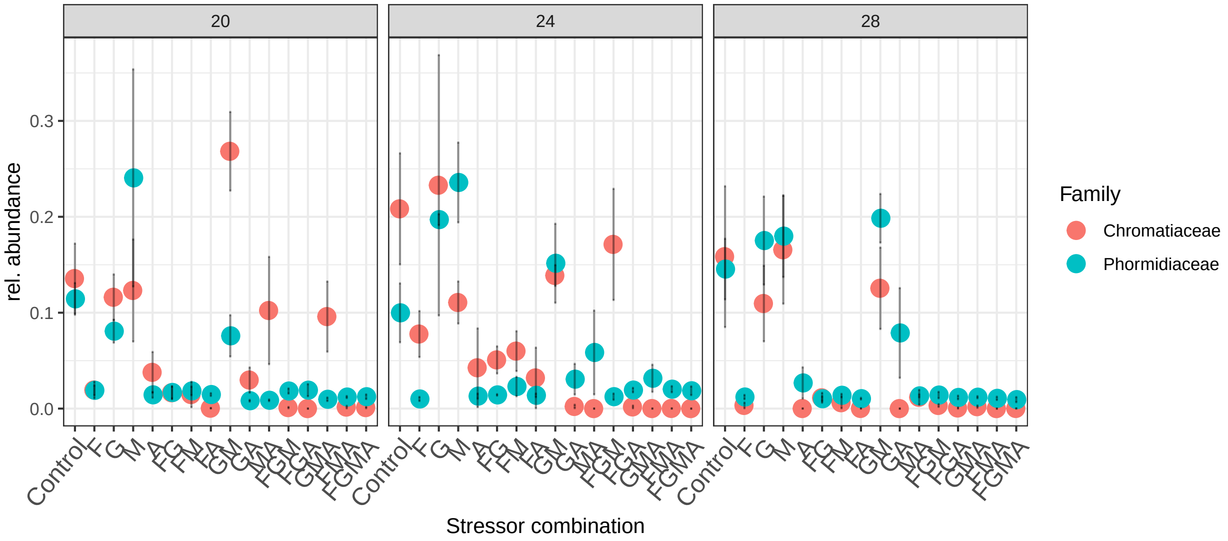


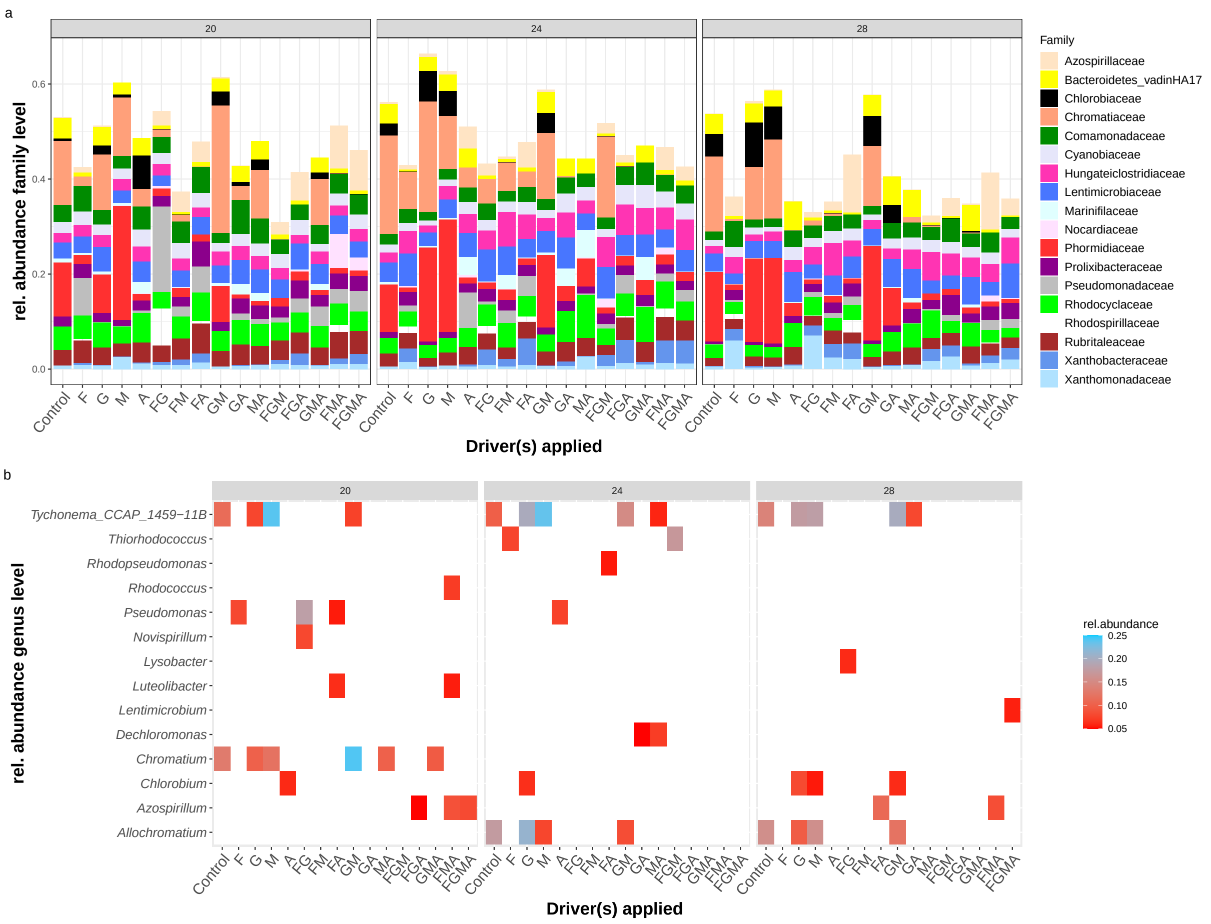
Fig 1: Combined figure at family and genus level

In the next code chunk the figure at genus level is created

```r
df_genus<- df_rel %>%
  group_by(Genus,Lable,Combinations,Temperature_factor) %>%
  dplyr::summarize(Abundance=sum(Abundance))

df_genus <- df_genus %>%
  dplyr::group_by(Genus,Combinations,Temperature_factor) %>%
  dplyr::summarize(Abundance = mean(Abundance))

index <- which(df_genus$Abundance>=0.05)
genus_to_keep <- unique(df_genus[index,"Genus"])
genus_to_keep <- unname(unlist(genus_to_keep))

df_genus $Genus_filter <- ifelse(df_genus$Genus %in% genus_to_keep, df_genus$Genus,"other")

#make other to one
df_genus<- df_genus %>%
  dplyr::group_by(Genus_filter,Combinations,Temperature_factor) %>%
  dplyr::summarize(Abundance = sum(Abundance))

df_genus <- df_genus %>%
  reorder_levels(Combinations, order = c("Control","F","G","M","A","FG","FM","FA","GM",
                                         "GA","MA","FGM","FGA","GMA","FMA","FGMA"))

genus_plot <- df_genus %>%
  filter(Combinations != "ori", Genus_filter != "other", Genus_filter !="Pirellulaceae",
         Genus_filter !="Cyanobium_PCC-6307", Genus_filter !="HN-HFO106") %>%
  ggplot(aes(x = Combinations, y = Genus_filter, fill=Abundance)) +
  geom_tile() +
  scale_fill_continuous(name = "rel.abundance", low = "#FF0000", high = "#00CCFF",
                        na.value="white", breaks=c(0.05,0.1,0.15,0.2,0.25),
                        limits=c(0.05,0.25)) +
  facet_wrap("Temperature_factor") +
  theme(axis.text.x = element_text(angle = 50, hjust = 1, size=14),
        axis.text.y = element_text(size=12,face="italic"),
        axis.title=element_text(size=16,face="bold")) +
  labs(x="Driver(s) applied ", y="rel. abundance genus level")
```

The combined figure (Fig. 1 in manuscript):

```r
(plot_spacer() + community_plot + plot_layout(widths=c(1,100))) /
  (plot_spacer() + genus_plot + plot_layout(widths=c(1,100))) +
  plot_layout(ncol=1,heights=c(1,1)) + plot_annotation(tag_levels = "a")
```



Alpha diversity

Here the alpha diversity is calculated and then used in a regression.

Calculation

```
diversity_test <- estimate_richness(ps_new_11, split=TRUE, measures=NULL)
```

```
## Warning in estimate_richness(ps_new_11, split = TRUE, measures = NULL): The data you have provided d
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```

```
diversity_test$File <- rownames(diversity_test)
diversity_test$File <- gsub("\\.\\.", "--", diversity_test$File)
div_raw <- left_join(diversity_test,sample_data(ps_new_11))
```

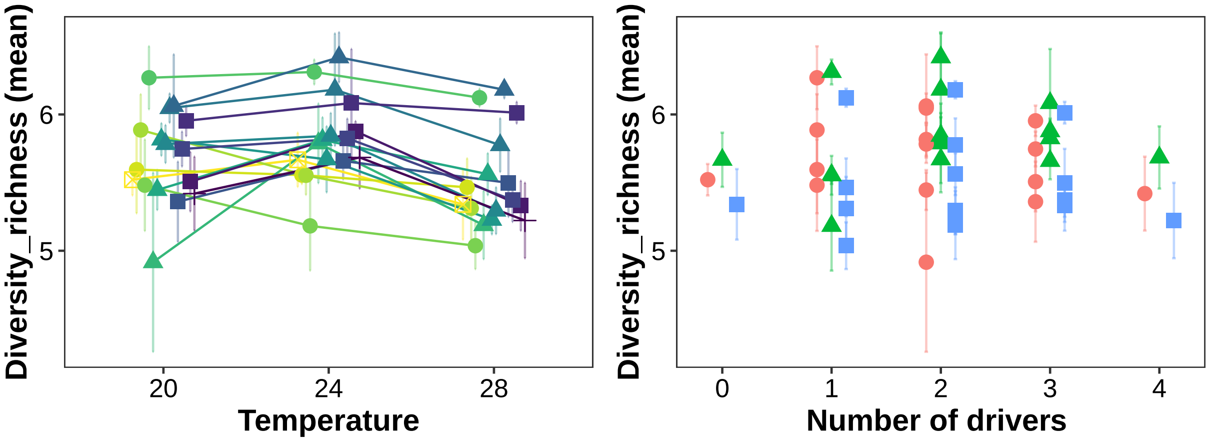
```
## Warning in class(x) <- c(setdiff(subclass, tibble_class), tibble_class): Setting
## class(x) to multiple strings ("tbl_df", "tbl", ...); result will no longer be an
## S4 object
```

```
#filter samples with low amount of reads
div_raw <- div_raw %>%
  filter(Observed >=100, Combination != "ori") %>%
  mutate(T2=Temperature^2,T=Temperature_factor) %>%
  select(-Temperature)%>%
  reorder_levels(Combinations, order = c("Control","F","G","M","A","FG","FM","FA","GM","GA","MA",
                                         "FGM","FGA","GMA","FMA","FGMA"))

#calculating mean of shannon and observed reads
div_means <- div_raw %>%
  filter(T %in% c("20", "24", "28")) %>%
  dplyr::group_by(Treatment, Combinations, T, Stressor,Stressor_numeric) %>%
  dplyr::summarize(Shannon_mean = mean(Shannon),
                   Observed_mean = mean(Observed),
                   Shannon_SE = stderr(Shannon),
                   Observed_SE = stderr(Observed))
```

Figures

Figures for the Shannon index (diversity)

```
# plots
plot_diversity_all <- div_raw %>%
  filter(T!="19")%>%
  ggplot() +
  geom_point(aes(x=Combinations, y=Shannon, shape=Stressor,col=Combinations),
             size = 3,position = position_dodge(width = 0.4)) +
  facet_wrap("T")+
  marcel_theme() +
  ylab ("Shannon index") +
  xlab("Driver(s) applied ") +
  scale_colour_viridis_d(direction = -1)+
  theme(axis.text.x = element_text(angle = 50, hjust = 1))

plot_diversity_means <- div_means %>%
  ggplot(aes(x=T, y=Shannon_mean, col=Combinations, group=Combinations)) +
  geom_errorbar(aes(ymin=Shannon_mean-Shannon_SE, ymax=Shannon_mean+Shannon_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4))+
  geom_line(position = position_dodge(width = 0.4)) +
  geom_point(aes(shape=Stressor), size = 3, position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  scale_colour_viridis_d(direction = -1) +
  labs(x="Temperature", y="Diversity_richness (mean)")

plot_diversity_means_numeric <- div_means %>%
  ggplot(aes(x=Stressor_numeric, y=Shannon_mean, col=T)) +
  geom_point(aes(shape= T), size = 3, position = position_dodge(width = 0.4)) +
  geom_errorbar(aes(ymin=Shannon_mean-Shannon_SE, ymax=Shannon_mean+Shannon_SE),
                 width=0.1,alpha=0.4,position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  labs(x="Number of drivers ", y="Diversity_richness (mean)")

#final plot
(plot_diversity_means + plot_diversity_means_numeric)
```



Anova

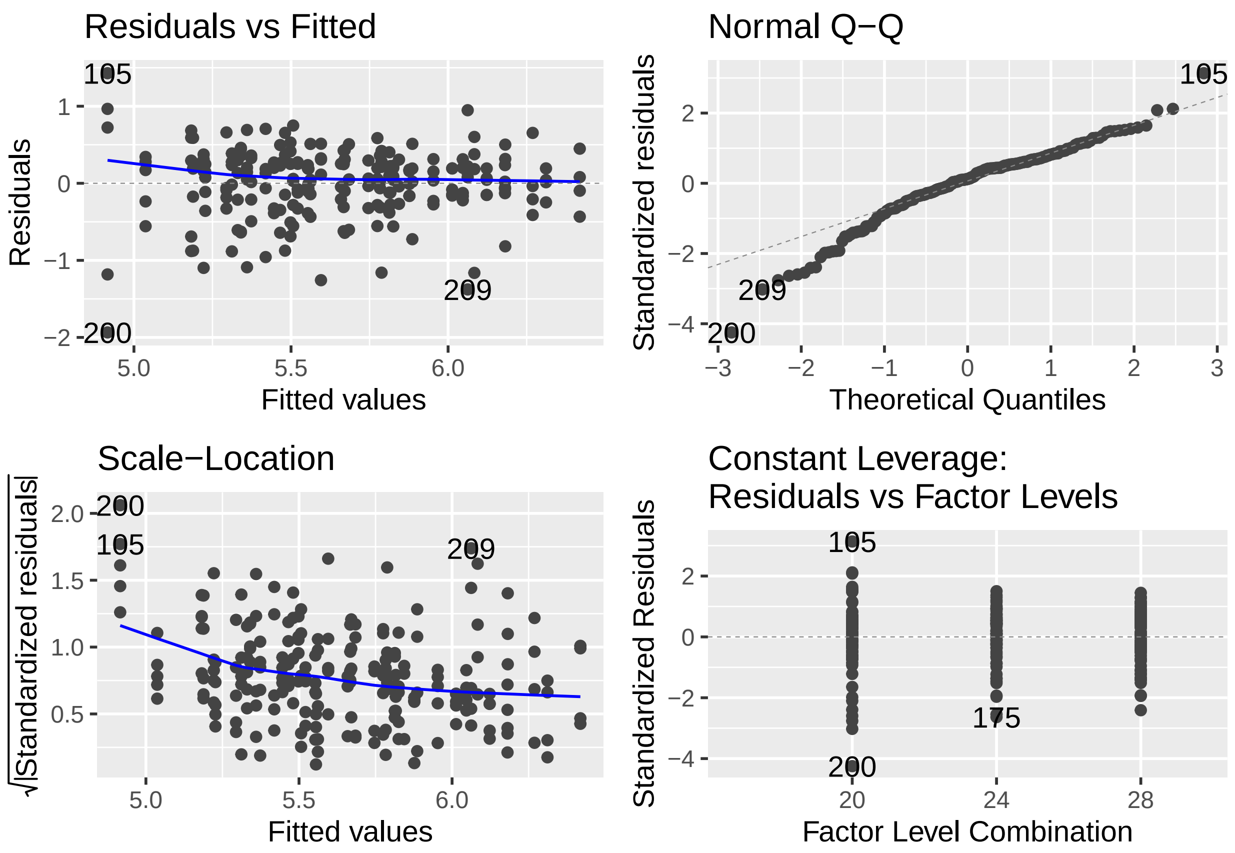
Regression with alpha diversity as response variable.

```
#shannon on combinations
anova_diversity <- lm(Shannon ~F*G*M*A*T, data = div_raw)
# anova(anova_diversity)
# summary(anova_diversity)
```

Model diagnostics

```
autoplot(anova_diversity)
```

```
## Warning: `arrange_()` was deprecated in dplyr 0.7.0.
## Please use `arrange()` instead.
## See vignette('programming') for more help
```



NMDS

Calculation

```
mds_whole <- ps_relative@otu_table %>%
  as.data.frame() %>%
  metaMDS(.,
          distance = "bray", trace=F, # trace = F silences the output
          k = 3, ## number of dimensions to reduce to
          try = 100, ## number of random starts to try
          autotransform = FALSE ## best not to use
  )
## 0.11
mds_whole
```

```
##
## Call:
## metaMDS(comm = ., distance = "bray", k = 3, try = 100, autotransform = FALSE,      trace = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     .
## Distance: bray
```

```
##
## Dimensions: 3
## Stress:     0.1147547
## Stress type 1, weak ties
## Two convergent solutions found after 100 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on '.'
```

```
mds_whole_res <- ps_relative@sam_data %>%
  as_tibble() %>%
  select(Treatment, Stressor,Stressor_numeric, Temperature, Combinations, Temperature_factor, Lable, Nar
  bind_cols(as_tibble(scores(mds_whole, display = "sites"))) %>%
  mutate(T2=Temperature^2,T=Temperature_factor) %>%
  select(-Temperature)  %>%
  filter(Temperature_factor != "19") %>%
  reorder_levels(Combinations, order = c("Control","F","G","M","A","FG","FM","FA","GM",
                                         "GA","MA","FGM","FGA","GMA","FMA","FGMA"))
```

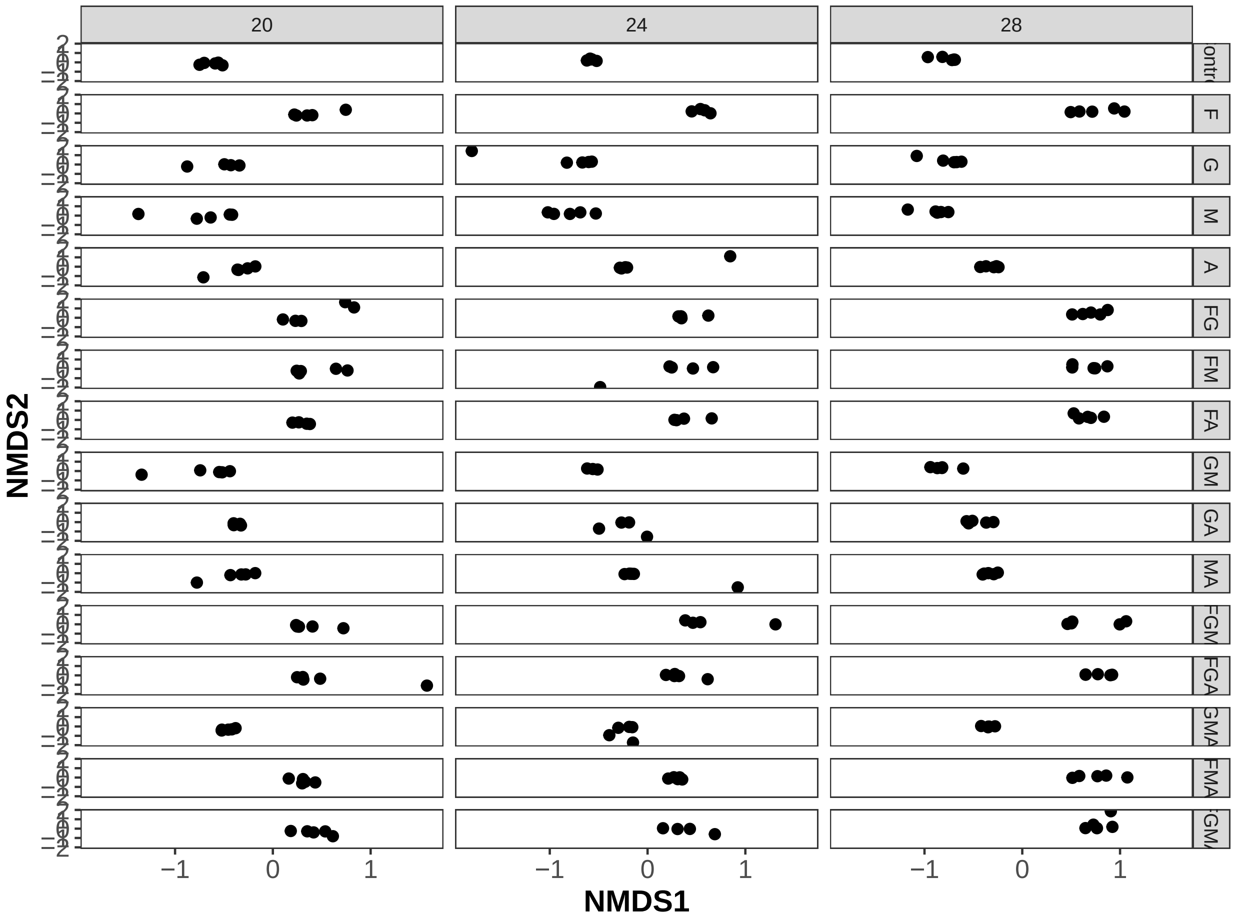
```
## Warning in class(x) <- c(setdiff(subclass, tibble_class), tibble_class): Setting
## class(x) to multiple strings ("tbl_df", "tbl", ...); result will no longer be an
## S4 object
```

```
#means of NMDS
means_NMDS <- mds_whole_res %>%
  filter(Temperature_factor %in% c("20", "24", "28")) %>%
  dplyr::group_by(Treatment,Combinations, Stressor, Stressor_numeric, Temperature_factor,F,G,M,A) %>%
  dplyr::summarize(NMDS1_mean = mean(NMDS1),
                   NMDS2_mean = mean(NMDS2),
                   NMDS3_mean = mean(NMDS3),
                   NMDS1_SE = stderr(NMDS1),
                   NMDS2_SE = stderr(NMDS2),
                   NMDS3_SE= stderr(NMDS3))
```

Figures

NMDS figure (1 & 2)

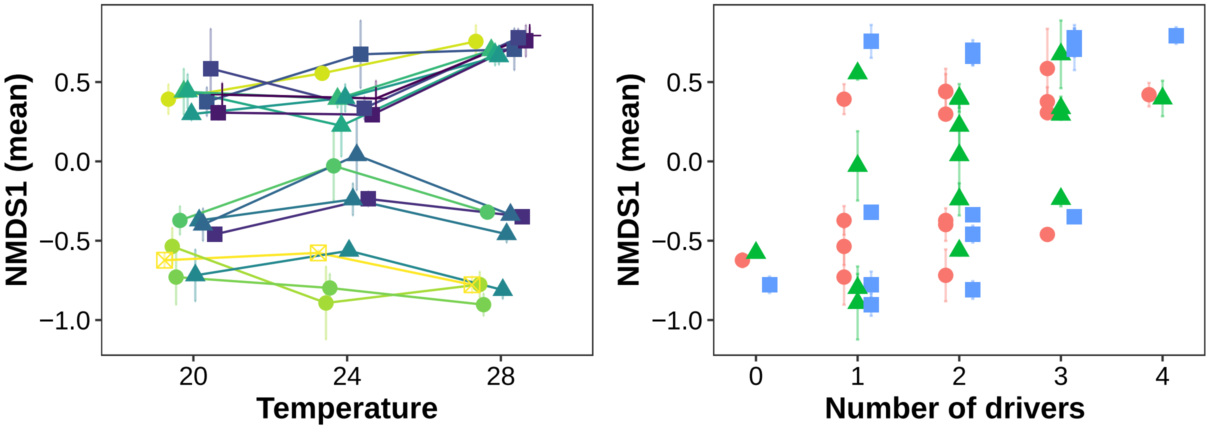
```
ggplot(mds_whole_res, aes(x = NMDS1, y = NMDS2)) +
  geom_point(size = 2) +
  facet_grid(Combinations~Temperature_factor)+
  theme_bw() + theme(panel.grid = element_blank()) +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"))
```



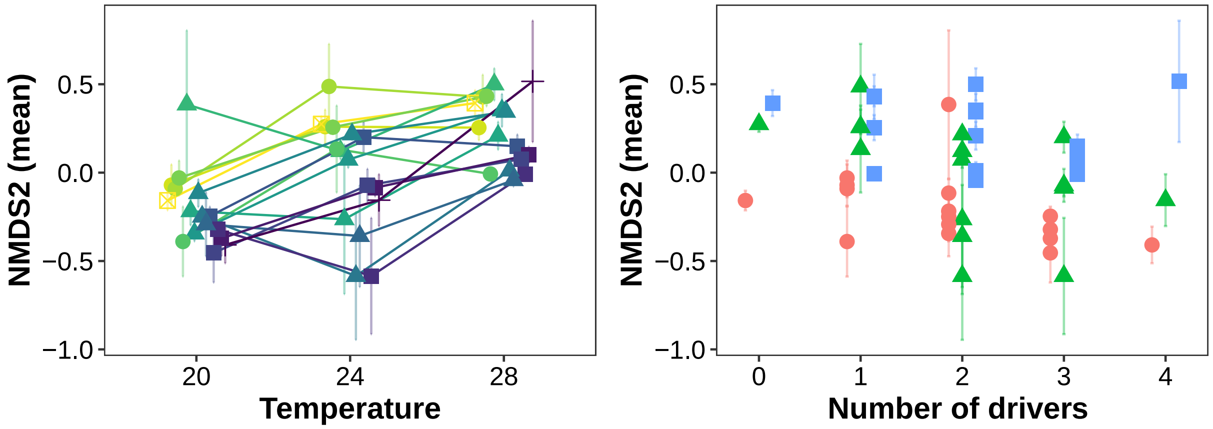
Plots of NMDS (1 and 2) distances that show high significance

```
#NMDS1 supplement plot
nmds1_plot_temperature <- means_NMDS %>%
  ggplot(aes(x = Temperature_factor, y=NMDS1_mean, col=Combinations, group=Combinations)) +
  geom_errorbar(aes(ymin = NMDS1_mean-NMDS1_SE, ymax = NMDS1_mean+NMDS1_SE),
              width=0.1, alpha=0.4, position = position_dodge(width = 0.4))+
  geom_line(position = position_dodge(width = 0.4)) +
  geom_point(aes(shape = Stressor), size = 3, position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  scale_colour_viridis_d(direction = -1)+
  labs(x="Temperature", y="NMDS1 (mean)")

plot_NMDS1_means_numeric <- means_NMDS %>%
  ggplot(aes(x=Stressor_numeric, y=NMDS1_mean, col=Temperature_factor)) +
  geom_point(aes(shape=Temperature_factor), size = 3, position = position_dodge(width = 0.4)) +
  geom_errorbar(aes(ymin=NMDS1_mean-NMDS1_SE, ymax=NMDS1_mean+NMDS1_SE),
              width=0.1,alpha=0.4,position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  labs(x="Number of drivers ", y="NMDS1 (mean)")

# plot NMDS1
```

```
nmds1_plot_temperature +  plot_NMDS1_means_numeric
```



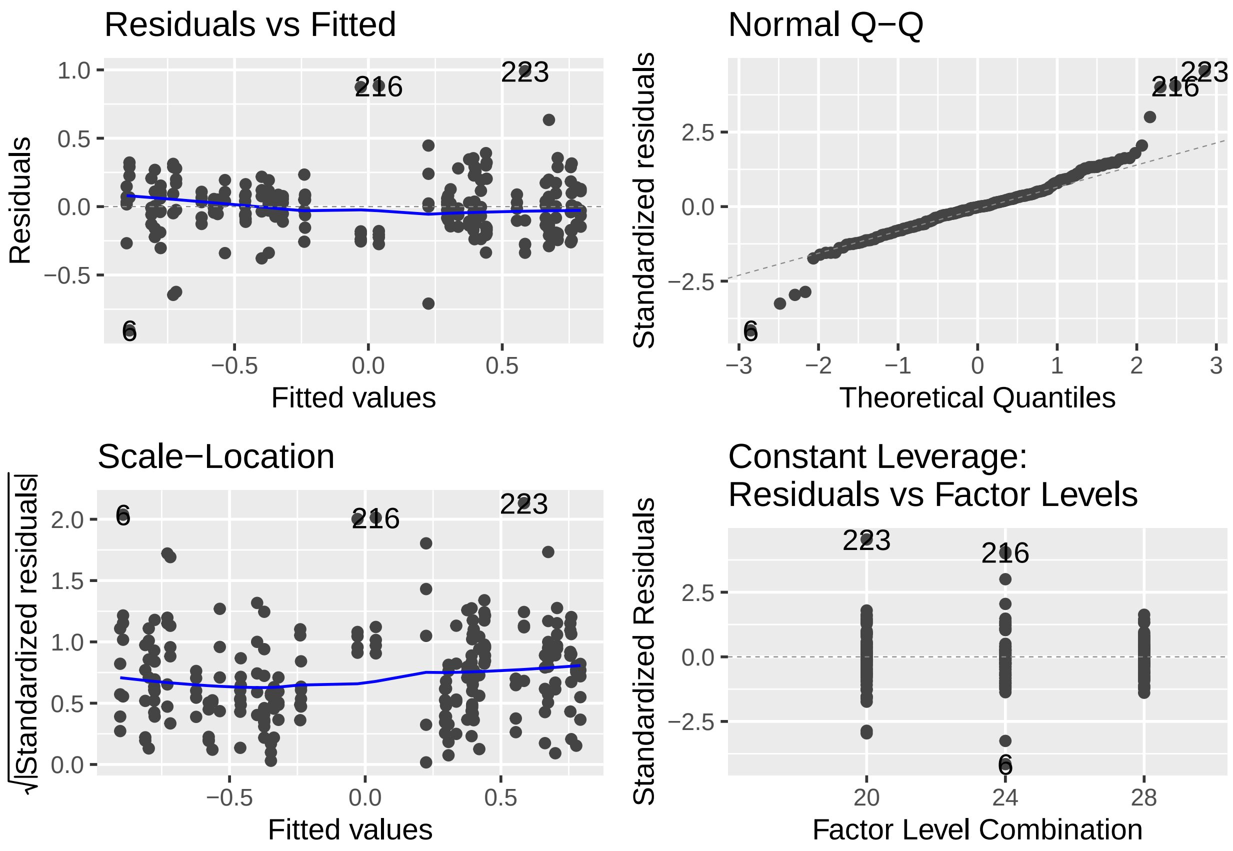
```
#NMDS2 supplement plot

NMDS2_plot_temperature <- means_NMDS %>%
  ggplot(aes(x=Temperature_factor, y=NMDS2_mean, col=Combinations, group=Combinations)) +
  geom_errorbar(aes(ymin=NMDS2_mean-NMDS2_SE, ymax=NMDS2_mean+NMDS2_SE),
                width=0.1,alpha=0.4, position = position_dodge(width = 0.4))+
  geom_line(position = position_dodge(width = 0.4)) +
  geom_point(aes(shape=Stressor), size = 3, position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  scale_colour_viridis_d(direction = -1)+
  labs(x="Temperature", y="NMDS2 (mean)")

plot_NMDS2_means_numeric <- means_NMDS %>%
  ggplot(aes(x=Stressor_numeric, y=NMDS2_mean, col=Temperature_factor)) +
  geom_point(aes(shape=Temperature_factor), size = 3, position = position_dodge(width = 0.4)) +
  geom_errorbar(aes(ymin=NMDS2_mean-NMDS2_SE, ymax=NMDS2_mean+NMDS2_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  labs(x="Number of drivers ", y="NMDS2 (mean)")

# plot NMDS2
NMDS2_plot_temperature +  plot_NMDS2_means_numeric
```



Anova

NMDS1

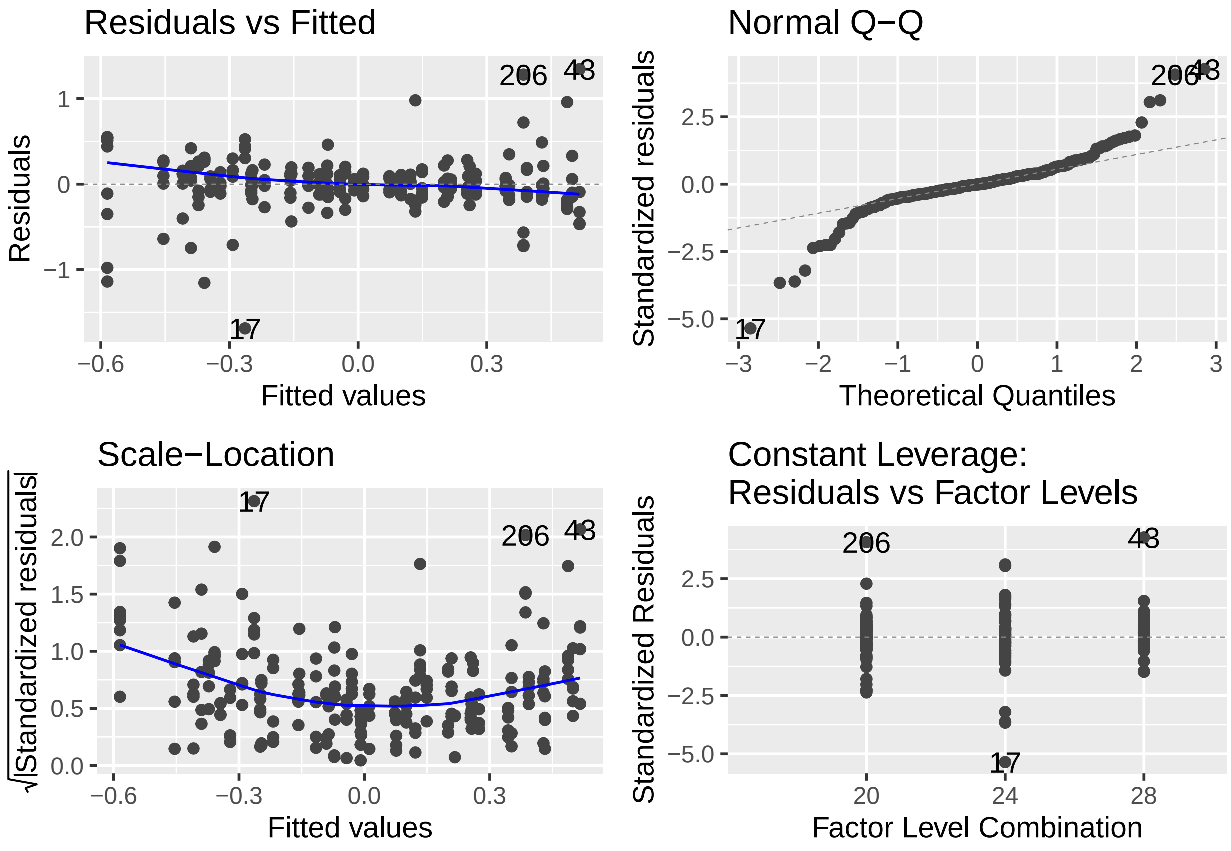
```r
anova_NMDS1 <- lm(NMDS1 ~F*G*M*A*T, data = mds_whole_res)
autoplot(anova_NMDS1)
```



```
# anova(anova_NMDS1)
# summary(anova_NMDS1)
```

NMDS2

```
anova_NMDS2 <- lm(NMDS2 ~F*G*M*A*T, data = mds_whole_res)
autoplot(anova_NMDS2)
```

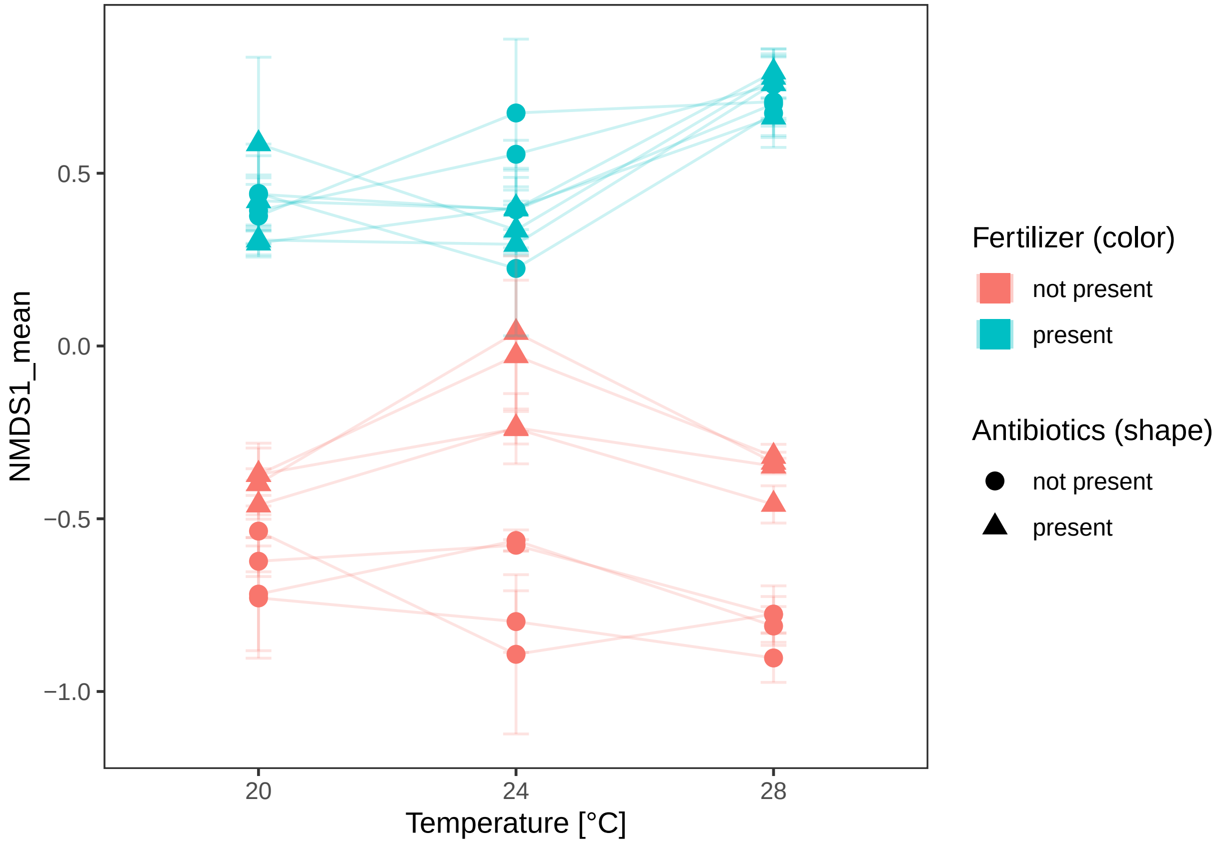


```
# anova(anova_NMDS2)
# summary(anova_NMDS2)
```

Fig 5: NMDS1 anova figure

Manuscript Fig. 5

```
#F:A:Temperature_factor
means_NMDS %>%
  ggplot(aes(x=Temperature_factor,y=NMDS1_mean, col=as.factor(F))) +
  geom_line(aes(group=interaction(as.factor(F),A,M,G)), alpha=0.2)+
  geom_point(mapping = aes(shape=as.factor(A)), size=3)+
  geom_errorbar(aes(ymin=NMDS1_mean-NMDS1_SE, ymax=NMDS1_mean+NMDS1_SE,col=as.factor(F)),
                width=0.1,alpha=0.2)+
  theme_bw() + theme(panel.grid = element_blank())+
  xlab("Temperature [°C]") +
  scale_color_discrete(name="Fertilizer (color)",labels=c("not present", "present")) +
  scale_shape_discrete(name="Antibiotics (shape)",labels=c("not present", "present")) +
  guides(colour = guide_legend(override.aes = list(shape = 15,size=5)))
```



Subpopulation analysis

Subpopulation calculation

Here, the data is filtered to only included some orders. Then, the NMDS analysis is repreated for this subpopulation

```
ps_rel <- transform_sample_counts(ps_new_11, function(x) x / sum(x) )

relab_threshold <- 0.001

ps_relab <- filter_taxa(ps_rel, function(x) !(sum(x < relab_threshold) == length(x)), TRUE)

ps_interest <- subset_taxa(ps_relab, Order %in% c("Cyanobacteriales","Chromatiales","Synechococcales",

paste("The subpopulation consists of",ntaxa(ps_interest),"taxa")
```

```
## [1] "The subpopulation consists of 493 taxa"
```

```
ps_relative_interest <- transform_sample_counts(ps_interest, function(x)  x / sum(x))

interest <-psmelt(ps_relative_interest)

ps_nmds <- subset_samples(ps_relative_interest, Treatment != "ori")

mds_whole_subset <- ps_nmds@otu_table %>%
  as.data.frame() %>%
  metaMDS(.,
          distance = "bray", trace =F,
          k = 3, ## number of dimensions to reduce to
          try = 300, ## number of random starts to try
          autotransform = FALSE ## best not to use
  )

## 0.1
mds_whole_subset
```

```
##
## Call:
## metaMDS(comm = ., distance = "bray", k = 3, try = 300, autotransform = FALSE,      trace = F)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     .
## Distance: bray
##
## Dimensions: 3
## Stress:     0.1036809
## Stress type 1, weak ties
## Two convergent solutions found after 300 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on '.'
```

```
mds_whole_res_subset <- ps_nmds@sam_data %>%
  as_tibble() %>%
  select(Treatment,Temperature, Temperature_factor, Stressor,Stressor_numeric, Combinations, Temperature
  bind_cols(as_tibble(scores(mds_whole_subset, display = "sites"))) %>%
  mutate(T2=Temperature^2,T=Temperature_factor) %>%
  select(-Temperature)
```

```
## Warning in class(x) <- c(setdiff(subclass, tibble_class), tibble_class): Setting
## class(x) to multiple strings ("tbl_df", "tbl", ...); result will no longer be an
## S4 object
```

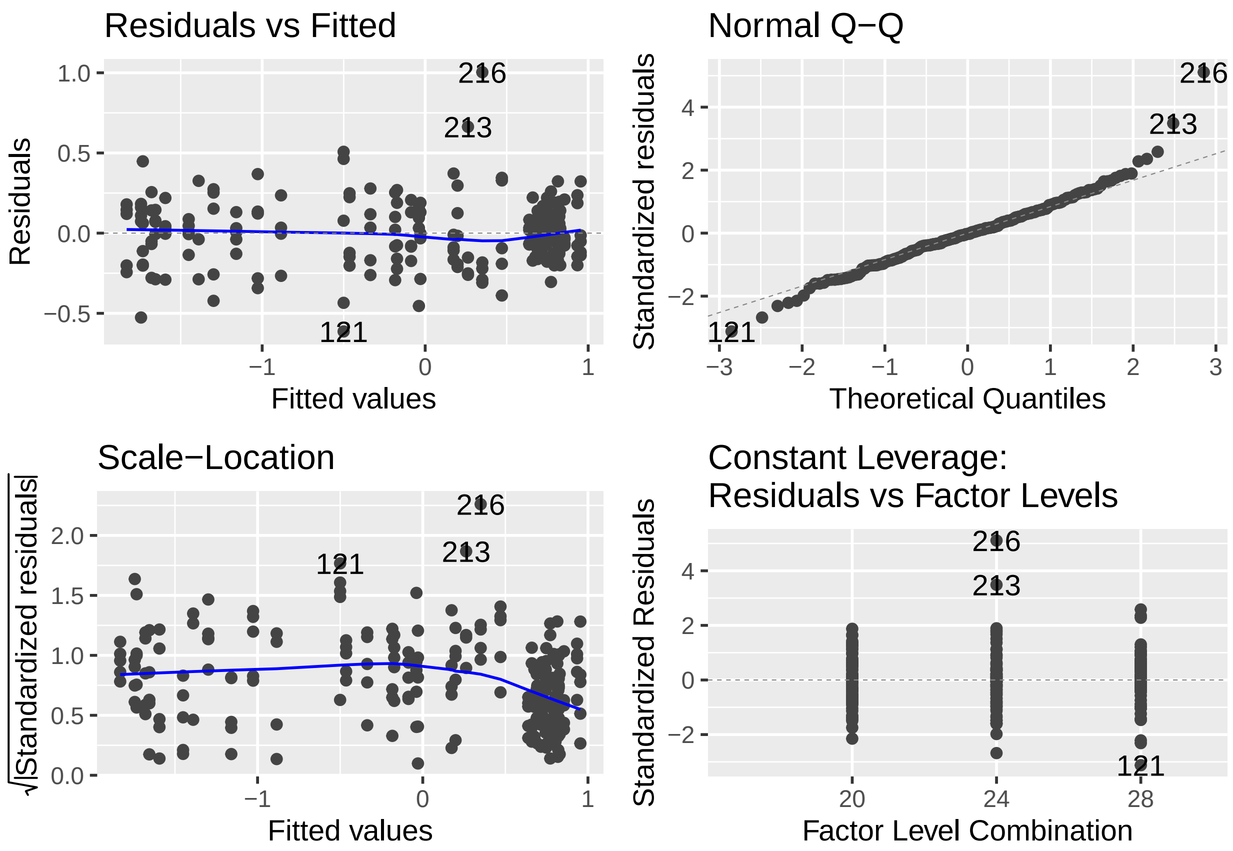
```
mds_whole_res_subset <- mds_whole_res_subset %>%
  reorder_levels(Combinations, order = c("Control","F","G","M","A","FG","FM","FA","GM",
                                         "GA","MA","FGM","FGA","GMA","FMA","FGMA"))

#means of NMDS
```

```r
means_NMDS_subset <- mds_whole_res_subset %>%
  filter(Temperature_factor %in% c("20", "24", "28")) %>%
  dplyr::group_by(Treatment,Combinations, Stressor, Stressor_numeric, Temperature_factor,F,G,M,A) %>%
  dplyr::summarize(NMDS1_mean = mean(NMDS1),
                   NMDS2_mean = mean(NMDS2),
                   NMDS3_mean = mean(NMDS3),
                   NMDS1_SE = stderr(NMDS1),
                   NMDS2_SE = stderr(NMDS2),
                   NMDS3_SE= stderr(NMDS3))
```

Anova (subpopulation)

NMDS1 (subpopulation)

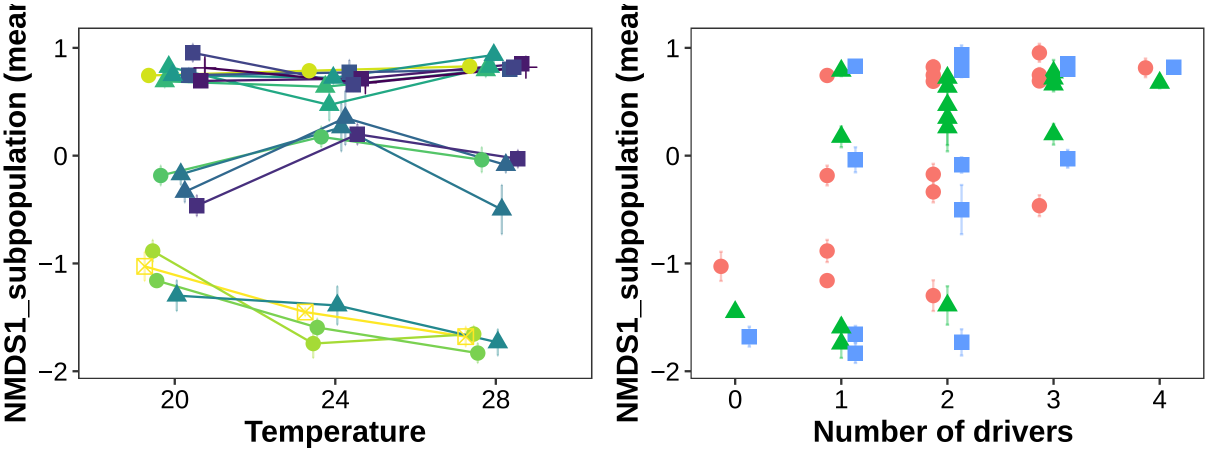
```r
#NMDS1 anova combination
anova_NMDS1_mixed <- lm(NMDS1 ~F*G*M*A*T, data = mds_whole_res_subset)
autoplot(anova_NMDS1_mixed)
```



```r
# anova(anova_NMDS1_mixed)
# summary(anova_NMDS1_mixed)
```

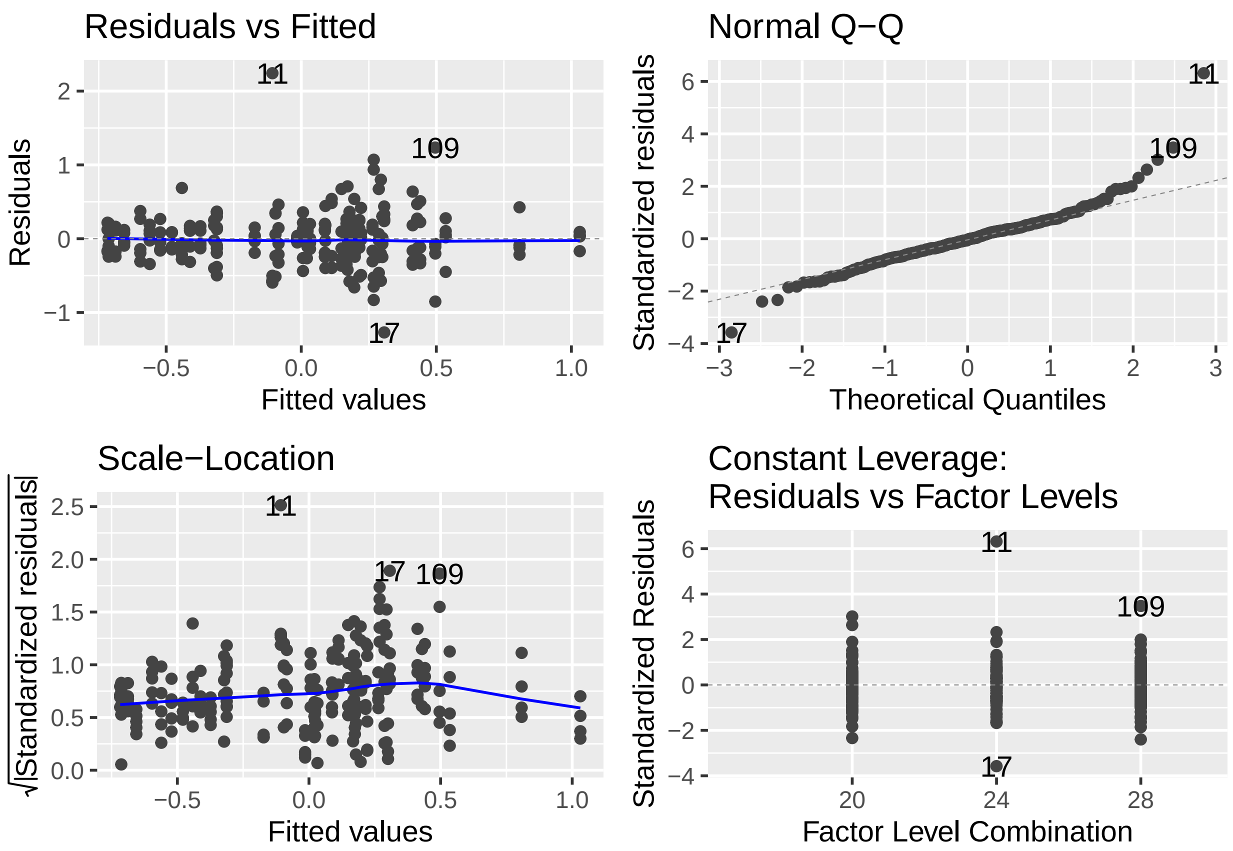
NMDS1 anova figure

```
#NMDS1 subpopulation
nmds1_plot_temperature_subset <- means_NMDS_subset %>%
  ggplot(aes(x=Temperature_factor, y=NMDS1_mean, col=Combinations, group=Combinations)) +
  geom_errorbar(aes(ymin=NMDS1_mean-NMDS1_SE, ymax=NMDS1_mean+NMDS1_SE),
                width=0.1,alpha=0.4, position = position_dodge(width = 0.4))+
  geom_line(position = position_dodge(width = 0.4)) +
  geom_point(aes(shape = Stressor), size = 3, position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  scale_colour_viridis_d(direction = -1) +
  labs(x="Temperature", y="NMDS1_subpopulation (mean)")

plot_NMDS1_means_numeric_subset<- means_NMDS_subset %>%
  ggplot(aes(x=Stressor_numeric, y=NMDS1_mean, col=Temperature_factor)) +
  geom_point(aes(shape = Temperature_factor), size = 3, position = position_dodge(width = 0.4)) +
  geom_errorbar(aes(ymin=NMDS1_mean-NMDS1_SE, ymax=NMDS1_mean+NMDS1_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  labs(x="Number of drivers ", y="NMDS1_subpopulation (mean)")

# plot
nmds1_plot_temperature_subset +  plot_NMDS1_means_numeric_subset
```



NMDS2 (subpopulation)

```
#NMDS2 anova combination
anova_NMDS2_mixed <- lm(NMDS2 ~F*G*M*A*T, data = mds_whole_res_subset)
autoplot(anova_NMDS2_mixed)
```

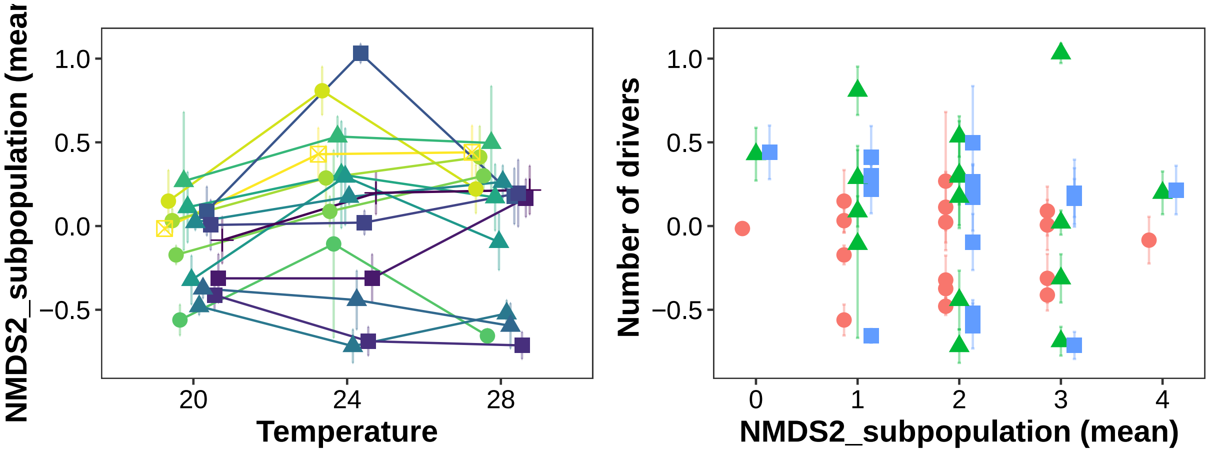


```
# anova(anova_NMDS2_mixed)
# summary(anova_NMDS2_mixed)
```

NMDS2 anova figure

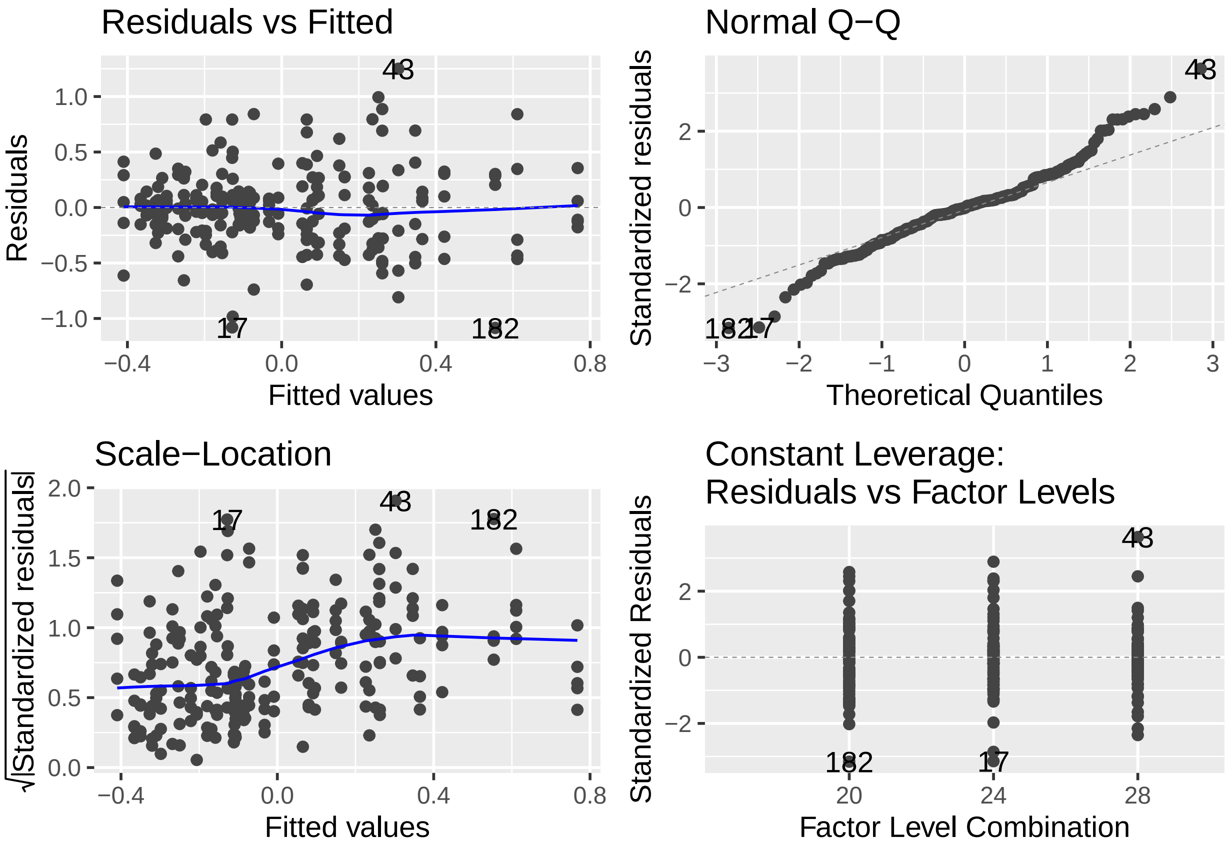
```
#NMDS2 subpopulation
NMDS2_plot_temperature_subset <- means_NMDS_subset %>%
  ggplot(aes(x=Temperature_factor, y=NMDS2_mean, col=Combinations, group=Combinations)) +
  geom_errorbar(aes(ymin=NMDS2_mean-NMDS2_SE, ymax=NMDS2_mean+NMDS2_SE),
              width=0.1,alpha=0.4, position = position_dodge(width = 0.4))+
  geom_line(position = position_dodge(width = 0.4)) +
  geom_point(aes(shape = Stressor), size = 3, position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  scale_colour_viridis_d(direction = -1) +
  labs(x="Temperature", y="NMDS2_subpopulation (mean)")

plot_NMDS2_means_numeric_subset <- means_NMDS_subset %>%
  ggplot(aes(x=Stressor_numeric, y=NMDS2_mean, col=Temperature_factor)) +
  geom_point(aes(shape = Temperature_factor), size = 3, position = position_dodge(width = 0.4)) +
  geom_errorbar(aes(ymin=NMDS2_mean-NMDS2_SE, ymax=NMDS2_mean+NMDS2_SE),
              width=0.1,alpha=0.4,position = position_dodge(width = 0.4)) +
  marcel_theme() +
  theme(legend.position = "none") +
  labs(x="NMDS2_subpopulation (mean)", y="Number of drivers ")
```

```
# plot
NMDS2_plot_temperature_subset +  plot_NMDS2_means_numeric_subset
```



NMDS3 (subpopulation)

```
#NMDS3 anova combination
anova_NMDS3 <- lm(NMDS3 ~F*G*M*A*T, data = mds_whole_res_subset)
autoplot(anova_NMDS3)
```



```
# anova(anova_NMDS3)
# summary(anova_NMDS3)
```

Analysis of single genera

Dataset of single genera, namely Tychonema_CCAP_1459-11B and Sulfuricurvum, calculating NMDS for all, respectively.

Tychonema_CCAP_1459-11B

Calculation

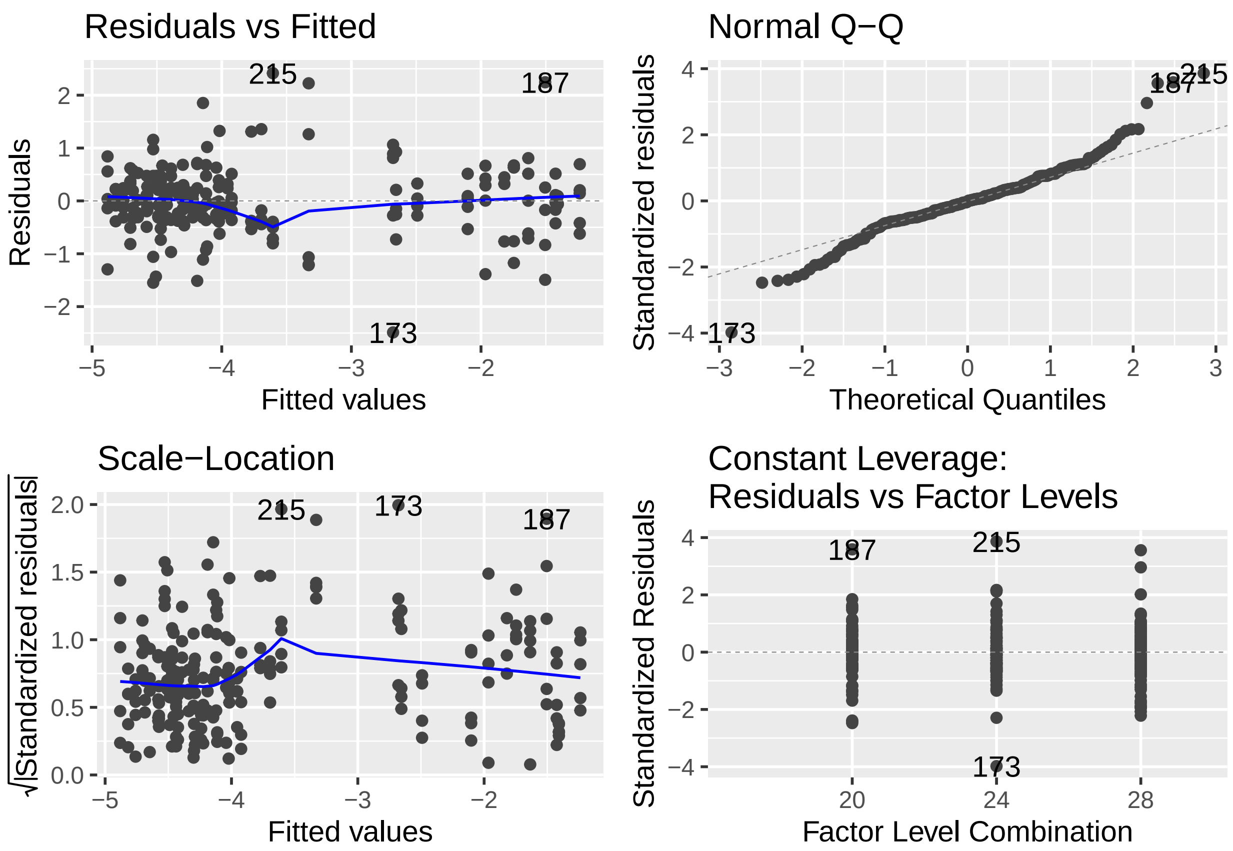
```
ps_relative_cyano <- subset_taxa(ps_relative, Genus == "Tychonema_CCAP_1459-11B")

df_cyano <- psmelt(ps_relative_cyano)

df_cyano <- df_cyano %>%
  mutate(T2=Temperature^2, T=Temperature_factor) %>%
  select(-Temperature)

df_cyano <- df_cyano %>%
  filter(Treatment != "ori")

df_cyano <- df_cyano %>%
  dplyr::group_by(Treatment,Stressor_numeric,Lable,Combinations, T,F,G,M,A) %>%
  dplyr::summarize(Abundance = sum(Abundance)) %>%
  reorder_levels(Combinations, order = c("Control","F","G","M","A","FG","FM","FA","GM",
                                         "GA","MA","FGM","FGA","GMA","FMA","FGMA"))
```

Anova

```
anova_tychonea <- lm(logit(Abundance) ~F*G*M*A*T, data = df_cyano)
# anova(anova_tychonea)
# summary(anova_tychonea)
```

Model diagnostics

```
autoplot(anova_tychonea)
```



Sulfuricurvum

Calculation

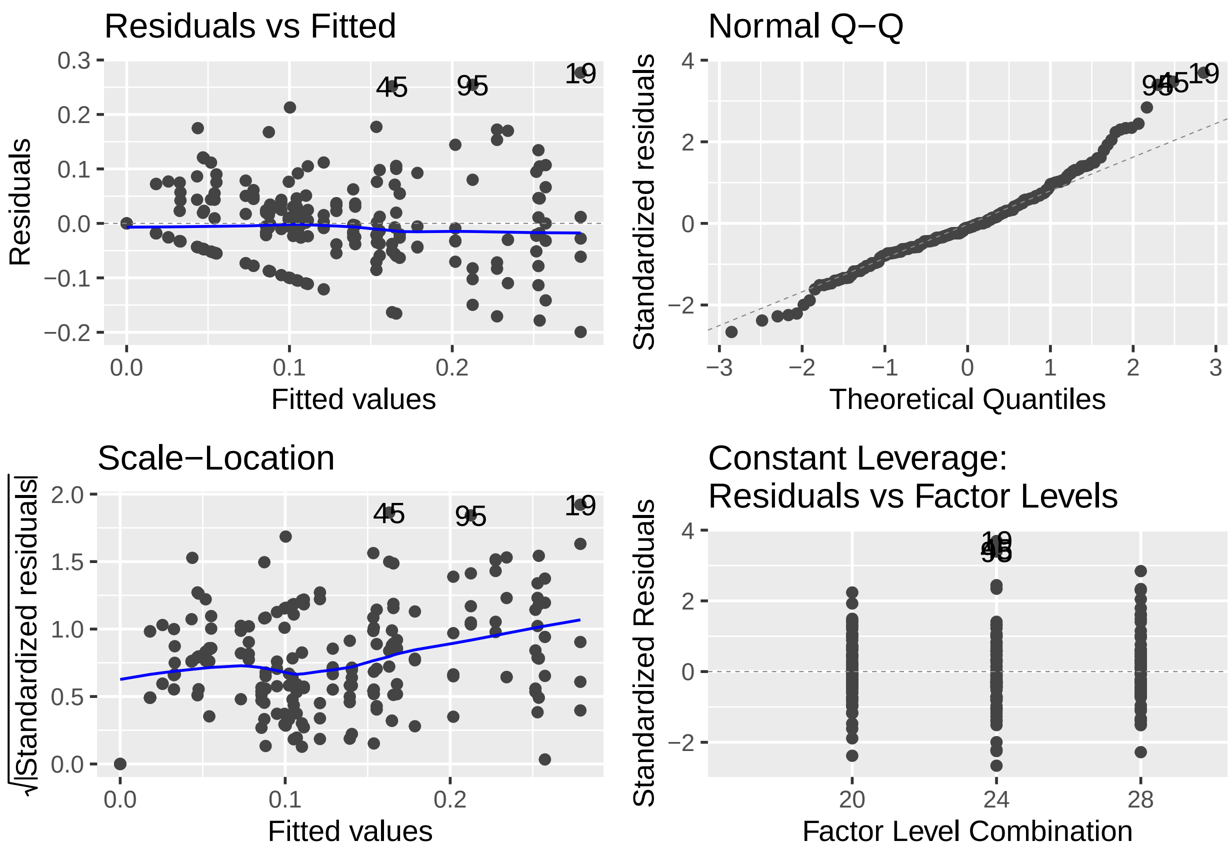
```
ps_sulfuri <- subset_taxa(ps_relative, Genus == "Sulfuricurvum")

df_sulfuri <- psmelt(ps_sulfuri)

df_sulfuri <- df_sulfuri %>%
  mutate(T2=Temperature^2,T=Temperature_factor) %>%
  select(-Temperature)

df_sulfuri <- df_sulfuri %>%
  filter(Treatment!="ori") %>%
  dplyr::group_by(Treatment,Stressor_numeric,Combinations, Lable,T,F,G,M,A) %>%
  dplyr::summarize(Abundance = sum(Abundance))
```

Anova

```
# anova_sulfuri <- lm(asin(sqrt(Abundance)) ~F*G*M*A*T, data = df_sulfuri)
anova_sulfuri <- lm((Abundance)^(1/3) ~F*G*M*A*T, data = df_sulfuri)
```

Model diagnostics

```
autoplot(anova_sulfuri)
```



```
# anova(anova_sulfuri)
# summary(anova_sulfuri)
```

Abiotic factors and oxygen

Calculation

```
abiotic<- ps_relative@sam_data %>%
  as_tibble() %>%
  select(Treatment,Temperature, Combinations, Temperature_factor, Lable,
         Name,F,G,M,A, Oxygen, TN,TC, pH,Stressor_numeric, Stressor) %>%
  mutate(T2=Temperature^2,T=Temperature_factor) %>%
  select(-Temperature)  %>%
  reorder_levels(Combinations, order = c("Control","F","G","M","A","FG","FM","FA","GM",
                                         "GA","MA","FGM","FGA","GMA","FMA","FGMA")) %>%
  dplyr::filter(Treatment!="ori")
```

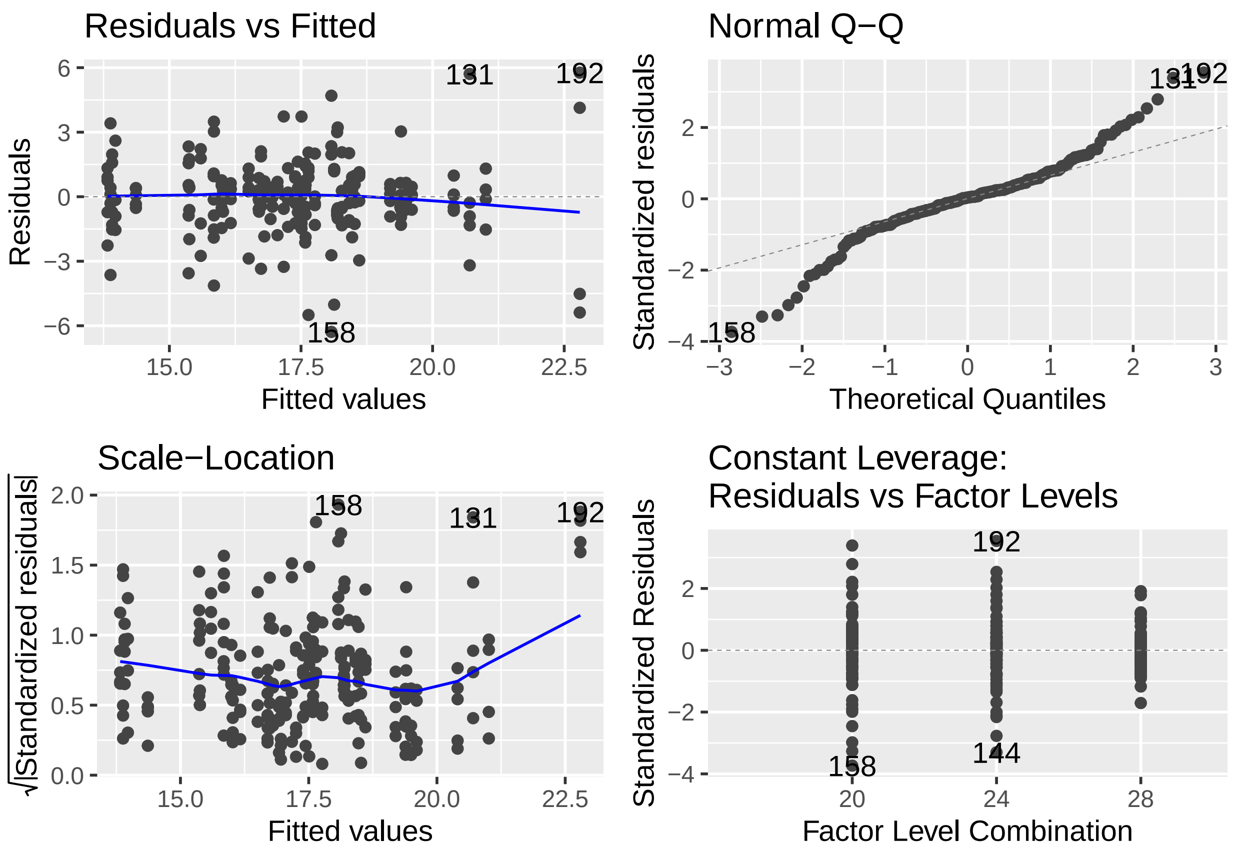
```
## Warning in class(x) <- c(setdiff(subclass, tibble_class), tibble_class): Setting
## class(x) to multiple strings ("tbl_df", "tbl", ...); result will no longer be an
## S4 object
```

```
#calculating mean of oxygen and pH
abiotic_means <- abiotic %>%
  filter(T %in% c("20", "24", "28")) %>%
  dplyr::group_by(Treatment, Combinations, T, Stressor,Stressor_numeric) %>%
  dplyr::summarize(Oxygen_mean = mean(Oxygen),
                   pH_mean = mean(pH),
                   Oxygen_SE = stderr(Oxygen),
                   pH_SE = stderr(pH),
                   TN_mean = mean(TN),
                   TC_mean = mean(TC),
                   TN_SE = stderr(TN),
                   TC_SE = stderr(TC))
```

Oxygen (anova)

Anova

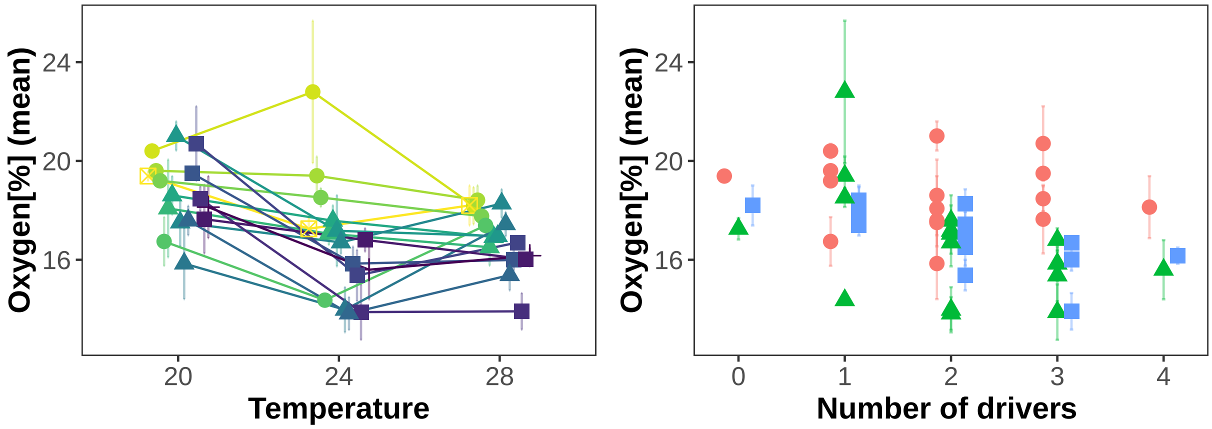
```
oxygen_anova <- lm(Oxygen ~F*G*M*A*T, data = abiotic)
autoplot(oxygen_anova)
```



```
# anova(oxygen_anova)
# summary(oxygen_anova)
```

Figure

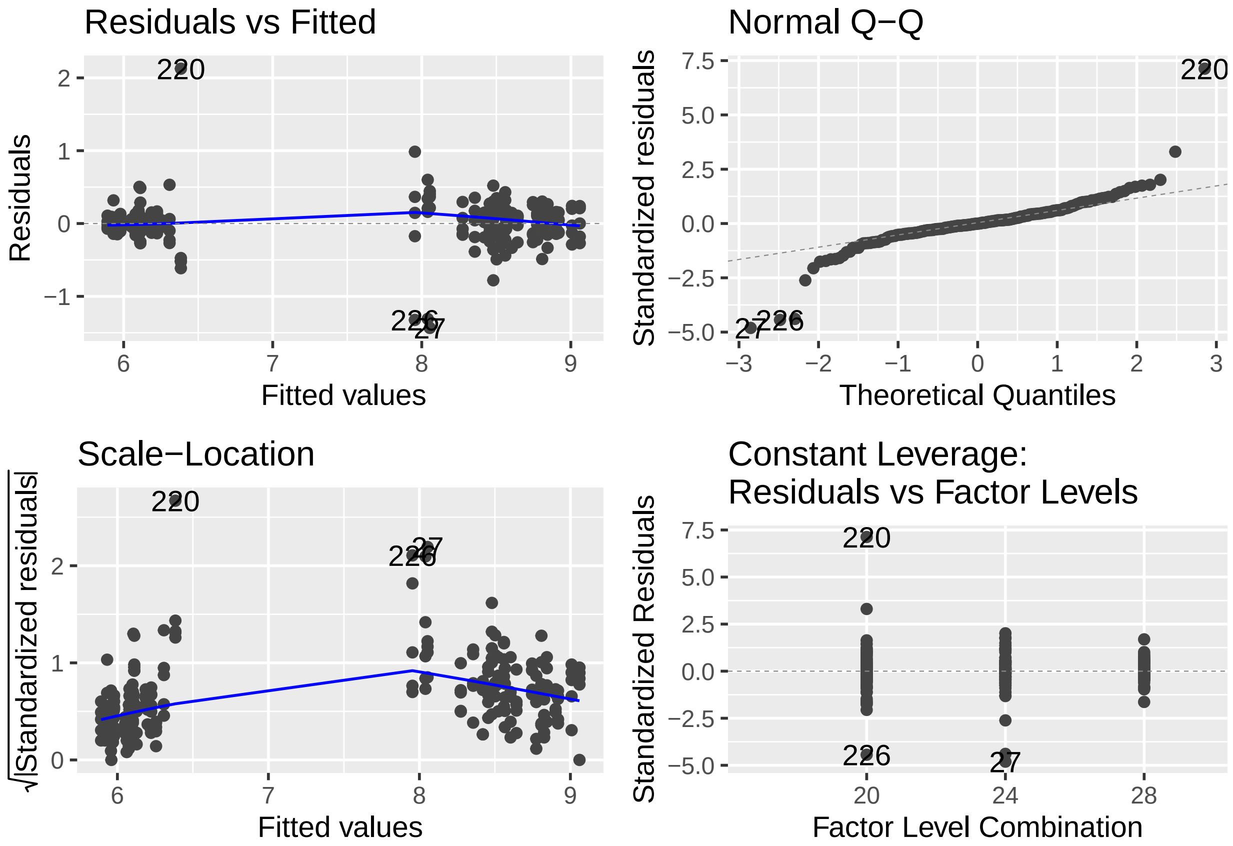
```
plot_oxygen_means <- abiotic_means %>%
  ggplot(aes(x=T, y=Oxygen_mean, col=Combinations, group=Combinations)) +
  geom_errorbar(aes(ymin=Oxygen_mean-Oxygen_SE, ymax=Oxygen_mean+Oxygen_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4))+
  geom_line(position = position_dodge(width = 0.4)) +
  geom_point(aes(shape=Stressor), size = 3, position = position_dodge(width = 0.4)) +
  theme_bw() +
  theme(panel.grid = element_blank(),
        axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"),
        legend.position = "none")+
  scale_colour_viridis_d(direction = -1)+
  labs(x="Temperature", y="Oxygen[%] (mean)")

plot_oxygen_numeric_means <- abiotic_means %>%
  ggplot(aes(x=Stressor_numeric, y=Oxygen_mean, col=T)) +
  geom_point(aes(shape= T), size = 3, position = position_dodge(width = 0.4)) +
  geom_errorbar(aes(ymin=Oxygen_mean-Oxygen_SE, ymax=Oxygen_mean+Oxygen_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4))+
  theme_bw() +
  theme(panel.grid = element_blank(),
        axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"),
        legend.position = "none")+
  labs(y="Oxygen[%] (mean)",x="Number of drivers ")

#plot oxygen
plot_oxygen_means + plot_oxygen_numeric_means
```



pH (anova)

Anova

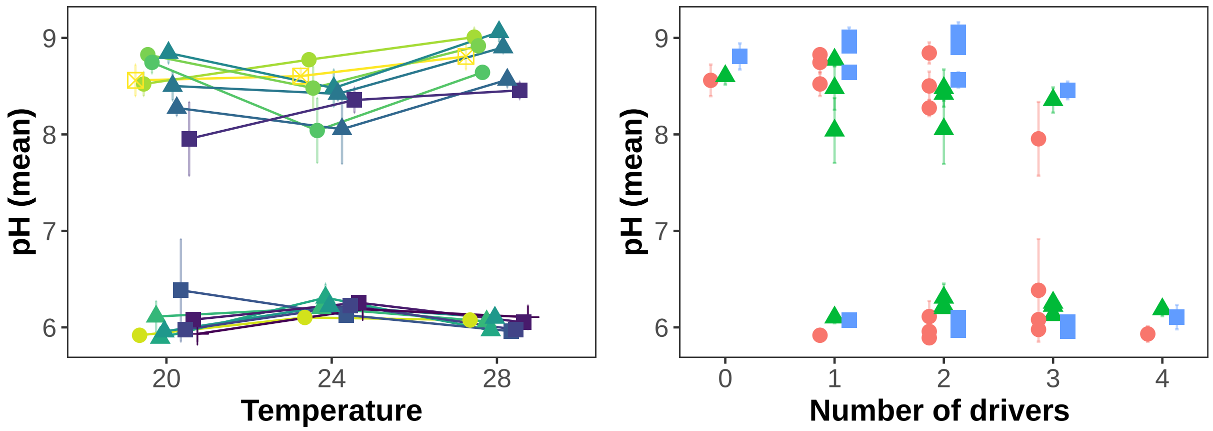
```
pH_anova <- lm(pH ~F*G*M*A*T, data = abiotic)
autoplot(pH_anova)
```



```
# anova(pH_anova)
# summary(pH_anova)
```

Figure

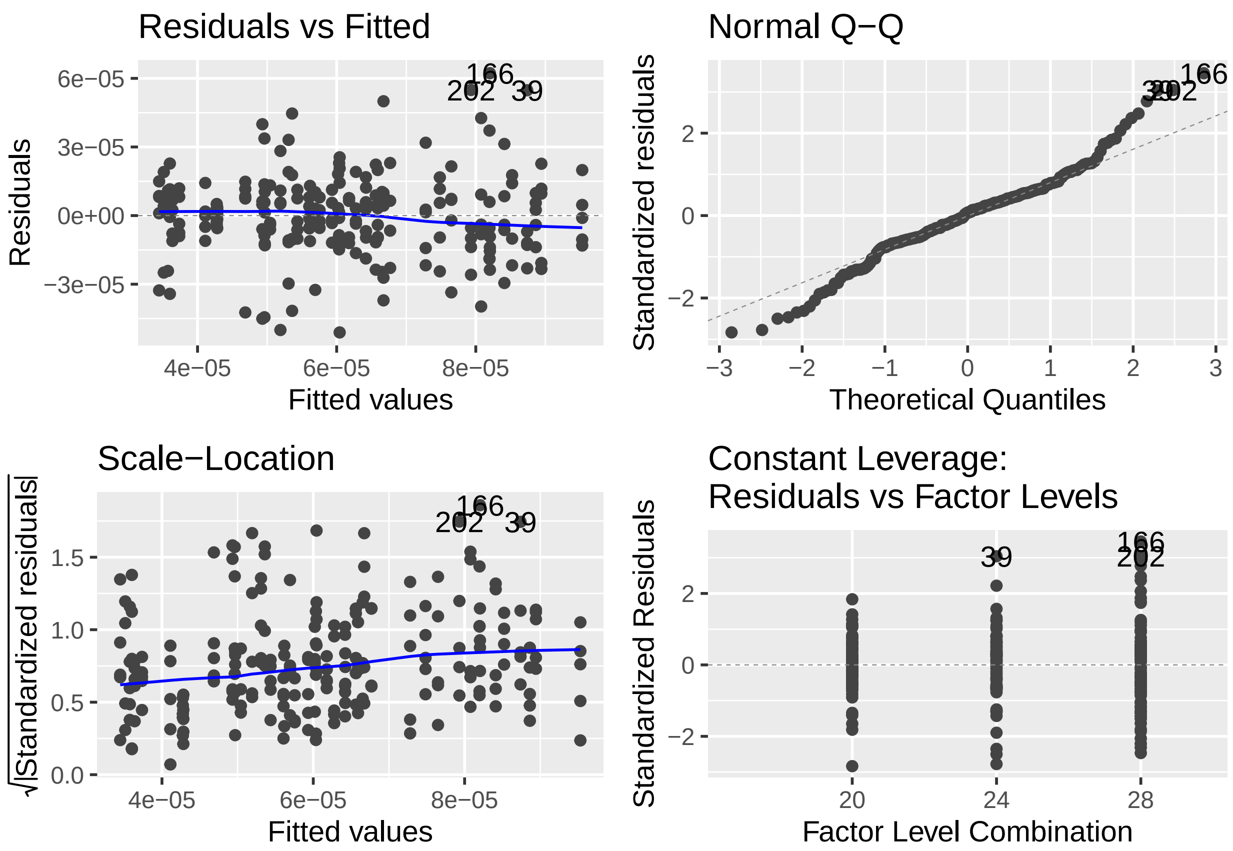
```
#pH
plot_pH_means <- abiotic_means %>%
  ggplot(aes(x=T, y=pH_mean, col=Combinations, group=Combinations)) +
  geom_errorbar(aes(ymin=pH_mean-pH_SE, ymax=pH_mean+pH_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4))+
  geom_line(position = position_dodge(width = 0.4)) +
  geom_point(aes(shape=Stressor), size = 3, position = position_dodge(width = 0.4)) +
  theme_bw() +
  theme(panel.grid = element_blank(),
        axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"),
        legend.position = "none")+
  scale_colour_viridis_d(direction = -1)+
  labs(x="Temperature", y="pH (mean)")

plot_pH_means_numeric <- abiotic_means %>%
  ggplot(aes(x=Stressor_numeric, y=pH_mean, col=T)) +
  geom_point(aes(shape=T), size = 3, position = position_dodge(width = 0.4)) +
  geom_errorbar(aes(ymin=pH_mean-pH_SE, ymax=pH_mean+pH_SE), width=0.1,alpha=0.4,position = position_doc
  theme_bw() +
  theme(panel.grid = element_blank(),
        axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"),
        legend.position = "none")+
  labs(y="pH (mean)",x="Number of drivers ")

#plot pH
plot_pH_means + plot_pH_means_numeric
```



TC (anova)

Anova

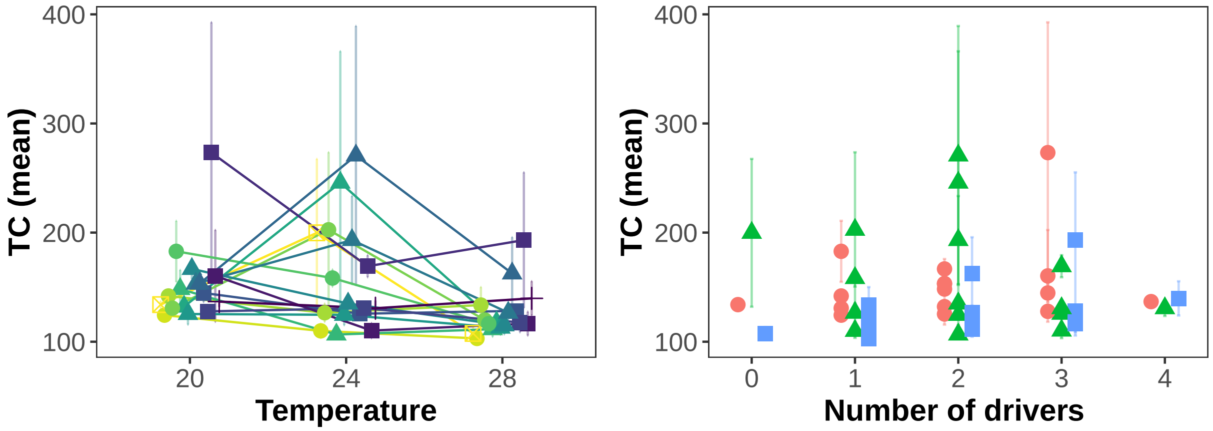
```
tc_anova <- lm((1/TC^2) ~F*G*M*A*T, data = abiotic)
# tc_anova1 <- lm(log10(TC) ~F*G*M*A*T, data = abiotic)
# tc_anova2 <- lm((TC)^(1/4) ~F*G*M*A*T, data = abiotic)

autoplot(tc_anova)
```



```
# anova(tc_anova)
# summary(tc_anova)
```

Figure

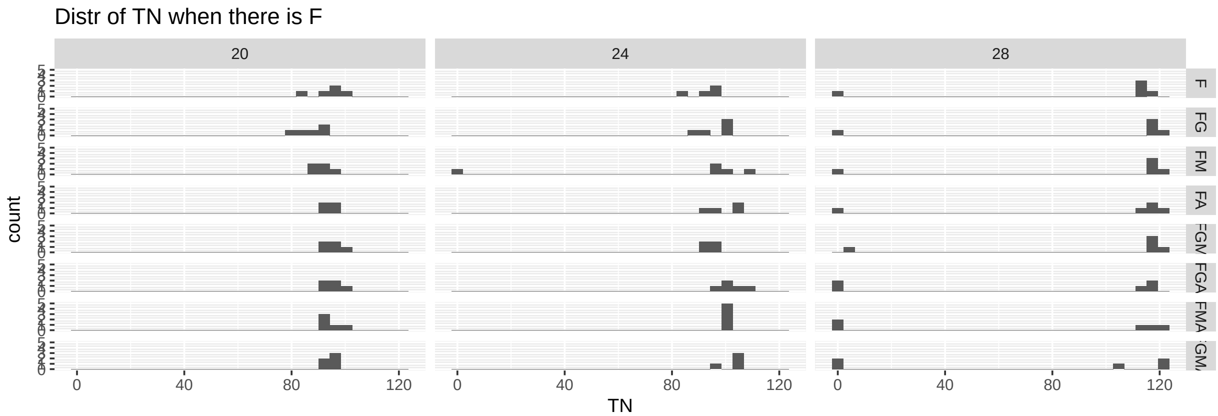
```
#total carbon

plot_TC_means <- abiotic_means %>%
  ggplot(aes(x=T, y=TC_mean, col=Combinations, group=Combinations)) +
  geom_errorbar(aes(ymin=TC_mean-TC_SE, ymax=TC_mean+TC_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4))+
  geom_line(position = position_dodge(width = 0.4)) +
  geom_point(aes(shape=Stressor), size = 3, position = position_dodge(width = 0.4)) +
  theme_bw() +
  theme(panel.grid = element_blank(),
        axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"),
        legend.position = "none")+
  scale_colour_viridis_d(direction = -1)+
  labs(x="Temperature", y="TC (mean)")

plot_TC_means_numeric <- abiotic_means %>%
  ggplot(aes(x=Stressor_numeric, y=TC_mean, col=T)) +
  geom_point(aes(shape=T), size = 3, position = position_dodge(width = 0.4)) +
  geom_errorbar(aes(ymin=TC_mean-TC_SE, ymax=TC_mean+TC_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4))+
  theme_bw() +
  theme(panel.grid = element_blank(),
        axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"),
        legend.position = "none")+
  labs(y="TC (mean)", x="Number of drivers ")

#plot total carbon

plot_TC_means + plot_TC_means_numeric
```

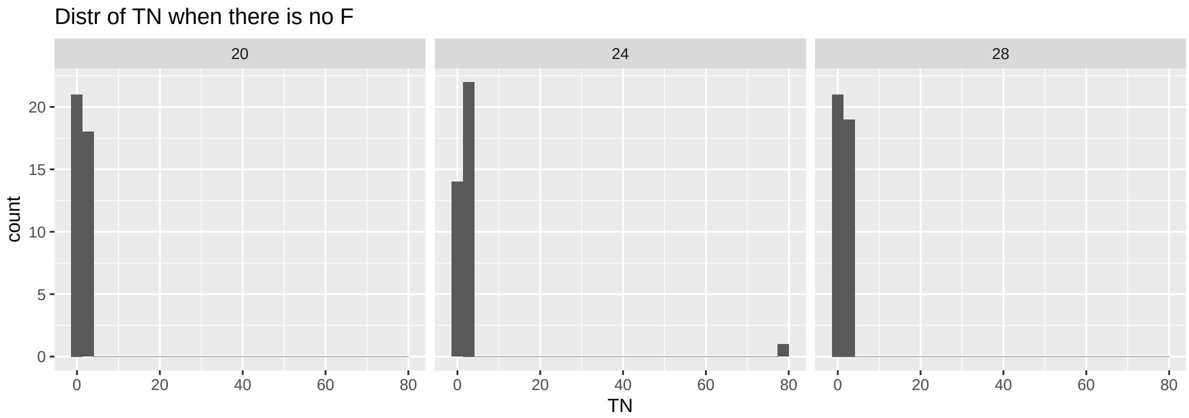


TN (anova)

Data

```
abiotic.F <- abiotic %>% dplyr::filter(F==1)
abiotic.noF <- abiotic %>% dplyr::filter(F==0)

abiotic.F %>% ggplot(aes(TN)) + geom_histogram() + labs(title = "Distr of TN when there is F") + facet_g
abiotic.noF %>% ggplot(aes(TN)) + geom_histogram() + labs(title = "Distr of TN when there is no F") + fa
  plot_layout(ncol = 1)
```

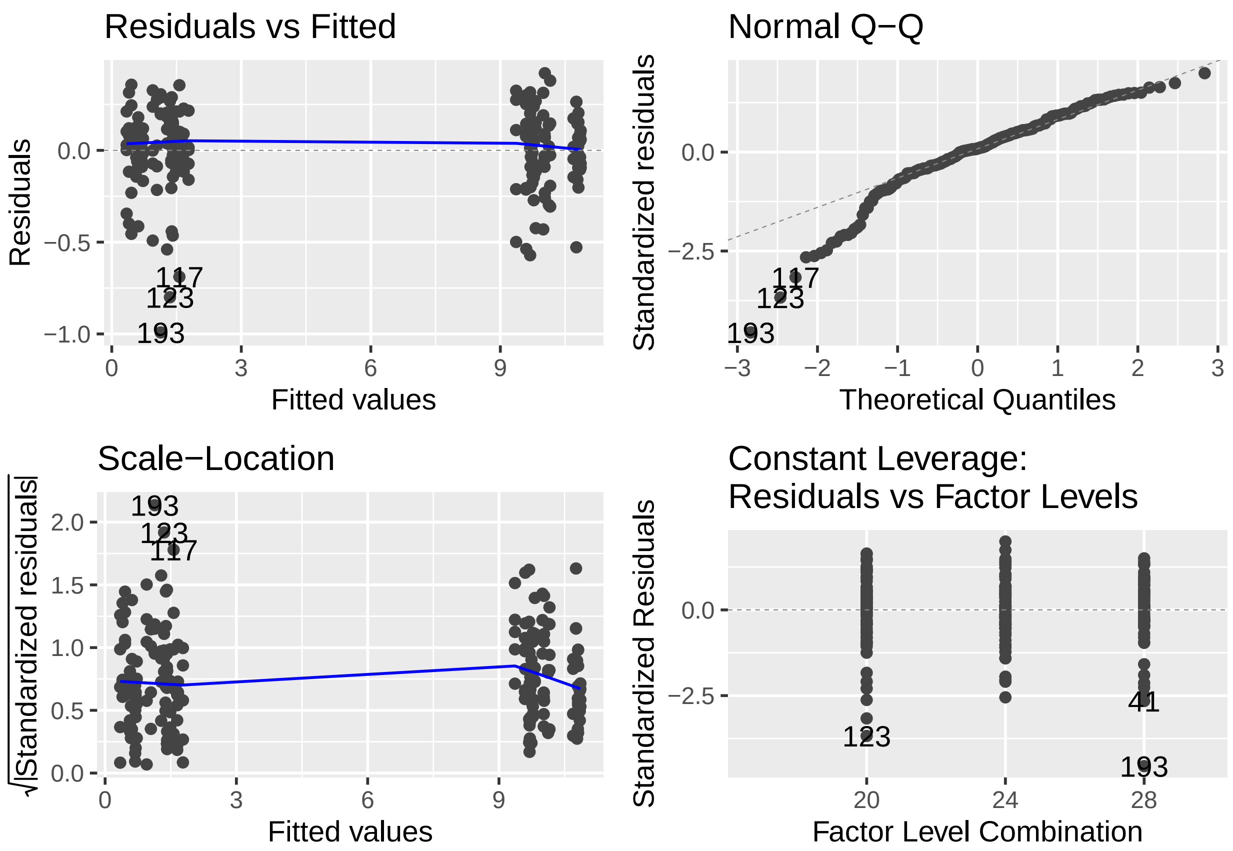




```
rm(abiotic.F,abiotic.noF)

abiotic.outliers.excluded <- abiotic %>%
  mutate(TN.no.outliers = ifelse(F=="1" & TN<5, NA,
                                 ifelse(F=="0" & TN>20,NA,TN))) %>%
  na.omit()

# abiotic.outliers.excluded %>% ggplot(aes(TN.no.outliers)) + geom_histogram() + labs(title = "Distr of
```

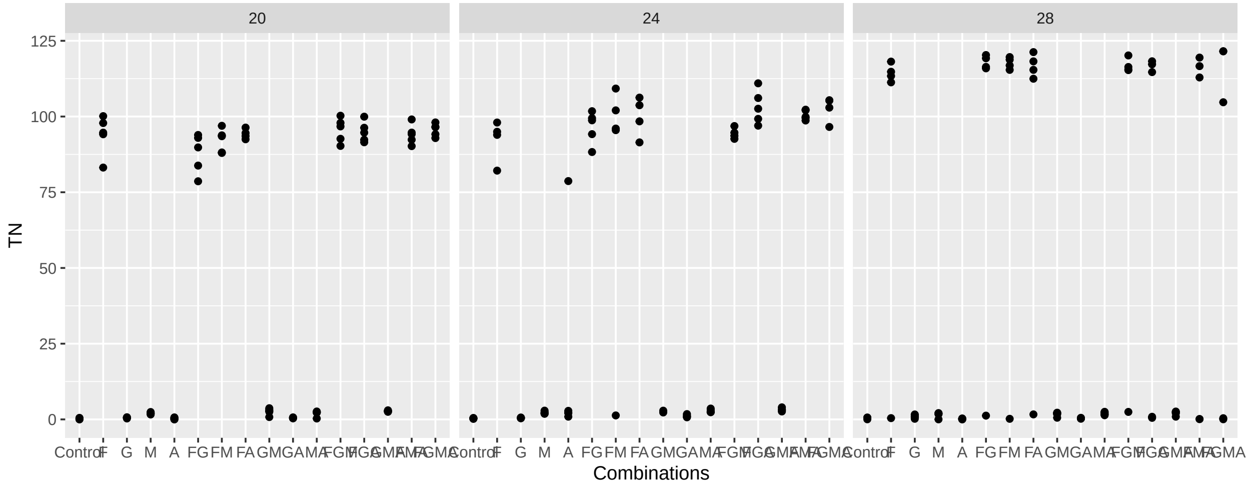
Anova

```
# tn_anova <- lm(TN ~F*G*M*A*T, data = abiotic)

# tn_anova.no.outliers <- lm(TN.no.outliers+0.001 ~ F*G*M*A*T, data = abiotic.outliers.excluded)
# autoplot(tn_anova.no.outliers)
# b <- boxcox(tn_anova.no.outliers, plotit = TRUE, lambda = seq(-1, 2, by = 0.05))

tn_anova.no.outliers <- lm(sqrt(TN.no.outliers) ~ F*G*M*A*T, data = abiotic.outliers.excluded)
autoplot(tn_anova.no.outliers)
```



```r
# anova(tn_anova.no.outliers)
# summary(tn_anova.no.outliers)
```

```r
abiotic %>%
  dplyr::filter(Stressor!="ori") %>%
  ggplot(aes(Combinations, TN)) +
  geom_point() +
  facet_wrap(~Temperature_factor)
```



Figure

```
#total nitrogen

plot_TN_means <- abiotic_means %>%
  ggplot(aes(x=T, y=TN_mean, col=Combinations, group=Combinations)) +
  geom_errorbar(aes(ymin=TN_mean-TN_SE, ymax=TN_mean+TN_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4))+
  geom_line(position = position_dodge(width = 0.4)) +
  geom_point(aes(shape=Stressor), size = 3, position = position_dodge(width = 0.4)) +
  theme_bw() +
  theme(panel.grid = element_blank(),
        axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"),
        legend.position = "none")+
  scale_colour_viridis_d(direction = -1)+
  labs(x="Temperature", y="TN (mean)")

plot_TN_means_numeric <- abiotic_means %>%
  ggplot(aes(x=Stressor_numeric, y=TN_mean, col=T)) +
  geom_point(aes(shape=T), size = 3, position = position_dodge(width = 0.4)) +
  geom_errorbar(aes(ymin=TN_mean-TN_SE, ymax=TN_mean+TN_SE),
                width=0.1,alpha=0.4,position = position_dodge(width = 0.4))+
  theme_bw() +
  theme(panel.grid = element_blank(),
        axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold"),
        legend.position = "none")+
  labs(y="TN (mean)", x="Number of drivers ")

#plot nitrogen

plot_TN_means +theme(legend.position = "bottom",
                     legend.box = "vertical")  + plot_TN_means_numeric + theme(legend.position = "botto
                                                                               legend.box = "vertical")
```

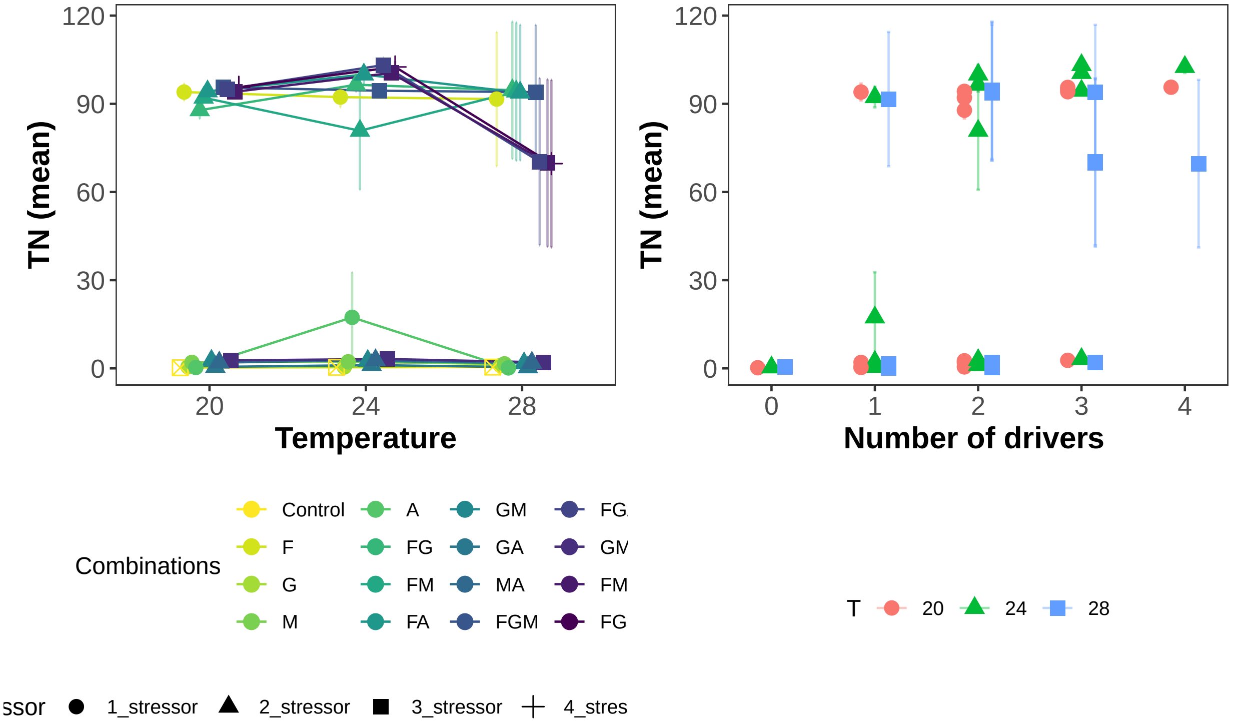
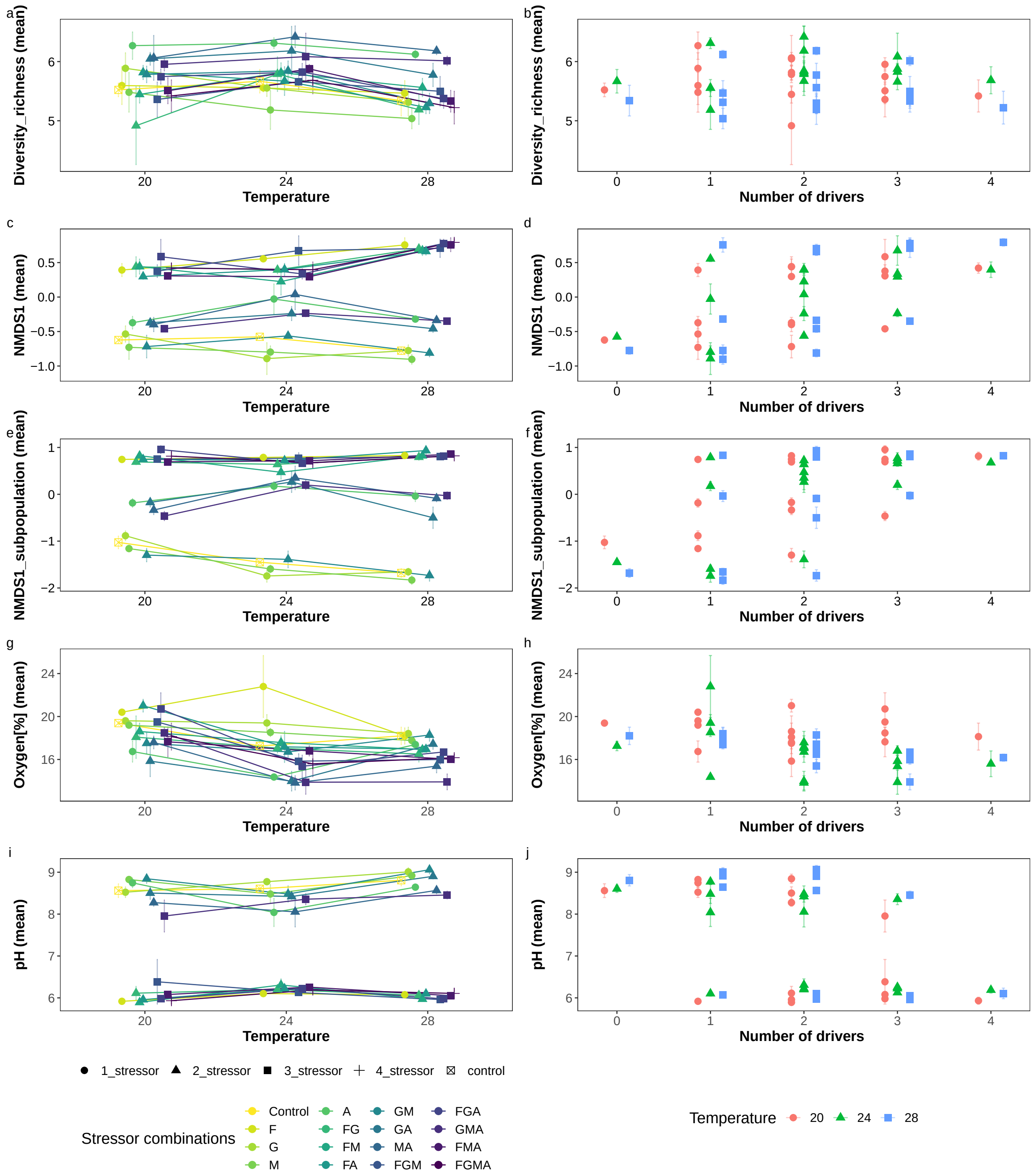


Fig 2: Response variables as a function of stressors, their combinations and temperature

Manuscript Fig 2

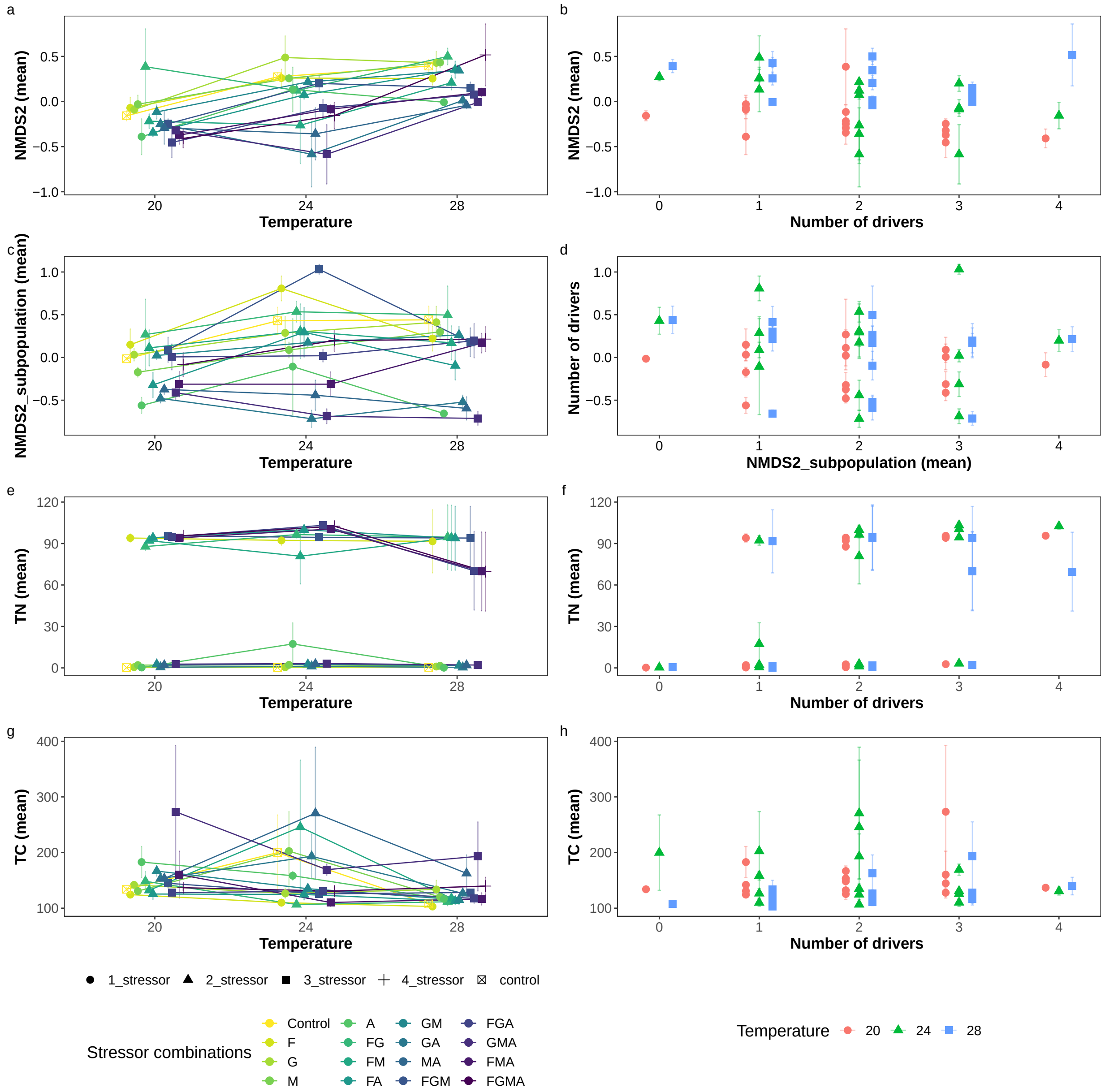
```
plot_pH_means <- plot_pH_means +
  theme(legend.position = "bottom",
        legend.box = "vertical",
        legend.text = element_text(size=12),
        legend.title = element_text(size=15)) +
  labs(col = "Stressor combinations",shape= "")

plot_pH_means_numeric <- plot_pH_means_numeric +
  theme(legend.position = "bottom",
        legend.box = "vertical",
        legend.text = element_text(size=12),
        legend.title = element_text(size=15)) +
  labs(col = "Temperature", shape= "Temperature")

plot_diversity_means +  plot_diversity_means_numeric + nmds1_plot_temperature +
  plot_NMDS1_means_numeric + nmds1_plot_temperature_subset + plot_NMDS1_means_numeric_subset +
  plot_oxygen_means + plot_oxygen_numeric_means + plot_pH_means + plot_pH_means_numeric +
  plot_annotation(tag_levels = "a") + plot_layout(ncol = 2)
```



Supplementary fig with remaining variables

```
plot_TC_means <- plot_TC_means +
  theme(legend.position = "bottom",
        legend.box = "vertical",
        legend.text = element_text(size=12),
```

```r
        legend.title = element_text(size=15)) +
  labs(col = "Stressor combinations", shape = "")

plot_TC_means_numeric <- plot_TC_means_numeric +
  theme(legend.position = "bottom",
        legend.box = "vertical",
        legend.text = element_text(size=12),
        legend.title = element_text(size=15))+
  labs(col = "Temperature", shape = "Temperature")

NMDS2_plot_temperature + plot_NMDS2_means_numeric + NMDS2_plot_temperature_subset +
  plot_NMDS2_means_numeric_subset + plot_TN_means + plot_TN_means_numeric +
  plot_TC_means + plot_TC_means_numeric +
  plot_annotation(tag_levels = "a") + plot_layout(ncol = 2)
```



Combined analysis of all response variables (heatmap)

Calculation: extraction of F- and t-values from fitted models

```
#NMDS1
nmds1 <- extract_test_values(anova_NMDS1, "NMDS1")
#NMDS2
nmds2 <- extract_test_values(anova_NMDS2, "NMDS2")
#NMDS1_CB_SB /1.8
nmds1_mixed <- extract_test_values(anova_NMDS1_mixed, "NMDS1_CB_SB")
#NMDS2_CB_SB
```

```
nmds2_mixed <- extract_test_values(anova_NMDS2_mixed, "NMDS2_CB_SB")
#Richness Shannon
richness <- extract_test_values(anova_diversity, "Shannon_richness")
#Tychonema Cyanobacteria /2.2
tyochonea <- extract_test_values(anova_tychonea, "Tychonema")
#Sulfuricurvum /1.27
sulfuri <- extract_test_values(anova_sulfuri, "Sulfuricurvum")
#Oxygen /9.26
oxygen <- extract_test_values(oxygen_anova, "oxygen")
#pH /2.62
pH <- extract_test_values(pH_anova, "pH")
#TN /93.77
tn <- extract_test_values(tn_anova.no.outliers, "tn")
#TC /65
tc <- extract_test_values(tc_anova, "tc")

# all abiotic and all heat
all_abiotic <- rbind(oxygen, pH, tn, tc)
all_heat <- rbind(tyochonea,sulfuri,nmds1,nmds2,nmds1_mixed,
                  nmds2_mixed,richness,oxygen,pH,tn,tc) %>%
  reorder_levels(variable,order=c("Tychonema","Sulfuricurvum","NMDS1","NMDS2",
                                  "NMDS1_CB_SB","NMDS2_CB_SB","Shannon_richness",
                                  "oxygen", "pH", "tn","tc"))
```

Fig 3: heat map of all factors

Manuscript figure 3

```
all_heat %>%
  filter(term != "(Intercept)", term!="Residuals")%>%
  ggplot(aes(x = variable, y = treatment_combination, fill=color)) +
  geom_tile() +
  scale_fill_gradient2(low="navy", mid="white", high="red", midpoint=0,
                       breaks=c(-1,0,1), na.value="lightgrey", limits=c(-1,1)) +
  theme_bw()+
  theme(axis.text.x = element_text(angle = 50, hjust = 1,size=10),
        axis.text.y = element_text(angle = 0, hjust = 1,size=10),
        strip.placement = "inside") +
  labs(y="Model term [main effect or interaction]")
```

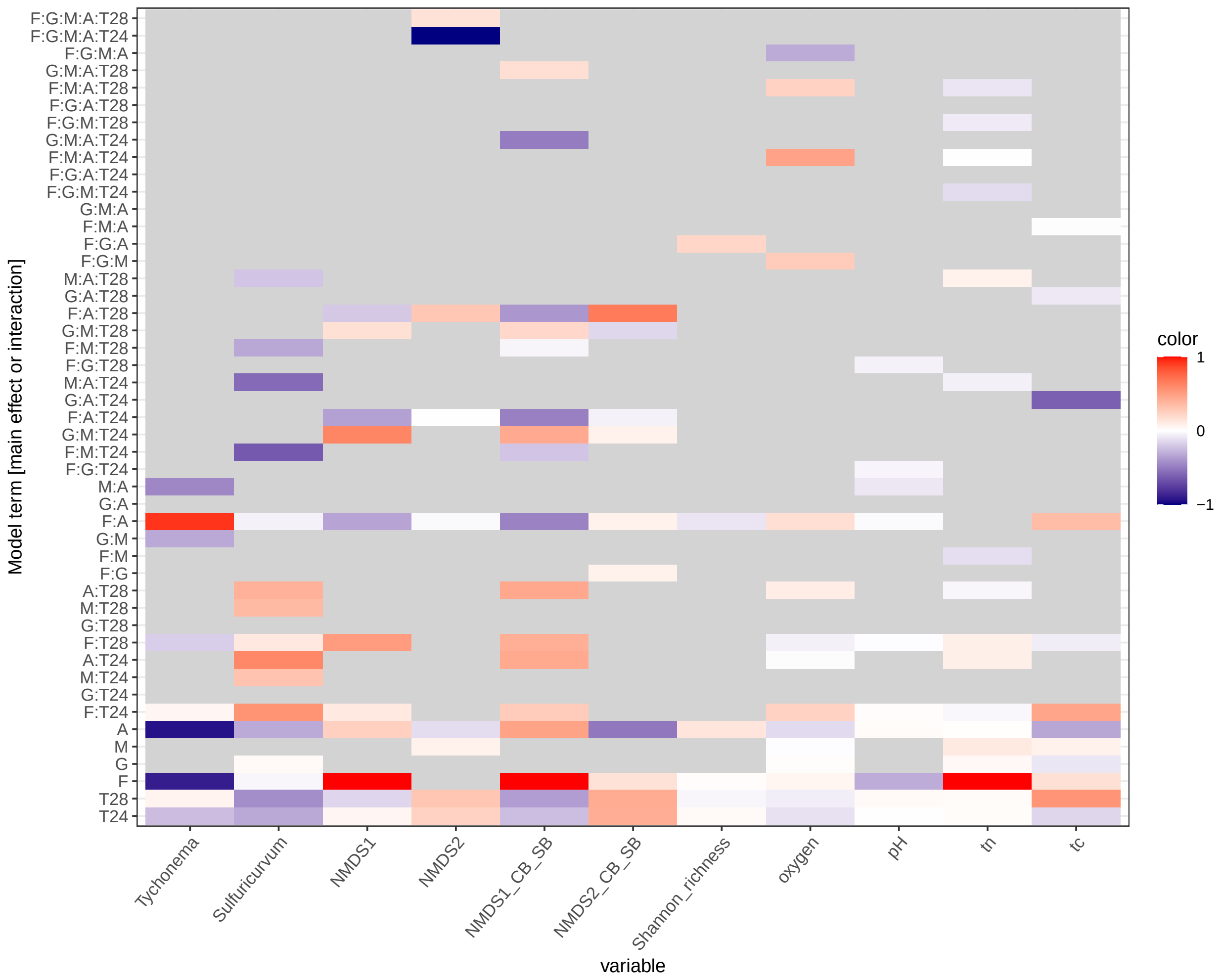
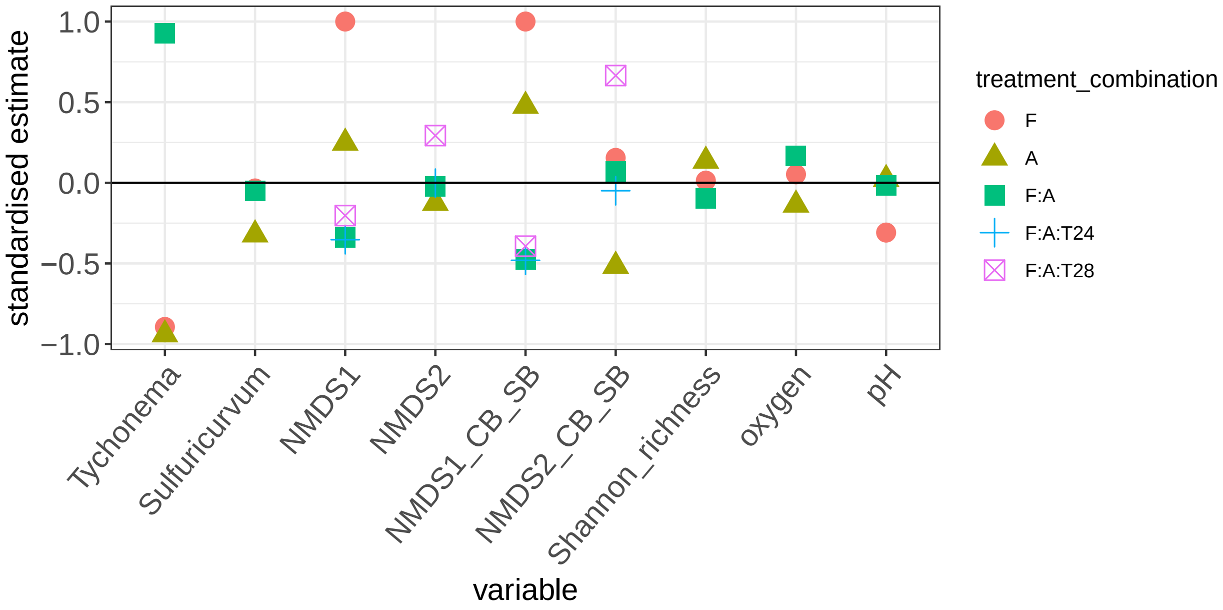


Fig 6: visualization of F:A with all factors

Manuscript figure 6

```
all_heat %>%
  filter(treatment_combination %in% c("F","A","F:A","F:A:T24","F:A:T28"),
         variable != "tn", variable != "tc", color != "NA") %>%
  ggplot()+
  geom_point(aes(x=variable, y=color, shape=treatment_combination,
                 col=treatment_combination), size=4)+
  geom_hline(yintercept=0) +
  theme_bw() +
  theme(axis.text=element_text(size=14),
        axis.title=element_text(size=14),
        axis.text.x = element_text(angle = 50, hjust = 1)) +
  labs(y="standardised estimate")
```



Effect of Number of drivers

The analysis is repeated with the Number of drivers included as a continuous variable

Calculations

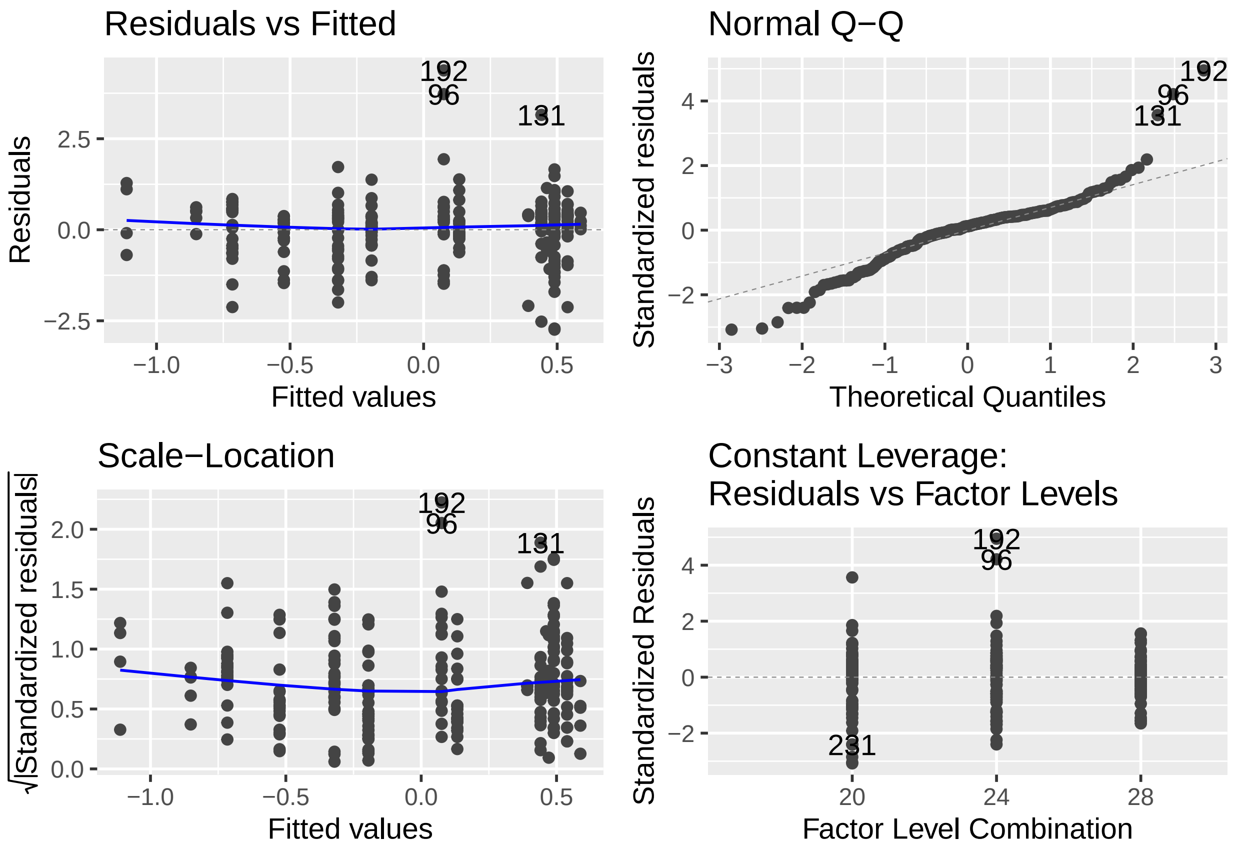
```
abiotic<- ps_relative@sam_data %>%
  as_tibble() %>%
  select(Treatment,Temperature, Combinations, Temperature_factor,
         Lable, Name, F, G, M, A, Oxygen, TN, TC, pH,
         Stressor_numeric, Stressor) %>%
  filter(Treatment!="ori")  %>%
  mutate(T2=Temperature^2,T=as.factor(Temperature)) %>%
  select(-Temperature)
```

```
## Warning in class(x) <- c(setdiff(subclass, tibble_class), tibble_class): Setting
## class(x) to multiple strings ("tbl_df", "tbl", ...); result will no longer be an
## S4 object
```

```
# abiotic$Oxygen <- (abiotic$Oxygen-mean(abiotic$Oxygen))/sd(abiotic$Oxygen)
# abiotic$pH <- (abiotic$pH-mean(abiotic$pH))/sd(abiotic$pH)
# abiotic$TN <- (abiotic$TN-mean(abiotic$TN))/sd(abiotic$TN)
# abiotic$TC <- (abiotic$TC-mean(abiotic$TC))/sd(abiotic$TC)
# mds_whole_res$NMDS1 <- (mds_whole_res$NMDS1-mean(mds_whole_res$NMDS1))/sd(mds_whole_res$NMDS1)
# mds_whole_res$NMDS2 <- (mds_whole_res$NMDS2-mean(mds_whole_res$NMDS2))/sd(mds_whole_res$NMDS2)
# mds_whole_res_subset$NMDS1 <- (mds_whole_res_subset$NMDS1-mean(mds_whole_res_subset$NMDS1))/
#   sd(mds_whole_res_subset$NMDS1)
# mds_whole_res_subset$NMDS2 <- (mds_whole_res_subset$NMDS2-mean(mds_whole_res_subset$NMDS2))/
#   sd(mds_whole_res_subset$NMDS2)
# div_raw$Shannon <- (div_raw$Shannon-mean(div_raw$Shannon))/sd(div_raw$Shannon)
# df_cyano$Abundance <- (df_cyano$Abundance-mean(df_cyano$Abundance))/sd(df_cyano$Abundance)
# df_sulfuri$Abundance <- (df_sulfuri$Abundance-mean(df_sulfuri$Abundance))/sd(df_sulfuri$Abundance)

# real.rt <- function(x, root=3) {
#     sign(x) * abs(x)^(1/root)
# }
```

Oxygen

```
#oxygen
oxygen_anova_numeric <- lm(scale(Oxygen) ~T+T:Stressor_numeric, data=abiotic)

oxy_slope<- tidy(oxygen_anova_numeric ,conf.int = TRUE) %>%
  filter(term!="(Intercept)", term!="T24", term !="T28")

oxy_slope$organisation="ecosystem"
oxy_slope$variable="oxygen"
oxy_slope$Fertilizer <- "Yes"

autoplot(oxygen_anova_numeric)
```



```
## getting f test parameters
oxygen_anova2_numeric <- lm(scale(Oxygen) ~T*Stressor_numeric, data=abiotic)
anova(oxygen_anova2_numeric)
```

```
## Analysis of Variance Table
##
## Response: scale(Oxygen)
##                      Df  Sum Sq Mean Sq F value    Pr(>F)
## T                     2  28.708 14.3542 17.8538 6.357e-08 ***
## Stressor_numeric      1  15.147 15.1473 18.8402 2.148e-05 ***
## T:Stressor_numeric    2   5.247  2.6237  3.2634   0.04008 *
## Residuals           225 180.897  0.8040
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
summary(oxygen_anova2_numeric)
```

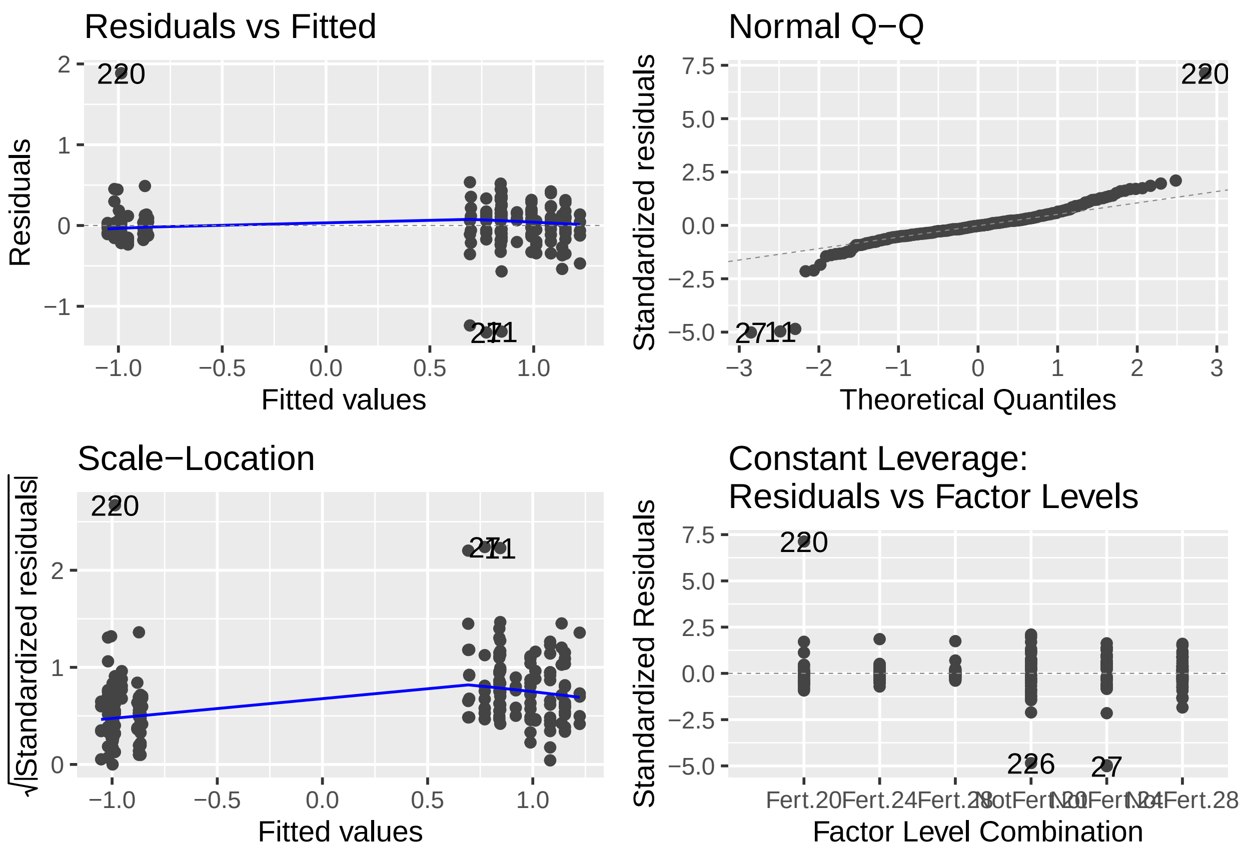
```
##
## Call:
## lm(formula = scale(Oxygen) ~ T * Stressor_numeric, data = abiotic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7439 -0.4290  0.1063  0.4152  4.3762
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.5880     0.2270   2.590   0.0102 *
## T24                     -0.1167     0.3238  -0.360   0.7190
## T28                     -0.1263     0.3191  -0.396   0.6925
## Stressor_numeric        -0.0490     0.1009  -0.486   0.6277
## T24:Stressor_numeric    -0.3468     0.1451  -2.390   0.0177 *
## T28:Stressor_numeric    -0.2793     0.1422  -1.964   0.0508 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.8967 on 225 degrees of freedom
## Multiple R-squared:  0.2135, Adjusted R-squared:  0.196
## F-statistic: 12.21 on 5 and 225 DF,  p-value: 1.746e-10
```

Comment UD: diagnostics are ok

pH

```
#pH

# pH_anova_numeric_old <- lm(pH ~T + T:Stressor_numeric, data=abiotic)
# b <- boxcox(pH_anova_numeric_old, plotit = TRUE, lambda = seq(-3, 3, by = 0.1))
#
# pH_anova_numeric_old2 <- lm(scale(log10(pH)) ~T + T:Stressor_numeric, data=abiotic)
```

```r
# abiotic$pH_stand <- scale(abiotic$pH)
# abiotic$pH_stand_transformed <- abiotic$pH_stand^(1/3)
# abiotic$pH_stand_transformed2 <- nthroot(abiotic$pH_stand,3)
# abiotic$pH_transformed <- abiotic$pH^(1/3)
# abiotic$pH_transformed_stand <- scale(abiotic$pH_transformed)
#
# abiotic %>% ggplot(aes(pH)) + geom_histogram() + labs(title="original") +
# abiotic %>% ggplot(aes(pH_stand)) + geom_histogram() + labs(title="standardized") +
# abiotic %>% ggplot(aes(pH_stand_transformed)) + geom_histogram() + labs(title="standardized then tran
# abiotic %>% ggplot(aes(pH_transformed)) + geom_histogram() + labs(title="transformed") +
# abiotic %>% ggplot(aes(pH_transformed_stand)) + geom_histogram() + labs(title="transformed then stand

abiotic$Fertilizer <- ifelse(abiotic$F==1, "Fert", "NotFert")
abiotic$F_and_T <- with(abiotic, interaction(Fertilizer,T))
pH_anova_numeric <- lm(scale(pH) ~-1+F_and_T + F_and_T:Stressor_numeric, data=abiotic)
# pH_anova_numeric1 <- lm(pH_stand ~-1+F_and_T*Stressor_numeric, data=abiotic)
# all.equal(predict(pH_anova_numeric),predict(pH_anova_numeric1))

pH_slope <- tidy(pH_anova_numeric,conf.int = TRUE) %>%
  slice_tail(n = 6)
pH_slope$term <- rep(c("T20:Stressor_numeric","T24:Stressor_numeric","T28:Stressor_numeric"),each=2)

pH_slope$organisation="ecosystem"
pH_slope$variable="pH"
pH_slope$Fertilizer <- rep(c("Yes","No"),3)
autoplot(pH_anova_numeric)
```



```r
## getting f test parameters

pH_anova2_numeric <- lm(scale(pH) ~ F*T*Stressor_numeric, data=abiotic)
anova(pH_anova2_numeric)
```

```
## Analysis of Variance Table
##
## Response: scale(pH)
##                       Df  Sum Sq Mean Sq   F value    Pr(>F)
## F                      1 210.837 210.837 2913.5480 < 2.2e-16 ***
## T                      2   0.498   0.249    3.4406  0.033788 *
## Stressor_numeric       1   0.318   0.318    4.3913  0.037271 *
## F:T                    2   1.840   0.920   12.7111 5.987e-06 ***
## F:Stressor_numeric     1   0.524   0.524    7.2475  0.007649 **
## T:Stressor_numeric     2   0.022   0.011    0.1493  0.861380
## F:T:Stressor_numeric   2   0.114   0.057    0.7843  0.457709
## Residuals            219  15.848   0.072
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(pH_anova2_numeric)
```

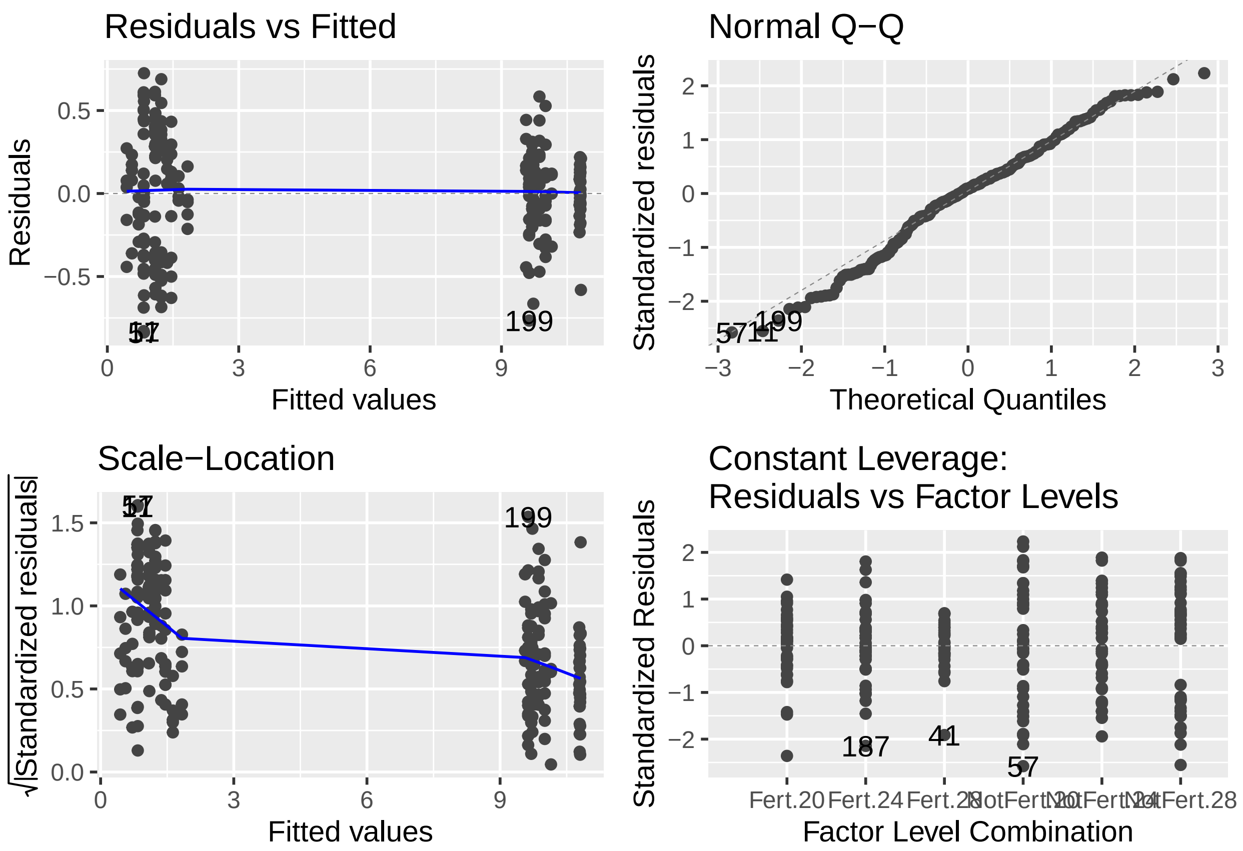
```
##
## Call:
## lm(formula = scale(pH) ~ F * T * Stressor_numeric, data = abiotic)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.32390 -0.09996 -0.00422  0.08922  1.88545
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.137097   0.086160  13.197   <2e-16 ***
## F                          -2.221978   0.156976 -14.155   <2e-16 ***
## T24                        -0.217964   0.121079  -1.800   0.0732 .
## T28                         0.085857   0.121079   0.709   0.4790
## Stressor_numeric           -0.148028   0.049325  -3.001   0.0030 **
## F:T24                       0.415202   0.227989   1.821   0.0699 .
## F:T28                       0.008239   0.220823   0.037   0.9703
## F:Stressor_numeric          0.180765   0.069756   2.591   0.0102 *
## T24:Stressor_numeric        0.074226   0.070086   1.059   0.2907
## T28:Stressor_numeric        0.077850   0.069607   1.118   0.2646
## F:T24:Stressor_numeric -0.099286   0.101191  -0.981   0.3276
## F:T28:Stressor_numeric -0.114043   0.098439  -1.159   0.2479
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.269 on 219 degrees of freedom
## Multiple R-squared:  0.9311, Adjusted R-squared:  0.9276
## F-statistic:   269 on 11 and 219 DF,  p-value: < 2.2e-16
```

Comment UD: diagnostics are now ok. I changed the model (see above)

TN

```
abiotic.outliers.excluded <- abiotic %>%
  mutate(TN.no.outliers = ifelse(F=="1" & TN<5, NA,
                                 ifelse(F=="0" & TN>20,NA,TN))) %>%
  na.omit()

# tn_anova_numeric <- lm(TN.no.outliers+0.001 ~ 1+F_and_T + F_and_T:Stressor_numeric, data=abiotic.outl
# b <- boxcox(tn_anova_numeric, plotit = TRUE, lambda = seq(-3, 3, by = 0.1))
# autoplot(tn_anova_numeric)
tn_anova_numeric <- lm(sqrt(TN.no.outliers) ~ 1+F_and_T + F_and_T:Stressor_numeric, data=abiotic.outlie

tn_slope <- tidy(tn_anova_numeric,conf.int = TRUE) %>%
  slice_tail(n = 6)
tn_slope$term <- rep(c("T20:Stressor_numeric","T24:Stressor_numeric","T28:Stressor_numeric"),each=2)

tn_slope$organisation="ecosystem"
tn_slope$variable="total nitrogen"
tn_slope$Fertilizer <- rep(c("Yes","No"),3)
autoplot(tn_anova_numeric)
```



```
## getting f test parameters

tn_anova2_numeric <- lm(scale(sqrt(TN.no.outliers)) ~ F*T*Stressor_numeric, data=abiotic.outliers.exclu
anova(tn_anova2_numeric)
```

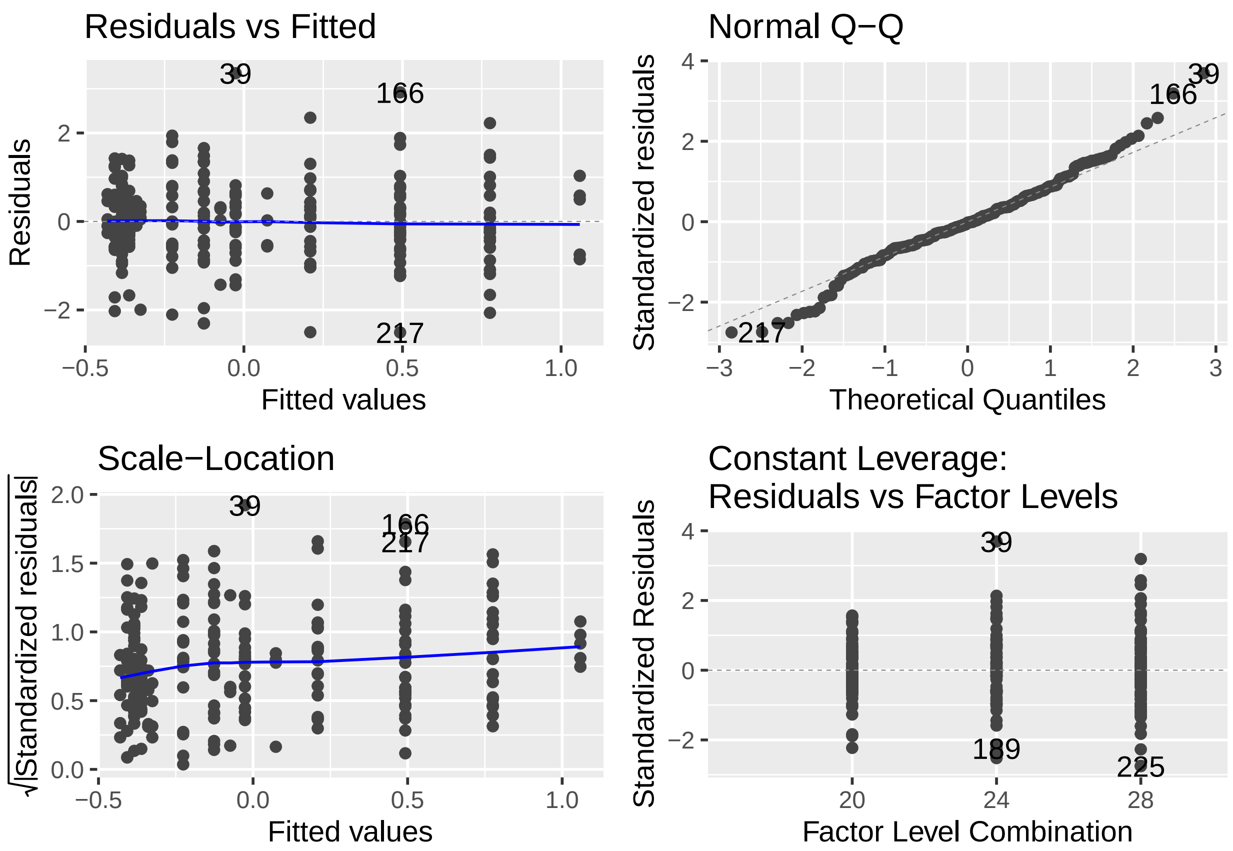
```
## Analysis of Variance Table
##
## Response: scale(sqrt(TN.no.outliers))
##                       Df  Sum Sq Mean Sq    F value    Pr(>F)
## F                      1 214.175 214.175 40286.4249 < 2.2e-16 ***
## T                      2   0.381   0.191    35.8392 4.400e-14 ***
## Stressor_numeric       1   0.361   0.361    67.8431 2.028e-14 ***
## F:T                    2   0.822   0.411    77.2893 < 2.2e-16 ***
## F:Stressor_numeric     1   0.144   0.144    27.0250 4.825e-07 ***
## T:Stressor_numeric     2   0.019   0.010     1.8247    0.1639
## F:T:Stressor_numeric   2   0.003   0.002     0.3020    0.7397
## Residuals            206   1.095   0.005
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(tn_anova2_numeric)
```

```
##
## Call:
## lm(formula = scale(sqrt(TN.no.outliers)) ~ F * T * Stressor_numeric,
##     data = abiotic.outliers.excluded)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.184781 -0.040651  0.006366  0.046991  0.160000
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -1.079542   0.023353 -46.226  < 2e-16 ***
## F                       1.997951   0.042548  46.958  < 2e-16 ***
## T24                     0.060305   0.033027   1.826 0.069305 .
## T28                     0.026150   0.032818   0.797 0.426464
## Stressor_numeric        0.087189   0.013369   6.522 5.28e-10 ***
## F:T24                  -0.039651   0.062176  -0.638 0.524359
## F:T28                   0.256103   0.063228   4.050 7.24e-05 ***
## F:Stressor_numeric     -0.071900   0.018907  -3.803 0.000188 ***
## T24:Stressor_numeric   -0.005034   0.019032  -0.265 0.791657
## T28:Stressor_numeric   -0.027919   0.018867  -1.480 0.140457
## F:T24:Stressor_numeric  0.020593   0.027492   0.749 0.454689
## F:T28:Stressor_numeric  0.014892   0.028061   0.531 0.596188
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07291 on 206 degrees of freedom
## Multiple R-squared:  0.995,  Adjusted R-squared:  0.9947
## F-statistic:  3692 on 11 and 206 DF,  p-value: < 2.2e-16
```

TC

```
#tc
# tc_anova_numeric <- lm(TC ~T+T:Stressor_numeric, data=abiotic)
# b <- boxcox(tc_anova_numeric, plotit = TRUE, lambda = seq(-8, 1.5, by = 0.1))
# lambda <- round(b$x[which.max(b$y)],1)

abiotic$TC_squared_inv <- scale(1/(abiotic$TC^2))

tc_anova_numeric <- lm(TC_squared_inv ~T+T:Stressor_numeric, data=abiotic)

tc_slope <- tidy(tc_anova_numeric,conf.int = TRUE) %>%
  filter(term!="(Intercept)", term!="T24", term !="T28")

tc_slope$organisation="ecosystem"
tc_slope$variable="total carbon"
tc_slope$Fertilizer <- "Yes"
autoplot(tc_anova_numeric)
```



```
## getting f test parameters
tc_anova2_numeric <- lm(scale(TC_squared_inv) ~T*Stressor_numeric, data=abiotic)
anova(tc_anova2_numeric)
```

```
## Analysis of Variance Table
##
## Response: scale(TC_squared_inv)
##                     Df  Sum Sq Mean Sq F value    Pr(>F)
## T                    2  32.154 16.0770 18.9735 2.429e-08 ***
## Stressor_numeric     1   1.250  1.2503  1.4756   0.22574
## T:Stressor_numeric   2   5.944  2.9718  3.5072   0.03163 *
## Residuals          225 190.652  0.8473
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

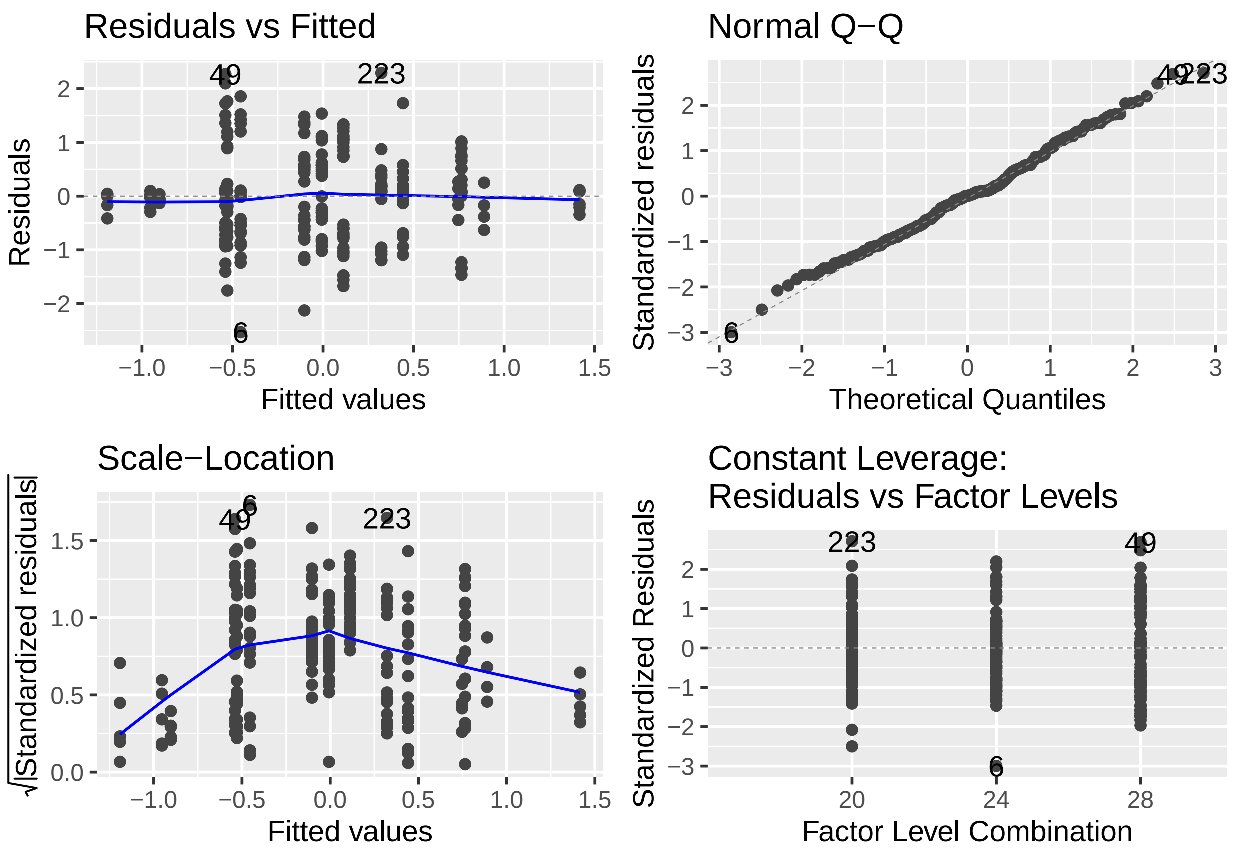
```
summary(tc_anova2_numeric)
```

```
##
## Call:
## lm(formula = scale(TC_squared_inv) ~ T * Stressor_numeric, data = abiotic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5080 -0.5335 -0.0203  0.5233  3.3508
##
```

```
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            -0.33878    0.23308  -1.454   0.1475    
## T24                     0.01293    0.33245   0.039   0.9690    
## T28                     1.39765    0.32754   4.267 2.92e-05 ***
## Stressor_numeric       -0.02276    0.10357  -0.220   0.8263    
## T24:Stressor_numeric    0.12265    0.14895   0.823   0.4111    
## T28:Stressor_numeric   -0.26042    0.14601  -1.784   0.0758 .  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.9205 on 225 degrees of freedom
## Multiple R-squared:  0.1711, Adjusted R-squared:  0.1527
## F-statistic: 9.287 on 5 and 225 DF,  p-value: 4.681e-08
```

NMDS1

```r
#NMDS1
anova_NMDS1_numeric <- lm(scale(NMDS1) ~T+T:Stressor_numeric, data=mds_whole_res)

nmds1_slope <- tidy(anova_NMDS1_numeric,conf.int = TRUE) %>%
  filter(term!="(Intercept)", term!="T24", term !="T28")

nmds1_slope$organisation="community"
nmds1_slope$variable="NMDS1"
nmds1_slope$Fertilizer <- "Yes"
autoplot(anova_NMDS1_numeric)
```



```
## getting f test parameters
anova2_NMDS1_numeric <- lm(scale(NMDS1) ~T*Stressor_numeric, data=mds_whole_res)
anova(anova2_NMDS1_numeric)
```

```
## Analysis of Variance Table
##
## Response: scale(NMDS1)
##                     Df  Sum Sq Mean Sq F value Pr(>F)
## T                    2   1.786   0.893  1.2170 0.2981
## Stressor_numeric     1  60.674  60.674 82.6972 <2e-16 ***
## T:Stressor_numeric   2   2.462   1.231  1.6778 0.1891
## Residuals          225 165.079   0.734
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

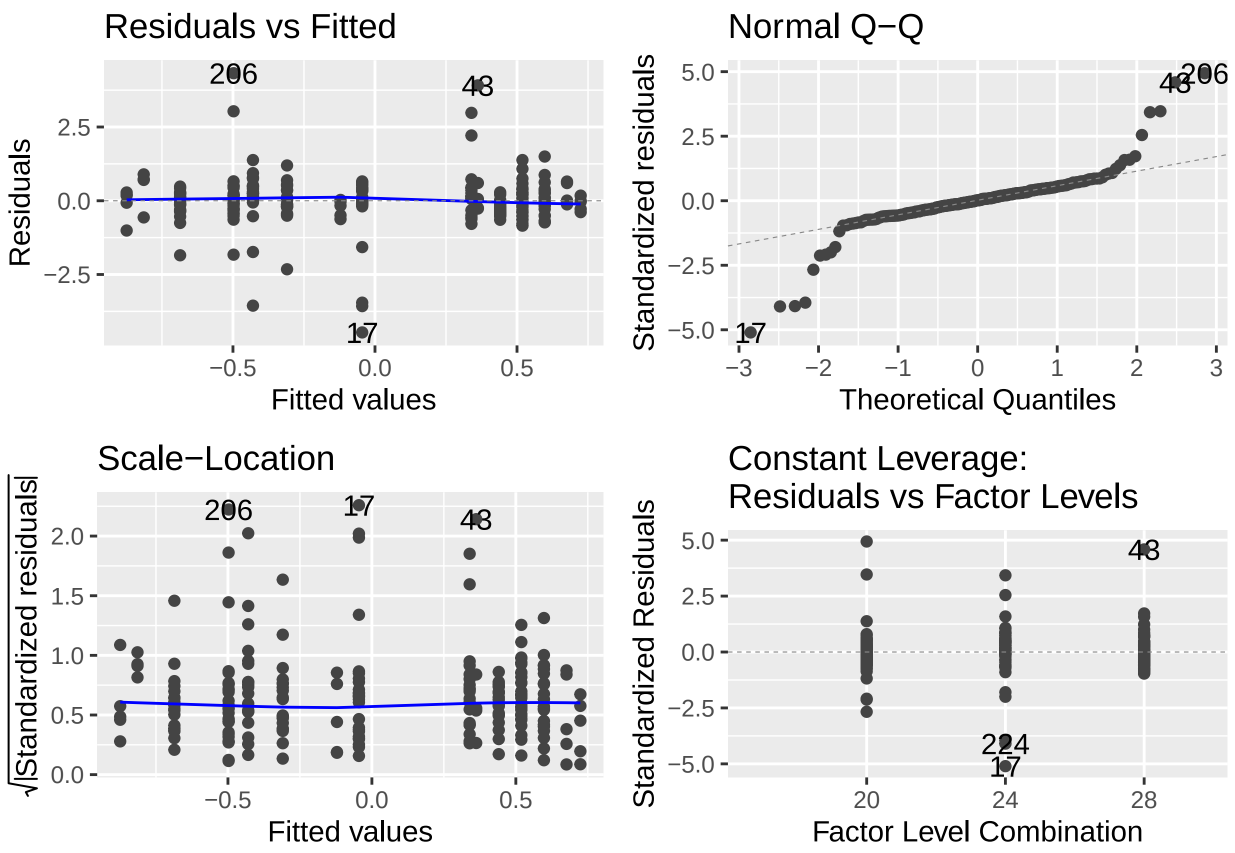
```
summary(anova2_NMDS1_numeric)
```

```
##
## Call:
## lm(formula = scale(NMDS1) ~ T * Stressor_numeric, data = mds_whole_res)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.53242 -0.62042  0.00301  0.54602  2.29343
##
```

```
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             -0.95356    0.21688  -4.397 1.70e-05 ***
## T24                      0.05042    0.30935   0.163   0.8707    
## T28                     -0.23768    0.30478  -0.780   0.4363    
## Stressor_numeric         0.42520    0.09638   4.412 1.59e-05 ***
## T24:Stressor_numeric     0.02299    0.13860   0.166   0.8684    
## T28:Stressor_numeric     0.22673    0.13587   1.669   0.0966 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8566 on 225 degrees of freedom
## Multiple R-squared:  0.2823, Adjusted R-squared:  0.2663 
## F-statistic:  17.7 on 5 and 225 DF,  p-value: 8.844e-15
```

NMDS2

```
#NMDS2
anova_NMDS2_numeric <- lm(scale(NMDS2) ~T+T:Stressor_numeric, data=mds_whole_res)

nmds2_slope <- tidy(anova_NMDS2_numeric,conf.int = TRUE) %>%
  filter(term!="(Intercept)", term!="T24", term !="T28")

nmds2_slope$organisation="community"
nmds2_slope$variable="NMDS2"
nmds2_slope$Fertilizer <- "Yes"
autoplot(anova_NMDS2_numeric)
```



```
## getting f test parameters
anova2_NMDS2_numeric <- lm(scale(NMDS2) ~T*Stressor_numeric, data=mds_whole_res)
anova(anova2_NMDS2_numeric)
```

```
## Analysis of Variance Table
##
## Response: scale(NMDS2)
##                     Df  Sum Sq Mean Sq F value    Pr(>F)
## T                    2  41.164 20.5818  26.521 4.549e-11 ***
## Stressor_numeric     1  10.554 10.5535  13.599 0.0002837 ***
## T:Stressor_numeric   2   3.674  1.8369   2.367 0.0960950 .
## Residuals          225 174.609  0.7760
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

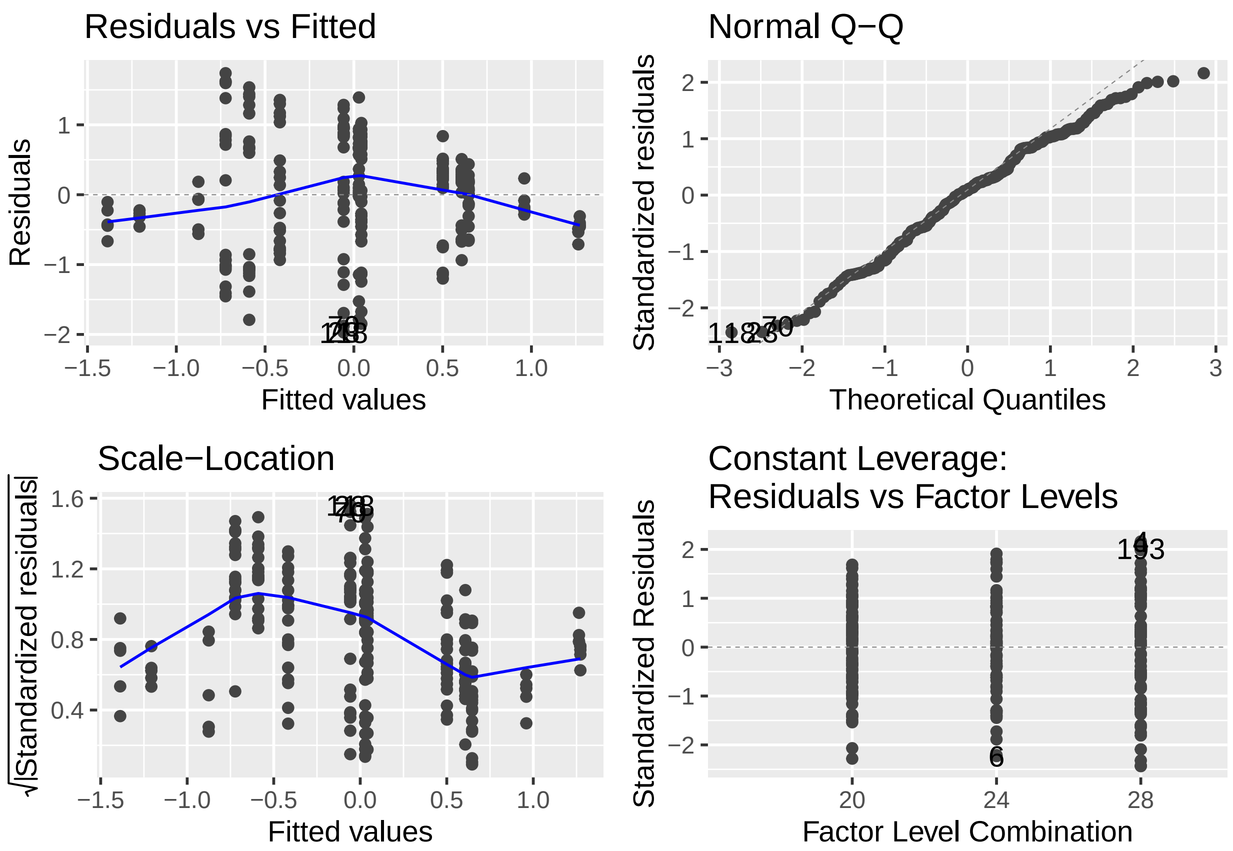
```
summary(anova2_NMDS2_numeric)
```

```
##
## Call:
## lm(formula = scale(NMDS2) ~ T * Stressor_numeric, data = mds_whole_res)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4642 -0.3158  0.0133  0.3477  4.3245
##
```

```
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -0.12132    0.22306  -0.544  0.58705
## T24                     0.84548    0.31815   2.657  0.00844 **
## T28                     0.79734    0.31346   2.544  0.01164 *
## Stressor_numeric       -0.18824    0.09912  -1.899  0.05883 .
## T24:Stressor_numeric -0.19634    0.14255  -1.377  0.16977
## T28:Stressor_numeric  0.10980    0.13973   0.786  0.43283
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8809 on 225 degrees of freedom
## Multiple R-squared:  0.2408, Adjusted R-squared:  0.224
## F-statistic: 14.28 on 5 and 225 DF,  p-value: 3.881e-12
```

NMDS1_subset

```
#NMDS1_subset
anova_NMDS1_mixed_numeric <- lm(scale(NMDS1) ~T+T:Stressor_numeric, data=mds_whole_res_subset)

nmds1_subset_slope <- tidy(anova_NMDS1_mixed_numeric,conf.int = TRUE) %>%
  filter(term!="(Intercept)", term!="T24", term !="T28")

nmds1_subset_slope$organisation="community"
nmds1_subset_slope$variable="NMDS1_subpopulation"
nmds1_subset_slope$Fertilizer <- "Yes"
autoplot(anova_NMDS1_mixed_numeric)
```



Residuals vs Fitted

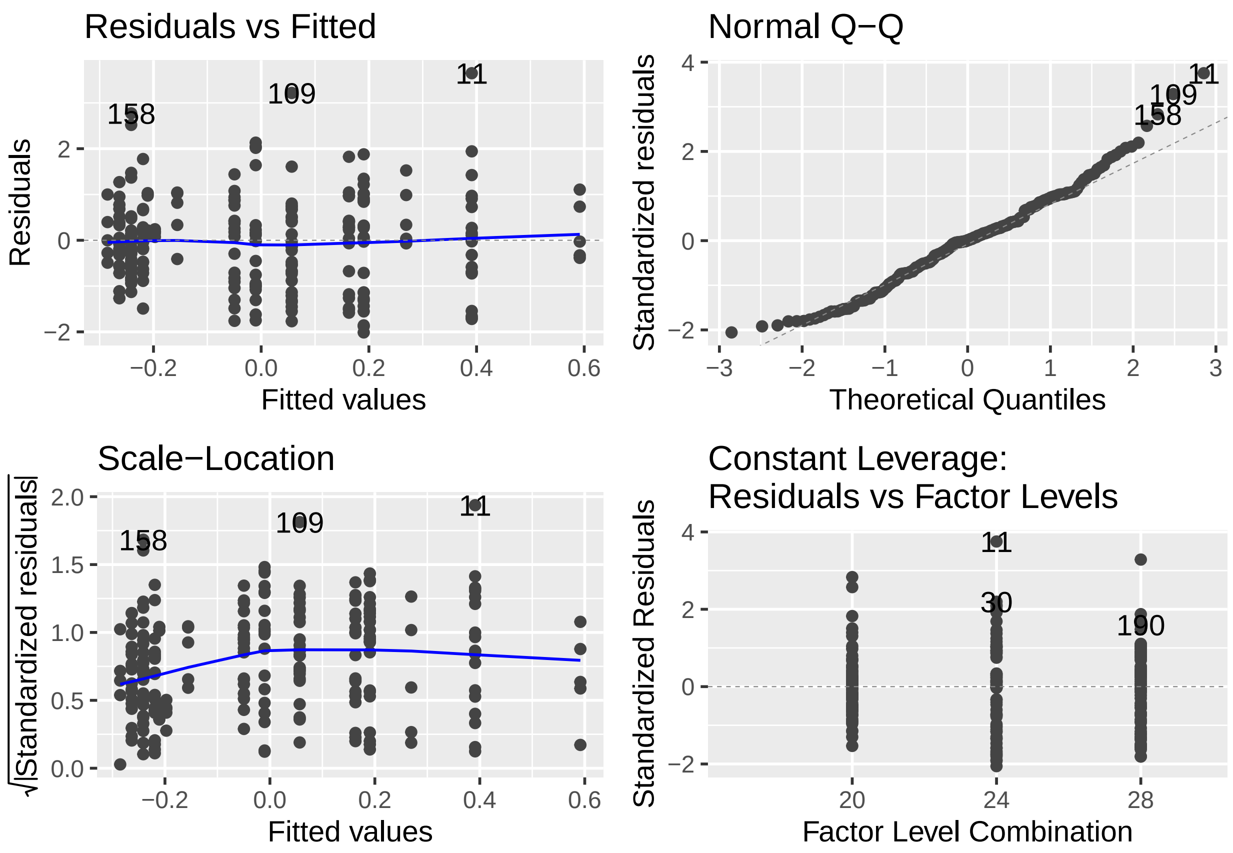
Normal Q−Q

Scale−Location

Constant Leverage:
Residuals vs Factor Levels

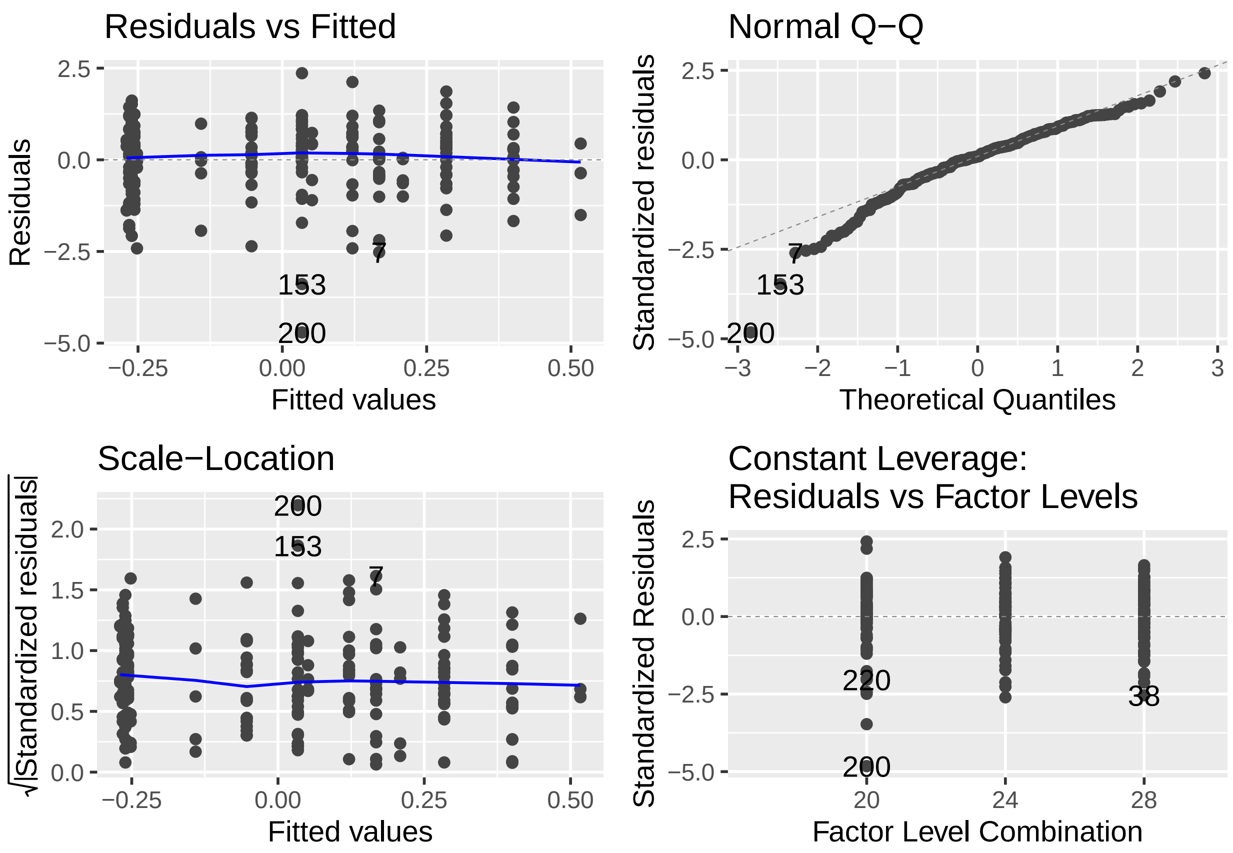
NMDS2_subset

```
#NMDS2_subset
anova_NMDS2_mixed_numeric <- lm(scale(NMDS2) ~T+T:Stressor_numeric, data=mds_whole_res_subset)

nmds2_subset_slope <- tidy(anova_NMDS2_mixed_numeric,conf.int = TRUE) %>%
  filter(term!="(Intercept)", term!="T24", term !="T28")

nmds2_subset_slope$organisation="community"
nmds2_subset_slope$variable="NMDS2_subpopulation"
nmds2_subset_slope$Fertilizer <- "Yes"
autoplot(anova_NMDS2_mixed_numeric)
```



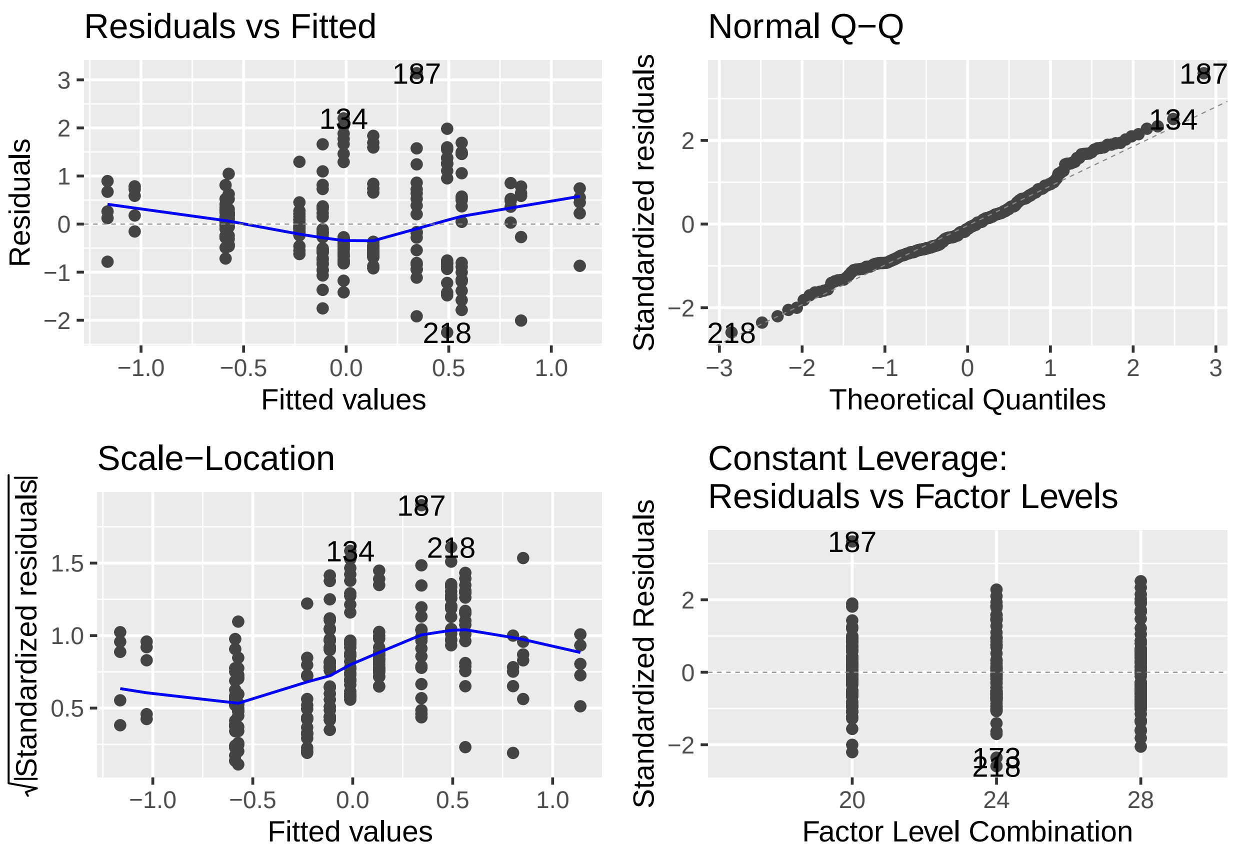
Shannon index

```
#Shannon
anova_diversity_numeric <- lm(scale(Shannon) ~T+T:Stressor_numeric, data=div_raw)

shannon_slope <- tidy(anova_diversity_numeric,conf.int = TRUE) %>%
  filter(term!="(Intercept)", term!="T24", term !="T28")

shannon_slope$organisation="community"
shannon_slope$variable="species_richness"
shannon_slope$Fertilizer <- "Yes"
autoplot(anova_diversity_numeric)
```



Tychonema

```
#Tychonema
# anova_tychonea_numeric <- lm(Abundance ~T+T:Stressor_numeric, data=df_cyano)
# b <- boxcox(anova_tychonea_numeric, plotit = TRUE, lambda = seq(-3, 3, by = 0.1))
anova_tychonea_numeric <- lm(scale(logit(Abundance)) ~T+T:Stressor_numeric, data=df_cyano)

tychonema_slope <- tidy(anova_tychonea_numeric,conf.int = TRUE) %>%
  filter(term!="(Intercept)", term!="T24", term !="T28")

tychonema_slope$organisation="species"
tychonema_slope$variable="Tychonema"
tychonema_slope$Fertilizer <- "Yes"
autoplot(anova_tychonea_numeric)
```



update.. i changed it to log10

Sulfuri

```
#Sulfuri
anova_sulfuri_numeric <- lm(scale(Abundance^(1/3)) ~T+T:Stressor_numeric, data=df_sulfuri)

sulfuri_slope <- tidy(anova_sulfuri_numeric,conf.int = TRUE) %>%
  filter(term!="(Intercept)", term!="T24", term !="T28")

sulfuri_slope$organisation="species"
sulfuri_slope$variable="Sulfuricurvum"
sulfuri_slope$Fertilizer <- "Yes"
autoplot(anova_sulfuri_numeric)
```

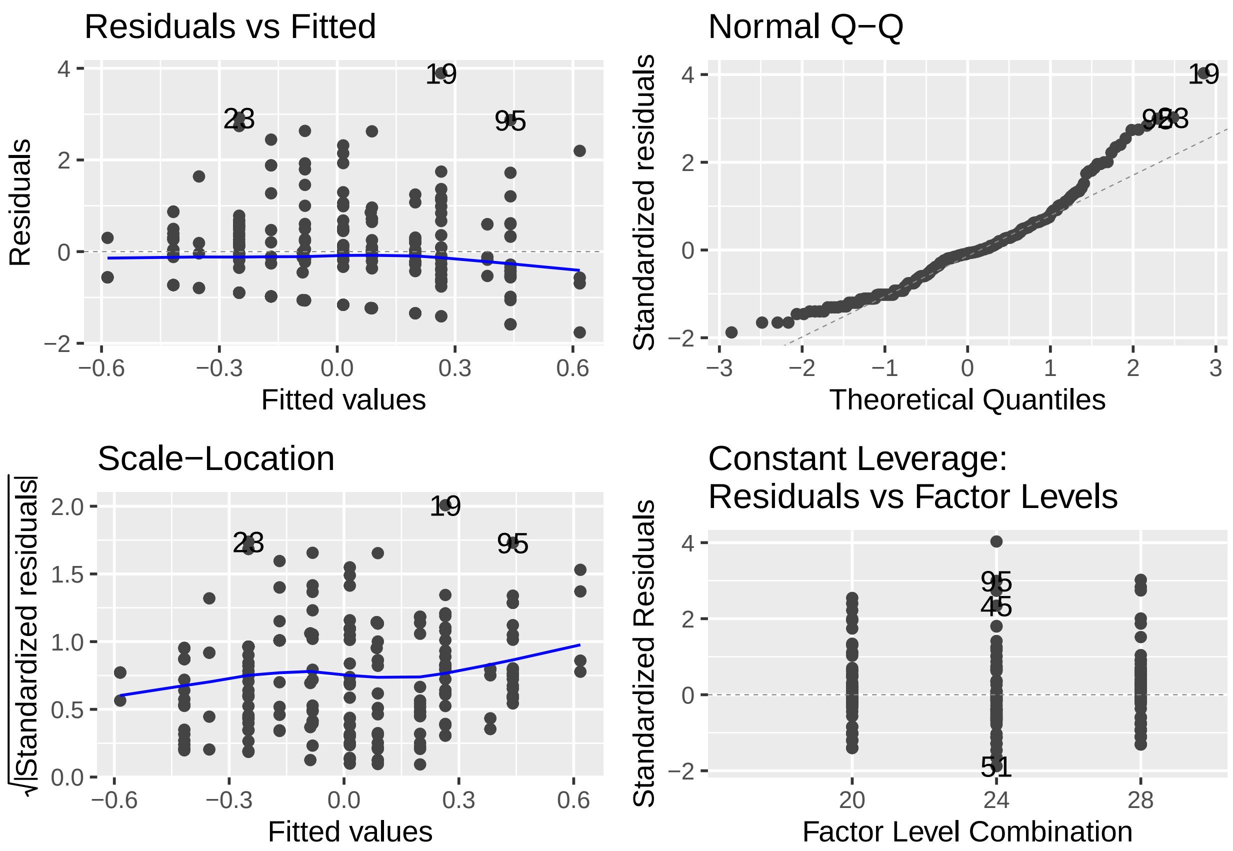
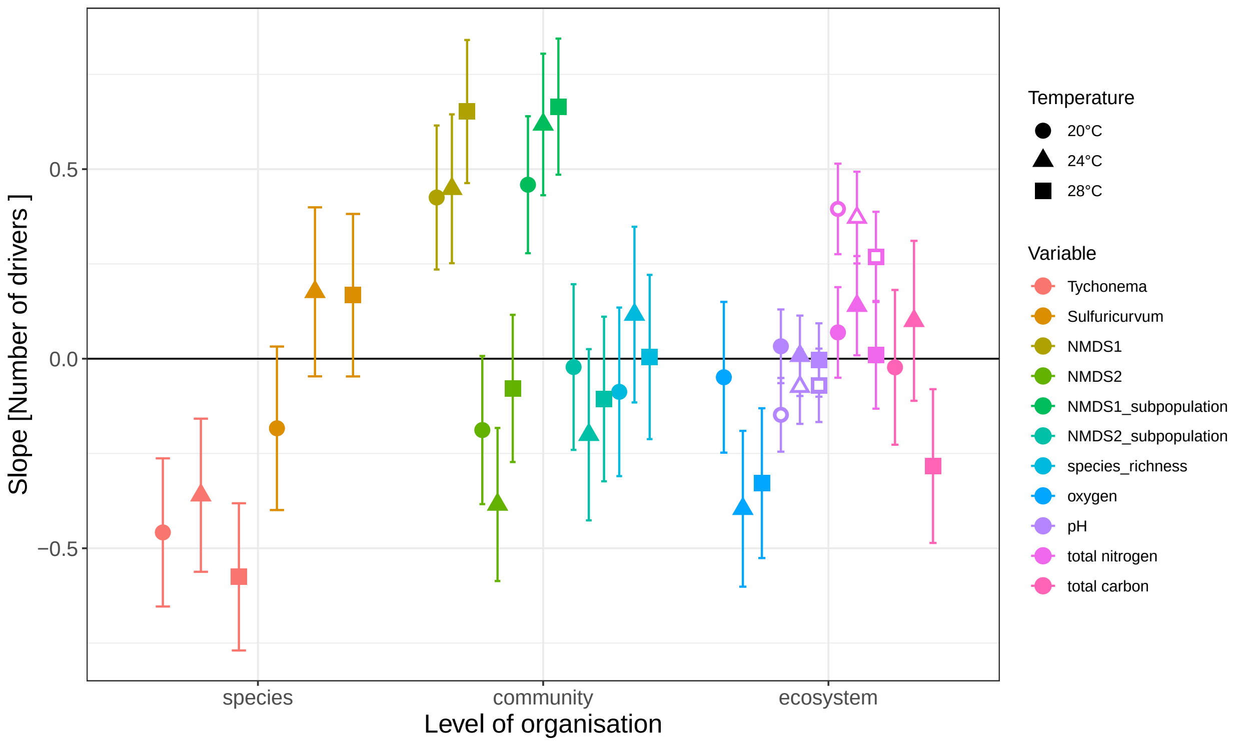


Fig4: effect of Number of drivers on response variables

```
all <- rbind(tychonema_slope,sulfuri_slope,shannon_slope,
             nmds1_slope,nmds2_slope,nmds1_subset_slope,
             nmds2_subset_slope,oxy_slope,pH_slope,
             tn_slope,tc_slope)  %>%
  reorder_levels(variable,order=c("Tychonema","Sulfuricurvum","NMDS1",
                                  "NMDS2","NMDS1_subpopulation",
                                  "NMDS2_subpopulation","species_richness",
                                  "oxygen","pH","total nitrogen",
                                  "total carbon"))

plot_slope <-
    all %>%
    ggplot(aes(x = organisation, y = estimate, colour = variable, shape=term, group=variable)) +
    geom_hline(aes(yintercept=0))+
    geom_errorbar(aes(ymin=conf.low, ymax=conf.high),
                  width=0.1,alpha=0.4,position = position_dodge(width = 0.8))+
    geom_point(size = 4,position = position_dodge(width = 0.8))  +
    # geom_point(colour = all$col.fert, size = 2, col="white", position = position_dodge(width = 0.8), 
    labs(y="Slope [Number of drivers ]", x="Level of organisation") +
    theme_bw() +
    theme(axis.title.x = element_text(size = 15),
          axis.text.x = element_text(size=12),
          axis.title.y = element_text(size = 15),
          axis.text.y = element_text(size=12)) +
    scale_x_discrete(limits = c("species","community","ecosystem"))

cols <- ggplot_build(plot_slope)
cols <- unique(cols$data[[2]]$colour)

var.cols <- c("Tychonema"=cols[1],
              "Sulfuricurvum"=cols[2],
              "NMDS1"=cols[4],
              "NMDS2"=cols[5],
              "NMDS1_subpopulation"=cols[6],
              "NMDS2_subpopulation"=cols[7],
              "species_richness"=cols[3],
              "oxygen"=cols[8],
              "pH"=cols[9],
              "total nitrogen"=cols[10],
              "total carbon"=cols[11])

all$col.fert <- ifelse(all$Fertilizer=="Yes",var.cols[all$variable],"white")

all$term2 <- ifelse(grepl("20", all$term, fixed = TRUE),"20°C",
                    ifelse(grepl("24", all$term, fixed = TRUE),"24°C","28°C"))

plot_slope <-
    all %>%
    ggplot(aes(x = organisation, y = estimate, colour = variable, shape=term2, group=interaction(term,va
    geom_hline(aes(yintercept=0))+
    geom_errorbar(aes(ymin=conf.low, ymax=conf.high), size=0.6,
                  width=0.3,position = position_dodge(width = 0.8))+
    geom_point(size = 4,position = position_dodge(width = 0.8))  +
    geom_point(col = all$col.fert, size = 2, position = position_dodge(width = 0.8), show.legend = F)  +
    labs(y="Slope [Number of drivers ]", x="Level of organisation",
         shape="Temperature", colour="Variable") +
    theme_bw() +
    theme(axis.title.x = element_text(size = 15),
          axis.text.x = element_text(size=12),
          axis.title.y = element_text(size = 15),
          axis.text.y = element_text(size=12)) +
    scale_x_discrete(limits = c("species","community","ecosystem"))

plot_slope
```



Slope [Number of drivers]
0.5
0.0
−0.5
species
community
ecosystem
Level of organisation
Temperature
● 20°C
▲ 24°C
■ 28°C
Variable
Tychonema
Sulfuricurvum
NMDS1
NMDS2
NMDS1_subpopulation
NMDS2_subpopulation
species_richness
oxygen
pH
total nitrogen
total carbon