

# Single Year Models

Advyth Ramachandran

2024-05-19

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.2      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.2      v tibble    3.2.1
## v lubridate  1.9.2      v tidyr     1.3.0
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggpubr)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:ggpubr':
##
##   get_legend
##
## The following object is masked from 'package:lubridate':
##
##   stamp
```

```
library(lme4)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##   expand, pack, unpack
```

```
library(glmmTMB)
```

```
## Warning in checkDepPackageVersion(dep_pkg = "TMB"): Package version inconsistency detected.  
## glmmTMB was built with TMB version 1.9.6  
## Current TMB version is 1.9.10  
## Please re-install glmmTMB from source or restore original 'TMB' package (see '?reinstalling' for more)
```

```
library(car)
```

```
## Loading required package: carData  
##  
## Attaching package: 'car'  
##  
## The following object is masked from 'package:dplyr':  
##  
##     recode  
##  
## The following object is masked from 'package:purrr':  
##  
##     some
```

```
library(emmeans)  
library(stringr)  
library(performance)  
library(purrr)  
library(gridExtra)
```

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

```
# options(scipen = 999)
```

```
analysis_df <- read.csv("analyses_manuscript/processed_data/analysis_variables.csv") %>%  
  mutate(year = as.factor(year)) %>%  
  mutate(log_shannon_native = log(shannon_native + 1)) %>%  
  mutate(shannon_native_backtransform = exp(log_shannon_native)-1)  
  
analysis_2022_df <- analysis_df %>%  
  filter(year == "2022")  
  
analysis_2023_df <- analysis_df %>%  
  filter(year == "2023")
```

## Function to test models

First, a function to test the model and return the test itself.

```

test_single_year_model <- function(fitted_model) {

  model_results <- list()

  chisqAnova <- car::Anova(fitted_model)
  print(chisqAnova)
  model_results[["chisquare_test"]] <- chisqAnova

  # Extract p-value for the treatment*year interaction term
  pval <- chisqAnova$'Pr(>Chisq)'[1]
  pval.r <- ifelse(
    pval < 0.001,
    "< 0.001",
    round(pval, digits = 3)
  )

  # Assess significance of p-value
  significance <- ifelse(pval < 0.05,
    "*",
    "N.S.")
  pval_signif <- paste(pval.r, significance, sep = " ")

  print(pval_signif)
  model_results[["significance"]] <- pval_signif

  return(model_results)
}

```

Now, a function to save the results to the global environment (every time this function is called, it will also save the results to a list in the global environment).

```

# Initialize a list to store all model results
single_year_results_list <- list()

save_model_results <- function(fitted_model, year) {

  model_results <- test_single_year_model(fitted_model = fitted_model)

  # Every time this function is called, it will add the results to the list of
# results in the global environment
  model_name <- deparse(substitute(fitted_model))
  model_name_with_year <- paste(model_name, year, sep = "_")
  single_year_results_list[[model_name_with_year]] <<- model_results

}

```

## Function to plot models

```

plot_single_year_model <- function(fitted_model, year, response_variable, response_name, dataset) {

  # Test the model

```

```

model_results <- test_single_year_model(fitted_model = fitted_model)

significance <- model_results[['significance']]

# Calculate position for text label
# rv_vec <- analysis_df[[response_variable]]
# label_position_x <- 'unraked'
# label_position_y <- max(rv_vec) + (max(rv_vec) * 0.05)

# Format label
significance_label <- paste("p-value:", significance, sep = " ")

# Plot title
# plot_title <- paste(year, sep = " ")
plot_title <- significance_label

# Make plot
model_plot <- dataset %>%
  ggplot(aes_string(
    x = "treatment",
    y = response_variable,
    fill = "treatment",
    color = "treatment",
    group = "treatment"
  )) +
  geom_jitter(
    alpha = 0.3,
    shape = 16,
    width = 0.3,
    height = 0,
    size = 3
  ) +
  geom_boxplot(color = "black", alpha = 0.4, outlier.shape = NA) +
  scale_fill_manual(values = c("antiquewhite4", "darkgoldenrod2")) +
  scale_color_manual(
    values = c("antiquewhite4", "darkgoldenrod2"),
    guide = guide_legend(override.aes = list(alpha = 1))
  ) +
  labs(title = plot_title) +
  theme_pubr() +
  theme(legend.position = "none",
    plot.title = element_text(size = 12)) +
  ylab(response_name)
# geom_text(aes(x = label_position_x,
#               y = label_position_y),
# label = significance_label,
# hjust = 1,
# vjust = 1,
# size = 3,
# color = "black")

return(model_plot)
}

```

# Individual year models

## Species richness

```
fit_sprich <- function(dataset, year) {  
  
  sprich_fm <- glmer(sprich ~ treatment + (1 | block),  
                    family = "poisson",  
                    data = dataset)  
  
  save_model_results(sprich_fm, year)  
  
  p <- plot_single_year_model(  
    fitted_model = sprich_fm,  
    year = year,  
    response_variable = "sprich",  
    response_name = "Species richness",  
    dataset = dataset  
  )  
  
  return(p)  
}  
  
p_sprich_2022 <- fit_sprich(dataset = analysis_2022_df, year = "2022")  
  
## boundary (singular) fit: see help('isSingular')  
  
## Analysis of Deviance Table (Type II Wald chisquare tests)  
##  
## Response: sprich  
##           Chisq Df Pr(>Chisq)  
## treatment 3.9326  1    0.04736 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
## [1] "0.047 *"  
## Analysis of Deviance Table (Type II Wald chisquare tests)  
##  
## Response: sprich  
##           Chisq Df Pr(>Chisq)  
## treatment 3.9326  1    0.04736 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
## [1] "0.047 *"  
  
## Warning: 'aes_string()' was deprecated in ggplot2 3.0.0.  
## i Please use tidy evaluation idioms with 'aes()'.  
## i See also 'vignette("ggplot2-in-packages")' for more information.  
## This warning is displayed once every 8 hours.  
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was  
## generated.
```

```
p_sprich_2023 <- fit_sprich(dataset = analysis_2023_df, year = "2023")
```

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

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: sprich
##           Chisq Df Pr(>Chisq)
## treatment 13.83  1  0.0002002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: sprich
##           Chisq Df Pr(>Chisq)
## treatment 13.83  1  0.0002002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"
```

## Native species richness

```
fit_sprich_native <- function(dataset, year) {
  sprich_native_fm <- glmer(sprich_native ~ treatment + (1 | block),
    family = "poisson",
    data = dataset)

  save_model_results(sprich_native_fm, year)

  p <- plot_single_year_model(
    fitted_model = sprich_native_fm,
    year = year,
    response_variable = "sprich_native",
    response_name = "Native species richness",
    dataset = dataset
  )

  return(p)
}
```

```
p_sprich_native_2022 <- fit_sprich_native(dataset = analysis_2022_df, year = "2022")
```

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

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: sprich_native
##           Chisq Df Pr(>Chisq)
```

```
## treatment 3.3589 1 0.06684 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "0.067 N.S."
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: sprich_native
##           Chisq Df Pr(>Chisq)
## treatment 3.3589 1 0.06684 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "0.067 N.S."
```

```
p_sprich_native_2023 <- fit_sprich_native(dataset = analysis_2023_df, year = "2023")
```

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

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: sprich_native
##           Chisq Df Pr(>Chisq)
## treatment 6.0138 1 0.01419 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "0.014 *"
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: sprich_native
##           Chisq Df Pr(>Chisq)
## treatment 6.0138 1 0.01419 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "0.014 *"
```

```
fit_shannon <- function(dataset, year) {

  shannon_fm <- lmer(shannon ~ treatment + (1|block),
                    data = dataset)

  save_model_results(shannon_fm, year)

  p <- plot_single_year_model(
    fitted_model = shannon_fm,
    year = year,
    response_variable = "shannon",
    response_name = "Shannon diversity",
    dataset = dataset
  )

  return(p)
}
```

```
p_shannon_2022 <- fit_shannon(dataset = analysis_2022_df, year = "2022")
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: shannon
##           Chisq Df Pr(>Chisq)
## treatment 52.844  1  3.611e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: shannon
##           Chisq Df Pr(>Chisq)
## treatment 52.844  1  3.611e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"
```

```
p_shannon_2023 <- fit_shannon(dataset = analysis_2023_df, year = "2023")
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: shannon
##           Chisq Df Pr(>Chisq)
## treatment 61.635  1  4.135e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: shannon
##           Chisq Df Pr(>Chisq)
## treatment 61.635  1  4.135e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"
```

## Shannon Native

```
fit_shannon_native <- function(dataset, year) {
  shnat_fm <- glmmTMB(log_shannon_native ~ treatment + (1|block),
    ziformula=~1,
    data = dataset,
    family = beta_family())
  save_model_results(shnat_fm, year)
  p <- plot_single_year_model(
```



```

    fitted_model = shnat_fm,
    year = year,
    response_variable = "shannon_native",
    response_name = "Native Shannon diversity",
    dataset = dataset
  )

  return(p)
}

```

```
p_shannon_native_2022 <- fit_shannon_native(dataset = analysis_2022_df, year = "2022")
```

```

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: log_shannon_native
##           Chisq Df Pr(>Chisq)
## treatment 0.1228 1      0.726
## [1] "0.726 N.S."
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: log_shannon_native
##           Chisq Df Pr(>Chisq)
## treatment 0.1228 1      0.726
## [1] "0.726 N.S."

```

```
p_shannon_native_2023 <- fit_shannon_native(dataset = analysis_2023_df, year = "2023")
```

```

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: log_shannon_native
##           Chisq Df Pr(>Chisq)
## treatment 0.5301 1      0.4666
## [1] "0.467 N.S."
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: log_shannon_native
##           Chisq Df Pr(>Chisq)
## treatment 0.5301 1      0.4666
## [1] "0.467 N.S."

```

## Exotic forb

```

fit_exotic_forb <- function(dataset, year) {

  ef_fm <- glmmTMB(exotic_forb ~ treatment + (1|block),
    ziformula=~1,
    data = dataset,
    family= beta_family()
  )
}

```

```

save_model_results(ef_fm, year)

p <- plot_single_year_model(
  fitted_model = ef_fm,
  year = year,
  response_variable = "exotic_forb",
  response_name = "Exotic forb abundance",
  dataset = dataset
)

return(p)
}

p_exotic_forb_2022 <- fit_exotic_forb(dataset = analysis_2022_df, year = "2022")

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: exotic_forb
##           Chisq Df Pr(>Chisq)
## treatment 95.107  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: exotic_forb
##           Chisq Df Pr(>Chisq)
## treatment 95.107  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"

p_exotic_forb_2023 <- fit_exotic_forb(dataset = analysis_2023_df, year = "2023")

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: exotic_forb
##           Chisq Df Pr(>Chisq)
## treatment 18.816  1  1.439e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: exotic_forb
##           Chisq Df Pr(>Chisq)
## treatment 18.816  1  1.439e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "< 0.001 *"

```

## Native forb

```
fit_native_forb <- function(dataset, year) {  
  nf_fm <- glmmTMB(native_forb ~ treatment + (1|block),  
                  ziformula=~1,  
                  data = dataset,  
                  family= beta_family()  
)  
  
  save_model_results(nf_fm, year)  
  
  p <- plot_single_year_model(  
    fitted_model = nf_fm,  
    year = year,  
    response_variable = "native_forb",  
    response_name = "Native forb abundance",  
    dataset = dataset  
  )  
  
  return(p)  
}
```

```
p_native_forb_2022 <- fit_native_forb(dataset = analysis_2022_df, year = "2022")
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)  
##  
## Response: native_forb  
##           Chisq Df Pr(>Chisq)  
## treatment 6.2402  1    0.01249 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
## [1] "0.012 *"  
## Analysis of Deviance Table (Type II Wald chisquare tests)  
##  
## Response: native_forb  
##           Chisq Df Pr(>Chisq)  
## treatment 6.2402  1    0.01249 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
## [1] "0.012 *"
```

```
p_native_forb_2023 <- fit_native_forb(dataset = analysis_2023_df, year = "2023")
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)  
##  
## Response: native_forb  
##           Chisq Df Pr(>Chisq)  
## treatment 6.5744  1    0.01035 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## [1] "0.01 *"
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: native_forb
##           Chisq Df Pr(>Chisq)
## treatment 6.5744  1    0.01035 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "0.01 *"
```

## Format and export plots

```
arrange_plots <- function(title, plots_list, ymin, ymax) {

  plots_adjusted_list <- list()

  for (plot_name in names(plots_list)) {

    plot <- plots_list[[plot_name]]
    plot_adj <- plot + ylim(min = ymin, max = ymax)

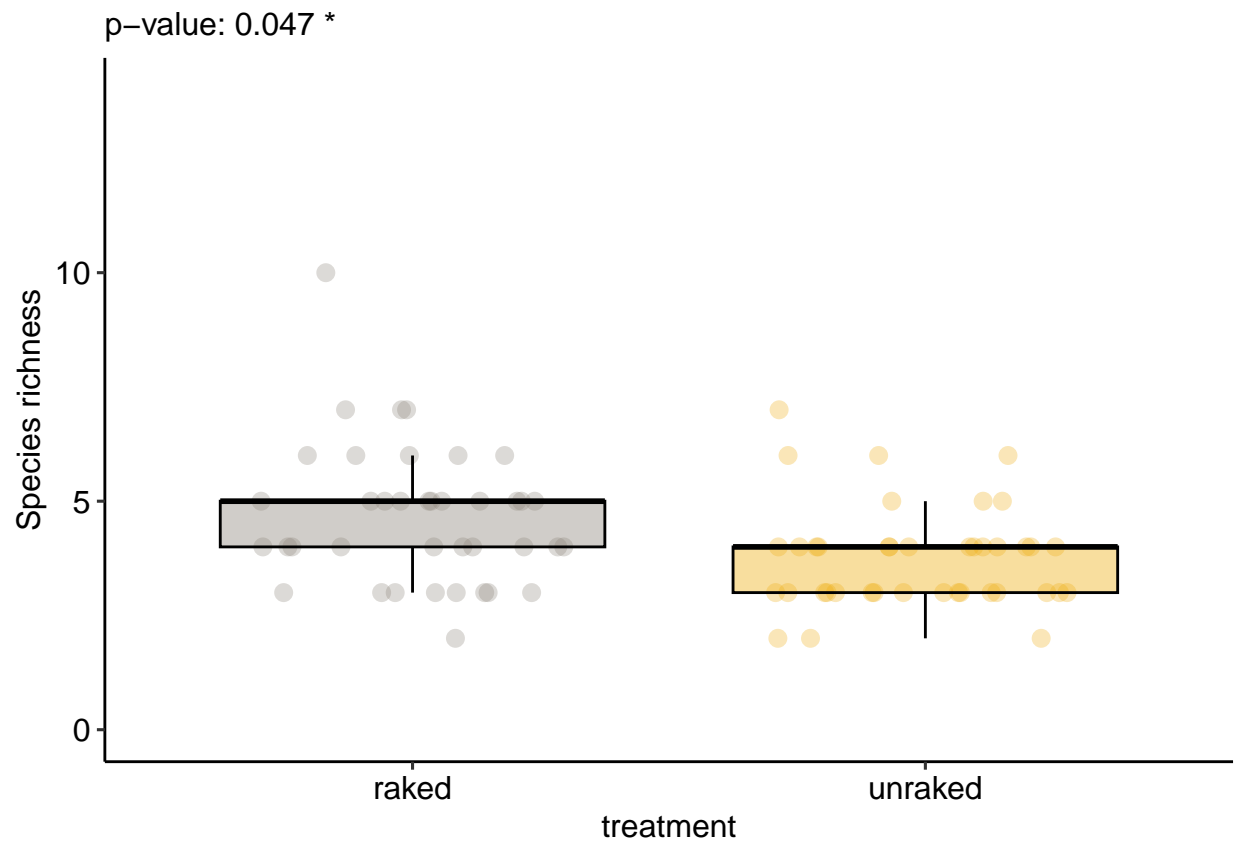
    plots_adjusted_list[[plot_name]] <- plot_adj
  }
  return(plots_adjusted_list)
}
```

```
sprich_as_unit_plots <- list(
  p_sprich_2022 = p_sprich_2022,
  p_sprich_2023 = p_sprich_2023,
  p_sprich_native_2022 = p_sprich_native_2022,
  p_sprich_native_2023= p_sprich_native_2023)

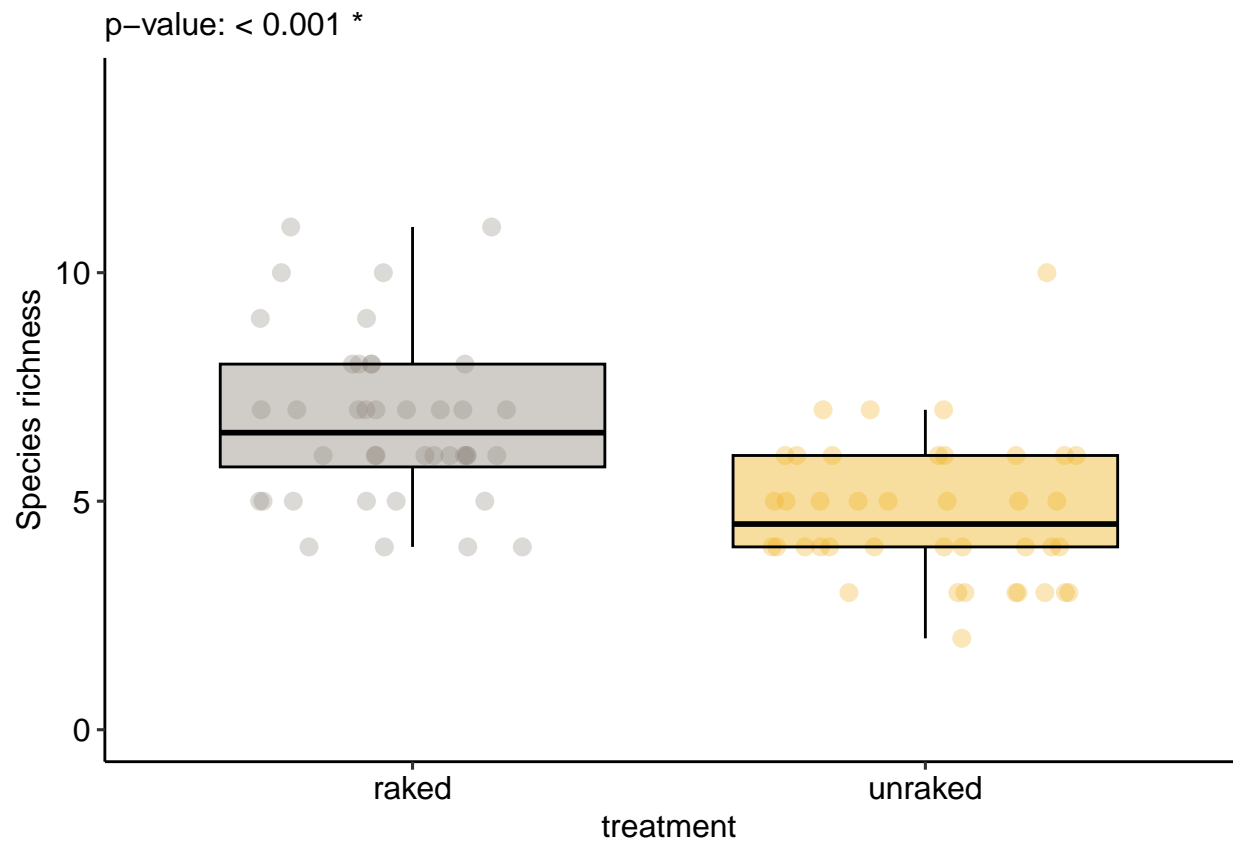
sprich_as_unit_plots_adj <- arrange_plots(
  title = "Species richness",
  plots_list = sprich_as_unit_plots, ymin = 0, ymax = 14)

sprich_as_unit_plots_adj
```

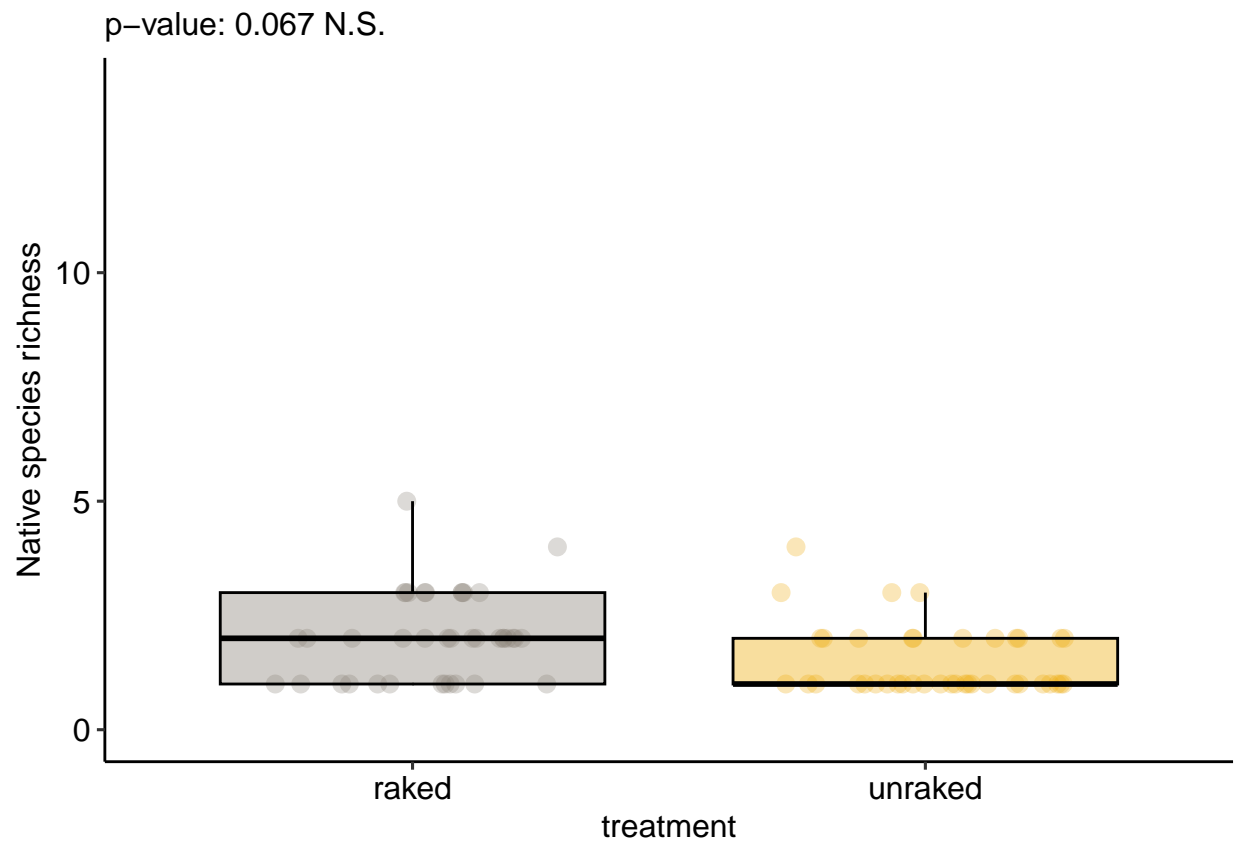
```
## $p_sprich_2022
```



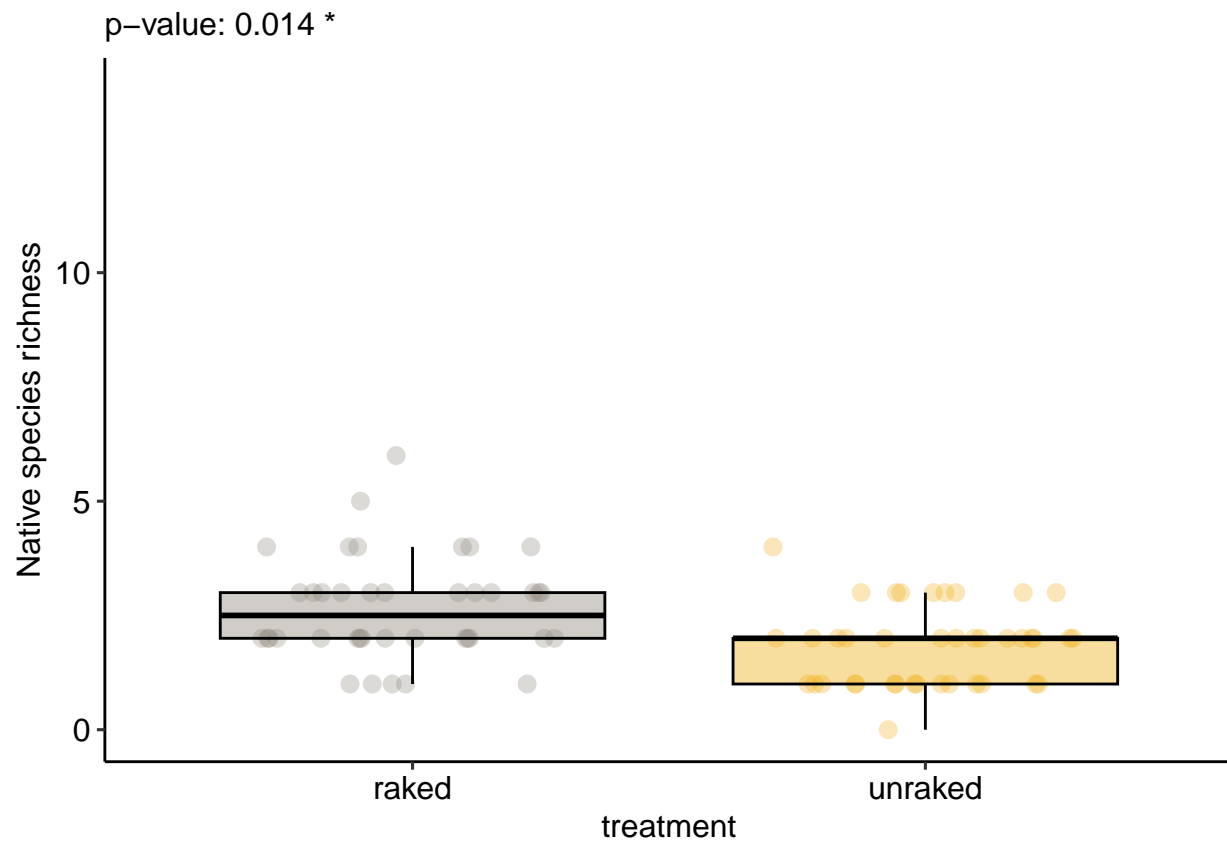
```
##  
## $p_sprich_2023
```



```
##
## $p_sprich_native_2022
```



```
##  
## $p_sprich_native_2023
```



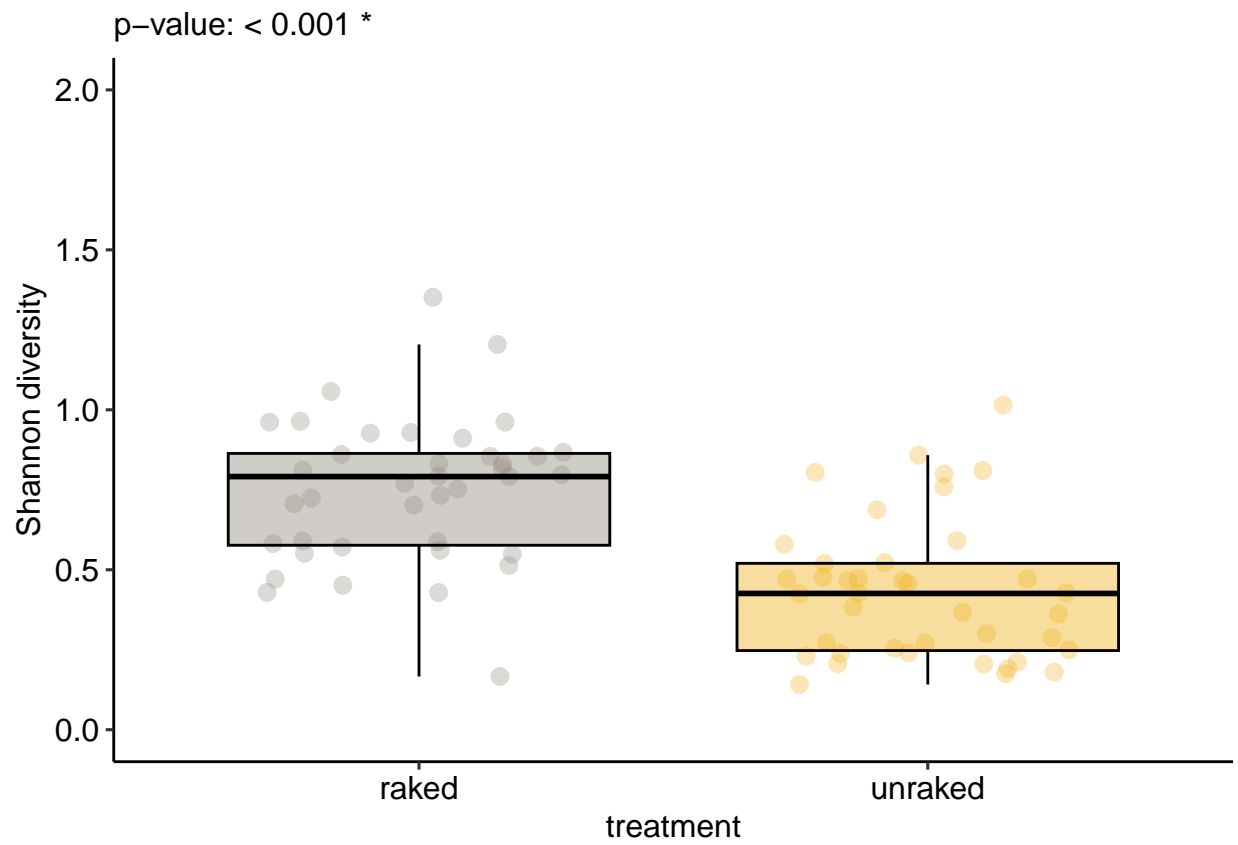
```
shannon_as_unit_plots <- list(
  p_shannon_2022 = p_shannon_2022,
  p_shannon_2023 = p_shannon_2023,
  p_shannon_native_2022 = p_shannon_native_2022,
  p_shannon_native_2023 = p_shannon_native_2023)

shannon_as_unit_plots_adj <- arrange_plots(
  title = "Shannon diversity",
  plots_list = shannon_as_unit_plots, ymin = 0, ymax = 2)

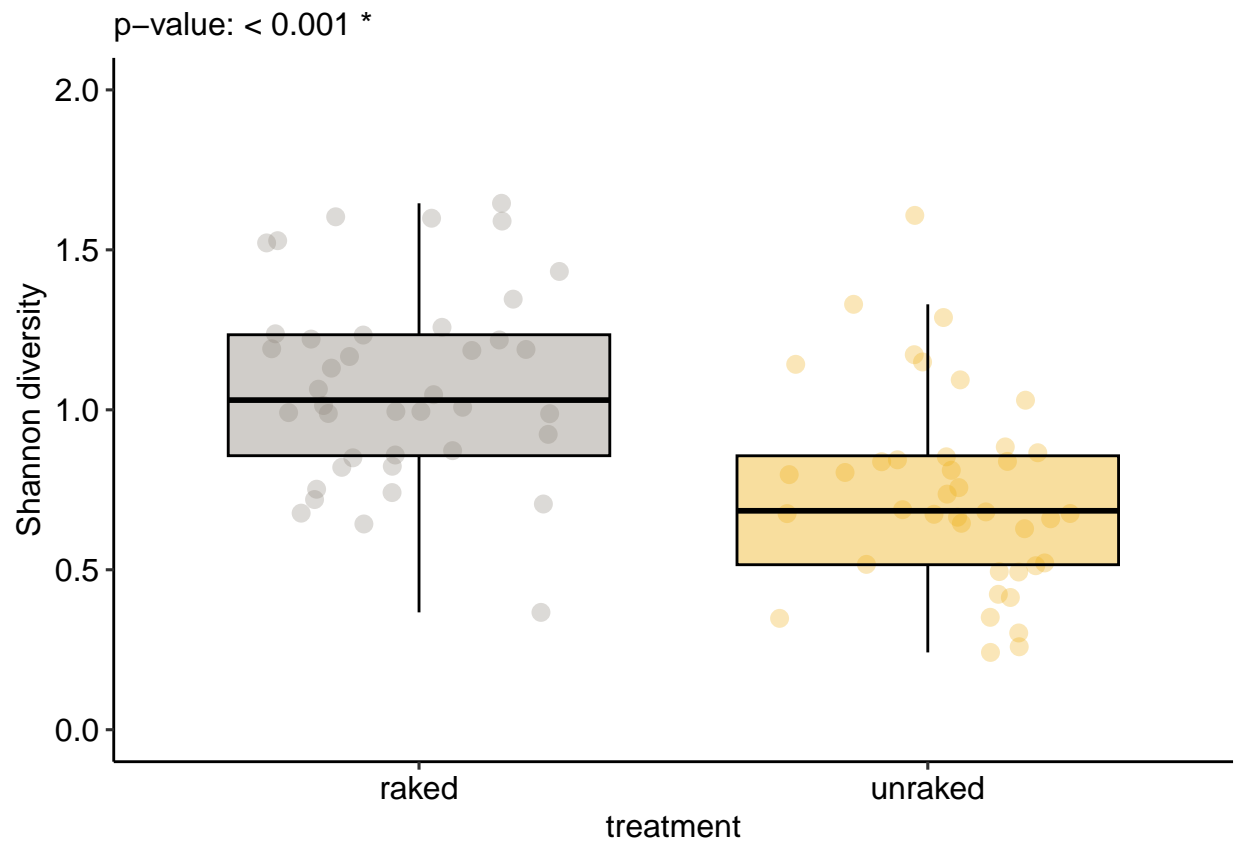
shannon_as_unit_plots_adj
```

```
## $p_shannon_2022
```

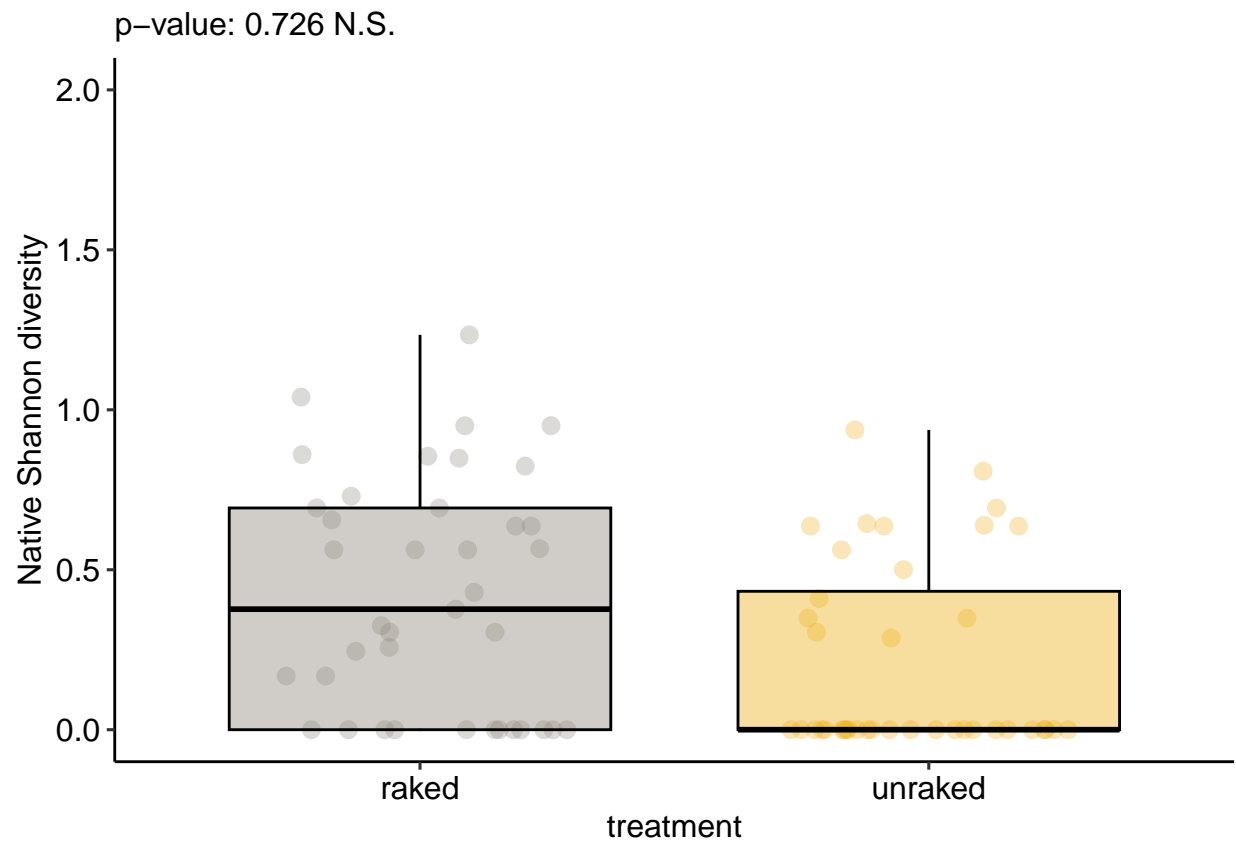




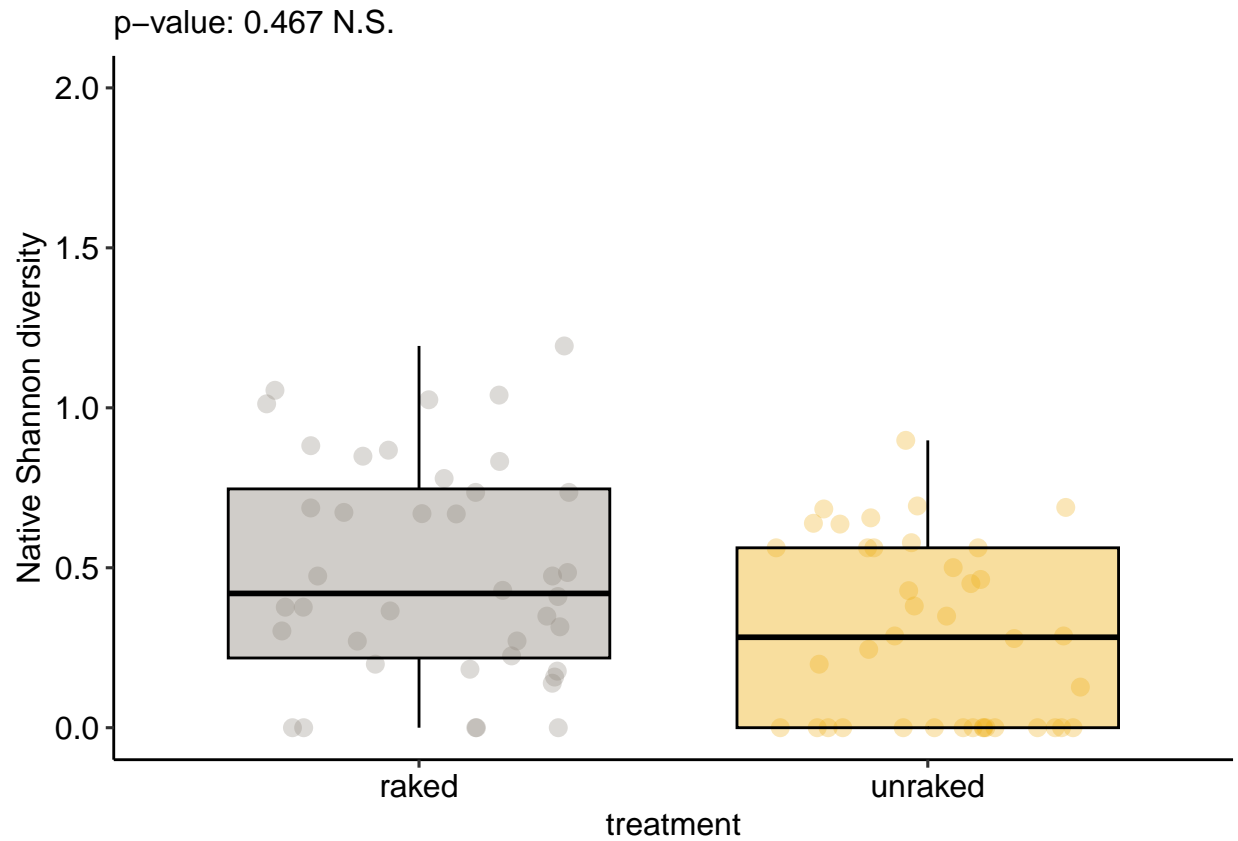
```
##  
## $p_shannon_2023
```



```
##  
## $p_shannon_native_2022
```



```
##
## $p_shannon_native_2023
```

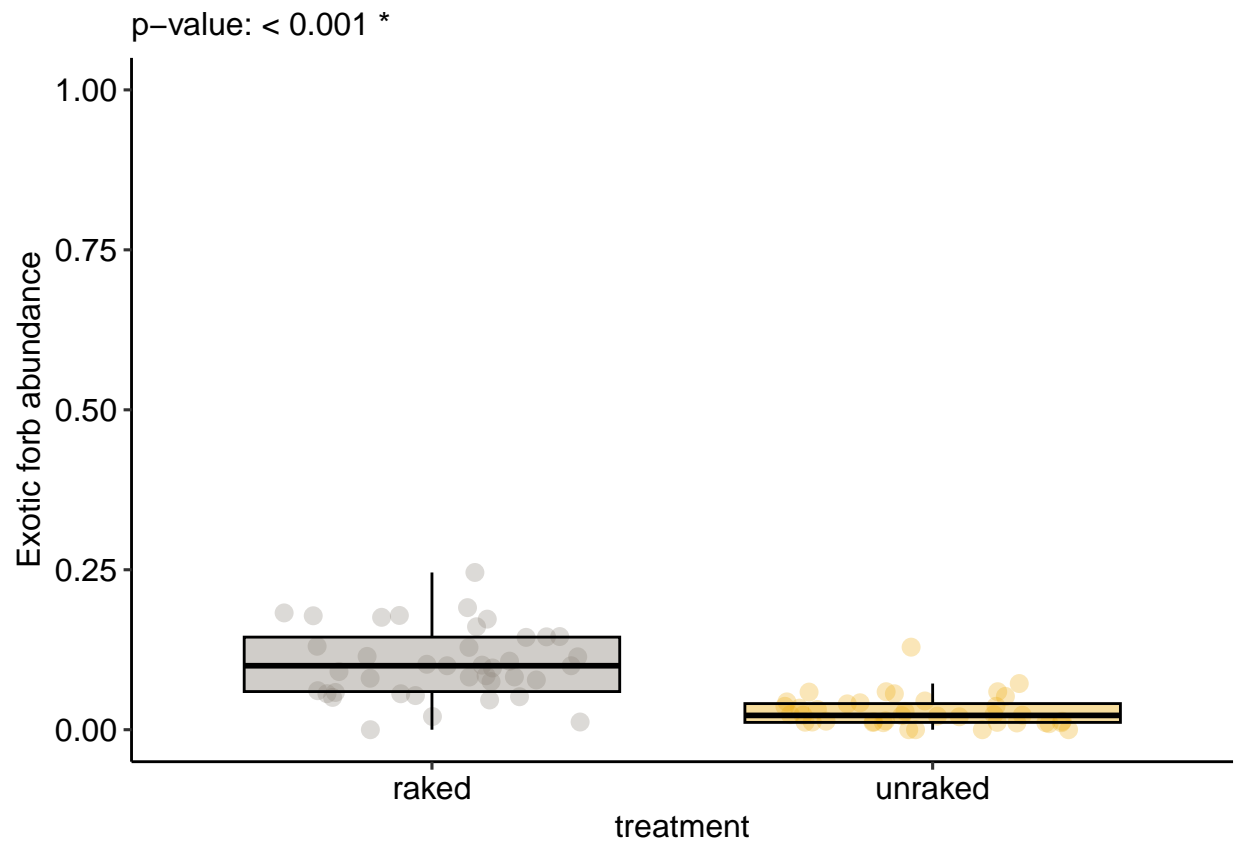


```
abundance_as_unit_plots <- list(
  p_exotic_forb_2022 = p_exotic_forb_2022,
  p_exotic_forb_2023 = p_exotic_forb_2023,
  p_native_forb_2022 = p_native_forb_2022,
  p_native_forb_2023 = p_native_forb_2023)

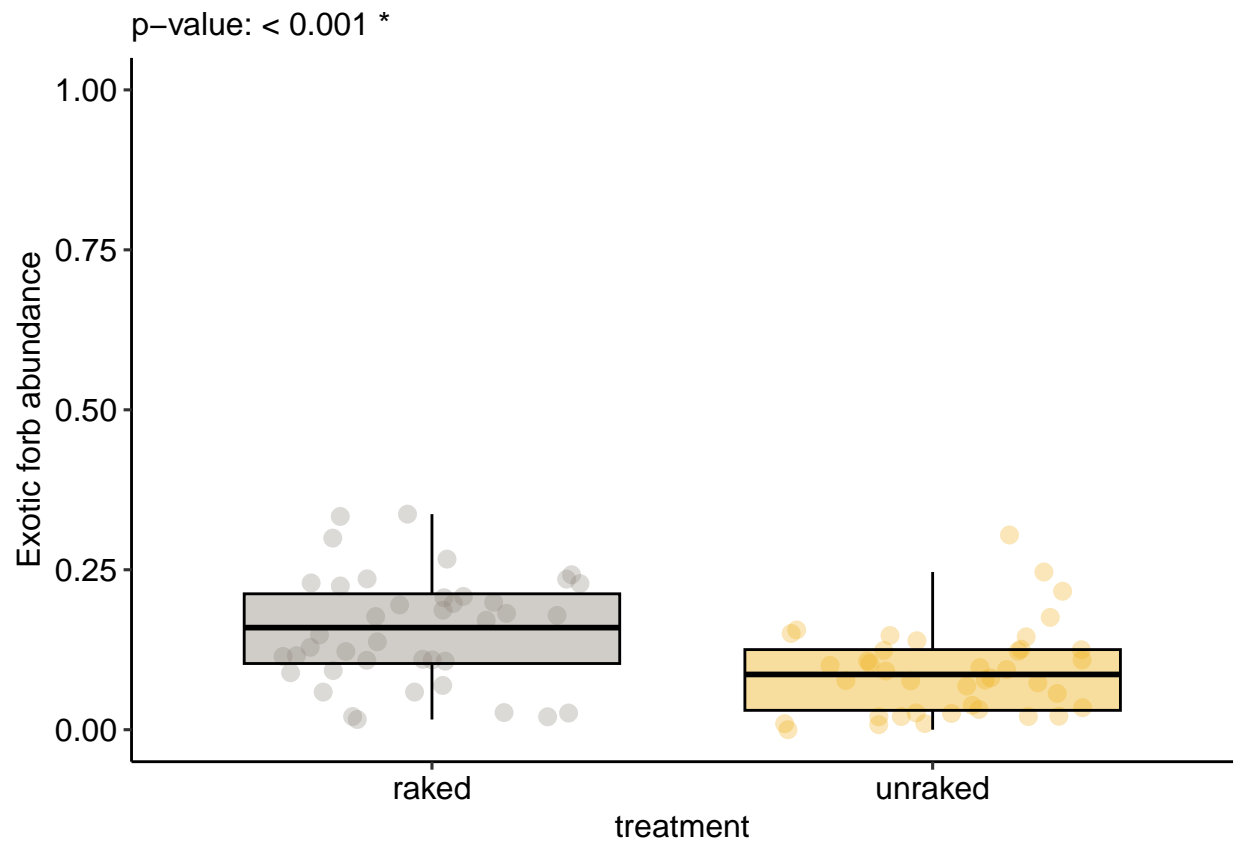
abundance_as_unit_plots_adj <- arrange_plots(
  title = "Abundance",
  plots_list = abundance_as_unit_plots, ymin = 0, ymax = 1)

abundance_as_unit_plots_adj
```

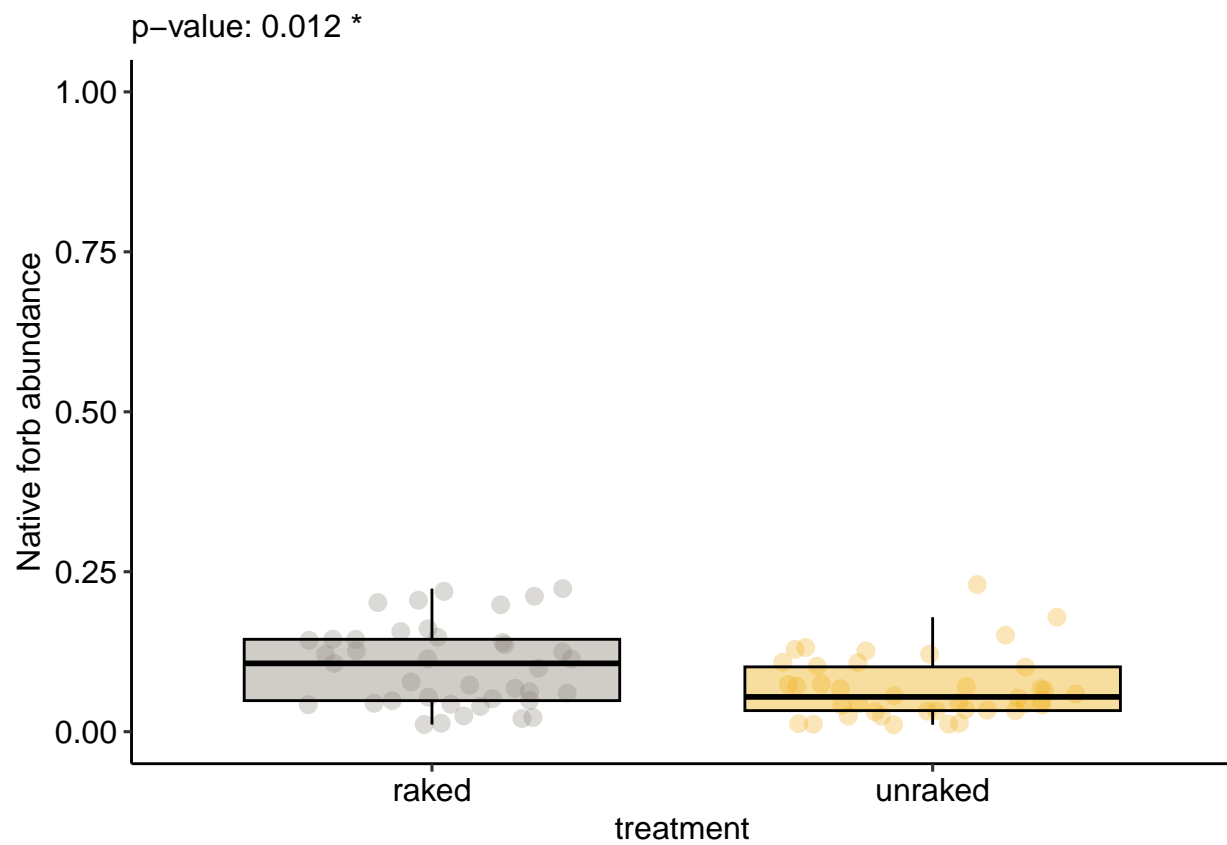
```
## $p_exotic_forb_2022
```



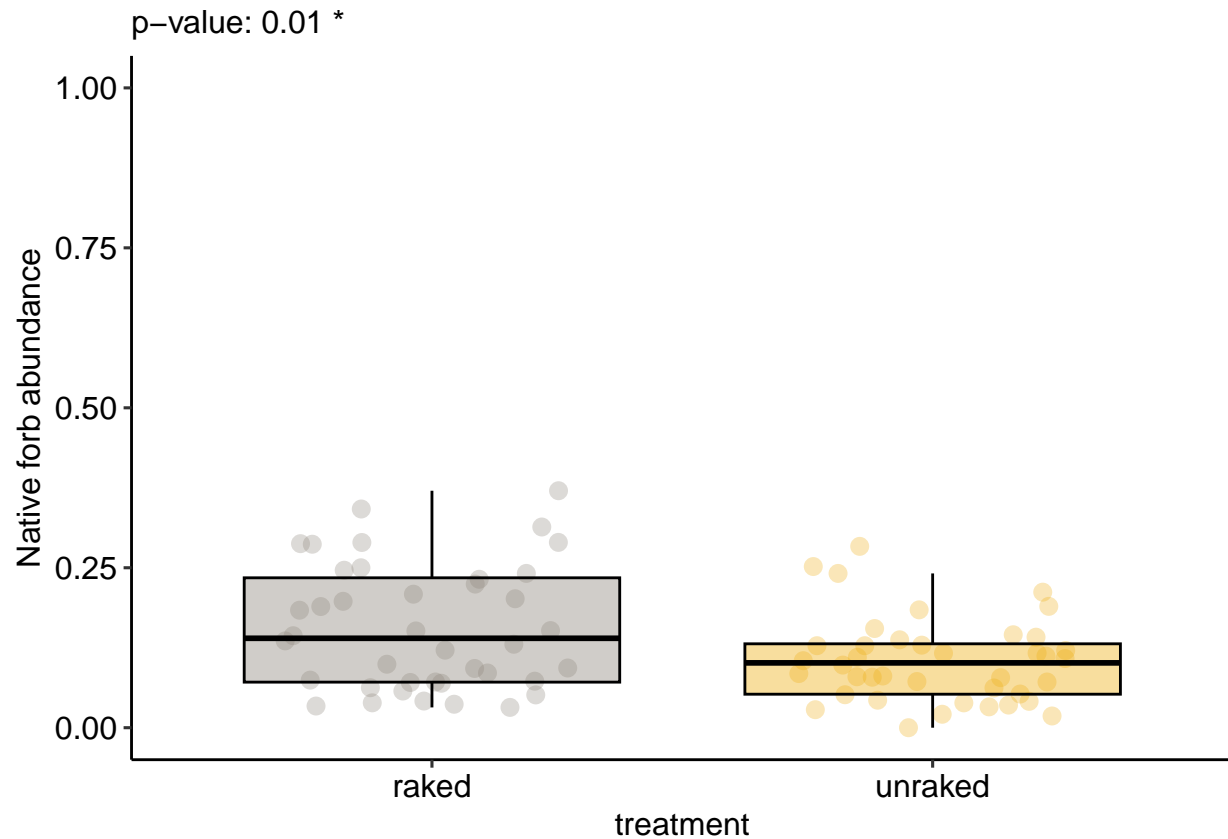
```
##  
## $p_exotic_forb_2023
```



```
##  
## $p_native_forb_2022
```



```
##  
## $p_native_forb_2023
```



```
plots_by_unit_list <- list(sprich_as_unit_plots_adj,
  shannon_as_unit_plots_adj,
  abundance_as_unit_plots_adj)
```

```
all_plots_list <- purrr::flatten(plots_by_unit_list)
```

```
names(all_plots_list)
```

```
## [1] "p_sprich_2022"      "p_sprich_2023"      "p_sprich_native_2022"
## [4] "p_sprich_native_2023" "p_shannon_2022"      "p_shannon_2023"
## [7] "p_shannon_native_2022" "p_shannon_native_2023" "p_exotic_forb_2022"
## [10] "p_exotic_forb_2023" "p_native_forb_2022"  "p_native_forb_2023"
```

```
order_plots_for_2022 <- c(
  "p_sprich_2022", "p_sprich_native_2022", "p_shannon_2022", "p_shannon_native_2022", "p_exotic_forb_2022"
)
```

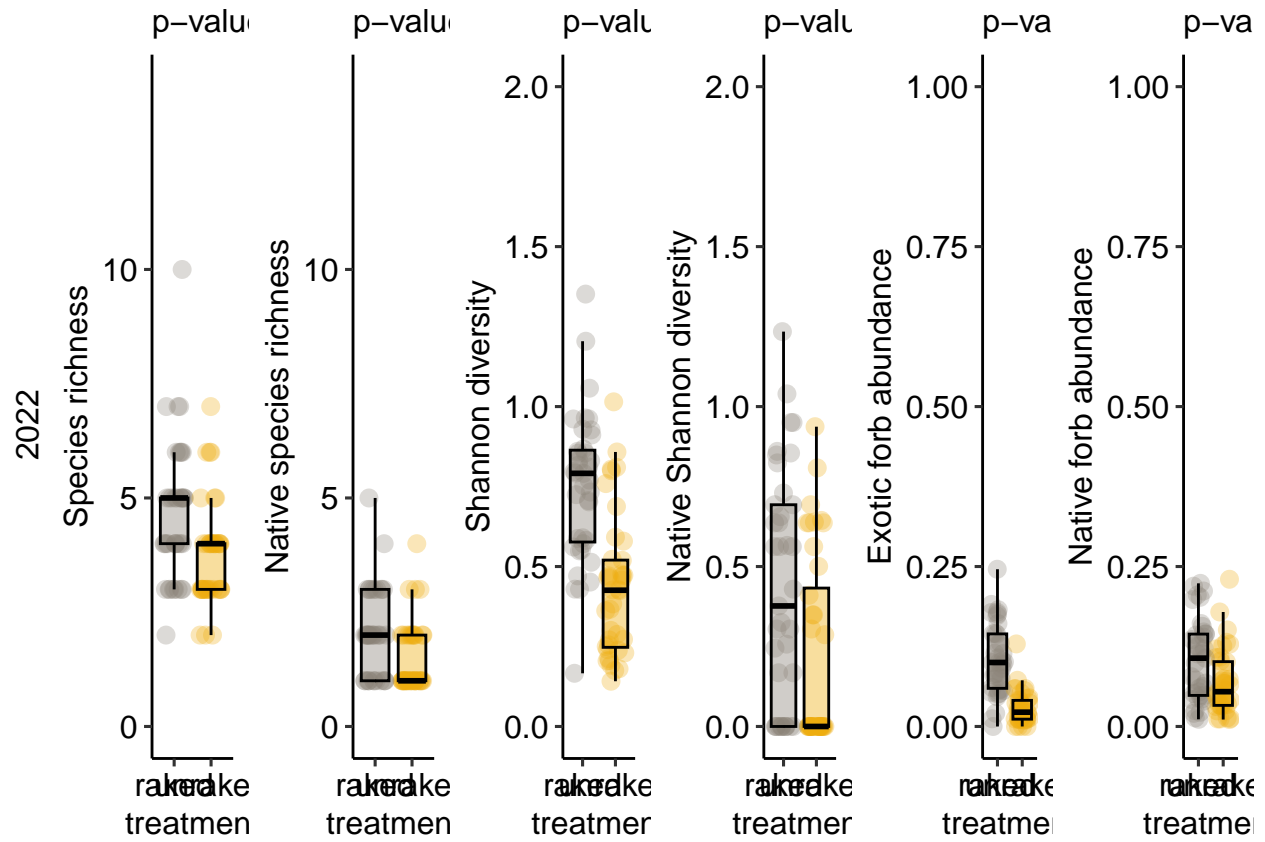
```
order_plots_for_2023 <- c(
  "p_sprich_2023", "p_sprich_native_2023", "p_shannon_2023", "p_shannon_native_2023", "p_exotic_forb_2023"
)
```

```
plots_for_2022 <- all_plots_list[order_plots_for_2022]
```

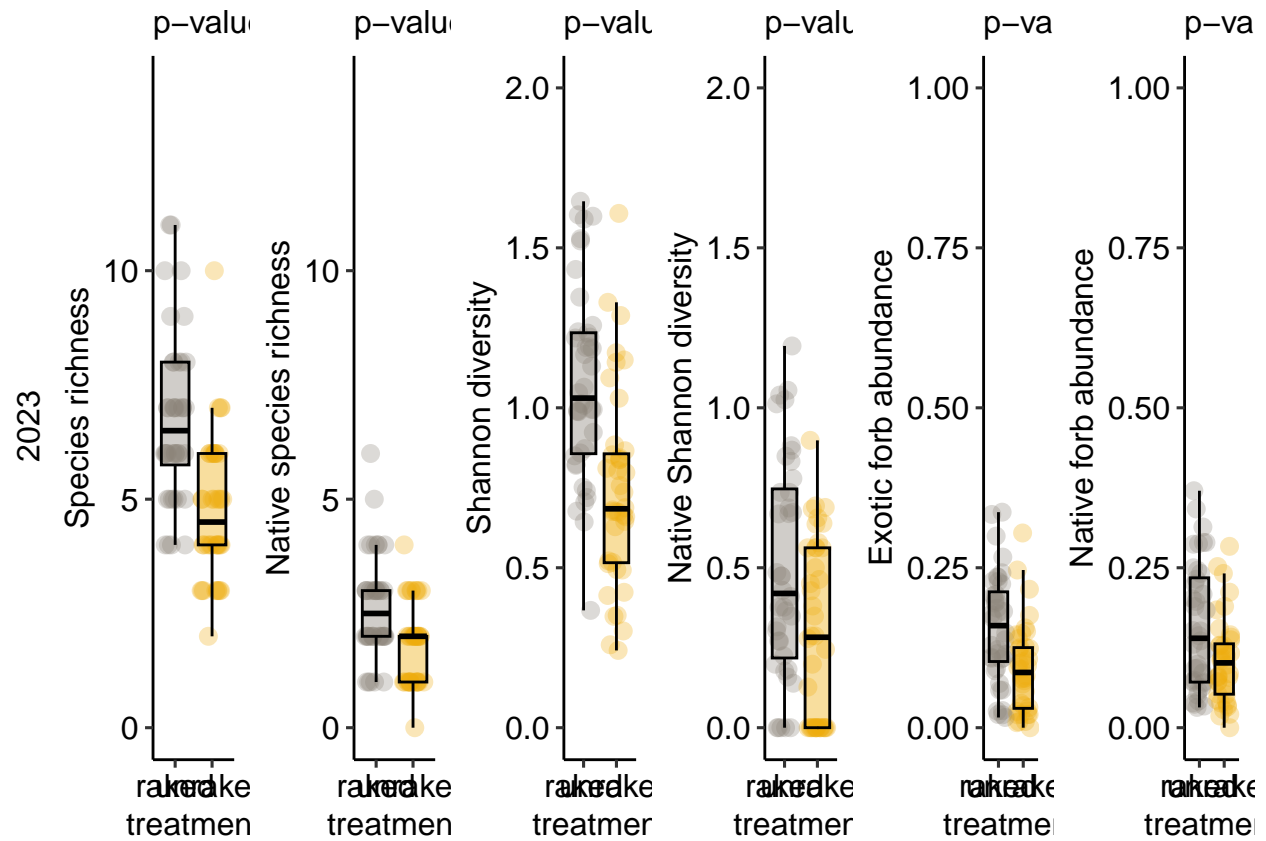
```
plots_for_2023 <- all_plots_list[order_plots_for_2023]
```



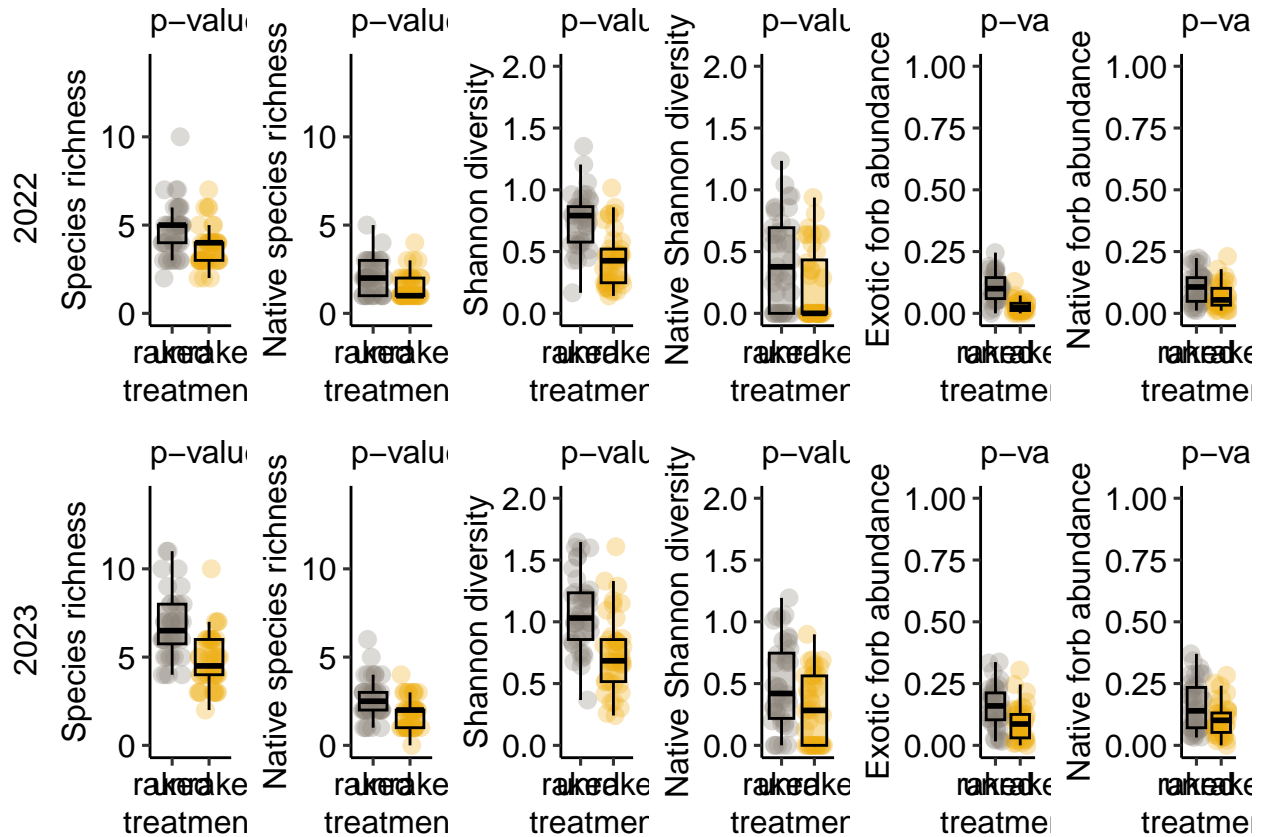
```
plots_combined_2022 <- grid.arrange(grobs = plots_for_2022, ncol = 6, nrow = 1, left = "2022")
```



```
plots_combined_2023 <- grid.arrange(grobs = plots_for_2023, ncol = 6, nrow = 1, left = "2023")
```



```
arranged_single_year_model_plots <- grid.arrange(plots_combined_2022, plots_combined_2023)
```



```
arranged_single_year_model_plots
```

```
## TableGrob (2 x 1) "arrange": 2 grobs
##      z      cells      name      grob
## 1 1 (1-1,1-1) arrange gtable[arrange]
## 2 2 (2-2,1-1) arrange gtable[arrange]
```

```
ggsave(filename = "arranged_single_year_model_plots.pdf",
        plot = arranged_single_year_model_plots,
        path = "analyses_manuscript/figures",
        width = 14,
        height = 5,
        scale = 1.2)
```

Save the results as a table.

```
# Save table of significance.
```

```
all_results_df <- data.frame(
  model = character(),
  predictor = character(),
  Chisq = numeric(),
  Df = numeric(),
  p_value = numeric()
)
```

```

for (model_name in names(single_year_results_list)) {

  anova_table <- single_year_results_list[[model_name]][['chisquare_test']]

  anova_table_formatted <- anova_table %>%
    rownames_to_column("predictor") %>%
    rename("p_value" = `Pr(>Chisq)` ) %>%
    mutate(model = model_name) %>%
    mutate(p_value_f = case_when(
      p_value < 0.001 ~ "< 0.001",
      p_value > 0.001 ~ paste(
        signif(p_value, digits = 2))
    )) %>%
    mutate(significance = case_when(
      p_value < 0.05 ~ "*",
      p_value > 0.05 ~ "N.S."
    ))

  all_results_df <- bind_rows(all_results_df, anova_table_formatted)

}

all_results_df

```

##	model	predictor	Chisq	Df	p_value	p_value_f
## 1	sprich_fm_2022	treatment	3.9325957	1	4.735879e-02	0.047
## 2	sprich_fm_2023	treatment	13.8296244	1	2.001554e-04	< 0.001
## 3	sprich_native_fm_2022	treatment	3.3589343	1	6.684131e-02	0.067
## 4	sprich_native_fm_2023	treatment	6.0137589	1	1.419476e-02	0.014
## 5	shannon_fm_2022	treatment	52.8441063	1	3.611006e-13	< 0.001
## 6	shannon_fm_2023	treatment	61.6346404	1	4.134800e-15	< 0.001
## 7	shnat_fm_2022	treatment	0.1228404	1	7.259741e-01	0.73
## 8	shnat_fm_2023	treatment	0.5300717	1	4.665767e-01	0.47
## 9	ef_fm_2022	treatment	95.1067629	1	1.803894e-22	< 0.001
## 10	ef_fm_2023	treatment	18.8160958	1	1.439472e-05	< 0.001
## 11	nf_fm_2022	treatment	6.2402313	1	1.248802e-02	0.012
## 12	nf_fm_2023	treatment	6.5744425	1	1.034534e-02	0.01
##	significance					
## 1			*			
## 2			*			
## 3			N.S.			
## 4			*			
## 5			*			
## 6			*			
## 7			N.S.			
## 8			N.S.			
## 9			*			
## 10			*			
## 11			*			
## 12			*			

```
write.csv(  
  all_results_df,  
  file = "analyses_manuscript/stats_results/single_year_model_results.csv",  
  row.names = FALSE  
)
```