Generalised linear mixed effects modelling for determining covariates of alpha

Jacob Westaway

Last updated on 2021-08-15

About

To examine the effect of several covariates on Shannon diversity, a generalised linear mixed effects regression model was created using lme4. Shannon diversity was calculated at the ASV level (normalised through TSS), and continuous predictors were scaled and centered. Multicollinearity was assessed with the AED package, and collinear variables were removed. To control for high inter-individual variation in the preterm infant microbiome, the infants identification was included as a random factor.

After creation of the initial model with lme4, backwards selection was used to find the least complex, yet adequate, model by comparing Akaike's Information Criterion (AIC) scores and removing predictors that did not contribute to variation in the model. A post-hoc pairwise Tukey comparison (correcting for multiple comparisons) was used to assess the effect of covariates using the emmeans package.

The code to create the data objects used in this workflow can be found in the 'Pipeline.Rmd'.

Packages

```
sapply(c("phyloseq", "tidyverse", "knitr", "lme4", "emmeans", "gridExtra",
         "MuMIn", "aods3", "sjPlot", "ggpubr", "lmerTest", "car"),
        require, character.only = TRUE)
```

Subset data.

- Subset ps to exclude SCN and NA values.
- Scale continuous variables.
- Estimate richness and save as object.
- create a new variable column with rownames.
- merge alpha diversity estimates (*ps_alpha_div*) with the metadata (*samdf*) by the *Label* column (orignially row names), for downstream analysis.

```
ps.NICU <- subset_samples(ps,
                 Primary_Group == "NICU" &
                 (Type == "Admission" | Type == "Discharge"))

centre_and_scale <- function(data){
# get numeric variables
data2 <- data %>%
  select_if(is.numeric)
# entering and scaling over variables
data3 <- sapply(data2, function(x) scale(x, center=T, scale = 2*sd(x))) %>%
  as.data.frame() %>%
  rownames_to_column("RowID")
# join scaled/centred data to non-numeric data
data %>%
  select_if(negate(is.numeric)) %>%
  rownames_to_column("RowID") %>%
  left_join(data3, by = "RowID") %>%
  select(-RowID)
}


sample_data(ps.NICU) <- sample_data(ps.NICU) %>%
  unclass() %>%
  as.data.frame() %>%
  filter(!is.na(Days_on_antibiotics)) %>% # omit NA samples for modelling
  filter(!is.na(Days_since_birth)) %>%
  centre_and_scale() %>%
  mutate("Sample" = Label) %>% # redo rownames to save it back into the original ps object
  column_to_rownames("Sample")

ps_alpha_div <- ps.NICU %>%
                estimate_richness(measures = c("Shannon")) %>%
                add_column(Label = row.names(sample_data(ps.NICU)))

ps_samp <- sample_data(ps.NICU) %>%
  unclass() %>%
  data.frame() %>%
  left_join(ps_alpha_div, by = "Label")
```

Centre and scale data

```
# define centre and scale function
centre_and_scale <- function(data){
# get numeric variables
data2 <- data %>%
  select_if(is.numeric)
# entering and scaling over variables
data3 <- sapply(data2, function(x) scale(x, center=T, scale = 2*sd(x))) %>%
  as.data.frame() %>%
  rownames_to_column("RowID")
# join scaled/centred data to non-numeric data
data %>%
  select_if(negate(is.numeric)) %>%
  rownames_to_column("RowID") %>%
  left_join(data3, by = "RowID") %>%
  select(-RowID)
}
```

```
glm_data <- ps_metadata %>%
  mutate(Shannon = as.factor(Shannon)) %>%
  centre_and_scale() %>%
  mutate(Shannon = as.character(Shannon)) %>%
  mutate(Shannon = as.numeric(Shannon))
```

Test for collinearity

```
# defin myvif function
myvif <- function(mod) {
  v <- vcov(mod)
  assign <- attributes(model.matrix(mod))$assign
  if (names(coefficients(mod)[1]) == "(Intercept)") {
    v <- v[-1, -1]
    assign <- assign[-1]
  } else warning("No intercept: vifs may not be sensible.")
  terms <- labels(terms(mod))
  n.terms <- length(terms)
  if (n.terms < 2) stop("The model contains fewer than 2 terms")
  if (length(assign) > dim(v)[1] ) {
    diag(tmp_cor)<-0
    if (any(tmp_cor==1.0)){
      return("Sample size is too small, 100% collinearity is present")
    } else {
      return("Sample size is too small")
    }
  }
  R <- cov2cor(v)
  detR <- det(R)
  result <- matrix(0, n.terms, 3)
  rownames(result) <- terms
  colnames(result) <- c("GVIF", "Df", "GVIF^(1/2Df)")
  for (term in 1:n.terms) {
    subs <- which(assign == term)
    result[term, 1] <- det(as.matrix(R[subs, subs])) * det(as.matrix(R[-subs, -subs]))/detR
    result[term, 2] <- length(subs)
  }
  if (all(result[, 2] == 1)) {
    result <- data.frame(GVIF=result[, 1])
  } else {
    result[, 3] <- result[, 1]^(1/(2 * result[, 2]))
  }
  invisible(result)
}

# corvif
corvif <- function(data) {
  data <- as.data.frame(data)

  form    <- formula(paste("fooy ~ ",paste(strsplit(names(data)," "),collapse = " + ")))
  data  <- data.frame(fooy = 1 + rnorm(nrow(data)) ,data)
  lm_mod  <- lm(form,data) # runs linear model with above formula and metadata

  cat("\n\nVariance inflation factors\n\n")
  print(myvif(lm_mod))
}

ps_samp %>%
  mutate(Days_since_birth = as.numeric(difftime(.$Date_Collected, .$DOB, units = "days"))) %>%
  mutate(Gest_at_collection = Days_since_birth + Gestational.Age.at.Birth) %>%
  select(Type, Feeding.Type , NEC , Sepsis , Mode.of.Delivery , Chorioamnionitis ,
          Preeclampsia, ROP, Diabetes , Antenatal.Antibiotics, Antenatal..Infections,
         Gest_at_collection, Prolonged..Membrane..Rupture, Died, Days_on_antibiotics) %>%
  corvif()
```

```
##
##
## Variance inflation factors
##
##                                  GVIF Df GVIF^(1/2Df)
## Type                         1.916320  1     1.384312
## Feeding.Type                 1.689688  2     1.140123
## NEC                          1.575315  1     1.255116
## Sepsis                       1.708989  1     1.307283
## Mode.of.Delivery             1.443920  1     1.201632
## Chorioamnionitis             1.829789  1     1.352697
## Preeclampsia                 1.481212  1     1.217050
## ROP                          1.306064  1     1.142831
## Diabetes                     1.324462  1     1.150853
## Antenatal.Antibiotics        1.703479  1     1.305174
## Antenatal..Infections        1.358500  1     1.165547
## Gest_at_collection           2.234118  1     1.494697
## Prolonged..Membrane..Rupture 1.816324  1     1.347711
## Died                         1.725737  1     1.313673
## Days_on_antibiotics          2.215848  1     1.488572
```

Fit Model

```
global <- lme4::lmer(Shannon ~  (Mode.of.Delivery + Feeding.Type +
                    Gestational.Age.at.Birth  + Antenatal.Antibiotics +
                    Antenatal..Infections + NEC + Sepsis +
                    Chorioamnionitis + Died +
                    Prolonged..Membrane..Rupture + Preeclampsia + Gest_at_collection +
                    Diabetes + ROP + Days_on_antibiotics) * Type + (1|URN), data = ps_samp)

summary(lmer(Shannon ~  (Mode.of.Delivery + Feeding.Type +
                    Gestational.Age.at.Birth  + Antenatal.Antibiotics +
                    Antenatal..Infections + NEC + Sepsis +
                    Chorioamnionitis + Neonatal.Antibiotics + Died +
                    Prolonged..Membrane..Rupture + Preeclampsia + Gest_at_collection +
                    Diabetes + ROP) * Type + (1|URN), data = ps_samp))
```

- Calculate the goodness of fit (how the sample data fits the distribution), the Pearsons Chi Square coefficient (how likely observed differences arose by chance) and the R2.
- Calculate these again post bakwards selection.

```
gof(global)
sum(residuals(global,"pearson")^2)
r.squaredGLMM(global)
```

Backwards Selection.

- Define a function that determines what variable is contributing least to the model, as determined by AIC score.
- Then apply that function to the model, and subsequent models, removing variables from the model that are not contributing (first from the interaction and then from the model entirely).

```
dfun <- function(x) {
  x$AIC <- x$AIC-min(x$AIC)
  names(x)[2] <- "dAIC"
  x
}

dfun(drop1(global))
```

```
global2 <- lme4::lmer(Shannon ~  Sepsis + Antenatal.Antibiotics + Gestational.Age.at.Birth
                      + Gest_at_collection + Feeding.Type + (Mode.of.Delivery +  NEC +
                     Preeclampsia + ROP + Days_on_antibiotics) * Type + (1|URN),
                     data = ps_samp)

dfun(drop1(global2))
```

- Calculate the goodness of fit (how the sample data fits the distribution), the Pearsons Chi Square coefficient (how likely observed differences arose by chance) and the R2.

```
gof(global)
sum(residuals(global,"pearson")^2)
r.squaredGLMM(global)
```

Final Model

```
lmer(Shannon ~  Sepsis + Antenatal.Antibiotics + Gestational.Age.at.Birth  +
                        Gest_at_collection + Feeding.Type + (Mode.of.Delivery +  NEC +
                     Preeclampsia + ROP + Days_on_antibiotics)* Type + (1|URN),
                     data = ps_samp) %>%
                  summary()
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Shannon ~ Sepsis + Antenatal.Antibiotics + Gestational.Age.at.Birth +
##     Gest_at_collection + Feeding.Type + (Mode.of.Delivery + NEC +
##     Preeclampsia + ROP + Days_on_antibiotics) * Type + (1 | URN)
##    Data: ps_samp
##
## REML criterion at convergence: 322.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4067 -0.5455 -0.0865  0.4611  2.6937
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  URN      (Intercept) 0.0000   0.0000
##  Residual             0.7105   0.8429
## Number of obs: 129, groups:  URN, 82
##
## Fixed effects:
##                                       Estimate Std. Error       df t value
## (Intercept)                             1.5855     0.2768 111.0000   5.728
## SepsisYes                              -0.8547     0.3939 111.0000  -2.170
## Antenatal.AntibioticsYes               -0.3022     0.1925 111.0000  -1.570
## Gestational.Age.at.Birth                0.4040     0.2332 111.0000   1.733
## Gest_at_collection                     -0.3296     0.2173 111.0000  -1.516
## Feeding.TypeBreastmilk and Formula      0.2697     0.2042 111.0000   1.320
## Feeding.TypeFormula                     0.6897     0.1883 111.0000   3.663
## Mode.of.DeliveryVaginal                 0.3141     0.2367 111.0000   1.327
## NECYes                                  0.4811     0.3985 111.0000   1.207
## PreeclampsiaYes                         0.2970     0.3270 111.0000   0.908
## ROPYes                                 -0.8201     0.2341 111.0000  -3.503
## Days_on_antibiotics                    -0.9952     0.7510 111.0000  -1.325
## TypeDischarge                           0.3389     0.2910 111.0000   1.165
## Mode.of.DeliveryVaginal:TypeDischarge  -0.6669     0.3475 111.0000  -1.919
## NECYes:TypeDischarge                   -1.1363     0.6947 111.0000  -1.636
## PreeclampsiaYes:TypeDischarge          -0.8495     0.4483 111.0000  -1.895
## ROPYes:TypeDischarge                    1.1471     0.3460 111.0000   3.315
## Days_on_antibiotics:TypeDischarge       1.3632     0.7387 111.0000   1.845
##                                       Pr(>|t|)
## (Intercept)                           8.82e-08 ***
## SepsisYes                             0.032151 *
## Antenatal.AntibioticsYes              0.119270
## Gestational.Age.at.Birth              0.085929 .
## Gest_at_collection                    0.132266
## Feeding.TypeBreastmilk and Formula    0.189392
## Feeding.TypeFormula                   0.000384 ***
## Mode.of.DeliveryVaginal               0.187090
## NECYes                                0.229875
## PreeclampsiaYes                       0.365704
## ROPYes                                0.000664 ***
## Days_on_antibiotics                   0.187843
## TypeDischarge                         0.246556
## Mode.of.DeliveryVaginal:TypeDischarge 0.057547 .
## NECYes:TypeDischarge                  0.104729
## PreeclampsiaYes:TypeDischarge         0.060711 .
## ROPYes:TypeDischarge                  0.001238 **
## Days_on_antibiotics:TypeDischarge     0.067648 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Analysis of deviance

```
car::Anova(global2) %>%
  as.data.frame(row.names = NULL)
```

```
##                                 Chisq Df   Pr(>Chisq)
## Sepsis                     4.70819559  1 0.0300191442
## Antenatal.Antibiotics      2.46477820  1 0.1164241357
## Gestational.Age.at.Birth   3.00219091  1 0.0831520006
## Gest_at_collection         2.29940346  1 0.1294236973
## Feeding.Type              13.43910706  2 0.0012070770
## Mode.of.Delivery           0.03375398  1 0.8542311025
## NEC                        0.04982333  1 0.8233709827
## Preeclampsia               0.18614811  1 0.6661420367
## ROP                        2.99968662  1 0.0832806238
## Days_on_antibiotics        1.19348600  1 0.2746275155
## Type                       0.25193531  1 0.6157156488
## Mode.of.Delivery:Type      3.68272965  1 0.0549787731
## NEC:Type                   2.67560093  1 0.1018968570
## Preeclampsia:Type          3.59054418  1 0.0581092115
## ROP:Type                  10.98982312  1 0.0009161355
## Days_on_antibiotics:Type   3.40541984  1 0.0649825760
```

Post-hoc pairwiset testing, accounting for multiple comparisons

```
emmeans(global2, list(pairwise ~ Sepsis), adjust = "tukey")
```

```
## $‘emmeans of Sepsis‘
##  Sepsis emmean    SE    df lower.CL upper.CL
##  No      1.683 0.214  78.4     1.20     2.17
##  Yes     0.829 0.418 104.7    -0.12     1.78
##
## Results are averaged over the levels of: Antenatal.Antibiotics, Feeding.Type, Mode.of.Delivery, NEC,
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 2 estimates
##
## $‘pairwise differences of Sepsis‘
##  contrast estimate    SE   df t.ratio p.value
##  No - Yes    0.855 0.396 93.4 2.159   0.0334
##
```

```
## Results are averaged over the levels of: Antenatal.Antibiotics, Feeding.Type, Mode.of.Delivery, NEC,
## Degrees-of-freedom method: kenward-roger
```

```
emmeans(global2, list(pairwise ~ Feeding.Type), adjust = "tukey")
```

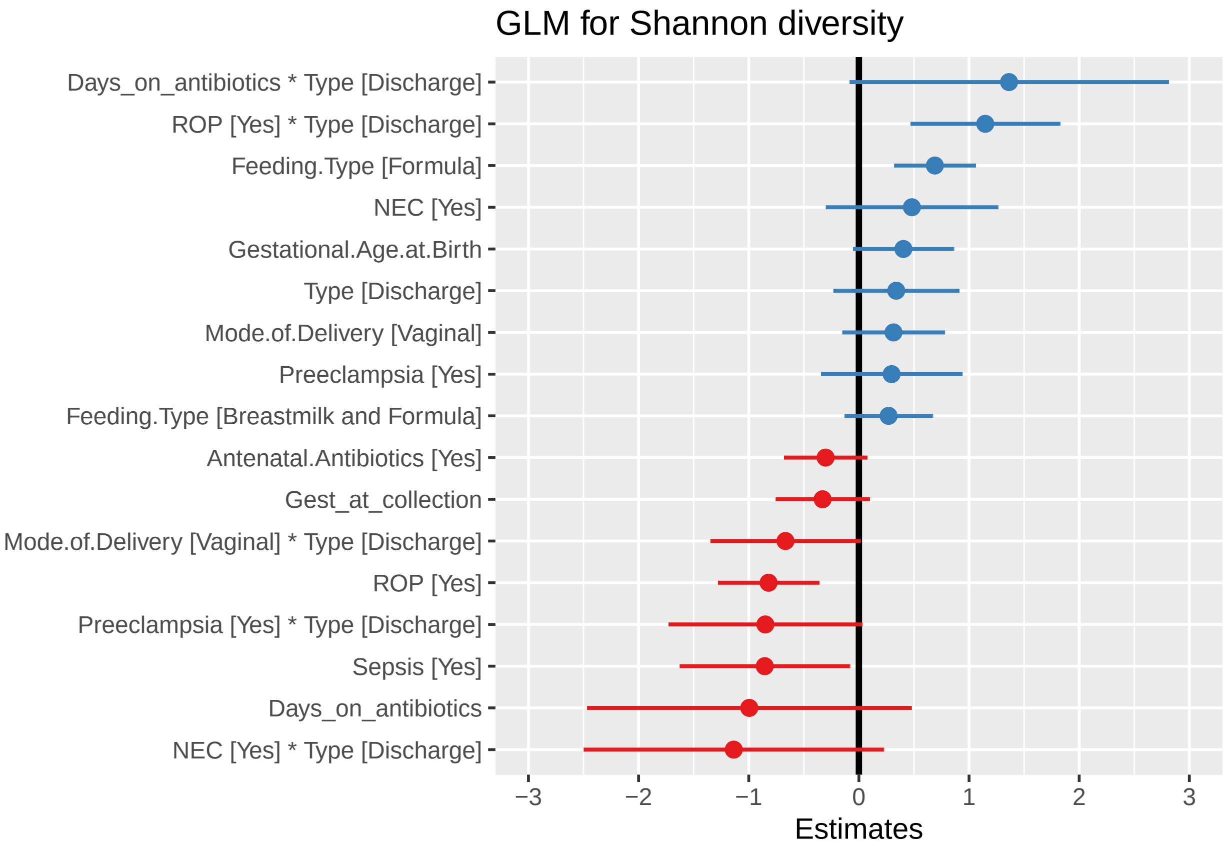
```
## $`emmeans of Feeding.Type`
##  Feeding.Type           emmean    SE    df lower.CL upper.CL
##  Breastmilk              0.936 0.277 104.5    0.265     1.61
##  Breastmilk and Formula  1.206 0.297  96.0    0.485     1.93
##  Formula                 1.626 0.304  90.5    0.887     2.37
##
## Results are averaged over the levels of: Sepsis, Antenatal.Antibiotics, Mode.of.Delivery, NEC, Preecl
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 3 estimates
##
## $`pairwise differences of Feeding.Type`
##  contrast                            estimate    SE   df t.ratio p.value
##  Breastmilk - Breastmilk and Formula    -0.27 0.206 64.4 -1.310  0.3950
##  Breastmilk - Formula                   -0.69 0.190 55.5 -3.639  0.0017
##  Breastmilk and Formula - Formula       -0.42 0.230 59.3 -1.824  0.1706
##
## Results are averaged over the levels of: Sepsis, Antenatal.Antibiotics, Mode.of.Delivery, NEC, Preecl
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
emmeans(global2, list(pairwise ~ ROP:Type), adjust = "tukey")
```

```
## $`emmeans of ROP, Type`
##  ROP Type      emmean    SE  df lower.CL upper.CL
##  No  Admission  1.873 0.290 110   1.1388     2.61
##  Yes Admission  1.053 0.329 109   0.2207     1.89
##  No  Discharge  0.886 0.431 110  -0.2065     1.98
##  Yes Discharge  1.213 0.452 110   0.0675     2.36
##
## Results are averaged over the levels of: Sepsis, Antenatal.Antibiotics, Feeding.Type, Mode.of.Delive
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 4 estimates
##
## $`pairwise differences of ROP, Type`
##  contrast                      estimate    SE  df t.ratio p.value
##  No Admission - Yes Admission     0.820 0.235 110  3.489  0.0038
##  No Admission - No Discharge      0.987 0.504 100  1.958  0.2109
##  No Admission - Yes Discharge     0.660 0.523 111  1.263  0.5883
##  Yes Admission - No Discharge     0.167 0.512 110  0.327  0.9879
##  Yes Admission - Yes Discharge   -0.160 0.530 105 -0.301  0.9904
##  No Discharge - Yes Discharge    -0.327 0.260 109 -1.259  0.5909
##
## Results are averaged over the levels of: Sepsis, Antenatal.Antibiotics, Feeding.Type, Mode.of.Delive
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```

Visulatisation: all variables in final model as an estimates sjplot

```
sjPlot::plot_model(global2,
                   vline.color = "black",
                   sort.est = TRUE,
                   title = "GLM for Shannon diversity")
```



Visualisation: plot significant variables as box plots.

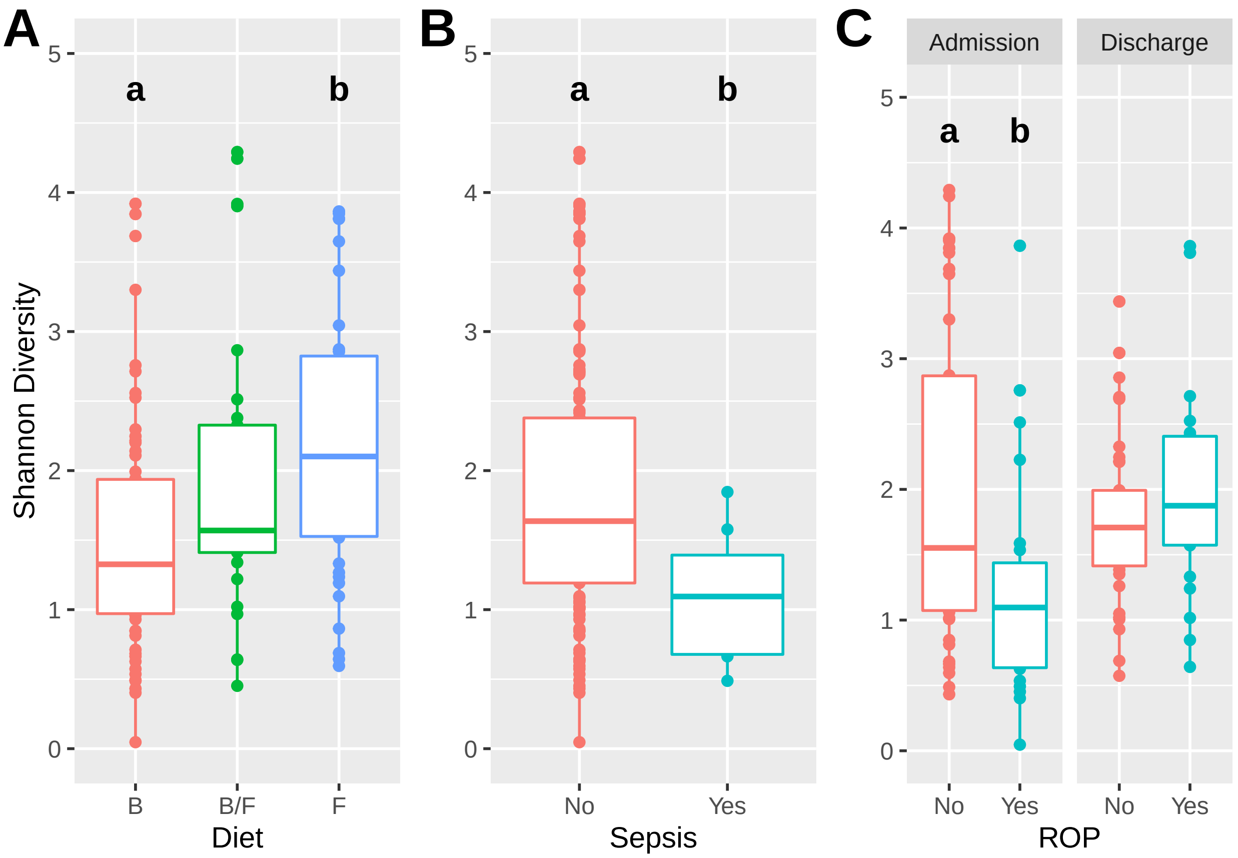
- Create a function for the boxplots that takes the data, the variable and any added annotation as arguments.
- Any other variable-specifc specifications for the plots can be added after the function. **eg.** Variables that interact with *Type* are faceted by the variable with `+ facet_wrap(~Type)`.
- Wrap all the plots into a function so intermediate objects don't need to be created in environment.

```
box_grid <- function(ps_samp){
# Diet
anno <- data.frame(xstar = c(1, 3), ystar = c(4.75, 4.75),lab = c("a", "b"))
ggplot_Feeding.Type <- (shannon_box_plot(ps_samp, "Feeding.Type", anno)  +
  scale_x_discrete(labels = c("B", "B/F", "F")) +
  xlab("Diet")) %>%
  annotate_figure(fig.lab = "A", fig.lab.face = "bold", fig.lab.size = 20)
# Sepsis
anno <- data.frame(xstar = c(1, 2), ystar = c(4.75, 4.75), lab = c("a", "b"))
ggplot_Sepsis <- (shannon_box_plot(ps_samp, "Sepsis", anno) +
                    ylab("")) %>%
  annotate_figure(fig.lab = "B", fig.lab.face = "bold", fig.lab.size = 20)

# ROP
anno <- data.frame(xstar = c(1, 2), ystar = c(4.75, 4.75),
        lab = c("a", "b"), Type = c("Admission", "Admission"))
ggplot_ROP <- (shannon_box_plot(ps_samp, "ROP", anno)  +
  facet_wrap(~Type)  +
  ylab("")) %>%
  annotate_figure(fig.lab = "C", fig.lab.face = "bold", fig.lab.size = 20)

# Create the grid
grid.arrange(ggplot_Feeding.Type, ggplot_Sepsis, ggplot_ROP, nrow = 1, ncol = 3)
}

box_grid(ps_samp)
```



```
ggsave("Figure_4.jpg", plot = (box_grid(ps_samp)), dpi = 600, height = 7, width = 10)
```

FINISHED