HEB1410 Gut Microbiome and Human Health

Computation Lab Section

Yijia Liow

2023-12-05

Outline

- Workshop on **final paper data anaylsis**
- **Your own gut microbiome** exploration
- **How to paper?**

Your own gut microbiome!

```
# Load necessary packages
library(tidyverse)
library(phyloseq)
library(Maaslin2)
library(lme4)
library(pheatmap)
library(RColorBrewer)
library(cowplot)
```

Load human gut microbiome data

```
# Load human microbiome data
ps_human <- readRDS("week-13/data/ps_human.rds")

# Convert phyloseq object to data frames
sample_df <- data.frame(sample_data(ps_human))
otu_df <- data.frame(otu_table(ps_human))
taxonomy_df <- data.frame(tax_table(ps_human))
```

Define a custom color palette

```
# Define a custom color palette with 20 distinct colors + gray for `Other`
claire_palette <- c(
  "#50394c", "#eea29a", "#f7cac9", "#d9ad7c", "#a2836e",
  "#d5f4e6", "#80ced6", "#fefbd8", "#ffef96", "#fbefcc",
  "#f9ccac", "#f4a688", "#7e4a35", "#cab577", "#dbceb0",
  "#838060", "#daebe8", "#bccad6", "#8d9db6", "#667292"
)

color_use <- c(claire_palette, "gray")
```

Choose your favorite color here

Or use a predefined palette

```
install.packages("wesanderson")
library("wesanderson")

# See all palettes
names(wes_palettes)

# Pick your favorite palette
palette <- wes_palette("Moonrise2")
```

Find your favorite palette here

Predominant phyla

```
# Glomming at the Phylum level
ps_human_phylum <- tax_glom(ps_human, "Phylum")
df_human_phylum <- psmelt(ps_human_phylum)

# Calculate relative abundance
df_human_phylum <- df_human_phylum %>%
  group_by(Sample) %>%
  mutate(RelativeAbundance = Abundance / sum(Abundance))

# List of specific mouseIDs you want to extract
specific_mouseIDs <- c("914072-01", "914072-02", "914072-03")

# Filter the dataframe for those specific mouseIDs
df_filtered_phylum <- df_human_phylum %>%
  filter(mouseID %in% specific_mouseIDs)
```

Predominant phyla

```r
# Identify the top 20 phyla based on total relative abundance
top_phyla_filtered <- df_filtered_phylum %>%
  group_by(Phylum) %>%
  summarize(TotalAbundance = sum(RelativeAbundance)) %>%
  ungroup() %>%
  slice_max(TotalAbundance, n = 20) %>%
  arrange(desc(TotalAbundance)) %>%
  pull(Phylum)

# Rename the Phylum column for entries not in the top phyla as 'Other'
df_filtered_phylum$Phylum <- ifelse(
  df_filtered_phylum$Phylum %in% top_phyla_filtered,
  df_filtered_phylum$Phylum, "Other")

df_filtered_phylum$Phylum <- factor(df_filtered_phylum$Phylum,
                                    levels = c(top_phyla_filtered, "Other"))
```

Plotting the top 20 phyla

```r
# Plot with the top 20 phyla and the 'Other' category
phyla_plot <- df_filtered_phylum %>%
  ggplot(aes(x = Sample, y = RelativeAbundance, fill = Phylum)) +
  geom_bar(stat = "identity",
           position = "stack",
           color = "gray80",
           linewidth = 0.2) +
  theme_minimal() +
  scale_fill_manual(values = color_use,
                    breaks = c(top_phyla_filtered, "Other")) +
  labs(x = "Sample", y = "Relative Abundance") +
  theme(axis.title.x = element_blank(),
        legend.position = "bottom",
        legend.text = element_text(size = 6),
        legend.key.size = unit(0.5, 'lines')) +
  guides(fill = guide_legend(nrow = 3))

# Save the plot
ggsave("phyla_plot.pdf", phyla_plot, width = 12, height = 8, dpi = 300)
```

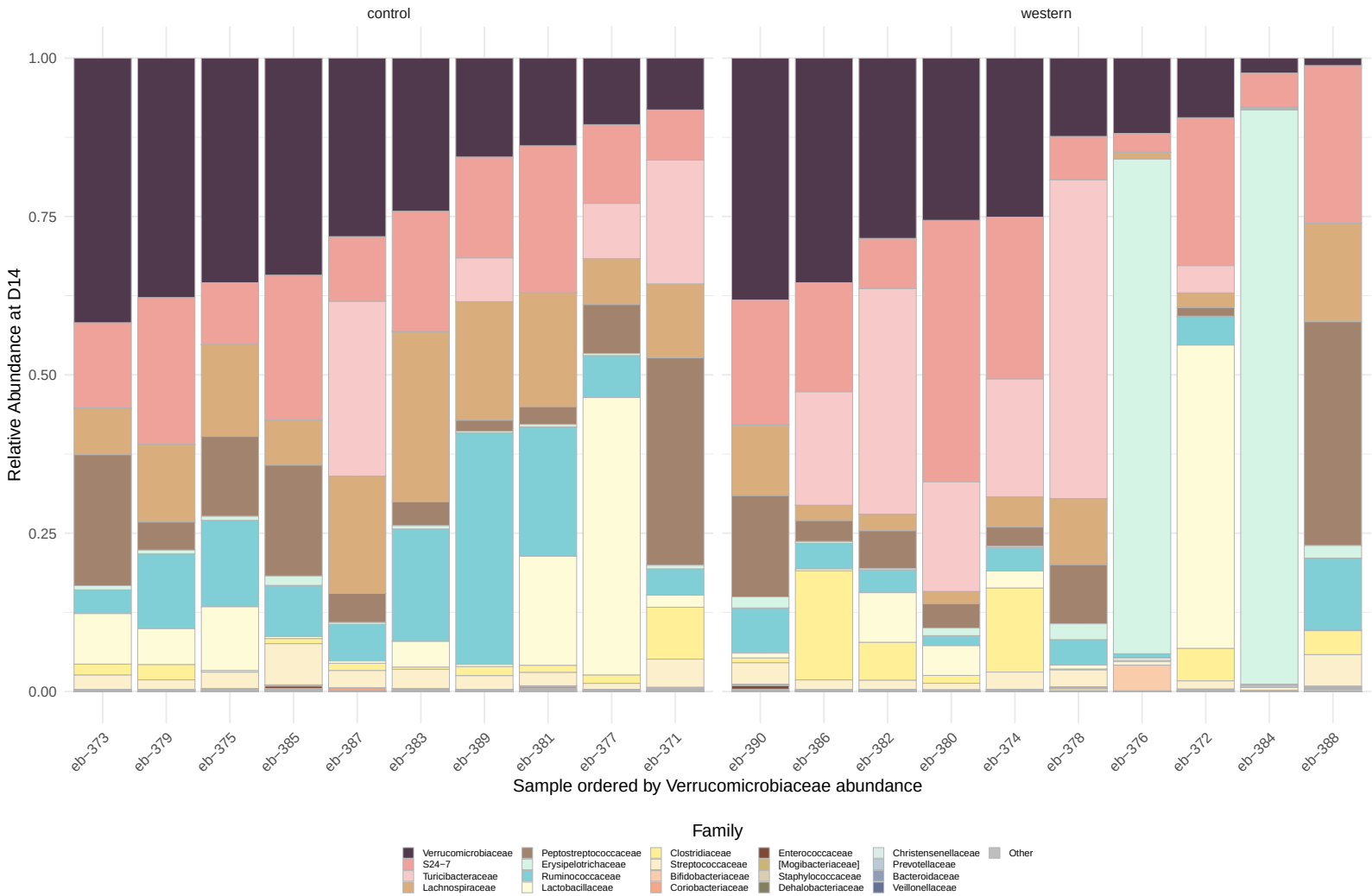
Combine multiple plots

```r
# Install the cowplot package
install.packages("cowplot")

# Load the package
library(cowplot)

# Combine your figures into one plot
combined_plot <- plot_grid(
  phyla_plot,
  family_plot,
  genus_plot,
  labels = c("A", "B", "C"),
  ncol = 1,
  nrow = 3
)

# Display and save the plot
print(combined_plot)
ggsave("combined_plot.pdf", combined_plot, width = 12, height = 8)
```

Refining gut microbiome plots

- Taxonomic plots: Order samples by taxon-of-interest
- Alpha-diversity plots: Multiple comparisons
- Beta-diversity plots: Adding R^2 and p-values

How to order samples by taxon-of-interest?



How to order samples by taxon-of-interest?

```r
# Load microbiome data
ps_course <- readRDS("computation/01-resources/ps_course.rds")

# Taxonomic composition at the Family level
ps_course_family <- tax_glom(ps_course, "Family")
df_course_family <- psmelt(ps_course_family)

# Calculate relative abundance
df_course_family <- df_course_family %>%
  group_by(Sample) %>%
  mutate(RelativeAbundance = Abundance / sum(Abundance))
```

How to order samples by taxon-of-interest?

```r
# Subset for D14 timepoint
df_d14_family <- df_course_family %>%
  filter(timepoint == "D14")

# Identify the top 20 families based on total relative abundance at D14
top_families_d14 <- df_d14_family %>%
  group_by(Family) %>%
  summarize(TotalAbundance = sum(RelativeAbundance)) %>%
  ungroup() %>%
  slice_max(TotalAbundance, n = 20) %>%
  arrange(desc(TotalAbundance)) %>%
  pull(Family)

# Order samples based on 'Verrucomicrobiaceae' abundance at D14
sample_order_d14 <- df_d14_family %>%
  filter(Family == "Verrucomicrobiaceae") %>%
  group_by(Sample) %>%
  summarize(VerrucomicrobiaceaeAbundance = sum(RelativeAbundance)) %>%
  arrange(desc(VerrucomicrobiaceaeAbundance)) %>%
  pull(Sample)
```

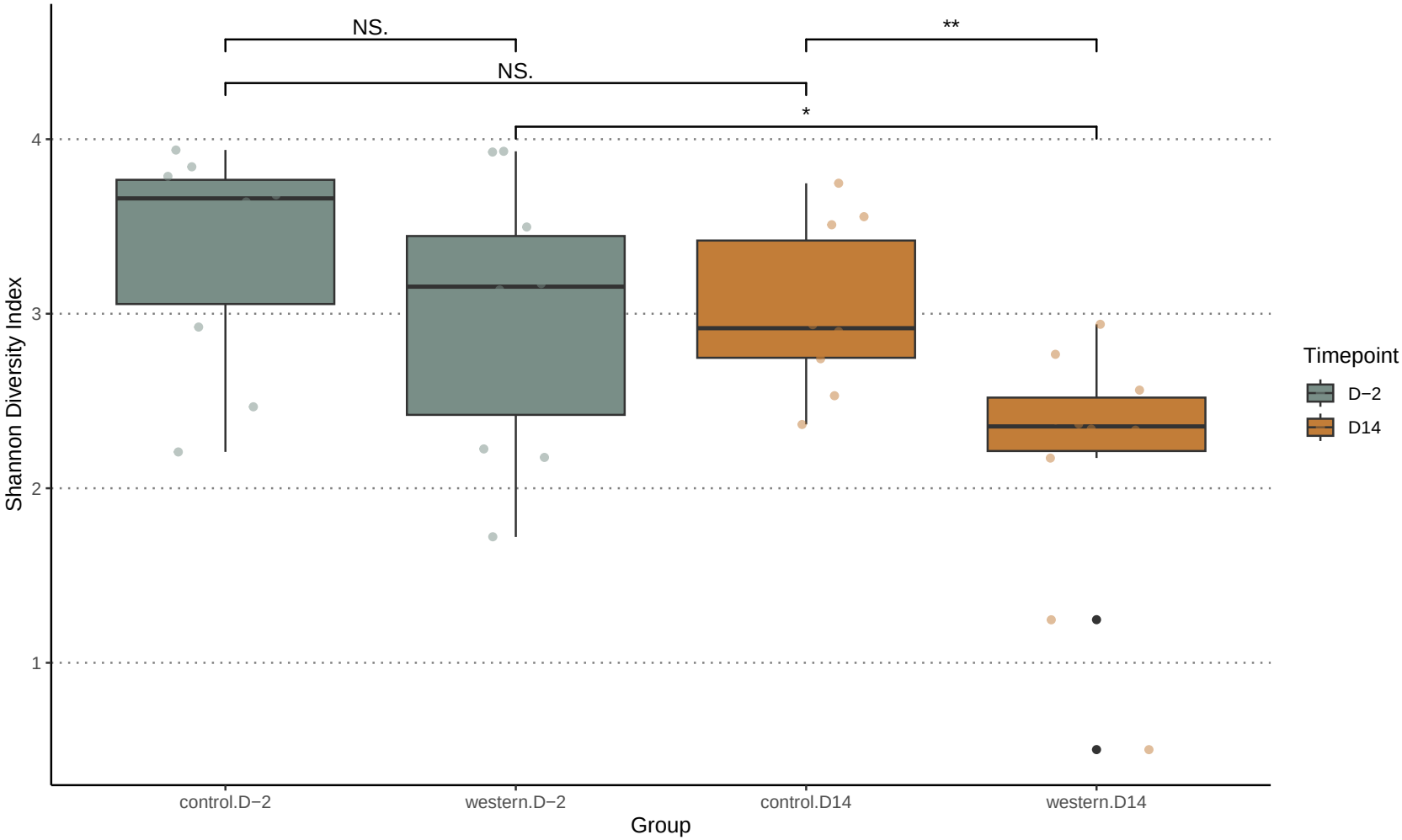
How to order samples by taxon-of-interest?

```
# Rename the Genus column
df_d14$Genus <- ifelse(
  df_d14$Genus %in% top_genera_d14, df_d14$Genus, "Other"
)

# Adjust factor levels for plotting genera
df_d14$Genus <- factor(df_d14$Genus,
                       levels = c(top_genera_d14, "Other"))
```

How to order samples by taxon-of-interest?

```
family_post <- df_d14_family %>%
  mutate(Sample = factor(Sample, levels = sample_order_d14)) %>%
  mutate(Family = factor(Family,
         levels = c(top_families_d14, "Other"))) %>%
  ggplot(aes(x = Sample, y = RelativeAbundance, fill = Family)) +
  geom_bar(stat = "identity",
           position = "stack",
           color = "gray70",
           linewidth = 0.2) +
  facet_wrap(~treatment, scales = "free_x") +
  theme_minimal() +
  scale_fill_manual(values = color_use,
                    breaks = c(top_families_d14, "Other")) +
  labs(x = "Sample ordered by Verrucomicrobiaceae abundance",
       y = "Relative Abundance at D14") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "bottom",
        legend.text = element_text(size = 6),
        legend.key.size = unit(0.5, 'lines')) +
  guides(fill = guide_legend(nrow = 4))
```

Refining alpha-diversity plot



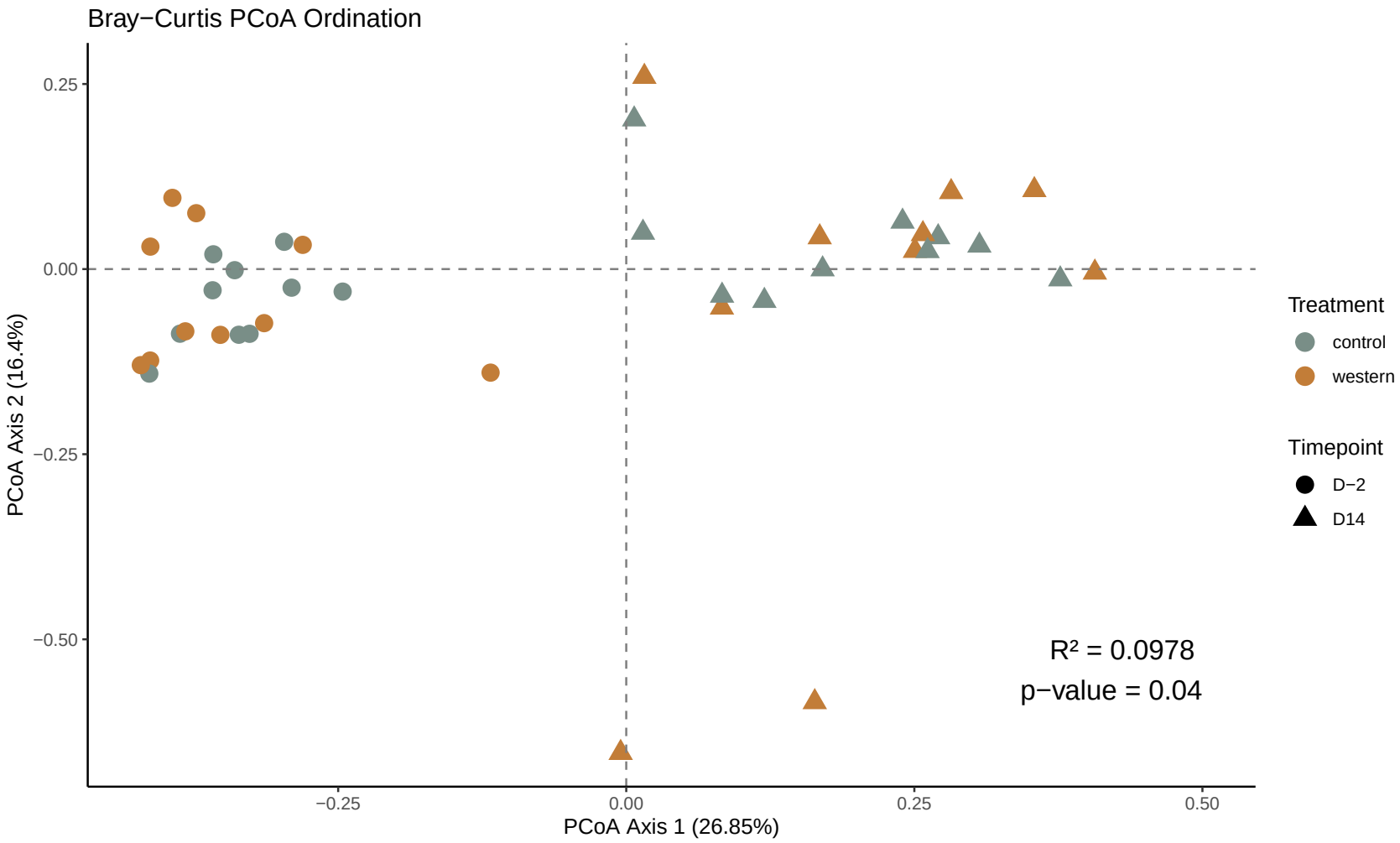
Refining alpha-diversity plot

```r
# Create a new interaction variable for group comparisons
alpha_diversity_joined$group <-
  interaction(alpha_diversity_joined$treatment,
              alpha_diversity_joined$Timepoint)

# Define the base plot
base_plot <- ggplot(alpha_diversity_joined, aes(x = group, y = Shannon)) +
  geom_boxplot(aes(fill = Timepoint)) +
  geom_jitter(aes(color = Timepoint),
              position = position_jitter(width = 0.2), alpha = 0.5) +
  scale_fill_manual(values = palette) +
  scale_color_manual(values = palette) +
  labs(x = "Group", y = "Shannon Diversity Index") +
  theme_classic()
```

Refining alpha-diversity plot

```r
# Specify y positions for the significance annotations
y_positions <- c(4.4, 4.4, 4.15, 3.9)

# Create the plot with significance annotations
p <- base_plot +
  geom_signif(comparisons = list(
    c("control.D-2", "western.D-2"),
    c("control.D14", "western.D14"),
    c("control.D-2", "control.D14"),
    c("western.D-2", "western.D14")
  ), map_signif_level = TRUE,
  y_position = y_positions, tip_length = 0.02, vjust = 0.2)
```

Refining beta-diversity plot



Refining beta-diversity plot

```
# Use your actual R-squared and p-values
r_squared_value <- 0.0978
p_value <- 0.04

# Format the p-value string properly
p_value_formatted <- ifelse(p_value < 0.001, "< 0.001",
                            as.character(signif(p_value, digits = 2)))
```

Refining beta-diversity plot

```
# Create the plot with the R² and p-value annotations
beta_d14 <- ord_bray_data_subset %>%
  ggplot(aes(x = Axis.1, y = Axis.2,
             color = treatment,
             shape = timepoint)) +
  geom_point(size = 2.5) +
  theme_classic() +
  scale_color_manual(values = palette) +
  labs(x = paste("PCoA Axis 1 (", axis1_var, "%)", sep = ""),
       y = paste("PCoA Axis 2 (", axis2_var, "%)", sep = ""),
       color = "Treatment",
       shape = "Timepoint",
       title = "Bray-Curtis PCoA Ordination") +
  annotate("text", x = 0.5, y = -0.5,
           label = paste("R² =", r_squared_value,
                         "\np-value =", p_value_formatted),
           hjust = 1, vjust = 1, size = 5, colour = "black") +
  theme(legend.position = c(0.9, 0.2))
```

How to paper[1]?

1. Finalize **empirical analyses** (figures and tables)
2. Determine a (informative) **title** for your paper
3. Draft the **abstract**
   - Concisely describe the problem and your solution to it
   - Remaining: Convey all the important information in your paper
4. **Introduction:**
   - Brief discussion of the motivation and main contributions
   - Elaborating each sentence in the abstract
5. Once that's done, remaining sections should follow naturally

[1] Source: https://imai.fas.harvard.edu/teaching/files/HowToPaper.pdf

How to paper?

- Each paragraph in Introduction becomes a section to describe your methods and empirical results
- **Precisely describe your analyses**: Readers should be able to replicate your findings without talking to you
- **Justify your decisions**
  - Why did you choose a particular method?
  - Why did you exclude a particular sample?
  - Why did you use a particular threshold?
- A *top-down* approach: Beginning of a paragraph/section/subsection should give the main message, and the rest to elaborate on it

Resources beyond this course

- R for Data Science
- R Markdown: The Definitive Guide
- R Graphics Cookbook
- R cheatsheets