

# Coding\_Challenge\_4

Anet Filipova, Brynleigh Payne

2025-02-27

Noel, Z.A., Roze, L.V., Breunig, M., Trail, F. 2022. Endophytic fungi as promising biocontrol agent to protect wheat from Fusarium graminearum head blight. Plant Disease. [DOI:] (<https://doi.org/10.1094/PDIS-06-21-1253-RE>) Installing the package necessary for generating a .pdf file

Question 1:

- What's a YAML header? It stands for yet another markdown language. It contains information such as title, author, date of creation, and the output files in the format we want them to be generated (e.g. .html, .word, .pdf, etc.)
- What's a literate programming? In an Rmarkdown file you can explain all the steps with text and then include a code chunk. Following that code chunk will be the output of the code.

Question 2: Read data, install packages and make a custom colored palette

```
Myco <- read.csv("MycotoxinData.csv", na.strings = "na")
library(ggplot2)
library(ggpubr)
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.3.3
```

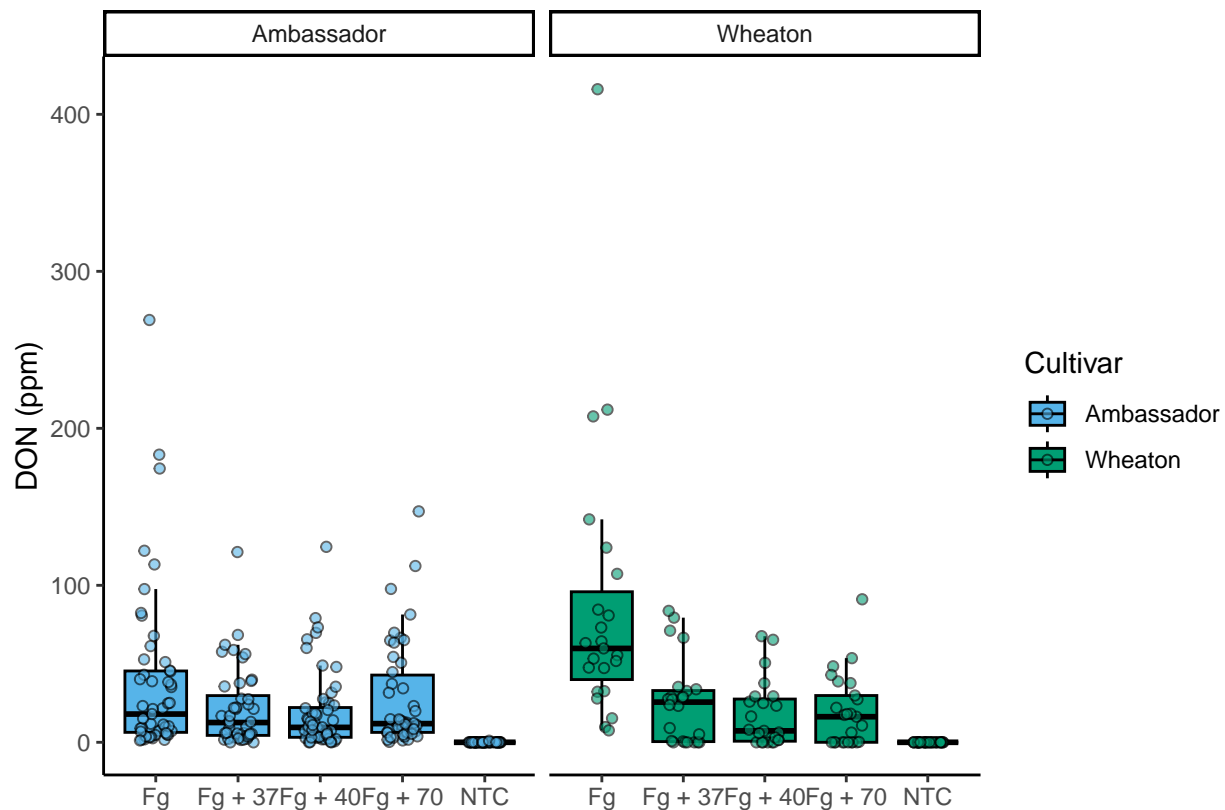
```
# Make vector containing custom color palette
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
```

This code creates a boxplot of DON by Treatment

```
# Create a boxplot of DON by Treatment
DONplot <- ggplot(Myco, aes(x = Treatment, y = DON, color = Cultivar, fill = Cultivar)) + # A - Define
  geom_boxplot(position = position_dodge(), color = "black", outliers = FALSE) + # A - Add boxplots with
  xlab("") + # C - Label the x-axis
  ylab("DON (ppm)") + # C - Label the y-axis
  geom_point(alpha = 0.6, color = "black", position = position_jitterdodge(dodge.width = 0.9), shape = 1) +
  theme_classic() + # D - Use a classic theme for the plot
  scale_color_manual(values = c(cbbPalette[[3]], cbbPalette[[4]]), name = "", labels = c("Ambassador", "Ambassador")) +
  scale_fill_manual(values = c(cbbPalette[[3]], cbbPalette[[4]])) + # A - Manually set colors
  facet_wrap(~Cultivar, ncol = 2, scales = "free_x") # E - Create separate panels for each Crop, allow
DONplot
```

```
## Warning: Removed 8 rows containing non-finite outside the scale range
## ('stat_boxplot()').
```

```
## Warning: Removed 8 rows containing missing values or values outside the scale range
## ('geom_point()').
```



This code replots the DON plot with specified axes

```
# Make variables "Time_Point" and "Crop" into a Factor instead of an Integer
Myco$Treatment <- as.factor(Myco$Treatment)
str(Myco)
```

```
## 'data.frame':  375 obs. of  6 variables:
## $ Treatment   : Factor w/ 5 levels "Fg","Fg + 37",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Cultivar    : chr  "Wheaton" "Wheaton" "Wheaton" "Wheaton" ...
## $ BioRep      : int   2 2 2 2 2 2 2 2 2 3 ...
## $ MassperSeed_mg: num   10.29 12.8 2.85 6.5 10.18 ...
## $ DON         : num   107.3 32.6 416 211.9 124 ...
## $ X15ADON     : num    3 0.85 3.5 3.1 4.8 3.3 6.9 2.9 2.1 0.71 ...
```

```
# Change ggplot's automatic alphabetical order
Myco$Treatment <- factor(Myco$Treatment, levels = c("NTC", "Fg", "Fg + 37", "Fg + 40", "Fg + 70"))
# Plot again
DONplot <- ggplot(Myco, aes(x = Treatment, y = DON, color = Cultivar, fill = Cultivar)) + # A - Define
  geom_boxplot(position = position_dodge(), color = "black", outliers = FALSE) + # A - Add boxplots with
  xlab("") + # C - Label the x-axis
  ylab("DON (ppm)") + # C - Label the y-axis
  geom_point(alpha = 0.6, color = "black", position = position_jitterdodge(dodge.width = 0.9), shape = 1)
```

```

theme_classic() + # D - Use a classic theme for the plot
scale_color_manual(values = c(cbbPalette[[3]], cbbPalette[[4]]), name = "", labels = c("Ambassador", "Wheaton")) + # A - Manually set colors
scale_fill_manual(values = c(cbbPalette[[3]], cbbPalette[[4]])) + # A - Manually set colors
facet_wrap(~Cultivar, ncol = 2, scales = "free_x") # E - Create separate panels for each Crop, allowing
DONplot # Save as object, run object

```

```

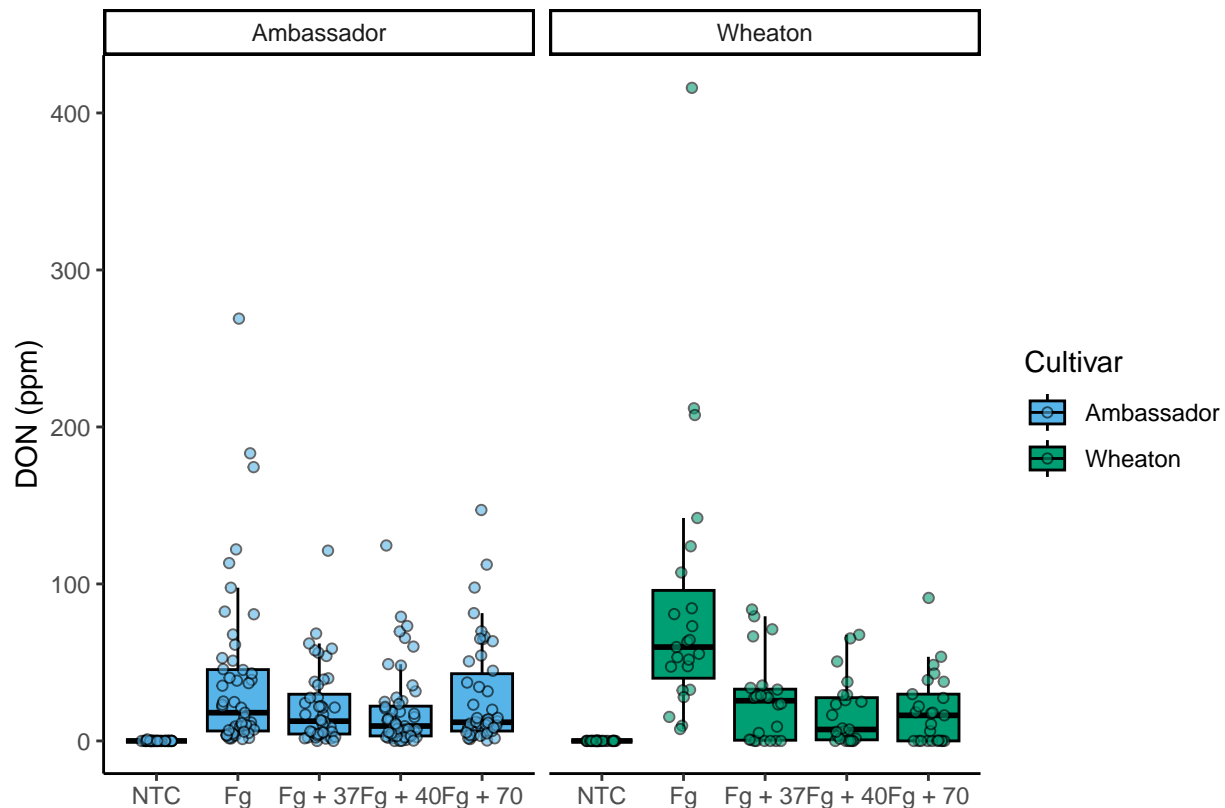
## Warning: Removed 8 rows containing non-finite outside the scale range
## ('stat_boxplot()').

```

```

## Warning: Removed 8 rows containing missing values or values outside the scale range
## ('geom_point()').

```



This code makes a X15DON plot

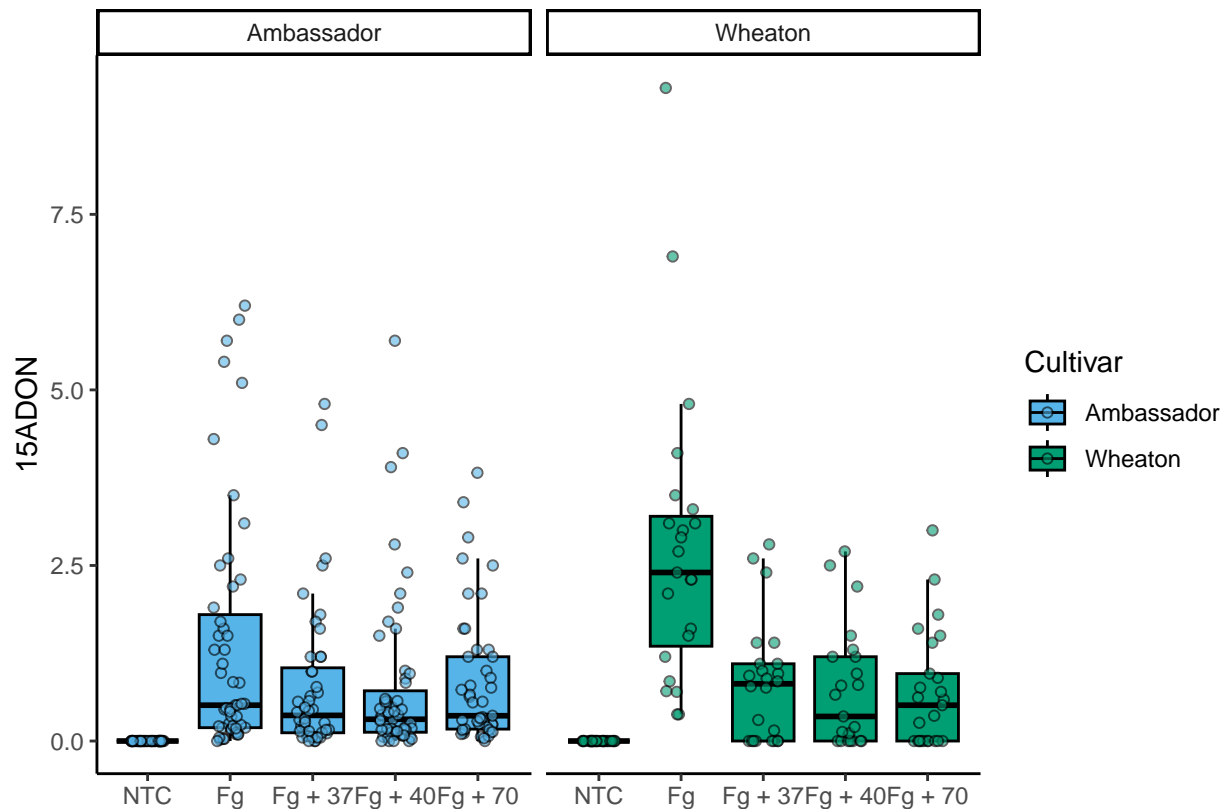
```

# Change the y to X15ADON in one plot & change the y to MassperSeed_mg in another plot
X15ADONplot <- ggplot(Myco, aes(x = Treatment, y = X15ADON, color = Cultivar, fill = Cultivar)) + # De
  geom_boxplot(position = position_dodge(), color = "black", outliers = FALSE) + # Add boxplots with do
  xlab("") + # Label the x-axis
  ylab("15ADON") + # Label the y-axis
  geom_point(alpha = 0.6, color = "black", position = position_jitterdodge(dodge.width = 0.9), shape = 2) +
  theme_classic() + # Use a classic theme for the plot
  scale_color_manual(values = c(cbbPalette[[3]], cbbPalette[[4]]), name = "", labels = c("Ambassador", "Wheaton")) + # A - Manually set colors
  scale_fill_manual(values = c(cbbPalette[[3]], cbbPalette[[4]])) + # A - Manually set colors
  facet_wrap(~Cultivar, ncol = 2, scales = "free_x") # Create separate panels for each Crop, allowing
X15ADONplot # Save as object, run object

```

```
## Warning: Removed 10 rows containing non-finite outside the scale range
## ('stat_boxplot()').
```

```
## Warning: Removed 10 rows containing missing values or values outside the scale range
## ('geom_point()').
```

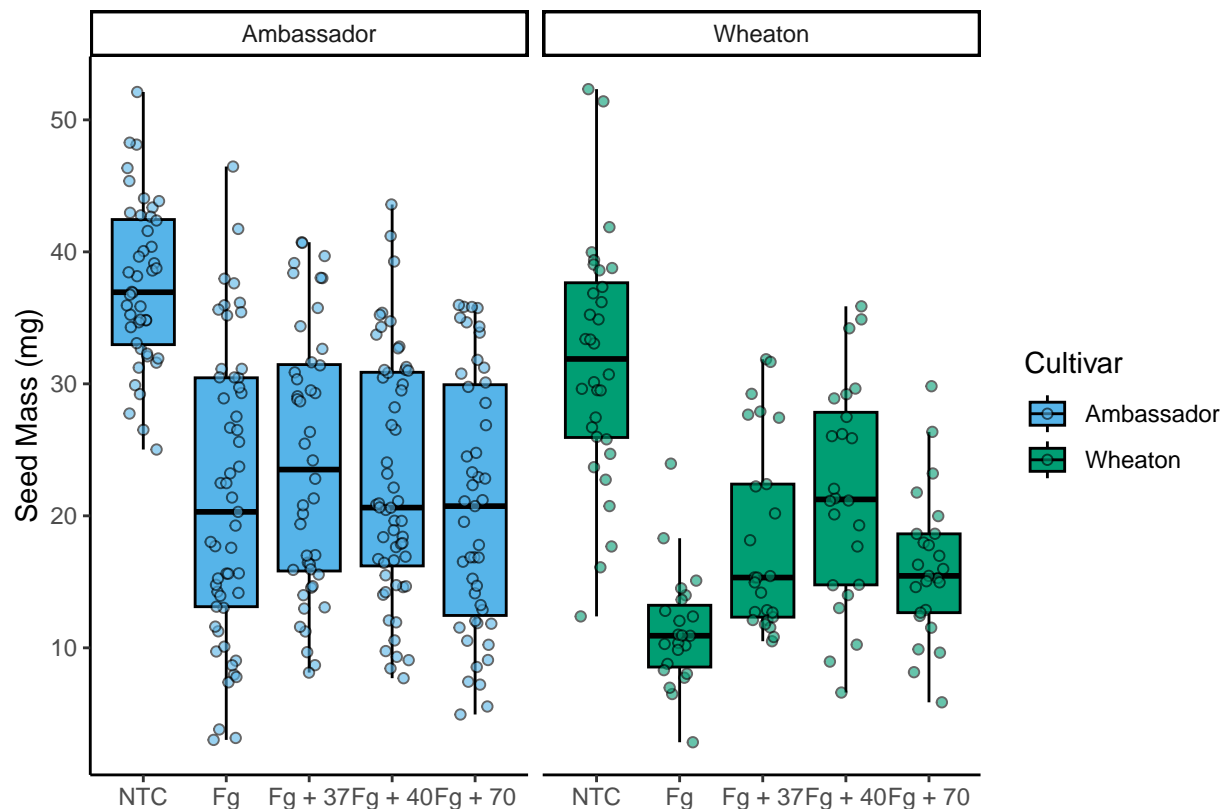


This code makes a SeedMass plot

```
Massplot <- ggplot(Myco, aes(x = Treatment, y = MassperSeed_mg, color = Cultivar, fill = Cultivar)) +
  geom_boxplot(position = position_dodge(), color = "black", outliers = FALSE) + # Add boxplots with dots
  xlab("") + # Label the x-axis
  ylab("Seed Mass (mg)") + # Label the y-axis
  geom_point(alpha = 0.6, color = "black", position = position_jitterdodge(dodge.width = 0.9), shape = 1) +
  theme_classic() + # Use a classic theme for the plot
  scale_color_manual(values = c(cbbPalette[[3]], cbbPalette[[4]]), name = "", labels = c("Ambassador", "Wheaton")) +
  scale_fill_manual(values = c(cbbPalette[[3]], cbbPalette[[4]])) + # Manually set colors
  facet_wrap(~Cultivar, ncol = 2, scales = "free_x") # Create separate panels for each Crop, allowing for different x-axis scales
Massplot # Save as object, run object
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## ('stat_boxplot()').
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## ('geom_point()').
```



This code combines all three plots into one figure

```
# Combine all three figures into one
figure1 <- ggarrange(
  DONplot, # First plot
  X15ADONplot, # Second plot
  Massplot, # Third plot
  labels = "auto", # Automatically label the plots (A, B, C, etc.)
  nrow = 1, # Arrange the plots in 1 rows
  ncol = 3, # Arrange the plots in 3 column
  common.legend = T # Include a legend in the combined figure
)

## Warning: Removed 8 rows containing non-finite outside the scale range
## ('stat_boxplot()').

## Warning: Removed 8 rows containing missing values or values outside the scale range
## ('geom_point()').

## Warning: Removed 8 rows containing non-finite outside the scale range
## ('stat_boxplot()').

## Warning: Removed 8 rows containing missing values or values outside the scale range
## ('geom_point()').
```

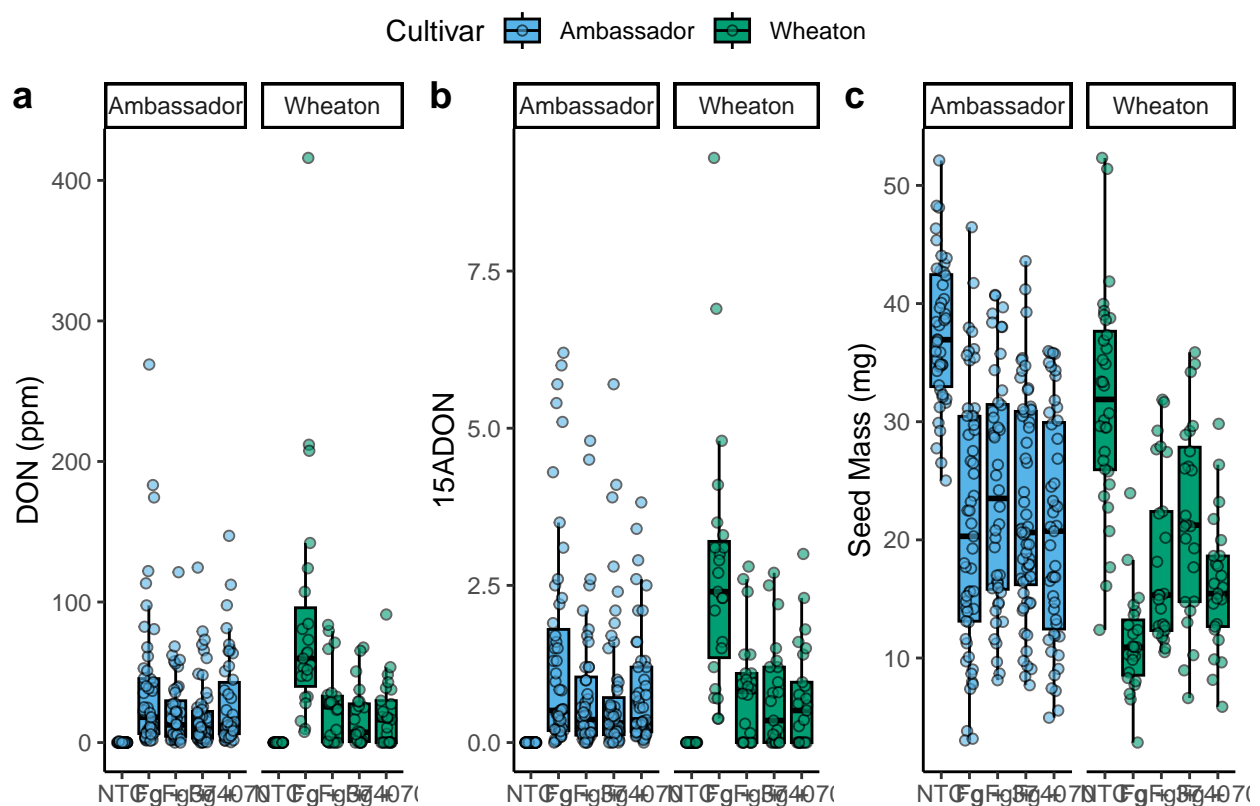
```
## Warning: Removed 10 rows containing non-finite outside the scale range
## ('stat_boxplot()').

## Warning: Removed 10 rows containing missing values or values outside the scale range
## ('geom_point()').

## Warning: Removed 2 rows containing non-finite outside the scale range
## ('stat_boxplot()').

## Warning: Removed 2 rows containing missing values or values outside the scale range
## ('geom_point()').
```

figure1



This code adds the t-test pairwise comparisons to the combined figure

```
# Add t test for pairwise comparisons
pDON <- DONplot +
  geom_pwc(aes(group = Treatment), method = "t_test", label = "{p.adj.format}{p.adj.signif}") # Combine

pX15 <- X15ADONplot +
  geom_pwc(aes(group = Treatment), method = "t_test", label = "{p.adj.format}{p.adj.signif}") # Combine

pMass <- Massplot +
  geom_pwc(aes(group = Treatment), method = "t_test", label = "{p.adj.format}{p.adj.signif}") # Combine
```

```

# Recombine all three plots
figure2 <- ggarrange(
  pDON, # First plot
  pX15, # Second plot
  pMass, # Third plot
  labels = "auto", # Automatically label the plots (A, B, C, etc.)
  nrow = 1, # Arrange the plots in 1 rows
  ncol = 3, # Arrange the plots in 3 column
  common.legend = T # Include a legend in the combined figure
)

## Warning: Removed 8 rows containing non-finite outside the scale range
## ('stat_boxplot()').

## Warning: Removed 8 rows containing non-finite outside the scale range
## ('stat_pwc()').

## Warning: Removed 8 rows containing missing values or values outside the scale range
## ('geom_point()').

## Warning: Removed 8 rows containing non-finite outside the scale range
## ('stat_boxplot()').

## Warning: Removed 8 rows containing non-finite outside the scale range
## ('stat_pwc()').

## Warning: Removed 8 rows containing missing values or values outside the scale range
## ('geom_point()').

## Warning: Removed 10 rows containing non-finite outside the scale range
## ('stat_boxplot()').

## Warning: Removed 10 rows containing non-finite outside the scale range
## ('stat_pwc()').

## Warning: Removed 10 rows containing missing values or values outside the scale range
## ('geom_point()').

## Warning: Removed 2 rows containing non-finite outside the scale range
## ('stat_boxplot()').

## Warning: Removed 2 rows containing non-finite outside the scale range
## ('stat_pwc()').

## Warning: Removed 2 rows containing missing values or values outside the scale range
## ('geom_point()').

```

figure2

