

# **Coding Practice 2**

## **Introduction to visualization in R**

**PLPA-5820 (Spring 2025)**

Gabriel A. A. Silva

2025-02-13

### **Instructions**

Introduction to visualization in R – 25 pts

PLEASE READ THIS BEFORE CONTINUING

This assignment is designed to help you visualize data in R and practice your R coding skills. You may collaborate with a partner to enhance your learning experience. Please ensure the following:

- Collaboration: If you worked with a partner, include their name.
- Submission: Only one person should submit the assignment.
- Answers: Provide written answers to all questions in the current document or a rendered markdown file.
- Code Submission: Submit your R code in an .R or .Rmd file by sharing a link to your GitHub repository.

Partner 1:

Partner 2:

### **Data description**

Here, we will analyze mycotoxin data collected from wheat heads grown in the greenhouse inoculated with *Fusarium graminearum* (Fg) with or without fungal endophytes to protect against Fg. The experiment was performed by inoculating the heads with Fg (positive control), Fg + 1 of 3 endophytes (Fg + 40, Fg + 70, or Fg + 37), and the non-treated control (NTC). Therefore, the experiment was five treatments with 6-10 technical replicates and three

experimental replicates. The authors quantified the parts per million (ppm) of mycotoxins Deoxynivalenol (DON) and the variant 15ADON from the wheat grain, and this is your primary response variable. In other words, we are interested in knowing if the fungal endophytes could reduce the concentration of DON on wheat heads. This was done across two wheat cultivars, a winter wheat variety, Ambassador, and a spring wheat variety, Wheaton. These data are published here:

- Noel, Z.A., Roze, L.V., Breunig, M., Trail, F. 2022. Endophytic fungi as promising bio-control agent to protect wheat from *Fusarium graminearum* head blight. Plant Disease. <https://doi.org/10.1094/PDIS-06-21-1253-RE>
- The following github repository is associated with this paper: <https://github.com/noelzach/EndophyteBiocontrol>

## Questions

### 5 pts. Explain the following concepts about ggplot

- What three elements do you need to produce a ggplot?

Data, aesthetics, and geoms.

- What is a geom?

It's a geometric object that represents the data, such as points, lines, bars, etc.

- What is a facet?

It's a way to split the data and display them in different panels.

- Explain the concept of layering.

I would say it's the process of adding different elements to the plot to make the final visualization. For example, first you add bars, then you add some statistic like standard deviation on top of it.

- Where do you add x and y variables and map different shapes, colors, and other attributes to the data?

X and y variables are used in the `aes()` function. To map those different attributes, the `geom` functions are used.

4 pts. Make a boxplot using ggplot with DON as the y variable, treatment as the x variable, and color mapped to the wheat cultivar. Show the code you use to load the libraries you need to read in the data and make the plot. Change the y label to "DON (ppm)" and make the x label blank.

```
# Load libraries
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.0.4

-- 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
```

```
# Read in the data
## Treatment, Cultivar, MassperSeed_mg, DON, 15ADON, BioRep
data <- read_csv("MycotoxinData.csv") %>%
  mutate(DON = as.numeric(DON),
         MassperSeed_mg = as.numeric(MassperSeed_mg),
         `15ADON` = as.numeric(`15ADON`),
         BioRep = as.factor(BioRep)) %>%
  filter(!is.na(DON))
```

```
Rows: 375 Columns: 6
```

```
-- Column specification -----
Delimiter: ","
chr (5): Treatment, Cultivar, MassperSeed_mg, DON, 15ADON
dbl (1): BioRep
```

```
i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

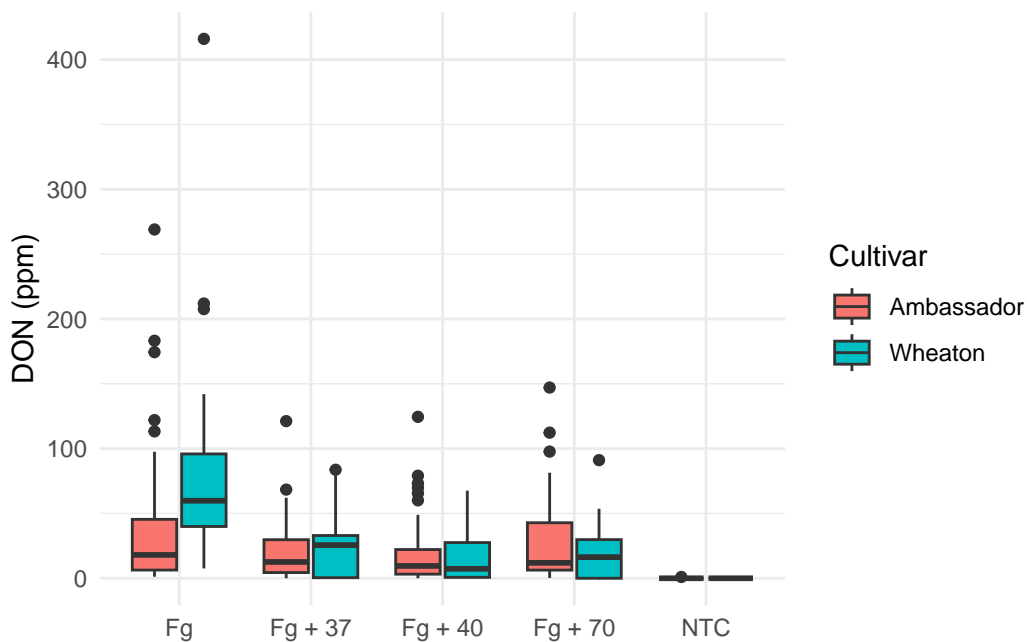
```
Warning: There were 3 warnings in `mutate()`.
The first warning was:
i In argument: `DON = as.numeric(DON)`.
```

Caused by warning:

! NAs introduced by coercion

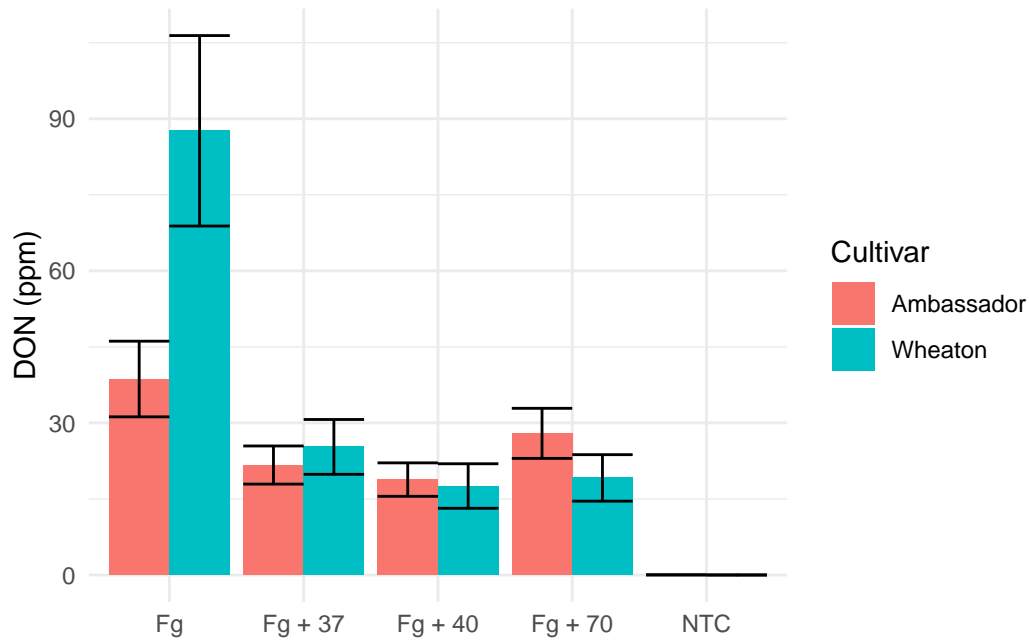
i Run ``dplyr::last_dplyr_warnings()`` to see the 2 remaining warnings.

```
# Make the boxplot
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  geom_boxplot() +
  labs(y = "DON (ppm)", x = element_blank()) +
  theme_minimal()
```



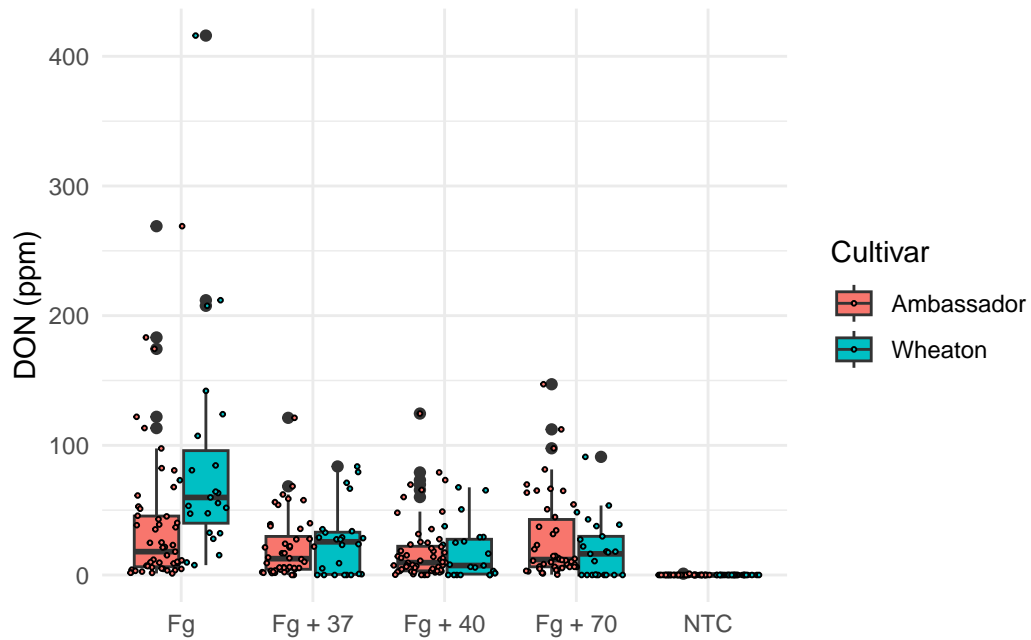
2 pts. Now convert this data into a bar chart with standard-error error bars using the `stat_summary()` command.

```
# Bar chart with error bars
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  stat_summary(fun=mean, geom="bar", position = "dodge") +
  stat_summary(fun.data = mean_se, geom = "errorbar", position = "dodge") +
  labs(y = "DON (ppm)", x = element_blank()) +
  theme_minimal()
```

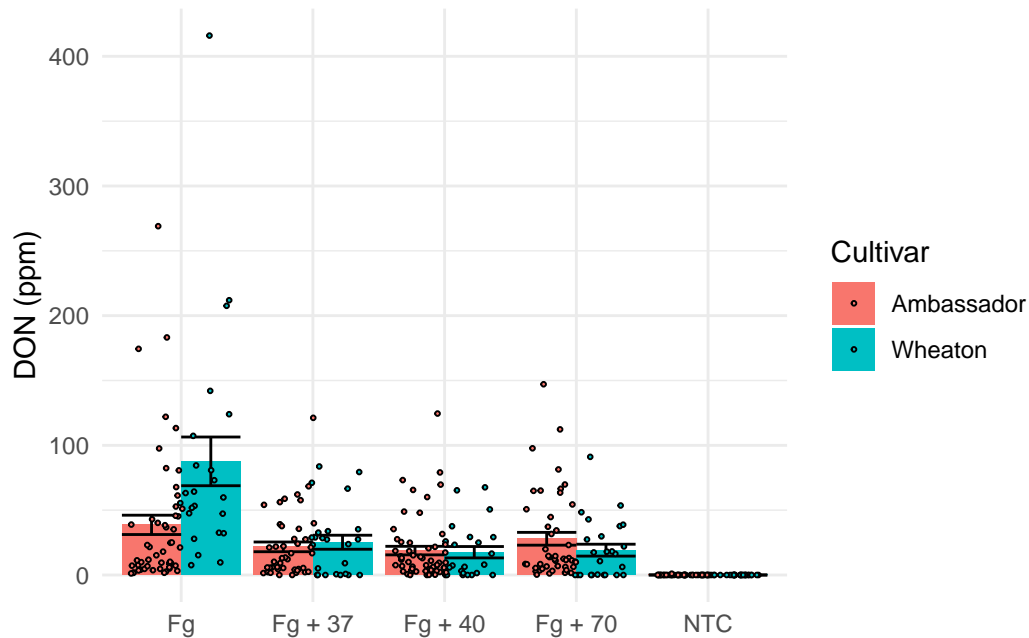


2 pts. Add points to the foreground of the boxplot and bar chart you made in question 3 that show the distribution of points over the boxplots. Set the shape = 21 and the outline color black (hint: use jitter\_dodge).

```
# Boxplot with jitter
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  geom_boxplot() +
  geom_point(position = position_jitterdodge(), shape = 21, color = "black", size = 0.1) +
  labs(y = "DON (ppm)", x = element_blank()) +
  theme_minimal()
```

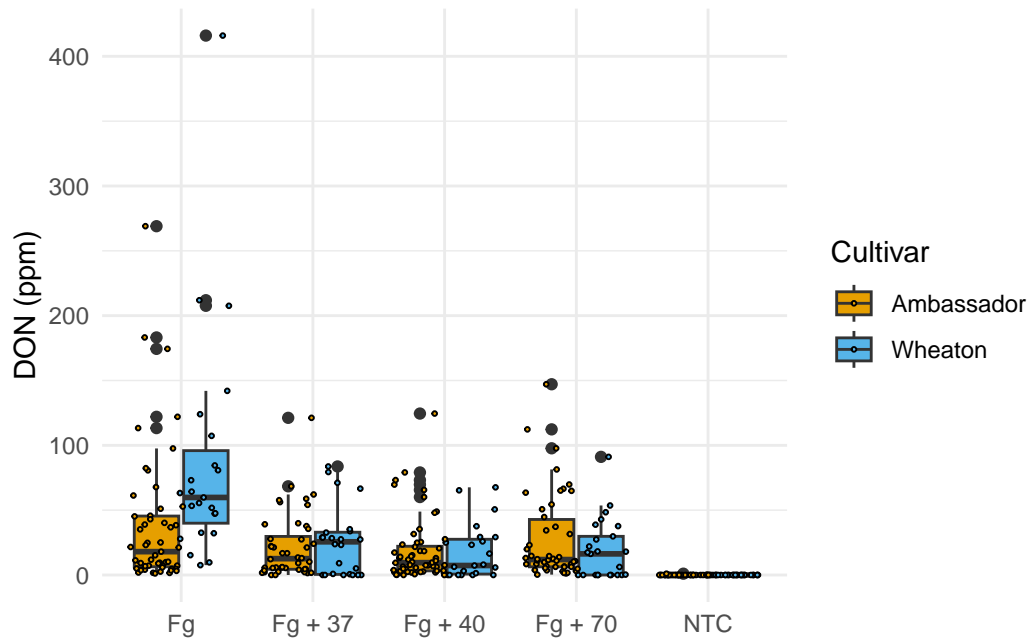


```
# Bar chart with jitter
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  stat_summary(fun=mean,geom="bar", position = "dodge") +
  stat_summary(fun.data = mean_se, geom = "errorbar", position = "dodge") +
  geom_point(position = position_jitterdodge(), shape = 21, color = "black", size = 0.5) +
  labs(y = "DON (ppm)", x = element_blank()) +
  theme_minimal()
```



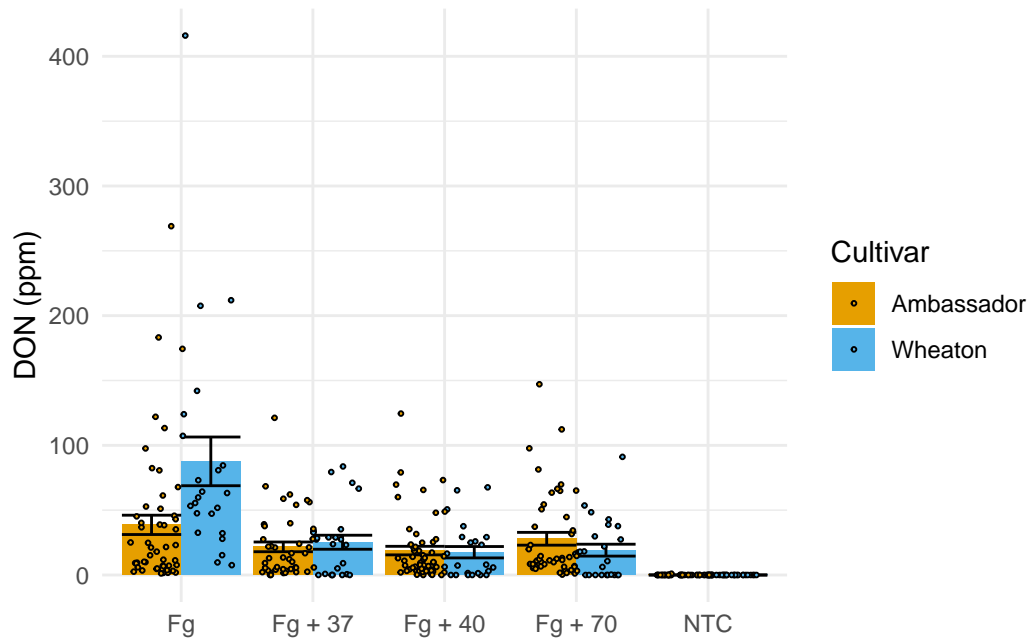
2 pts. Change the fill color of the points and boxplots to match some colors in the following colorblind pallet.

```
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00",
# Boxplot
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  geom_boxplot() +
  geom_point(position = position_jitterdodge(), shape = 21, color = "black", size = 0.5) +
  labs(y = "DON (ppm)", x = element_blank()) +
  scale_fill_manual(values = cbbPalette[2:3], name = "Cultivar", labels = c("Ambassador", "Wheaton")) +
  theme_minimal()
```



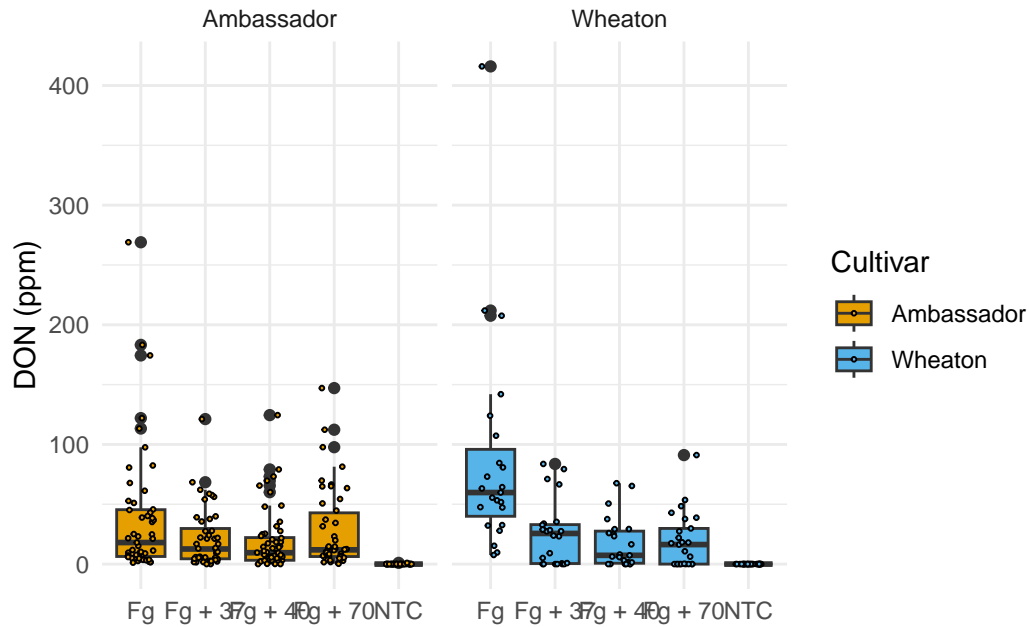
```
# Bar chart
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  stat_summary(fun=mean,geom="bar", position = "dodge") +
  stat_summary(fun.data = mean_se, geom = "errorbar", position = "dodge") +
  geom_point(position = position_jitterdodge(), shape = 21, color = "black", size = 0.5) +
  labs(y = "DON (ppm)", x = element_blank()) +
  scale_fill_manual(values = cbbPalette[2:3], name = "Cultivar", labels = c("Ambassador", "Wheaton")) +
  theme_minimal()
```





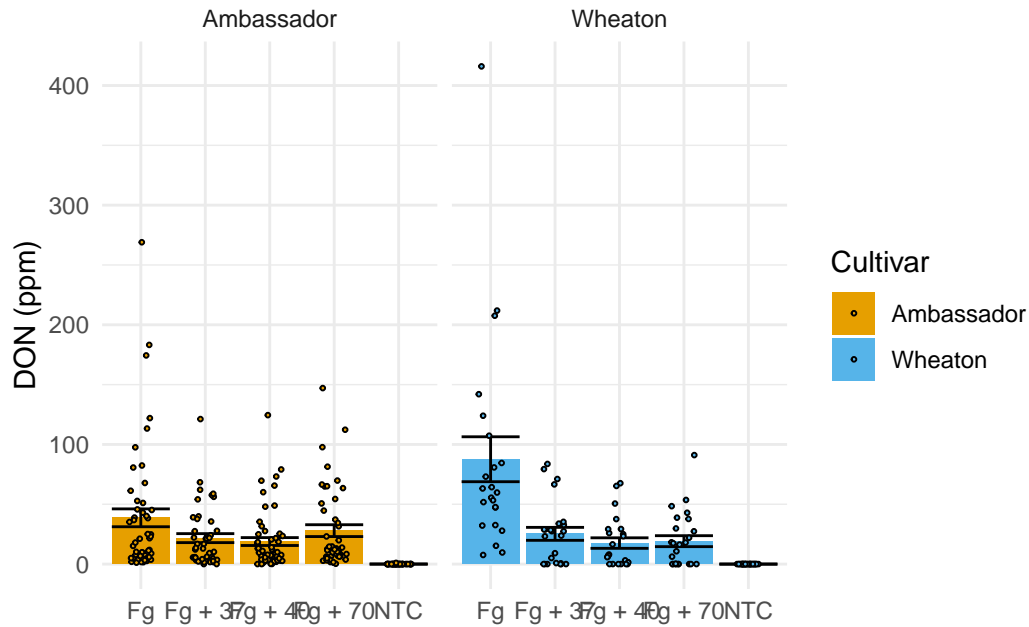
2 pts. Add a facet to the plots based on cultivar.

```
# Boxplot
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  geom_boxplot() +
  geom_point(position = position_jitterdodge(), shape = 21, color = "black", size = 0.5) +
  labs(y = "DON (ppm)", x = element_blank()) +
  scale_fill_manual(values = cbbPalette[2:3], name = "Cultivar", labels = c("Ambassador", "Wheaton")) +
  theme_minimal() +
  facet_wrap(~ Cultivar)
```



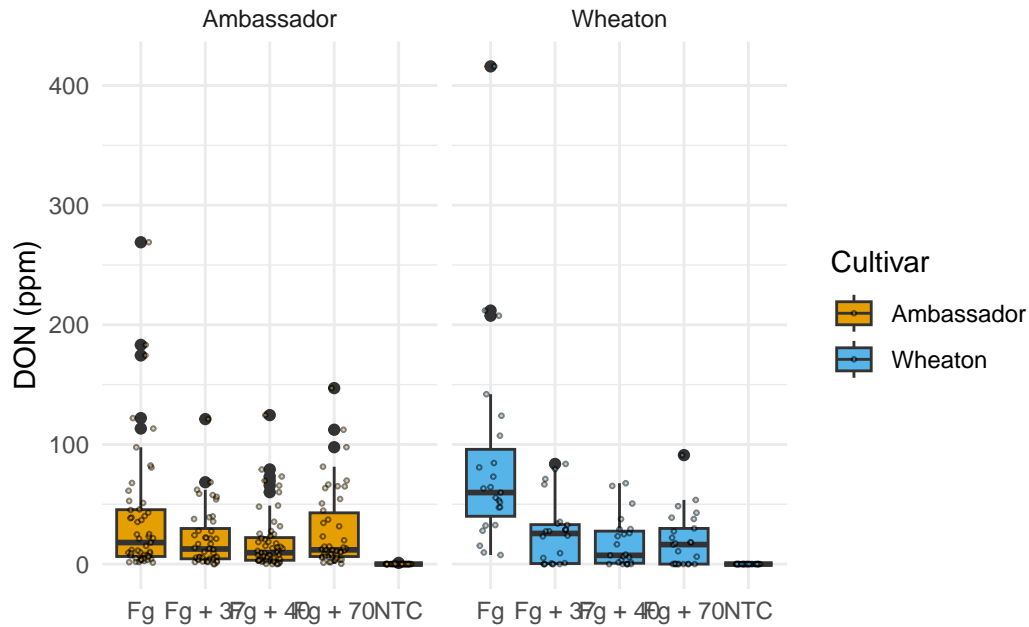
# Bar chart

```
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  stat_summary(fun=mean, geom="bar", position = "dodge") +
  stat_summary(fun.data = mean_se, geom = "errorbar", position = "dodge") +
  geom_point(position = position_jitterdodge(), shape = 21, color = "black", size = 0.5) +
  labs(y = "DON (ppm)", x = element_blank()) +
  scale_fill_manual(values = cbbPalette[2:3], name = "Cultivar", labels = c("Ambassador", "Wheaton")) +
  theme_minimal() +
  facet_wrap(~ Cultivar)
```



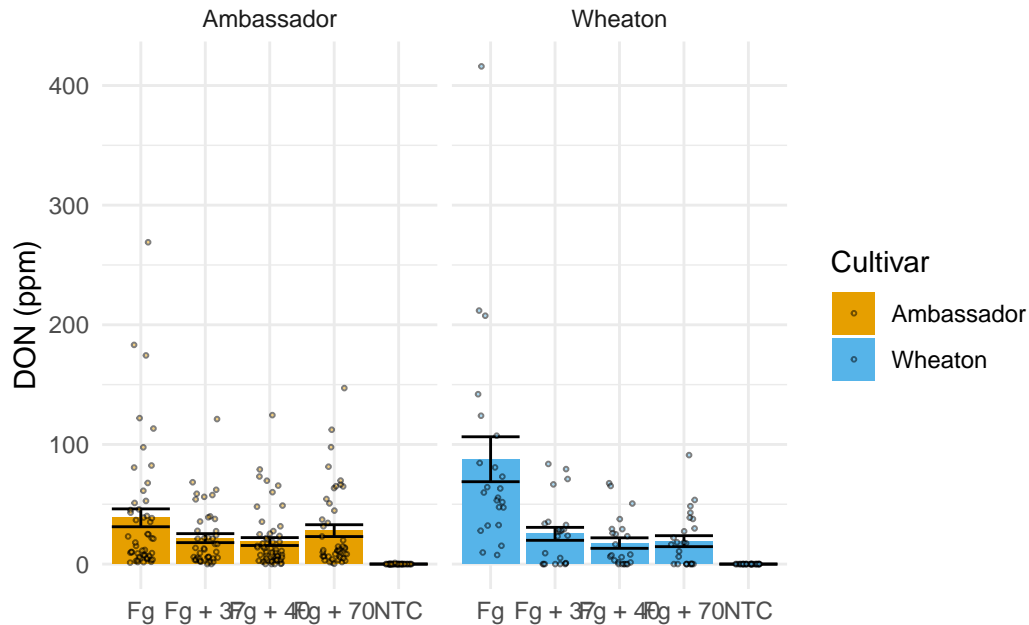
2 pts. Add transparency to the points so you can still see the boxplot or bar in the background.

```
# Boxplot
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  geom_boxplot() +
  geom_point(position = position_jitterdodge(), shape = 21, color = "black", size = 0.5) +
  labs(y = "DON (ppm)", x = element_blank()) +
  scale_fill_manual(values = cbbPalette[2:3], name = "Cultivar", labels = c("Ambassador", "Wheaton")) +
  theme_minimal() +
  facet_wrap(~ Cultivar)
```



# Bar chart

```
ggplot(data, aes(x = Treatment, y = DON, fill = Cultivar)) +
  stat_summary(fun=mean, geom="bar", position = "dodge") +
  stat_summary(fun.data = mean_se, geom = "errorbar", position = "dodge") +
  geom_point(position = position_jitterdodge(), shape = 21, color = "black", size = 0.5) +
  labs(y = "DON (ppm)", x = element_blank()) +
  scale_fill_manual(values = cbbPalette[2:3], name = "Cultivar", labels = c("Ambassador", "Wheaton")) +
  theme_minimal() +
  facet_wrap(~ Cultivar)
```

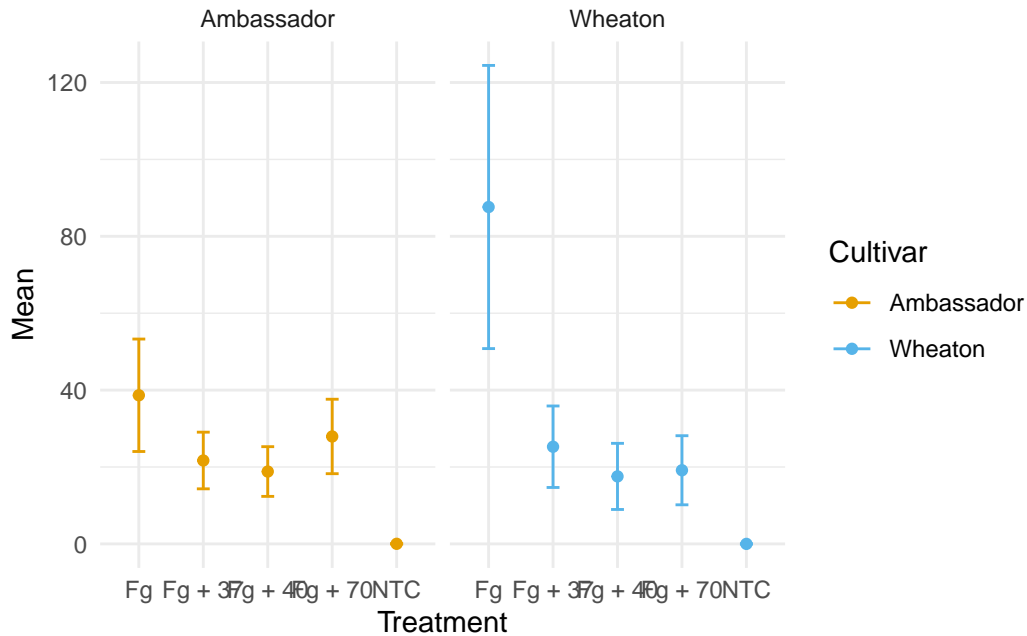


2 pts. Explore one other way to represent the same data  
<https://ggplot2.tidyverse.org/reference/> . Plot them and show the code here. Which one would you choose to represent your data and why?

```
data_CI <- data %>%
  group_by(Cultivar, Treatment) %>%
  summarize(Mean = mean(DON), SD = sd(DON), N = n(),
            CI_L = Mean - (SD * 1.96)/sqrt(N),
            CI_U = Mean + (SD * 1.96)/sqrt(N))
```

`summarise()` has grouped output by 'Cultivar'. You can override using the  
 `.groups` argument.

```
ggplot(data_CI, aes(x = Treatment, y = Mean, color = Cultivar)) +
  geom_point() +
  geom_errorbar(aes(ymin = CI_L, ymax = CI_U), width = 0.2) +
  scale_color_manual(values = cbbPalette[2:3], name = "Cultivar", labels = c("Ambassador", "Wheaton")) +
  theme_minimal() +
  facet_wrap(~ Cultivar)
```



I tried to do a violin plot, but for some reason it wouldn't shape correctly. I chose to do a mean point with 95% confidence intervals instead. I think since the data has some outliers, showing a confidence interval reduces the plotting area and makes the data more readable.

**4 pts. Annotate your code and push it to github. Now, find a partner if you don't already have one and have that person "fork" the repository containing the code you just pushed to github. Describe what just happened when you "forked" the repository. Please include the links (URLs) below to answer this question.**

Forking a repository creates a copy of the original repository in your own account, which can then be edited without modifying the original repository.