

# Codingchallenge4

Priya Shahi

2026-02-20

## Contents

<b>1 Question1</b>	<b>1</b>
<b>2 Question2a</b>	<b>1</b>
<b>3 Question2b</b>	<b>1</b>
<b>4 Question2c</b>	<b>2</b>
4.1 DON_PLOT1 . . . . .	2
4.2 Plot X15ADON . . . . .	3
4.3 SeedMass . . . . .	4
<b>5 Question5</b>	<b>5</b>
<b>6 Question6</b>	<b>5</b>

## 1 Question1

*In the context of R Markdown , the YAML header appears at the top of the document, enclosed by — lines, and it contains metadata about the document. Literate Programming(For eg. R Markdown) is documentation and code live together in the same file, making it easy for others (and your future self!) to understand not just what the code does, but why it was written that way.*

## 2 Question2a

-PaperLink

## 3 Question2b

```
library(tidyverse)
```

```

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.2.0     v readr     2.1.6
## vforcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.2     v tibble    3.3.1
## v lubridate 1.9.4     v tidyv     1.3.2
## v purrr    1.2.1
## -- 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)
mycotoxin = read.csv("MycotoxinData.csv", na.strings = "na")
mycotoxin$Treatment <- as.factor(mycotoxin$Treatment)
mycotoxin$Cultivar <- as.factor(mycotoxin$Cultivar)
str(mycotoxin)

## 'data.frame': 375 obs. of 6 variables:
##   $ Treatment : Factor w/ 5 levels "Fg","Fg + 37",...: 1 1 1 1 1 1 1 1 1 ...
##   $ Cultivar  : Factor w/ 2 levels "Ambassador","Wheaton": 2 2 2 2 2 2 2 2 2 ...
##   $ 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 ...

```

## 4 Question2c

```
cbbPalette <- c("#56B4E9", "#009E73")
```

### 4.1 DON\_PLOT1

```

Plot1 <- ggplot(mycotoxin, aes(x = Treatment, y = DON, fill = Cultivar)) +
  geom_boxplot(position = position_dodge(0.85)) +
  xlab("") +
  ylab("DON (ppm)") +
  geom_point(alpha = 0.6, pch = 21, color = "black") +
  scale_fill_manual(values = cbbPalette) +
  facet_wrap(~Cultivar) +
  theme_classic()
Plot1

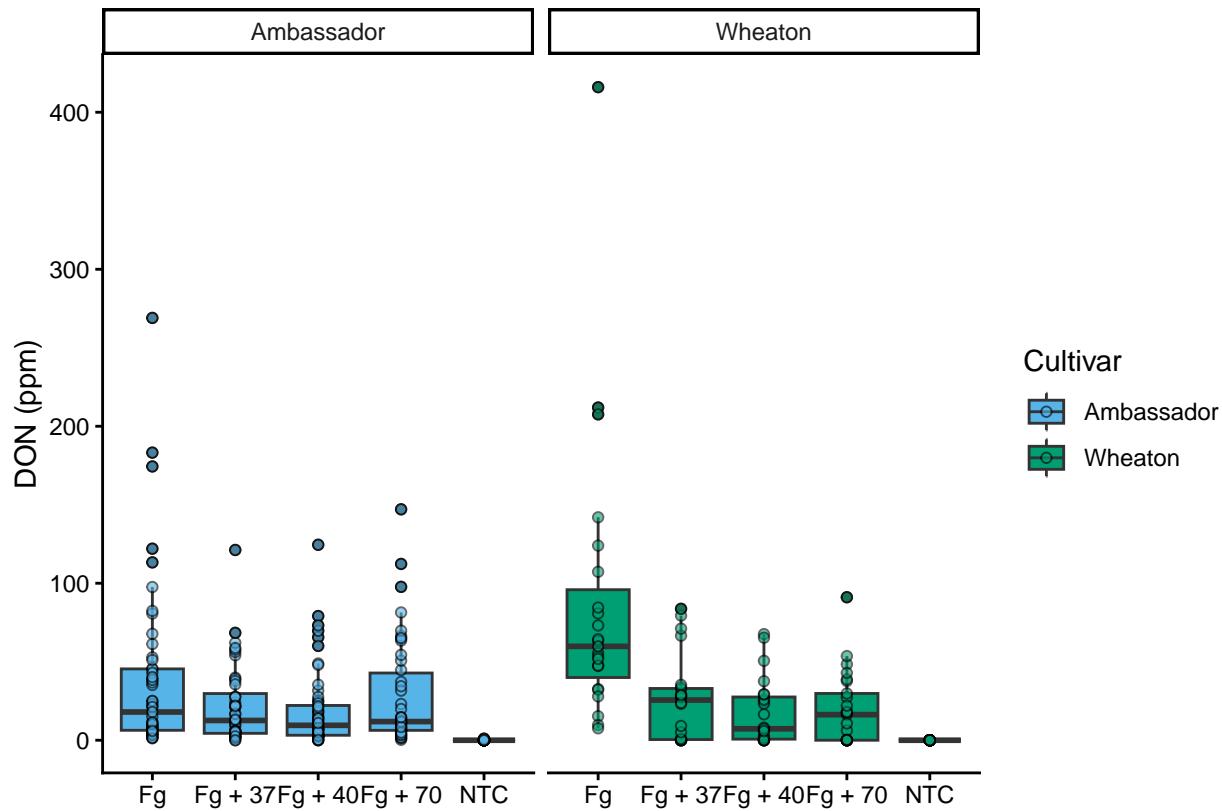
```

```

## 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()').

```



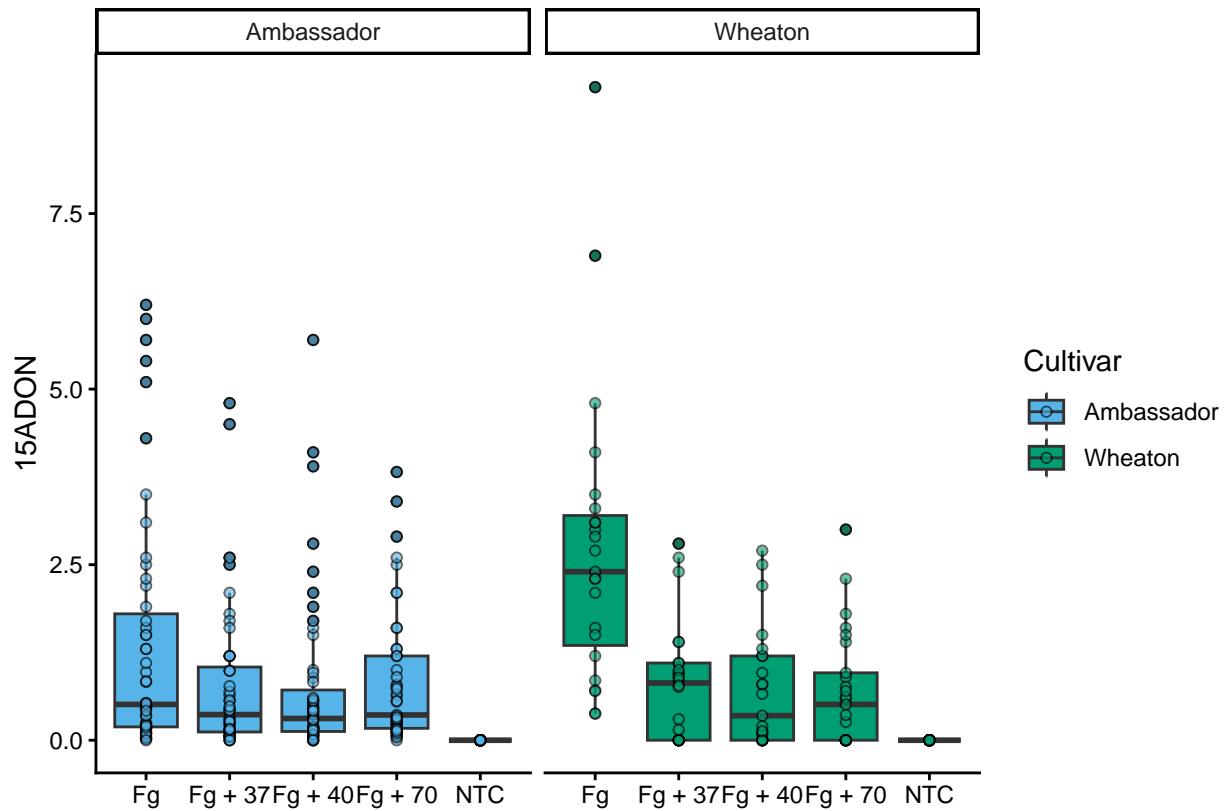
## 4.2 Plot X15ADON

```
Plot2 <- ggplot(mycotoxin, aes(x = Treatment, y = X15ADON, fill = Cultivar)) +
  geom_boxplot(position = position_dodge(0.85)) +
  xlab("") +
  ylab("X15ADON") +
  geom_point(alpha = 0.6, pch = 21, color = "black") +
  scale_fill_manual(values = cbbPalette) +
  facet_wrap(~Cultivar) +
  theme_classic()
```

Plot2

```
## 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()').
```



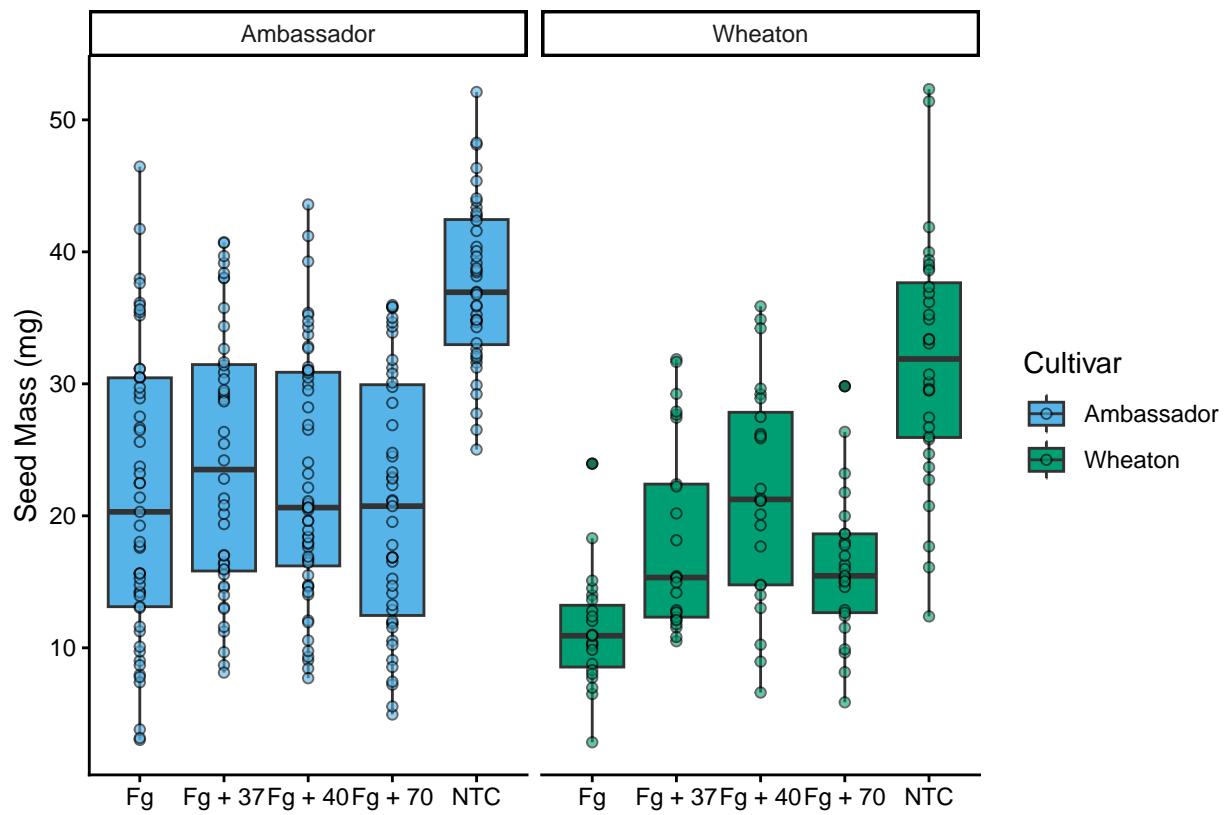
### 4.3 SeedMass

```
#Plot 3

Plot3<- ggplot(mycotoxin, aes(x = Treatment, y = MassperSeed_mg, fill = Cultivar)) +
  geom_boxplot(position = position_dodge(0.85)) +
  xlab("") +
  ylab("Seed Mass (mg)") +
  geom_point(alpha = 0.6,pch = 21, color = "black") +
  scale_fill_manual(values = cbbPalette) +
  facet_wrap(~Cultivar) +
  theme_classic()
Plot3

## 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()').
```



## 5 Question5

README

## 6 Question6

Github