Assignment 3: Data Exploration

Darpan Barua

Spring 2025

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Exploration.

Directions

1. Rename this file `<DarpanBarua>_A03_DataExploration.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Assign a useful **name to each code chunk** and include ample **comments** with your code.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.
7. After Knitting, submit the completed exercise (PDF file) to the dropbox in Canvas.

TIP: If your code extends past the page when knit, tidy your code by manually inserting line breaks.

TIP: If your code fails to knit, check that no `install.packages()` or `View()` commands exist in your code.

Set up your R session

1. Load necessary packages (tidyverse, lubridate, here), check your current working directory and upload two datasets: the ECOTOX neonicotinoid dataset (ECOTOX_Neonicotinoids_Insects_raw.csv) and the Niwot Ridge NEON dataset for litter and woody debris (NEON_NIWO_Litter_massdata_2018-08_raw.csv). Name these datasets "Neonics" and "Litter", respectively. Be sure to include the sub-command to read strings in as factors.

```
library(tidyverse)
library(lubridate)
library(here)

# Below we read the csv and read strings in as factors. We then set a variable
# as directed in the question.
Neonics <- read_csv(here("Assignments", "ECOTOX_Neonicotinoids_Insects_raw.csv"),
    col_types = cols(.default = col_factor()))


Litter <- read_csv(here("Assignments", "NEON_NIWO_Litter_massdata_2018-08_raw.csv"),
    col_types = cols(.default = col_factor()))
```

Learn about your system

2. The neonicotinoid dataset was collected from the Environmental Protection Agency's ECOTOX Knowledgebase, a database for ecotoxicology research. Neonicotinoids are a class of insecticides used widely in agriculture. The dataset that has been pulled includes all studies published on insects. Why might we be interested in the ecotoxicology of neonicotinoids on insects? Feel free to do a brief internet search if you feel you need more background information.

   Answer: With reference to what I learned online neonicotinoids can be used to protect crops from sap-sucking and leaf-chewing insects which is essential on farms. However, they can also harm bees, butterflies, and other flower-visitng insects - which may adversely affect the farmers. If we can better understand the impact of neonicotinoid, we can assess patterns of toxicity across species, concentrations, and expsoure conditions. This could help us determine how much or when or where nenicotinoid should be administered.

3. The Niwot Ridge litter and woody debris dataset was collected from the National Ecological Observatory Network, which collectively includes 81 aquatic and terrestrial sites across 20 ecoclimatic domains. 32 of these sites sample forest litter and woody debris, and we will focus on the Niwot Ridge long-term ecological research (LTER) station in Colorado. Why might we be interested in studying litter and woody debris that falls to the ground in forests? Feel free to do a brief internet search if you feel you need more background information.

   Answer: With reference to what I learned online, litter and woody debris decompose over time, releasing essential nutrients such as nitrogen, phosphorus, and carbon back into the soil. This can be essential as it helps plant growth and enriches the forest ecosystem. Forest litter and woody debris also store carbon, acting as a carbon sink. On the negative end - they can act as fuel for wildfires. Overall the data collected from LTER station could help us understand how the ecosystems change over time, limit wild-fire damages, and track forest health better.

4. How is litter and woody debris sampled as part of the NEON network? Read the NEON_Litterfall_UserGuide.pdf document to learn more. List three pieces of salient information about the sampling methods here:

   Answer: 1.Using elevated and ground traps for collection. Helps ensure consistent, representative sampling across different forest types. 2.Using a standardized spatial sampling design that are either randomized/targeted depending on vegetation density. 3.Varying Temporal Sampling Frequency by Ecosystem Type. Ground traps sampled once annually. Elevated traps (Deciduous forests - every 2 weeks; Evergreen forests - every 1/2 months; Extreme weather sites - up to 6 month intervals).

Obtain basic summaries of your data (Neonics)

5. What are the dimensions of the dataset?

```r
# Below notes the number of rows in Neonics.
nrow(Neonics)
```

```
## [1] 4623
```

```r
# Below notes the number of columns in Neonics.
ncol(Neonics)
```

```
## [1] 30
```

6. Using the **summary** function on the "Effect" column, determine the most common effects that are studied. Why might these effects specifically be of interest? [Tip: The `sort()` command is useful for listing the values in order of magnitude...]

```r
# Below simply employs the 'summary' function on the 'Effect' column within
# Neonics.
summary(Neonics$Effect)
```

```
##        Mortality           Growth       Population    Immunological
##             1493               38             1803               16
##          Cell(s)         Behavior     Reproduction      Development
##                9              360              197              136
##         Genetics        Enzyme(s) Feeding behavior        Avoidance
##               82               62              255              102
##     Intoxication     Biochemistry       Hormone(s)     Accumulation
##               12               11                1               12
##       Morphology        Histology       Physiology
##               22                5                7
```

```r
# Below we assign a variable 'effect_summary' to the summary function on the
# 'Effect' column within Neonics.
effect_summary <- summary(Neonics$Effect)

sort(effect_summary, decreasing = TRUE)
```

```
##       Population        Mortality         Behavior Feeding behavior
##             1803             1493              360              255
##     Reproduction      Development        Avoidance         Genetics
##              197              136              102               82
##        Enzyme(s)           Growth       Morphology    Immunological
##               62               38               22               16
##     Intoxication     Accumulation     Biochemistry          Cell(s)
##               12               12               11                9
##       Physiology        Histology       Hormone(s)
##                7                5                1
```

Answer: Seems like Population and Mortality are the most common effects that are studied. This helps us know which effects are studied the most and least. It can help prioritize ecological concerns, craft informed pesticide regulations, and develop safer alternatives.

7. Using the `summary` function, determine the six most commonly studied species in the dataset (common name). What do these species have in common, and why might they be of interest over other insects? Feel free to do a brief internet search for more information if needed.[TIP: Explore the help on the `summary()` function, in particular the `maxsum` argument...]

```r
# Below we summarize the common names of species
species_summary <- summary(Neonics$"Species Common Name", maxsum = 6)
# Below we sort the species in descending order
sort(species_summary, decreasing = TRUE)
```

```
##                (Other)               Honey Bee         Parasitic Wasp 
##                   3196                     667                    285 
## Buff Tailed Bumblebee   Carniolan Honey Bee             Bumble Bee 
##                    183                     152                    140 
```

> Answer:They are all pollinators (mostly bees and some paristic wasps). Since these insects forage on flowers, they're directly exposed to pesticide-treated crops. This affects these pollinators and could lead to adverse effects in food or other product production and ecosystem balance. Therefore we see that they're of economic and ecological importance.

8. Concentrations are always a numeric value. What is the class of `Conc.1..Author.` column in the dataset, and why is it not numeric? [Tip: Viewing the dataframe may be helpful…]

```r
# Below we check the class of the Conc.1..Author. column
class(Neonics$"Conc 1 (Author)")
```

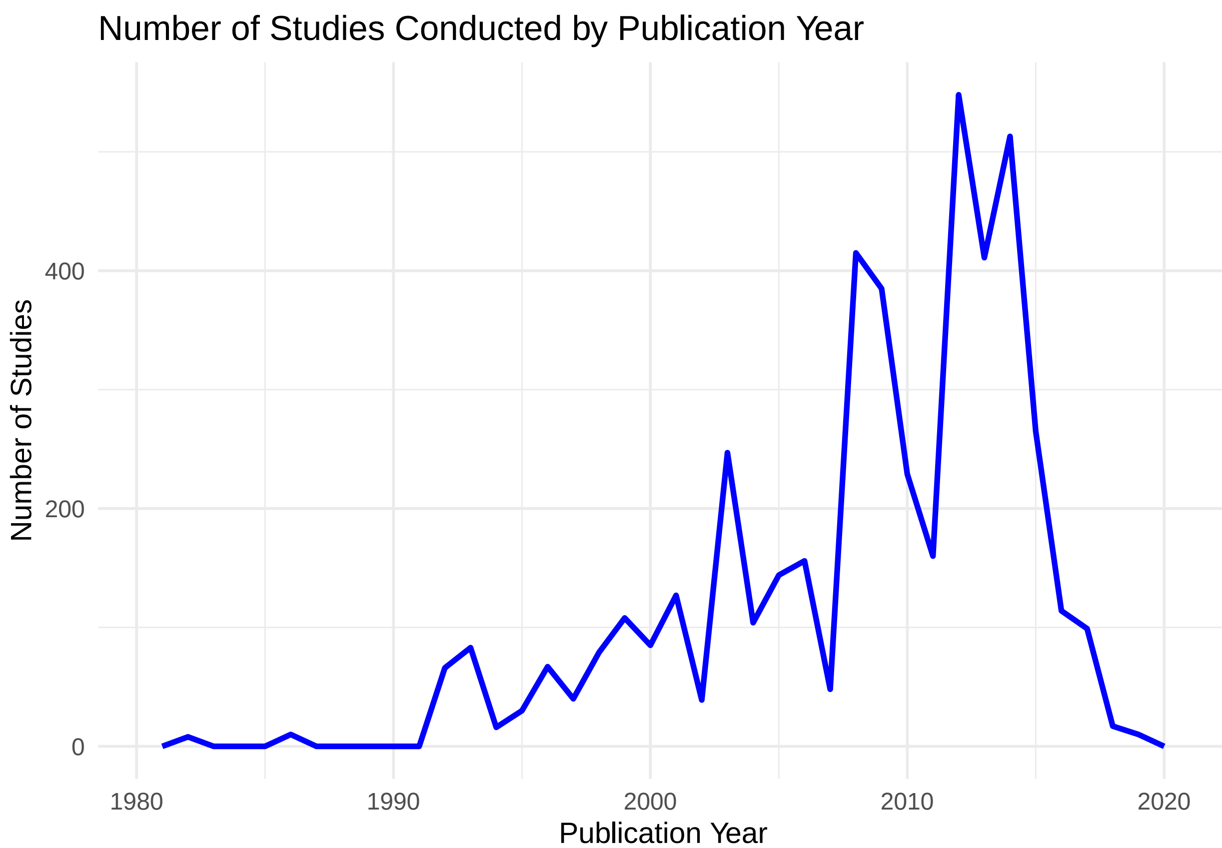
```
## [1] "factor"
```

> Answer:Maybe because it contains texts or other non-number characters.

Explore your data graphically (Neonics)

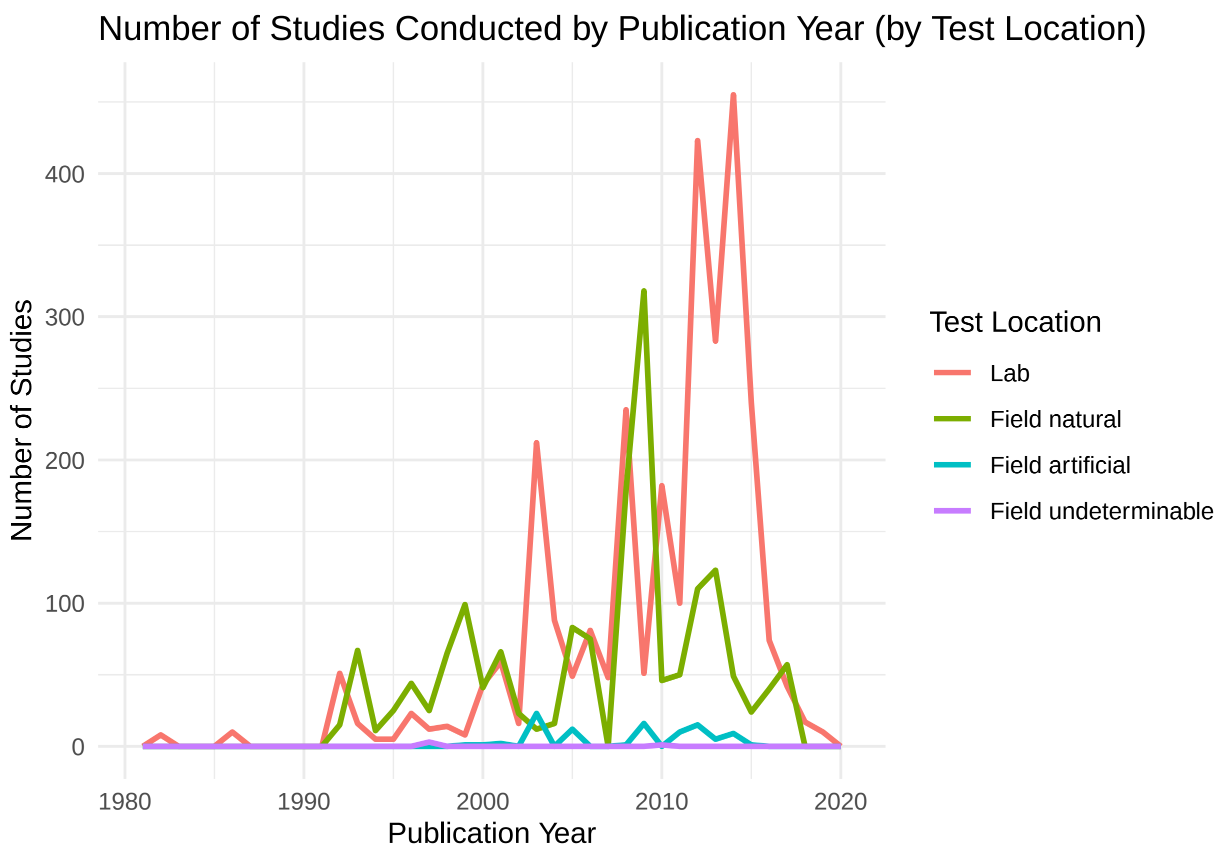
9. Using `geom_freqpoly`, generate a plot of the number of studies conducted by publication year.

```r
# Below we convert the publication year column to numeric; when checking class
# before it showed factor instead of numeric.

Neonics$Publication_Year <- as.numeric(as.character(Neonics$"Publication Year"))

# library(ggplot2)

# Below we create frequency polygon of study counts by publication year. We
# also use 'theme_minimal' to have a clean theme for better visualization.
ggplot(Neonics, aes(x = Publication_Year)) + geom_freqpoly(binwidth = 1, color = "blue",
    linewidth = 1) + labs(title = "Number of Studies Conducted by Publication Year",
    x = "Publication Year", y = "Number of Studies") + theme_minimal()
```



10. Reproduce the same graph but now add a color aesthetic so that different Test.Location are displayed as different colors.

```
# Below we ensure the Publication Year is numeric (as done in previous
# question)
Neonics$Publication_Year <- as.numeric(as.character(Neonics$Publication_Year))

# Below we ensure Test.Location is a factor
Neonics$Test_Location <- as.factor(Neonics$"Test Location")


# Below we regenerate the plot. This time the color is based on the Test
# Location which is a factor column.

ggplot(Neonics, aes(x = Publication_Year, color = Test_Location)) + geom_freqpoly(binwidth = 1,
    linewidth = 1) + labs(title = "Number of Studies Conducted by Publication Year (by Test Location)",
    x = "Publication Year", y = "Number of Studies", color = "Test Location") + theme_minimal()
```



Interpret this graph. What are the most common test locations, and do they differ over time?

> Answer:The most common test locations are lab, followed by field natural and field artificial. Until around 2000, field natural was more common, before it was overtaken by labs. Then around before 2010, field natural took over once again as the higher number of studies before seeing a steady decline. More tests were later conducted in labs and it seems like cumulatively, there may have been more tests conducted in the lab versus field natural, especially in decade preceding 2020.

11. Create a bar graph of Endpoint counts. What are the two most common end points, and how are they defined? Consult the ECOTOX_CodeAppendix for more information.

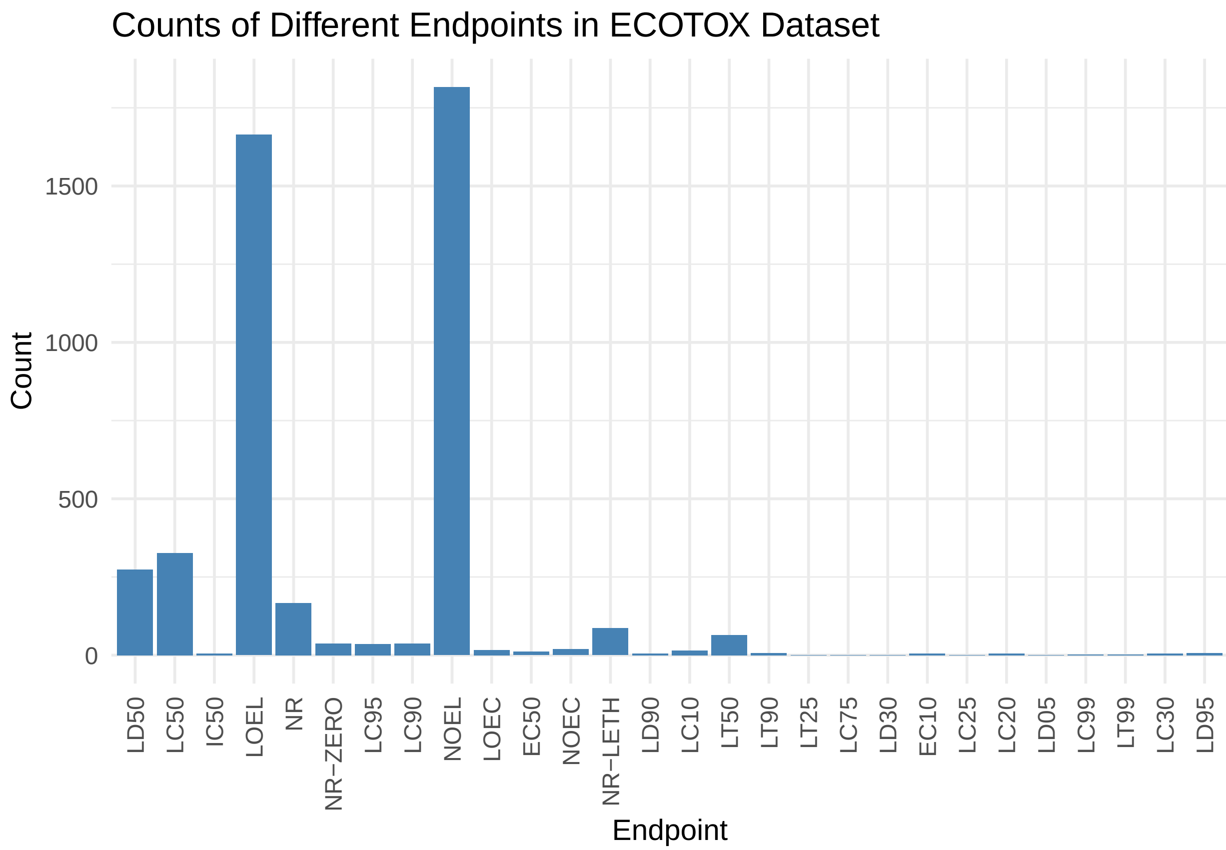
[**TIP**: Add `theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))` to the end of your plot command to rotate and align the X-axis labels…]

```
# Below checks if Neonics$Endpoint is factor or not.
class(Neonics$Endpoint)
```

```
## [1] "factor"
```

```
# Below counts the number of occurences of each endpoint.
endpoint_counts <- as.data.frame(table(Neonics$Endpoint))

# Below creates a bar plot.
ggplot(endpoint_counts, aes(x = Var1, y = Freq)) + geom_bar(stat = "identity", fill = "steelblue") +
```

```r
  labs(title = "Counts of Different Endpoints in ECOTOX Dataset", x = "Endpoint",
      y = "Count") + theme_minimal() + theme(axis.text.x = element_text(angle = 90,
  vjust = 0.5, hjust = 1))  # Here we rotate the x labels for better visualization
```



Answer:LOEL and NOEL are the two most common endpoint. LOEL is defined as 'Lowest-observable-effect-level: lowest dose (concentration) producing effects that were significantly different (as reported by authors) from responses of controls (LOEAL/LOEC). NOEL is defined as 'No-observable-effect-level: highest dose (concentration) producing effects not significantly different from responses of controls according to author's reported statistical rest (NOEAL/NOEC).

Explore your data (Litter)

12. Determine the class of collectDate. Is it a date? If not, change to a date and confirm the new class of the variable. Using the `unique` function, determine which dates litter was sampled in August 2018.

```r
library(lubridate)

class(Litter$collectDate)
```

```
## [1] "factor"
```

```r
# Below extracts unique dates for August 2018
unique_dates <- unique(Litter$collectDate[month(Litter$collectDate) == 8 & year(Litter$collectDate) ==
    2018])
```

```
## Warning: tz(): Don't know how to compute timezone for object of class factor;
## returning "UTC".
## Warning: tz(): Don't know how to compute timezone for object of class factor;
## returning "UTC".
```

```r
unique_dates
```

```
## [1] 2018-08-02 2018-08-30
## Levels: 2018-08-02 2018-08-30
```

13. Using the `unique` function, determine how many different plots were sampled at Niwot Ridge. How is the information obtained from `unique` different from that obtained from `summary`?

```r
# Below finds unique plots sampled at Niwot Ridge.

unique_plots <- unique(Litter$plotID)

# Below counts the number of unique plots.

num_unique_plots <- length(unique_plots)

num_unique_plots
```

```
## [1] 12
```

```r
# Below compares unique and summary function


# Using unique()
unique(Litter$plotID)
```

```
##  [1] NIWO_061 NIWO_064 NIWO_067 NIWO_040 NIWO_041 NIWO_063 NIWO_047 NIWO_051
##  [9] NIWO_058 NIWO_046 NIWO_062 NIWO_057
## 12 Levels: NIWO_061 NIWO_064 NIWO_067 NIWO_040 NIWO_041 NIWO_063 ... NIWO_057
```

```r
# Using summary()
summary(Litter$plotID)
```

```
## NIWO_061 NIWO_064 NIWO_067 NIWO_040 NIWO_041 NIWO_063 NIWO_047 NIWO_051
##       17       16       17       20       19       14       15       14
## NIWO_058 NIWO_046 NIWO_062 NIWO_057
##       16       18       14        8
```

Answer: Unique() lists all distinct values in a column. It's useful when we want to know how many distinct plots exist. Summary() provides summary statistics of a column, including counts. It's useful when we want to see how frequently each plot was sampled.

14. Create a bar graph of functionalGroup counts. This shows you what type of litter is collected at the Niwot Ridge sites. Notice that litter types are fairly equally distributed across the Niwot Ridge sites.

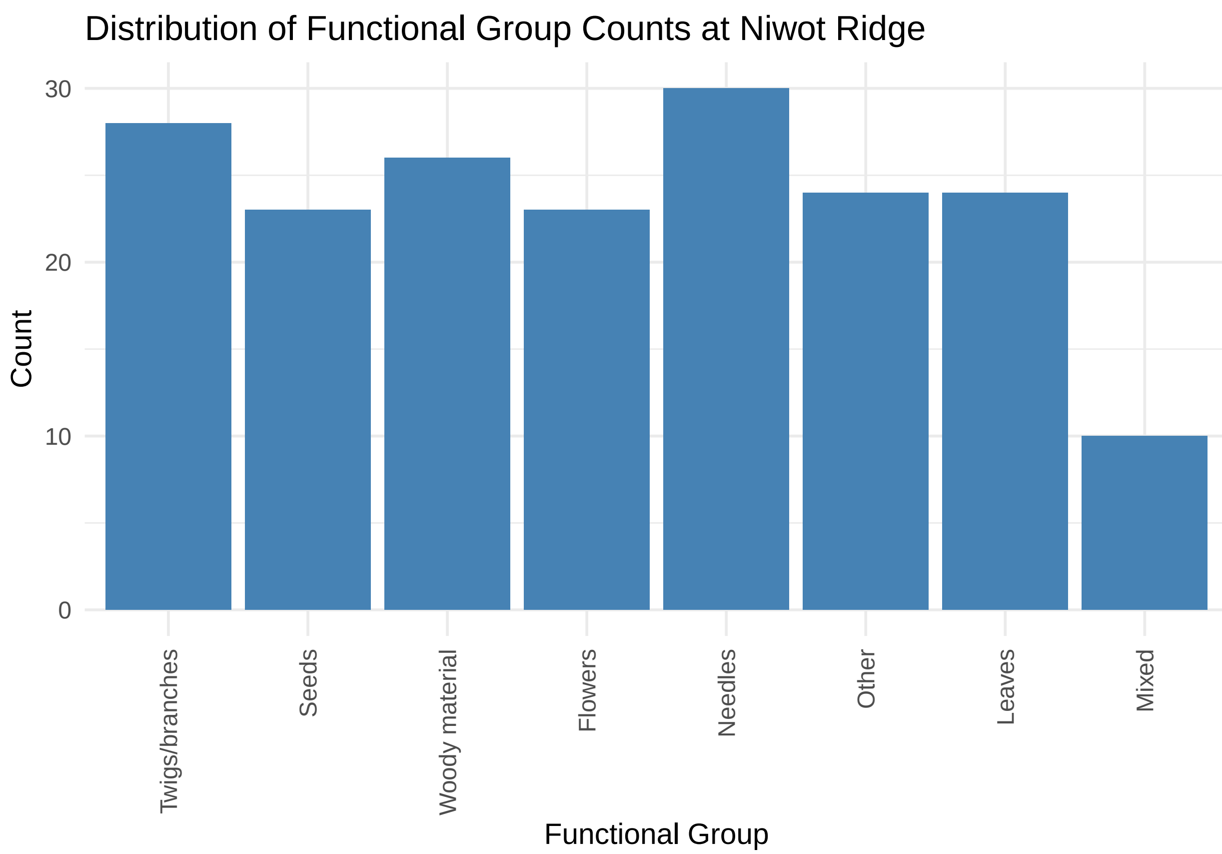
```
# Below verifies the categories present...
unique(Litter$functionalGroup)
```

```
## [1] Twigs/branches Seeds          Woody material Flowers        Needles
## [6] Other          Leaves         Mixed
## 8 Levels: Twigs/branches Seeds Woody material Flowers Needles Other ... Mixed
```

```
# Below counts the occurences of each functional group

functional_group_counts <- as.data.frame(table(Litter$functionalGroup))


functional_group_counts
```

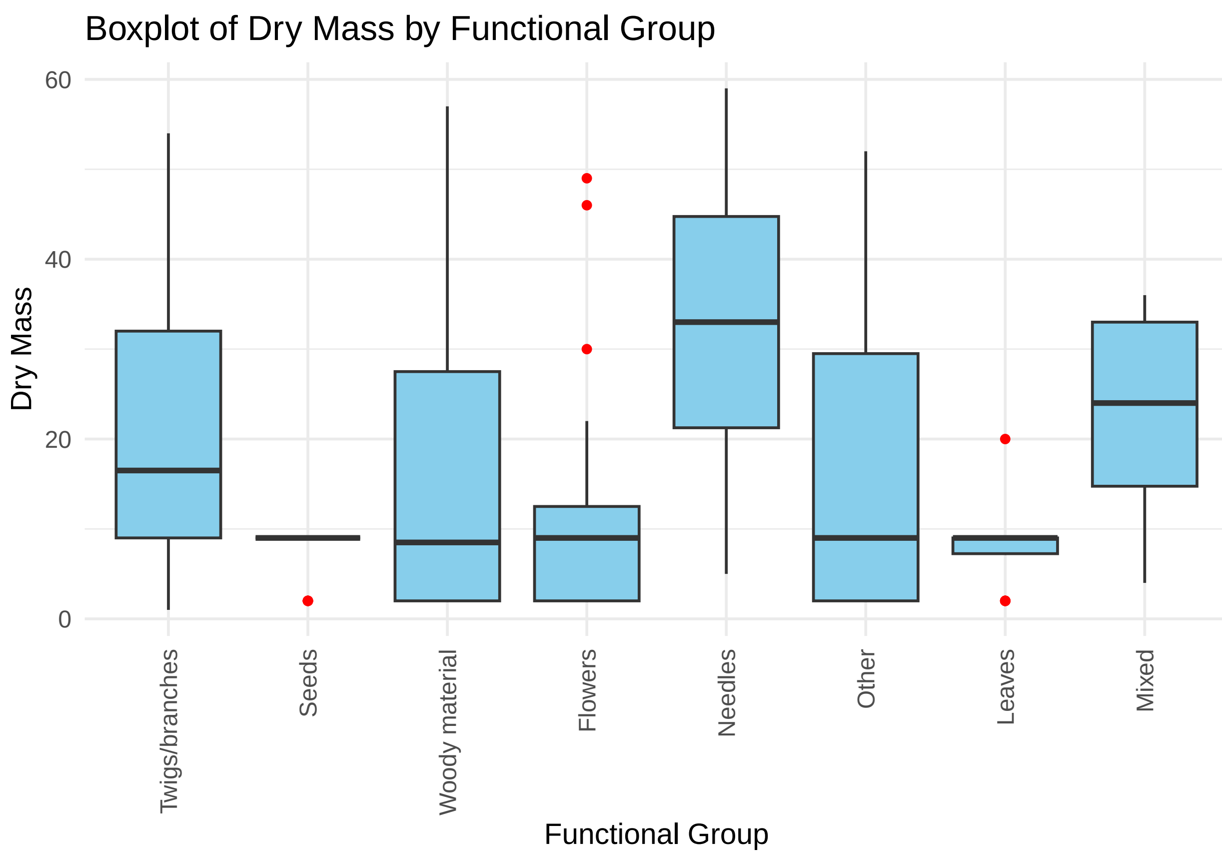
```
##             Var1 Freq
## 1 Twigs/branches   28
## 2          Seeds   23
## 3 Woody material   26
## 4        Flowers   23
## 5        Needles   30
## 6          Other   24
## 7         Leaves   24
## 8          Mixed   10
```

```
library(ggplot2)

# Below plots the bar graph
ggplot(functional_group_counts, aes(x = Var1, y = Freq)) + geom_bar(stat = "identity",
    fill = "steelblue") + labs(title = "Distribution of Functional Group Counts at Niwot Ridge",
    x = "Functional Group", y = "Count") + theme_minimal() + theme(axis.text.x = element_text(angle = 9
    vjust = 0.5, hjust = 1))  # This rotates the label for better visualization.
```

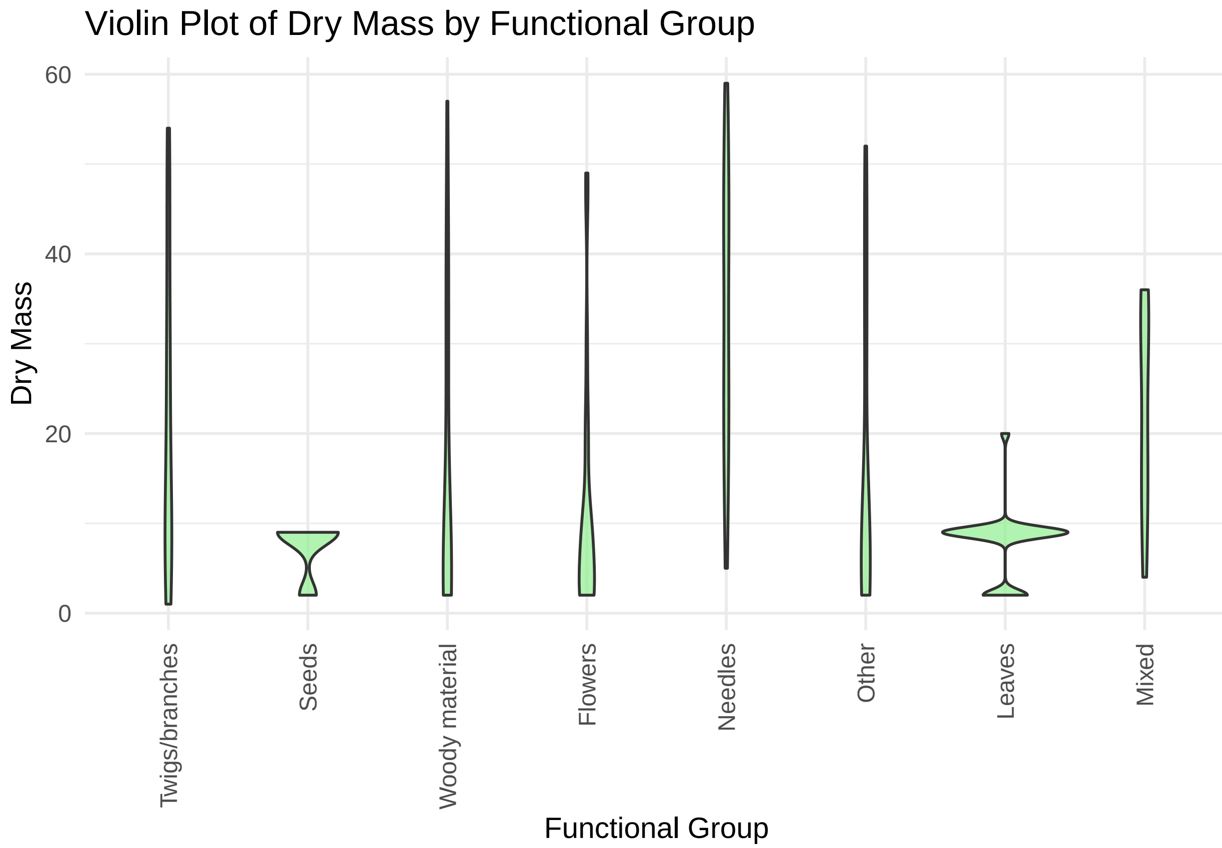


15. Using `geom_boxplot` and `geom_violin`, create a boxplot and a violin plot of dryMass by functionalGroup.

```
# Below converts functional group to a factor if needed.... .

Litter$functionalGroup <- as.factor(Litter$functionalGroup)

# Below converts dryMass to a numeric if needed....
Litter$dryMass <- as.numeric(Litter$dryMass)


# Below creates the boxplot

ggplot(Litter, aes(x = functionalGroup, y = dryMass)) + geom_boxplot(fill = "skyblue",
    outlier.color = "red", outlier.shape = 16) + labs(title = "Boxplot of Dry Mass by Functional Group"
    x = "Functional Group", y = "Dry Mass") + theme_minimal() + theme(axis.text.x = element_text(angle =
    vjust = 0.5, hjust = 1))  # This rotates the x-axis labels for better readability....
```




```r
# Below creates the violin plot

ggplot(Litter, aes(x = functionalGroup, y = dryMass)) + geom_violin(fill = "lightgreen",
    alpha = 0.7) + labs(title = "Violin Plot of Dry Mass by Functional Group", x = "Functional Group",
    y = "Dry Mass") + theme_minimal() + theme(axis.text.x = element_text(angle = 90,
    vjust = 0.5, hjust = 1))  # This rotates the x-axis labels for better readability....
```



Why is the boxplot a more effective visualization option than the violin plot in this case?

> Answer: Boxplot clearly shows the spread (range, median, quartiles) and outliers. It's more readable and interpretable. In this case, the data for many functional groups is highly skewed/contains few data points; which makes the violin plots appear thin and stereteched. It's hard to interpret anything from this illustration of data density. Boxplot on the other hand is more informative with clear box lengths, lines, and red dots.

What type(s) of litter tend to have the highest biomass at these sites?

> Answer: Needles and Twigs/Branches seem to have the highest biomass. Their medians are higher than others and their IQRs extend higher - greater overall dry mass.