Assignment 3: Data Exploration

Jake Whisler

Fall 2023

OVERVIEW

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

Directions

1. Rename this file `<FirstLast>_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 Sakai.

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. Check your working directory, load necessary packages (tidyverse, lubridate), 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 subcommand to read strings in as factors.

```
# Loading the tidyverse and lubridate packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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(lubridate)

# Reading in two comma-separated value files for data and naming them as variables in our environment
Neonics <- read.csv("./Data/Raw/ECOTOX_Neonicotinoids_Insects_raw.csv", stringsAsFactors = TRUE)
Litter <- read.csv("./Data/Raw/NEON_NIWO_Litter_massdata_2018-08_raw.csv", stringsAsFactors = TRUE)
```

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: Insects are often a core part of ecosystem food webs, acting as sustenance for other creatures, both large and small. Not only will toxins potentially harm their survival in particular, but even if the toxic substances in the insecticides don't outright kill the insects, the creatures consuming those insects may suffer repercussions later on.

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: Litter and woody debris can be an indicator of ecosystem health, as it plays a role in moving nutrients through an ecosystem and is a source of microhabitat for smaller creatures. It can also influence nearby freshwater ecosystems, as too much woody debris can divert the flow of water in a stream, for instance.

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. Sampling occurs at NEON sites that contain woody vegetation at least 2 meters tall. 2. Trap placement can be either randomized or targeted, depending on vegetation. 3. Ground traps are sampled once per year.

Obtain basic summaries of your data (Neonics)

5. What are the dimensions of the dataset?

```
#Finding the number of columns and rows in the Neonics dataframe, respectively
print(ncol(Neonics))
```

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

```r
print(nrow(Neonics))
```

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

```r
#There are 4623 observations, and 30 variables recorded for each observation.
```

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?

```r
# Using the summary function on the specific Effect column of the dataframe,
# sorted by prevalence
sort(summary(Neonics$Effect))
```

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

Answer: The most common effects studied are "Mortality", "Population", and "Behavior". These effects are directly related to the insects' chances of survival, as changes in the species' mortality and population will have rippling effects throughout the ecosystem. Changes to behavior may also directly or indirectly affect their odds of survival, or alter the rate of their consumption by higher predators.

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: The `sort()` command can sort the output of the summary command...]

```r
# Using the summary function on the specific Species.Common.Name column of the dataframe,
# sorted by prevalence
tail(sort(summary(Neonics$Species.Common.Name)))
```

```
##            Bumble Bee   Carniolan Honey Bee Buff Tailed Bumblebee
##                   140                   152                   183
##        Parasitic Wasp             Honey Bee               (Other)
##                   285                   667                   670
```

Answer: The six most commonly studied species in the dataset are the Honey Bee, the Parasitic Wasp, the Buff Tailed Bumblebee, the Carniolan Honey Bee, the Bumble Bee, and the Italian Honeybee. These are obviously all bees or bee-adjacent species, and they likely hold more scientific interest due to their disproportionately large role in shaping their ecosystem through the movement of pollen.

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?

```
data.class(Neonics$Conc.1..Author.)
```

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

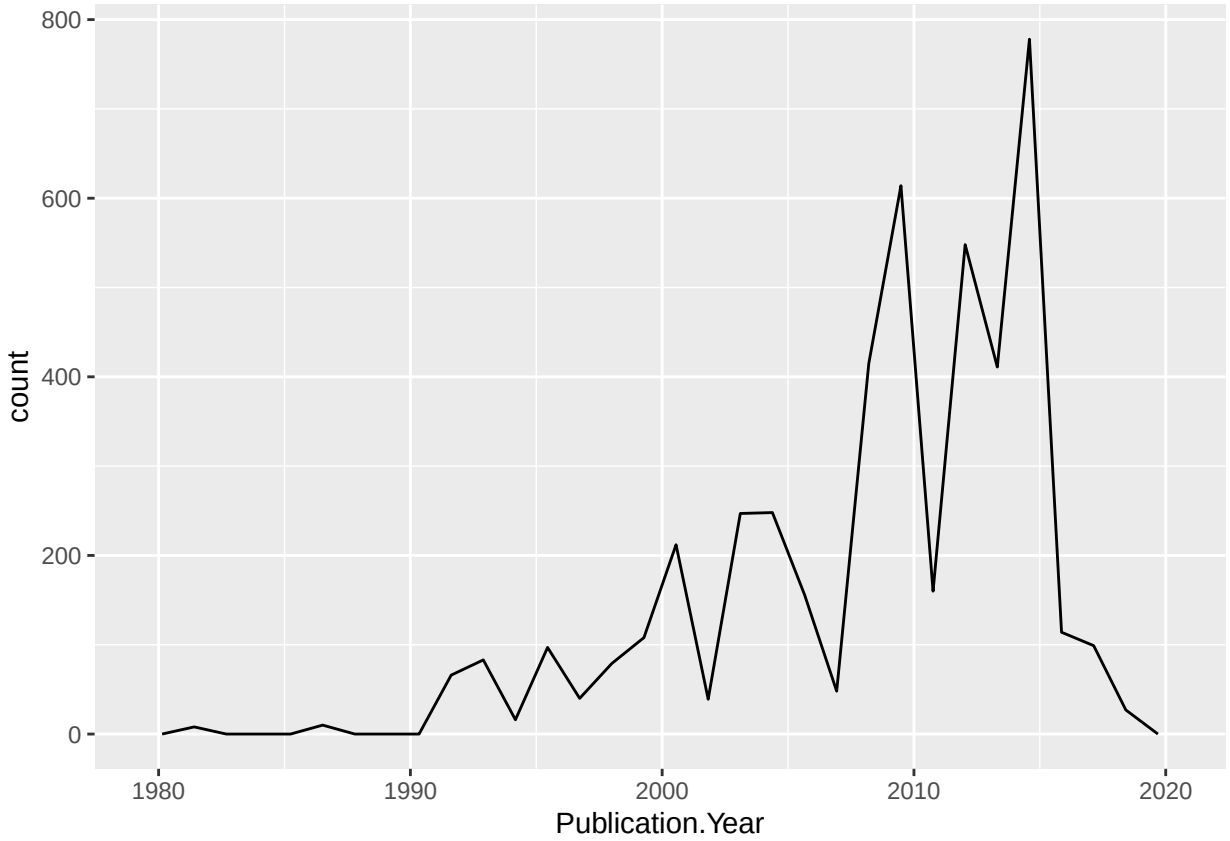
Answer: The Conc.1..Author column's data class is "factor". This is because this column is not actually numerical data, and is instead being read as a factor by the read.csv() function that was used to import the data in the first place.

Explore your data graphically (Neonics)

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

```
# Printing a simple frequency histogram using Publication.Year as the variable
print(ggplot(Neonics) +
  geom_freqpoly(aes(x = Publication.Year)))
```

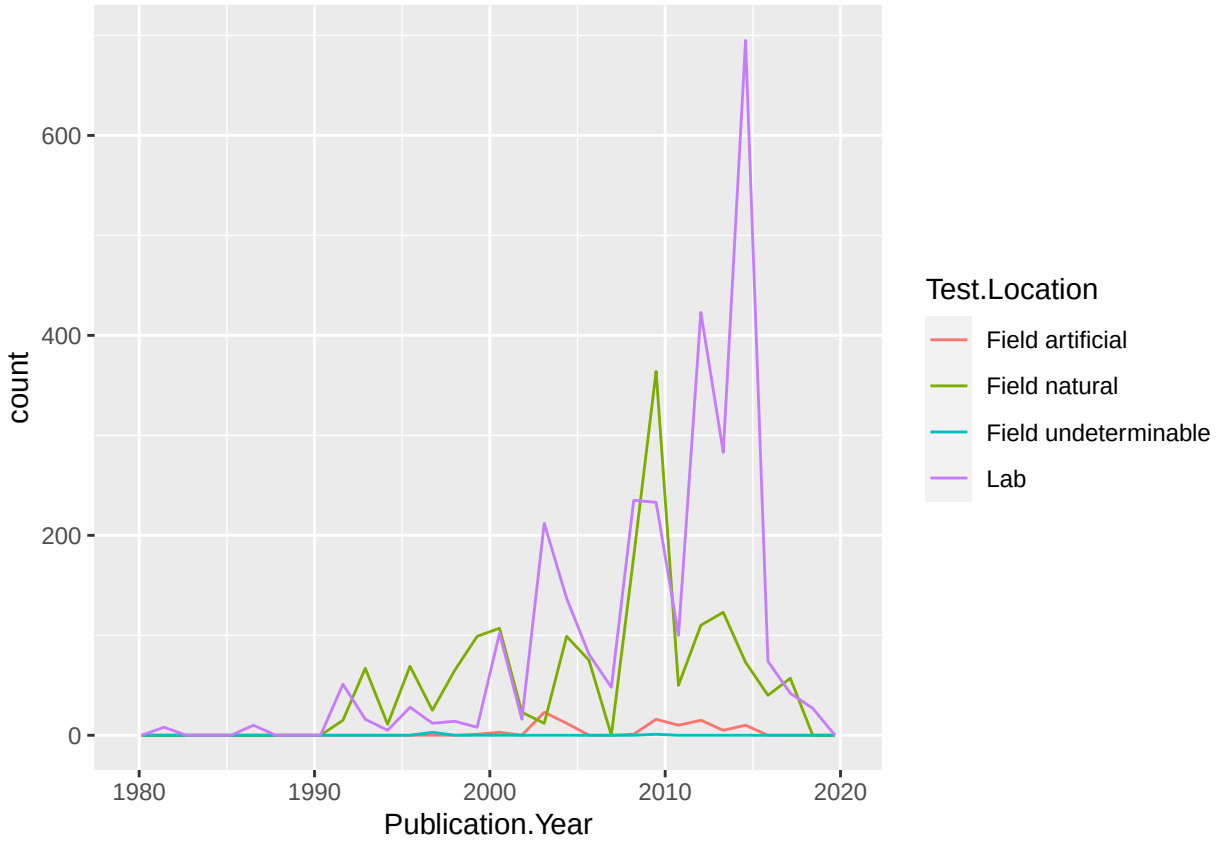
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

```
print(ggplot(Neonics, aes(x = Publication.Year, group = Test.Location, col = Test.Location)) +
        geom_freqpoly())
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



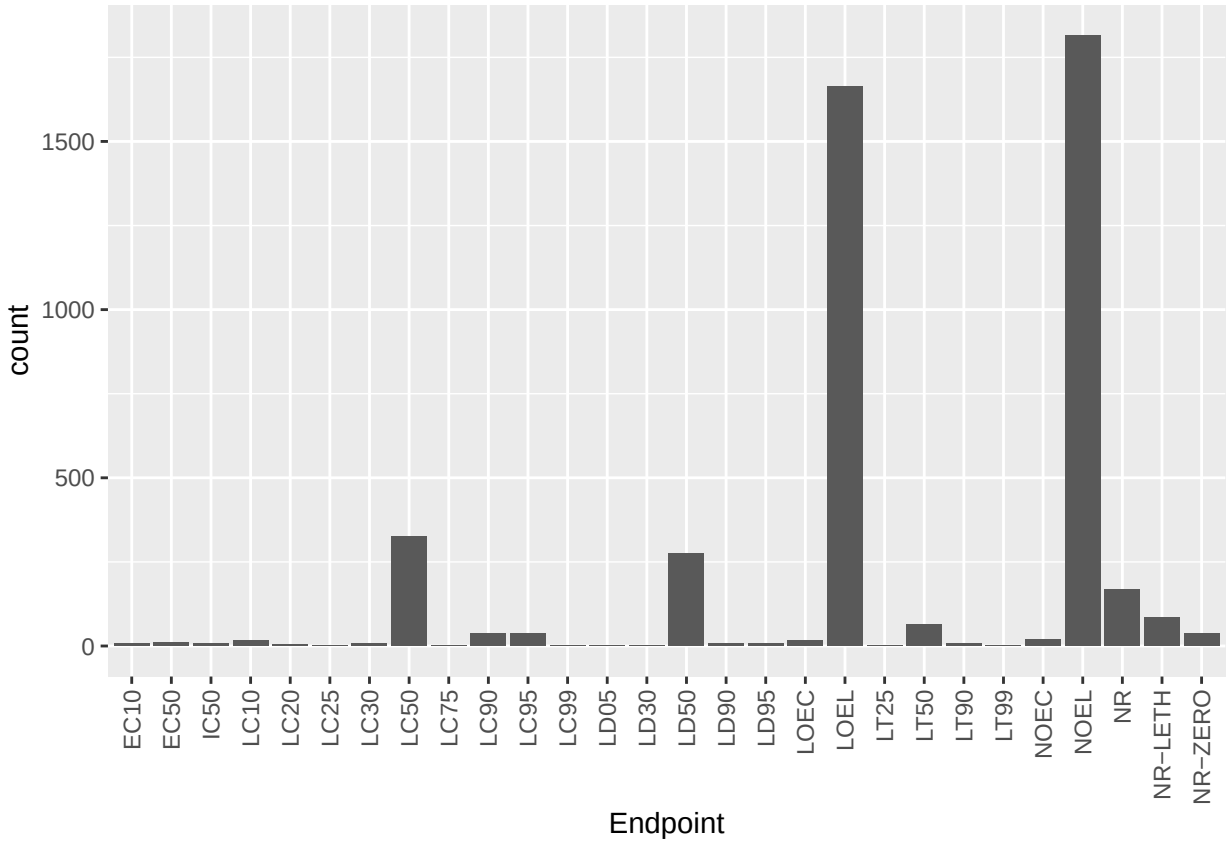
Interpret this graph. What are the most common test locations, and do they differ over time?

> Answer: It appears as though most of the tests occurred in the laboratory, followed by in the field naturally, then in the field artificially, and finally, a few instances of underdeterminable field circumstances. These graphs do differ over time; after the year 2010, lab tests became far more prevalent than any other test location.

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...]

```
print(ggplot(Neonics, aes(x = Endpoint)) +
        geom_bar() +
        theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)))
```



Answer: The most common end points are LOEL and NOEL, which are defined as "lowest-observable-effect-level" and "no-observable-effect-level" according to the Code Appendix.

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.

```
# Checking the collectDate object class (it returns as a factor)
data.class(Litter$collectDate)
```

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

```
# Changing the collectDate object class to Date
Litter$collectDate <- as.Date(Litter$collectDate)

# Checking to make sure it changed the object class
data.class(Litter$collectDate)
```

```
## [1] "Date"
```

```
# Using the unique function to find dates sampled in August 2018
unique(x = Litter$collectDate)
```

```
## [1] "2018-08-02" "2018-08-30"
```

```
# Litter was sampled on August 8th and August 30th in 2018.
```

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

```
# Using the unique function to determine how many plots were sampled at Niwot Ridge
unique(x = 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_040 NIWO_041 NIWO_046 NIWO_047 NIWO_051 NIWO_057 ... NIWO_067
```

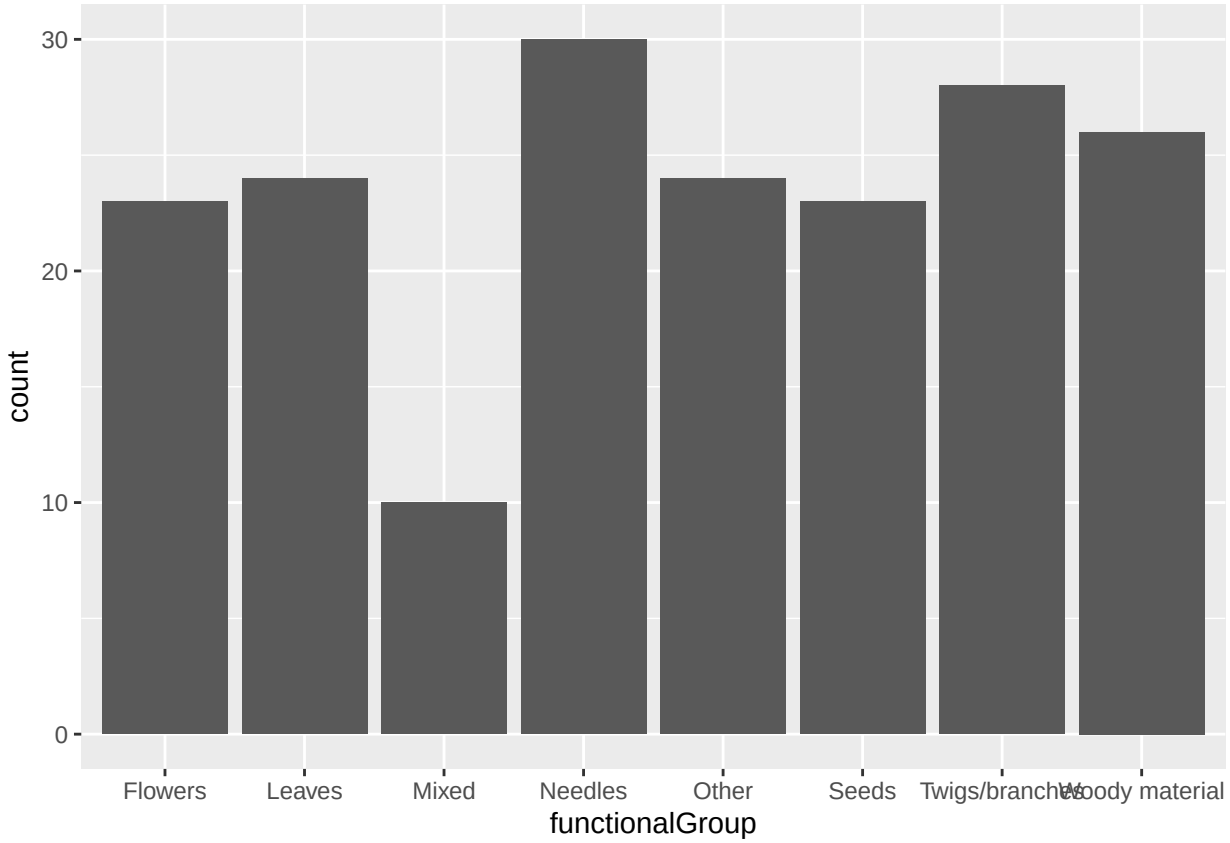
```
# Using the summary function with plotID
summary(Litter$plotID)
```

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

Answer: Twelve unique plots were sampled at Niwot Ridge. The "unique" function shows how many unique plots there are, while the "summary" function shows how many samples were taken of each plot.

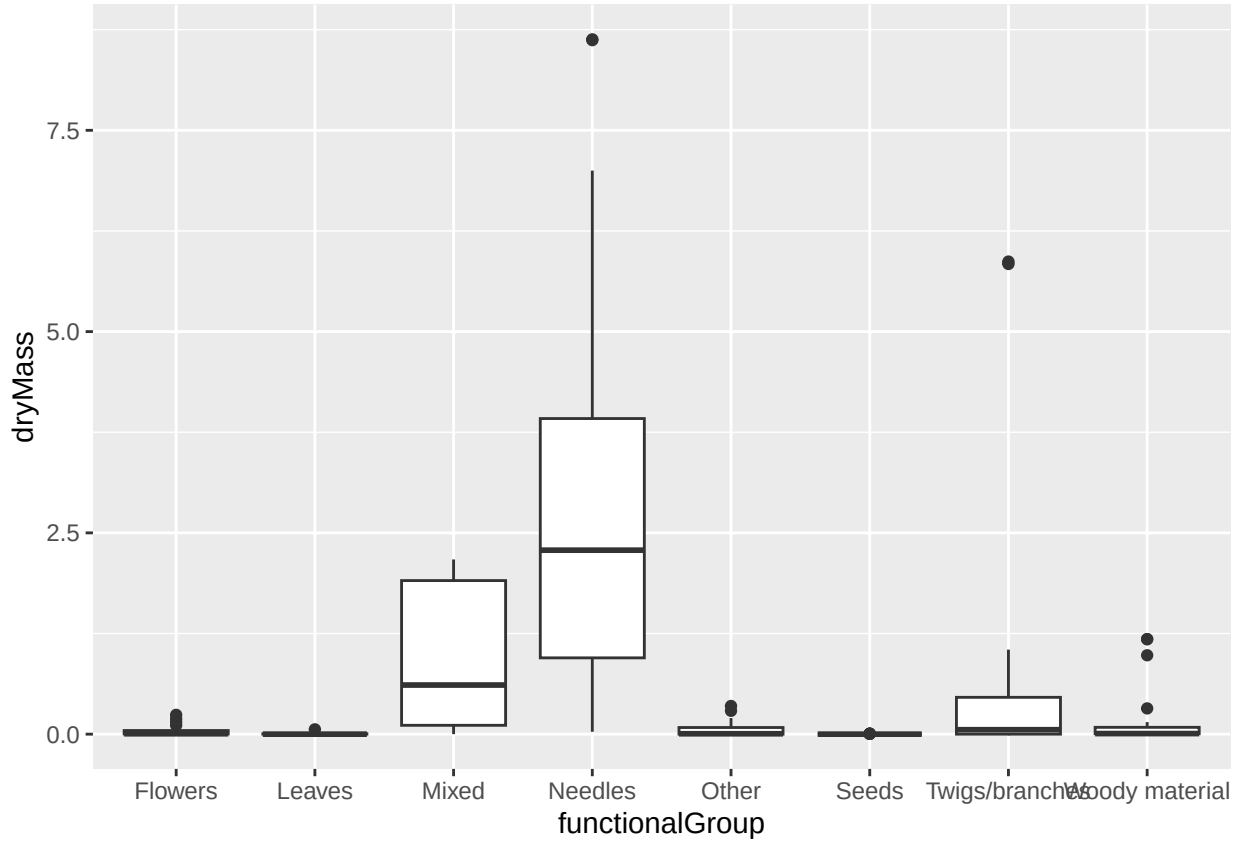
14. Create a bar graph of functionalGroup counts. This shows you what type of litter is collected at the Niwot Ridge sites. Notice that litter types are fairly equally distributed across the Niwot Ridge sites.

```
# Using ggplot and geom_bar() to create a bar graph of functionalGroup counts
print(ggplot(Litter, aes(x = functionalGroup)) +
        geom_bar())
```

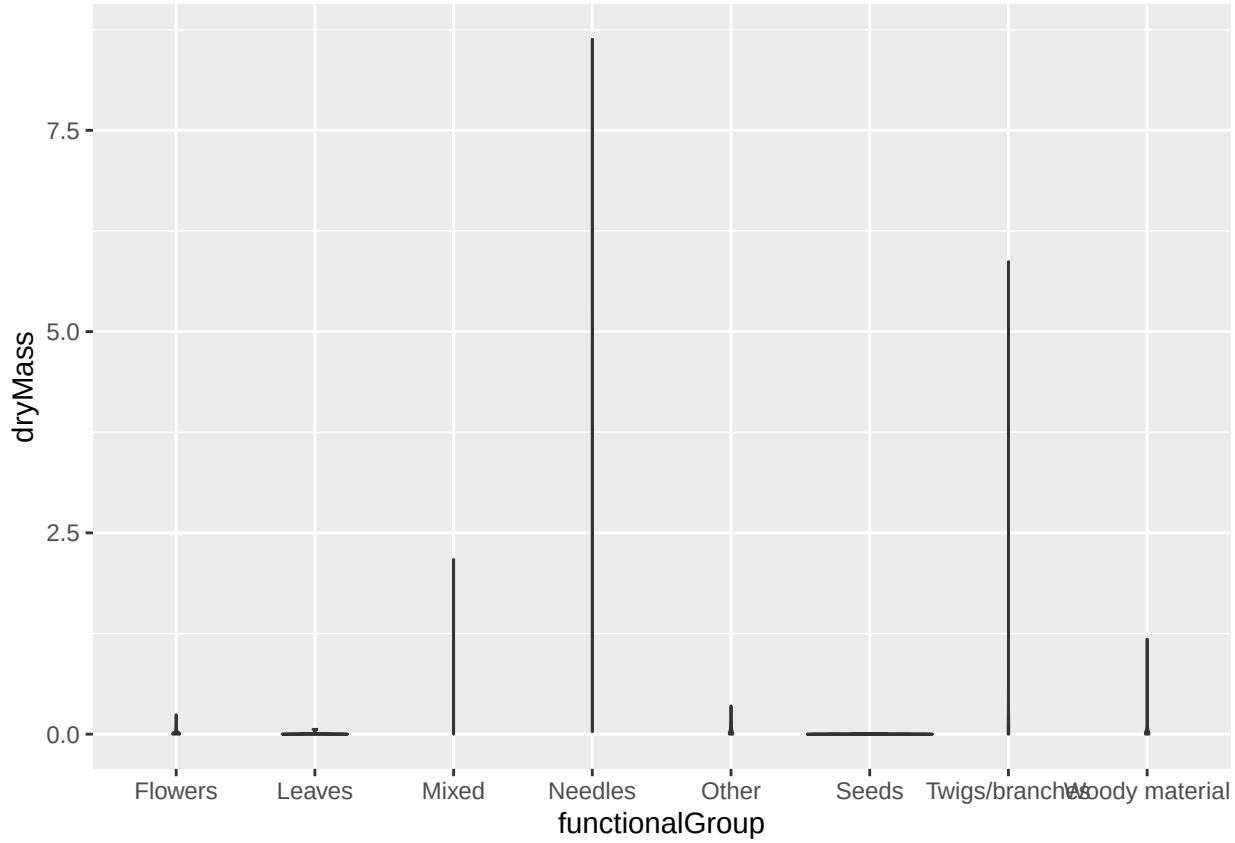


15. Using `geom_boxplot` and `geom_violin`, create a boxplot and a violin plot of dryMass by functionalGroup.

```
# Creating a boxplot with geom_boxplot()
print(ggplot(Litter) +
        geom_boxplot(aes(x = functionalGroup, y = dryMass)))
```




```
# Creating a violin plot with geom_violin()
print(ggplot(Litter) +
        geom_violin(aes(x = functionalGroup, y = dryMass), draw_quantiles = c(0.25, 0.5, 0.75)))
```



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

> Answer: The boxplot is more effective because it shows the influence of outliers on the mean dryMass for certain types of litter, giving a greater understanding of the data's distribution.

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

> Answer: Needles tend to have the highest biomass at these sites, followed by mixed litter and twigs/branches. Twigs/branches has a very strong outlier that has exacerbated the overall mean of its biomass.