Assignment 3: Data Exploration

Jenn McNeill

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.

```
# determine working directory
getwd()
```

```
## [1] "/Users/jennifermcneill/EDE_Fall2023/EDE_Fall2023"
```

```
# load necessary packages
library(tidyverse)
library(lubridate)

# use a relative filepath to upload datasets
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: Understanding the effects that neonicotinoids have on insects is important because insects serve a wide variety of purposes in agriculture. Insects are responsible for pollination and decomposition, they serve as a food source for other animals, and they can cause serious damage to crops. It is important to study the effect of the insecticide on all different insects to discern how different insects are affected by it. The agricultural industry is interested in having as much information as possible so that when this insecticide is sprayed, they know how the ecosystem will respond. The "effect" column of the data shows that different insects responded differently to the insecticide; effects ranged between "mortality,"growth," "reproduction," "development," and more.

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: Studying litter and woody debris that falls to the ground in forests is important for a variety of ecological and environmental reasons. Litter and woody debris on the forest floor plays a role in nutrient cycling and carbon sequestration, and it serves as an ecosystem for smaller species that are vital to forest health. These indicator variables can also reveal information about how the public is interacting with the forest, tree health, the risk of future forest fires, and potential need for rehabilitation or restoration. A forest is a complex ecosystem, and the forest floor is the foundation from which it all grows. The health of the forest floor is crucial to the health of the entire ecosystem, and it should not be overlooked.

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. One litter trap pair (one elevated trap and one ground trap) is deployed for every 400 m^2 plot area, resulting in 1-4 trap pairs per plot. 2. Plot edges must be separated by a distance 150% of one edge of the plot. 3. Trap placement within plots may be either targeted or randomized, depending on the vegetation.

Obtain basic summaries of your data (Neonics)

5. What are the dimensions of the dataset?

```
dim(Neonics)
```

```
## [1] 4623   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?

```r
# point to a specific column using the $
summary(Neonics$Effect)
```

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

```r
# create a variable to store the summary of the effect column
common_effects <- sort(summary(Neonics$Effect))

# print the effects in sorted order to discern that mortality and population
# are the most common effects
common_effects
```

```
##       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: Mortality and Population are the two effects that are studied significantly more than others. These two effects are of interest because they are directly related to the effectiveness and safety of the insecticide. This data is critical for deciding whether to move forward with the usage of the insecticide on a larger scale after the testing stages.

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
# create a variable to store the sorted summary of the species common name column
commonly_studied <- sort(summary(Neonics$Species.Common.Name))

# find the length of the species common name column
# to determine where the "top six" will fall
length(summary(Neonics$Species.Common.Name))
```

```
## [1] 100
```

```
# create a variable to store the values for the last six values
# in the variable commonly_studied
top_six <- commonly_studied[95:100]

# print the top six species
top_six
```

```
##            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 are 6. Bumble Bee 5. Carniolan Honey Bee 4. Buff Tailed Bumblebee 3. Parasitic Wasp 2. Honey Bee 1. Other The commonality between most of these species is that they are pollinators. In agriculture, pollination is essential for crop production. Without these insects, crop yields and ecosystem health would be negatively impacted. Thus, it is logical that impact of the insecticide on these insects is more widely studied than the impact on other insects.

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?

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

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

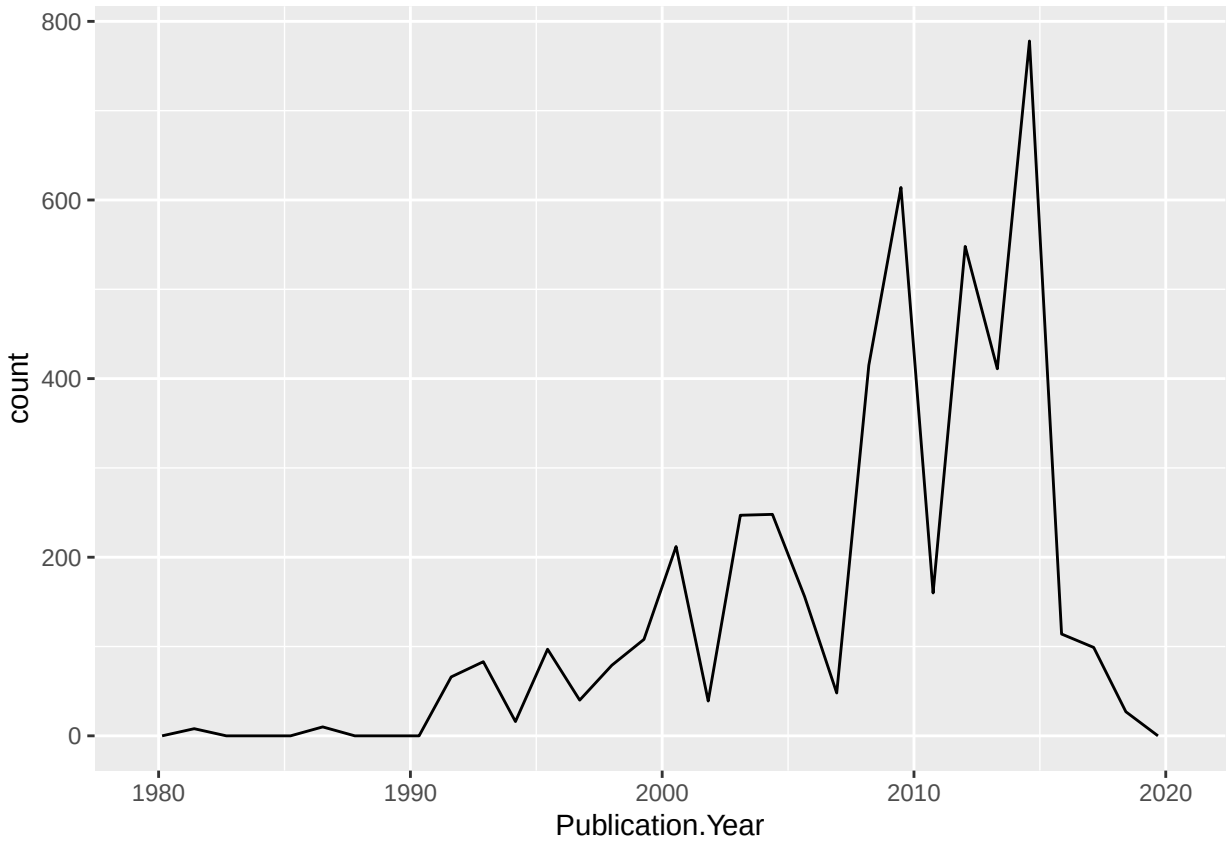
Answer: The class of the Conc.1..Author. column is factor because when I read the dataset into R (question 1), I included "stringsasFactors=TRUE" in my read.csv function.

Explore your data graphically (Neonics)

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

```
# set the aesthetics of the geom_freqpoly function to display
# Publication Year on the x-axis
ggplot(Neonics) +
  geom_freqpoly(aes(x = Publication.Year))
```

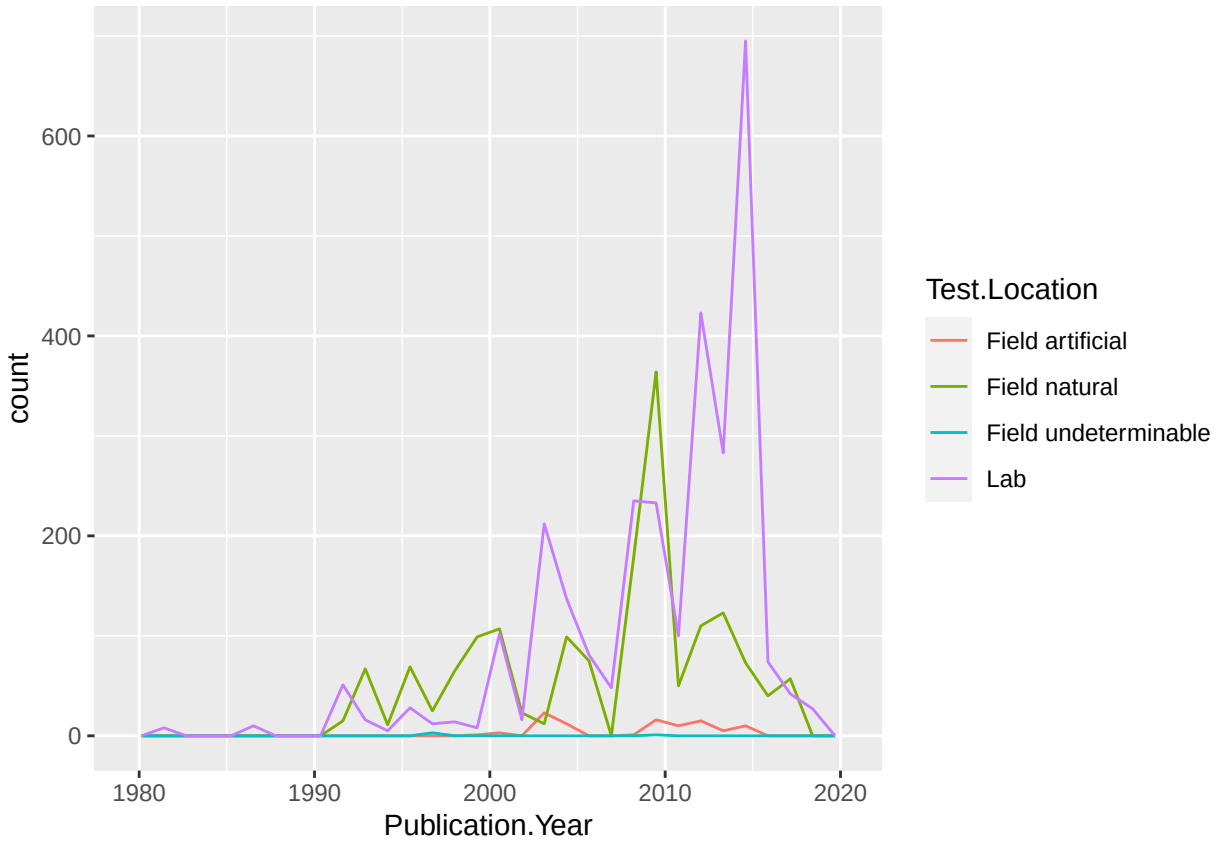
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

```
# add 'color =' to the aesthetics of the plot so that each test location
# displays as a different line in a different color
ggplot(Neonics) +
  geom_freqpoly(aes(x = Publication.Year, color = Test.Location))
```

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



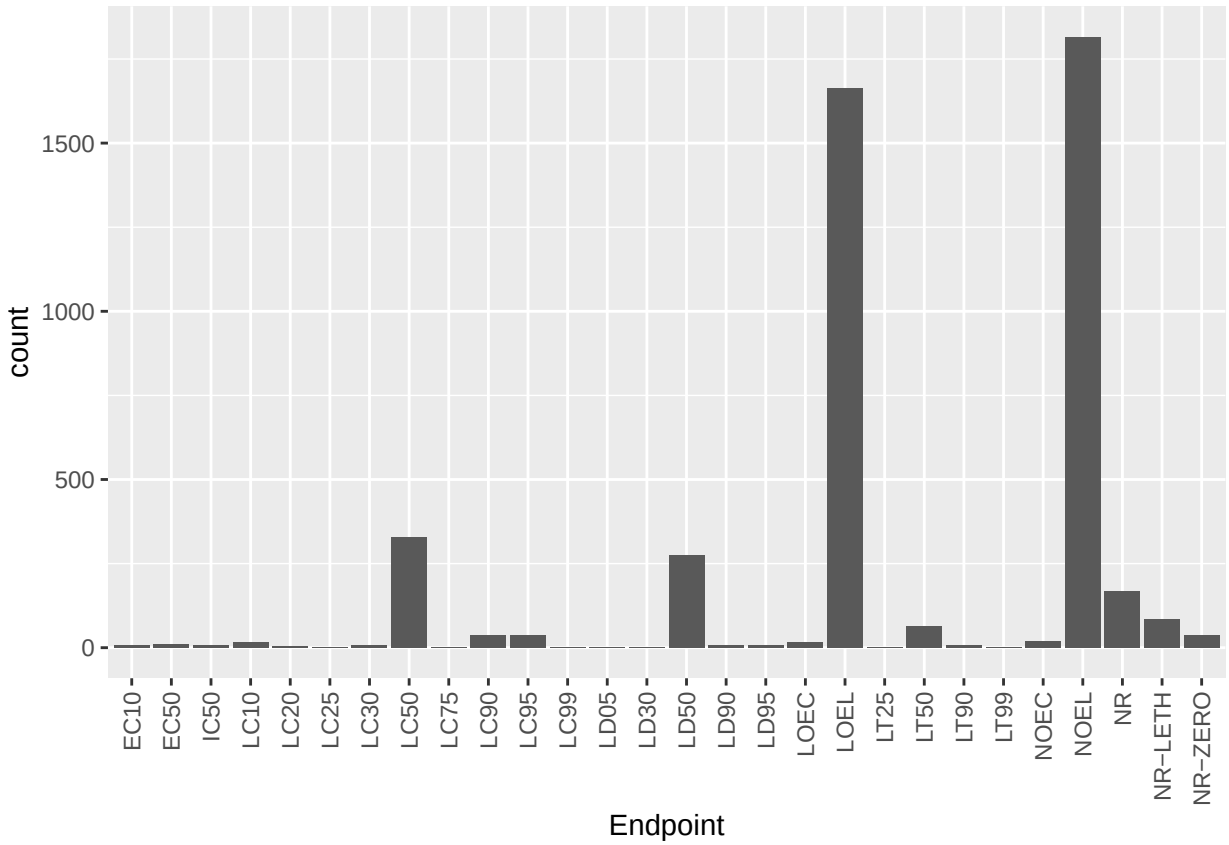
Interpret this graph. What are the most common test locations, and do they differ over time?

> Answer: The two most common test locations are the natural field and the lab. Up until 2000, these natural field was slightly more common then the lab. After 2000, the lab mostly overtook the natural field as the most common test location. Since 2012, the lab has increased sharply in popularity, leaving the natural field far behind in comparison. The artificial field is quite low in popularity, only having been used sparingly around 2003.

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...]

```
# set the aesthetics of the geom_bar function to display Endpoint on the x-axis
# alter the theme of the bar graph as instructed
ggplot(Neonics) +
  geom_bar(aes(x = Endpoint)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



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

```
class(Litter$collectDate)
```

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

```
# original class is factor
Litter$collectDate <- as.Date(Litter$collectDate, format = "%Y-%m-%d")
# use the as date function to change the class
class(Litter$collectDate)
```

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

```
# updated class is a date
Litter$collectDate
```

```
##   [1] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##   [6] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [11] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [16] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [21] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [26] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [31] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [36] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [41] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [46] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [51] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [56] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [61] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [66] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [71] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [76] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [81] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [86] "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02" "2018-08-02"
##  [91] "2018-08-02" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
##  [96] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [101] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [106] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [111] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [116] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [121] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [126] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [131] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [136] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [141] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [146] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [151] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [156] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [161] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [166] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [171] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [176] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [181] "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30" "2018-08-30"
## [186] "2018-08-30" "2018-08-30" "2018-08-30"
```

```
# print the dates to check that they were properly formatted
# after changing the class
```

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`?

```
# test unique function and find length of unique function
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_040 NIWO_041 NIWO_046 NIWO_047 NIWO_051 NIWO_057 ... NIWO_067
```

```
length(unique(Litter$plotID))
```

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

```
# compare with summary function and length of summary function
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
```

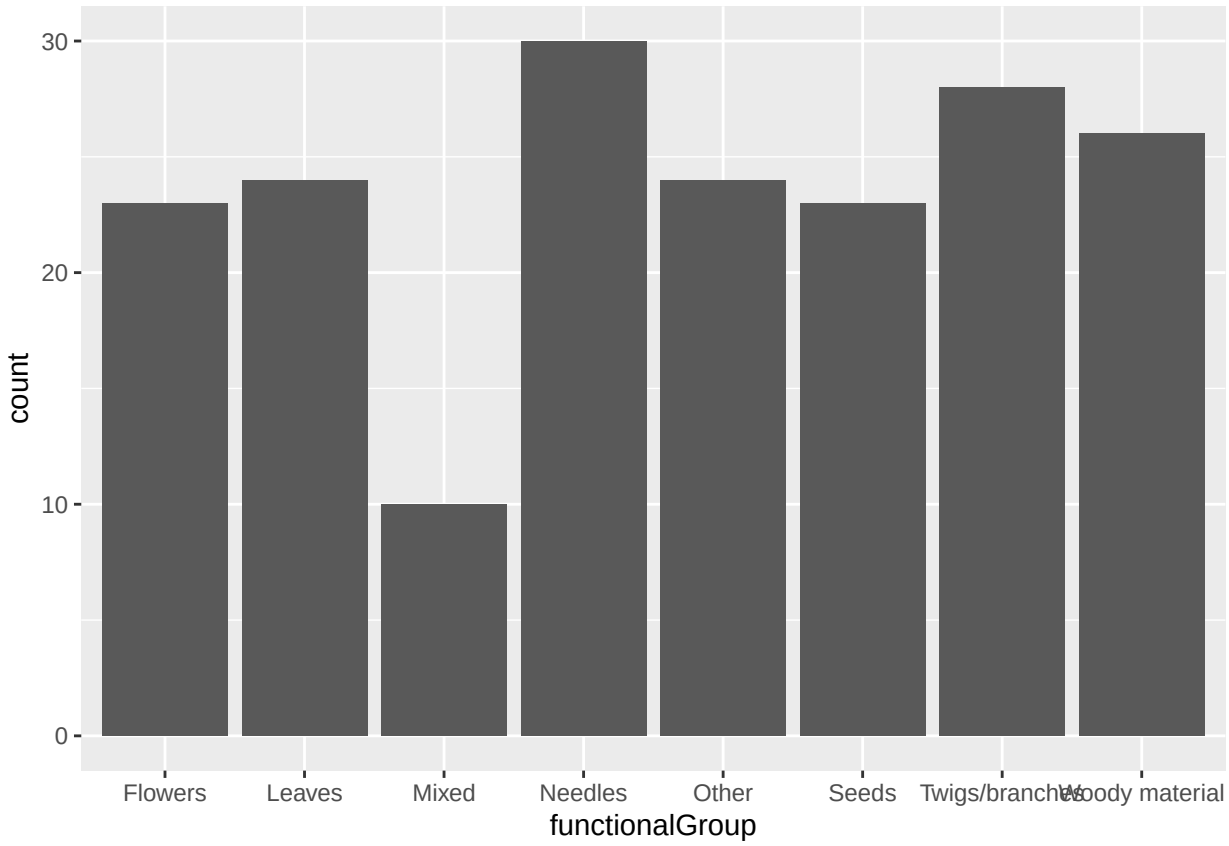
```
length(summary(Litter$plotID))
```

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

Answer: The unique function prints all 12 names of the unique plots that were sampled at Niwot ridge. The summary function prints all 12 names of the unique plots that were sampled at Niwot ridge and gives a count of how many times each unique value appears in the data set. The unique function is more broad and only gives the specific unique values you seek. The summary function is more specific and gives extra detail corresponding to the unique values.

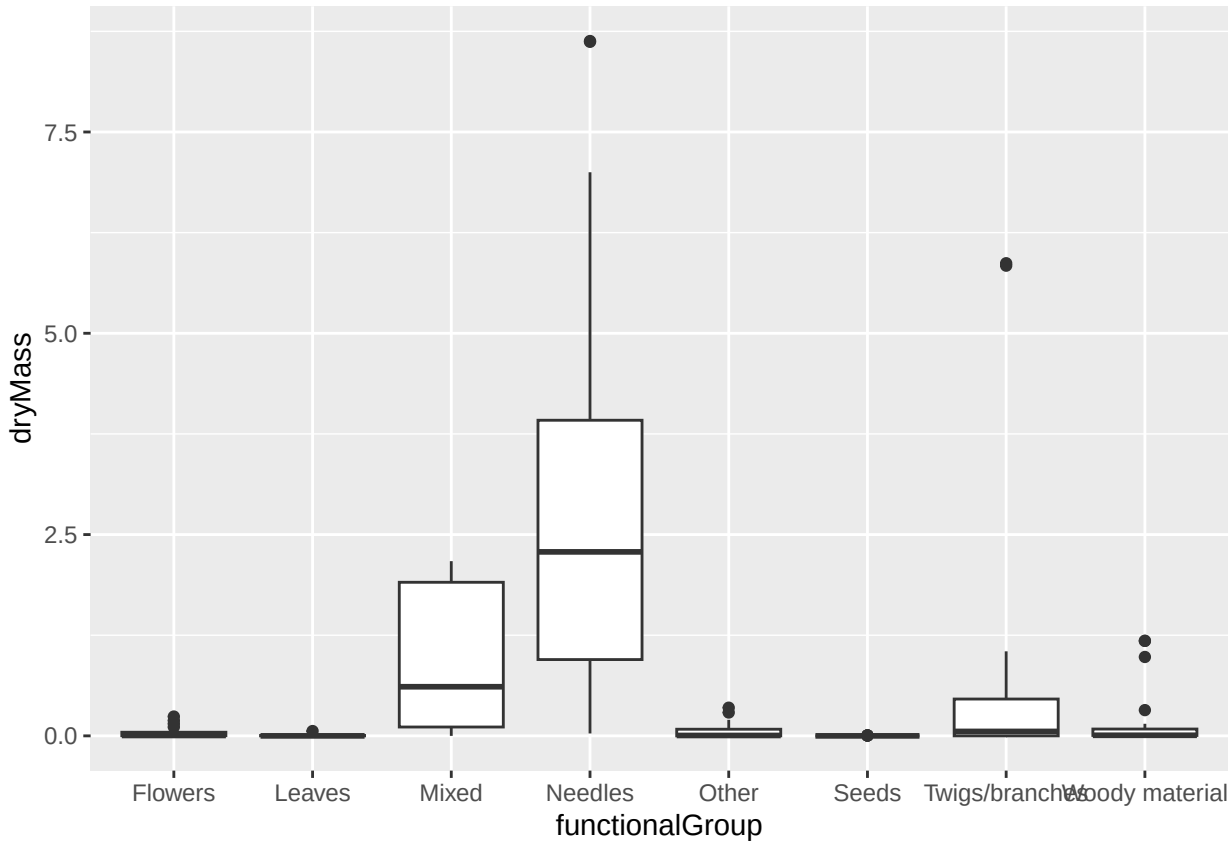
14. Create a bar graph of functionalGroup counts. This shows you what type of litter is collected at the Niwot Ridge sites. Notice that litter types are fairly equally distributed across the Niwot Ridge sites.

```
# create bar graph
ggplot(Litter) +
  geom_bar(aes(x = functionalGroup))
```

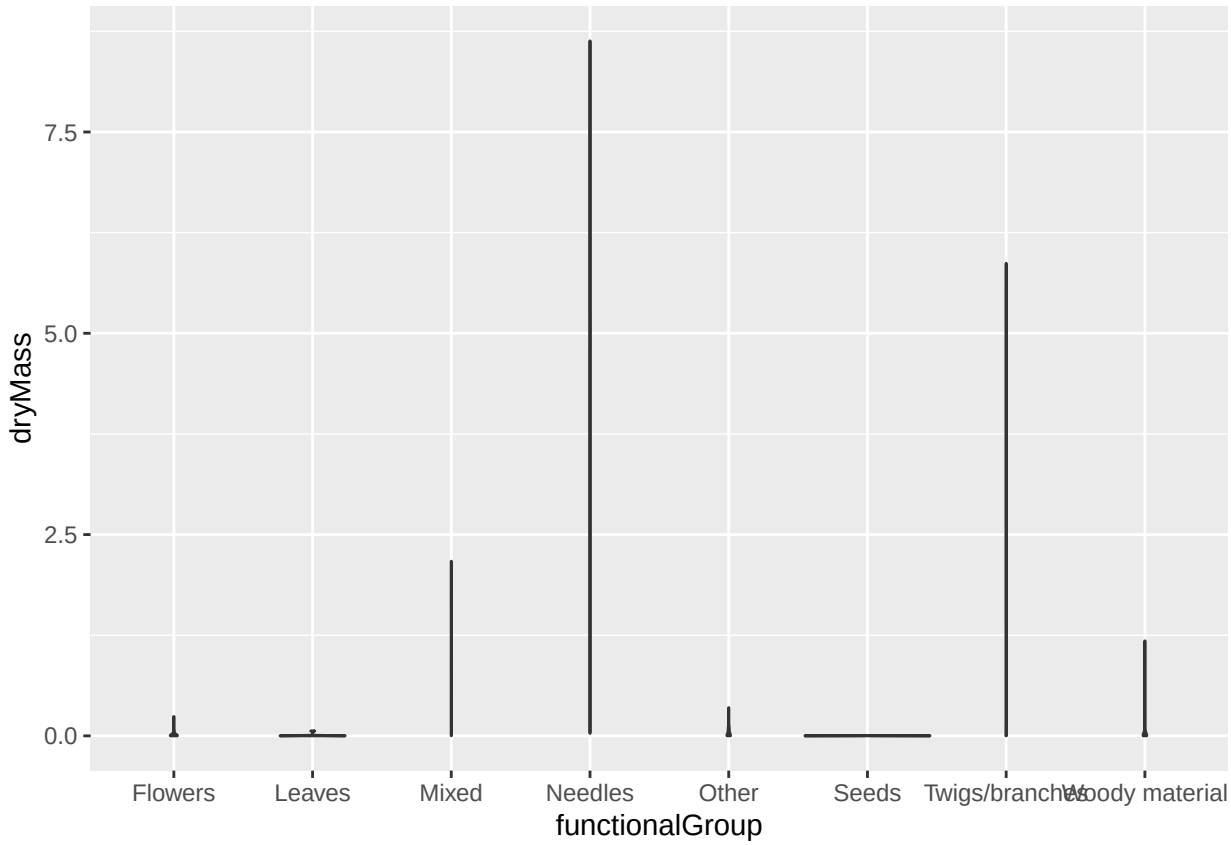


15. Using `geom_boxplot` and `geom_violin`, create a boxplot and a violin plot of dryMass by functionalGroup.

```
# create boxplot
ggplot(Litter) +
  geom_boxplot(aes(x = functionalGroup, y = dryMass))
```




```
# create violin plot
ggplot(Litter) +
  geom_violin(aes(x = functionalGroup, y = dryMass))
```



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

> Answer: The boxplot is more effective because this is not a very large dataset. The boxplot shows the outliers and the mean of each functional group clearly. The violin plot in this case is not effective because there are not enough data points to create clear definition in the shape of each "violin." Because the data set is small with some variability, the violin plot for each functional group appears as a vertical or horizontal line. With the vertical lines in functional groups such as needles and twigs/branches, the mean is undiscernable. In general, I believe a boxplot is better at displaying results for small data sets and violin plots are better at displaying results for large data sets.

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. The boxplot shows the mean for needles being the highest of all the different functional groups.