Assignment 3: Data Exploration

Fiona Price

Fall 2024

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 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 subcommand to read strings in as factors.

```
#Load the dplyr, tidyverse, ggplot2, and here package
library(dplyr)
library(ggplot2)
library(here)
library(tidyverse)

#Upload the Neonics dataset
neonics <- read.csv(
  file = here("./Data/Raw/ECOTOX_Neonicotinoids_Insects_raw.csv"),
  stringsAsFactors = TRUE)
```

```r
#Upload the Litter dataset
litter <- read.csv(
  file = here("./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: Neonicotinoids were originally introduced as a less harmful insecticide. However, research now shows that they are harmful to insects. For example, they are very harmful to bees. The chemicals are absorbed by plants and are thus present in pollen and nectar. Additionally, they have long-lasting lifespans in plants. Overall, we are interested in the ecotoxicology of neonicotinoids on insects because they will affect insect populations, which will then have a cascading effect on ecosystems, as many of these insects are beneficial.

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 has many important ecological functions. First, litter and woody debris breaks down organic matter and returns nutrients back into the soil. Second, they provide habitat for organisms (including insects), helping to sustain biodiversity. Third, litter and woody debris helps retain moisture in the soil which is necessary for plant growth. They can also stabilize soils, helping to prevent erosion. Litter and woody debris are crucial for preserving biodiversity in ecosystems.

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. Litter and woody debris are collected from ground traps and elevated traps. These samples are then weighed. 2. Traps are placed in both targeted and randomized ways, depending on vegetation. Traps are only placed in tower plots. 3. Ground traps are sampled once per year, while elevated traps are sampled at varying frequencies depending on amount of vegetation.

Obtain basic summaries of your data (Neonics)

5. What are the dimensions of the dataset?

```r
#Use the `dim` function to find the number of rows and columns in the dataset
dim(neonics)
```

```
## [1] 4623   30
```

```
#The dataset has 4623 rows and 30 columns.
```

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...]

```
#Use the `summary` function to look at an overview of the types of effects
summary(neonics$Effect)

#Create a table summarizing these effects
neonics_summary <- table(neonics$Effect)

#Now, sort the table using the `sort` function. Decreasing = TRUE indicates that
#the sort should be decreasing (so it starts with the most common effect).
sorted_neonics_summary <- sort(neonics_summary, decreasing = TRUE)
sorted_neonics_summary

#Population is the most common effect. Mortality is the second most common.
```

Answer: Population is the most common effect. Mortality is the second most common. These effects are specifically of interest because they reflect the impact that neonicotinoids have on insect population sizes. This data can be useful for predicting ecosystem dynamics.

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...]

```
#Use the summary function to look at an overview of the common species names.
summary(neonics$Species.Common.Name)

#Create a table summarizing these species.
neonics_species <- table(neonics$Species.Common.Name)

#Now, sort the table using the sort function. Decreasing = TRUE indicates that
#the sort should be decreasing (so it starts with the most common species).
sorted_neonics_species <- sort(neonics_species, decreasing = TRUE)
sorted_neonics_species

#The six most commonly studied species are the honey bee, parasitic wasp, buff
#tailed bumblebee, carniolan honey bee, bumble bee, and Italian honeybee.

#This is another way to do it using maxsum. Here, I'm indicating I want seven
#levels to show (because the 7th will be (other)).
summary(neonics$Species.Common.Name, maxsum = 7)
```

Answer: The six most commonly studied species are the honey bee, parasitic wasp, buff tailed bumblebee, carniolan honey bee, bumble bee, and Italian honeybee. Most of these species are pollinators; they are therefore of more interest because pollination is crucial for plant reproduction. Although parasitic wasps are not pollinators, they too are can indirectly contribute to plant reproduction.

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...]

```
#Use the `view` function to view the dataframe.
view(neonics)

#Use the `class` function to view the class of the `Conc.1..Author` column.
class(neonics$Conc.1..Author.)
```

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

```
#The class is factor.

#Examine the specific categories.
summary(neonics$Conc.1..Author.)
```

```
##    0.37/      10/      NR/       NR        1     1023    0.40/       2/
##      208      127      108       94       82       80       69       63
##       10   0.053/      100      50/     0.5/     0.03    0.05/     0.45
##       62       59       56       51       45       44       43       43
##     0.1/    0.45/     1.0/    2.27/       50    0.125     500/      0.5
##       42       40       40       40       36       33       33       32
##   0.048/    0.15/       1/       48    25.0/      12/    0.027      2.4
##       30       30       30       30       28       27       26       26
##     0.2/    0.56/     100/        3    0.01/    1000/       3/    0.336
##       25       24       23       23       22       22       22       21
##     1.5/     0.05      1.5    2.60/    20.0/        6    6.80/    62.5/
##       21       20       20       20       20       20       20       20
##    0.005     0.4/    0.18/     0.3/     1000       40 0.00355/      0.1
##       18       18       17       17       17       17       16       16
##      0.4     150/      300      80/    0.053     0.24     0.28     125/
##       16       16       16       16       15       15       15       15
##        9   0.0001  0.0004/   0.084/     0.15      0.6    12.5/   144.0/
##       15       14       14       14       14       14       14       14
##     350/    40.0/      48/       56      84/    0.17/      125       14
##       14       14       14       14       14       13       13       13
##       16       17   0.047/    0.25/    0.28/    1.28/    1.81/      112
##       13       13       12       12       12       12       12       12
##      150      2.5/       25      60/      75/    0.02/   0.025/     0.29
##       12       12       12       12       12       11       11       11
##    37.5/       4/        5  (Other)
##       11       11       11     1817
```

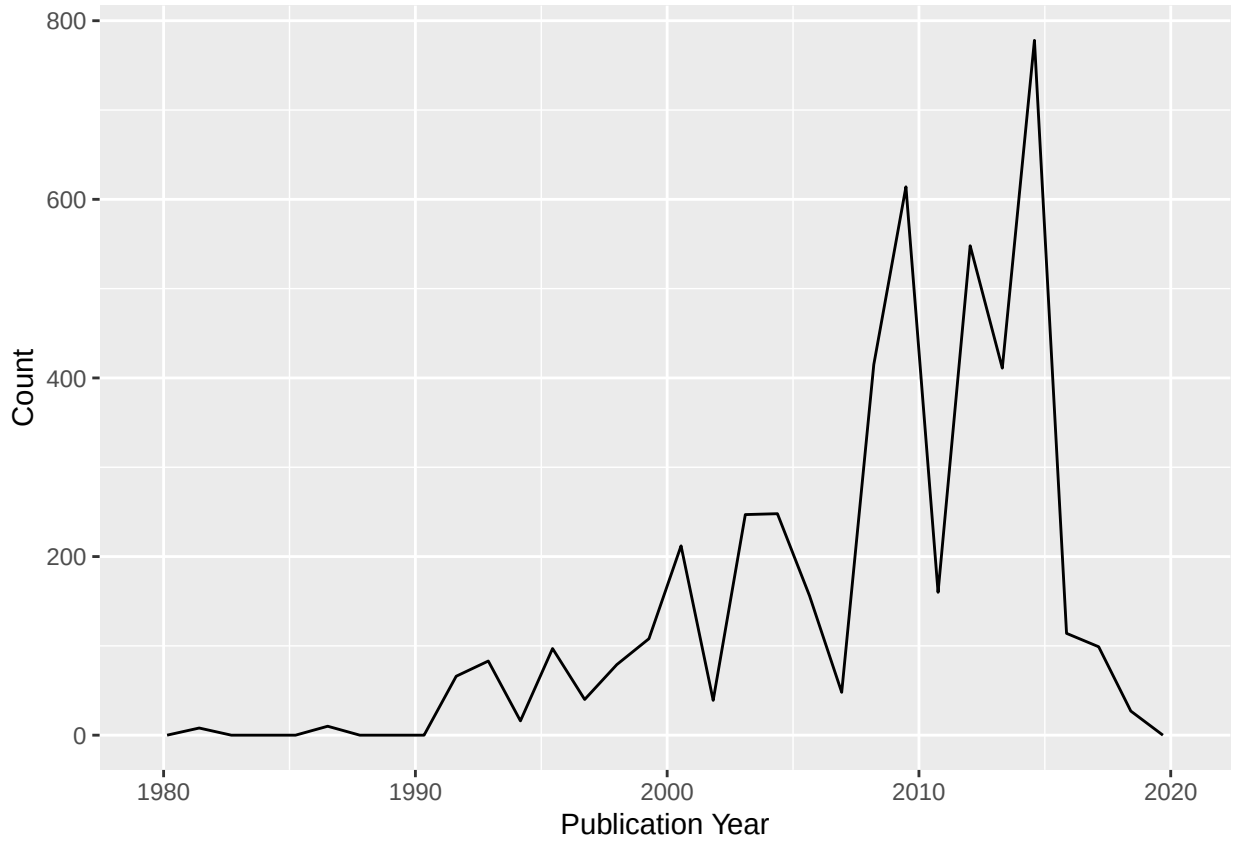
Answer: The class is factor. It is not numeric. Factors are categorical; because some of the data entries include "/" after the number, R reads this data as a factor rather than a number.

Explore your data graphically (Neonics)

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

```
ggplot(neonics, aes(Publication.Year)) + #Specify which data frame and which
  #column to use along the x-axis
        geom_freqpoly() + #Specify the type of plot to be used
  labs(x = "Publication Year", y = "Count") #Change axis lables
```

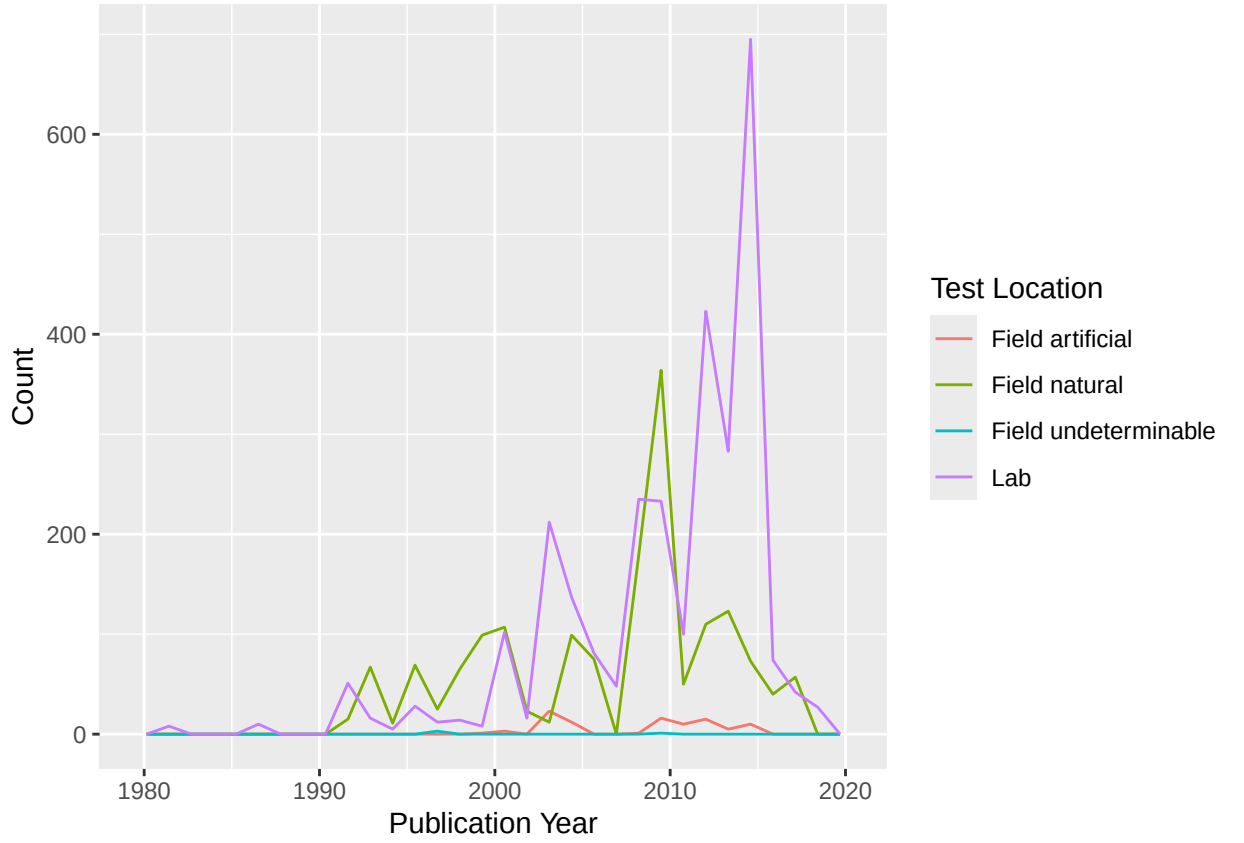
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

```
ggplot(neonics, aes(Publication.Year, color = Test.Location)) + #Specify which
  #data frame and which column to use along the x-axis and that color should
  #vary based on test location
        geom_freqpoly() + #Specify the type of plot to be used
  labs(x = "Publication Year", y = "Count", color = "Test Location") #Change the
```

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




```
#axis labels and legend title
```

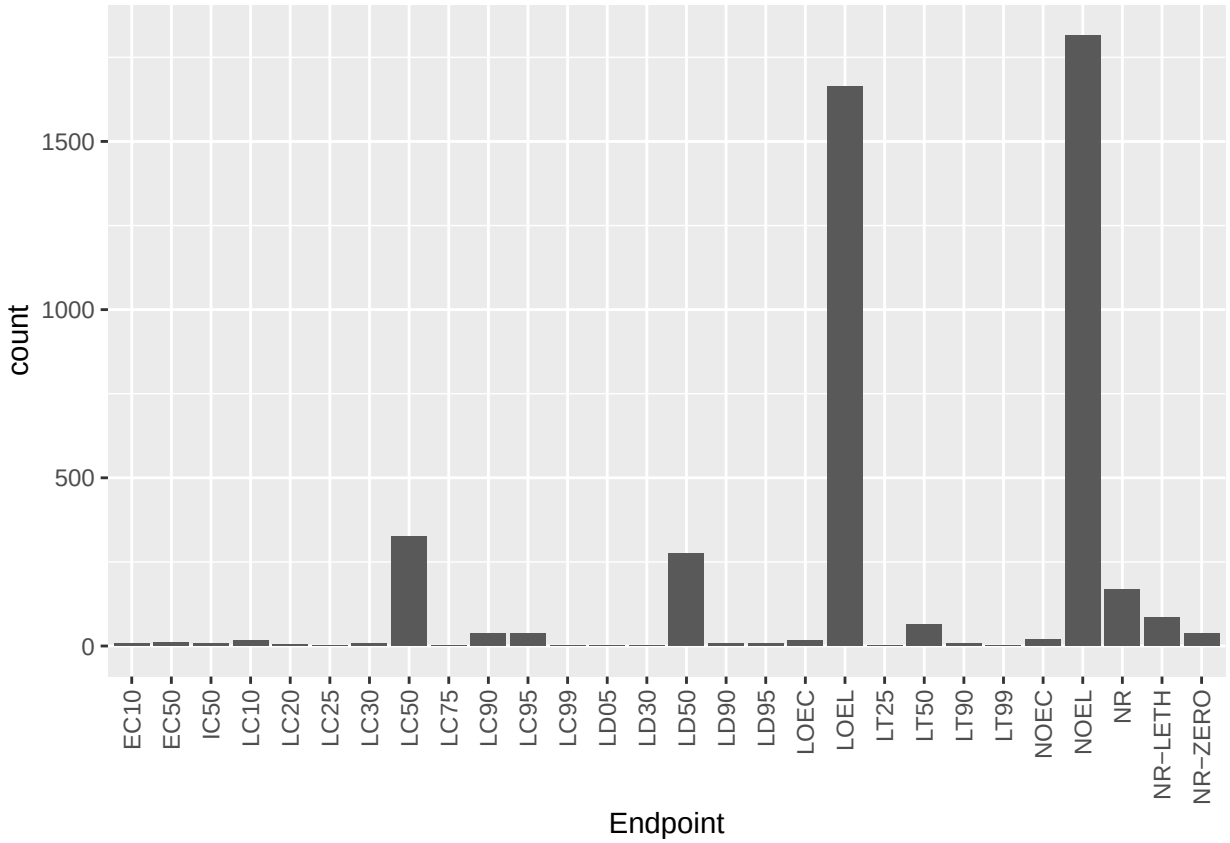
Interpret this graph. What are the most common test locations, and do they differ over time?

> Answer: The most common test locations are the lab and natural field. Natural field peaked right before 2010 and then dropped; lab peaked in about 2015 and then dropped drastically. Both natural field and lab were far less common before the 2000s.

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...]

```
ggplot(neonics, aes(Endpoint)) + #Specify which
  #data frame and which column to use along the x-axis.
  geom_bar() + #Specify we want to use a bar graph.
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) #This
```




```
#rotates and aligns and the x-axis labels so we can see them clearly.
```

Answer: The two most common endpoints are LOEL and NOEL. LOEL is defined as the 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 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.

```
#Use the `class` function to determine the class of collectDate.
class(litter$collectDate)
```

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

```
#The class is factor.

#Change the class of collectDate to date using the `as.Date` function.
litter$collectDate <- as.Date(litter$collectDate, format = "%y-%b-%d")

#Confirm the new class of the variable.
class(litter$collectDate)
```

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

```
#The class is now date.

#Use the `unique` function to determine which dates litter were sampled in
#August 2018.
unique(litter$collectDate)
```

```
## [1] NA
```

```
#Sampling occurred on August 2nd and August 30th.
```

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`?

```
#Use `unique` to determine how many different plots were sampled at Niwot Ridge.
unique(litter$namedLocation)
```

```
##  [1] NIWO_061.basePlot.ltr NIWO_064.basePlot.ltr NIWO_067.basePlot.ltr
##  [4] NIWO_040.basePlot.ltr NIWO_041.basePlot.ltr NIWO_063.basePlot.ltr
##  [7] NIWO_047.basePlot.ltr NIWO_051.basePlot.ltr NIWO_058.basePlot.ltr
## [10] NIWO_046.basePlot.ltr NIWO_062.basePlot.ltr NIWO_057.basePlot.ltr
## 12 Levels: NIWO_040.basePlot.ltr ... NIWO_067.basePlot.ltr
```

```
#There are 12 levels, so 12 different plots were sampled.

#Use `summary` to determine how many different plots were sampled at Niwot Ridge.
summary(litter$namedLocation)
```

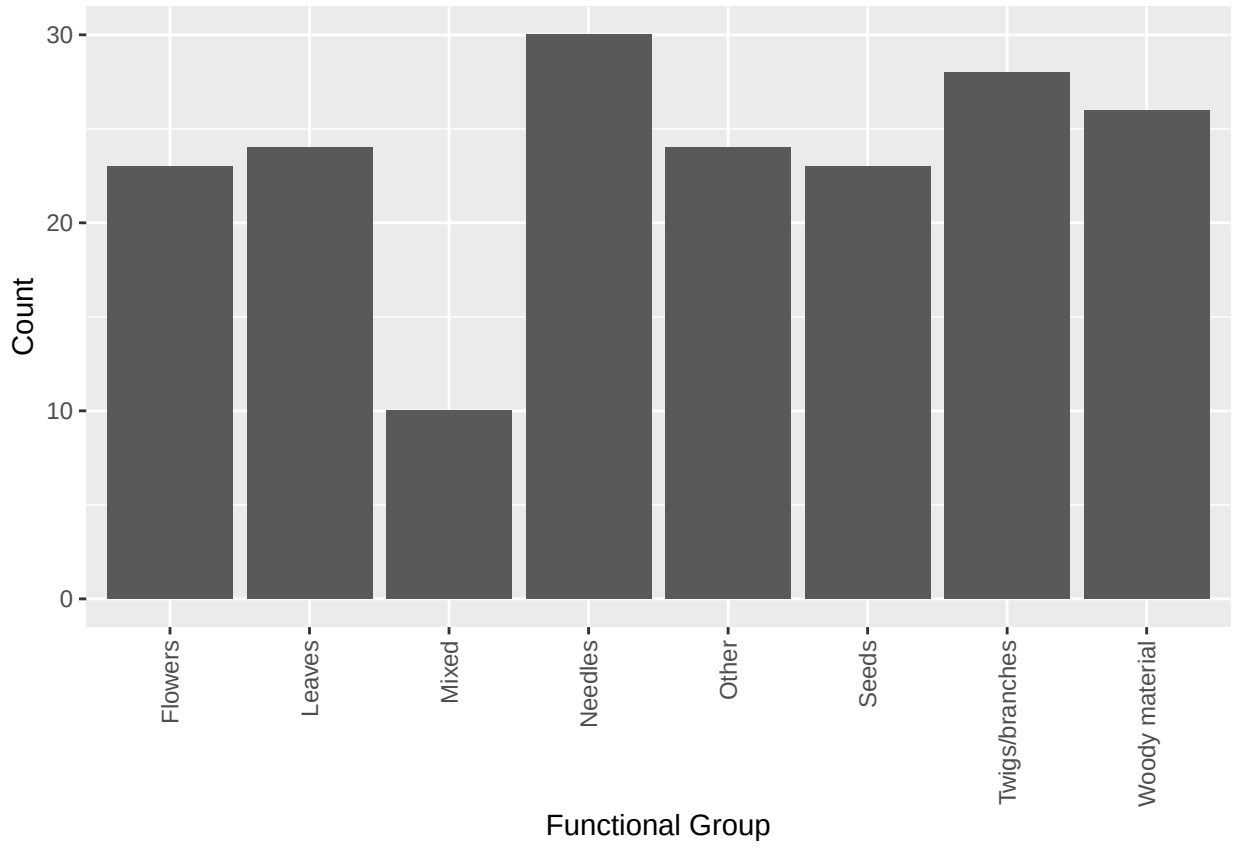
```
## NIWO_040.basePlot.ltr NIWO_041.basePlot.ltr NIWO_046.basePlot.ltr
##                    20                    19                    18
## NIWO_047.basePlot.ltr NIWO_051.basePlot.ltr NIWO_057.basePlot.ltr
##                    15                    14                     8
## NIWO_058.basePlot.ltr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

```
#This returned the sample counts at each plot.
```

Answer: `unique` returns the number of levels of our data (12). `summary` returns the number of counts of each level. We can still find the number of levels using summary by manually counting the number of categories in the output.

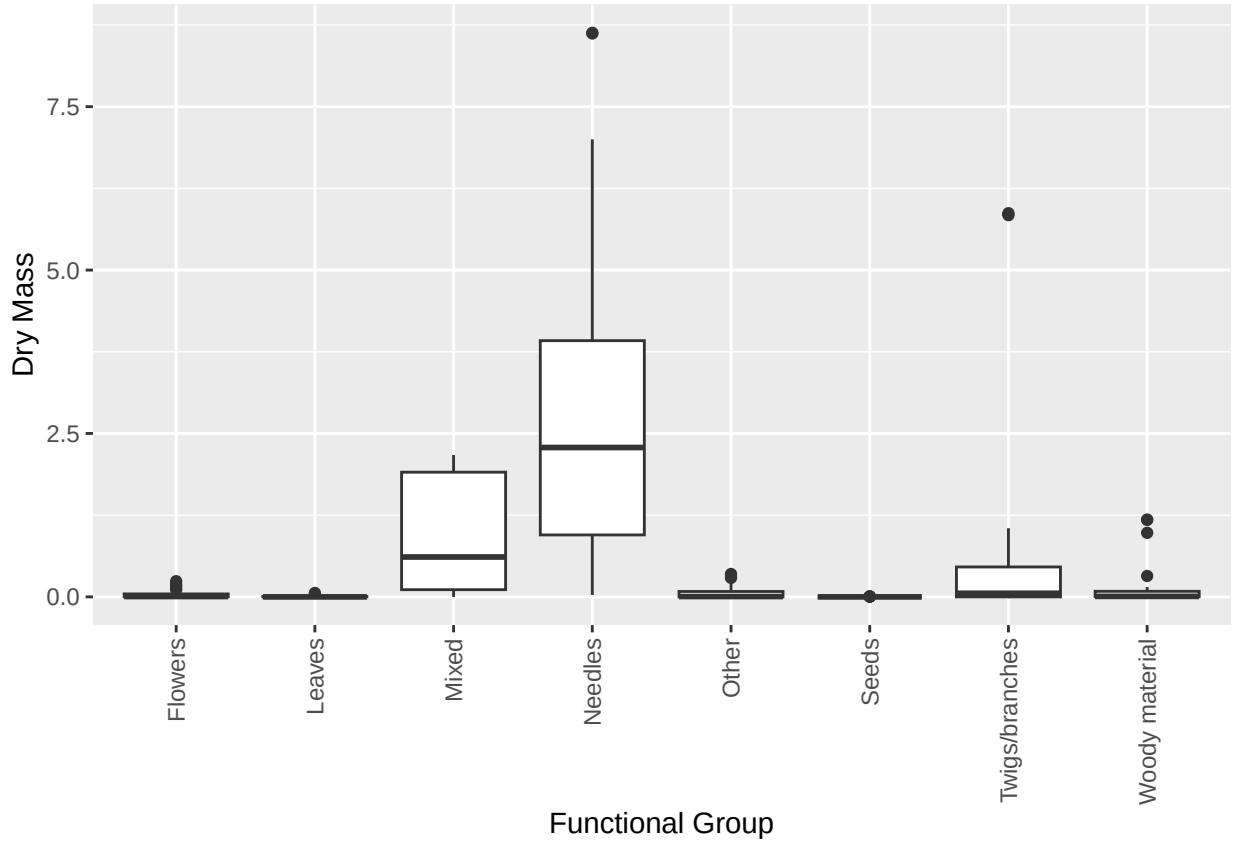
14. Create a bar graph of functionalGroup counts. This shows you what type of litter is collected at the Niwot Ridge sites. Notice that litter types are fairly equally distributed across the Niwot Ridge sites.

```
ggplot(litter, aes(functionalGroup)) + #Specify which
  #data frame and which column to use along the x-axis.
  geom_bar() + #Specify we want to use a bar graph.
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) + #This
#rotates and aligns and the x-axis labels so we can see them clearly.
  labs(x = "Functional Group", y = "Count") #Change axis labels.
```

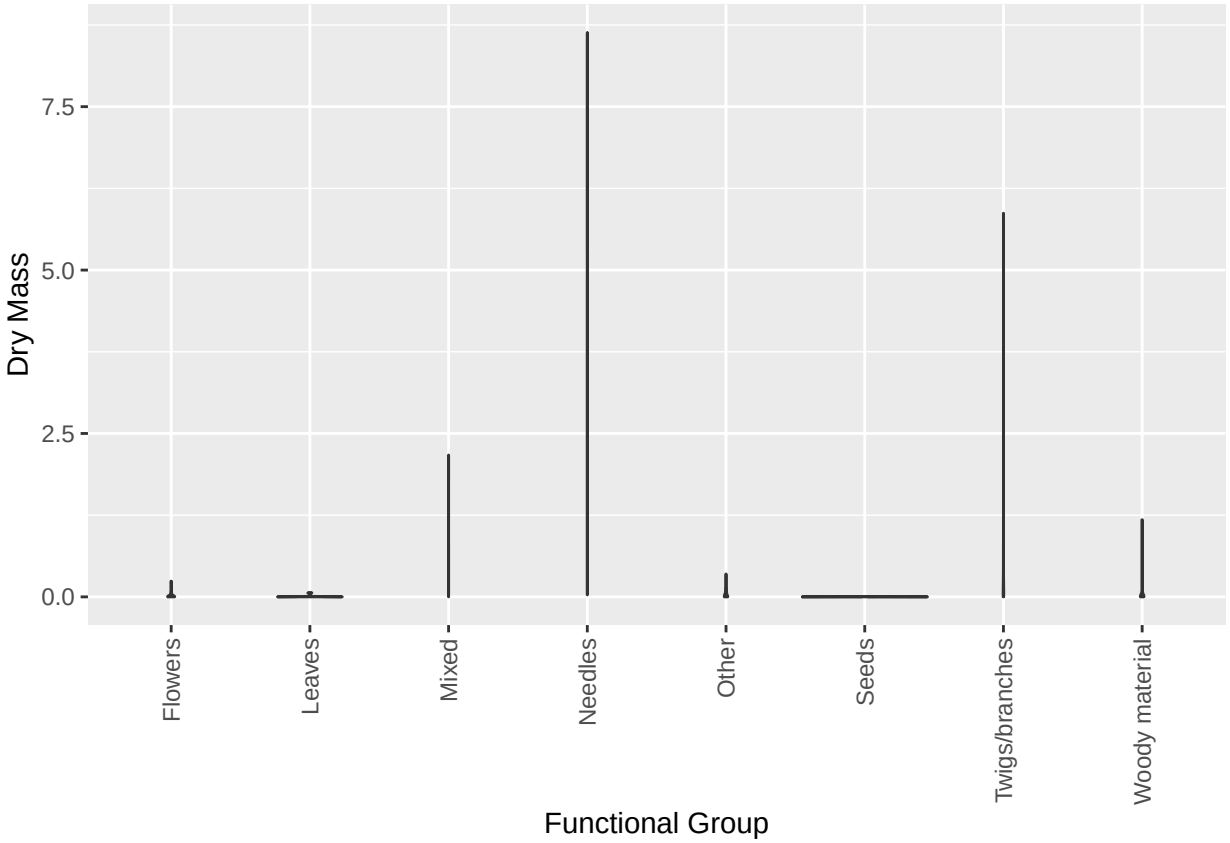


15. Using `geom_boxplot` and `geom_violin`, create a boxplot and a violin plot of dryMass by functionalGroup.

```
#Start with boxplot:
ggplot(litter) + #Specify which data frame to use.
  geom_boxplot(aes(x = functionalGroup, y = dryMass)) + #Specify we want to use
  #a boxplot with functional group on the x-axis and dry mass on the y-axis
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) + #This
#rotates and aligns and the x-axis labels so we can see them clearly.
  labs(x = "Functional Group", y = "Dry Mass") #Change axis labels.
```




```
#Now a violin plot:
ggplot(litter) + #Specify which data frame to use.
  geom_violin(aes(x = functionalGroup, y = dryMass)) + #Specify we want to use
  #a violin plot with functional group on the x-axis and dry mass on the y-axis
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) + #This
#rotates and aligns and the x-axis labels so we can see them clearly.
  labs(x = "Functional Group", y = "Dry Mass") #Change axis labels.
```



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

> Answer: The violin plot displays density distributions. However, the width #only changes with a continuous variable along the x-axis. Since functional group is categorical, the "violin" does not have a width.

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

> Answer: Needlles tend to have the highest biomass at these sites.