Assignment 3: Data Exploration

Kamil Burak Karayel

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.

```
library(tidyverse);library(lubridate);library(here)
getwd()
```

```
## [1] "/home/guest/EDA_Spring2025_kbk"
```

```
here()
```

```
## [1] "/home/guest/EDA_Spring2025_kbk"
```

```
Neonics <- read.csv(here('Data/Raw/ECOTOX_Neonicotinoids_Insects_raw.csv'), #Read csv file to Neonics d
  stringsAsFactors = TRUE)

Litter <- read.csv(here('Data/Raw/NEON_NIWO_Litter_massdata_2018-08_raw.csv'), #Read csv file to Litter
  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: Insecticides like neonicotinoids might have toxic effects on earth like other pesticides. These effects might include soil and water contaminations. It must be beneficial to measure and store these contamination levels. The policy makers or agricultural companies can benefit gathered data from different researches to decide usage levels or bans.

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: We can get information about carbon cycling or soil formation using these data. Forest litter and woody debris data can also be used for calculation fire spread risk.

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. Debris is collected by ground and elevated traps. Litter butt end diameter is defined as <2cm and <50cm and collected by elevated traps. Wood debris butt end diameter is defined as <2cm and >50cm and collected by ground traps. 2. The sampling site should consist of minimum 2m tall woody vegetation. The sampling area changes with the forest or low-statured vegetation. The plot centers should be at least 50m from large paved roads and plot edges should be 10m from dirt roads and there should be no water stream more than 1m wide. 3. The collected material is sorted into functional groups like leaves, needles etc. Each sample is weighed with 0.01g sensitivity level.

Obtain basic summaries of your data (Neonics)

5. What are the dimensions of the dataset?

```
dim(Neonics) #4623 rows and 30 columns
```

```
## [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? [Tip: The `sort()` command is useful for listing the values in order of magnitude...]

```
sort(summary(Neonics$Effect),decreasing = TRUE) #The most common effect is Population with 1803 observa
```

```
##       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
##     Accumulation      Intoxication     Biochemistry           Cell(s)
##               12                12                11                 9
##       Physiology         Histology       Hormone(s)
##                7                 5                 1
```

Answer: The "Effect" column should be consisted of observation groups which represents probable effects on insects. Each group is consisted of measurement effects which are the specific effects of the insecticides on the insects.

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...]

```
sort(summary(Neonics$Species.Common.Name,maxsum = 6),decreasing = TRUE) #6 most common species sorted i
```

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

Answer: The most common specie is "Other", the following five are all bee species. These species might be the most affected species by the insecticides. Hence the data might give idea about biodiversity, pollination and effects on plants.

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...]

```
class(Neonics$Conc.1..Author.) #The vector type of "Conc.1..Author" is factor.
```

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

```
#view(Neonics$Conc.1..Author.) #Insert comment not to fail the knit.
```

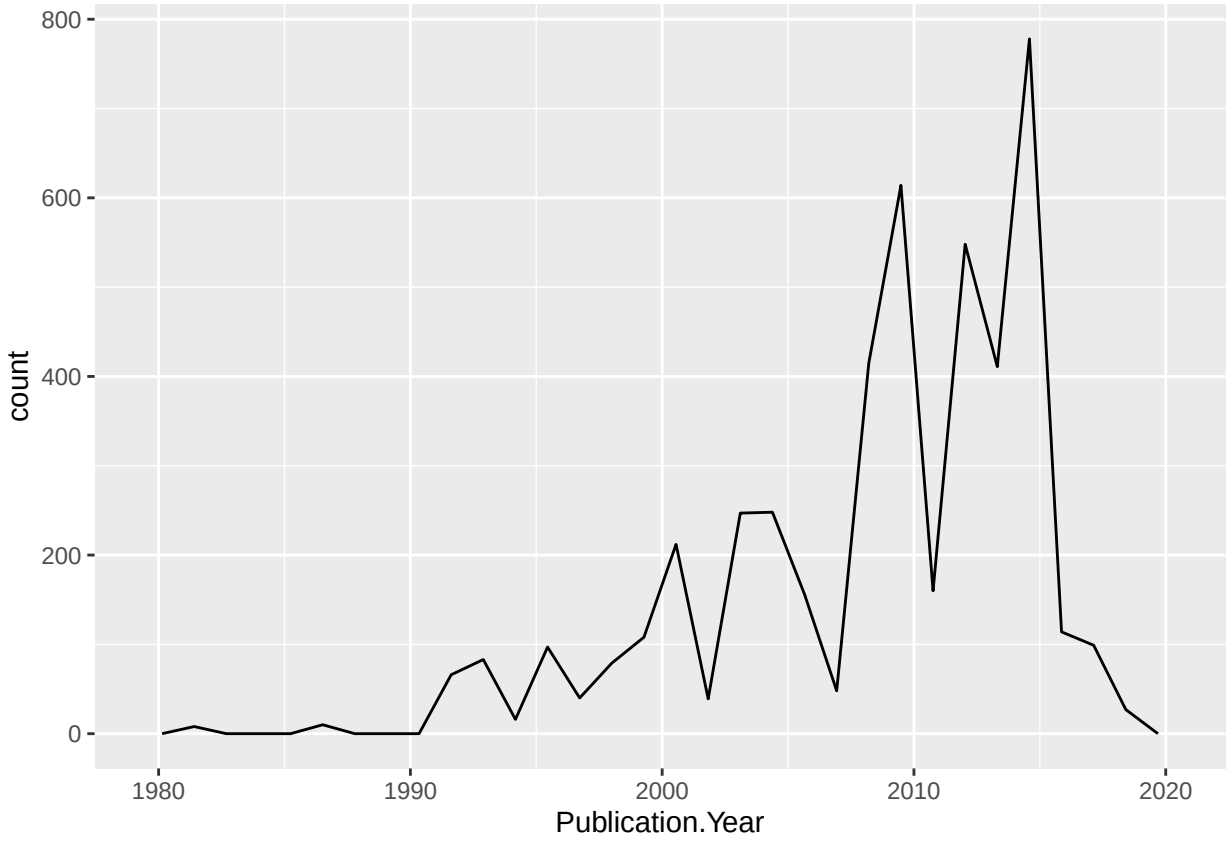
Answer: By viewing the column we can see that the vector includes both numeric and characters for different observations. So it can't be numeric. And stringsAsFactors command makes the vector type as factor.

Explore your data graphically (Neonics)

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

```
ggplot(Neonics,aes(x=Publication.Year)) + #Number of studies by Publication.Year
  geom_freqpoly()
```

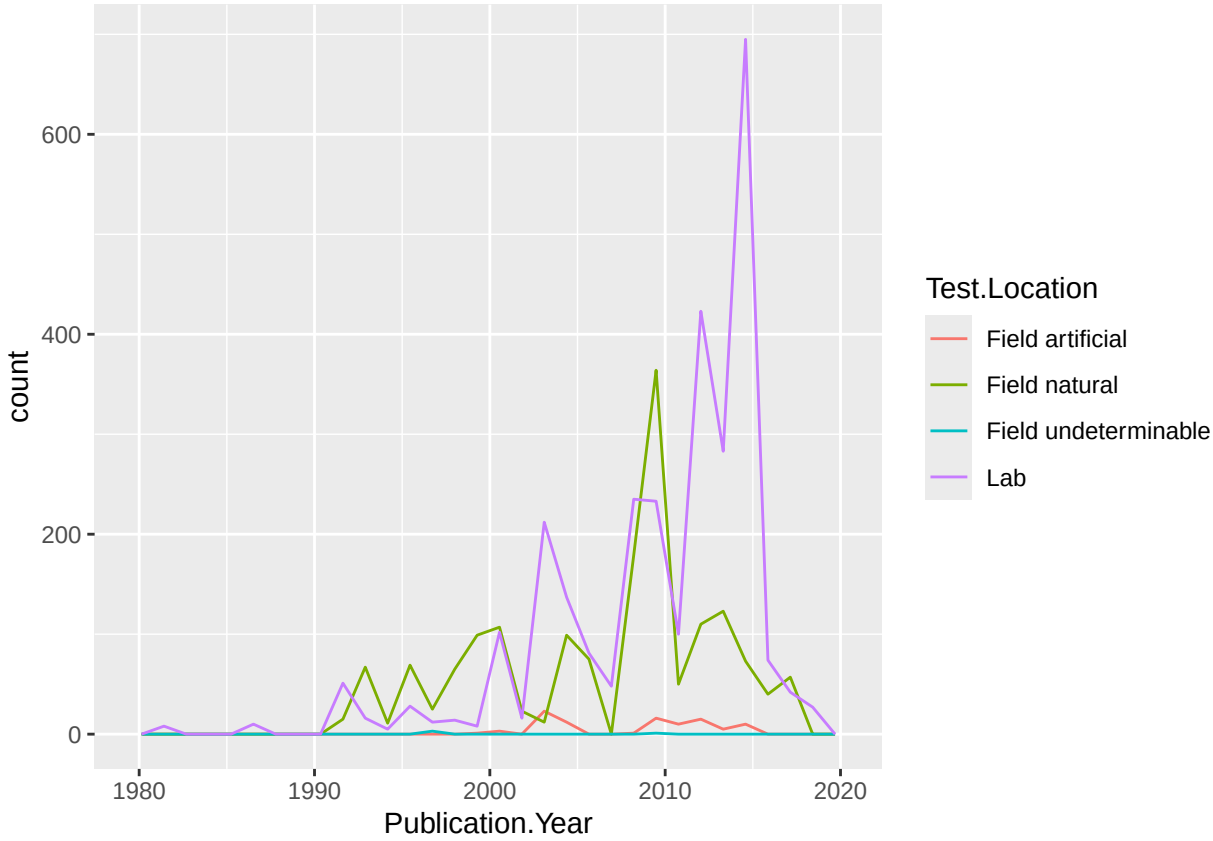
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

```
ggplot(Neonics,aes(x=Publication.Year,colour = Test.Location)) + #Test.Location data is added as color
  geom_freqpoly()
```

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




```
sort(summary(Neonics$Test.Location),decreasing = TRUE) #Added for next answer (most common test locatio
```

```
##                  Lab       Field natural    Field artificial
##                 2860                1663                  96
## Field undeterminable
##                    4
```

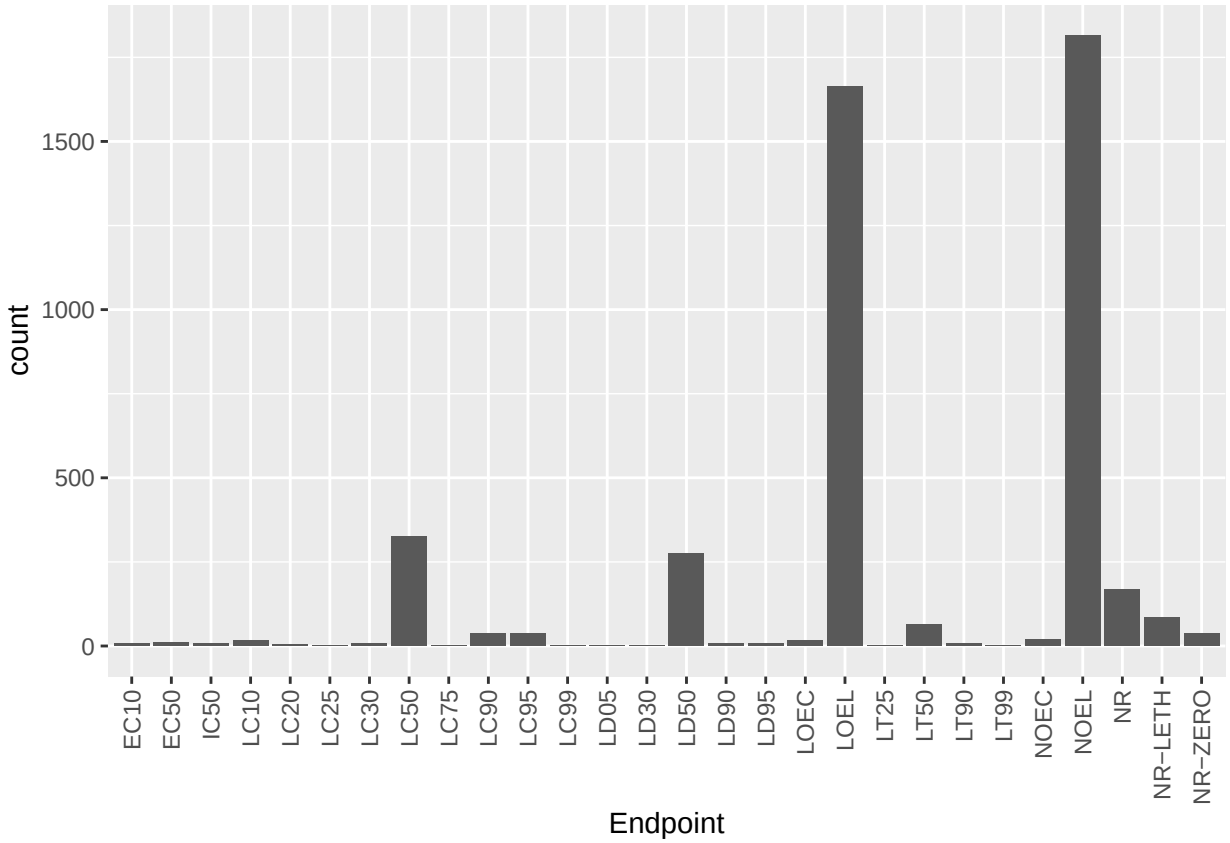
Interpret this graph. What are the most common test locations, and do they differ over time?

> Answer: By eyeballing we can say the most common test location is Lab. In some years Field natural studies exceed Lab studies. By the command above, the most common test location is Lab with 2860, Field natural is following by 1663.

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(x=Endpoint)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) #make x axis labels vertical. vju
```



Answer: LOEL and NOEL are the most common end points. LOEL is defined as "Lowest-observable-effect-level" and NOEL is defined as "No-observable-effect-level".

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) #Vector class is factor, not a date.
```

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

```
date_obj_collectDate <- ymd(Litter$collectDate) #Create new object by using ymd function on data from c
date_obj_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"
```

```
class(date_obj_collectDate) #Now the class of this new object is date.
```

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

```
Litter$collectDate <- date_obj_collectDate #change the content of collectDate with new vector which has
unique(Litter$collectDate) #The litter is sampled on 2nd 30th of August.
```

```
## [1] "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`?

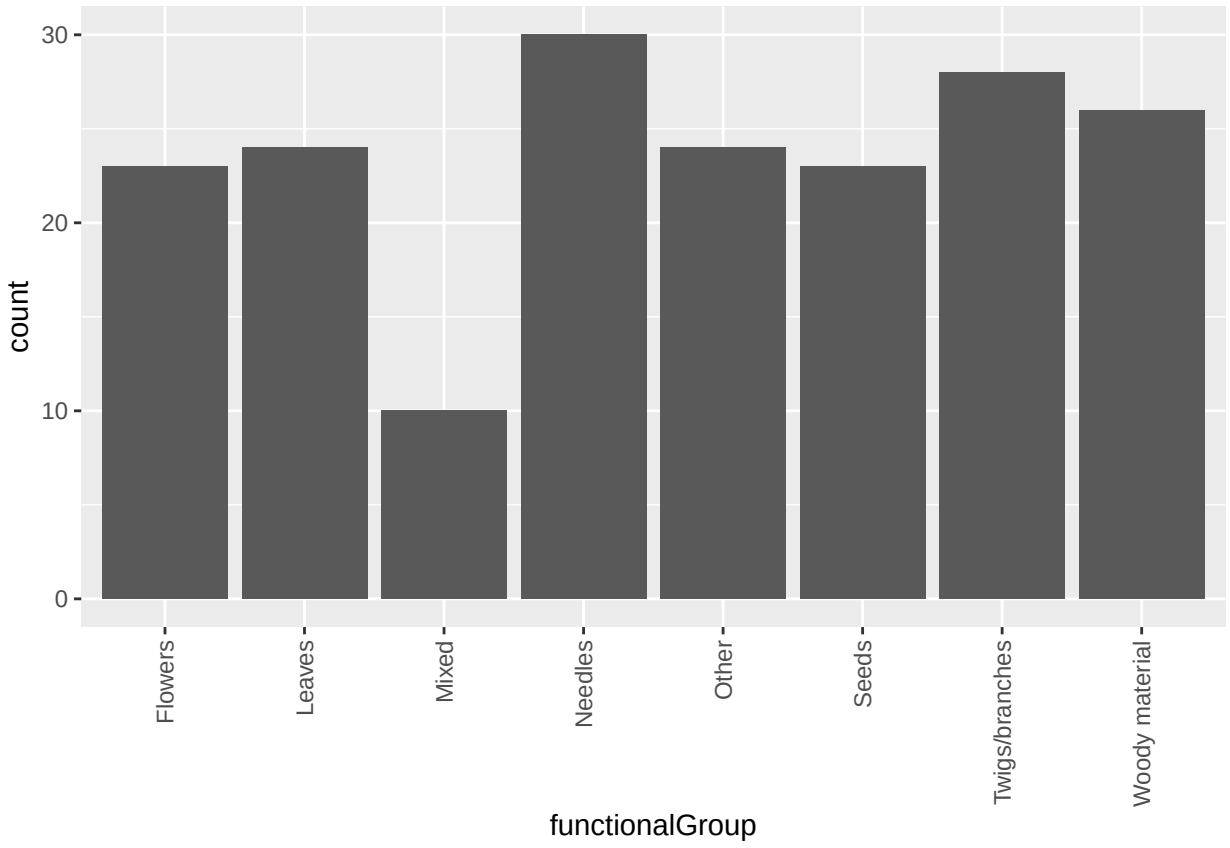
```
unique(Litter$plotID) #There are 12 levels which means there are 12 different observations in plotID.
```

```
##  [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
```

Answer: 'Unique' function gives the distinct values and the number of distinct values in the vector. But 'summary' function gives the number of occurence of every unique value.

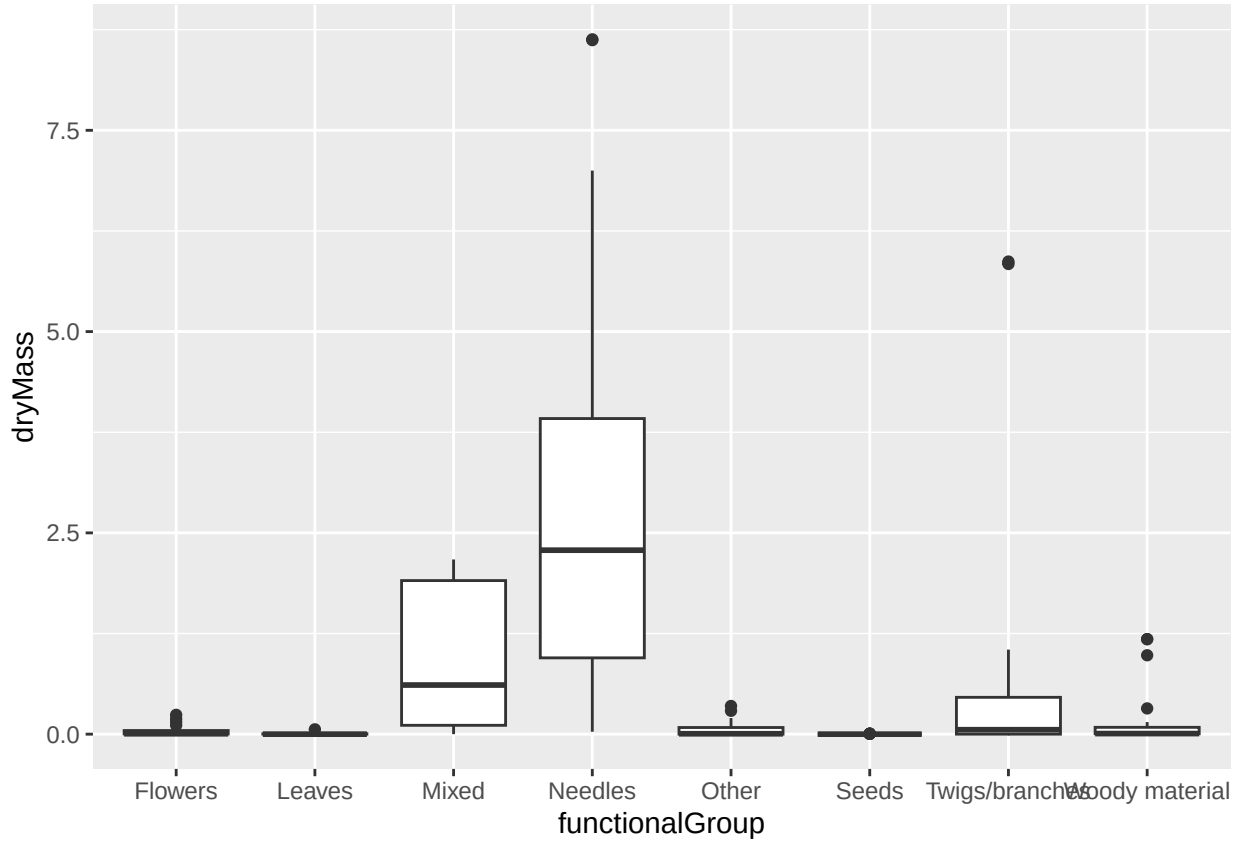
14. Create a bar graph of functionalGroup counts. This shows you what type of litter is collected at the Niwot Ridge sites. Notice that litter types are fairly equally distributed across the Niwot Ridge sites.

```
ggplot(Litter,aes(x=functionalGroup)) + #The most collected litter is 'Needless' and second is 'Twigs/b
  geom_bar() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) #Rotate and align X-
```

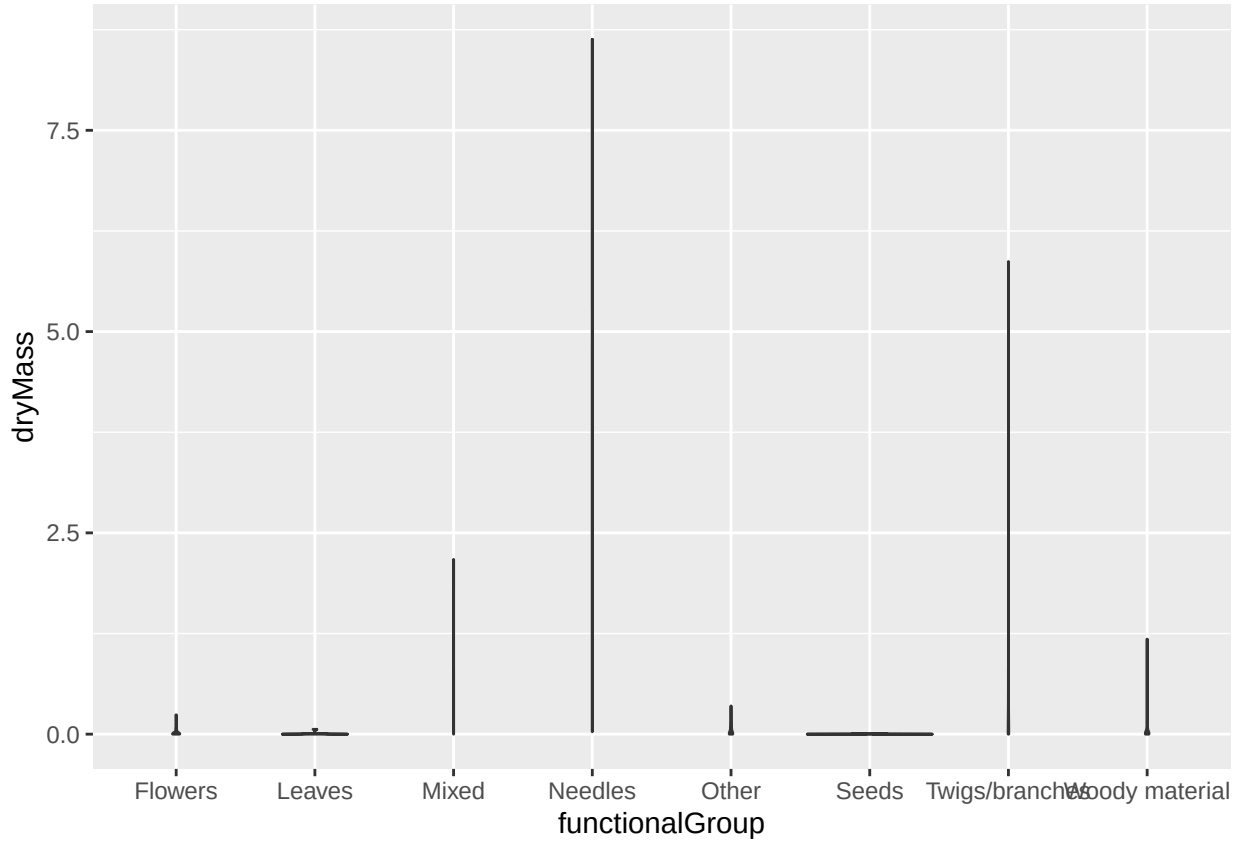


15. Using `geom_boxplot` and `geom_violin`, create a boxplot and a violin plot of dryMass by functionalGroup.

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




```
ggplot(Litter, aes(x = functionalGroup, y = dryMass)) +
  geom_violin()
```




```
sum(subset(Litter, functionalGroup == "Needles")$dryMass) #added for last question.
```

```
## [1] 81.15
```

```
sum(subset(Litter, functionalGroup == "Twigs/branches")$dryMass) #added for last question
```

```
## [1] 17.445
```

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

> Answer: By the boxplot, it is easier to understand the basic statistical value like median and quartiles or outliers. Because of limited data points violin plot can't give distribution.

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

> Answer: Needles is the highest drymass in the samples and Twigs/branches follows it. Drymass of "Needles" is 81.15 and it is 17.445 for Twigs/branches. We can see from the codes above line 182 and 183.