Assignment 3: Data Exploration

Keanu Valibia

Spring 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 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.

```
library(tidyverse) # Load tidyverse package
library(lubridate) # Load lubridate package

getwd() # Retrieve current working directory
```

```
## [1] "/home/guest/R/R Projects/EDA_Spring2024"
```

```r
# Create Neonics object using ecotox .csv
Neonics <- read.csv("~/R/R Projects/EDA_Spring2024/Data/Raw/ECOTOX_Neonicotinoids_Insects_raw.csv",
  stringsAsFactors = TRUE)

#Create Litter object using NEON .csv
Litter <- read.csv("~/R/R Projects/EDA_Spring2024/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: We might be interested in understanding the specific chemical effects on insects, which may have drastic effects on their populations and surrounding environments. Chemical use may also have unpredictable effects, including changes in behavior or lifespan.

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: Forest litter and debris is a source of energy for forests and aquatic systems. It also provides shelter for smaller animals / insects.

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. (Spatial sampling) Sampling is executed at terrestrial NEON sites that contain woody vegetatioon less than 2 meters tall. 2. (Spatial sampling) In sites with forested tower airsheds, the litter sampling is targeted to take place in 20 40m x 40m plots 3. (Temporal sampling) Frequent sampling (1x every 2 weeks) in deciduous forest sites & infrequent sampling (1x every 1-2 months) at evergreen sites

Obtain basic summaries of your data (Neonics)

5. What are the dimensions of the dataset?

```r
# Use dim() function to retrieve dimensions of Neonics dataset
dim(Neonics)
```

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

   Answer: 4,623 Rows and 30 Fields/Columns. These same dimensions can be viewed in the "Environment" panel.

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?

```
# Create vector object containing summary() of Effect field from Neonics
neonics.effect.summary <- summary(Neonics$Effect)
# Call newly created object
neonics.effect.summary
```

```
##     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
```

```
# Sort summary by descending order and print results
neonics.effect.summary[order(neonics.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
##     Accumulation     Intoxication     Biochemistry          Cell(s)
##               12               12               11                9
##       Physiology        Histology        Hormone(s)
##                7                5                1
```

Answer: The most common effects are Population and Mortality. We are interested in these effects to see: 1. Population-level effects as a result of chemical use and 2. Effects of insect mortality due to chemical use

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…]

```
# Create vector object containing summary() of Species.Common.Name field from Neonics
neonics.species.summary <- sort(summary(Neonics$Species.Common.Name), decreasing = TRUE)
# Call newly created object
neonics.species.summary
```

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

```
##             Carniolan Honey Bee                      Bumble Bee
##                             152                             140
##               Italian Honeybee                 Japanese Beetle
##                             113                              94
##               Asian Lady Beetle                  Euonymus Scale
##                              76                              75
##                        Wireworm               European Dark Bee
##                              69                              66
##               Minute Pirate Bug            Asian Citrus Psyllid
##                              62                              60
##                   Parastic Wasp          Colorado Potato Beetle
##                              58                              57
##                 Parasitoid Wasp              Erythrina Gall Wasp
##                              51                              49
##                    Beetle Order     Snout Beetle Family, Weevil
##                              47                              47
##        Sevenspotted Lady Beetle                  True Bug Order
##                              46                              45
##           Buff-tailed Bumblebee                    Aphid Family
##                              39                              38
##                  Cabbage Looper            Sweetpotato Whitefly
##                              38                              37
##                   Braconid Wasp                    Cotton Aphid
##                              33                              33
##                  Predatory Mite           Ladybird Beetle Family
##                              33                              30
##                      Parasitoid                   Scarab Beetle
##                              30                              29
##                   Spring Tiphia                     Thrip Order
##                              29                              29
##            Ground Beetle Family              Rove Beetle Family
##                              27                              27
##                   Tobacco Aphid                    Chalcid Wasp
##                              27                              25
##         Convergent Lady Beetle                   Stingless Bee
##                              25                              25
##               Spider/Mite Class             Tobacco Flea Beetle
##                              24                              24
##                Citrus Leafminer                 Ladybird Beetle
##                              23                              23
##                       Mason Bee                        Mosquito
##                              22                              22
##                   Argentine Ant                          Beetle
##                              21                              21
##       Flatheaded Appletree Borer             Horned Oak Gall Wasp
##                              20                              20
##              Leaf Beetle Family               Potato Leafhopper
##                              20                              20
##      Tooth-necked Fungus Beetle                    Codling Moth
##                              20                              19
##        Black-spotted Lady Beetle                    Calico Scale
##                              18                              18
##            Fairyfly Parasitoid                     Lady Beetle
##                              18                              18
```

```
##             Minute Parasitic Wasps                              Mirid Bug 
##                                 18                                     18 
##                   Mulberry Pyralid                               Silkworm 
##                                 18                                     18 
##                     Vedalia Beetle                 Araneoid Spider Order 
##                                 18                                     17 
##                          Bee Order                       Egg Parasitoid 
##                                 17                                     17 
##                       Insect Class            Moth And Butterfly Order 
##                                 17                                     17 
##       Oystershell Scale Parasitoid Hemlock Woolly Adelgid Lady Beetle 
##                                 17                                     16 
##             Hemlock Wooly Adelgid                                   Mite 
##                                 16                                     16 
##                        Onion Thrip                Western Flower Thrips 
##                                 16                                     15 
##                       Corn Earworm                    Green Peach Aphid 
##                                 14                                     14 
##                          House Fly                             Ox Beetle 
##                                 14                                     14 
##                 Red Scale Parasite                   Spined Soldier Bug 
##                                 14                                     14 
##              Armoured Scale Family                      Diamondback Moth 
##                                 13                                     13 
##                      Eulophid Wasp                    Monarch Butterfly 
##                                 13                                     13 
##                      Predatory Bug               Yellow Fever Mosquito 
##                                 13                                     13 
##                Braconid Parasitoid                          Common Thrip 
##                                 12                                     12 
##       Eastern Subterranean Termite                                Jassid 
##                                 12                                     12 
##                         Mite Order                             Pea Aphid 
##                                 12                                     12 
##                   Pond Wolf Spider             Spotless Ladybird Beetle 
##                                 12                                     11 
##            Glasshouse Potato Wasp                              Lacewing 
##                                 10                                     10 
##            Southern House Mosquito             Two Spotted Lady Beetle 
##                                 10                                     10 
##                         Ant Family                          Apple Maggot 
##                                  9                                      9
```

Answer: Excluding (Other), the top six include: 1. Honeybee 2. Parasitic Wasp 3. Buff Tailed Bumblebee 4. Carniolan Honey Bee 5. Bumble Bee 6. Italian Honeybee We may be interested particularly in bees due to their prominent role in pollination.

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?

```r
# Check field class using class() function
class(Neonics$Conc.1..Author.)
```

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

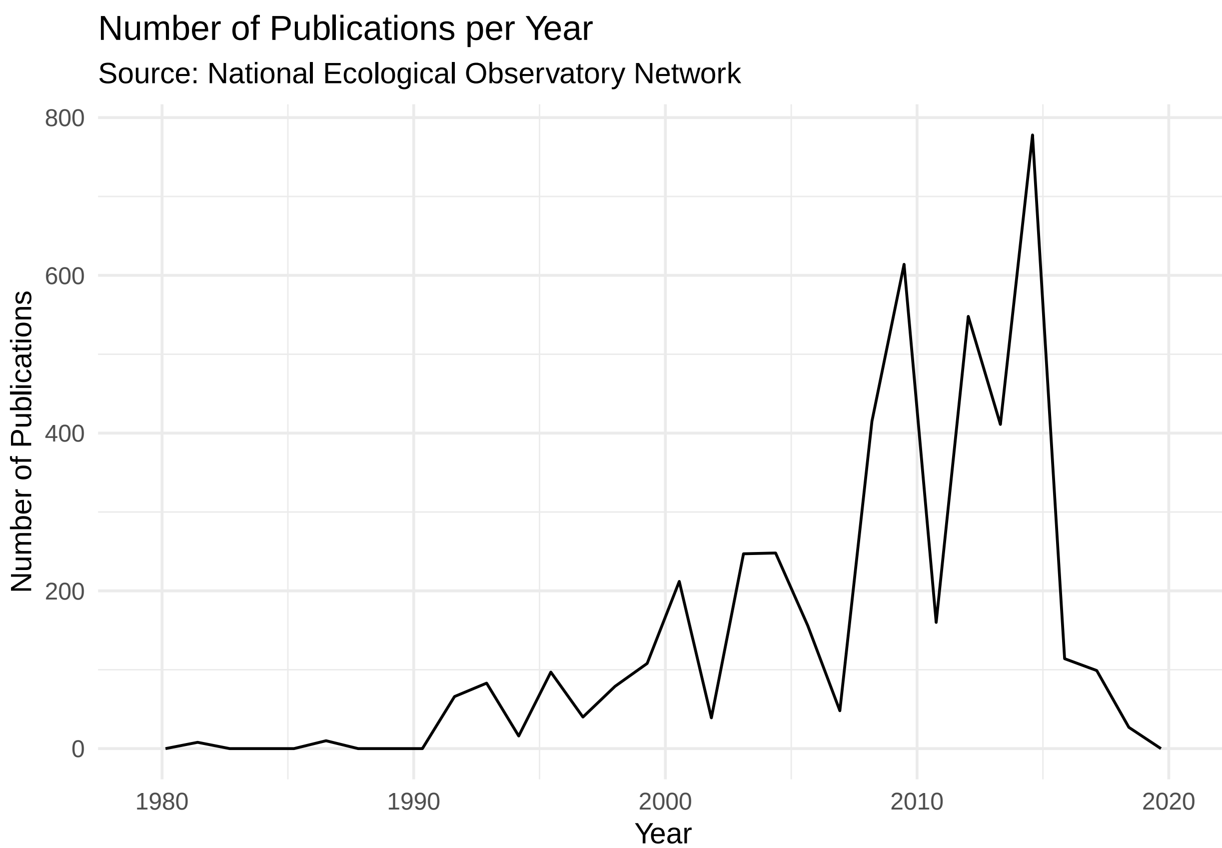
Answer: It is a factor. It is not numeric because it includes non-numeric values. Thus RStudio defaults this field class to the most inclusive class.

Explore your data graphically (Neonics)

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

```
# Create new object to save plot
freqPlot.publications1 <- ggplot(Neonics, aes(x = Publication.Year)) +
  # Frequency plot type
  geom_freqpoly() +
  # Change theme
  theme_minimal() +
  # Create title and subtitle
  ggtitle("Number of Publications per Year",
          "Source: National Ecological Observatory Network") +
  # Change label names
  labs(x = "Year", y = "Number of Publications")

# Call frequency plot
freqPlot.publications1
```

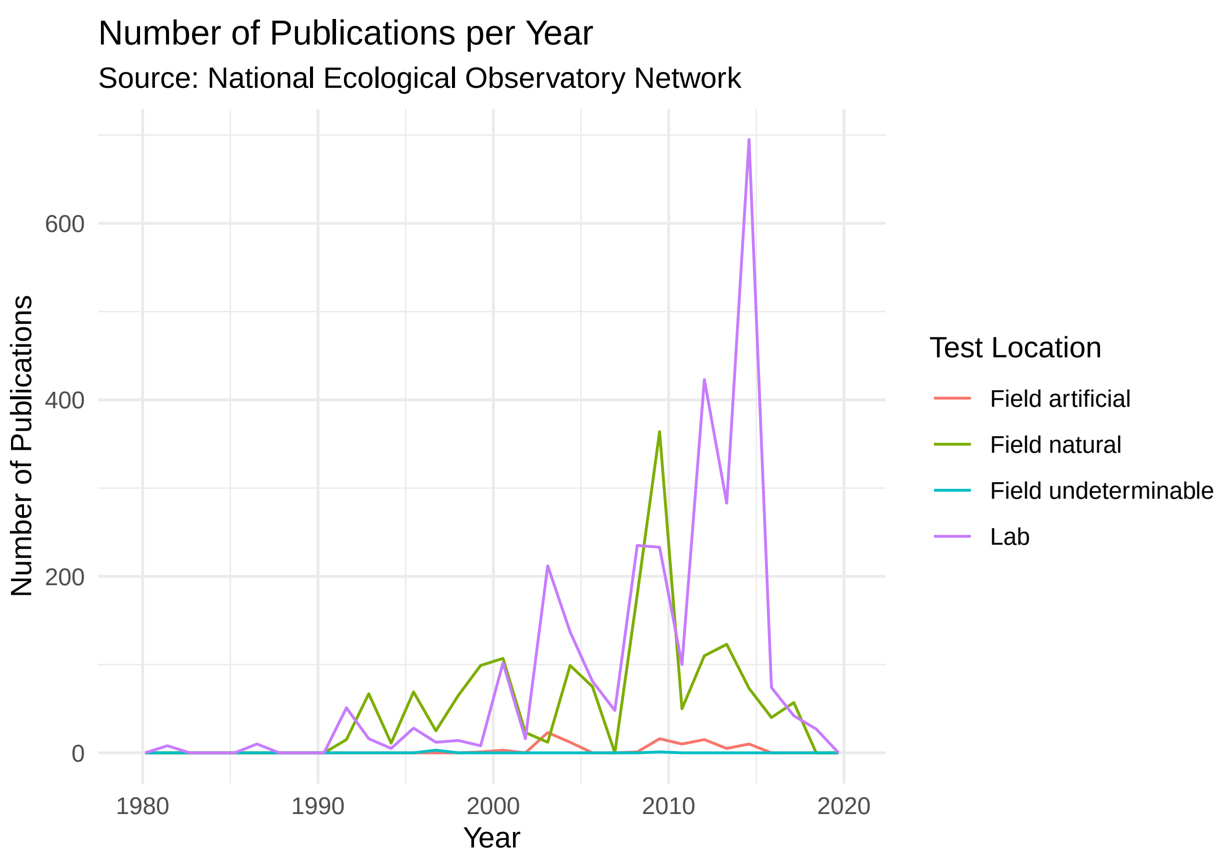
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

```
freqPlot.publications2 <- ggplot(Neonics, aes(x = Publication.Year, color = Test.Location)) +
  # Frequency plot type
  geom_freqpoly() +
  # Change plot theme
  theme_minimal() +
  # Set title and subtitle
  ggtitle("Number of Publications per Year",
          "Source: National Ecological Observatory Network") +
  # Set x- and y-axis lables. Change legend title.
  labs(x = "Year", y = "Number of Publications", color = "Test Location")

# Call frequency plot
freqPlot.publications2
```

```
## `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?

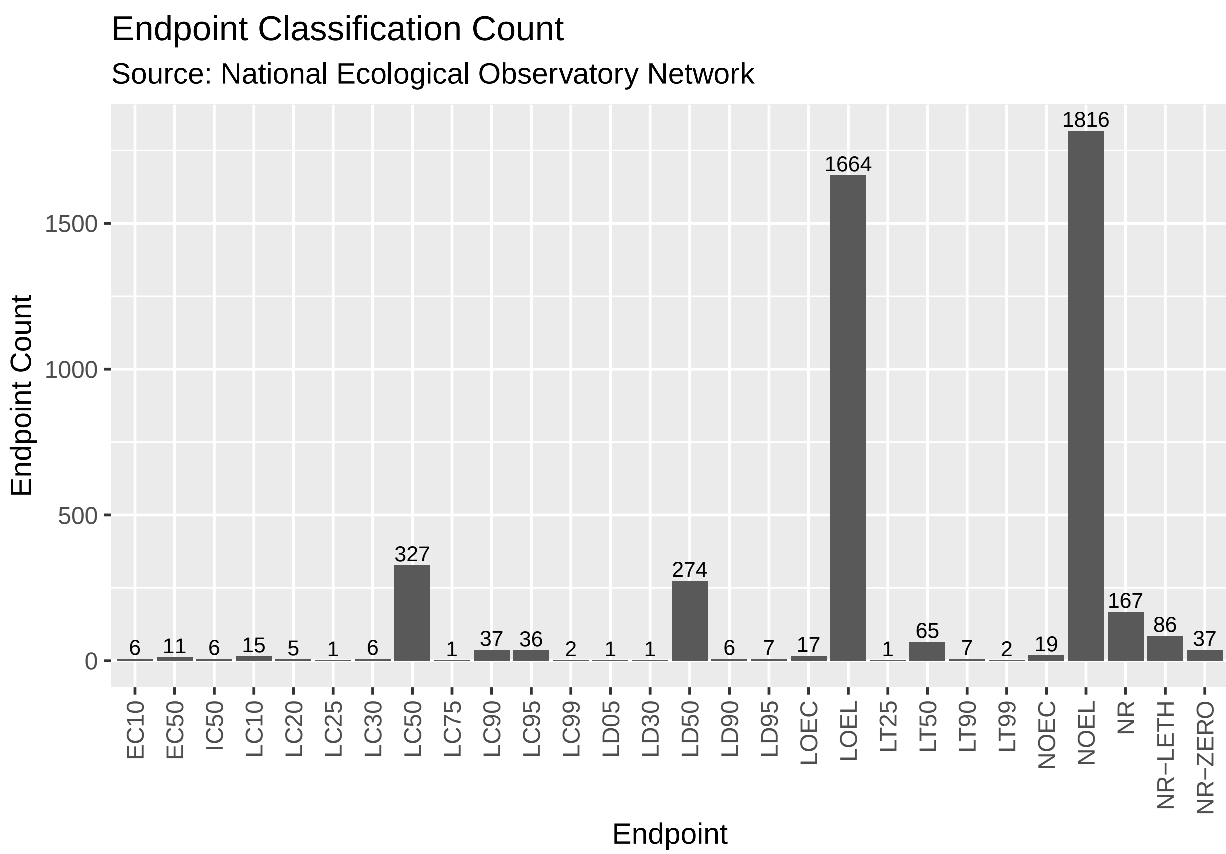
> Answer: The most common test locations are (in descending order): 1. "Lab" 2. "Field Natural" 3. "Field Artificial" 4. "Undeterminable" As time increases, the number of lab locations increases dramatically. Natural fields spike, then decrease. Artificial labs are stable, as are undeterminable fields.

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…]

```r
# Create and save plot
barGraph.endpoints <- ggplot(Neonics, aes(x = Endpoint))+
  # Set bar chart type
  geom_bar() +
  # Set title and subtitle
  ggtitle("Endpoint Classification Count",
          "Source: National Ecological Observatory Network") +
  # Change x- and y-axis labels
  labs(x = "Endpoint", y = "Endpoint Count") +
  # Adjust plot text to make more legible
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  # Add bar counts to top of individual bars
  geom_text(size = 2.8, aes(label=..count..),
            stat='count',
            position=position_dodge(0.5),
            vjust=-0.27)

# Call bar graph
barGraph.endpoints
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Answer: The two most common endpoints are: 1. L0EL (Lowest-observable-effect-level): lowest dose (concentration) producing effects that were significantly different (as reported by authors) from responses of controls 2. NOEL (No-observable-effect-level): highest dose (concentration) producing effects not significantly different from responses of controls according to author's reported statistical test

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"
```

```
Litter$collectDate <- as.Date(Litter$collectDate, "%Y-%m-%d")
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"
```

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

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

```r
unique(Litter$collectDate)
```

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

Answer: Litter was sampled on two days: 08/02/18 and 08/30/18

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`?

```r
# Retrieve unique values in plot ID field
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
```

```r
# Retrieve summary stats of plot ID field
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
```

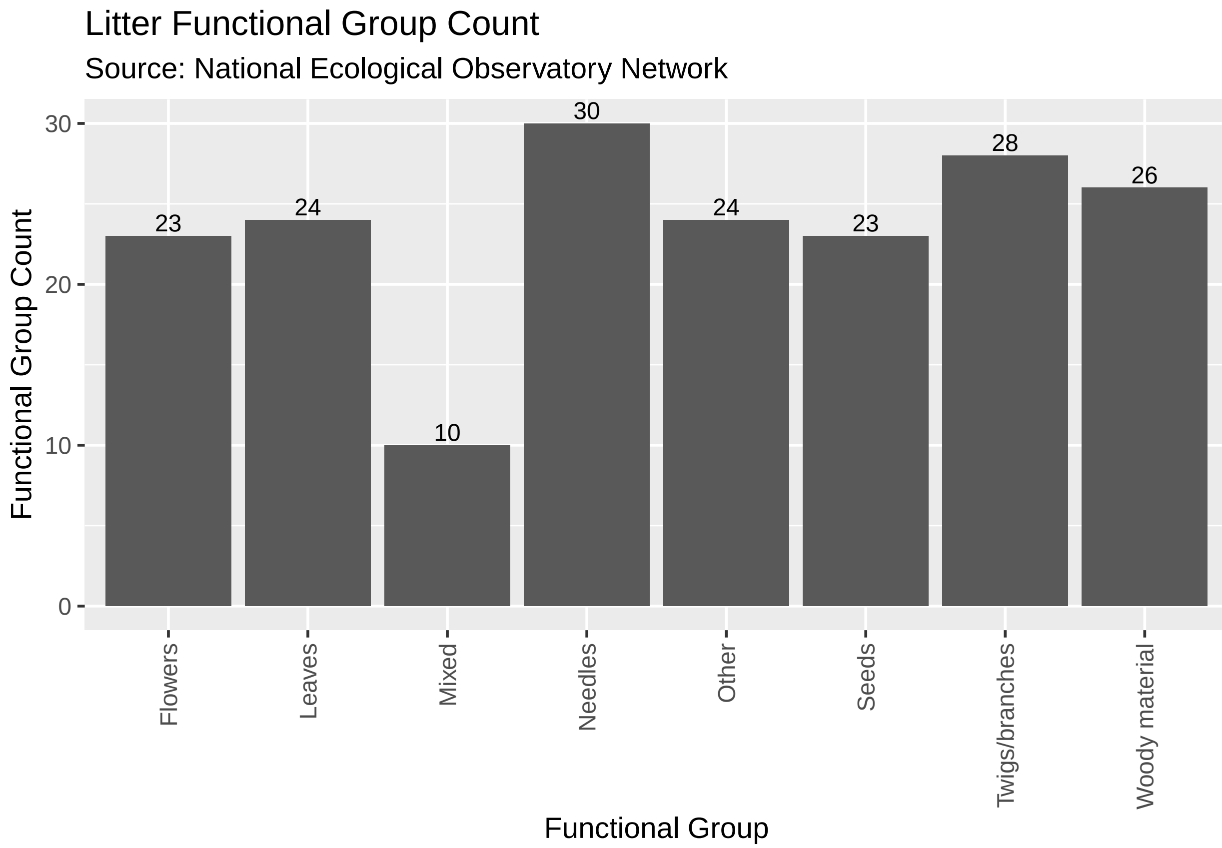
```r
# sum number of unique values in plot ID field
sum(summary(unique(Litter$plotID)))
```

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

> Answer: 12 plots were sampled at Niwot Ridge. The information obtained from unique() returns the specific plot IDs, without number of occurrences. The summary() function returns each plot ID as well as the number of occurrences.

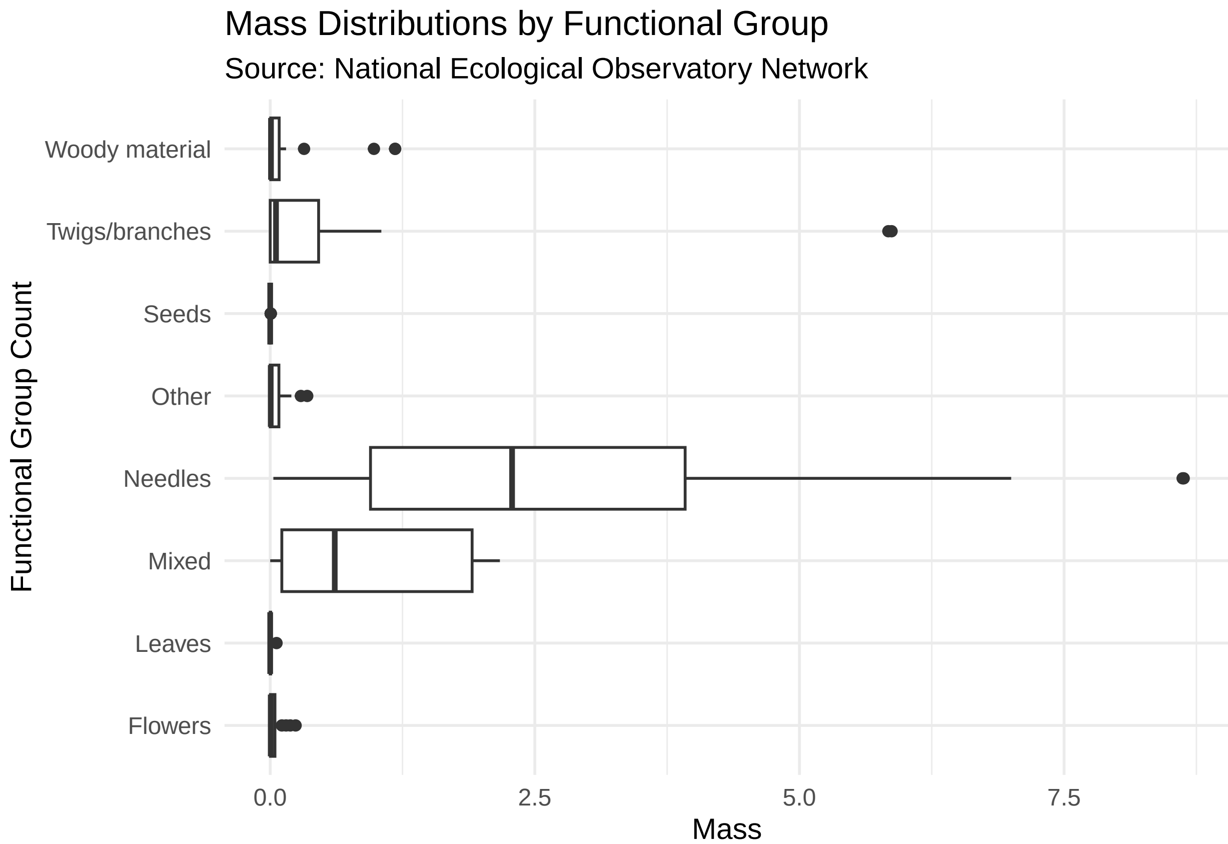
14. Create a bar graph of functionalGroup counts. This shows you what type of litter is collected at the Niwot Ridge sites. Notice that litter types are fairly equally distributed across the Niwot Ridge sites.

```r
# Create and save bar graph
barGraph.functionalGroup <- ggplot(Litter, aes(x = functionalGroup)) +
  # Set geometry bar graph type
  geom_bar() +
  # Set title and subtitle
  ggtitle("Litter Functional Group Count",
          "Source: National Ecological Observatory Network") +
  # Set x- and y-axis labels
  labs(x = "Functional Group", y = "Functional Group Count") +
  # Adjust label orientation
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  # Add bar counts, and adjust label position
  geom_text(size = 3, aes(label=..count..),
            stat='count',
            position=position_dodge(0.5),
            vjust=-0.27)

# Call bar graph
barGraph.functionalGroup
```

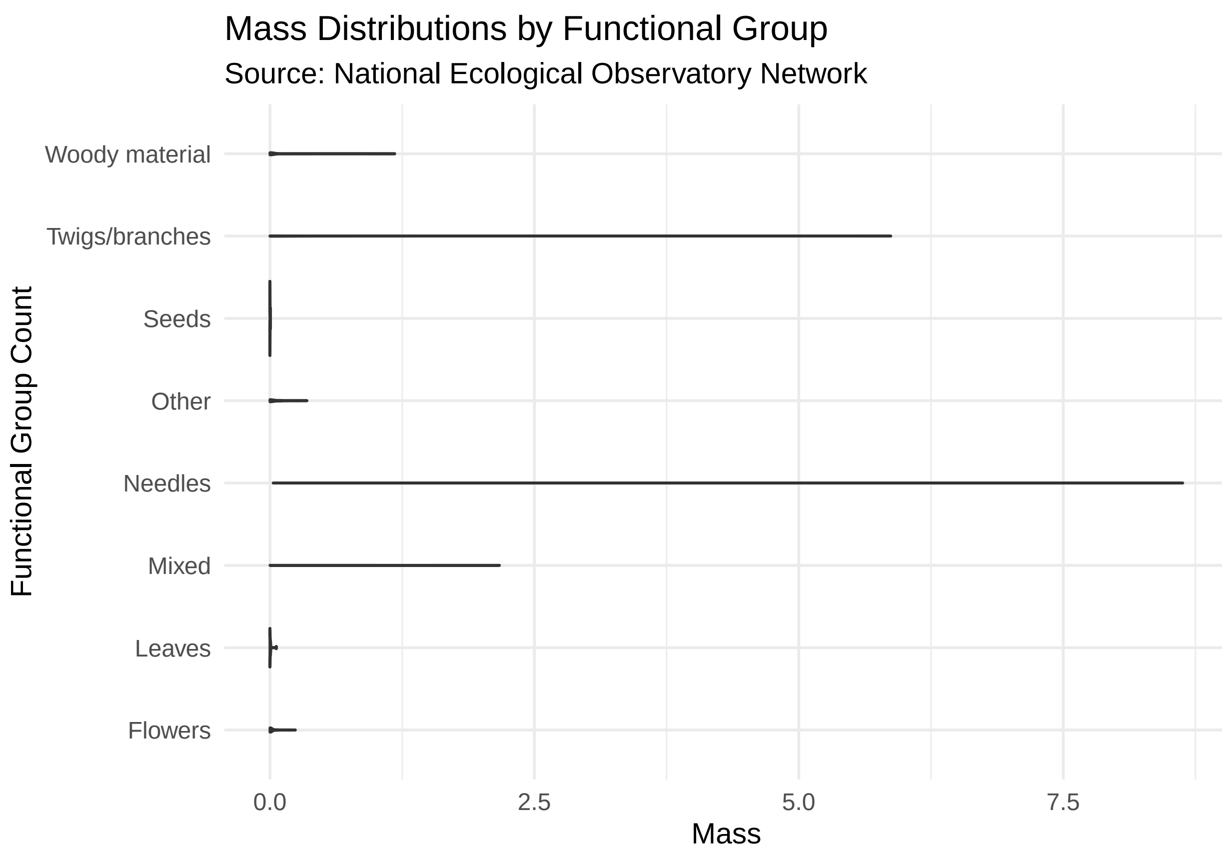


15. Using `geom_boxplot` and `geom_violin`, create a boxplot and a violin plot of dryMass by functionalGroup.

```
# Create boxplot
boxPlot.functionalGroup <- ggplot(Litter, aes(x = dryMass, y = functionalGroup)) +
  # Set boxplot type
  geom_boxplot() +
  # Set title and subtitle
  ggtitle("Mass Distributions by Functional Group",
          "Source: National Ecological Observatory Network") +
  # Set x- and y-axis labels
  labs(x = "Mass", y = "Functional Group Count") +
  theme_minimal()

# Call box plot
boxPlot.functionalGroup
```




```r
violinPlot.functionalGroup <- ggplot(Litter, aes(x = dryMass, y = functionalGroup)) +
  geom_violin() +
  # Set title and subtitle
  ggtitle("Mass Distributions by Functional Group",
          "Source: National Ecological Observatory Network") +
  # Set x- and y-axis labels
  labs(x = "Mass", y = "Functional Group Count") +
  theme_minimal()

# Call violin plot
violinPlot.functionalGroup
```



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

> Answer: The boxplot is more effective in this case because: 1. It provides meaningful summary statistics, such as median, quartiles, etc. 2. Shows outliers The violin plot does not show any density around the the center lines, possibly because the mass distributions are so small.

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

> Answer: Needles (on average)