Homework 5

In the neonDivData data package, there is a data frame named as data_plant. This data frame records plant coverage (percentage at 1 m^2 scale indicated by the sample_area_m2 column) and plant presence information in larger plots (10 and 100 m^2 indicated by the sample_area_m2 column). Use this data frame and functions we learned during lectures to do the steps below.

(2 points) Create a new column named as genus for data_plant from the taxon_name column. The genus name is the first word of the scientific names. For example, if a record has taxon_name of "Bunchosia glandulosa (Cav.) DC.", then the genus is "Bunchosia". You probably want to use regular expression to do so. Take a look at all the names (sort(unique(data_plant$taxon_name))) to look at possible genus names and think about how to specify the regular expression pattern. Randomly select 100 values from the genus column and print it out.

```
data_plant <- neonDivData::data_plant
genus <- stringr::str_extract(data_plant$taxon_name, '[A-Za-z]+')
data_plant <- data.frame(data_plant, genus)
head(data_plant$genus, 100)
```

```
##   [1] "Viburnum"    "Acer"        "Fagus"       "Fagus"       "Fagus"
##   [6] "Fagus"       "Acer"        "Fagus"       "Acer"        "Acer"
##  [11] "Fagus"       "Viburnum"    "Viburnum"    "Betula"      "Viburnum"
##  [16] "Betula"      "Acer"        "Fagus"       "Viburnum"    "Betula"
##  [21] "Viburnum"    "Acer"        "Tsuga"       "Picea"       "Fagus"
##  [26] "Acer"        "Picea"       "Acer"        "Tsuga"       "Tsuga"
##  [31] "Picea"       "Acer"        "Fagus"       "Tsuga"       "Picea"
##  [36] "Acer"        "Picea"       "Tsuga"       "Picea"       "Acer"
##  [41] "Picea"       "Tsuga"       "Acer"        "Tsuga"       "Acer"
##  [46] "Acer"        "Trillium"    "Acer"        "Picea"       "Tsuga"
##  [51] "Picea"       "Acer"        "Fagus"       "Fagus"       "Acer"
##  [56] "Acer"        "Trillium"    "Acer"        "Acer"        "Picea"
##  [61] "Acer"        "Fagus"       "Cypripedium" "Quercus"     "Acer"
##  [66] "Acer"        "Picea"       "Tsuga"       "Fagus"       "Quercus"
##  [71] "Prunus"      "Uvularia"    "Picea"       "Tsuga"       "Fraxinus"
##  [76] "Abies"       "Viburnum"    "Acer"        "Tsuga"       "Betula"
##  [81] "Fagus"       "Acer"        "Picea"       "Fraxinus"    "Acer"
##  [86] "Viburnum"    "Picea"       "Acer"        "Acer"        "Picea"
##  [91] "Viburnum"    "Tsuga"       "Acer"        "Fagus"       "Acer"
##  [96] "Picea"       "Fraxinus"    "Acer"        "Fraxinus"    "Fagus"
```

(2 points) Looking at the taxon_name values, it is clear that some scientific names probably are the same species (as different subspecies). For example, we may want to treat "Calamagrostis canadensis (Michx.) P. Beauv." and "Calamagrostis canadensis (Michx.) P. Beauv. var. langsdorffii (Link) Inman" as the same species. Create a new column taxon_name2 for data_plant based on taxon_name. taxon_name2 should just contain the first two words of taxon_name. For example, "Calamagrostis canadensis (Michx.) P. Beauv." and "Calamagrostis canadensis (Michx.) P. Beauv. var. langsdorffii (Link) Inman" should both be "Calamagrostis canadensis". Randomly select 100 values from the taxon_name2 column and print it out.

```
taxon_name2 <- stringr::str_extract(data_plant$taxon_name, '\\w+\\s+\\w+')
data_plant <- data.frame(data_plant, taxon_name2)
head(data_plant$taxon_name2, 100)
```

```
##   [1] "Viburnum lantanoides"   "Acer saccharum"         "Fagus grandifolia"
##   [4] "Fagus grandifolia"      "Fagus grandifolia"      "Fagus grandifolia"
##   [7] "Acer saccharum"         "Fagus grandifolia"      "Acer pensylvanicum"
##  [10] "Acer saccharum"         "Fagus grandifolia"      "Viburnum lantanoides"
##  [13] "Viburnum lantanoides"   "Betula alleghaniensis"  "Viburnum lantanoides"
##  [16] "Betula alleghaniensis"  "Acer pensylvanicum"     "Fagus grandifolia"
##  [19] "Viburnum lantanoides"   "Betula sp"              "Viburnum lantanoides"
##  [22] "Acer pensylvanicum"     "Tsuga canadensis"       "Picea sp"
##  [25] "Fagus grandifolia"      "Acer rubrum"            "Picea sp"
##  [28] "Acer rubrum"            "Tsuga canadensis"       "Tsuga canadensis"
##  [31] "Picea sp"               "Acer rubrum"            "Fagus grandifolia"
##  [34] "Tsuga canadensis"       "Picea sp"               "Acer sp"
##  [37] "Picea sp"               "Tsuga canadensis"       "Picea sp"
##  [40] "Acer rubrum"            "Picea sp"               "Tsuga canadensis"
##  [43] "Acer rubrum"            "Tsuga canadensis"       "Acer rubrum"
##  [46] "Acer pensylvanicum"     "Trillium sp"            "Acer rubrum"
##  [49] "Picea sp"               "Tsuga canadensis"       "Picea sp"
##  [52] "Acer rubrum"            "Fagus grandifolia"      "Fagus grandifolia"
##  [55] "Acer rubrum"            "Acer rubrum"            "Trillium sp"
##  [58] "Acer pensylvanicum"     "Acer rubrum"            "Picea sp"
##  [61] "Acer rubrum"            "Fagus grandifolia"      "Cypripedium sp"
##  [64] "Quercus rubra"          "Acer pensylvanicum"     "Acer saccharum"
##  [67] "Picea sp"               "Tsuga canadensis"       "Fagus grandifolia"
##  [70] "Quercus rubra"          "Prunus pensylvanica"    "Uvularia sessilifolia"
##  [73] "Picea sp"               "Tsuga canadensis"       "Fraxinus pennsylvanica"
##  [76] "Abies sp"               "Viburnum lantanoides"   "Acer saccharum"
##  [79] "Tsuga canadensis"       "Betula alleghaniensis"  "Fagus grandifolia"
##  [82] "Acer rubrum"            "Picea sp"               "Fraxinus pennsylvanica"
##  [85] "Acer pensylvanicum"     "Viburnum lantanoides"   "Picea sp"
##  [88] "Acer rubrum"            "Acer rubrum"            "Picea sp"
##  [91] "Viburnum lantanoides"   "Tsuga canadensis"       "Acer saccharum"
##  [94] "Fagus grandifolia"      "Acer rubrum"            "Picea sp"
##  [97] "Fraxinus pennsylvanica" "Acer pensylvanicum"     "Fraxinus pennsylvanica"
## [100] "Fagus grandifolia"
```

(2 points) Calculate the number of species (based on taxon_name2) of each site observed based on different sizes of plot:

based on 1 m^2 plots; this would be all observations with sample_area_m2 == "1". This would result in a data frame named as n_1 with two columns: siteID and richness_1m2.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(magrittr)
n1 <- filter(data_plant, sample_area_m2 == "1")
n1 <- n1 %>% group_by(siteID) %>% summarise(richness_1m2 = length(unique(taxon_name2)))
n1
```

```
## # A tibble: 47 x 2
##    siteID richness_1m2
##    <chr>         <int>
##  1 ABBY            188
##  2 BARR             71
##  3 BART             80
##  4 BLAN            268
##  5 BONA             72
##  6 CLBJ            413
##  7 CPER            185
##  8 DCFS            223
##  9 DEJU            152
## 10 DELA            303
## # ... with 37 more rows
```

based on 10 m^2 plots; this would be all observations with sample_area_m2 %in% c("1", "10"). This would result in a data frame named as n_10 with two columns: siteID and richness_10m2.

```
n10 <- filter(data_plant, sample_area_m2 %in% c("1", "10"))
n10 <- n10 %>% group_by(siteID) %>% summarise(richness_10m2 = length(unique(taxon_name2)))
n10
```

```
## # A tibble: 47 x 2
##    siteID richness_10m2
##    <chr>          <int>
##  1 ABBY             228
##  2 BARR              87
##  3 BART             104
##  4 BLAN             313
##  5 BONA              88
##  6 CLBJ             477
##  7 CPER             222
##  8 DCFS             264
##  9 DEJU             183
## 10 DELA             391
## # ... with 37 more rows
```

based on 100 m^2 plots; this would be all observations with sample_area_m2 %in% c("1", "10", "100"). This would result in a data frame named as n_100 with two columns: siteID and richness_100m2.

```
n_100 <- filter(data_plant, sample_area_m2 %in% c("1", "10", "100"))
n_100 <- n_100 %>% group_by(siteID) %>% summarise(richness_100m2 = length(unique(taxon_name2)))
n_100
```

```
## # A tibble: 47 x 2
##    siteID richness_100m2
##    <chr>           <int>
##  1 ABBY              261
##  2 BARR               91
##  3 BART              127
##  4 BLAN              378
##  5 BONA              100
##  6 CLBJ              517
##  7 CPER              241
##  8 DCFS              293
##  9 DEJU              198
## 10 DELA              457
## # ... with 37 more rows
```

then, use dplyr::left_join() to join n_1, n_10, and n_100 as one data frame n_all, which should have 47 rows and four columns: siteID, richness_1m2, richness_10m2, and richness_100m2. Note: dplyr::left_join() can only join two data frames at each time, so you may use pipe (e.g., xyz = left_join(x, y) %>% left_join(z)).

```
n_all <- left_join(n1, n10, by = "siteID") %>% left_join(n_100, by = "siteID")
n_all
```

```
## # A tibble: 47 x 4
##    siteID richness_1m2 richness_10m2 richness_100m2
##    <chr>         <int>         <int>          <int>
##  1 ABBY            188           228            261
##  2 BARR             71            87             91
##  3 BART             80           104            127
##  4 BLAN            268           313            378
##  5 BONA             72            88            100
##  6 CLBJ            413           477            517
##  7 CPER            185           222            241
##  8 DCFS            223           264            293
##  9 DEJU            152           183            198
## 10 DELA            303           391            457
## # ... with 37 more rows
```

(2 points) Transform n_all to a long format data frame named as n_all_long with three columns: siteID, spatial_scale, and richness. Hint: tidyr::pivot_longer().

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:magrittr':
##
##     extract
```

```
n_all_long <- n_all %>% pivot_longer(cols = -c(siteID), names_to = 'spatial_scale', values_to = "richnes
n_all_long
```

```
## # A tibble: 141 x 3
##    siteID spatial_scale  richness
##    <chr>  <chr>             <int>
##  1 ABBY   richness_1m2        188
##  2 ABBY   richness_10m2       228
##  3 ABBY   richness_100m2      261
##  4 BARR   richness_1m2         71
##  5 BARR   richness_10m2        87
##  6 BARR   richness_100m2       91
##  7 BART   richness_1m2         80
##  8 BART   richness_10m2       104
##  9 BART   richness_100m2      127
## 10 BLAN   richness_1m2        268
## # ... with 131 more rows
```

(2 points) Use ggplot2 and n_all_long to generate the plot below. Each line links the three values of each site (hint: aes(group = siteID)).

```
library(ggplot2)
siteID <- n_all_long$siteID
richness <- n_all_long$richness
spatial_scale <- n_all_long$spatial_scale
richness_plot <- ggplot(n_all_long, aes(x = spatial_scale, y = richness, group = siteID)) + geom_line()
richness_plot
```