INF 550 Intro to NEON Exercises Part 1

Natasha Wesely

2022-09-16

2.9.1.1 Part 1: Sign up for and Use an NEON API Token:

```
# call in your unique NEON API token
load('neon_token_source.Rdata')

# demonstrate the you can use your NEON API token to pull some data
veglist <- loadByProduct(dpID="DP1.10098.001", site="WREF", package="basic", check.size=FALSE, token = 
```

```
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 18.79386 MB
## Downloading 15 files
##   |                                                                              |
##
## Unpacking zip files using 1 cores.
## Stacking operation across a single core.
## Stacking table vst_apparentindividual
## Stacking table vst_mappingandtagging
## Stacking table vst_perplotperyear
## Stacking table vst_non-woody
## Copied the most recent publication of validation file to /stackedFiles
## Copied the most recent publication of categoricalCodes file to /stackedFiles
## Copied the most recent publication of variable definition file to /stackedFiles
## Finished: Stacked 4 data tables and 4 metadata tables!
## Stacking took 0.408643 secs
```

```
# prep the data further
vegmap <- getLocTOS(veglist$vst_mappingandtagging,
                    "vst_mappingandtagging")
```

```
##   |                                                                              |
```

```
veg <- merge(veglist$vst_apparentindividual, vegmap,
             by=c("individualID","namedLocation",
                  "domainID","siteID","plotID"))
```

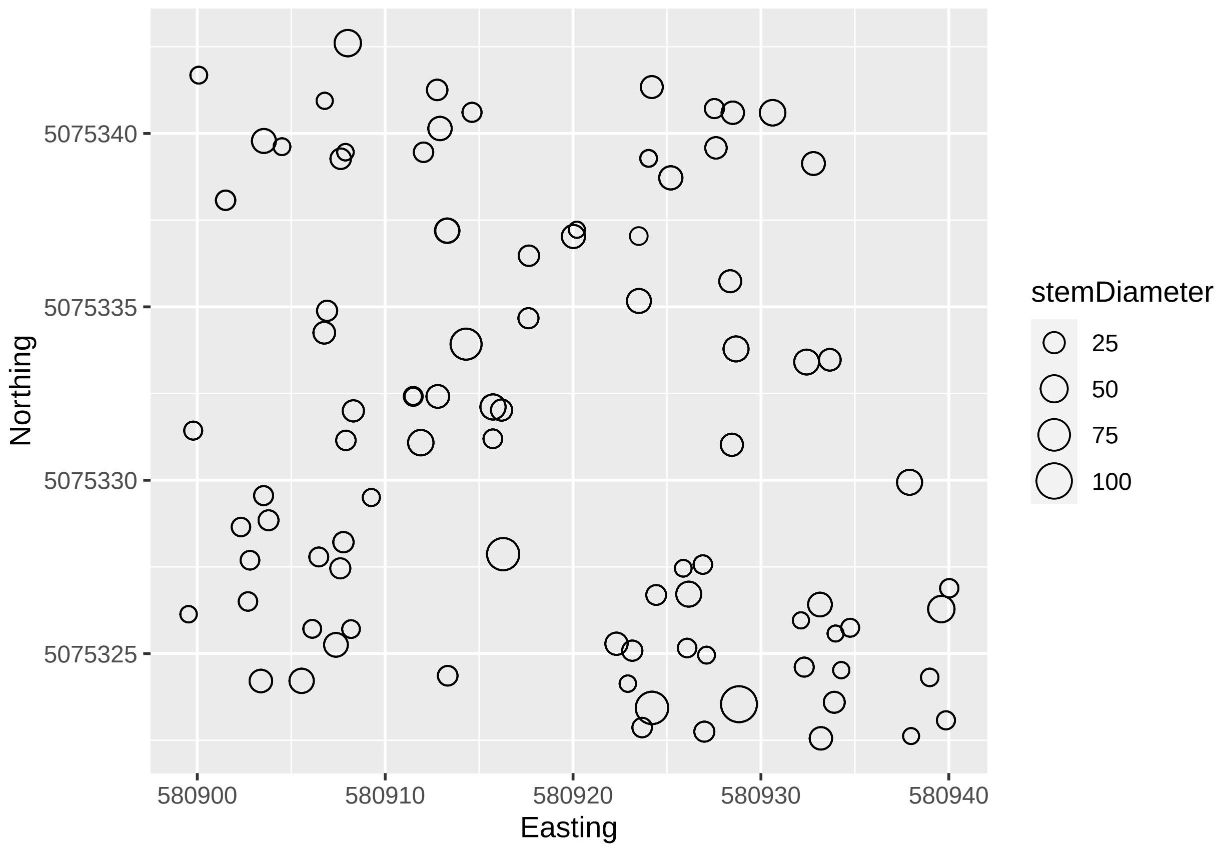
2.9.1.2 Part 2: Further Investigation of NEON TOS Vegetation Structure Data

Question 1

Convert the previous diagram plot into a ggplot.

```
veg %>%
  filter(plotID == "WREF_075") %>%
  ggplot() +
  geom_point(aes(x = adjEasting, y = adjNorthing,
                 size = stemDiameter), shape = 1) +
  labs(x = "Easting", y = "Northing")
```

```
## Warning: Removed 116 rows containing missing values (geom_point).
```

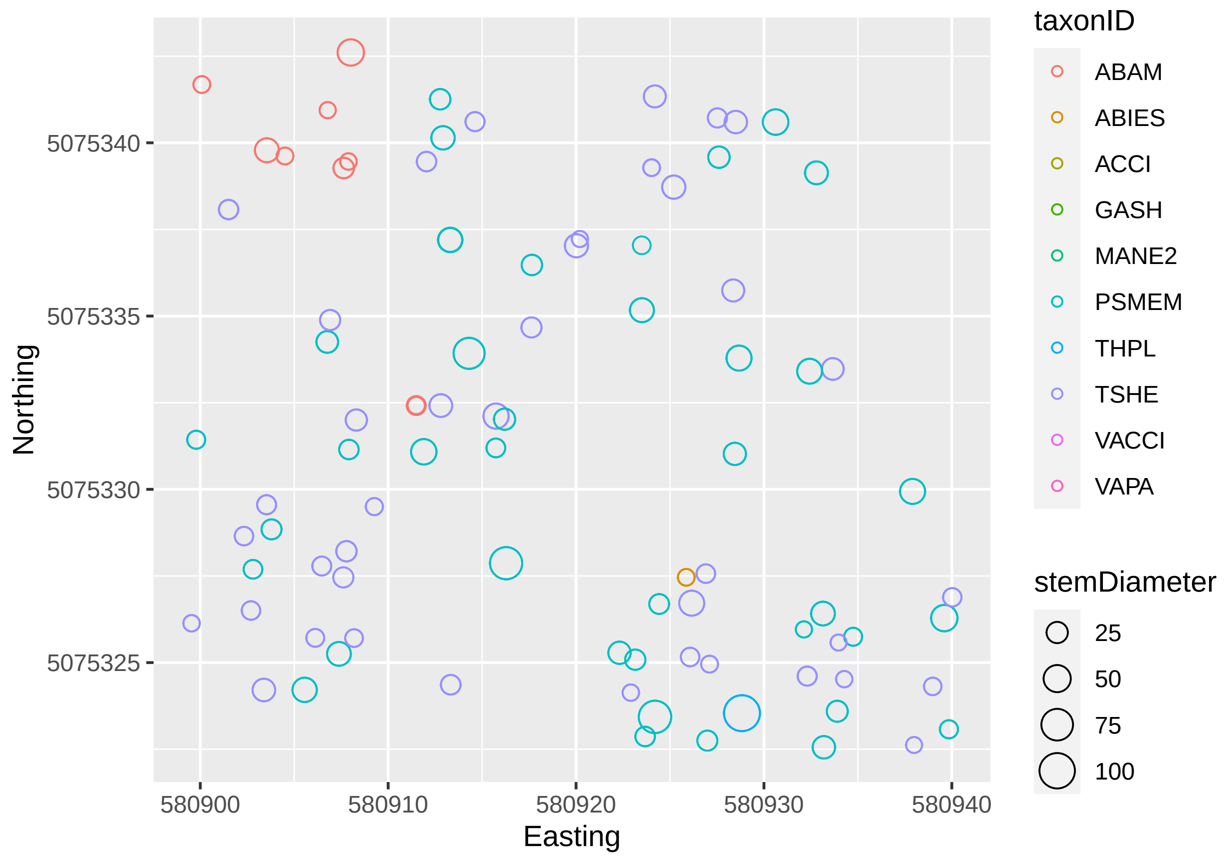


Question 2

Set the color your circles to be a function of each species.

```
veg %>%
  filter(plotID == "WREF_075") %>%
  ggplot() +
  geom_point(aes(x = adjEasting, y = adjNorthing,
                 size = stemDiameter, color = taxonID), shape = 1) +
  labs(x = "Easting", y = "Northing")
```

```
## Warning: Removed 116 rows containing missing values (geom_point).
```



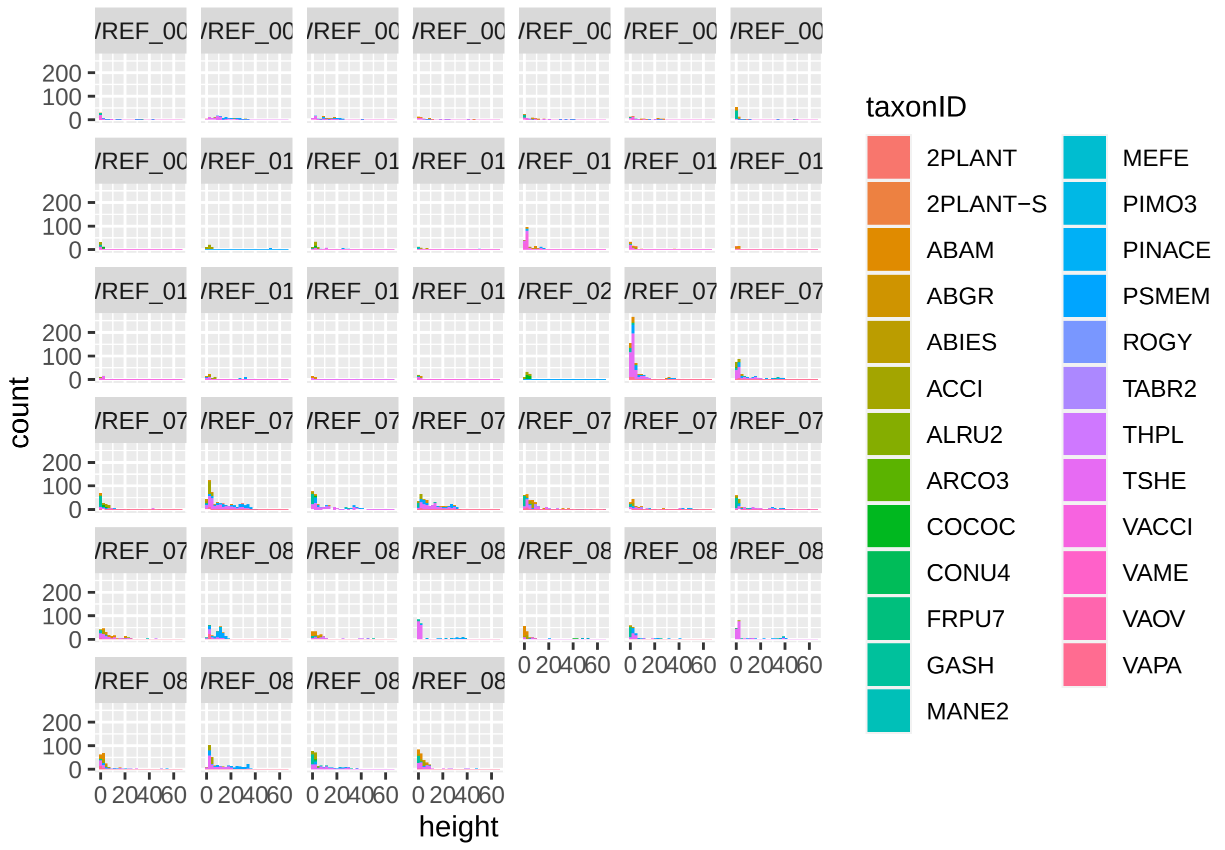
Question 3

Generate a histogram of tree heights for each plot. Color your stacked bar as a function of each species.

```
veg %>% ggplot() +
  geom_histogram(aes(x = height, fill = taxonID)) +
  facet_wrap(~ plotID)
```

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

```
## Warning: Removed 296 rows containing non-finite values (stat_bin).
```



Question 4

Use dplyr to remove dead trees.

```
# what are all the options in plantStatus?
unique(veg$plantStatus)
```

```
##  [1] "Live"                        "Dead, broken bole"
##  [3] "Live, physically damaged" "Lost, fate unknown"
##  [5] "Standing dead"               "No longer qualifies"
##  [7] "Live, broken bole"           "Live, disease damaged"
##  [9] "Downed"                      "Live,  other damage"
## [11] "Live, insect damaged"
```

```
# filter out any dead/downed trees
vegLiv = veg %>%
  filter(
    plantStatus != "Dead, broken bole",
    plantStatus != "Lost, fate unknown",
    plantStatus != "Standing dead",
    plantStatus != "No longer qualifies",
    plantStatus != "Downed")

# make sure your filtering worked
unique(vegLiv$plantStatus)
```

```
## [1] "Live"                      "Live, physically damaged"
## [3] "Live, broken bole"         "Live, disease damaged"
## [5] "Live,  other damage"       "Live, insect damaged"
```

Question 5

Create a simple linear model that uses Diameter at Breast Height (DBH) and height to predict allometries. Print the summary information of your model.

```
model = lm(baseCrownHeight ~ stemDiameter + height,
           data = vegLiv)
summary(model)
```

```
##
## Call:
## lm(formula = baseCrownHeight ~ stemDiameter + height, data = vegLiv)
##
## Residuals:
##     1981     1982     2126     2127     4648     4903     4904     4905
## -0.31195 -0.14978 -0.02367 -0.26239  0.34934  0.16106  0.04491  0.19248
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.9080     2.0333  -0.938    0.391
## stemDiameter  -0.3872     0.5436  -0.712    0.508
## height         1.8827     0.9633   1.954    0.108
##
## Residual standard error: 0.2743 on 5 degrees of freedom
##   (5754 observations deleted due to missingness)
## Multiple R-squared:  0.5298, Adjusted R-squared:  0.3417
## F-statistic: 2.816 on 2 and 5 DF,  p-value: 0.1516
```

Question 6

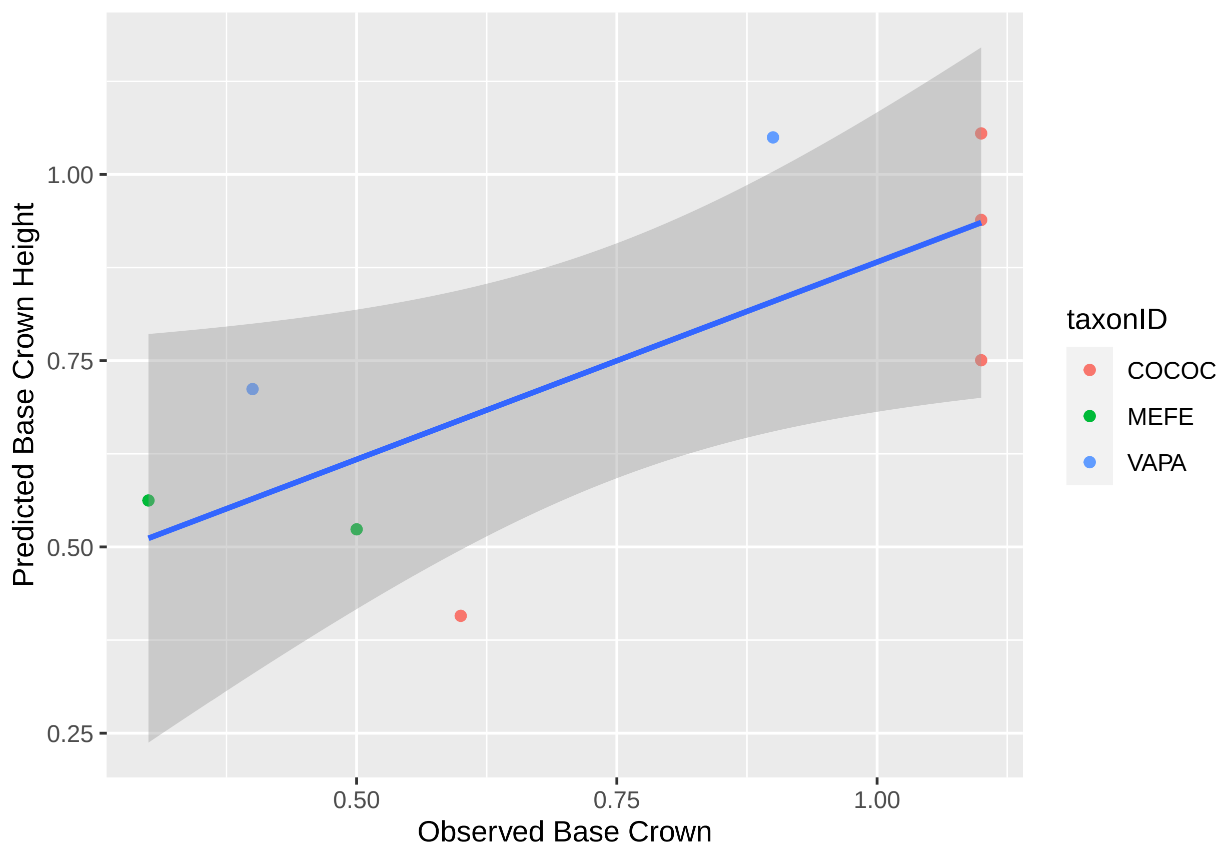
Plot your linear model.

```
plot.df = vegLiv %>%
  select(stemDiameter, height, baseCrownHeight, taxonID) %>%
  na.omit()

plot.df$predictions = predict(model)

plot.df %>% ggplot() +
  geom_point(aes(x = baseCrownHeight, y = predictions, color = taxonID)) +
  geom_smooth(aes(x = baseCrownHeight, y = predictions), method = "lm") +
  labs(y = "Predicted Base Crown Height", x = "Observed Base Crown")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



I realize I did not make separate lines for the different species, but that is because I only have 8 observations (2 observations for VAPA, 2 observations for MEFE, and 3 observations for COCOC). Having separate lines would be nonsensical.

Question 7

Answer the following questions.

What do you think about your simile linear model? What are its limitations?

My simple linear model is not great but not terrible. The biggest issue with this model that makes this model questionable is the extremely small sample size. Linear models have a lot of limitations, including the assumption of normally distributed residuals, the assumption of a linear relationship between your response and predictor(s), and the assumption of homoscedasticty. It's very challenging characterize complicated relationships with a simple linear model in general.

How many unique species are present at WREF?

```
length(unique(veg$taxonID))
```

```
## [1] 25
```

There are 25 unique species at the WREF site.

What are the top_5 trees based on height? Diameter?

```r
# What are the top 5 trees based on height?
vegHT = vegLiv %>% arrange(desc(height)) %>%
  select(individualID, height)

head(vegHT, n = 5)
```

```
##               individualID height
## 1 NEON.PLA.D16.WREF.07530   66.0
## 2 NEON.PLA.D16.WREF.04630   65.5
## 3 NEON.PLA.D16.WREF.04806   59.7
## 4 NEON.PLA.D16.WREF.02242   58.8
## 5 NEON.PLA.D16.WREF.05896   58.3
```

```r
# what are the top 5 trees based on diameter?
vegDM = vegLiv %>% arrange(desc(stemDiameter)) %>%
  select(individualID, stemDiameter)

head(vegDM, n = 5)
```

```
##               individualID stemDiameter
## 1 NEON.PLA.D16.WREF.09998        140.9
## 2 NEON.PLA.D16.WREF.04803        139.3
## 3 NEON.PLA.D16.WREF.05890        137.0
## 4 NEON.PLA.D16.WREF.04803        135.0
## 5 NEON.PLA.D16.WREF.02139        130.1
```

What proportion of sampled trees are dead?

```r
liveTrees = length(unique(vegLiv$individualID))
allTrees = length(unique(veg$individualID))
livePropor = liveTrees / allTrees
(deadTreesProportion = 1 - livePropor)
```

```
## [1] 0.1175657
```