

# INF 550 Intro to NEON Exercises Part 1

Natasha Wesely

2022-09-13

## 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 = I

## 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 0.5473118 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"))
```

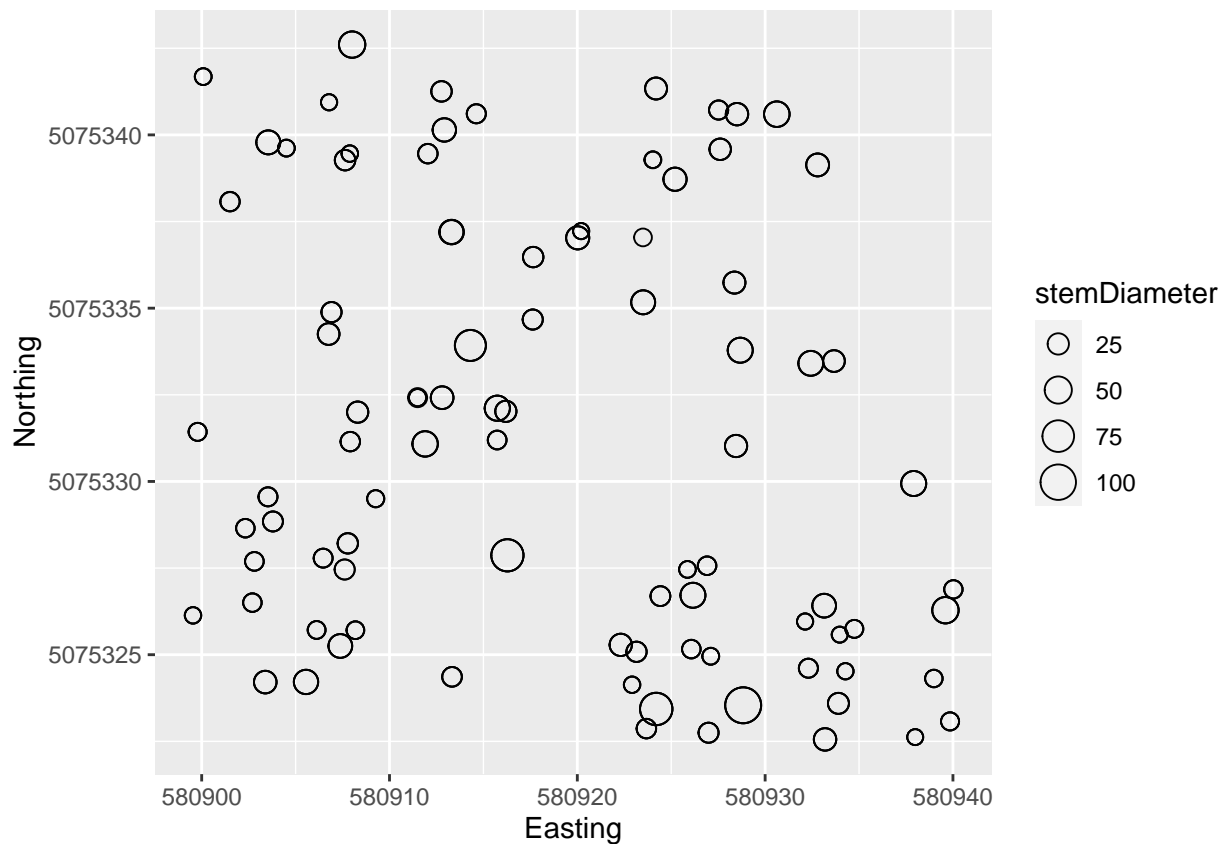
## 2.9.1.2 Part 2: Further Investigation of NEON TOS Vegetation Structure Data

### 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).
```



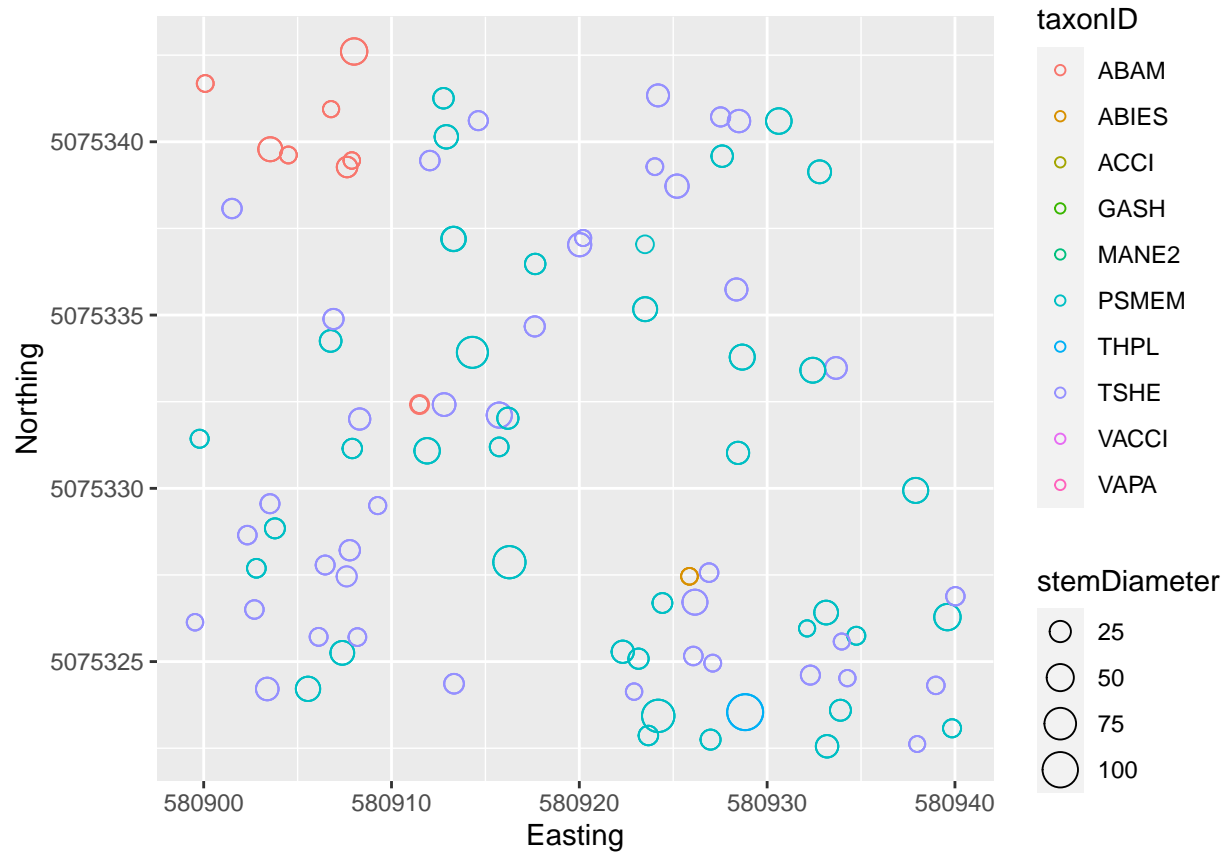
### Question 2

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).
```



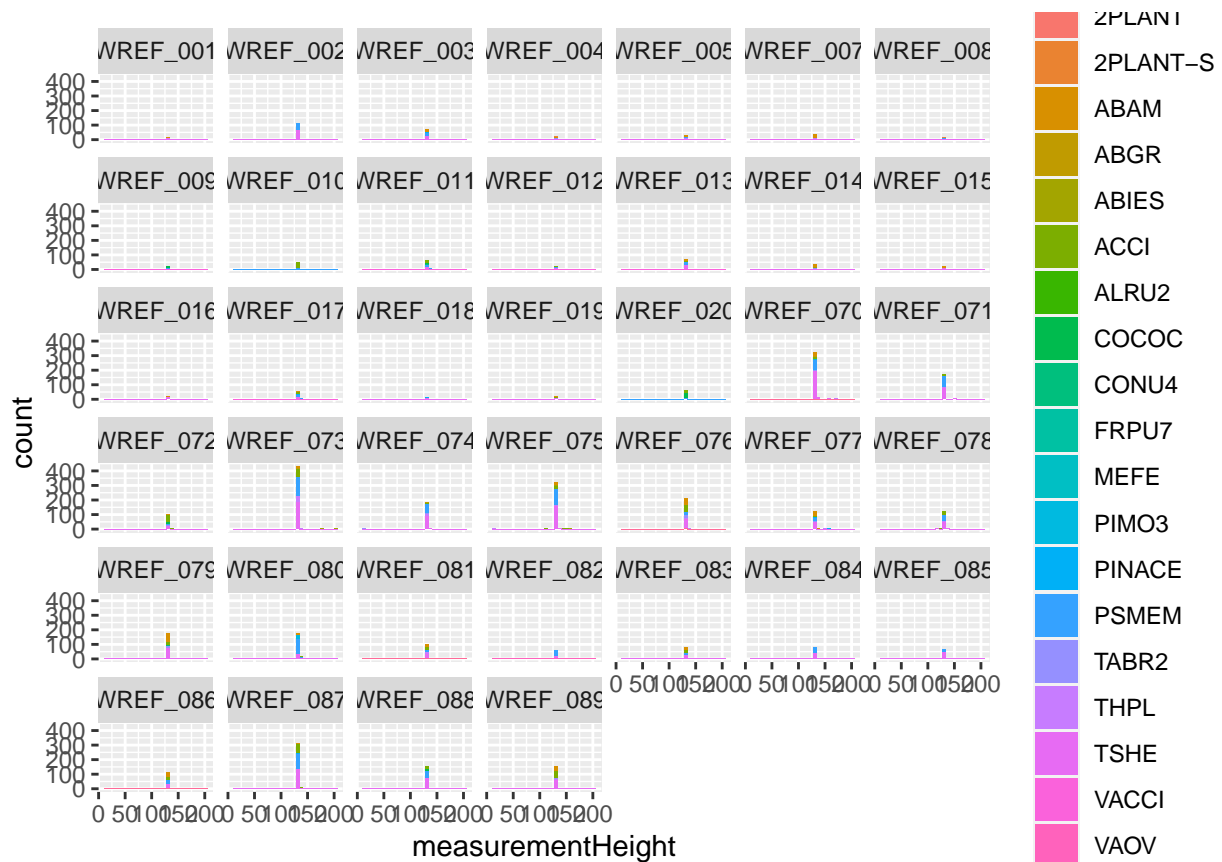
### Question 3

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 = measurementHeight, fill = taxonID)) +
  facet_wrap(~ plotID)
```

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

```
## Warning: Removed 2372 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(bandStemDiameter ~ stemDiameter + measurementHeight,
            data = vegLiv)
summary(model)
```

```
##
## Call:
## lm(formula = bandStemDiameter ~ stemDiameter + measurementHeight,
##     data = vegLiv)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4622  -0.9490  -0.0946   0.6023  26.1150
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.49634    18.92143  -0.132   0.895
## stemDiameter    0.94449     0.00896 105.408 <2e-16 ***
## measurementHeight 0.02349     0.14504   0.162   0.872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.867 on 149 degrees of freedom
## (5610 observations deleted due to missingness)
## Multiple R-squared:  0.9868, Adjusted R-squared:  0.9867
## F-statistic: 5583 on 2 and 149 DF, p-value: < 2.2e-16
```

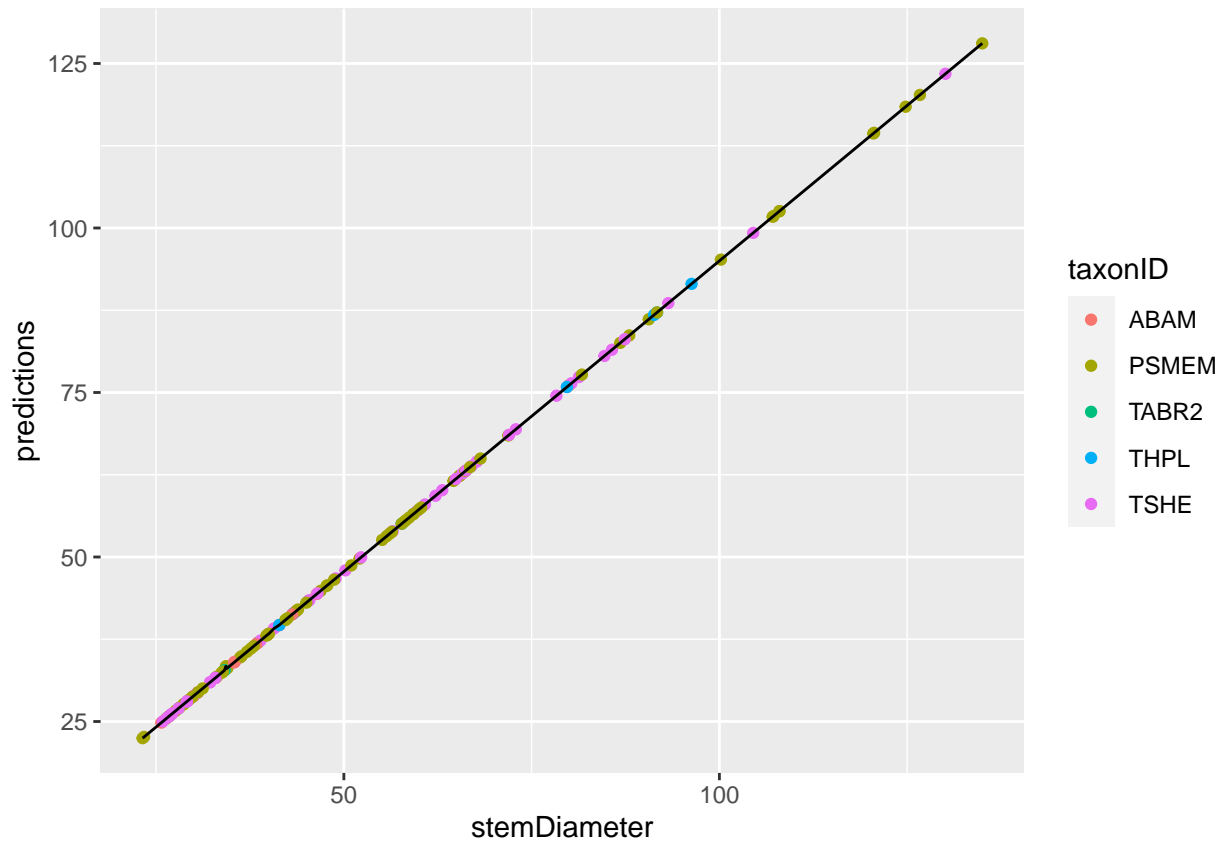
## Question 6 !!!!

Plot your linear model.

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

plot.df$predictions = predict(model)

plot.df %>% ggplot() +
  geom_point(aes(x = stemDiameter, y = predictions, color = taxonID)) +
  geom_line(aes(x = stemDiameter, y = predictions))
```



## Question 7

Answer the following questions.

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

My simple linear model is pretty poor, with a very low  $R^2$ . 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 homoscedasticity. It's very challenging to 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?**

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

head(vegHT, n = 5)
```

```
##           individualID measurementHeight
## 1 NEON.PLA.D16.WREF.05734             200
## 2 NEON.PLA.D16.WREF.05375             195
## 3 NEON.PLA.D16.WREF.05375             195
## 4 NEON.PLA.D16.WREF.05119             180
## 5 NEON.PLA.D16.WREF.05119             180
```

```
# 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?

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

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