

SUNDQUIST_INF550_2.8

Shelby Sundquist

2022-09-12

```
options(stringsAsFactors = F)
library(neonUtilities)
```

```
## Warning: package 'neonUtilities' was built under R version 4.1.3
```

```
library(geoNEON)
library(sp)
```

```
## Warning: package 'sp' was built under R version 4.1.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
```

```
## 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
```

```
source('neon_token_source.R')
```

Section 2.8

```
veglist <- loadByProduct(dpID="DP1.10098.001", site="WREF", package="basic", check.size=FALSE, token = 1)
```

```
## 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 1.698833 secs

vegmap <- getLocTOS(veglist$vst_mappingandtagging,
                    "vst_mappingandtagging")

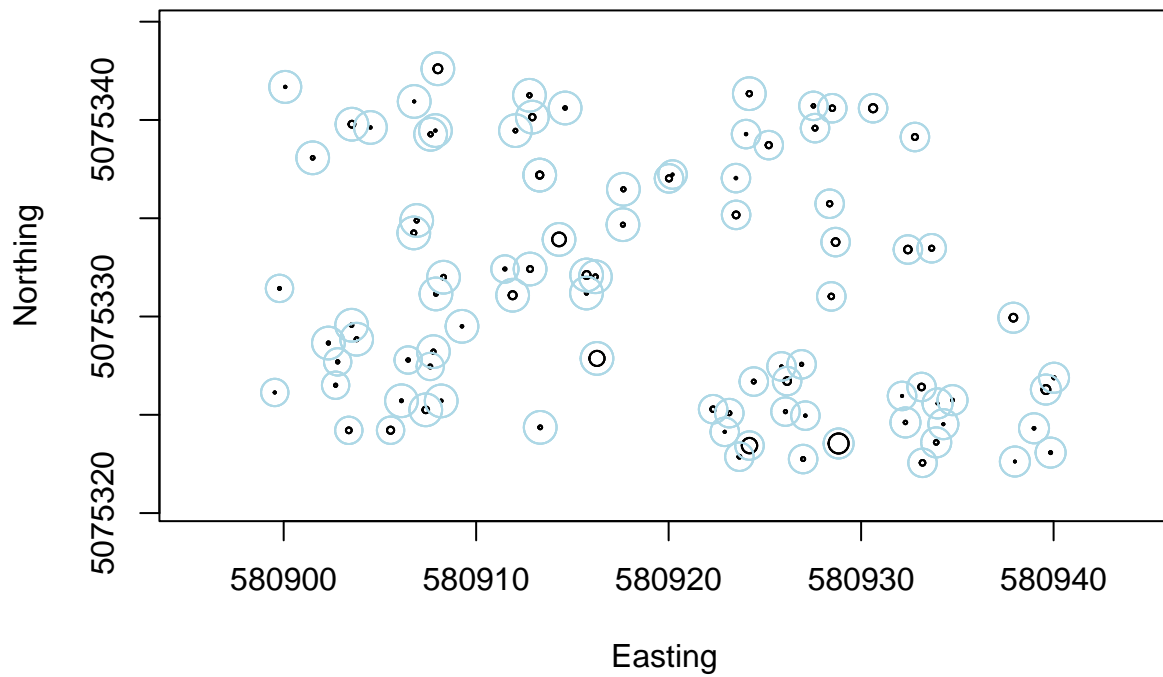
##      |
veg <- merge(veglist$vst_apparentindividual, vegmap,
            by=c("individualID", "namedLocation",
                 "domainID", "siteID", "plotID"))

symbols(veg$adjEasting[which(veg$plotID=="WREF_075")],
        veg$adjNorthing[which(veg$plotID=="WREF_075")],
        circles=veg$stemDiameter[which(veg$plotID=="WREF_075")]/100/2,
        inches=F, xlab="Easting", ylab="Northing")

symbols(veg$adjEasting[which(veg$plotID=="WREF_075")],
        veg$adjNorthing[which(veg$plotID=="WREF_075")],
        circles=veg$stemDiameter[which(veg$plotID=="WREF_075")]/100/2,
        inches=F, xlab="Easting", ylab="Northing")

symbols(veg$adjEasting[which(veg$plotID=="WREF_075")],
        veg$adjNorthing[which(veg$plotID=="WREF_075")],
        circles=veg$adjCoordinateUncertainty[which(veg$plotID=="WREF_075")],
        inches=F, add=T, fg="lightblue")

```

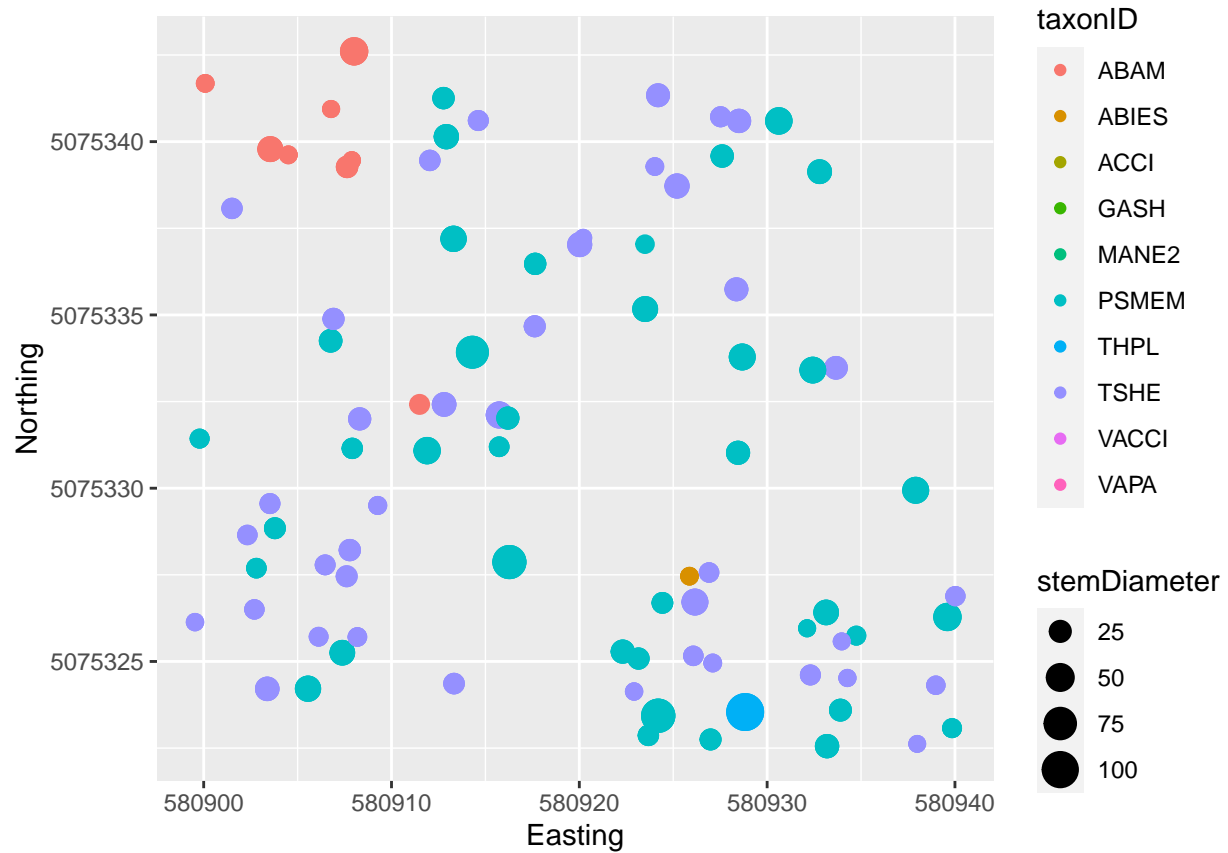


Section 2.9

#1 - 2

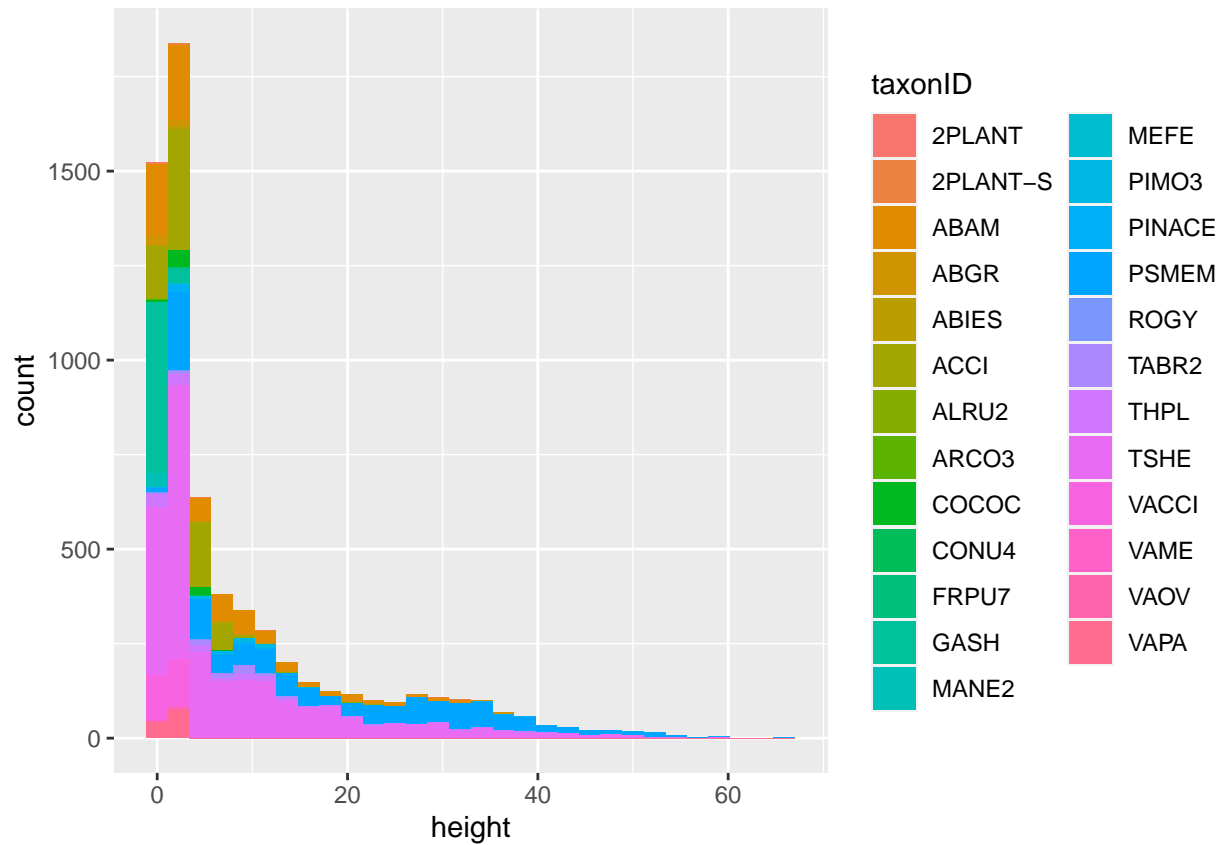
```
ggplot(subset(veg, plotID == "WREF_075"), aes(x = adjEasting, y = adjNorthing, color = taxonID, size = 
  geom_point() + ylab("Northing") + xlab("Easting")
```

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



```
#3
ggplot(subset(veg, !is.na(height)), aes(height, fill = taxonID)) + geom_histogram()

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

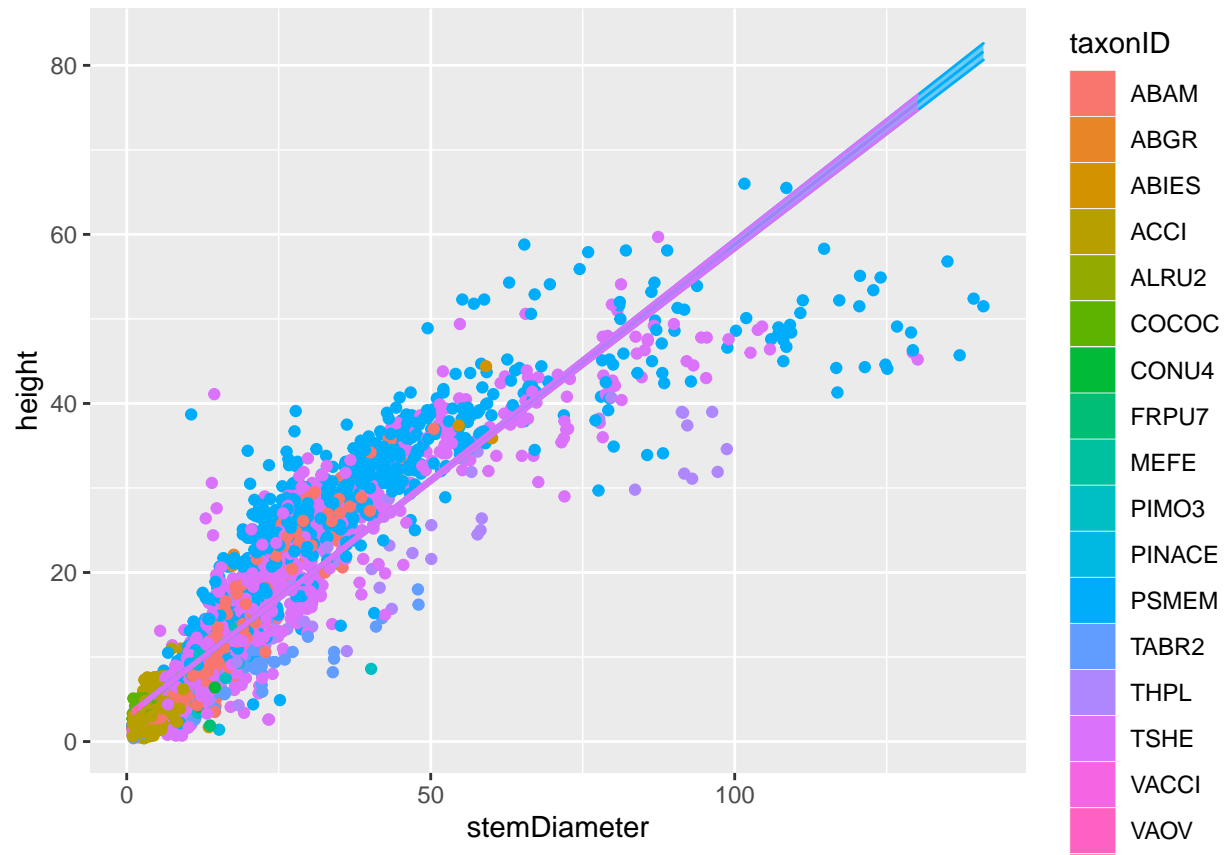


#4 - 6

```
live = veg %>% filter(!plantStatus %in% c("Standing dead", "Dead, broken bole", "Downed"))
live$taxonID = as.factor(live$taxonID)
live = live[!is.na(live$height),]
live = live[!is.na(live$stemDiameter),]
model = lm(height ~ stemDiameter, data = live)
```

```
fit = live %>% select(-matches("fit"), -matches("lwr"), -matches("upr")) %>%
cbind(predict(model, newdata = ., interval = 'confidence'))
```

```
ggplot(fit, aes(x = stemDiameter, y = height, color = taxonID)) + geom_point() + geom_line(aes(y = fit)) +
  geom_ribbon(aes(ymin = lwr, ymax = upr, alpha = 0.2, fill = taxonID))
```



```
#7
#Linear model assessment: it's not great, mostly because the data doesn't appear linear.
#Height increases more slowly for trees with large DBH.
#Also, the confidence interval hardly appears to encompass the data, indicating that there are factors
#adequately considering

#Number of unique species:
length(unique(veg$taxonID))

## [1] 25

#Tallest/ greatest diameter trees:
veg[order(veg$height),]$individualID[1:5]

## [1] "NEON.PLA.D16.WREF.02138" "NEON.PLA.D16.WREF.02138"
## [3] "NEON.PLA.D16.WREF.02481" "NEON.PLA.D16.WREF.02497"
## [5] "NEON.PLA.D16.WREF.04487"

veg[order(veg$stemDiameter),]$individualID[1:5]

## [1] "NEON.PLA.D16.WREF.02003" "NEON.PLA.D16.WREF.02027"
## [3] "NEON.PLA.D16.WREF.02411" "NEON.PLA.D16.WREF.03257"
## [5] "NEON.PLA.D16.WREF.04211"

#What proportion of sampled trees are dead?
veg %>% group_by(plantStatus) %>% count() %>% mutate(proportion = n/nrow(veg))

## # A tibble: 11 x 3
## # Groups:   plantStatus [11]
```

| ## | plantStatus | n | proportion |
|-------|--------------------------|-------|------------|
| ## | <chr> | <int> | <dbl> |
| ## 1 | Dead, broken bole | 419 | 0.0618 |
| ## 2 | Downed | 15 | 0.00221 |
| ## 3 | Live | 5091 | 0.751 |
| ## 4 | Live, other damage | 140 | 0.0207 |
| ## 5 | Live, broken bole | 94 | 0.0139 |
| ## 6 | Live, disease damaged | 118 | 0.0174 |
| ## 7 | Live, insect damaged | 11 | 0.00162 |
| ## 8 | Live, physically damaged | 308 | 0.0455 |
| ## 9 | Lost, fate unknown | 55 | 0.00812 |
| ## 10 | No longer qualifies | 40 | 0.00590 |
| ## 11 | Standing dead | 484 | 0.0714 |

#Between 13-14% of trees sampled were dead