NEON AOP Coding Lab

Emma Reich

10/8/2020

1: How are these two forests similar? How are they different? *(3-5 sentences)*

The GUAN forest is a dry neotropical forest near the coast with deciduous and semi-evergreen species with scrub along the coast. The branching patterns at GUAN seem sparser and spread out compared the BART. The climate is warmer at GUAN than BART, so I would expect higher photosynthetic activity at GUAN for most of the year. The BART forest is a boreal ecotone Northeastern deciduous forest with some evergreen conifers, which means it experiences more seasonality than GUAN. The canopy at BART seems denser with significant leaf litter lining the understory.

2: Using this NEON tutorial and the tutorials we've covered in this textbook *(hint: you did half of this workflow in your very first coding lab)* pull the NEON AOP derived Canopy Height Model (CHM, `DP3.30015.001`) and High-resolution orthorectified camera imagery mosaic `DP3.30010.001` for each forest and overlay the NEON TOS Woody Vegetation Structure `DP1.10098.001` basal diameter data product to evaluate how well AOP captures trees in each forest.
Generate a labeled 2x2 plot panel including:

- **Each RGB image with basal diameter overlaid**
- **Each CHM with basal diameter overlaid**

```
soi <- c("BART","GUAN") # sites of interest

# Download veg structure data from NEON API
veglist <- loadByProduct(dpID="DP1.10098.001", # Veg structure
                         site=soi,
                         package="expanded",
                         check.size=FALSE,
                         token = NEON_TOKEN)
```

```
## Finding available files
##   |                                                                                |
```

```
## No expanded package found for site BART and month 2019-07. Basic package downloaded instead.
## No expanded package found for site BART and month 2019-08. Basic package downloaded instead.
## No expanded package found for site BART and month 2019-09. Basic package downloaded instead.
## No expanded package found for site BART and month 2019-10. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2015-05. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2015-06. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2015-07. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2015-08. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2015-09. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2015-10. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2015-11. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-02. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-03. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-04. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-05. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-06. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-07. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-08. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-09. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-10. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2016-11. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2017-02. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2017-03. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2017-04. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2017-12. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2018-01. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2018-02. Basic package downloaded instead.
## No expanded package found for site GUAN and month 2019-01. Basic package downloaded instead.
## Downloading files totaling approximately 12.690605 MB
## Downloading 42 files
##   |                                                                              |
```

```r
# Unlist veg data frames
list2env(veglist,.GlobalEnv)
```

```
## <environment: R_GlobalEnv>
```

```r
vegmap <- getLocTOS(veglist$vst_mappingandtagging, "vst_mappingandtagging") # returns vegmap object
```

```
##   |                                                                              |
```

```
## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
```

```
## BART_001.basePlot.vst.32

##   |                                                                            |=

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_005.basePlot.vst.32

##   |                                                                            |===

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_006.basePlot.vst.32

##   |                                                                            |====

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_015.basePlot.vst.32

##   |                                                                            |====

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_006.basePlot.vst.39

##   |                                                                            |=====

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_011.basePlot.vst.32

##   |                                                                            |=======

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_025.basePlot.vst.32

##   |                                                                            |=========

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_026.basePlot.vst.32

##   |                                                                            |=========

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_010.basePlot.vst.50

##   |                                                                            |==========

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_010.basePlot.vst.32

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_010.basePlot.vst.40

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_010.basePlot.vst.42
```

```
##   |                                                                              |==========

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_023.basePlot.vst.32

##   |                                                                              |===========

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_004.basePlot.vst.32

##   |                                                                              |============

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_018.basePlot.vst.32

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_028.basePlot.vst.32

##   |                                                                              |==============

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_004.basePlot.vst.39

##   |                                                                              |================

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_013.basePlot.vst.32

##   |                                                                              |=================

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_003.basePlot.vst.32

##   |                                                                              |==================

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_024.basePlot.vst.32

##   |                                                                              |===================

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_027.basePlot.vst.32

##   |                                                                              |===================

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_016.basePlot.vst.32

##   |                                                                              |===================

## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_033.basePlot.vst.50
```

```
##   |                                                                          |==================
```

```
## Warning in geoNEON::getLocByName(data, locCol = locCol, locOnly = T,
## token = token): WARNING: the following namedLocation was not found:
## BART_075.basePlot.vst.42
```

```
##   |                                                                          |==================
```

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

# Separate by plotID
vegbart <- veg %>%
   filter(plotID == "BART_013") %>%
   drop_na(adjNorthing, adjEasting)

vegguan <- veg %>%
   filter(plotID == "GUAN_043") %>%
   drop_na(adjNorthing, adjEasting)
```

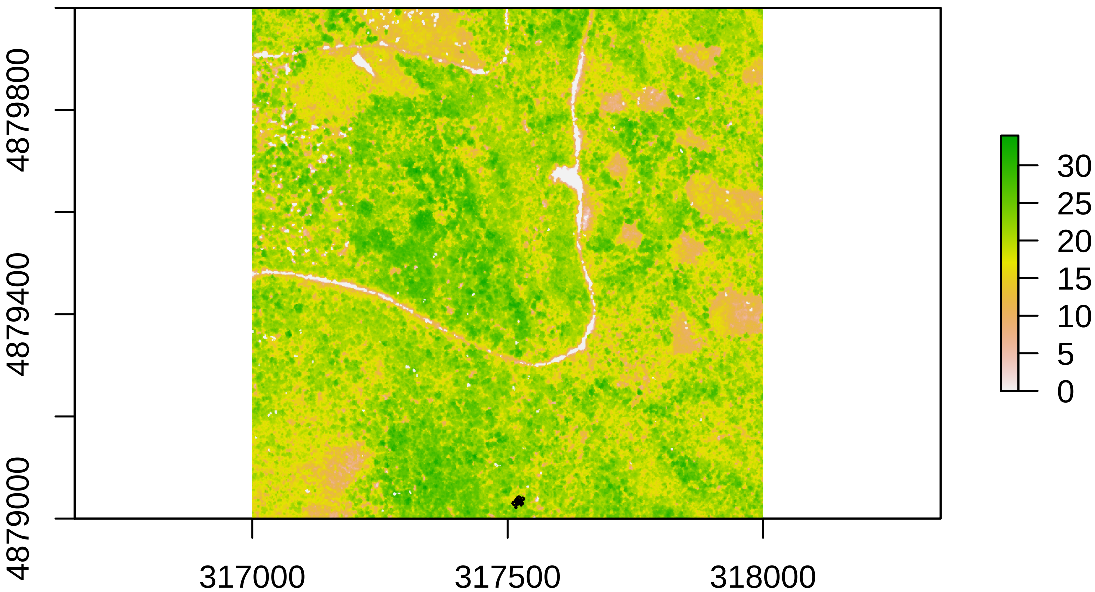
```r
yoi <- "2018" # years of interest

# Download Canopy Height Model from NEON API
byTileAOP(dpID="DP3.30015.001", # CHM
          site= "BART",
          year= yoi,
          easting = vegbart$adjEasting,
          northing = vegbart$adjNorthing,
          check.size=FALSE,
          token = NEON_TOKEN)

byTileAOP(dpID="DP3.30015.001", # CHM
          site= "GUAN",
          year= yoi,
          easting = vegguan$adjEasting,
          northing = vegguan$adjNorthing,
          check.size=FALSE,
          token = NEON_TOKEN)

# Download High-resolution orthorectified camera imagery mosaic from NEON API
byTileAOP(dpID="DP3.30010.001", # imagery
          site="BART",
          year = yoi,
          easting = vegbart$adjEasting,
          northing = vegbart$adjNorthing,
          check.size=FALSE,
          token = NEON_TOKEN)

byTileAOP(dpID="DP3.30010.001", # imagery
          site="GUAN",
          year = yoi,
          easting = vegguan$adjEasting,
```

```
          northing = vegguan$adjNorthing,
          check.size=FALSE,
          token = NEON_TOKEN)
```

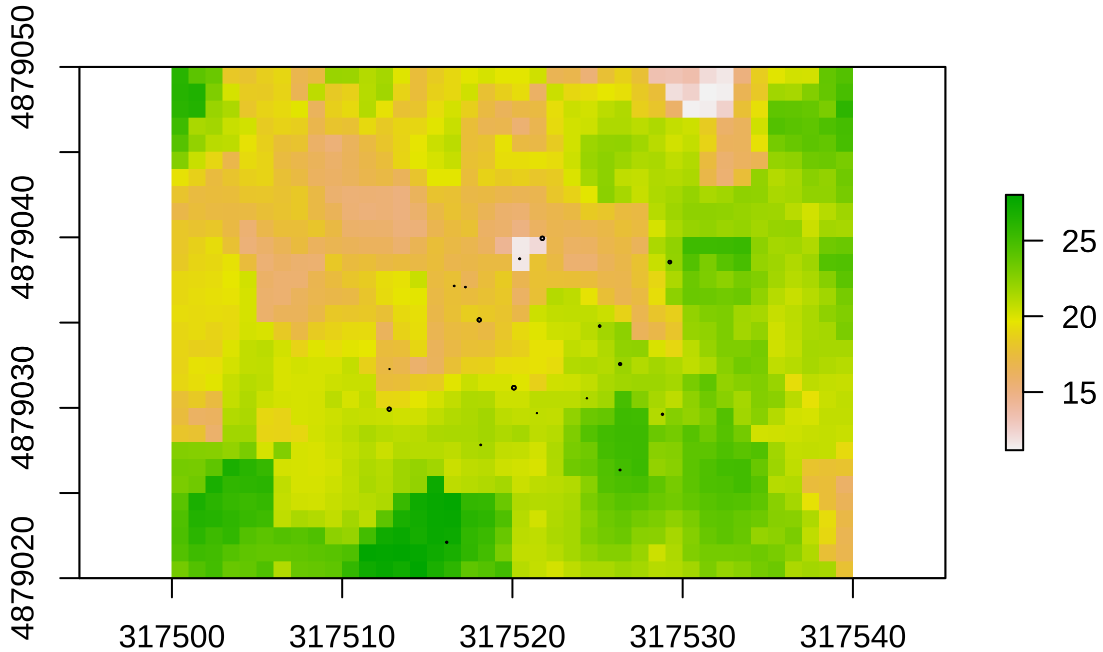
```
chmbart <- raster('./DP3.30015.001/2018/FullSite/D01/2018_BART_4/L3/DiscreteLidar/CanopyHeightModelGtif/

chmguan <- raster('./DP3.30015.001/2018/FullSite/D04/2018_GUAN_1/L3/DiscreteLidar/CanopyHeightModelGtif/
```

First I'll plot BART

```
# PLot CHM BART
plot(chmbart)
points(vegbart$adjEasting,vegbart$adjNorthing, pch=1, cex=vegbart$stemDiameter/100/2)
```

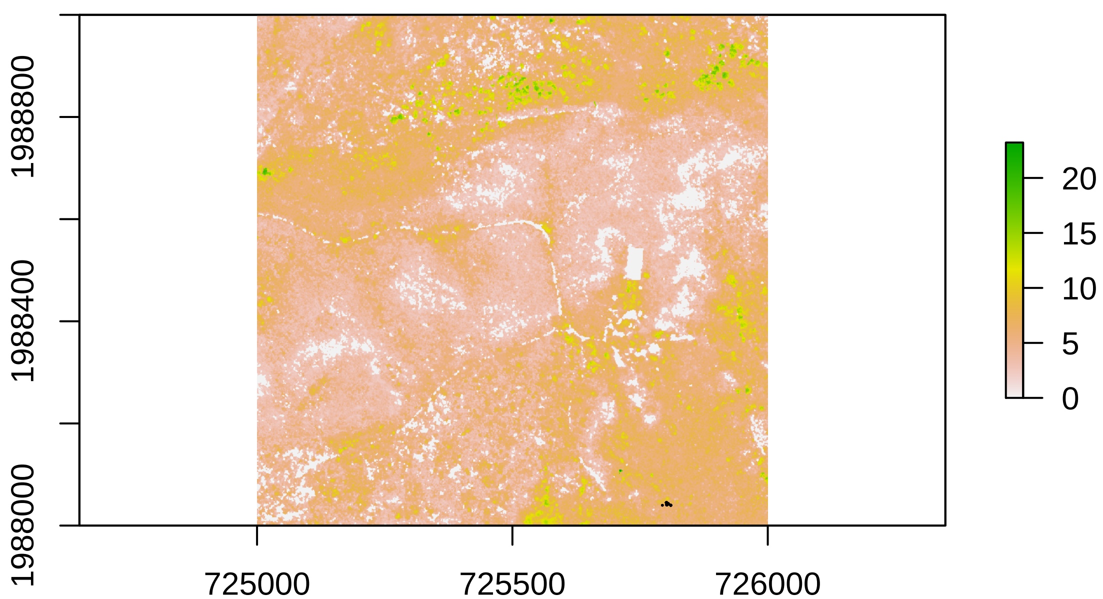



```
# Let's zoom in
plot(chmbart, xlim=c(317500,317540), ylim=c(4879020,4879050))
points(vegbart$adjEasting,vegbart$adjNorthing, pch=1, cex=vegbart$stemDiameter/100/2)
```

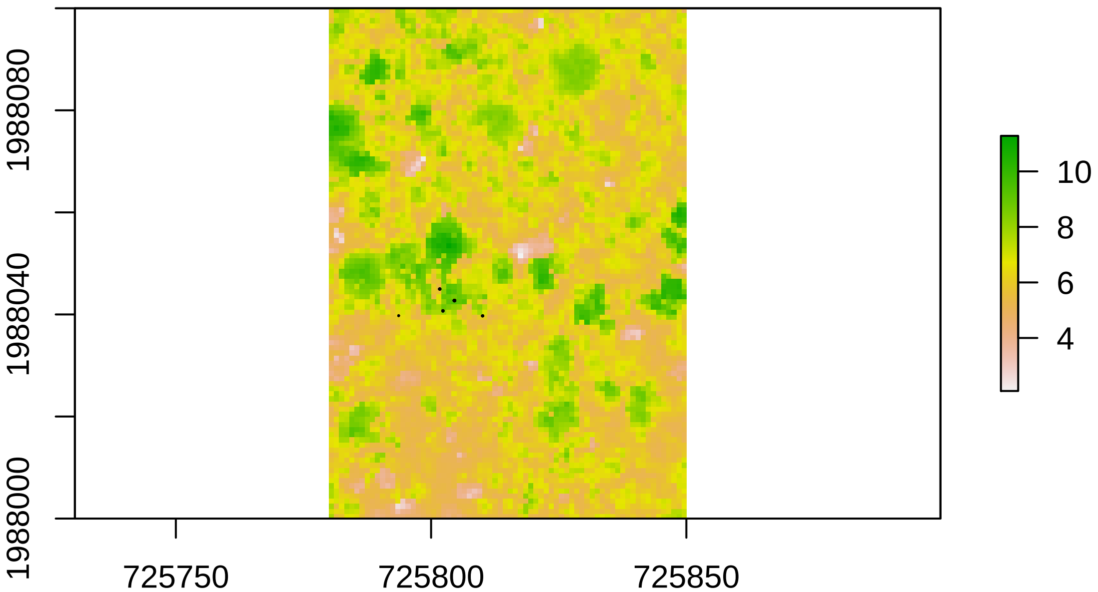


Now I'll plot GUAN

```r
# Plot CHM GUAM
plot(chmguan)
points(vegguan$adjEasting,vegguan$adjNorthing, pch=1, cex=vegguan$stemDiameter/100/2)
```



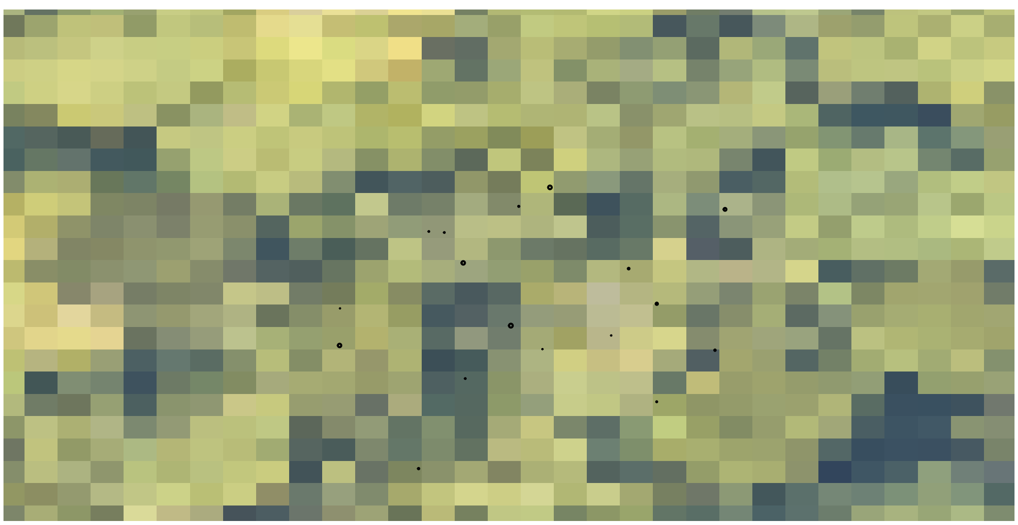

```r
# Let's zoom in
plot(chmguan, xlim=c(725780,725850), ylim=c(1988000,1988100))
points(vegguan$adjEasting,vegguan$adjNorthing, pch=1, cex=vegguan$stemDiameter/100/2)
```




```
imagebart <-brick('./DP3.30010.001/2018/FullSite/D01/2018_BART_4/L3/Camera/Mosaic/2018_BART_4_317000_48

imageguan <- brick('./DP3.30010.001/2018/FullSite/D04/2018_GUAN_1/L3/Camera/Mosaic/2018_GUAN_1_725000_19
```

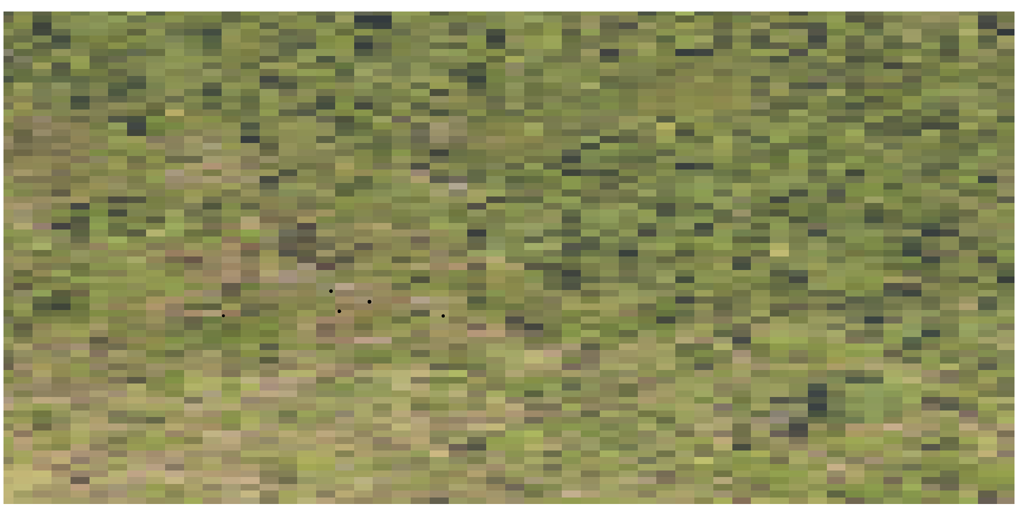
```
# Plot high-res camera imagery for BART

plotRGB(imagebart, r=1,g=2,b=3)
points(vegbart$adjEasting,vegbart$adjNorthing, pch=1, cex=vegbart$stemDiameter/100/2)
```

```
# Zoom in
plot.new()
plot.window(xlim=c(317500,317540), ylim=c(4879020,4879050))
plotRGB(imagebart, r=1,g=2,b=3, add=TRUE)
points(vegbart$adjEasting,vegbart$adjNorthing, pch=1, cex=vegbart$stemDiameter/100/2)
```



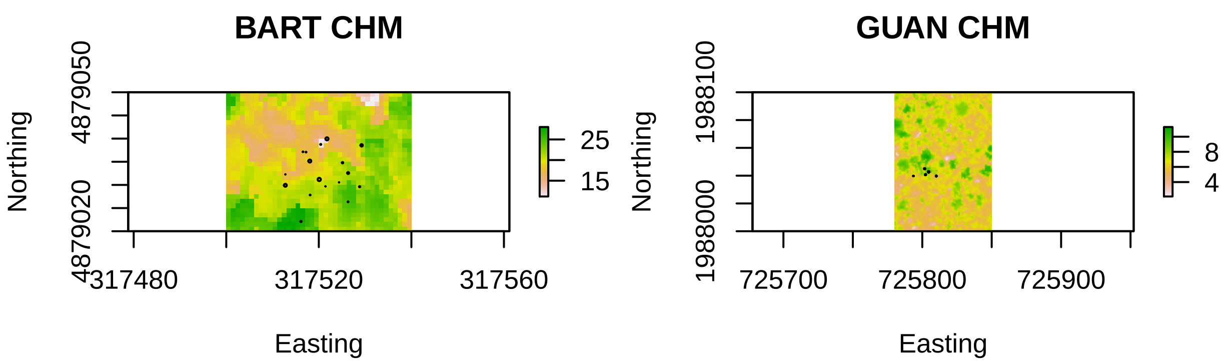
```
# PLot high-res imagery for GUAN
plotRGB(imageguan, r=1,g=2,b=3)
points(vegguan$adjEasting,vegguan$adjNorthing, pch=1, cex=vegguan$stemDiameter/100/2)
```

```
# Zoom in
plot.new()
plot.window(xlim=c(725780,725850), ylim=c(1988000,1988100))
plotRGB(imageguan, r=1,g=2,b=3, add=TRUE)
points(vegguan$adjEasting,vegguan$adjNorthing, pch=1, cex=vegguan$stemDiameter/100/2)
```



```
par(mfrow=c(2,2))

plot(chmbart, xlim=c(317500,317540), ylim=c(4879020,4879050), main="BART CHM", xlab = "Easting", ylab =
points(vegbart$adjEasting,vegbart$adjNorthing, pch=1, cex=vegbart$stemDiameter/100/2)
```

```
plot(chmguan, xlim=c(725780,725850), ylim=c(1988000,1988100), main="GUAN CHM", xlab = "Easting", ylab =
points(vegguan$adjEasting,vegguan$adjNorthing, pch=1, cex=vegguan$stemDiameter/100/2)

plot.new()
plot.window(xlim=c(317500,317540), ylim=c(4879020,4879050))
plotRGB(imagebart, r=1,g=2,b=3, axes = TRUE, main="BART Image", add=TRUE)
points(vegbart$adjEasting,vegbart$adjNorthing, pch=1, cex=vegbart$stemDiameter/100/2)

plot.new()
plot.window(xlim=c(725780,725850), ylim=c(1988000,1988100))
plotRGB(imageguan, r=1,g=2,b=3, axes = TRUE, main="GUAN Image", add=TRUE)
points(vegguan$adjEasting,vegguan$adjNorthing, pch=1, cex=vegguan$stemDiameter/100/2)
```



Write 2-3 sentences summarizing your findings and thoughts.

These AOP data seem too coarse to represent individual trees well by itself. It seems that the veg structure data could potentially help us identity individual trees in the AOP data. Perhaps if we were doubtful of the AOP data quality for canopy height, we could use the TOS measurements are a way to validate or correct our data using a linear prediction model.

3: Use the `byTileAOP` function of the `neonUtilities` package to pull a subset of the discrete LiDAR pointcloud for each forest (*Hint: You can feed `byTileAOP` Easting and Northing from your `Vegetation Structure` dataframe(s)*). Use the `structural_diversity_metrics` function that you defined in section 5.6 of the textbook to process discrete return LiDAR for each site and generate structural diversity metrics.

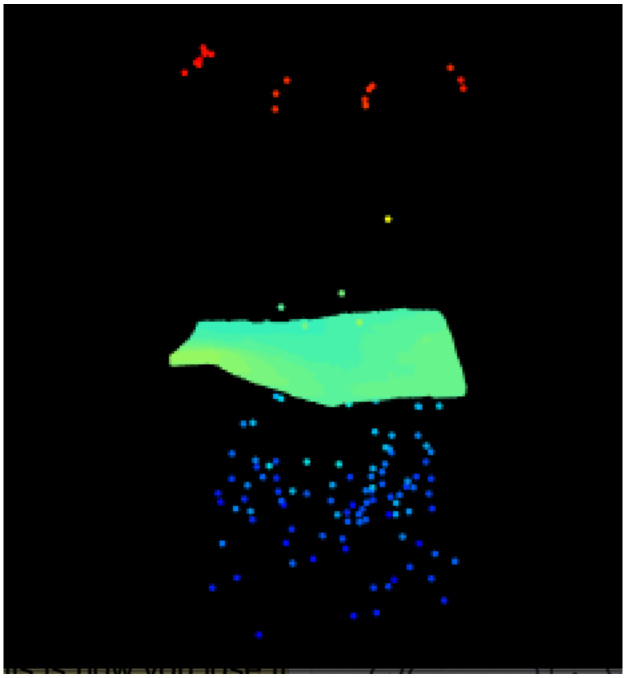
- Using `lidR` generate a labeled 2-panel plot of your canopy height model for each forest
- Using `lidR` generate a labeled 2-panel plot of a cross-section for each forest
- Use section `5.6.3 Comparing Metrics Between Forests` to compare each forest and generate a clean summary table via `kable`
- Using **Table 2** from LaRue, Wagner, et al. (2020) as a reference, write 1-2 paragraphs summarizing the differences in forest structural diversity between the two forests and how they may relate to your answers to **Question 1**.

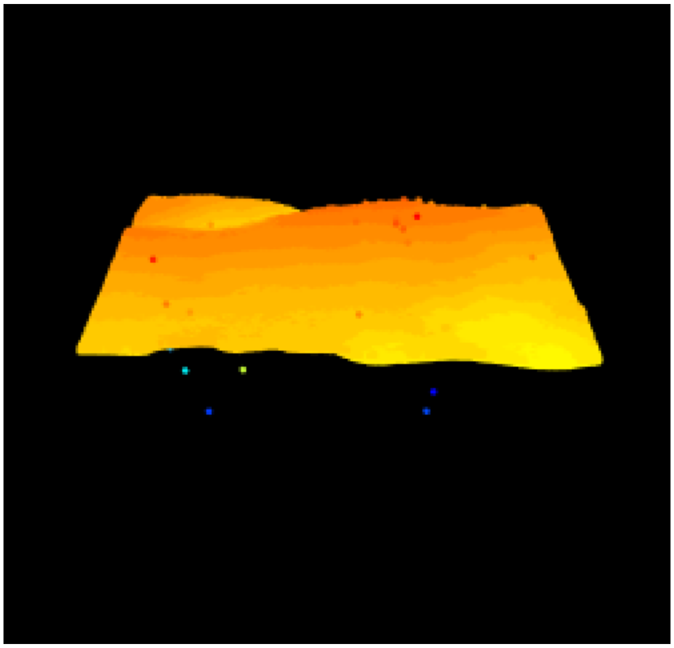
```
byTileAOP(dpID="DP1.30003.001",
          site= "BART",
          year= yoi,
          easting = extent(chmbart)[1],
          northing = extent(chmbart)[3],
          check.size=FALSE,
          token = NEON_TOKEN)

byTileAOP(dpID="DP1.30003.001",
          site= "GUAN",
          year= yoi,
          easting = extent(chmguan)[1],
          northing = extent(chmguan)[3],
          check.size=FALSE,
          token = NEON_TOKEN)
```

```
pcbart <- lidR::readLAS('./DP1.30003.001/2018/FullSite/D01/2018_BART_4/L1/DiscreteLidar/ClassifiedPoint
pcguan <- lidR::readLAS('./DP1.30003.001/2018/FullSite/D04/2018_GUAN_1/L1/DiscreteLidar/ClassifiedPoint
```

```
lidR::plot(pcbart)
```

```
lidR::plot(pcguan)
```

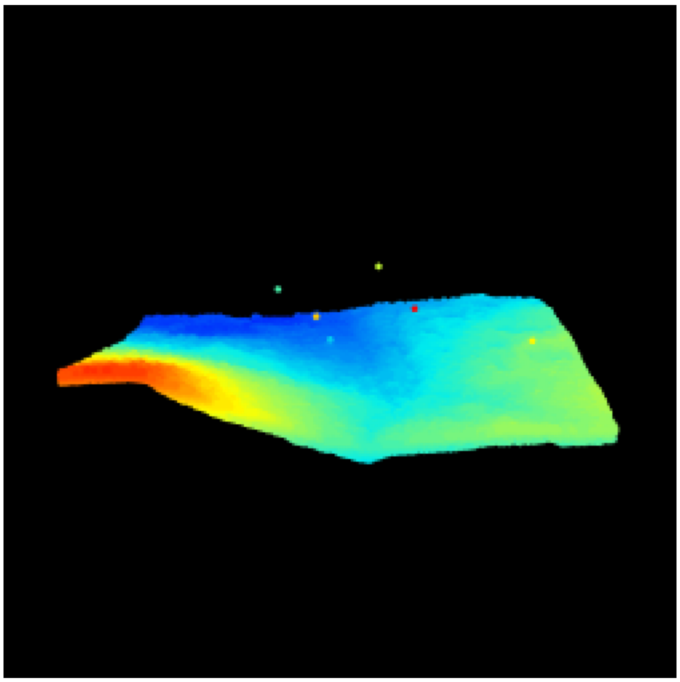
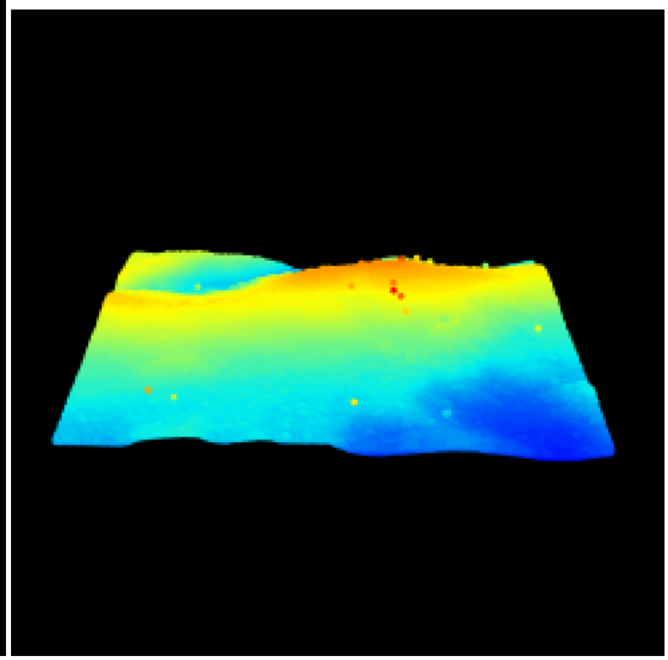




```
# remove outlier lidar point outliers using mean and sd statistics
Z_sd_bart=sd(pcbart@data$Z)
Z_mean_bart=mean(pcbart@data$Z)

Z_sd_guan=sd(pcguan@data$Z)
Z_mean_guan=mean(pcguan@data$Z)

# make filter string in form filter = "-drop_z_below 50 -drop_z_above 1000"
# You can increase or decrease (from 4) the number of sd's to filter outliers
fbart = paste("-drop_z_below",(Z_mean_bart-4*Z_sd_bart),"-drop_z_above",(Z_mean_bart+4*Z_sd_bart))
```

```r
fguan = paste("-drop_z_below",(Z_mean_guan-4*Z_sd_guan),"-drop_z_above",(Z_mean_guan+4*Z_sd_guan))

# Read in LAS file, trimming off vertical outlier points
pcbart = readLAS('./DP1.30003.001/2018/FullSite/D01/2018_BART_4/L1/DiscreteLidar/ClassifiedPointCloud/N

pcguan = readLAS('./DP1.30003.001/2018/FullSite/D04/2018_GUAN_1/L1/DiscreteLidar/ClassifiedPointCloud/N
```

```r
lidR::plot(pcbart)
lidR::plot(pcguan)
```

```r
# Check summaries to get x and y
summary(pcbart)
```

```
## class        : LAS (v1.3 format 1)
## memory       : 554.6 Mb
## extent       : 317000, 318000, 4879000, 4880000 (xmin, xmax, ymin, ymax)
## coord. ref.  : +proj=utm +zone=19 +datum=WGS84 +units=m +no_defs
## area         : 1 km²
## points       : 6.92 million points
## density      : 6.92 points/m²
## File signature:           LASF
## File source ID:           0
## Global encoding:
##  - GPS Time Type: GPS Week Time
##  - Synthetic Return Numbers: no
##  - Well Know Text: CRS is GeoTIFF
##  - Aggregate Model: false
## Project ID - GUID:        00000000-0000-0000-0000-000000000000
## Version:                  1.3
## System identifier:        LAStools (c) by rapidlasso GmbH
## Generating software:      lasmerge (version 160730)
## File creation d/y:        259/2018
```

```
## header size:              235
## Offset to point data:     575
## Num. var. length record:  2
## Point data format:        1
## Point data record length: 32
## Num. of point records:    6922789
## Num. of points by return: 4391402 1938270 510226 82891 0
## Scale factor X Y Z:       0.01 0.01 0.01
## Offset X Y Z:             3e+05 4800000 0
## min X Y Z:                317000 4879000 313.98
## max X Y Z:                318000 4880000 603.3
## Variable length records:
##    Variable length record 1 of 2
##        Description:
##        Tags:
##           Key 1024 value 1
##           Key 1025 value 2
##           Key 3072 value 32619
##           Key 4099 value 9001
##    Variable length record 2 of 2
##        Description: by LAStools of rapidlasso GmbH
##        Extra Bytes Description:
##           reversible index (lastile): original index before tiling
```

```r
summary(pcguan)
```

```
## class        : LAS (v1.3 format 1)
## memory       : 563.7 Mb
## extent       : 725000, 726000, 1988000, 1989000 (xmin, xmax, ymin, ymax)
## coord. ref.  : +proj=utm +zone=19 +datum=WGS84 +units=m +no_defs
## area         : 1 km²
## points       : 7.04 million points
## density      : 7.04 points/m²
## File signature:           LASF
## File source ID:           0
## Global encoding:
##  - GPS Time Type: GPS Week Time
##  - Synthetic Return Numbers: no
##  - Well Know Text: CRS is GeoTIFF
##  - Aggregate Model: false
## Project ID - GUID:        00000000-0000-0000-0000-000000000000
## Version:                  1.3
## System identifier:        LAStools (c) by rapidlasso GmbH
## Generating software:      lasmerge (version 160730)
## File creation d/y:        178/2018
## header size:              235
## Offset to point data:     575
## Num. var. length record:  2
## Point data format:        1
## Point data record length: 32
## Num. of point records:    7035964
## Num. of points by return: 5062553 1874746 97456 1209 0
## Scale factor X Y Z:       0.01 0.01 0.01
## Offset X Y Z:             7e+05 1900000 0
## min X Y Z:                725000 1988000 108.4
```

```
## max X Y Z:                726000 1989000 245.28
## Variable length records:
##    Variable length record 1 of 2
##        Description:
##        Tags:
##           Key 1024 value 1
##           Key 1025 value 2
##           Key 3072 value 32619
##           Key 4099 value 9001
##    Variable length record 2 of 2
##        Description: by LAStools of rapidlasso GmbH
##        Extra Bytes Description:
##           reversible index (lastile): original index before tiling
```

```
## BART
# Correct for elevation
# Choose a 40 x 40 m spatial extent, which is the extent for NEON base plots.
# First set the center of where you want the plot to be
xbart <- 317500 #easting
ybart <- 4879500 #northing

#Cut out a 200 x 200 m buffer by adding 100 m to easting and
#northing coordinates (x,y).
data.200m.b <-
   lasclipRectangle(pcbart,
                    xleft = (xbart - 100), ybottom = (ybart - 100),
                    xright = (xbart + 100), ytop = (ybart + 100))

# Correct for ground height using a kriging function to interpolate
#elevation from ground points in the .laz file.
dtm.b <- grid_terrain(data.200m.b, 1, kriging(k = 10L))
data.200m.b <- lasnormalize(data.200m.b, dtm.b)

#Will often give a warning if not all points could be corrected,
#but visually check to see if it corrected for ground height.
lidR::plot(data.200m.b)
#There's a few uncorrected points

#Clip 20 m out from each side of the easting and northing
#coordinates (x,y).
data.40m.b <-
   lasclipRectangle(data.200m.b,
                    xleft = (xbart - 20), ybottom = (ybart - 20),
                    xright = (xbart + 20), ytop = (ybart + 20))

data.40m.b@data$Z[data.40m.b@data$Z <= .5] <- NA
#This line filters out all z_vals below .5 m as we are less interested in shrubs/trees.
```

```
# visualize the clipped plot point cloud
lidR::plot(data.40m.b)
```

```
## GUAN
# Correct for elevation
# Choose a 40 x 40 m spatial extent, which is the extent for NEON base plots.
# First set the center of where you want the plot to be
```

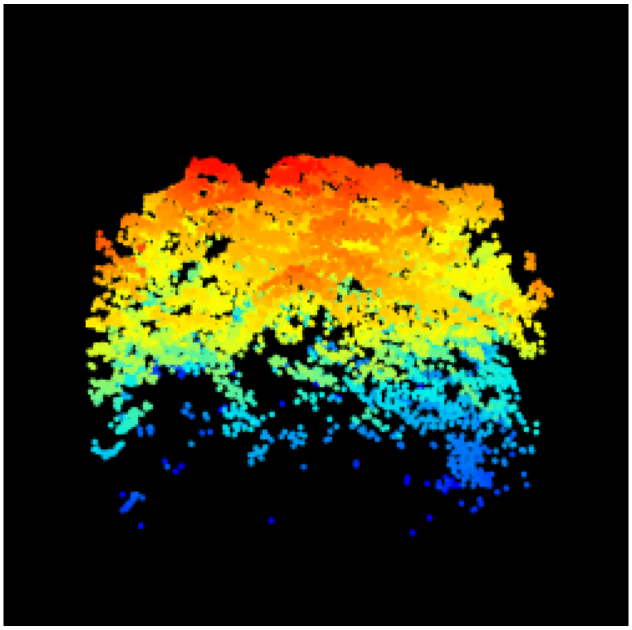
```r
xguan <- 725500 #easting
yguan <- 1988500 #northing

# Cut out a 200 x 200 m buffer by adding 100 m to easting and
#northing coordinates (x,y).
data.200m.g <-
   lasclipRectangle(pcguan,
                    xleft = (xguan - 100), ybottom = (yguan - 100),
                    xright = (xguan + 100), ytop = (yguan + 100))

# Correct for ground height using a kriging function to interpolate
#elevation from ground points in the .laz file.
dtm.g <- grid_terrain(data.200m.g, 1, kriging(k = 10L))
```

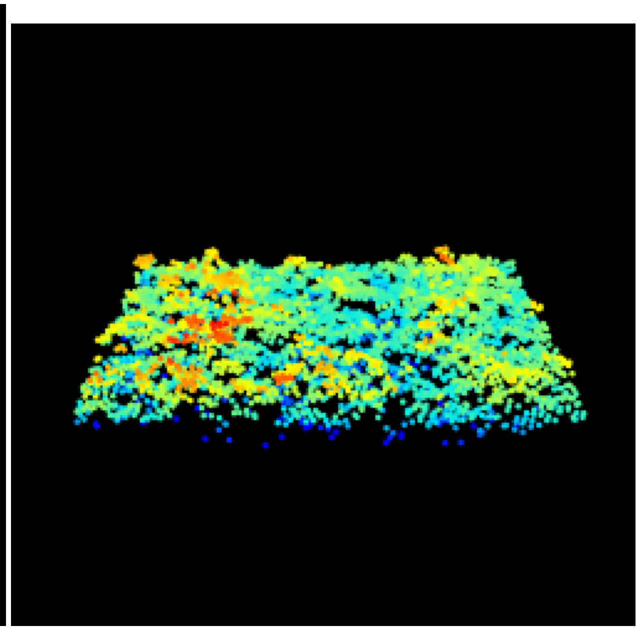
```
## Warning: There were 23 degenerated ground points. Some X Y coordinates were
## repeated but with different Z coordinates. min Z were retained.
```

```r
data.200m.g <- lasnormalize(data.200m.g, dtm.g)

#Will often give a warning if not all points could be corrected,
#but visually check to see if it corrected for ground height.
lidR::plot(data.200m.g)
#There's a few uncorrected points

#Clip 20 m out from each side of the easting and northing
#coordinates (x,y).
data.40m.g <-
   lasclipRectangle(data.200m.g,
                    xleft = (xguan - 20), ybottom = (yguan - 20),
                    xright = (xguan + 20), ytop = (yguan + 20))

data.40m.g@data$Z[data.40m.g@data$Z <= .5] <- NA
# This line filters out all z_vals below .5 m
```

```r
# visualize the clipped plot point cloud
lidR::plot(data.40m.g)
```

BART cross section (left) and GUAN cross section (right):

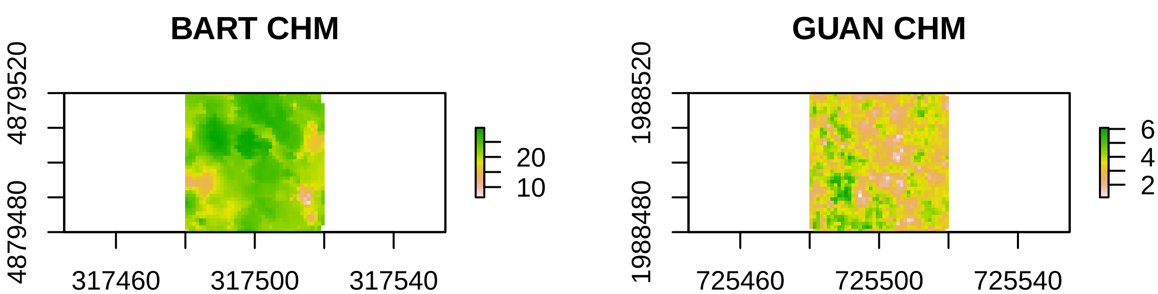




CHM plots:

```
chm.b <- grid_canopy(data.40m.b, res = 1, dsmtin())
chm.g <- grid_canopy(data.40m.g, res = 1, dsmtin())
```

```
par(mfrow=c(2,2))
lidR::plot(chm.b, main = "BART CHM")
lidR::plot(chm.g, main = "GUAN CHM")
```




```
# write function to run all 13 veg structure metrics in a single function.
structural_diversity_metrics <- function(data.40m,x,y) {
   chm <- grid_canopy(data.40m, res = 1, dsmtin())
   mean.max.canopy.ht <- mean(chm@data@values, na.rm = TRUE)
   max.canopy.ht <- max(chm@data@values, na.rm=TRUE)
   rumple <- rumple_index(chm) #calculate rumple, a ratio of outer canopy surface area to ground surfac
   top.rugosity <- sd(chm@data@values, na.rm = TRUE) #top rugosity, the standard deviation of pixel val
   cells <- length(chm@data@values)
   chm.0 <- chm
   chm.0[is.na(chm.0)] <- 0
   zeros <- which(chm.0@data@values == 0)
   deepgaps <- length(zeros) #number of deep gaps
   deepgap.fraction <- deepgaps/cells #deep gap fraction, the number of deep gaps in the chm relative t
   cover.fraction <- 1 - deepgap.fraction #cover fraction, the inverse of deep gap fraction
   vert.sd <- cloud_metrics(data.40m, sd(Z, na.rm = TRUE)) #height SD, the standard deviation of height
```

```r
  sd.1m2 <- grid_metrics(data.40m, sd(Z), 1)
  sd.sd <- sd(sd.1m2[,3], na.rm = TRUE) # The standard deviation of height values
  Zs <- data.40m@data$Z
  Zs <- Zs[!is.na(Zs)]
  entro <- entropy(Zs, by = 1) # Quantifies diversity & evenness of point cloud heights
  gap_frac <- gap_fraction_profile(Zs, dz = 1, z0=3)
  GFP.AOP <- mean(gap_frac$gf) # Gap fraction profile, assesses the distribution of gaps in the canopy
  LADen<-LAD(Zs, dz = 1, k=0.5, z0=3)
  VAI.AOP <- sum(LADen$lad, na.rm=TRUE) # Leaf area density
  VCI.AOP <- VCI(Zs, by = 1, zmax=100) # A vertical complexity index
  out.plot <- data.frame(
    matrix(c(x, y, mean.max.canopy.ht,max.canopy.ht,
             rumple,deepgaps, deepgap.fraction,
             cover.fraction, top.rugosity, vert.sd,
             sd.sd, entro, GFP.AOP, VAI.AOP,VCI.AOP),
           ncol = 15))
  colnames(out.plot) <-
    c("easting", "northing", "mean.max.canopy.ht.aop",
      "max.canopy.ht.aop", "rumple.aop", "deepgaps.aop",
      "deepgap.fraction.aop", "cover.fraction.aop",
      "top.rugosity.aop","vert.sd.aop","sd.sd.aop",
      "entropy.aop", "GFP.AOP.aop",
      "VAI.AOP.aop", "VCI.AOP.aop")
  print(out.plot)
}
```

```r
BART_structural_diversity <- structural_diversity_metrics(data.40m.b,xbart,ybart)
```

```
##   easting northing mean.max.canopy.ht.aop max.canopy.ht.aop rumple.aop
## 1  317500  4879500                22.7105          29.67878   1.857444
##   deepgaps.aop deepgap.fraction.aop cover.fraction.aop top.rugosity.aop
## 1            5             0.003125           0.996875         3.278998
##   vert.sd.aop sd.sd.aop entropy.aop GFP.AOP.aop VAI.AOP.aop VCI.AOP.aop
## 1    5.388723  1.213976   0.8902085   0.8244346    10.98084   0.6574729
```

```r
GUAN_structural_diversity <- structural_diversity_metrics(data.40m.g,xguan,yguan)
```

```
##   easting northing mean.max.canopy.ht.aop max.canopy.ht.aop rumple.aop
## 1  725500  1988500               3.413074          6.080559   1.281618
##   deepgaps.aop deepgap.fraction.aop cover.fraction.aop top.rugosity.aop
## 1            3             0.001875           0.998125         0.761247
##   vert.sd.aop sd.sd.aop entropy.aop GFP.AOP.aop VAI.AOP.aop VCI.AOP.aop
## 1   0.9552243 0.4976439   0.7231822     0.85184    1.474833   0.3055799
```

```r
combined_results=rbind(BART_structural_diversity,
                       GUAN_structural_diversity)

# Add row names for clarity
row.names(combined_results)=c("BART","GUAN")

# Take a look to compare
combined_results
```

```
##      easting northing mean.max.canopy.ht.aop max.canopy.ht.aop rumple.aop
## BART  317500  4879500              22.710504         29.678781   1.857444
```

```
## GUAN  725500  1988500              3.413074          6.080559   1.281618
##      deepgaps.aop deepgap.fraction.aop cover.fraction.aop top.rugosity.aop
## BART           5             0.003125           0.996875         3.278998
## GUAN           3             0.001875           0.998125         0.761247
##      vert.sd.aop sd.sd.aop entropy.aop GFP.AOP.aop VAI.AOP.aop VCI.AOP.aop
## BART   5.3887227 1.2139760   0.8902085   0.8244346   10.980836   0.6574729
## GUAN   0.9552243 0.4976439   0.7231822   0.8518400    1.474833   0.3055799
```

BART forest has a much taller canopy than GUAN. The leaf area density at BART is also much higher, which follows my initial assumption of denseness based on the leaf litter in the video. The rugosity and vertical standard deviation of BART is higher, which likely means there are more types of species interdispersed within the same space compared to GUAN. I would predict from this assessment that the tree species at BART specialize at different light levels, which is a common life strategy in dense northeastern forests. In comparison, the species at GUAN likely do not have to compete for light because of the sparser canopy, so there is more homogeneity in tree height.