# Profile curves

Robert J. McGaughey

4/22/2020

```
# load libraries
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(webshot)
library(lidR)
library(fusionwrapr)
```

## Profile Area

Setup and read data. Tree point data are in FUSION's LDA format so we have to use readLDA() to read points into an LAS object for the lidR package. The tree points have been normalized relative to ground.

Compute the percentile heights in 1% steps. These are used to find the maximum slope of the percentile height line.

```
FilePath <- "../extras/"

# read point cloud for individual tree...points have height in Z, not elevation.
# Example tree is in FUSION's LDA format, not LAS or LAZ. readLDA() converts the data into LAS object
# If you have tree points in LAS format, use readLAS() function from lidR to read points.
#LAS <- readLDA(paste0(FilePath, "trees_normalized_Clip_0000011.lda"), type = "LAS", LASTemplate = "../
LAS <- readLDA(paste0(FilePath, "trees_Clip_0001510.lda"), type = "LAS", LASTemplate = "../extras/DataF
```

```r
# compute the percentile data
PercentileData <- quantile(LAS$Z, seq(0, 1, 0.01))

# wipe out the names
names(PercentileData) <- NULL
```

## Data Processing

Normalize the percentile heights using the 99th percentile. This gives us heights that range from 0 to 1.0.

```r
# percentile heights start in column 1 and P99 is column 100
NormalizedPercentileData <- PercentileData / PercentileData[100]
Label <- "Tree_0000011"
```

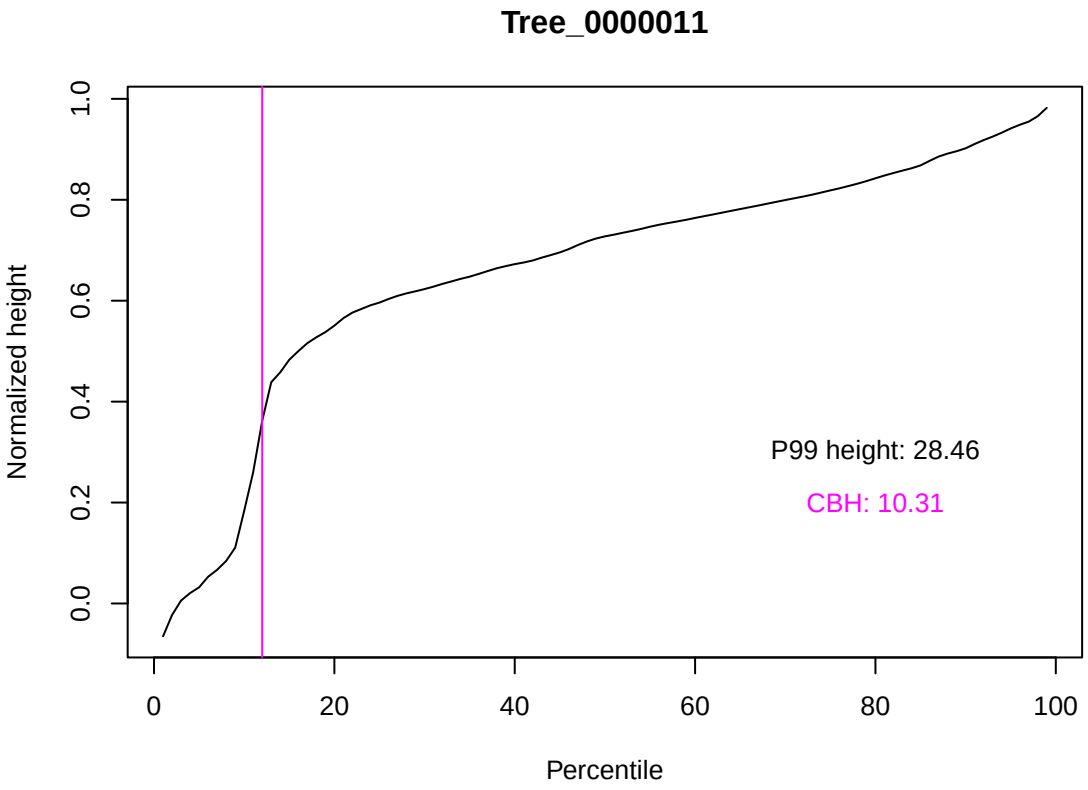
## Plots

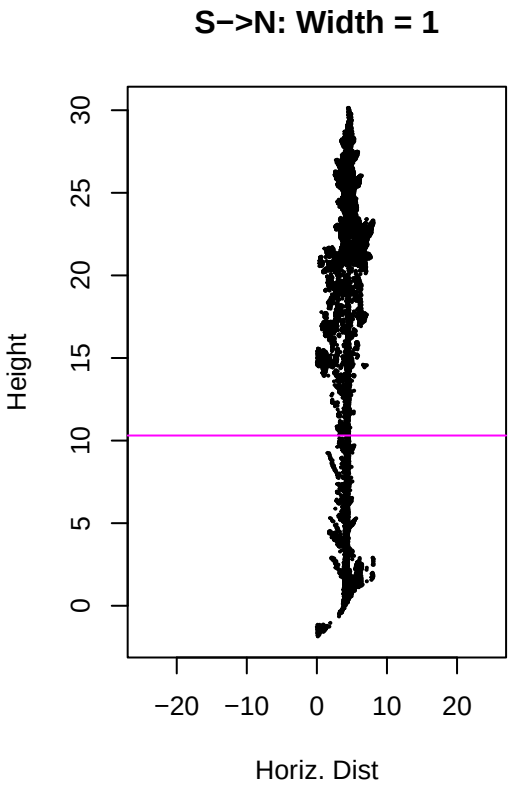
Find the maximum slope and scale this back to actual height. Then, build the plots. First plot shows the relative heights and second two plots show cross sections through the point cloud with the CBH labeled. Note that the y-axis scaling for the cross section plots may be different making the CBH line look like it is at a different height in each graph.

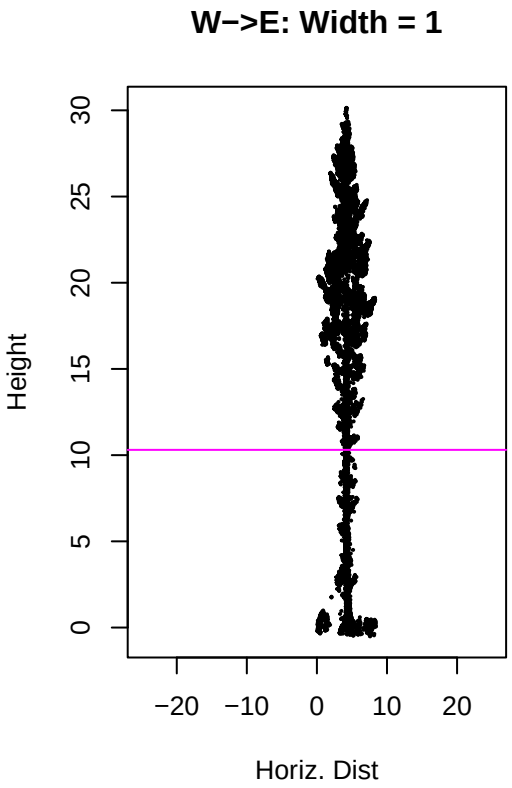
```r
CorridorWidth <- 1

# compute slopes using normalized percentile data...ignore the 1st percentile so we
# don't include ground points. However, on steep slopes, we may need to ignore more
# points close to the ground.
x <- c(1:99)
slopes <- vector()
for (j in 2:99) {
  x1 <- x[j - 1]
  x2 <- x[j]
  y1 <- NormalizedPercentileData[j -1]
  y2 <- NormalizedPercentileData[j]
  slope_i <- (y2-y1)/(x2-x1)
  slopes <- append(slopes, slope_i)
}

# get max slope and scale back to actual height from original PercentileData
maxSlopeIndex <- which.max(slopes)
cbhIndex <- maxSlopeIndex + 1
cbh <- PercentileData[cbhIndex]

# set up for graphs
nf <- layout( matrix(c(1,1,2,3), nrow=2, byrow=TRUE) )

plot(x = c(1:99),
     y = NormalizedPercentileData[1:99],
     type = "l",
     xlab = "Percentile",
     ylab = "Normalized height",
     main = Label
```

```
)

# add vertical line for cbhIndex
abline(v = x[cbhIndex], col = "magenta")

text(80, 0.3, paste("P99 height:", round(PercentileData[100], 2)))
text(80, 0.2, paste("CBH:", round(cbh, 2)), col = "magenta")

# load lidar data for plot
#  PlotLAS <- readLAS(NormalizedPercentileData[i, "Plot"])
PlotLAS <- LAS

# get center of plot
#centerX <- (PlotLAS@bbox[1,2] + PlotLAS@bbox[1,1]) / 2
#centerY <- (PlotLAS@bbox[2,2] + PlotLAS@bbox[2,1]) / 2
centerX <- (PlotLAS@header$`Min X` + PlotLAS@header$`Max X`) / 2
centerY <- (PlotLAS@header$`Min Y` + PlotLAS@header$`Max Y`) / 2

# clip a N-S transect and plot
NStransect <- clip_rectangle(PlotLAS,
                      xleft = centerX - CorridorWidth / 2,
                      ybottom = PlotLAS@header$`Min Y`,
                      xright = centerX + CorridorWidth / 2,
                      ytop = PlotLAS@header$`Max Y`)

plot(x = NStransect@data$Y - PlotLAS@header$`Min Y`,
     y = NStransect@data$Z,
     type = "p",
     xlim = c(-25, 25),
     pch = 20,
     cex = .25,
     main = paste("S->N: Width =", CorridorWidth),
     xlab = "Horiz. Dist",
     ylab = "Height")

# add horizontal lines
# abline(h = PercentileData[i, 6], col = "red")
# abline(h = PercentileData[i, 8], col = "blue")
# abline(h = PercentileData[i, 10], col = "green")
abline(h = cbh, col = "magenta")

# clip an E-W transect and plot
EWtransect <- clip_rectangle(PlotLAS,
                      xleft = PlotLAS@header$`Min X`,
                      ybottom = centerY - CorridorWidth / 2,
                      xright = PlotLAS@header$`Max X`,
                      ytop = centerY + CorridorWidth / 2)

plot(x = EWtransect@data$X - PlotLAS@header$`Min X`,
     y = EWtransect@data$Z,
     type = "p",
     xlim = c(-25, 25),
     pch = 20,
```

```
     cex = .25,
     main = paste("W->E: Width =", CorridorWidth),
     xlab = "Horiz. Dist",
     ylab = "Height")

# add horizontal lines
# abline(h = PercentileData[i, 6], col = "red")
# abline(h = PercentileData[i, 8], col = "blue")
# abline(h = PercentileData[i, 10], col = "green")
abline(h = cbh, col = "magenta")
```


Tree_0000011
Normalized height
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Percentile
P99 height: 28.46
CBH: 10.31


S−>N: Width = 1
Height
0
5
10
15
20
25
30
−20
−10
0
10
20
Horiz. Dist


W−>E: Width = 1
Height
0
5
10
15
20
25
30
−20
−10
0
10
20
Horiz. Dist