

# where to place TLS plots in BCI: (3) Make figures to evaluate plot placement options

Helene Muller-Landau

2023-09-11

## Background

This assumes have already run “whereTLSonBCI\_part1a.Rmd”, “whereTLSonBCI\_part1b.Rmd” and “whereTLSonBCI\_part2.Rmd” or have the products of those functions available (in directory usedata).

## Objectives

The objective of this document is to explore potential locations for three (or possibly more) 1-ha plots within the BCI 50-ha plot to be scanned with a Terrestrial Laser Scanner (TLS) for GEOTREES. These plots should represent the variation in forest structure within the site. Ideally forest structure is relatively homogeneous within the chosen plots, and overall the set of plots spans the variation in forest structure within the entire 50-ha plot, but with a bias towards capturing more bigger trees.

## Potential future additions

Get the density of large trees (>50 or 60 cm dbh) per quadrat in the analyses of plot data, and use that as well.

Update to using the 2023 50-ha plot data as far as possible, but this would require processing the preliminary data and merging across datasets.

Would be nice to include liana stem density and basal area per quadrat in the analyses as well; could request these from Stefan Schnitzer.

## Background code and output

### Make sure needed packages are available

Some R packages are used in the codes, and must be installed on the computer beforehand. The following code automatically installs the missing packages.

```
# list of package dependencies
req_packages <- c("rdryad", "data.table", "utils", "truncnorm", "openxlsx", "docxtractr", "readxl", "jpeg")

# packages that are not yet installed on the computer
ins_packages <- req_packages[!(req_packages %in% rownames(installed.packages()))]
```

```

# install missing packages
if (length(ins_packages) > 0)
  install.packages(ins_packages)

for (i in 1:length(req_packages))
  require(req_packages[i], character.only=T)

## Loading required package: rdryad

## Loading required package: data.table

## Loading required package: truncnorm

## Warning: package 'truncnorm' was built under R version 4.3.1

## Loading required package: openxlsx

## Warning: package 'openxlsx' was built under R version 4.3.1

## Loading required package: docxtractr

## Warning: package 'docxtractr' was built under R version 4.3.1

## Registered S3 method overwritten by 'httr':
##   method           from
##   print.cache_info hoardr

## Loading required package: readxl

## Warning: package 'readxl' was built under R version 4.3.1

## Loading required package: jpeg

## Loading required package: tiff

## Warning: package 'tiff' was built under R version 4.3.1

## Loading required package: fields

## Warning: package 'fields' was built under R version 4.3.1

## Loading required package: spam

## Warning: package 'spam' was built under R version 4.3.1

## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

```

```

## 
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##       backsolve, forwardsolve

## Loading required package: viridisLite

## 
## Try help(fields) to get started.

## Loading required package: vioplot

## Warning: package 'vioplot' was built under R version 4.3.1

## Loading required package: sm

## Warning: package 'sm' was built under R version 4.3.1

## Package 'sm', version 2.2-5.7: type help(sm) for summary information

## Loading required package: zoo

## 
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##       as.Date, as.Date.numeric

```

### Load and map subplots and layers of interest

Load the habitat data (moved up from later in code)

```

load("usedata/bci_habitat.rda")
# change q20 coordinates to match quadrat center
bci_habitat$x <- bci_habitat$x + 10
bci_habitat$y <- bci_habitat$y + 10

```

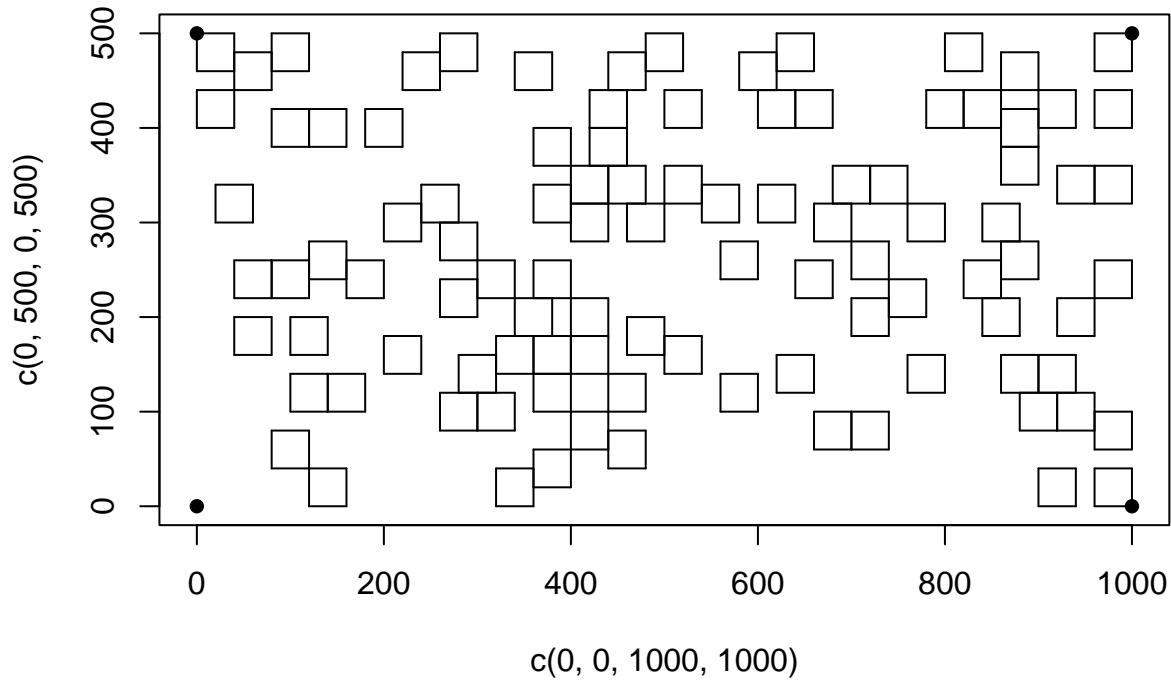
Load the centers of the 40x40s of dendrometer plots and make a map of them.

```

dend40centers <- read.table("usedata/bci40x40sxy.txt", header=T, sep="\t")
plot(c(0,0,1000,1000),c(0,500,0,500), pch=16, main="Dendrometer subplots")
for (i in 1:nrow(dend40centers)) {
  lines(x=rep(dend40centers$x[i]-20,2), y=c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(x=rep(dend40centers$x[i]+20,2), y=c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(x=c(dend40centers$x[i]-20,dend40centers$x[i]+20), y=rep(dend40centers$y[i]-20,2))
  lines(x=c(dend40centers$x[i]-20,dend40centers$x[i]+20), y=rep(dend40centers$y[i]+20,2))
}

```

## Dendrometer subplots

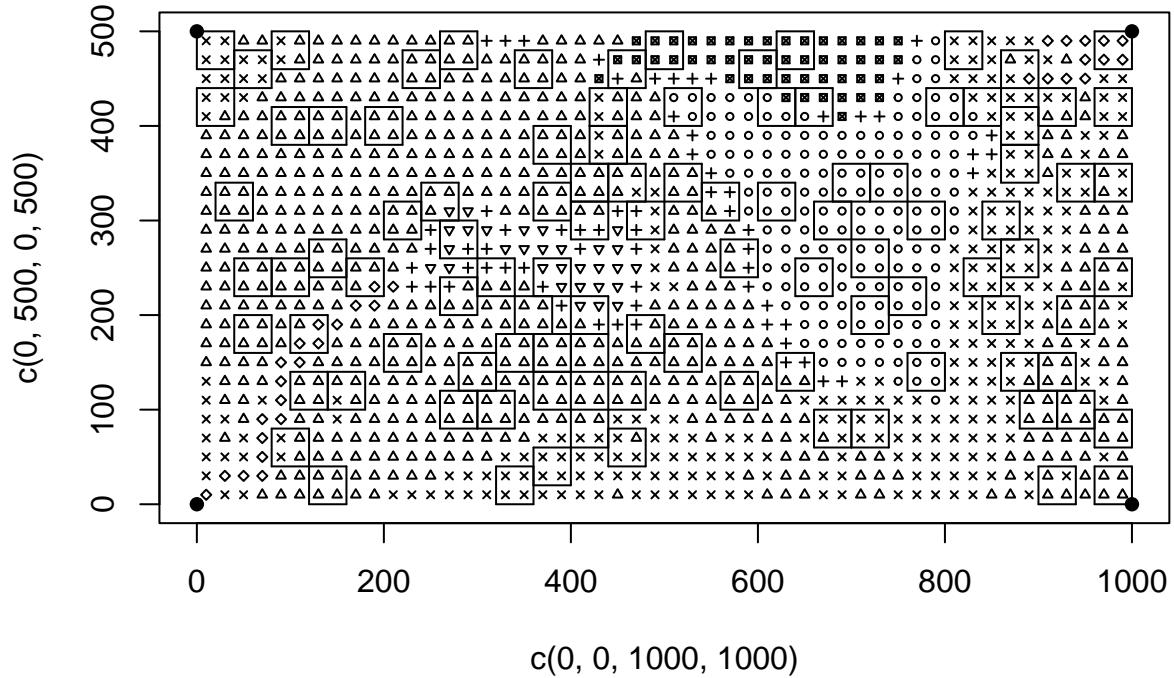


```

plot(c(0,0,1000,1000),c(0,500,0,500),pch=16, main = "Dendrometer subplots & Habitat class")
for (i in 1:nrow(dend40centers)) {
  lines(x=rep(dend40centers$x[i]-20,2), y=c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(x=rep(dend40centers$x[i]+20,2), y=c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(x=c(dend40centers$x[i]-20,dend40centers$x[i]+20), y=rep(dend40centers$y[i]-20,2))
  lines(x=c(dend40centers$x[i]-20,dend40centers$x[i]+20), y=rep(dend40centers$y[i]+20,2))
}
points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.5)

```

## Dendrometer subplots & Habitat class



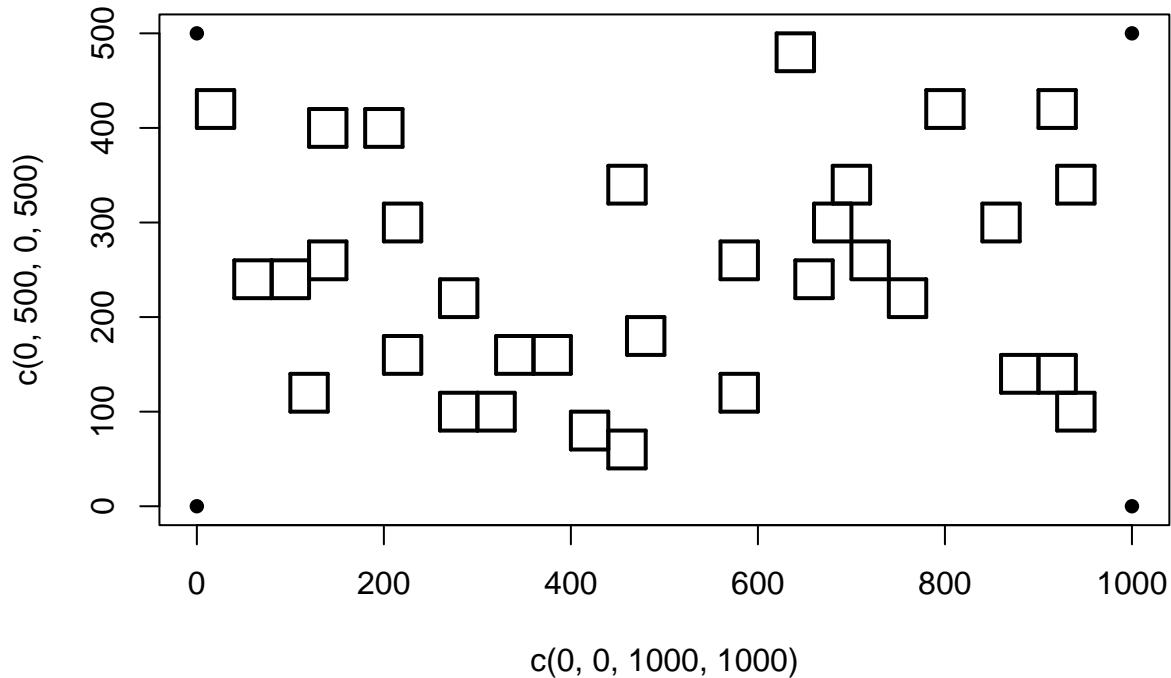
Load the centers of the 40x40s scanned by Sruthi Moorthy in 2019 (a subset of the dendrometer 40x40s) and make a map of them.

```

tls40centers <- read.table("usedata/bci_tls_2019_subplotcenters.txt", header=T, sep="\t")
plot(c(0,0,1000,1000),c(0,500,0,500),pch=16, main="TLS 2019 subplots")
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20), lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20), lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2), lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2), lwd=2)
}

```

## TLS 2019 subplots

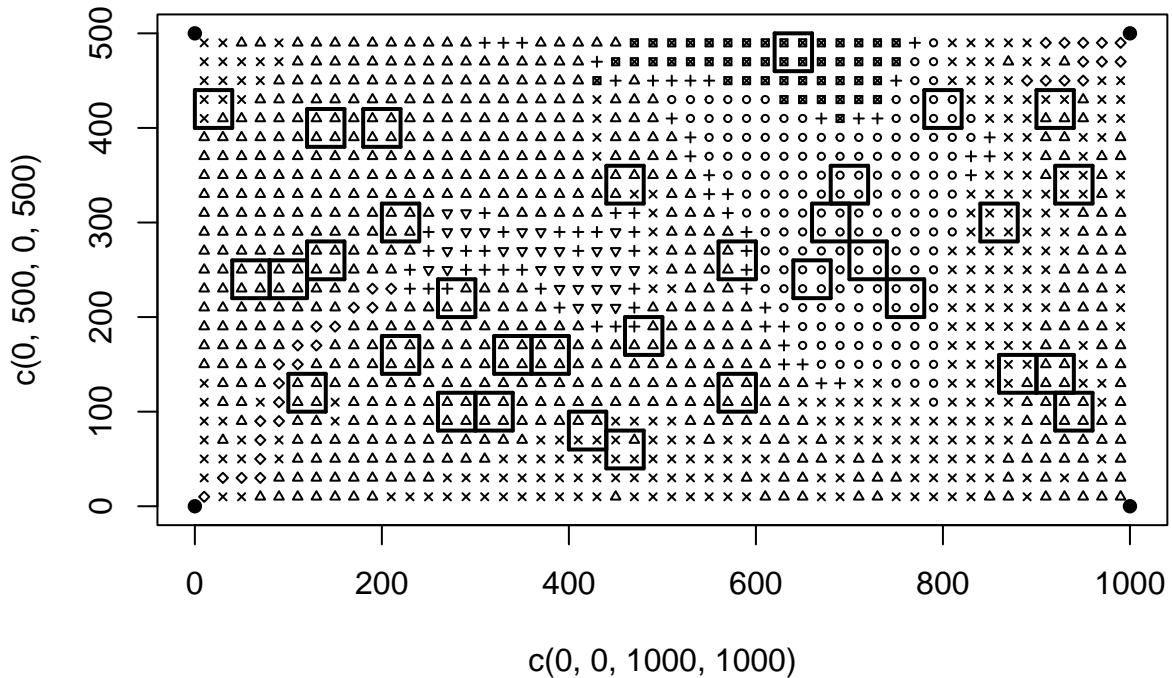


```

plot(c(0,0,1000,1000),c(0,500,0,500),pch=16, main = "TLS 2019 subplots & Habitat class")
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.5)

```

# TLS 2019 subplots & Habitat class



## Canopy height data (from 2023 airborne lidar)

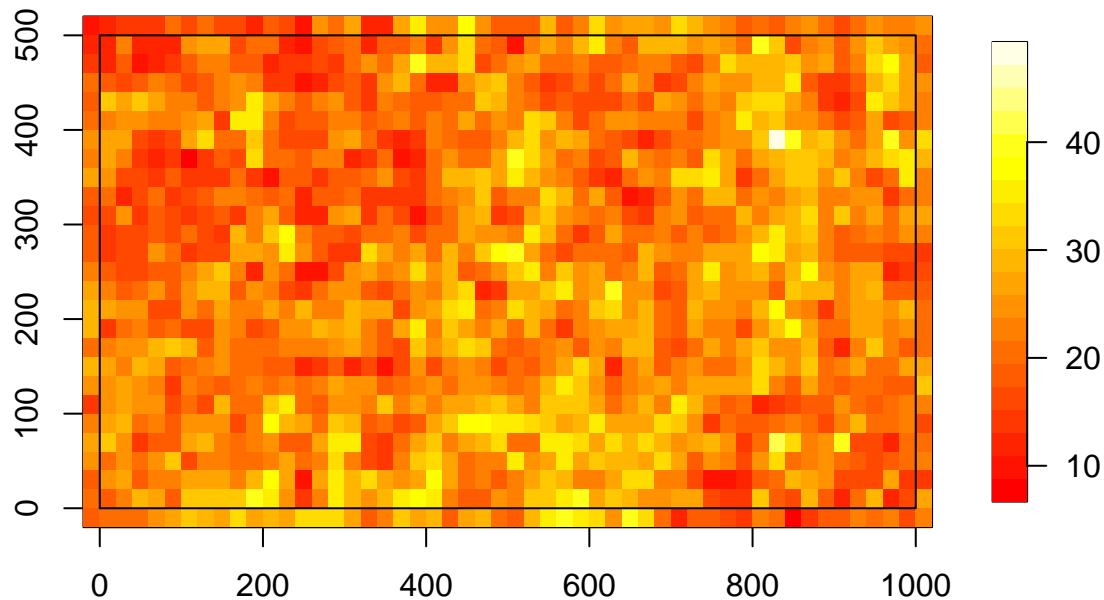
Load the 20-m and 5-m mean and sd 2023 lidar-derived canopy height stats processed in whereTLSonBCI\_part1b.Rmd. (Note those weren't fully adjusted for the plot not being oriented perfectly N-S, so they are actually a few meters off in parts of the plot.)

```

q20canht2023 <- read.table("usedata/canht2023q20.txt",header=T,sep="\t")
x20s <- sort(unique(q20canht2023$x))
y20s <- sort(unique(q20canht2023$y))
mnch23q20 <- matrix(q20canht2023$mnch2023,nrow=length(x20s),ncol=length(y20s))
sdch23q20 <- matrix(q20canht2023$sdch2023,nrow=length(x20s),ncol=length(y20s))
image.plot(x20s,y20s,mnch23q20,col=heat.colors(20),
            main="20-m Mean Lidar canopy height May 2023",xlab="",ylab="")
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))

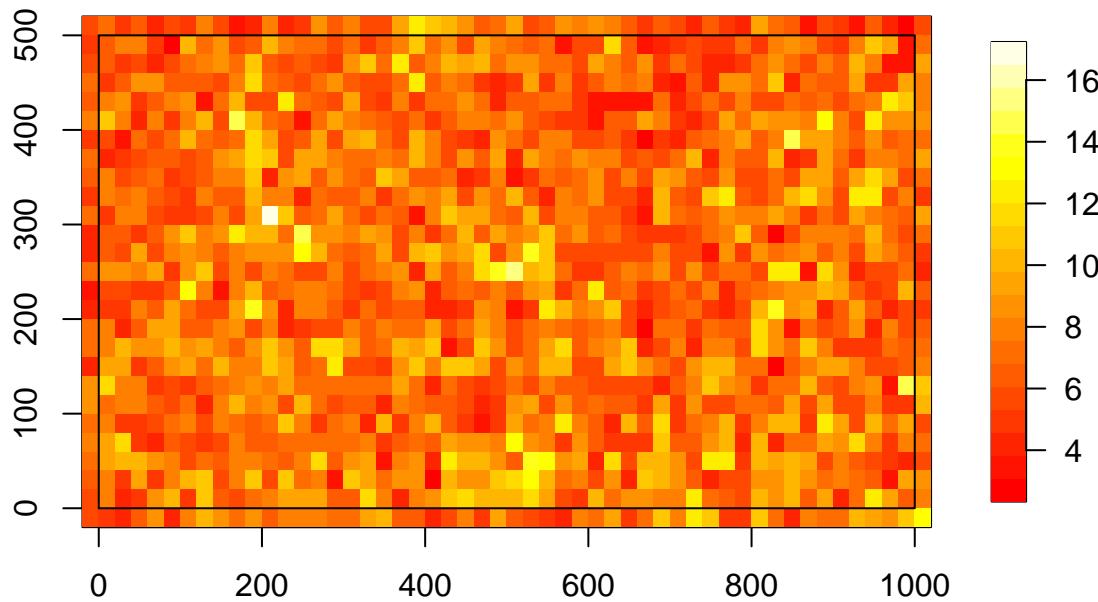
```

**20-m Mean Lidar canopy height May 2023**



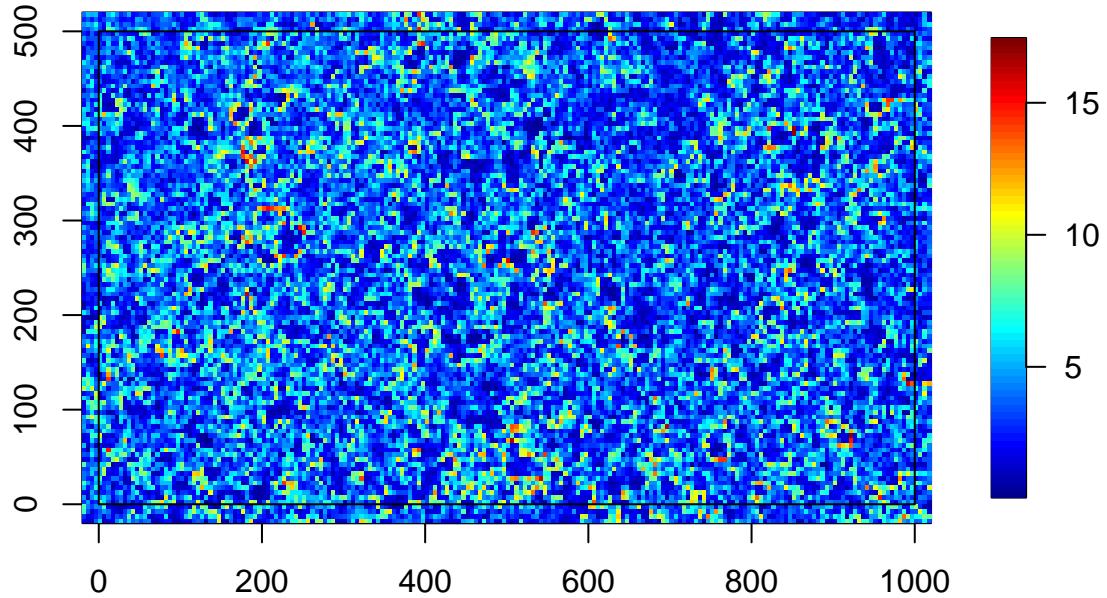
```
image.plot(x20s,y20s,sdch23q20,col=heat.colors(20),
           main="20-m SD Lidar canopy height May 2023",xlab="",ylab="")
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))
```

## 20-m SD Lidar canopy height May 2023



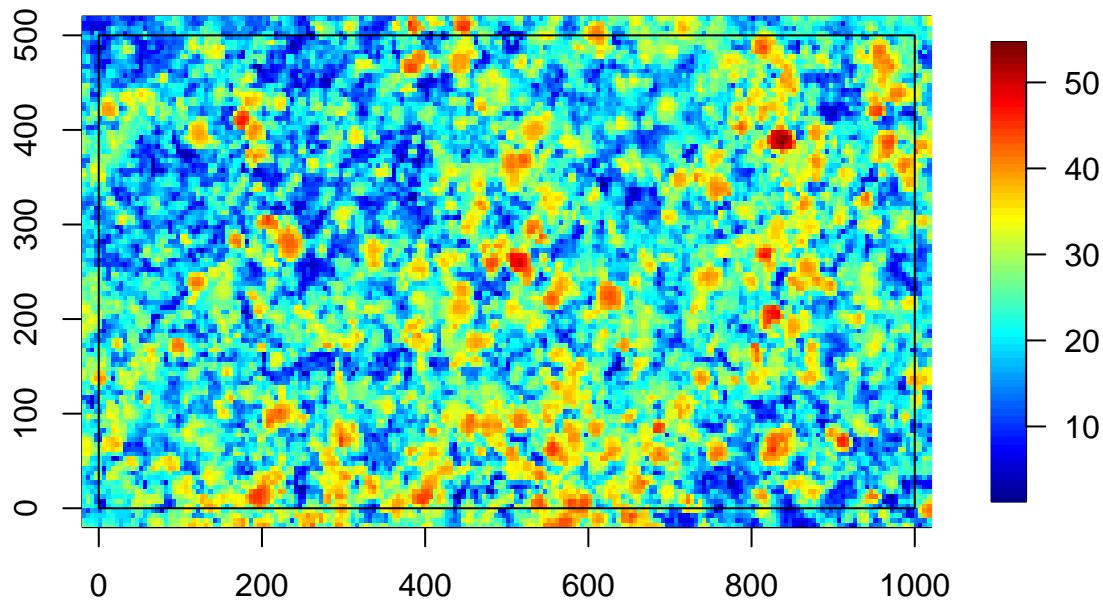
```
p5canht2023 <- read.table("usedata/canht2023p5.txt", header=T, sep="\t")
x5s <- sort(unique(p5canht2023$x))
y5s <- sort(unique(p5canht2023$y))
mnch23p5 <- matrix(p5canht2023$mnccanht2023, nrow=length(x5s), ncol=length(y5s))
sdch23p5 <- matrix(p5canht2023$sdccanht2023, nrow=length(x5s), ncol=length(y5s))
image.plot(x5s, y5s, sdch23p5,
           main="5-m SD Lidar canopy height May 2023", xlab="", ylab="")
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))
```

**5-m SD Lidar canopy height May 2023**



```
image.plot(x5s,y5s,mnch23p5,
           main="5-m Mean Lidar canopy height May 2023",xlab="",ylab="")
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))
```

## 5-m Mean Lidar canopy height May 2023



### Forest structure stats by 20 x 20 quadrat (from 2015 census)

Assumes have already run code to make df\_quad from function agb\_dyn\_byquad to estimate quadrat-level annual AGB and AWP (in whereTLSonBCI\_part2.rmd)

Now make data frames that have the quadrat level information too

```
load("usedata/cachequad.rda")

addleadingzeroquad <- function(q20) {
  q20char <- as.character(q20)
  nchars <- nchar(q20char)
  quadrat4 <- ifelse(nchars==4,q20char,
                       ifelse(nchars==3,paste0("0",q20char),
                             ifelse(nchars==2,paste0("00",q20char),
                                   paste0("000",q20char))))
  return(quadrat4)
} # end addleadingzeroquad

bci_habitat$quadrat <- addleadingzeroquad(bci_habitat$q20)

use_quad <- merge(df_quad,bci_habitat,by="quadrat",all=T)
use_quad <- use_quad[order(use_quad$y,use_quad$x),]
q20x <- sort(unique(use_quad$x))
```

```

q20y <- sort(unique(use_quad$y))

agb2015 <- subset(use_quad, year==2015 & group=="chave14_h")

ba2015 <- subset(use_quad, year==2015 & group=="ba")

n10y2015 <- subset(use_quad, year==2015 & group=="n10")

bci_habitat$x_y <- paste0(bci_habitat$x, "_", bci_habitat$y)

```

## Map habitat types

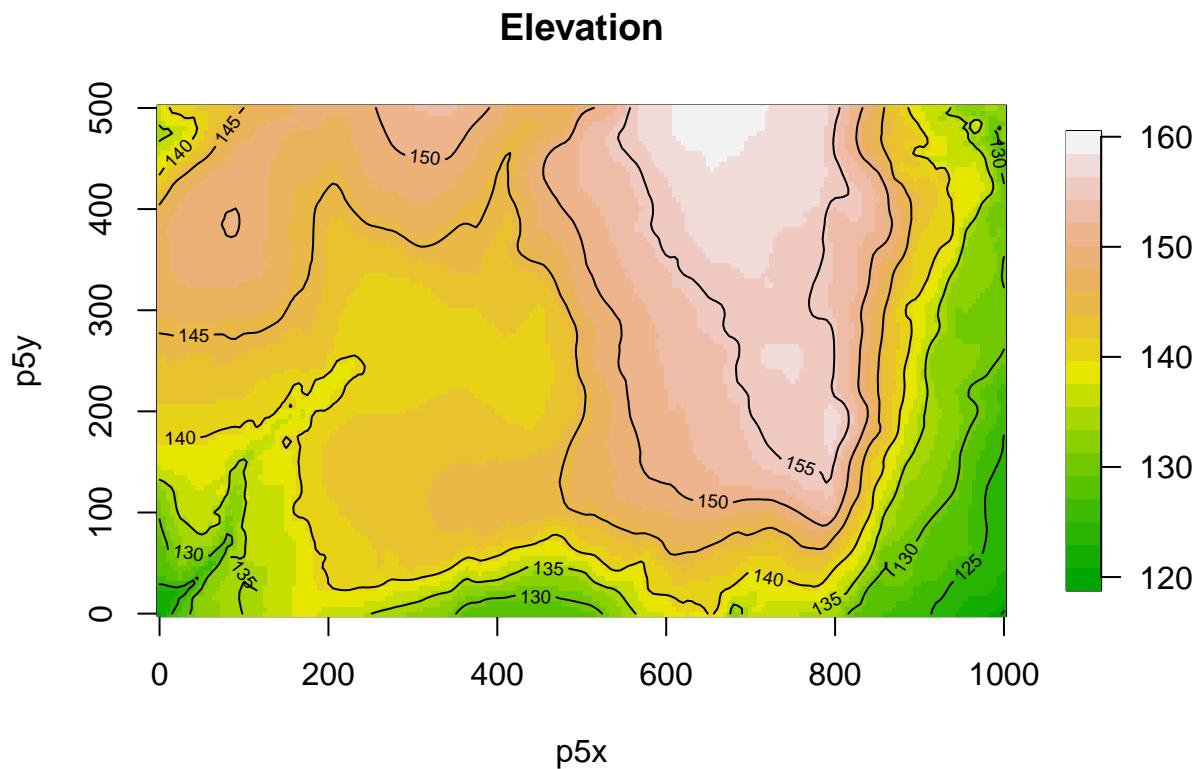
Make figures of the habitat types

```

agb2015 <- agb2015[order(agb2015$y, agb2015$x),]
habitatm <- matrix(as.numeric(as.factor(agb2015$habitat)), nrow=length(q20x), ncol=length(q20y))

elevation <- read.table("usedata/elev.TXT", header=T, sep="\t")
elevation <- elevation[order(elevation$y, elevation$x),] # put in the right order for default matrix fil
p5x <- sort(unique(elevation$x))
p5y <- sort(unique(elevation$y))
elevm5 <- matrix(elevation$elev, nrow=length(p5x), ncol=length(p5y))
image.plot(p5x, p5y, elevm5, main="Elevation", col=terrain.colors(20))
contour(p5x, p5y, elevm5, add=T, nlevels=10)

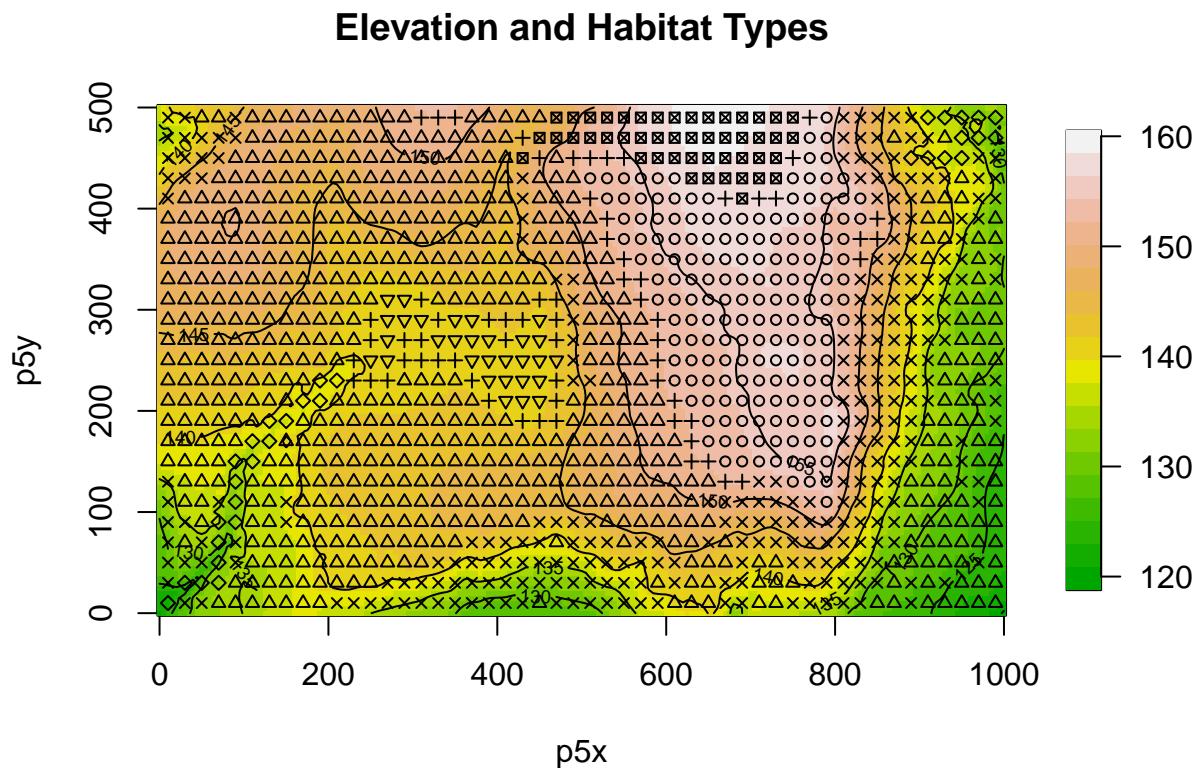
```



```

image.plot(p5x,p5y,elevm5,main="Elevation and Habitat Types",col=terrain.colors(20))
contour(p5x,p5y,elevm5,add=T,nlevels=10)
points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.7)

```

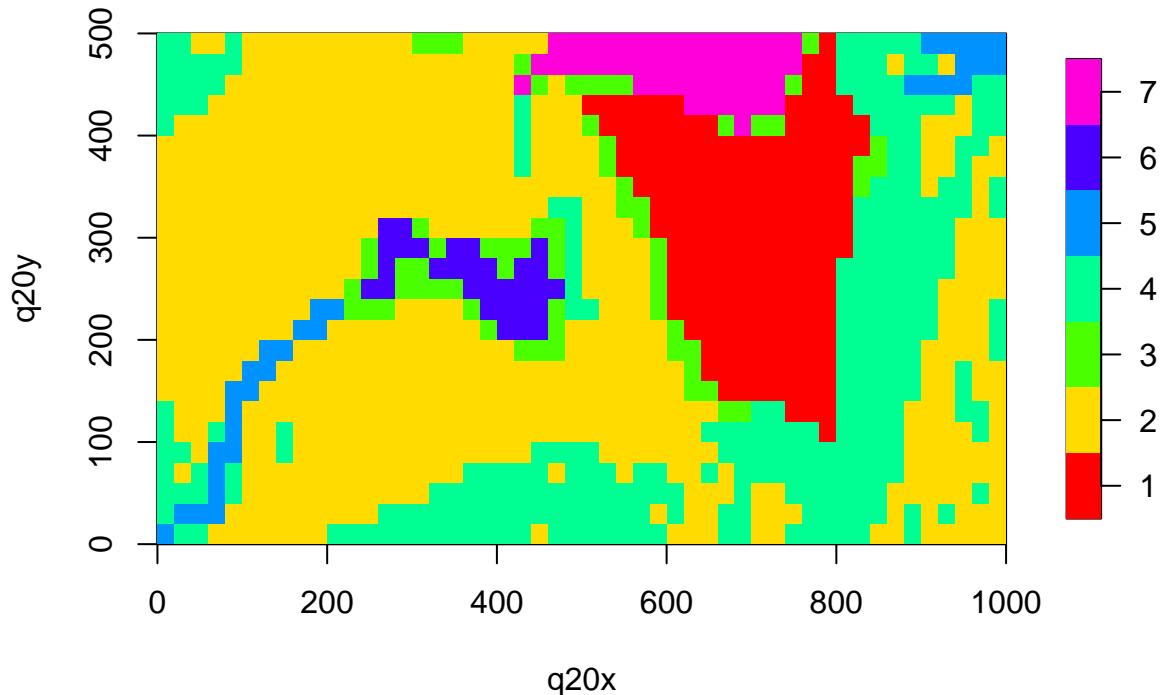


```

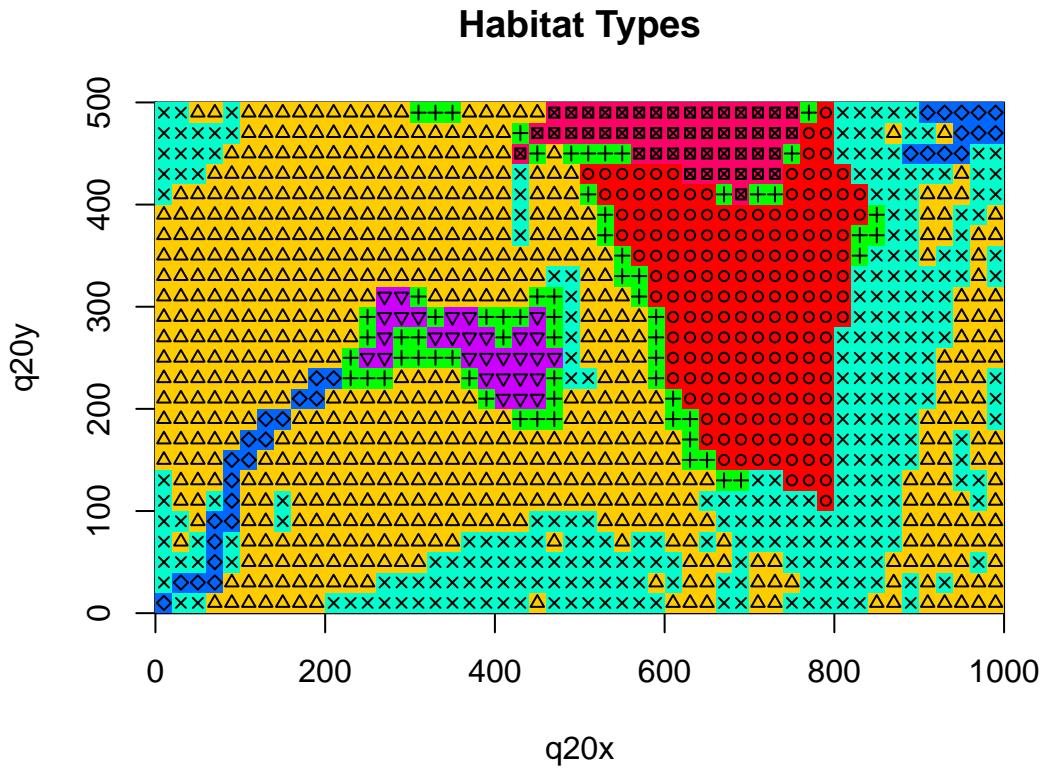
image.plot(q20x,q20y,habitatm, main="Topographic habitat types",col=rainbow(7))

```

## Topographic habitat types



```
image.plot(q20x,q20y,habitatm, main="Habitat Types",col=rainbow(15))
points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.7)
```



Double check that the x and y coordinates are being assigned and graphed correctly in image.

```
# following is to doublecheck the matrices are being assembled and graphed correctly
xm <- matrix(agb2015$x,nrow=length(q20x),ncol=length(q20y))
ym <- matrix(agb2015$y,nrow=length(q20x),ncol=length(q20y))

#image.plot(q20x,q20y,xm)
#image.plot(q20x,q20y,ym)
```

## Liana density

Just liana stem map from Schnitzer et al. 2023 to figure out how to display it.

```
# load the liana map image
lianamap <- readJPEG("liana/82 Schnitzer LianaSpecies Figure 1.jpg")
lianares <- dim(lianamap)[2:1]

# make this map semi-transparent
lianamap4 <- array(NA,dim=c(dim(lianamap)[1],dim(lianamap)[2],4))
lianamap4[,1] <- lianamap[,1]
lianamap4[,2] <- lianamap[,2]
lianamap4[,3] <- lianamap[,3]
lianamap4[,4] <- 0.5

plot(c(0,0,1000,1000),c(0,500,0,500),pch=16, main = "Liana stem map alone, check location")
```

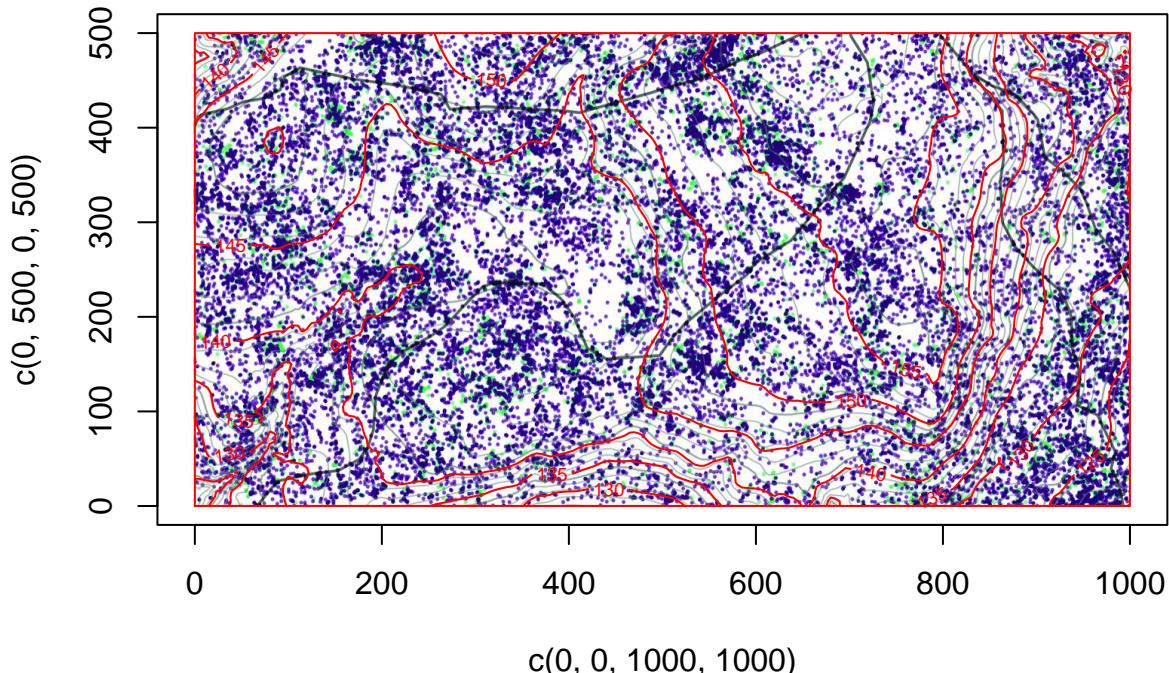
```

offset=16
rasterImage(lianamap,xleft=-offset,ybottom=-offset,xright=1000+offset,ytop=500+offset)
lines(c(0,0,1000,1000,0),c(0,500,500,0,0),col="red")

contour(p5x,p5y,elevm5,add=T,nlevels=10,col="red")

```

## Liana stem map alone, check location



Combining liana stem map from Schnitzer et al. 2023 with dendrometer plots and habitat class.

```

dend40centers <- read.table("usedata/bci40x40sxy.txt",header=T,sep="\t")
plot(c(0,0,1000,1000),c(0,500,0,500),pch=16, main = "Dendrometer subplots, Habitat class, and Liana stem map alone, check location")

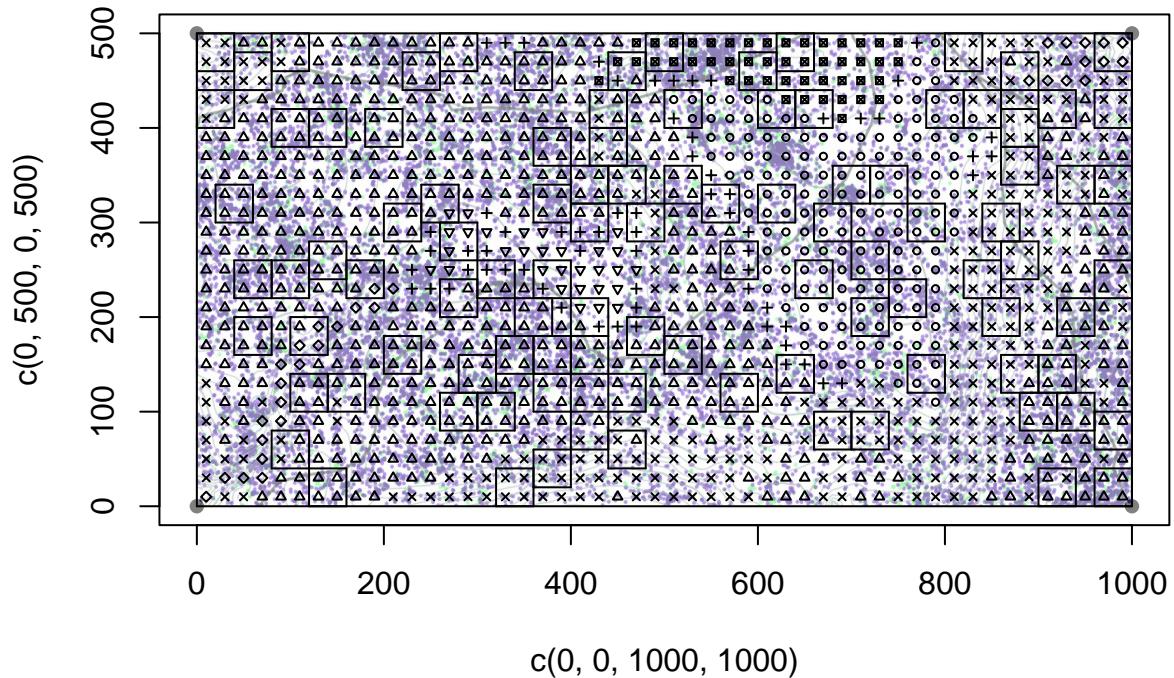
rasterImage(lianamap4,xleft=-offset,ybottom=-offset,xright=1000+offset,ytop=500+offset)
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))

for (i in 1:nrow(dend40centers)) {
  lines(x=rep(dend40centers$x[i]-20,2), y=c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(x=rep(dend40centers$x[i]+20,2), y=c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(x=c(dend40centers$x[i]-20,dend40centers$x[i]+20), y=rep(dend40centers$y[i]-20,2))
  lines(x=c(dend40centers$x[i]-20,dend40centers$x[i]+20), y=rep(dend40centers$y[i]+20,2))
}

points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.5)

```

## Dendrometer subplots, Habitat class, and Liana stem map

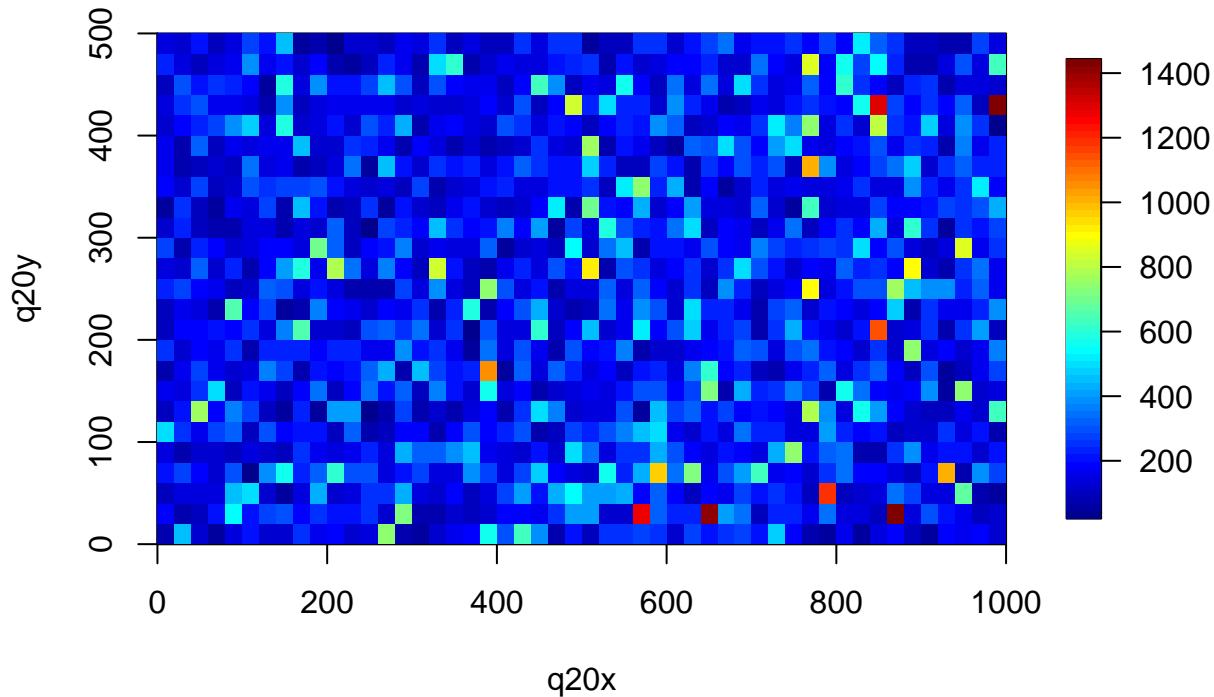


```
## Maps of AGB and N10, untransformed and log-transformed, by 20x20
```

```
agb2015mbq20 <- matrix(agb2015$stock, nrow=length(q20x), ncol=length(q20y))
ba2015mbq20 <- matrix(ba2015$stock, nrow=length(q20x), ncol=length(q20y))
n10y2015mbq20 <- matrix(n10y2015$stock, nrow=length(q20x), ncol=length(q20y))

#quantile(agb2015mbq20, probs=seq(0.05, 0.95, by=0.1))
#hist(agb2015mbq20)
image.plot(q20x, q20y, agb2015mbq20, main = "Above-Ground Biomass (AGB) in 2015 (Mg/ha), by 20x20")
```

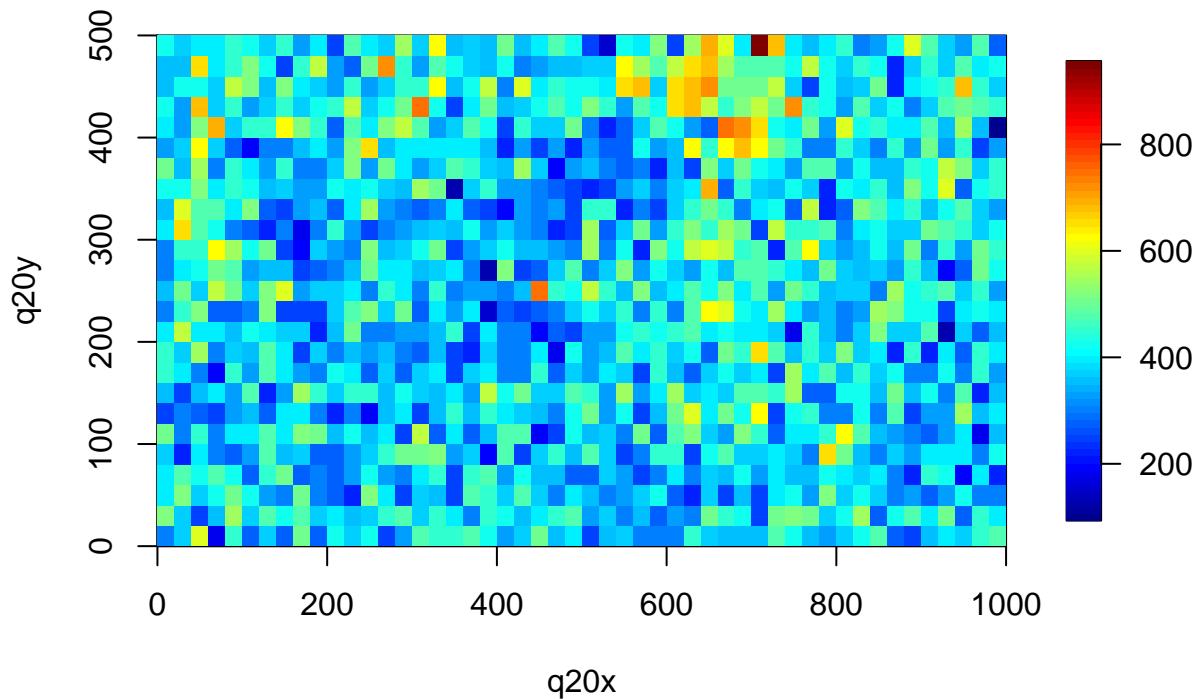
## Above-Ground Biomass (AGB) in 2015 (Mg/ha), by 20x20



```
# basal area patterns mirror those of AGB, so don't bother showing
#quantile(ba2015mbyq20,probs=seq(0.05,0.95,by=0.1))
#hist(ba2015mbyq20)
#image.plot(q20x,q20y,ba2015mbyq20, main = "Basal Area (BA) in 2015 (m2/ha), by 20x20")

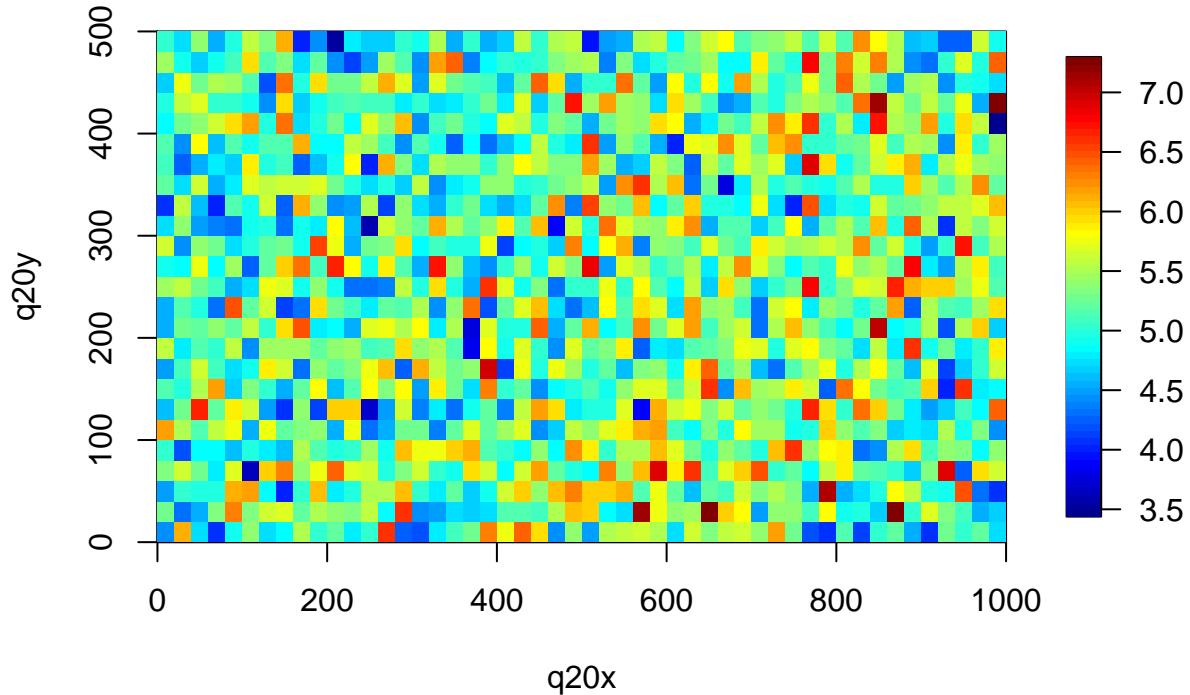
#quantile(n10y2015mbyq20,probs=seq(0.05,0.95,by=0.1))
#hist(n10y2015mbyq20)
image.plot(q20x,q20y,n10y2015mbyq20, main = "Tree Density (N>=10 cm dbh) in 2015 (N/ha), by 20x20")
```

### Tree Density ( $N \geq 10$ cm dbh) in 2015 (N/ha), by 20x20



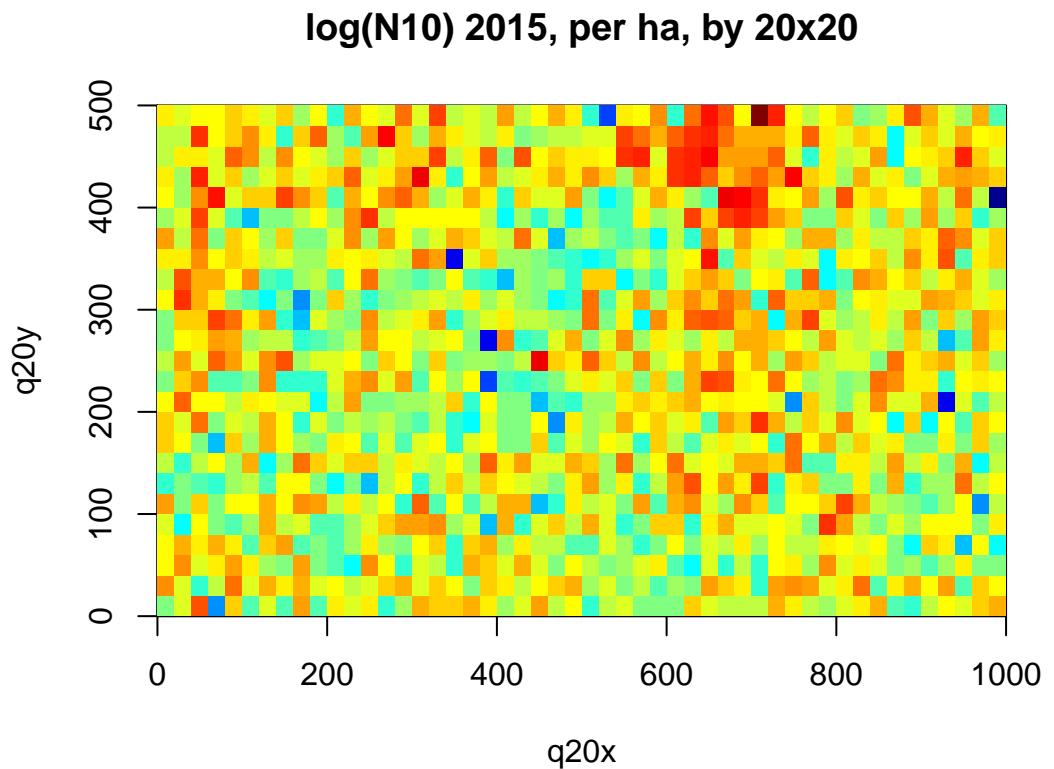
```
#quantile(log10(agb2015mbyq20), probs = seq(0.05, 0.95, by = 0.1))
#hist(log10(agb2015mbyq20))
image.plot(q20x, q20y, log(agb2015mbyq20), main = "log(AGB) 2015, Mg/ha, by 20x20")
```

### **log(AGB) 2015, Mg/ha, by 20x20**



```
#quantile(log10(ba2015mbyq20),probs=seq(0.05,0.95,by=0.1))
#hist(log10(ba2015mbyq20))
#image.plot(q20x,q20y,log(ba2015mbyq20), main = "log(BA) 2015, m2/ha, by 20x20")

#quantile(log10(n10y2015mbyq20),probs=seq(0.05,0.95,by=0.1))
#hist(log10(n10y2015mbyq20))
image.plot(q20x,q20y,log(n10y2015mbyq20), main = "log(N10) 2015, per ha, by 20x20")
```



Load and map crown-distributed AGB, and compare with stem-localized

```

load("usedata/cachecrown.rda")

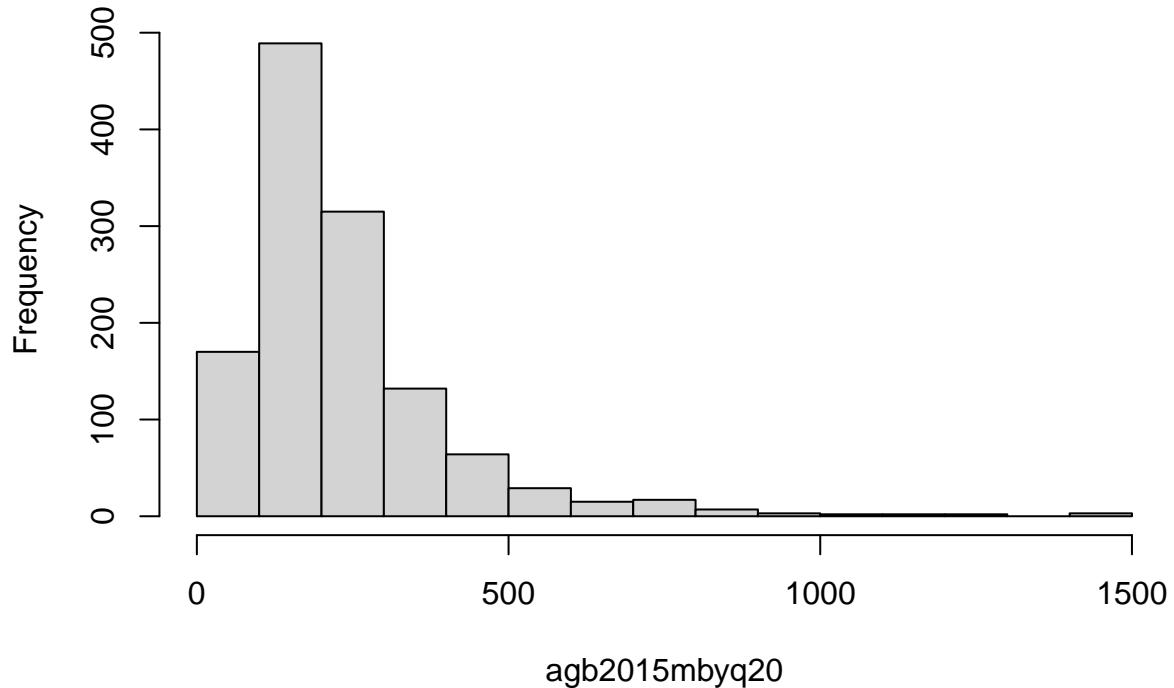
df_crown_2015 <- df_crown_2015[order(df_crown_2015$y,df_crown_2015$x),] # insure order is good for defa
cagb2015mbbyq20 <- matrix(df_crown_2015$agb,nrow=length(q20x),ncol=length(q20y))

#quantile(agb2015mbbyq20,probs=seq(0.05,0.95,by=0.1))
#quantile(cagb2015mbbyq20,probs=seq(0.05,0.95,by=0.1))

hist(agb2015mbbyq20, main = "Stem-localized AGB 2015, Mg/ha, by 20x20")

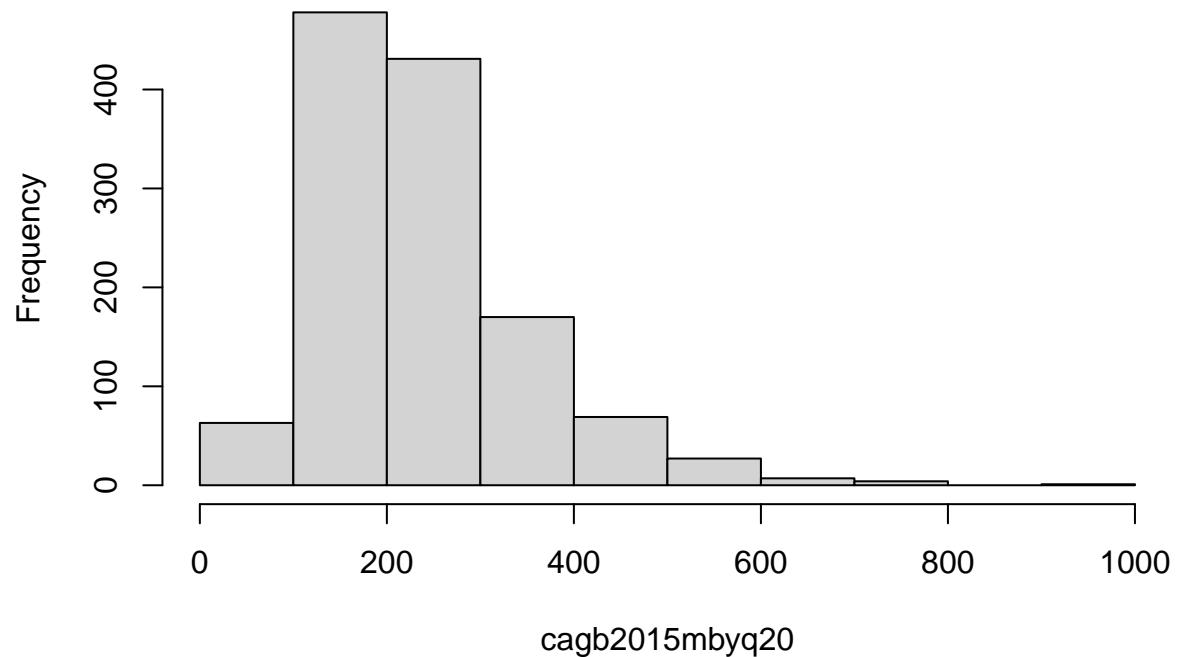
```

### Stem-localized AGB 2015, Mg/ha, by 20x20



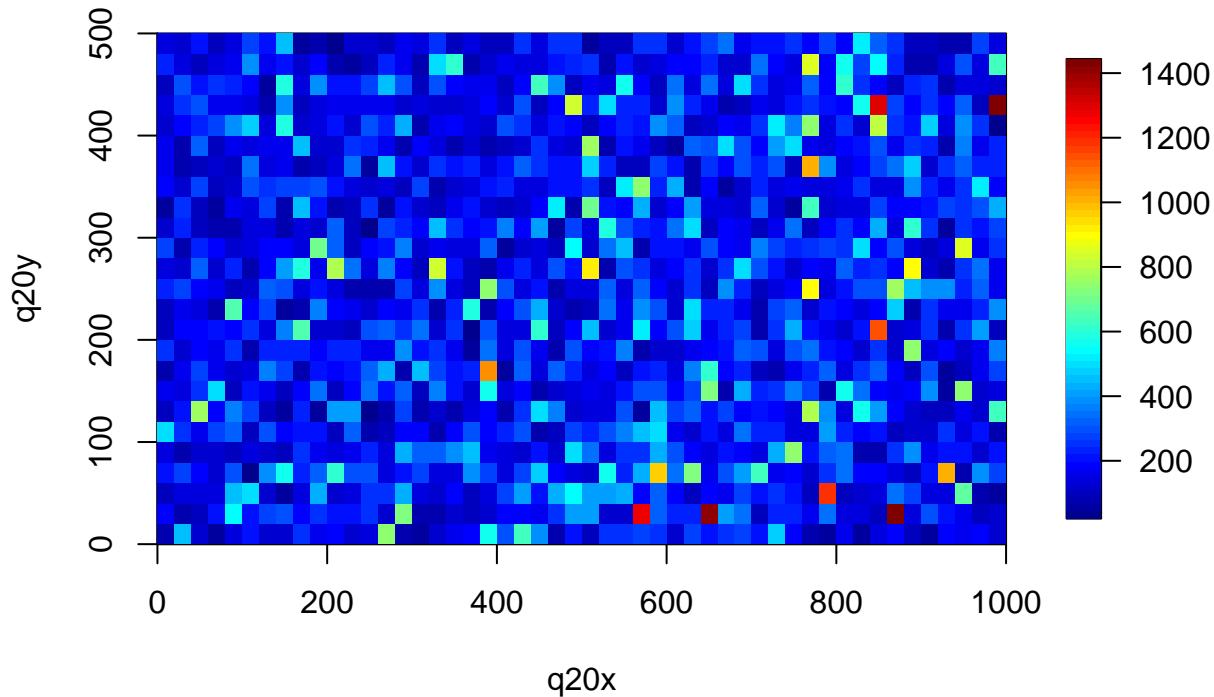
```
hist(cagb2015mbyq20, main = "Crown-distributed AGB 2015, Mg/ha, by 20x20")
```

### Crown-distributed AGB 2015, Mg/ha, by 20x20



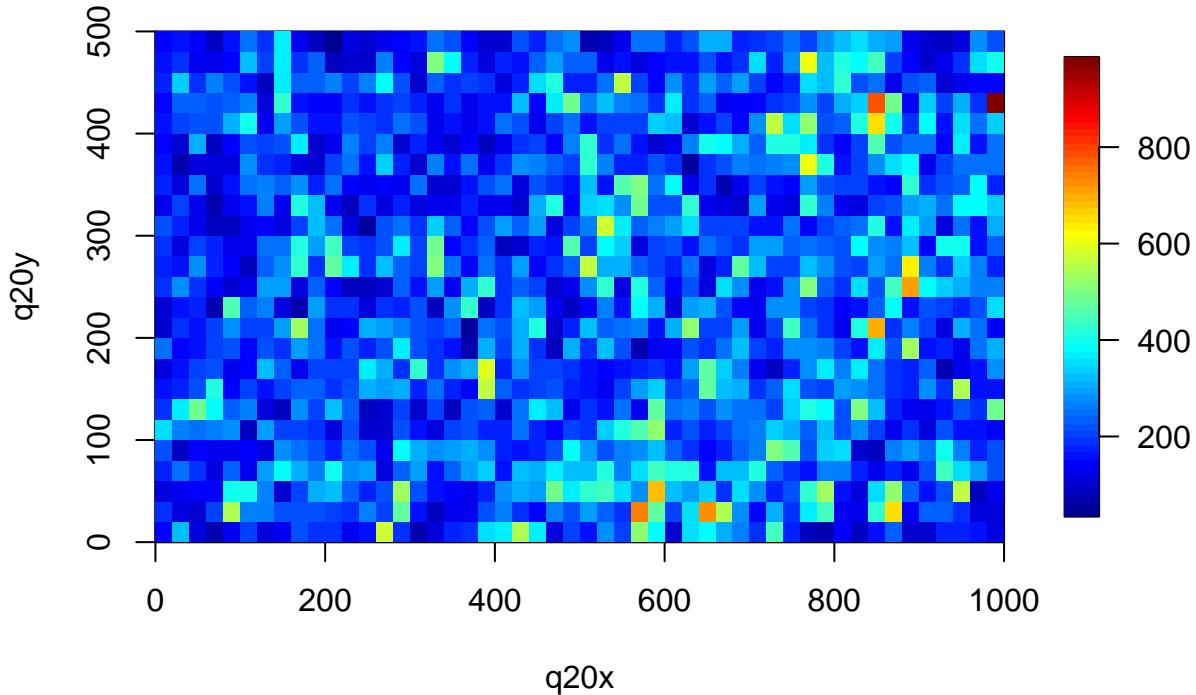
```
image.plot(q20x,q20y,agb2015mbyq20, main = "Stem-localized AGB 2015, Mg/ha, by 20x20")
```

**Stem-localized AGB 2015, Mg/ha, by 20x20**



```
image.plot(q20x,q20y,cagb2015mbyq20, main = "Crown-distributed AGB 2015, Mg/ha, by 20x20")
```

## Crown-distributed AGB 2015, Mg/ha, by 20x20



Get 1-ha running means and sds of stats of interest and map them

Function to do running means of matrices.

```
# function to return a matrix of the same dimensions as the input
# with each cell holding the mean of the original matrix cells
# centered on the current cell and with row and column difference <= sidedist
# so for example, sidedist=1 would mean a mean of a 3x3 unit centered on the focal one
# and sidedist=2 would mean a mean of a 5x5 unit centered on the focal one
# for inmatrices corresponding to 20x20 quadrats, and desired 1-ha data, use sidedist=2
getmatrixrunmean <- function (inmatrix,sidedist=2) {
  nrowm <- dim(inmatrix)[1]
  ncolm <- dim(inmatrix)[2]
  outmatrix <- matrix(NA, nrow=nrowm,ncol=ncolm)
  for (i in 1:nrowm) {
    for (j in 1:ncolm) {
      thism <- inmatrix[(max(1,i-sidedist):min(nrowm,i+sidedist)),(max(1,j-sidedist):min(ncolm,j+sidedist))]
      outmatrix[i,j] <- mean(thism)
    }
  }
  return(outmatrix)
}
```

Function to do running standard deviation of matrices.

```

# function to return a matrix of the same dimensions as the input
# with each cell holding the sd of the original matrix cells
# centered on the current cell and with row and column difference <= sidedist
# so for example, sidedist=1 would return an sd of a 3x3 unit centered on the focal one
getmatrixrunsd <- function (inmatrix,sidedist=2) {
  nrowm <- dim(inmatrix)[1]
  ncolm <- dim(inmatrix)[2]
  outmatrix <- matrix(NA, nrow=nrowm,ncol=ncolm)
  for (i in 1:nrowm) {
    for (j in 1:ncolm) {
      thism <- inmatrix[(max(1,i-sidedist):min(nrowm,i+sidedist)),(max(1,j-sidedist):min(ncolm,j+sidedist))]
      outmatrix[i,j] <- sd(thism)
    }
  }
  return(outmatrix)
}

```

Get 1-ha running means, SDs, and CVs of all the key variables

```

# for AGB
agb2015by1ha <- getmatrixrunmean(agb2015mbyq20,2)
sdagb2015by1ha <- getmatrixrunsd(agb2015mbyq20,2)
cvagb2015by1ha <- sdagb2015by1ha/agb2015by1ha

# for basal area, BA
ba2015by1ha <- getmatrixrunmean(ba2015mbyq20,2)
sdba2015by1ha <- getmatrixrunsd(ba2015mbyq20,2)
cvba2015by1ha <- sdba2015by1ha/ba2015by1ha

# for density of trees >= 10 cm dbh
n10y2015by1ha <- getmatrixrunmean(n10y2015mbyq20,2)
sdn10y2015by1ha <- getmatrixrunsd(n10y2015mbyq20,2)
cvn10y2015by1ha <- sdn10y2015by1ha/n10y2015by1ha

# for canopy height
canht2023by1ha <- getmatrixrunmean(mnch23q20,2)
sdcanht2023by1ha <- getmatrixrunsd(mnch23q20,2)
cvcanht2015by1ha <- sdcanht2023by1ha/canht2023by1ha

```

Evaluate habitat heterogeneity within larger subplots. Specifically, obtain most common habitat type, number of habitat types, and the relative abundance of the most common habitat type.

```

gethabitathet <- function (inmatrix,sidedist=2) {
  nrowm <- dim(inmatrix)[1]
  ncolm <- dim(inmatrix)[2]
  mainhabitatm <- nhabitatem <- habdominancem <- matrix(NA, nrow=nrowm,ncol=ncolm)
  for (i in 1:nrowm) {
    for (j in 1:ncolm) {
      thism <- inmatrix[(max(1,i-sidedist):min(nrowm,i+sidedist)),
                         (max(1,j-sidedist):min(ncolm,j+sidedist))]
      thishtable <- as.data.frame(table(thism))
      thishtable <- thishtable[order(thishtable$Freq,decreasing=T),]
      mainhabitatm[i,j] <- as.numeric(as.character(thishtable$thism[1]))
    }
  }
}

```

```

    nhabitatem[i,j] <- nrow(thishabtable)
    habdominancem[i,j] <- thishabtable$Freq[1]/sum(thishabtable$Freq)
}
}
return(list(mainhabitatem=mainhabitatem,nhabitatem=nhabitatem,habdominancem=habdominancem))
}

habhet <- gethabitathet(habitatm)
mainhabitatem <- habhet$mainhabitatem
nhabitatem <- habhet$nhabitatem
habdominancem <- habhet$habdominancem

```

Make matrices for all 20x20s of presence/absence of dendrometer plots and 2019 TLS plots. Use these to make matrices of proportions of 1-ha plots that are in dendrometer plots and 2019 TLS plots.

```

indend <- matrix(FALSE,nrow=50,ncol=25)
dendq20x <- floor(dend40centers$x/20)
dendq20y <- floor(dend40centers$y/20)
for (i in 1:length(dend40centers$x)) {
  indend[dendq20x[i]:(dendq20x[i]+1),dendq20y[i]:(dendq20y[i]+1)] <- TRUE
}

intls19 <- matrix(FALSE,nrow=50,ncol=25)
tlsoldq20x <- floor(tls40centers$x/20)
tlsoldq20y <- floor(tls40centers$y/20)
for (i in 1:length(tls40centers$x)) {
  intls19[tlsoldq20x[i]:(tlsoldq20x[i]+1),tlsoldq20y[i]:(tlsoldq20y[i]+1)] <- TRUE
}

indendiha <- getmatrixrunmean(indend)
inoldtls1ha <- getmatrixrunmean(intls19)

```

Make a combined data.frame of all the variables for 1-ha subplots. Also make variables of the corresponding quantiles. One entry per 1-ha subplot that has boundaries on 20x20 quadrat edges (so 1-ha plots overlap). Note that quadrats at plot borders are excluded because can't be centers of 1-ha subplots within the plot.

```

q20xmatrix <- matrix(rep(q20x,times=length(q20y)),nrow=length(q20x),ncol=length(q20y))
q20ymatrix <- matrix(rep(q20y,each=length(q20x)),nrow=length(q20x),ncol=length(q20y))

# omit values for areas too close to edge (not truly 1-ha plots)
xcent <- 3:48
ycent <- 3:23

# canopy height data includes a set of quadrats around the edge, so one more in for those:
xcentch <- 4:49
ycentch <- 4:24

df1ha <- data.frame(xcenter=as.vector(q20xmatrix[xcent,ycent]),
                      ycenter=as.vector(q20ymatrix[xcent,ycent]),
                      mnagb=as.vector(agb2015by1ha[xcent,ycent]),
                      sdagb=as.vector(sdagb2015by1ha[xcent,ycent]),
                      cvagb=as.vector(cvagb2015by1ha[xcent,ycent]),
                      mnba=as.vector(ba2015by1ha[xcent,ycent]),

```

```

sdba=as.vector(sdba2015by1ha[xcent,ycent]),
cvba=as.vector(cvba2015by1ha[xcent,ycent]),
mnn10=as.vector(n10y2015by1ha[xcent,ycent]),
sdn10=as.vector(sdn10y2015by1ha[xcent,ycent]),
cvn10=as.vector(cvn10y2015by1ha[xcent,ycent]),
mncanht=as.vector(canht2023by1ha[xcentch,ycentch]),
sdcanht=as.vector(sdcanht2023by1ha[xcent,ycent]),
cvcanht=as.vector(cvcanht2015by1ha[xcent,ycent]),
mainhabitat=as.vector(mainhabitatem[xcent,ycent]),
nhabitat=as.vector(nhabitatem[xcent,ycent]),
habdominance=as.vector(habdominancem[xcent,ycent]),
indend=as.vector(indendiha[xcent,ycent]),
inoldtls=as.vector(inoldtls1ha[xcent,ycent]))
df1ha$namemainhabitat <- levels(bci_habitat$habitat)[df1ha$mainhabitat]

# add variables that are quantiles
ngood=nrow(df1ha)
df1ha$qmnagb <- rank(df1ha$mnagb)/ngood
df1ha$qmnb <- rank(df1ha$mnba)/ngood
df1ha$qmnn10 <- rank(df1ha$mnn10)/ngood
df1ha$qmncanht <- rank(df1ha$mncanht)/ngood
df1ha$qsdagb <- rank(df1ha$sdagb)/ngood
df1ha$qsdba <- rank(df1ha$sdba)/ngood
df1ha$qsdn10 <- rank(df1ha$sdn10)/ngood
df1ha$qsdcanht <- rank(df1ha$sdcanht)/ngood
df1ha$qc vagb <- rank(df1ha$cvagb)/ngood
df1ha$qcvba <- rank(df1ha$cvba)/ngood
df1ha$qc vn10 <- rank(df1ha$cvn10)/ngood
df1ha$qcvcanht <- rank(df1ha$cvcanht)/ngood

```

Make matrices of quantiles for each of the potential 1-ha plots

```

# get matrices of just the usable values, for use in subsequent figures
n10use <- n10y2015by1ha[xcent,ycent]
bause <- ba2015by1ha[xcent,ycent]
agbuse <- agb2015by1ha[xcent,ycent]
canhtuse <- canht2023by1ha[xcentch,ycentch]

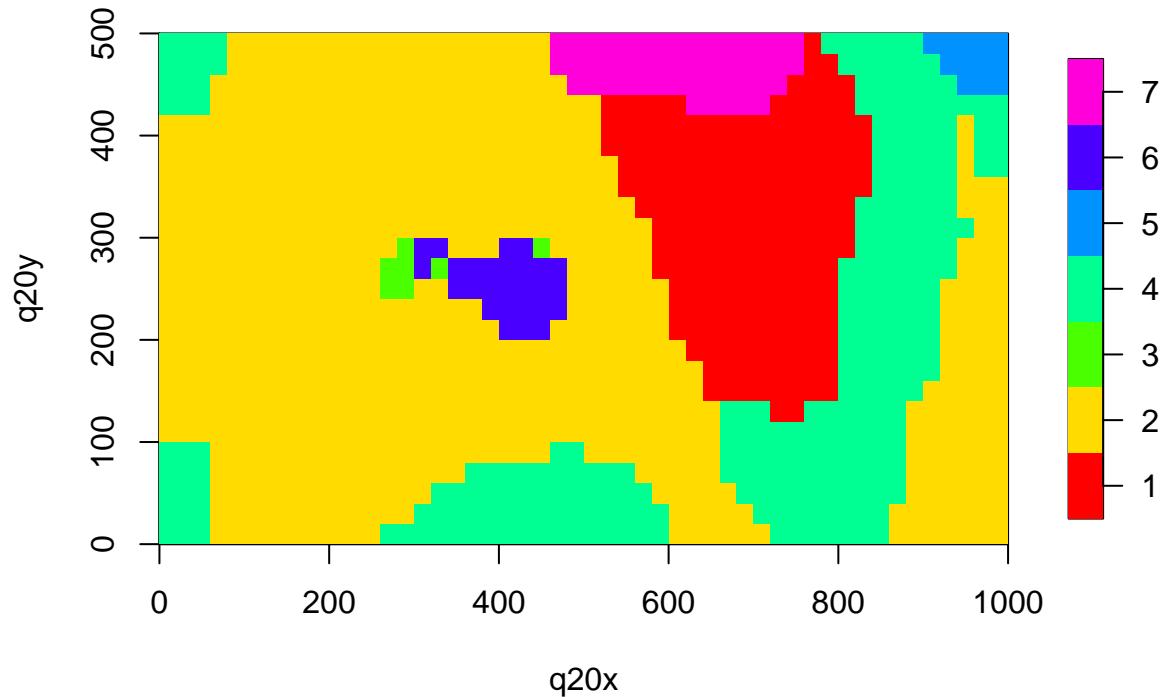
# now get parallel matrices of the quantiles
ntot <- length(as.vector(n10use))
n10quant <- matrix(rank(n10use),nrow=nrow(n10use),ncol=ncol(n10use))/ntot
baquant <- matrix(rank(bause),nrow=nrow(bause),ncol=ncol(bause))/ntot
agbquant <- matrix(rank(agbuse),nrow=nrow(agbuse),ncol=ncol(agbuse))/ntot
canhtquant <- matrix(rank(canhtuse),nrow=nrow(canhtuse),ncol=ncol(canhtuse))/ntot

```

## Figures of habitat heterogeneity.

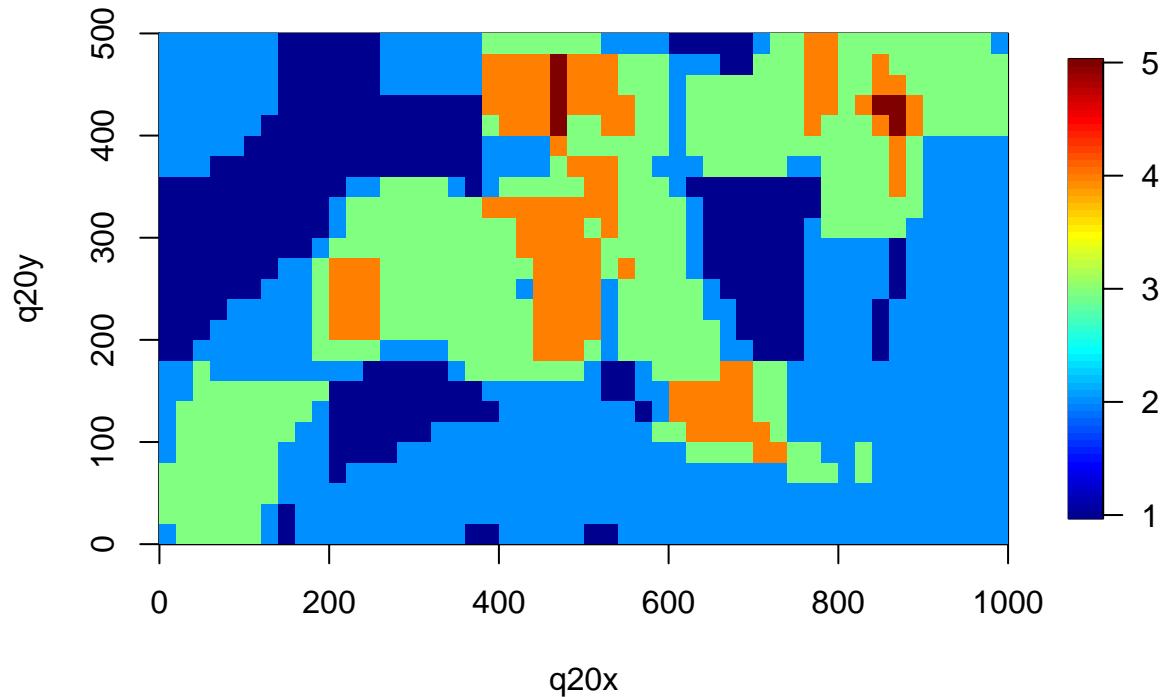
```
image.plot(q20x,q20y,mainhabitatem, main="Dominant topographic habitat type in 1-ha subplot",col=rainbow
```

## Dominant topographic habitat type in 1-ha subplot



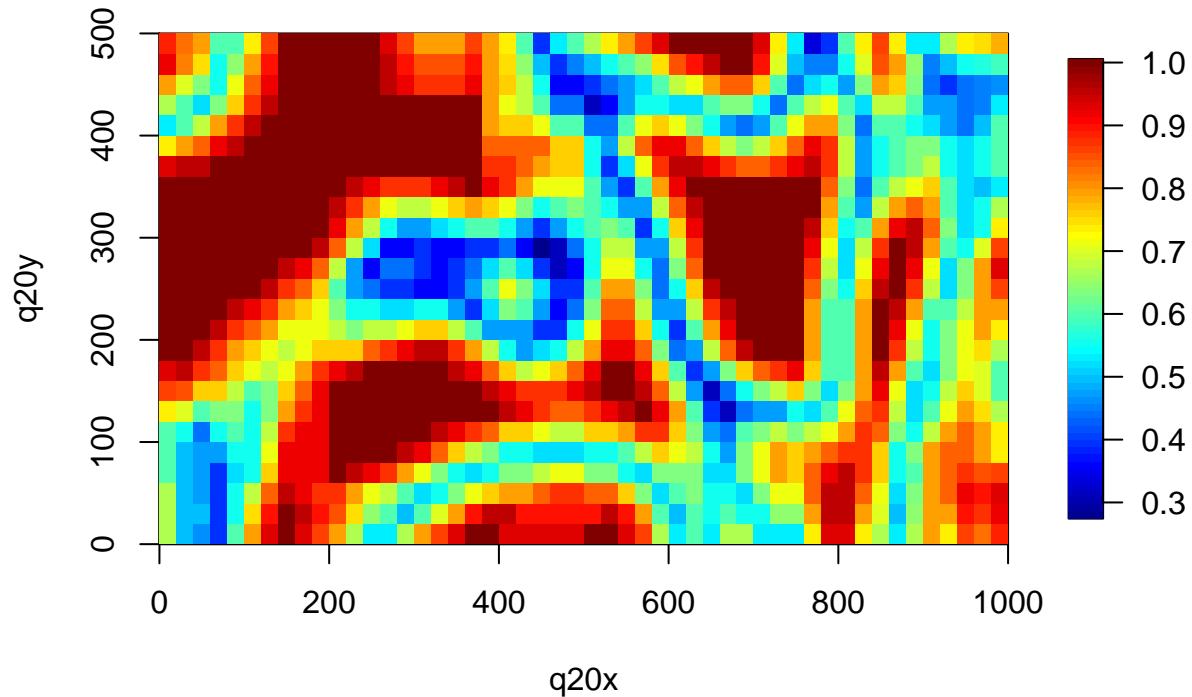
```
image.plot(q20x,q20y,nhabitatm, main="Number of habitat types per 1-ha subplot")
```

**Number of habitat types per 1-ha subplot**



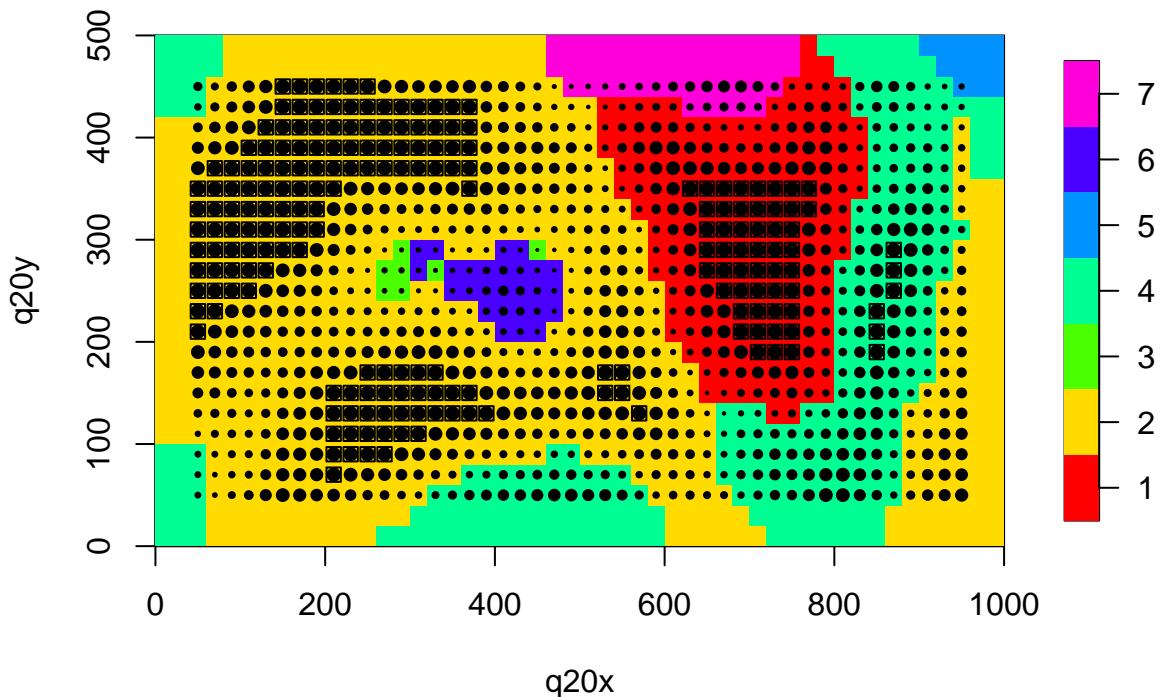
```
image.plot(q20x,q20y,habdominancem, main="Dominance of most common habitat within 1-ha subplot")
```

## Dominance of most common habitat within 1-ha subplot



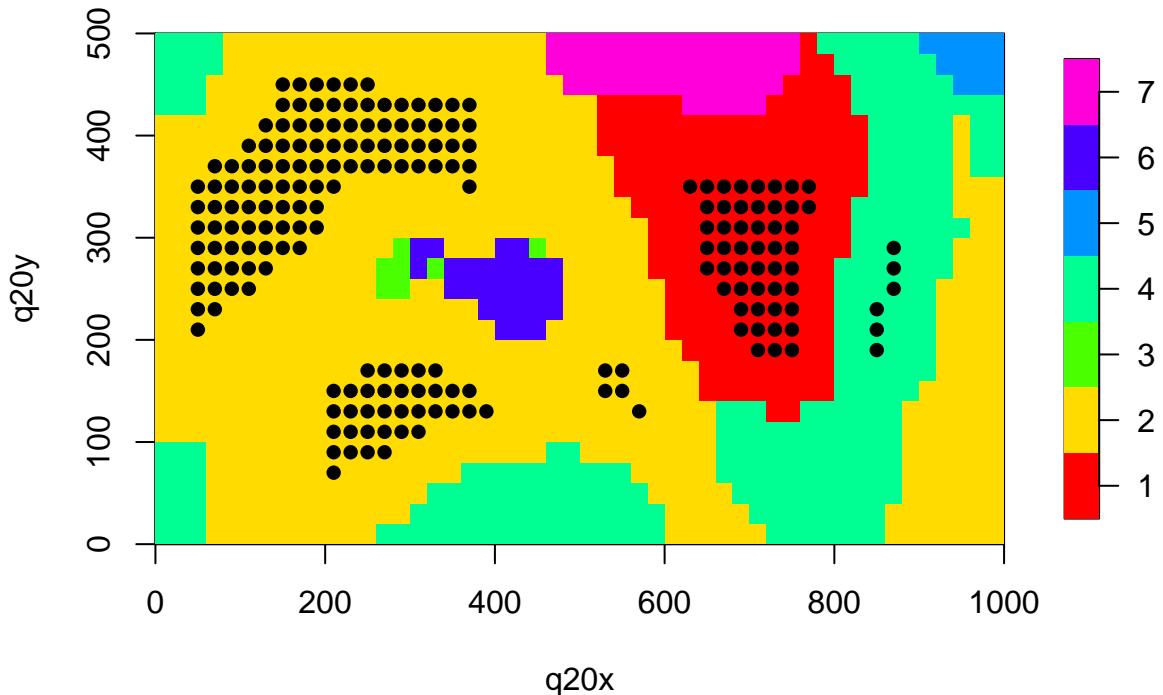
```
image.plot(q20x,q20y,main="Dominant habitat type in 1-ha, with point size indicating dominance")
points(df1ha$xcenter,df1ha$ycenter,pch=16,cex=df1ha$habdominance)
inc <- df1ha$nhabitat==1
points(df1ha$xcenter[inc],df1ha$ycenter[inc],pch=0)
```

## Dominant habitat type in 1-ha, with point size indicating dominance



```
image.plot(q20x,q20y,mainhabitatem, main="Dominant habitat type in 1-ha, with points indicating 100% that  
points(df1ha$xcenter[inc],df1ha$ycenter[inc],pch=16)
```

## Dominant habitat type in 1-ha, with points indicating 100% that type



Figures of forest structure metrics at 1-ha scales (spatial averages)

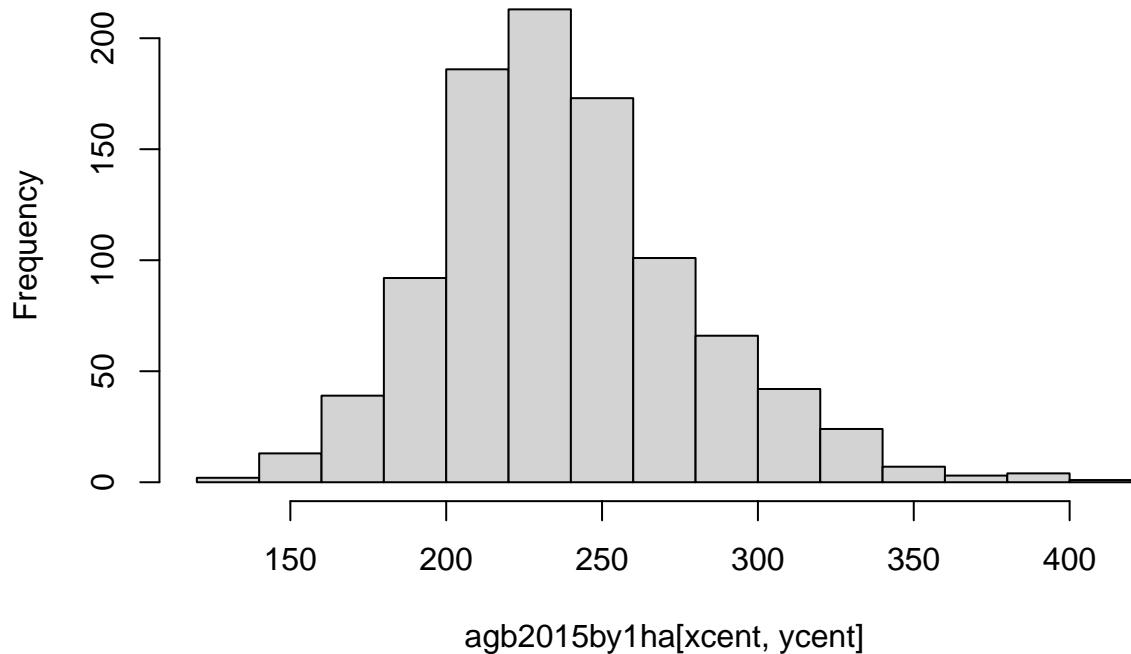
Figures of AGB density at 1-ha scale

```
quantile(agb2015by1ha,probs=seq(0.05,0.95,by=0.1))
```

```
##      5%     15%     25%     35%     45%     55%     65%     75%
## 173.7108 194.1687 207.3990 218.5525 227.6694 238.3038 248.2501 260.9420
##     85%     95%
## 283.2689 316.9524
```

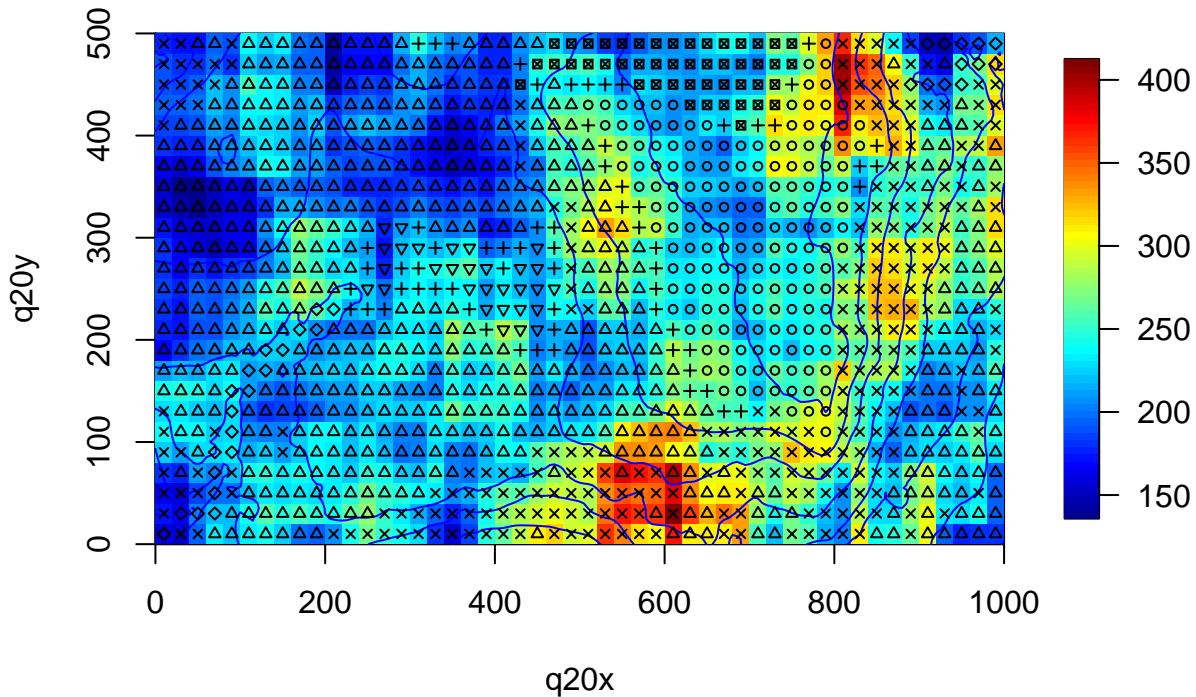
```
# in drawing histograms, omit quadrats near the edge which don't represent true possible 1-ha subplots
hist(agb2015by1ha[xcent,ycent],main="1-ha subplot AGB density (Mg/ha) in 2015")
```

## 1-ha subplot AGB density (Mg/ha) in 2015



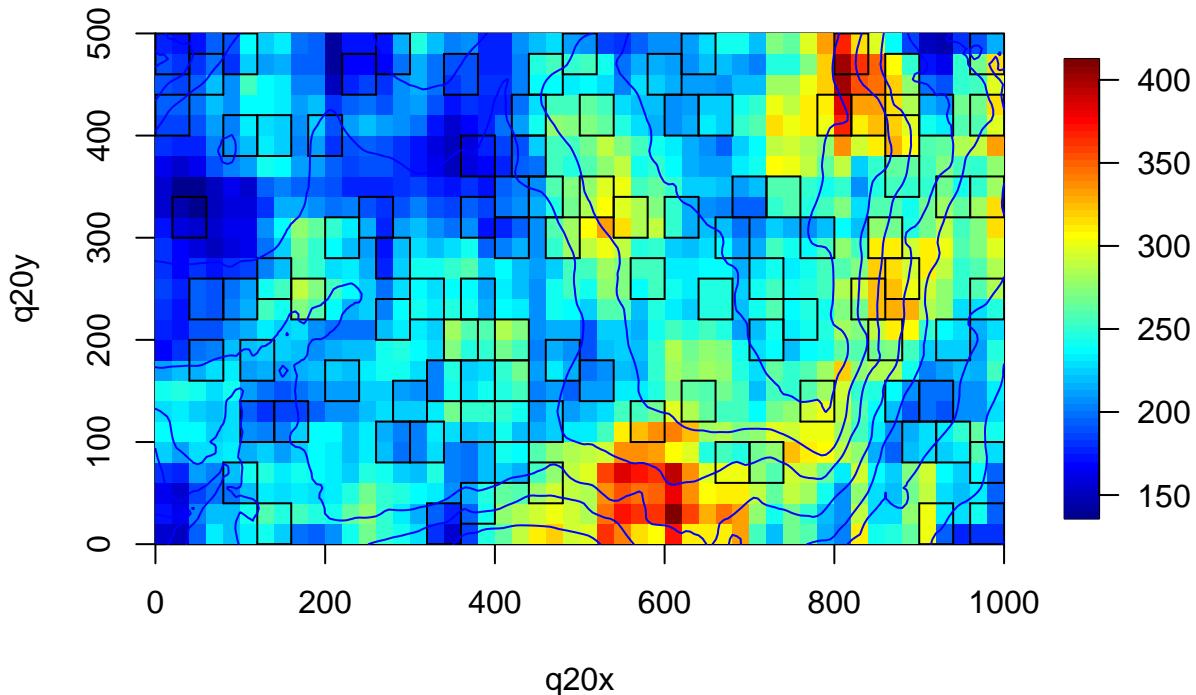
```
image.plot(q20x,q20y,agb2015by1ha,main="1-ha mean AGB density (Mg/ha) in 2015 & contours & habitat types")
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)
```

## -ha mean AGB density (Mg/ha) in 2015 & contours & habitat types



```
image.plot(q20x,q20y,agb2015by1ha,main="1-ha mean AGB density (Mg/ha) in 2015 & Dendrometer subplots")
for (i in 1:nrow(dend40centers)) {
  lines(rep(dend40centers$x[i]-20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(rep(dend40centers$x[i]+20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]-20,2))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]+20,2))
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

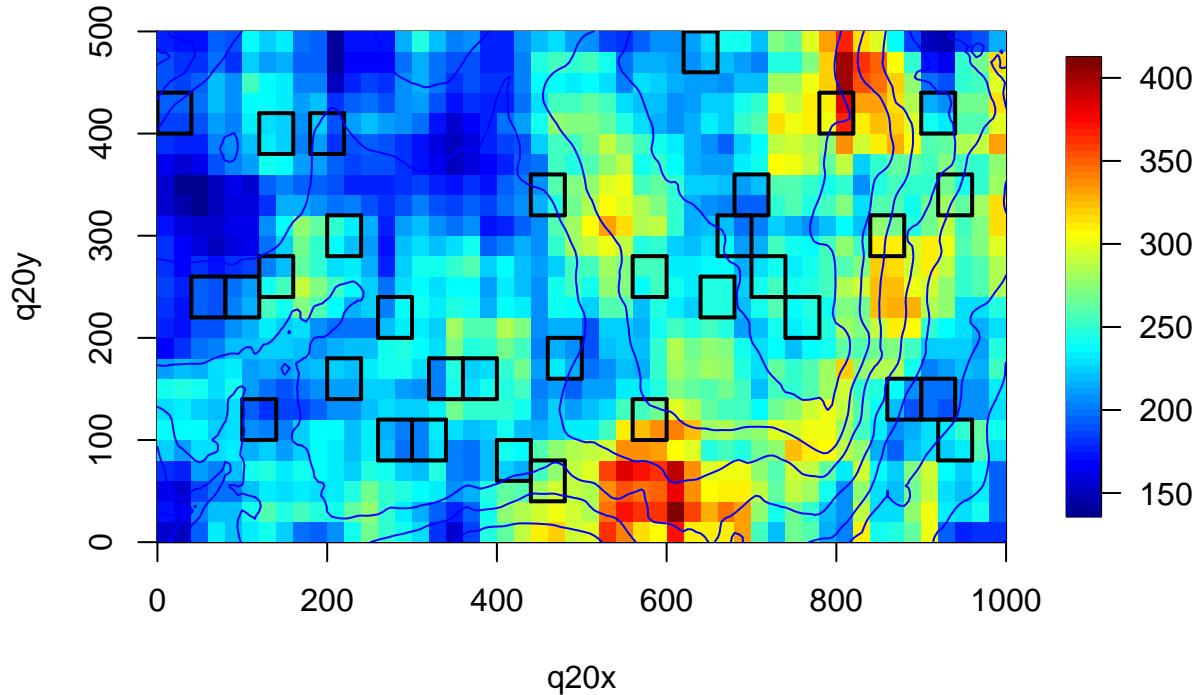
## 1-ha mean AGB density (Mg/ha) in 2015 & Dendrometer subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)

image.plot(q20x,q20y,argb2015by1ha,main="1-ha mean AGB density (Mg/ha) in 2015 & TLS 2019 subplots")
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

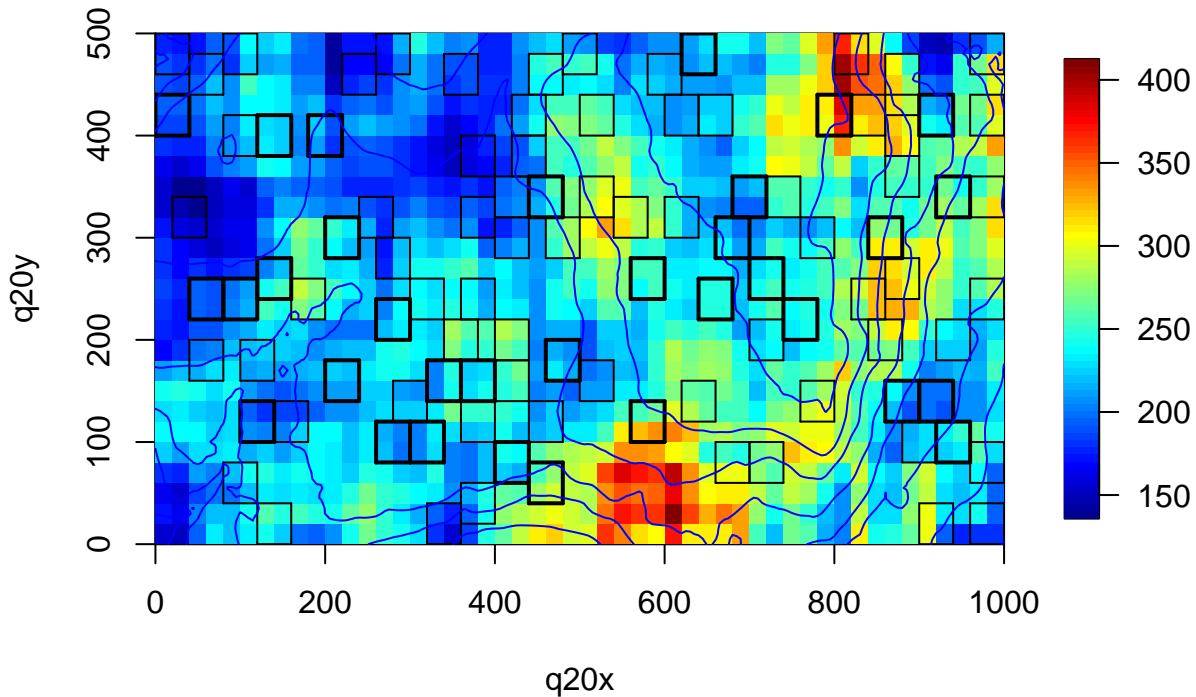
## 1-ha mean AGB density (Mg/ha) in 2015 & TLS 2019 subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)

image.plot(q20x,q20y,agb2015by1ha,main="1-ha mean AGB density (Mg/ha) in 2015 & All subplots")
for (i in 1:nrow(dend40centers)) {
  lines(rep(dend40centers$x[i]-20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(rep(dend40centers$x[i]+20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]-20,2))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]+20,2))
}
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

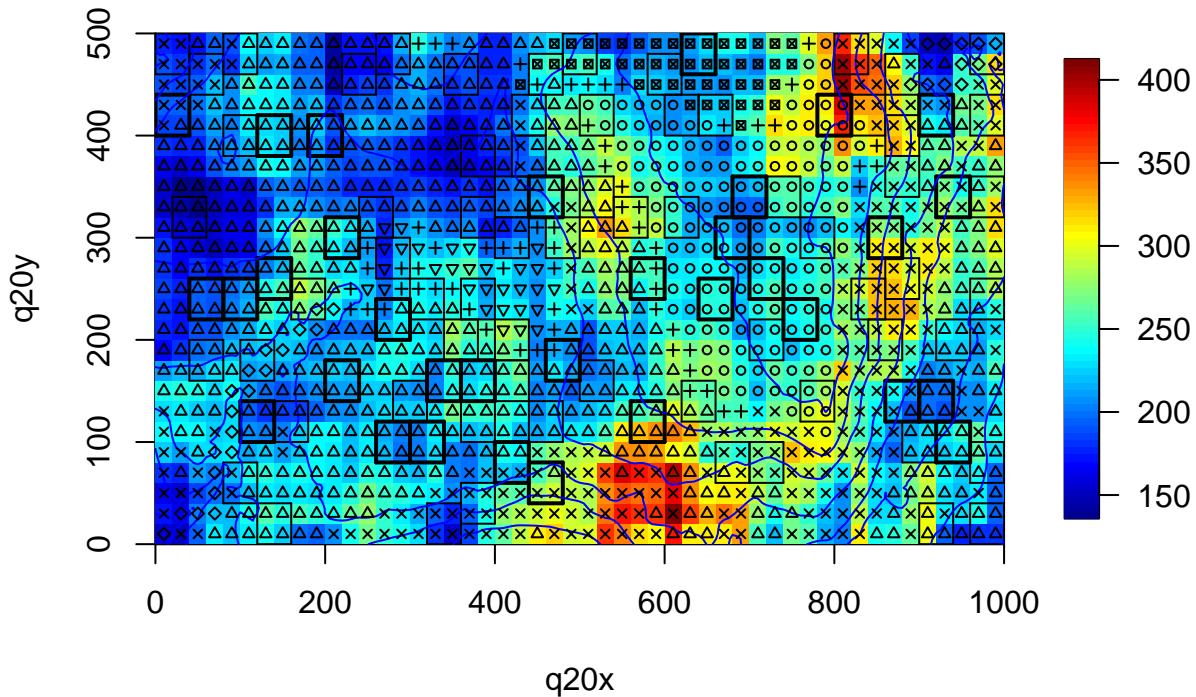
## 1-ha mean AGB density (Mg/ha) in 2015 & All subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)

image.plot(q20x,q20y,argb2015by1ha,main="1-ha mean AGB density (Mg/ha) in 2015 & All subplots & Habitat"
for (i in 1:nrow(dend40centers)) {
lines(rep(dend40centers$x[i]-20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
lines(rep(dend40centers$x[i]+20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]-20,2))
lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]+20,2))
}
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)
```

## ha mean AGB density (Mg/ha) in 2015 & All subplots & Habitat types



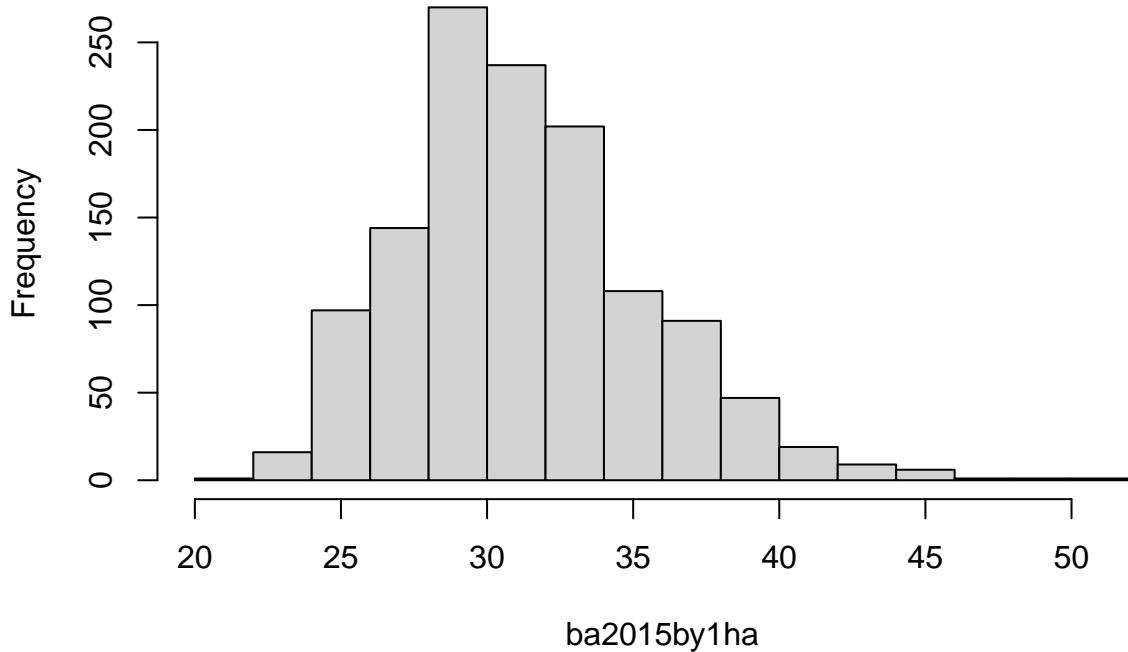
Figures of BA density at 1-ha scales

```
quantile(ba2015by1ha,probs=seq(0.05,0.95,by=0.1))
```

```
##      5%     15%     25%     35%     45%     55%     65%     75%
## 25.15414 27.19077 28.54048 29.39339 30.27327 31.23739 32.35891 33.68633
##      85%     95%
## 35.59069 38.64826
```

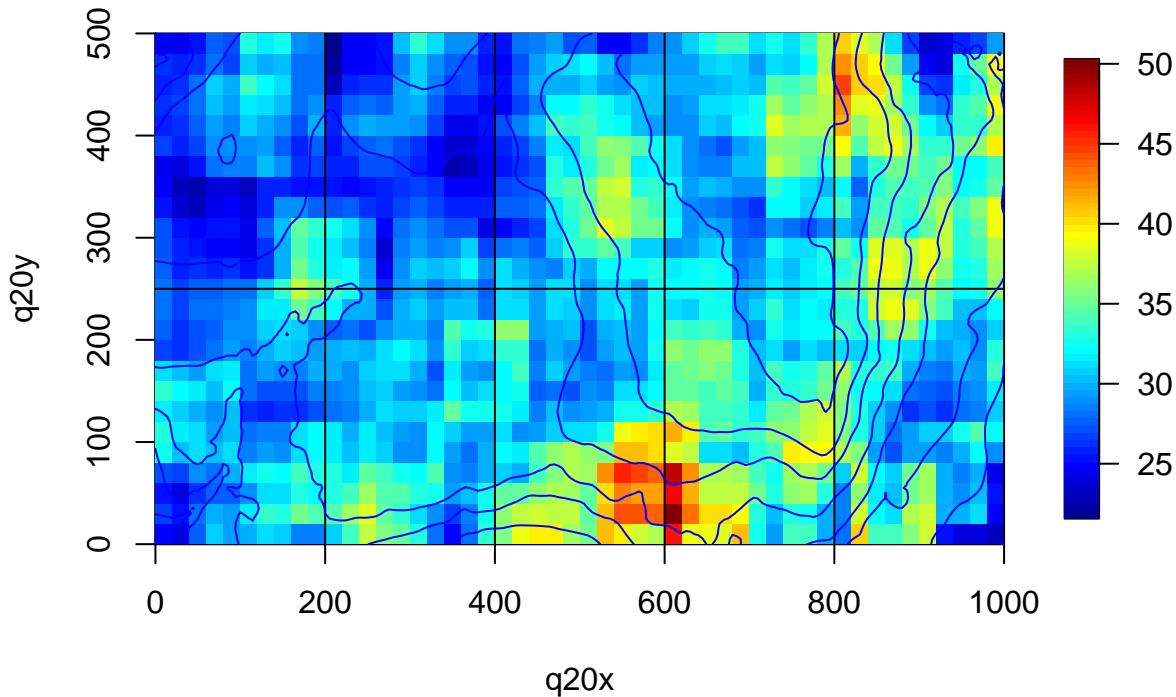
```
hist(ba2015by1ha)
```

## Histogram of ba2015by1ha



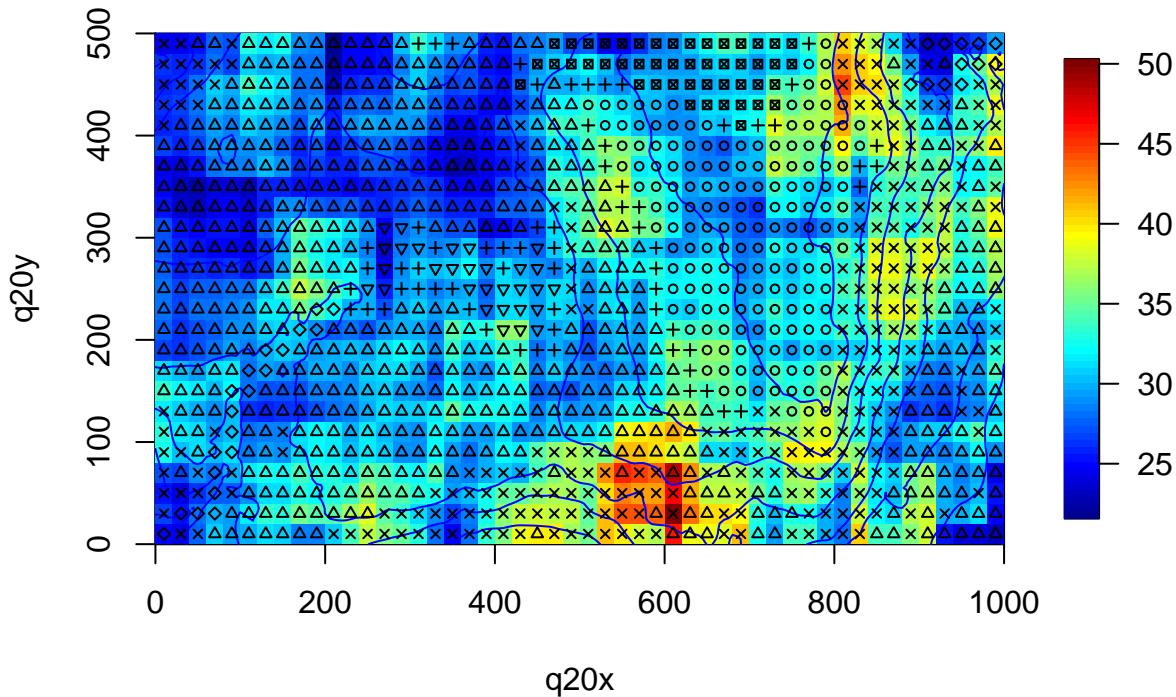
```
image.plot(q20x,q20y,ba2015by1ha,main="1-ha mean BA density (m2/ha) in 2015")
abline(h=250)
abline(v=c(200,400,600,800,1000))
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

### 1-ha mean BA density (m<sup>2</sup>/ha) in 2015



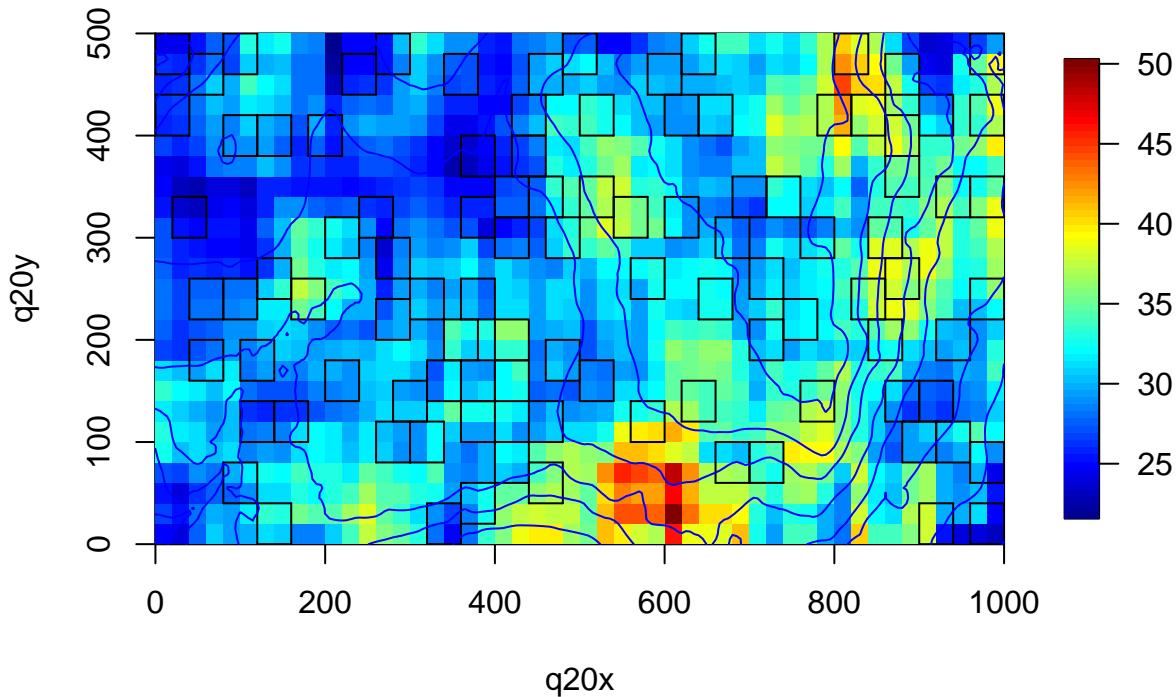
```
image.plot(q20x,q20y,ba2015by1ha,main="1-ha mean BA density (m2/ha) in 2015 & contours & habitat types")
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)
```

## 1-ha mean BA density (m<sup>2</sup>/ha) in 2015 & contours & habitat types



```
image.plot(q20x,q20y,ba2015by1ha,main="1-ha mean BA density (m2/ha) in 2015 & Dendrometer subplots")
for (i in 1:nrow(dend40centers)) {
  lines(rep(dend40centers$x[i]-20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(rep(dend40centers$x[i]+20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]-20,2))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]+20,2))
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

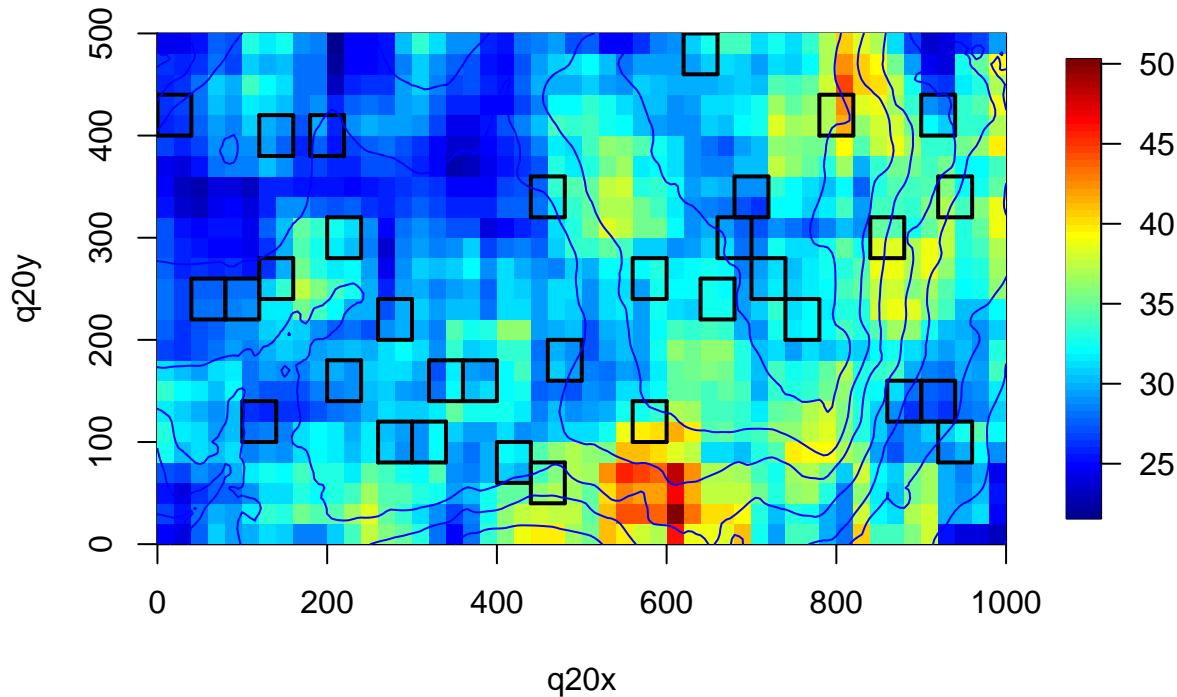
## 1-ha mean BA density (m<sup>2</sup>/ha) in 2015 & Dendrometer subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)

image.plot(q20x,q20y,ba2015by1ha,main="1-ha mean BA density (m2/ha) in 2015 & TLS 2019 subplots")
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

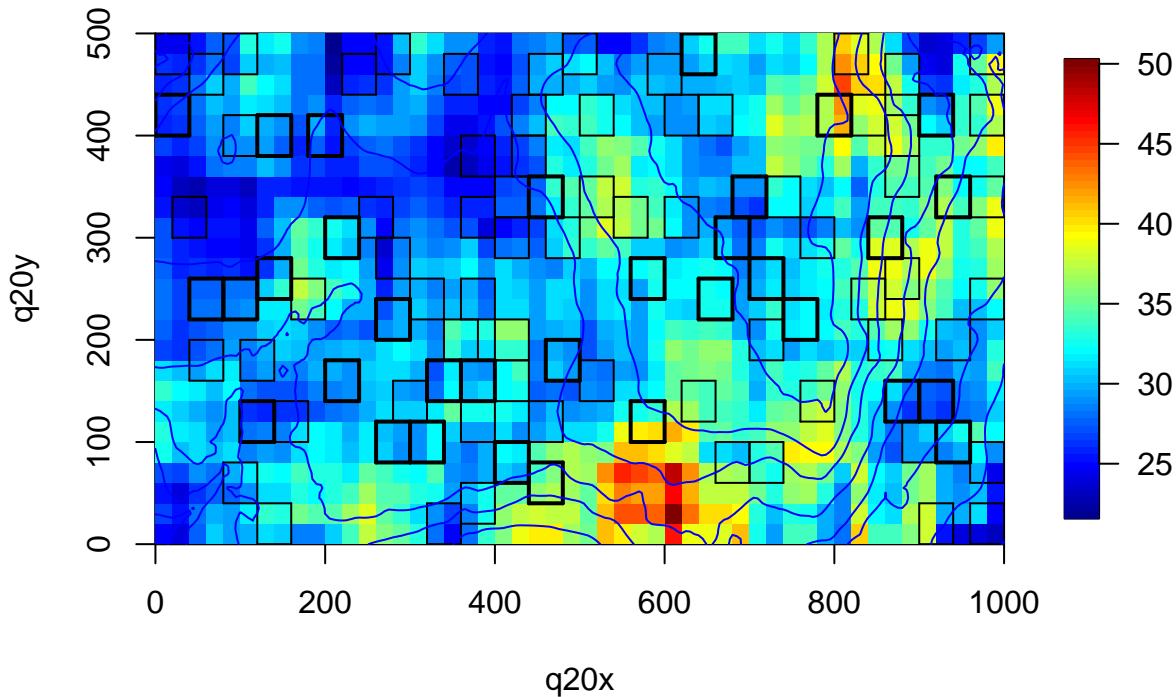
## 1-ha mean BA density (m<sup>2</sup>/ha) in 2015 & TLS 2019 subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)

image.plot(q20x,q20y,ba2015by1ha,main="1-ha mean BA density (m2/ha) in 2015 & All subplots")
for (i in 1:nrow(dend40centers)) {
  lines(rep(dend40centers$x[i]-20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(rep(dend40centers$x[i]+20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]-20,2))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]+20,2))
}
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

## 1-ha mean BA density (m<sup>2</sup>/ha) in 2015 & All subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)
```

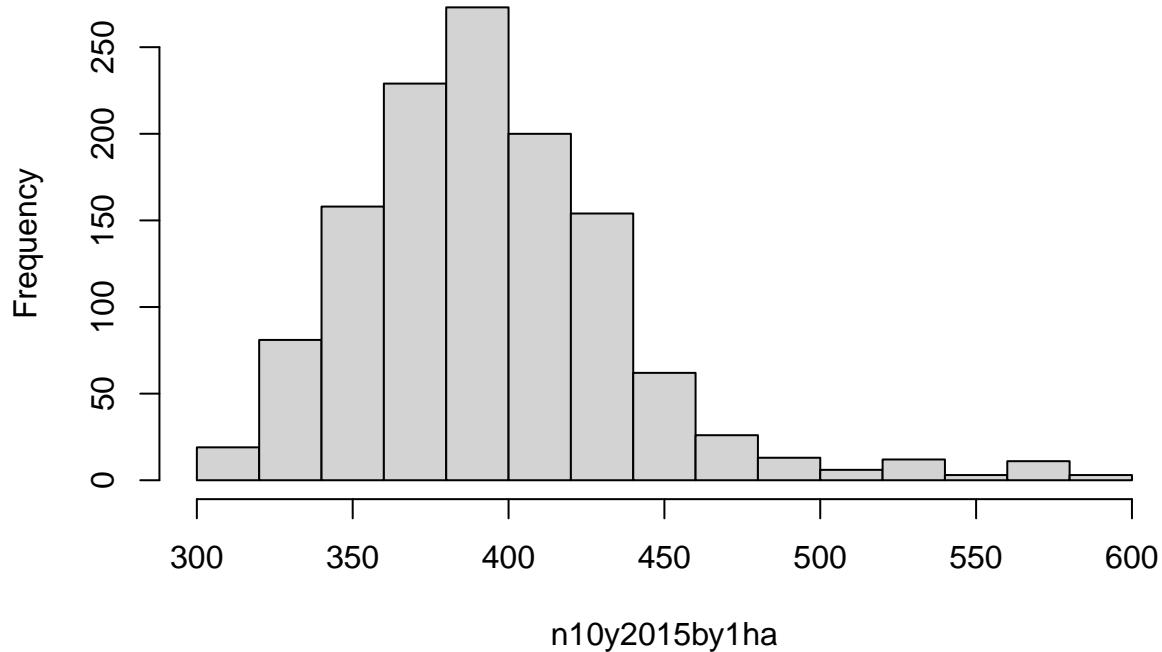
Figures of N>10 density at 1-ha scales

```
quantile(ba2015by1ha,probs=seq(0.05,0.95,by=0.1))
```

```
##      5%      15%      25%      35%      45%      55%      65%      75%
## 25.15414 27.19077 28.54048 29.39339 30.27327 31.23739 32.35891 33.68633
##     85%     95%
## 35.59069 38.64826
```

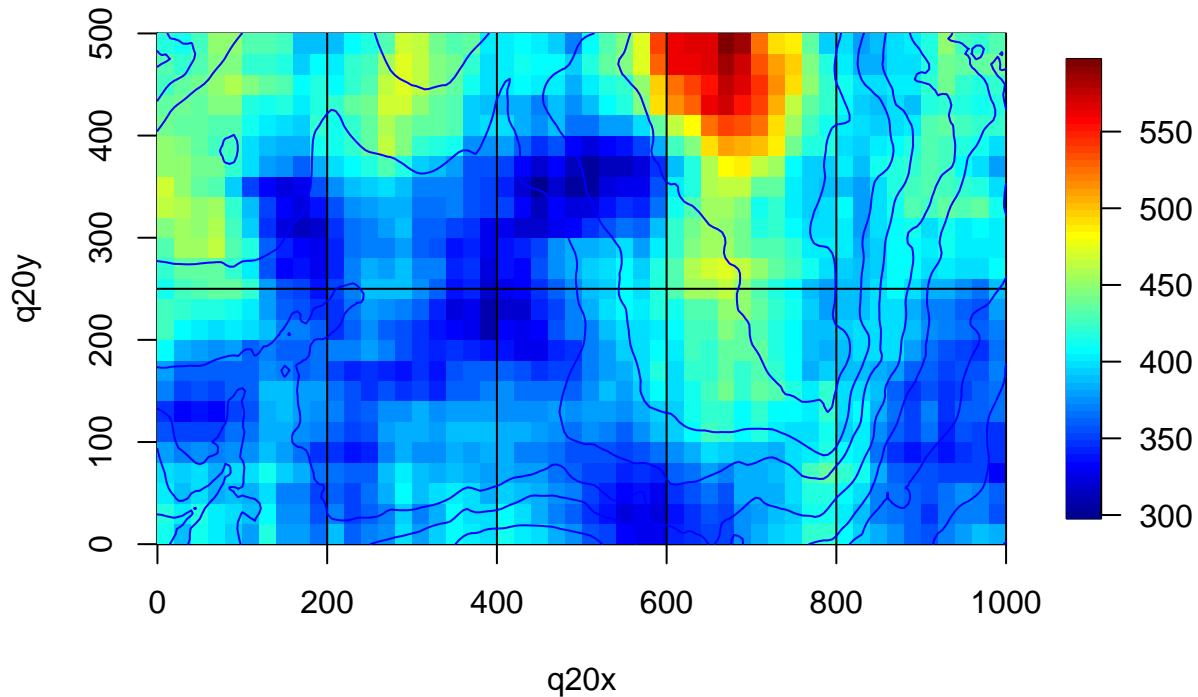
```
hist(n10y2015by1ha)
```

## Histogram of n10y2015by1ha



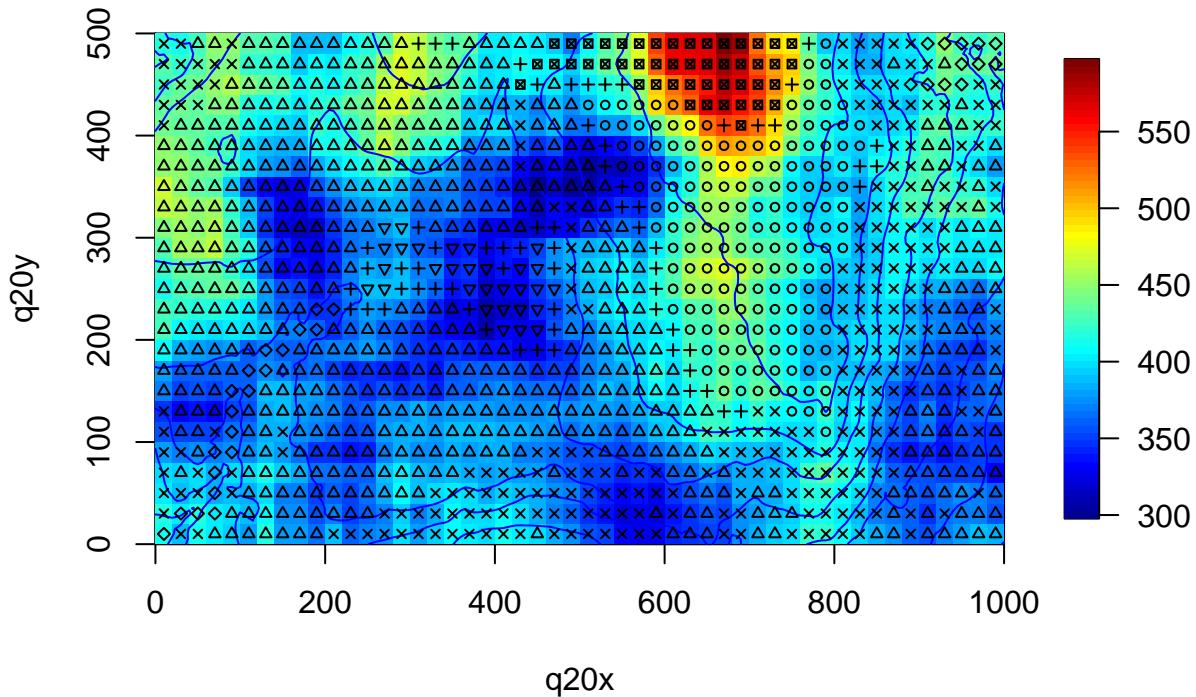
```
image.plot(q20x,q20y,n10y2015by1ha,main="1-ha mean N>10cm density (/ha) in 2015")
abline(h=250)
abline(v=c(200,400,600,800,1000))
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

### 1-ha mean N>10cm density (/ha) in 2015



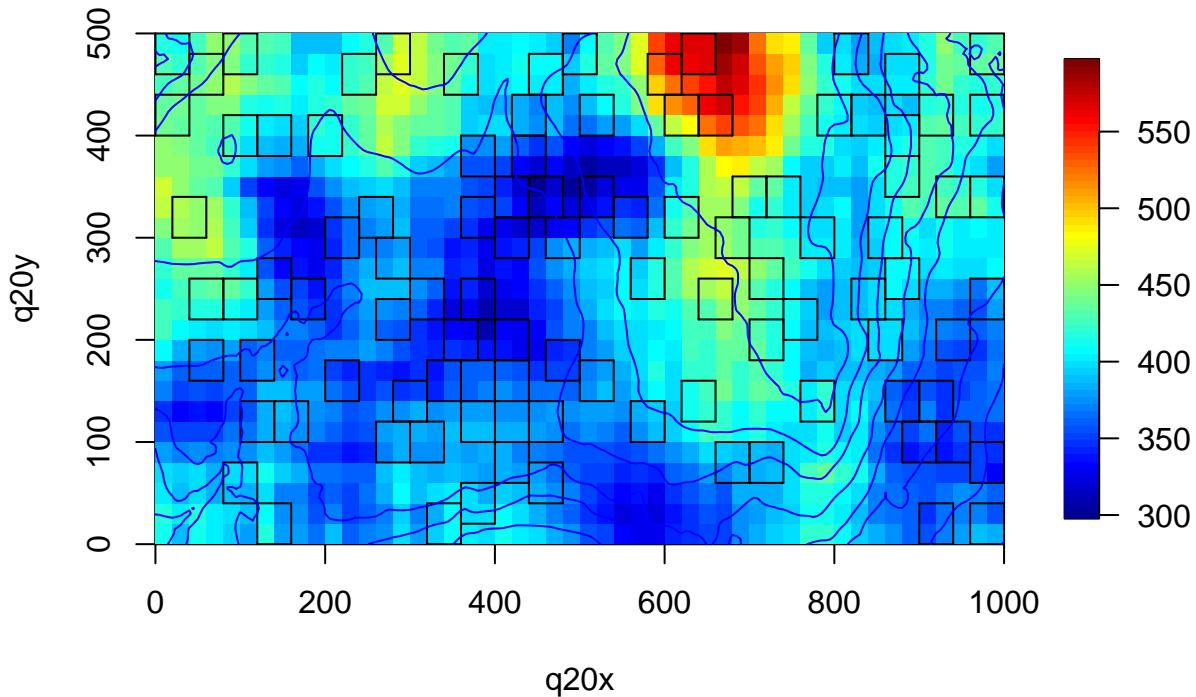
```
image.plot(q20x,q20y,n10y2015by1ha,main="1-ha mean N>10cm density (/ha) in 2015 & contours & habitat type")
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)
```

## -ha mean N>10cm density (/ha) in 2015 & contours & habitat types



```
image.plot(q20x,q20y,n10y2015by1ha,main="1-ha mean N>10cm density (/ha) in 2015 & Dendrometer subplots")
for (i in 1:nrow(dend40centers)) {
  lines(rep(dend40centers$x[i]-20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(rep(dend40centers$x[i]+20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]-20,2))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]+20,2))
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

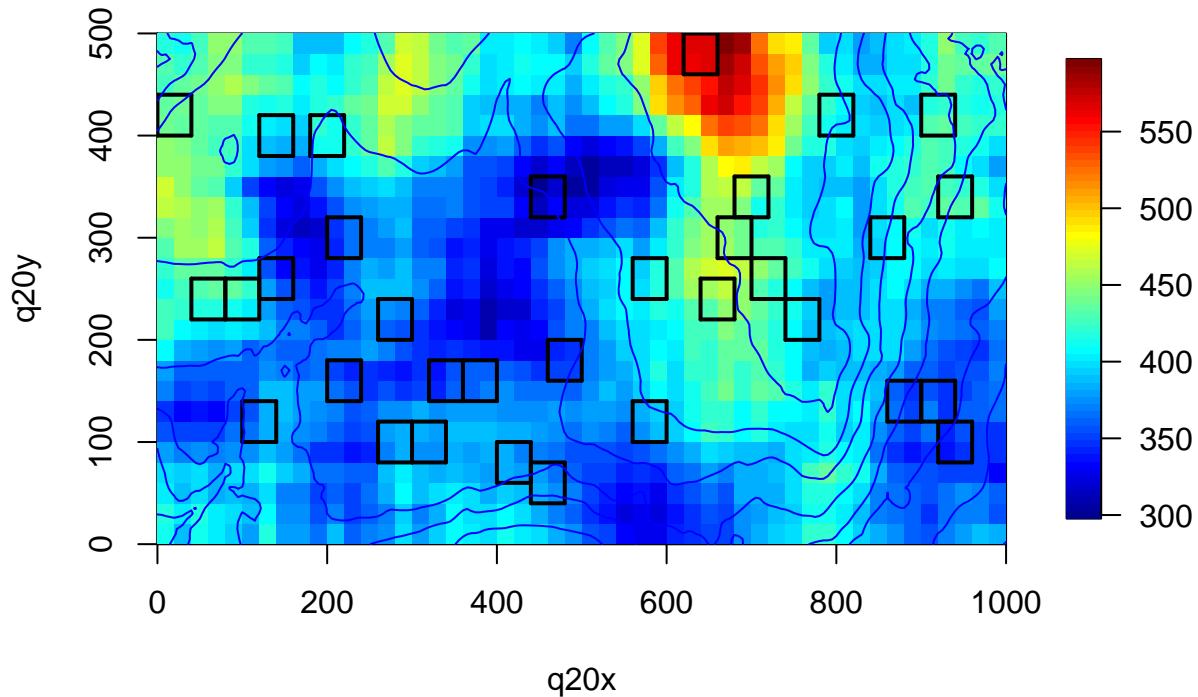
## 1-ha mean N>10cm density (/ha) in 2015 & Dendrometer subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)

image.plot(q20x,q20y,n10y2015by1ha,main="1-ha mean N>10cm density (/ha) in 2015 & TLS 2019 subplots")
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

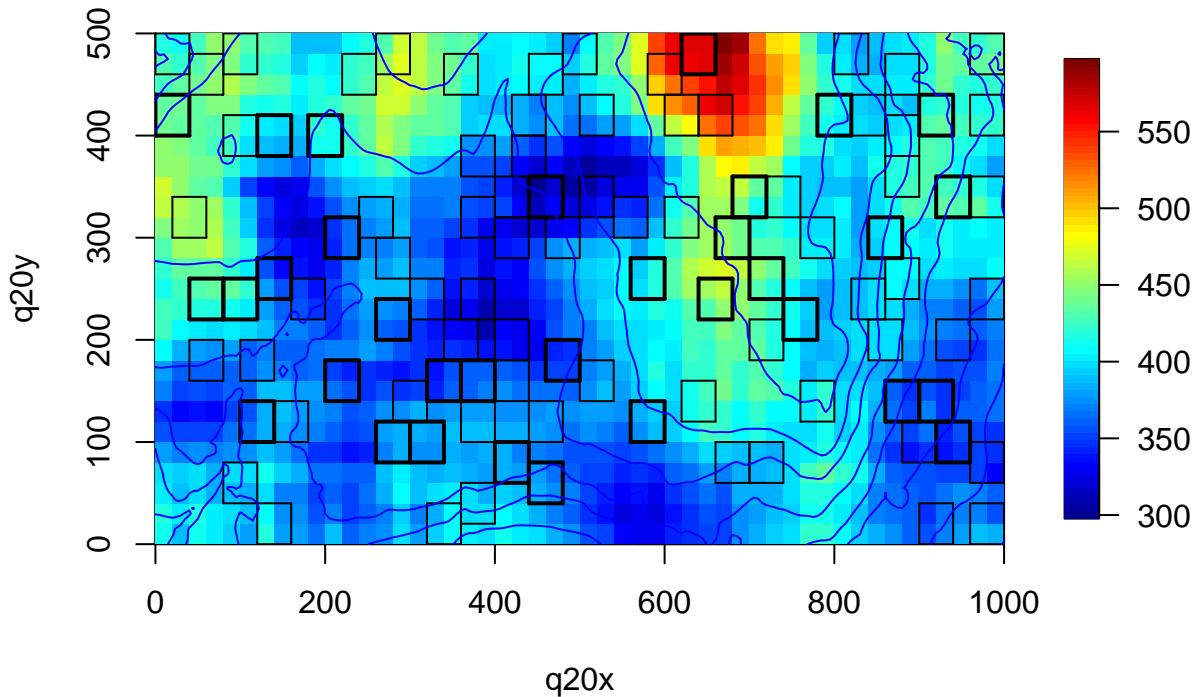
## 1-ha mean N>10cm density (/ha) in 2015 & TLS 2019 subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)

image.plot(q20x,q20y,n10y2015by1ha,main="1-ha mean N>10cm density (/ha) in 2015 & All subplots")
for (i in 1:nrow(dend40centers)) {
  lines(rep(dend40centers$x[i]-20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(rep(dend40centers$x[i]+20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]-20,2))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]+20,2))
}
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

## 1-ha mean N>10cm density (/ha) in 2015 & All subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)
```

## Figures of mean canopy height at 1-ha scales

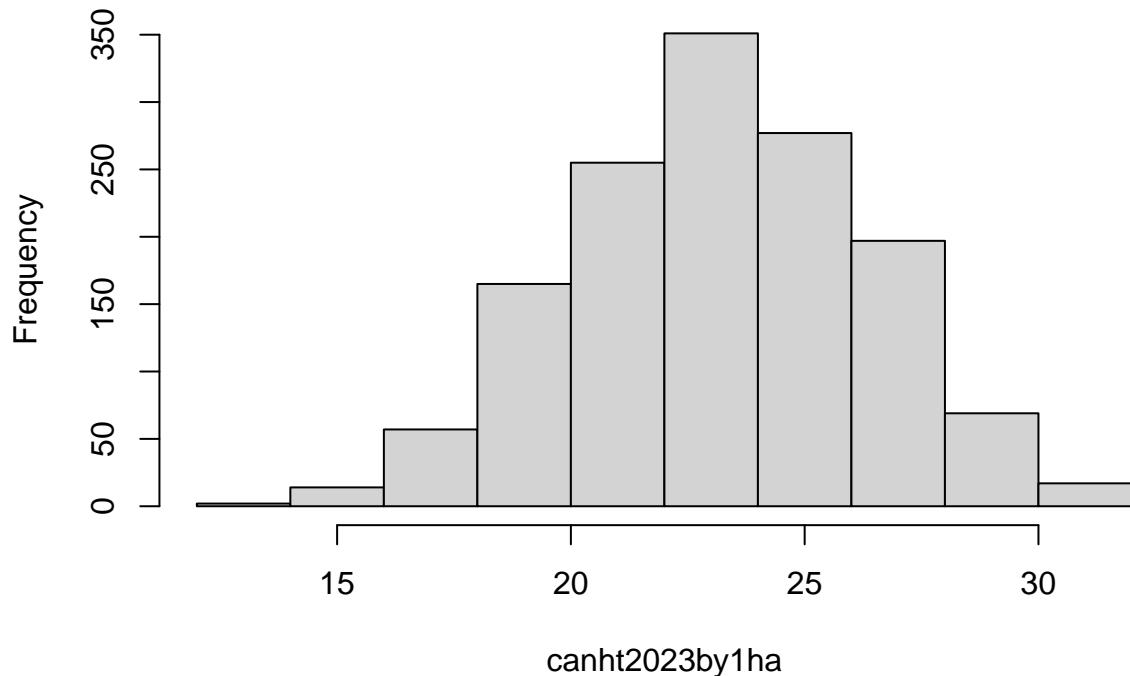
```
# recall that q20canht2023 is a data.frame
# x20s is the xs, y20s is the ys
# mnch23q20 is a matrix
# and importantly, this matrix extends past the plot boundaries

quantile(canht2023by1ha,probs=seq(0.05,0.95,by=0.1))

##          5%         15%        25%        35%        45%        55%        65%        75%
## 17.92859 19.78790 21.06430 21.99211 22.73807 23.53878 24.46520 25.45675
##          85%         95%
## 26.55740 28.22318

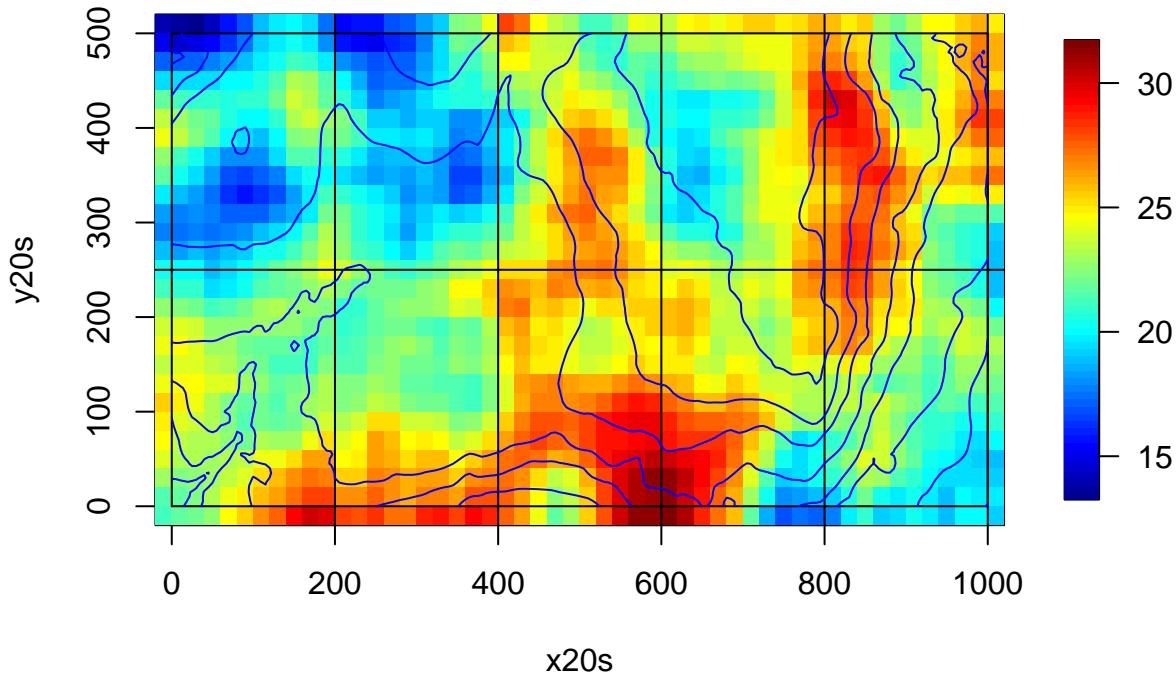
hist(canht2023by1ha)
```

### Histogram of canht2023by1ha



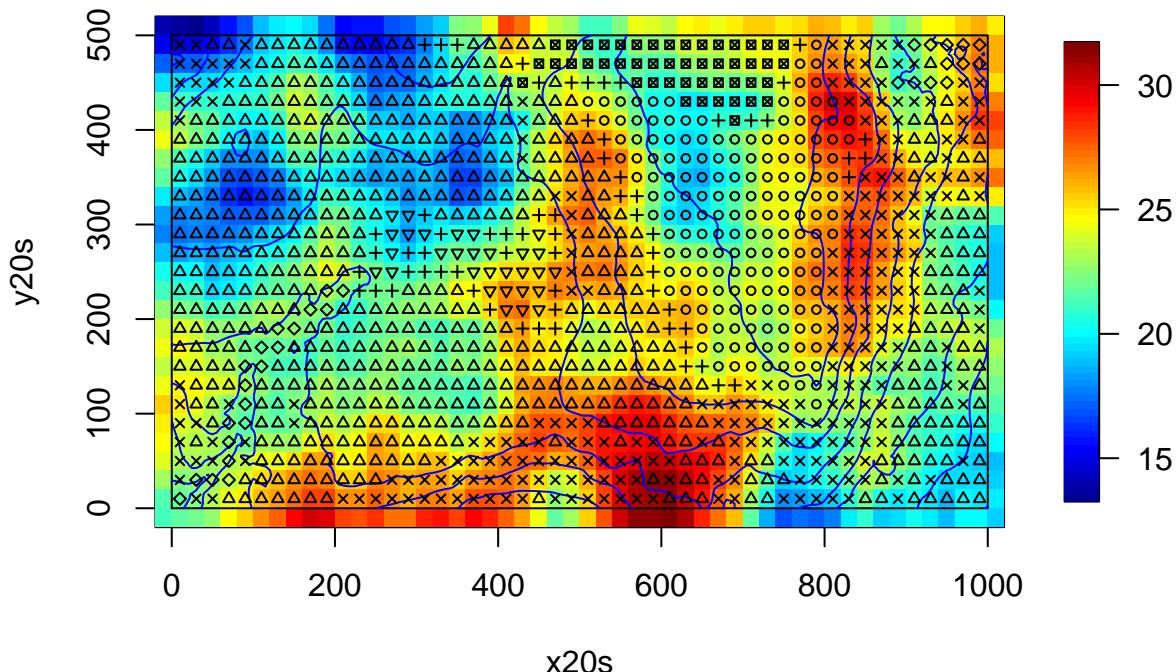
```
image.plot(x20s,y20s,canht2023by1ha,main="1-ha mean Canopy Height (m) in 2023")
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))
abline(h=250)
abline(v=c(200,400,600,800,1000))
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

### 1-ha mean Canopy Height (m) in 2023



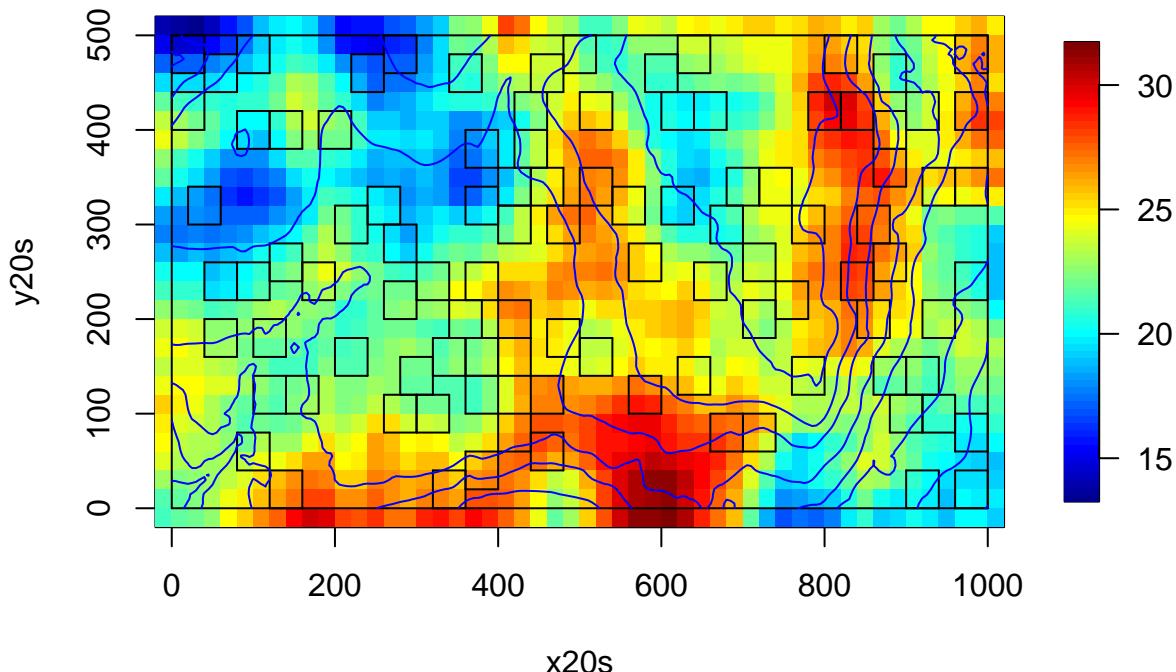
```
image.plot(x20s,y20s,canht2023by1ha,main="1-ha mean Canopy Height (m) in 2023 & contours & habitat types"
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)
```

## 1-ha mean Canopy Height (m) in 2023 & contours & habitat types



```
image.plot(x20s,y20s,canht2023by1ha,main="1-ha mean N>10cm density (/ha) in 2015 & Dendrometer subplots
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))
for (i in 1:nrow(dend40centers)) {
  lines(rep(dend40centers$x[i]-20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(rep(dend40centers$x[i]+20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]-20,2))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]+20,2))
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

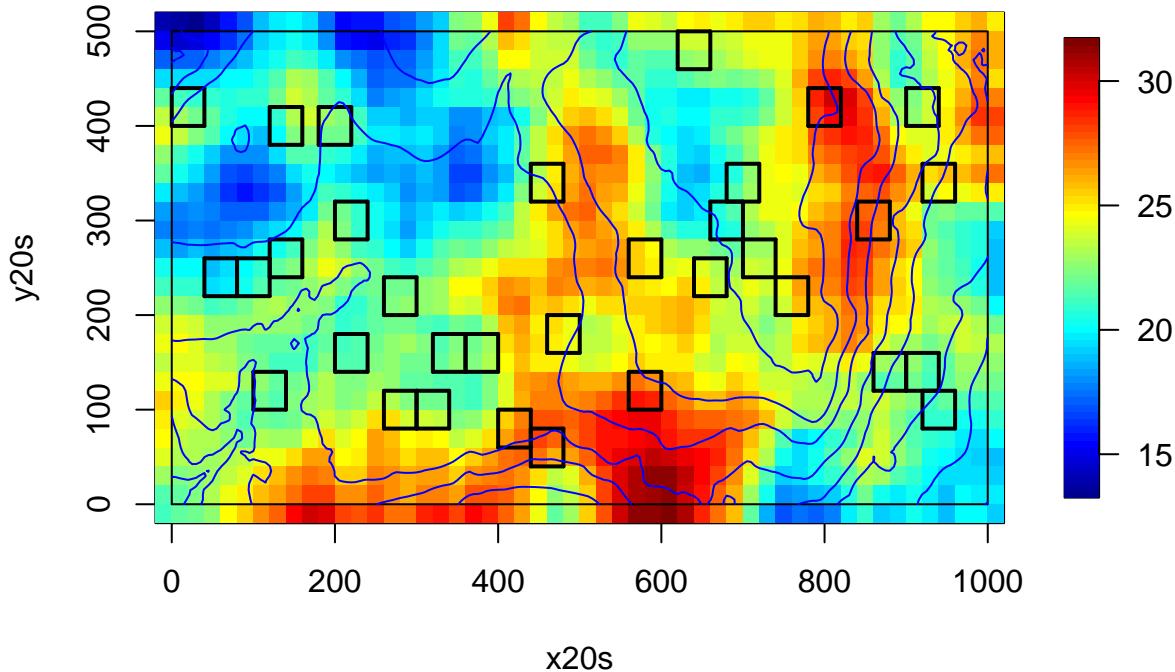
## 1-ha mean N>10cm density (/ha) in 2015 & Dendrometer subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)

image.plot(x20s,y20s,canht2023by1ha,main="1-ha mean Canopy Height (m) in 2023 & TLS 2019 subplots")
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

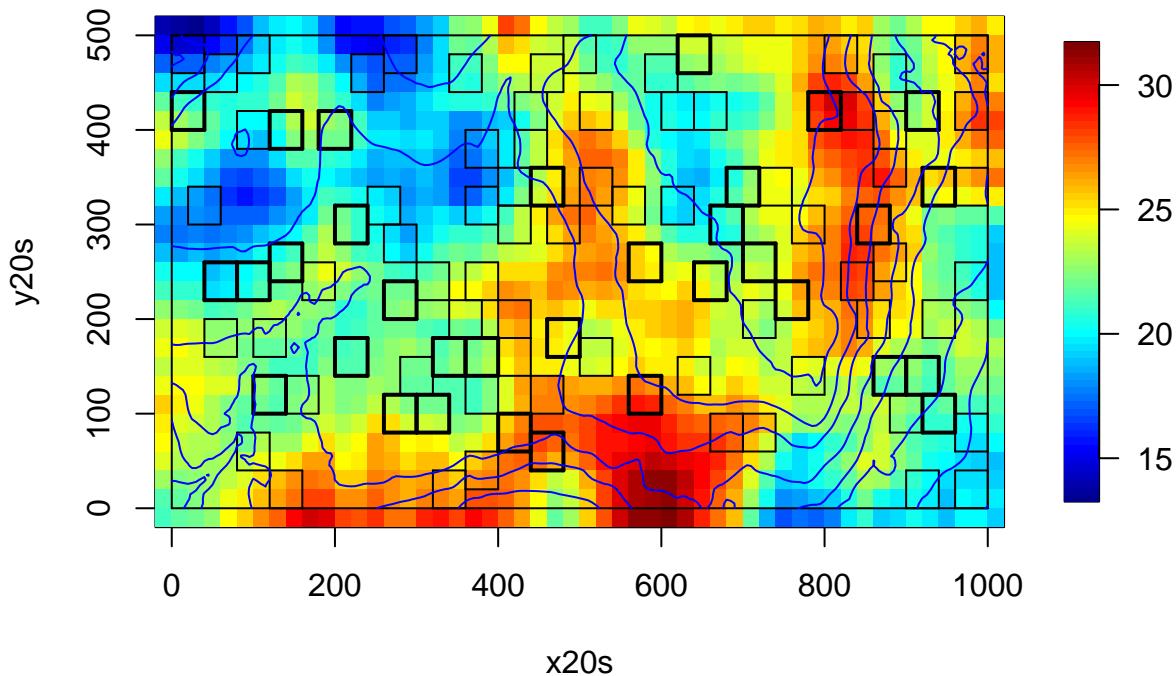
## 1-ha mean Canopy Height (m) in 2023 & TLS 2019 subplots



```
#points(bci_habitat$x,bci_habitat$y,pch=as.numeric(as.factor(bci_habitat$habitat)),cex=0.6)

image.plot(x20s,y20s,canht2023by1ha,main="1-ha mean Canopy Height (m) in 2023 & All subplots")
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))
for (i in 1:nrow(dend40centers)) {
  lines(rep(dend40centers$x[i]-20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(rep(dend40centers$x[i]+20,2),c(dend40centers$y[i]-20,dend40centers$y[i]+20))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]-20,2))
  lines(c(dend40centers$x[i]-20,dend40centers$x[i]+20),rep(dend40centers$y[i]+20,2))
}
for (i in 1:nrow(tls40centers)) {
  lines(x=rep(tls40centers$x[i]-20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=rep(tls40centers$x[i]+20,2), y=c(tls40centers$y[i]-20,tls40centers$y[i]+20),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]-20,2),lwd=2)
  lines(x=c(tls40centers$x[i]-20,tls40centers$x[i]+20), y=rep(tls40centers$y[i]+20,2),lwd=2)
}
contour(p5x,p5y,elevm5,add=T,nlevels=10,drawlabels=F,col="blue")
```

## 1-ha mean Canopy Height (m) in 2023 & All subplots



## Proposed locations of 1-ha plots

```

#newtlscenters=data.frame(x=c(690,650,870,550),y=c(270,450,270,50))
#newtlscenters=data.frame(x=c(650,870,710,370,310),y=c(450,270,230,390,130),
newtlscenters=data.frame(x=c(650,870,710),y=c(450,270,230),
                           n10=NA,ba=NA,argb=NA,percn10=NA,percba=NA,percagb=NA)
nnew <- length(newtlscenters$x)
cols <- c("red","blue","forest green", "purple","orange")
df1ha$newtls <- NA
for (i in 1:nnew)
  df1ha$newtls[df1ha$xcenter==newtlscenters$x[i] & df1ha$ycenter==newtlscenters$y[i]] <- i

# calculate corresponding q20 center points
newtls1hax <- floor(newtlscenters$x/20)+1
newtls1hay <- floor(newtlscenters$y/20)+1

for (i in 1:nnew) {
  newtlscenters$n10[i] <- n10y2015by1ha[newtls1hax[i],newtls1hay[i]]
  newtlscenters$ba[i] <- ba2015by1ha[newtls1hax[i],newtls1hay[i]]
  newtlscenters$agb[i] <- agb2015by1ha[newtls1hax[i],newtls1hay[i]]
  newtlscenters$canht[i] <- canht2023by1ha[newtls1hax[i]+1,newtls1hay[i]+1]
}

```

```

# canopy height matrix includes a 20-m buffer around plot, so coordinates offset by 1 quadrat
# following are percentiles (quantiles) of the variables

# for quantiles, the matrices already eliminate 2 rows around edge, so adjust coordinates
newtlscenters$n10quant[i] <- n10quant[newtls1hax[i]-2,newtls1hay[i]-2]
newtlscenters$baquant[i] <- baquant[newtls1hax[i]-2,newtls1hay[i]-2]
newtlscenters$agbquant[i] <- agbquant[newtls1hax[i]-2,newtls1hay[i]-2]
newtlscenters$canhtquant[i] <- canhtquant[newtls1hax[i]-2,newtls1hay[i]-2]
}

}

```

Function to make figures illustrating where TLS plots fall with respect to one focal variable

```

wheretls1var <- function(df1ha,zvarname,zvartitle) {
  df1ha <- df1ha[order(df1ha$ycenter,df1ha$xcenter),] # just make sure ordering is right
  xs <- sort(unique(df1ha$xcenter))
  ys <- sort(unique(df1ha$ycenter))
  focvar <- df1ha[,names(df1ha)==zvarname]
  quantfoc <- rank(focvar)/nrow(df1ha)
  focmat <- matrix(focvar,nrow=46,ncol=21)
  inc <- !is.na(df1ha$newtls)
  newtls1ha <- data.frame(newtls=df1ha$newtls[inc],focvar=focvar[inc],quantfoc=quantfoc[inc])
  nnewtls <- nrow(newtls1ha)

  # make a map of the variable, and note where the plots would be located
  image.plot(xs,ys,focmat,main=zvartitle)
  text(df1ha$xcenter[inc],df1ha$ycenter[inc],df1ha$newtls[inc])
  inc2 <- df1ha$nhabitat==1
  points(df1ha$xcenter[inc2],df1ha$ycenter[inc2],pch=16,cex=0.5)
  # the points show 1-ha subplots that are entirely in one habitat type

  # make a density plot of the variable, and note plot locations
  plot(density(focvar),main=zvartitle)
  for (i in 1:nnewtls) {
    abline(v=newtls1ha$focvar[i],col=cols[newtls1ha$newtls[i]])
    text(x=newtls1ha$focvar[i],y=0,newtls1ha$newtls[i],col=cols[newtls1ha$newtls[i]])
  }

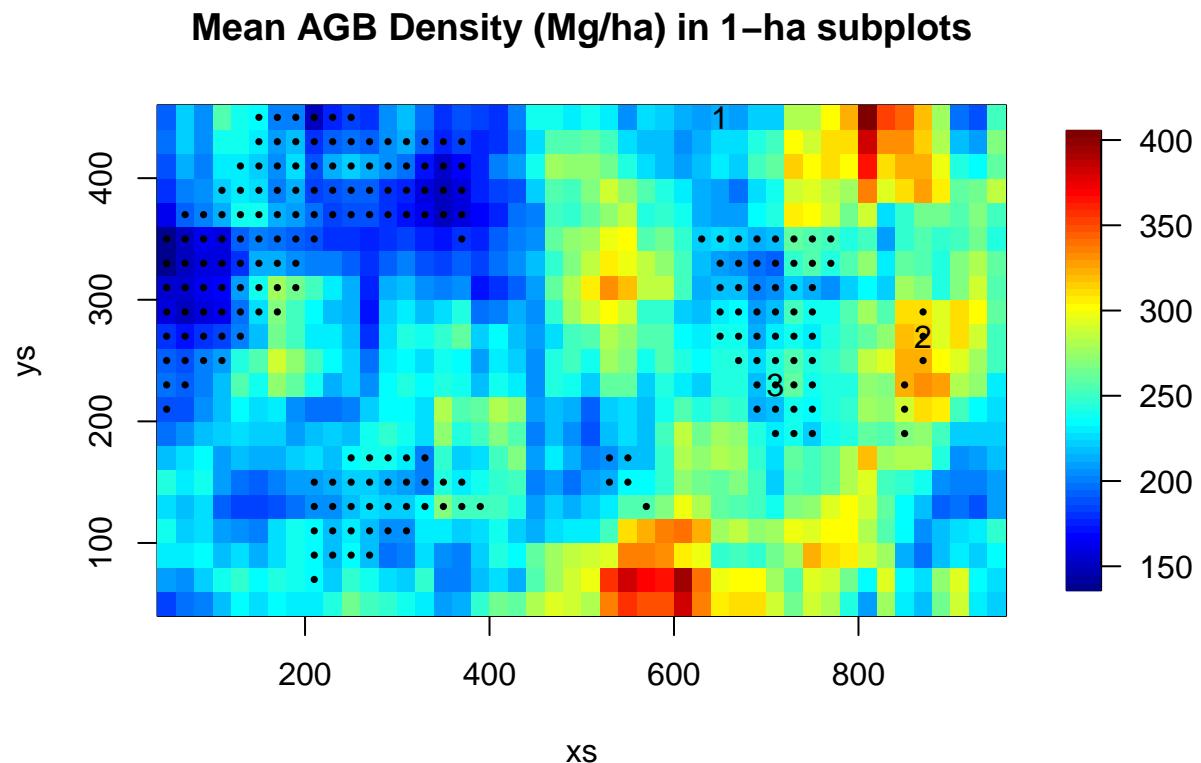
  # make a cumulative density plot of the variable, and note plot locations
  plot(ecdf(focvar),cex=0.1,main=zvartitle)
  for (i in 1:nnewtls) {
    abline(v=newtls1ha$focvar[i],col=cols[newtls1ha$newtls[i]])
    text(x=newtls1ha$focvar[i],y=0,newtls1ha$newtls[i],col=cols[newtls1ha$newtls[i]])
    abline(h=newtls1ha$quantfoc[i],col=cols[newtls1ha$newtls[i]])
  }
}

```

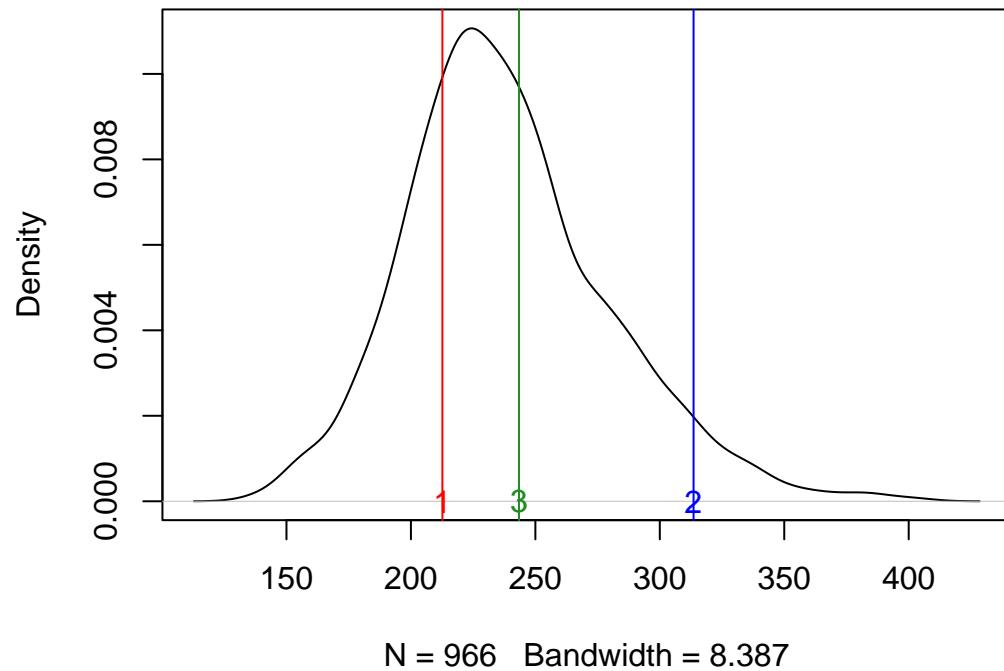
Figures showing subplot locations and quantiles for mean variables of interest.

Note that we want to subplots to span plot-level variation in the following mean variables (so better if more different in pdfs and cdfs).

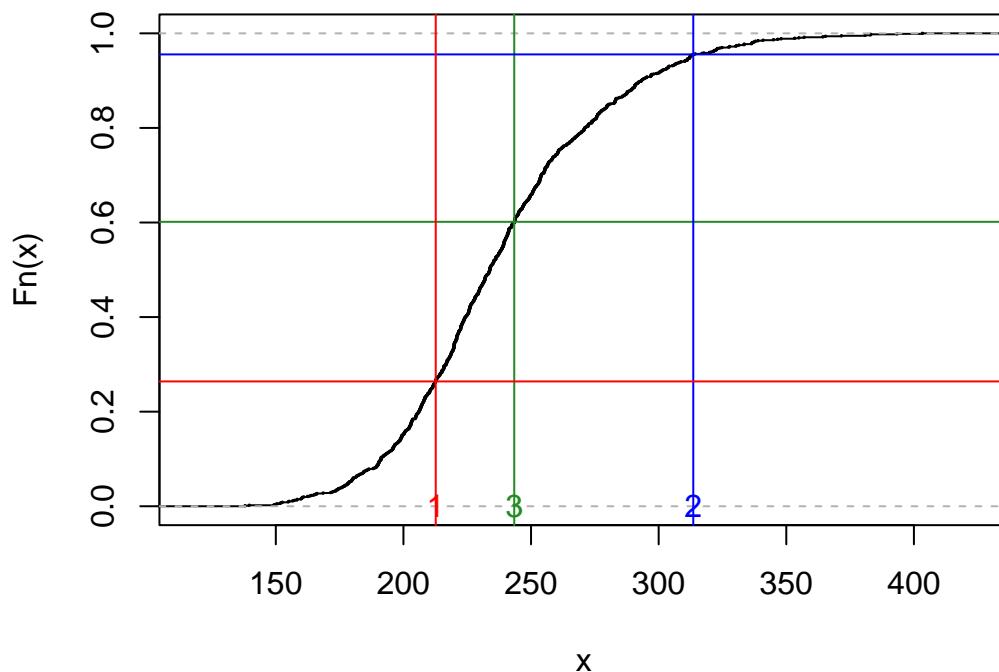
```
wheretls1var(df1ha,"mnagb","Mean AGB Density (Mg/ha) in 1-ha subplots")
```



### Mean AGB Density (Mg/ha) in 1-ha subplots

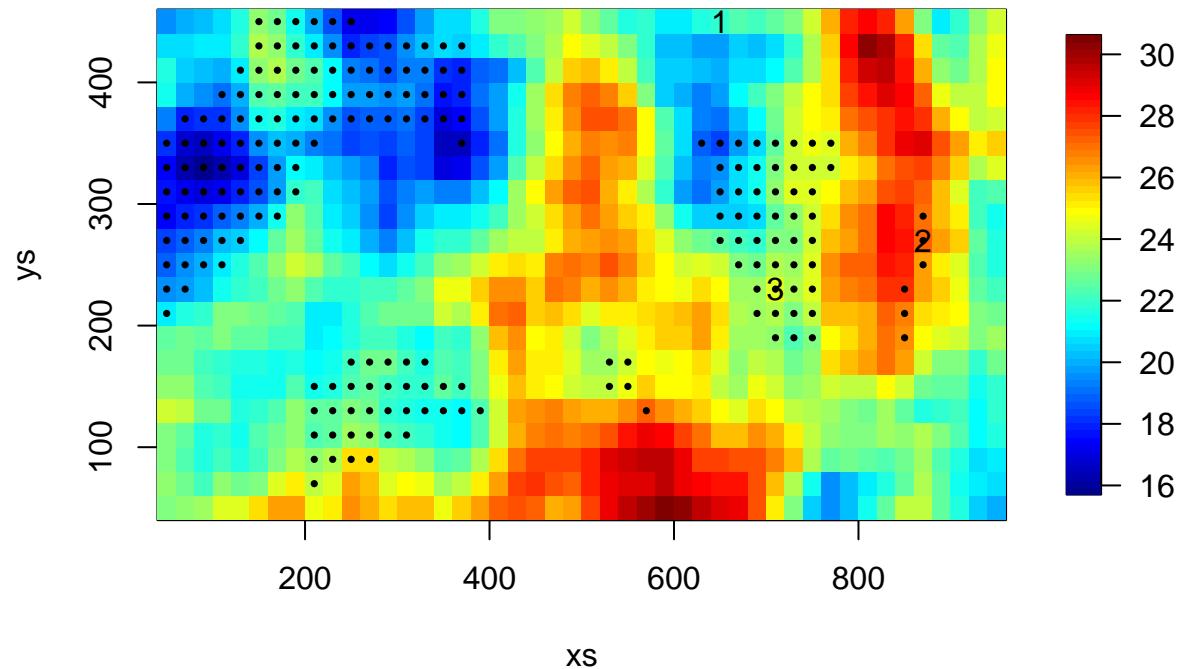


## Mean AGB Density (Mg/ha) in 1-ha subplots

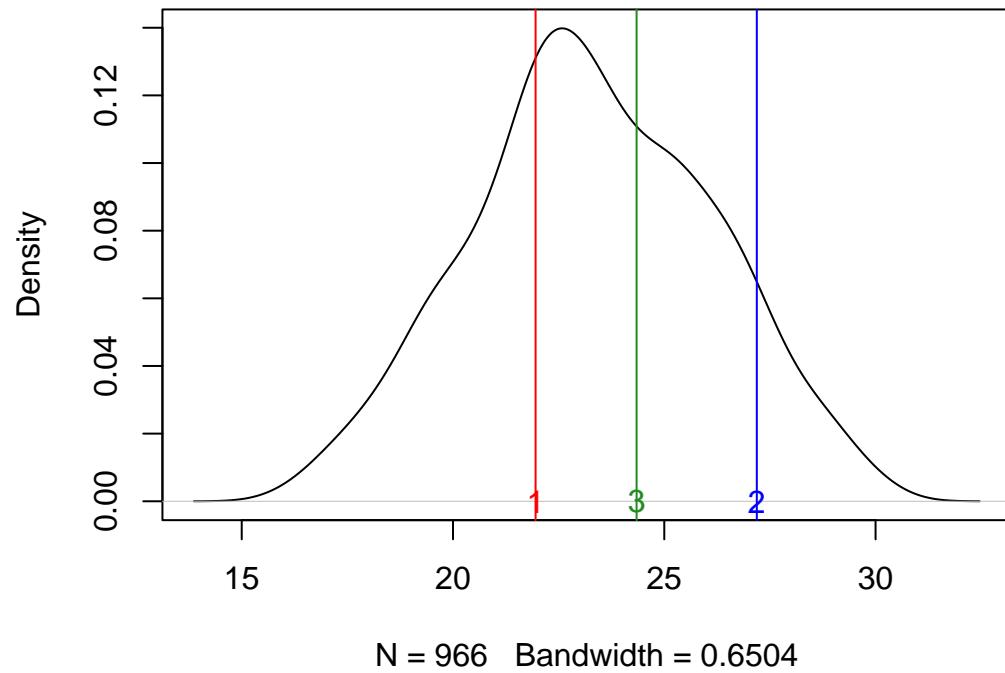


```
wheretlsivar(df1ha, "mncanht", "Mean Canopy Height in 1-ha subplots")
```

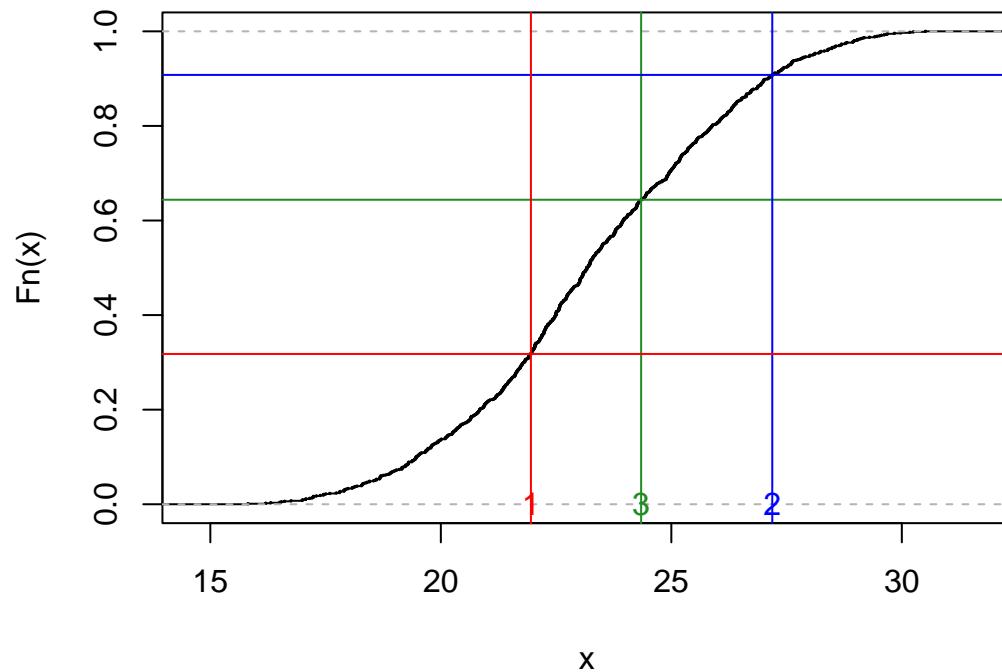
### Mean Canopy Height in 1-ha subplots



### Mean Canopy Height in 1-ha subplots

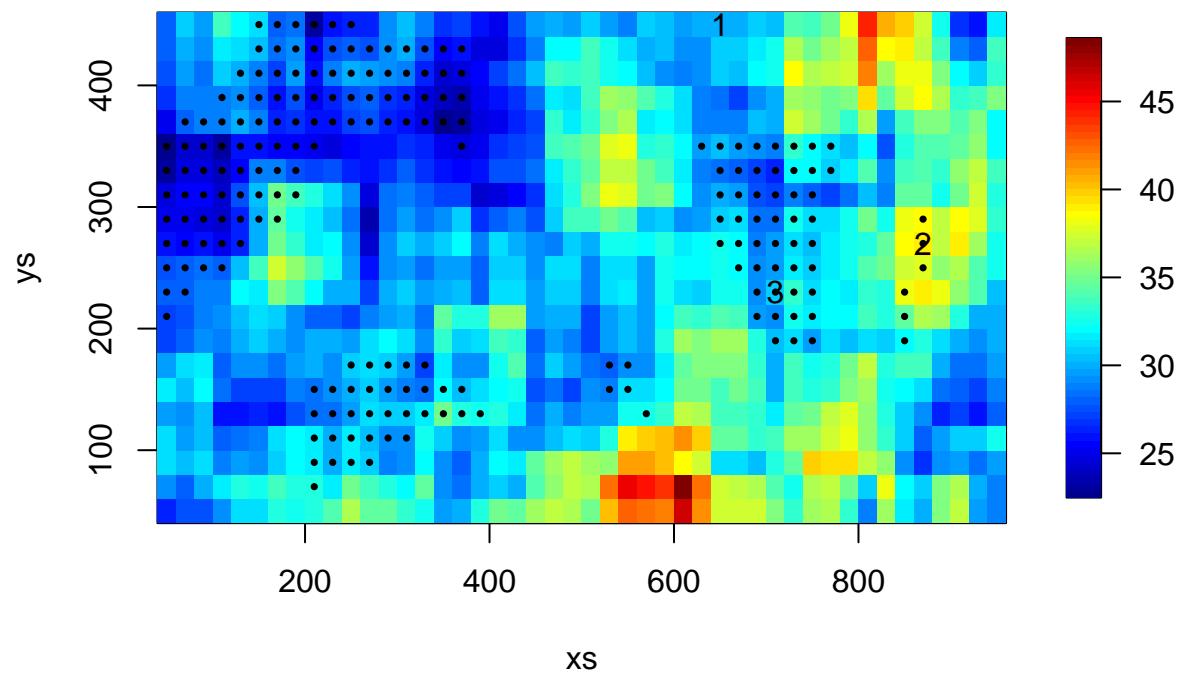


## Mean Canopy Height in 1-ha subplots

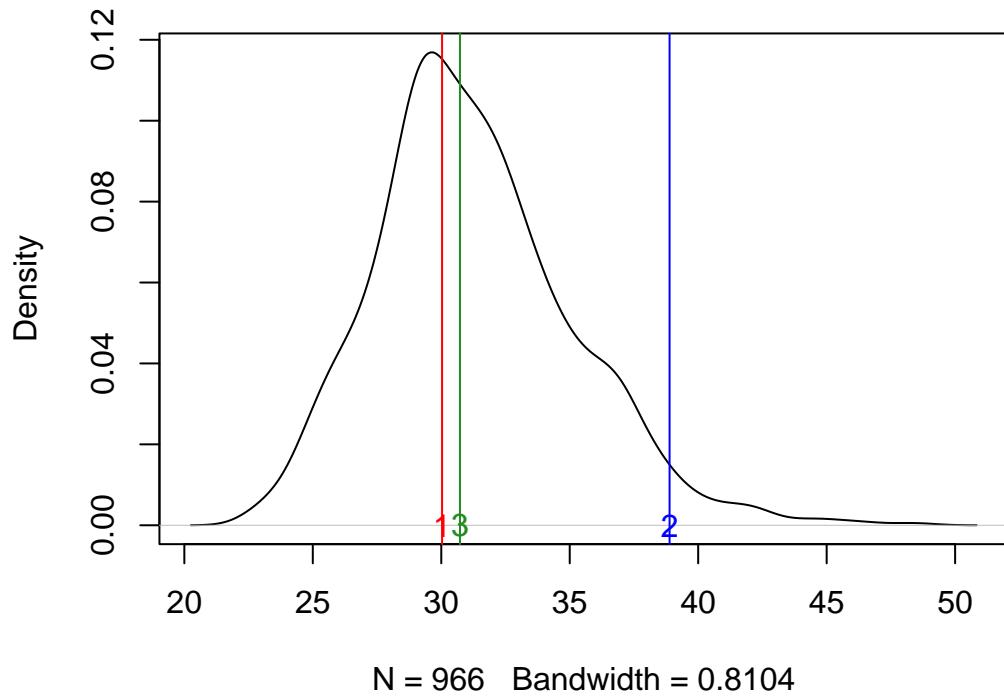


```
wheretlsivar(df1ha,"mnba","Mean Basal Area Density in 1-ha subplots")
```

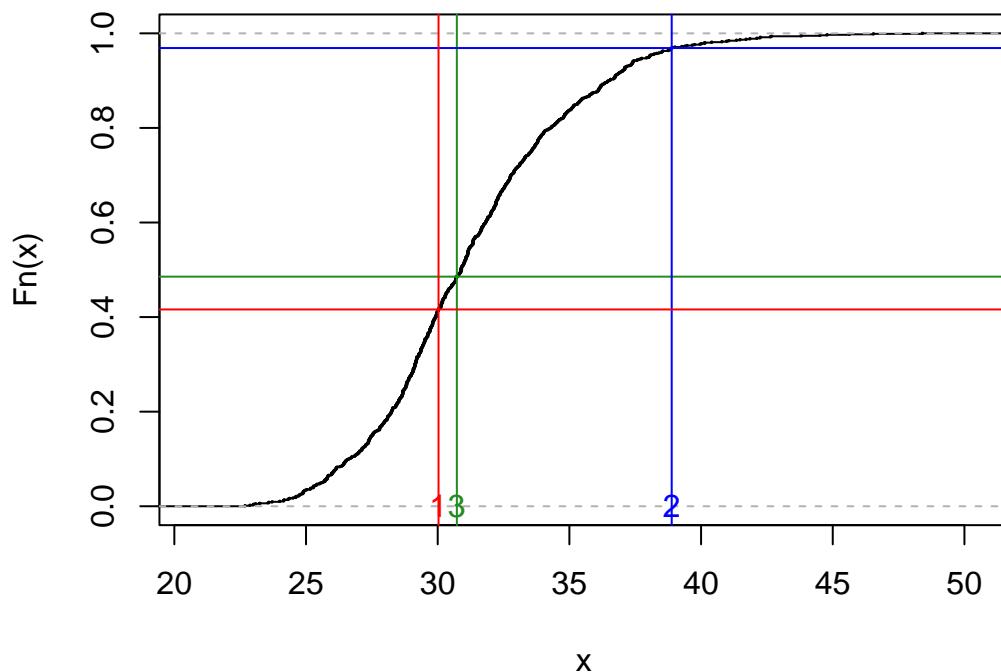
## Mean Basal Area Density in 1-ha subplots



### Mean Basal Area Density in 1-ha subplots

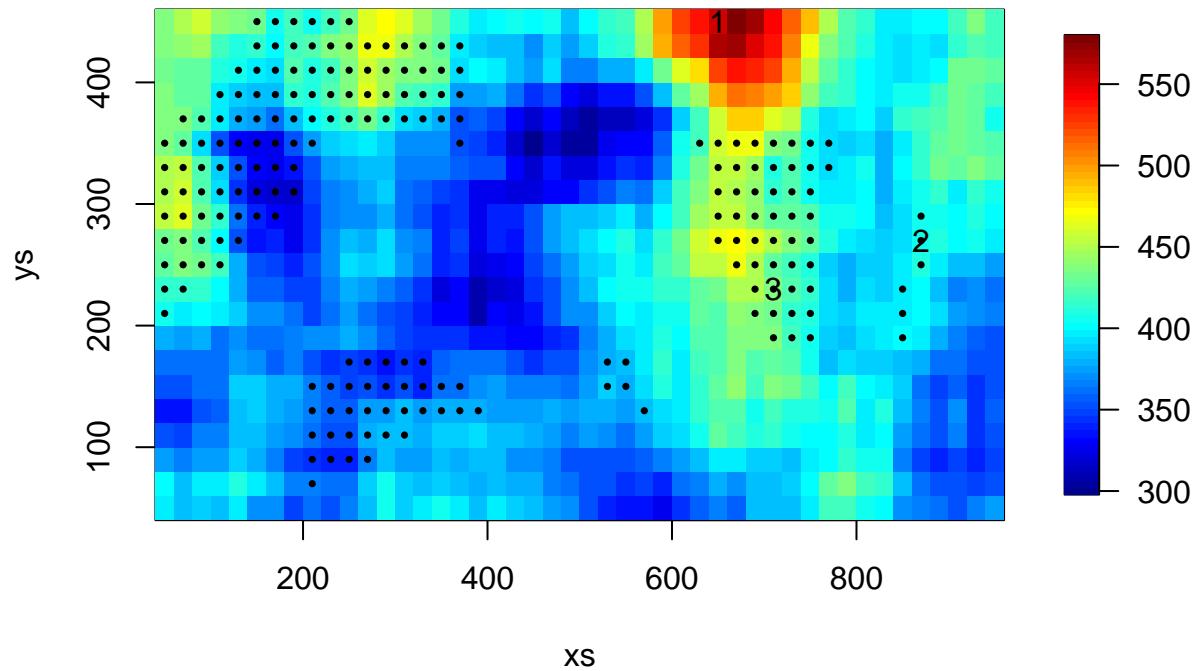


## Mean Basal Area Density in 1-ha subplots

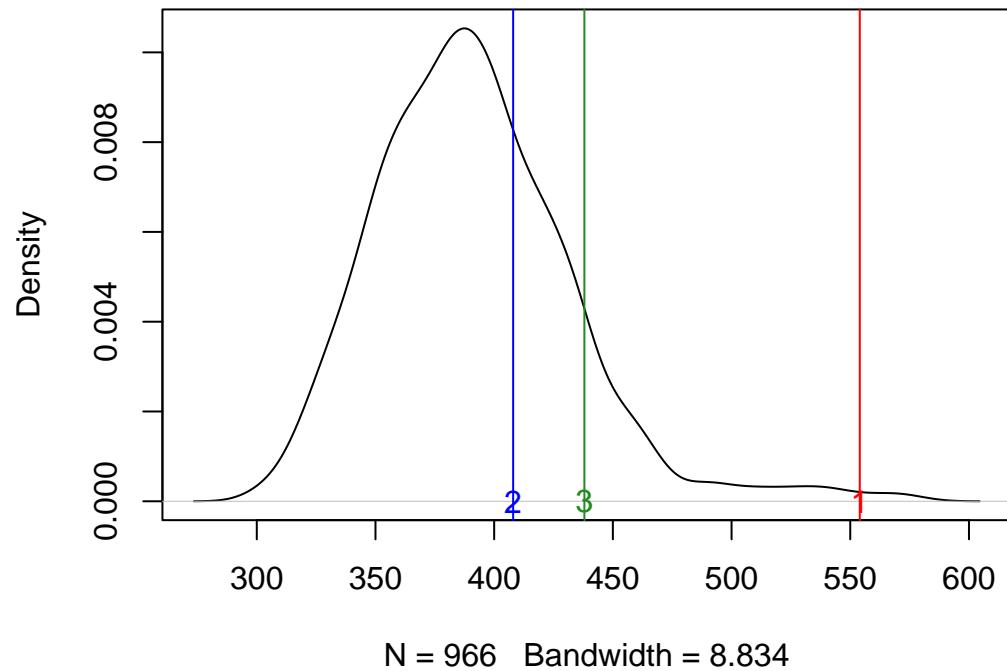


```
wheretlsivar(df1ha,"mnn10","Mean Density Trees >= 10 cm DBH in 1-ha subplots")
```

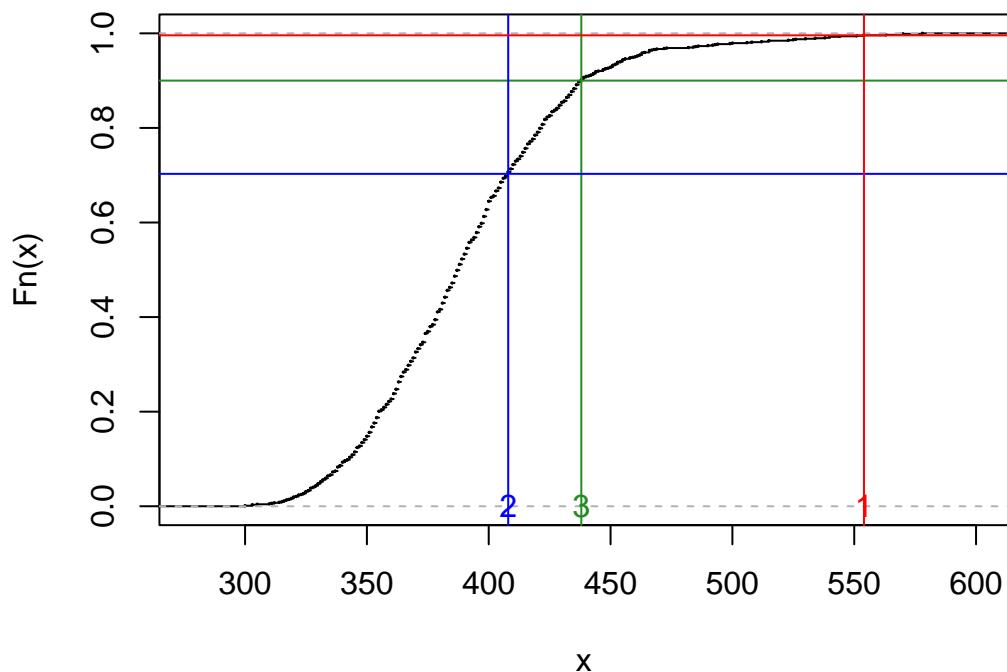
### Mean Density Trees >= 10 cm DBH in 1-ha subplots



## Mean Density Trees >= 10 cm DBH in 1-ha subplots



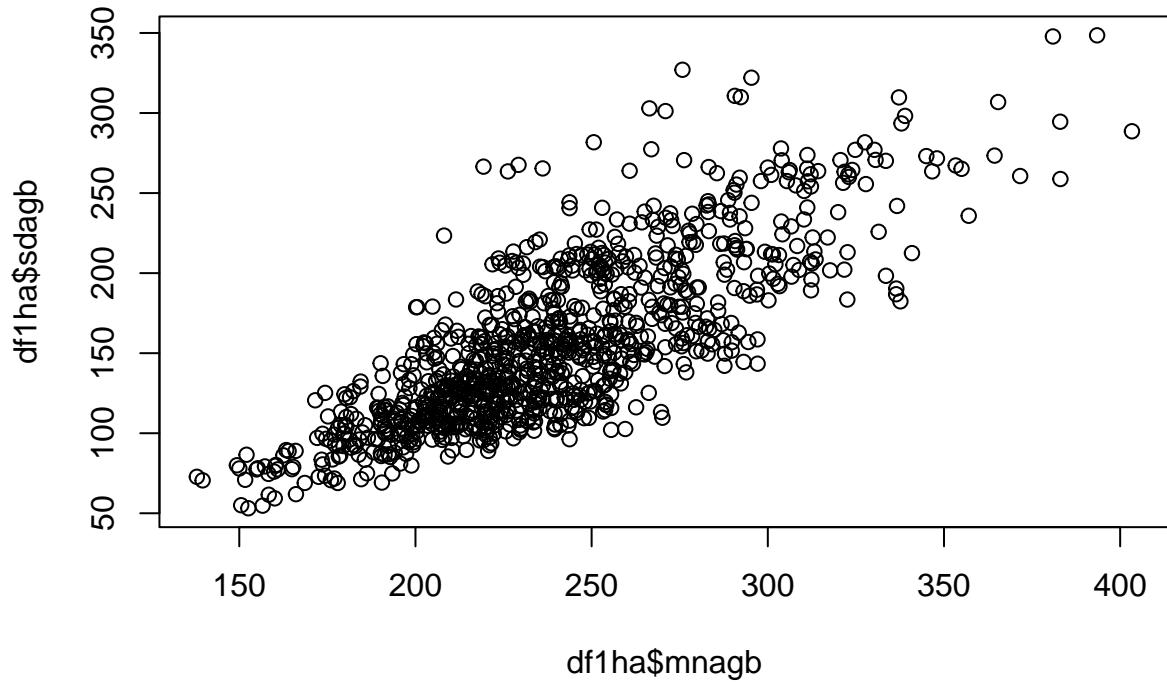
## Mean Density Trees >= 10 cm DBH in 1-ha subplots



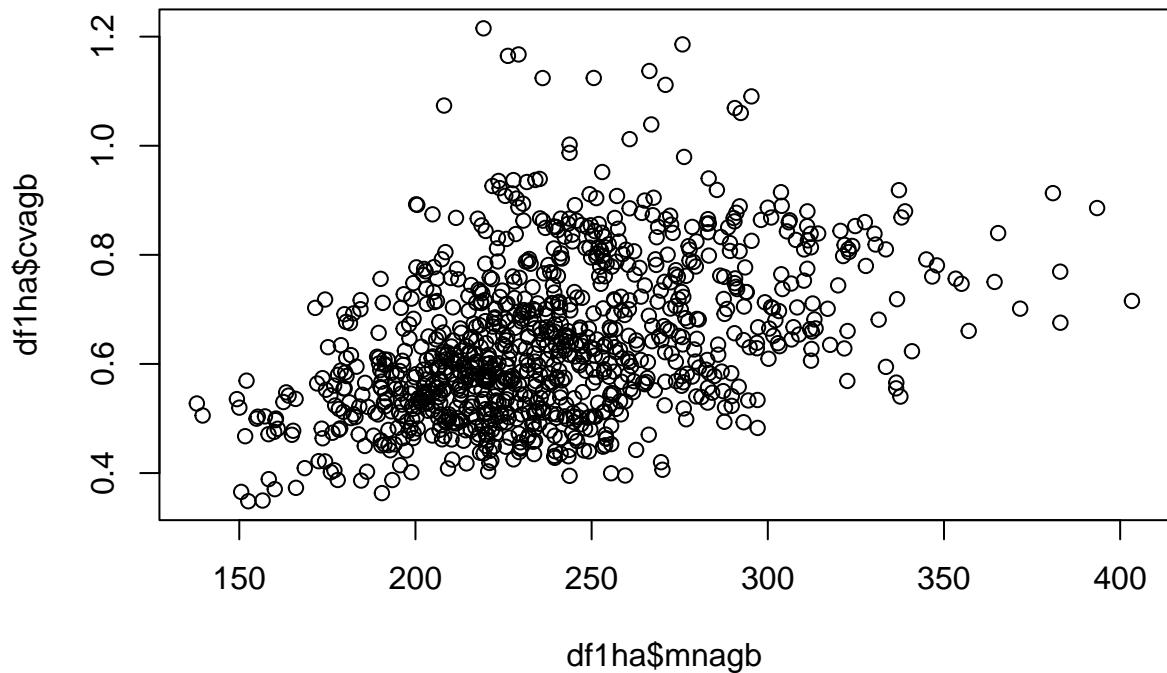
Evaluate heterogeneity within subplots.

We want the chosen subplots to show low variation within subplots. They don't need to be the lowest possible, but they shouldn't be in the highest range of within-plot variation.

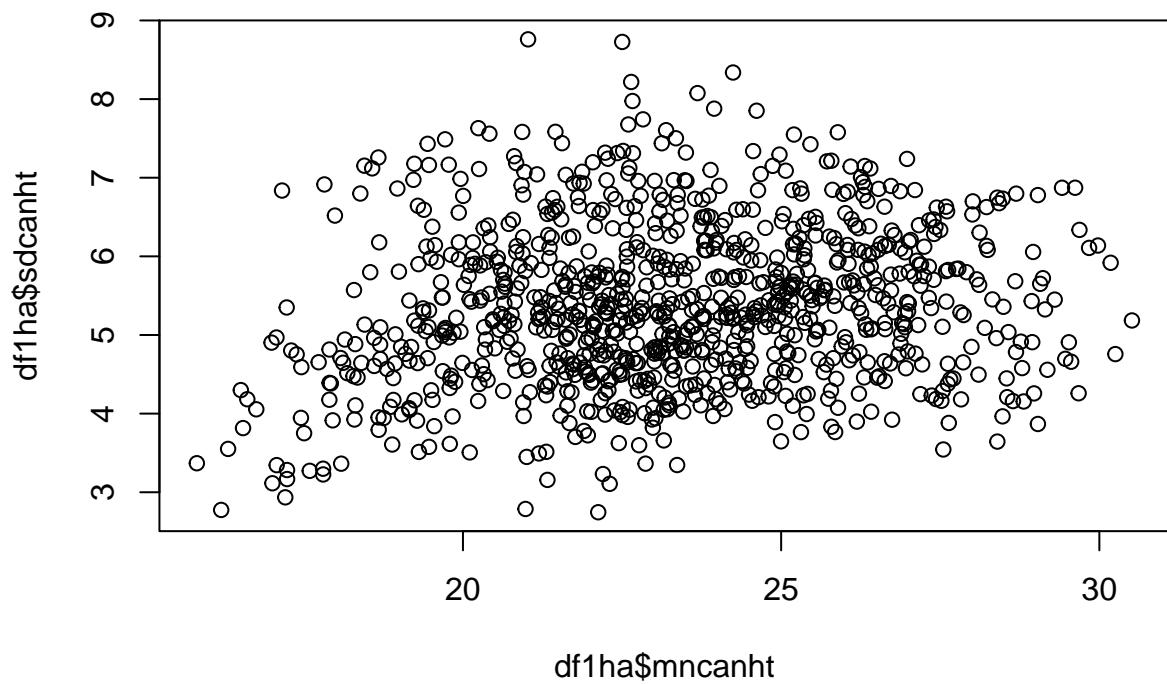
```
# first check how sds and cvs vary with means
plot(df1ha$mnagb,df1ha$sdagb)
```



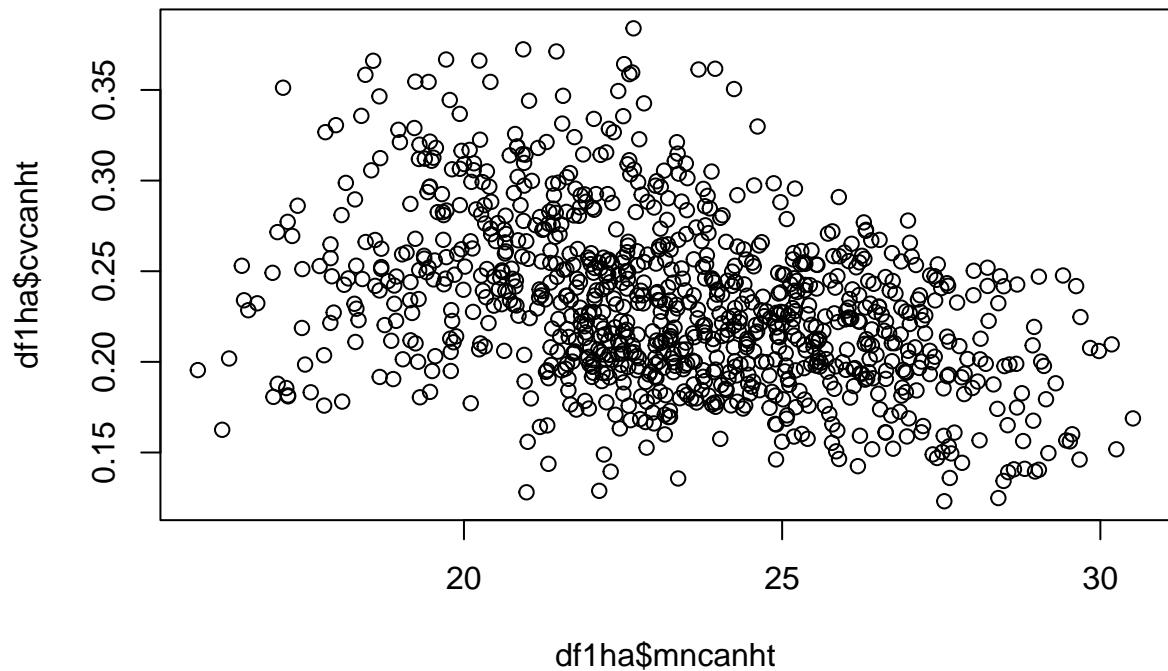
```
plot(df1ha$mnagb,df1ha$cvagb)
```



```
plot(df1ha$mncanht,df1ha$sdcanht)
```

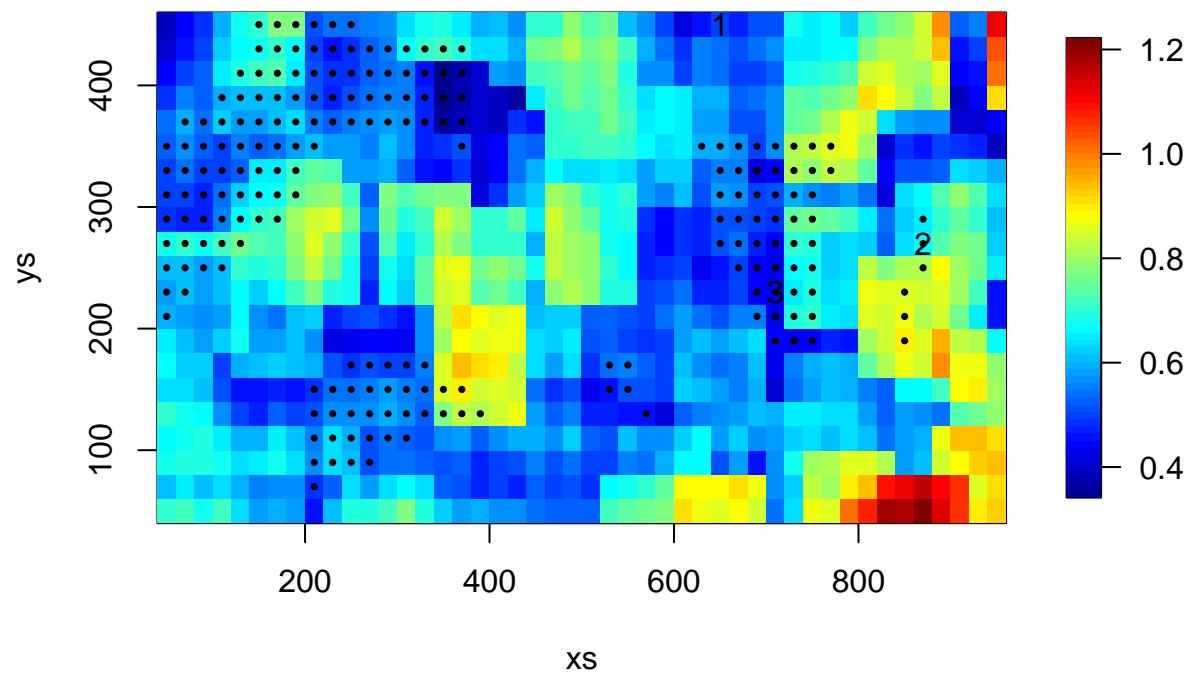


```
plot(df1ha$mncanht,df1ha$cvcanht)
```

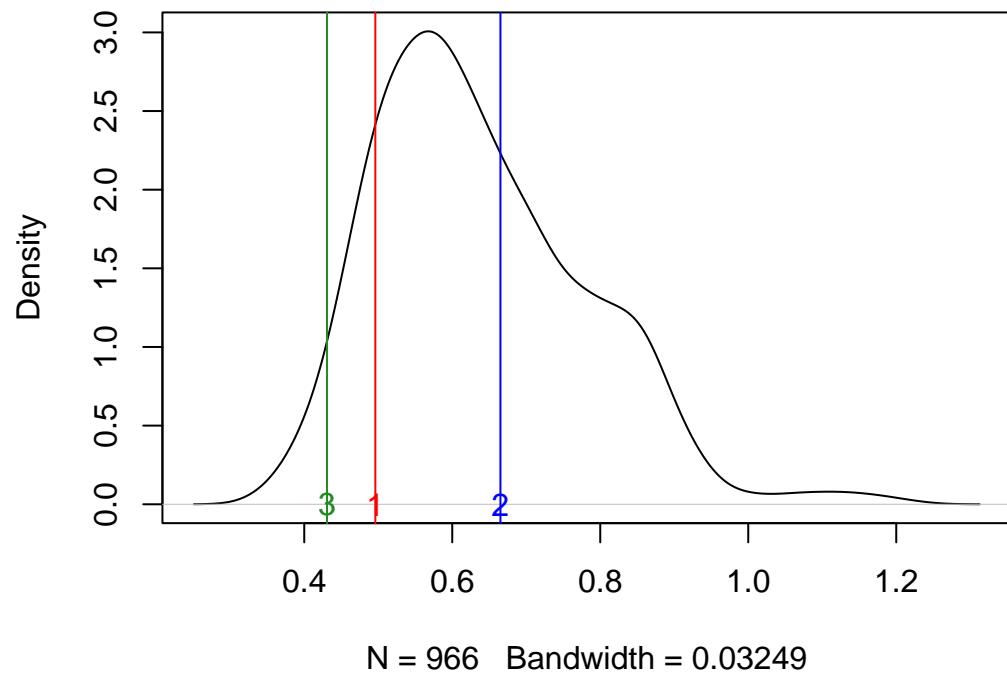


```
# based on these patterns, evaluate CVs not SDs
wheretlsivar(df1ha,"cvagb","CV (over 20x20s) of AGB Density (Mg/ha) in 1-ha subplots")
```

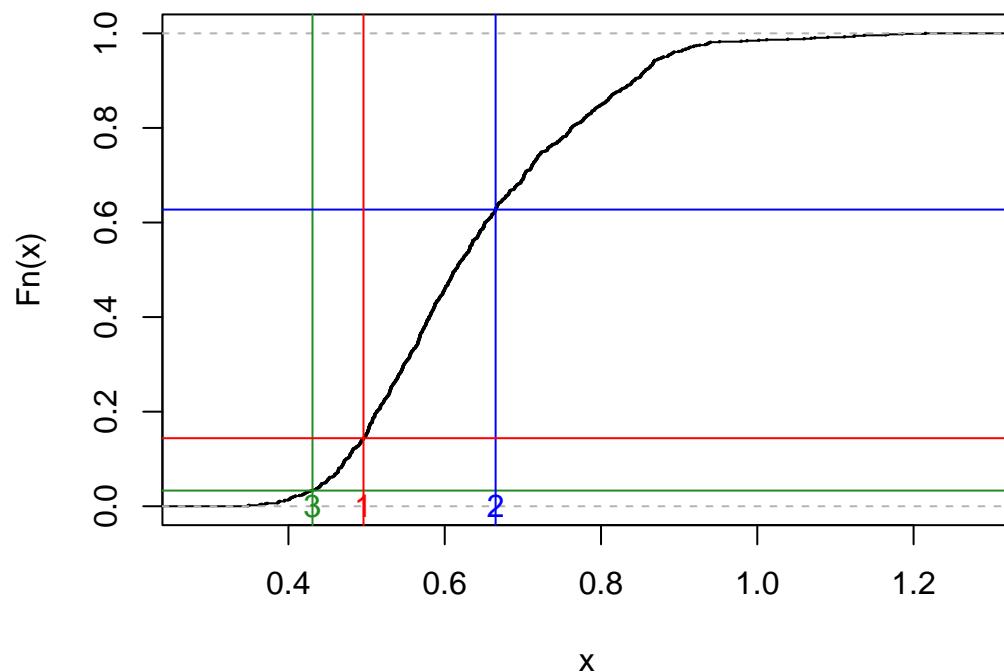
## CV (over 20x20s) of AGB Density (Mg/ha) in 1-ha subplots



## CV (over 20x20s) of AGB Density (Mg/ha) in 1-ha subplots

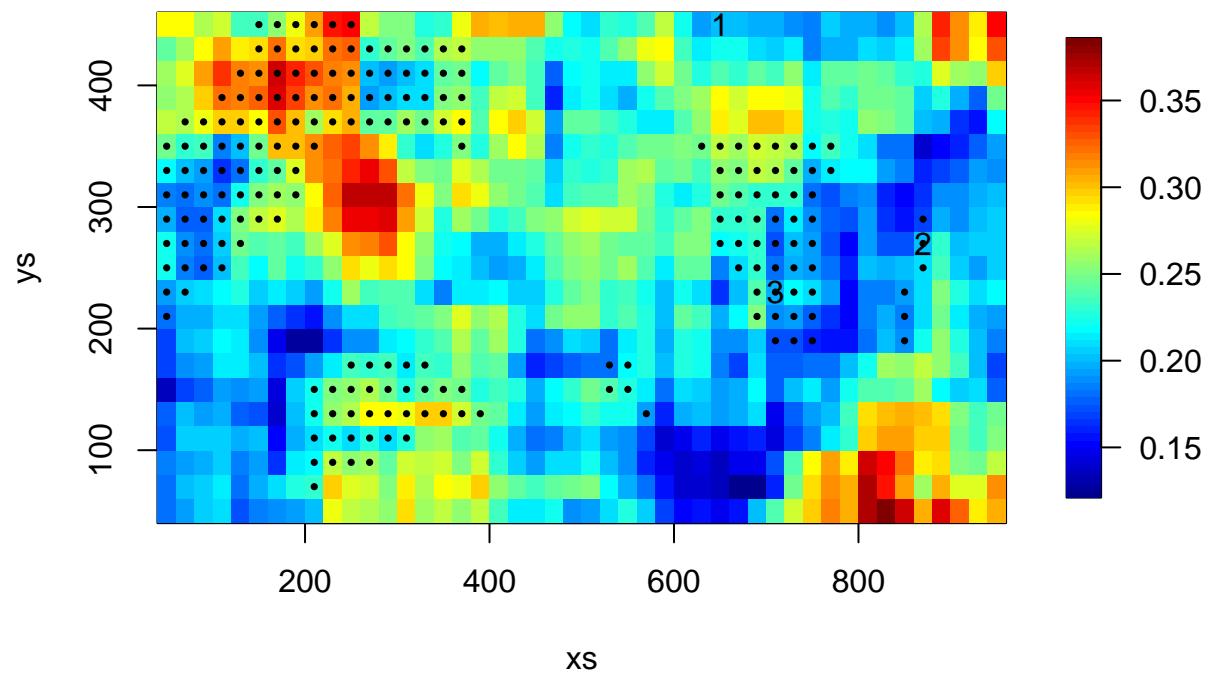


## CV (over 20x20s) of AGB Density (Mg/ha) in 1-ha subplots

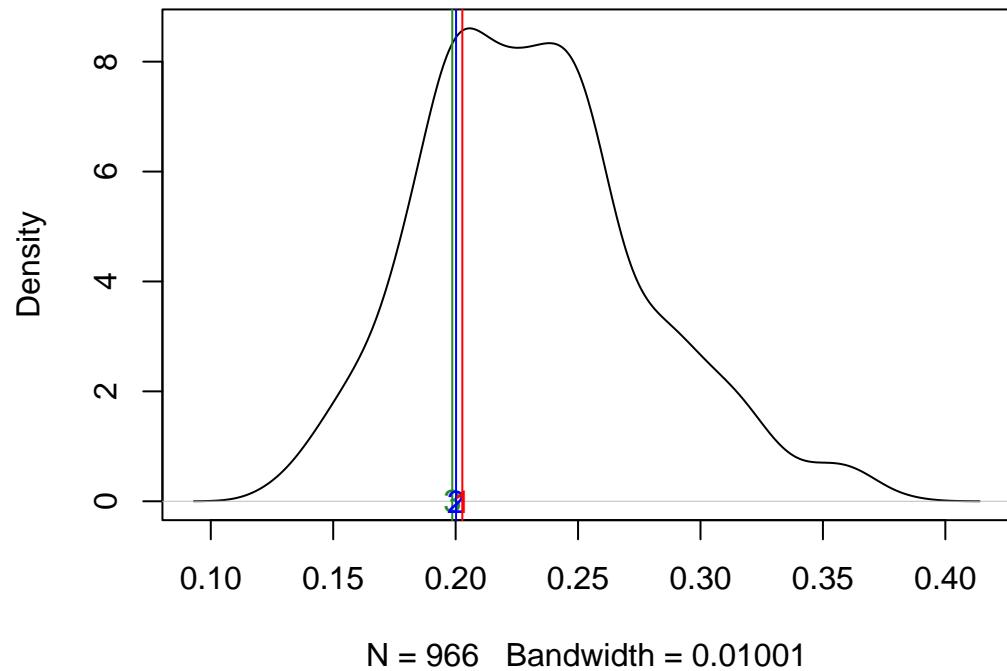


```
wheretlsivar(df1ha, "cvcanht", "CV (over 20x20s) of Mean Canopy Height in 1-ha subplots")
```

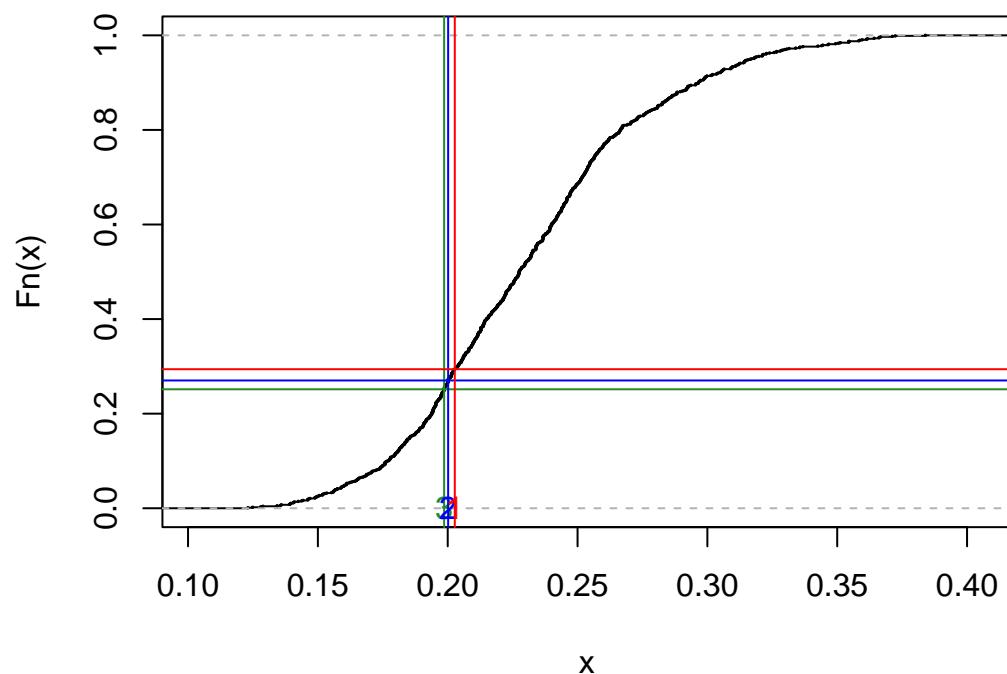
## CV (over 20x20s) of Mean Canopy Height in 1-ha subplots



## CV (over 20x20s) of Mean Canopy Height in 1-ha subplots

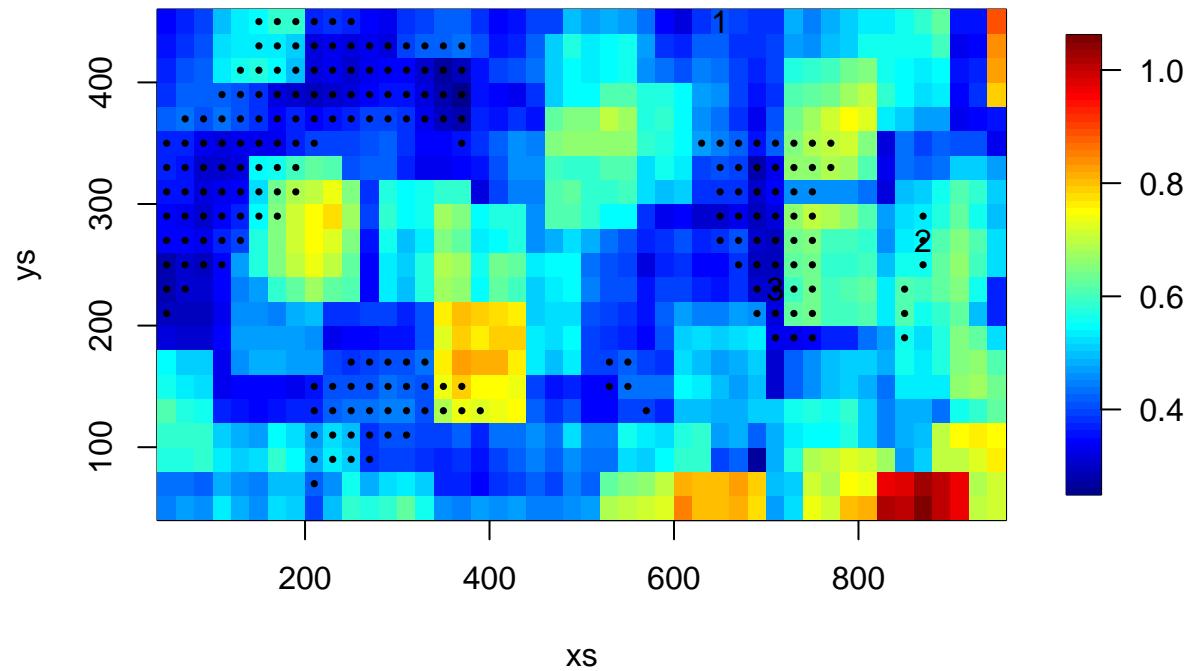


## CV (over 20x20s) of Mean Canopy Height in 1-ha subplots

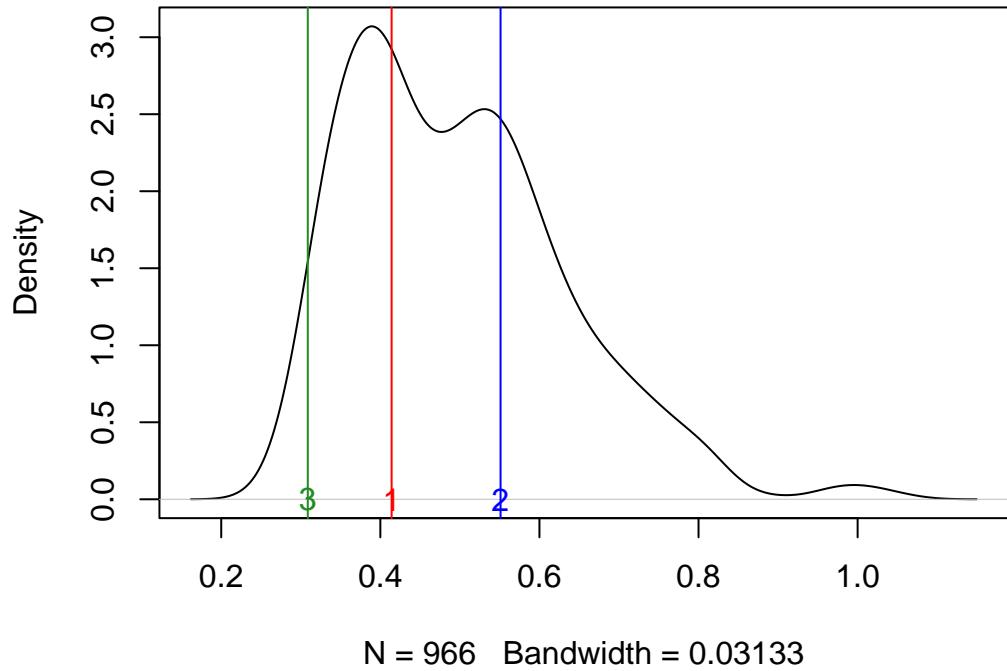


```
wheretlsivar(df1ha,"cvba","CV (over 20x20s) of Basal Area Density in 1-ha subplots")
```

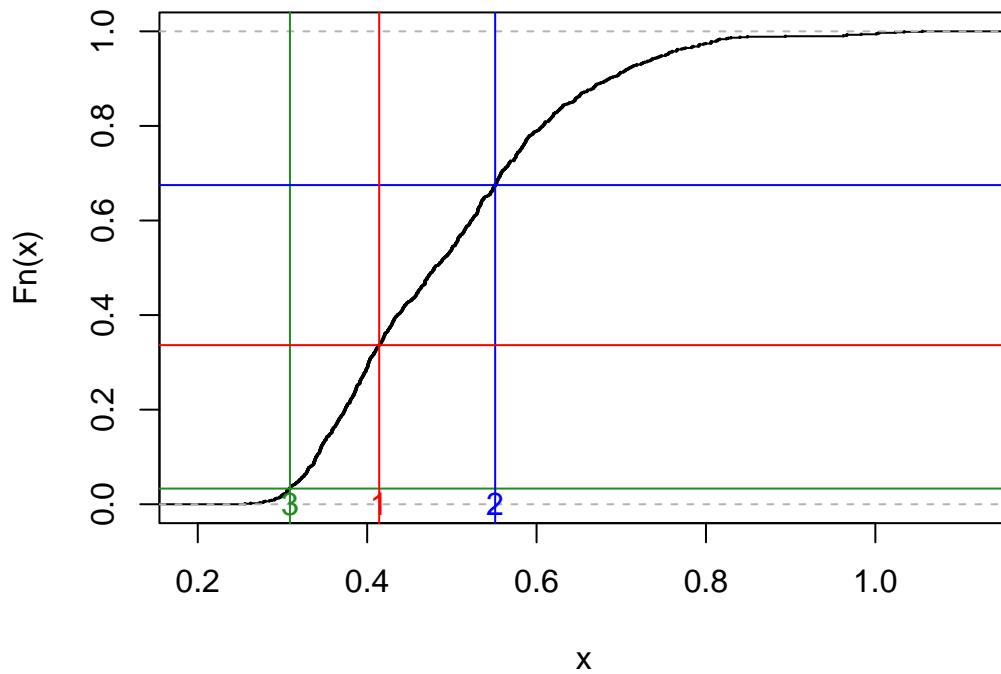
## CV (over 20x20s) of Basal Area Density in 1-ha subplots



## CV (over 20x20s) of Basal Area Density in 1-ha subplots

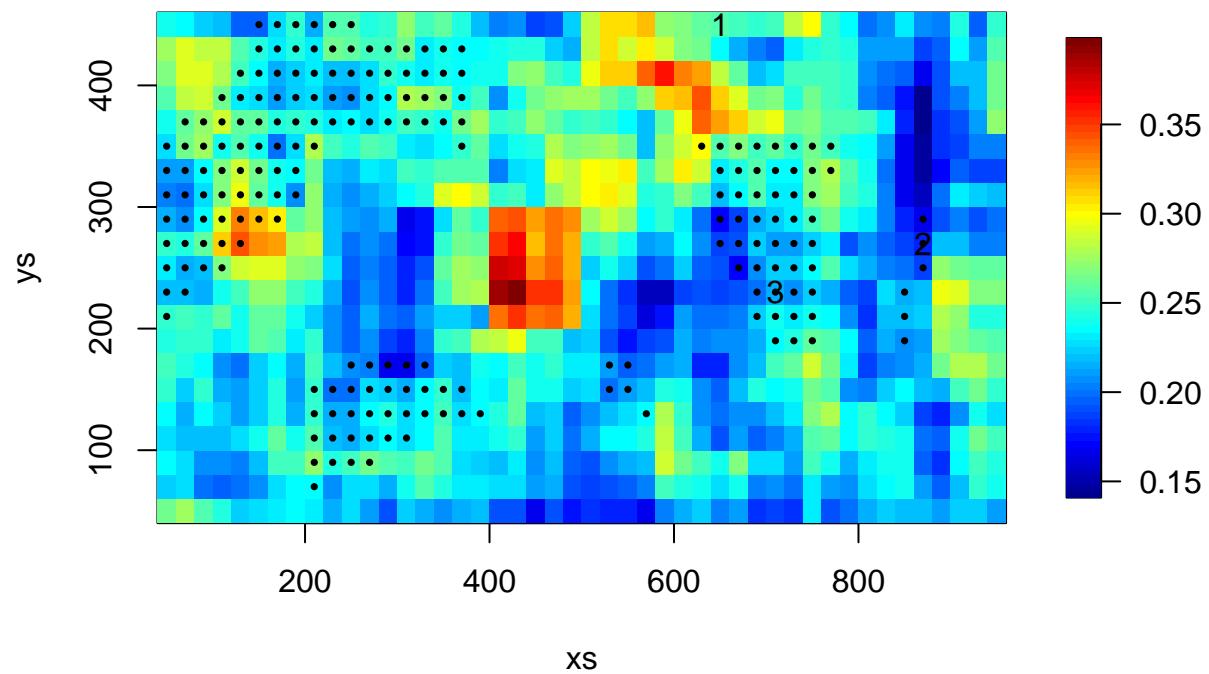


## CV (over 20x20s) of Basal Area Density in 1-ha subplots

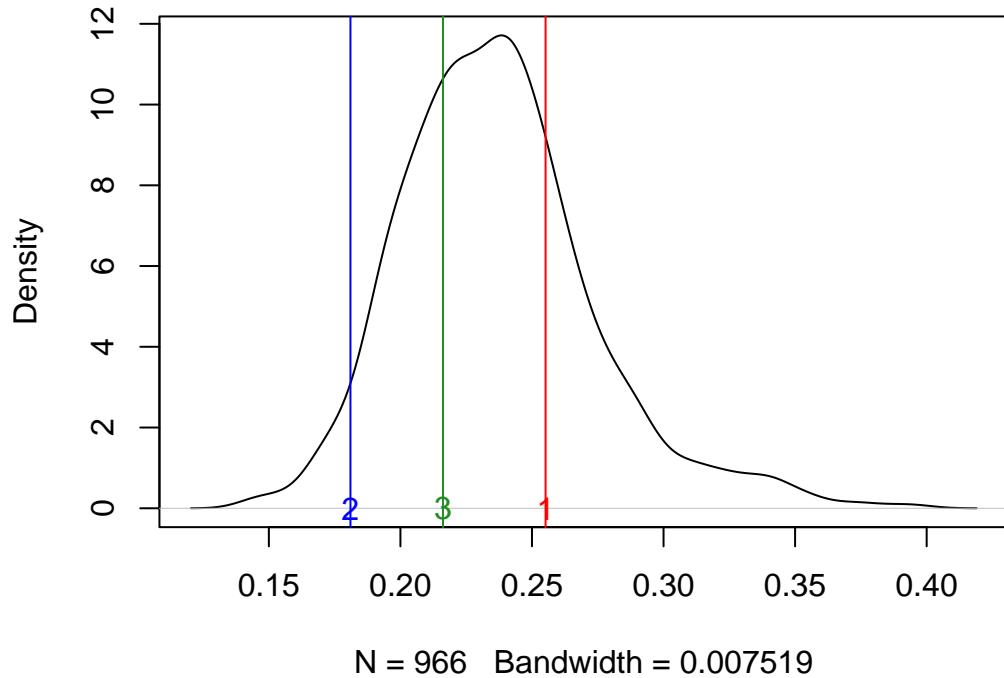


```
wheretlsivar(df1ha,"cvn10","CV (over 20x20s) of N Trees>= 10 cm in 1-ha subplots")
```

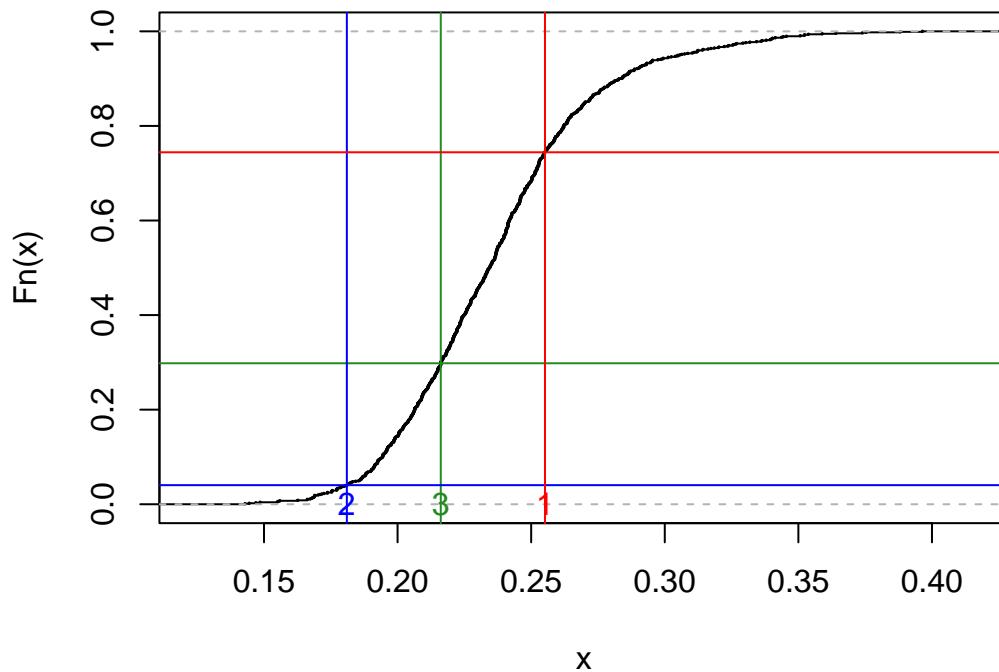
### CV (over 20x20s) of N Trees $\geq$ 10 cm in 1-ha subplots



### CV (over 20x20s) of N Trees $\geq$ 10 cm in 1-ha subplots



## CV (over 20x20s) of N Trees>= 10 cm in 1-ha subplots



## Variation across 1-ha subplots in relation to habitat type

Evaluate to what degree habitat type explains variation across 1-ha subplots, and see where proposed TLS plots fall within relevant datasets.

```
focvarvshabiha <- function(df1ha,zvarname,zvartitle) {
  focvar <- df1ha[,names(df1ha)==zvarname]
  inc <- !is.na(df1ha$newtls)
  newtls1ha <- data.frame(nplot=df1ha$newtls[inc],focvar=focvar[inc],
                           namehab=df1ha$namemainhabitat[inc],mainhab=df1ha$mainhabitat[inc])
  newtls1ha <- newtls1ha[order(newtls1ha$mainhab),]
  nnewtls <- nrow(newtls1ha)

  inc100 <- df1ha$habdominance==1
  inc80 <- df1ha$habdominance >=0.8
  inc60 <- df1ha$habdominance >=0.6

  vioplot(focvar[inc100]~df1ha$namemainhabitat[inc100],areaEqual=F,
          main=paste(zvartitle,"vs Habitat, for 1-ha 100% that type"),
          ylab=zvartitle,xlab="Habitat Type")
  xloc=seq(0.7,3.3,length=nnewtls)
  for (i in 1:nrow(newtls1ha)) {
    abline(h=newtls1ha$focvar[i],col=cols[newtls1ha$nplot[i]])
    text(xloc[i],newtls1ha$focvar[i],newtls1ha$namehab[i],col=cols[newtls1ha$nplot[i]])
  }

  vioplot(focvar[inc80]~df1ha$namemainhabitat[inc80],areaEqual=F,
```

```

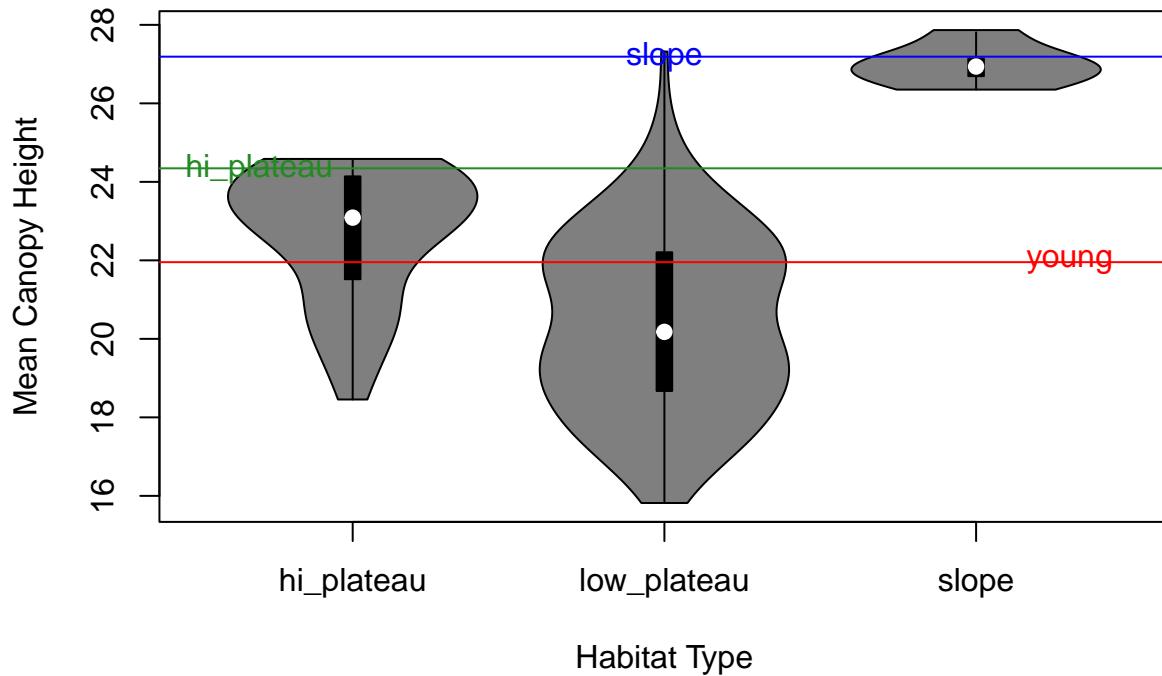
    main=paste(zvartitle,"vs Habitat, for 1-ha >= 80% that type"),
    ylab=zvartitle,xlab="Habitat Type")
for (i in 1:nrow(newtls1ha)) {
  abline(h=newtls1ha$focvar[i],col=cols[newtls1ha$nplot[i]])
  text(xloc[i],newtls1ha$focvar[i],newtls1ha$namehab[i],col=cols[newtls1ha$nplot[i]])
}

vioplot(focvar[inc60]~df1ha$namemainhabitat[inc60],areaEqual=F,
        main=paste(zvartitle,"vs Habitat, for 1-ha >= 60% that type"),
        ylab=zvartitle,xlab="Habitat Type")
for (i in 1:nrow(newtls1ha)) {
  abline(h=newtls1ha$focvar[i],col=cols[newtls1ha$nplot[i]])
  text(xloc[i],newtls1ha$focvar[i],newtls1ha$namehab[i],col=cols[newtls1ha$nplot[i]])
}

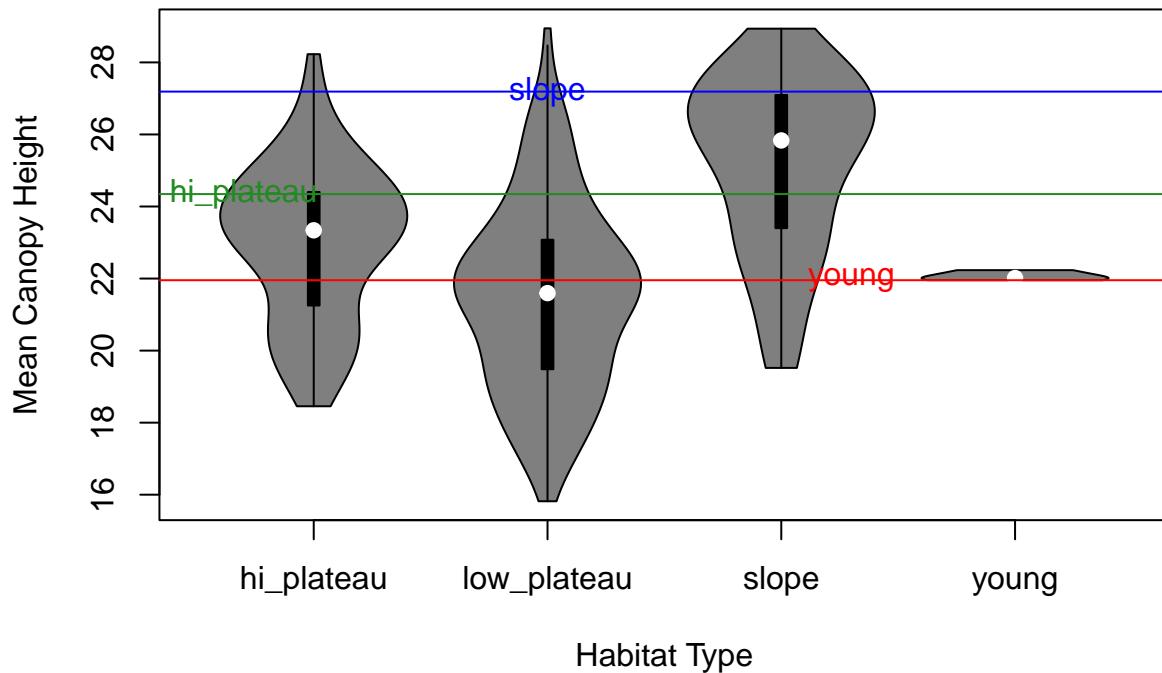
}
focvarvshab1ha(df1ha,"mncanht","Mean Canopy Height")

```

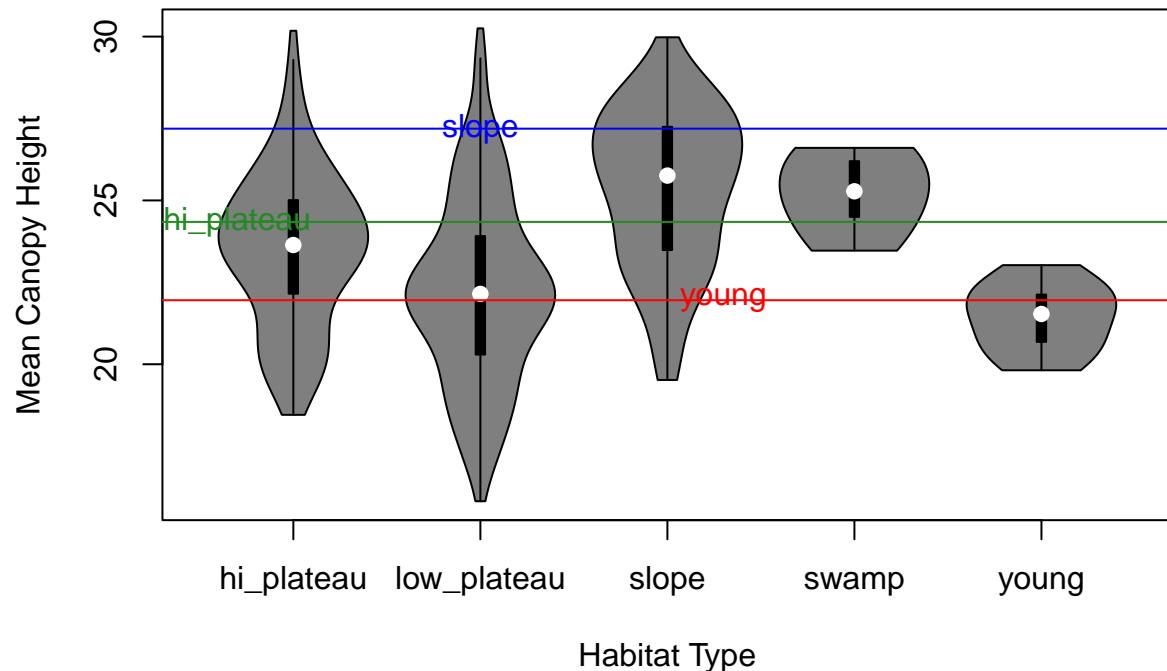
## Mean Canopy Height vs Habitat, for 1-ha 100% that type



### Mean Canopy Height vs Habitat, for 1-ha $\geq$ 80% that type

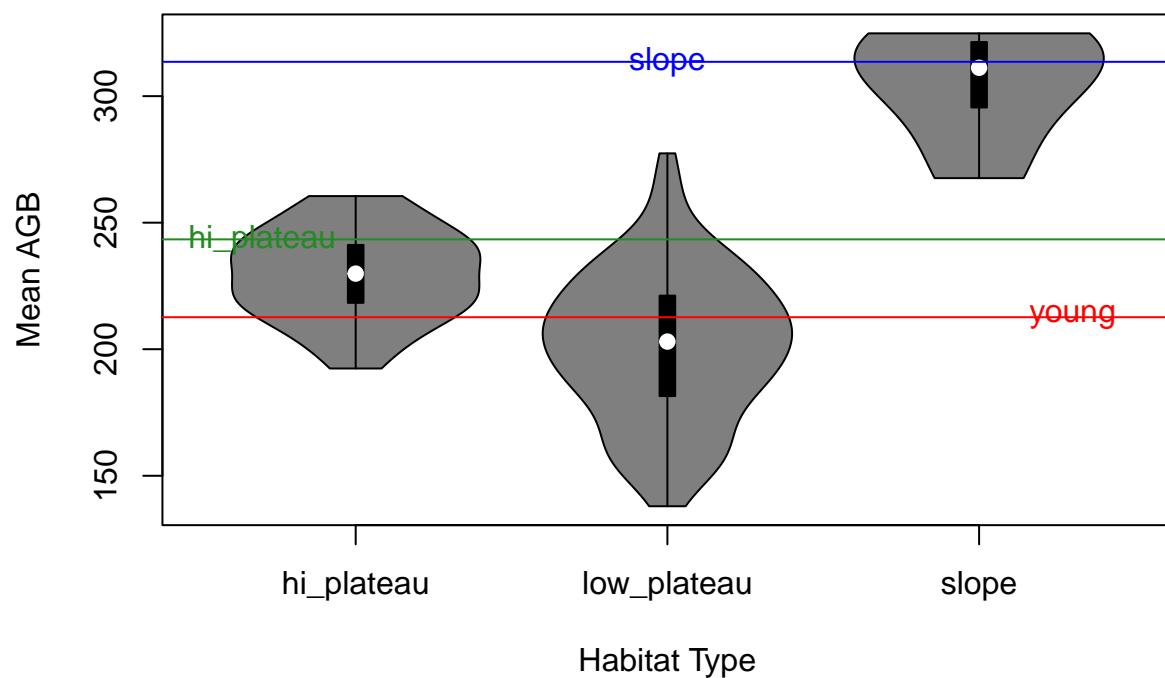


## Mean Canopy Height vs Habitat, for 1-ha $\geq$ 60% that type

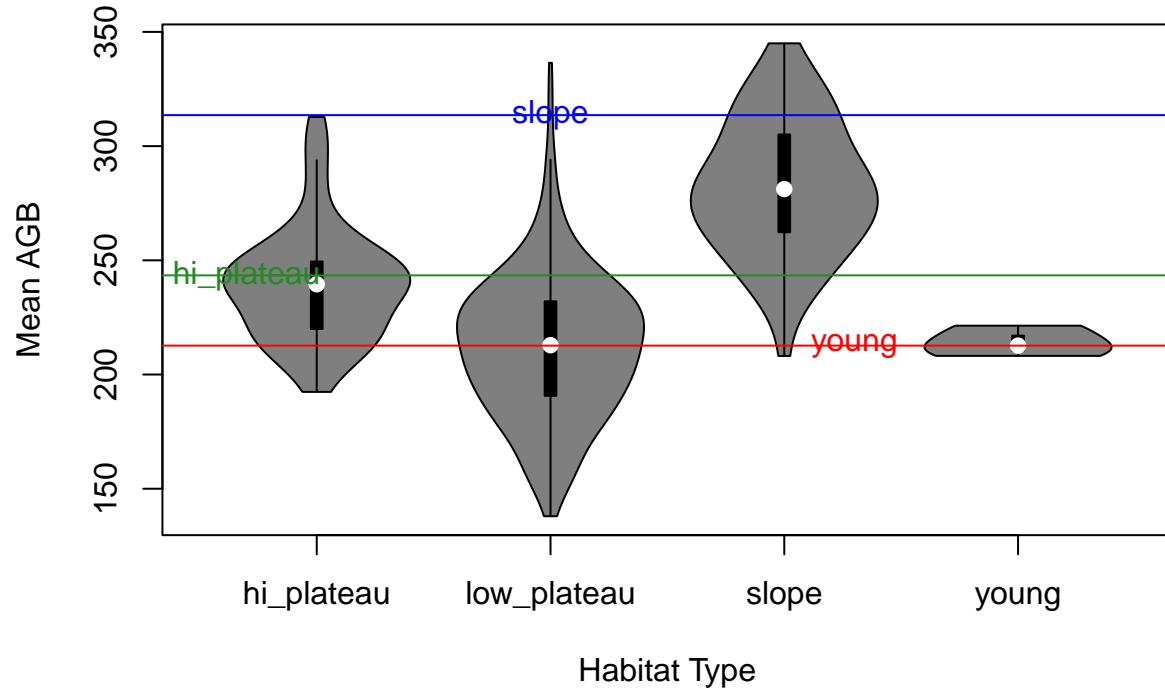


```
focvarvshabiha(df1ha, "mnagb", "Mean AGB")
```

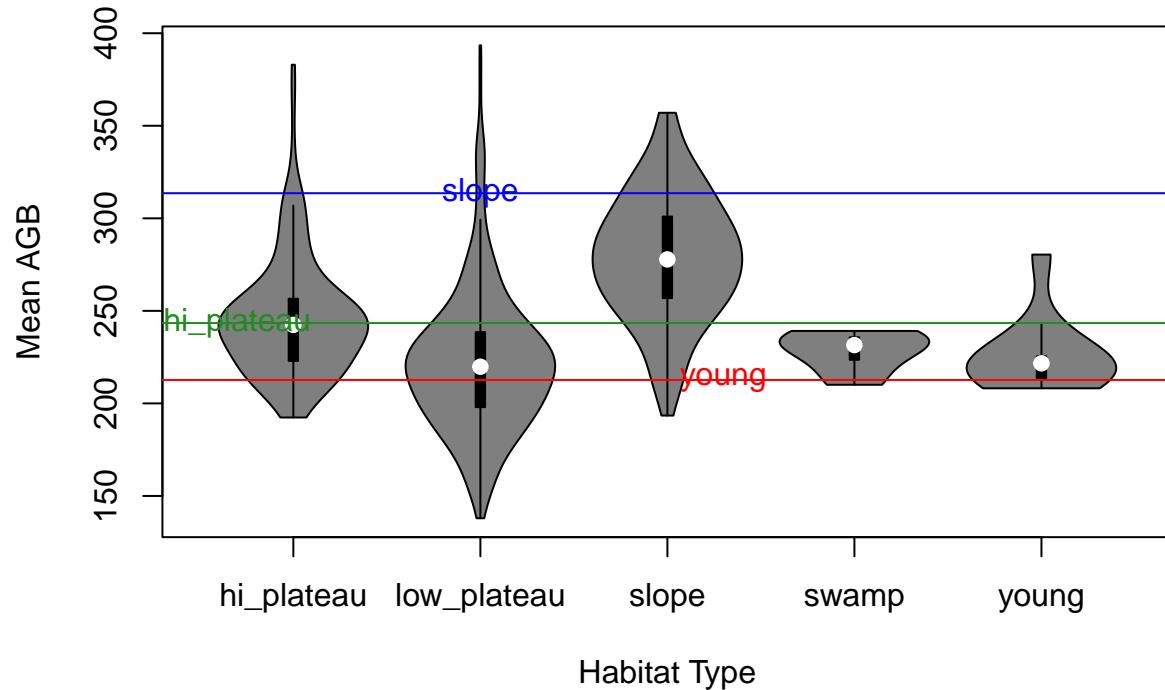
### Mean AGB vs Habitat, for 1-ha 100% that type



### Mean AGB vs Habitat, for 1-ha $\geq 80\%$ that type

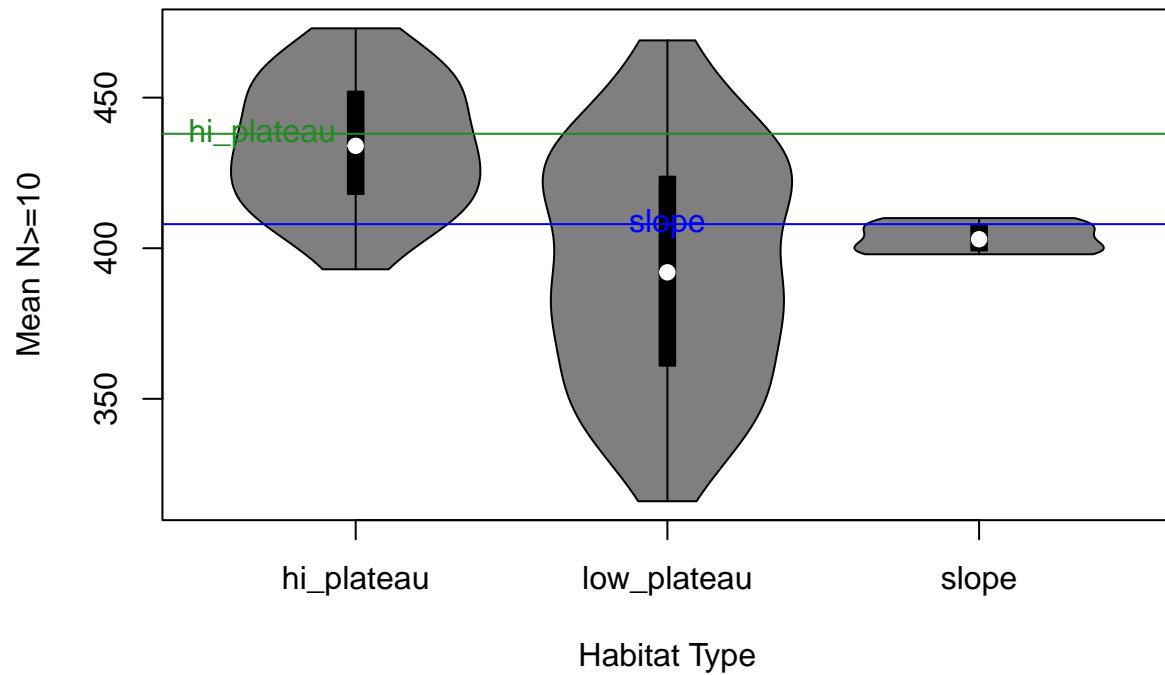


### Mean AGB vs Habitat, for 1-ha $\geq$ 60% that type

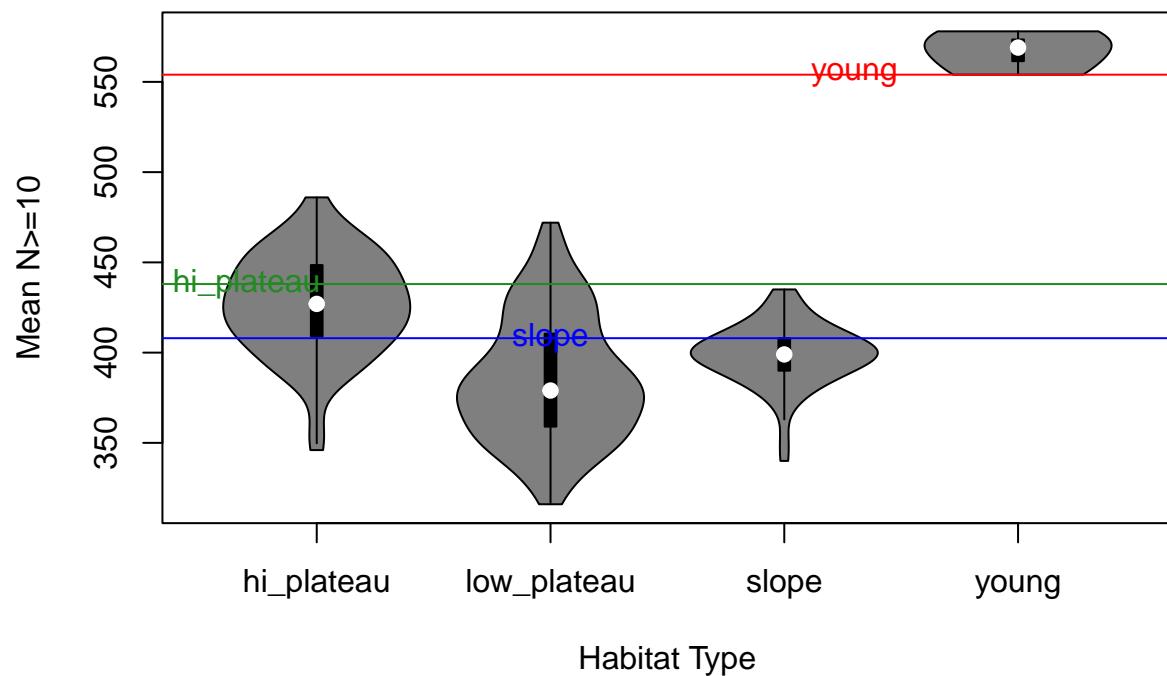


```
focvarvshabiha(df1ha, "mnn10", "Mean N>=10")
```

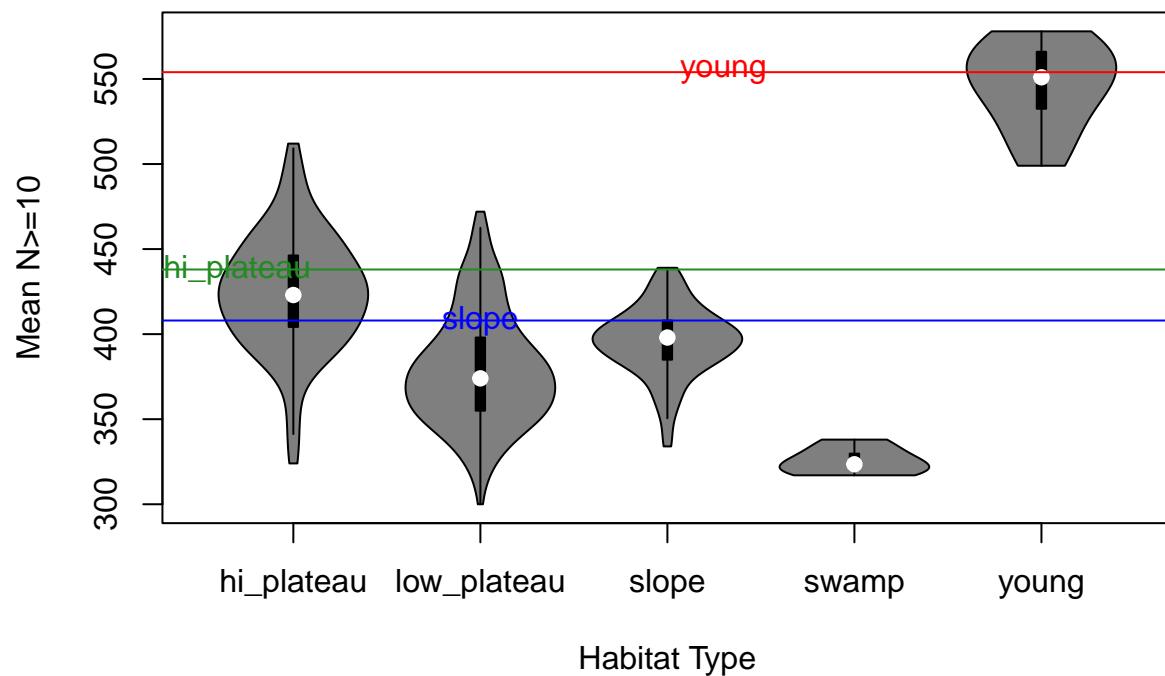
### Mean N $\geq$ 10 vs Habitat, for 1-ha 100% that type



### Mean N $\geq$ 10 vs Habitat, for 1-ha $\geq$ 80% that type

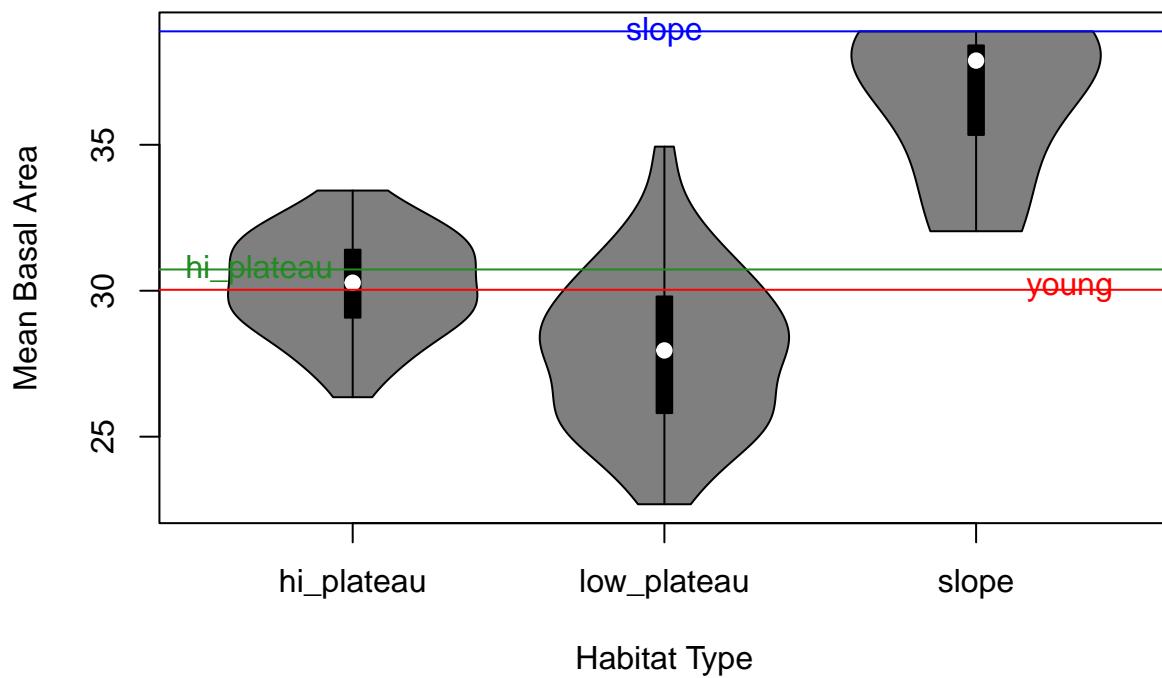


### Mean N $\geq$ 10 vs Habitat, for 1-ha $\geq$ 60% that type

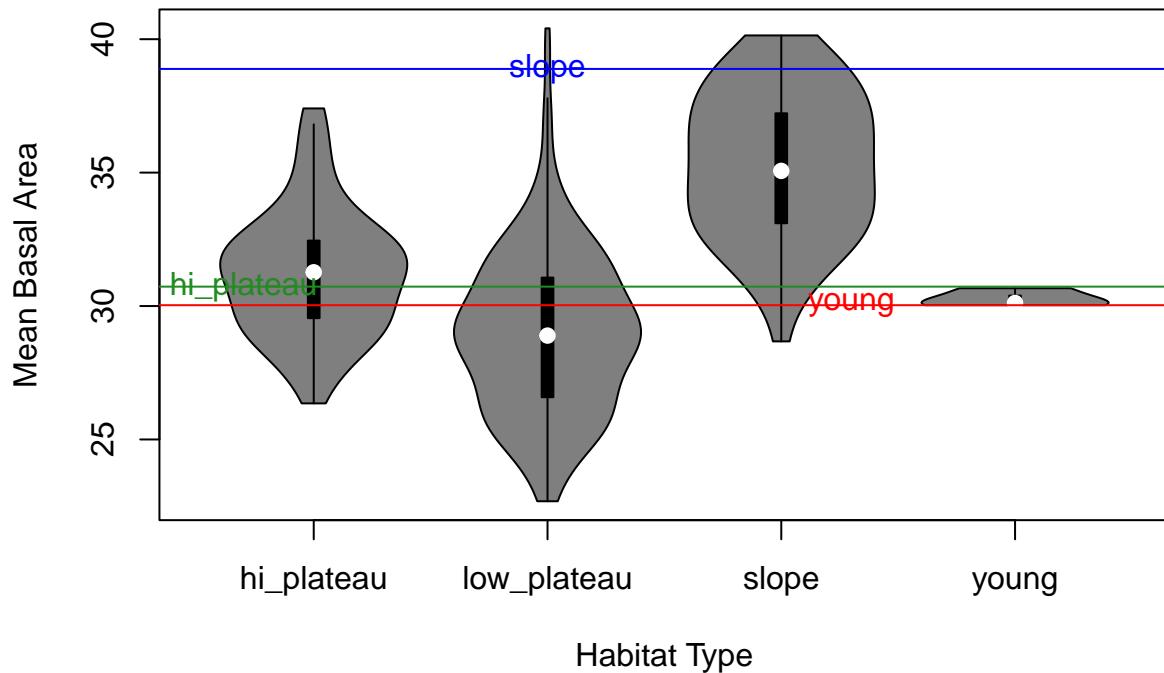


```
focvarvshabiha(df1ha, "mmba", "Mean Basal Area")
```

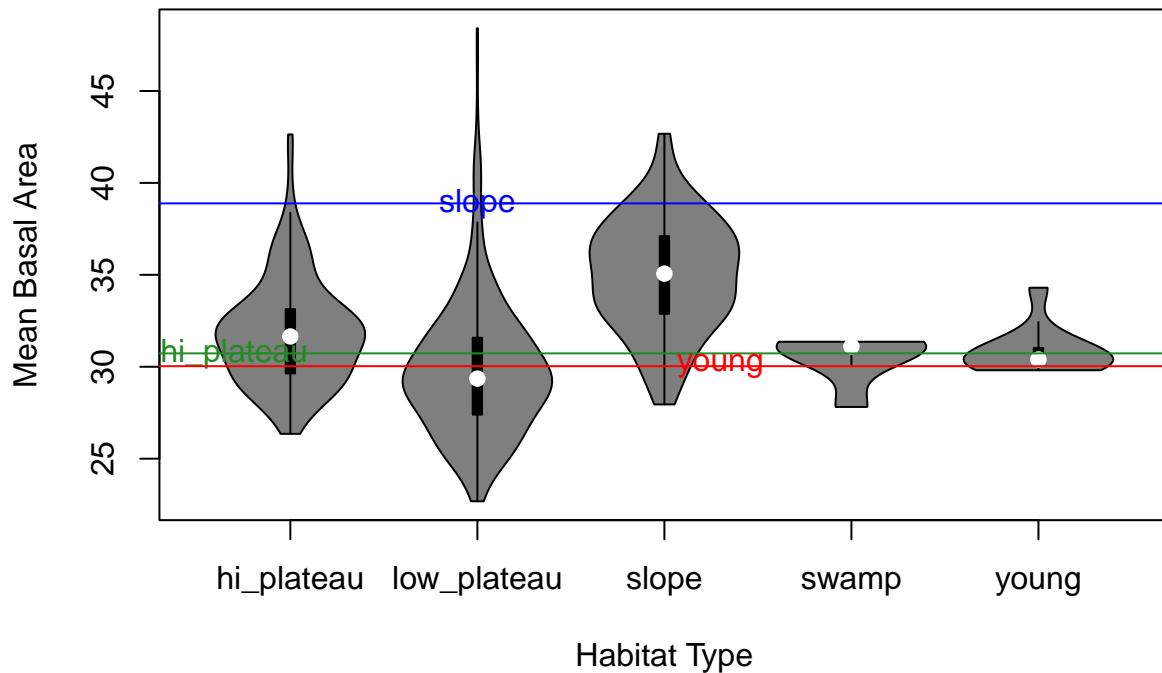
## Mean Basal Area vs Habitat, for 1-ha 100% that type



### Mean Basal Area vs Habitat, for 1-ha $\geq$ 80% that type



## Mean Basal Area vs Habitat, for 1-ha >= 60% that type



### Positions of plots in scatterplots of pairs of variables

Here symbols indicate habitat type

```
foc2varvshab1ha <- function(df1ha,z1varname,z2varname,z1vartitle,z2vartitle,mindominance=1) {
  focvar1 <- df1ha[,names(df1ha)==z1varname]
  focvar2 <- df1ha[,names(df1ha)==z2varname]
  quantfoc1 <- rank(focvar1)/nrow(df1ha)
  quantfoc2 <- rank(focvar2)/nrow(df1ha)
  inc <- !is.na(df1ha$newtls)
  newtls1ha <- data.frame(nplot=df1ha$newtls[inc],focvar1=focvar1[inc],quantfoc1=quantfoc1[inc],
                           focvar2=focvar2[inc],quantfoc2=quantfoc2[inc],
                           namehab=df1ha$namemainhabitat[inc],mainhab=df1ha$mainhabitat[inc])
  newtls1ha <- newtls1ha[order(newtls1ha$nplot),]
  nnewtls <- nrow(newtls1ha)

  inc <- df1ha$habdominance>=mindominance

  par(mar=c(5,4,4,8),xpd=TRUE)
  plot(focvar1[inc],focvar2[inc],xlab=z1vartitle,ylab=z2vartitle,
        xlim=range(focvar1),ylim=range(focvar2),
        cex=0.8,pch=df1ha$mainhabitat[inc],
        col=rainbow(7)[df1ha$mainhabitat[inc]],
        main=paste("Only Plots >=",100*mindominance,"% Single Habitat"))
  points(newtls1ha$focvar1,newtls1ha$focvar2,
```

```

    pch=newtls1ha$mainhab,lwd=2)
legend("topright",inset=c(-0.2,0),legend=newtls1ha$nplot,
#      col=cols[1:nnewtls],
      lwd=2,pch=newtls1ha$mainhab,title="1-ha plots")
legend("bottomright",inset=c(-0.3,0),legend=levels(bci_habitat$habitat),
#col="grey",
      col=rainbow(7)[1:7],
      pch=1:7, title="habitats")

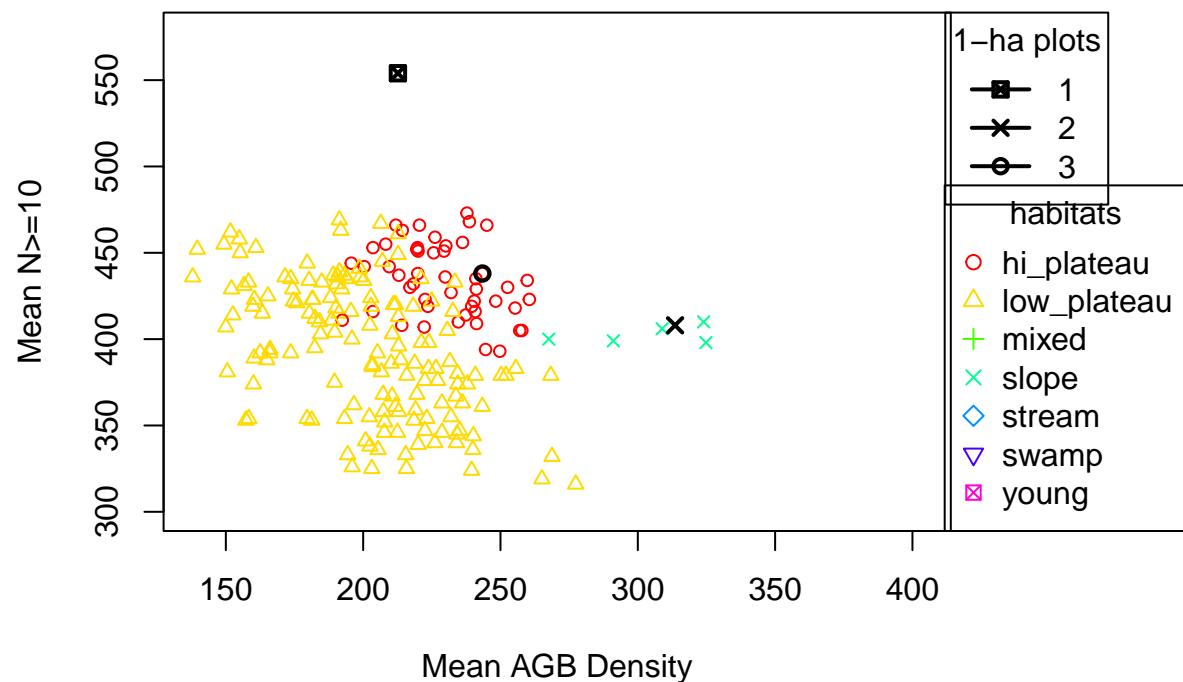
par(mar=c(5,4,4,8),xpd=TRUE)
plot(quantfoc1[inc],quantfoc2[inc],
      xlab=paste("Quantile", z1vartitle),ylab=paste("Quantile",z2vartitle),
      cex=0.8,pch=df1ha$mainhabitat[inc],
#      col="grey",
      col=rainbow(7)[df1ha$mainhabitat[inc]],
      main=paste("Only Plots >=",100*mindominance,"% Single Habitat"))
points(newtls1ha$quantfoc1,newtls1ha$quantfoc2,
      #col=cols[newtls1ha$nplot],
      pch=newtls1ha$mainhab,lwd=2)
legend("topright",inset=c(-0.2,0),legend=newtls1ha$nplot,
      pch=newtls1ha$mainhab,lwd=2,
#      col=cols[1:nnewtls],
      title="1-ha plots")
legend("bottomright",inset=c(-0.3,0),legend=levels(bci_habitat$habitat),
#col="grey",
      col=rainbow(7)[1:7],
      pch=1:7, title="habitats")

}

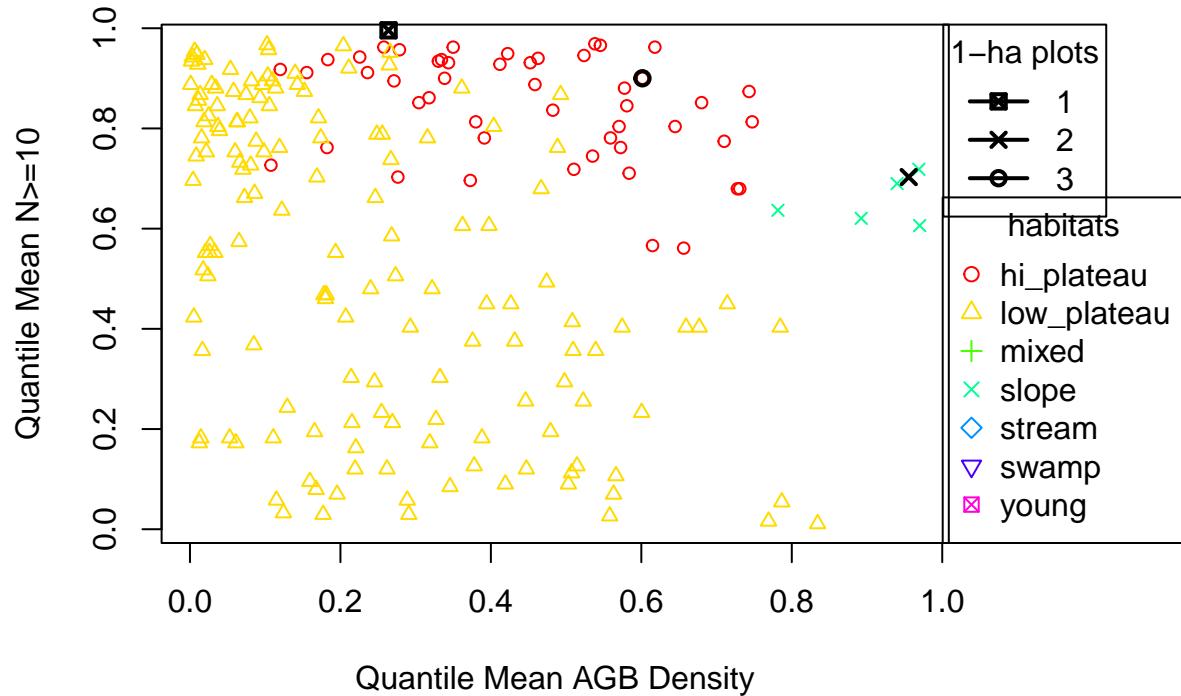
foc2varvshab1ha(df1ha,z1varname="mnagb",z2varname="mnn10",
                  z1vartitle="Mean AGB Density",z2vartitle="Mean N>=10",
                  mindominance=1)

```

### Only Plots $\geq 100\%$ Single Habitat

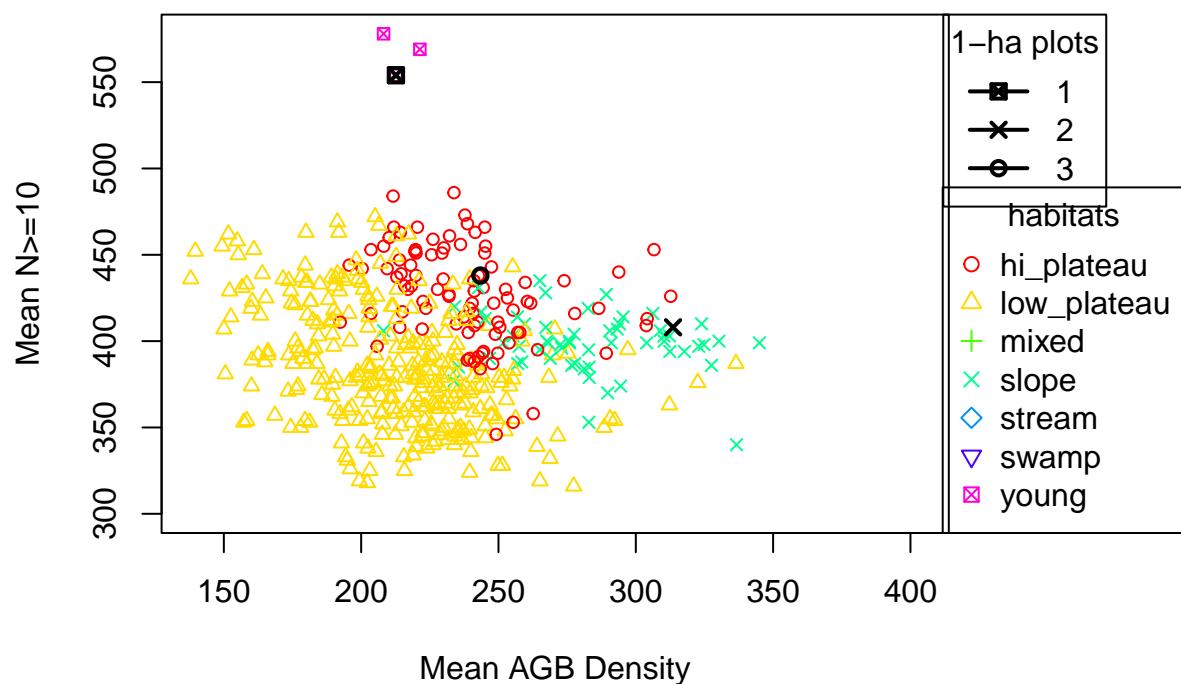


## Only Plots $\geq 100\%$ Single Habitat

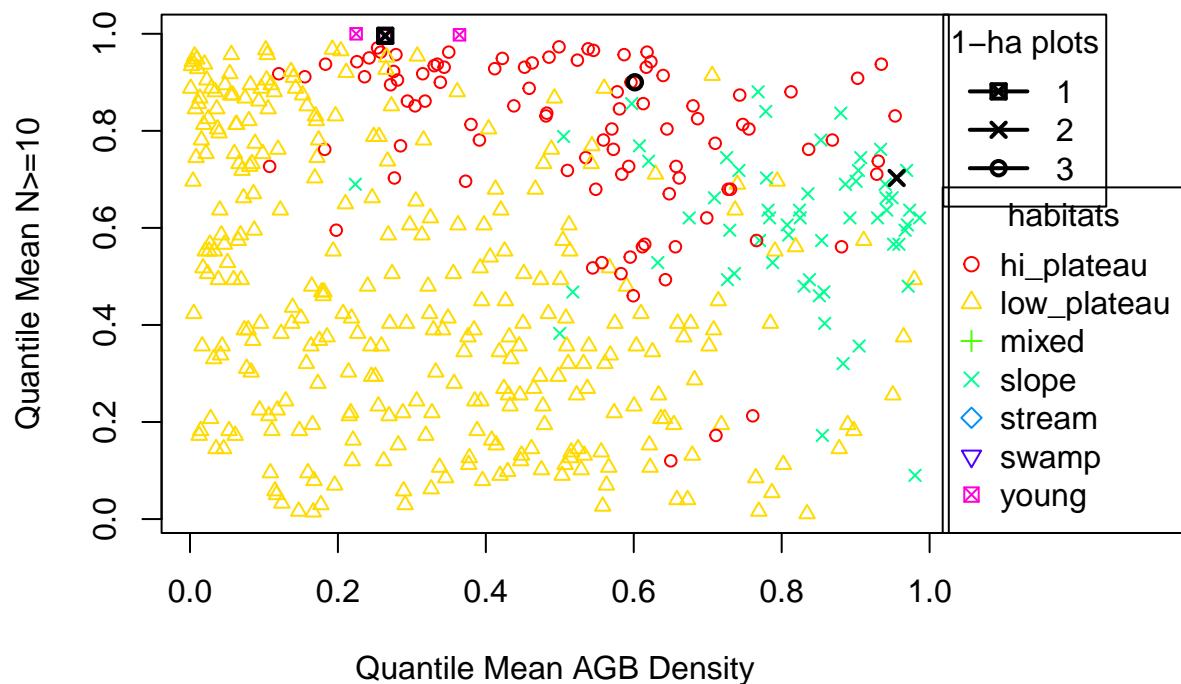


```
foc2varvshab1ha(df1ha,z1varname="mnagb",z2varname="mnn10",
                  z1vartitle="Mean AGB Density",z2vartitle="Mean N>=10",
                  mindominance=0.8)
```

### Only Plots $\geq 80\%$ Single Habitat

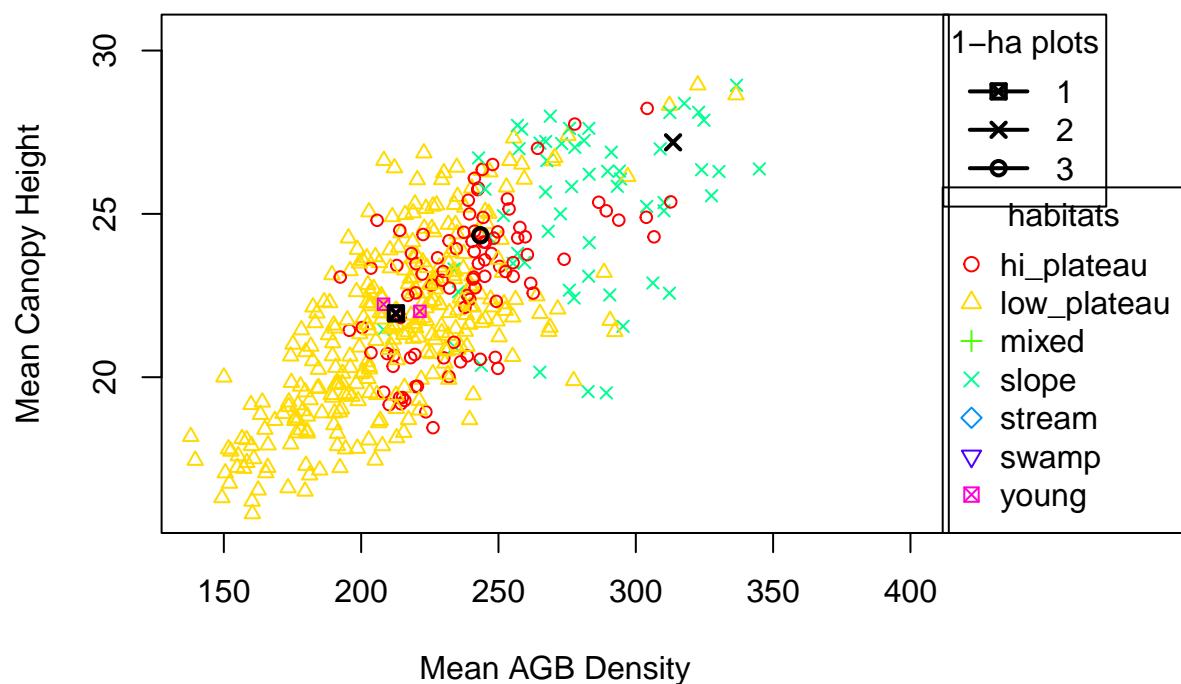


## Only Plots >= 80 % Single Habitat

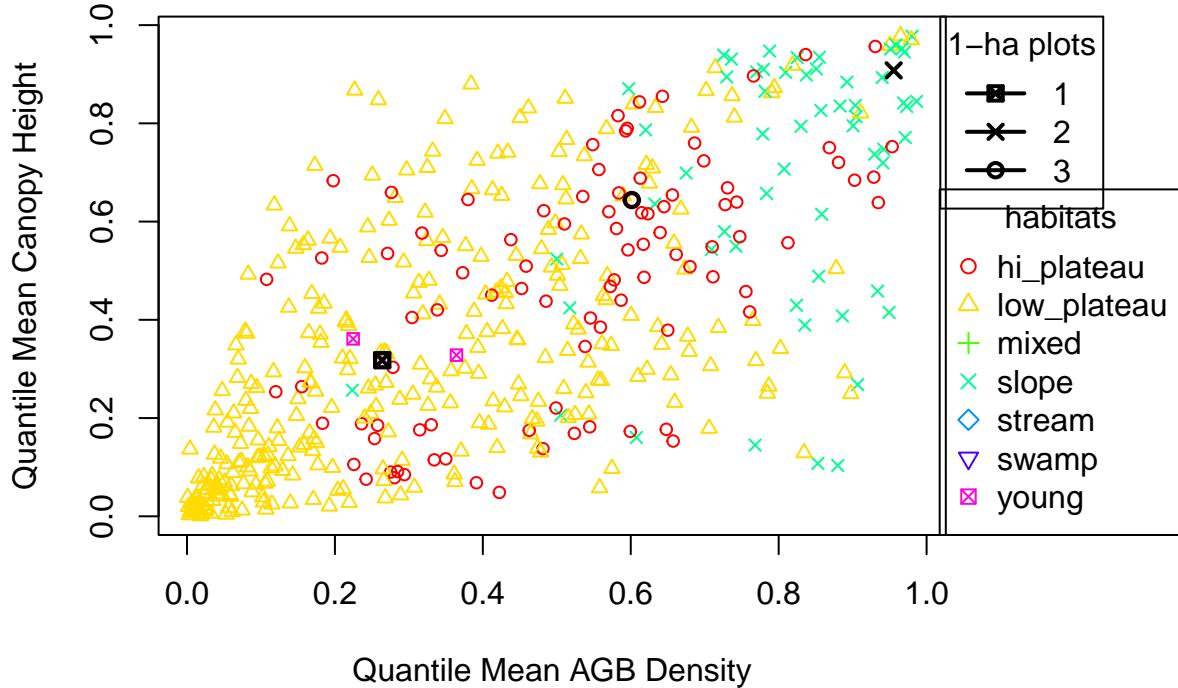


```
foc2varvshab1ha(df1ha,z1varname="mnagb",z2varname="mncanht",
                  z1vartitle="Mean AGB Density",z2vartitle="Mean Canopy Height",
                  mindominance=0.8)
```

### Only Plots $\geq 80\%$ Single Habitat



## Only Plots >= 80 % Single Habitat



```
# this set isn't very useful because so closely correlated, so drop
#foc2varushab1ha(df1ha,z1varname="mnagb",z2varname="mnba",
#                  z1vartitle="Mean AGB Density",z2vartitle="Mean Basal Area",
#                  mindominance=0.8)
```

### Proposed plots in relation to liana stem map

```
plot(c(0,0,1000,1000),c(0,500,0,500),pch=16, main = "Liana stem map and potential 1-ha plots")
rasterImage(lianamap,xleft=-offset,ybottom=-offset,xright=1000+offset,ytop=500+offset)
lines(c(0,0,1000,1000,0),c(0,500,500,0,0))
col1ha <- "red"
for (i in 1:nrow(newtlscenters)) {
  lines(x=rep(newtlscenters$x[i]-50,2), y=c(newtlscenters$y[i]-50,newtlscenters$y[i]+50),lwd=3,lty=3,col=col1ha)
  lines(x=rep(newtlscenters$x[i]+50,2), y=c(newtlscenters$y[i]-50,newtlscenters$y[i]+50),lwd=3,lty=3,col=col1ha)
  lines(x=c(newtlscenters$x[i]-50,newtlscenters$x[i]+50), y=rep(newtlscenters$y[i]-50,2),lwd=3,lty=3,col=col1ha)
  lines(x=c(newtlscenters$x[i]-50,newtlscenters$x[i]+50), y=rep(newtlscenters$y[i]+50,2),lwd=3,lty=3,col=col1ha)
}
```

## Liana stem map and potential 1-ha plots

