Remote Sensing in R

Atkins, Stovall, and Silva

7/28/2020

Remote Sensing of Forest Succession in R with `lidR` and `ForestGapR`

Remote sensing science requires handling large data often hundreds to thousands of megabytes in size per file, with analyses typically including multiple files requiring complex operations, atmospheric correction, height normalization, band algebra–necessitating both access to sufficient computing and skills development resources. The R environment and package development community have helped to meet this need. In this example, we will show a brief analysis of forest succession using NEON Airborne Observatory (AOP) discrete return lidar data acquired at the Blandy Experimental Farm (NEON site BLAN) in 2019. Blandy is a 300-ha biological field station owned by the University of Virginia that includes 120 ha of pasture and cropland, 40 ha of woodland, the 60 ha Virginia State Arboretum, and 80 ha of old fields in early, middle, and late succession (Bowers, 1997). We will be focusing this tutorial on one of these successional chronosquences

Installation of lidR and ForestGapR

Many remote sensing centered packages have other package dependencies, meaning they necessitate the installation of additional packages. For this example, we will be using functions from the `lidR`, `ForestGapR`, and `sf` packages. However, additionally, we will need to make sure that the `devtools`, `ggplot`, `raster` and `viridis` package are installed.

```
#The development version install requires devtools:
library(devtools)
```

```
## Loading required package: usethis
```

```
# lidR
if(!require(lidR)){install.packages("lidR")}
```

```
## Loading required package: lidR
```

```
## Loading required package: raster
```

```
## Loading required package: sp
```

```
library(lidR)
# sf
if(!require(sf)){install.packages("sf")}
```

```
## Loading required package: sf
```

```
## Linking to GEOS 3.9.0, GDAL 3.2.1, PROJ 7.2.1
```

```
library(sf)
# sf
if(!require(ggplot2)){install.packages("ggplot2")}
```

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

```
library(ggplot2)
# install the package if not already installed and call it via library()
if(!require(ForestGapR)){devtools::install_github("carlos-alberto-silva/ForestGapR")}
```

```
## Loading required package: ForestGapR
```

```
library(ForestGapR)
# ForestGapR additionally requires the `raster` packages
if(!require(raster)){install.packages("raster")}
library(raster)
# ForestGapR additionally requires the `viridis` packages
if(!require(viridis)){install.packages("viridis")}
```

```
## Loading required package: viridis
```

```
## Loading required package: viridisLite
```

```
library(viridis)
```

Importing Raw Point Clouds with `lidR`

Raw lidar point clouds often come in the `.las` file format, which the `lidR` package works with quite readily. For this tutorial, our example data set is a mosaicked point cloud of four separate `.las` files chosen because they include the area of the Blandy Experimental Farm that includes the old field abandonment study, specfiically the southwestern successional chronosquence which includes an early succesional field abandoned in 2001, a middle successional field abandoned in 1986, and a late succesional field abandoned in 1910. Wang et al. (2010) and Aneece et al. (2017) provide greater context and detail on these successional chronosequences.

First we import our raw point cloud using the `readLAS()` function from the `lidR` package:

```
# Raw point cloud of the approximate succession field in Blandy
las <- lidR::readLAS("../inst/extdata/remote_sensing/blandy_successional_gradient.las")

# double check for outlier points here with an upper limit of 40 m
las <- lidR::filter_poi(las, Z > 0, Z < 60)
```

In mosaicking this point cloud, we have already ran appropriate corrections and normalizations. However, if a user were to be working directly with raw `.las` files, it is necessary to run those processes, The code below, which created the source mosaicked point cloud we are using, accomplishes this using the `normalize_height()` function from `lidR` with an

```
# create a directory object
las.dir <- "D:/Blandy/NEON_lidar-point-cloud-line_BLANDY/NEON_lidar-point-cloud-line/NEON.D02.BLAN.DP1.3

# list file names within that directory
# specfically here I have included a pattern to match the files I need that include all four tiles with
# 752000 - 754000, and 4327000 - 4329000, by specifically naming those files in the `pattern = ` option
# late.file
file.names <- list.files(path = las.dir, pattern = "*_752000_4327000_classified_point_cloud_colorized.la

# 1
las <- readLAS(file.names[1])

las.norm <- lidR::normalize_height(las, algorithm = tin(), na.rm = TRUE, res = 1)
las.norm <- filter_poi(las.norm, Z > 0, Z < 60)

# 2
las2 <- readLAS(file.names[2])

las.norm2<- lidR::normalize_height(las2, algorithm = tin(), na.rm = TRUE, res = 1)
las.norm2 <- filter_poi(las.norm2, Z > 0, Z < 60)

# 3
las3 <- readLAS(file.names[3])

las.norm3<- lidR::normalize_height(las3, algorithm = tin(), na.rm = TRUE, res = 1)
las.norm3 <- filter_poi(las.norm3, Z > 0, Z < 60)

# 4
las4 <- readLAS(file.names[4])

las.norm4 <- lidR::normalize_height(las4, algorithm = tin(), na.rm = TRUE, res = 1)
las.norm4 <- filter_poi(las.norm4, Z > 0, Z < 60)

#plot(las.norm3)
x <- rbind(las.norm, las.norm2, las.norm3, las.norm4)

plot(x)

writeLAS(x, "blandy_successional_gradient.las")
```

Our point cloud includes some areas outside of our study area, necessitating that we clip our point cloud using a standard shapefile which can be uploaded in the workspace using the `st_read()` function from the `sf` package. Then we can clip our point cloud using that shapefile within th `clip_roi()` function from the `lidR` package and then plot:

```
# load in shapefile
blandy.success <- st_read("../inst/extdata/remote_sensing/blandy_succesional_forest_one-polygon.shp")
```

```
## Reading layer 'blandy_succesional_forest_one-polygon' from data source
##   'C:\R\rforestanalysis\inst\extdata\remote_sensing\blandy_succesional_forest_one-polygon.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 1 feature and 3 fields
## Geometry type: POLYGON
```

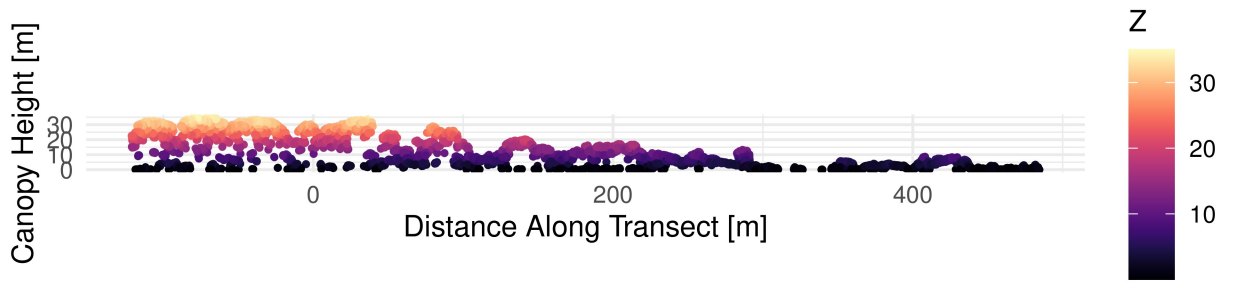
```
## Dimension:     XY
## Bounding box:  xmin: -78.07989 ymin: 39.05629 xmax: -78.07019 ymax: 39.06372
## Geodetic CRS:  WGS 84
```

```r
# convert that shapefile to UTM
blandy.success <- st_transform(blandy.success, crs = st_crs(las))

# now we clip the forest
las.clip <- lidR::clip_roi(las, geometry = blandy.success)

# plotting
lidR::plot(las.clip, colorPalette = viridis::magma(24), bg = "white", axis = FALSE, legend = TRUE)
```

`lidR` also has functionality to clip circles, rectangles, etc. from data, depending on the user's need.

For visualization purpose, let's cut a transect out of this data. This can be done using the `clip_transet()` function in `lidR` with specified end points:

```r
# Endpoints
p1 <- c(752849, 4327711) # these are coordinates
p2 <- c(753399, 4327436) # these are coordinates

# clipping the transect
las_tr <- clip_transect(las, p1, p2, width = 4, xz = TRUE)

# now let's plot the transect using ggplot2
# x11(width = 12, height = 2)
ggplot(las_tr@data, aes(X,Z, color = Z))+
  geom_point(size = 0.75)+
  coord_equal()+
  theme_minimal()+
  xlab("Distance Along Transect [m]")+
  ylab("Canopy Height [m]")+
  scale_color_gradientn(colours =  viridis::magma(24))
```



Statistical Analysis of Lidar Data in R

One method of analyzing lidar data is to work with structural diversity and complexity metrics (Parker and Russ, 2004; Atkins et al. 2018) which can distill complex, rich data sets like lidar point clouds which may have millions and millions of data points into tractable metrics and indices. This is analgous to established methods in landscape ecology and forestry in using landscape metrics such as contagion and connectivity, or ecological methods such as vegetation indices (e.g. NDVI, EVI, SAVI) which can be derived from spectrometer data (e.g. Landsat, MODIS).

`lidR` includs a function `grid_metrics()` which can be modifed using custom functions or used with built in functions, to calculate gridded, lidar derived metrics. Here we first define a custom `myMetrics()` function. Within this, we create inputs for return height (`z`), return number (`rn`), return intensity (`i`):

```
myMetrics <- function(z, rn, i){
  first  = rn == 1L
  zfirst = z[first]
  nfirst = length(zfirst)
  nz = length(z)


  metrics = list(
    fhd =   (entropy(z, zmax = max(z)) * log(max(z))),  #foliar height diversity
    vdr = ( (max(z) - median(z)) / max(z)),
    top_rug = sd(zfirst),
    p95 = quantile(z, probs = 0.95))
```

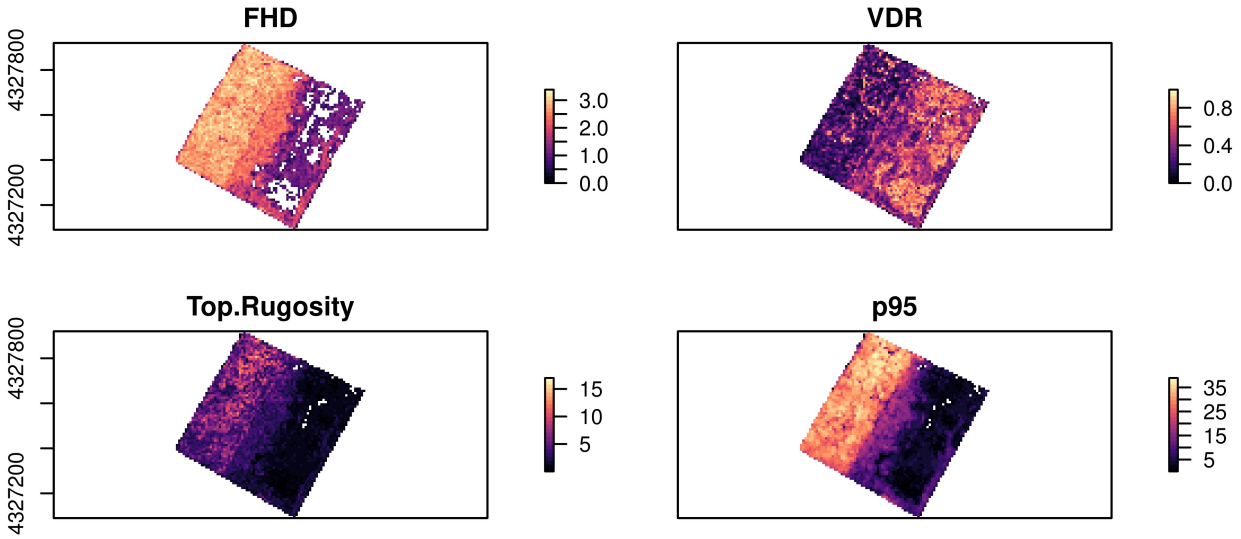
```
  return(metrics)
}
```

Within this function we have included four lidar metrics that have been used broadly in forest research, especially for describing succesional gradients and forest age:

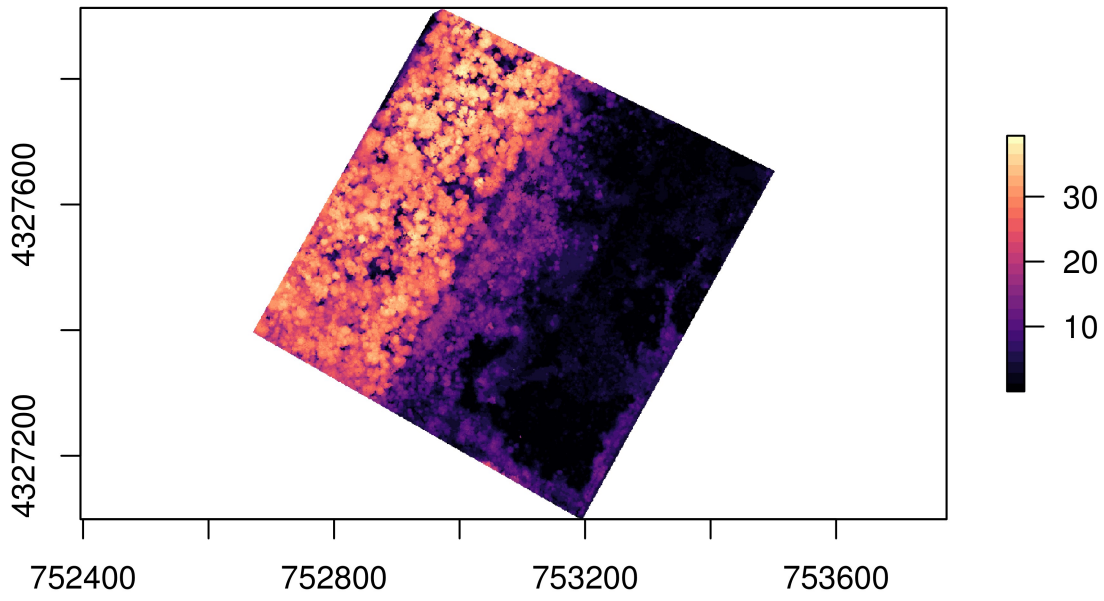
1. fhd - Foliar height diversity (MacArthur and MacArthur, 1969)
2. vdr - Vertical Distribution Ratio (XXXX)
3. top_rug - Top rugosity (Parker and Russ, 2004)
4. p95 - 95th height percentile (XXXX)

We then pass this custom function to the `grid_metrics()` function, which produces a raster stack with each layer a raster of our calculated metric at the resolution declared in the `res =` option in `grid_metrics()`, here we use 10, corresponding to a 10 x 10 meter grid–the units are defined by the units of the `.las` file. Our `.las` file is in UTM coordinates, thus in meters. Be aware of this! We then plot the rasters:

```
# custom metrics function, note input variables and resolution, which is set at 10 as in 10 meters
  metrics <- lidR::grid_metrics(las.clip, ~myMetrics(z = Z, rn=ReturnNumber, i = Intensity), res = 10 )

  #this renames the layers in our raster stack
  stack.names = c("FHD", "VDR", "Top Rugosity", "p95")
  names(metrics) <- stack.names

  # we now plot using base R
  plot(metrics, col = viridis::magma(24), nc = 2, nr = 3)
```



```r
# first make th
chm <- lidR::grid_canopy(las.clip, res = 1, algorithm = dsmtin())

plot(chm, col = viridis::magma(24))
```



Analzing Forest Gap Development Using `ForestGapR`

Often forest gaps develop as a forest ages. Detection of forest gaps has been rapidly advanced by the use of lidar. While forest gap detection methods exist using raw lidar point clouds, `ForestGapR` works directly with canopy height models (CHM) which are in raster file format (`.tif`)–a file format that is two to three orders or more smaller in file size than `.las` files and far more readily available and accessible than `.las` files.

To use `ForestGapR` we first must declare a detection threshold value, here we use 5 m. This means the algorithm is going to look for ....

And we also want to define the area of our gap, here a minimum of 1 m in area and a maximum of 1000 m, with anything greater than 1000 m ignored.

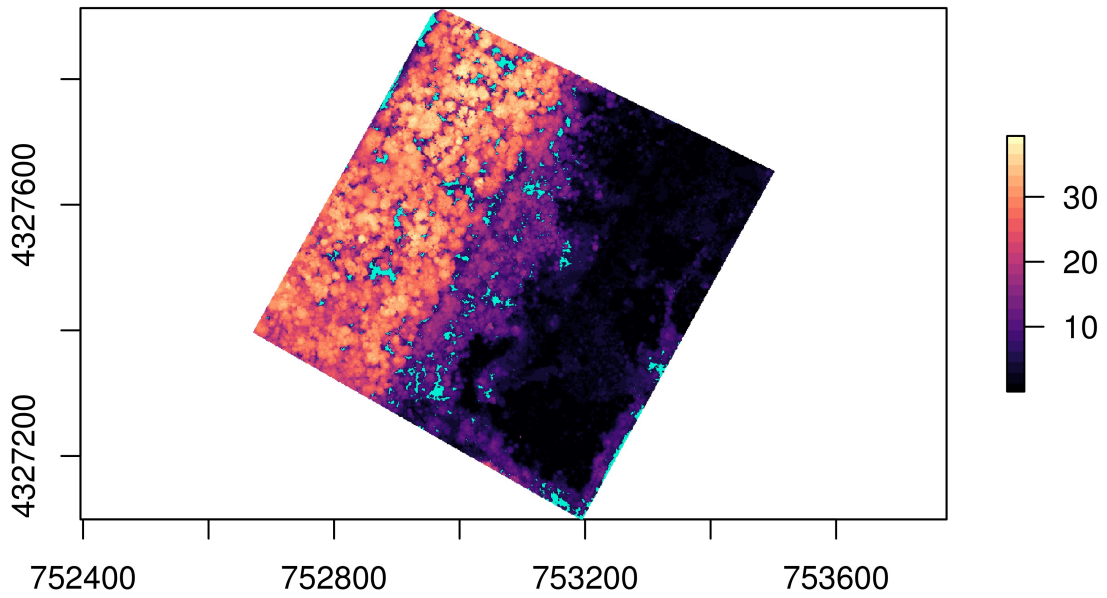
```r
# Setting height thresholds (e.g. 10 meters)
threshold <- 5
size <- c(1,1000) # m2
```

Now that we have our parameters we rune the `getForestGaps()` function with our defined parameters and then plot those data by adding the gaps layer to our canopy height model raster layer using base R plotting:

```
# Detecting forest gaps
gaps_blandy <- ForestGapR::getForestGaps(chm_layer = chm, threshold = threshold, size = size)
```

```
## Loading required namespace: igraph
```

```
# Plotting chm
plot(chm, col = viridis::magma(24))
# Plotting gaps
plot(gaps_blandy, col = "#00F0D4", add = TRUE, main="Forest Canopy Gap", legend = FALSE)
```



We can then use the `GapStats()` function to run additional analyses on our detected forest gaps and plot those distributions:

```
# Computing basic statistics of forest gap
gaps_stats <- ForestGapR::GapStats(gap_layer = gaps_blandy, chm_layer = chm)

# Gap-size Frequency Distributions
ForestGapR::GapSizeFDist(gaps_stats = gaps_stats, method = "Hanel_2017", col = "forestgreen", pch=16, c
axes=FALSE,ylab="Gap Frequency",xlab=as.expression(bquote("Gap Size" ~ (m^2) )))
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 478 y values <= 0 omitted from
## logarithmic plot
```

```
## $lambda
```

```
## [1] 1.74692
##
## $gap.freq
##        gap.size gap.freq
##   [1,]        1      227
##   [2,]        2       93
##   [3,]        3       49
##   [4,]        4       37
##   [5,]        5       18
##   [6,]        6       19
##   [7,]        7       11
##   [8,]        8       13
##   [9,]        9        9
##  [10,]       10        7
##  [11,]       11        9
##  [12,]       12        6
##  [13,]       13        9
##  [14,]       14        3
##  [15,]       15        5
##  [16,]       16        4
##  [17,]       17        6
##  [18,]       18        5
##  [19,]       19        4
##  [20,]       20        1
##  [21,]       21        4
##  [22,]       22        3
##  [23,]       23        2
##  [24,]       24        2
##  [25,]       25        3
##  [26,]       26        2
##  [27,]       27        3
##  [28,]       28        4
##  [29,]       29        2
##  [30,]       30        3
##  [31,]       31        3
##  [32,]       32        1
##  [33,]       33        0
##  [34,]       34        0
##  [35,]       35        2
##  [36,]       36        3
##  [37,]       37        0
##  [38,]       38        2
##  [39,]       39        0
##  [40,]       40        1
##  [41,]       41        3
##  [42,]       42        3
##  [43,]       43        0
##  [44,]       44        2
##  [45,]       45        2
##  [46,]       46        0
##  [47,]       47        1
##  [48,]       48        0
##  [49,]       49        4
##  [50,]       50        0
```

```
##  [51,]       51        1
##  [52,]       52        0
##  [53,]       53        0
##  [54,]       54        0
##  [55,]       55        2
##  [56,]       56        2
##  [57,]       57        0
##  [58,]       58        1
##  [59,]       59        1
##  [60,]       60        0
##  [61,]       61        0
##  [62,]       62        1
##  [63,]       63        1
##  [64,]       64        0
##  [65,]       65        0
##  [66,]       66        1
##  [67,]       67        1
##  [68,]       68        0
##  [69,]       69        0
##  [70,]       70        0
##  [71,]       71        1
##  [72,]       72        0
##  [73,]       73        0
##  [74,]       74        1
##  [75,]       75        1
##  [76,]       76        2
##  [77,]       77        0
##  [78,]       78        1
##  [79,]       79        0
##  [80,]       80        0
##  [81,]       81        1
##  [82,]       82        0
##  [83,]       83        0
##  [84,]       84        0
##  [85,]       85        1
##  [86,]       86        0
##  [87,]       87        0
##  [88,]       88        0
##  [89,]       89        1
##  [90,]       90        0
##  [91,]       91        0
##  [92,]       92        0
##  [93,]       93        0
##  [94,]       94        0
##  [95,]       95        0
##  [96,]       96        0
##  [97,]       97        0
##  [98,]       98        0
##  [99,]       99        0
## [100,]      100        0
## [101,]      101        1
## [102,]      102        0
## [103,]      103        0
## [104,]      104        0
```

```
## [105,]      105        1
## [106,]      106        0
## [107,]      107        0
## [108,]      108        1
## [109,]      109        0
## [110,]      110        0
## [111,]      111        0
## [112,]      112        0
## [113,]      113        0
## [114,]      114        0
## [115,]      115        0
## [116,]      116        0
## [117,]      117        0
## [118,]      118        0
## [119,]      119        0
## [120,]      120        0
## [121,]      121        0
## [122,]      122        0
## [123,]      123        0
## [124,]      124        2
## [125,]      125        1
## [126,]      126        0
## [127,]      127        0
## [128,]      128        0
## [129,]      129        0
## [130,]      130        0
## [131,]      131        0
## [132,]      132        0
## [133,]      133        0
## [134,]      134        1
## [135,]      135        0
## [136,]      136        0
## [137,]      137        1
## [138,]      138        0
## [139,]      139        2
## [140,]      140        0
## [141,]      141        0
## [142,]      142        0
## [143,]      143        1
## [144,]      144        1
## [145,]      145        0
## [146,]      146        0
## [147,]      147        0
## [148,]      148        0
## [149,]      149        0
## [150,]      150        0
## [151,]      151        0
## [152,]      152        0
## [153,]      153        0
## [154,]      154        0
## [155,]      155        0
## [156,]      156        0
## [157,]      157        0
## [158,]      158        0
```

```
## [159,]      159        0
## [160,]      160        0
## [161,]      161        0
## [162,]      162        0
## [163,]      163        0
## [164,]      164        0
## [165,]      165        0
## [166,]      166        0
## [167,]      167        0
## [168,]      168        0
## [169,]      169        0
## [170,]      170        0
## [171,]      171        0
## [172,]      172        0
## [173,]      173        0
## [174,]      174        0
## [175,]      175        0
## [176,]      176        0
## [177,]      177        0
## [178,]      178        0
## [179,]      179        0
## [180,]      180        0
## [181,]      181        0
## [182,]      182        0
## [183,]      183        0
## [184,]      184        0
## [185,]      185        0
## [186,]      186        0
## [187,]      187        0
## [188,]      188        0
## [189,]      189        0
## [190,]      190        0
## [191,]      191        0
## [192,]      192        0
## [193,]      193        0
## [194,]      194        0
## [195,]      195        0
## [196,]      196        0
## [197,]      197        0
## [198,]      198        0
## [199,]      199        0
## [200,]      200        0
## [201,]      201        0
## [202,]      202        0
## [203,]      203        0
## [204,]      204        0
## [205,]      205        0
## [206,]      206        0
## [207,]      207        0
## [208,]      208        0
## [209,]      209        0
## [210,]      210        0
## [211,]      211        1
## [212,]      212        0
```

```
## [213,]      213        0
## [214,]      214        0
## [215,]      215        0
## [216,]      216        0
## [217,]      217        0
## [218,]      218        0
## [219,]      219        0
## [220,]      220        0
## [221,]      221        0
## [222,]      222        0
## [223,]      223        0
## [224,]      224        0
## [225,]      225        0
## [226,]      226        0
## [227,]      227        0
## [228,]      228        0
## [229,]      229        0
## [230,]      230        0
## [231,]      231        0
## [232,]      232        0
## [233,]      233        0
## [234,]      234        0
## [235,]      235        0
## [236,]      236        0
## [237,]      237        0
## [238,]      238        0
## [239,]      239        0
## [240,]      240        0
## [241,]      241        0
## [242,]      242        0
## [243,]      243        0
## [244,]      244        0
## [245,]      245        0
## [246,]      246        0
## [247,]      247        0
## [248,]      248        0
## [249,]      249        0
## [250,]      250        0
## [251,]      251        0
## [252,]      252        0
## [253,]      253        0
## [254,]      254        0
## [255,]      255        0
## [256,]      256        0
## [257,]      257        0
## [258,]      258        0
## [259,]      259        0
## [260,]      260        0
## [261,]      261        0
## [262,]      262        0
## [263,]      263        1
## [264,]      264        0
## [265,]      265        0
## [266,]      266        0
```

```
## [267,]      267        0
## [268,]      268        0
## [269,]      269        0
## [270,]      270        0
## [271,]      271        0
## [272,]      272        0
## [273,]      273        0
## [274,]      274        0
## [275,]      275        0
## [276,]      276        0
## [277,]      277        0
## [278,]      278        0
## [279,]      279        0
## [280,]      280        0
## [281,]      281        0
## [282,]      282        0
## [283,]      283        0
## [284,]      284        0
## [285,]      285        0
## [286,]      286        0
## [287,]      287        0
## [288,]      288        0
## [289,]      289        0
## [290,]      290        0
## [291,]      291        0
## [292,]      292        0
## [293,]      293        0
## [294,]      294        0
## [295,]      295        0
## [296,]      296        0
## [297,]      297        0
## [298,]      298        0
## [299,]      299        0
## [300,]      300        0
## [301,]      301        0
## [302,]      302        0
## [303,]      303        0
## [304,]      304        0
## [305,]      305        0
## [306,]      306        0
## [307,]      307        0
## [308,]      308        0
## [309,]      309        0
## [310,]      310        0
## [311,]      311        0
## [312,]      312        1
## [313,]      313        0
## [314,]      314        0
## [315,]      315        0
## [316,]      316        0
## [317,]      317        0
## [318,]      318        0
## [319,]      319        0
## [320,]      320        0
```

```
## [321,]      321        0
## [322,]      322        0
## [323,]      323        0
## [324,]      324        0
## [325,]      325        0
## [326,]      326        0
## [327,]      327        0
## [328,]      328        0
## [329,]      329        0
## [330,]      330        0
## [331,]      331        0
## [332,]      332        0
## [333,]      333        0
## [334,]      334        0
## [335,]      335        0
## [336,]      336        0
## [337,]      337        0
## [338,]      338        0
## [339,]      339        0
## [340,]      340        1
## [341,]      341        0
## [342,]      342        0
## [343,]      343        0
## [344,]      344        0
## [345,]      345        0
## [346,]      346        0
## [347,]      347        0
## [348,]      348        0
## [349,]      349        0
## [350,]      350        0
## [351,]      351        0
## [352,]      352        0
## [353,]      353        0
## [354,]      354        0
## [355,]      355        0
## [356,]      356        1
## [357,]      357        0
## [358,]      358        0
## [359,]      359        0
## [360,]      360        0
## [361,]      361        0
## [362,]      362        0
## [363,]      363        0
## [364,]      364        0
## [365,]      365        1
## [366,]      366        0
## [367,]      367        0
## [368,]      368        0
## [369,]      369        0
## [370,]      370        0
## [371,]      371        0
## [372,]      372        0
## [373,]      373        0
## [374,]      374        0
```

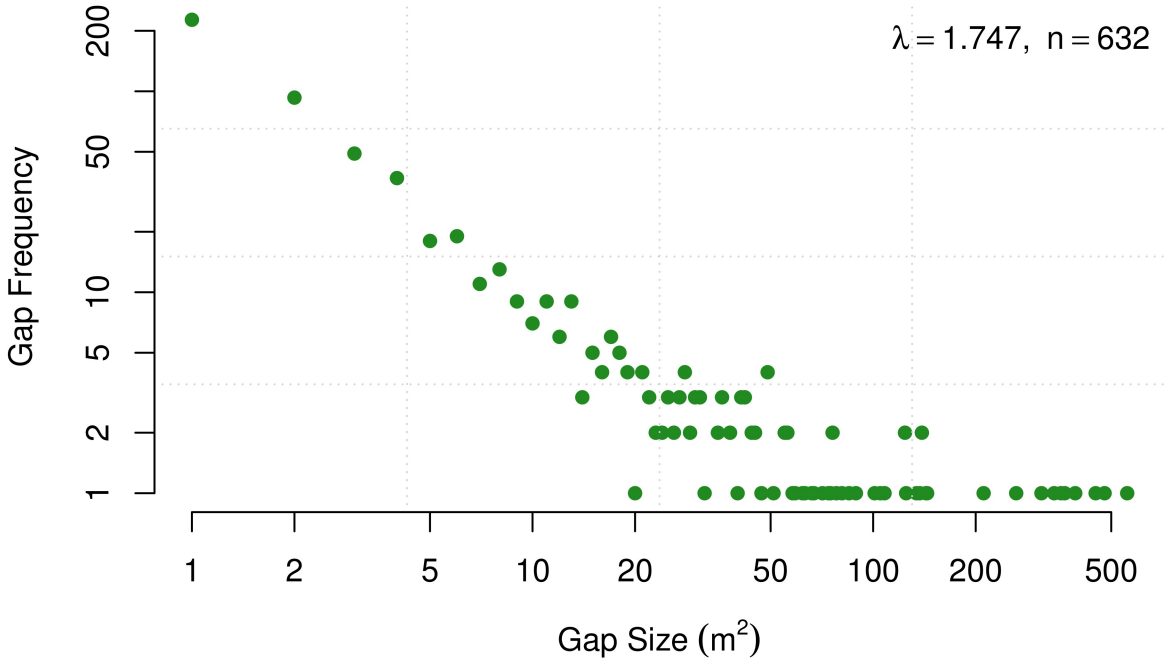
```
## [375,]      375        0
## [376,]      376        0
## [377,]      377        0
## [378,]      378        0
## [379,]      379        0
## [380,]      380        0
## [381,]      381        0
## [382,]      382        0
## [383,]      383        0
## [384,]      384        0
## [385,]      385        0
## [386,]      386        0
## [387,]      387        0
## [388,]      388        0
## [389,]      389        0
## [390,]      390        0
## [391,]      391        0
## [392,]      392        1
## [393,]      393        0
## [394,]      394        0
## [395,]      395        0
## [396,]      396        0
## [397,]      397        0
## [398,]      398        0
## [399,]      399        0
## [400,]      400        0
## [401,]      401        0
## [402,]      402        0
## [403,]      403        0
## [404,]      404        0
## [405,]      405        0
## [406,]      406        0
## [407,]      407        0
## [408,]      408        0
## [409,]      409        0
## [410,]      410        0
## [411,]      411        0
## [412,]      412        0
## [413,]      413        0
## [414,]      414        0
## [415,]      415        0
## [416,]      416        0
## [417,]      417        0
## [418,]      418        0
## [419,]      419        0
## [420,]      420        0
## [421,]      421        0
## [422,]      422        0
## [423,]      423        0
## [424,]      424        0
## [425,]      425        0
## [426,]      426        0
## [427,]      427        0
## [428,]      428        0
```

```
## [429,]      429        0
## [430,]      430        0
## [431,]      431        0
## [432,]      432        0
## [433,]      433        0
## [434,]      434        0
## [435,]      435        0
## [436,]      436        0
## [437,]      437        0
## [438,]      438        0
## [439,]      439        0
## [440,]      440        0
## [441,]      441        0
## [442,]      442        0
## [443,]      443        0
## [444,]      444        0
## [445,]      445        0
## [446,]      446        0
## [447,]      447        0
## [448,]      448        0
## [449,]      449        0
## [450,]      450        1
## [451,]      451        0
## [452,]      452        0
## [453,]      453        0
## [454,]      454        0
## [455,]      455        0
## [456,]      456        0
## [457,]      457        0
## [458,]      458        0
## [459,]      459        0
## [460,]      460        0
## [461,]      461        0
## [462,]      462        0
## [463,]      463        0
## [464,]      464        0
## [465,]      465        0
## [466,]      466        0
## [467,]      467        0
## [468,]      468        0
## [469,]      469        0
## [470,]      470        0
## [471,]      471        0
## [472,]      472        0
## [473,]      473        0
## [474,]      474        0
## [475,]      475        0
## [476,]      476        0
## [477,]      477        0
## [478,]      478        1
## [479,]      479        0
## [480,]      480        0
## [481,]      481        0
## [482,]      482        0
```

```
## [483,]      483        0
## [484,]      484        0
## [485,]      485        0
## [486,]      486        0
## [487,]      487        0
## [488,]      488        0
## [489,]      489        0
## [490,]      490        0
## [491,]      491        0
## [492,]      492        0
## [493,]      493        0
## [494,]      494        0
## [495,]      495        0
## [496,]      496        0
## [497,]      497        0
## [498,]      498        0
## [499,]      499        0
## [500,]      500        0
## [501,]      501        0
## [502,]      502        0
## [503,]      503        0
## [504,]      504        0
## [505,]      505        0
## [506,]      506        0
## [507,]      507        0
## [508,]      508        0
## [509,]      509        0
## [510,]      510        0
## [511,]      511        0
## [512,]      512        0
## [513,]      513        0
## [514,]      514        0
## [515,]      515        0
## [516,]      516        0
## [517,]      517        0
## [518,]      518        0
## [519,]      519        0
## [520,]      520        0
## [521,]      521        0
## [522,]      522        0
## [523,]      523        0
## [524,]      524        0
## [525,]      525        0
## [526,]      526        0
## [527,]      527        0
## [528,]      528        0
## [529,]      529        0
## [530,]      530        0
## [531,]      531        0
## [532,]      532        0
## [533,]      533        0
## [534,]      534        0
## [535,]      535        0
## [536,]      536        0
```

```
## [537,]      537        0
## [538,]      538        0
## [539,]      539        0
## [540,]      540        0
## [541,]      541        0
## [542,]      542        0
## [543,]      543        0
## [544,]      544        0
## [545,]      545        0
## [546,]      546        0
## [547,]      547        0
## [548,]      548        0
## [549,]      549        0
## [550,]      550        0
## [551,]      551        0
## [552,]      552        0
## [553,]      553        0
## [554,]      554        0
## [555,]      555        0
## [556,]      556        0
## [557,]      557        1
##
## $method
## [1] "Hanel_2017"
```

```
axis(1);axis(2)
grid(4,4)
```



More information can be found at: <https://github.com/carlos-alberto-silva/ForestGapR >