Raster data

Víctor Granda

Ecosystem Modelling Facility

15/03/2022

Raster data

Raster data consist of a matrix of pixels (or cells) organized into rows and columns (also know as grid), where each pixel contains a value representing information (such an attribute like in the simple features, i.e. biomass).

Raster data

R packages

There are several packages to work with raster data:

- `raster`: original and old standard to work with raster data in R.
- `terra`: modern standard to work with raster data in R.
- `stars`: work with raster, and also datacubes (more than 3 dimensions).

In this session we will explore the `terra` package, as it has most of the most common tools to work with raster data implemented, as well as a nice online documentation (https://rspatial.org/terra/pkg/index.html)

Raster data

`terra` package

```
install.package(terra)
```

Raster data

Read data

```
library(terra)
```

```
## terra 1.5.21
```

```
##
## Attaching package: 'terra'
```

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

```
## The following object is masked from 'package:dplyr':
##
##     src
```

```
lidar_ab_raster <- rast('lidar_ab_raster.tif')
lidar_ab_raster
```

```
## class       : SpatRaster
## dimensions  : 660, 680, 1  (nrow, ncol, nlyr)
## resolution  : 400, 400  (x, y)
## extent      : 256000, 528000, 4488000, 4752000  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=31 +ellps=GRS80 +units=m +no_defs
## source      : lidar_ab_raster.tif
## name        : lidar_ab_raster
```

Raster data

Layers

```
lidar_dbh_raster <- rast('lidar_dbh_raster.tif')
lidar_raster <- c(lidar_ab_raster, lidar_dbh_raster)
lidar_raster
```

```
## class       : SpatRaster
## dimensions  : 660, 680, 2  (nrow, ncol, nlyr)
## resolution  : 400, 400  (x, y)
## extent      : 256000, 528000, 4488000, 4752000  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=31 +ellps=GRS80 +units=m +no_defs
## sources     : lidar_ab_raster.tif
##               lidar_dbh_raster.tif
## names       : lidar_ab_raster, lidar_dbh_raster
## min values  :           0.840,            9.562
## max values  :          86.988,           40.000
```

Raster data

Write data

```
writeRaster(lidar_raster, 'lidar_raster.tif', overwrite = TRUE)
rast('lidar_raster.tif')
```

```
## class       : SpatRaster
## dimensions  : 660, 680, 2  (nrow, ncol, nlyr)
## resolution  : 400, 400  (x, y)
## extent      : 256000, 528000, 4488000, 4752000  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=31 +ellps=GRS80 +units=m +no_defs
## source      : lidar_raster.tif
## names       : lidar_ab_raster, lidar_dbh_raster
## min values  :           0.840,            9.562
## max values  :          86.988,           40.000
```

Raster data

CRS

CRS is a framework used to precisely measure locations on the surface of the Earth as coordinates.

```
crs(lidar_raster)
```

```
## [1] "PROJCRS[\"unknown\",\n    BASEGEOGCRS[\"unknown\",\n        DATUM[\"Unknown_based_on_GRS80_ellip
```

Raster data

transform CRS

```
lidar_raster <- terra::project(lidar_raster, 'epsg:3043')
lidar_raster
```

```
## class       : SpatRaster
## dimensions  : 660, 680, 2  (nrow, ncol, nlyr)
## resolution  : 400, 400  (x, y)
## extent      : 256000, 528000, 4488000, 4752000  (xmin, xmax, ymin, ymax)
## coord. ref. : ETRS89 / UTM zone 31N (N-E) (EPSG:3043)
## source      : memory
## names       : lidar_ab_raster, lidar_dbh_raster
## min values  :           0.840,            9.562
## max values  :          86.988,           40.000
```

Raster data

Aggregate / Dissaggregate

Aggregate (less resolution = bigger pixel)

```
terra::aggregate(lidar_raster, fact = 2, fun = 'max')
```

```
## class       : SpatRaster
## dimensions  : 330, 340, 2  (nrow, ncol, nlyr)
## resolution  : 800, 800  (x, y)
## extent      : 256000, 528000, 4488000, 4752000  (xmin, xmax, ymin, ymax)
## coord. ref. : ETRS89 / UTM zone 31N (N-E) (EPSG:3043)
## source      : memory
## names       : lidar_ab_raster, lidar_dbh_raster
## min values  :        2.962121,        13.256105
## max values  :          86.988,           40.000
```

Dissaggregate (more resolution = smaller pixel)

```
terra::disagg(lidar_raster, fact = 2, method = 'bilinear')
```

```
## class       : SpatRaster
## dimensions  : 1320, 1360, 2  (nrow, ncol, nlyr)
## resolution  : 200, 200  (x, y)
## extent      : 256000, 528000, 4488000, 4752000  (xmin, xmax, ymin, ymax)
## coord. ref. : ETRS89 / UTM zone 31N (N-E) (EPSG:3043)
## source      : memory
## names       : lidar_ab_raster, lidar_dbh_raster
```

Raster data

Extract values from vector data

```
library(sf)
lidar_data_sf <- st_read('lidar_data.gpkg', quiet = TRUE) %>%
  st_transform(crs = 3043)
lidar_data_terra <- vect(lidar_data_sf)
lidar_data_terra
```

```
##  class       : SpatVector
##  geometry    : polygons
##  dimensions  : 42, 9  (geometries, attributes)
##  extent      : 260445.7, 526467.6, 4488771, 4747976  (xmin, xmax, ymin, ymax)
##  coord. ref. : ETRS89 / UTM zone 31N (N-E) (EPSG:3043)
##  names       :     poly_id poly_km2 AB_pixels AB_average AB_sd AB_min AB_max AB_km2 AB_km2_perc
##  type        :       <chr>    <num>     <int>      <num> <num>  <num>  <num>  <num>        <num>
##  values      :    Alt Camp    537.6    482120      16.69 8.207   0.01    100  192.8        35.87
##                Alt Empordà     1357   1393054      20.31 8.609   0.01    100  557.2        41.07
##                Alt Penedès    592.8    488554      17.66 9.036   0.01    100  195.4        32.97
```

Raster data

Extract values from vector data

```
terra::extract(lidar_raster, lidar_data_terra, mean, na.rm = TRUE)
```

```
##    ID lidar_ab_raster lidar_dbh_raster
## 1   1       15.246024         19.97279
## 2   2       17.112733         20.13811
## 3   3       16.668480         22.16253
## 4   4       22.944323         21.61816
## 5   5       23.772981         24.41451
## 6   6       14.912623         21.48832
## 7   7       15.089065         21.33481
## 8   8       14.756012         19.87355
## 9   9       13.563222         20.58355
## 10 10       17.493316         21.91562
## 11 11       17.376353         22.18640
## 12 12       12.982869         19.38242
## 13 13       17.507935         22.71191
## 14 14       22.657111         22.42527
## 15 15       28.354471         25.89686
## 16 16       13.906846         20.03831
## 17 17       13.558100         19.26940
## 18 18       10.117328         20.26445
## 19 19       23.841450         20.91880
## 20 20       20.705040         22.35493
## 21 21       20.138874         22.27460
## 22 22       12.530326         19.37777
## 23 23       10.137778         18.17510
## 24 24       21.333362         22.68933
```

Raster data

Extract values from vector data

```
lidar_centroids <- st_read('lidar_data.gpkg', quiet = TRUE) %>%
  st_transform(crs = 3043) %>%
  st_centroid() %>%
  vect()
```

```
## Warning in st_centroid.sf(.): st_centroid assumes attributes are constant over geometries of x
```

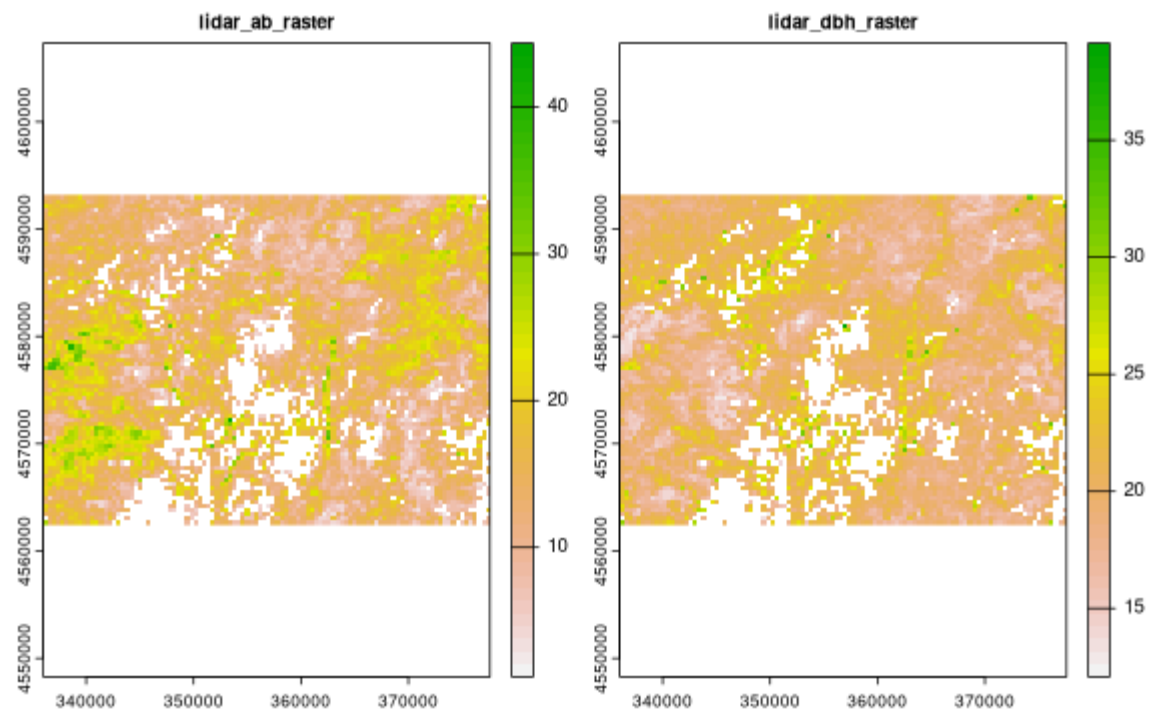
```
terra::extract(lidar_raster, lidar_centroids, method = 'simple')
```

```
##    ID lidar_ab_raster lidar_dbh_raster
## 1   1       17.454269         20.25319
## 2   2       20.164095         19.13410
## 3   3              NaN              NaN
## 4   4       14.875556         20.12202
## 5   5       18.549231         29.85897
## 6   6       14.612204         26.47458
## 7   7       17.752705         23.13799
## 8   8       19.885937         25.16281
## 9   9       10.533671         20.75886
## 10 10       22.389166         34.05958
## 11 11       23.931147         24.90984
## 12 12              NaN              NaN
## 13 13              NaN              NaN
## 14 14       22.744896         21.64656
## 15 15       18.794941         16.31494
## 16 16       10.836884         19.16870
```

Raster data

Crop and mask

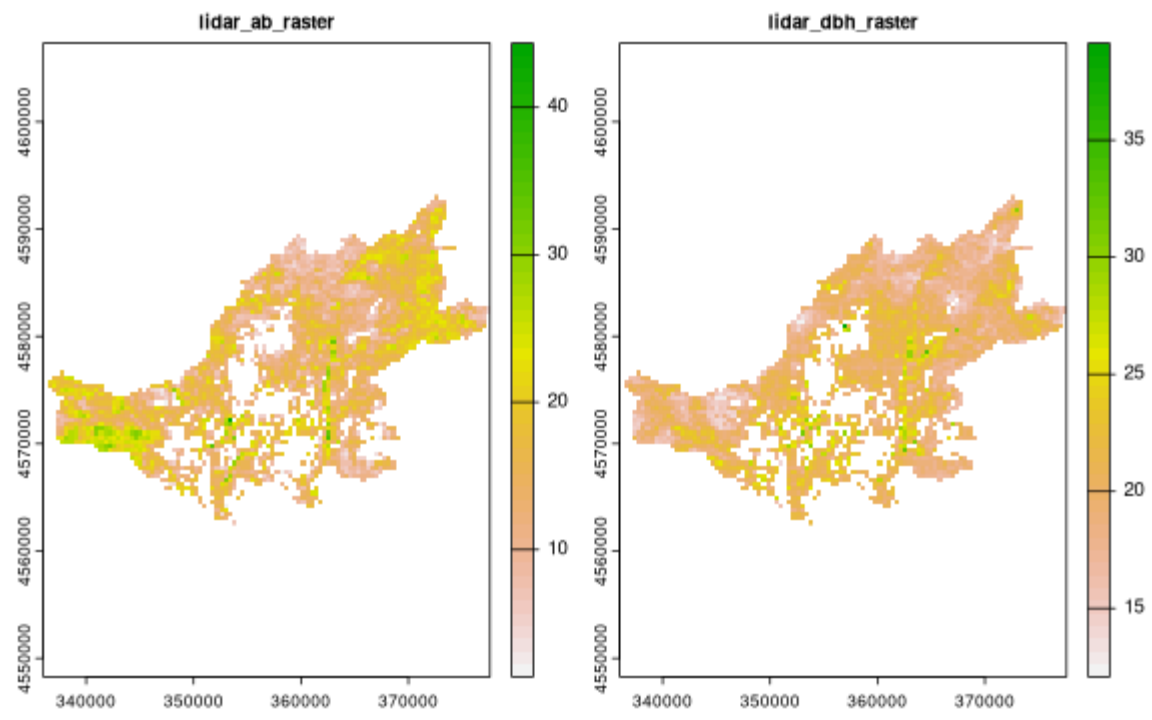
```
lidar_raster_alt_camp <- terra::crop(lidar_raster, lidar_data_terra[1,])
plot(lidar_raster_alt_camp)
```



Raster data

Crop and mask

```
lidar_raster_alt_camp_masked <-
  terra::crop(lidar_raster, lidar_data_terra[1,], mask = TRUE)
plot(lidar_raster_alt_camp_masked)
```



Raster data

Raster algebra

```
lidar_raster + c(1000, -1000)
```

```
## class       : SpatRaster
## dimensions  : 660, 680, 2  (nrow, ncol, nlyr)
## resolution  : 400, 400  (x, y)
## extent      : 256000, 528000, 4488000, 4752000  (xmin, xmax, ymin, ymax)
## coord. ref. : ETRS89 / UTM zone 31N (N-E) (EPSG:3043)
## source      : memory
## names       : lidar_ab_raster, lidar_dbh_raster
## min values  :        1000.840,         -990.438
## max values  :        1086.988,         -960.000
```

Raster data

Raster algebra

```
c(lidar_raster, lidar_ab_raster + lidar_dbh_raster)
```

```
## class       : SpatRaster
## dimensions  : 660, 680, 3  (nrow, ncol, nlyr)
## resolution  : 400, 400  (x, y)
## extent      : 256000, 528000, 4488000, 4752000  (xmin, xmax, ymin, ymax)
## coord. ref. : ETRS89 / UTM zone 31N (N-E) (EPSG:3043)
## sources     : memory  (2 layers)
##               memory
## names       : lidar_ab_raster, lidar_dbh_raster, lidar_ab_raster
## min values  :           0.840,            9.562,          16.400
## max values  :         86.9880,          40.0000,        122.4799
```

Raster data

Pixels algebra

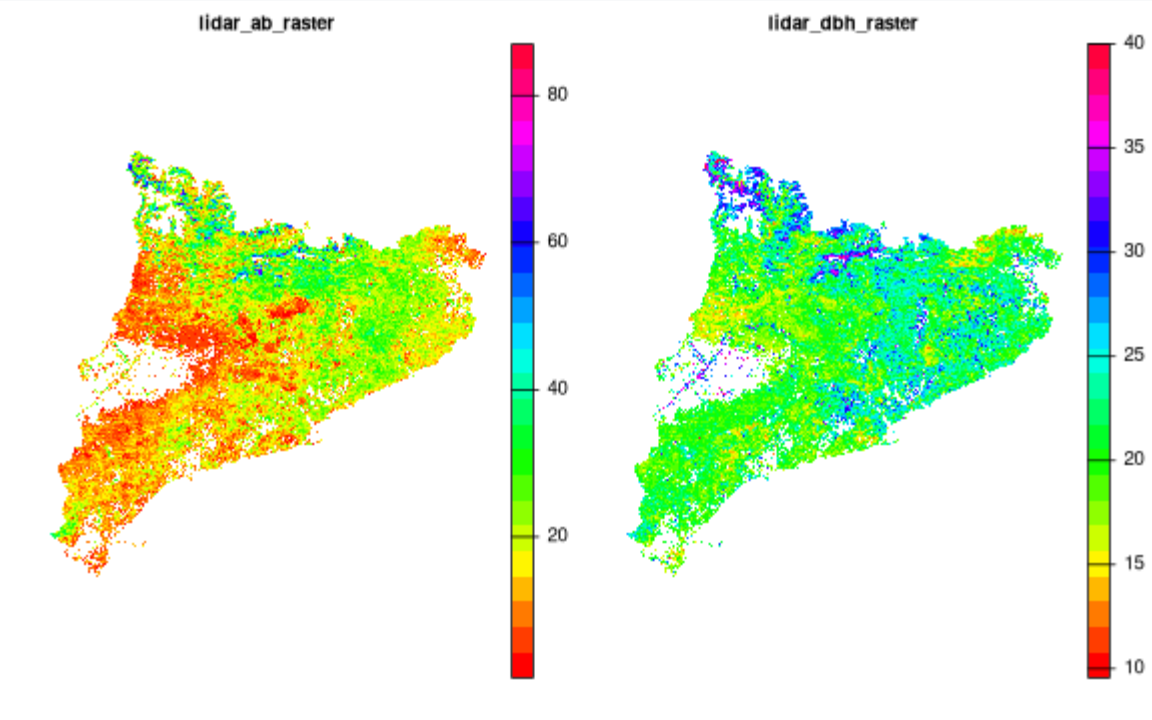
```
global(lidar_raster, "mean", na.rm = TRUE)
```

```
##                       mean
## lidar_ab_raster  17.86761
## lidar_dbh_raster 21.46871
```

Raster data

Plot

```
plot(lidar_raster, axes= FALSE, col = rainbow(25))
```



Raster data

`stars` package

```
install.packages('stars')
```

Raster data

`stars` package

```
library(stars)
```

```
## Loading required package: abind
```

```
lidar_raster_stars <- read_stars('lidar_raster.tif')
lidar_raster_stars
```

```
## stars object with 3 dimensions and 1 attribute
## attribute(s):
##                   Min. 1st Qu.  Median     Mean  3rd Qu.   Max.   NA's
## lidar_raster.tif  0.84    15.8 19.8065 19.66816 23.28197 86.988 560210
## dimension(s):
##      from  to  offset delta                       refsys point                                values
## x       1 680  256000   400 +proj=utm +zone=31 +ellps... FALSE                                  NULL
## y       1 660 4752000  -400 +proj=utm +zone=31 +ellps... FALSE                                  NULL
## band    1   2      NA    NA                           NA    NA lidar_ab_raster , lidar_dbh_raster
##      x/y
## x    [x]
## y    [y]
## band
```

Raster data

Bands and attributes

```
lidar_raster <- split(lidar_raster_stars)
```

Raster data

Bands and attributes

```
merge(lidar_raster)
```

```
## stars object with 3 dimensions and 1 attribute
## attribute(s):
##    Min. 1st Qu.  Median     Mean  3rd Qu.   Max.   NA's
## X  0.84    15.8 19.8065 19.66816 23.28197 86.988 560210
## dimension(s):
##            from  to  offset delta                       refsys point
## x             1 680  256000   400 +proj=utm +zone=31 +ellps... FALSE
## y             1 660 4752000  -400 +proj=utm +zone=31 +ellps... FALSE
## attributes    1   2      NA    NA                           NA    NA
##                                       values x/y
## x                                       NULL [x]
## y                                       NULL [y]
## attributes lidar_ab_raster , lidar_dbh_raster
```

Raster data

Write stars objects

```
write_stars(lidar_raster_stars, 'lidar_raster.nc')
read_stars('lidar_raster.nc')
```

```
## Band1, Band2,

## stars object with 2 dimensions and 2 attributes
## attribute(s):
##         Min.  1st Qu.   Median     Mean  3rd Qu.   Max.   NA's
## Band1  0.840 11.12243 16.66869 17.86761 22.80815 86.988 280105
## Band2  9.562 18.82708 20.89650 21.46871 23.50256 40.000 280105
## dimension(s):
##   from  to  offset delta                       refsys point values x/y
## x    1 680  256000   400 +proj=utm +zone=31 +ellps...    NA   NULL [x]
## y    1 660 4752000  -400 +proj=utm +zone=31 +ellps...    NA   NULL [y]
```

Raster data

CRS transformations

```
st_crs(lidar_raster_stars)
```

```
## Coordinate Reference System:
##   User input: unknown
##   wkt:
## PROJCRS["unknown",
##     BASEGEOGCRS["unknown",
##         DATUM["Unknown_based_on_GRS80_ellipsoid",
##             ELLIPSOID["GRS 1980",6378137,298.257222101004,
##                 LENGTHUNIT["metre",1],
##                 ID["EPSG",7019]]],
##         PRIMEM["Greenwich",0,
##             ANGLEUNIT["degree",0.0174532925199433,
##                 ID["EPSG",9122]]]],
##     CONVERSION["Transverse Mercator",
##         METHOD["Transverse Mercator",
##             ID["EPSG",9807]],
##         PARAMETER["Latitude of natural origin",0,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8801]],
##         PARAMETER["Longitude of natural origin",3,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8802]],
##         PARAMETER["Scale factor at natural origin",0.9996,
##             SCALEUNIT["unity",1],
##             ID["EPSG",8805]],
##         PARAMETER["False easting",500000,
```

Raster data

CRS transformations

```
st_transform(lidar_raster_stars, crs = 3043)
```

```
## stars object with 3 dimensions and 1 attribute
## attribute(s):
##                   Min. 1st Qu.  Median     Mean  3rd Qu.   Max.   NA's
## lidar_raster.tif  0.84    15.8 19.8065 19.66816 23.28197 86.988 560210
## dimension(s):
##      from  to offset delta                       refsys point                              values x/y
## x       1 680     NA    NA ETRS89 / UTM zone 31N (N-E) FALSE        [680x660] 256200,...,527800 [x]
## y       1 660     NA    NA ETRS89 / UTM zone 31N (N-E) FALSE      [680x660] 4488200,...,4751800 [y]
## band    1   2     NA    NA                            NA    NA lidar_ab_raster , lidar_dbh_raster
## curvilinear grid
```

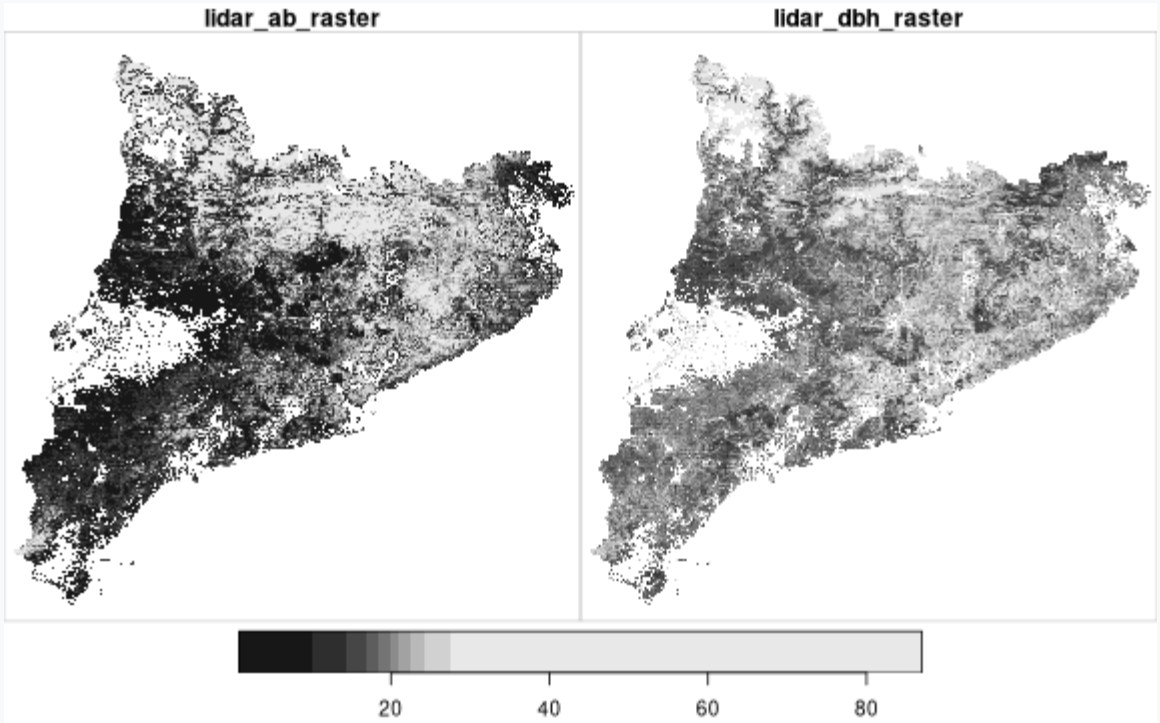
```
lidar_raster_stars <- st_warp(lidar_raster_stars, crs = 3043)
```

Raster data

Aggregate / Dissaggregate

```
reference_grid <- st_as_stars(st_bbox(lidar_raster_stars), dx = 200, dy = 200)
plot(st_warp(lidar_raster_stars, reference_grid))
```

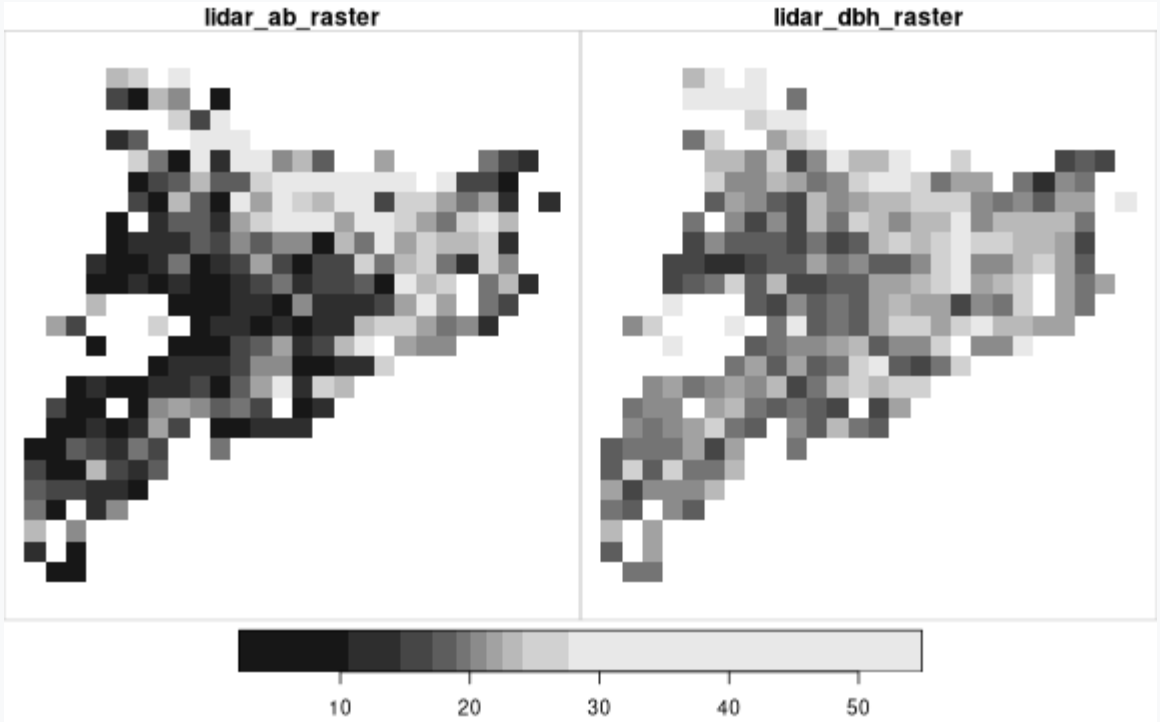
```
## downsample set to 3
```



Raster data

Aggregate / Dissaggregate

```
reference_grid <- st_as_stars(st_bbox(lidar_raster_stars), dx = 10000, dy = 10000)
plot(st_warp(lidar_raster_stars, reference_grid))
```



Raster data

Extract values from vector data

```
lidar_centroids <- st_centroid(lidar_data_sf)
```

```
## Warning in st_centroid.sf(lidar_data_sf): st_centroid assumes attributes are constant over
## geometries of x
```

```
st_extract(lidar_raster_stars, lidar_centroids)
```

```
## stars object with 2 dimensions and 1 attribute
## attribute(s):
##                       Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lidar_raster.tif  7.467778 16.31219 19.95963 19.57132 22.66794 34.05958   16
## dimension(s):
##      from to offset delta                       refsys point
## geom    1 42     NA    NA ETRS89 / UTM zone 31N (N-E)  TRUE
## band    1  2     NA    NA                           NA    NA
##                                                   values
## geom POINT (358422.6 4576554),...,POINT (423874.9 4627749)
## band                    lidar_ab_raster , lidar_dbh_raster
```

Raster data

Extract values from vector data

```
st_extract(lidar_raster_stars, lidar_centroids) %>%
  st_as_sf()
```

```
## Simple feature collection with 42 features and 2 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 279185.7 ymin: 4504748 xmax: 505392.1 ymax: 4732649
## Projected CRS: ETRS89 / UTM zone 31N (N-E)
## First 10 features:
##    lidar_ab_raster lidar_dbh_raster                     geom
## 1         17.45427         20.25319 POINT (358422.6 4576554)
## 2         20.16409         19.13410 POINT (496188.9 4683595)
## 3               NA               NA POINT (391065.8 4582623)
## 4         14.87556         20.12202 POINT (366720.8 4681336)
## 5         18.54923         29.85897 POINT (321143.6 4706110)
## 6         14.61220         26.47458 POINT (381909.9 4606116)
## 7         17.75270         23.13799 POINT (400743.5 4626667)
## 8         19.88594         25.16281 POINT (330797.4 4556041)
## 9         10.53367         20.75886   POINT (294404 4523912)
## 10        22.38917         34.05958 POINT (505392.1 4645040)
```

Raster data

Extract values from vector data

```
 aggregate(lidar_raster_stars, lidar_data_sf[1,], mean, na.rm = TRUE) %>%
   st_as_sf()
```

```
## Simple feature collection with 1 feature and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 336192.8 ymin: 4562380 xmax: 377678.6 ymax: 4593064
## Projected CRS: ETRS89 / UTM zone 31N (N-E)
##   lidar_ab_raster lidar_dbh_raster                              geom
## 1        15.24602         19.97279 MULTIPOLYGON (((374342.1 45...
```

Raster data

Crop

```
plot(st_crop(lidar_raster_stars, lidar_data_sf[1,]))
```

Raster data

Raster algebra

```
lidar_raster_stars*5
```

```
## stars object with 3 dimensions and 1 attribute
## attribute(s):
##                   Min. 1st Qu.  Median     Mean  3rd Qu.   Max.   NA's
## lidar_raster.tif   4.2      79 99.0325 98.34081 116.4098 434.94 560210
## dimension(s):
##      from  to  offset delta                       refsys point                                values
## x       1 680  256000   400 ETRS89 / UTM zone 31N (N-E)    NA                                  NULL
## y       1 660 4752000  -400 ETRS89 / UTM zone 31N (N-E)    NA                                  NULL
## band    1   2      NA    NA                           NA    NA lidar_ab_raster , lidar_dbh_raster
##      x/y
## x    [x]
## y    [y]
## band
```

Raster data

ggplot2

```
library(ggplot2)
ab_stars <- slice(lidar_raster_stars, band, 1)
ggplot() +
  geom_stars(data = ab_stars) +
  scale_fill_fermenter(palette = 2, direction = 1, type = 'seq', na.value = 'transparent')
```