Stands and Managment Areas maps

Vincent BISQUAY GRACIA

2021

I. Set up

Used pakages :

```
setwd("C:/Users/181248/Documents/181248/LANDIS II/R_calculus")
library(raster)
library(rgdal)
library(devtools)
library(tidyverse)
library(sf)
library(nngeo)
```

3 rasters are imported :

The raster layer Ecoregions *ER_OnlyCA* has the extent of the study zone (The Californian part of the Klamath mountains).

The two raster layers Managment Areas *Ownership_TotKlamath* and Stands *Stands_TotKlamath* have an extent superior to the study zone (the Oregon part of the Klamath mountain is also contained).

```
Ownership_TotKlamath = raster(
  "C:/Users/181248/Documents/181248/LANDIS II/R_calculus/Study_site/E4_ownership_BAU_v4.tif")
Stands_TotKlamath =  raster(
  "C:/Users/181248/Documents/181248/LANDIS II/R_calculus/Study_site/E4_stands_BAU_v5.tif")

ER_OnlyCA = raster(
  "C:/Users/181248/Documents/181248/LANDIS II/R_calculus/Study_site/ecoregions_v2.tif")
# this have the extent we want
```

II. Creating a shape file with the borders of the study area

Polygons are created with the Ecoregions raster. Then, all the polygons contained are merge in an unique polygon *Study site* that have the extent of the study area.

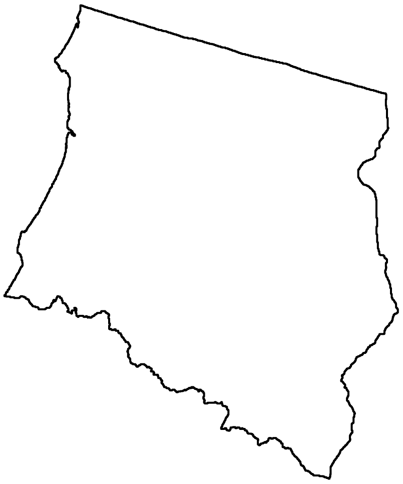
```
ER_OnlyCA[ER_OnlyCA == 0] <- NA # We don't want the background
ER_OnlyCA_poly = rasterToPolygons(ER_OnlyCA, fun=function(x){x > 0}, dissolve = T)
```

```
## Loading required namespace: rgeos
```

```
Study_site = st_union(ER_OnlyCA_poly %>% st_as_sf())
Study_site = st_remove_holes(Study_site %>% st_as_sf())

plot(Study_site, main = "Figure 1: Borders of the study area")
```

Figure 1: Borders of the study area



III. Extracting the part of the Ownership and Stand that is in the study area

III.a) All cells out of the study area are set to NA

In this fist step, all the cells of the Stand raster or the Ownership layer that are not in the Study Area are set to NA.

```
Ownership_OnlyCA_select = mask(Ownership_TotKlamath, Study_site)
Stands_OnlyCA_select = mask(Stands_TotKlamath, Study_site)

par(mfrow=c(2,2))
plot(Ownership_TotKlamath, main = "Figure 2.a: Ownership layer", xaxt="n", yaxt="n")
plot(Ownership_OnlyCA_select, main = "Figure 2.b: Ownership in study area", xaxt="n",
     yaxt="n")
plot(Stands_TotKlamath, main = "Figure 2.c: Stands layer", xaxt="n", yaxt="n")
plot(Stands_OnlyCA_select, main = "Figure 2.d: Stands in study area", xaxt="n", yaxt="n")
```

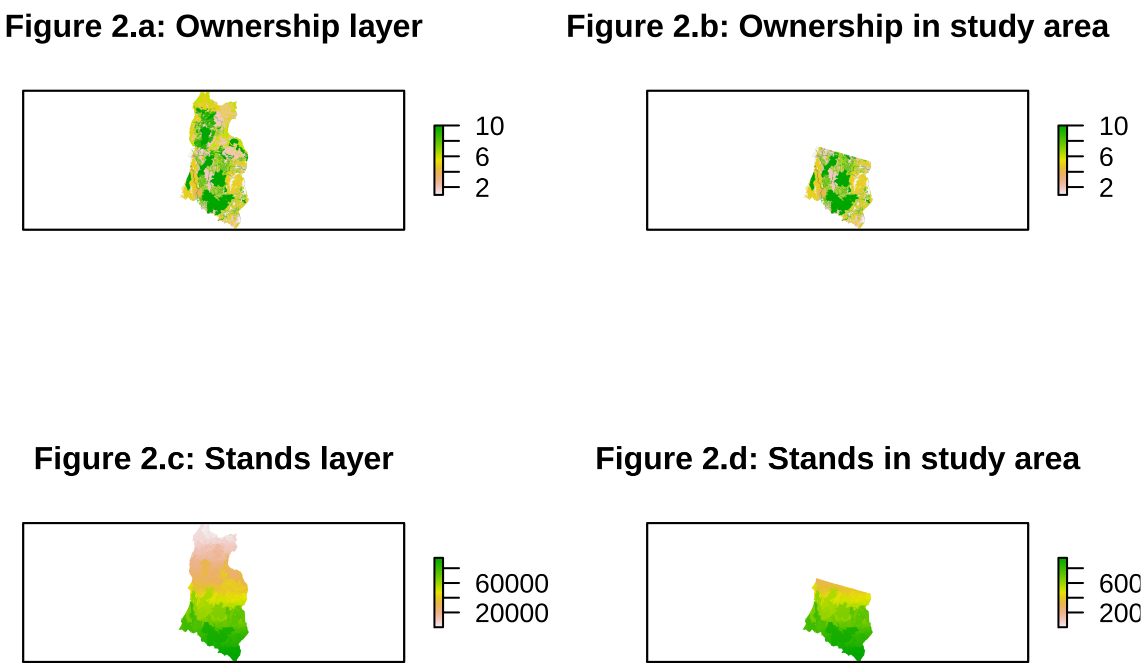


Figure 2.a: Ownership layer

Figure 2.b: Ownership in study area

Figure 2.c: Stands layer

Figure 2.d: Stands in study area

```
print(c(ER_OnlyCA@ncols, ER_OnlyCA@nrows))
```

```
## [1] 1084 1343
```

```
print(c(Ownership_OnlyCA_select@ncols, Ownership_OnlyCA_select@nrows))
```

```
## [1]  602 1197
```

```
print(c(Stands_OnlyCA_select@ncols, Stands_OnlyCA_select@nrows))
```

```
## [1]  602 1197
```

III.b) The rasters are crop to the study area extent

```
Ownership_OnlyCA_Crop = crop(Ownership_OnlyCA_select,ER_OnlyCA)
Stands_OnlyCA_Crop = crop(Stands_OnlyCA_select,ER_OnlyCA)

par(mfrow=c(2,2))
plot(Ownership_OnlyCA_select, main = "Figure 3.a: Ownership in study area", xaxt="n",
     yaxt="n")
plot(Ownership_OnlyCA_Crop, main = "Figure 3.b: Ownership croped", xaxt="n", yaxt="n")
plot(Stands_OnlyCA_select, main = "Figure 3.c: Stands in study area", xaxt="n", yaxt="n")
plot(Stands_OnlyCA_Crop, main = "Figure 3.d: Stands croped", xaxt="n", yaxt="n")
```

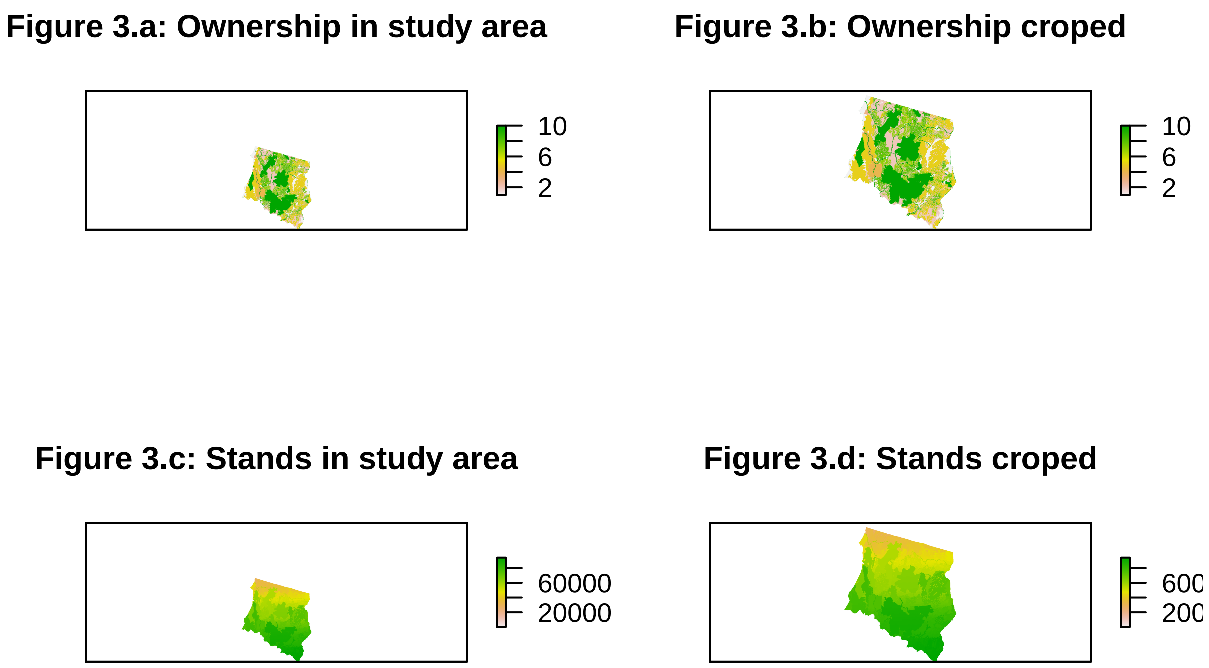


Figure 3.a: Ownership in study area
10
6
2
Figure 3.b: Ownership croped
10
6
2
Figure 3.c: Stands in study area
60000
20000
Figure 3.d: Stands croped
600
200

The plots seems identical at this sage.

```
print(c(ER_OnlyCA@ncols, ER_OnlyCA@nrows))
```

```
## [1] 1084 1343
```

```
print(c(Ownership_OnlyCA_Crop@ncols, Ownership_OnlyCA_Crop@nrows))
```

```
## [1] 602 746
```

```
print(c(Stands_OnlyCA_Crop@ncols, Stands_OnlyCA_Crop@nrows))
```

```
## [1] 602 746
```

However, the number of rows and columns of the obtained layers (602, 746) is different than the targeted one (1084, 1343)

III.c) A resampling is done to have the same resolution

```
Ownership_OnlyCA = resample(Ownership_OnlyCA_Crop, ER_OnlyCA, method="ngb")
Stands_OnlyCA = resample(Stands_OnlyCA_Crop, ER_OnlyCA, method="ngb")

par(mfrow=c(2,2))
plot(Ownership_OnlyCA_Crop, main = "Figure 4.a: Ownership croped", xaxt="n", yaxt="n")
plot(Ownership_OnlyCA, main = "Figure 4.b: Ownership resampled", xaxt="n", yaxt="n")
plot(Stands_OnlyCA_Crop, main = "Figure 4.c: Stands croped", xaxt="n", yaxt="n")
plot(Stands_OnlyCA, main = "Figure 4.d: Stands resampled", xaxt="n", yaxt="n")
```

Figure 4.a: Ownership croped

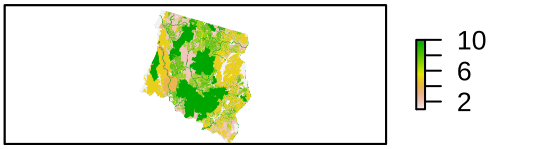


Figure 4.b: Ownership resampled

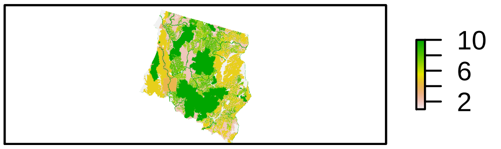


Figure 4.c: Stands croped

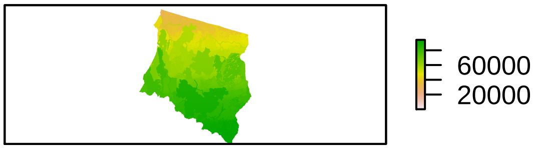
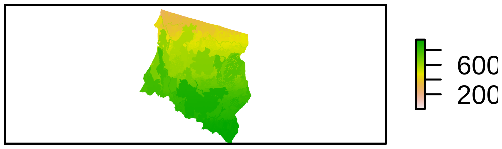


Figure 4.d: Stands resampled



```
print(c(ER_OnlyCA@ncols, ER_OnlyCA@nrows))
```

```
## [1] 1084 1343
```

```
print(c(Ownership_OnlyCA@ncols, Ownership_OnlyCA@nrows))
```

```
## [1] 1084 1343
```

```
print(c(Stands_OnlyCA@ncols, Stands_OnlyCA@nrows))
```

```
## [1] 1084 1343
```

This time the resolution is the same.

IV. Exporting the obtained layers

The *datatype* parameter is important, LANDIS-II works only with integers. For a raster as heavy as *Stands_OnlyCA*, the data type "INT4S" is the only possibility.

```
Ownership_OnlyCA[is.na(Ownership_OnlyCA[])] = 0
Stands_OnlyCA[is.na(Stands_OnlyCA[])] = 0

writeRaster(Ownership_OnlyCA,
            filename  = "C:/Users/181248/Documents/181248/LANDIS II/R_calculus/Study_site/Ownership_Onl
            format = "GTiff", datatype ="INT1U",overwrite=T)

writeRaster(Stands_OnlyCA,
            filename  = "C:/Users/181248/Documents/181248/LANDIS II/R_calculus/Study_site/Stands_OnlyCA
            format = "GTiff", datatype ="INT4S",overwrite=T)
```