Stands and Managment Areas maps

Vincent BISQUAY GRACIA

07/2021

I. Set up

Used pakages :

```
wd = "D:/Internship 3A/GitHub/Project-Klamath-2021/Model Parameterization"
setwd(wd)
library(sf)
library(raster)
library(rgdal)
library(devtools)
library(magrittr)
library(nngeo)
library(knitr)
```

Used palettes for the ploting :

```
pal.2 = colorRampPalette(c("purple","yellow"))
pal.2.op = colorRampPalette(c("yellow","purple"))
pal.11 = colorRampPalette(c("white", "grey", "purple","grey","tomato","green4",
                             "green","yellow","red","grey","grey"))
pal.13 = colorRampPalette(c("white", "grey", "purple","grey","tomato","green4",
                             "green","yellow","red","grey","grey", "cyan","blue"))
pal.R = colorRampPalette(c("white", "grey", "grey","grey","grey","grey","grey",
                            "grey","grey","grey","grey", "cyan","blue"))
```

3 rasters are imported :

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

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

```
ER_OnlyCA =
  paste0(wd,"/Study_site/ecoregions_v2.tif") %>% raster()

Ownership_TotKlamath =
  paste0(wd,"/Study_site/E4_ownership_BAU_v4.tif") %>% raster()

Stands_TotKlamath =
  paste0(wd,"/Study_site/E4_stands_BAU_v5.tif") %>% raster()
```

We want that the management area and stands rasters have the same extent and defintion than the ecoregion one. This will allow all rasters to have the same definition and allow LANDIS-II to compute the landscape properly.

II. Selection of the Ownership and Stand areas included in the study area

II.a) A shapefile with all the borders of the study area is created

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

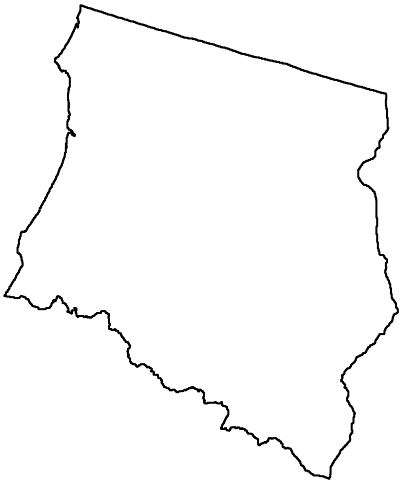
```
ER_OnlyCA[ER_OnlyCA == 0] <- NA # We don't want the background
ER_OnlyCA_poly = rasterToPolygons(ER_OnlyCA, fun=function(x){x > 0}, dissolve = T)
```

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

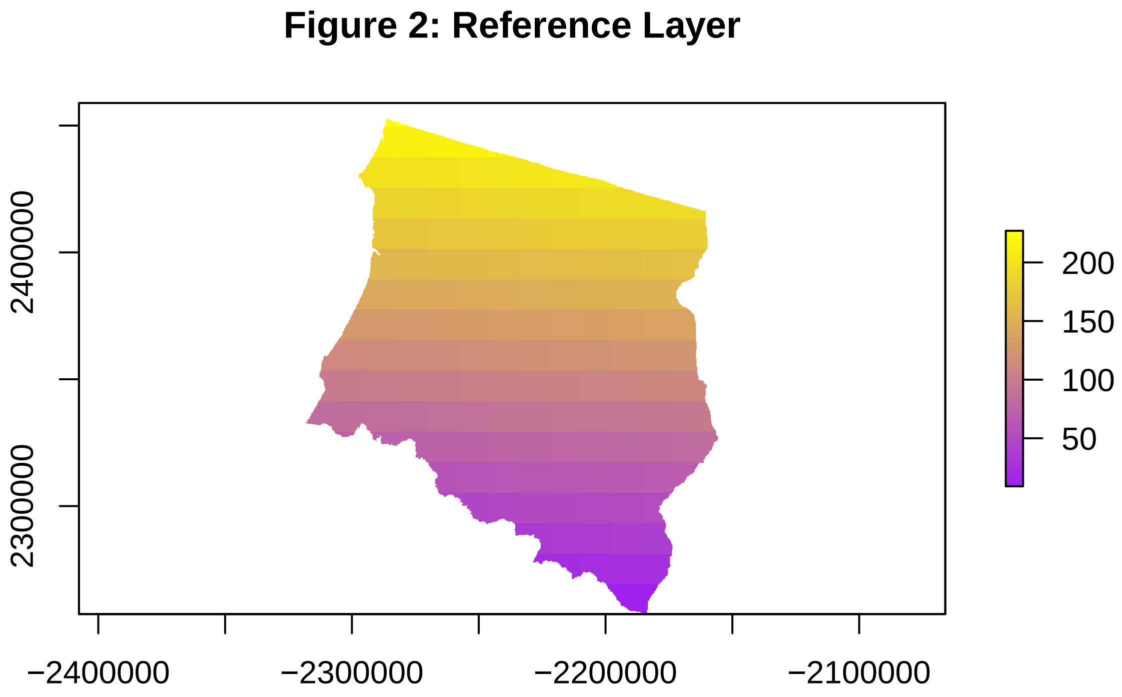
```
## Warning in rasterToPolygons(ER_OnlyCA, fun = function(x) {: package rgeos is not
## available. Cannot dissolve
```

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

par(mfrow = c(1,2))
plot(Study_site, main = "Figure 1: Borders of the study area")
```

Figure 1: Borders of the study area



```r
plot(ER_OnlyCA, main = "Figure 2: Reference Layer",
     col = pal.2(250))
```



II.b) All cells out of the study area are set to NA

In this fist step, all the cells of the stand raster and the management area raster 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 3.a: Ownership layer",
     xaxt="n", yaxt="n", col = pal.2(10))
plot(Ownership_OnlyCA_select, main = "Figure 3.b: Ownership in study area",
     xaxt="n", yaxt="n", col = pal.2(10))
plot(Stands_TotKlamath, main = "Figure 3.c: Stands layer",
     xaxt="n", yaxt="n", col = pal.2(500))
plot(Stands_OnlyCA_select, main = "Figure 3.d: Stands in study area",
     xaxt="n", yaxt="n", col = pal.2(500))
```

Figure 3.a: Ownership layer

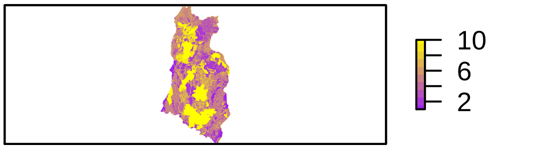


Figure 3.b: Ownership in study area

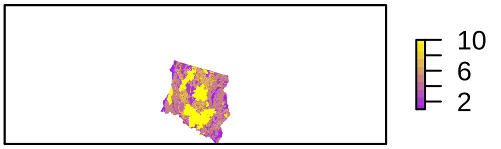


Figure 3.c: Stands layer

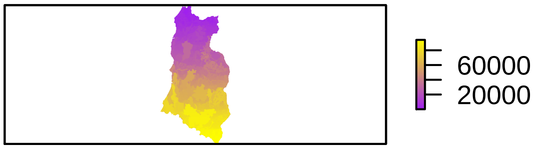
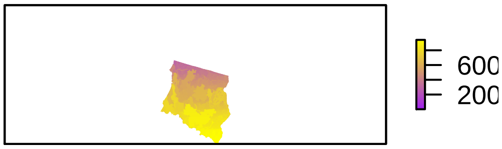


Figure 3.d: Stands in study area



After this operation, the two layers we are modifying present a large number of empty cells that does not exist on the reference layer. Thus, other transformations are needed.

II.c) The rasters are croped 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 4.a: Ownership in study area",
     xaxt="n", yaxt="n", col = pal.2(10))
plot(Ownership_OnlyCA_Crop, main = "Figure 4.b: Ownership croped",
     xaxt="n", yaxt="n", col = pal.2(10))
plot(Stands_OnlyCA_select, main = "Figure 4.c: Stands in study area",
     xaxt="n", yaxt="n", col = pal.2(500))
plot(Stands_OnlyCA_Crop, main = "Figure 4.d: Stands croped",
     xaxt="n", yaxt="n", col = pal.2(500))
```

Figure 4.a: Ownership in study area

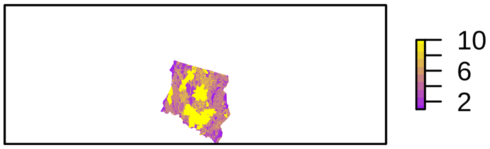


Figure 4.b: Ownership croped

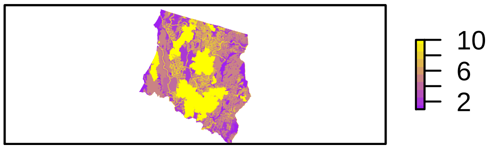


Figure 4.c: Stands in study area

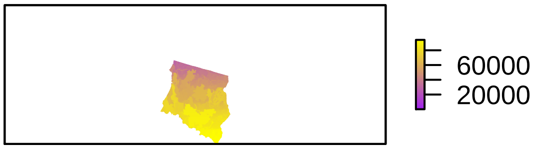
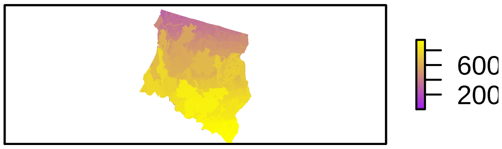


Figure 4.d: Stands croped



The plots seems identical to the ecoregion one at this sage.

```
a = c(ER_OnlyCA@ncols, ER_OnlyCA@nrows)
b = c(Ownership_OnlyCA_Crop@ncols, Ownership_OnlyCA_Crop@nrows)
c = c(Stands_OnlyCA_Crop@ncols, Stands_OnlyCA_Crop@nrows)

df = rbind(a,b,c) %>% as.data.frame(row.names = c("Ecoregion", "Management area", "Stands"))
colnames(df) = c("X","Y")

kable(df, caption = "Number of cells by layer after croping")
```

Table 1: Number of cells by layer after croping

	X	Y
Ecoregion	1084	1343
Management area	602	746
Stands	602	746

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

II.d) 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 5.a: Ownership croped",
     xaxt="n", yaxt="n", col = pal.2(10))
plot(Ownership_OnlyCA, main = "Figure 5.b: Ownership resampled",
     xaxt="n", yaxt="n", col = pal.2(10))
plot(Stands_OnlyCA_Crop, main = "Figure 5.c: Stands croped",
     xaxt="n", yaxt="n", col = pal.2(500))
plot(Stands_OnlyCA, main = "Figure 5.d: Stands resampled",
     xaxt="n", yaxt="n", col = pal.2(500))
```

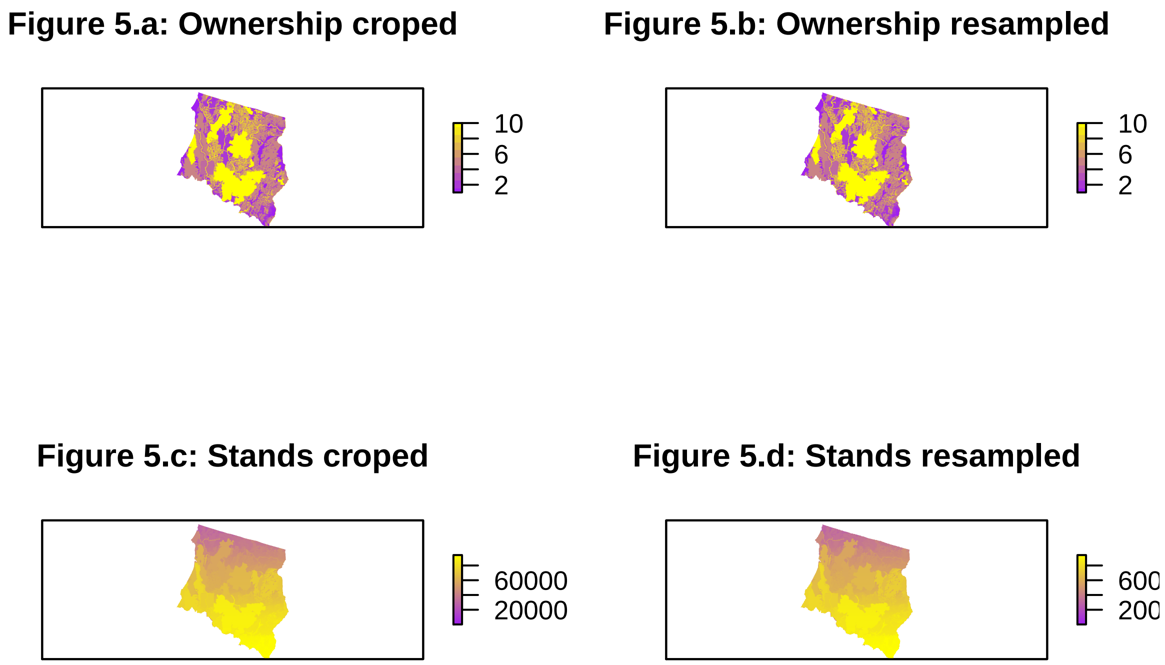


Figure 5.a: Ownership croped

Figure 5.b: Ownership resampled

Figure 5.c: Stands croped

Figure 5.d: Stands resampled

```
a = c(ER_OnlyCA@ncols, ER_OnlyCA@nrows)
b = c(Ownership_OnlyCA@ncols, Ownership_OnlyCA@nrows)
c = c(Stands_OnlyCA@ncols, Stands_OnlyCA@nrows)

df = rbind(a,b,c) %>% as.data.frame(row.names = c("Ecoregion", "Management area", "Stands"))
colnames(df) = c("X","Y")

kable(df, caption = "Number of cells by layer after resampling")
```

Table 2: Number of cells by layer after resampling

	X	Y
Ecoregion	1084	1343
Management area	1084	1343
Stands	1084	1343

This time the resolution is the same. Thus, the layers can be stacked and used in LANDIS-II.

II.e) The obtained layers are exported

The *datatype* parameter is important, LANDIS-II works only with integers for these two layers. 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  = paste0(wd,"/Study_site/Ownership_OnlyCA.tif"),
            format = "GTiff", datatype ="INT1U",overwrite=T)

writeRaster(Stands_OnlyCA,
            filename  = paste0(wd,"/Study_site/Stands_OnlyCA.tif"),
            format = "GTiff", datatype ="INT4S",overwrite=T)
```

III. Creation of the Resistence Management Areas

The Pro-active management scenario have a resistence prescription. This consists on identifying climate refugias and apply a productive management with species traditionaly used on such places.

As these refugias concentrate an important part of the economical insights, a buffer zone is created with intensive fire and insects prescriptions.

Some of the rasters used in this part have not a defined system of projection. The one of the management area is used as a reference.

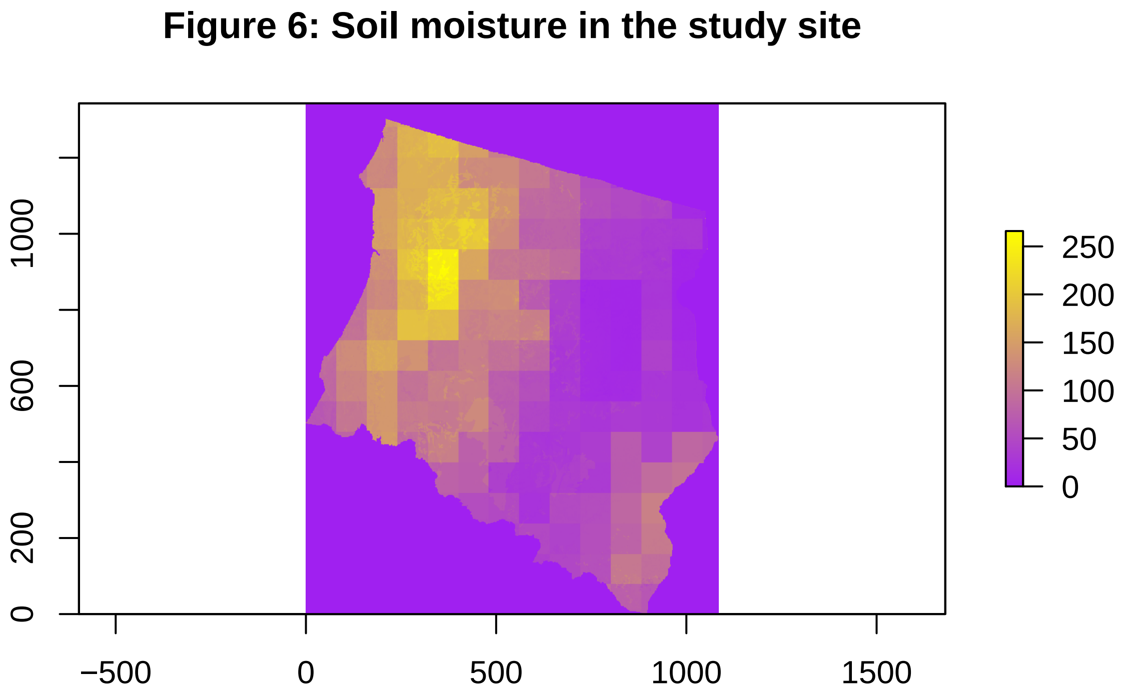
```
CRSRef = crs(Ownership_OnlyCA)
NrowRef = Ownership_OnlyCA@nrows
NcolRef = Ownership_OnlyCA@ncols
ResRef = res(Ownership_OnlyCA)[1]
XminRef = Ownership_OnlyCA@extent@xmin
XmaxRef = Ownership_OnlyCA@extent@xmax
YminRef = Ownership_OnlyCA@extent@ymin
YmaxRef = Ownership_OnlyCA@extent@ymax
```

III.a) Identification of the climate refugias

The first step is to identify the climate change refugias, i.e. zones where the effect of the climate change on the ecosystem (and particularly tree health) is limited comared to the wole landscape.

The proxy used to identify these zones are the soil moisture. This choice is justified and discussed in the report. Among the input files used in LANDIS-II, there is no raster of the soil moisture. The layer used here is the soil moisture modelled on year 1 by the NECN extention, in the scenario Business As Usual.

```
SoilMoist = paste0(wd, "/Study_site/Annual-water-budget-1.img") %>% raster()

plot(SoilMoist, main = "Figure 6: Soil moisture in the study site",  col = pal.2(300))
```



The quatile 0.95, i.e. the 5% of the cells with the highest value of soil moisture, is selected (the cells take the value 11, while the others are eraised).

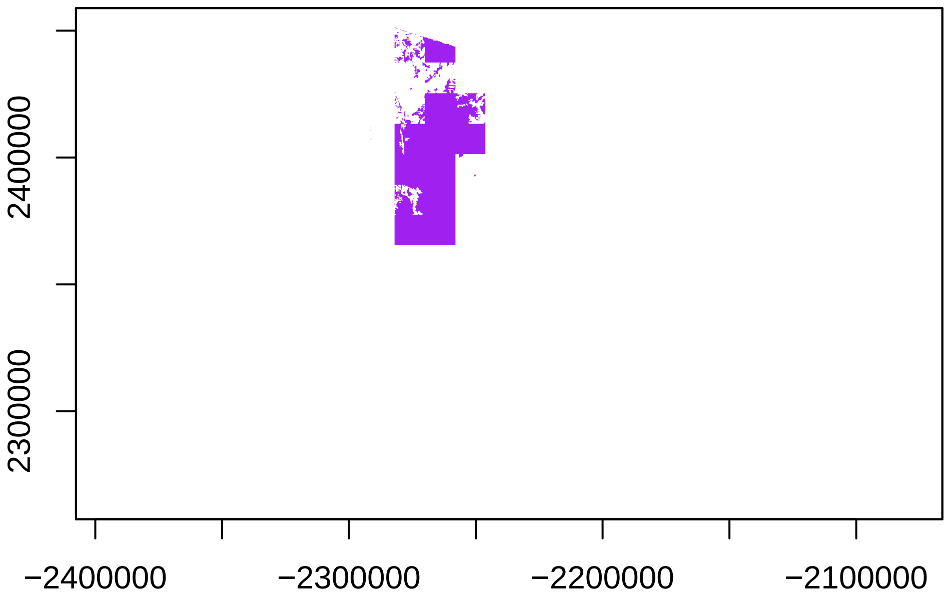
```
Q = quantile(SoilMoist, probs = c(0.05, 0.95))

Q3 = as.numeric(Q[2])

values(SoilMoist)[values(SoilMoist) < Q3] = NA

values(SoilMoist)[values(SoilMoist) > 0] = 11
```

A raster is created with this quantile using the data of the reference raster.

```
SoilMoist = raster(vals=getValues(SoilMoist),
                   nrow=NrowRef,ncol=NcolRef,resolution=ResRef,
                   xmn=XminRef, xmx=XmaxRef,ymn =YminRef,ymx=YmaxRef,crs=CRSRef)
```

```
plot(SoilMoist, main = "Figure 7: Zones with high soil moisture",
     legend = F, col = pal.2(1))
```

Figure 7: Zones with high soil moisture



```
freq(SoilMoist, merge=T) %>% as.data.frame() %>% kable(caption = "Number of cells in the potential refug
```

Table 3: Number of cells in the potential refugias raster by value

value	count
11	75833
NA	1379979

This zone of high soil moisture is considered as the potentials climate refugias and will be used for the resistence prescription.

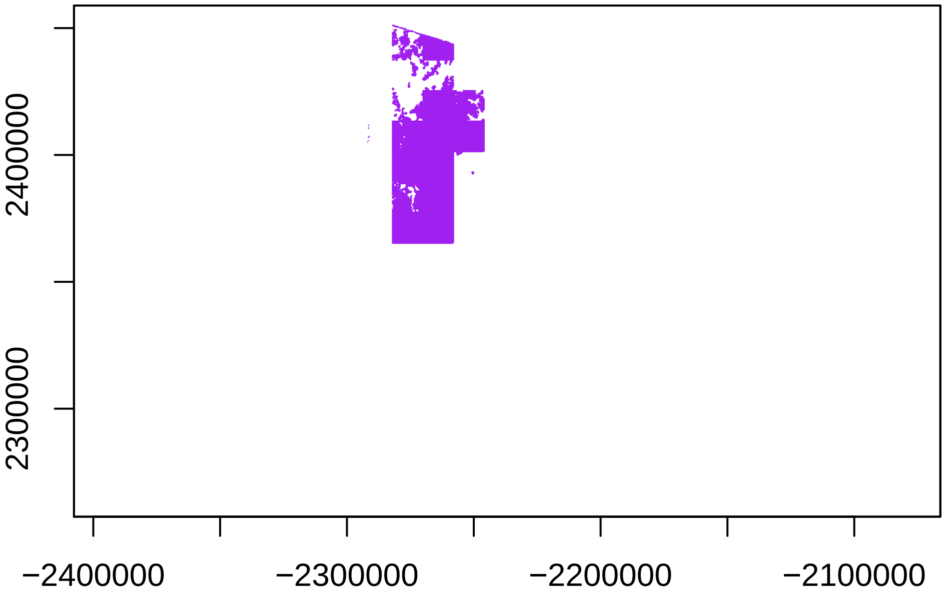
III.b) Creation of the buffer zone

A buffer of 300 m, corresponding to a width of 2 cells, is created. Similarly to before, all cells in it are set to 12 and the others eraised.

```
Buf = buffer(SoilMoist, width=300)

values(Buf)[values(Buf) == 1] = 12
```

```
Buf = raster(vals=getValues(Buf),
                    nrow=NrowRef,ncol=NcolRef,resolution=ResRef,
                    xmn=XminRef, xmx=XmaxRef,ymn =YminRef,ymx=YmaxRef,crs=CRSRef)

plot(Buf, main = "Figure 8: Buffer zone around the resistence management area",
     legend = F, col = pal.2(1))
```

Figure 8: Buffer zone around the resistence management area



```
freq(Buf, merge=T) %>% as.data.frame() %>% kable(caption = "Number of cells in the buffer raster by valu
```

Table 4: Number of cells in the buffer raster by value

value	count
12	86105
NA	1369707

III.c) Merging the management areas in one layer

The potential climate refugia raster and the buffer zone raster are merged.

```
Resistence = cover(SoilMoist, Buf)
```

```
plot(Resistence, main = "Figure 9: Buffer zone and resistence area ",
     legend = F, col = pal.2.op(2))
```

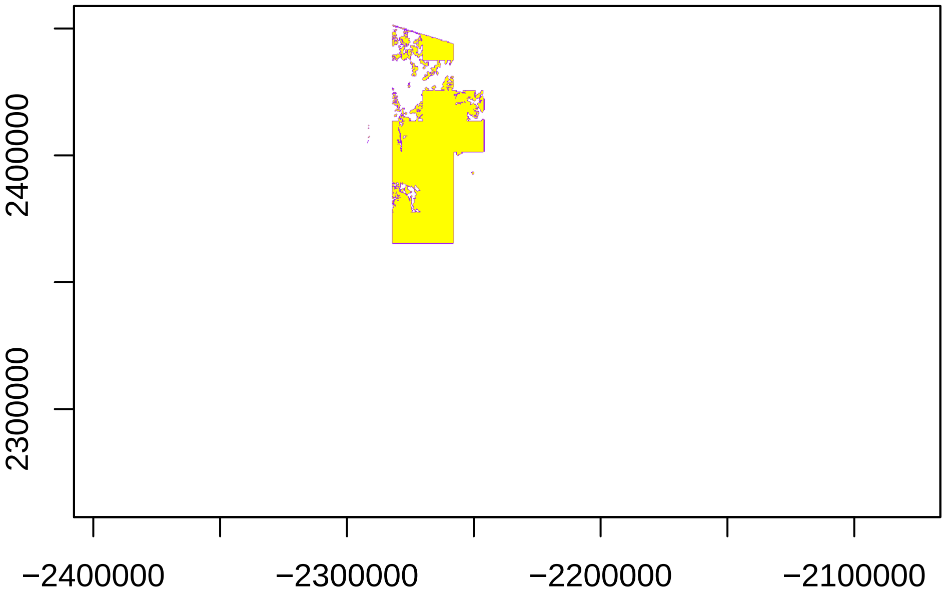


Figure 9: Buffer zone and resistence area

After that, the two new management areas are added to the others.

```
New_Ma = cover(Resistence, Ownership_OnlyCA)
```

III.d) Export of the completed management area and stand rasters

III.d.1) Export the new stands raster

To obtain new codes for the cells situated on the resistence and buffer zones, a sum is performed.

This way, the cells in question have new code that is different depending of if it is in the resistent area or the buffer zone.

This new layer is then exported.

```
SoilMoist[is.na(SoilMoist)] = 0

Buf[is.na(Buf)] = 0

StandRefugia = Stands_OnlyCA + SoilMoist + Buf

writeRaster(StandRefugia,
filename =  paste0(wd,"/Study_site/Stand_CA_Refugias.tif"),
format = "GTiff", datatype ="INT4S",overwrite=T)
```

III.d.2) Dealing with stands on several management areas

The change on the stands operated before result on many stands being on different management areas. As this makes little sence on a practical point of view and is forbidden on LANDIS-II, the limits of the management areas are modified to make sure one stand is included in only one management area.

In practice, there is no stands that where on more than two management areas, so the following script only deal with stands on two management areas, one of them being the buffer zone or the resistence area.

```
names(New_Ma) = "New_Ma"
names(StandRefugia) = "StandRefugia"

# The management area raster and the stands one are stacked
# to better see where they overlap :
Stack = stack(New_Ma,StandRefugia)
Stack_df = as.data.frame(Stack)

Val = unique(Stack_df[ , 1:2 ])

Ndup = sum(duplicated(Val[,2]))

Dupl = subset(Val, duplicated(Val[,2])) # list of the ID of the duplicated stands
# (that are on 2 management areas)

for(i in c(1:Ndup)){
  DupStand = Dupl[i,2]
  DupMA = Stack$New_Ma[Stack$StandRefugia==DupStand] %>% unique()

  # If the 1st management area where the duplicate stand is
  # is the buffer zone or the resistence area,
  # then this is the management area kept :

  if(DupMA[1] %in% c(11,12)) {
    Select_Ma = DupMA[1]
    No_Ma = DupMA[2]
  # if not, the 2nd is kept :
  }else{
    Select_Ma = DupMA[2]
    No_Ma = DupMA[1]}

  # All the cells of the management area raster that are included in the duplicaded stand
  # are given the value of the keps management area.
  Stack$New_Ma[Stack$StandRefugia==DupStand &Stack$New_Ma==No_Ma] = Select_Ma
}
```

This way, the management areas are change to make all the problematic stands placed in the buffer zone or in the resistence area. The reverse operation, change the stands to fit them on a given management area could have been done.

III.2.3) Export the new management area layer

Finally, this layer is exported and will be used on the Pro-active scenario along with the corresponding stand layer.

```
MA_Refugia = Stack$New_Ma

writeRaster(MA_Refugia,
filename =  paste0(wd,"/Study_site/Ownership_CA_Refugias.tif"),
format = "GTiff", datatype ="INT1U",overwrite=T)
```

VI. Plot of the Management areas

A summary of the different raster layers of management areas is given here.

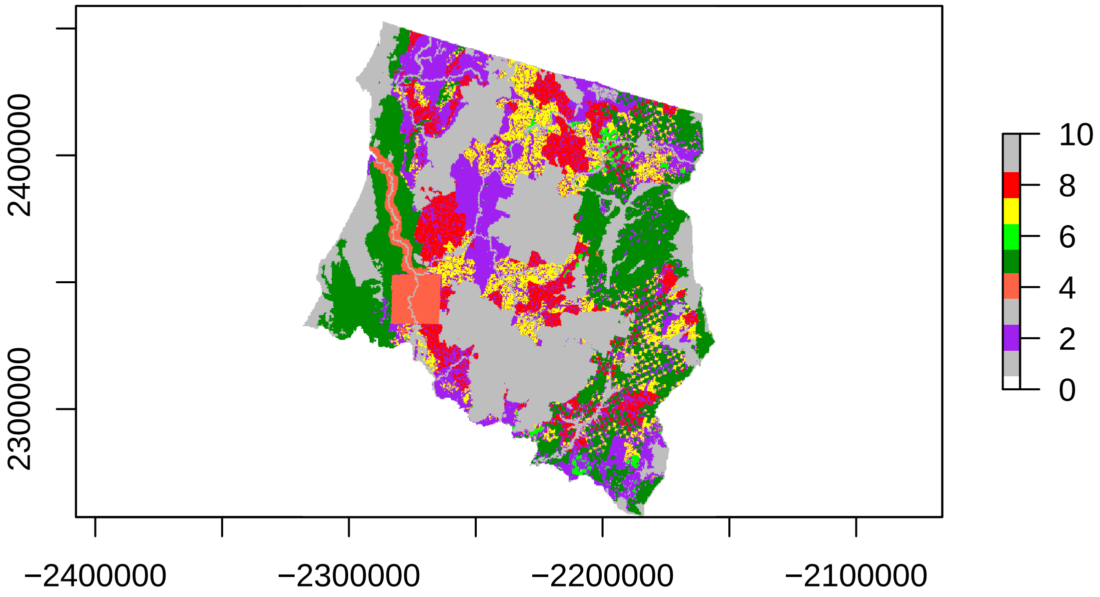
```
data.frame("Map_code" =c("1","2","3","4","5","6","7","8","9","10"),
           "Management_area" =c("Unmanaged","Federal","Unmanaged","Tribal Area","PIF", "PNIF",
                                "Matrix timberlands", "AMA","Unmanaged","Unmanaged")) %>%
  kable(caption = "Management areas map code")
```

Table 5: Management areas map code

Map_code	Management_area
1	Unmanaged
2	Federal
3	Unmanaged
4	Tribal Area
5	PIF
6	PNIF
7	Matrix timberlands
8	AMA
9	Unmanaged
10	Unmanaged

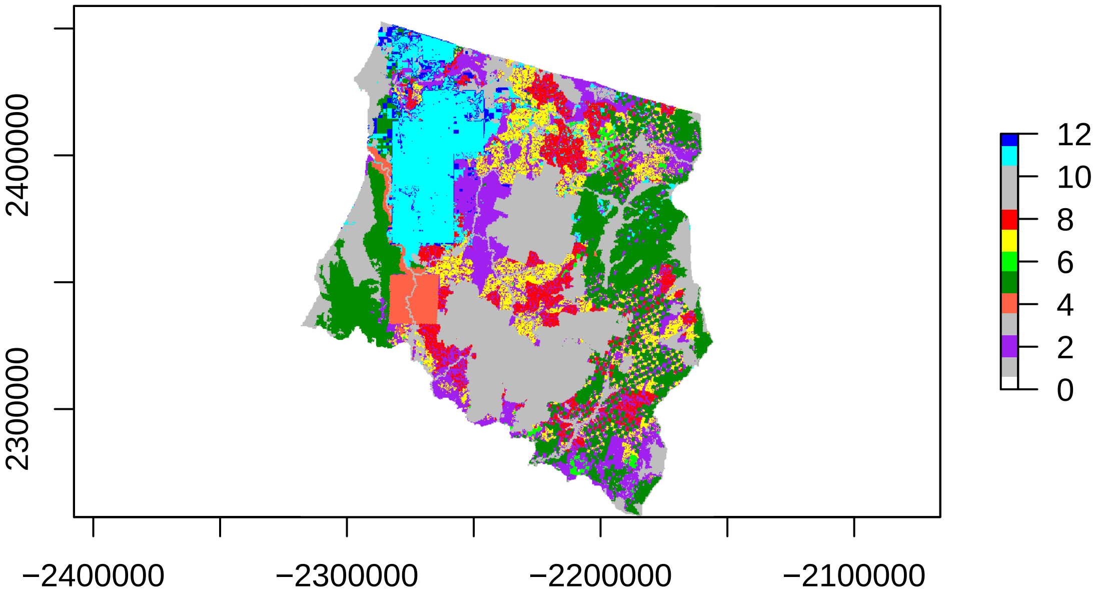
```
plot(Ownership_OnlyCA, main="Figure 10: Managment areas (BAU and Adaptability)",
     col=pal.11(11))
```

Figure 10: Managment areas (BAU and Adaptability)



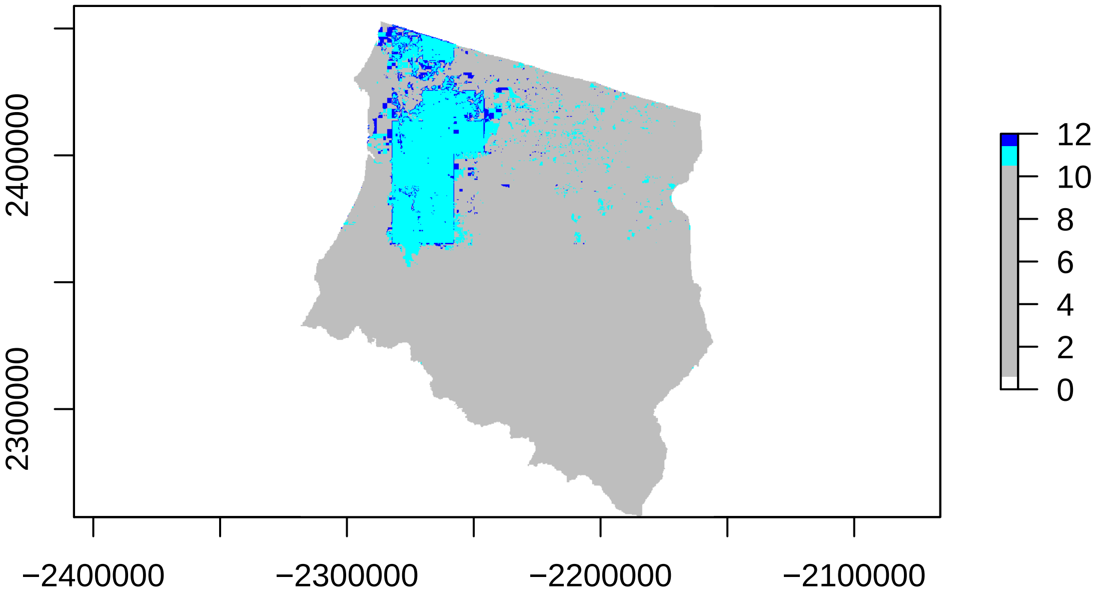
```r
plot(MA_Refugia, main="Figure 11: Managment areas (Pro-Active)",
     col=pal.13(13))
```

Figure 11: Managment areas (Pro−Active)



```
plot(MA_Refugia, main="Figure 12: Resistence area and buffer zone",
     col=pal.R(13))
```

Figure 12: Resistence area and buffer zone



2400000
2300000
−2400000
−2300000
−2200000
−2100000
12
10
8
6
4
2
0