

## Core area analysis

Load the packages.

```
library(rgdal)
library(rgeos)
library(raster)
library(rmapshaper)
```

### Set the file to work with

The path to the working directory will not need to be set as long as the file is in the directory below the shapefile directory containing the results from vectorisation.

The code contains lines for each of the files. To change the file you are working with comment out the rest by adding a hash # to the start of the line. Remove the hash from the one you want. Make sure that only one line is uncommented.

```
file_name<-"forest2012"
#file_name<-"forest2000"
#file_name<-"nonforest2000"
#file_name<-"nonforest2012"
```

### Set the buffer width

Change this for each buffer width you want to run. If it is set so high that there are no core areas at all then the code will fail on line 49.

```
buffer_width = 30
```

### Load the data

```
frags<-readOGR("shapefiles",file_name)
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "shapefiles", layer: "forest2012"
## with 10 features
## It has 7 fields
```

```
str(frags@bbox)
```

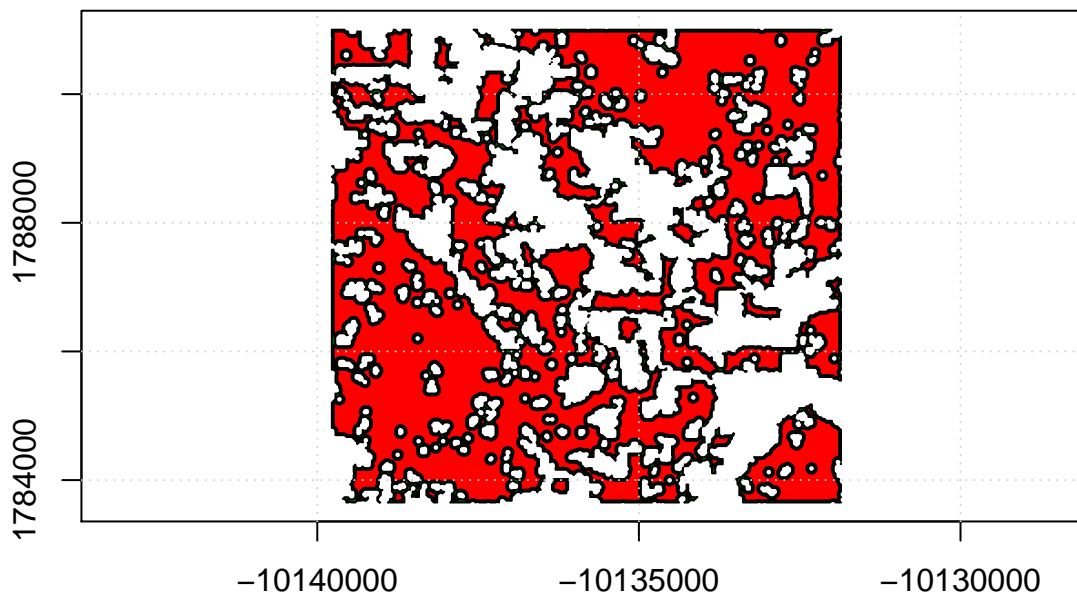
```
##  num [1:2, 1:2] -10139775 1783651 -10131857 1791003
##  - attr(*, "dimnames")=List of 2
##    ..$ : chr [1:2] "x" "y"
##    ..$ : chr [1:2] "min" "max"
```

```
totalarea<-(frags@bbox[1,2]-frags@bbox[1,1])*(frags@bbox[2,2]-frags@bbox[2,1])/10000
totalarea
```

```
## [1] 5821.194
```

```
cores<-gBuffer(frags,width=-buffer_width,byid=TRUE)

plot(frags,col="darkgreen")
plot(cores,col="red",add=T)
axis(1)
axis(2)
box()
grid()
```



Percent of total area that is core

```
100*(gArea(cores)/10000)/totalarea
```

```
## [1] 41.57532
```

```
cores$c_area<-gArea(cores,byid=T)/10000
cores$corepercent<-(cores$c_area/cores$area)*100
cores$edgepercent<-100-cores$corepercent
cores$ncores <- unlist(lapply(lapply( cores@polygons , slot , "Polygons" ),length))

frags@data<-merge(frags@data,cores@data[, -c(2:4)],by="id",all.x=TRUE)
frags@data[is.na(frags@data)]<-0
frags@data[, -1]<-round(frags@data[, -1],2)
d<-cores@data

d<-d[order(d$area,decreasing=T),]
head(d)
```

```
##      id      area      perims      shape      cumarea      ptot      parea
## 1    3 2772.03610 237559.631 12.728237 2823.00682 47.2529986 85.5699711
## 5 134 176.53946 22712.727 4.822180 3068.70585 3.0093471 5.4495958
## 8 242 146.50643 10738.166 2.502631 3234.51066 2.4973947 4.5225065
## 0    1  50.97072  5367.914 2.120998  50.97072 0.8688629 1.5734149
## 3    75  47.96188  8280.206 3.372781 2881.57128 0.8175733 1.4805350
## 2     6  10.60258  2442.716 2.116227 2833.60940 0.1807350 0.3272911
##      c_area corepercent edgepercent ncores
## 1 2104.342928  75.91326  24.08674  108
## 5  114.385345  64.79308  35.20692    9
## 8  115.305748  78.70354  21.29646    6
## 0   35.671667  69.98463  30.01537    2
## 3   27.569197  57.48148  42.51852    5
## 2    4.729938  44.61118  55.38882    3
```

```
out_file<-paste("results",file_name,"buffer",buffer_width,".csv",sep="_")
out_file
```

```
## [1] "results_forest2012_buffer_30_.csv"
```

```
write.csv(d,out_file)
```

```
proj4string(cores)<-CRS("+init=epsg:3857")

#cores<-ms_explode(cores, force_FC = TRUE)
shape_file<-paste(file_name,"cores",buffer_width,sep="_")
writeOGR(cores,dsn="shapefiles",shape_file,driver="ESRI Shapefile",over=TRUE)

proj4string(frag)<-CRS("+init=epsg:3857")
shape_file<-paste(file_name,"edge_analysis",buffer_width,sep="_")
writeOGR(frag,dsn="shapefiles",shape_file,driver="ESRI Shapefile",over=TRUE)
```