ZOO955 - Overlays

Hilary Dugan

2/22/2018

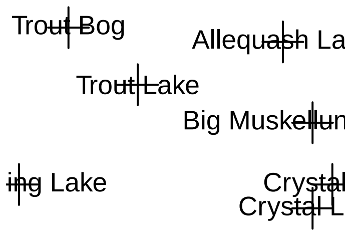
Vector and Raster Data

It is common that you will have to use both vector and raster data in your work. Today we'll explore some common techniques for working with multiple data types.

Lake point data

Read in lake point data from .csv

```
library(sp)
library(raster)
library(rgdal)
library(rgeos)
lakes = read.csv('Data/nlakes.csv',stringsAsFactors = F)
coordinates(lakes) = ~Longitude+Latitude
crs(lakes) <- CRS("+proj=longlat +datum=WGS84")
plot(lakes,cex=2)
text(lakes,labels = lakes$Lake,cex=0.8)
```



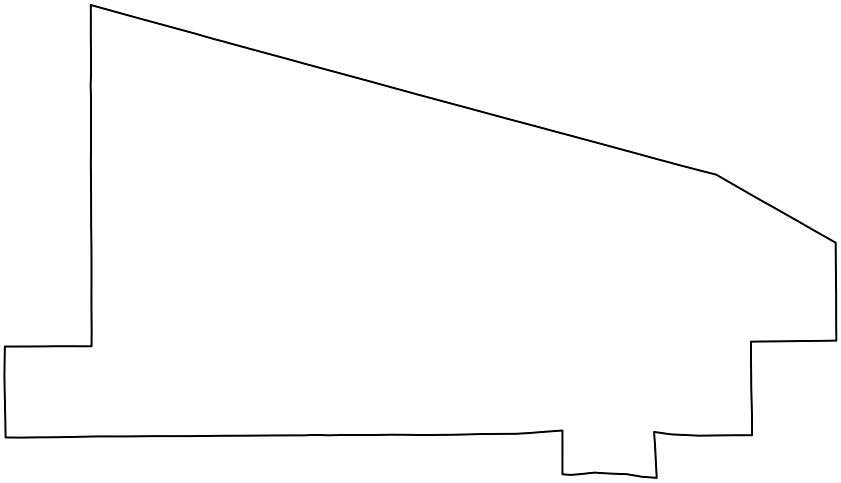
Counties data

Read in lake point data from .csv

```
counties = readOGR('C:/Users/hdugan/Dropbox/RandomR/GIS_Wisconsin/County_Boundaries_24K/County_Boundari
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:/Users/hdugan/Dropbox/RandomR/GIS_Wisconsin/County_Boundaries_24K/County_Boundaries_24K.s
## with 72 features
## It has 7 fields
## Integer64 fields read as strings:  OBJECTID DNR_CNTY_C
```

```
vilas = counties[counties@data$COUNTY_NAM == 'Vilas',]
plot(vilas)
```



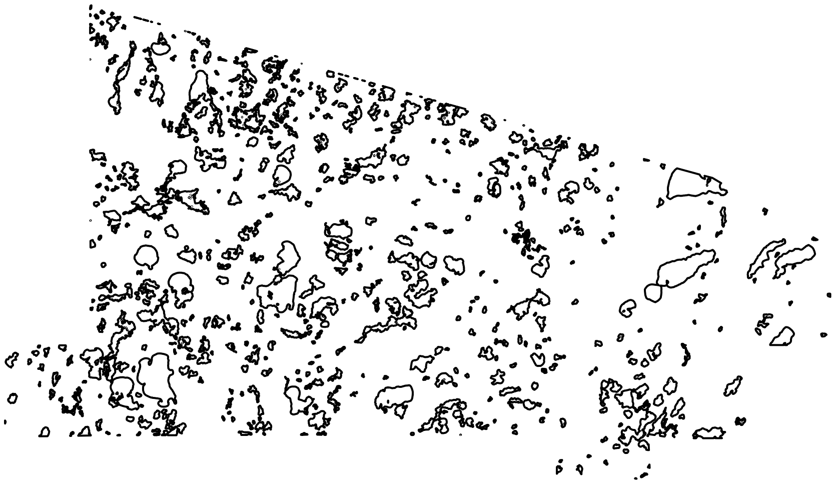
National hydrography shapefiles

Read in lake point data from .csv

```
library(rgdal)
vilasNHD = readOGR('Data/vilasNHD.shp',layer = 'vilasNHD')
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "Data/vilasNHD.shp", layer: "vilasNHD"
## with 616 features
## It has 13 fields
```

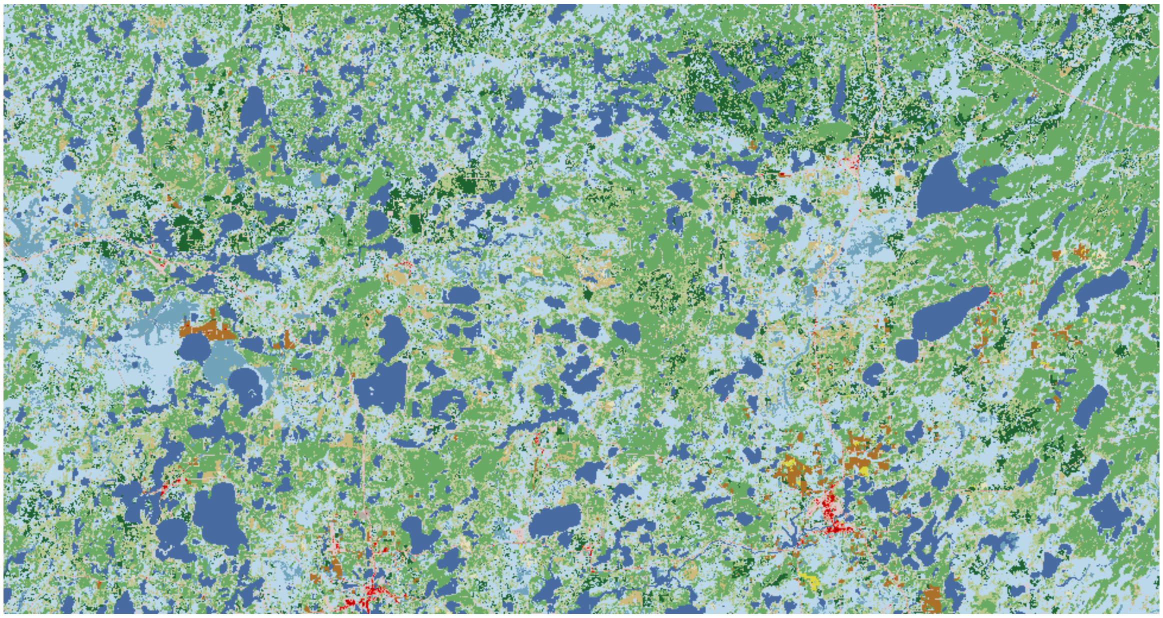
```
plot(vilasNHD)
```



NLCD raster data

Read in lake point data from .csv

```
nlcd = raster('Data/vilasNLCD.grd')
plot(nlcd)
```



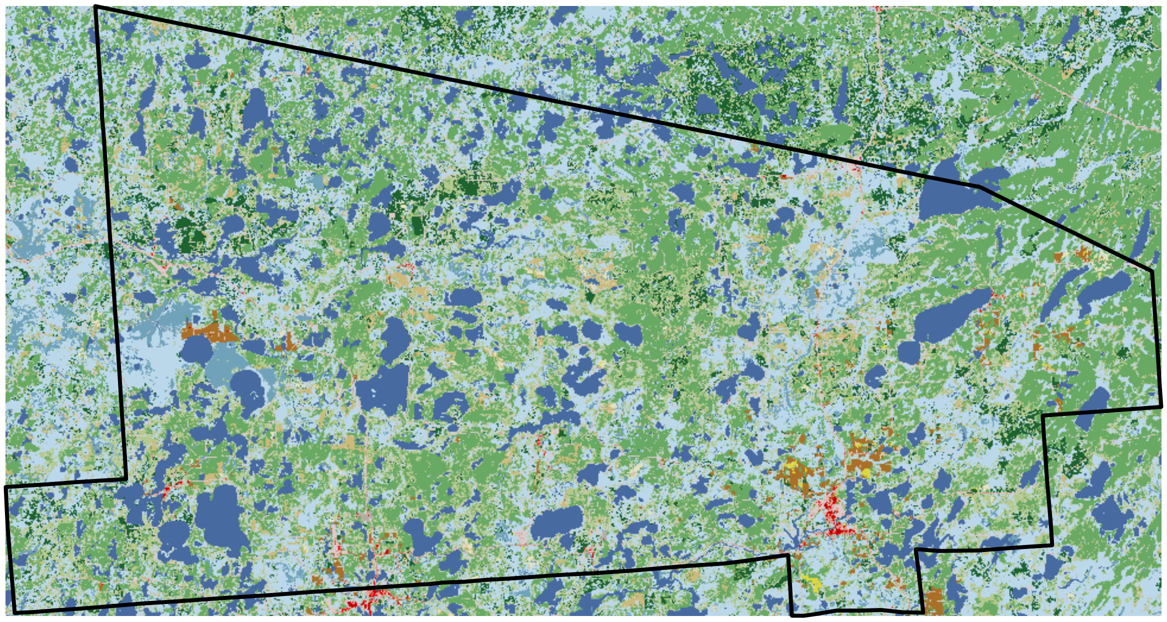
Cropping rasters

You can only crop rasters to be square. So the above raster is already cropped to vilas county. However, if we wanted to be sure:

```
nlcdvilas = crop(nlcd,vilas)
```

```
## Error in .local(x, y, ...): extents do not overlap
```

```
# Need to make sure they have identical CRS!
vilas = spTransform(vilas,crs(nlcd)) # transform CRS (usually easier to transform the vector data)
nlcdvilas = crop(nlcd,vilas) # crop the raster the extent of the county
plot(nlcdvilas)
plot(vilas,add = T,lwd=2)
```



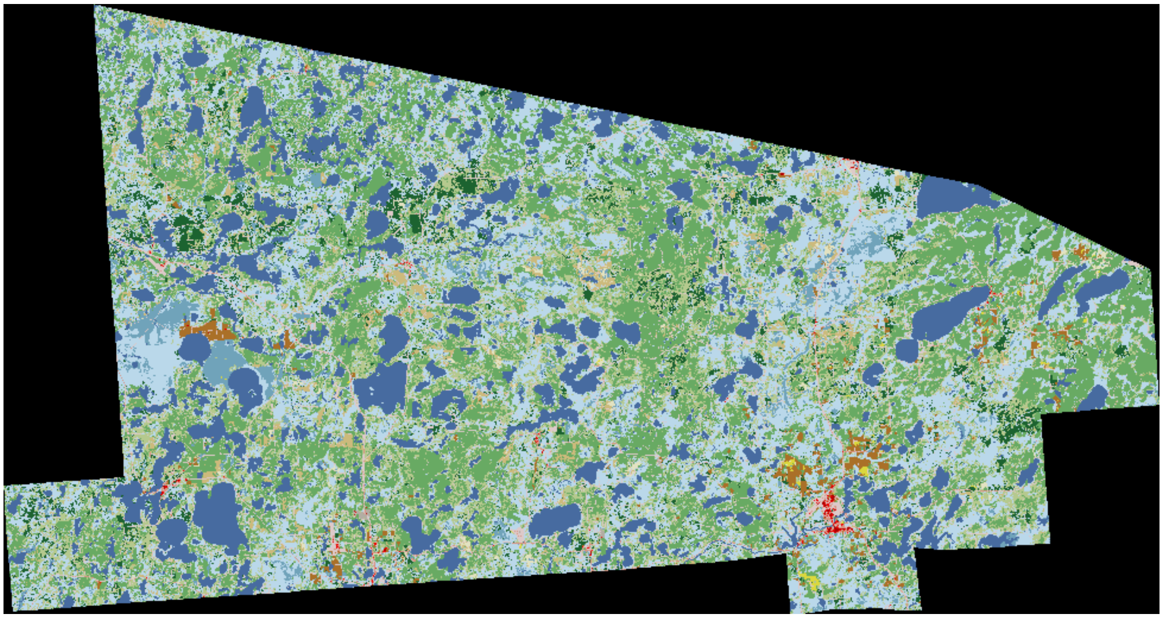
Mask raster

now use the mask function

Instead we can `mask` the raster.

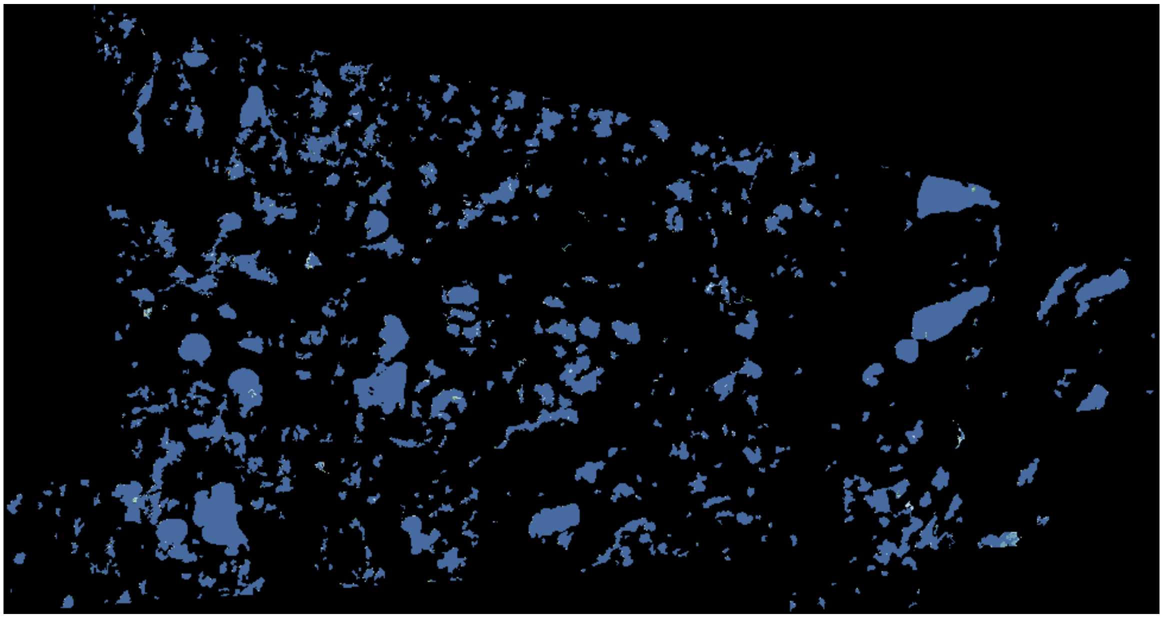
```
Create a new Raster* object that has the same values as x, except for the cells that are
NA (or other maskvalue) in a 'mask'.
```

```
#now use the mask function
nlcdvilas <- mask(nlcd, vilas)
plot(nlcdvilas)
```



Check if NHD matches up with NLCD

```
vilas.crs = spTransform(vilasNHD,crs(nlcd))
nlcd.nhd = mask(nlcd,vilas.crs)
plot(nlcd.nhd)
```



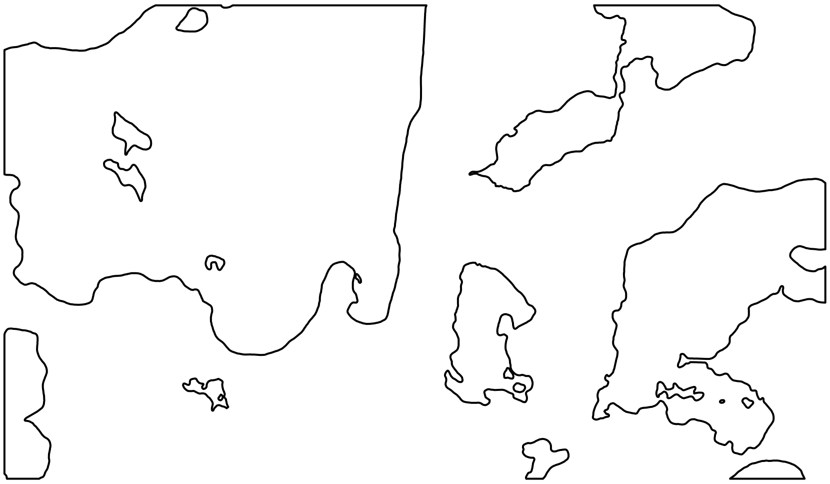
Looks like we're dealing with an island issue

Extract northern study lakes

```
identicalCRS(lakes,vilasNHD)
```

```
## [1] FALSE
```

```
lakes.crs = spTransform(lakes,crs(vilasNHD))
lakesNHD = crop(vilasNHD,lakes.crs)
plot(lakesNHD)
```



Well that didn't work. This just cropped to the extent.

Reserve `crop` for raster data sets. Instead we can use `over`.

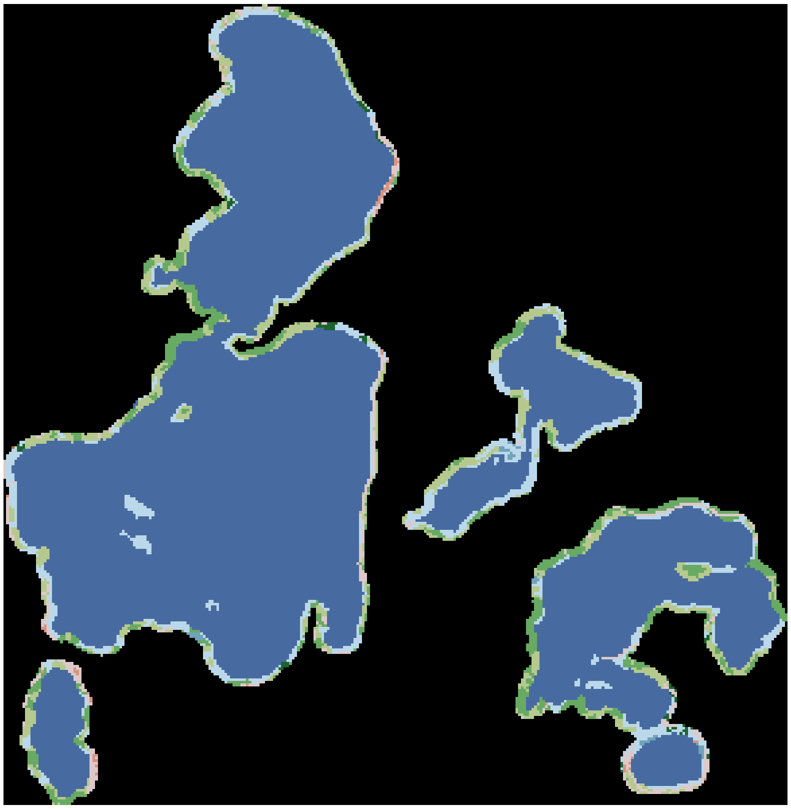
```
b = over(vilasNHD,lakes.crs)
nLakes = vilasNHD[which(!is.na(b)),]
plot(nLakes)
```

Buffers around study lakes

```
nLakes.m = spTransform(nLakes,crs(nlcd))
buff500 = gBuffer(nLakes.m,width = 100)
plot(buff500)
plot(nLakes.m,add=T,col='lightblue4')
```

Extract nlcd data

```
a = crop(nlcd,buff500)
a = mask(a,buff500)
plot(a)
```



Can ask questions like:

- What is the percent wetlands around these five lakes?

```
afreq = freq(a)
afreq
```

```
##       value count
##  [1,]    11 24870
##  [2,]    21   608
##  [3,]    22    60
##  [4,]    41  1355
##  [5,]    42    96
##  [6,]    43  1856
##  [7,]    52     1
##  [8,]    82     1
##  [9,]    90  1981
## [10,]    95    57
## [11,]    NA 55542
```

```
perWetlands = 100 * sum(afreq[9:10,2])/sum(afreq[1:10,2])
perWetlands
```

```
## [1] 6.598672
```

Convert polygons to raster

Things to make note of:

- CRS - Units and area are important!
- Grid size

```
# Albers Equal Area Conic
AEA <- CRS("+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=37.5 +lon_0=-96
                                 +x_0=0 +y_0=0 +datum=NAD83 +units=m +no_defs")
# Transform projection of NHD
nhdAEA = spTransform(vilasNHD,AEA)


# Length and width of area in km
width = (extent(nhdAEA)[4] - extent(nhdAEA)[3])/1000
length = (extent(nhdAEA)[2] - extent(nhdAEA)[1])/1000

# Create an empty raster
r1 = raster(ncol=width*10, nrow=length*10) #approximately 100 m grid
r1
```

```
## class       : RasterLayer
## dimensions  : 854, 451, 385154  (nrow, ncol, ncell)
## resolution  : 0.7982262, 0.2107728  (x, y)
## extent      : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
```

```
extent(r1) = extent(nhdAEA)
r1
```

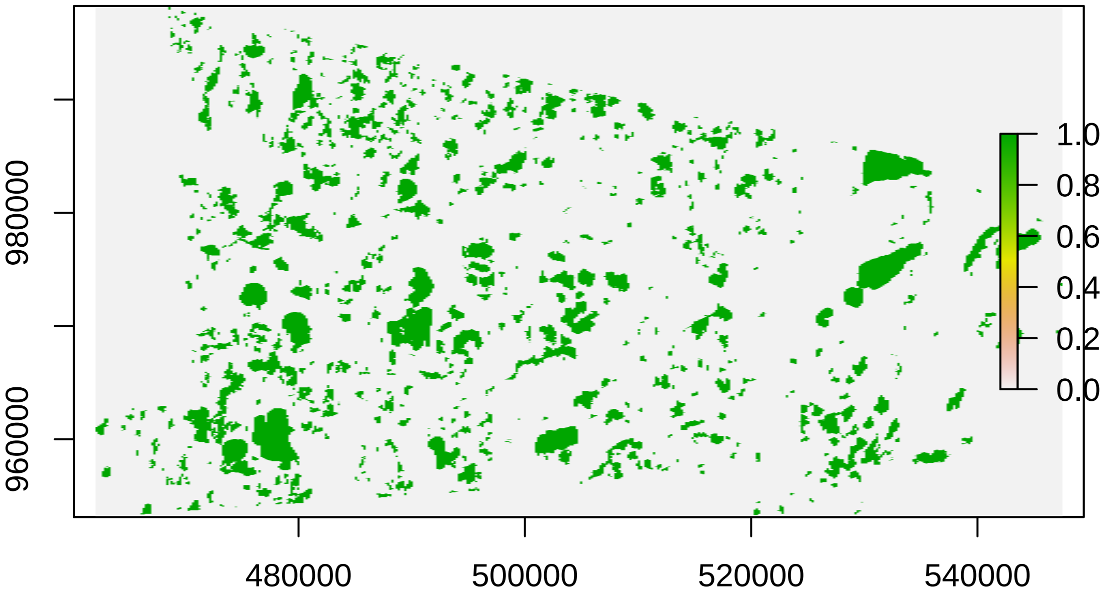
```
## class       : RasterLayer
## dimensions  : 854, 451, 385154  (nrow, ncol, ncell)
## resolution  : 189.4169, 52.8522  (x, y)
## extent      : 462061.9, 547488.9, 953127.5, 998263.3  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
```

We now have an empty raster that has the desired geometry

```
r2 = rasterize(nhdAEA,r1,field = 1, background = 0)
r2
```

```
## class       : RasterLayer
## dimensions  : 854, 451, 385154  (nrow, ncol, ncell)
## resolution  : 189.4169, 52.8522  (x, y)
## extent      : 462061.9, 547488.9, 953127.5, 998263.3  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=37.5 +lon_0=-96 +x_0=0 +y_0=0 +datum=NAD83 +u
## data source : in memory
## names       : layer
## values      : 0, 1  (min, max)
```

```
plot(r2)
```

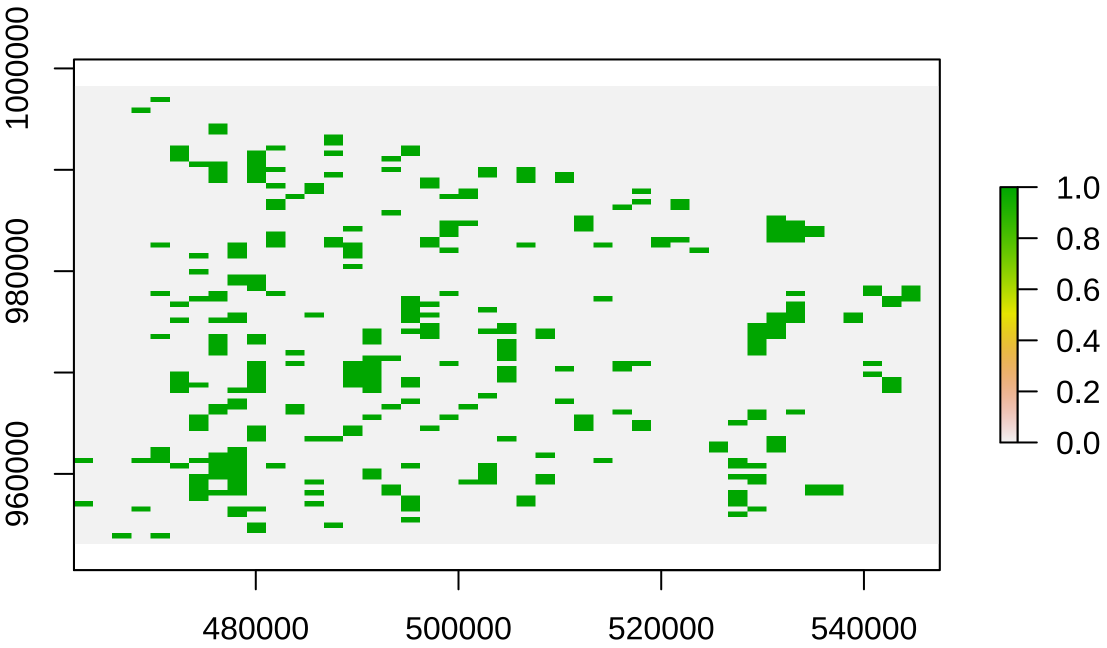


```
# Reclassify raster where all zero values are NA.
class1 <- reclassify(r2, cbind(0, NA))
class1
```

```
## class       : RasterLayer
## dimensions  : 854, 451, 385154  (nrow, ncol, ncell)
## resolution  : 189.4169, 52.8522  (x, y)
## extent      : 462061.9, 547488.9, 953127.5, 998263.3  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=37.5 +lon_0=-96 +x_0=0 +y_0=0 +datum=NAD83 +u
## data source : in memory
## names       : layer
## values      : 1, 1  (min, max)
```

What would it look like with a 1km grid cell?

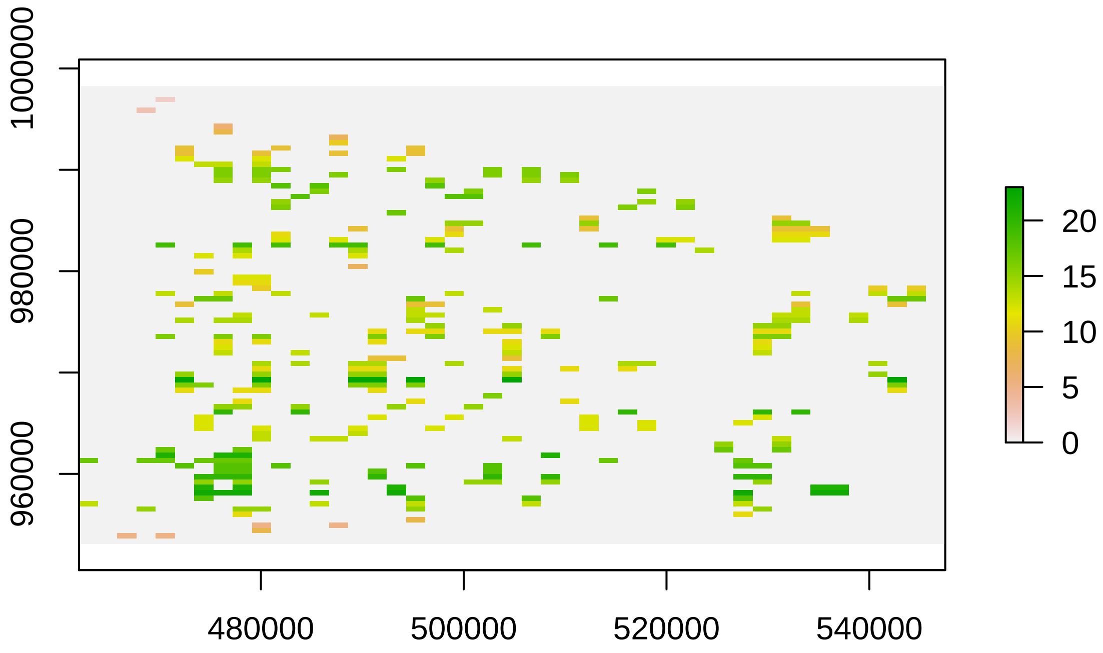
```
r1k = raster(ncol=width, nrow=length) #approximately 1 km grid
extent(r1k) = extent(nhdAEA)
r2k = rasterize(nhdAEA,r1k,field = 1, background = 0)
plot(r2k)
```



Can change the function of how cells are counted.

```
To determine what values to assign to cells that are covered by multiple spatial features.
You can use functions such as min, max, or mean, or one of the following character values:
'first', 'last', 'count'. The default value is 'last'.
```

```
r2kc = rasterize(nhdAEA,r1k,field = 1,background = 0, fun=function(x,...){length(x)})
plot(r2kc)
```



Note:

`You can also pass multiple functions using a statement like fun=function(x, ...) c(length(x),mean(x)), in which case the returned object is a RasterBrick (multiple layers).`

Homework

NHD data for Wisconsin: http://prd-tnm.s3-website-us-west-2.amazonaws.com/?prefix=StagedProducts/Hydrography/NHD/State/HighResolution/Shape/

1) Calculate the percent water in each county in Wisconsin using the 2011 NLCD.
2) Calculate the percent water in each county using the NHD.