Data Preprocessing and Raster Manipulation

Mark Baker

Importing Key Libraries

```
library(raster)
```

```
## Loading required package: sp
```

```
library(sp)
library(sf)
```

```
## Linking to GEOS 3.7.2, GDAL 2.4.2, PROJ 5.2.0
```

```
library(rgdal)
```

```
## rgdal: version: 1.4-8, (SVN revision 845)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.4.2, released 2019/06/28
##  Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/3.6/Resources/library/rgdal/gda
##  GDAL binary built with GEOS: FALSE
##  Loaded PROJ.4 runtime: Rel. 5.2.0, September 15th, 2018, [PJ_VERSION: 520]
##  Path to PROJ.4 shared files: /Library/Frameworks/R.framework/Versions/3.6/Resources/library/rgdal/p
##  Linking to sp version: 1.3-2
```

```
library(leaflet)
library(ggplot2)
library(rasterVis)
```

```
## Loading required package: lattice
```

```
## Loading required package: latticeExtra
```

```
##
## Attaching package: 'latticeExtra'
```

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

Loading in Raster Necessary Data

```
cdl2019 <- raster("/Users/markbaker/Downloads/GEOG28602/final/2019_30m_cdls/2019_IL.tif")
cdl2019
```

```
## class      : RasterLayer
## dimensions : 16994, 12330, 209536020  (nrow, ncol, ncell)
## resolution : 0.0003258965, 0.0003259018  (x, y)
```

```
## extent     : -91.51352, -87.49521, 36.96997, 42.50835  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## source     : /Users/markbaker/Downloads/GEOG28602/final/2019_30m_cdls/2019_IL.tif
## names      : X2019_IL
## values     : 0, 255  (min, max)
```

```
cdl2018 <- raster("/Users/markbaker/Downloads/GEOG28602/final/2018_30m_cdls/2018_IL.tif")
cdl2018
```

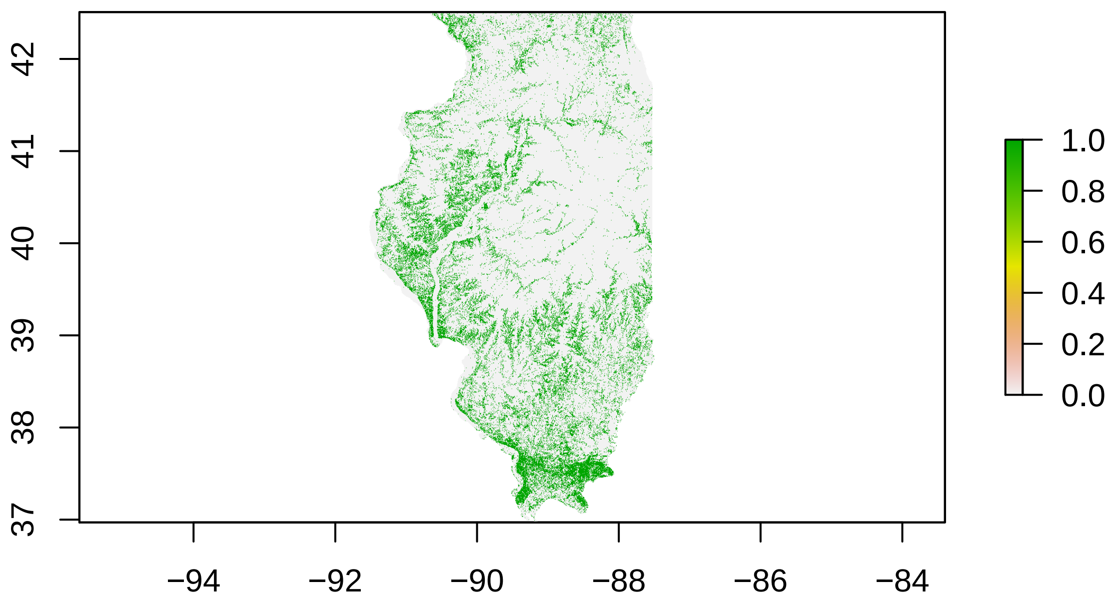
```
## class      : RasterLayer
## dimensions : 16994, 12330, 209536020  (nrow, ncol, ncell)
## resolution : 0.0003258965, 0.0003259018  (x, y)
## extent     : -91.51352, -87.49521, 36.96997, 42.50835  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## source     : /Users/markbaker/Downloads/GEOG28602/final/2018_30m_cdls/2018_IL.tif
## names      : X2018_IL
## values     : 0, 255  (min, max)
```

Raster Manipulation

In this selection, I will work to generate the key output file that will utlized in the creation of the Shiny Application

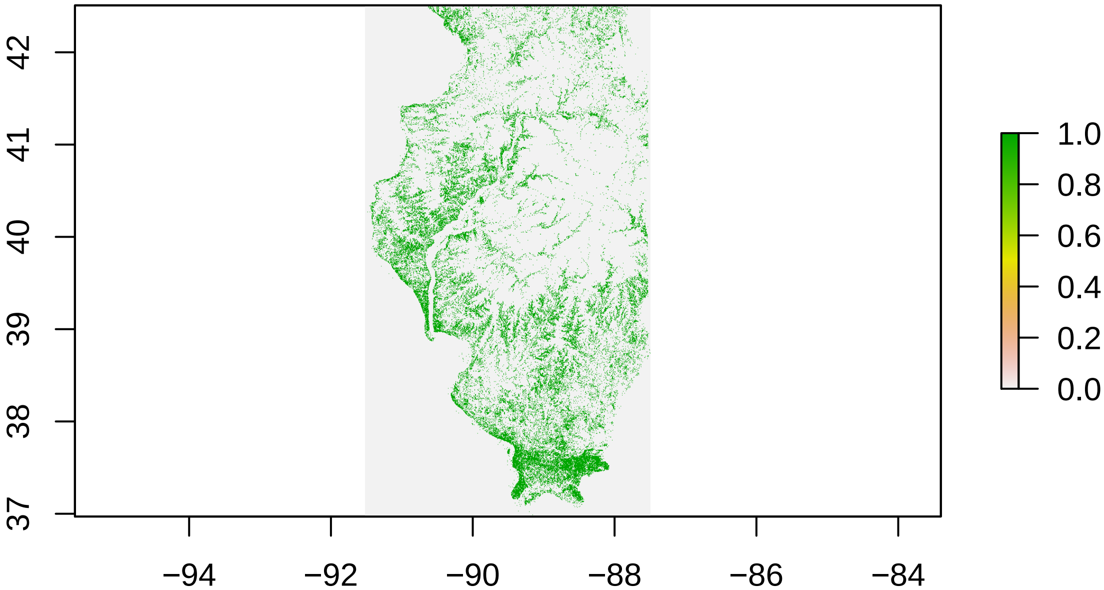
Selecting Forestry Cover

```
d2018 <- cdl2018 == 141 #Deciduous
e2018 <- cdl2018 == 142 #Evergreen
m2018 <- cdl2018 == 143 #Mixed
forest2018 <- d2018 + e2018 + m2018 #Total
plot(forest2018)
```



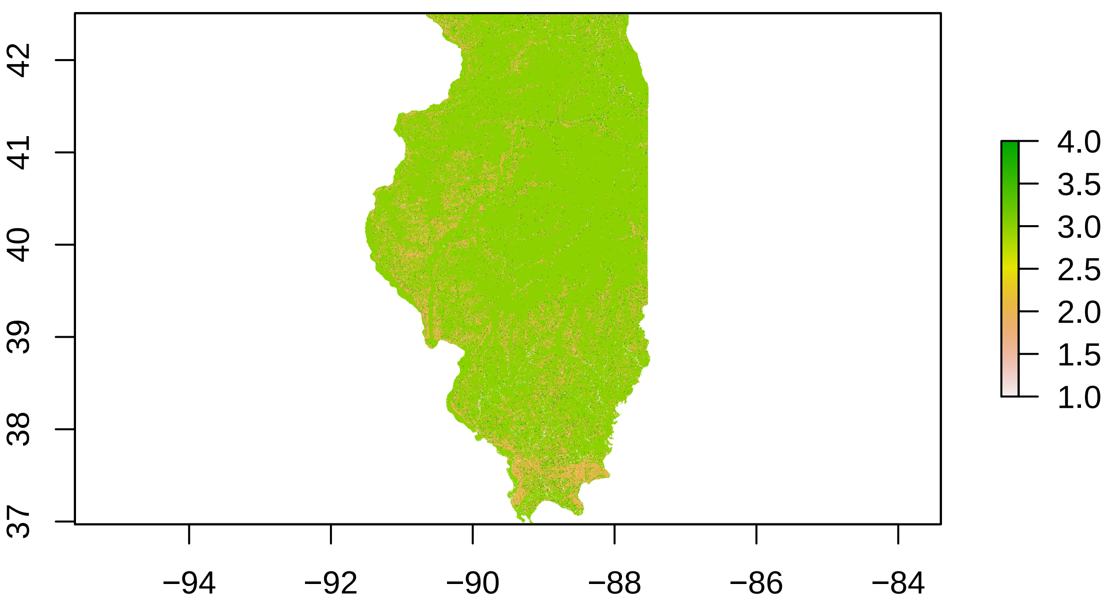

```
d2019 <- cdl2019 == 141 #Deciduous
e2019 <- cdl2019 == 142 #Evergreen
```

```
m2019 <- cdl2019 == 143 #Mixed
forest2019 <- d2019 + e2019 + m2019 #Total
plot(forest2019)
```



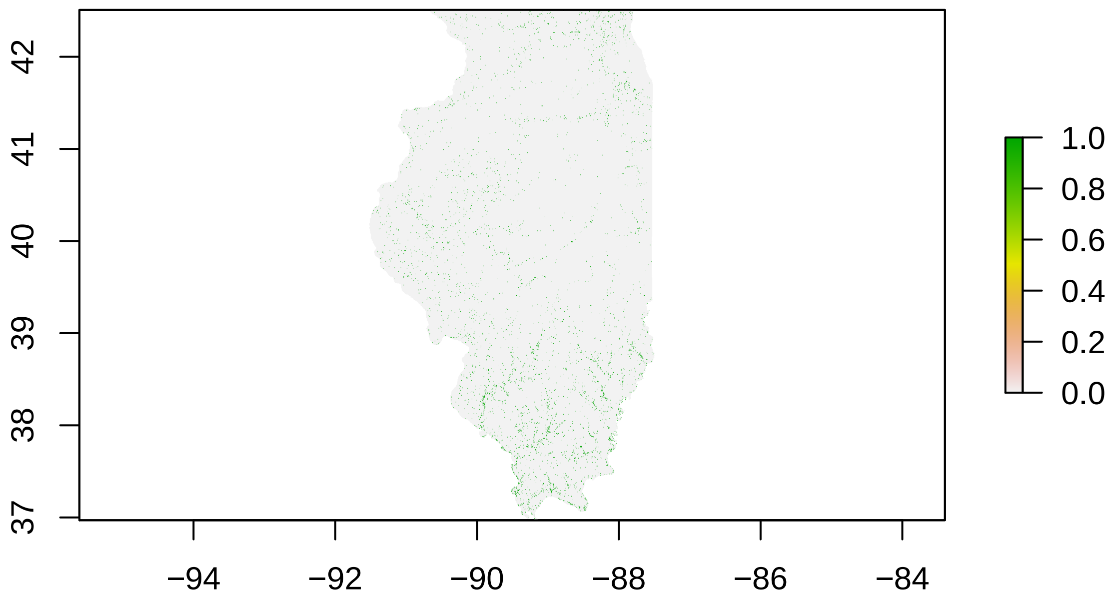
Raster Algebra

```
c1819 <- -2*forest2018 + forest2019 + 3
plot(c1819)
```

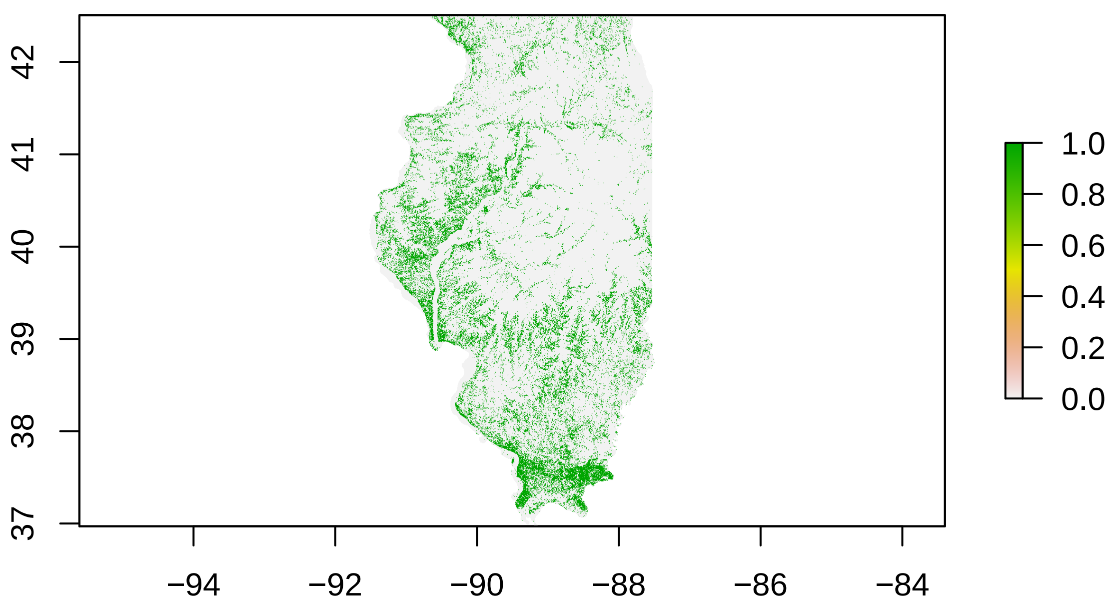


Displaying deforested, reforested, and areas of no change. Confirming that the deforested, reforested, and areas of no change are apparent.

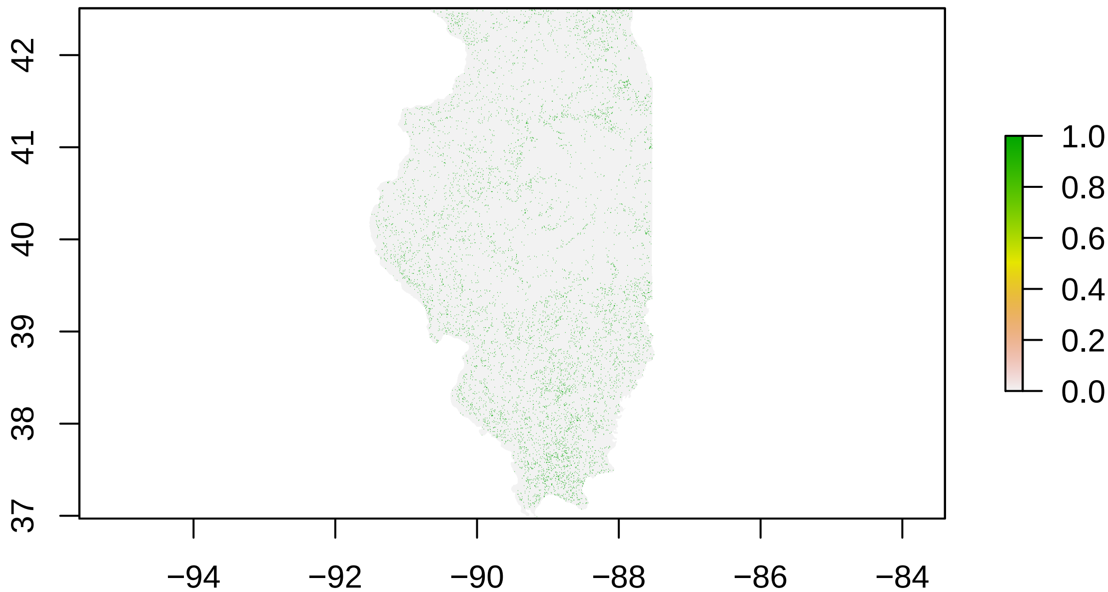
```
df1819 <- c1819 == 1
nf1819 <- c1819 == 2
rf1819 <- c1819 == 4
plot(df1819)
```




```
plot(nf1819)
```




```
plot(rf1819 )
```



Writing Raster Data to Computer for Shiny Application.

```
writeRaster(c1819, filename="./change/c1819.tif", format="GTiff", overwrite=TRUE)
```

Implementing Vector Layer

Here, I am reading in the Illinois County Shapefile for future manipulation.

```
il <- st_read("/Users/markbaker/Downloads/GEOG28602/final/IL_BNDY_County/IL_BNDY_County_Py.shp")
```

```
## Reading layer `IL_BNDY_County_Py' from data source `/Users/markbaker/Downloads/GEOG28602/final/IL_BN
## Simple feature collection with 102 features and 2 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: -91.51352 ymin: 36.96997 xmax: -87.49521 ymax: 42.50835
## CRS:            4269
```

```
il <- st_transform(il, "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0")
```

Cropping Raster Data to Particular Regions (A Case Example)

Examining key avenues to examine when implementing crop feature for raster data. This will be heavily utilized when attempting to display particular counties in Illinois in the Shiny App.

Cropping by Extent

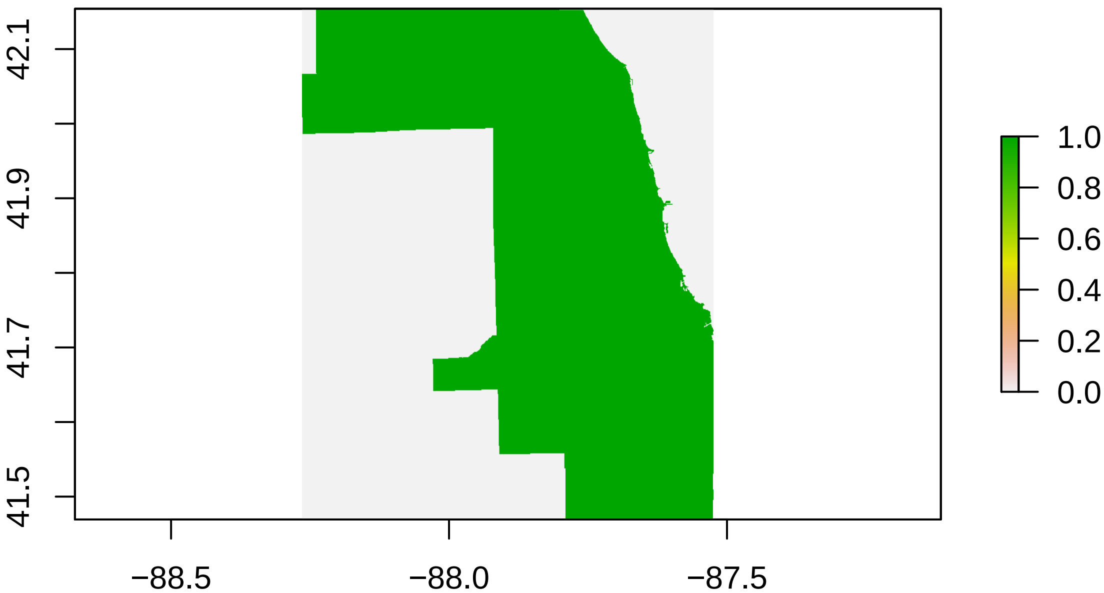
By cropping by extent first, we will minimize the time it takes to process the raster when attempting to display the entire raster data.

```
cook <- il[il$COUNTY_NAM == "COOK",]
p_cook_forest2018 <- intersect(c1819, cook)
```

Creating a mask of the polygon of interest

Assigning value of 1 to every pixel within area of interest and 0 to other pixels.

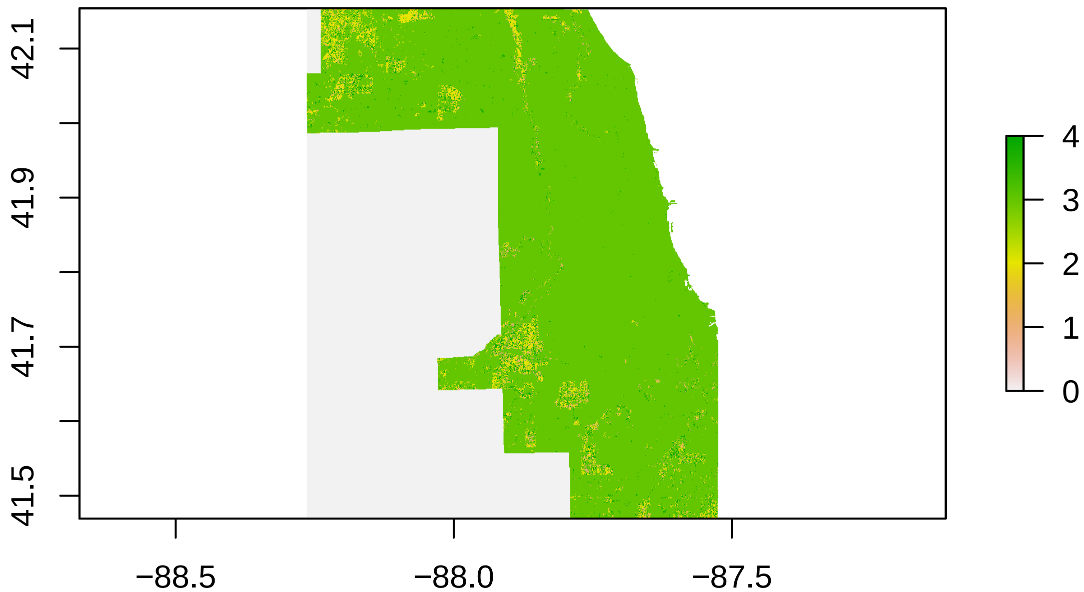
```
blank <- raster(p_cook_forest2018)
land_raster <- rasterize(cook, blank)
land_raster[!(is.na(land_raster))] <- 1
land_raster[(is.na(land_raster))] <- 0
plot(land_raster)
```



Applying the Mask to the forestry cropped extent

Using this, we will reach the end goal of cropping the raster to a particular polygon used.

```
cook_raster <- p_cook_forest2018 * land_raster
plot(cook_raster)
```



Generating Land Cover Calculator

Below is a function that counts the total number of pixels in the forestry change raster file and outputs a well-formated data.frame.

```
change_table <- function(raster) {
  freq_table <- freq(raster, digits=0, value=NULL)
  freq_table <- as.data.frame(freq_table)
  freq_table <- freq_table[(freq_table$value == 1| freq_table$value == 2 |
                          freq_table$value == 3 | freq_table$value == 4) ,]
  rfreq <- head(freq_table[freq_table$value == 4, ],1)$count
  dfreq <- head(freq_table[freq_table$value == 1, ],1)$count
  nfreq <- head(freq_table[freq_table$value == 2, ],1)$count
  other <- head(freq_table[freq_table$value == 3, ],1)$count
  forest_calc <- data.frame("Type" = c("Reforested", "Deforested", "No Change in Forest", "Other Land")
                          "Value" = c(4,1,2, 3),
                          "Percent" = c(rfreq/ sum(head(freq_table, 4)$count) *100,
                                        dfreq/ sum(head(freq_table, 4)$count) *100,
                                        nfreq/ sum(head(freq_table, 4)$count) *100,
                                        other/ sum(head(freq_table, 4)$count) *100))
  forest_calc
}
```

Here I am determine a simple way to remove the "Value" column from the generated table from change_table function.

```
table<- change_table(cook_raster)
table <- table[, -2]
table
```

```
##                  Type   Percent
## 1          Reforested  2.219263
## 2          Deforested  1.279240
## 3 No Change in Forest  4.206623
## 4          Other Land 92.294874
```