Lab 5: Quantifying and Characterizing Landscape Patterns

Instructor: Michael Treglia

Landscape Analysis and Modeling, The University of Tulsa, Spring 2016

Due Date: Thursday, 1 March 2016

There will be 10 questions for Part 1 (provided on Thursday) and additional questions for Part 2 (provided on Thursday), for a total of 20 points. All answers are due on the date specified above. PLEASE WRITE YOUR NAME ON ALL WORK

Part 1: Analyzing Categorical Rasters using FRAGSTATS

Part 1 Questions - Answer the following questions as you go through this lab. Feel free to use any materials from lecture, the internet, and FRAGSTATS documentation. Be sure to include your name with your answers. Unless otherwise noted, use the 8 cell rule for defining patches.

1) Before running any formal analysis in FRAGSTATS, look at the landscapes for Catoosa and Claremore.
   a) Which do you think has a higher value for Contagion? Justify your answer.
   b) Which do you think has higher Shannon's Evenness? Justify your answer

2) In each landscape, which land cover class do you expect has the largest Related Circumscribing Circle (CIRCLE)? Why?

For the next questions run FRAGSTATS for Catoosa and Claremore with the 8 cell patch rule. You can simply chose all analyses of interest (look through the questions, identify the analyses you need with the the 8 cell rule, check the appropriate boxes in FRAGSTATS, and Run the program).

3) a) which landscape actually does have a higher contagion value? (What are the values?)
   b) which landscape has a higher Shannon's Evenness values? (What are the values?)

4) In the Claremore landscape, which land cover class has the highest mean radius of gyration? (Use the Class metrics -> Area-Edge tab -> Radius of Gyration. Does the answer differ when considering the weighted mean vs the non-weighted mean?

5) What land cover class has the highest average (*not weighted average*) Fractal Dimension Index (FRAC) in each landscape?

6) In each landscape, what is the land cover type of the patch with the largest value for Related Circumscribing Circle(CIRCLE)? (You can import the results into R and use the 'order' function - see this website for help http://www.statmethods.net/management/sorting.html).

7) For the Catoosa landscape and only the developed land cover, compute the average and area-weighted average Core Area of patches (a class metric) with the edge depth set to 30.

The next questions involve computing separate analyses from those listed above. It may be easiest to run individual analyses to answer for each question.

8) For the Catoosa landscape and only the developed land cover, compute the average and area-weighted average Core Area of patches (a class metric) with the edge depth set to 60.

    a) How does changing the edge depth affect these core area metrics?

9) Carry out a moving window analysis of patch type richness for Catoosa (this is a landscape metric), with a square window and radius of 60; Present your result - it can simply be a screen-shot from QGIS if you want.

    a) What area appears to have the highest patch type richness? (e.g., Northwest, Northeast, Center?)
    b) What is the maximum richness found for the specified window size across the landscape?

10) Run FRAGSTATS for both Catoosa and Claremore using both the 8 cell patch rule and the 4 cell patch rule (you will need to run the analyses separately for the 8 cell and 4 cell rule). How many patches are identified in each of the landscapes based on each of those rules? (The Number of Patches calculation will be under Landscape Metrics - the Aggregation Tab - Subdivision section).

Introduction

This assignment involves a few steps, which involve pre-processing of categorical raster datasets, and loading them into and analyzing them in FRAGSTATS. Though this exercise covers some basics of preparing data for, and analyzing them with FRAGSTATS, a more comprehensive tutorial is available from the program developers at http://www.umass.edu/landeco/research/fragstats/downloads/fragstats_downloads.html#tutorials. FRAGSTATS is available for download and installation on Windows computers at this link: http://www.umass.edu/landeco/research/fragstats/downloads/fragstats_downloads.html#FRAGSTATS. After downloading it, simply un-zip the folder, and run the .exe file.

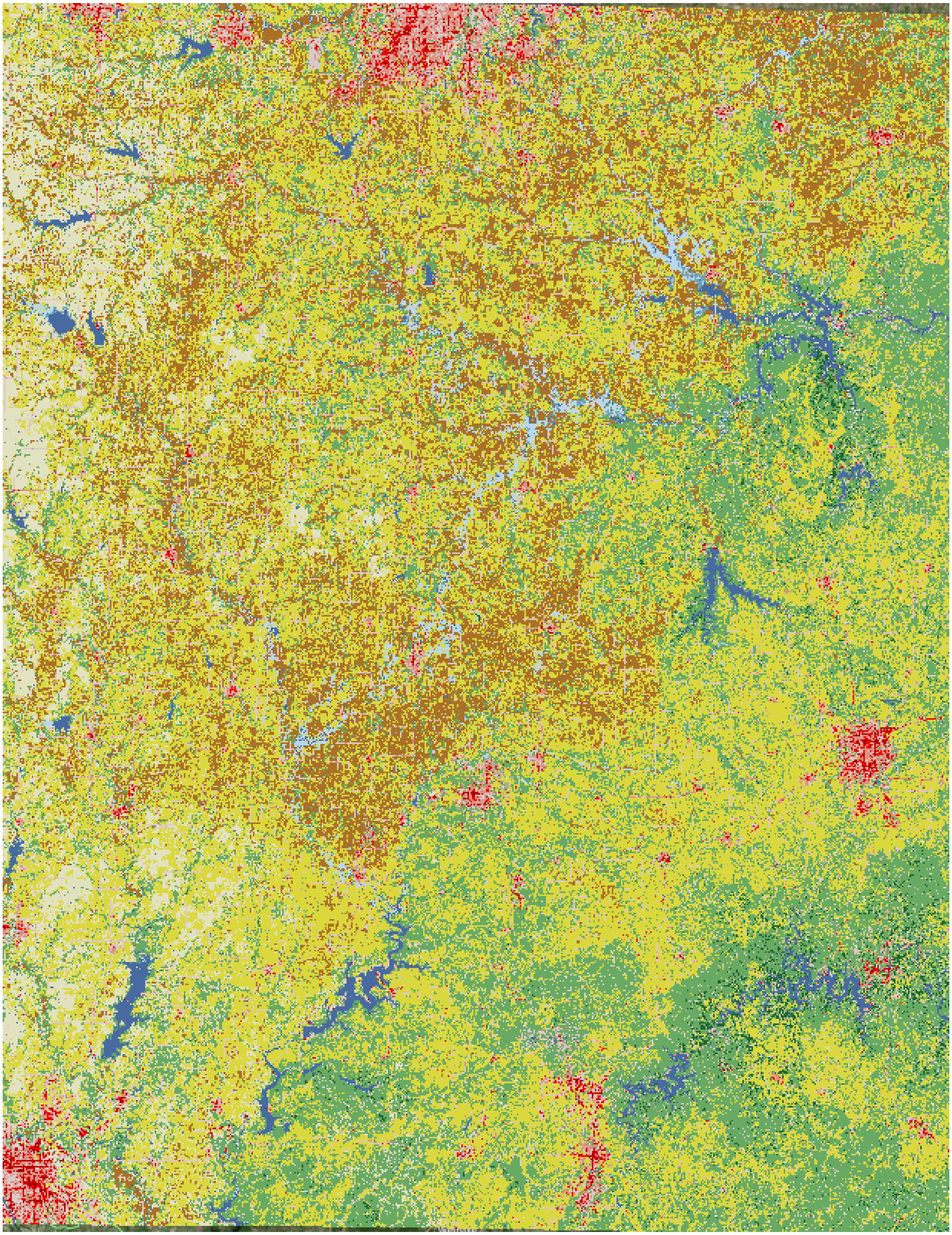
We will use data from the 2011 National Land Cover Dataset (NLCD), for a couple of areas near Tulsa, OK. The NLCD is available at 30 meter resolution, for the entire United States; multiple layers are available for this, including categorical land cover data, percent canopy cover and percent impervious pavement per pixel. These data are available for download via both the Multi-Resolution Land Characteristics Consortium (MRLC) and The National Map Viewer and Download Platform. You can store and work with these data in the directory of your choice.

For this lab, we will start with a 3 x 3 degree tile of the categorical NLCD layer for north-eastern Oklahoma, which includes part of Tulsa. This dataset is available through the University of Tulsa Harvey page, and can be downloaded, with full metadata by clicking this link (and it can be downloaded from the sources listed above). Before we get into doing actual analyses, we will go through a few pre-processing steps, often necessary in setting up the types of analyses we will carry out. There are lots of different tools that can be used for these operations, but we'll use QGIS (version 2.12). We will use R to manage and process our results.

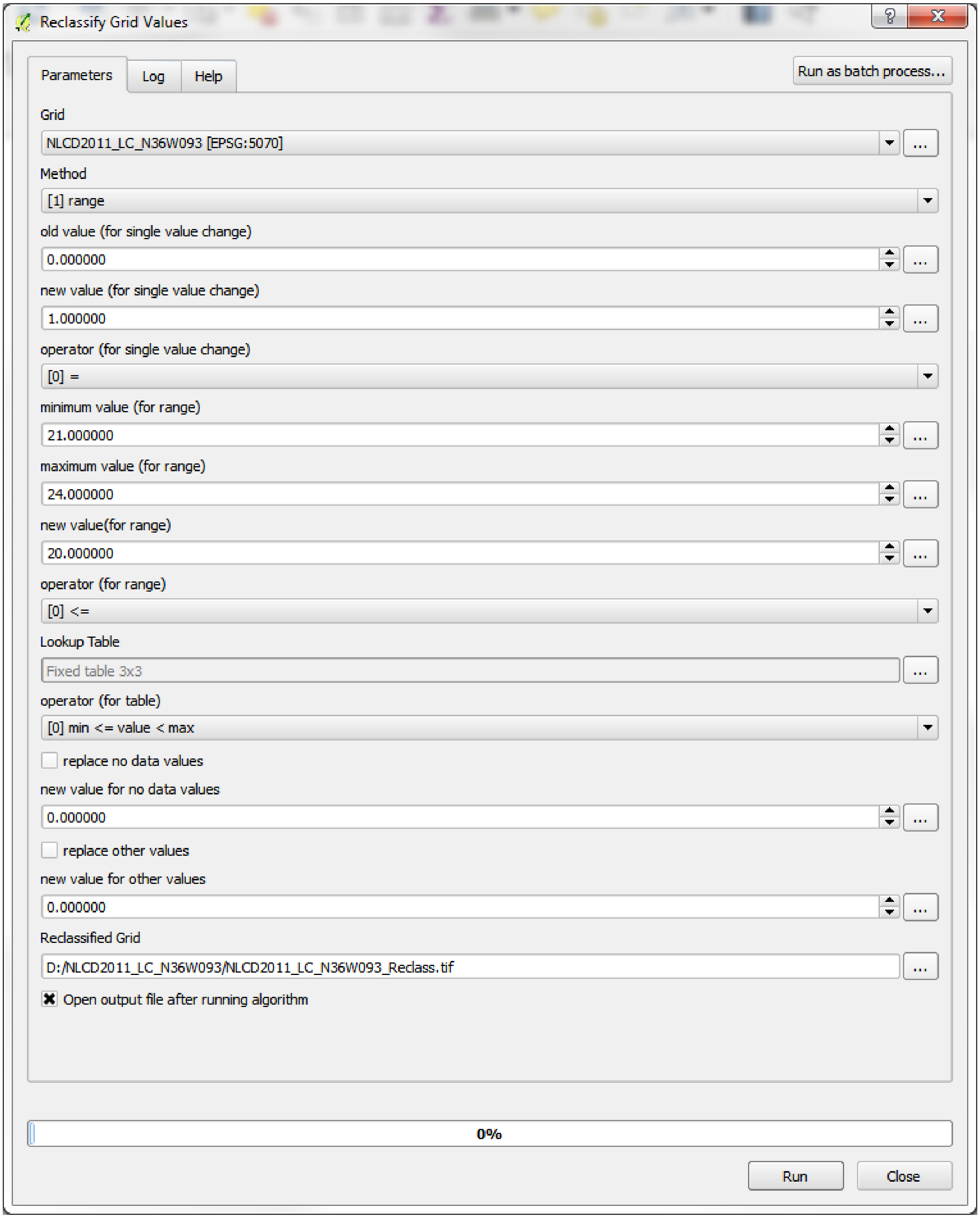
Reclassify Raster

The 2011 NLCD has 20 specific potential land cover types, 15 of which occur in our focal tile. Each class is assigned a class number, as seen in the legend for this layer, available online at http://www.mrlc.gov/nlcd11_leg.php. If you use the 'inspect' tool of QGIS, you will see that each pixel has a numeric value, corresponding to legend in the previous link. The land cover classes are also grouped coarsely, as indicated by the 10s place of the ID number (e.g., all numbers in the 20s indicate varying levels of development). Sometimes this scheme may not meet our needs perfectly. For example, in this lab we will consider all classes of development as the same. To deal with this, we will reclassify the raster, using a similar process as we used in the first lab of the semester.

Import the NLCD layer into QGIS as you would any other raster layer (use the 'NLCD2011_LC_N36W093.tif') - the resulting layer should look something like the image below. According to the metadata (available in an .html file that comes in the folder with this dataset), the data are in an Albers Equal Area projection; if you import some Google or Bing (or other)imagery and ensure that on the fly projection is enabled, that layer should appear in northeastern Oklahoma. The horizontal units for this projection layer are meters, which is important to know for later steps. You can also set the CRS for this layer to the standard EPSG code, 5070. (This won't change the projection, but will designate the corresponding EPSG code).



We will use SAGA, available form the Processing Toolbox to do the reclassification. In the processing toolbox, expand the SAGA menu, and find “Grid-Tools”; then, double click “Reclassify grid values”. If the processing toolbox is not visible, click the “Processing” menu at the top of the QGIS screen, and click “Toolbox”. This may open in the “Simplified Interface”, showing only a couple of tools - if this is the case, look to the bottom of the Processing Toolbox pane (typically appears on the right side) - there will be a drop-down menu where you can choose either “Advanced Interface” or “Simplified Interface” - choose the Advanced Interface.



Grid
NLCD2011_LC_N36W093 [EPSG:5070]

Method
[1] range

old value (for single value change)
0.000000

new value (for single value change)
1.000000

operator (for single value change)
[0] =

minimum value (for range)
21.000000

maximum value (for range)
24.000000

new value(for range)
20.000000

operator (for range)
[0] <=

Lookup Table
Fixed table 3x3

operator (for table)
[0] min <= value < max

☐ replace no data values

new value for no data values
0.000000

☐ replace other values

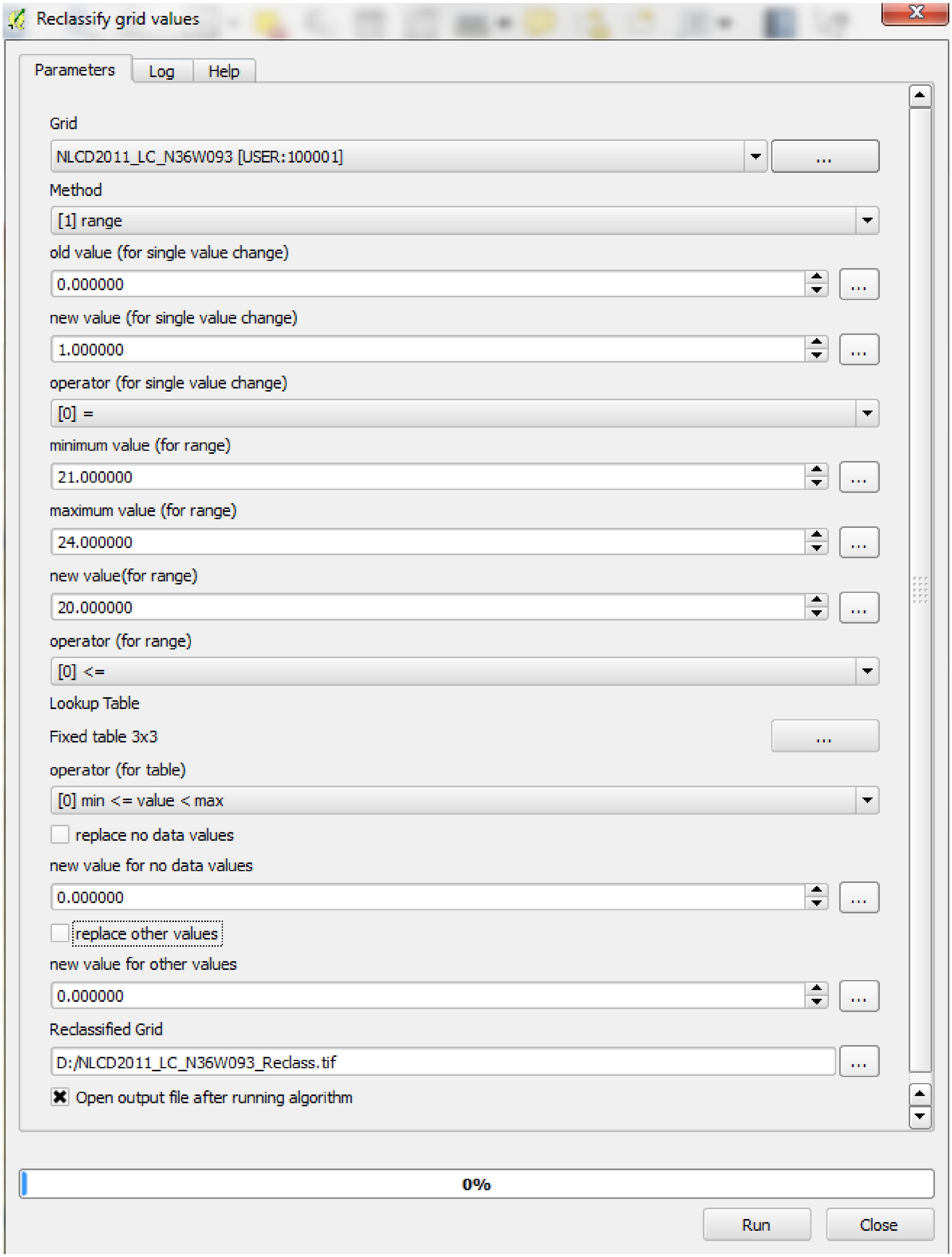
new value for other values
0.000000

Reclassified Grid
D:/NLCD2011_LC_N36W093/NLCD2011_LC_N36W093_Reclass.tif

☒ Open output file after running algorithm

In the window that pops up, you will designate the settings for this operation. These below settings will work for this lab - in the future, you can adjust these settings according to your own needs. After setting up all of the options, click 'Run'.

- Under 'Grid', chose the appropriate layer (NLCD2011_LC_N36W093).
- For 'Method', choose 'Range'.
- Set the minimum and maximum values for the range option as 21 and 24, respectively. This tells the operation that you want to reclassify values from 21 through 24 to a new value.
- For 'new value (for range)', set the value to 20, and leave the 'operator (for range)' set to the default ('[0]<=').
- Un-check the boxes for 'replace no data values' and 'replace other values'. This is important, as it will ensure other values in the layer will remain as they were originally.
- Designate a 'Reclassified grid' for your output file - saving it as a .tif file generally works well.


Reclassify grid values

Grid
NLCD2011_LC_N36W093 [USER:100001]

Method
[1] range

old value (for single value change)
0.000000

new value (for single value change)
1.000000

operator (for single value change)
[0] =

minimum value (for range)
21.000000

maximum value (for range)
24.000000

new value(for range)
20.000000

operator (for range)
[0] <=

Lookup Table
Fixed table 3x3

operator (for table)
[0] min <= value < max

□ replace no data values

new value for no data values
0.000000

□ replace other values

new value for other values
0.000000

Reclassified Grid
D:/NLCD2011_LC_N36W093_Reclass.tif

☒ Open output file after running algorithm

After this operation, your newly reclassified layer will appear in QGIS in gray-scale colors. In the list of layers in QGIS you can simply right click on the original NLCD, select ‘Copy Style’, and then right click on the new layer and select 'Paste Style' - this will apply the original color palette to this new layer. Note, the areas classified as 'Developed', now designated as class 20, will appear black - that is because there was previously no class number 20 - you can go to the style settings (in the layer properties) and designate a color for class 20.

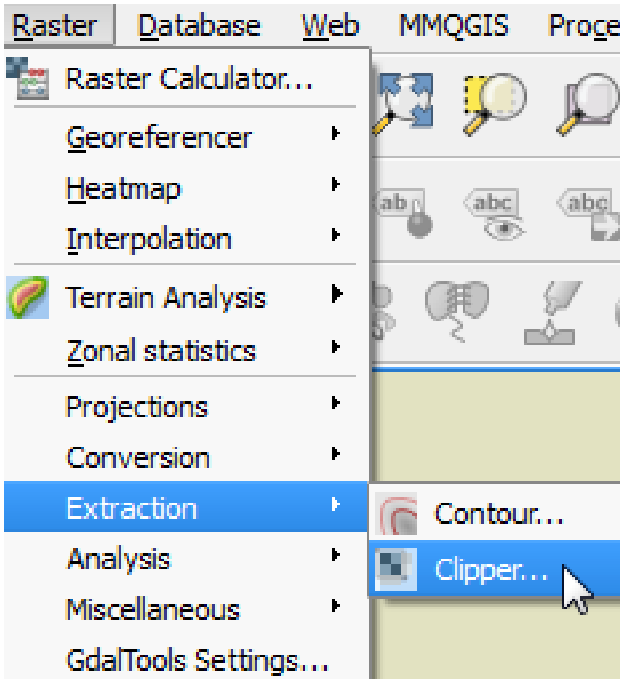
Clipping Areas for Analysis

Now that we have the desired land cover classes for this analysis, we can focus on extracting a couple of areas for analysis. Again, there are numerous ways to do this - what is shown here is probably among the simplest and easiest, though for this exercise, we will ignore bordering pixels along our focal areas.

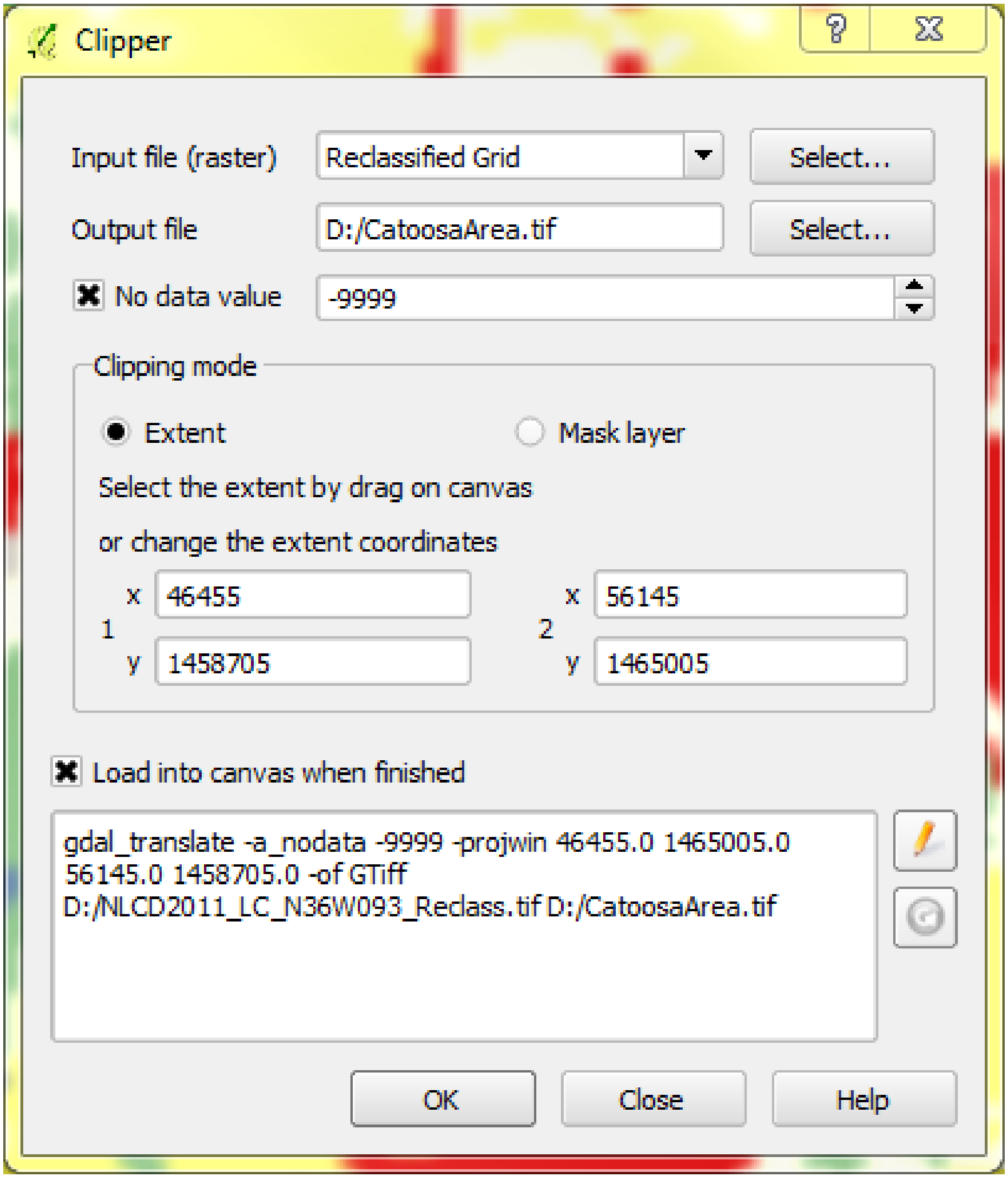
We will use the Raster Clipper tool in QGIS to extract two focal areas for this analysis: an area near Catoosa, OK that we have highlighted during lab, and a nearby area of Claremore, OK. The corresponding extents will be as follows:

- **Catoosa Area** Lower Left: x = 46455, y = 1458705; Upper Right: x = 56145, y = 1465005
- **Claremore Area** Lower Left: x = 30075, y = 1469655; Upper Right: x = 39765, y = 1475955

For each area, go to the Raster menu at the top of the QGIS window, mouse over 'Extraction' and click 'Clipper'.



In the window that opens, you can select the layer you want to clip (the reclassified NLCD layer), designate the output file, set the 'Clipping mode' to 'Extent', and fill in the coordinates. I typically set the 'No data value' to -9999, to avoid any confusion of my focal area includes locations without data. This is not the case for this area, but can be good practice. The image below is the setup for the Catoosa area; *you will also need to do this for the Claremore area (designated above).*



Clipper

Input file (raster): Reclassified Grid

Output file: D:/CatoosaArea.tif

☒ No data value: -9999

Clipping mode

● Extent ○ Mask layer

Select the extent by drag on canvas

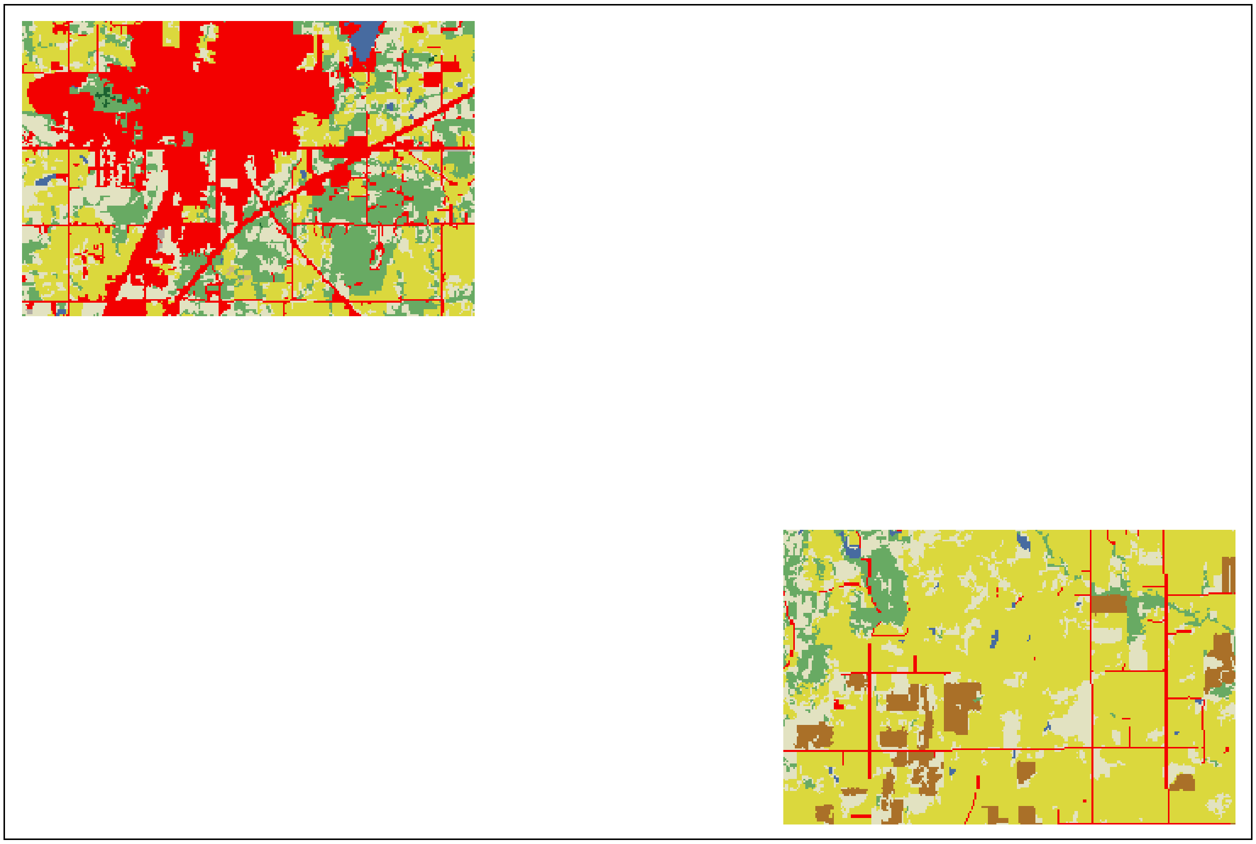
or change the extent coordinates

1	x	46455
	y	1458705
2	x	56145
	y	1465005

☒ Load into canvas when finished

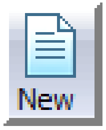
```
gdal_translate -a_nodata -9999 -projwin 46455.0 1465005.0
56145.0 1458705.0 -of GTiff
D:/NLCD2011_LC_N36W093_Reclass.tif D:/CatoosaArea.tif
```

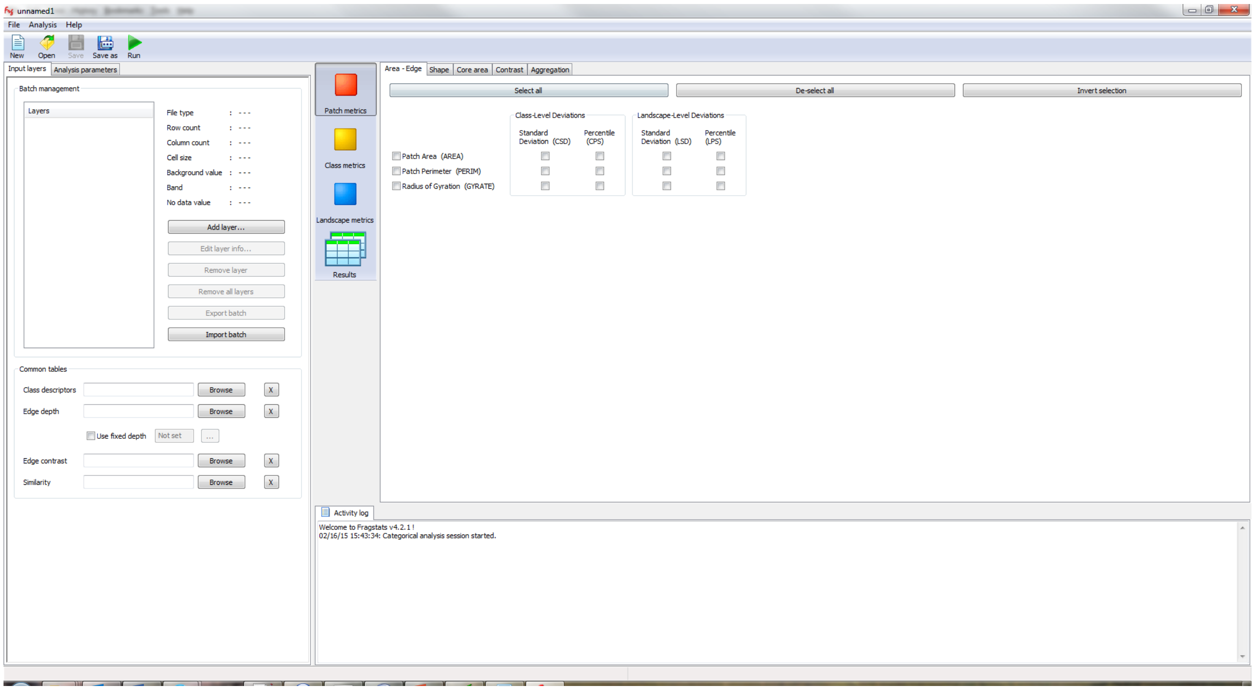
The two areas are nearby, and your results should look like this:



Load Data Into FRAGSTATS

Now that we have a couple of landscapes to analyze, we can import them to FRAGSTATS. Open FRAGSTATS on your computer - you should be able to find it in your Start Menu. When the program opens up, the screen will appear blank; click the ‘New’ icon at the top and the screen below should appear.

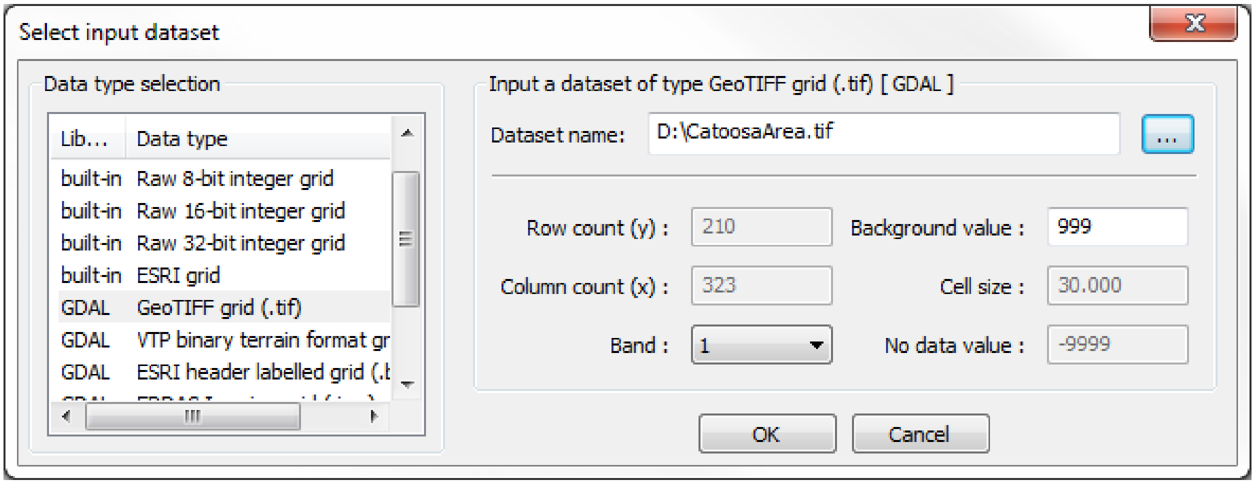




File type : - - -
Row count : - - -
Column count : - - -
Cell size : - - -
Background value : - - -
Band : - - -
No data value : - - -

```
Welcome to Fragstats v4.2.1 !
02/16/15 15:43:34: Categorical analysis session started.
```

Click the 'Add layer. . . ' button. Then, chose the file type you will import (GeoTIFF grid for these data), and browse to your dataset. In this case we have no 'Background value', but if you wanted to consider one of your classes as a background, you could indicate that class number in that box. The rest of the information should be filled in automatically. For the Catoosa area, this window should look like the image below.



Lib...	Data type
built-in	Raw 8-bit integer grid
built-in	Raw 16-bit integer grid
built-in	Raw 32-bit integer grid
built-in	ESRI grid
GDAL	GeoTIFF grid (.tif)
GDAL	VTP binary terrain format gr
GDAL	ESRI header labelled grid (.b

Dataset name: D:\CatoosaArea.tif

Row count (y) : 210
Background value : 999
Column count (x) : 323
Cell size : 30.000
Band : 1
No data value : -9999

Note the 'Cell Size' is identified as '30.00'. FRAGSTATS assumes this is in meters, and all area and perimeter calculations will be based on this cell size. This is correct, and can be confirmed via the metadata and checked by measuring pixels while zoomed in within QGIS. If your layer is in projected coordinates (i.e., Lat/Long), it is wise to reproject to a meter-based coordinate reference system prior to analysis.

You can import the NLCD for the Claremore area too.

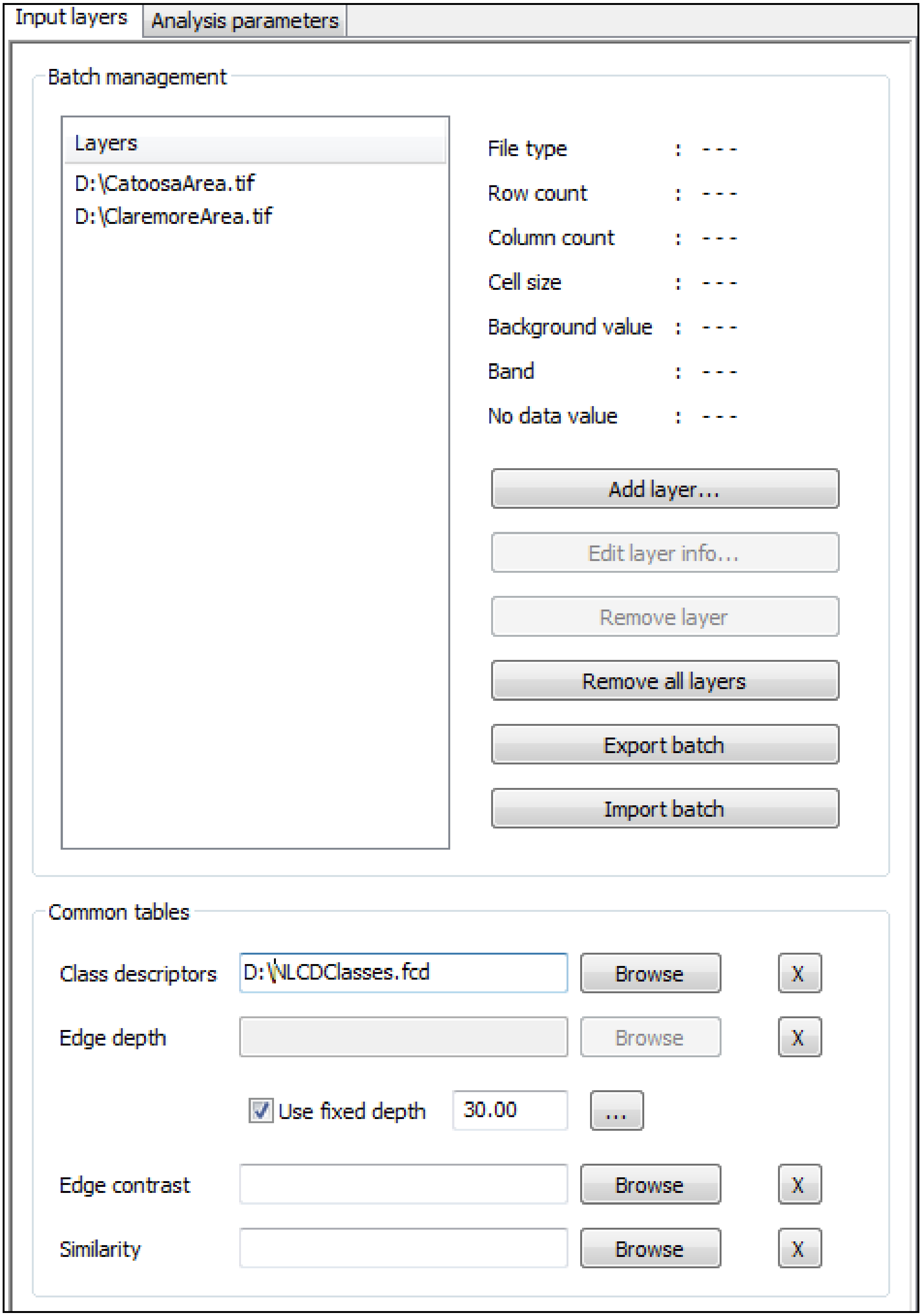
For this lab, we will only import one other file - the 'Class Descriptors' file, 'NLCDClasses.fcd'. You can download it from the Harvey website under Week 6 lab material, or you can view it at this link. If you are simply viewing this online, you can copy and paste it into a text file (e.g., with Notepad on Windows), and save it as 'NLCDClasses.fcd'. If you downloaded the file from Harvey, open it with Notepad and explore it -

this file simply tells FRAGSTATS which names correspond to the numeric land cover codes. I set this up for you, with the land cover classes that might be in these areas (and with the Developed classes all categorized as a single class), but you can easily create a similar file or adjust this for your own purposes.

One last thing that we will set is the edge depth, required for computing Core Area metrics. We will set a fixed edge depth for all land cover classes; check the box for 'Use fixed edge depth', click the '. . . ' icon, and for now set the value to 30.

If you wanted to use different edge depths for different land cover classes, you could use a file, similar to the Class Descriptors file; similarly, you can use files to specify similarity and contrast between different classes. This is described in detail in the tutorial material available from the FRAGSTATS developers.

At this point the left portion of the screen, with the 'Input Layers' tab, should look similar to this:



Layers

D:\CatoosaArea.tif

D:\ClaremoreArea.tif

File type : - - -

Row count : - - -

Column count : - - -

Cell size : - - -

Background value : - - -

Band : - - -

No data value : - - -

Class descriptors: D:\NLCDClasses.fcd

Edge depth:

☑ Use fixed depth 30.00

Edge contrast:

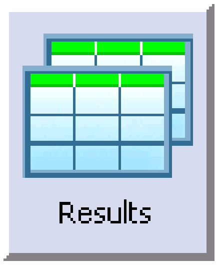
Similarity:

Analysis in FRAGSTATS

Now that we have data imported to FRAGSTATS, we can get on to some analyses. In the pane on the left, use the tabs at the top switch to ‘Analysis Parameters’. There are lots of things to adjust. For now we will leave the box checked for ‘Use 8 cell neighborhood rule’. We will also save the results - check the box for ‘Automatically save results’, and browse to designate a results file. I will name mine ‘CatoosaClaremore1’, and subsequent analyses can be named in a logical way of your choosing. There are various sampling strategies that can be chosen, in which only specified parts of the landscape are analyzed. We will leave the Sampling Strategy set to the default (No Sampling), and we will check the boxes for Patch Metrics, Class Metrics, Landscape Metrics, and Generate Patch ID file. You can browse through the Patch Metrics, Class Metrics, and Landscape Metrics tabs in the middle of the screen to see what analyses are available. We discussed many of the metrics in class, and you can read further about the details in the FRAGSTATS Documentation.

Notice that a lot of metrics can be calculated at all of the three scales - Patch metrics apply to individual patches; Class metrics describe the arrangement of specific land cover types within the landscape; and Landscape metrics describe the relationship of all land cover types, and the arrangement of all types with respect to all other types. For patch characteristics, we selected he option to generate a ‘Patch ID’ file - this is a raster, where each pixel is assigned a value that identified what patch number it is associated with.

After you have the analyses you wish to run selected, simply click the ‘Run’ icon at the top of the screen. The results can be accessed from within FRAGSTATS by clicking the ‘Results’ icon near the center of the screen.



In the Analysis Parameters pane (on the left), you can also select to carry out Moving Window Analyses. These create new rasters for each class and landscape metric selected (which can be imported into GIS software just like any other raster). For class metrics, it will create a separate result raster for each metric of each class, so be aware the results can add up quickly.

Viewing Results

As aforementioned, you can view FRAGSTATS results directly within the software. When you run analyses with the option to automatically save results, the results are saved as comma delimited files, albeit the extension (i.e., suffix for the file names) will read ‘.patch’, ‘.class’, or ‘.land’, corresponding to results for patch, class, and landscape metrics. These can be opened in a text editor like Notepad, and they can be imported into R using the simple ‘read.csv()’ command as illustrated below.

```r
CatoosaClaremorePatches <- read.csv("CatoosaClaremore1.patch")
CatoosaClaremoreClass <- read.csv("CatoosaClaremore1.class")
CatoosaClaremoreLand <- read.csv("CatoosaClaremore1.land")
```

There is also an R package available for download that aggregates landscape and class metrics into a single data frame for analysis. The package is available for download at http://www.umass.edu/landeco/research/fragstats/downloads/fragstats_downloads.html#Rfrag.

To install the package in R, open R, and use the 'Packages' menu at the top of the screen to navigate to 'Install package(s) from local zip file' - then browse your computer to your download location and select the appropriate file (Rfrag_1.0.zip). Load the packages as you would any other:

```r
library(Rfrag)
```

There is currently only one function: 'frag.combine'. You can use the help for this function to see the specific arguments, and here is an example of its usage:

```r
example <- frag.combine(path='D:/',inland = 'CatoosaClaremore1.land',
    inclass = 'CatoosaClaremore1.class')
```

You can compare this to the original files - the result of this function is a table where each row represents an individual landscape (in this case, Catoosa and Claremore landscapes), and columns represent the landscape metrics, followed by the metrics for each class. This can make it easy to compare characteristics of multiple landscapes by simply comparing entries for respective columns across multiple rows.

Part 2: Calculating Some Surface Metrics

Part 2 Questions - Answer the following questions as you go through this lab. Feel free to use any materials from lecture, the internet, and supplemental reading materials posted on Harvey (in particular McGarigal, K., S. Tagil, and S. Cushman. 2009. Surface metrics: an alternative to patch metrics for the quantification of landscape structure. Landscape Ecology 24:433-450, and the supplement associated with it). Be sure to include your name with your answers.

1) Produce basic plots of elevation for Catoosa and Claremore (include them in your answer sheet).
2) Which landscape (Catoosa or Claremore) has a higher mean elevation?
3) How do the histograms of the two focal landscapes compare? Is one more skewed or peaked than the other? (which area, and in which direction - does the tail extend further to the left or right?) Which landscape seems to have more even terrain based on the histograms?
4) Which landscape is rougher? Is the answer the same when you compare Average Roughness and the Standard Deviation of elevation? (If not, why might that be?)
5) Which landscape has higher average slope? Give the value in both radians and degrees.
6) Compare the histograms for slope - do you notice any major differences in the shapes? or are they generally uniform? If you see major differences, elucidate what might cause that (feel free to look at the digital elevation model and slope maps if that helps).
7) Calculate the 'average roughness' for slope of each landscape. Provide the R code you used. (Remember, this is a metric of how much variation there is in slopes present.)
8) Surface metrics could potentially be combined with the patch mosaic paradigm. How would you apply surface metrics to individual patches? (Just a brief description is sufficient).
9) Surface metrics for continuous raster layers and landscape/patch mosaics each have benefits and applicability in different situations. When/how might surface metrics be better for quantifying a pattern of a landscape? Provide an example from one of your study systems.
10) When/how might landsacpe/class/patch metrics be better for quantifying a pattern of a landscape? Provide an example from one of your study systems.

Introduction

In its current version, FRAGSTATS is a powerful tool for working with categorical raster layers. Though planned future versions will calculate surface metrics for continuous rasters (described in the McGarigal et al. 2009 paper listed above), this feature is not currently implemented. Surface metrics can allow you to characterize landscapes (and patches) according to continuous variables (e.g., elevation, slope, percent canopy cover, percent Software designed to calculate surface metrics can be expensive, though we can calculate some metrics pretty easily using R. We will only focus on a couple here will, but this lab will also familiarize you with working with raster datasets in R.

Though R was developed for statistical computing, the R programming language can be used for a wide variety of tasks, and it can integrate with other languages. Thus, myriad packages have been developed for importing data of various formats, and manipulating them in countless ways. For working with spatial data in particular, numerous packages are available, generally outlined in this Task View: http://cran.r-project.org/web/views/Spatial.html. For this lab we will use the 'raster' package and s ome dependencies: * rgdal; * raster; and * rgeos. * sp;

We will do some minor data preparation, similar to as we did in Part 1, but this time we will use R instead of QGIS, to highlight some of what we can do in R. In your own work, you can use the tools of your preference, both are powerful and have their own advantages (e.g., R is a simple language to program and script your work in, but might take more work to figure out commands and can be slower for big datasets; QGIS has a GUI (graphical user interface) and is fast for processing, but programming/scripting requires learning another language, Python).

Our analyses will focus on the 30 meter National Elevation Dataset for the same areas as we used in Part 1 (near Catoosa and Claremore, Oklahoma, USA). * An appropriate 1 x 1 degree tile (tile n36w096) is available on the Harvey website for this class and here as a .img file. * You should un-zip the file and place it in your preferred file location before conducting analyses. * You can also download this layer from from The National Map Viewer.

For your own analyses, some layers that you may be interested in are Percent Impervious Cover and Percent Canopy Cover from the the 2011 National Land Cover Dataset. You can download these for the conterminuous United States from the NLCD website, or in 3 x 3 degree tiles or state extents from The National Map Viewer.

Installing and Loading R Packages

For an introduction to R and R packages, see the Intro to R material and the beginning of the Lab 3 material associated with this course.

You will need to install the packages listed above. the package 'sp' comes with R, but the others need to be installed - you can do this using the 'install.packages' function. Once packages are installed, you can load them using the 'library' function.

Note: For Macs, check out instructions for installing rgdal and rgeos at the following link - note, you'll need to install plain geos and gdal before installing the respective R packages: http://tlocoh.r-forge.r-project.org/mac_rgeos_rgdal.html

For example, for the raster package, use this code:

```r
# To install a single package:
install.packages("raster")

# To install multiple packages at once, follow this example:
install.packages(c("raster", "rgeos"))

# To load packages (must be done individually for each package; when a
# package has dependencies, the dependencies will be loaded when you load
# the main package you are loading):
library(raster)
library(rgeos)
```

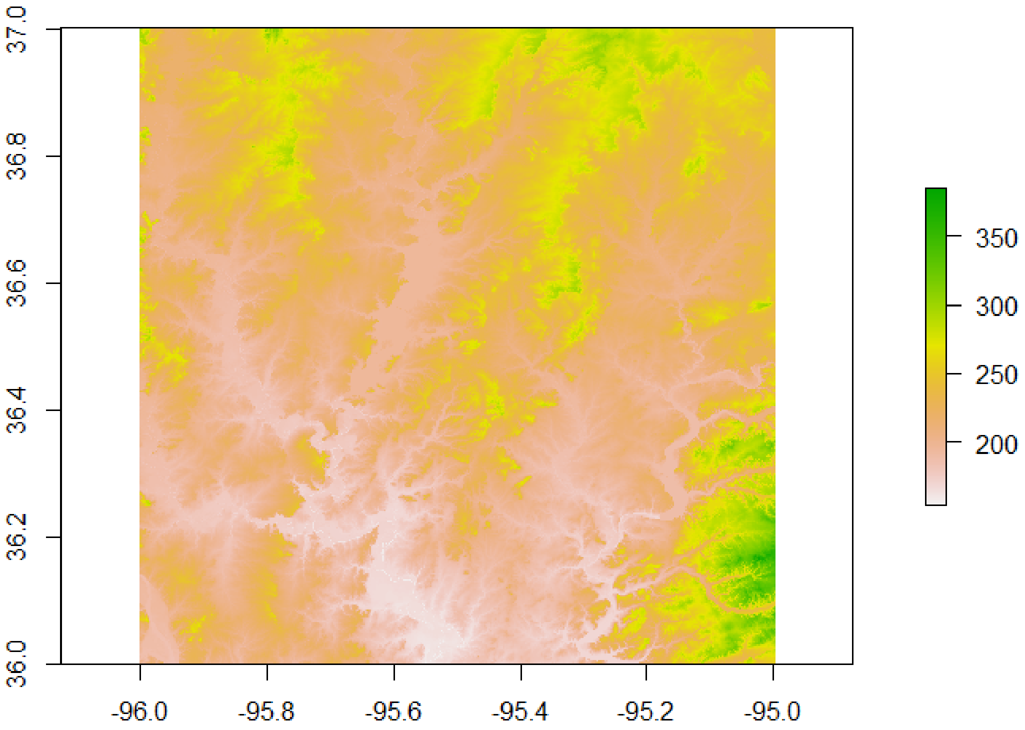
Loading Raster Layers Into R; Finding Projection Info & Reprojecting Layer

As you get started, you should set your working directory to your desired folder location (use the 'setwd' function), where you should have the data you'll be working with stored.

The function to import a raster is pretty simple - just the 'raster' command in the package of that name. This function is pretty flexible, and can be used to create blank rasters, or import existing raster files. You can plot a raster using the 'plot' command. *Note: with the raster package open, if you look up the help for 'plot' in R, you will be asked to choose help for generic plotting functions or for plotting a raster - in this case you would want the help for plotting a raster.*

```
dem <- raster("imgn37w096_1.img")

# Plotting may take a minute or two - if you wait until we have clipped
# rasters for our focal areas, it will be much quicker as this function will
# be plotting smaller areas (i.e., less pixels).
plot(dem)
```



As with any other R object, we can look at the structure, identify the class, etc. You can also calculate the summary statistics - because Raster layers typically have a large number of values (in individual pixels), it will typically take a sample of values for these computations and note that.

```
str(dem)
class(dem)
summary(dem)
```

In the structure there is a lot of information, summarizing some important characteristics of the raster (e.g., the boundaries of the layer, the class of object, and the coordinate reference system). The raster layer is an object of class 'RasterLayer', associated with the 'raster' package. Also, note that the coordinate reference system is "+proj=longlat +datum=NAD83 +no_defs +ellps=GRS80 +towgs84=0,0,0" - thus, the data are in projected coordinates (latitude/longitude), in datum WgS 84.

If you look at the metadata for this layer (the .html file in the folder with the DEM), you will find that the height values (i.e., vertical units) are in meters; for some calculations such as slope, it is better to have the horizontal units the same as the vertical units. Thus, we will reproject the layer to a meter-based CRS. For consistency with the National Land Cover Dataset, we will simply use an Albers Equal Area Projection (EPSG code: 5070). The command for this is 'projectRaster'. As shown in the examples within the help, you can simply assign the re-projected raster to a new object, operating the function on the existing raster, and indicating what CRS you wish to use. You can also specify the interpolation (nearest neighbor would be for a categorical raster; bilinear is the default, and that or cubic spline is generally appropriate for continuous rasters). This function expects the CRS in a different format: 'PROJ.4' description of the CRS. To figure out how to convert the EPSG code to the PROJ.4 format, I typically do a web search something like "EPSG 5070 PROJ.4". The first result is http://epsg.io/5070-1252 - if you scroll down, on the left hand side you will see a list under the heading 'Export' - click on 'PROJ.4' and it will show you how this EPSG code is represented in the PROJ.4 format.

- "+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96 +x_0=0 +y_0=0 +ellps=GRS80 +towgs84=1,1,-1,0,0,0,0 +units=m +no_defs"

You could alternatively get this from QGIS - when you are in the window for setting the projection of a layer, it displays the text for the PROJ.4 code of the projection that is highlighted. We can now reproject the data:

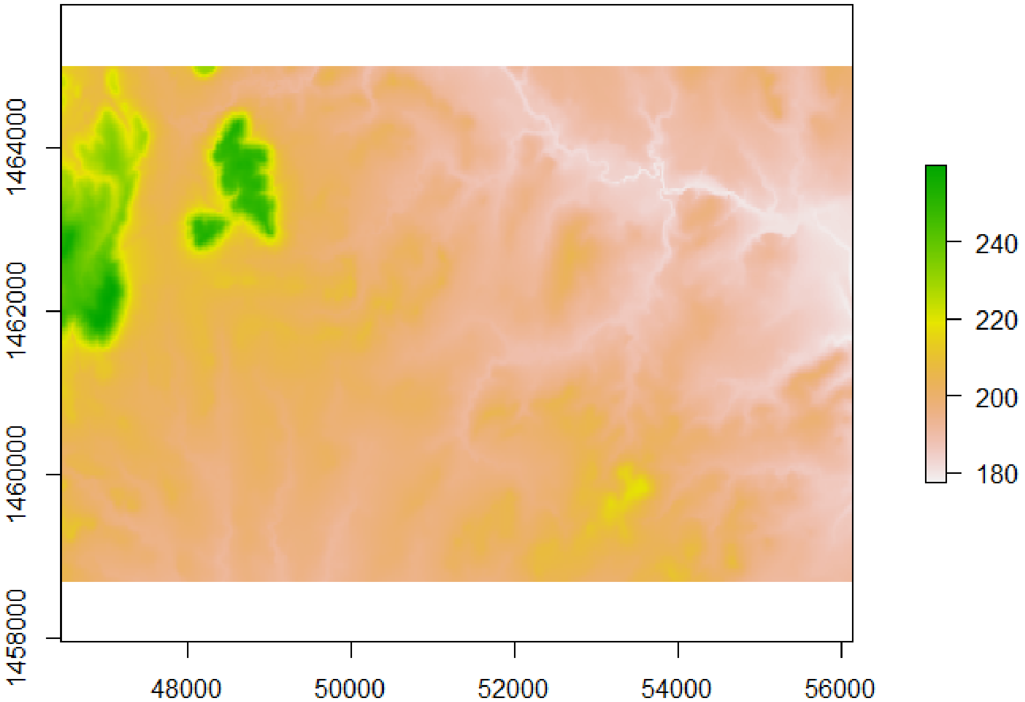
```
demAlbers <- projectRaster(dem, crs = "+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96 +x_0=0 +y
```

Now we are ready to clip the raster to the respective areas - we will use the same boundaries for Catoosa and Claremore as in part 1 of this lab (see the lab 1 handout for the coordinates). We will set up extents that we wish to crop the layer to using the appropriate coorinates, and crop to those. The main functions we will use are 'extent' and 'crop'. Below is an example for the Catoosa area - use do the same for the Claremore area on your own.

```
CatoosaExtent <- extent(46455, 56145, 1458705, 1465005)
# the order expected is 'xmin, xmax, ymin, ymax'

demCatoosa <- crop(demAlbers, CatoosaExtent)

# Plot to verify that this appears coorect:
plot(demCatoosa)
```



Calculation of Surface Metrics

Now that you have these focal landscapes clipped out, you can calculate surface metrics. Note that you can do regular math with rasters, adding and subtracting, multiplying, and dividing, etc., but to get specific statistics for a layer, you need to use the 'cellStats' function (see below. You can also view the histogram for this each layer using the simple 'hist' function.

```
hist(demCatoosa)
demCatoosaMean <- cellStats(demCatoosa, mean)
```

One of the surface metrics we can look at is Average Roughness - or the average deviation from the mean. There are a few calculations we need to do: 1) calculate the mean (as shown above); 2) then, calculate a new value for every pixel, as the absolute value of the differene between the pixel value and the mean for the area; and 3) calculate the mean of the raster computed in Step 2.

Here is an example of this for Catoosa, broken out into multiple steps, but it could be combined into a single line of code.

```
demCatoosaMean <- cellStats(demCatoosa, mean)
demCatoosaDeviance <- abs(demCatoosa - demCatoosaMean)
CatoosaAvgRough <- cellStats(demCatoosaDeviance, mean)
```

Another metric we can look at for roughness is the standard deviation of the elevation within a landscape. I won't provide this code here, but you will need to calculate it for the assignment (hint: search around for the R function to calculate the standard deviation; you will need to use the cellStats function).

Deriving Other Topographic Variables from Raster Layers

With digital elevation models, we often want to compute derivitives from the layer - for example, slope and aspect are among the most commonly used layers derived from DEMs. A convenient function for this is 'terrain' in the raster package. Check out the help for this, and look at the options and details for what can be computed using this function.

As an example, we will calculate slope.

```
slopeCatoosa <- terrain(demCatoosa, opt='slope', units='degrees')
```

Although this example of slope is for elevation, you can also calculate the 'slope' of other values, as slope is just the first derivative (rate of change) of values across the horizontal plane. Thus, you can do this for percent (or proportion) of impervious surface per pixel, or of percent canopy cover per pixel (e.g., from the National Land Cover Dataset).

Note, you can set the units for slope and aspect as either degrees or radians - degrees are typically more intuitive to most users, but radians are often used for some other calculations, and might sometimes be preferable. In particular, aspect is frequently converted to metrics of 'northness' and 'eastness', each ranging -1 - 1. This is because an aspect of 0 and 360 is the same (both face north), so northness and eastness break this down into two other components. Northness is calculated as the cosine of aspect (in radians), and eastness is calculated as the cosine of aspect (again, in radians). 1 represents East or North, and -1 represents West or South, respectively. With the resulting rasters from these terrain calculations, you can easily calculate the roughness metrics previously mentioned. *You do not need to do this for this lab, but it is important to be aware of these calculations if you are working with continuous raster datasets. If you want to calculate other surface metrics described in the McGarigal et al 2009 paper, let me know and we can try to do this in R.*

It is also be possible to calculate the surface metrics for individual patches (e.g., average roughness of a patch) - you would need to import a Patch ID output file from FRAGSTATS, convert the raster to polygons, and use the extract function to extract information from the raster layers to your polygons file. The functions you would need are 'rasterToPolygons' and 'extract'; to export the resulting dataset (polygons with the corresponding metrics in the attribute table), you should be able to use the 'writeOGR' function. We will cover some of these steps in a future lab.