Geospatial Portfolio

Sagert Sheets

MS Geography (GIScience),
University of New Mexico 2018

sagert.sheets@gmail.com
505.263.0538
https://**LinkedIn**.com/in/**sagertsheets**
https://**GitHub**.com/**sagert585**

CONTENTS

Master's Project
(Selected Elements)

Implementing the Distance Decay Function in an Enhanced Two-Step Floating Catchment Area Analysis and Interpreting Its Effect on Results: A Tool for Geographic Information Systems

Abstract: The Enhanced Two-Step Floating Catchment Area (E2SFCA) method of measuring potential spatial access to a resource is becoming increasingly common in research assessing the interaction between populations and health care providers. It uses a Gaussian decay function to weight access scores as distance from a resource (or from a population center) increases. However, the decay coefficient of the decay function and the resultant weights are prone to exaggerating or understating real-world access.

The current research introduces a tool for geographic information systems (GIS) software that allows the user to: 1) execute the E2SFCA on a variety of datasets; 2) choose between using a known weight or determining weights based on a coefficient; 3) output a ratio-based score that minimizes variations across coefficients; and 4) output documentation that records data inputs, the coefficient and weights used, and spatial access scores. This tool aims to increase the consistency, transparency, and understanding of the application of the E2SFCA method.

Note: The processes that went into preparing data for this analysis are described in the Appendix.

Elements:
Code Sample (Functions)
Mapped Output
Interface & Code Sample
Gaussian Decay Functions & Code Sample

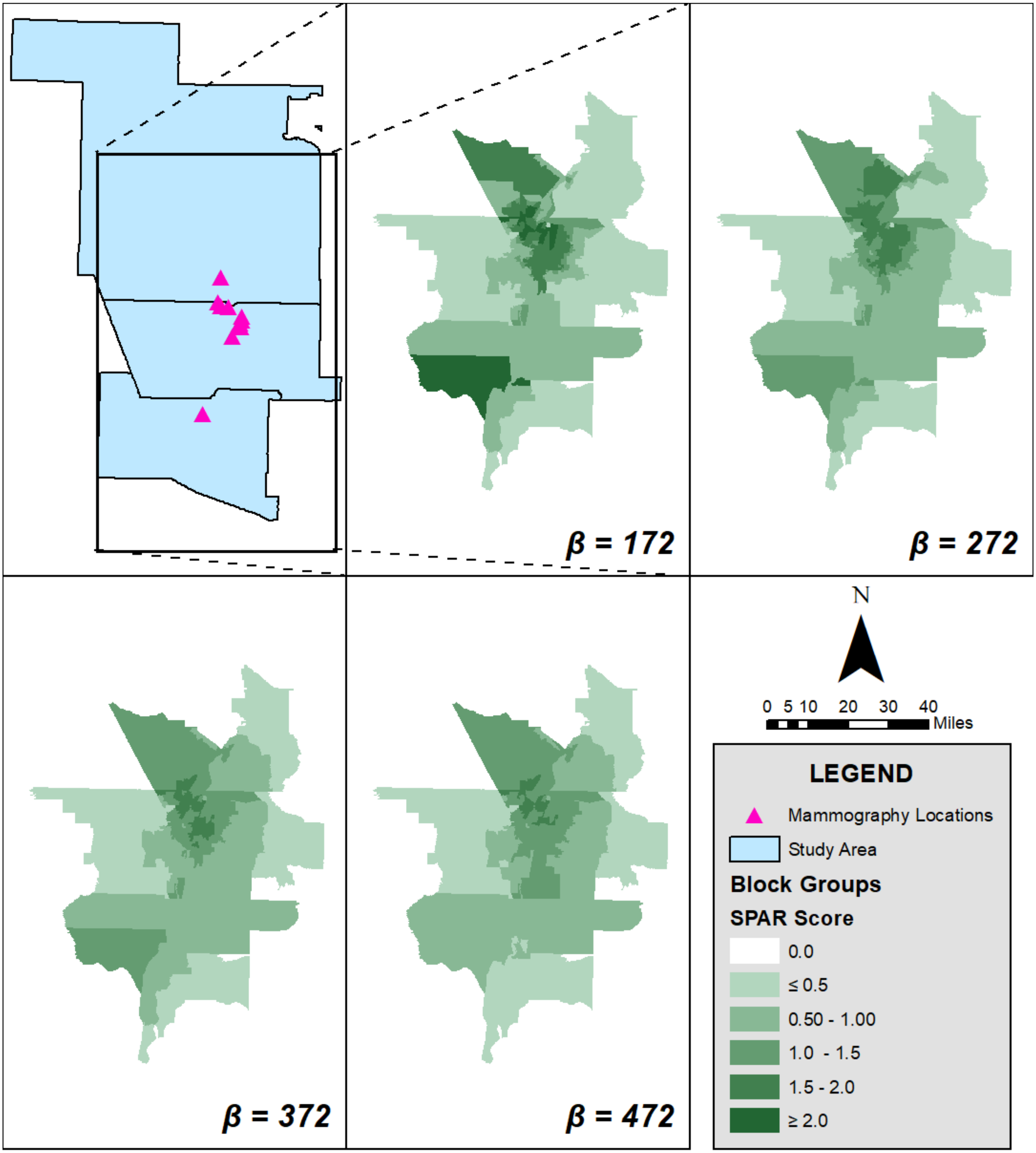
Code Sample:
(Functions)

A full version of this script is available (for review purposes only) on GitHub (https://www.github.com/sagert585).

```
# A function for calculating weights
def gaussianWeights(catch,coefficient):
    '''Weights may be an approximation'''
    weight = numpy.exp(-numpy.power(catch, 2.0)/coefficient)
    return weight

# A function for finding the coefficient, if necessary.
def gaussianSolve(catch,targetWeight):
    '''Coefficient may be an approximation'''
    coefficient = -numpy.power(catch, 2.0)/numpy.log(targetWeight)
    return coefficient
```

Mapped Output:

SPAR (spatial access ratio) is the ratio of the access score (measured as services per population) at a given population area to the mean of all access scores in the area of interest. A SPAR of 1.0 indicates average access.



Coordinate System: NAD 1983 UTM Zone 13N
Projection: Tansverse Mercator
Datum: North American 1983

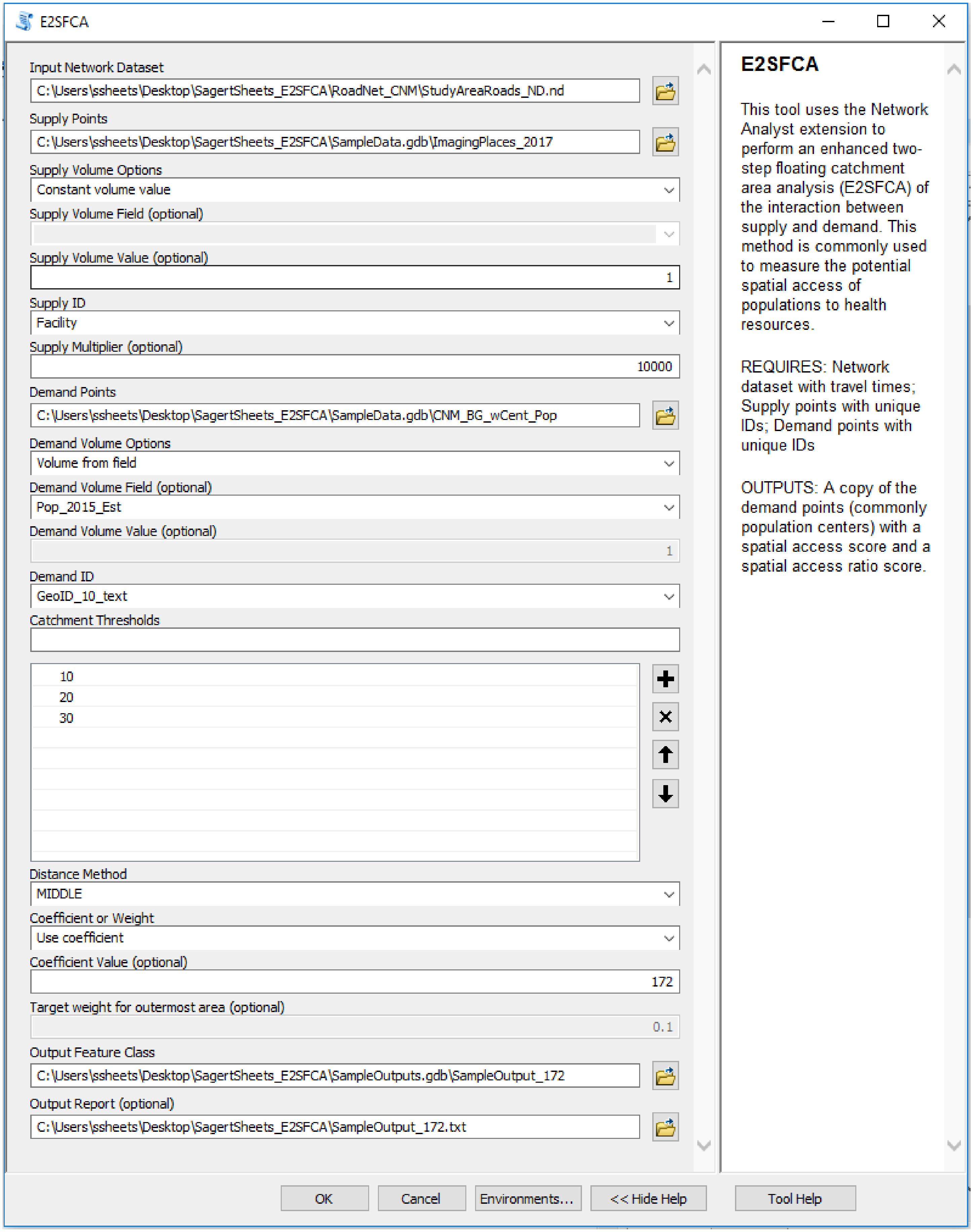
Units: Meter

False Easting: 500,000.0
False Northing: 0.0
Central Meridian: -105.0
Scale Factor: 0.9996
Latitude of Origin: 0.0

Interface & Code Sample:

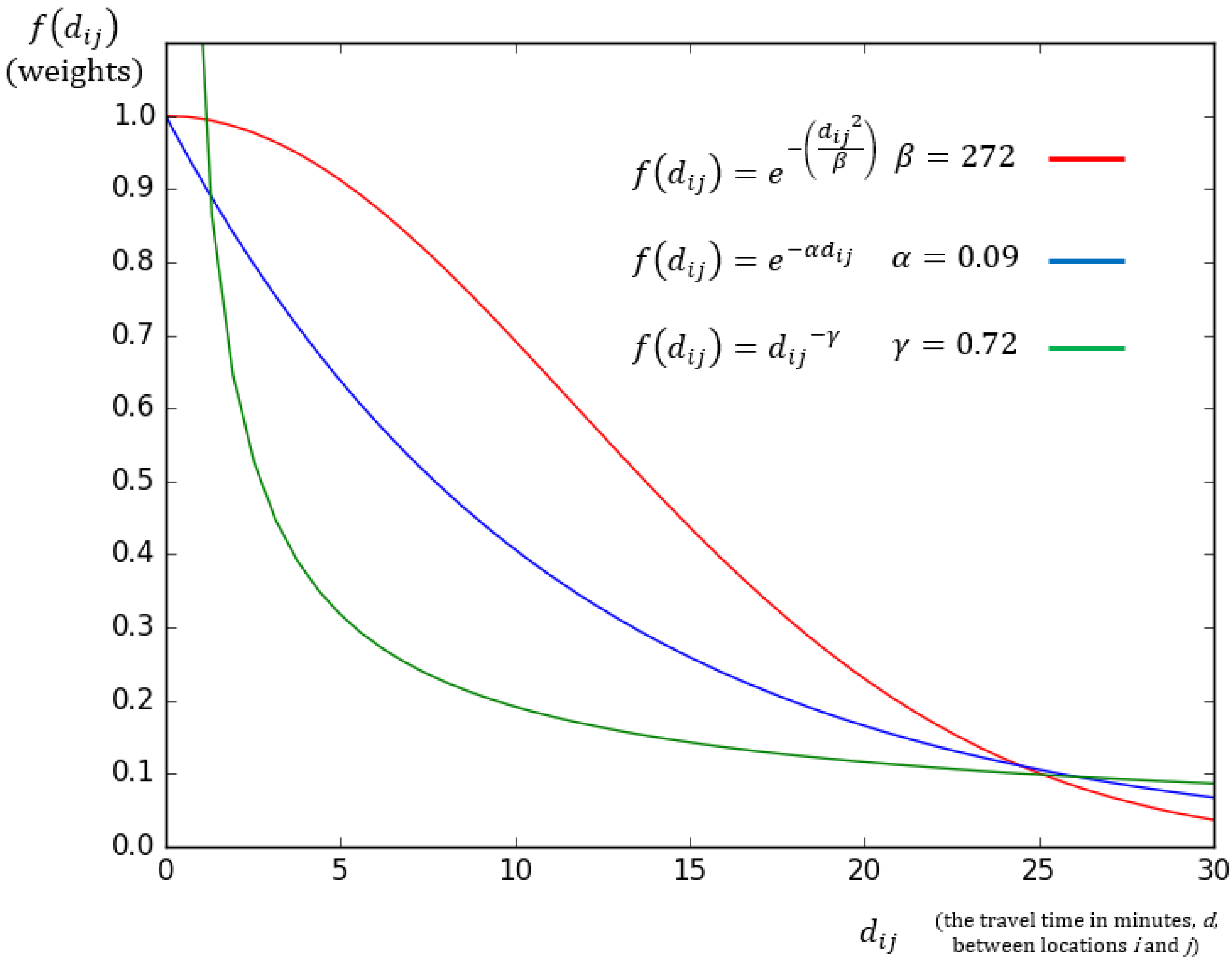
ArcPy's parameter retrieval functions allow parameter settings made by the user in the graphical interface to be processed by the script as variables.

```
'''-------------------------------------------------------------------------------
# Name:         E2SFCA_Version_1.0
# Purpose:      Implement the Enhanced Two-Step Floating Catchment Area method
# Author:       Sagert Sheets
# Created:      October 2017
#-----------------------------------------------------------------------------'''

# Import necessary modules
import numpy
import arcpy
from arcpy import env

arcpy.CheckOutExtension("Network")

# Main function
def e2sfca():

    arcpy.AddMessage("Retrieving parameters...")
    arcpy.SetProgressor("default", "Retrieving parameters...")
    # Get parameters from ArcGIS interface
    #Network Dataset - REQUIRED
    inputND = arcpy.GetParameterAsText(0)
    #Supply(Resource) Points - REQUIRED
    inputSupply = arcpy.GetParameterAsText(1)
    #Volume of Supply (e.g. beds per hospital) - REQUIRED
        #Choice
    supplyVolumeOpt = arcpy.GetParameterAsText(2)
        #Field
    supplyVolumeField = arcpy.GetParameterAsText(3)
        #Class value
    supplyVolumeValue = float(arcpy.GetParameter(4))
    #Unique ID - REQUIRED
    inputSupplyID = arcpy.GetParameterAsText(5)
    #Multiplier (in case, e.g. low facility to high pop. ratio) - OPTIONAL
    supplyMultiplier = float(arcpy.GetParameter(6))
    #Demand (Population) Points - REQUIRED
    inputDemand = arcpy.GetParameterAsText(7)
    #Volume of Demand (e.g. population) - REQUIRED
        #Choice
    demandVolumeOpt = arcpy.GetParameterAsText(8)
        #Field
    demandVolumeField = arcpy.GetParameterAsText(9)
        #Class value
    demandVolumeValue = float(arcpy.GetParameter(10))
    #Unique ID - REQUIRED
    inputDemandID = arcpy.GetParameterAsText(11)
    #Distance of regions - REQUIRED
    distList = arcpy.GetParameter(12)                       # Returns a list
    #How distance is used in Decay Function
    distanceMethod = arcpy.GetParameterAsText(13)
    #Use coefficient or calculate based on weight
```


E2SFCA
Input Network Dataset
C:\Users\ssheets\Desktop\SagertSheets_E2SFCA\RoadNet_CNM\StudyAreaRoads_ND.nd
Supply Points
C:\Users\ssheets\Desktop\SagertSheets_E2SFCA\SampleData.gdb\ImagingPlaces_2017
Supply Volume Options
Constant volume value
Supply Volume Field (optional)
Supply Volume Value (optional)
1
Supply ID
Facility
Supply Multiplier (optional)
10000
Demand Points
C:\Users\ssheets\Desktop\SagertSheets_E2SFCA\SampleData.gdb\CNM_BG_wCent_Pop
Demand Volume Options
Volume from field
Demand Volume Field (optional)
Pop_2015_Est
Demand Volume Value (optional)
1
Demand ID
GeoID_10_text
Catchment Thresholds
10
20
30
Distance Method
MIDDLE
Coefficient or Weight
Use coefficient
Coefficient Value (optional)
172
Target weight for outermost area (optional)
0.1
Output Feature Class
C:\Users\ssheets\Desktop\SagertSheets_E2SFCA\SampleOutputs.gdb\SampleOutput_172
Output Report (optional)
C:\Users\ssheets\Desktop\SagertSheets_E2SFCA\SampleOutput_172.txt
OK
Cancel
Environments...
<< Hide Help
Tool Help
E2SFCA
This tool uses the Network Analyst extension to perform an enhanced two-step floating catchment area analysis (E2SFCA) of the interaction between supply and demand. This method is commonly used to measure the potential spatial access of populations to health resources.
REQUIRES: Network dataset with travel times; Supply points with unique IDs; Demand points with unique IDs
OUTPUTS: A copy of the demand points (commonly population centers) with a spatial access score and a spatial access ratio score.

Guassian Decay Curves & Code Sample:

The curves below were plotted with Python's numpy and matplotlib packages. Labels were added in Microsoft Word.




```
#---------------------------------------------------
# Name:        Decay_curve_scripts
# Purpose:     The Project
# Author:      Sagert Sheets
# Created:     11/07/2017
#---------------------------------------------------

import numpy
import matplotlib
import matplotlib.pyplot as plt

#Variables for plots
d = numpy.linspace(0, 30)
dx = numpy.linspace(0.1, 30)
g = numpy.exp(-(d**2)/272.0)
i = numpy.power(d, -.72)
e = numpy.exp(-d*(0.09))

plt.plot(d, g, "r", d, e, "b", dx, i, "g")
plt.ylim(0, 1.1)
plt.yticks(numpy.arange(0,1.1,0.1))
plt.show()
```

**A Comparison of Models for Measuring
Potential Spatial Access to Health Resources**

Description: Following the completion and academic defense of the E2SFCA tool for my master's project, I created a second tool based on the code of the original. This represented a significant achievement in the following ways:

1) The second tool is distinct from the original and based on a related model.
2) Increased the number of reusable functions, reducing the length of the code.
3) Decreased the parameters the user is required to input.
4) The code can be used to update the original tool and create new tools based on related models.

Note: The processes that went into preparing data for this analysis are described in the Appendix.

Code Samples:

A full version of this script is available (for review purposes only) on GitHub (https://www.github.com/sagert585).

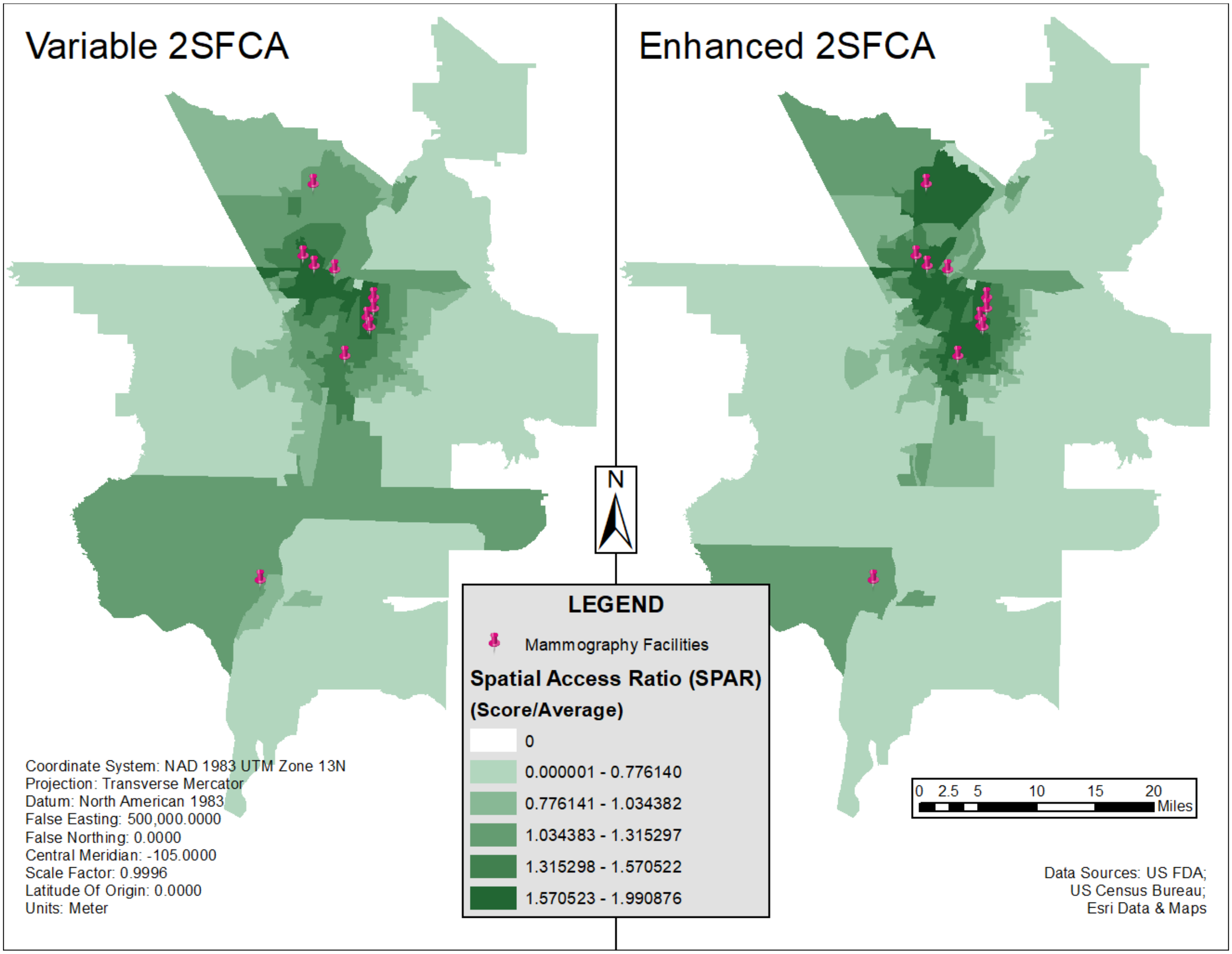
Functions:

```
# A function for updating weight values after each step
def writeWeights(ODMatrixTable, distance, coefficient):
    with arcpy.da.UpdateCursor(ODMatrixTable, [accumMinutes,
                                weightField]) as weightWriter:
        for row in weightWriter:
            if (0 < row[0] <= distance) == True:
                row[1] = gaussianWeights(row[0], coefficient)
            else:
                row[1] = 0
            weightWriter.updateRow(row)
    return

# A function for applying weights to volume values
def applyWeights(ODMatrixTable, inField, outField):
    with arcpy.da.UpdateCursor(ODMatrixTable, [inField, weightField,
                                outField]) as weightApplier:
        for row in weightApplier:
            row[2] = row[0] * row[1]
            weightApplier.updateRow(row)
    return

# A function for implementing the user's choice for volume
def writeVolume(inTable, newField, volumeValue):
    arcpy.AddField_management(inTable, newField, "DOUBLE")
    with arcpy.da.UpdateCursor(inTable, newField) as volumeCursor:
        for row in volumeCursor:
            row[0] = volumeValue
            volumeCursor.updateRow(row)
    return newField
```

Error Handling:

```
except:
    ODMatrixError = arcpy.GetMessages(2)
    arcpy.AddMessage(ODMatrixError)
    arcpy.AddError("The second O-D Matrix could not be completed.")
finally:
    arcpy.CheckInExtension("Network")
```

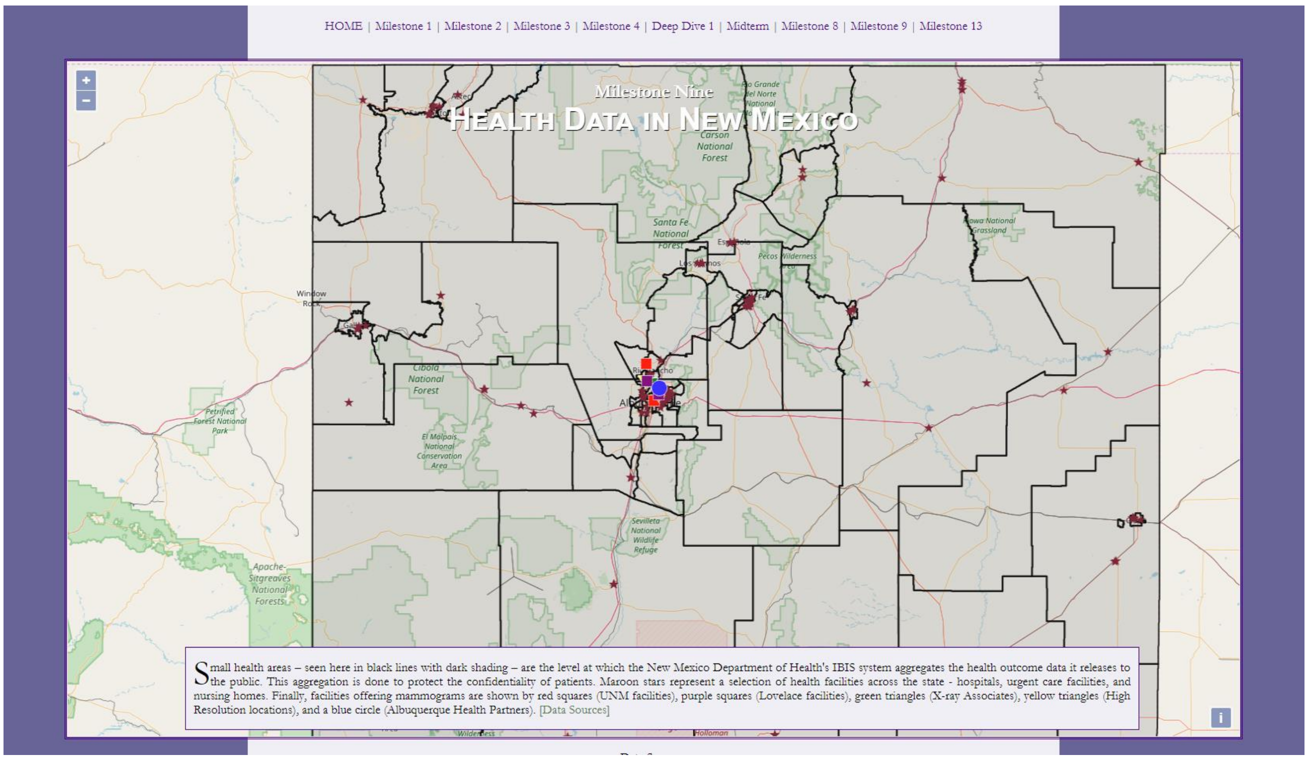
Mapped Outputs:

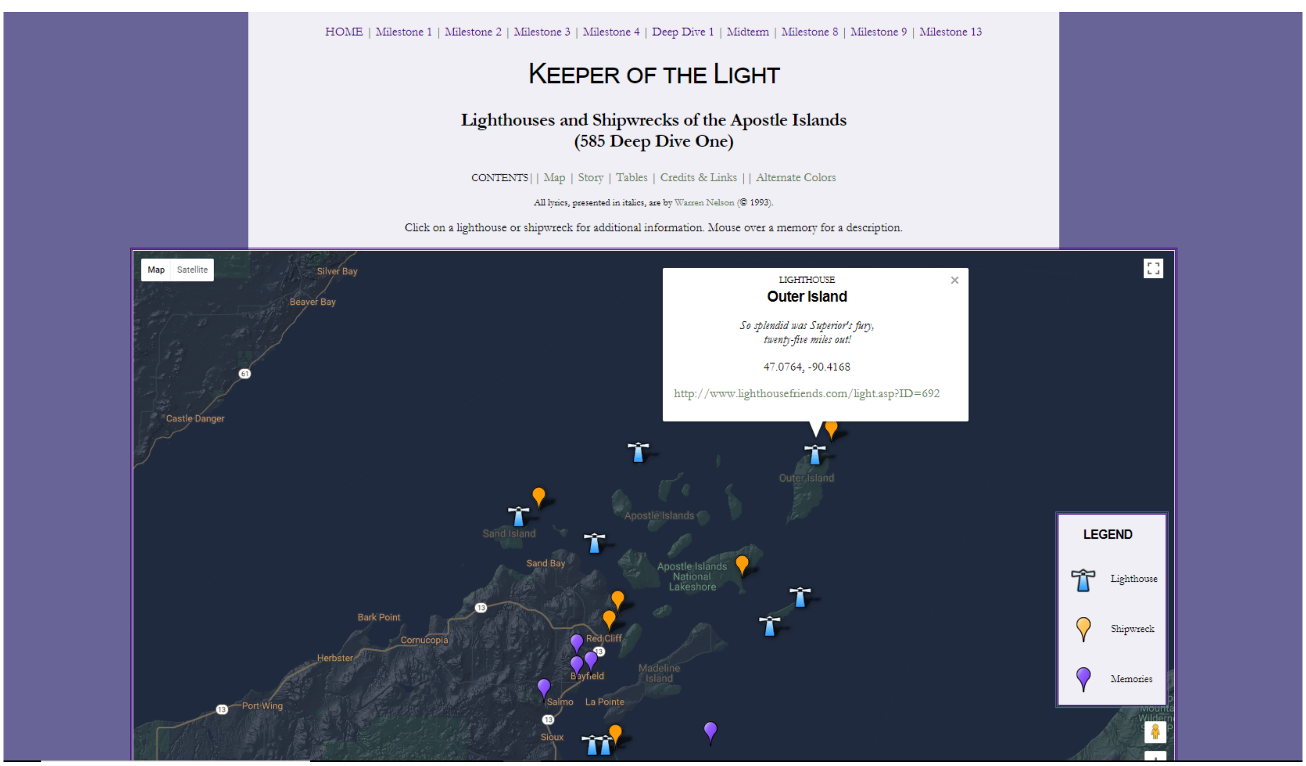


Internet Mapping
(Geography 585)

Description: Coursework focused on creating a portfolio of online mapping applications, presented on pages written in HTML and styled with CSS (cascading style sheets). Interactive maps incorporated a combination of the Google Maps API, Google Fusion Tables, OpenLayers, GeoServer, and javascript.

GitHub: https://github.com/sagert585
Published pages: https://sagert585.github.io/portfolio/





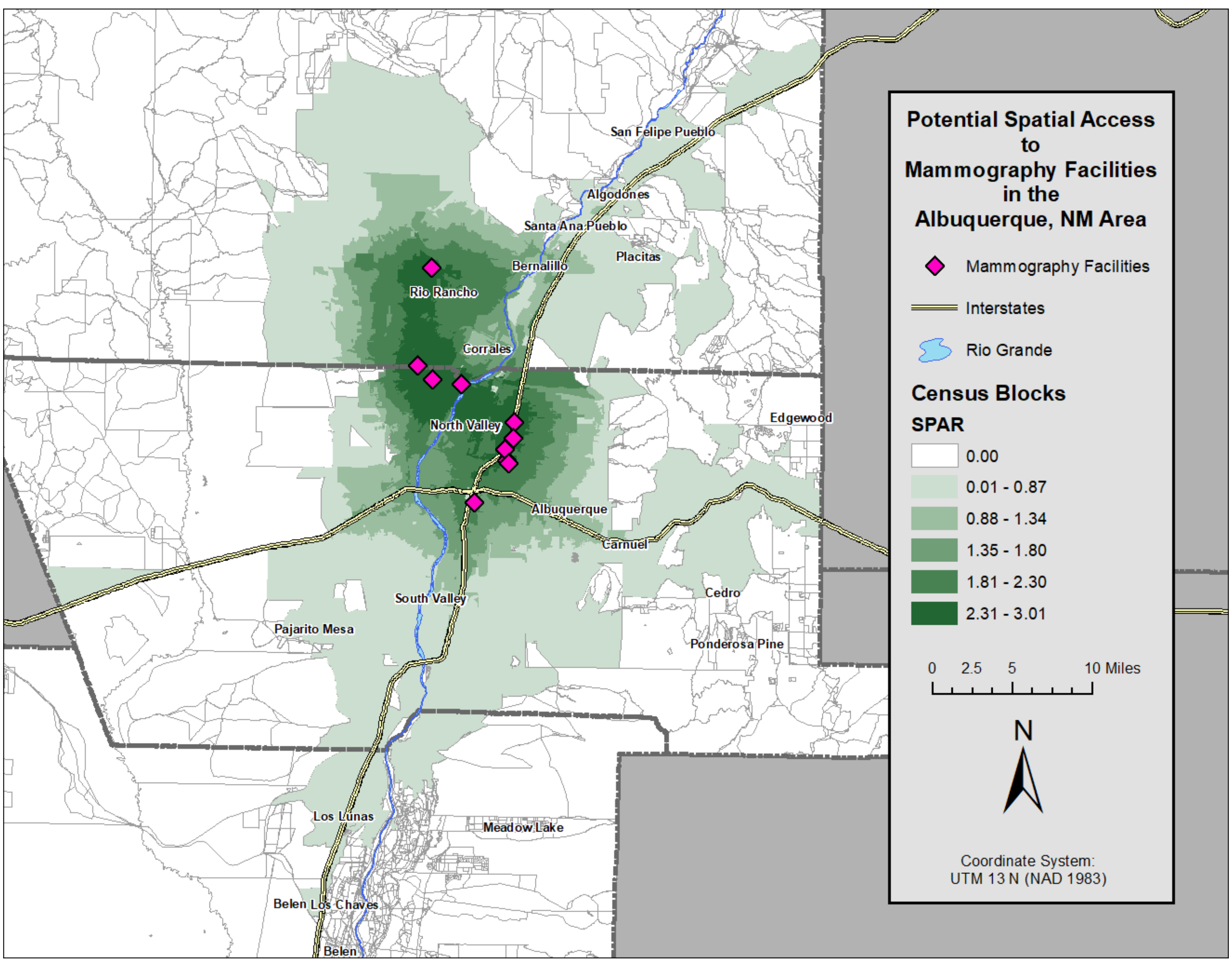
Description: Before beginning work on my master's project, this class project presented a chance to familiarize myself with the GIS processes involved in the E2SFCA model.

Abstract: This project uses the enhanced two-step floating catchment area (E2SFCA) method to assess potential spatial accessibility to mammography facilities in the Albuquerque, NM, area. Mammography is an important part of breast cancer screening and early detection, which in turn are key factors in increasing survival rates and decreasing mortality rates. The E2SFCA method uses GIS software, network analysis, and points of supply and demand to measure accessibility to a resource. Weights are used in the analysis to decrease access scores as travel time from a resource increases, and accessibility scores are ratioed against their average to create a spatial access ratio (SPAR). The results are mapped and spatial accessibility in the area of interest is assessed. Limitations and opportunities for future research are considered.

Notes: This project was completed in Fall 2016, before an 11th facility offering mammography services was opened in the study area (in Los Lunas). Data sources and some of the processes involved in preparing this data are described in the Appendix.

Mapped output:

Spatial access indices (SPAI) and spatial access ratios (SPAR) were compared for both census blocks (finer scale) and census block groups (aggregated from blocks). The following map shows SPAR scores for blocks.

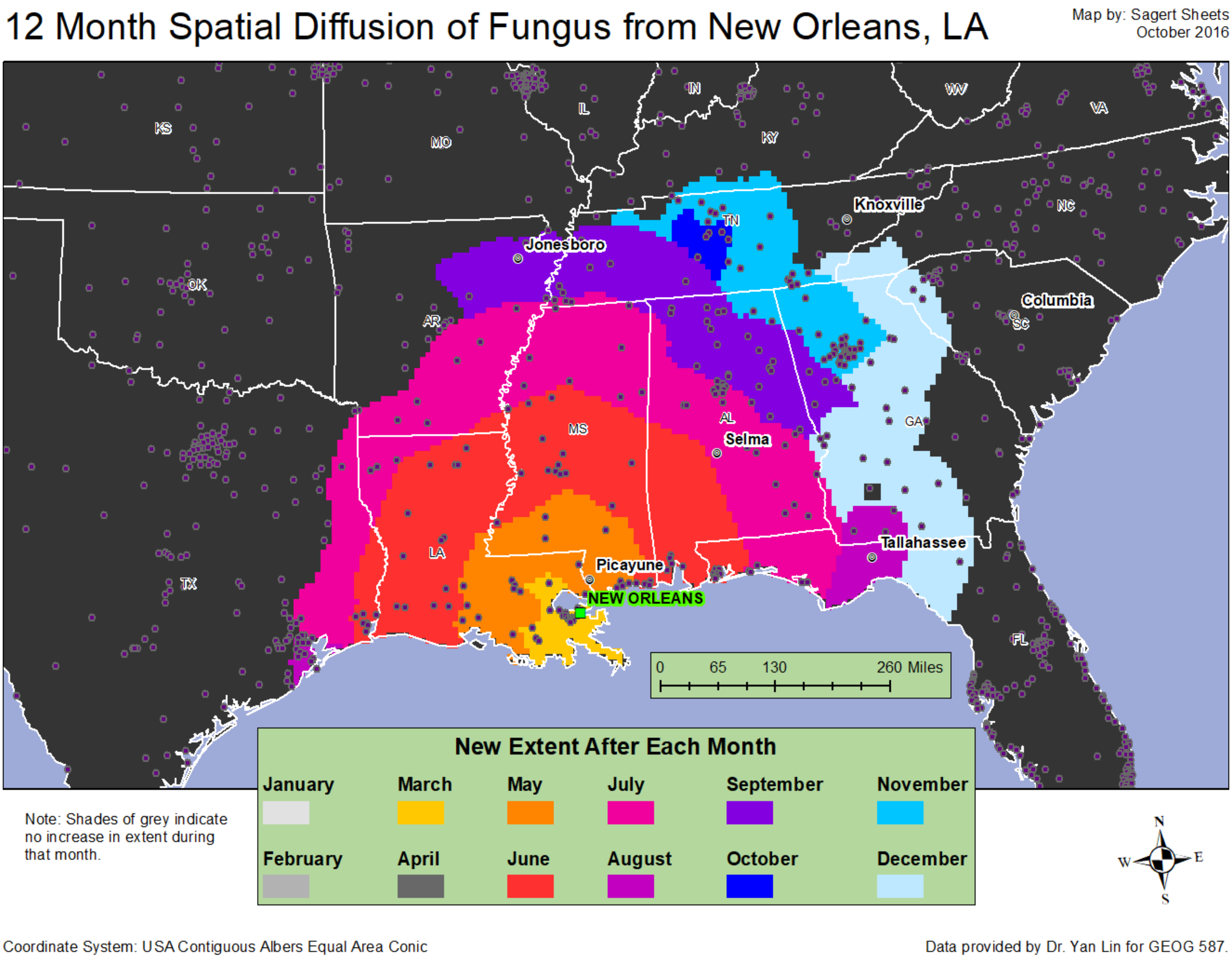


Lab Exercises

Description: These maps were created for lab exercises in GIS courses. The goals were set and the data was provided by an instructor. I completed the exercises and created the final outputs.

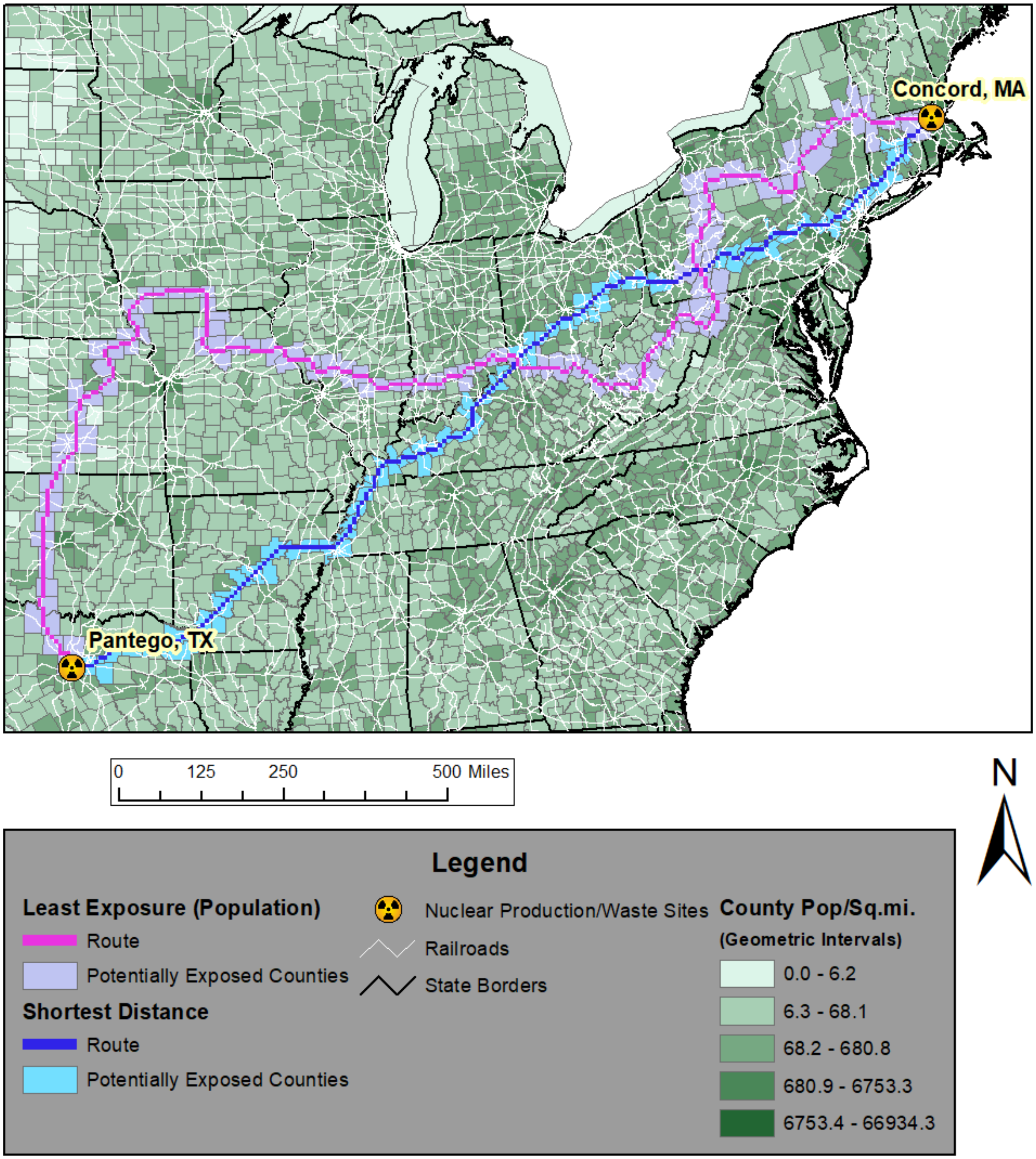
Exercises:
Spatial Diffusion Modeling
Cost Path Analysis
Hydrology and Watershed Modeling

Spatial Diffusion Modeling:



Cost Path Analysis:

Potential Nuclear Material Transport Routes Between Pantego, TX, and Concord, MA



Map by Sagert Sheets

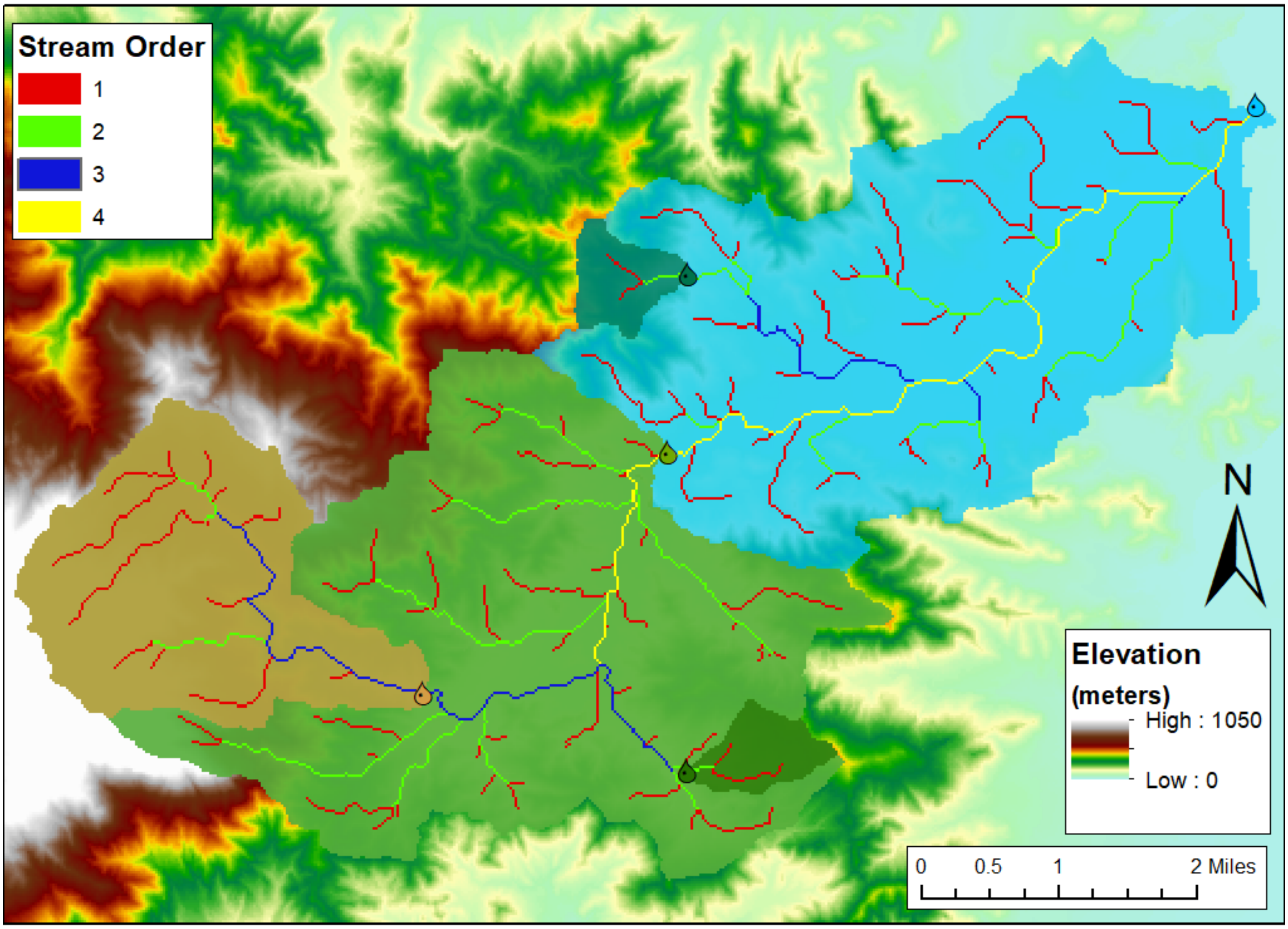
Data Sources:
Lab data provided by Dr. Yan Lin for GEOG 587;
Esri Maps & Data; US Census Bureau.

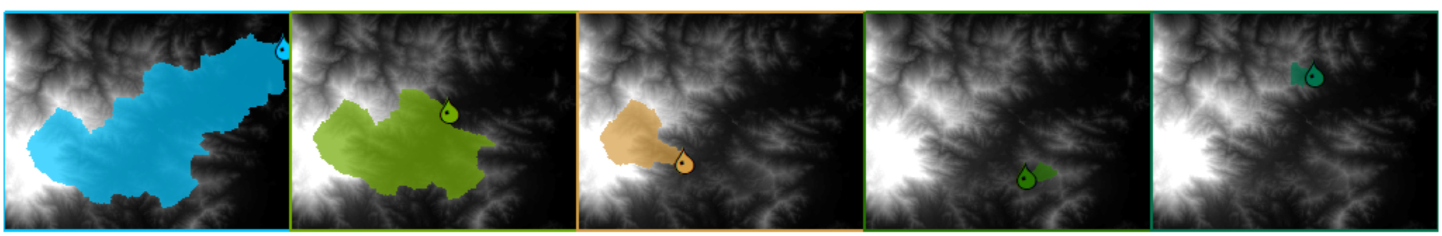
Coordinate System:
USA Contiguous Lambert
Conformal Conic

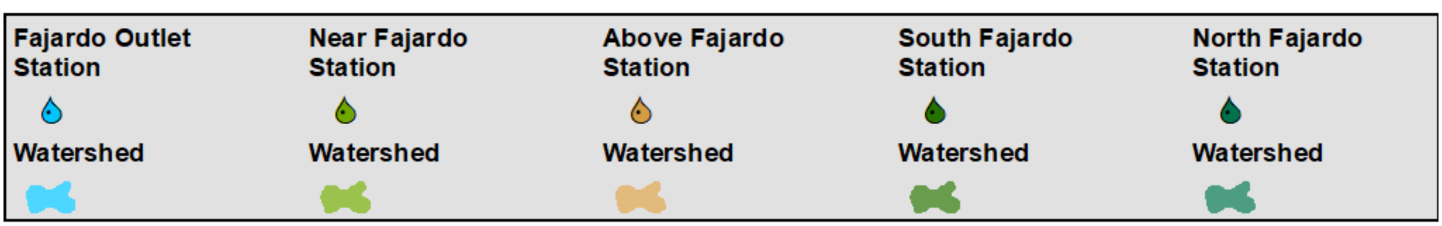
Hydrology and Watershed Modeling:

Rio Fajardo Watershed and Sub-watersheds

Map by: Sagert Sheets
October 2016







Note: For the watershed modeling and analysis, a "filled" elevation model was used.
The above map shows filled elevation data.

Coordinate System: NAD 1983 UTM Zone 20N

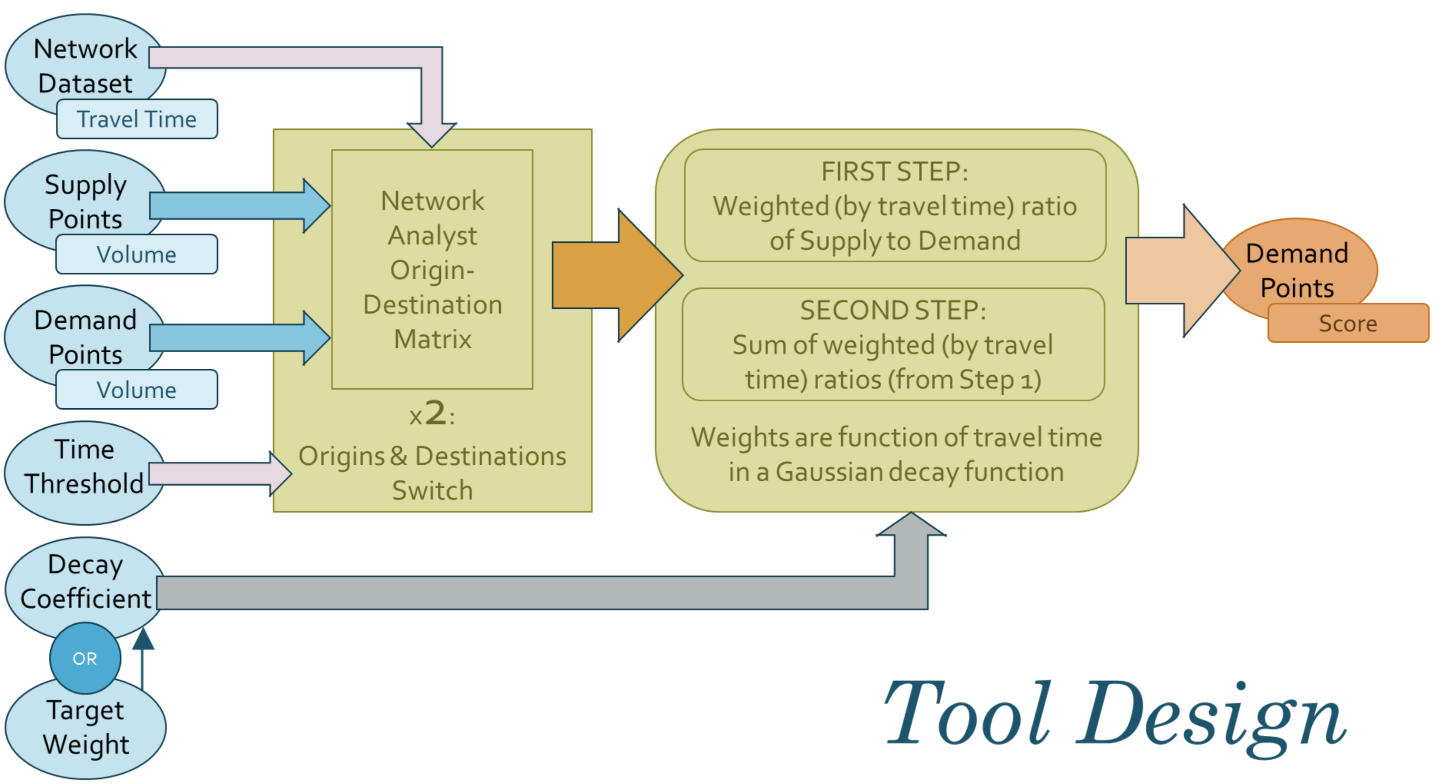
Data Source: Provided by Dr. Lin for GEOG 587, Fall 2016.

Appendix:
Data, Tool Design, Processes

Data Sources:
(unless otherwise noted on a map)

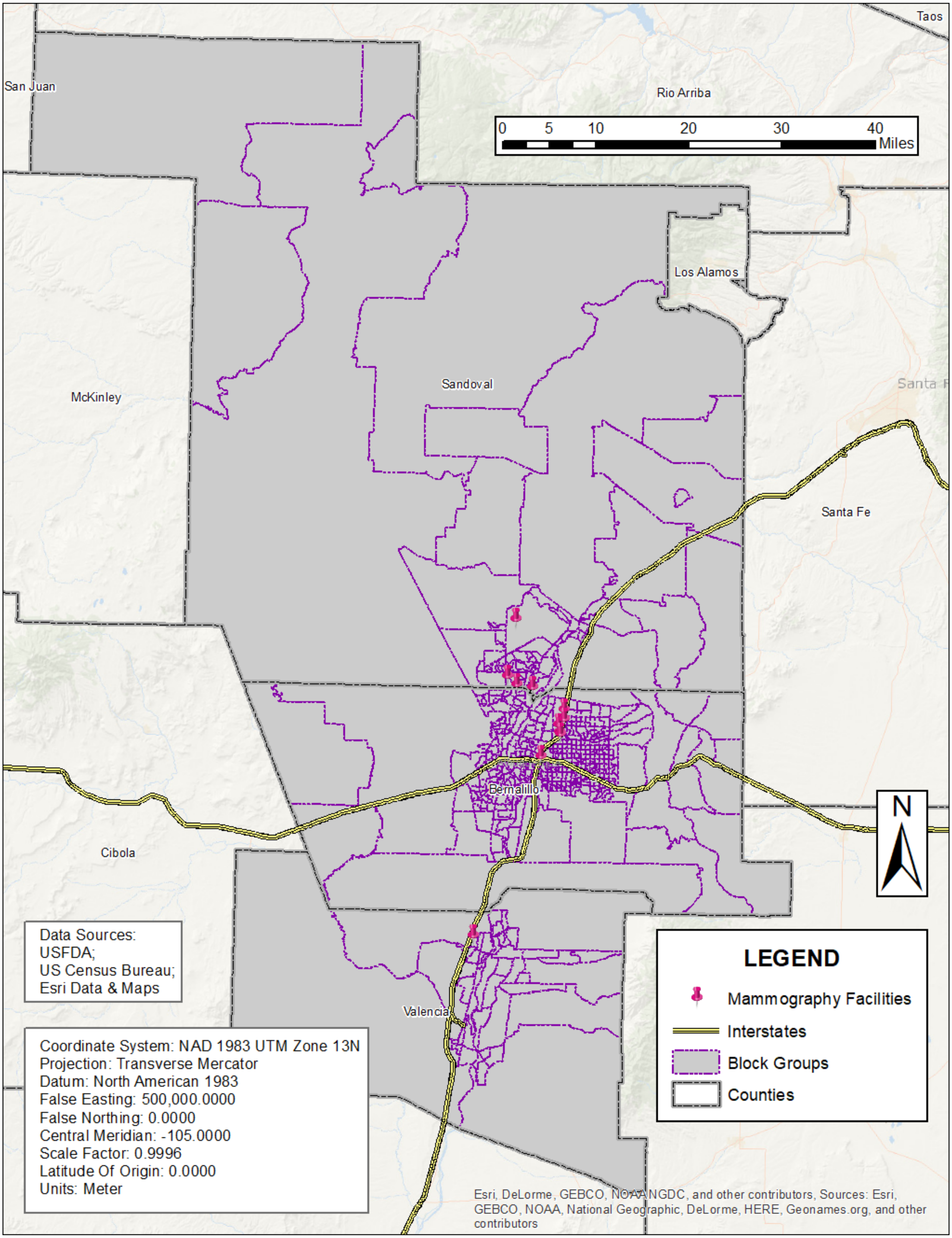
- Esri Data and Maps
- US Census Bureau
- USFDA Mammography Facility Database
- UNM Resource Geographic Information System

Tool Design:
The chart below is a summary of the inputs, analysis steps, and output of two-step floating catchment area (2SFCA) models. The key distinguishing factor is often how weights are determined. For example, in the E2SFCA method, weights are assigned to a sub-zone of the catchment and then to supply or demand points based on the sub-zone in which they are located.



Data and Processes:
Described on the following two pages, with mapped examples.

Study Area: Sandoval, Bernalillo, and Valencia Counties in Central New Mexico served as the study area for much of the research in this portfolio. Counties are divided into block groups, and block groups are made up of census block enumeration units (blocks). The interstates (pictured) and road networks used for impedance-based network analysis are subsets of Esri Data and Maps' North America Detailed Streets dataset.



Aggregation and Centroids: Blocks were joined to tabular population counts by unique geo-identifier and then aggregated into block groups, while the population-weighted centroid of each block group was calculated based on the population of the component blocks. These centroid points were located on the network for analysis.