2023

EEMS Pro Manual

Environmental Evaluation Modeling System
Manual Version 1.0.1

An introduction to the principles of fuzzy logic, and an instruction manual for the ArcGIS implementation of the Environmental Evaluation Modeling System (EEMS).

Mike Gough;Tim Sheehan;James Strittholt
Conservation Biology Institute
1/16/2023

Table of Contents

I. INTRODUCTION

1.1 Purpose and Contents

This document serves to introduce the user to the principles of fuzzy logic and to provide instruction on the use of EEMS Pro – the ArcGIS implementation of the EEMS Fuzzy Logic Modeling System (Sheehan and Gough, 2016).

How this document is organized

This manual is divided into three sections:

Section I provides an introduction to the EEMS Pro manual and lists the software and technical requirements needed to run EEMS Pro.

Section II introduces the user to principles of fuzzy logic, and provides a high level overview of the EEMS Modeling Framework.

Section III begins by describing the EEMS modeling process – as implemented from within the ArcGIS Model Builder environment – and concludes with a tutorial on building and running an EEMS model using a subset of the tools in the EEMS Pro Toolbox.

It is recommended that the user read through Sections I, II, and III in order to obtain a conceptual understanding of the EEMS modeling process prior to working through the tutorial at the end of section III.

1.2 Software and Technical Requirements

1. **Windows 7 or greater**

2. **The EEMS Pro Folder**

    If you do not already have the EEMS Pro folder, it can be downloaded from the URL below. (Click *Code -> Download Zip* or clone the repository to your local machine).

    https://github.com/consbio/eems-pro

    Where should I keep the EEMS Pro Folder?

    You can keep the EEMS Pro folder in a known location on a local or network drive where it can be accessed and used on a variety of different projects and studies. For example:

    - `D:\GIS\Tools\EEMS_Pro`
    - `\\Server\GIS\Tools\EEMS_Pro`

    Alternatively, you may choose to keep a copy of it in multiple separate workspaces (folders) corresponding to different projects, studies, etc., in order to make your workspaces portable.

    The choice is yours. There are advantages and disadvantages to each approach. However a discussion of this topic goes beyond the purview of this manual.

3. **ArcGIS Pro or ArcGIS Desktop (v10.6 or greater)**

    While earlier versions of ArcGIS Desktop may work, they have not been tested.

4. **MPilot (v1.2.5 or greater)**

    MPilot is a plugin-based environmental modeling framework written in Python:

    https://github.com/consbio/mpilot

    It must be installed in order for EEMS Pro to work. To install MPilot, please follow the instructions below corresponding to the version of ArcGIS you have installed and plan on using with EEMS Pro.

    ! IMPORTANT: Please close all ArcGIS products before proceeding with the MPilot installation.

Installing MPilot for ArcGIS Pro:

To use EEMS Pro in ArcGIS Pro you must install MPilot into a cloned conda environment (more information on conda environments in ArcGIS Pro can be found on the ArcGIS Pro website).

If you already have a cloned environment in ArcGIS Pro in which you'd like to install MPilot, you can simply click the **ArcGIS -> Python Command Prompt** shortcut from the Windows Start Menu, activate the cloned environment (`activate <environment name>`), then install MPilot using pip (`pip install mpilot`).

If you do not already have a cloned environment or you would like to install MPilot into a new cloned environment, or, **if you are unsure**, follow the instructions below:

i.   Click the Windows Start Button in the bottom left hand corner of the Windows Taskbar.

ii.   Click **ArcGIS -> Python Command Prompt**

iii.   Clone the default ArcGIS Pro environment using the command below. This will take a few minutes. Note that you can replace the cloned environment name (`arcgispro-py3-clone-mpilot-eems`) with a name of your choosing:

```
conda create --clone arcgispro-py3 --name arcgispro-py3-clone-mpilot-eems
```

iv.   Activate the new environment:
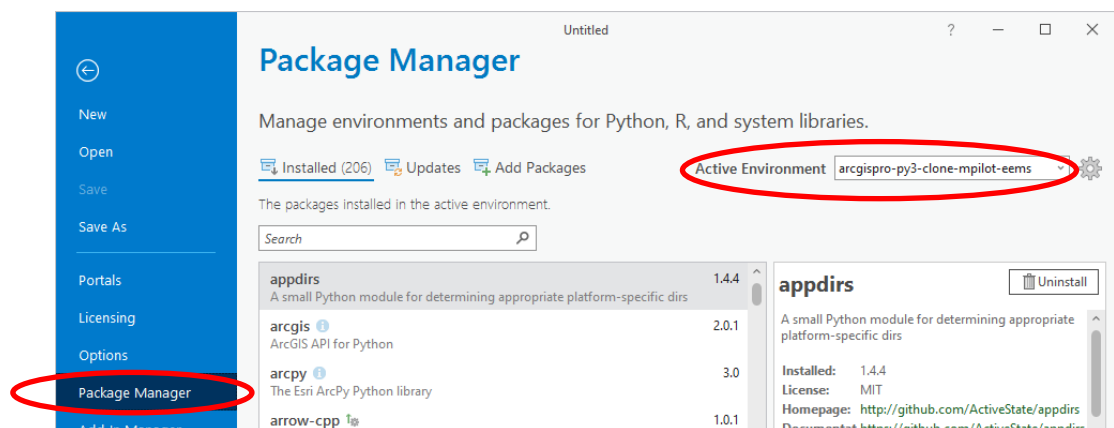
```
activate arcgispro-py3-clone-mpilot-eems
```

v.   Install MPilot using pip:

```
pip install mpilot
```

vi.   Once MPilot has finished installing, you can close the Python Command Prompt.

vii.   Now, when you are ready to build an EEMS Model, start ArcGIS Pro and switch to the new cloned environment from the **Package Manager** -> **Active Environment** dropdown menu.

Installing MPilot for ArcGIS Desktop:

    i.   Open up a command prompt (in Windows 10, you can do this by typing "cmd" into the Search box in the bottom left hand corner of your screen).

    ii.   Type in the following command, replacing "X" with the version of ArcGIS Desktop you have installed:

```
cd C:\Python27\ArcGIS10.X\Scripts
```

    iii.   Install MPilot:

```
 pip install mpilot
```

In addition to the software requirements listed above, the user should have formal GIS training, and be experienced with ArcGIS and ArcGIS Model Builder.

II. OVERVIEW OF EEMS & FUZZY LOGIC

2.1 About EEMS

The Environmental Evaluation Modeling System (EEMS) is an implementation of an evaluative logic modeling system. EEMS was developed by Tim Sheehan, an ecological modeler at the Conservation Biology Institute, and has been applied by CBI scientists in a range of ecological evaluations. In the Tehachapi and Southern Sierra regions of California, a model incorporating data for habitat presence, habitat linkage, and disturbance was used to find areas of high ecological value and to provide guidance for reserve design and to inform siting wind energy. For the Bureau of Land Management Rapid Ecological Assessments of the Sonoran Desert and Colorado Plateau ecoregions, several EEMS models were developed and used to evaluate a variety of current and projected ecological metrics.

2.2 An Overview of Fuzzy Logic

In the discipline of GIS evaluative modeling, a logic model is a cognitive map (Jensen et al. 2009) that presents networks of various spatial data components and their logical relationships to explain the process used to evaluate a complex topic such as terrestrial intactness.

EEMS logic models rely solely on spatial data layers arranged in a tree-based, hierarchical fashion (Figure 1-1, Figure 1-2) to answer a primary question. Data and analysis flow from the bottom up. At the bottom of the tree, the *leaf nodes* represent the initial data inputs. Data from the leaves are combined recursively, moving up *branches*, until the single *root node*, representing the answer to the primary question, is reached. In this example, the evaluation pertains to the level of terrestrial landscape intactness.

Unlike conventional GIS applications that use Boolean logic (True/False or 1/0) or scored input layers, evaluative logic models rely on fuzzy logic. Simply put, fuzzy logic allows the user to assign shades of gray to thoughts and ideas rather than being restricted to black (false) and white (true) determinations. All data inputs (regardless of the type—ordinal, nominal, or continuous) are converted into fuzzy values between -1 (false) and +1 (true) up to six decimal places. A user defined function converts inputs from original values (often referred to as *raw values* or values in *raw space)* to fuzzy values (values in fuzzy space). Values in raw space may be combined using mathematical operators (e.g. sum, weighted mean) before being translated into fuzzy space. Values in fuzzy space are combined using fuzzy logic operators (e.g. AND, OR). There are many advantages of this modeling approach: (1) it is highly interactive and flexible; (2) it is easy to visualize thought processes; (3) the logic components are modular making it easy to include or exclude pieces of the logic design; (4) the logic can be managed using a number of different mechanisms; and (5) numerous, diverse topics can be included into a single integrated analysis. Raw, or non-fuzzy, spatial data source inputs (boxes in the example) are populated by one or more GIS data layers (indicated by the stack of gray files). Moving up the diagram, these data are arranged and analyzed to form intermediate map products (purple boxes), which are then arranged and analyzed to generate the final results (green box). One way the user controls the logic of the information is the

arrangement of the various data inputs and intermediate products—the higher up in the diagram, the greater the influence on the final result.
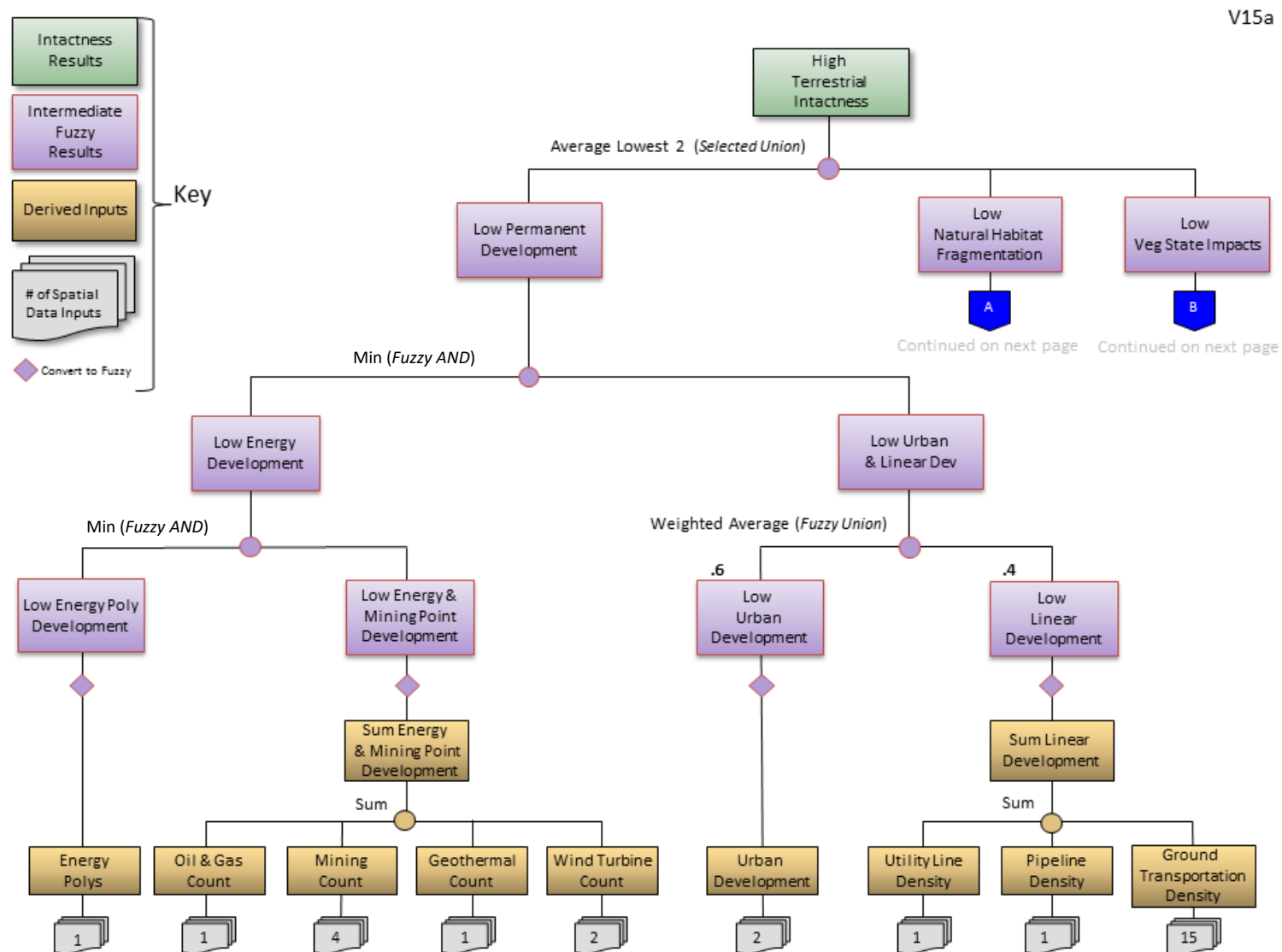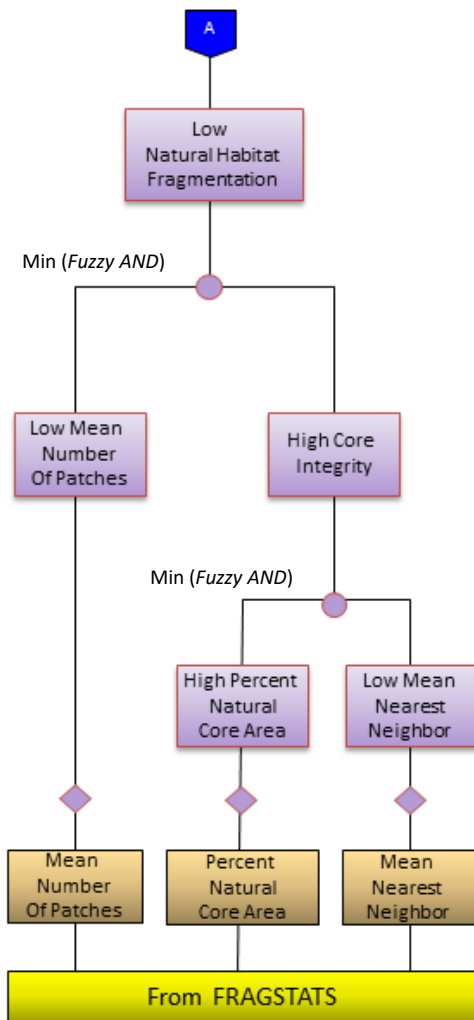
EEMS Pro Manual, 2023 (VERSION 1.0.1)

Figure 1-1. Logic model for terrestrial landscape intactness (Page 1 of 2)

EEMS Pro Manual, 2023 (VERSION 1.0.1)

A

Low
Natural Habitat
Fragmentation

Min (*Fuzzy AND*)

Low Mean
Number
Of Patches

High Core
Integrity

Min (*Fuzzy AND*)

High Percent
Natural
Core Area

Low Mean
Nearest
Neighbor

Mean
Number
Of Patches

Percent
Natural
Core Area

Mean
Nearest
Neighbor

From FRAGSTATS

B

Low
Veg State Impacts

Sum
Veg State Impacts

Weighted Sum

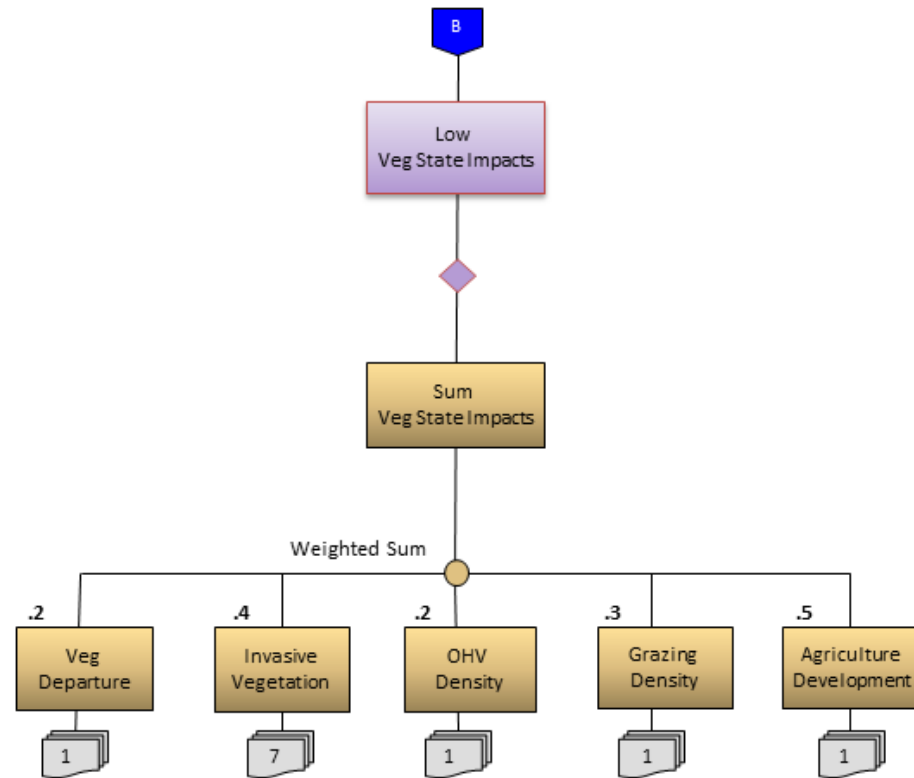| .2	.4	.2	.3	.5
Veg Departure	Invasive Vegetation	OHV Density	Grazing Density	Agriculture Development
1	7	1	1	1

Figure 1-2. Logic model for terrestrial landscape intactness (Page 2 of 2)

Using fuzzy logic as the core modeling principle, evaluative logic model performance is achieved in several ways. For every spatial data input, the user determines how to assign the range of values along a truth continuum. For example, when trying to determine and map the most suitable habitat from the standpoint of road density for wildlife—the greater the road density, the greater is the risk to wildlife through habitat degradation and direct mortality. In our example, road density ranges from 0 km/km^2 to 24.5 km/km^2. To assign a fuzzy logic continuum for this range of values, one could assign a -1 to the high value (this value is totally harmful for wildlife or false) and a +1 to the lowest value (this value is totally beneficial for wildlife, or true; red line in Figure 1-3). However, mountain lion research has shown that mountain lion populations have a low probability of persistence in areas with road densities > 0.6 km/km2 (Van Dyke et al. 1986). A more meaningful alternative then for setting fuzzy thresholds for this parameter would be that a road density of > 0.6 km/km^2 is totally false (-1) and 0 remains totally true (+1, green line in Figure 1-3). Of course, not all wildlife species have the same sensitivity to roads, but this example illustrates how the logic in the model can be altered for known thresholds.
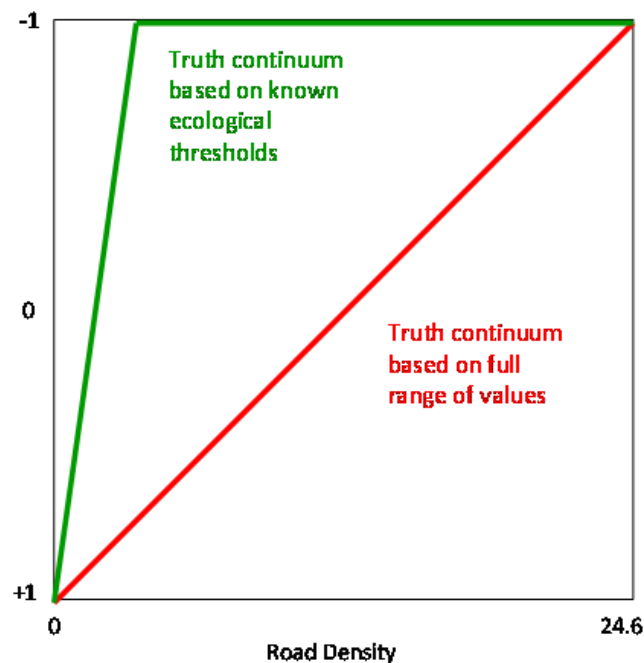
Figure 1-3. Diagram of two treatments of road density in fuzzy logic modeling illustrating important model control options, one based on a full range of values (red line) and the other based on a known threshold for road density (> 0.60 km/km^2 is totally false [-1], green line).

2.3 EEMS Tools (Operators)

EEMS can be implemented from within an ArcGIS Model Builder Model, or as a standalone program. The tables below list the EEMS tools available within the EEMS Pro Toolbox as well as the corresponding EEMS command that the tool references. The last column describes the functionality of the tool.

The first table below lists the core set of EEMS tools which perform functions specific to the EEMS framework.

ARCGIS TOOL	EEMS COMMAND	DESCRIPTION
EEMS MODEL INITIALIZE	(N/A, applies only to Arc version)	Specifies the paths to the files needed to run EEMS (input reporting units & EEMS command file). Clears the contents of the EEMS command file if it exists
EEMS MODEL LOGIC CHECK	(N/A, applies only to Arc version)	Check for logical consistencies within the EEMS command file.
EEMS MODEL RUN	(N/A, applies only to Arc version)	Executes the commands in the EEMS command file on the input reporting units. Stores the results in the output reporting units.
EEMS READ	EEMSRead	Reads a variable from the input reporting units.

EEMS also comes with the set of logic and mathematical operators used to process or combine input data or derived data. These operators are described in the table below. The operators used and the order in which they are executed will depend on the question under evaluation. Note that some of the operators are intended to operate on raw values while others are intended to operate on fuzzy values.

ARCGIS TOOL	EEMS COMMAND	INPUT DATA	DESCRIPTION
CONVERT FROM FUZZY	CvtFromFuzzy	Fuzzy	Converts input fuzzy values into non-fuzzy values using linear interpolation
CONVERT TO BINARY	CvtToBinary	Raw	Converts input values into binary 0 or 1 based on threshold. Direction = LowToHigh for values below threshold to be false and above to be true. Direction = HighToLow for values below threshold to be true and above to be false.
CONVERT TO FUZZY	CvtToFuzzy	Raw	Converts input values into fuzzy values using linear interpolation.
CONVERT TO FUZZY CATEGORY	CvtToFuzzyCat	Raw	Converts integer input values into fuzzy based on user specification.
CONVERT TO FUZZY CURVE	CvtToFuzzyCurve	Raw	Converts input values into fuzzy based on user-defined curve.

CONVERT TO FUZZY MEAN TO MID	CvtToFuzzyMeanToMid	Raw	Converts input values into fuzzy based on the mean to mid-value normalization method. The mean is assigned the middle fuzzy value from the provided list. The mean of all the values below the mean (low mean) is assigned the low mid fuzzy value from the list. The same is done for the high mean. The remaining values are scaled linearly between these five points (-1, -0.5, 0, 0.5, and 1).
CONVERT TO FUZZY CURVE Z SCORE	CvtToFuzzyCurveZScore	Raw	Converts input values into fuzzy based on user-defined curve.
CONVERT TO FUZZY Z SCORE	CvtToFuzzyZScore	Raw	Converts input values into fuzzy values using linear interpolation based on Z Score.
FUZZY AND	FuzzyAnd	Fuzzy	Takes the fuzzy And (minimum) of fuzzy input variables.
FUZZY NOT	FuzzyNot	Fuzzy	Logical NOT for fuzzy modeling. Reverses the sign of values of the input field.
FUZZY OR	FuzzyOr	Fuzzy	Takes the fuzzy Or (maximum) of fuzzy input variables.
FUZZY SELECTED UNION	FuzzySelectedUnion	Fuzzy	Takes the fuzzy Union (mean) of N Truest or Falsest fuzzy input variables.
FUZZY UNION	FuzzyUnion	Fuzzy	Takes the fuzzy Union (mean) of fuzzy input variables.
FUZZY WEIGHTED UNION	FuzzyWeightedUnion	Fuzzy	Takes the weighted fuzzy Union (mean) of fuzzy input variables. Multiply each value by its weight, take the sum of those products, then divide by the sum of the weights.
FUZZY X OR	FuzzyXOr	Fuzzy	Computes Fuzzy XOr: Truest - (Truest - 2nd Truest) * (2nd Truest - full False)/(Truest - full False).
MAXIMUM	Maximum	Raw	Takes the maximum of the input variables.
MEAN	Mean	Raw	Takes the mean of the input variables.
MINIMUM	Minimum	Raw	Takes the minimum of the input variables.
MULTIPLY	Multiply	Raw	Multiplies input variables.
NORMALIZE	Normalize	Raw	Normalizes the data from another field to range (default 0:1).
SUM	Sum	Raw	Sums input variables.
WEIGHTED MEAN	WeightedMean	Raw	Takes the weighted mean of input variables. Multiply each value by its weight, take the sum of those products, then divide by the sum of the weights.

WEIGHTED SUM	WeightedSum	Raw	Takes the weighted sum of input variables. Multiplies each field by its weight before adding. Like a weighted mean without the division.
X DIVIDED BY Y	ADividedByB	Raw	Performs X / Y
X MINUS Y	AMinusB	Raw	Performs X - Y

Table 1.1: EEMS Tools (Operators)

Choosing Operators for Your Logic Model

EEMS presents the user with choices for many operators and finding the right one can be confusing at first. The guidelines presented here will help you choose the right operator, but remember, sometimes it is best to experiment with several choices to make sure the operator you choose is appropriate for your model.

EEMS has operators designed to work on data before they are converted into fuzzy numerical space (i.e. when they are still in *raw* space) and those designed to work on data after they are converted into fuzzy space (see the above table). A user should respect that distinction. Using a non-fuzzy operator on fuzzy data can produce a result that falls outside the -1 to +1 continuum of fuzzy space. Doing this produces an invalid model.

Weighted Sum

The operators used in raw space are for the most part pretty straightforward. However the Weighted Sum operator merits a discussion. A **Weighted Sum** takes two or more inputs, and multiplies each of them by a weight before adding them. It has proven especially valuable with combining data of very similar types into one result that is then converted into fuzzy space. For example, if you were evaluating a region for intactness, the negative impact of paved roads might be considered similar to but greater than that of dirt roads. Their effects are additive, but a sum operator is not available in fuzzy space. To apply the **Weighted Sum** operator you might provide a weight of 1 to the paved road density and a weight of 0.5 to the dirt road density. In models that have done this, the result has been labeled "Effective Road Density."

And, Or, and **Union**

*Note that while the tools that operate on Fuzzy data are technically called "**Fuzzy And**", "**Fuzzy Or**", "**Fuzzy Union**", etc. they will be generally be referred to simply as **And, Or, Union,** etc., for the remainder of this document.*

And, Or and **Union** are the most common EEMS operators used. The choice between **And, Or,** and **Union** depends on the relationship of the input data to the question you are asking. **Or** returns the highest fuzzy value of any of the inputs, it is appropriate when any of the inputs is sufficient for your desired outcome. For example if you were evaluating a region in which three critically endangered species were present in

some locations, you could use an **Or** to combine *presence of species A*, *presence of species B*, and *presence of species C* into *high preserve value*. The presence of any of the three species would cause a map reporting unit to have a high fuzzy value. **And** is used when all inputs are necessary for the result to be high. For instance, if both habitat for and presence of a species of interest were required to consider a location as a preserve, you could combine *species presence* and *habitat density* with an **And** to produce *high preservation value*. **And** chooses the lowest fuzzy value of the inputs so that high fuzzy values for both conditions are necessary to yield a high fuzzy value for the result. **Union** takes the mean of the input values. **Union** allows each input to exert an influence on the result. If all inputs have a high value, the result will have a high fuzzy value; if all have a low value, the result will be low. If some are high and some are low, the result will be somewhere in between. Going back to our preserve example, we know if the species is present, the location has value as a preserve. If the habitat is present there is some value, too. If they are both present then the value is the highest. Union will yield that result. A **Weighted Union** is similar to **Union**, except that it allows a weight to the inputs. In our preserve example, if *habitat density* is more important than *species presence* (for instance in an area where remnant populations are under stress and habitat has been restored in areas where the species has not been able to recolonize) then you could provide a greater weight to *habitat density*.

Selected Union

The Selected Union represents a combination of **Or** (or **And**) and **Union**. Consider a study area that includes many different types of habitat, for example, a basin and range terrain. Some species of concern are found in valleys, others inhabit the foothills, and others the high mountains. What if there are 30 species of concern? The more species of concern in a location, the more valuable the location, but nowhere are they all found together. The **Selected Union** allows for the evaluation of such a study area. With the **Selected Union**, you choose a number of the truest (or falsest) of inputs to evaluate. In the basin and range example, you might choose five. A location with a high density of five (or more) species of concern would have a high fuzzy value for *high species diversity*. As the density of species of concern falls, so does the fuzzy value for *high species diversity*. A **Selected Union** with a parameter of 5 Truest would do just that. It performs a **Union** operation on the five inputs with the highest fuzzy values.

Final Word

It may not seem so at first, but fuzzy logic evaluative models are not hard to understand, and they are not hard to construct. With some time, some trial and error, and some thought, you can easily manage the concepts, the operators, and the construction of these models. The key is to trust that with experience comes both understanding and expertise.

III. EEMS Pro

The following section describes the process of designing and building an EEMS model within the ArcGIS Pro or ArcGIS Desktop environment. It is organized into 3 parts. The first part (3.1) provides an overview of the EEMS Pro framework (The ABC's of EEMS Pro). The second part (3.2) describes the four phases of the EEMS modeling process. And the third part (3.3) consists of a tutorial which will walk you through the steps involved in building an EEMS model.
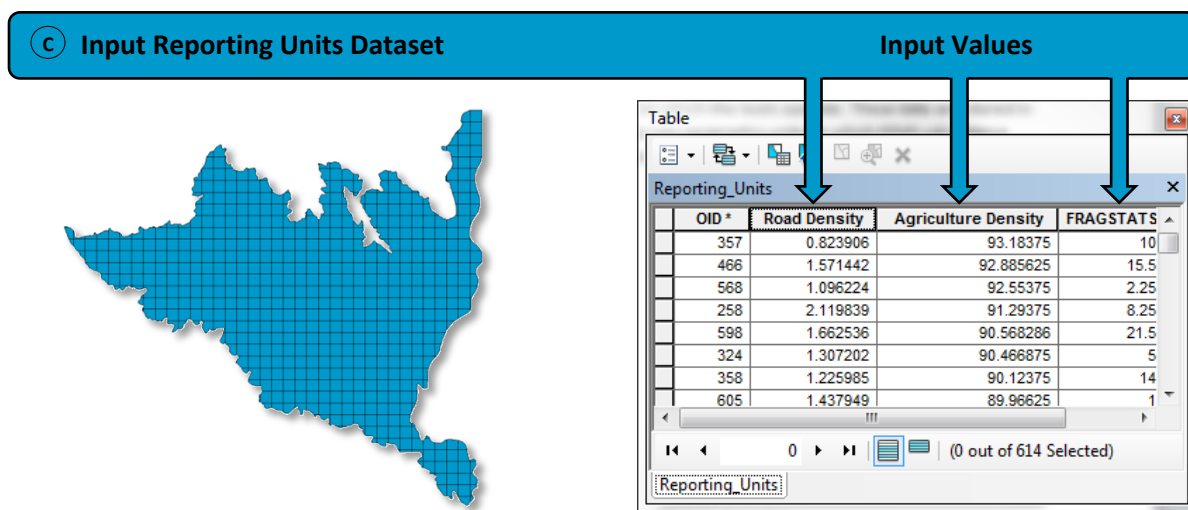
3.1 The ABC's of EEMS Pro

(A) **EEMS Pro** is written in Python and deployed as a collection of script tools within an ArcGIS Toolbox.

(B) **EEMS models** are constructed by bringing these tools into an ArcGIS Model Builder model, where they are connected and arranged in a logical hierarchy. A full list of the EEMS Tools and a description of their functionality is presented in table 1.1.
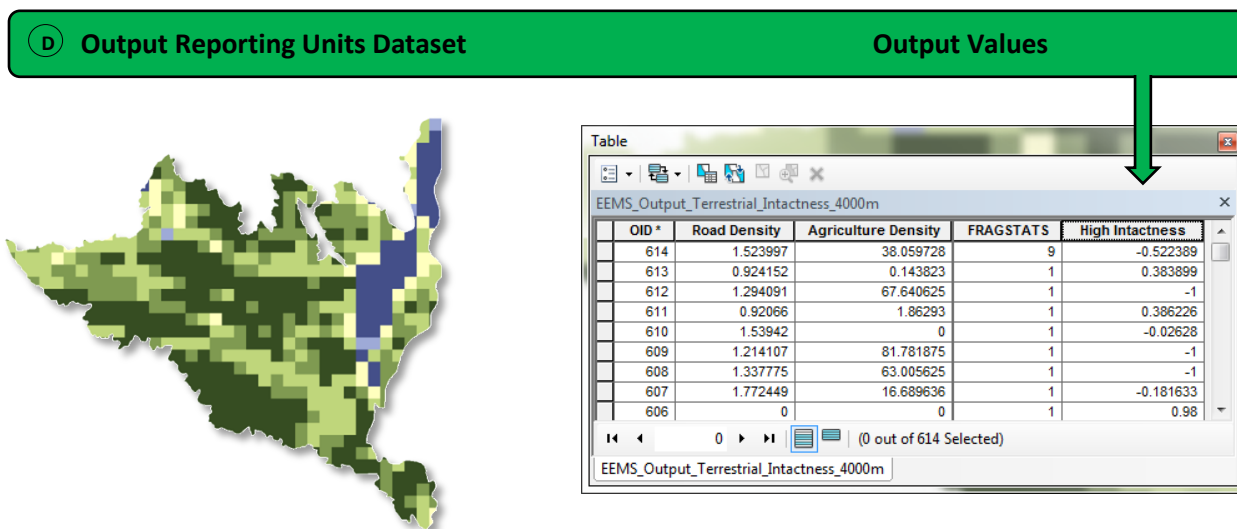
EEMS Pro Manual, 2023 (VERSION 1.0.1)
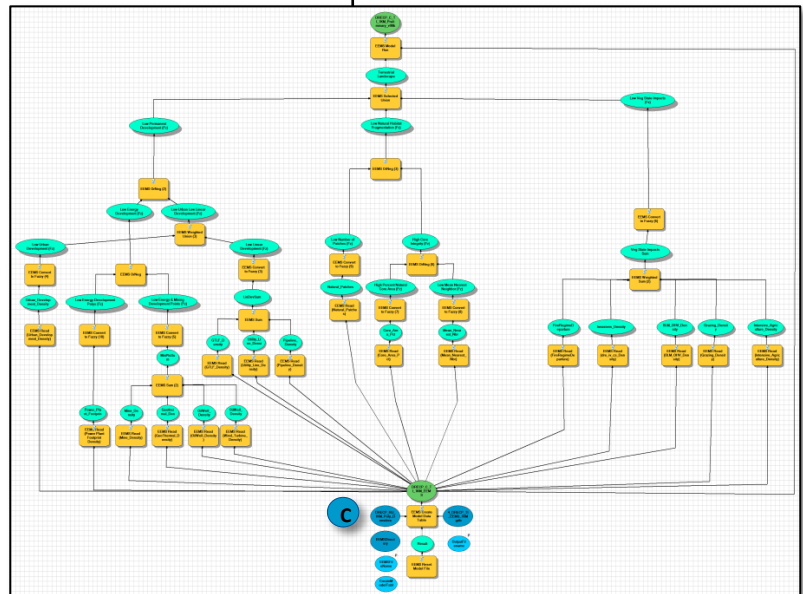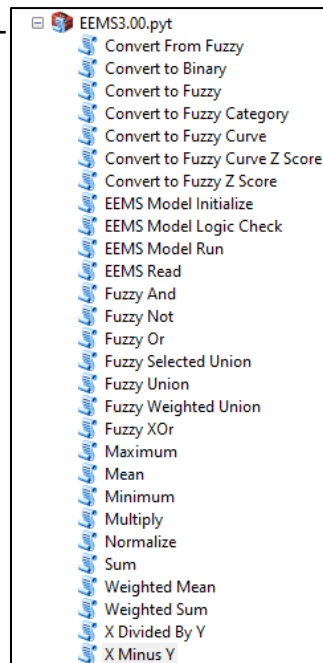
EEMS tools operate on what is referred to as the **Input Reporting Units Dataset** (C). All the values for the input variables (e.g., road density) need to be stored in a *single* vector dataset composed of reporting units. Reporting units are discrete geographic units containing quantitative measures of how much (or how little) of each variable is present within that reporting unit. Common measurements include densities, counts, and/or statistical measurements such as the mean or max. Reporting units may be regular geometric shapes (e.g., squares), or they may be in the form of irregular natural or political boundaries, such as watersheds or counties.

C Input Reporting Units Dataset	Input Values

OID *	Road Density	Agriculture Density	FRAGSTATS
357	0.823906	93.18375	10
466	1.571442	92.885625	15.5
568	1.096224	92.55375	2.25
258	2.119839	91.29375	8.25
598	1.662536	90.568286	21.5
324	1.307202	90.466875	5
358	1.225985	90.12375	14
605	1.437949	89.96625	1

(0 out of 614 Selected)

After the input reporting units dataset has been created and the EEMS model has been constructed, the model can be executed. When the model is executed, the EEMS tools read in the values in the input reporting units dataset (C), the calculations are performed, and an **output reporting unit dataset** (D) is created which contains a final output value for each reporting unit.

D Output Reporting Units Dataset	Output Values

EEMS_Output_Terrestrial_Intactness_4000m

OID *	Road Density	Agriculture Density	FRAGSTATS	High Intactness
614	1.523997	38.059728	9	-0.522389
613	0.924152	0.143823	1	0.383899
612	1.294091	67.640625	1	-1
611	0.92066	1.86293	1	0.386226
610	1.53942	0	1	-0.02628
609	1.214107	81.781875	1	-1
608	1.337775	63.005625	1	-1
607	1.772449	16.689636	1	-0.181633
606	0	0	1	0.98

(0 out of 614 Selected)

EEMS_Output_Terrestrial_Intactness_4000m

EEMS Pro Manual, 2023 (VERSION 1.0.1)

3.2 The EEMS Modeling Process

PRELIMINARY STEPS (BUILDING A CONCEPTUAL MODEL DIAGRAM):

Before preparing your input data and constructing an EEMS model, it is important to perform the following preliminary steps:

1. Think about the question you are trying to answer.
2. Think about what information (raw data) you will need to acquire in order to answer that question.
3. Construct a conceptual model diagram describing how you might fit these pieces of information together in order to answer your question (Figure 2-1)
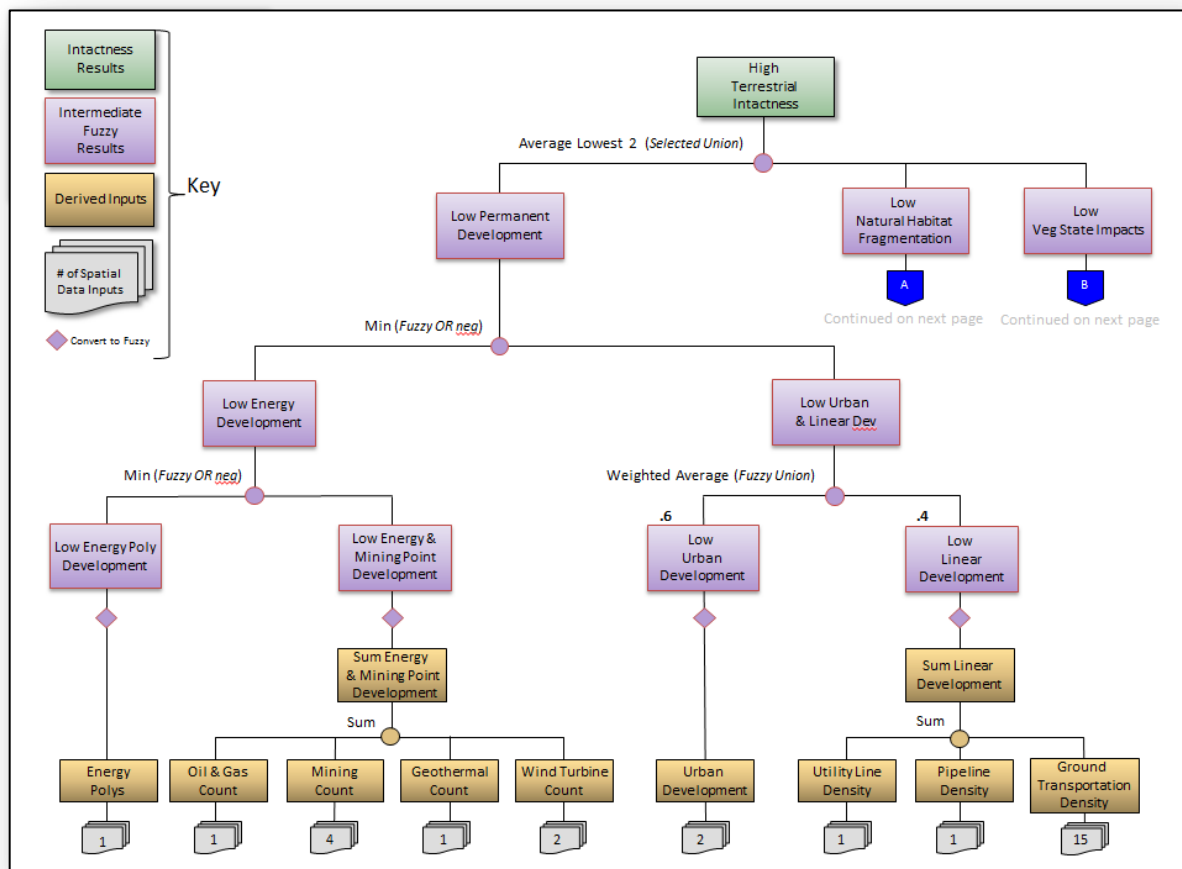
Figure 2-1: A conceptual model diagram (created in MS PowerPoint) describing the logic and operators to be used in constructing a Terrestrial Intactness logic model.

EEMS Pro Manual, 2023 (VERSION 1.0.1)

After you've created a conceptual representation of your model, you can proceed with the EEMS modeling process. There are typically four phases involved in this process:

Phase 1: Preparing input data
Phase 2: Creating or defining reporting units
Phase 3: Calculating densities (quantitative evaluation by reporting unit)
Phase 4: Building and Running the EEMS model

As discussed in section 1.4, EEMS operates on a single input dataset (represented in blue below). Each field in the attribute table of this dataset corresponds to an input variable, and each record corresponds to a reporting unit (discrete geographic units for which EEMS calculates a final output value). Consequently, in order to build and execute an EEMS model, one must first generate this master input reporting units dataset which contains all of the input data. This is accomplished in phases 1-3.

GENERATING THE INPUT REPORTING UNITS DATASET.
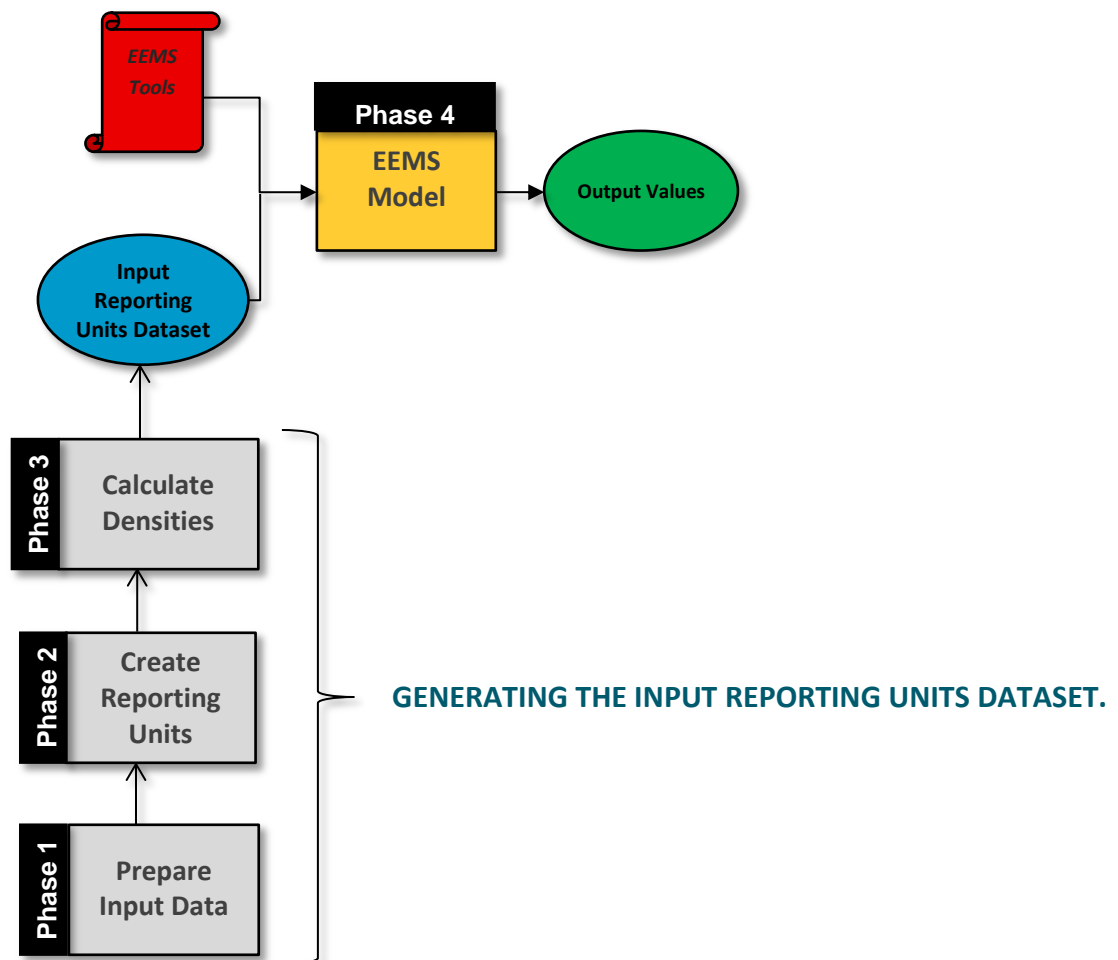
The process of generating the input reporting units dataset in phases 1-3 is generally carried out using python scripts and/or models which employ the standard suite of ArcGIS geoprocessing (e.g., clip and project (phase 1), create fishnet (phase 2), zonal statistics (phase 3), etc.). CBI has also created a set of support tools that can help with this process. These tools can be downloaded at the URL below.

EEMS Support Tools:

https://eemsonline.org/static/EEMS_Support_Tools.zip

Please note that these tools are not officially a part of the EEMS software package, and are provided *as-is*. CBI does not offer support for these tools at this time. Use of these tools is not required in order to use EEMS, but may be of value in assisting with the preparation and processing of input data during phases 1-3. For example, it includes a tool called **Calculate Feature Density by Reporting Unit** which will calculate the density of point, lines, or polygons within each reporting unit, which can be useful for phase 3.

The following section provides a detailed overview of each of the phases involved in the EEMS modeling process.

PHASE 1 (PREPARING INPUT DATA):

During this phase of the analysis, you will define your study area and pre-processes your input data. The purpose of this phase is to ensure that all of the input data are ready to be quantitatively evaluated within each reporting unit.

The process of preparing input data is unique to each analysis. Consequently, you will always need to construct your own models and scripts to conduct this phase of the process.

**A common workflow for phase 1 is as follows:**

1. Obtain or create a dataset which defines the study area.
2. Obtain the input data needed in order to answer the question under evaluation.
3. Project each dataset to a common projected coordinate system.
4. Select features of interest and generate new datasets as necessary.
5. Merge or combine existing datasets as necessary (e.g., roads from multiple counties).
6. Create points from x,y coordinates as necessary.
7. Dissolve input data as necessary.
8. Perform any necessary raster pre-processing

   NOTE: Raster datasets which will be used in in calculating percent cover by reporting unit should be classified to binary during this phase (1 for presence, 0 for absence).

9. Clip all of your input datasets to the study area.
10. Store all of your input data in a centralized location for input to phase 3.

## PHASE 2 (CREATING OR DEFINING REPORTING UNITS):

During phase 2, a reporting unit feature class is generated from the study area. The reporting units feature class is typically a vector based grid composed of regular square polygons (for example, 4km^2 or 1km^2). However, as mentioned previously, it may also be in the form of irregular natural or political boundaries, such as watersheds or counties.
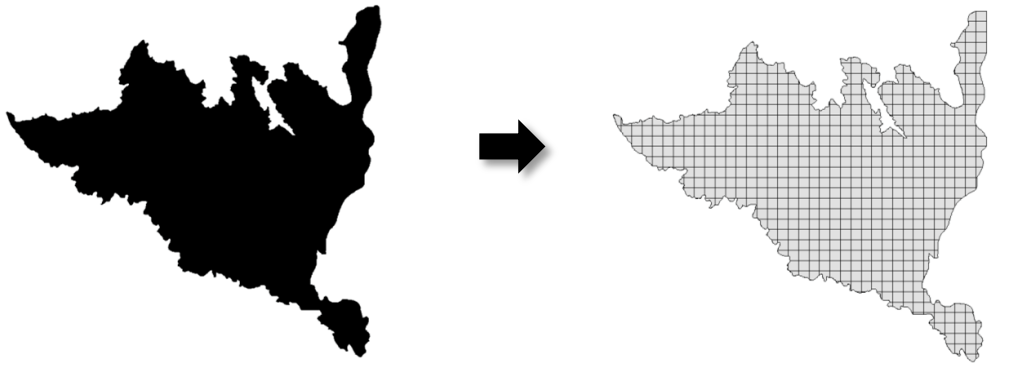
The general procedure for creating the input reporting units dataset within ArcGIS is to:

1. Use the **Fishnet** tool to create an empty reporting units dataset, setting the extent to match that of your study area boundary
2. **Clip** the fishnet to your study area boundary.

A corresponding raster version of the reporting units dataset should also be generated for processing any raster data inputs. The raster version of the reporting units dataset is used for generating zonal statistics (e.g., mean vegetation departure within each reporting unit). The cell size of the raster reporting units dataset should match the resolution of your input raster datasets (although ESRI's zonal statistics tool will automatically perform a resampling if this is not the case). When the raster version of the reporting units dataset is created, the output cell values should match the OBJECTIDs of the vector reporting units dataset within which they fall.

EEMS Pro Manual, 2023 (VERSION 1.0.1)
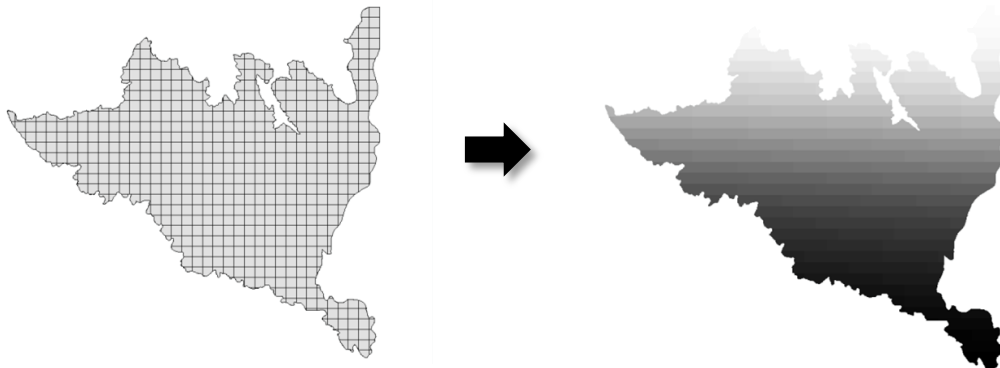
PHASE 3 (QUANTITATIVE EVALUATION, CALCULATING DENSITIES):

During this phase, the input datasets created in phase 1 are quantitatively evaluated by reporting unit. That is, within each reporting unit, a value is calculated and stored which summarizes or aggregates the data for each variable within that reporting unit (e.g., sum, mean, density, % cover, count, etc.). The populated reporting units dataset created during this phase will be the input to your EEMS Model.
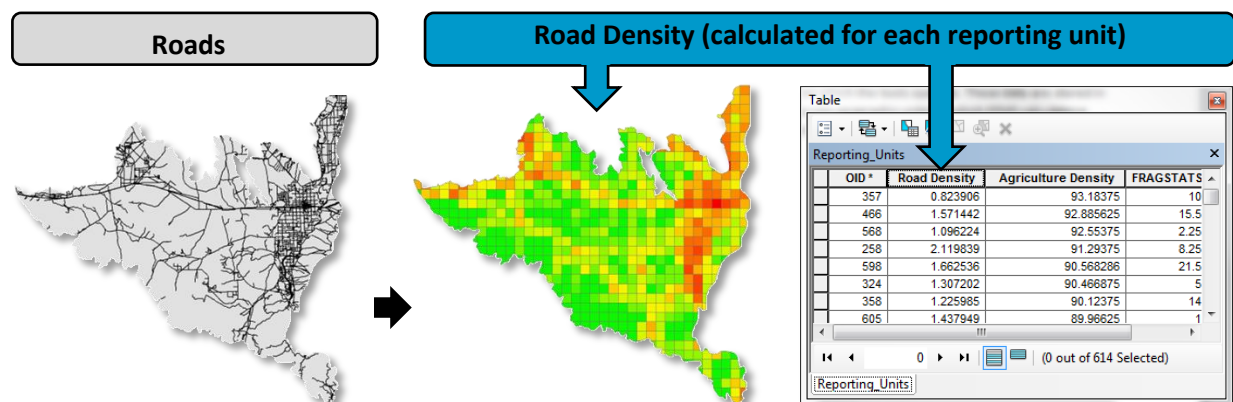
There are a number of steps involved in this phase. While there is no universal method of conducting this phase of the analysis, for raster data this is often carried out using **Zonal Statistics as Table** followed by a **Join** to the input reporting units. The **EEMS Support Tools** mentioned earlier also includes a set of tools in in the *EEMS_Support_Tools\EEMS Support Tools.tbx\Reporting Unit Tools* toolset that can help simplify much of this process (e.g., *Calculate Feature Density by Reporting Unit* & *Calculate Raster Density By Reporting Unit*).

Common metrics calculated for vector input data are as follows:

- **Points:** a count of the number of points within each reporting unit.
- **Lines:** the linear density within each reporting unit (e.g., km/km^2).
- **Polygons:** The percent of each reporting unit occupied by a polygon.

Common metrics calculated for raster input data are as follows:

- Percent coverage within each reporting unit
- Statistical values (e.g., mean, max, min) within each reporting unit (i.e., zonal statistics).

The general procedure for conducting this phase of the analysis is as follows:

1. Make a copy of the empty reporting units dataset created in phase 2.
2. Perform the necessary operations to calculate the desired metric within each reporting unit (either using the *Reporting Units Tools* in the EEMS Support Tools or the standard suite of tools in ArcToolbox, e.g., *Zonal Statistics as Table*).
3. The *EEMS Support Tools* and *Zonal Statistics as Table* tool will generate a separate Feature Class, or Table, respectively. This must then be joined back to the copy of the reporting units dataset based on the OBJECTID field.
4. Repeat steps 2 and 3 for each of your input datasets.

PHASE 4 (BUILDING AND RUNNING AN EEMS MODEL):

Phase 4 of the EEMS modeling process involves constructing your model in ArcGIS Model Builder based on the structure and logical relationships defined in your conceptual model diagram using the fuzzy logic operators in the EEMS Pro toolbox (*EEMS_Pro\EEMS_Pro\EEMS_Pro.pyt).* The input to your EEMS Model will be the populated reporting units dataset created in phase 3, which contains all of the input variables summarized by reporting unit.

Continue on with the tutorial below to learn how to build and execute a simple EEMS model.

3.3 Tutorial

BUILDING AND RUNNING AN EEMS MODEL IN ARCGIS

This tutorial will walk you through the final phase (phase 4) of the EEMS modeling process in ArcGIS. This will involve building a simple logic model using an input reporting units dataset that has already been prepared for you.

The purpose of this model will be to determine the level of human impact across the landscape based on two input variables: Road Density and Agricultural Density. Below is our conceptual model diagram:
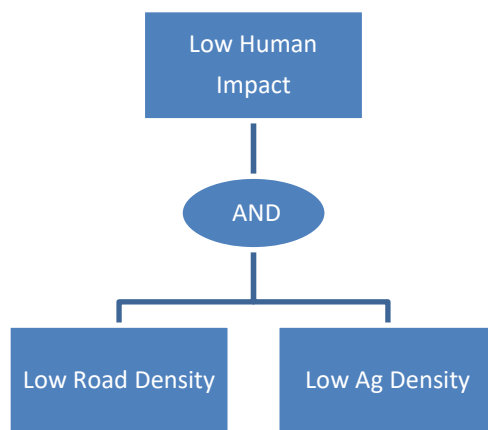
This can be expressed with the following logical statement:

If Road Density is Low, and Ag Density is Low, then Human Impacts are Low.

As you learned in section 2.3, the **AND** operator takes the lowest (falsest) of the inputs and is appropriate to use in situations where ALL of the inputs must be true in order for the output to be true.

Each node in our model diagram represents a logical **proposition**. A proposition is simply a declarative statement that can be evaluated for truth or falsity. In traditional logic, each proposition would either be True (1) or False (0) at a given location (there is no gray area). But in fuzzy logic (EEMS), each proposition can either be Totally True (+1), Totally False (-1), or somewhere in-between on the truth continuum.

Take note of how these propositions are phrased (*Low* Road Density, *Low* Ag Density, *Low* Human Impacts). Since propositions are statements that are evaluated for truth or falsity, they should be stated in a way that makes that evaluation possible. For example *Road Density* is not a statement that you could say was True or False, but *Low Road Density* is. So is *High Road Density*. When building an EEMS model, you first need to think about what variables contribute to your final proposition (Road Density and Ag Density), and then determine *how they relate* to your final proposition. Is it *Low Road Density* or *High Road Density* that contributes to *Low Human Impacts*? This will determine how you set your True and False thresholds (more on this later).

Let us begin…

Note: Before proceeding with the tutorial, you should have the EEMS Pro folder downloaded and stored on a local or network drive, and MPilot should be installed (refer to section 1.2 for instructions).

STEP 1 (SETUP YOUR WORKSPACE)

The first step in building an EEMS model is to setup a workspace in which to store your models, data, and other necessary files. So let's start by preparing a workspace:

I. Create a workspace folder somewhere on your local hard drive called **My_EEMS_Pro_Tutorial_Workspace**.

II. Copy the **EEMS_Pro_Tutorial_Data** folder from the EEMS_Pro folder into the workspace you created above (we want to work with a copy of the Tutorial Data in order to avoid any accidental overwrites).

III. Start ArcGIS Pro or ArcCatalog.

   Note: If you are using ArcGIS Pro, be sure to set the Active Environment to the conda clone that has MPilot installed (refer to section 1.2 for instructions).

IV. Using the Catalog Pane in ArcGIS Pro, or ArcCatalog in ArcGIS Desktop, make a connection to the workspace you created on step I.

V. Create a new Toolbox in your workspace called **EEMS_Pro_Tutorial_Models**.

VI. Create a new Model in your Toolbox called **EEMS_Pro_Tutorial_Model**.

   At this point, your workspace should be accessible through ArcGIS and it should resemble the contents and structure shown below:

EEMS Pro Manual, 2023 (VERSION 1.0.1)
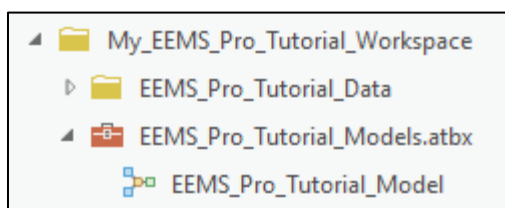
VII. Next, add the EEMS Toolbox to ArcGIS following the instructions below corresponding to the version of ArcGIS you are running.

ArcGIS Pro:

From the Catalog Pane -> Right click the Toolboxes Container 📁 -> click Add Toolbox -> Browse to EEMS_Pro.pyt in the downloaded EEMS Pro folder.

ArcGIS Desktop:

From the ArcToolbox pane in ArcCatalog-> Right Click the ArcToolbox icon 📕 click "Add Toolbox". Browse to EEMS_Pro.pyt in the downloaded EEMS Pro folder.

STEP 2 (INITIALIZE)

Next, edit your **EEMS_Pro_Tutorial_Model** model and add the **EEMS Model Initialize** tool from the EEMS_Pro.pyt toolbox. If dragging and dropping the tool doesn't work, right click the tool and select "Add to Model".
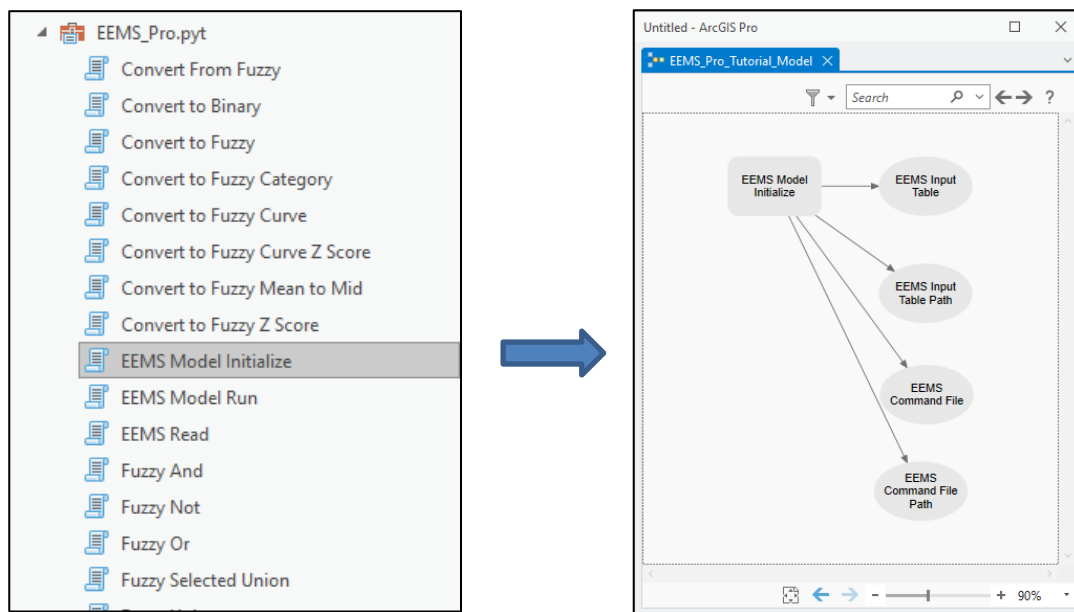
The **EEMS Model Initialize** tool performs the following functions:

1. It specifies the name and location of the Input Reporting Units feature class and makes the fields in this file accessible to EEMS as a table.
2. It specifies the name and location for the EEMS Command File (more on this later).
3. It stores the paths to these files as Model Builder variables called **EEMS Input Table Path**, and **EEMS Command File Path**. These variable names will be used throughout the tools in your model and consequently should not be changed.

Double click the **EEMS Model Initialize** tool, browse to the **Input_Reporting_Units** dataset located in *EEMS_Pro_Tutorial_Data\2_EEMS_Input.gdb*, and specify a name and output location for the **EEMS Command File** following the example below, updating the paths based on the location of the workspace you created in Step 1:

The input reporting units dataset you added to the EEMS Model Initialize tool has a field for **Road Density** and a field for **Agriculture Density.** There are other fields in the attribute table as well, but we won't be using them in this tutorial.

INPUT REPORTING UNITS DATASET

EEMS COMMAND FILE

The **EEMS Command File** is a text file that will contain all the instructions required to tell EEMS how to build and execute your EEMS model. Each time an EEMS tool in your model runs, an EEMS command gets written to the EEMS Command File. When all of the commands have been written to this file, EEMS (specifically, the EEMS libraries within MPilot), will be able to read in and execute the lines of code in the EEMS command file to generate your results.

Right click on the **EEMS Model Initialize** tool in your model, and click **Run**.

EEMS Pro Manual, 2023 (VERSION 1.0.1)

STEP 3 (READ IN THE DATA)

Next, bring in two **EEMS Read** commands from the EEMS Toolbox and connect the **EEMS Input Table** to them. Double click the first EEMS Read command and set the **Input Field** to "Road_Density". Then double click the second EEMS Read command and set the **Input Field** to "Agriculture_Density".

At this point, you should organize your model as shown on the left (in order be consistent with the screen captures in the tutorial).

You should also rename the blue output bubbles from the EEMS Read commands to something more meaningful (e.g., "Road Density" & "Ag Density"). These are just labels and can have spaces, whereas the Input Field names cannot.

The **EEMS Read** tool is used to specify which of the fields in your input reporting units dataset should be read in and processed by your EEMS model.

The two light blue bubbles attached to the EEMS Model Initialize tool are variable names. One of them stores a path to the Input Reporting Units dataset and the other one stores the path to the EEMS Command File. As mentioned earlier, it is important that you don't change these variable names as they will be used by several other EEMS tools throughout your model.

Click the **Run** button on your model. It is necessary to do this at this point in order to make the Road_Density and Agriculture_Density fields available for subsequent operations.

STEP 4 (CONVERT TO FUZZY SPACE)

As you learned in section 2.2, all input data, regardless of type (ordinal, nominal, or continuous) or units (e.g., km/km², percent cover, counts), must be converted into fuzzy space in order to combine the data using fuzzy logic operators (e.g., AND, OR, UNION). So, in this step, we will convert the raw Road Density and Ag Density values into fuzzy space.

To do this, bring in two **Convert to Fuzzy** tools and connect them to the output from the two **EEMS Read** tools as "Input Fields".

Your model should now match the example shown below.

Double click each of the **Convert to Fuzzy** tools and set the **True and False Thresholds** to match the values specified in the tool dialogs shown to the right.
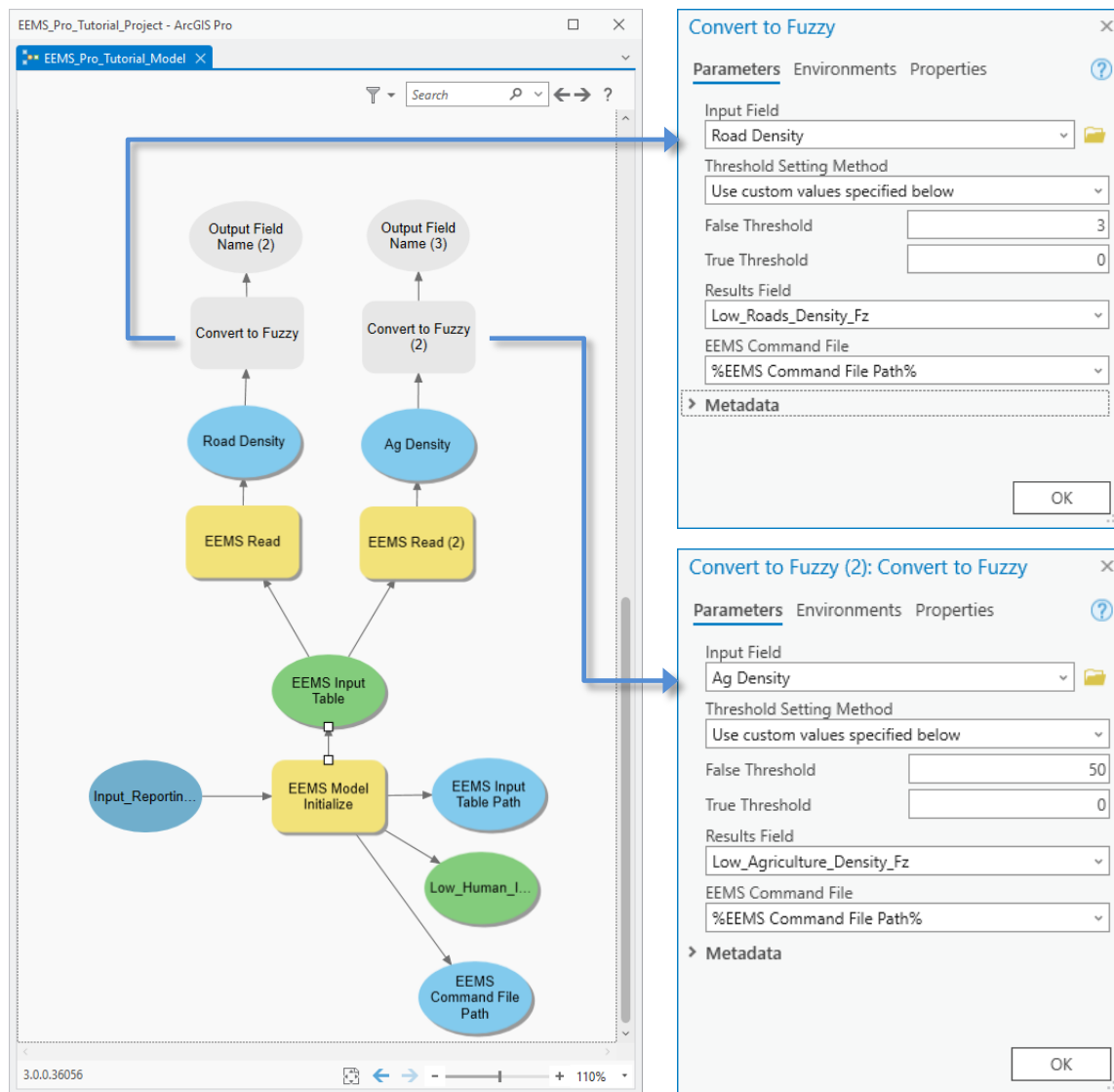
As you do this, you'll notice that the **Results Field** auto-populates with a default name prefaced with a directionality ("Low" or "High") based on which threshold you assigned a larger value to. The tool will also append "Fz" to the end of the results field name. It's good practice to include this so that when you are viewing the attribute table you can easily distinguish between fuzzy and non-fuzzy fields.

You have the option of changing the default field name, but the name should always reflect the proposition you are evaluating for degrees of truth.

After you have clicked ok, rename the blue output bubbles to *Low Road Density* and *Low Ag Density*.

Click the **Run** button on your model. Again, it is necessary to do this at this point in order to make these fields available for subsequent operations.

SETTING THRESHOLDS

In the step above, you entered in a **True** and **False Threshold** for each of your input propositions (*Low Road Density* and *Low Ag Density*). But what are those, and how do you decide what they should be?

As you have learned, fuzzy values in EEMS are represented over a continuum of -1.0 for totally false to +1.0 for totally true. 0.0 represents neither true nor false (Sheehan and Gough, 2016). Converting raw values into fuzzy space is a matter of determining where on that truth continuum each raw value falls in relationship to a logical proposition. For example, we might consider the proposition "High Species Richness" to be totally true in places where there are more than 100 different species, totally false in places where there are less than 20 different species, and somewhere in between true and false in places that have between 20 and 100 different species. As the analyst, you control how that transformation from raw values to fuzzy (truth) values occurs by setting the True and False Thresholds at those cutoffs.

- The **True Threshold** indicates the point at which a proposition becomes totally true (+1)
- The **False Threshold** indicates the point at which a proposition becomes totally false (-1)
- Raw input values falling between the True and False thresholds are transformed to a fuzzy value between -1 and +1 based on a linear transformation between the two thresholds.

For example, given a **True Threshold** of 100 and a **False Threshold** of 20, the transformation between raw input values to fuzzy values would occur along the line shown below.
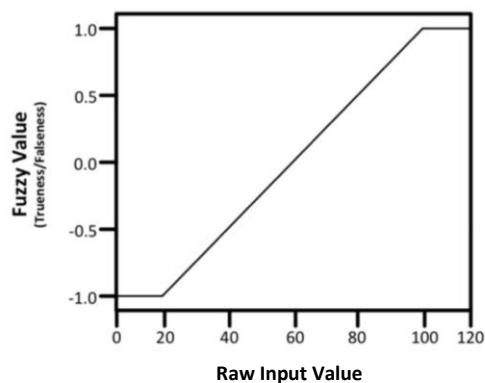
Raw input values >= the True Threshold of 100 will be transformed to a fuzzy value of +1 (Totally True).

Raw input values <= the False Threshold of 20 will be transformed to a fuzzy value of -1 (Totally False).

Raw input values that fall between the True and False thresholds will be transformed to a fuzzy value between -1 and +1.

If we were asking the opposite question (e.g., our proposition was "*Low* Species Richness"), we would simply reverse the True and False Thresholds. So if we set the True Threshold to 20 and the False Threshold to 100, the transformation would occur along the line shown below.
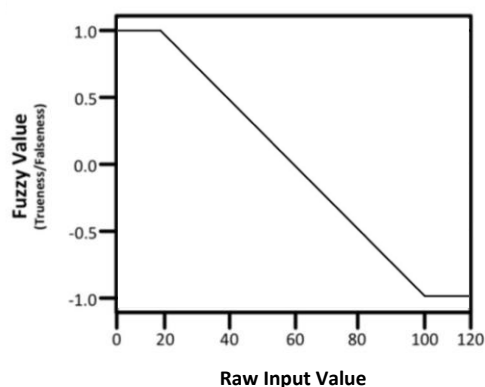

Raw input values <= the True Threshold of 20 will be transformed to a fuzzy value of +1 (Totally True).

Raw input values >= the False Threshold of 100 will be transformed to a fuzzy value of -1 (Totally False).

Raw input values that fall between the True and False thresholds will be transformed to a fuzzy value between -1 and +1.

For the "Low Road Density" proposition in the tutorial model, we set the True Threshold at 0 and the False Threshold at 3. By doing so we have indicated that the proposition *Low Road Density* is totally true (+1) when the road density is 0, totally false (-1) when the road density is greater than or equal to 3 km/km^2, and somewhere in between when the road density is between 0 and 3km/km^2. So the transformation from the raw input values to fuzzy space will occur along the line shown in the chart below.

Proposition: Low Road Density
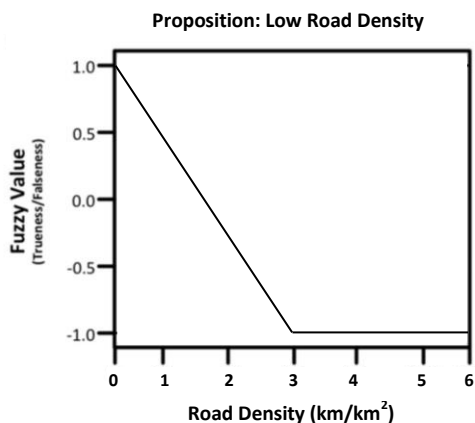
The proposition *Low Road Density* is totally true only when the road density is 0 (since there are no lower values).

The closer a road density value is to the True Threshold of 0, the more true the proposition becomes.

The closer a road density value is to the False Threshold of 3, the more false the proposition becomes.

Once a road density value exceeds 3, the proposition *Low Road Density* becomes totally false.

For the Low Ag Density proposition we, again, set the True Threshold at 0, and we set the False Threshold at 50. By doing so, we have indicated that the proposition *Low Ag Density* is totally true (+1) when the ag density is 0, totally false (-1) when the ag density is greater than or equal to 50 %, and somewhere in between when the ag density is between 0 and 50%. So the transformation between the raw input data and fuzzy space will occur along the line shown in the chart below.

Proposition: Low Ag Density
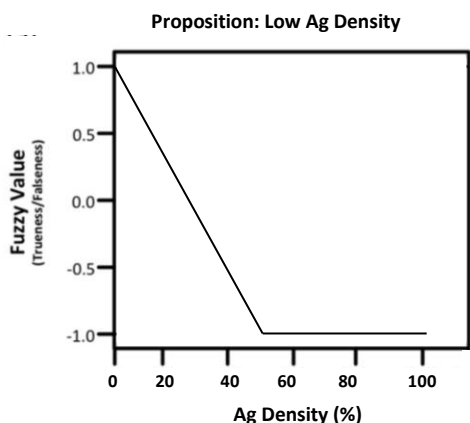
The proposition *Low Ag Density* is totally true only when the ag density is 0 (since there are no lower values).

The closer an ag density value is to the True Threshold of 0, the more true the proposition becomes.

The closer an ag density value is to the False Threshold of 50, the more false the proposition becomes.

Once an ag density value exceeds 50, the proposition *Low Ag Density* becomes totally false.

So how does one determine where to set the True and False thresholds outside of a tutorial? There are several different approaches that can be used. One common approach is to set them at 1-2 standard deviations from the mean. Another approach is to set them using the minimum and maximum values in the raw input field. In fact, you may have also noticed that the **Convert to Fuzzy** tool now provides a *Threshold Setting Method* dropdown which allows the user to automatically set the thresholds at the Min/Max value or at x Std Dev from the mean.

Min/Max and Std Dev are both good defensible approaches to setting thresholds because you are setting cutoffs for determining what constitutes "High" or "Low" based on the distribution of data in your study area. For example, setting the True and False Thresholds at 1.5 Std Dev from the mean ensures that in order for a value to be considered "High" or "Low" for a given proposition, it must fall within the tails of the distribution. It allows you to identify, for example, "High Species Richness" relative to the number of species that tend to occur within your study area.
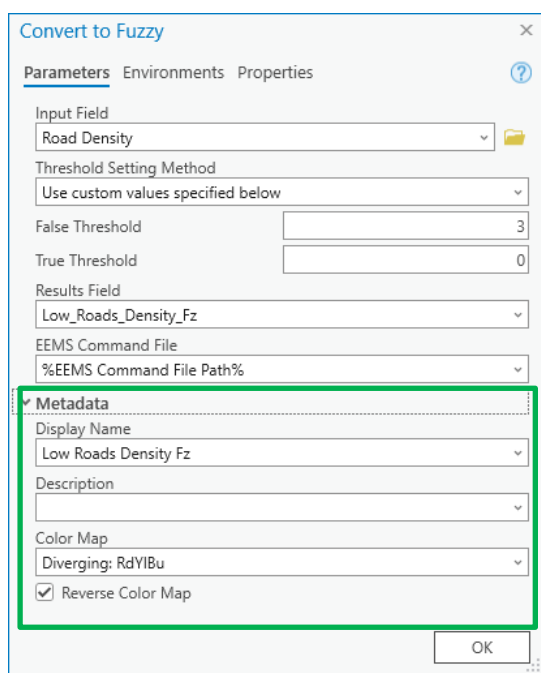
As you learned in section 2.2, thresholds can also be set using ecological thresholds (and that the thresholds may be set differently depending on the species you are studying). Thresholds can also be set using legal definitions, or expert opinion. For example, there may be a city or county ordinance that defines "High Housing Density" as x dwelling units per square mile.

You can also combine approaches. For example, if the raw input data has a skewed distribution, it is fairly common to use the min or max value for setting one threshold and standard deviations for setting the other.

Ultimately, the approach you use to setting thresholds is up to you. You just need to make the determination as to what point each proposition in your model becomes **Totally True** and **Totally False**, and then set the thresholds accordingly.

EEMS Pro also provides additional tools for converting raw input data to fuzzy space. For example, the **Convert to Fuzzy Category** tool allows you to assign each unique raw value from a numeric categorical value to a specific fuzzy value (e.g., convert all raw values of 5 to a fuzzy value of -1). See table 1.1 in section 2.3 for a full list of the fuzzy conversion tools.

METADATA

When you were setting the thresholds in the **Convert to Fuzzy** tool, you may have also noticed a set of optional fields under the **Metadata** header (as well as in other tools in the EEMS Pro Toolbox). These fields can be used to add auxiliary information to each node in your EEMS model. This information will be written to the EEMS Command file, where it can be accessed and used by EEMS Online, Data Basin, and other web applications that consume EEMS models.

Currently, there are three options available:

1. Display Name: Used to provide a human readable alias for the field name. For example, field names in ArcGIS must not have spaces, but the Display Name can. The Display Name is what will appear in the interactive model diagram in EEMS Online and Data Basin.

2. Description: Used to provide a brief description about the data stored in the Results Field. When a description is added, EEMS Online will add an "*i*" icon to the node which will display the description when a user hovers over it.

3. Color Map: Used by EEMS Online to control the colors used to render the data in the Results Field (plus an additional checkbox that can be used to reverse the selected color ramp). Visit the URL below for a visual representation of the color map options:

https://matplotlib.org/3.1.1/gallery/color/colormap_reference.html

The **EEMS Read** tool includes a fourth metadata parameter (**Data Sources**), where you can add the name and URL for the data source(s) used to create the data stored in the Input Field. In order to ensure EEMS compatibility with EEMS Online, any data sources entered here will be added to the **Description** field in the command file.

STEP 5

Let us now continue on with the tutorial.
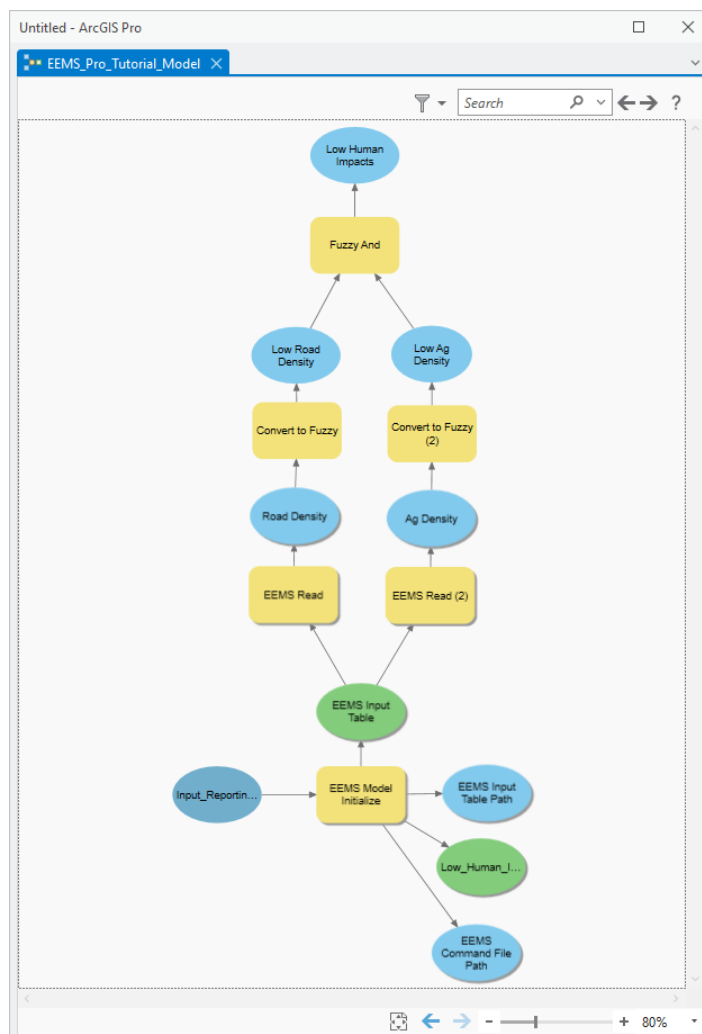
Bring in the **Fuzzy And** tool and connect the two fuzzy nodes to it as "Input Fields", as shown to the left.

Double click the **Fuzzy And** tool and change the Results Field to "Low_Human_Impacts_Fz".

Rename the output bubble from the **Fuzzy And** command to something more meaningful (e.g., "Low Human Impacts").

Run the model**.**

Note that up until this point we haven't created any new data and EEMS hasn't done any processing of the input fields. The only thing that has happened up to his point is that each time you pushed the run button on your model, a line (or lines) of code have been written to the **EEMS Command File**.

Let's open up this file and take a look inside. It can be opened with any text editor (e.g., Notepad or VIM).

EEMS Pro Manual, 2023 (VERSION 1.0.1)

Navigate to your EEMS Command File, which, if you followed the steps above, should be a file called *Low_Human_Impacts.mpt* located in your EEMS_Pro_Tutorial_Data*EEMS_Command_Files* directory. In it you should see the following lines of code:

```
Roads_Density = EEMSRead(
    InFileName = <your workspace path>\Input_Reporting_Units,
    InFieldName = Roads_Density,
    DataType = Float,
    Metadata = [
        DisplayName: Roads Density,
        ColorMap: binary
    ]
)
Low_Roads_Density_Fz = CvtToFuzzy(
    InFieldName = Roads_Density,
    FalseThreshold = 3.0,
    TrueThreshold = 0.0,
    Metadata = [
        DisplayName: Low Roads Density Fz,
        ColorMap: RdYlBu_r
    ]
)
Agriculture_Density = EEMSRead(
    InFileName = <your workspace path>\Input_Reporting_Units,
    InFieldName = Agriculture_Density,
    DataType = Float,
    Metadata = [
        DisplayName: Agriculture Density,
        ColorMap: binary
    ]
)
Low_Agriculture_Density_Fz = CvtToFuzzy(
    InFieldName = Agriculture_Density,
    FalseThreshold = 50.0,
    TrueThreshold = 0.0,
    Metadata = [
        DisplayName: Low Agriculture Density Fz,
        ColorMap: RdYlBu_r
    ]
)
Low_Human_Impacts_Fz = FuzzyAnd(
    InFieldNames = [Low_Roads_Density_Fz,Low_Agriculture_Density_Fz],
    Metadata = [
        DisplayName: Low Human Impacts Fz,
        ColorMap: RdYlBu_r
    ]
)
```

These are the commands that will instruct EEMS to read in the input data in the two input fields and generate new output data in the three fuzzy fields.
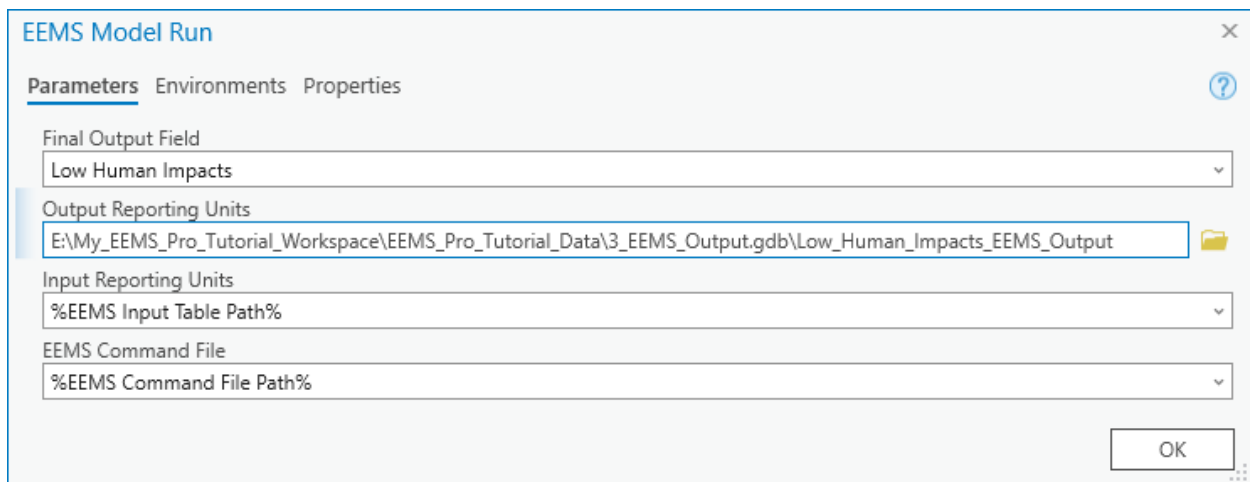
If you see more than the five commands listed above, that suggests that you ran one of the tools more than once. Duplicate commands can cause EEMS to fail. If this is the case, close your EEMS Command File and run your entire model again (In **ArcGIS Desktop**: Model -> Run Entire Model, or in **ArcGIS Pro**: Right

Click Validate -> Run). By rerunning your tool from the beginning, the EEMS Model Initialize tool will re-run first, which will clear out the EEMS command file and all of the subsequent commands will be rewritten. Consequently, the first step to take if you encounter an error when running your EEMS model should be to clear out the command file by rerunning the entire model.

Close the EEMS Command File if you haven't already done so.

STEP 6

Now that we've constructed our model, bring the **EEMS Model Run** tool in and connect the Top Node (Low Human Impacts Fz) to it as the "Final Output Field". Double click the tool, and specify a path for the Output Reporting Units (you can send it to the **3_EEMS_Output.gdb** geodatabase in the Tutorial Data workspace. Name it **Low_Human_Impacts_EEMS_Output**):

Click **Ok**.

The **EEMS Model Run** tool creates a CSV file from attribute table of the input reporting units feature class. The CSV File and the EEMS Command File will then get handed off to the EEMS libraries in MPilot where the model will be executed and the resulting output data will be added to the CSV. The CSV file (containing both the input and output fields) will then get joined back to the Output Reporting Units.

At this point your model is complete and ready to be run. Please confirm that it appears as shown on the following page, then **Run** the model.

Final output values will be stored in the attribute table of the feature class you specified as the Output File Name.

EEMS Pro Manual, 2023 (VERSION 1.0.1)
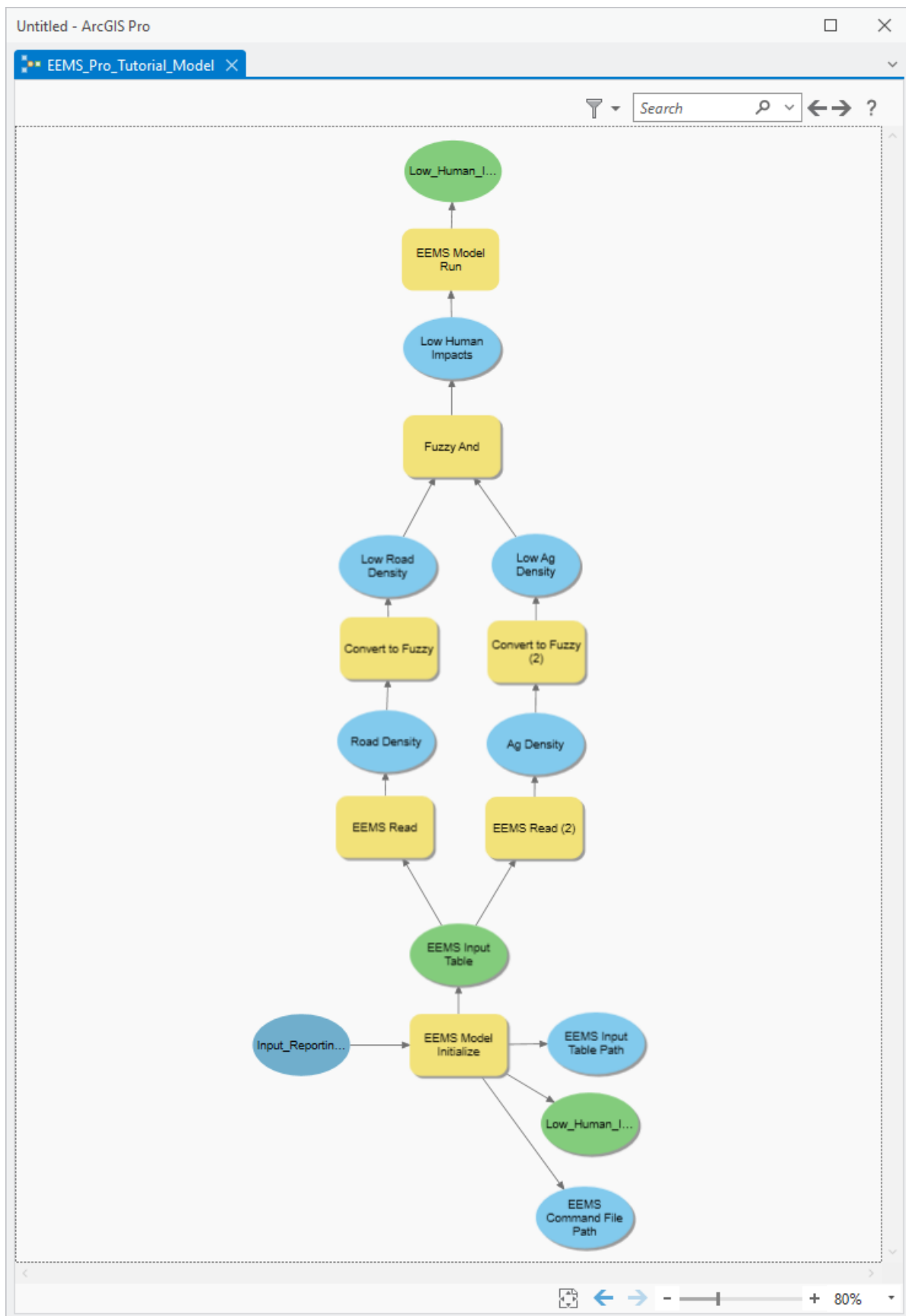
STEP 9

Bring your output into ArcMap and observe the results. Use the lyr file located in the EEMS_Pro_Tutorial_Data\Layer_Files directory to symbolize your output using the same class breaks and color scheme used in the solution below. Do your results match?

Blue areas are places with Low or Very Low fuzzy values. These are areas that have been most affected by roads and agricultural development. In these areas, our **Low Human Impacts** proposition is closer to false than to true, and in some reporting units our proposition is totally false (-1).

Greener areas are those reporting units in which our proposition is closer to true than false. And in some areas, our proposition is totally true (+1). Since we set our **True Threshold** at zero, these will only occur in areas where there are no roads and no agricultural development.
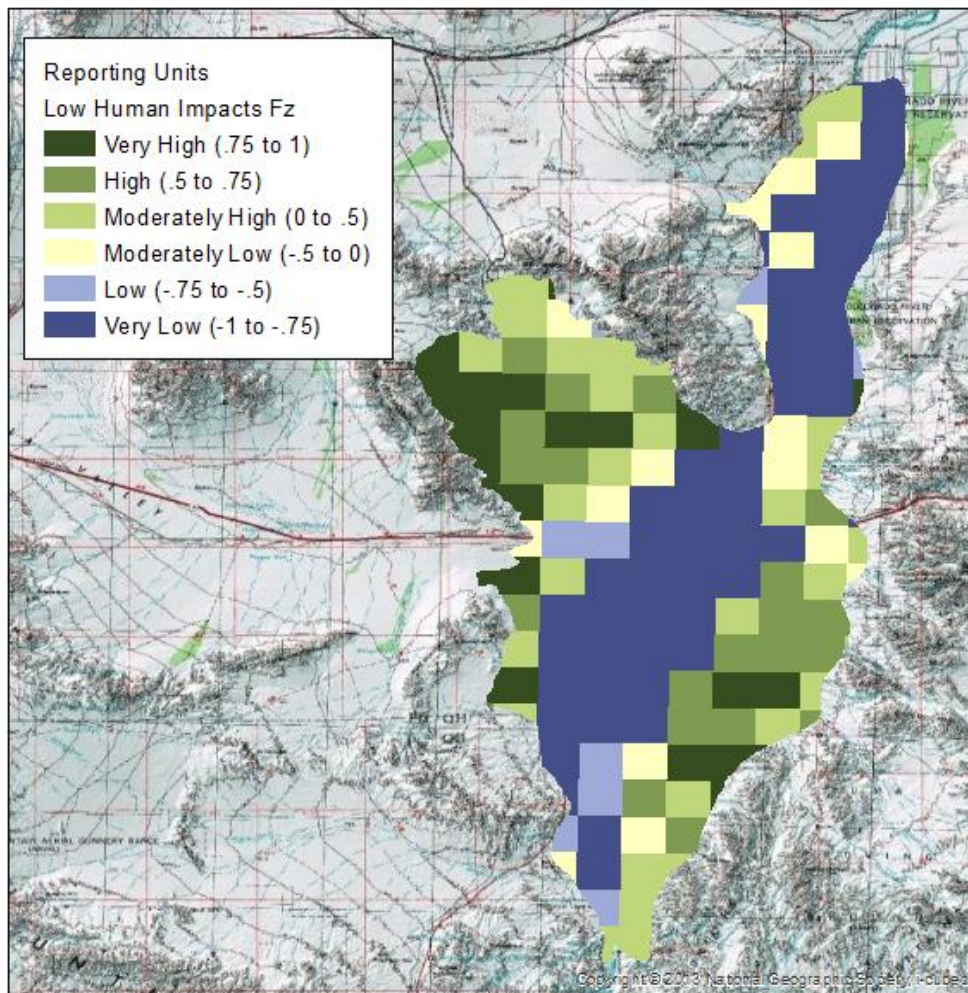
The roads and agriculture data used to calculate the densities are in the \EEMS_Pro _Tutorial_Data\1_RAW_Input.gdb geodatabase. If you bring them into your map, you can see that there

EEMS Pro Manual, 2023 (VERSION 1.0.1)

are indeed areas in the Northwest side of the study area that don't have any roads or agricultural development. The output values in the Low_Human_Impacts_Fz field should be +1 in these areas.
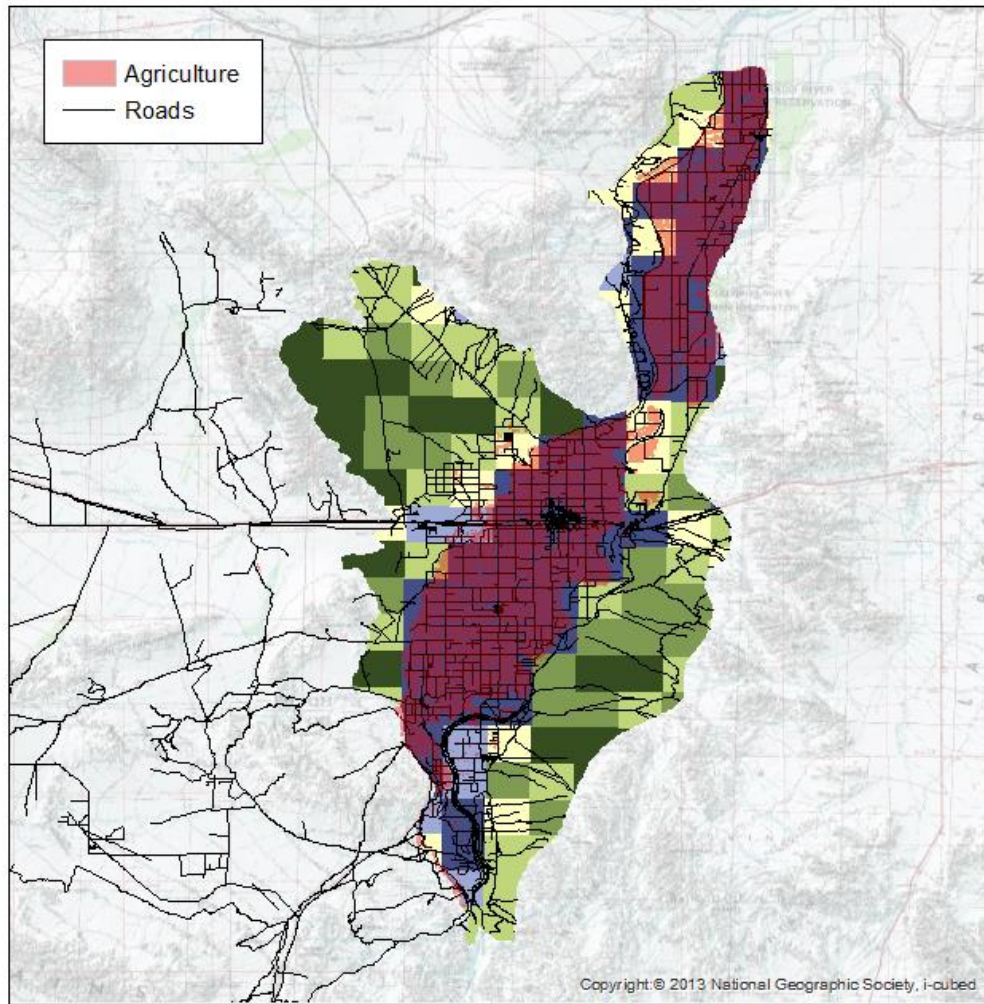
This concludes the tutorial. At this point, you are encouraged to experiment with different thresholds and operators in order to observe the effects on the final output values. You are also encouraged to experiment with your own input data and study area boundary.

References Cited

Jensen, M., K. Reynolds, U. Langner, and M. Hart. 2009. Application of logic and decision models in sustainable ecosystem management. 2009. Proceedings of the 42nd Hawaii International Conference on Systems Sciences. Waikoloa, Hawaii. 5-8 January 2009.

Reynolds, K.M. 1999. NetWeaver for EMDS version 2.0 user guide: A knowledge base development system. U.S. Forest Service, General Technical Report PNW-GTR-471, U.S. Forest Service, Pacific Northwest Research Station, Portland, Oregon.

Sheehan, T. and Gough, M. (2016) A platform-independent fuzzy logic modeling framework for environmental decision support. Ecological Informatics. 34:92-101