

Geoprocessing and Automation Lab Exercises

Table of Contents

Slide Notes	4
<i>Introduction</i>	4
<i>Basic Geoprocessing</i>	4
<i>Environments</i>	5
ENVIRONMENT EXAMPLES:.....	5
<i>ModelBuilder</i>	7
MODEL VARIABLES.....	8
<i>Python</i>	9
Geoprocessing and Automation GIS Lab Exercises:	11
<i>Heads-Up: What you will turn in from these labs</i>	11
<i>Note: The Navigation Pane</i>	12
<i>Exercise 1: Set the Default Geodatabase for your ArcGIS Pro Project</i>	13
OBJECTIVES.....	13
DATA USED.....	13
HOMEWORK ASSIGNMENT:.....	13
EXERCISE.....	13
<i>PREPARE YOUR ArcGIS PRO PROJECT</i>	13
<i>CHECK THE CURRENT DEFAULT GEODATABASE:</i>	14
<i>OBSERVE HOW THE TOOLS USE YOUR DEFAULT GEODATABASE:</i>	17
<i>CHANGE YOUR DEFAULT GEODATABASE FOR THIS ArcGIS PRO PROJECT:</i>	19
<i>OBSERVE THAT THE TOOLS NOW DEFAULT TO THIS NEW GEODATABASE:</i>	22
<i>OBSERVE THAT THE DEFAULT HAS BEEN SAVED WITH YOUR ArcGIS PRO PROJECT:</i>	22
<i>OBSERVE THAT YOUR DEFAULT GEODATABASE IS ONLY CHANGED FOR THIS PROJECT:</i>	23
<i>Exercise 2: Learn the Merge tool</i>	25
OBJECTIVES.....	25
DATA USED.....	25
HOMEWORK ASSIGNMENT:.....	25
EXERCISE.....	25
<i>PREPARE YOUR ArcGIS PRO PROJECT</i>	25
<i>REVIEW ATTRIBUTE FIELDS</i>	26
<i>OPEN AND RUN MERGE TOOL</i>	27
<i>Exercise 3: Learn the Dissolve tool</i>	31
OBJECTIVES.....	31
DATA USED.....	31
HOMEWORK ASSIGNMENT:.....	31
EXERCISE.....	31
<i>PREPARE YOUR ArcGIS PRO PROJECT</i>	31
<i>OPEN AND RUN DISSOLVE TOOL</i>	32
<i>Exercise 4: Learn the Buffer tool</i>	35
OBJECTIVES.....	35
DATA USED.....	35

HOMEWORK ASSIGNMENT:	35
EXERCISE:	35
PREPARE YOUR ArcGIS Pro PROJECT:	36
SELECT FIRST SET OF ROADS:	36
OPEN AND RUN BUFFER TOOL:	37
SELECT SECOND SET OF ROADS:	39
OPEN AND RUN BUFFER TOOL:	40
Exercise 5: Learn the Union tool:	43
OBJECTIVES:	43
DATA USED:	43
HOMEWORK ASSIGNMENT:	43
EXERCISE:	43
PREPARE YOUR ArcGIS Pro PROJECT:	43
ADD AN ATTRIBUTE FIELD TO EACH FEATURE CLASS:	44
OPEN AND RUN UNION TOOL:	47
Exercise 6: Learn the Clip tool:	51
OBJECTIVES:	51
DATA USED:	51
HOMEWORK ASSIGNMENT:	51
EXERCISE:	51
PREPARE YOUR ArcGIS Pro PROJECT:	51
SELECT GAME MANAGEMENT UNIT POLYGON:	52
OPEN AND RUN CLIP TOOL:	53
Exercise 7: Build and Run a Model:	56
OBJECTIVES:	56
DATA USED:	56
HOMEWORK ASSIGNMENT:	56
EXERCISE:	56
PREPARE YOUR ArcGIS Pro PROJECT:	56
MAKE A NEW TOOLBOX AND MODEL:	57
BUILD THE MODEL:	60
Exercise 8: Export Model to Python Code:	72
OBJECTIVES:	72
DATA USED:	72
HOMEWORK ASSIGNMENT:	72
EXERCISE:	72
PREPARE YOUR ArcGIS Pro PROJECT:	72
Exercise 9: GeoProcessing Intersect and Landfire:	81
OBJECTIVES:	81
DATA USED:	81
HOMEWORK ASSIGNMENT:	81
EXERCISE:	81
PRELIMINARY STEPS: WHAT IS LANDFIRE?	81
THE ACTUAL LAB EXERCISE:	82
RUN THE INTERSECT TOOL:	88
USE SUMMARIZE TOOL TO CALCULATE ACREAGES OF EACH SAF_SRM TYPE IN PROJECT AREA:	92
OPTIONAL:	95
OPTIONAL II:	95
OPTIONAL III:	95

Slide Notes

Introduction

Slide 2: Main topics for today

- 1) **Slide 3:** General Geoprocessing
- 2) **Slide 4:** Environments
- 3) **Slide 5:** ModelBuilder
- 4) **Slide 6:** Python

Basic Geoprocessing

Slide 7: Geoprocessing refers to any type of modeling, analysis or just anything you would do with any of the tools in ArcGIS. All of the tools in the Toolbox are considered “Geoprocessing” tools.

There are over a thousand of these geoprocessing tools available (I count 1,045 in ArcMap 10.8, and ArcGIS Pro has several that ArcMap does not have).

Slide 8, 9, 10: They can be run individually by clicking on them, strung together in models, or run by command line in Python.

Slide 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21: I will explain some of the most common tools here, and we will use the Buffer, Clip, Union, Merge and Dissolve tools in the lab. These are some of the oldest geoprocessing functions produced by Esri, going back to ARC/INFO in the 1980's.

Slide 22: The **Buffer** tool creates a polygon around existing features (points, lines or polygons), or even inside polygons (if you use a negative buffer distance).

Slide 23: You can generate internal buffers by using a negative buffer distance. This only works on polygons, though. There's no “inside” to a point or a polyline.

Slide 24: The **Clip** tool cuts a feature class to some Area of Interest (AOI) polygon. This tool only works on vector data; there are different tools that do similar things for Raster data.

Slide 25: The **Intersect** tool creates a new feature class showing only the regions that had overlapping original features. **Slide 26:** The intersect tool can intersect all shape types.

Slide 27: The **Identity** tool sort of stamps Identity features into an existing feature class. **Slide 28:** The new feature class does not get any additional area added to it, but any areas overlapping an Identity feature will be cut along the Identity boundary and will get Identity attributes.

Slide 29: The **Union** tool combines one or more feature classes into a new feature class. It also intersects all features with each other so that the new feature class has separate features for each overlapping part of each of the source features. In this illustration, 4 rectangular features are unioned into a new set of 12 square polygons.

Slide 30: The **Erase** tool simply removes parts of an existing feature class based on a polygon feature. There's not a lot of mystery with this tool.

Slide 31, 32, 33: The **Symmetric Difference** tool produces a feature class showing only those areas that did not have any overlapping areas.

Slide 34, 35: The **Merge** and **Append** tools combine feature classes into a single feature class without modifying the geometry any. Merge creates a new feature class, and Append adds a feature class to an existing feature class. In my experience, **Merge** is also a lot less likely to crash than the **Append** tool.

Slide 36: The **Dissolve** tool combines features based on common attribute values. You can also use it to just dissolve every feature in the feature class into a single object as well.

Slide 37: Pop Quiz: How could we get this border region, which represents the area $\leq 50\text{m}$ from the stand boundary? There are several ways we could get this, but I used the internal buffer on the stand polygon, then the “Erase” tool to delete the internal buffer from the original stand polygon.

Environments

Slide 38: Whenever you run a tool in ArcGIS Pro, you have to tell the tool a few things. In ArcGIS, these things are referred to as “Parameters”. **Slide 39:** For example, when you run the Buffer tool, you need to specify parameters for:

- 1) Input Feature Class
- 2) Output Feature Class
- 3) Distance
- 4) Distance Units
- 5) Optionally some buffer shape types
- 6) Optionally whether to buffer using geodesic or planar distances
- 7) Optionally some Dissolve methods.

Click “OK” and the tool runs. **Slide 40:** However, there are actually several more parameters that ArcGIS can use in the analysis that are not revealed in the tool dialog. **Slide 41:** These are called “Environments” and they give you a lot more control over exactly how the tool is run.

Slide 42: Not all tools honor all environment settings, but if a tool **can** then these settings will show up in that tool’s **Environments** pane.

Slide 43: For example the **Buffer** tool uses these six environment settings, **Slide 44:** while the **Slope** tool uses these 5. Each tool shares the *Output Coordinates* and *Processing Extent* settings.

Slide 45: If a tool does support an environment, it’ll usually say so at the bottom of the Help file for that tool.

Environment Examples:

Slide 46: Workspaces: Set your **Current** and **Scratch** data locations to somewhere you actually have room, like your Microsoft OneDrive. Your “Current” workspace is similar to your Default geodatabase, and is initially set to the same location. **Slide 47:** Your “Scratch” workspace is where you might generate temporary intermediate data. **Slide 48:** Some of the tools have to create multiple datasets as they progress toward the end result, and these intermediate datasets are usually deleted automatically when the tool finishes.

ArcGIS Pro initially sets these as the geodatabase that’s created with your project, but you can change them if you want.

Slide 49: Output Coordinates lets you force all your new data to be in a particular coordinate system. It forces the data to be projected if it is not already in the specified coordinate system.

Slide 50: Processing Extent lets you restrict new data to a particular geographic area. **Slide 51:** This is useful when your datasets cover a large geographic area, but your analysis area is very focused.

Slide 52: The **Raster Analysis** parameters are important when a tool uses more than one raster. **Slide 53:** There are lots of times when the rasters have different cell sizes. What happens if you add two rasters that have different cell sizes? In this example, we are creating a new raster to show areas that meet both Elevation and Slope criteria. If the source DEM and Slope rasters had different cell sizes, then what cell size would the new raster have had? **Slide 54:** This “Cell Size” parameter lets you specify the value.

Slide 55: The “Mask” parameter lets you clip all your new rasters to some region of interest without using the “Extract by Mask” tool separately. However, the “Mask” parameter doesn’t work on some rasters like those produced by the **Cost Distance** tool. You’ll always get a full-sized raster with that tool. If you want to clip it to an analysis area, you need to do it in a separate step after you create the original raster.

Slide 56: The **Snap Raster** option is really useful when working with rasters.

Slide 57: Any time you analyze a raster along with another dataset (such as clipping them, or doing mathematical or logical operations between two rasters), you run the risk of shifting your new raster in a way you didn’t expect. This can happen anytime the two datasets don’t have exactly the same extent.

This has caused a lot of confusion over the years! People clip their DEM to a polygon analysis area, then they look closely and observe the cells in their clipped DEM don’t quite match the original DEM.

Slide 58: Environmental parameters apply to both **Tools** and **Models**.

Slide 59: Environments are Hierarchical! 4 levels, where higher levels override lower levels.

- 1) **Slide 60:** Level 1: Application level
 - a) Stored in ArcGIS Pro Project
- 2) **Slide 61:** Level 2: Tool level
 - a) Overrides Application-level settings, and only applies to single run of tool
- 3) **Slide 62:** Level 3: Model level
 - a) Overrides application and tool settings, applies to all tools in model.
- 4) **Slide 63:** Level 4: Model Process level
 - a) Applies to a single tool in a model, overrides general model settings.

Slide 64: Let’s look at how to get to the environmental settings. To get to the Application-level settings, go to the “Analysis” tab, then “Environments”.

Slide 65: Be careful with application-level settings! They can trip you up if you are not careful. This is an example where I wasted lots of time trying to figure out why none of the tools worked on the Miller fire. I had set the Processing Extent to the Wallow fire area, then used the same map document to try and analyze the Miller fire.

Slide 66: Here is a good example of an Application-level environment setting. You can set your Default geodatabase for this ArcGIS Pro Project easily, letting you avoid your C drives on the NAU system. And

you can set this environment setting by connecting a geodatabase with your project, and then just right-clicking on the geodatabase to set it to the default.

Slide 67: But remember this is only saved with the project. If you open a new project, you will have to reset the default geodatabase again (although there is an exception to this rule; there is a general setting in the ArcGIS Pro Options that will let you designate a single geodatabase as the default for all new ArcGIS Pro Projects.)

If you open your project-level environments window after you do this, you'll see the current and scratch geodatabase have been set to this default geodatabase.

Slide 68: Tool-level settings override any application-level settings, but they only apply to a single run of the tool. For example, if you open a tool (like the "Extract Values to Points" tool in this slide), then you can click the "Environments" command to set environmental parameters for this single run of the tool. If you run the tool a second time, you will need to set the environment settings a second time.

Slide 69: Model-level settings apply to all tools and processes in the model. The tool looks exactly the same as the Application-level environmental settings, but you get to it from the **ModelBuilder** tab.

Slide 70: Model Process-level. Applies to a single process (tool) in the model, overrides application and general model settings.

Slides 71, 72, 73 and 74: Homework questions about Environment Hierarchies.

ModelBuilder

Slide 75: Now let's look at Models and ModelBuilder. ModelBuilder provides a method of linking multiple tools together. For example, making this buffer region around high-quality roads within the Rock Project unit would normally take 3 steps.

Slide 76: In ModelBuilder you can link up the steps and run the entire series in one operation.

Slide 77: ModelBuilder has several cool qualities.

- **Slide 78:** It is easy to delete intermediate datasets.
- **Slide 79:** You can re-use your model.
- **Slide 80:** You can share your model.
- **Slide 81:** Makes it easier to document the steps you took in your analysis.
- **Slide 82:** Easy to specify which outputs should be automatically added to your map.

Slide 83: Building a model is very similar to writing code, except that you don't actually write any code.

Slide 84: Conceptually it is similar, though, because you are working with variables and functions, and using components that you often use in code (like looping structures). **Slide 85:** We'll try some basic models, although we won't get into iterations in this lab. **Slide 86:** Because of its similarity to writing code, ModelBuilder is sometimes referred to as a "Visual Programming Language".

Slide 87: Models often get quite complex. This is a model I made to analyze point densities, using a manual method that was similar to Kernel Densities.

Slide 88: Another model I made to analyze annual and seasonal current speed using various ocean datasets.

Both previous models have a lot of components, but I suspect that this level of complexity is not uncommon.

Slide 89: Model components are distinguished by their shape. Ovals are variables, Rounded rectangles are tools, Lines are connectors and Hexagons are iterators. We won't use iterators today.

Slide 90: Variables are distinguished by their colors. You can have "Dataset" variables or "Value" variables, and you have Input variables (that you specify initially) and Derived variables (that the tool creates). Then the Derived variables can serve as inputs to other tools.

Slide 91: Here we have a model with three tools, two input Dataset variables, three Derived dataset variables and one derived Value variable.

Slide 92: Components are also distinguished by whether they are filled in or if they have shadows. Hollow components are not ready to run. Filled in components are ready to run, and components with shadows have already run.

Slide 93: Models not ready to run.

Slide 94: Models ready to run and complete.

Slide 95: There are a large number of Model tools in the **ModelBuilder** tab. Especially cool is the "Auto-Layout" command.

There are several more tools that just affect how the model is drawn in the ModelBuilder window, which affect the aesthetic quality of what you see. These tools are all in the **Diagram** tab..

Slide 96: You can set individual tool parameters by double-clicking on the tool in the window.

Slide 97, 98: You can declare data to be Intermediate, meaning that it is only created to serve as inputs to later tools and you don't need to keep it forever in your hard drive. This function gives you an easy way to delete all datasets created before the final product.

Slide 99: Also a nice easy way to designate whether any derived dataset should be added to your map.

Model Variables

Slide 100: So far we've been looking at a model in which all the operating parameters are specified in the model itself. This is fine when we want to build a model that does one very specific task, but it is possible to make your model "generic" so that it is useful for different scenarios. For example, the illustrated model is specifically designed to select a specific road ('Fry Park Road'), buffer it, and then clip the buffers to a specific polygon. You construct and run it right inside the ModelBuilder window.

Slide 101: What if the model works well, and you want to run it again on different roads? As it stands now, you need to manually change the tool parameters and rerun the model each time.

Slide 102: There is a better way!

We can set any of the tool parameters to be "Model-level" parameters (aka "Model Variables"). When you set Model Variables, then you can run the model as a stand-alone object, just like any individual tool.

ArcGIS will ask you to specify the Model Variables in a single window, and then run the entire model.

Slide 103: For example, the first tool in the model is "Select Layer by Attribute", and it requires you to set the input layer and the "Select by Attribute" SQL Query String:

Slide 104: It is currently preset to make a new selection of roads named "Fry Park"

Note that this tool requires an input dataset plus 3 operating parameters (the *Selection Type*, the *Expression* and whether we want to invert the expression). We can set any or all of these 3 parameters, or even the input dataset, to be “Model-Level Parameters”:

Slide 105: For this example, we will only make the query expression a model variable:

Slide 106: Now the model looks a little different. Notice that the “Expression” has now been separated into a new element, connected with a line to the “Select Layer by Attribute” tool.

Slide 107: If we right-click on this new “Expression” element, we can set it to be a model parameter.

Slide 108: When we do so, we see a “P” above the new empty input variable “Expression”.

Slide 109: We also want to make the output feature class a model variable so the user can set it each time. Otherwise it will get overwritten each time we run the model.

Slide 110: Now the model has two model-level variables (each with a “P” in the upper right corner):

Slide 111, 112: We can now double-click on the model itself, set these two parameters, and get a new output buffer feature class.

Slide 113: There is a fairly decent Esri virtual campus course you can take that goes into a little more detail on ModelBuilder than we are going into here. It doesn’t describe Iterators, but it does show how you can use a model as a single component inside larger models.

Python

Slide 114, 115: Python: There is just no way we can get into this in enough detail to make it worthwhile. I think you could easily make a full-semester 3-credit course just on Python.

Slide 116: Lots of resources available, though. Books, web resources, **Slide 117:** and Esri Virtual Campus courses.

Slide 118: In general, Python is very popular in a wide variety of fields. **Slide 119:** There are lots of programming environments you can use to write Python code, and **Slide 120:** it’s used in more software than just GIS software, **Slide 121:** and in more GIS software than just ArcGIS. **Slide 122:** QGIS and Google Earth Engine both offer ways to run analyses using Python. **Slide 123:** The language is also very popular in engineering, science and math software in general. **Slide 124:** If you learn Python, you can use it lots of places!

Slide 125: In ArcGIS, we usually use Python as a scripting language to **Slide 126:** automate the use of **existing** tools. It is not used as much to create brand new tools.

Slide 127: If you’re a good Python programmer, though, you can create your own tools!

Slide 128: Almost all existing Toolbox tools can be run in Python, by typing into the Python command window.

Slide 129: Pro offers two environments for running Python code. **Slide 130:** The Python window is a good environment for copying in scripts or typing Python code directly. The Jupyter Notebook is a more sophisticated environment that lets you save in multiple Python functions, add documentation and illustrations, then save the entire package in a single object.

Slide 131: You can also run Python code in the Field Calculator tool. You can even set up multiple functions, and have each function call others, all inside that window.

Slide 132: We aren't going to get into language structure and code conventions and rules in our brief class period, but it would be worth your while to learn Python! **Slide 133:** You'll be the ArcGIS wizard in the room if you know Python.

Slide 134, 135: Fortunately, for the time being, you can actually use Python without knowing how to write it. **Slide 136:** It is often easier to **modify** Python code (or any computer language) than it is to write it from scratch.

Slide 137: Whenever you run a tool, you can get a "snippet" of Python that would have run the tool with the exact same parameters that you set.

Slide 138: You can even get this snippet of code from the tool before you've run it. Just click the little down-arrow next to the "Run" tool, and there's an option to copy the code. Sometimes you still want to run the tool, just to be sure it's doing what you want it to, but you can get the code either before or after running the tool.

Slide 139: It's actually a common strategy to run the tool a single time just to get a clean set of Python code, which you can be confident will do a particular task.

Slide 140: Once you get that Python snippet, you can then **Slide 141:** copy that snippet, possibly hundreds of times, and just modify each copy with a slightly different parameter value. **Slide 142:** Then they'll copy the entire set of hundreds of commands back into the Python window, then just go to lunch or something while it works. **Slide 143:** There's a demo of how to do this at the end of the video on Lab Exercise 8.

Slide 144: You can also get samples of Python code in the help windows for the individual tools.

Slide 145: A word of warning! **Don't** modify Python code in a fancy word processor like Microsoft Word! These word processors often modify the Quotation symbol into an "Open Quote" and "Close Quote" using different characters, and these characters are not valid Python code.

Slide 146: If you are interested, I have an example of a combination of Python and ModelBuilder I did recently. A former student contacted me about a month ago, to discuss methods for calculating the slope at regular intervals along polylines based on elevation from a DEM. This was Especially tricky because the DEM and polylines would be in Latitude/Longitude (i.e. angular) units.

Slide 147: I made him a model that would:

- **Slide 148:** Intersect his polylines with the DEM surface to create PolylineZ features with vertices roughly every 30m.
- **Slide 149:** Convert the PolylineZ features to points.
- **Slide 150:** Add attribute fields for Latitude, Longitude, Elevation, Vertical and Horizontal Distance from Previous Point, and Slope from Previous Point.
- **Slide 151:** Calculate these values using Python code, with the code embedded in the "Calculate Field" tools.

Slide 152: It might be an educational example if you want to look it over.

- See "**OPTIONAL: Sample Model and Data**" on BBLearn, in folder *Basic Geoprocessing and Automating Analyses*.

Slide 153: This is an image of the actual model. Notice that it uses model parameters, so the entire thing can be run by double-clicking on the model itself.

Geoprocessing and Automation GIS Lab Exercises:

Heads-Up: What you will turn in from these labs...

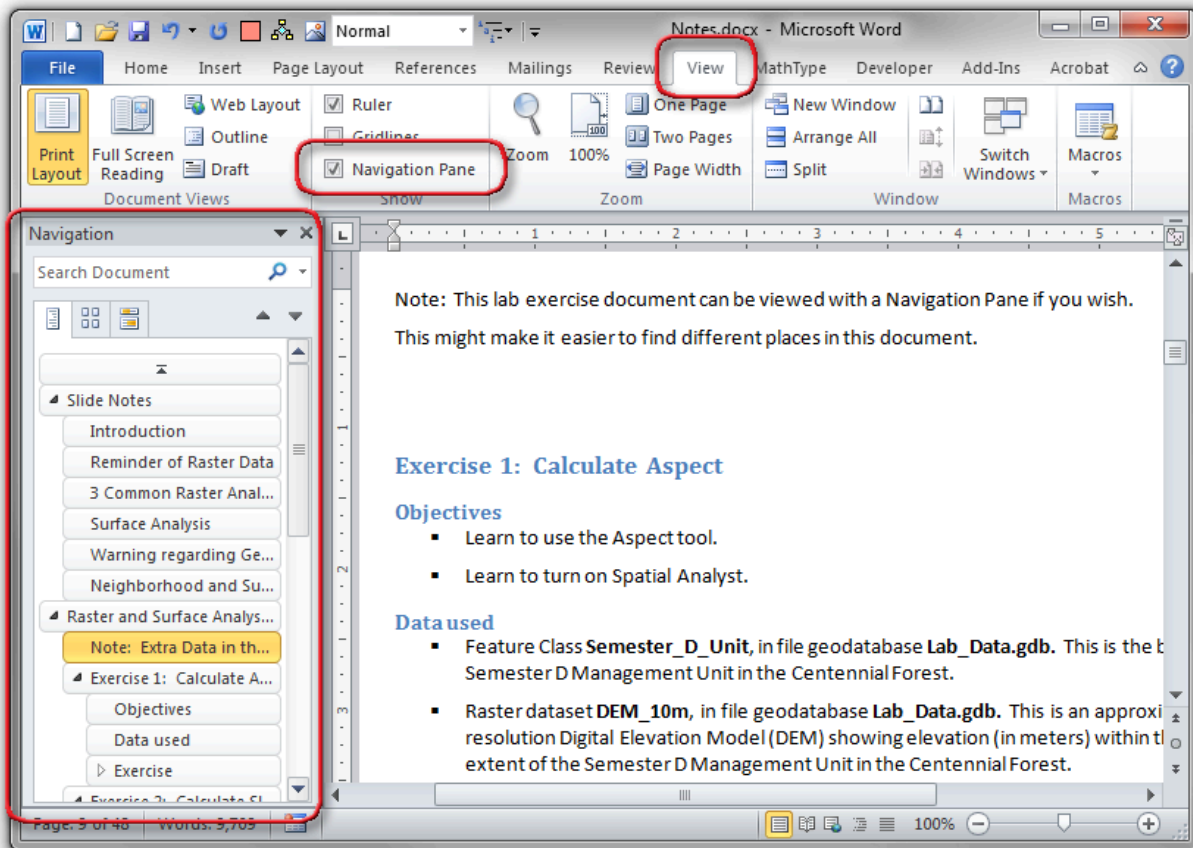
Your homework assignment for this lecture will be based entirely on these labs and the 4 questions on Slides 71 – 74. For the labs, you will create and turn in a file geodatabase containing the following data:

- From Lab Exercise 1 (**Geodatabase**): You will create a geodatabase to contain data from rest of labs.
 - There is also a question to answer from Lab Exercise 1.
- From Lab Exercise 2 (**Merge** tool): Polygon feature class **Merged_Project_Areas**
 - There is also a question to answer from Lab Exercise 2.
- From Lab Exercise 3 (**Dissolve** tool): Polygon feature class **Rock_Project_Unit**
- From Lab Exercise 4 (**Buffer** tool): Two polygon feature classes:
 - **Highway_Legal_Yearlong**
 - **All_Vehicles_Yearlong**
- From Lab Exercise 5 (**Union** tool): Polygon feature class **Roads_Union**
 - This exercise also adds an attribute field to the feature classes **Highway_Legal_Yearlong** and **All_Vehicles_Yearlong** from Lab Exercise 4.
- From Lab Exercise 6 (**Clip** tool): Polygon feature class **Roads_in_6A**
- From Lab Exercise 7 (**ModelBuilder**): Several items
 - A toolbox with a model
 - The model will produce an **intermediate** dataset name **Coconino_NF_MVUM_Road_Buffer**. **This dataset should be deleted from the geodatabase!** Part of the lab exercise is deleting intermediate data, so you will lose a few points if this intermediate dataset is still present.
 - The model will produce a **final** dataset named **Rock_Project_Buffer**. **This dataset should be present in the geodatabase!**
- From Lab Exercise 8 (**Python**): Python script version of your model from Lab Exercise 7.

There is also a question to answer from Lab Exercise 8.

Note: The Navigation Pane

If you wish, and if you didn't already know about it, Word includes a navigation pane that might make it easier to find different places in this document.



Exercise 1: Set the Default Geodatabase for your ArcGIS Pro Project

Objectives

- Learn how to find the current default geodatabase for your ArcGIS Pro Project.
- Learn to set the default geodatabase for your ArcGIS Pro Project.
- Learn more ways to effectively use the Catalog window in ArcGIS Pro.
- Observe how the default geodatabase is saved with the ArcGIS Pro Project.

Data used

- Point Feature Class **Plots_RockPrj_CF_2014**, in folder `..\Class_GIS_Data\RockPrj_CF_2014\..`

Homework Assignment:

For your homework, you will send me the file geodatabase you create for this exercise. This file geodatabase should be named **Your_Name_Automation_Lab.gdb**.

Exercise

You may have realized by now that you have a **Default** geodatabase in ArcGIS Pro. You have probably seen it almost every time you have run a geoprocessing tool. This is the default location where all your tools have tried to save your new datasets. Unless you have changed the location of your default geodatabase, or changed some of the basic rules that ArcGIS Pro follows (for example, changing the ArcGIS Pro General Options), then ArcGIS Pro will use a default file geodatabase located in the same folder as your project file. ArcGIS Pro actually creates this default geodatabase as part of any new ArcGIS Pro Project.

This **default location** is usually a convenient place to keep data. If you can find your ArcGIS Pro Project file (which ends in *.aprx), then you can find the default geodatabase because it's in the same folder.

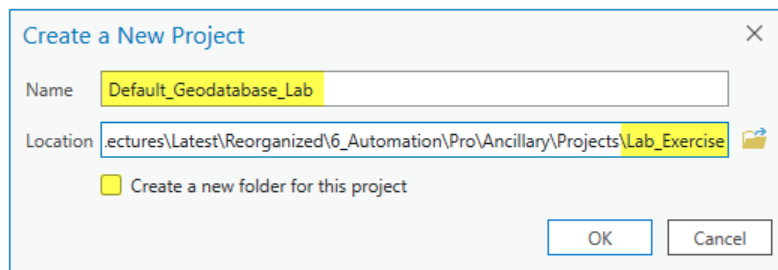
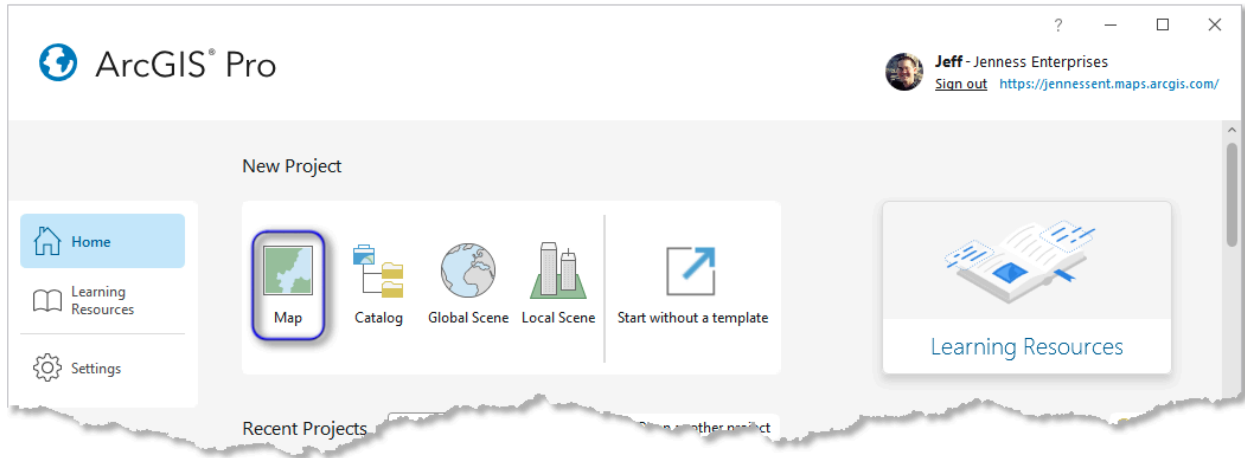
However, sometimes you want to change that default geodatabase. It might be more convenient for you to keep a library of GIS data in a different location where multiple people or projects can access it.

This exercise will show you how to create a new file geodatabase and set it to be the Default geodatabase for your ArcGIS Pro Project.

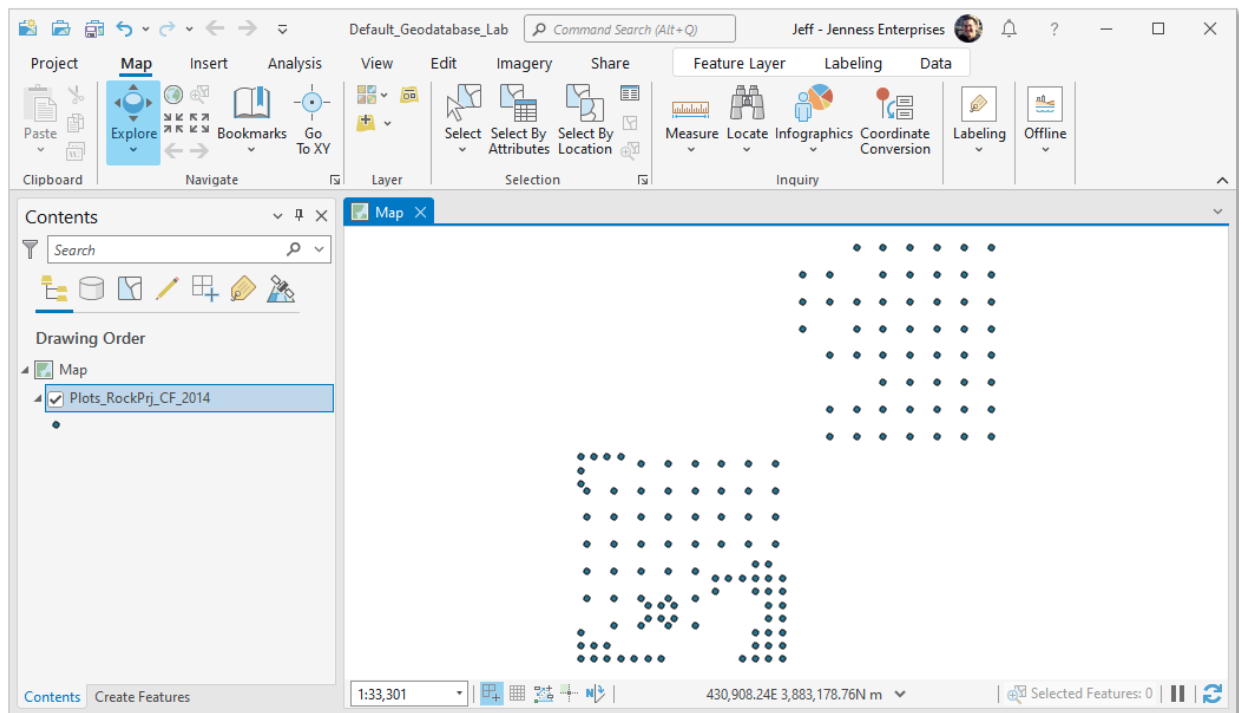
Just to be clear: When you run a tool that creates new data, you can save it anywhere you want. The tools give you an option to say where you want to save the data. This **Default** geodatabase is just the location where the tool **initially** offers to save the data. If you don't explicitly change the output location, then it will go into the Default geodatabase.

Prepare your ArcGIS Pro Project

- 1) Decide where you are going to save your data for this lab exercise. You will need to create both an ArcGIS Pro Project and a file geodatabase, so I recommend you create a new folder somewhere on your Microsoft OneDrive to hold your new data.
 - a) **Note:** You can use either the **ArcGIS Pro Catalog View** or **Windows Explorer** to create your new folder. Please call me over if you need help creating a new folder.
- 2) Open a new ArcGIS Pro Project and **save it to your new folder**. The illustrations below will assume you saved your project with the name **Default_Geodatabase_Lab**, in a folder named **Lab_Exercise**.

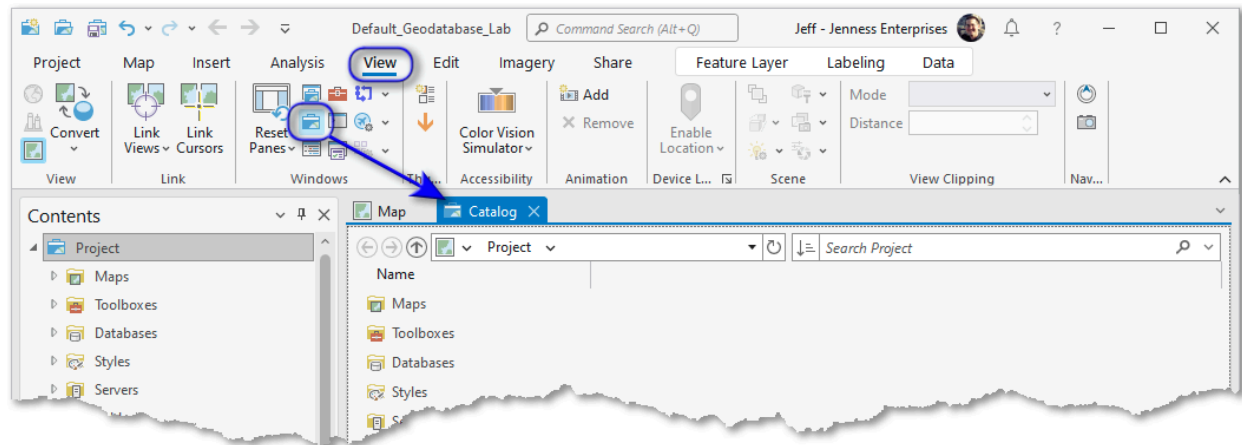


3) Load the shapefile **Plots_RockPrj_CF_2014** from the folder **..\Class_GIS_Data\RockPrj_CF_2014\..**

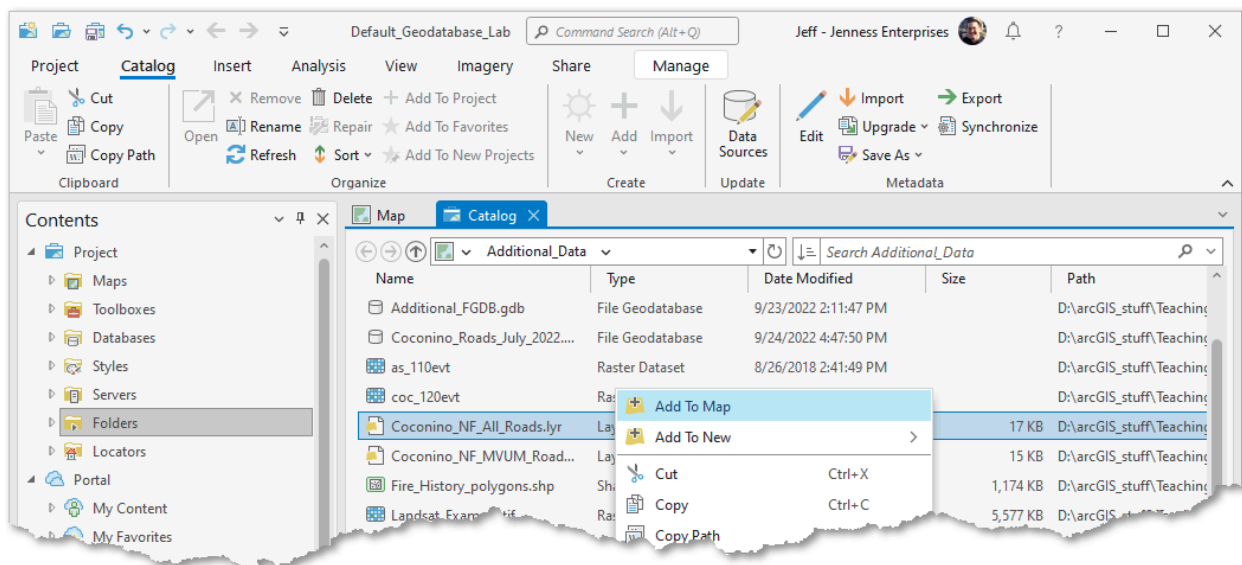


Check the Current Default Geodatabase:

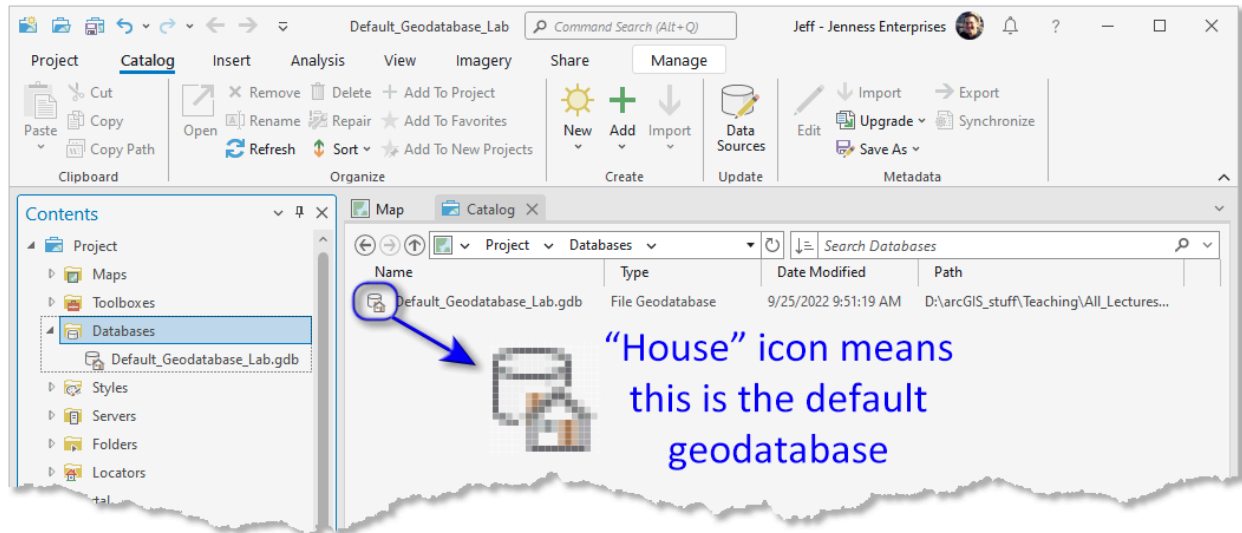
- 4) If the Catalog view is not already open, click the “View” tab, then the “Catalog View” command 



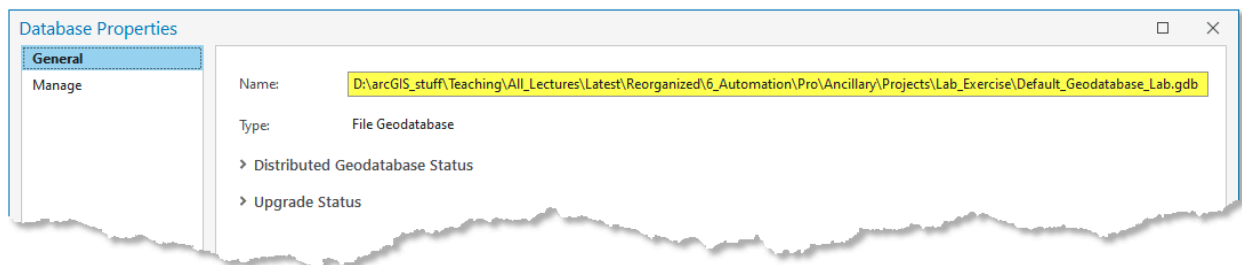
- **Tip:** The **Catalog View** shows all the workspaces you have connected to in this project. You can open any folder, database or server you have connected to and see the data in that workspace. You can also easily add that data to your map by just dragging it into the map, or right-clicking on the data and choosing “Add to [Your Map Name]” or “Add To New...”.



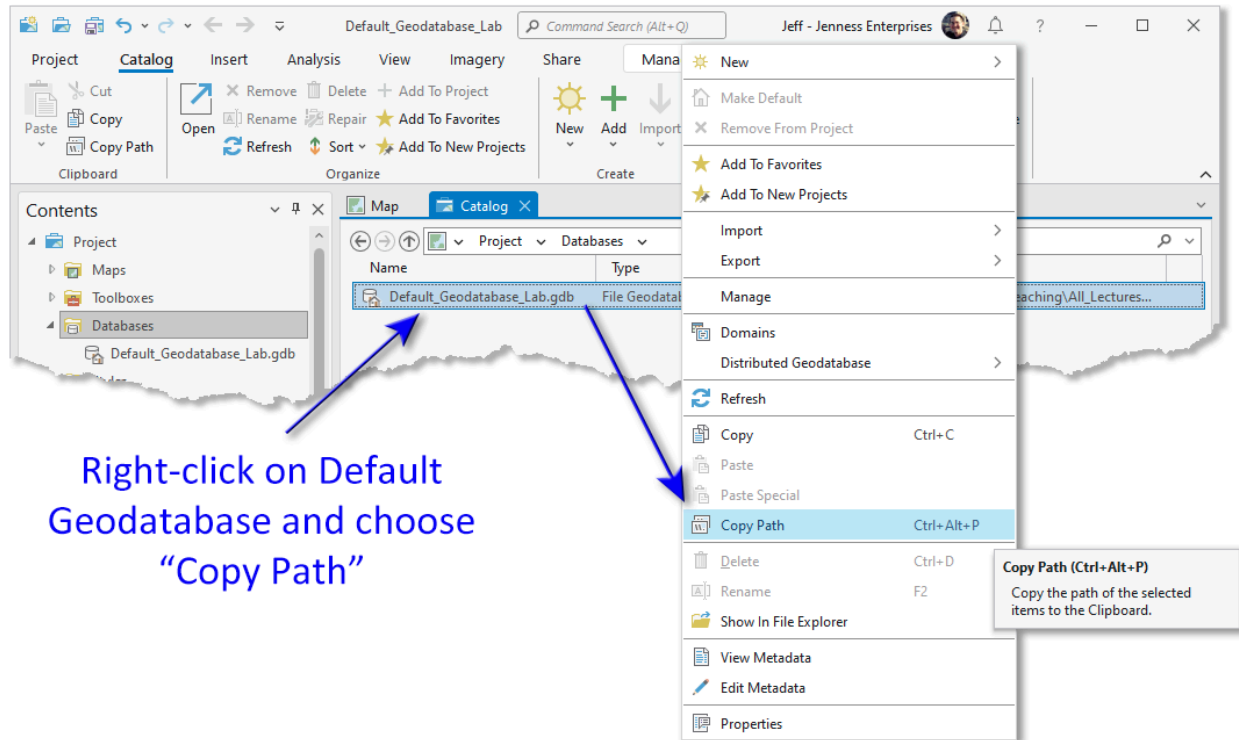
- **Tip:** This **Catalog View** is also an excellent way to delete data. Just select the data in the Catalog view and hit the [Delete] key. If any of the datasets are currently loaded as layers into any maps in this project, then the layers will be removed automatically. **However**, if any of the layers are loaded into any other open ArcGIS Pro projects or ArcGIS Pro Projects, then you won't be able to delete them. You need to close those other applications first.
- 5) Check the location of your Default Geodatabase:
- a) Go to your **Contents** pane, then click on **Databases** to see what geodatabases you are currently connected to. You should see your default geodatabase symbolized with a little “House” icon:



- b) Where is this geodatabase located? You might be able to determine the answer by looking at the “Path” column in your Catalog view. Another way is to right-click the geodatabase and choose Properties, and you’ll see it in the “General” tab:



- c) Another good way is to right-click on your default geodatabase and choose “Copy Path”:



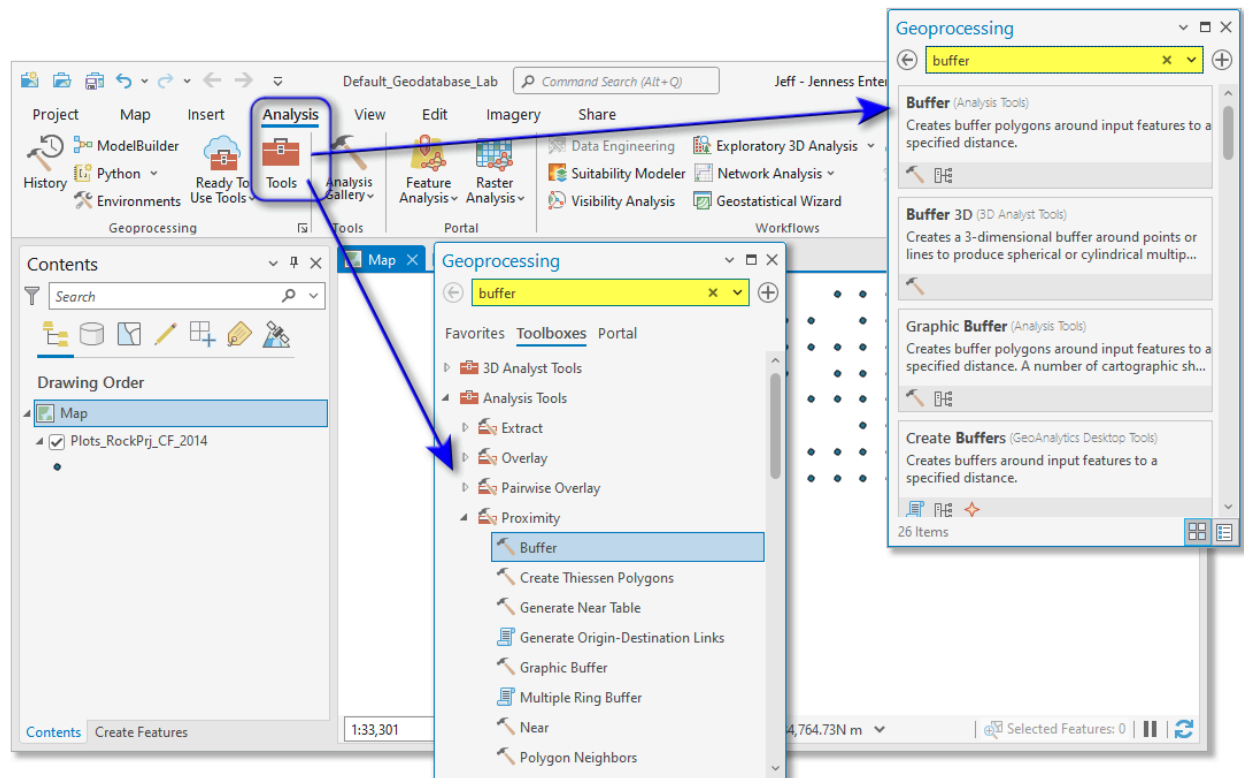
d) Based on the database properties, I can see that my default geodatabase is located at:

`D:\arcGIS_stuff\Teaching\All_Lectures\Latest\Reorganized\6_Automation\Pro\Ancillary\Projects\Lab_Exercise\Default_Geodatabase_Lab.gdb`

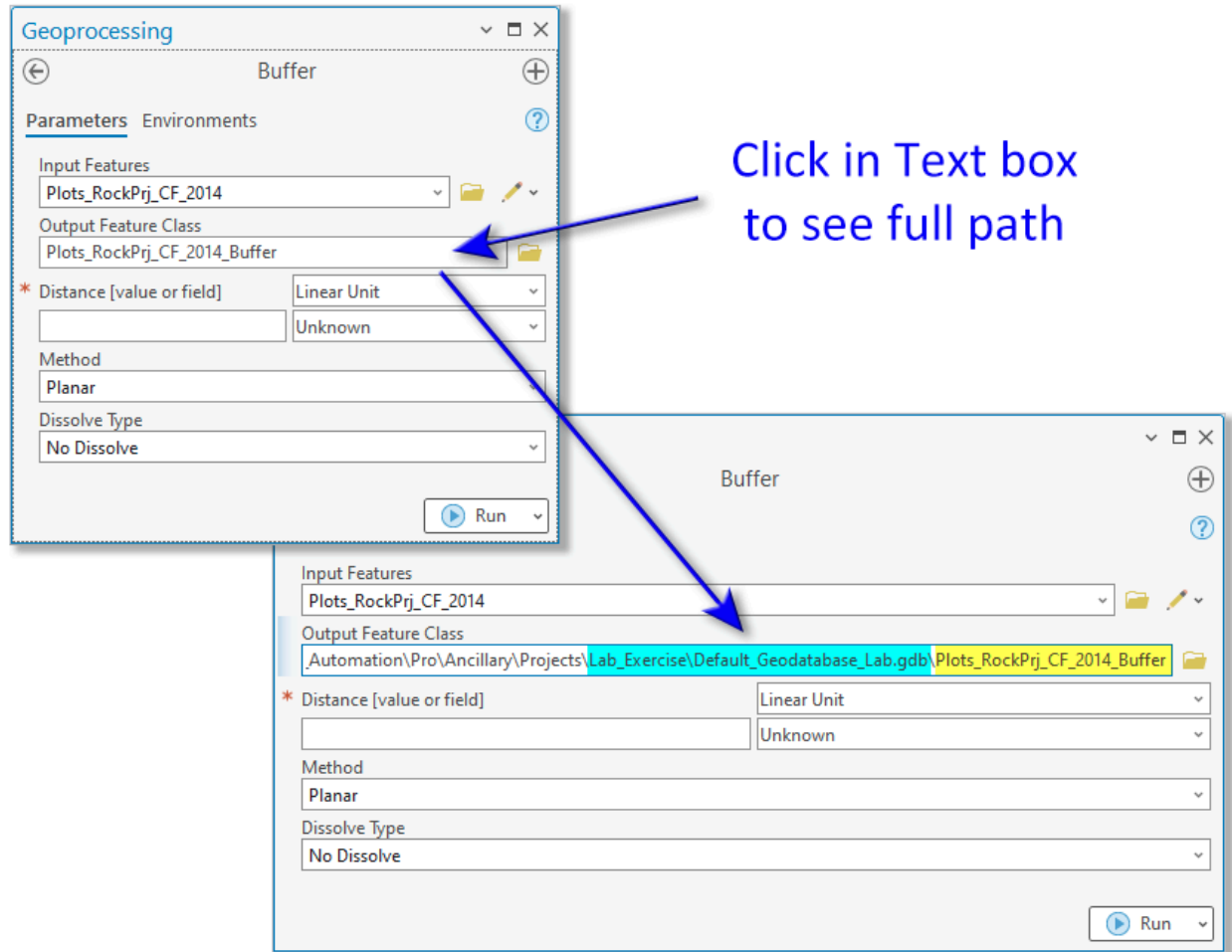
- 6) **Question:** Where is **your** default geodatabase located? **Hint:** The "Right-Click", then "Copy Path" is a very easy way to get this answer.
- a) **Remember**, this is the location automatically designated **before you have changed it**. We will change the location shortly.

Observe how the tools use your Default Geodatabase:

- 7) Return to your map, then open your **Buffer** tool (Click the "Analysis" tab, then "Tools". You'll find the **Buffer** tool in "Toolbox", then "Analysis Tools", then "Proximity", then "Buffer". You can also just search for "Buffer" in the geoprocessing pane.):
- a) **Note:** You will not actually run this tool! You are just going to look at the "Output Feature Class" and then close it. This step is only to illustrate how tools use the Default Geodatabase.



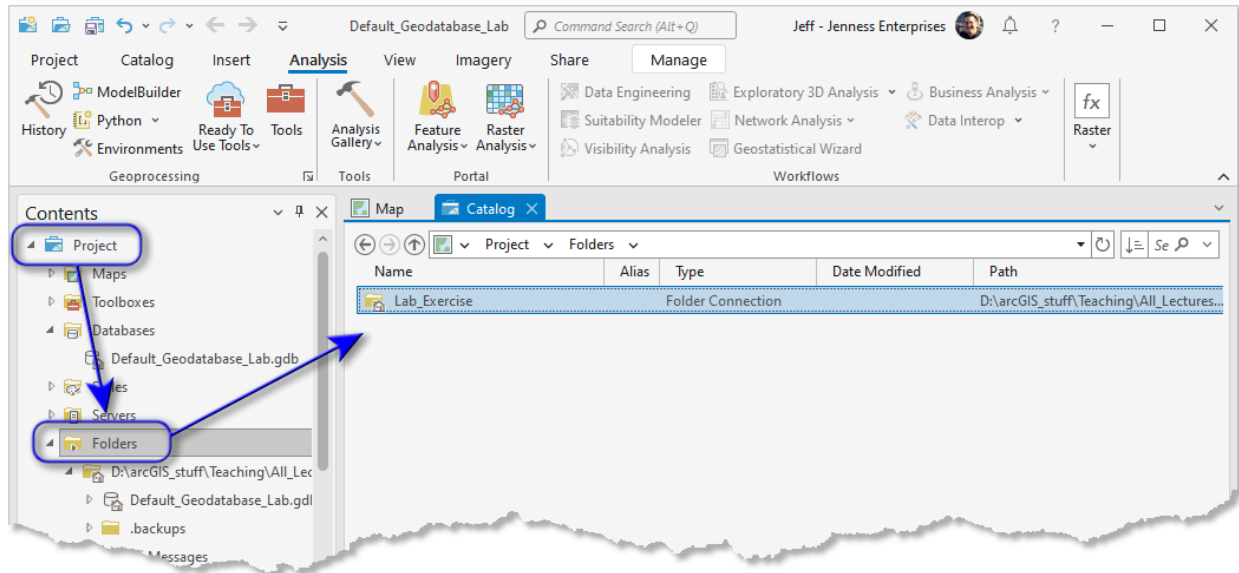
- 8) In the Buffer window, set your Input Features parameter to **Plots_RockPrj_CF_2014**.
- 9) After you leave the “Input Features” box, the tool should automatically fill in a name and location to save the Output Feature Class. Observe that it automatically chooses your Default Geodatabase to save the data.
 - a) Initially you will likely only see the suggested dataset name reported in the “Output Feature Class” text box. If you click in the box, though, you should see the full path:



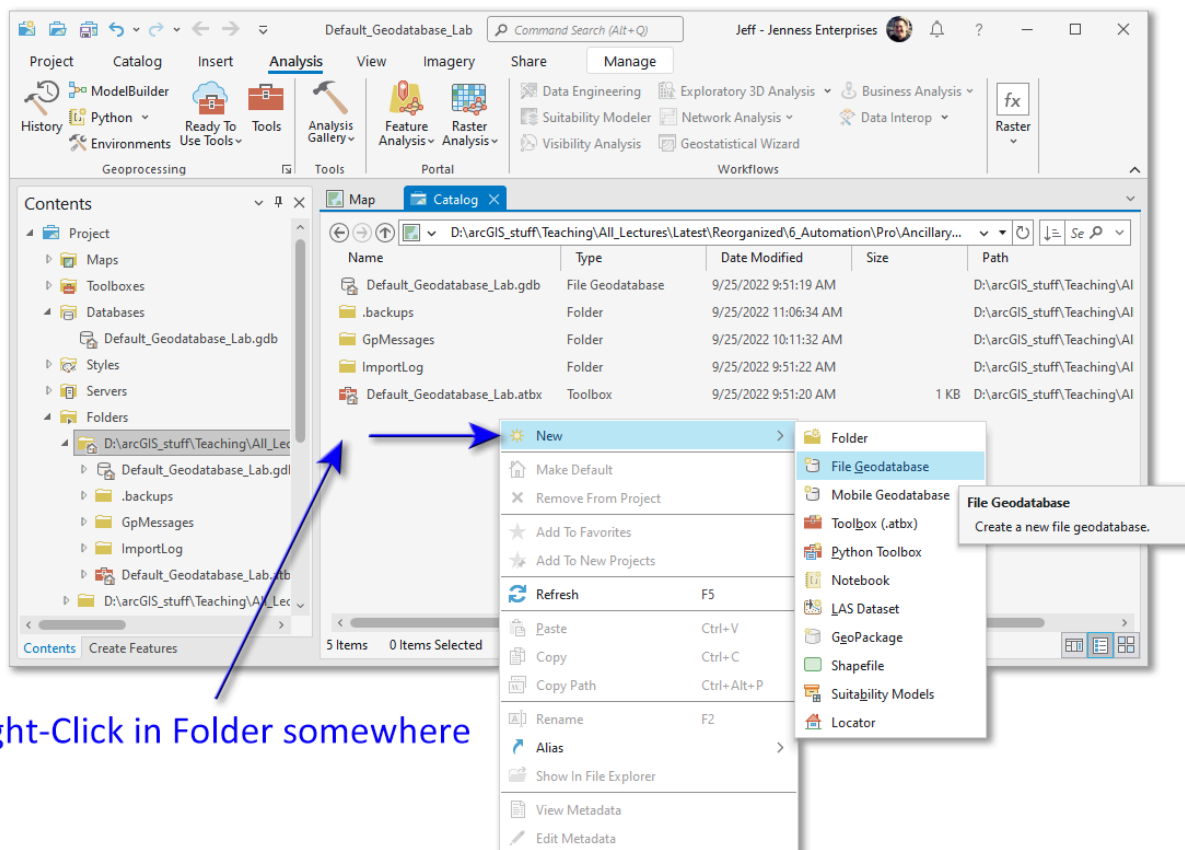
10) Close the Buffer tool without running it.

Change your Default Geodatabase for this ArcGIS Pro Project:

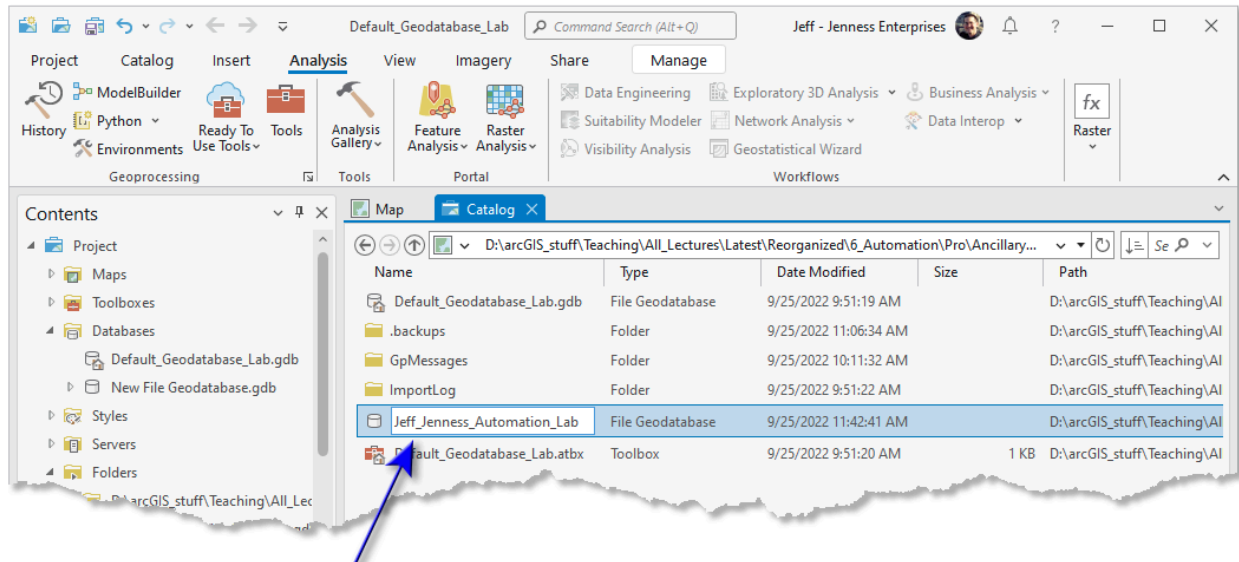
- 11) First make a new geodatabase. You will send me this geodatabase as part of your homework, so please name it correctly.
- a) I suggest you put your new geodatabase in the new folder you created at the beginning of this lab (I will put it in the folder “Lab Exercise” in these illustrations). You can reach it in the **Catalog** view by going to your **Contents** Pane, looking in the **Project** section and opening the **Folders**:
 - i) **Note:** The place you create your new geodatabase is not as important as the fact that you remember where you put it. Also, if you are on the NAU system, make sure you put it on your **Microsoft OneDrive** drive somewhere.



- b) Name it **Your_Name_Automation_Lab.gdb** (obviously substituting your name for “Your_Name”).
- c) You can create the geodatabase easily in the **Catalog** view. First open the new folder you created, then right-click inside it and choose “New”, and then “File Geodatabase”. When the new geodatabase is created, you can change the name by typing in a new one.



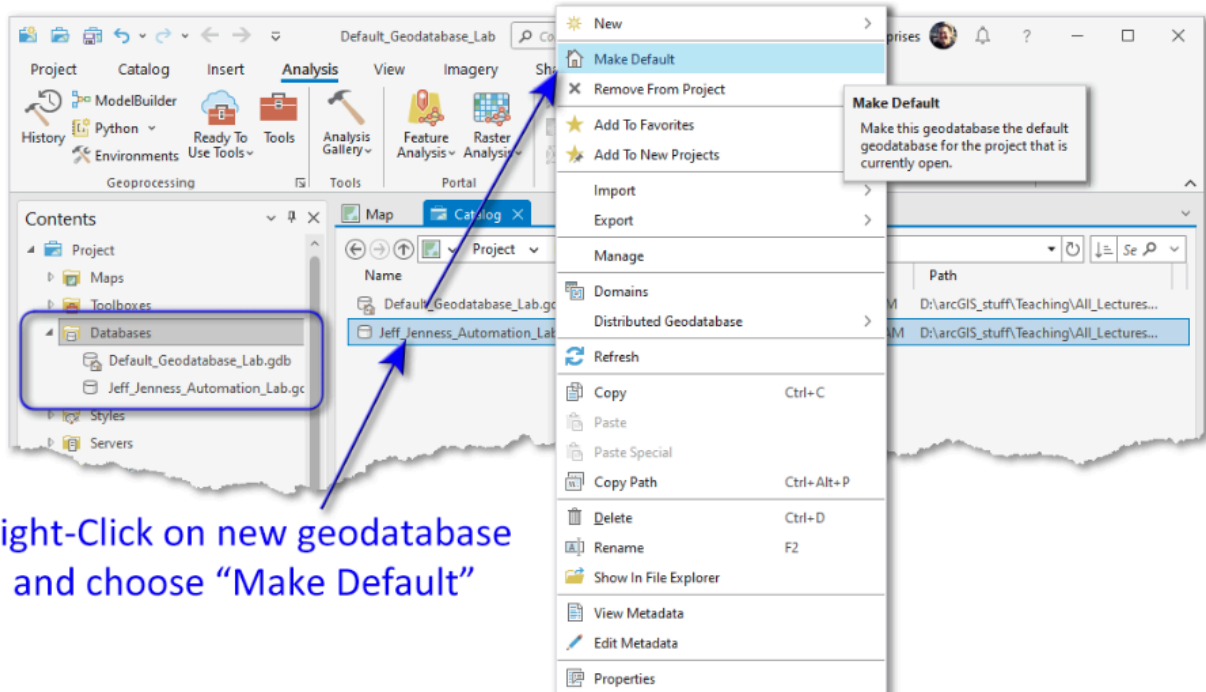
Right-Click in Folder somewhere



Type in new name

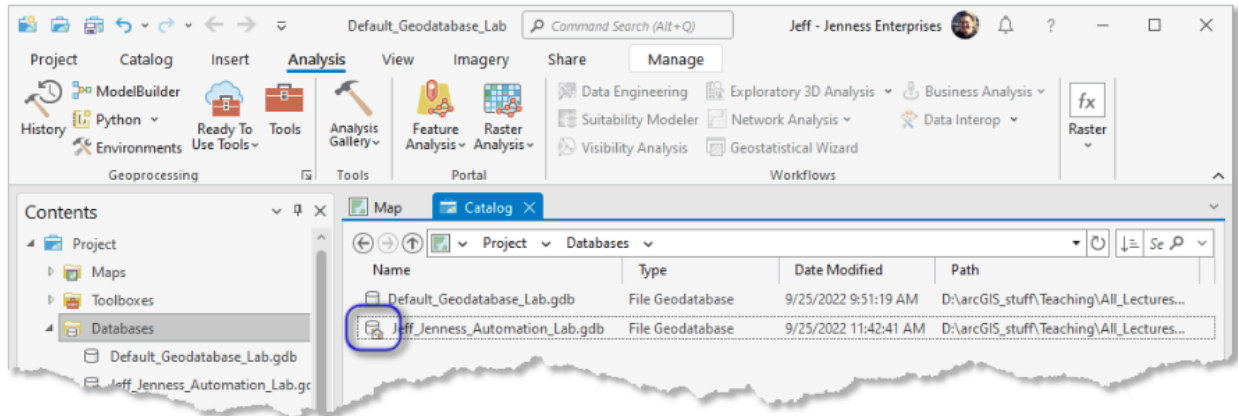
12) Set your new geodatabase as the Default Geodatabase for this map document.

- Important:** In your **Catalog** view, click on the “Databases” in your “Project” section. You can only set the database to be the Default if you are in the “Databases” section.
- Right-click on your new geodatabase and choose “Make Default”.



Right-Click on new geodatabase and choose “Make Default”

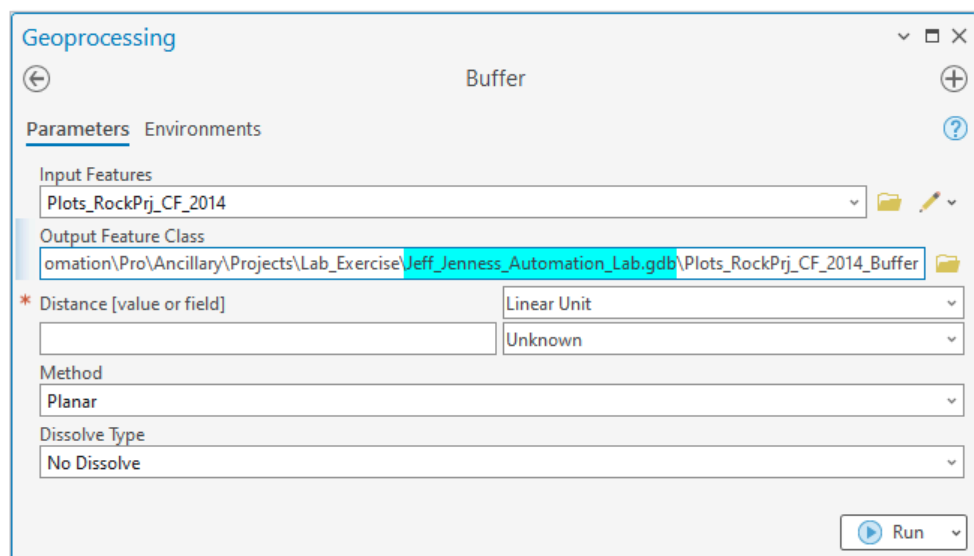
- Observe that your new geodatabase now has a file geodatabase symbol with a little house on it, indicating that it is now the default geodatabase:



Observe that the tools now default to this new geodatabase:

13) Open your **Buffer** tool again (in Toolbox, then “Analysis Tools”, then “Proximity”, then “Buffer”) and set your Input Features parameter to **Plots_RockPrj_CF_2014**.

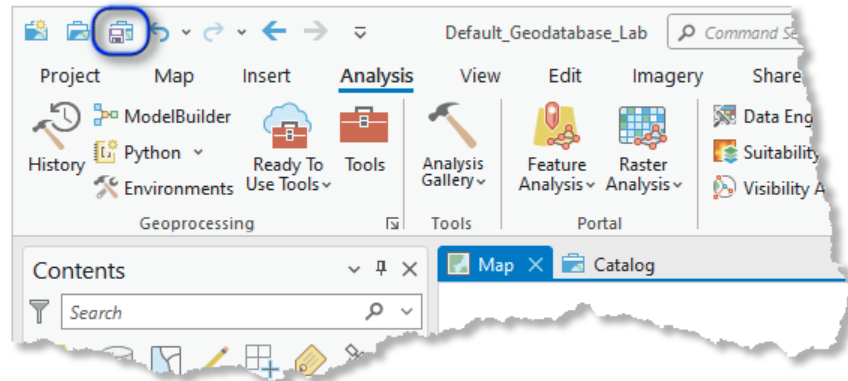
14) Observe that it now chooses your new Default Geodatabase to save the data.



15) Close the Buffer tool without running it.

Observe that the default has been saved with your ArcGIS Pro Project:

16) Save your ArcGIS Pro project by either clicking [Control]-S, or by clicking the “Save” command 



17) Close your ArcGIS Pro project and reopen it.

18) Confirm that your default geodatabase is still set to your new geodatabase

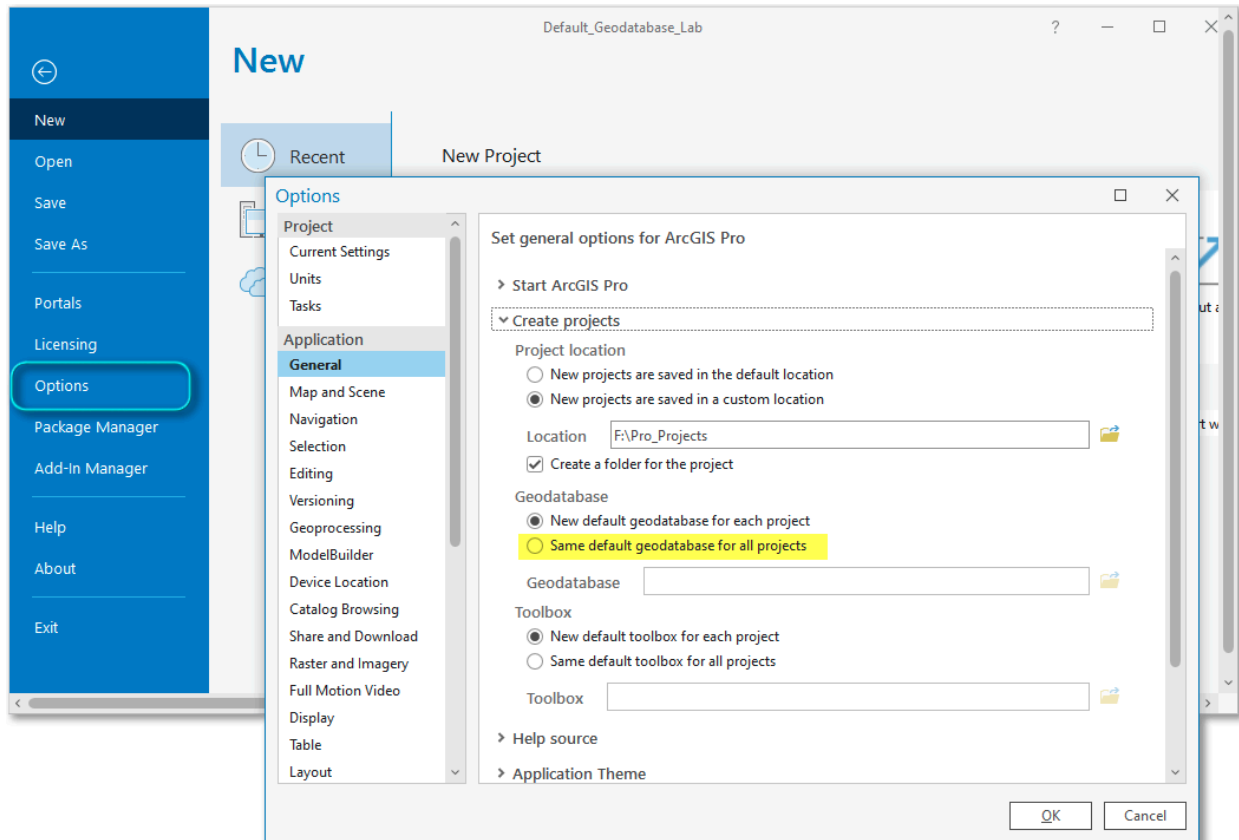
Observe that your default geodatabase is only changed for this project:

This is an important point, so please make sure you understand what is going on here. When you changed your default geodatabase, you **only** changed it for your current ArcGIS Pro Project. Whenever you open that project again, it will remember to use your new geodatabase as the default.

However, this doesn't apply to any other ArcGIS Pro Project, or to ArcGIS Pro in general. If you open a new project, that new project will use a different default geodatabase.

19) Open a new **blank** ArcGIS Pro Project and find the default geodatabase (use the **Catalog** view like you did above). Observe that it is not the default location you specified in this lab exercise.

Tip: If you ever find yourself in the position of wanting every new ArcGIS Pro project to use the same default geodatabase automatically, you can set this rule in the ArcGIS Pro **Options**, in the "General" tab, and in the "Create projects" section (see below):



Exercise 2: Learn the Merge tool

Objectives

- Learn how to use the Merge tool.
- You will create a new dataset that combines 4 existing datasets.

Data used

- Feature Classes **Eastsideprojectarea**, **EldenSmallprojectarea**, **JackSmith_Schultzprojectarea** and **Uppersprucewatershed_boundary**, all located in folder `..\Class_GIS_Data\Dry_Lake_Hills\..`. These are all boundaries of project areas in and around Flagstaff, and especially in the Dry Lake Hills area, between Mt. Elden and the San Francisco Peaks.
- File Geodatabase **Your_Name_Automation_Lab.gdb**, which you created in the first lab.

Homework Assignment:

There is one homework question in the lab exercise below, and you will add a new feature class named **Merged_Project_Areas** to the **Your_Name_Automation_Lab.gdb** geodatabase you created in Exercise 1. **You will send me the geodatabase as part of your homework.**

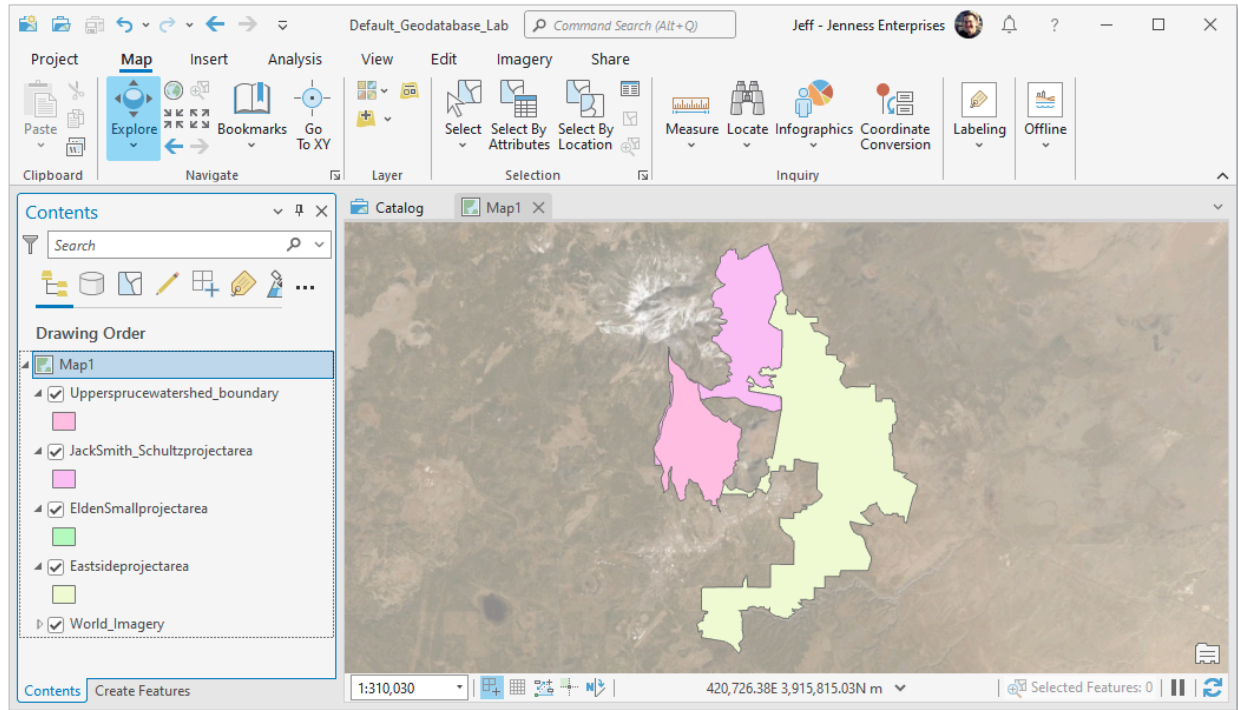
Exercise

This lab exercise will show you how to use the **Merge** tool to combine multiple feature classes into a single feature class. You will also see how the **Merge** tool works with feature classes that have different attribute fields.

Note: The **Merge** tool is very similar to the **Append** tool. The only difference is that the **Merge** tool combines multiple feature classes into a new feature class, while the **Append** tool adds feature classes to an existing feature class.

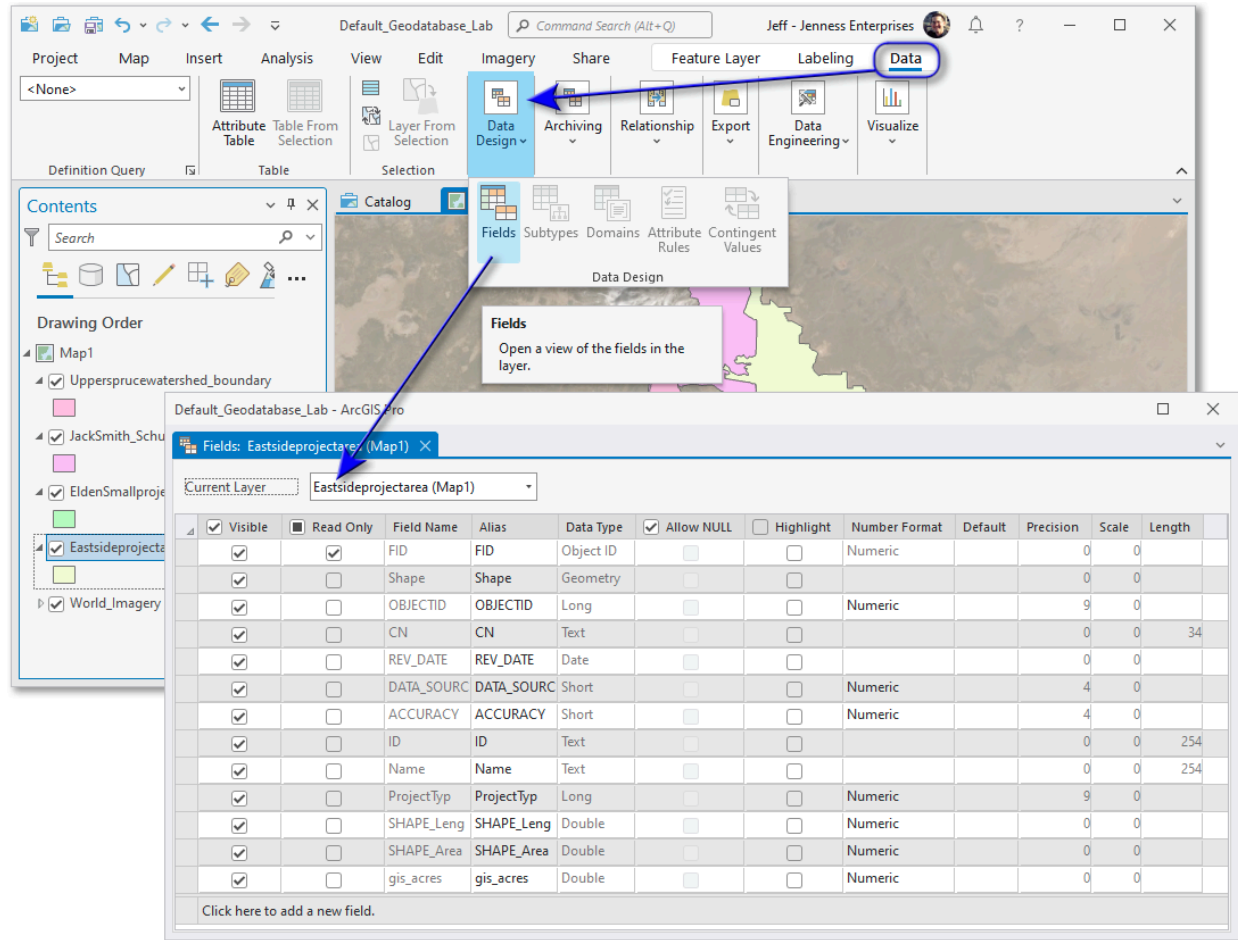
Prepare your ArcGIS Pro Project

- 1) If it is not still open, then open your ArcGIS Pro Project from Lab Exercise 1. Confirm that your default geodatabase is **Your_Name_Automation_Lab.gdb**. If it is not, then follow the instructions in Lab Exercise 1 to set the default geodatabase correctly.
 - a) **Note:** **You will be sending me this geodatabase as part of your homework**, so it is important that you have it set as the default. Otherwise you may not correctly save the required feature class to the database.
- 2) Open a new map and load the shapefiles **Eastsideprojectarea**, **EldenSmallprojectarea**, **JackSmith_Schultzprojectarea** and **Uppersprucewatershed_boundary** from the folder `..\Class_GIS_Data\Dry_Lake_Hills\..`
- 3) If you like, feel free to add a background dataset like **USA_Topo_Maps** or **Imagery**, just to give a general reference to the project. These background datasets will not be used in the lab, though.



Review Attribute Fields

- 4) One of these feature classes is not like the others. Three feature class have exactly the same 13 attribute fields, and one only has 3 fields. You can review the attribute fields by opening the **Fields** view for each feature class: You can get to the **Fields** view by doing the following:
 - **Select the Layer** in the **Contents** pane.
 - Click the **Data** tab, then the “Data Design” group, and choose **Fields**:

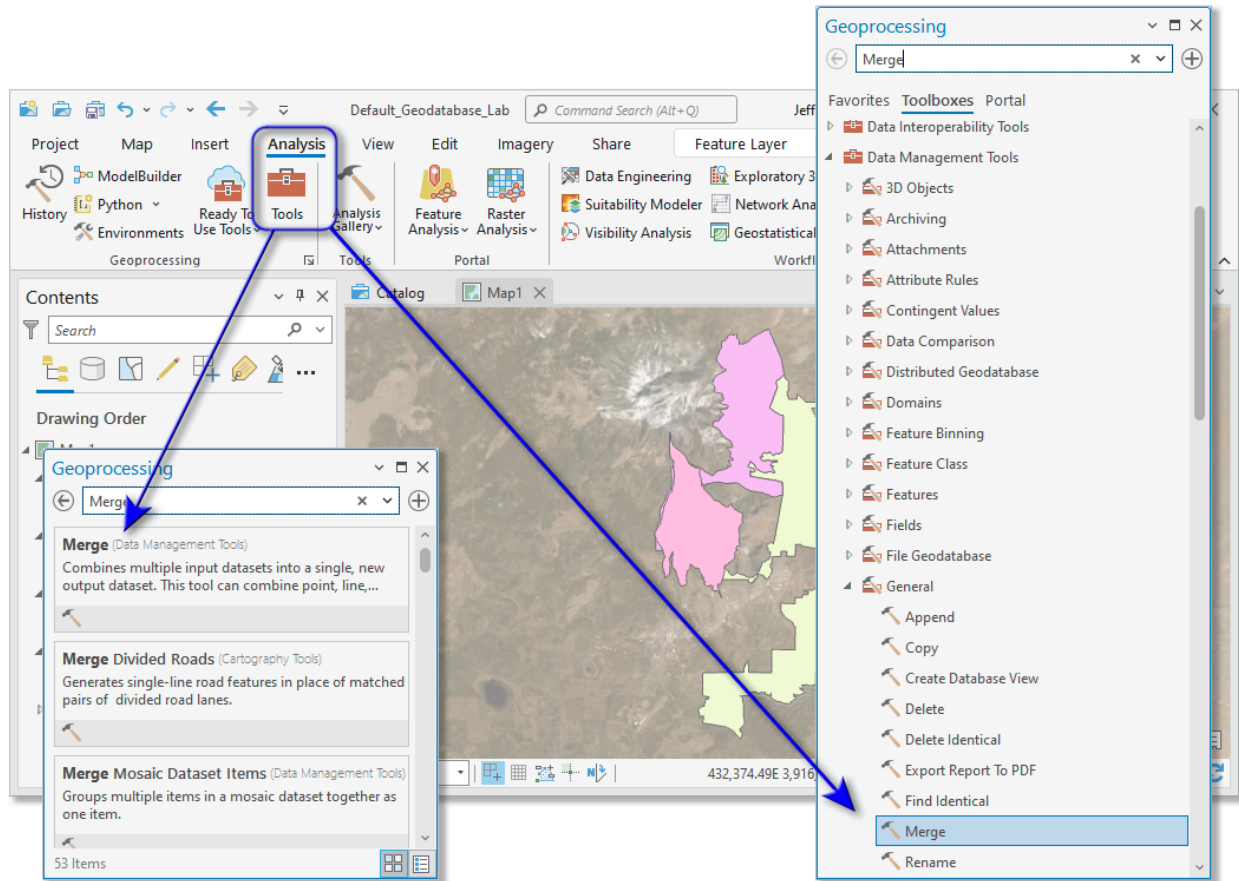


- Each attribute field is described in a separate row. Once you have reviewed the attribute fields of the selected layer, then simply follow the steps above to open the **Fields** view for the other three feature classes.

5) **Homework Question:** Which feature class has only 3 attribute fields?

Open and run Merge tool

- Open your **Merge** tool (Click the "Analysis" tab, then "Tools". You'll find the **Merge** tool in "Toolbox", then "Data Management Tools", then "General", then "Merge". You can also just search for "Merge" in the geoprocessing pane):
 - Clicking the **Geoprocessing** menu, and then choosing **Merge**:

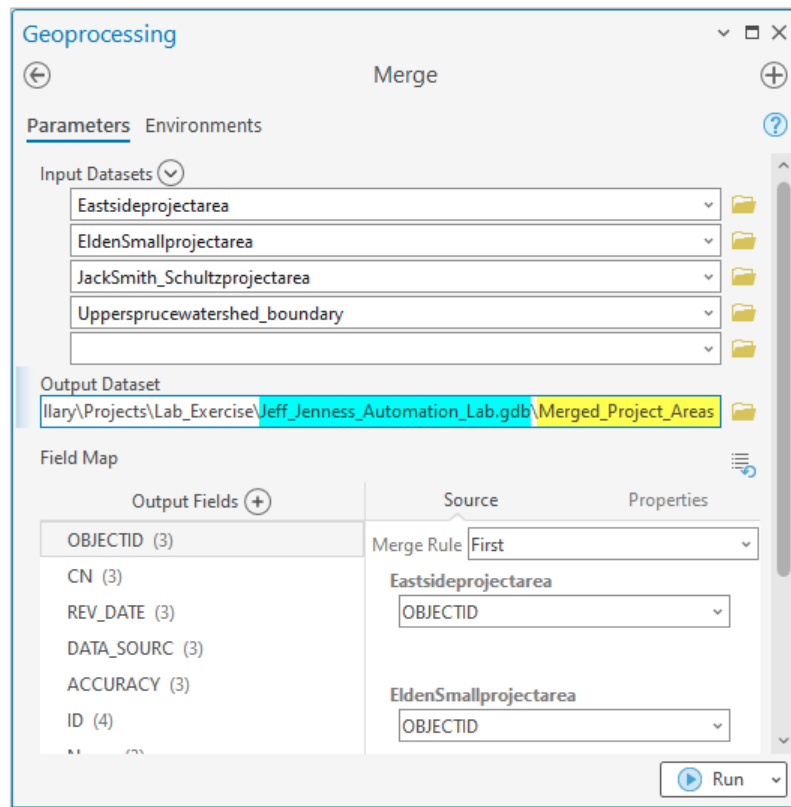


b) In the **Merge** parameters window, set the following:

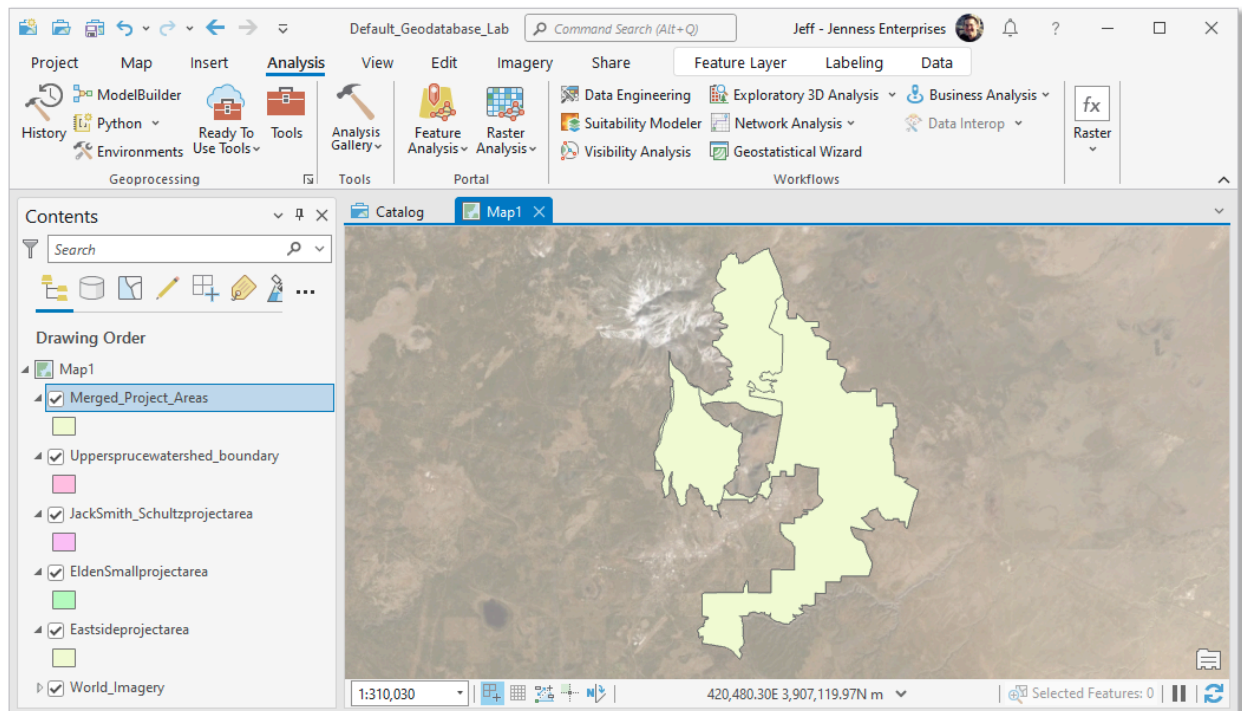
- i) *Input Feature Classes* = **Eastsideprojectarea**, **EldenSmallprojectarea**, **JackSmith_Schultzprojectarea** and **Uppersprucewatershed_boundary**

(1) **Tip:** You can select all four feature classes at once in the **Contents** pane and drag them all into the “Input Feature Classes” box.

- ii) *Output Feature Class:* Name your new feature class **Merged_Project_Areas**. Confirm that you are saving to your default geodatabase from Lab Exercise 1. **Important:** You will turn in your file geodatabase as part of the homework, so it is important that this feature class be included!



- c) Observe the *Field Map* section. This is the combined list of all attribute fields that exist in all four input datasets.
- d) Hit **Run** and the tool will combine all project areas into a single feature class.



- 7) Open the attribute table to see the attribute values.
- 8) Find the records for the feature class that only had three attribute fields.
- 9) This feature class did not have many of the attribute fields that the other feature classes did. **For example**, the other feature classes had attribute fields named "DATA_SOURC", "ACCURACY" and "Name", and they had attribute values in these fields.
- 10) **Question** (Not for homework): What attribute values did the **Merge** tool assign when the original features did not have the attribute fields? **For example**, what are the "DATA_SOURC", "ACCURACY" and "Name" values for those records that came from the feature class that did not have these attribute fields?

Exercise 3: Learn the Dissolve tool

Objectives

- Learn how to use the Dissolve tool.
- You will create a new dataset that will be used in the lab exercise 7.

Data used

- Feature Class **Stands_RockPrj_CF_2014.shp**, in folder `..\Class_GIS_Data\RockPrj_CF_2014\..`. These are the stands in the Rock Project Management Unit in the NAU Centennial Forest.
- File Geodatabase **Your_Name_Automation_Lab.gdb**, which you created in the first lab.

Homework Assignment:

You will create one new feature class named **Rock_Project_Unit** in the geodatabase you created in Exercise 1. **You will send me the geodatabase as part of your homework.**

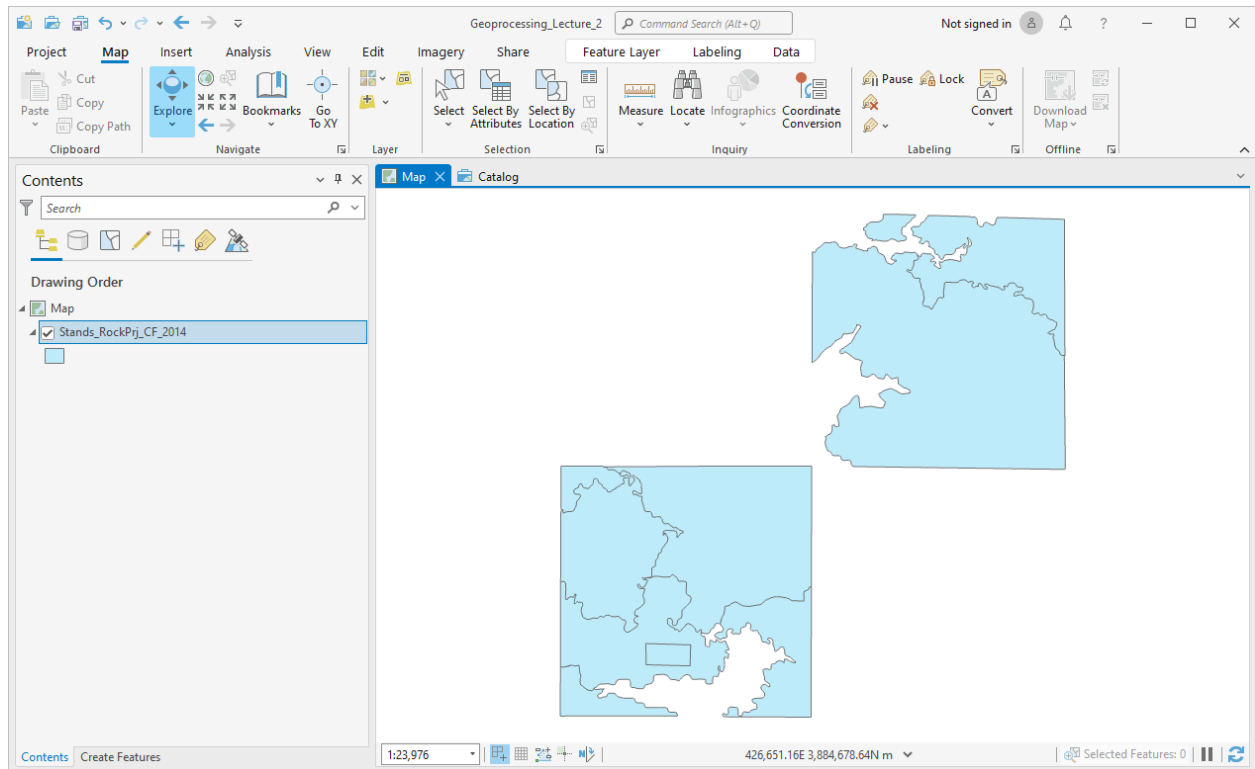
Exercise

This lab exercise will show you how to dissolve multiple features into a single feature. You will produce a new feature class containing a single multipart polygon representing the extent of the Rock Project Management Unit in the NAU Centennial Forest.

This exercise will dissolve all features into a single combined feature, but be aware that you can also use the Dissolve tool to combine features that all share particular attribute values. For example, if you had a feature class of US Counties, and if those features each had a “State Name” attribute, then you could dissolve all counties into their respective states. You would probably end up with a feature class containing 50 features, give or take a few depending on whether Hawaii, Alaska or the territories are included.

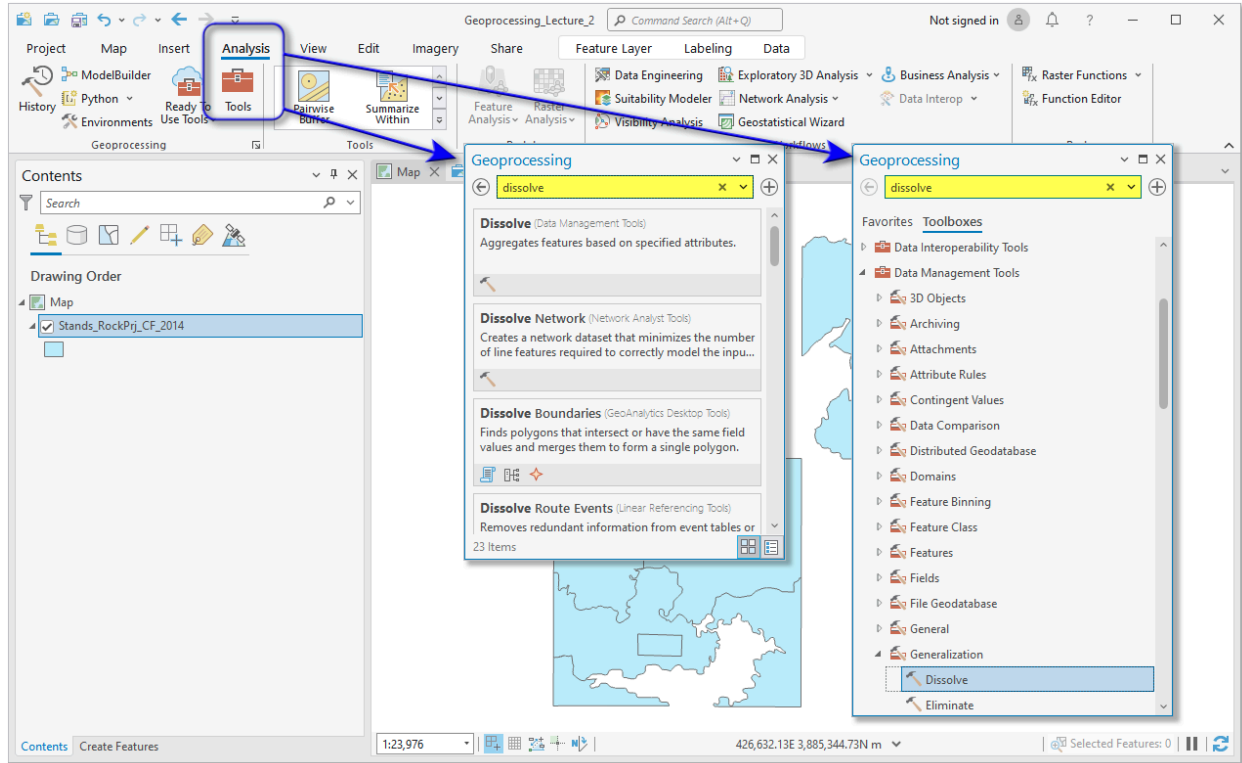
Prepare your ArcGIS Pro Project

- 1) If it is not still open, then open your ArcGIS Pro project from Lab Exercise 1. Confirm that your default geodatabase is **Your_Name_Automation_Lab.gdb**. If it is not, then follow the instructions in Lab Exercise 1 to set the default geodatabase correctly.
 - a) **Note:** **You will be sending me this geodatabase as part of your homework**, so it is important that you have it set as the default. Otherwise you may not correctly save the required feature class to the database.
- 2) Open a new map and load the shapefile **Stands_RockPrj_CF_2014** from the folder `..\Class_GIS_Data\RockPrj_CF_2014\..`.

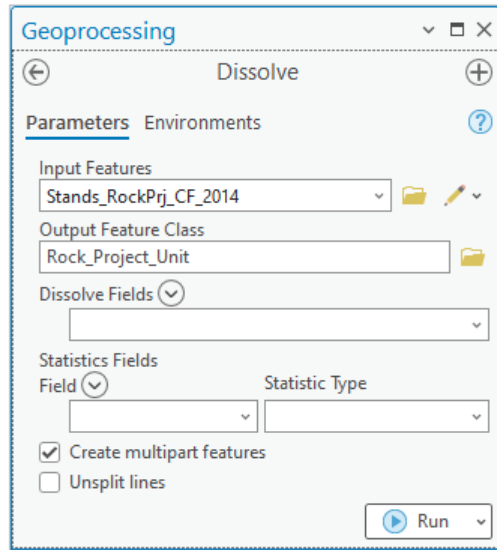


Open and run Dissolve tool

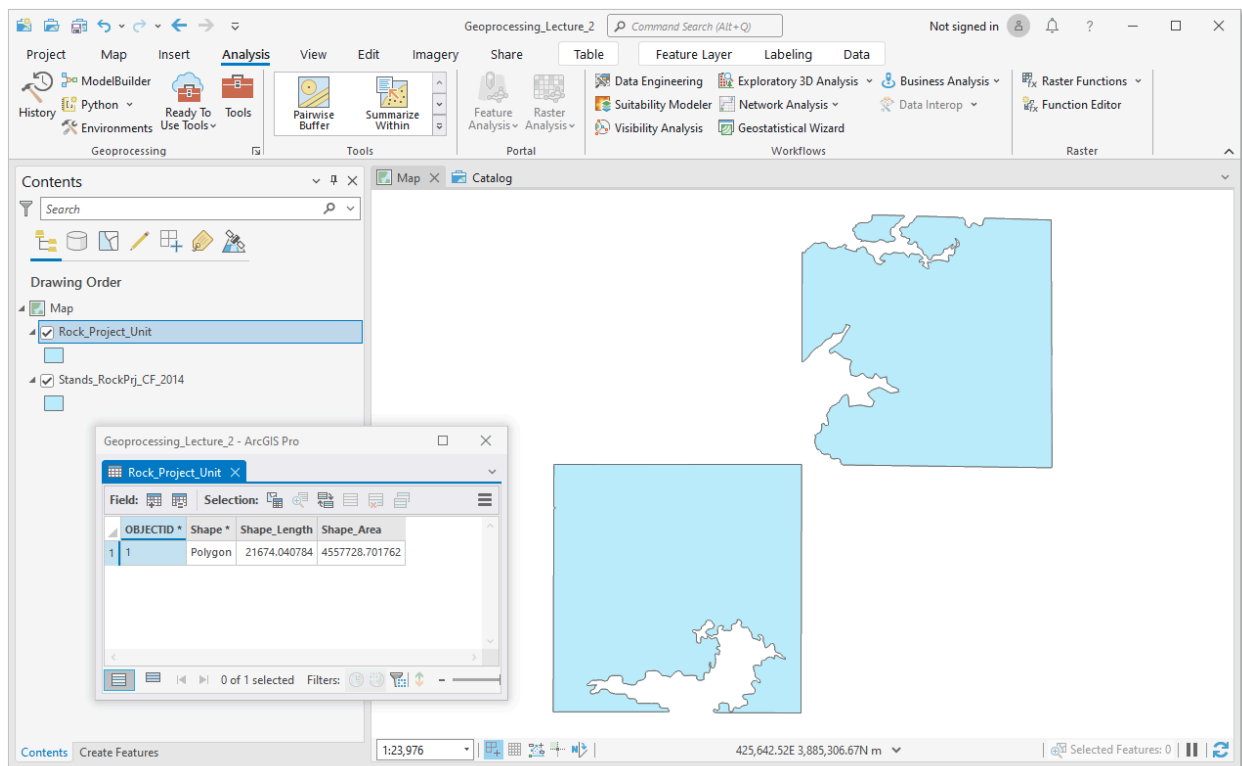
- 3) Open your Dissolve tool by clicking the **Analysis** tab, and **Tools**, and then choosing **Dissolve**:
 - a) You can find the **Dissolve** tool by either searching for it directly, or by open up the **Toolboxes**, then **Data Management Tools**, then **Generalization**, then **Dissolve**.



- b) In the **Dissolve** parameters window, set the following:
- i) *Input Features* = **Stands_RockPrj_CF_2014**.
 - ii) *Output Feature Class*: Name your new feature class **Rock_Project_Unit**. **Confirm that you are saving to your default geodatabase from Lab Exercise 1.** **Important:** You will turn in your file geodatabase as part of the homework, so it is important that this feature class is included!
 - iii) *Dissolve Fields*: Leave empty. We are dissolving all features into a single object, not dissolving separate classes of features.
 - iv) *Statistics Fields*: Leave empty
 - v) *Create multipart features* = **Checked**
 - vi) *Unsplit Lines* = **Unchecked**; this option only applies to dissolving line features anyway.



- c) Hit **Run** and the tool should go to work. Upon completion, you should have a new feature class containing a single multipart polygon representing the overall Rock Project Unit boundary.



Exercise 4: Learn the Buffer tool

Objectives

- Learn how to use the **Buffer** tool.
- Learn how the **Buffer** tool **does** honor selected features. If a particular feature or set of features are selected, then the **Buffer** tool will only consider those selected features in the analysis.
- You will create two new datasets that will be used in the next lab exercise.

Data used

- Polyline layer **Coconino_NF_MVUM_Roads** (this is the Coconino road status as of 2022) in workspace `..\Class_GIS_Data\Additional_Data`.
- File Geodatabase **Your_Name_Automation_Lab.gdb**, which you created in the first lab.

Homework Assignment:

You will add two new feature classes named **All_Vehicles_Yearlong** and **Highway_Legal_Yearlong** to the geodatabase you created in Exercise 1. **You will send me the geodatabase as part of your homework.**

Exercise

This lab exercise will show you how to create buffer polygons around existing features. This exercise will also demonstrate how selecting features first will cause the buffer tool to only buffer the selected features.

Specifically, you will create two new polygon datasets. One will contain 1-mile buffers around all roads classified as **Roads Open to All Vehicles, Yearlong** on the Coconino, and the other will contain 0.25-mile buffers around all roads classified as **Roads Open to Highway Legal Vehicles Only, Yearlong**. In a later lab exercise, you will use the Union tool to combine them into a single polygon, and later still you will clip that polygon to the Coconino boundary.

Note: ArcGIS Pro has **two versions** of the Buffer tool: The standard **Buffer** tool we are using in this exercise, and the **“Pairwise”** Buffer tool. And in fact ArcGIS Pro has a whole class of “Pairwise” tools that replicate the behavior of the traditional Overlay tools.

Pairwise, in this case, means the tool uses **parallel processing** (i.e. using multiple processes running simultaneously in the computer). ArcGIS Pro’s terminology here has nothing to do with the standard statistical definition of “Pairwise” (i.e. analyzing statistical patterns in paired data), and in fact I struggle to appreciate why Esri would use this term to describe tools whose internal algorithms are coded to take advantage of multiple processors.

Because these “Pairwise” tools take advantage of multiple processors, they do better with large datasets. And in fact they are written specifically to handle large datasets much faster than the traditional tools. However, at least with respect to the **Buffer** tool, the “Pairwise” version creates a less smooth buffer than the standard version. For small datasets such as our lab exercise data, the processing time is reasonably fast for both versions of the **Buffer** tool so we will use the traditional version.

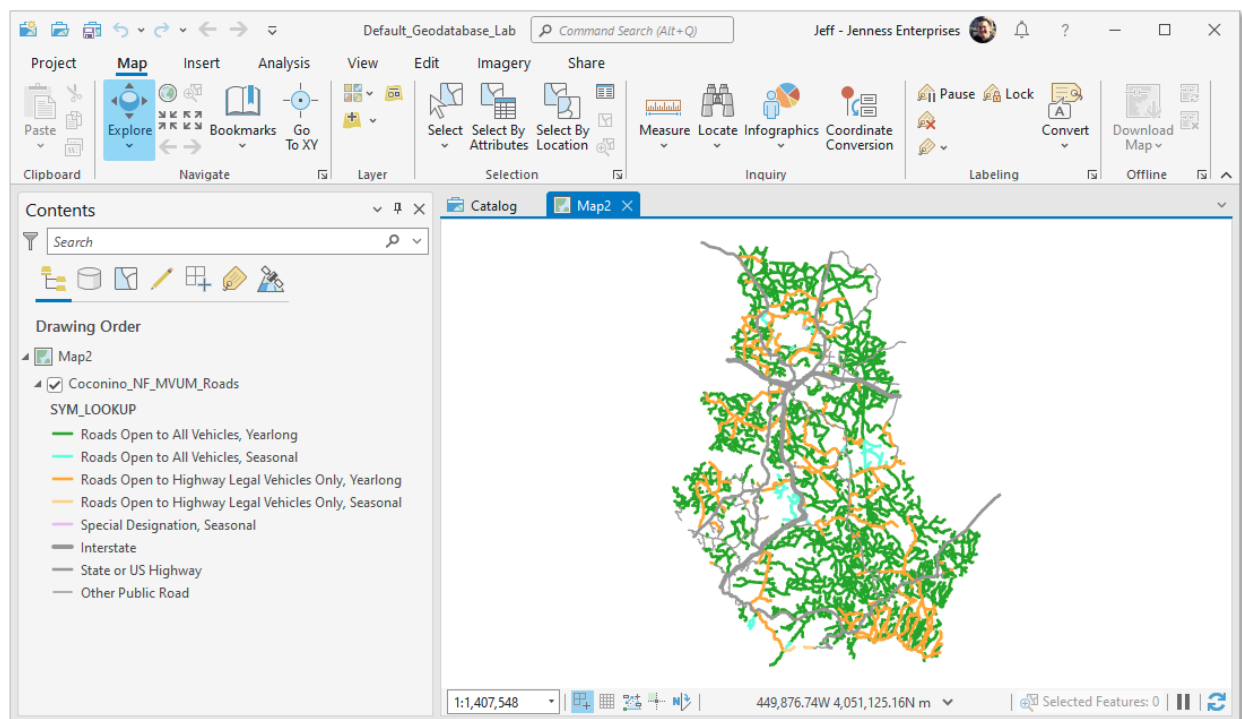
There are other differences between the two versions of the **Buffer** tool. The traditional version offers choices about how to buffer the endpoints of lines, and gives the option to only buffer one side of a polyline or polygon. More importantly, the “Pairwise” version of the buffer tool can, under certain

conditions, produce output with true curves in the form of circular arcs, or actual true circles if buffered around points. These circles and circular arcs are defined only by the center point plus a radius (i.e. the buffer is **not** a series of vertices around the edge of the circle or arc). This may sound like a good idea in theory, but these buffers behave **badly** if you change the coordinate system of your map. A true circle, defined only by a centerpoint and radius, will cover different regions of the landscape if you change the map coordinate system. In general I don't totally trust how the "Pairwise" version of the buffer tool works, so I avoid it.

However, if you would like to try the "Pairwise" version in this lab, please feel free to do so! Every tool you try gives you more experience, and will make you better at GIS.

Prepare your ArcGIS Pro Project

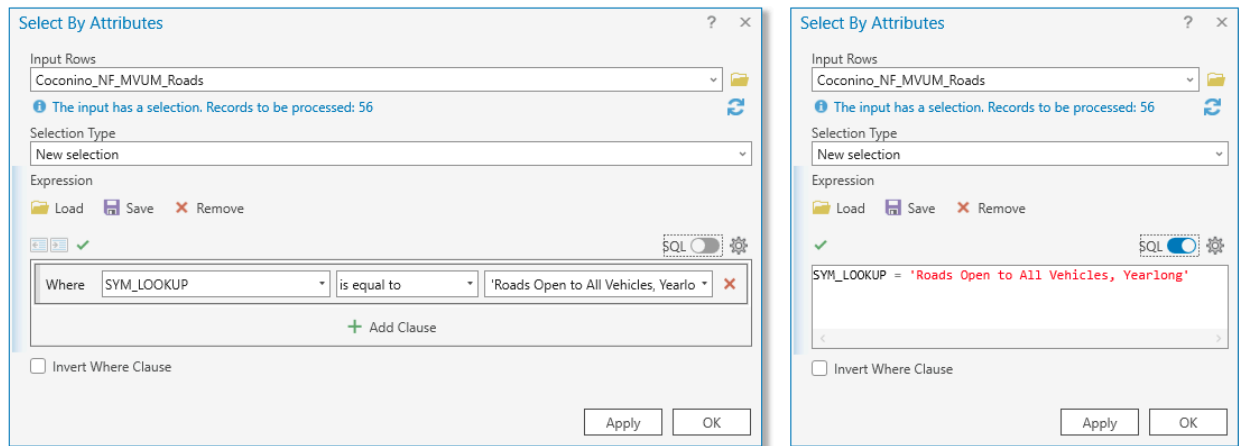
- 1) If it is not still open, then open your ArcGIS Pro Project from Lab Exercise 1. Confirm that your default geodatabase is **Your_Name_Automation_Lab.gdb**. If it is not, then follow the instructions in Lab Exercise 1 to set the default geodatabase correctly.
 - a) **Note:** You will be sending me this geodatabase as part of your homework, so it is important that you have it set as the default. Otherwise you may not correctly save the required feature classes to the database.
- 2) Open a new map and load the layer file **Coconino_NF_MVUM_Road.lyrx** from the folder **..\Class_GIS_Data\Additional_Data\.**
- 3) If you like, feel free to add a background dataset like **USA_Topo_Maps** or **Imagery**, just to give a general reference to the project. These background datasets will not be used in the lab, though.



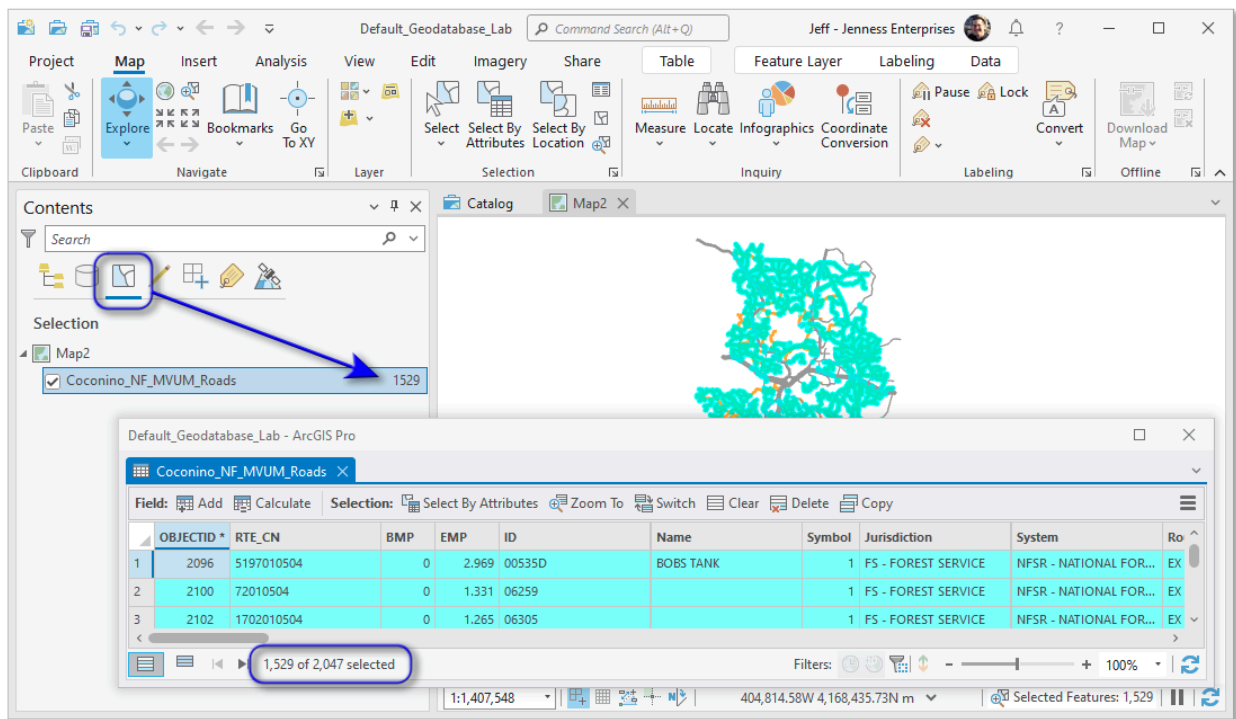
Select First Set of Roads

- 4) Select all features from **Coconino_NF_MVUM_Road** that are classified as **Roads Open to All Vehicles, Yearlong**.

a) Open the “Select by Attributes” window as you have done before, and fill out the tool as follows:



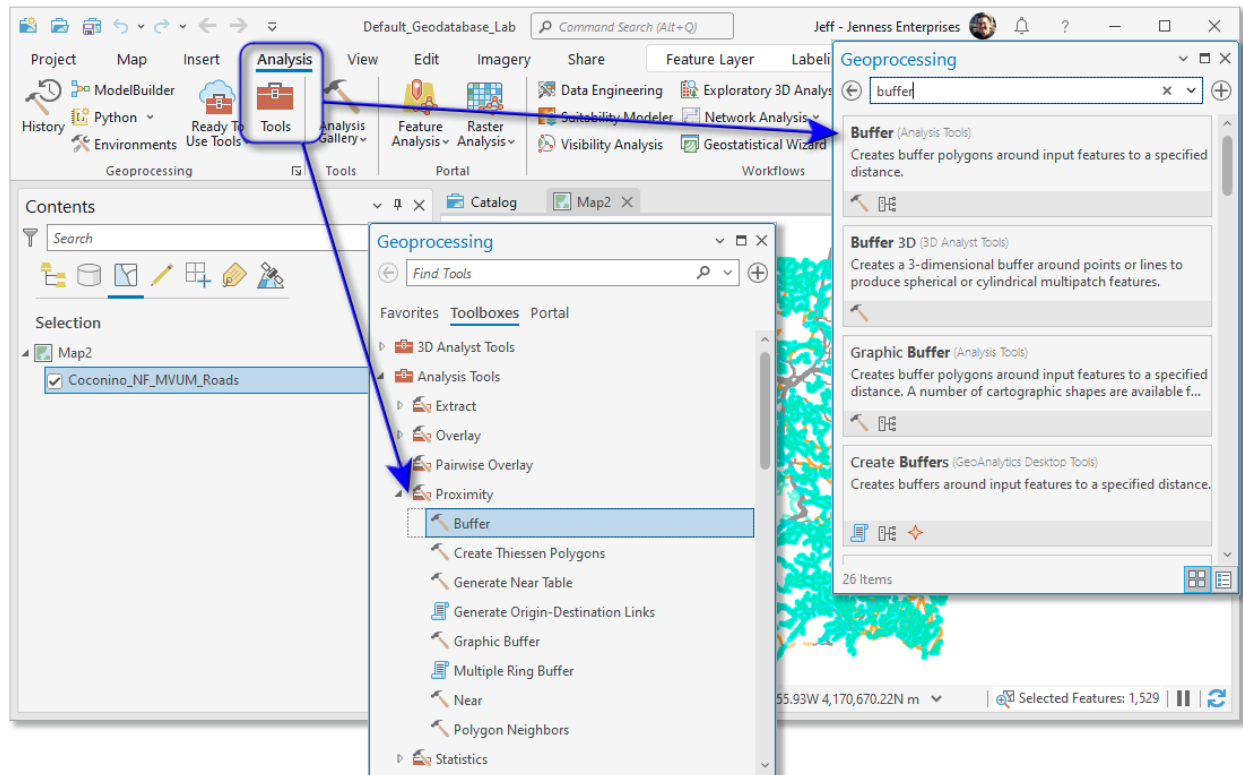
b) Hit OK and you should get 1,529 roads selected.



Open and run Buffer tool

5) Open your *Buffer* tool by clicking the **Analysis** tab, and **Tools**, and then choosing **Buffer**:

a) You can find the **Buffer** tool by either searching for it directly, or by open up the **Toolboxes**, then **Analysis Tools**, then **Proximity**, then **Buffer**.



b) In the **Buffer** parameters window, set the following:

i) **Input Features** = **Coconino_NF_MVUM_Road**

(1) **Important:** Make sure you select the input features by either clicking the down-arrow in

the “Input Features” box , or else by dragging the layer from your map into that box. **Do not use the  button!** The reason for not using the  button is important to understand:

Remember that we want to create a buffer around only the **selected** features. This selection only exists in the actual **layer** in your map, not in the dataset on the hard drive. The selection is part of the **layer definition**.

The **Buffer** tool lets us load data to buffer by either using a **Layer** from your map, or a **Feature Class** from the hard drive. The  button is what we use to load a feature class from the hard drive. But if we do this, then the **Buffer** tool will not know about our selected features. It will buffer every road.

Therefore, in order to restrict the buffers to only our selected features, we have to

specify the **Layer**. And we do that with either the Down-arrow  or by simply dragging the layer over from your map into this tool.

ii) **Output Feature Class:** Name your new feature class **All_Vehicles_Yearlong**. Confirm that you are saving to your default geodatabase from Lab Exercise 1. **Important:** You will turn in

your file geodatabase as part of the homework, so it is important that this feature class is included!

iii) *Distance* = **1 US Survey Mile**

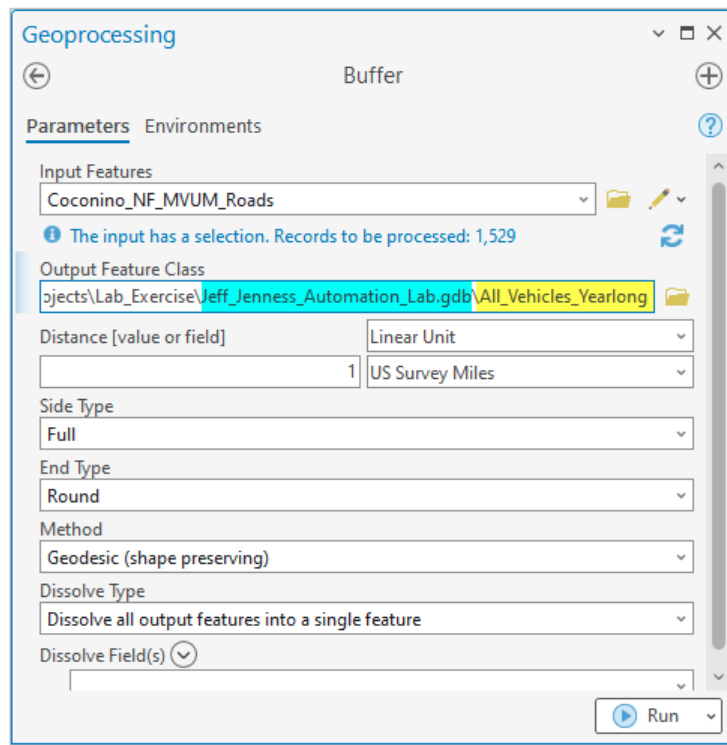
iv) *Side Type* = **Full**

v) *End Type* = **Round**

vi) *Method* = **Geodesic (shape preserving)**

vii) *Dissolve Type* = **Dissolve all output features into a single feature**

- (1) **This is an important parameter, too.** By setting the **Dissolve Type** value to be ***Dissolve all output features into a single feature***, we force all buffer polygons to be merged into a single object. The lab exercise would still work if you chose the type **None**, but in that case you would get 1,529 overlapping buffers instead of a single combined buffer.



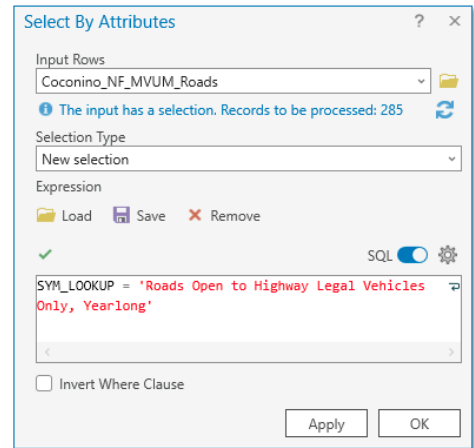
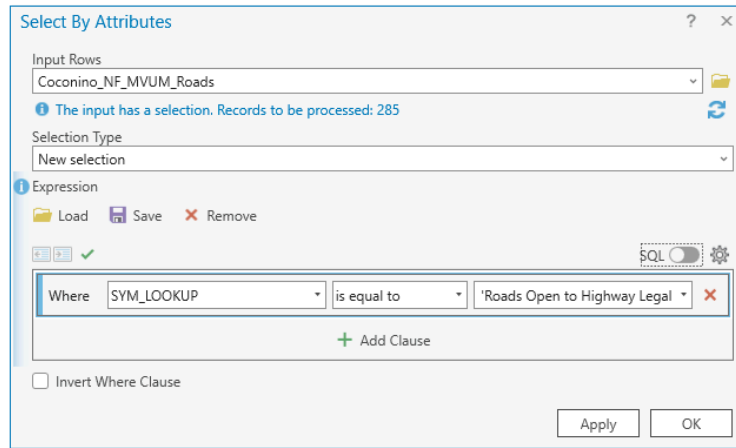
c) Click **Run** and create your 1-mile buffers.

d) Check the map to confirm that these buffers are only around the ***Roads Open to All Vehicles, Yearlong*** features.

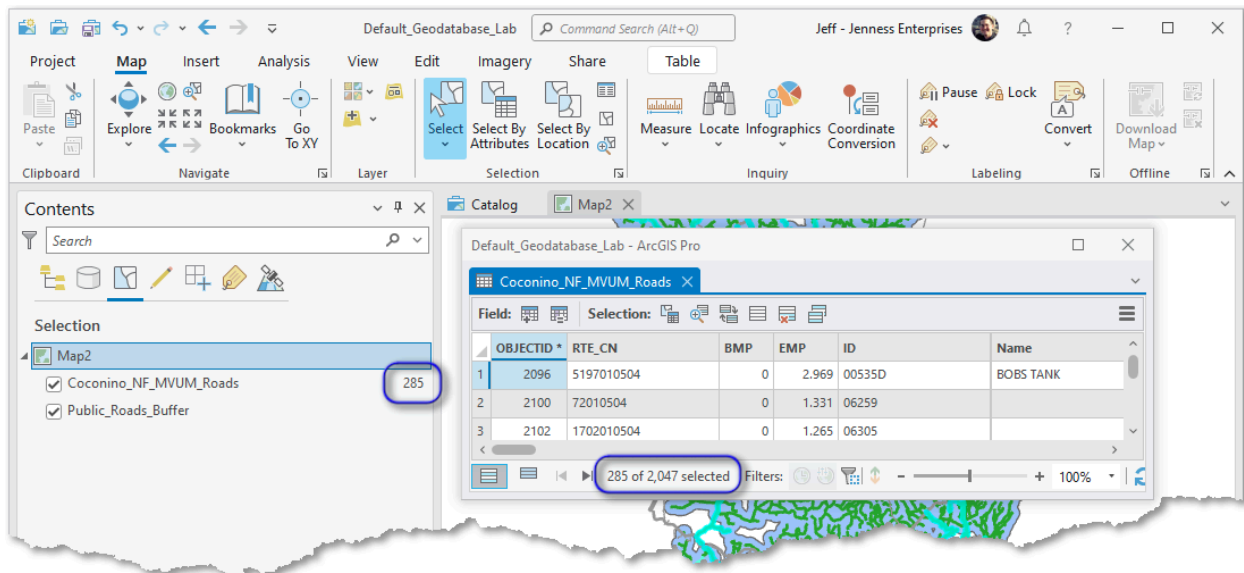
Select Second Set of Roads

6) Select all features from **Coconino_NF_MVUM_Road** that are classified as ***Roads Open to Highway Legal Vehicles Only, Yearlong***.

a) Open the "Select by Attributes" window as you have done before, and fill out the tool as follows:



b) Hit OK and you should get 285 roads selected.



Open and run Buffer tool

7) Open the buffer tool as you did before and set the following parameters:

i) **Input Features = Coconino_NF_MVUM_Road**

(1) **Important:** Again, make sure you have selected the **Layer**. Do **not** use the  button to load the data.

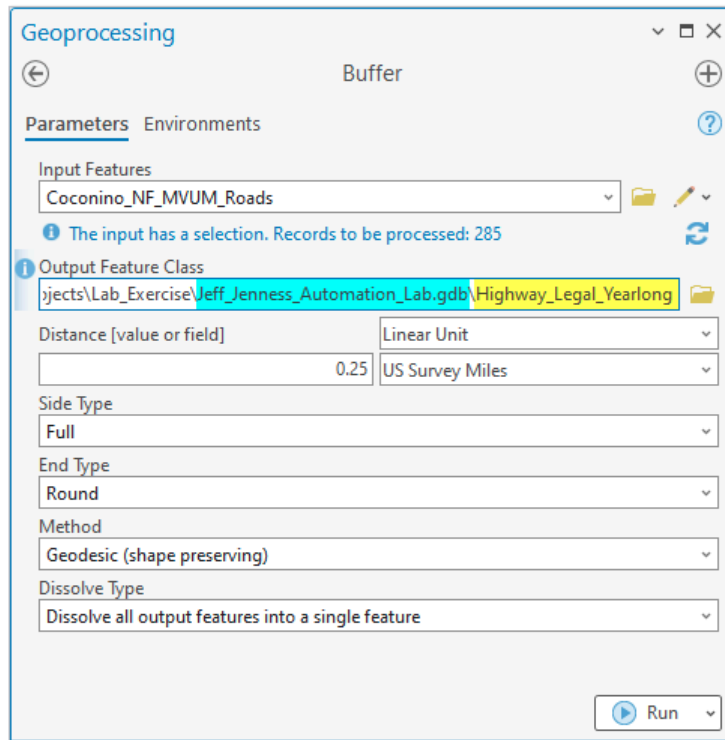
ii) **Output Feature Class:** Name your new feature class **Highway_Legal_Yearlong**. Confirm that you are saving to your default geodatabase from Lab Exercise 1. **Important:** You will turn in your file geodatabase as part of the homework, so it is important that this feature class is included!

iii) **Distance = 0.25 US Survey Miles**

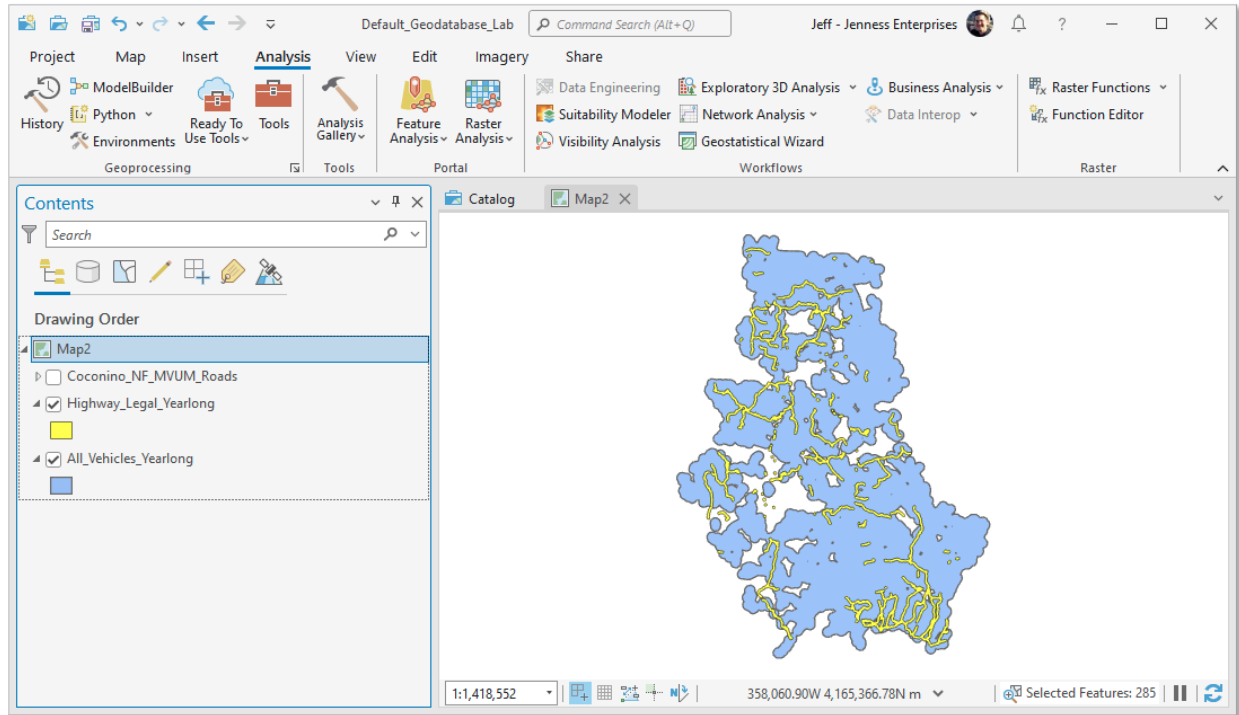
iv) **Side Type = Full**

v) **End Type = Round**

- vi) *Method* = **Geodesic (shape preserving)**
- vii) *Dissolve Type* = **Dissolve all output features into a single feature**



- 8) Click OK and create your 0.25-mile buffers.
- 9) Check the map to confirm that these buffers are only around the **Roads Open to Highway Legal Vehicles Only, Yearlong**. If you turn off your actual road layer, your map should look something like this:



10) **Don't close ArcGIS Pro yet!** You will use these two feature classes in your next lab exercise.

Exercise 5: Learn the Union tool

Objectives

- Learn how to use the Union tool.
- Learn two ways to add attribute values to a table: 1) The **Calculate Field** command, and 2) typing directly into the table.
- You will create one new dataset that will be used in the next lab exercise.

Data used

- Feature class **All_Vehicles_Yearlong**, in the geodatabase you created in Lab Exercise 1.
 - Note: You just created this feature class in Lab Exercise 4.
- Feature class **Highway_Legal_Yearlong**, in the geodatabase you created in Lab Exercise 1.
 - **Note:** You just created this feature class in Lab Exercise 4.
- File Geodatabase **Your_Name_Automation_Lab.gdb**, which you created in the first lab.

Homework Assignment:

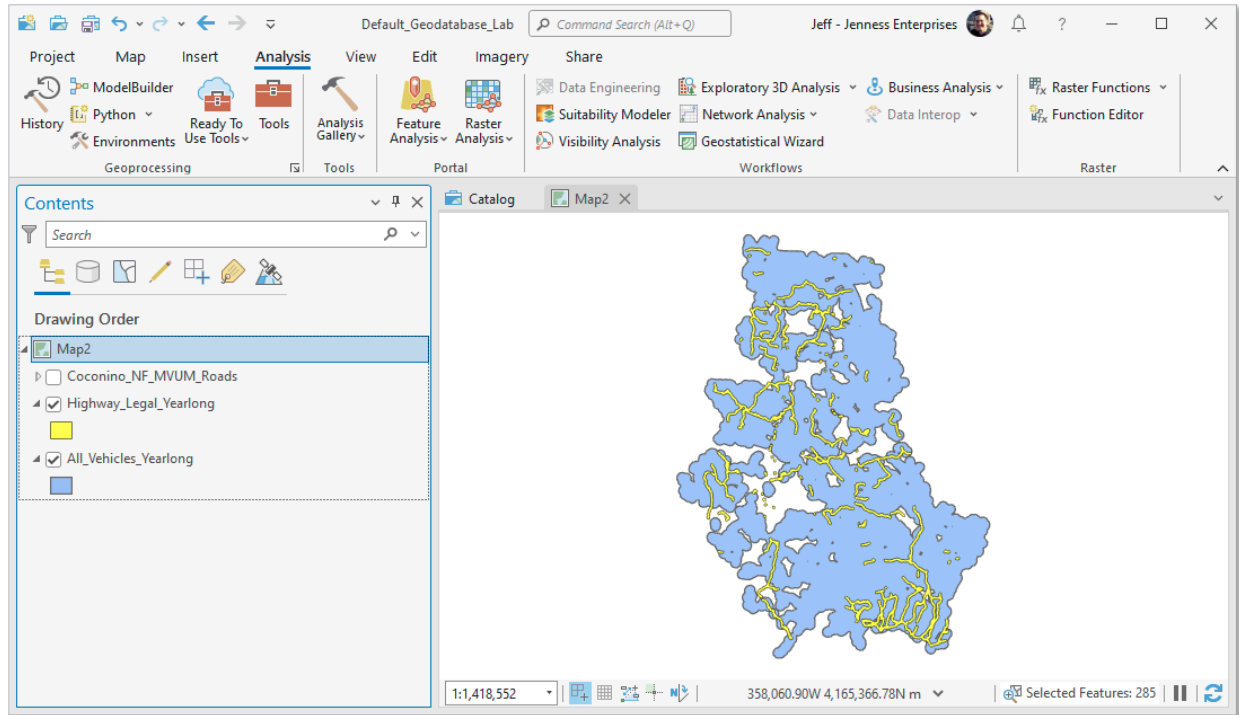
You will add one new feature class named **Roads_Union** to the geodatabase you created in Exercise 1. You will also add an attribute field named **All_Vehicles_Text** to your feature class **All_Vehicles_Yearlong**, and add an attribute field named **Highway_Legal_Text** to your feature class **Highway_Legal_Yearlong**. You will send me the geodatabase as part of your homework.

Exercise

This lab exercise will show you how to **Union** two feature classes into a new feature class. This exercise will also demonstrate how the Union transfers attributes from the source feature classes to the Union feature class.

Prepare your ArcGIS Pro Project

- 1) If it is not still open, then open your ArcGIS Pro Project from Lab Exercise 1. Confirm that your default geodatabase is **Your_Name_Automation_Lab.gdb**. If it is not, then follow the instructions in Lab Exercise 1 to set the default geodatabase correctly.
 - a) **Note:** You will be sending me this geodatabase as part of your homework, so it is important that you have it set as the default. Otherwise you may not correctly save the required feature classes to the database.
- 2) If they are not already in your map, load **All_Vehicles_Yearlong** and **Highway_Legal_Yearlong** from your file geodatabase **Your_Name_Automation_Lab.gdb**, which you created in the first lab.

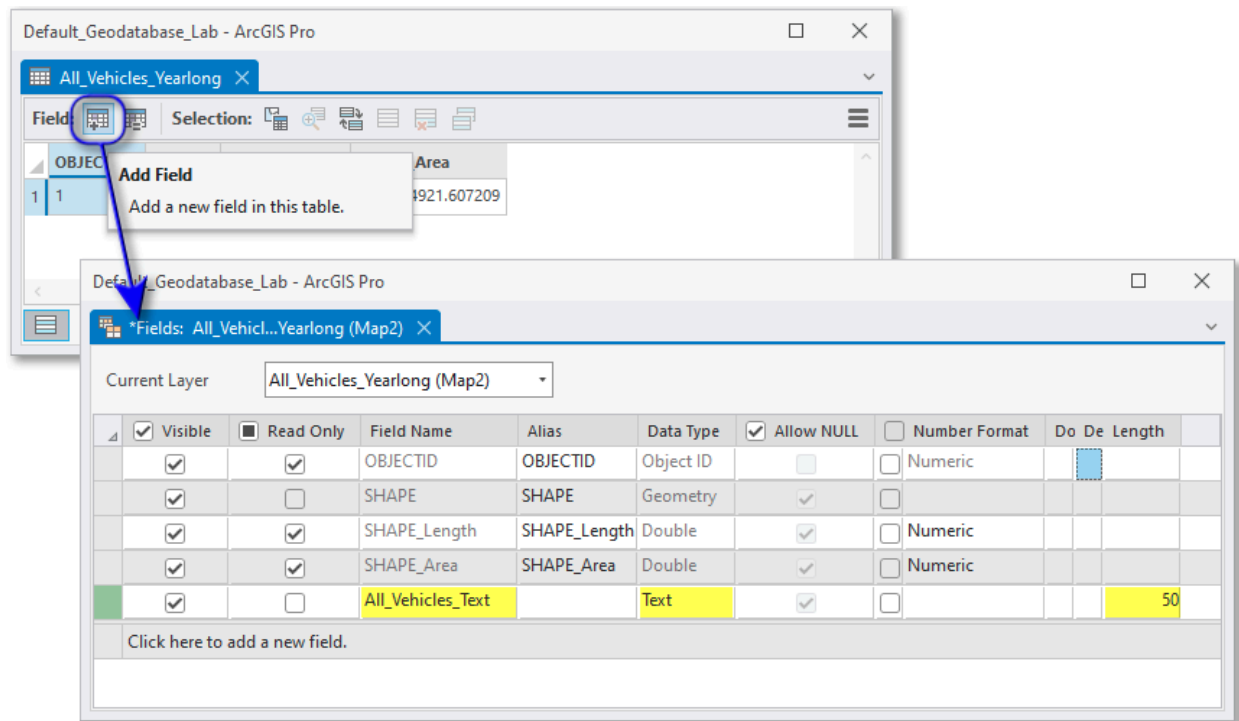


Add an Attribute Field to Each Feature Class

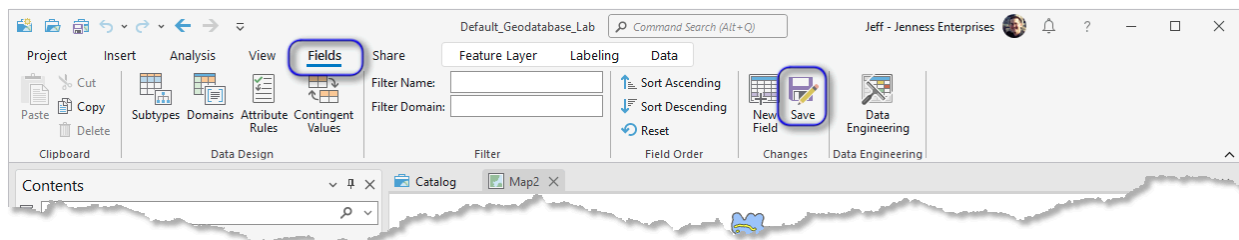
The **Union** tool will combine feature class in a particular way, producing a new dataset containing all the attribute fields of each feature class. We will add a single new attribute field to each feature class, in order to observe how the **Union** tool brings in these attributes to the final output feature class.

Note: The next few pages will show you how to add an attribute field to your feature class, and then two methods to add data to that field. The first method will use the **Calculate Field** tool, which is a good way to write values to lots of features at the same time. The second method shows you how to write text directly into the table, which is an easier way to write values to a single feature.

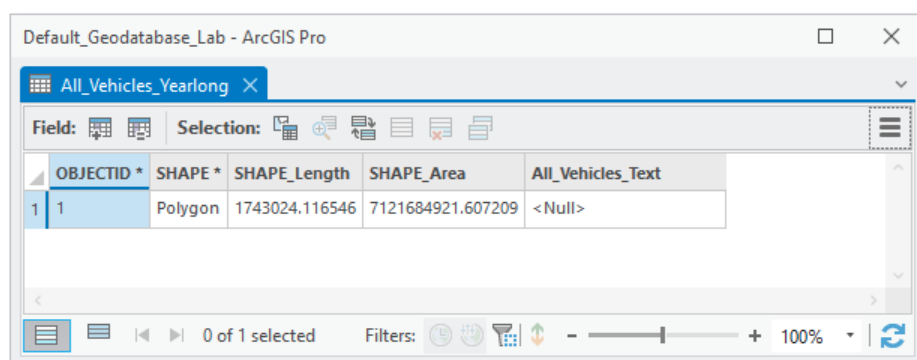
- 3) Open your attribute table for **All_Vehicles_Yearlong**.
 - a) Click the “Add Field” button in the top left corner of the table to open up the “Fields” view for **All_Vehicles_Yearlong**, and create a new Text field named *All_Vehicles_Text*, with a length of 50:



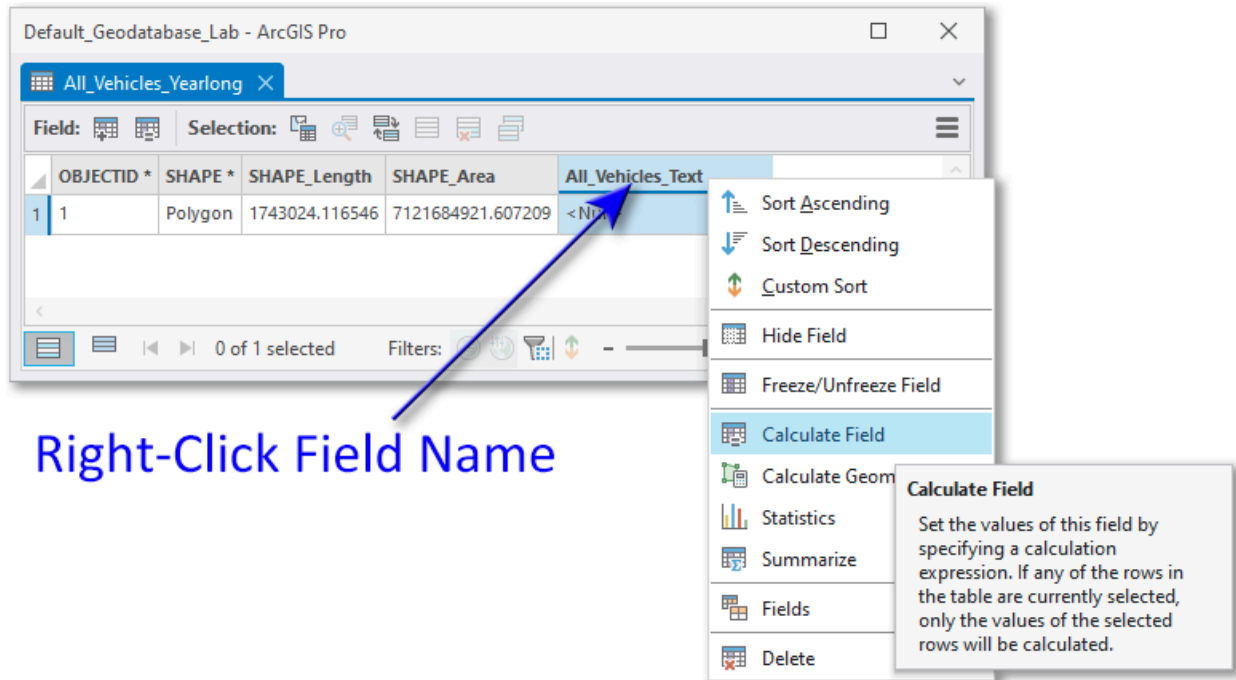
- b) At the top of ArcGIS Pro, make sure the **Fields** tab is selected and click the Save command to save your edits:



- c) Close your **Fields** view to return to the attribute table.



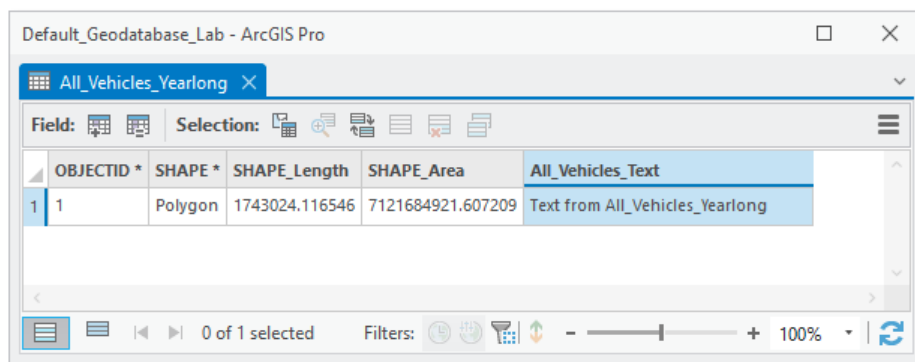
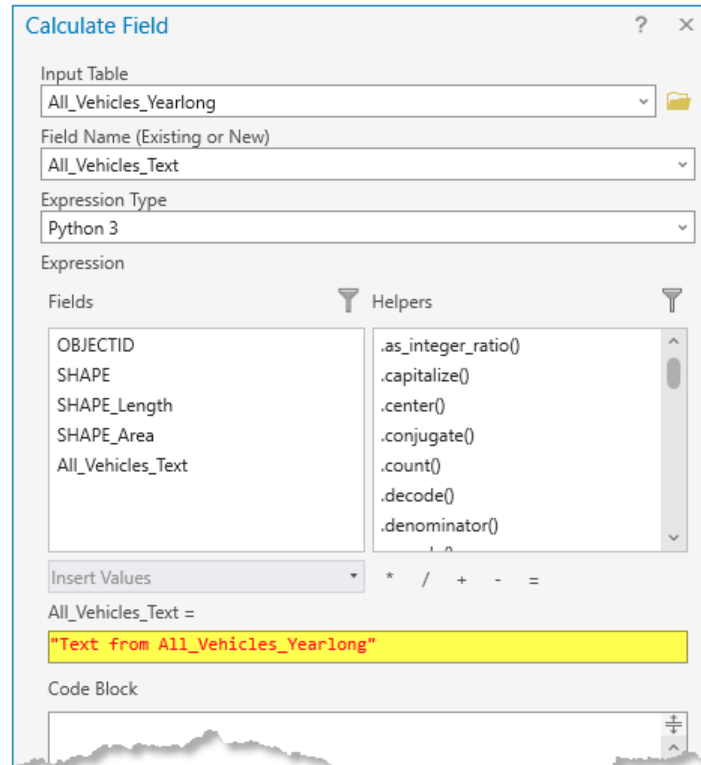
- d) Right-click on your new **All_Vehicles_Text** field and choose *Calculate Field*.



e) There are a lot of code tools in the *Calculate Field* window, but we don't need them for this operation. Simply type in the following text in the textbox beneath **All_Vehicles_Text** =.

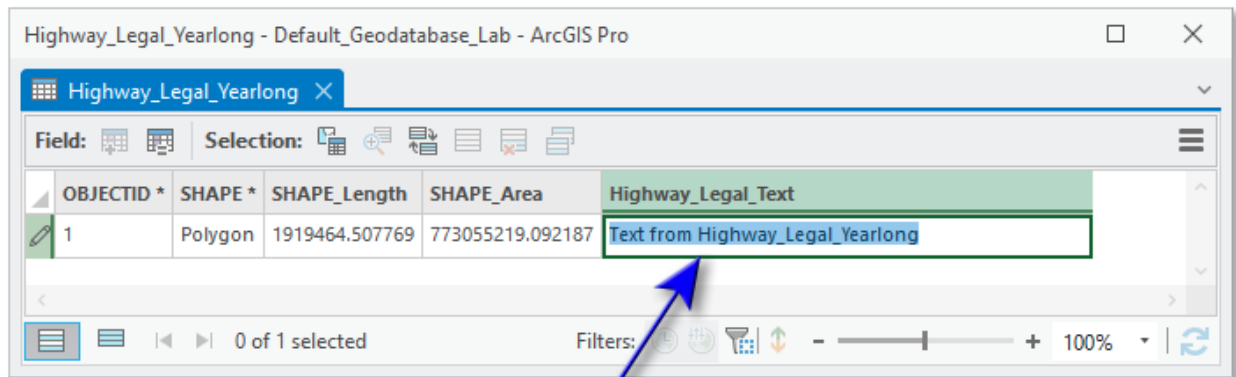
i) **Note:** The text you type in should be enclosed in double-quote characters (").

"Text from All_Vehicles_Yearlong"



- ii) **Note:** Using this *Calculate Field* function is very good for writing values to large numbers of rows at once. If you only need to write a value to a single row, it's easier to just type it in the table cell directly, which we will do shortly.
- f) Repeat the process above to add a new field to your feature class **Highway_Legal_Yearlong**, except this time name the new field "Highway_Legal_Text".
- g) This time, just double-click in the box below the field name "Highway_Legal_Text", and type in the following text (and note that it does **not** have to be contained in double-quotes this time):

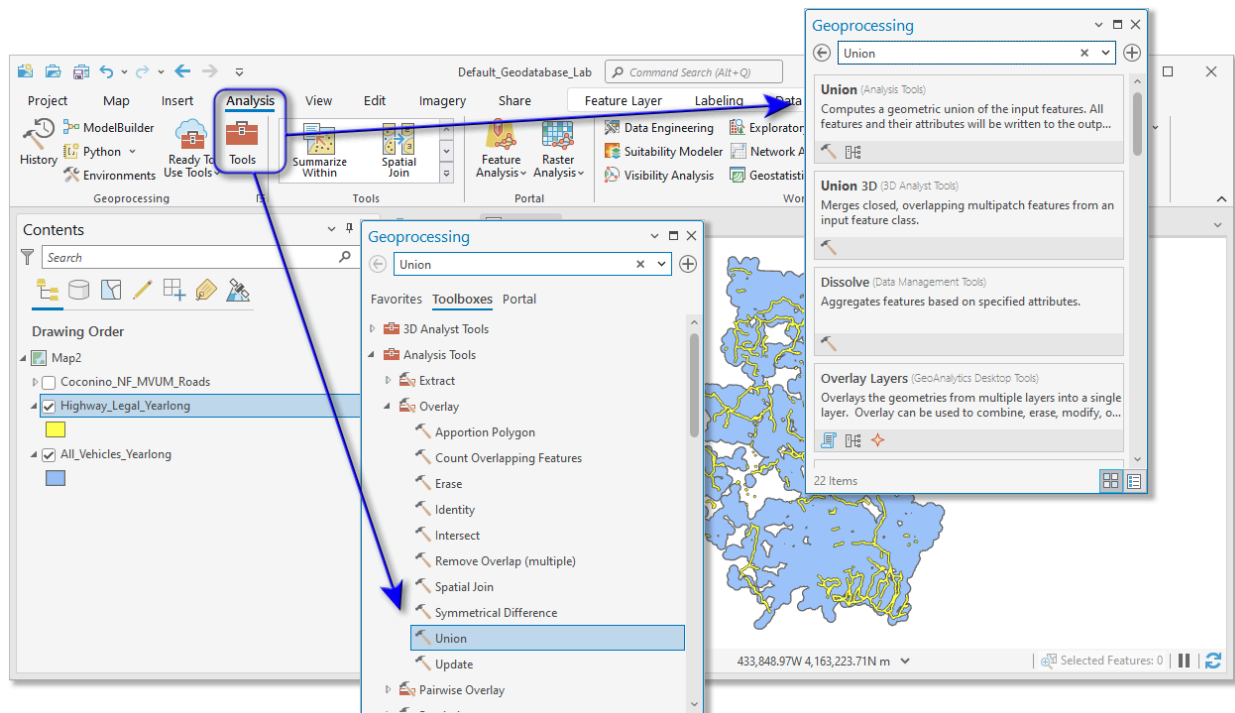
Text from Highway_Legal_Yearlong



Double-Click in here and start typing

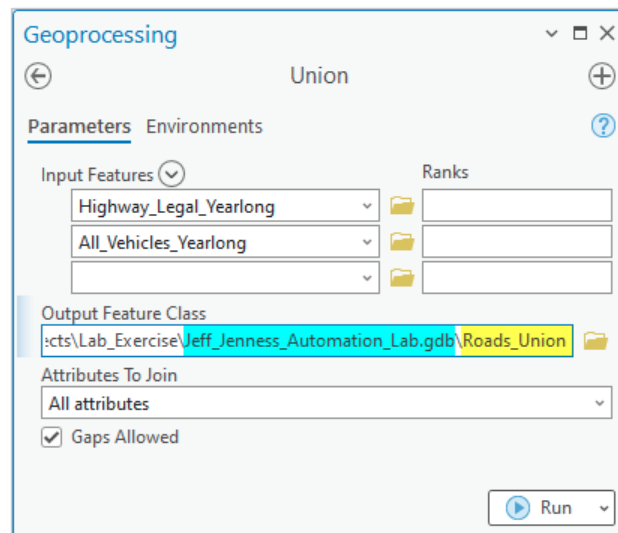
Open and run Union tool

- 4) Open your **Union** tool by clicking the **Analysis** tab, and **Tools**, and then choosing **Union**:
 - a) You can find the **Union** tool by either searching for it directly, or by open up the **Toolboxes**, then **Analysis Tools**, then **Overlay**, then **Union**.

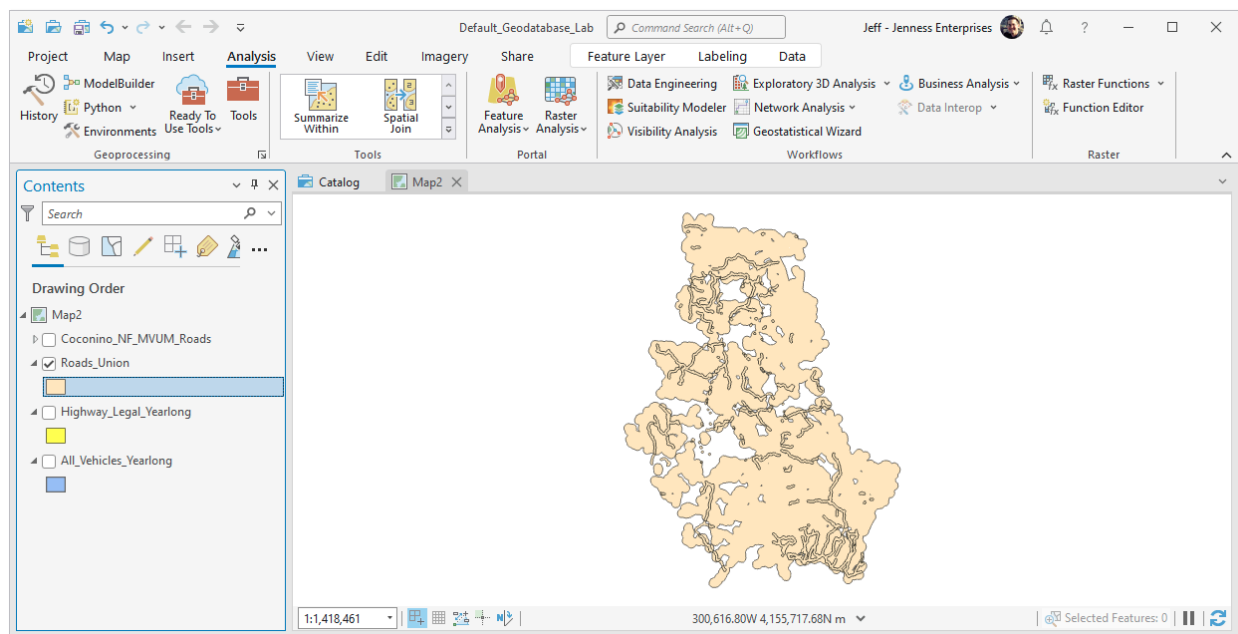


- 5) In the **Union** parameters window, set the following:
 - a) **Input Features** = **All_Vehicles_Yearlong** and **Highway_Legal_Yearlong**
 - i) **Note:** You can select both of these layers at once in the **Contents** pane and drag them into the "Input Features" box.

- ii) Don't worry about Input Feature Ranks. This just affects how feature geometries which are extremely close to one another get snapped to a common set of vertices. This helps prevent "sliver" polygons, which are very tiny artifacts that sometimes get created when edges don't align perfectly.
- b) *Output Feature Class*: Name your new feature class **Roads_Union**. Confirm that you are saving to your default geodatabase from Lab Exercise 1. **Important: You will turn in your file geodatabase as part of the homework**, so it is important that this feature class is included!
- c) *Attributes to Join* = **All attributes**
- d) *Gaps Allowed* = **True**



- 6) You should now have a new polygon feature class with separate polygons for all regions covered by the feature classes **All_Vehicles_Yearlong** and **Highway_Legal_Yearlong**.



- 7) Open the attribute table and observe all the attribute fields. Notice how the **Union** tool brings in all attribute fields and attribute values from both datasets.

	OBJECTID *	SHAPE *	FID_Highway_Legal_Yearlong	Highway_Legal_Text	FID_All_Vehicles_Yearlong	All_Vehicles_Text	SHAPE_Length
1	1	Polygon	1	Text from Highway_Hi...	-1		335335.0146
2	2	Polygon	-1		1	Text from All_Vehicles_...	3327153.6096
3	3	Polygon	1	Text from Highway_Hi...	1	Text from All_Vehicles_...	1827901.5305

- 8) Especially notice the attribute fields **FID_All_Vehicles_Yearlong** and **FID_Highway_Legal_Yearlong**. These contain the **FID (Feature ID)** values of the original data. These values come from the original attribute fields **ObjectID** from the feature classes **All_Vehicles_Yearlong** and **Highway_Legal_Yearlong**.
- 9) **Question** (Not for homework): What do you suppose the value -1 means in the attribute fields **FID_All_Vehicles_Yearlong** and **FID_Highway_Legal_Yearlong**?
- 10) **Don't close your map yet!** You will use this **Union** feature class in your next lab exercise.

11)

Exercise 6: Learn the Clip tool

Objectives

- Learn how to use the Clip tool, to trim vector datasets to some area of interest.
- Learn how the Clip tool **does** honor selected features. If a particular feature or set of features are selected, then the **Clip** tool will only consider those selected features in the analysis.

Data used

- Feature Class **Roads_Union**, in the geodatabase you created in Lab Exercise 1.
 - **Note:** You just created this feature class in Lab Exercise 5.
- Feature Class **AGFDGameMgmtUnits**, located in folder `..\Class_GIS_Data\Selecting_and_Creating_Data\..`
- File Geodatabase **Your_Name_Automation_Lab.gdb**, which you created in the first lab.

Homework Assignment:

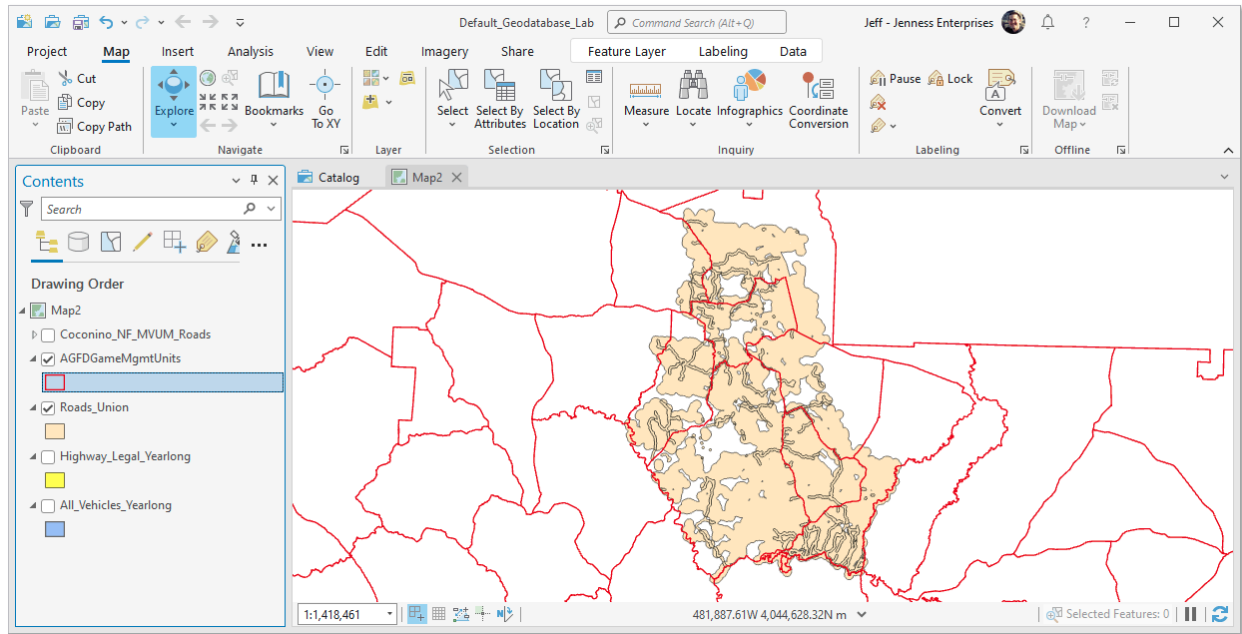
You will add one feature class named **Roads_in_6A** to the geodatabase you created in Exercise 1. **You will send me the geodatabase as part of your homework.**

Exercise

This lab exercise will show you how to trim an existing vector dataset to a polygon. Specifically, you will trim the feature class **Roads_Union**, which you created in Lab Exercise 5, to the boundary of Arizona Game and Fish Game Management Unit 6A, which is located south of Flagstaff.

Prepare your ArcGIS Pro Project

- 1) If it is not still open, then open your ArcGIS Pro Project from Lab Exercise 1. Confirm that your default geodatabase is **Your_Name_Automation_Lab.gdb**. If it is not, then follow the instructions in Lab Exercise 1 to set the default geodatabase correctly.
 - a) **Note:** **You will be sending me this geodatabase as part of your homework**, so it is important that you have it set as the default. Otherwise you may not correctly save the required feature class to the database.
- 2) If it is not already present, open a map and load the feature class **Roads_Union** from your file geodatabase **Your_Name_Automation_Lab.gdb**.
- 3) Load the Feature Class **AGFDGameMgmtUnits**, located in folder `..\Class_GIS_Data\Selecting_and_Creating_Data\..`



Note: I have changed the symbology of **AZGFDGameMgmtUnits** above so we can see the **Roads_Union** layer underneath...

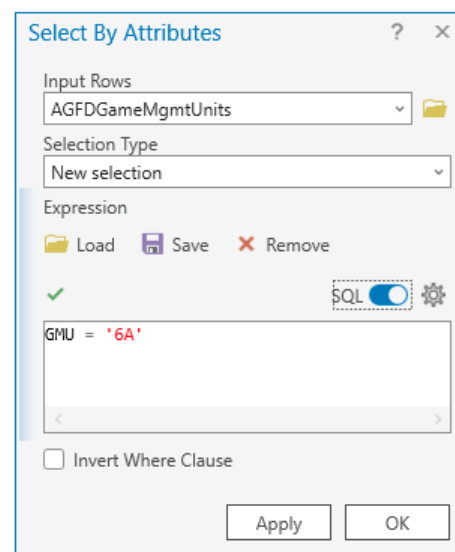
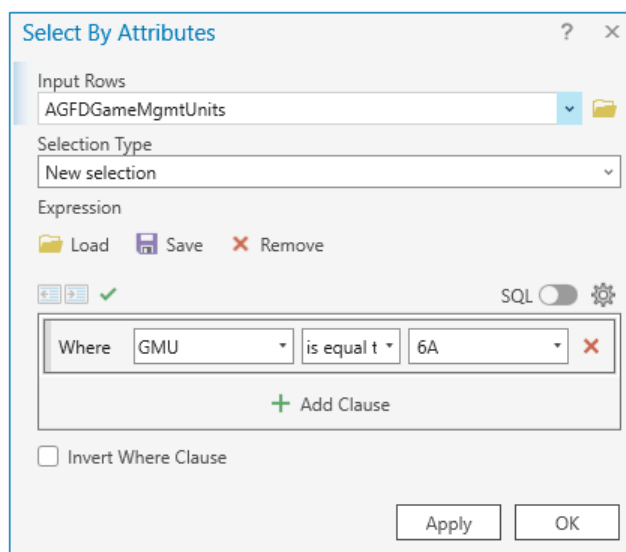
Select Game Management Unit Polygon

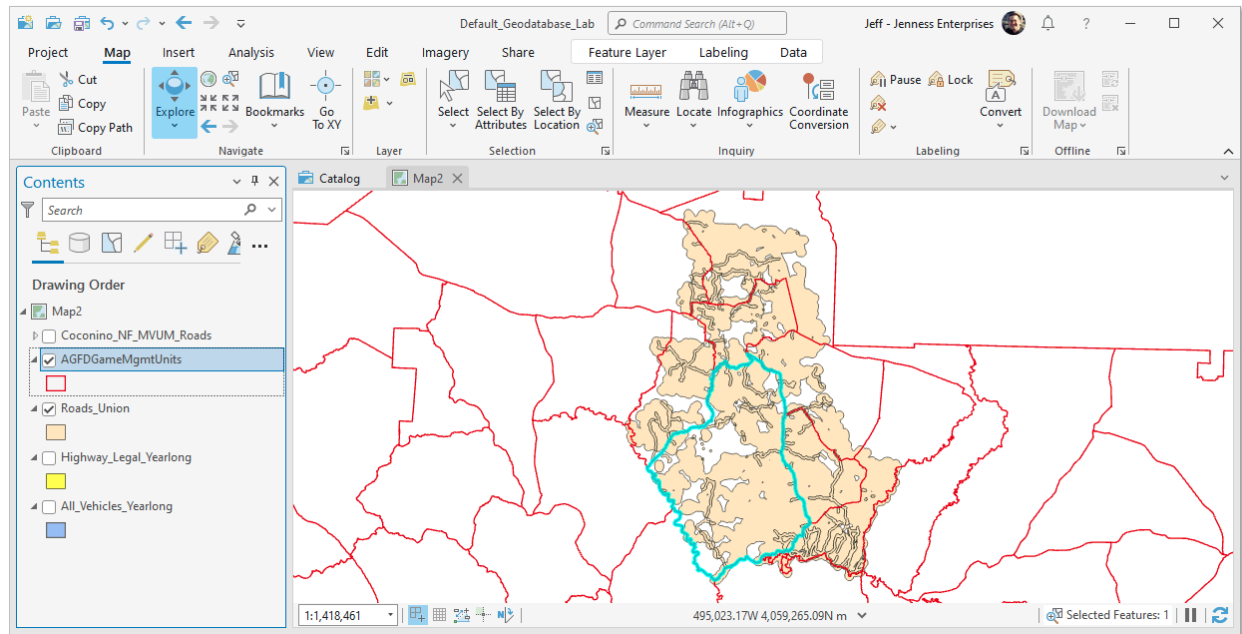
- 4) First select Arizona Game and Fish Management Unit 6A (see location above). If you know where it is (which you might if you hunt this area) the simplest way to do this would be to just use the



“Select” command **Select** to click on the polygon. Alternatively, you may use the *Select by Attributes* tool to select this feature, using the query string:

"GMU" = '6A'

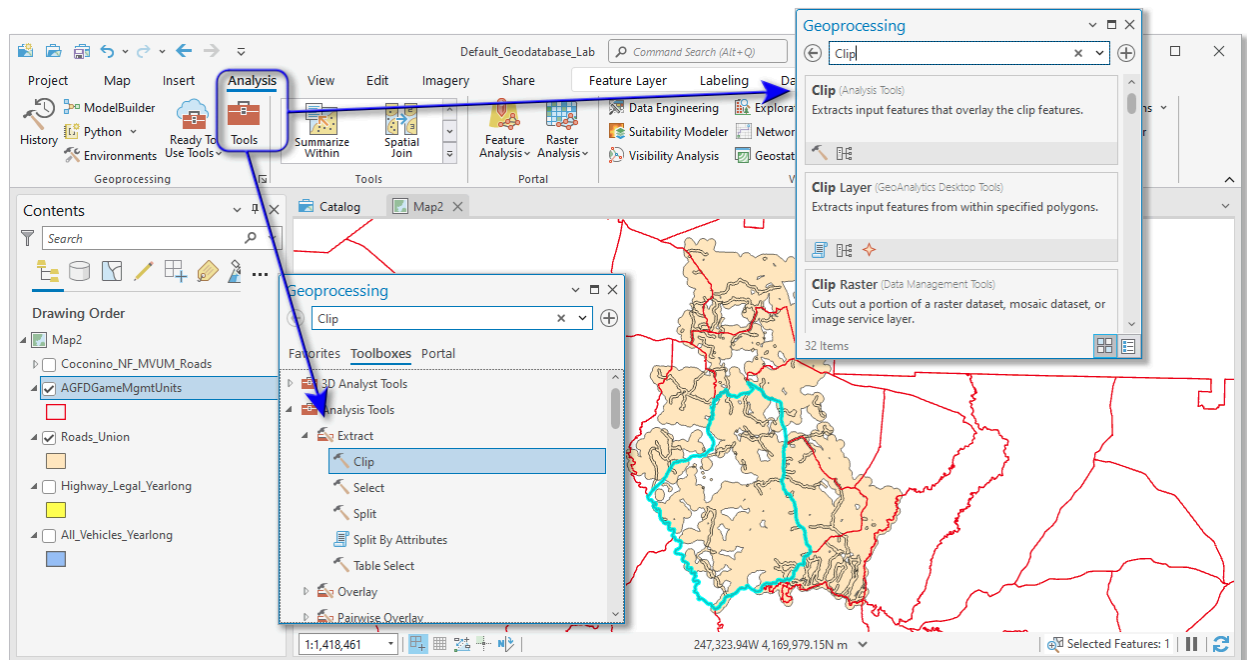




- a) **Important!** Do **not** select any features from **Roads_Union**! If you accidentally selected any **Roads_Union** features, clear that selection by opening the attribute table and clicking the *Clear Selection* command .

Open and run Clip tool

- 5) Open your **Clip** tool by clicking the **Analysis** tab, and **Tools**, and then choosing **Clip**:
- a) You can find the **Clip** tool by either searching for it directly, or by open up the **Toolboxes**, then **Analysis Tools**, then **Extract**, then **Clip**.

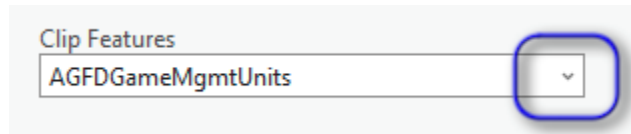


6) In the **Clip** parameters window, set the following:

a) *Input Features or Dataset* = **Roads_Union**

b) *Clip Features* = **AGFDGameMgmtUnits**

i) **Important:** As you did with the **Buffer** tool exercise, make sure you select the Clip feature dataset by either clicking the down-arrow in the “Clip Features” box:



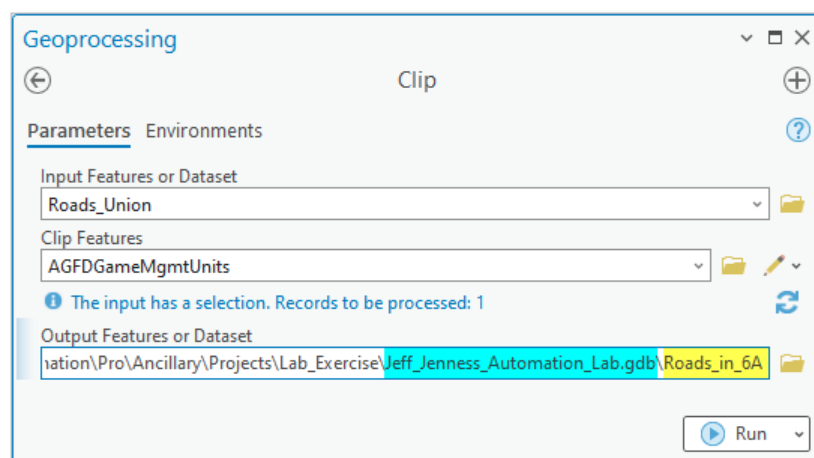
... or else by dragging the layer from your map into that box. **Do not use the  button!** The reason for not using the  button is important to understand:

Remember that we want to clip the road buffers to Management Unit 6A, and we have that polygon selected in **AGFDGameMgmtUnits**. Therefore we want to restrict the analysis to the **selected** features. This selection only exists in the actual **layer** in the map, not in the dataset on the hard drive. The selection is part of the layer definition.

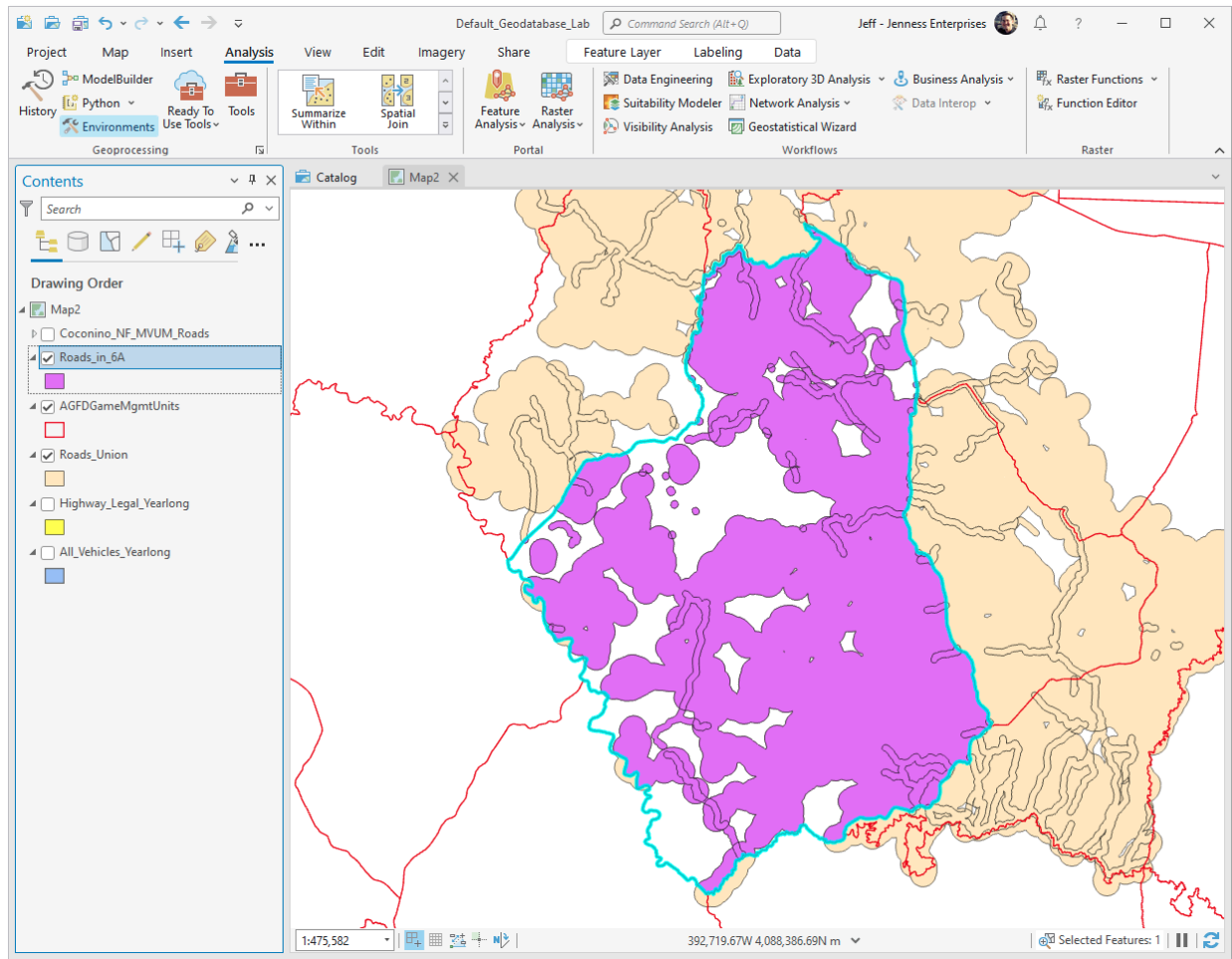
The **Clip** tool lets us load data to clip to by either using a **Layer** from our map, or a **Feature Class** from the hard drive. The  button is what we use to load a feature class from the hard drive. But if we do this, then the **Clip** tool will not know about our selected management unit. It will clip the road buffers to the combined region of all game management units.

Therefore, in order to restrict the clip operation to only our **selected** management unit, we have to specify the **Layer**. And we do that with either the Down-arrow  or by simply dragging the layer over from the map.

c) *Output Feature Class*: Name your new feature class **Roads_in_6A**. Confirm that you are saving to your default geodatabase from Lab Exercise 1. **Important:** You will turn in your file geodatabase as part of the homework, so it is important that this feature class is included!



- 7) Upon completion, you should have a new feature class of Road buffers, clipped to Management Unit 6A.



Exercise 7: Build and Run a Model

Objectives

- Learn how to create a new toolbox.
- Learn to create a new model.
- Learn how to set model component parameters.
- Learn how to delete intermediate data.
- Learn how to run the model.

Data used

- File Geodatabase **Your_Name_Automation_Lab.gdb**, which you created in the first lab.
- Feature Class **Rock_Project_Unit**, in file geodatabase **Your_Name_Automation_Lab.gdb** from Lab Exercise 1. This is the boundary of the Rock Project Management Unit in the NAU Centennial Forest, and you made it in Lab Exercise 3.
- Polyline layer **Coconino_NF_MVUM_Roads** (this is the Coconino road status as of 2022) in workspace **Additional_Data**.

Homework Assignment:

You will add a **toolbox**, a **model**, and a **feature class** to the geodatabase you created in Exercise 1. **You will send me the geodatabase as part of your homework.**

Exercise

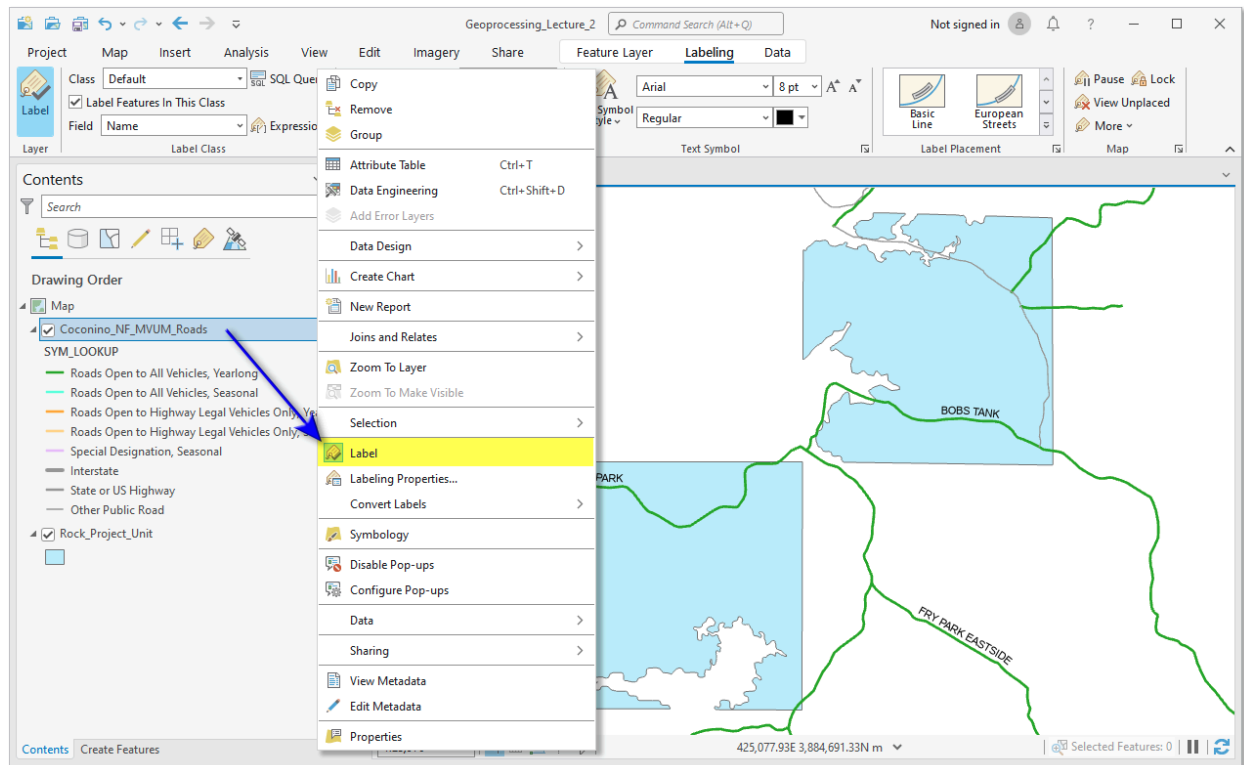
This lab exercise will show you how to create a new model and a new toolbox, and how to set various model parameters. After you run the model, you will learn how to delete intermediate datasets.

Your model will identify that portion of the Rock Project unit which is within ½-mile of Fry Park Road. To do this, your model will first select Fry Park Road from the Coconino National Forest road map, then generate a ½-mile buffer around it. This buffer will extend through the middle of the Rock Project unit. Your model will then clip that buffer to the Rock Project polygon and add the final feature class to your map.

Prepare your ArcGIS Pro Project

- 1) If it is not still open, then open your ArcGIS Pro project from Lab Exercise 1. Confirm that your default geodatabase is **Your_Name_Automation_Lab.gdb**. If it is not, then follow the instructions in Lab Exercise 1 to set the default geodatabase correctly.
 - a) **Note:** **You will be sending me this geodatabase as part of your homework**, so it is important that you have it set as the default. Otherwise you may not correctly save the required feature class to the database.
- 2) Load the feature class **Rock_Project_Unit** (from your file geodatabase **Your_Name_Automation_Lab.gdb** from Lab Exercise 1) and the layer file **Coconino_NF_MVUM_Roads.lyr** from the folder **..\Class_GIS_Data\Additional_Data\...**

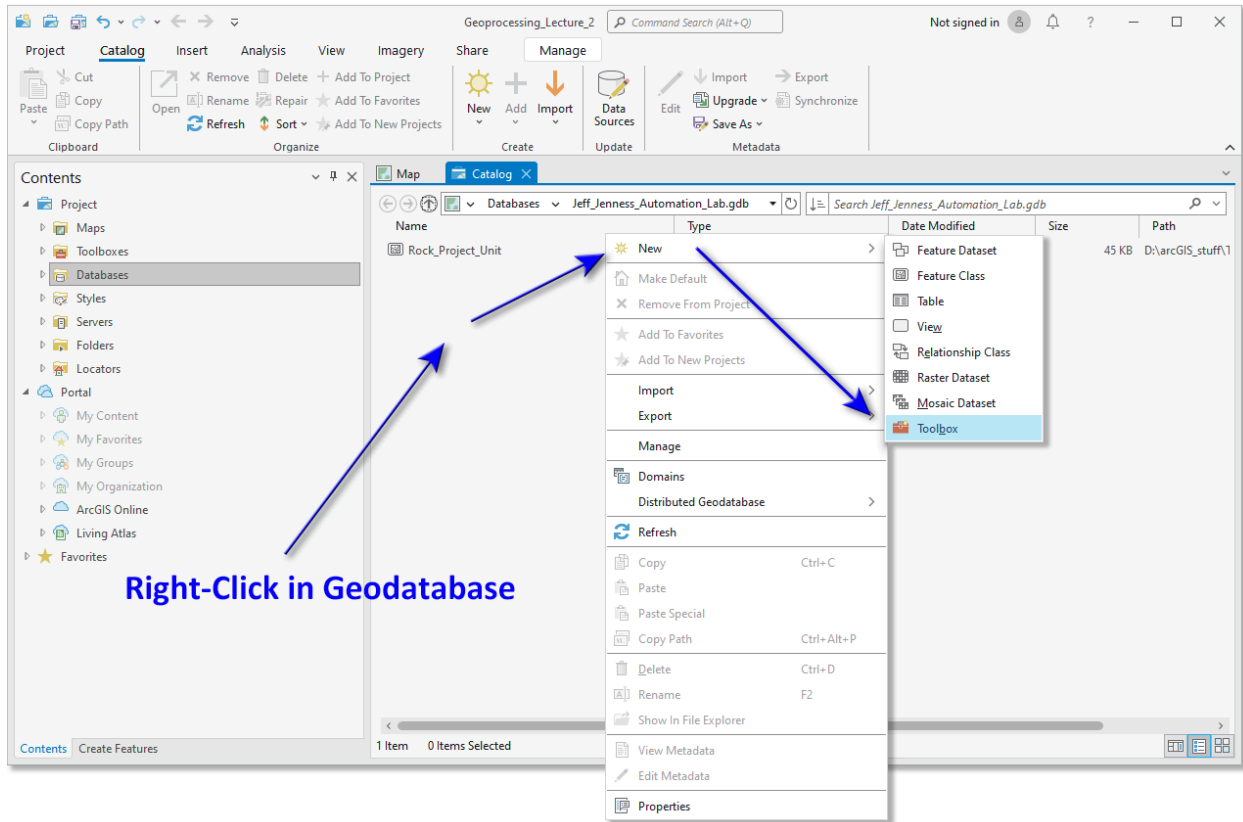
- 3) Turn on the Labels for **Coconino_NF_MVUM_Roads** by right-clicking on the layer and clicking “Label”.



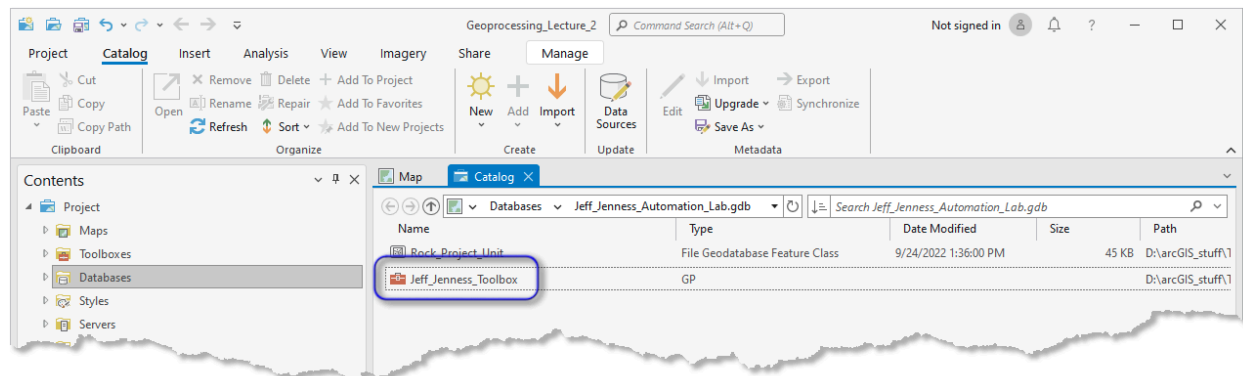
Make a new toolbox and Model

All models **must** be located in a toolbox. You cannot create a model outside of a toolbox, so making the toolbox is the first step. Fortunately it is easy to do.

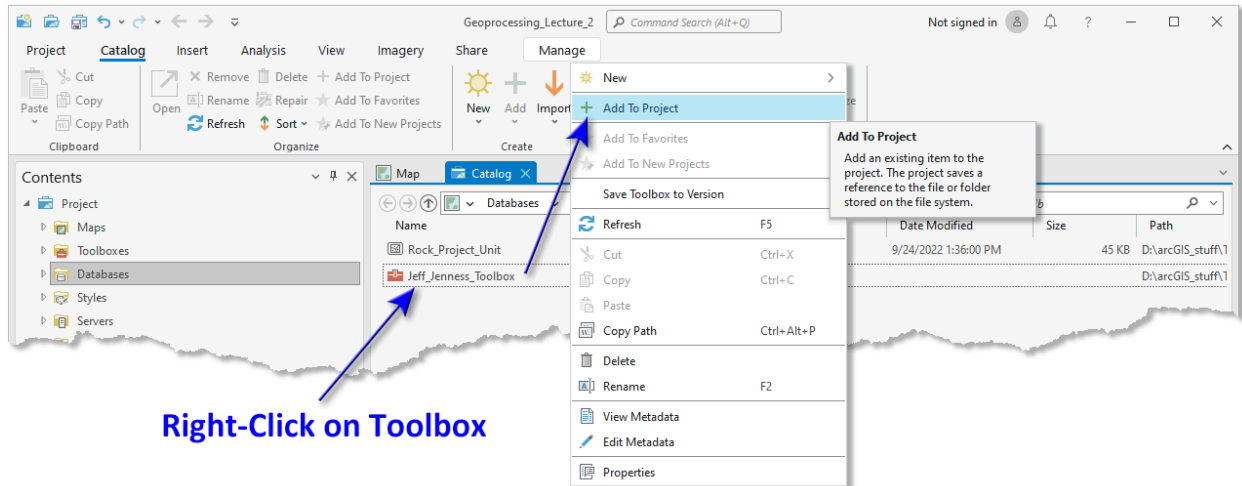
- 4) Open a Catalog view and open your geodatabase **Your_Name_Automation_Lab.gdb**
- 5) Right-click inside your **Your_Name_Automation_Lab.gdb** geodatabase and choose “New”, then “Toolbox”.



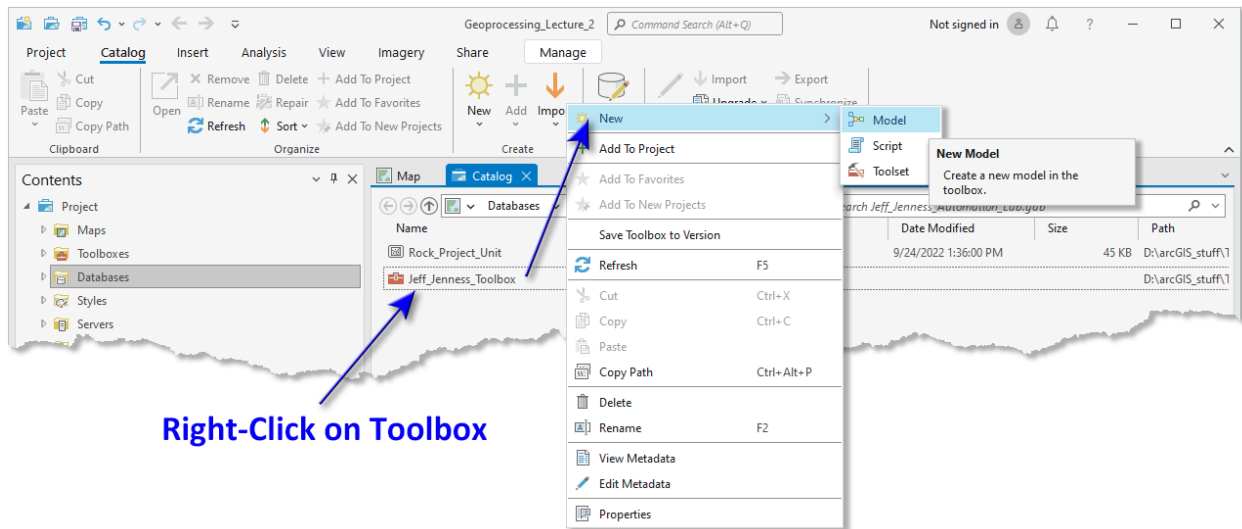
6) Name your new toolbox “Your_Name_Toolbox”



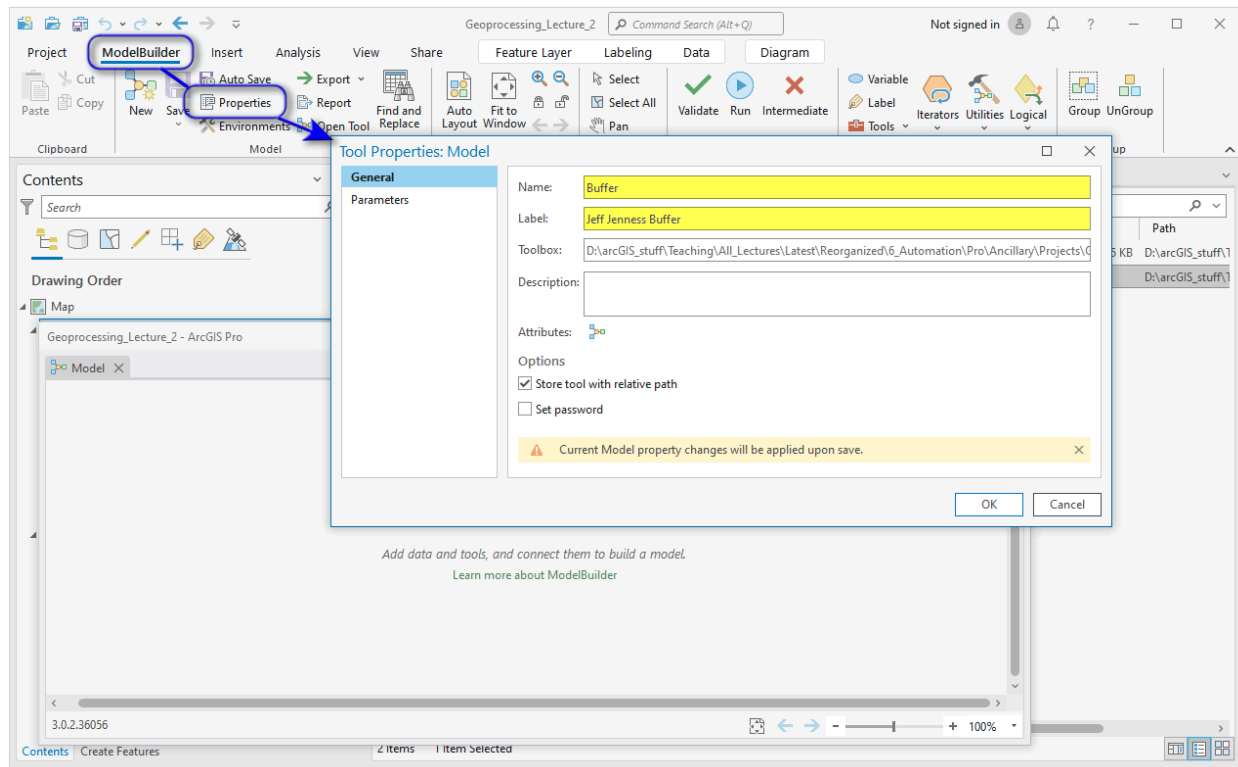
7) Connect your new toolbox to your project by right-clicking on it, and choosing “Add to Project”.



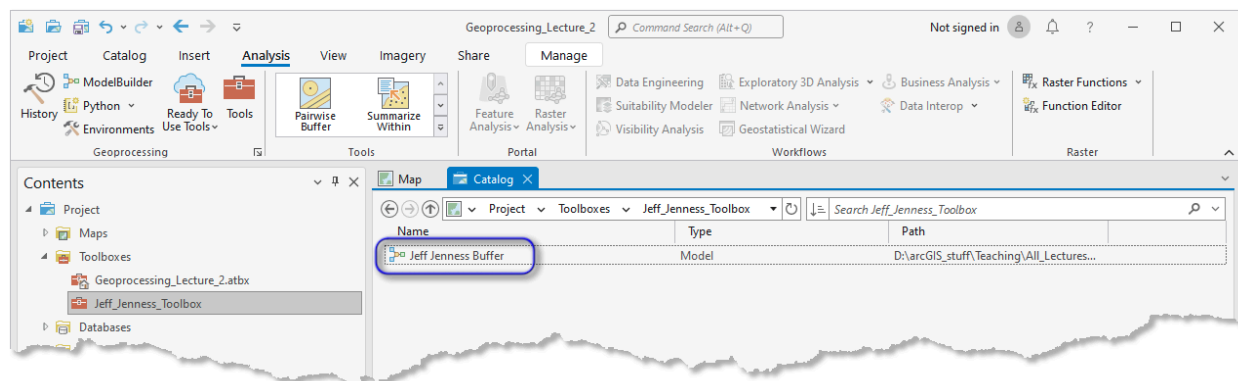
- 8) You now have a new toolbox that you can create models in. Right-click your toolbox and choose “New”, then “Model”:



- 9) Your new model should open in the ModelBuilder window. Click the “Model” menu item, then “Model Properties”. Set the **Name** of your new model to **Buffer** and the **Label** to **Your Name Buffer**.



10) Save your model and close your model window. You should now see your model appear in your toolbox with the correct name.



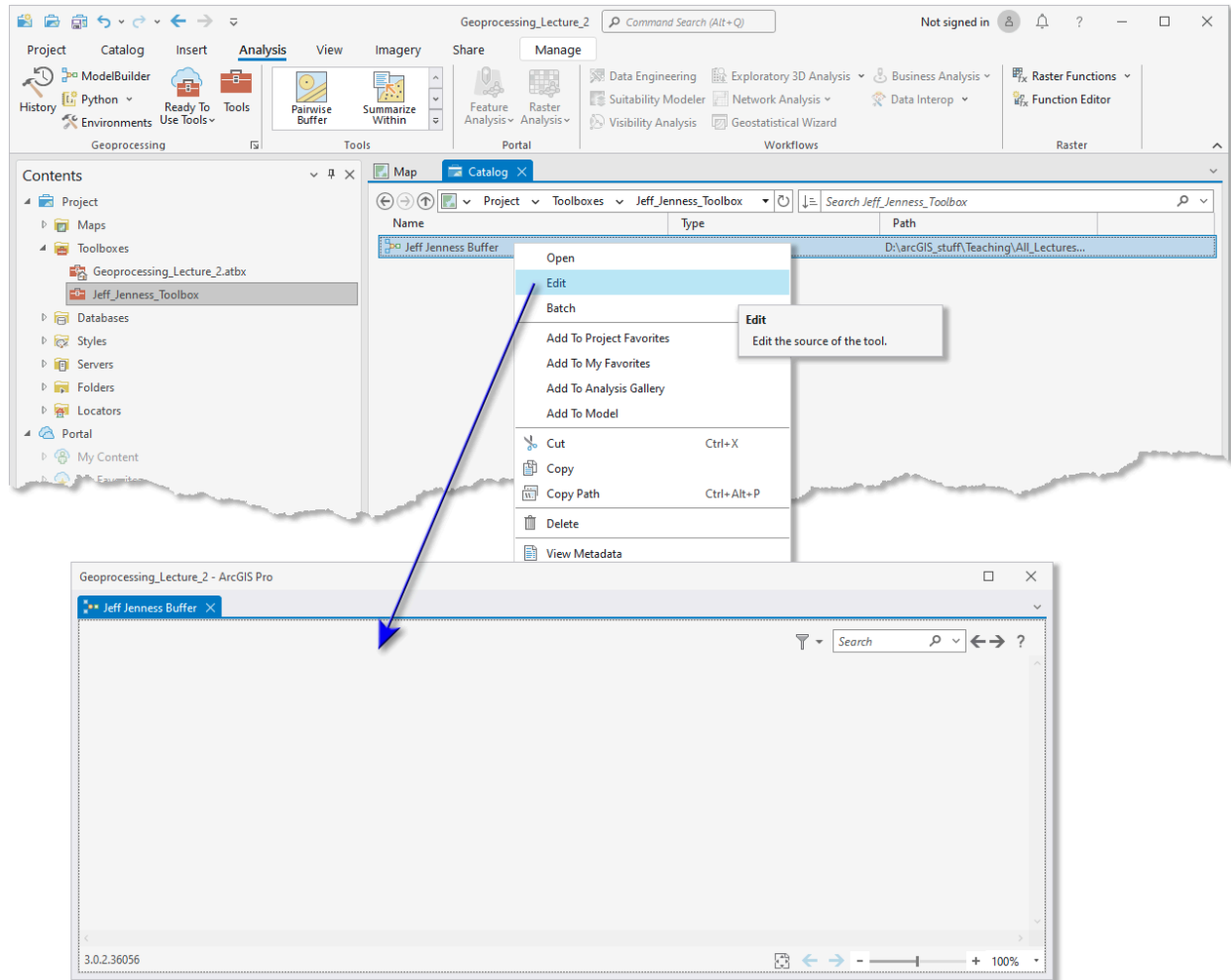
Build the Model:

Reminder: Our final model will take 3 steps:

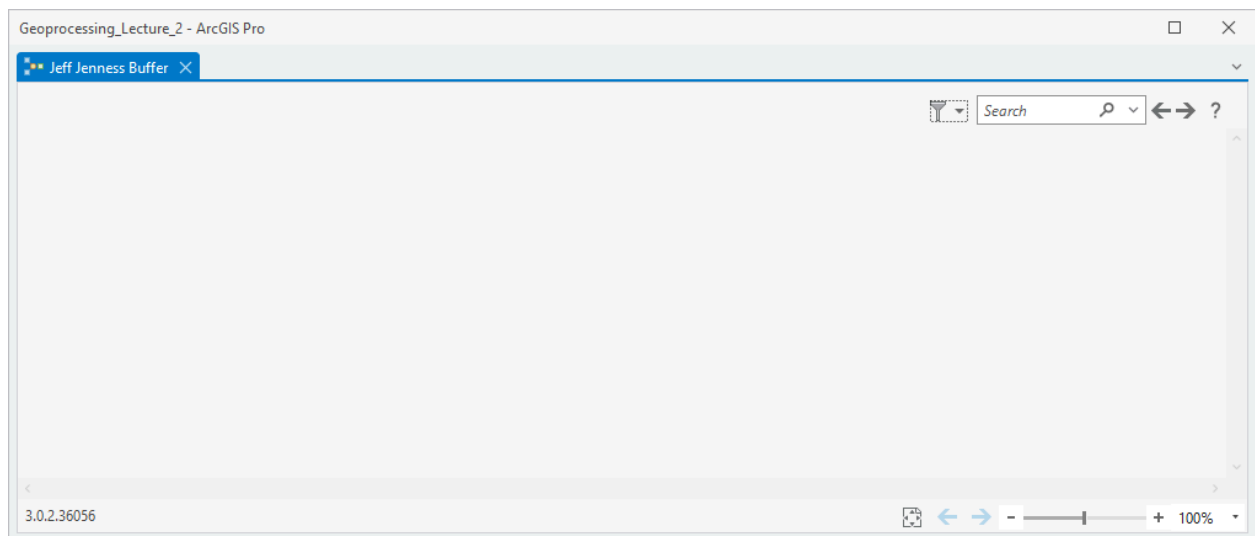
- Select the road named **Fry Park**.
- Build a ½-mile buffer around the **Fry Park** road.
- Clip the buffer to the Rock Project polygon.

11) So now you have an empty model. Let's build it and try it out.

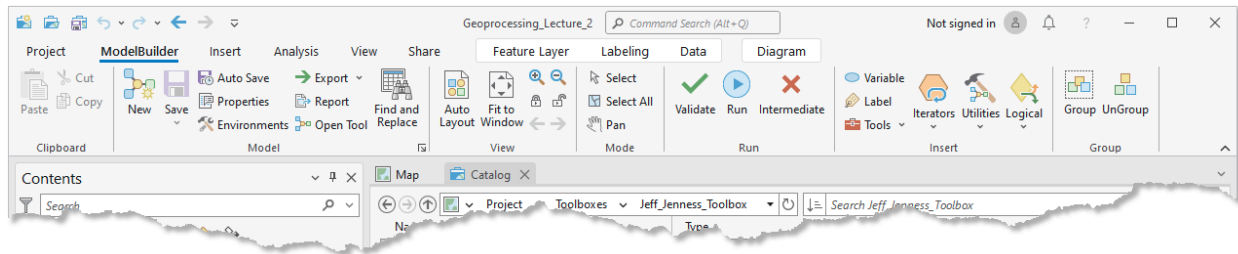
12) Right-click on your new model and click "Edit" to reopen the ModelBuilder window.



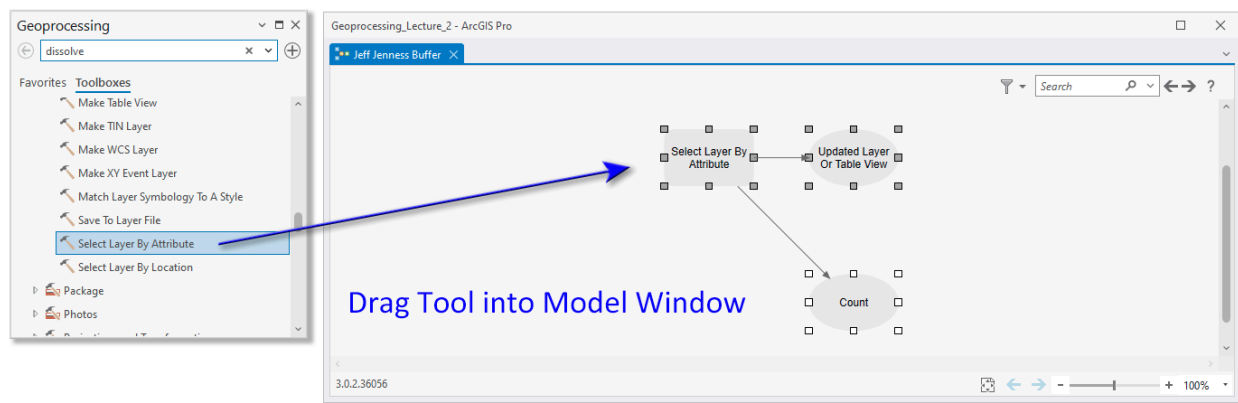
13) Your model should now be open in a ModelBuilder window:



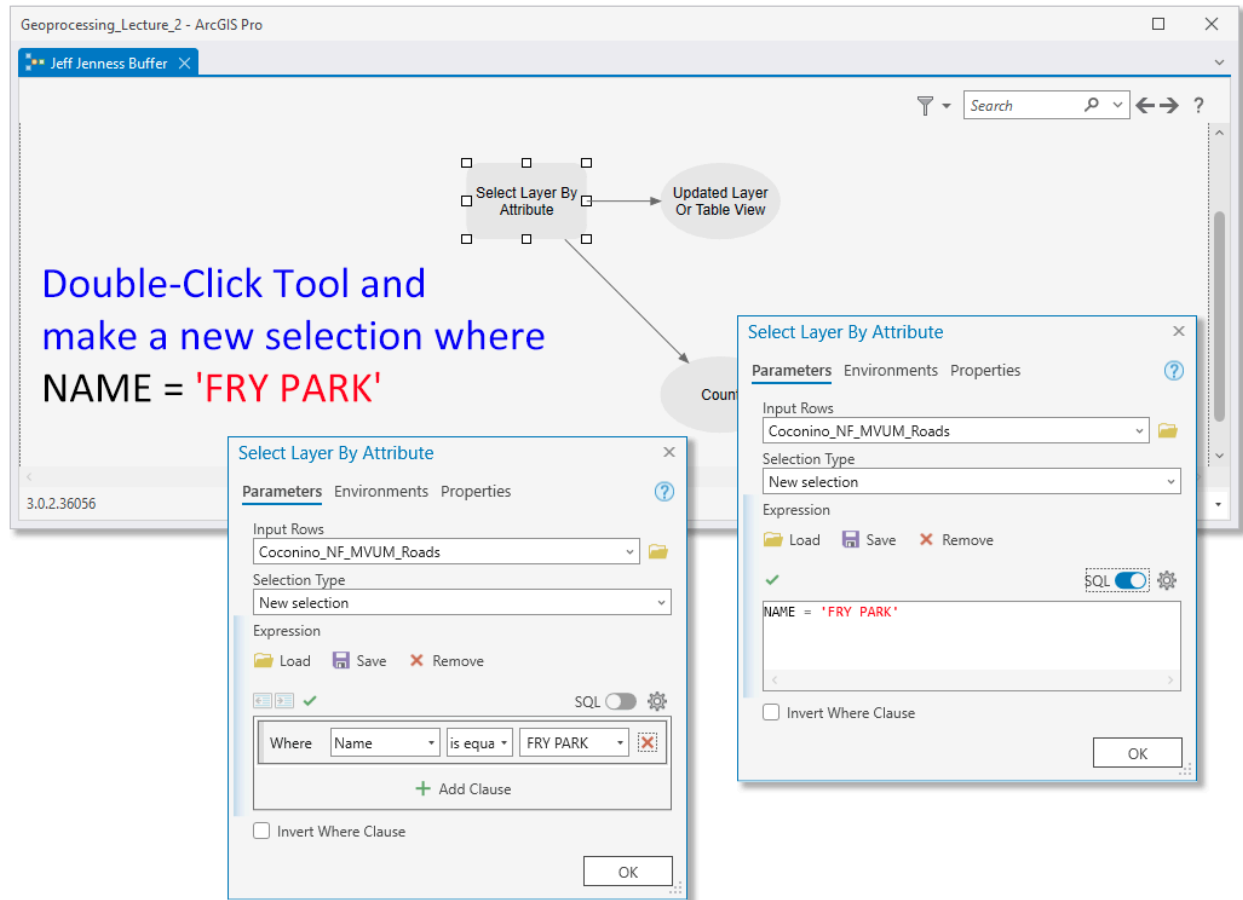
14) The tools to work with the model are in the **Model Builder** ribbon at the top of ArcGIS Pro:



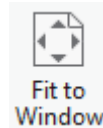
- 15) To add tools, just drag them directly from the Toolbox pane into the ModelBuilder window. First add the “Select Layer by Attribute” tool (located in “Toolbox”, then “Data Management Tools”, then “Layers and Table Views”, then “Select Layer by Attribute”): **Note: Remember to drag the tool from the Toolbox pane into ModelBuilder! Do not just click on the tool to run it. We’re not going to run it until we get the entire model built..**



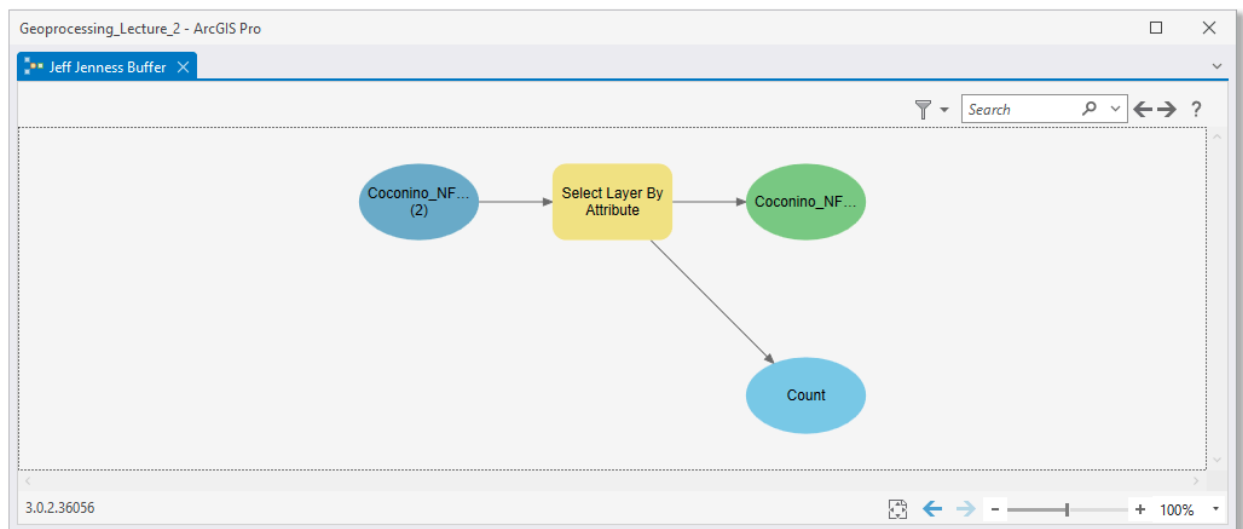
- 16) After adding the tool to ModelBuilder, you should notice that there are three components: A tool (rounded rectangle) and two variables (oval), both of which are grayed out, indicating that they are **not** ready to be run. You need to set the tool parameters.
- 17) Double-click on the tool symbol in ModelBuilder to open the “Select Layer by Attribute” window. Set the *Input Rows* to be **Coconino_NF_MVUM_Roads**, the *Selection Type* to be **New Selection**, and the query to be where the road name is equal to Fry Park.



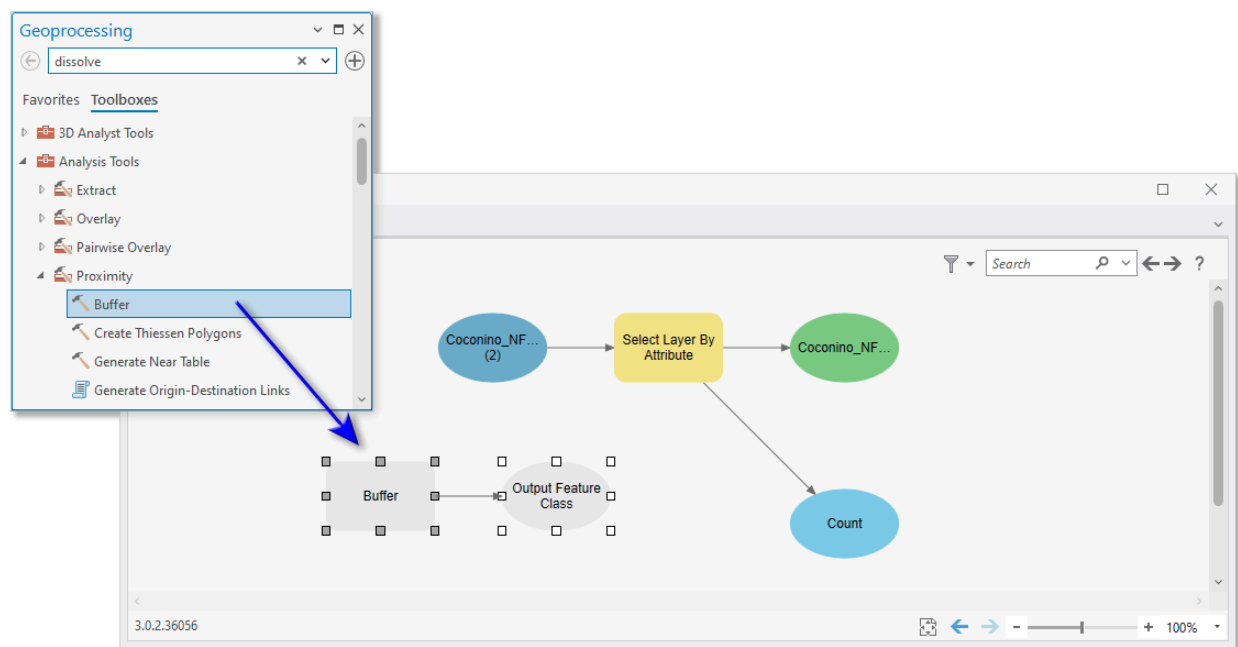
18) Click “OK and observe that all the components in your model now take on colors, meaning they are



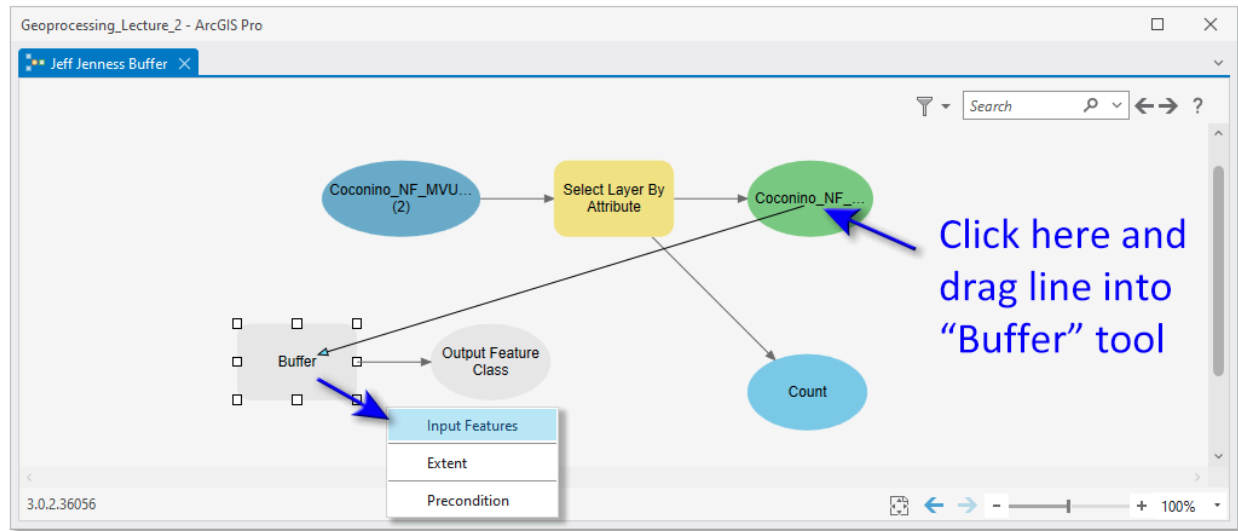
ready to run. Click the “Fit to Window” command in the **ModelBuilder** ribbon to expand the model to fit the window:



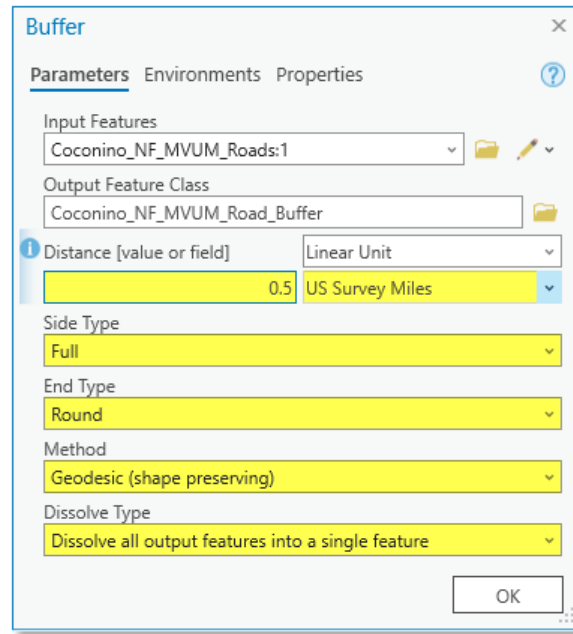
- 19) Notice that your **Select Layer by Attribute** tool has a **Dark Blue oval** in front of it, indicating the input dataset being used by the operation.
- 20) Notice that your **Select Layer by Attribute** tool is producing two outputs:
- a) A new layer showing the results of your selection. This layer should only contain the two road segments named **Fry Park**. This oval is **Green**, indicating that it is data to be derived from the operation.
 - b) A **Number** representing the number of features that satisfy the query. This output oval is **Light Blue**, indicating that it's a single value derived from the operation. **Note:** We won't actually use this "Count" output for anything.
- 21) So now we are 1/3 of the way done. Next drag the **Buffer** tool into the ModelBuilder window (from "Toolbox", then "Analysis Tools", then "Proximity", then "Buffer"). It doesn't matter particularly where you put it, but just make sure you can see it.



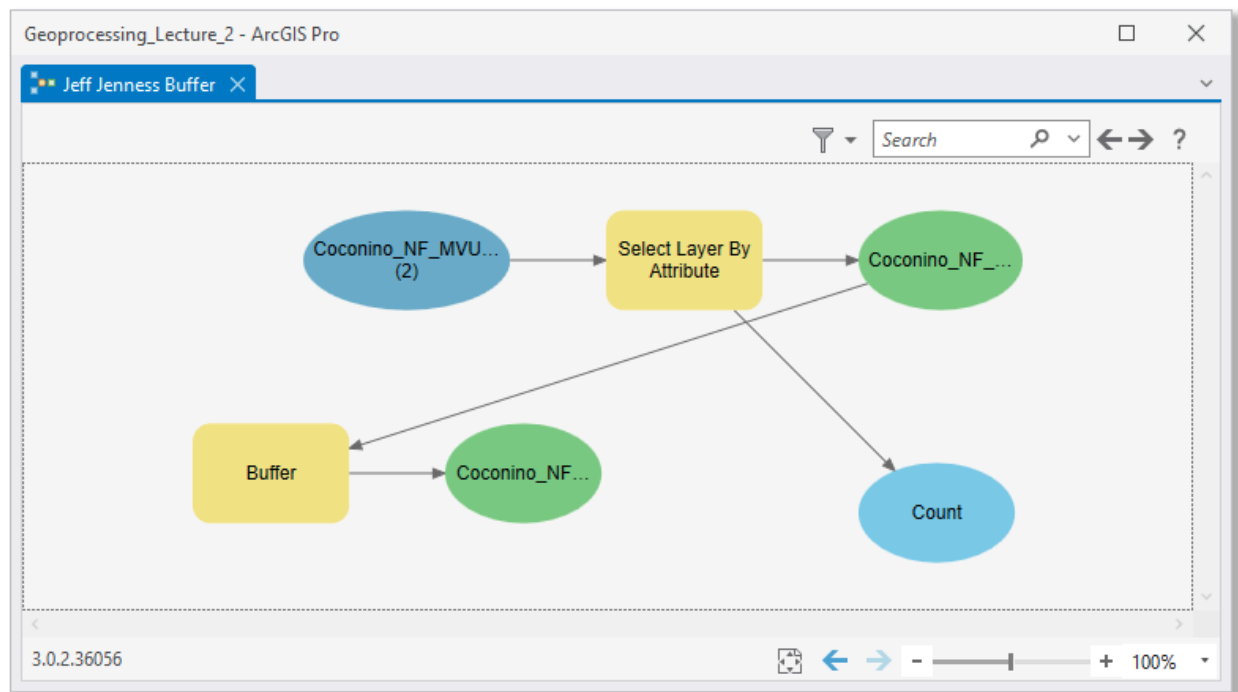
- 22) As before, the Buffer tool is not ready to run because you have not set any of the parameters. Click in your **Green** oval (representing the selected roads) and drag a line down to the **Buffer** tool. When you let up off the mouse key, select the option "Input Features".



- 23) Double-click on the Buffer rectangle in the ModelBuilder window to set the rest of the Buffer parameters:
- Your “Input Features” should already be set. The line you drew connecting the **Green** oval to the buffer tool set this value.
 - Don’t worry too much about the Output Feature Class. This is only an intermediate dataset that we will delete after the tool has run.
 - Set your “Linear Unit” to 0.5 US Survey Miles, and your “Dissolve Type” to “Dissolve all output features into a single feature”. Your “Buffer” tool parameter window should look like this before you click “OK”.
 - Set “Side Type” = **Full**
 - This option lets you buffer just one side of a line if you ever want to. You can choose Left, Right or All.
 - Set “End Type” = **Round**
 - This option lets you create flat ends at the ends of lines if you want, such that the buffer is trimmed off right at the end of the line. We want the buffer to extend around the endpoint, so we choose “Round”
 - Set “Method” = **Geodesic (shape preserving)**
 - Set “Dissolve Type” = **Dissolve all output features into a single feature**
 - This option lets you generate separate buffer polygons around each road polyline. For this exercise, it makes more sense to produce a single dissolved polygon at the end.

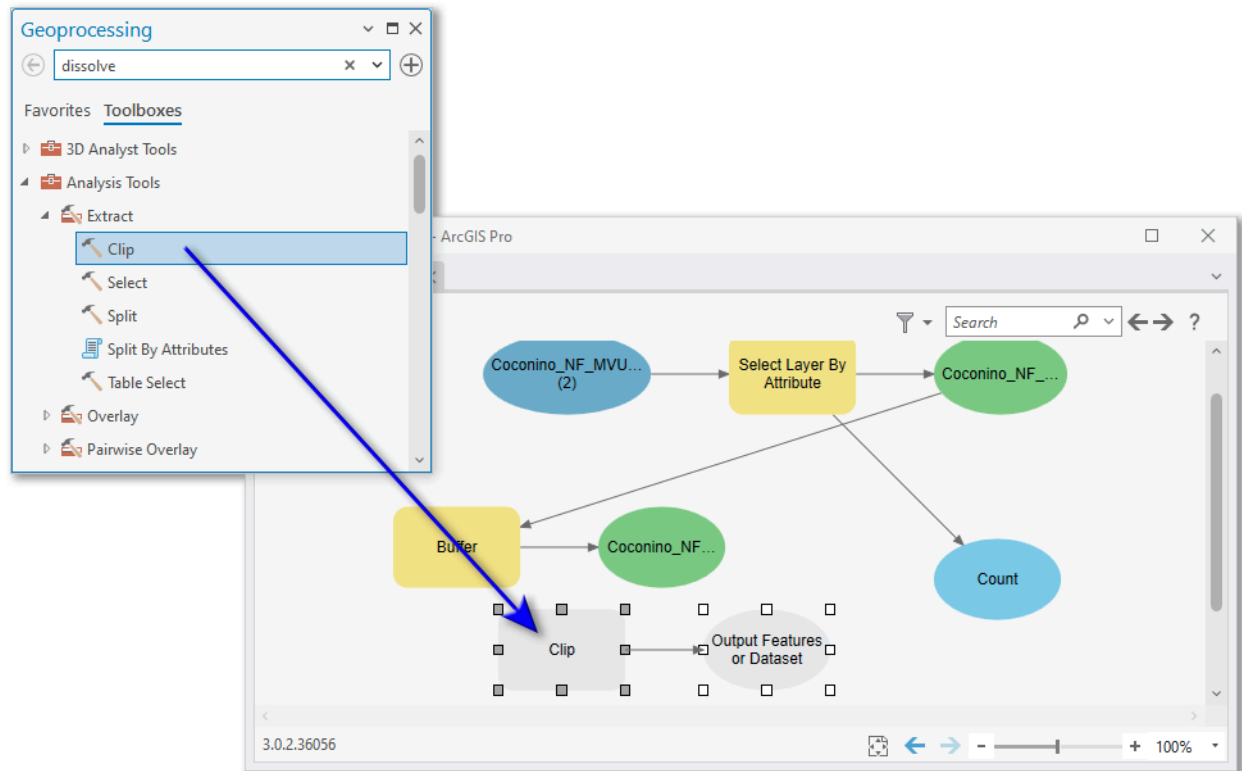


24) Click OK and watch the new colors and lines appear in your model.



25) Now we are 2/3 of the way through. The last step of the model is to clip the buffer to the Rock Project polygon.

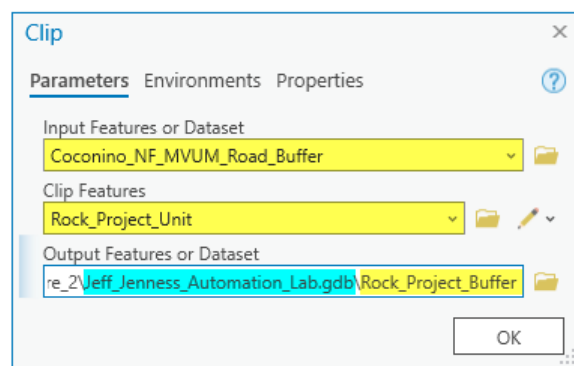
26) Drag in the "Clip" tool (from "Toolbox", then "Analysis Tools", then "Extract", then "Clip").



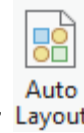
27) Double-click on the “Clip” rectangle to fill in the parameters.

- Set the “Input Features” to be **Coconino_NF_MVUM_Road_Buffer** (or whatever you named the output of your “Buffer” tool).
- Set your “Clip Features” to be your Rock Project feature class **Rock_Project_Unit**. If you have it loaded in your map, you should see it in the drop-down list under “Clip Features”.
- Set the name of your “Output Feature Class” to **Rock_Project_Buffer**. **Make sure your output feature class is being saved to your Your_Name_Automation_Lab.gdb geodatabase!**

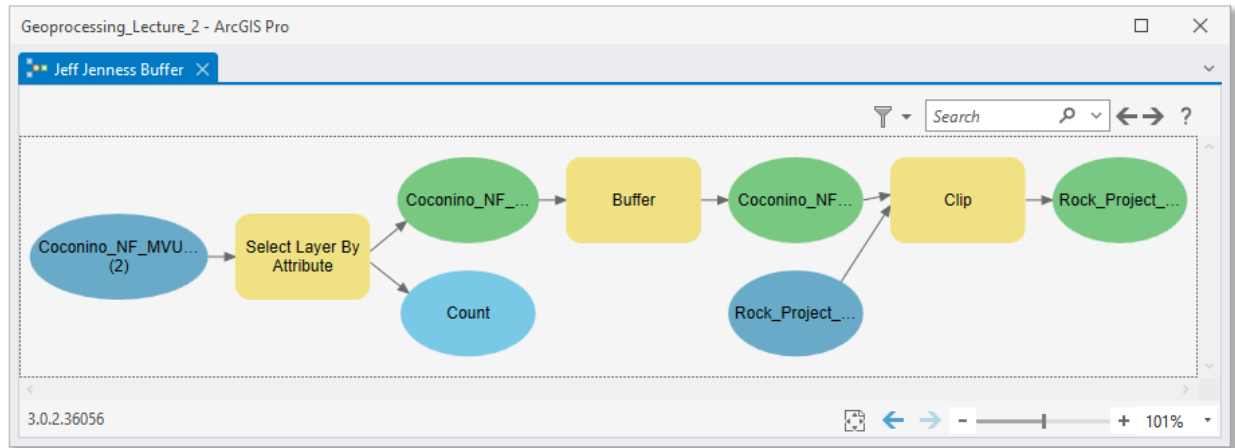
28) Your “Clip” parameters should look like this before you click “OK”.



29) Click “OK” and look at your model. You may wish to click the “Auto-Layout” and “Fit to



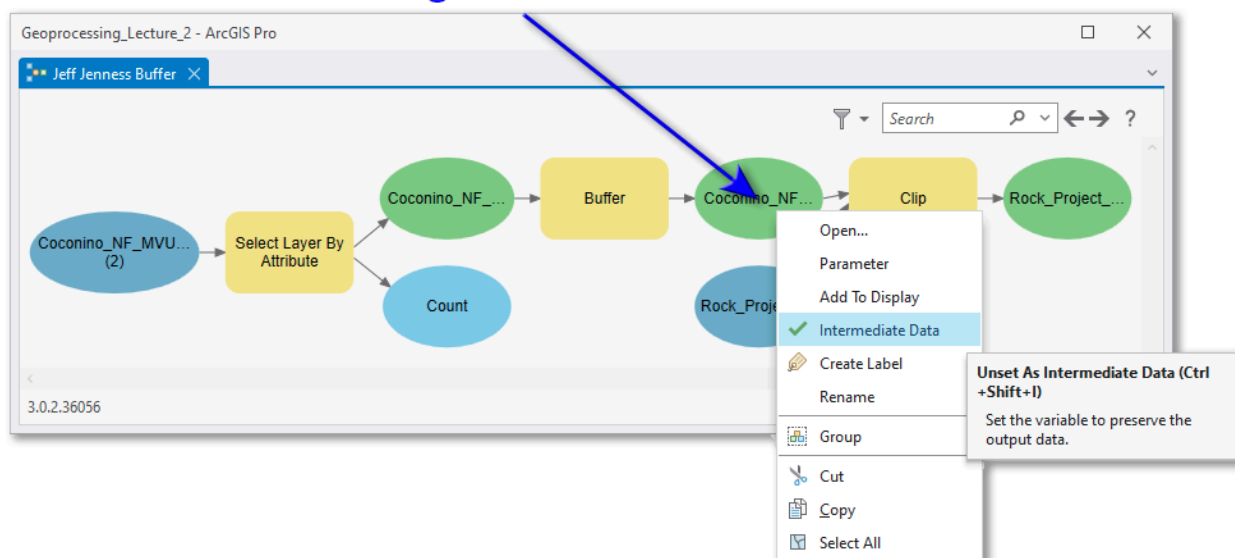
Window” commands in your **ModelBuilder** ribbon at this point to rearrange and reposition your model.



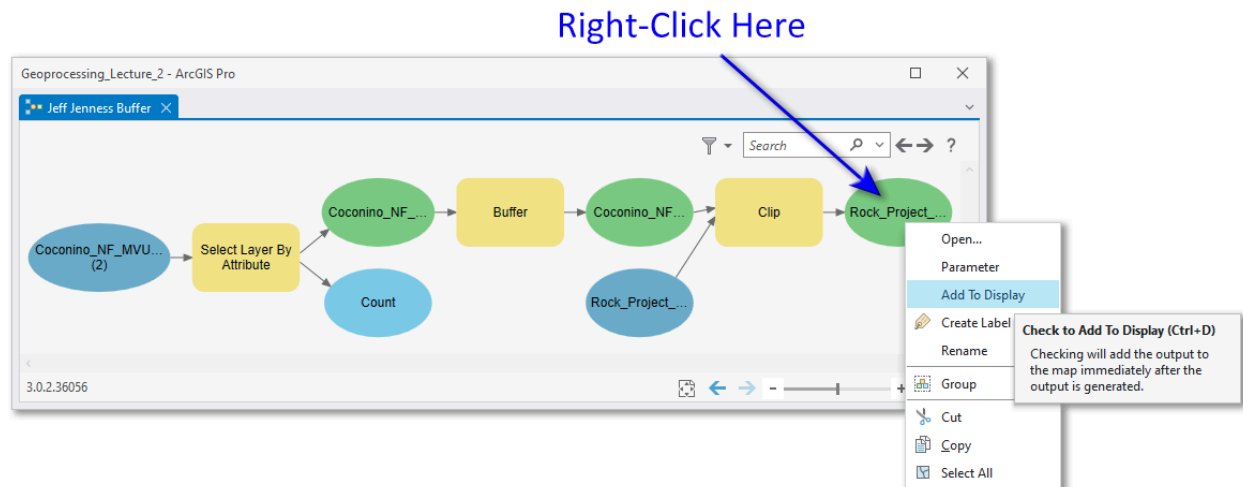
30) At this point your model should be capable of running. However, let’s do two more things before we run it.

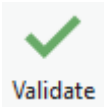
31) Your model makes one intermediate dataset right after the “Buffer” command. Right-click on **Coconino_NF_MVUM_Road_Buffer** (the **Green Oval** produced by your **Buffer** tool) and make sure it is set to “Intermediate” so we can easily delete it after the model finishes.

Right-Click Here



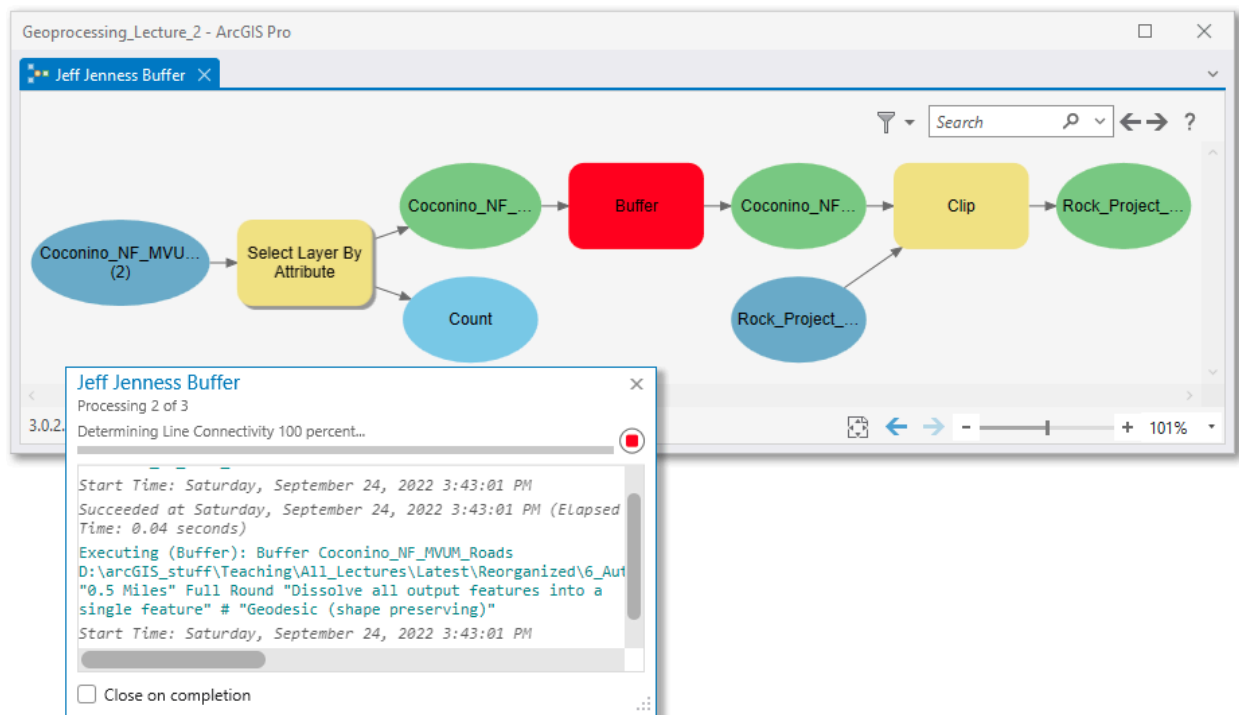
- 32) Then, let's make the final clipped buffer dataset appear in your Map after it is created. Right-click on "Rock_Project_Buffer" and click on "Add To Display".



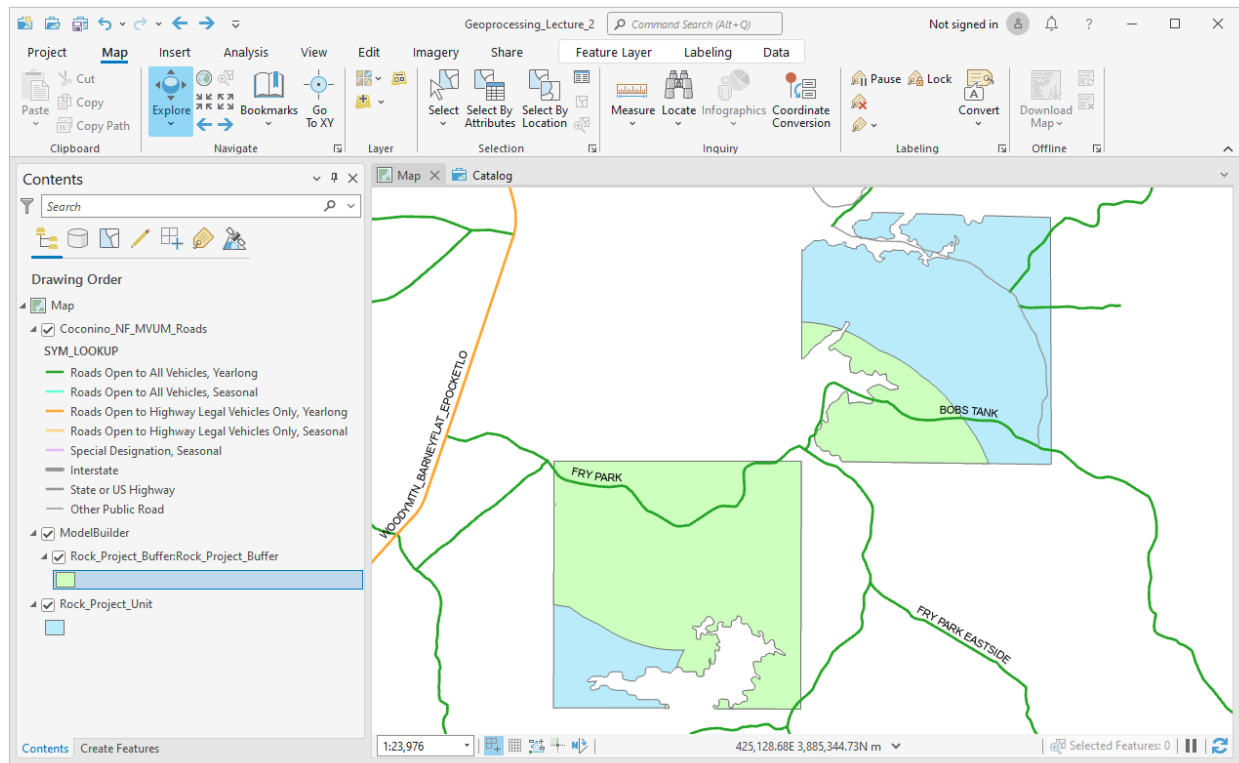
- 33) It should now be ready to run. Give it one last check by clicking the "Validate" command  in your **ModelBuilder** ribbon. If you don't get an error message, then run the model with the "Run"

 command  in the **ModelBuilder** ribbon.

- 34) As the model runs, you should see the current tool highlighted in Red. As tools finish, they should get a shadowed appearance.

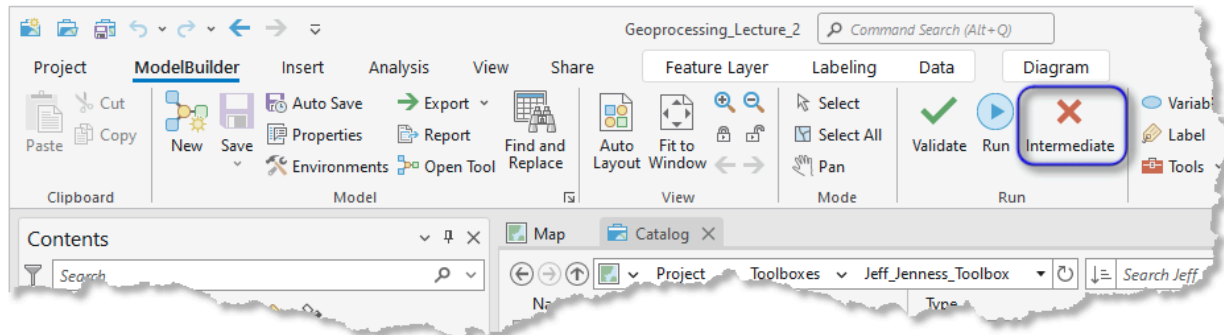


35) Upon completion, you should see your new buffer layer appear in your map.



36) You may also notice that you have both your final **Rock_Project_Buffer** and your intermediate **Coconino_NF_MVUM_Road_Buffer** feature classes in your geodatabase. We never intended to save the intermediate data, so let's delete this. We could delete it fairly easily right from the Catalog window, but let's try the **ModelBuilder** function instead.

37) In the **ModelBuilder** ribbon, click the "Delete Intermediate Data" command **Intermediate** :



38) Notice that your **Coconino_NF_MVUM_Road_Buffer** feature class is deleted, leaving you with only the **Rock_Project_Buffer** feature class.

- Important:** Part of this lab exercise is learning to use the "Delete Intermediate Data" function, so you will lose points if your intermediate **Coconino_NF_MVUM_Road_Buffer** feature class is still in the geodatabase when I check it!

For your Homework:

39) **Save** and **close** your model.

40) **Save your ArcGIS Pro project!** You are going to use it again it for Lab Exercise 8 below.

41) Close ArcGIS Pro. ArcGIS Pro must be closed before you can do step 42 below.

42) Zip your **Your_Name_Automation_Lab.gdb** geodatabase containing your toolbox and your feature classes from the exercises above into a zip file. You will eventually send it to me along with your homework document.

Exercise 8: Export Model to Python Code

Objectives

- Learn how to export a model to Python code.
- Learn how to review and modify Python code.
- Learn how to run modified Python code.

Data used

- ArcGIS Pro Project from Lab Exercise 7 above. This document should have the default geodatabase set to **Your_Name_Automation_Lab.gdb**, which contains the toolbox and model you created.

Homework Assignment:

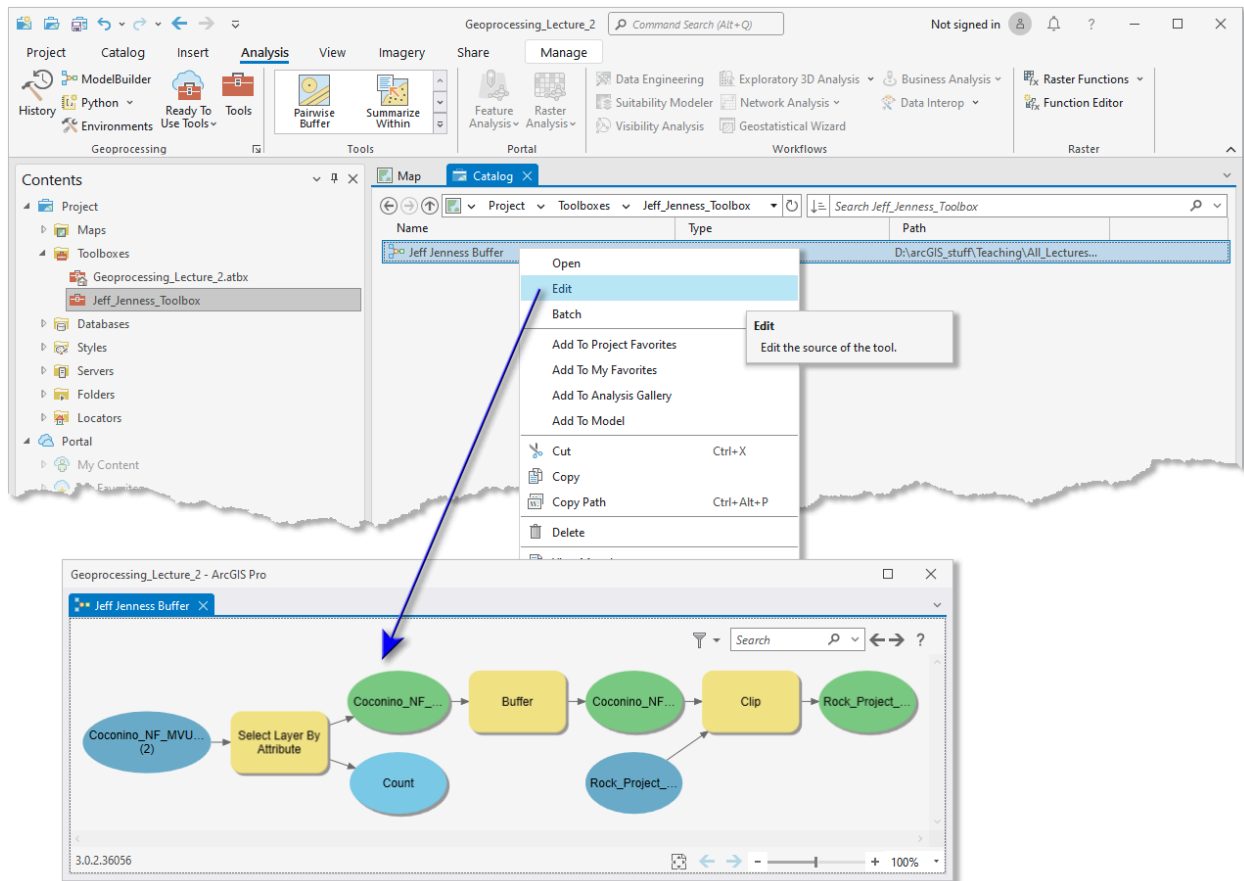
You will export your model to a Python script, then copy-and-paste that Python code into your Homework document. There is also a question to answer in the lab exercise below

Exercise

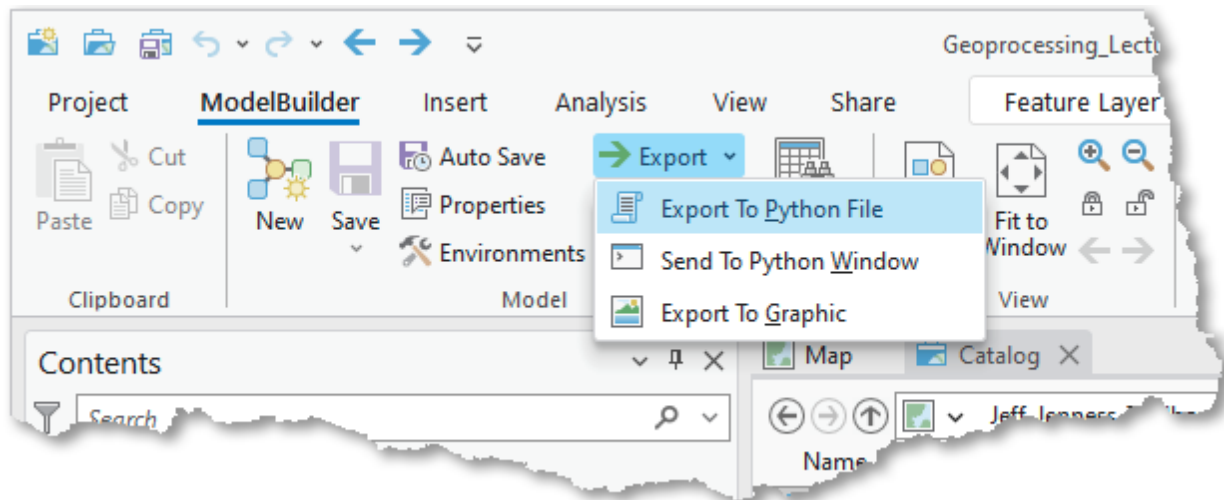
This lab exercise will show you how to export a model to a Python script. It will then show you how to open it and see the different components of the script, and how you can easily change some of the parameters. Finally, you will run your revised Python script to create 0.25-mile buffers in your Rock Project Unit polygon

Prepare your ArcGIS Pro Project

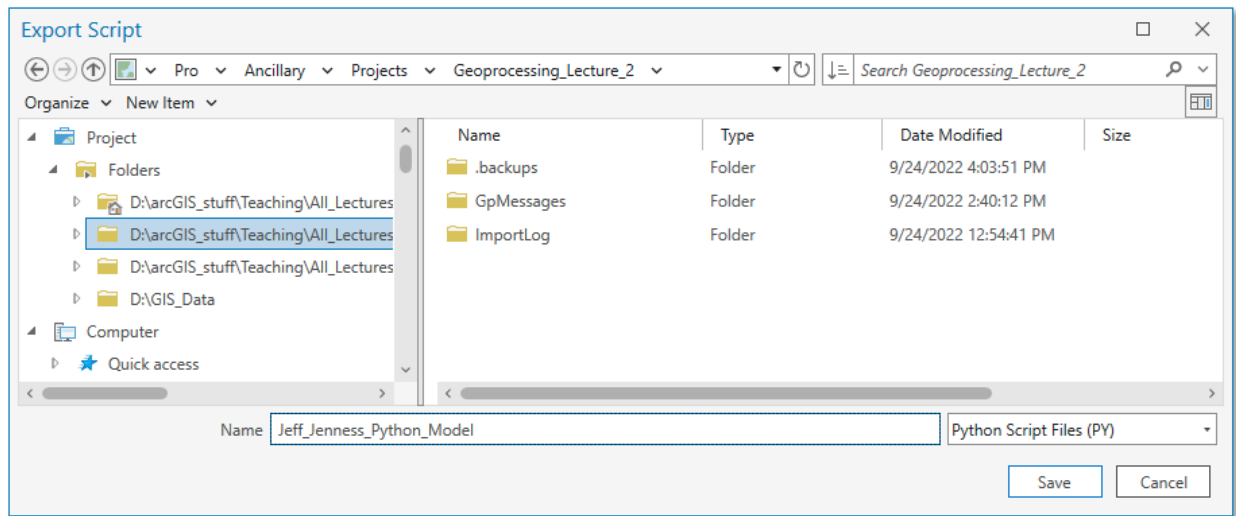
- 1) Open your ArcGIS Pro Project from Lab Exercise 7. Confirm that your default geodatabase is **Your_Name_Automation_Lab.gdb**. If it is not, then follow the instructions in Lab Exercise 1 to set the default geodatabase correctly.
- 2) Open your toolbox, then right-click on your model and click “Edit”:



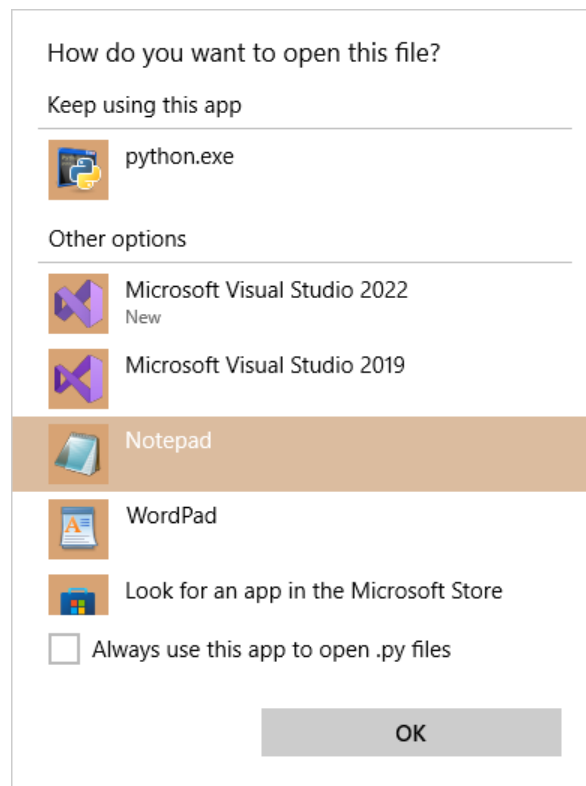
3) Your model will open. In the **ModelBuilder** toolbar, click **Export**, then **Export to Python File**:



4) Name it something like “Your_Name_Python_Model.py”, and remember where you save it.



- 5) You should now have a Python code file on your computer. Open it in either **Notepad++** (if it is installed), or just **Notepad**.
 - a) Using Windows File Explorer, right-click on the file and choose “Open With”, then choose **Notepad** or **Notepad++** from the options. Please let me know if you have trouble with this! If you don’t see **Notepad** in the list of options, or if there is no list of options, then you’ll have to select click “Choose Default Program” and pick **Notepad** out of the list:



- 6) If opening in **Notepad**, the Python code should look something like this:



```
Jeff_Jenness_Python_Model.py - Notepad
File Edit Format View Help
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-09-24 16:08:05
"""
import arcpy

def Buffer(): # Jeff Jenness Buffer

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

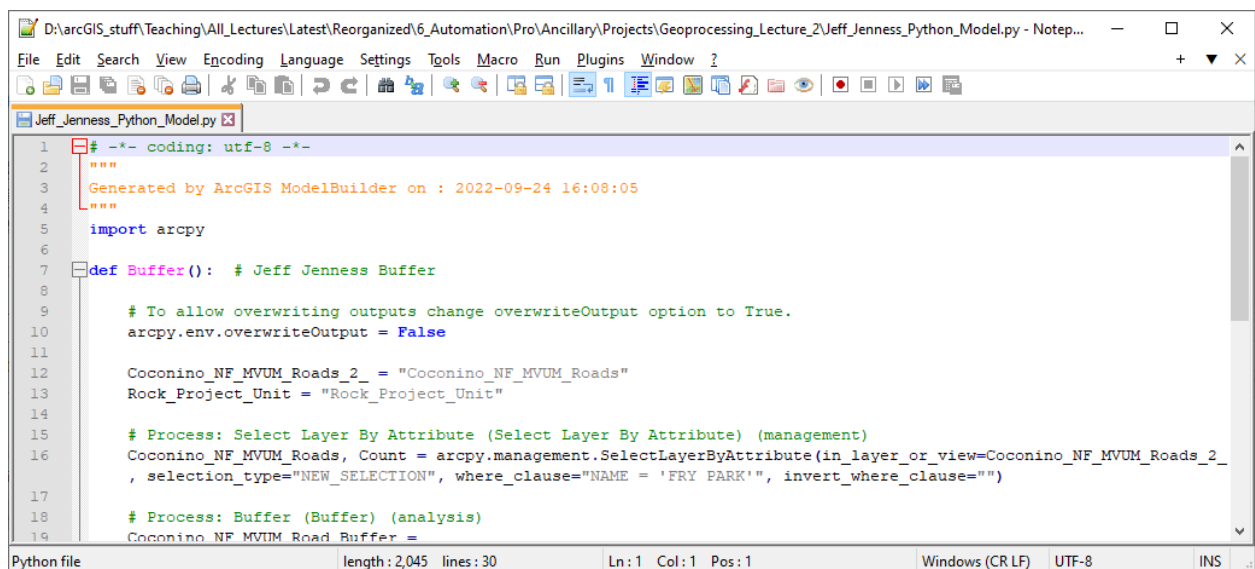
    Coconino_NF_MVUM_Roads_2_ = "Coconino_NF_MVUM_Roads"
    Rock_Project_Unit = "Rock_Project_Unit"

    # Process: Select Layer By Attribute (Select Layer By Attribute) (management)
    Coconino_NF_MVUM_Roads, Count = arcpy.management.SelectLayerByAttribute(in_layer_or_view=Coconino_NF_MVUM_Roads_2, selection_type="NEW_SELECTION", wh

    # Process: Buffer (Buffer) (analysis)
    Coconino_NF_MVUM_Road_Buffer = "D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\Ancillary\\Projects\\Geoprocessing_Lecture_2\\
    arcpy.analysis.Buffer(in_features=Coconino_NF_MVUM_Roads, out_feature_class=Coconino_NF_MVUM_Road_Buffer, buffer_distance_or_field="0.5 Miles", line_side="FUL

    # Process: Clip (Clip) (analysis)
    Rock_Project_Buffer = "D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\Ancillary\\Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_
    arcpy.analysis.Clip(in_features=Coconino_NF_MVUM_Road_Buffer, clip_features=Rock_Project_Unit, out_feature_class=Rock_Project_Buffer, cluster_tolerance="")
```

7) If opening in **Notepad++**, the Python code should look something like this:



```
D:\arcGIS_stuff\Teaching\All_Lectures\Latest\Reorganized\6_Automation\Pro\Ancillary\Projects\Geoprocessing_Lecture_2\Jeff_Jenness_Python_Model.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Jeff_Jenness_Python_Model.py
1 # -*- coding: utf-8 -*-
2 """
3 Generated by ArcGIS ModelBuilder on : 2022-09-24 16:08:05
4 """
5 import arcpy
6
7 def Buffer(): # Jeff Jenness Buffer
8
9     # To allow overwriting outputs change overwriteOutput option to True.
10     arcpy.env.overwriteOutput = False
11
12     Coconino_NF_MVUM_Roads_2_ = "Coconino_NF_MVUM_Roads"
13     Rock_Project_Unit = "Rock_Project_Unit"
14
15     # Process: Select Layer By Attribute (Select Layer By Attribute) (management)
16     Coconino_NF_MVUM_Roads, Count = arcpy.management.SelectLayerByAttribute(in_layer_or_view=Coconino_NF_MVUM_Roads_2_
17     , selection_type="NEW_SELECTION", where_clause="NAME = 'FRY PARK'", invert_where_clause="")
18
19     # Process: Buffer (Buffer) (analysis)
20     Coconino_NF_MVUM_Road_Buffer =
```

Python file length: 2,045 lines: 30 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS

8) Here is a close-up of the left side of the code, with “Word Wrap” turned on:

```

D:\arcGIS_stuff\Teaching\All_Lectures\Latest\Reorganized\6_Automation\Pro\Ancillary\Projects\Geoprocessing_Lecture...
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Jeff_Jenness_Python_Model.py
1 # -*- coding: utf-8 -*-
2 """
3 Generated by ArcGIS ModelBuilder on : 2022-09-24 16:08:05
4 """
5 import arcpy
6
7 def Buffer(): # Jeff Jenness Buffer
8
9     # To allow overwriting outputs change overwriteOutput option to True.
10    arcpy.env.overwriteOutput = False
11
12    Coconino_NF_MVUM_Roads_2_ = "Coconino_NF_MVUM_Roads"
13    Rock_Project_Unit = "Rock_Project_Unit"
14
15    # Process: Select Layer By Attribute (Select Layer By Attribute) (management)
16    Coconino_NF_MVUM_Roads_Count = arcpy.management.SelectLayerByAttribute(in_layer_or_view=
    Coconino_NF_MVUM_Roads_2_, selection_type="NEW_SELECTION", where_clause="NAME = 'FRY
    PARK'", invert_where_clause="")
17
18    # Process: Buffer (Buffer) (analysis)
19    Coconino_NF_MVUM_Road_Buffer =
    "D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\Ancil
    lary\\Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb\\Coconino_NF_MV
    UM_Road_Buffer"
20    arcpy.analysis.Buffer(in_features=Coconino_NF_MVUM_Roads, out_feature_class=
    Coconino_NF_MVUM_Road_Buffer, buffer_distance_or_field="0.5 Miles", line_side="FULL",
    line_end_type="ROUND", dissolve_option="ALL", dissolve_field=[], method="GEODESIC")
21
22    # Process: Clip (Clip) (analysis)
23    Rock_Project_Buffer =
    "D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\Ancil
    lary\\Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb\\Rock_Project_B
    uffer"
24    arcpy.analysis.Clip(in_features=Coconino_NF_MVUM_Road_Buffer, clip_features=
    Rock_Project_Unit, out_feature_class=Rock_Project_Buffer, cluster_tolerance="")
25
26 if __name__ == '__main__':
27     # Global Environment settings
28     with arcpy.EnvManager(scratchWorkspace=
    r"D:\arcGIS_stuff\Teaching\All_Lectures\Latest\Reorganized\6_Automation\Pro\Ancillary\Pr
    ojects\Geoprocessing_Lecture_2\Jeff_Jenness_Automation_Lab.gdb", workspace=
    r"D:\arcGIS_stuff\Teaching\All_Lectures\Latest\Reorganized\6_Automation\Pro\Ancillary\Pr
    ojects\Geoprocessing_Lecture_2\Jeff_Jenness_Automation_Lab.gdb"):
29         Buffer()
30
Python file length: 2,045 lines: 30 Ln: 9 Col: 74 Pos: 224 Windows (CR LF) UTF-8 INS

```

- 9) When the line starts with a # symbol, Python interprets it as a **Comment**. Comments are not run as part of the code. They are just added there by programmers to help them keep track of what the code does. This means that all of the lines in the text that start with “#” will be ignored by ArcGIS Pro. Those lines are only there for you.

10) Observe the three “Process” sections in the middle of the code:

```
<-- Preceding Code cut out here -->

# Process: Select Layer By Attribute (Select Layer By Attribute) (management)

Coconino_NF_MVUM_Roads, Count =
arcpy.management.SelectLayerByAttribute(in_layer_or_view=Coconino_NF_MVUM_Roads_2,
selection_type="NEW_SELECTION", where_clause="NAME = 'FRY PARK'", invert_where_clause="")

# Process: Buffer (Buffer) (analysis)

Coconino_NF_MVUM_Road_Buffer =
"D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\Ancillary\\
Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb\\Coconino_NF_MVUM_Road_Buffer"

arcpy.analysis.Buffer(in_features=Coconino_NF_MVUM_Roads,
out_feature_class=Coconino_NF_MVUM_Road_Buffer, buffer_distance_or_field="0.5 Miles",
line_side="FULL", line_end_type="ROUND", dissolve_option="ALL", dissolve_field=[],
method="GEODESIC")

# Process: Clip (Clip) (analysis)

Rock_Project_Buffer =
"D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\Ancillary\\
Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb\\Rock_Project_Buffer"

arcpy.analysis.Clip(in_features=Coconino_NF_MVUM_Road_Buffer,
clip_features=Rock_Project_Unit, out_feature_class=Rock_Project_Buffer, cluster_tolerance="")
```

11) Do you recognize that these are the three tools you ran (“Select Layer by Attribute”, then “Buffer”, then “Clip”)?

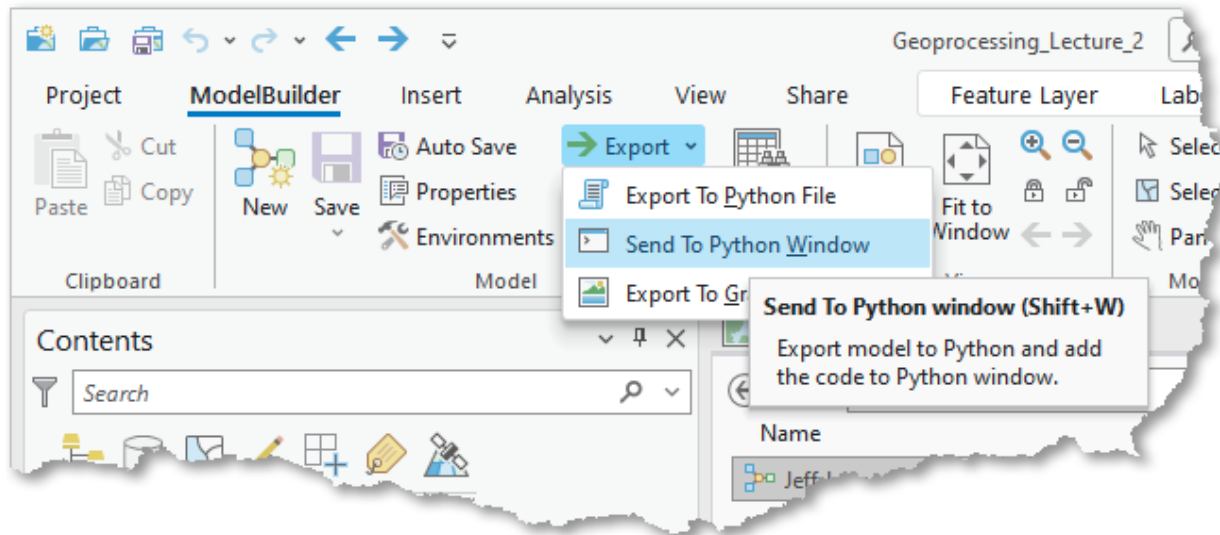
12) Look closely at the lines of code that executes each tool. Do you recognize the parameters in that line of code? For example, do you see the SQL code you used to select the **Fry Park** road? Do you see where you set the Buffer distance to be 0.5 miles?

13) **For your Homework:** You will copy-and-paste all the Python code into your homework document.

14) **Question for Homework:** How would you change the line of code that executes the “Buffer” tool so that it would create an 0.25-mile buffer rather than an 0.5-mile buffer? **Hint:** You’ll do this in the next part of the lab.

15) Continuing on with the lab...

- We will modify the code slightly to create a 0.25-mile buffer rather than a 0.5-mile buffer, and run it in the Python window.
- Return to the **ModelBuilder** ribbon, and click the **Export** drop-down list and choose **Send To Python Window**:



- a) If you stretch out the Python window so you can see all the code, it should look something like this:

```
Python

# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-09-24 16:05:31
"""
import arcpy

def Buffer(): # Jeff Jenness Buffer

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    Coconino_NF_MVUM_Roads_2_ = "Coconino_NF_MVUM_Roads"
    Rock_Project_Unit = "Rock_Project_Unit"

    # Process: Select Layer By Attribute (Select Layer By Attribute) (management)
    Coconino_NF_MVUM_Roads_Count = arcpy.management.SelectLayerByAttribute
    (in_layer_or_view=Coconino_NF_MVUM_Roads_2_, selection_type="NEW_SELECTION", where_clause="NAME = 'FRY
    PARK'", invert_where_clause="")

    # Process: Buffer (Buffer) (analysis)
    Coconino_NF_MVUM_Road_Buffer = "D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation
    \\Pro\\Ancillary\\Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb\\
    \\Coconino_NF_MVUM_Road_Buffer"
    arcpy.analysis.Buffer(in_features=Coconino_NF_MVUM_Roads, out_feature_class=Coconino_NF_MVUM_Road_Buffer,
    buffer_distance_or_field="0.5 Miles", line_side="FULL", line_end_type="ROUND", dissolve_option="ALL",
    dissolve_field=[], method="GEODESIC")

    # Process: Clip (Clip) (analysis)
    Rock_Project_Buffer = "D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\
    \\Ancillary\\Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb\\Rock_Project_Buffer"
    arcpy.analysis.Clip(in_features=Coconino_NF_MVUM_Road_Buffer, clip_features=Rock_Project_Unit,
    out_feature_class=Rock_Project_Buffer, cluster_tolerance="")

    if __name__ == '__main__':
        # Global Environment settings
        with arcpy.EnvManager(scratchWorkspace=r"D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized
        \\6_Automation\\Pro\\Ancillary\\Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb", workspace=r"D:
        \\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\Ancillary\\Projects
        \\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb"):
            Buffer()
```

b) The code is almost ready to run, but we need to change the buffer value and we need to provide new names for the output datasets. Right now it is written to create a new dataset named **Rock_Project_Buffer**, but we already created a dataset with that name in Lab Exercise 7. This will cause the script to crash. Just to be safe, we'll also change the name of intermediate buffer dataset.

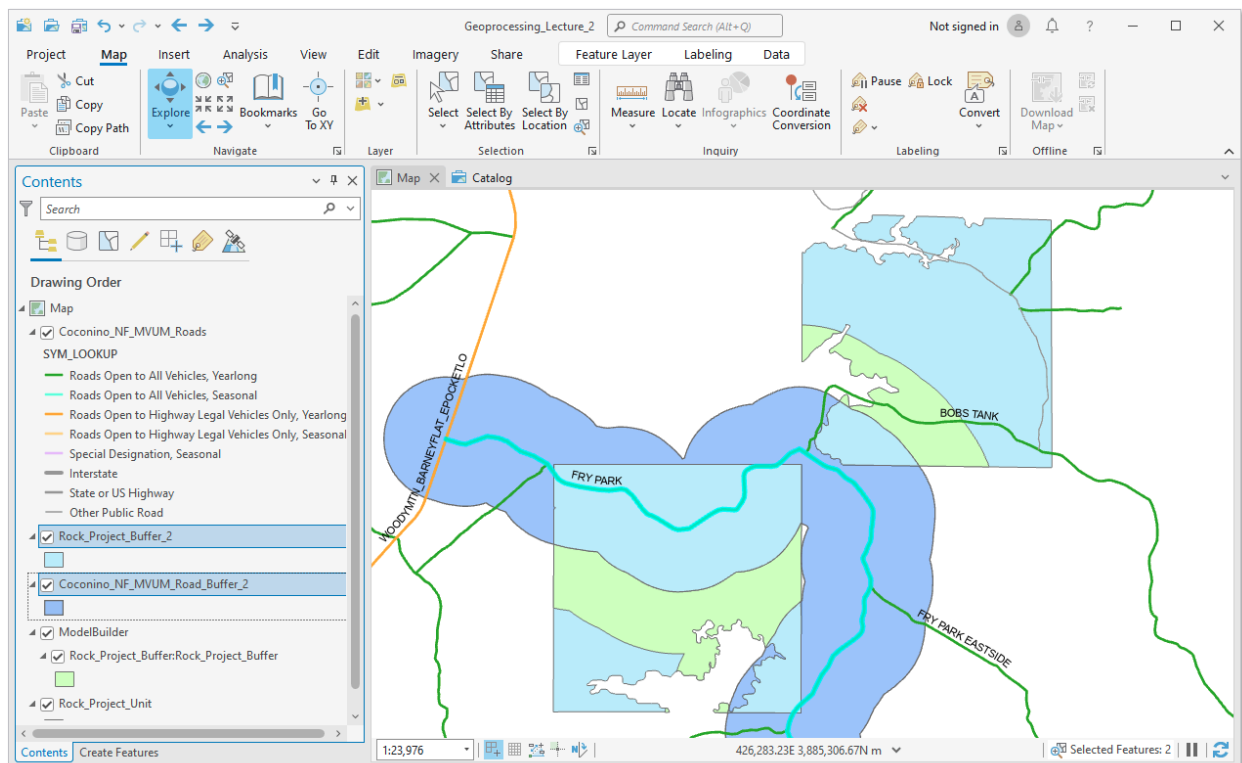
- i) In the **Buffer** process, change the name of the output feature class to **\\Coconino_NF_MVUM_Road_Buffer_2** and change the buffer distance to "0.25 Miles"

```
# Process: Buffer (Buffer) (analysis)
Coconino_NF_MVUM_Road_Buffer = "D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\
\\Ancillary\\Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb\\Coconino_NF_MVUM_Road_Buffer_2"
arcpy.analysis.Buffer(in_features=Coconino_NF_MVUM_Roads, out_feature_class=Coconino_NF_MVUM_Road_Buffer,
buffer_distance_or_field="0.25 Miles", line_side="FULL", line_end_type="ROUND", dissolve_option="ALL", dissolve_field=[],
method="GEODESIC")
```

- ii) In the **Clip** process, change the name of the output feature class to

```
# Process: Clip (Clip) (analysis)
Rock_Project_Buffer = "D:\\arcGIS_stuff\\Teaching\\All_Lectures\\Latest\\Reorganized\\6_Automation\\Pro\\Ancillary\\
\\Projects\\Geoprocessing_Lecture_2\\Jeff_Jenness_Automation_Lab.gdb\\Rock_Project_Buffer_2"
arcpy.analysis.Clip(in_features=Coconino_NF_MVUM_Road_Buffer, clip_features=Rock_Project_Unit,
out_feature_class=Rock_Project_Buffer, cluster_tolerance="")
```

- iii) When you are ready to run the code, just hit the **[Enter]** key and it will run.
- iv) When the code finishes, take a look at your map. You should see that the code added both the intermediate buffer and the final clipped buffer.



- 2) Food for Thought:** One cool thing about Python is that you can use it to run ArcGIS functions without ArcGIS actually being turned on. People often do this when they have regular updates to their data. They will have a place for new data to be stored, and then they'll have a Python script set to run in the middle of the night every night that will process that data and store it in the appropriate geodatabases.

Exercise 9: GeoProcessing Intersect and Landfire

Objectives

- Learn the **Intersect** tool.
- Learn the **Summarize** tool.
- Learn what Landfire data looks like.
- Summarize Landfire Vegetation characteristics for your project area.

Data used:

- Your stand boundaries feature class **Stands_RockPrj_CF_2014** in workspace `..\Class_GIS_Data\RockPrj_CF_2014\..`
- LandFire “Existing Vegetation **Type**” feature class **Landfire_EVT_140_Rock_Prj** in workspace `..\Class_GIS_Data\RockPrj_CF_2014\Rock_Project_Landfire.gdb`.
- **Optionally**, Landfire “Existing Vegetation **Cover**” feature class **Landfire_EVC_140_Rock_Prj** in workspace `..\Class_GIS_Data\RockPrj_CF_2014\Rock_Project_Landfire.gdb`.
- **Optionally**, Landfire “Existing Vegetation **Height**” feature class **Landfire_EVH_140_Rock_Prj** in workspace `..\Class_GIS_Data\RockPrj_CF_2014\Rock_Project_Landfire.gdb`.
- **Optionally**, any background data you wish to add, such as Esri Basemap layers **Imagery** or **USA Topo Maps**.

Homework Assignment:

No homework assignment for this one. Just some questions throughout the lab to emphasize particular points.

Exercise:

Preliminary Steps: What is Landfire?

- ☐ This lab exercise uses Landfire data downloaded from <http://www.landfire.gov>. I would have preferred to have all students go to the web site themselves, but downloading Landfire data can sometimes be a little tricky. I'd also like to illustrate a few vector tools in this lab, which don't work on the Landfire data because it comes in raster format.
- ☐ Therefore, I have downloaded a few datasets that cover your project area, and converted the original rasters to polygon feature classes and reprojected them into NAD 1983 UTM Zone 12 coordinates.
- ☐ Take a few minutes to go to <http://www.landfire.gov> and look it over. And please feel free to download additional data for your own purposes! It's a great resource.

- ☐ **Question:** The data included with this lab is version 1.4.0, meaning that this is the “Landfire 2014 Update” to the data. What does the Landfire 2014 Update include? **Hint:** See http://www.landfire.gov/data_overviews.php, and especially https://www.landfire.gov/lf_140.php. What is the text following the words “LF 2014 is a vegetation and fuels mapping effort to...” on that site?

Answer:

- ☐ **Question:** This lab includes 3 datasets: **Existing Vegetation Type (EVT)**, **Existing Vegetation Cover (EVC)** and **Existing Vegetation Height (EVH)**. What are brief descriptions of these three datasets? **Hint:** For brief descriptions, see <https://www.landfire.gov/vegetation.php>, https://www.landfire.gov/documents/Data_Products_Table_2016.pdf and https://www.landfire.gov/documents/LF_Data_Product_Descriptions_2016.pdf

- Existing Vegetation Type:
- Existing Vegetation Cover:
- Existing Vegetation Height:

- ☐ **Question:** What are the names of 3 other Landfire datasets that you feel would be useful to you as a Forester? **Hint:** The same links https://www.landfire.gov/documents/Data_Products_Table_2016.pdf, https://www.landfire.gov/documents/LF_Data_Product_Descriptions_2016.pdf and https://www.landfire.gov/version_comparison.php list and describe a lot more than 3 additional datasets. I just want you to look some of them over and decide if they might be useful to you in the future.

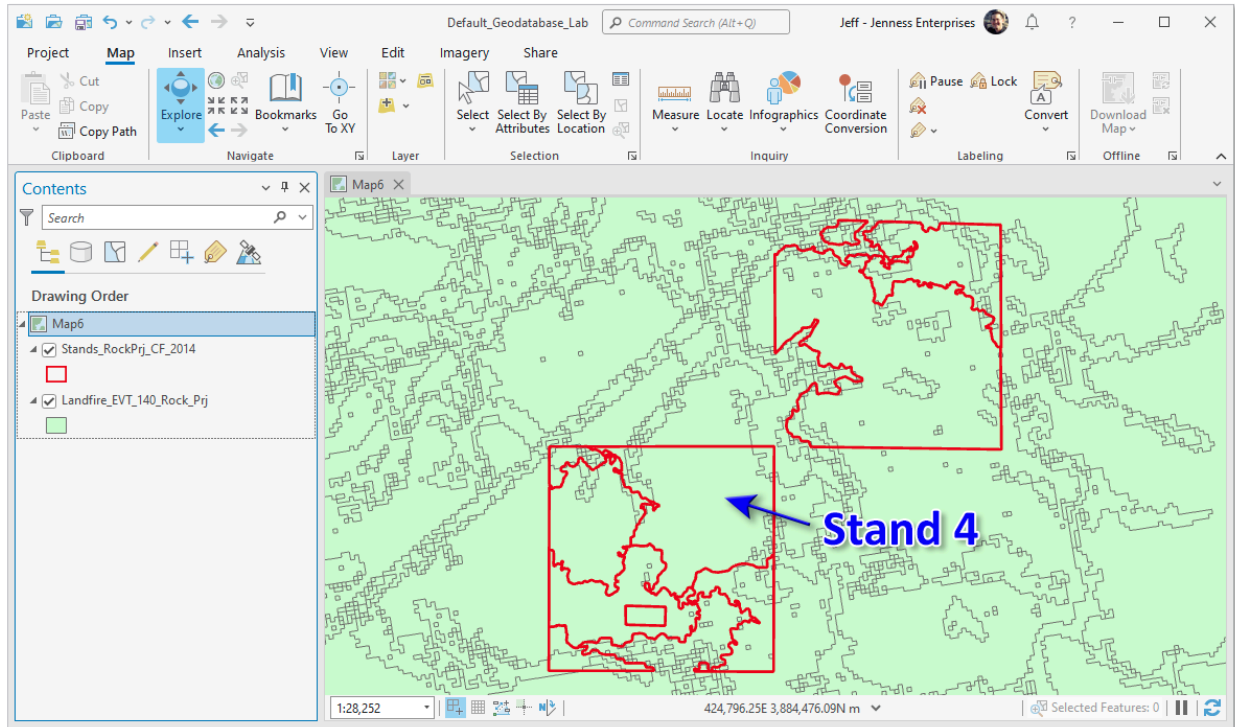
- 1.
- 2.
- 3.

- ☐ For your own information: You can find a lot more detailed description of the vegetation data types by clicking on the appropriate links at <http://www.landfire.gov/vegetation.php>. See especially <https://www.landfire.gov/documents/LF-GAPMapUnitDescriptions.pdf>.

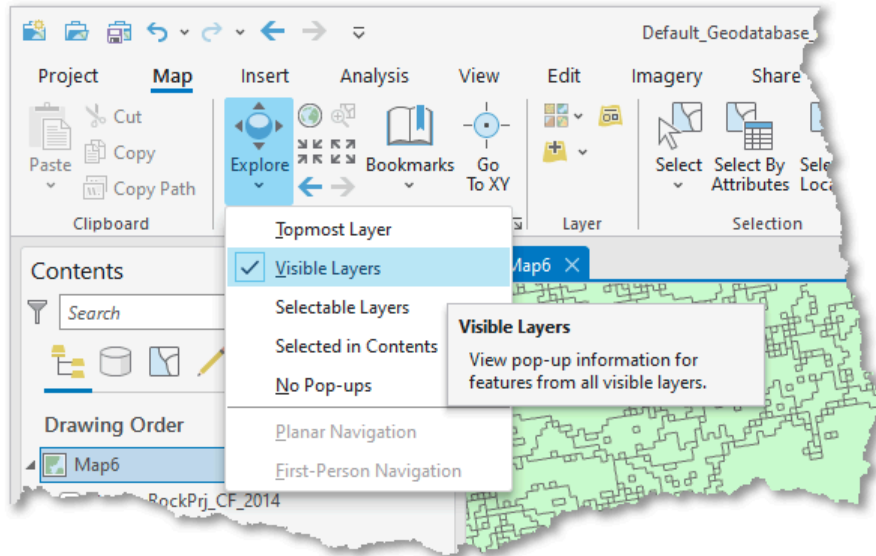
The Actual Lab Exercise

- ☐ Open **ArcGIS Pro** to a new empty map.
- ☐ Add the **Stands_RockPrj_CF_2014** from workspace **..\Class_GIS_Data\RockPrj_CF_2014\..**

- ☐ Add the LandFire "Existing Vegetation Type" feature class **Landfire_EVT_140_Rock_Prj** from workspace **Rock_Project_Landfire.gdb**. **Important!** This is the feature class that has "EVT" in the name!
- ☐ If you wish, add any background data that you would like to see in your map.



- ☐ Use the Explore command  to click on a few Landfire **Landfire_EVT_140_Rock_Prj** polygons and see what they say. **Notice all the different ways each polygon is classified!** These different classification systems mean we can use and analyze Landfire data in multiple ways and at multiple levels of vegetation class precision.
- ☐ **For Example:** Use the Identify command  to click on a the **Landfire_EVT_140_Rock_Prj** polygon underneath Rock Project Stand 4 (see above). **Note:** In the Explore tool drop-down menu, you may have to switch the current choice from "Topmost layer" to one of the other options:



- Now, when clicking on Stand 4, you should see the following classifications for the Landfire polygon underneath Stand 4:

Pop-up

- Stands_RockPrj_CF_2014 (1)
 - 4
- Landfire_EVT_140_Rock_Prj (1)
 - Southern Rocky Mountain Ponderosa Pine Woodland**

Landfire_EVT_140_Rock_Prj - Southern Rocky Mountain Po...

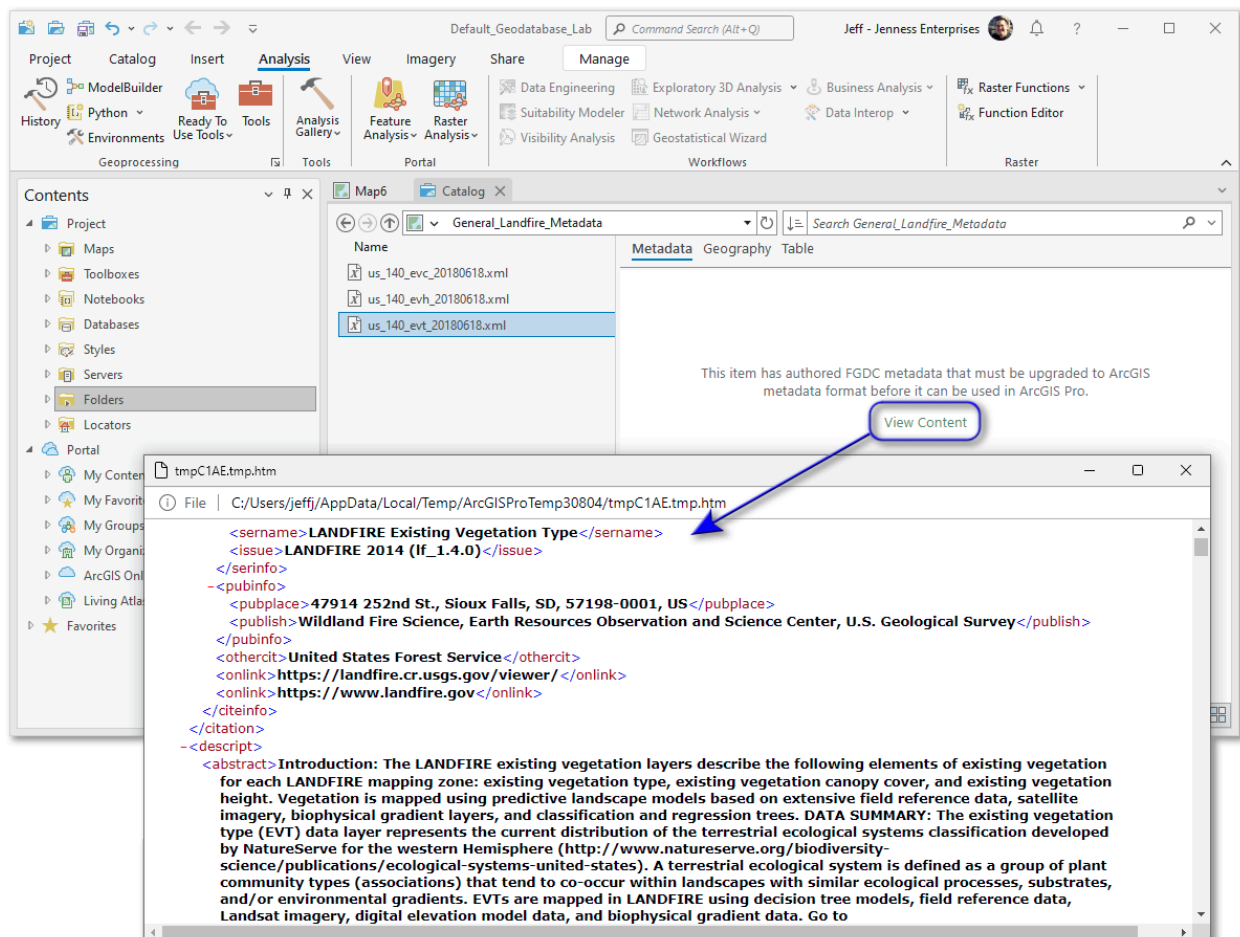
object identifier	4384
VALUE	3054
CLASSNAME	Southern Rocky Mountain Ponderosa Pine Woodland
EVT_Fuel	2054
EVT_Fuel_N	Tr Southern Rocky Mountain Ponderosa Pine Woodland
EVT_LF	Tree
EVT_GP	631
EVT_PHYS	Conifer
EVT_GP_N	Ponderosa Pine Forest, Woodland and Savanna
SAF_SRM	SAF 237: Interior Ponderosa Pine
EVT_ORDER	Tree-dominated
EVT_CLASS	Open tree canopy
EVT_SBCLS	Evergreen open tree canopy
r	214
g	255
b	181
Red	0.839216
Green	1
Blue	0.709804
Shape_Length	52510.262621
Shape_Area	6917282.11654

◀ 1 of 1 ▶ 426,616.90E 3,882,312.26N m

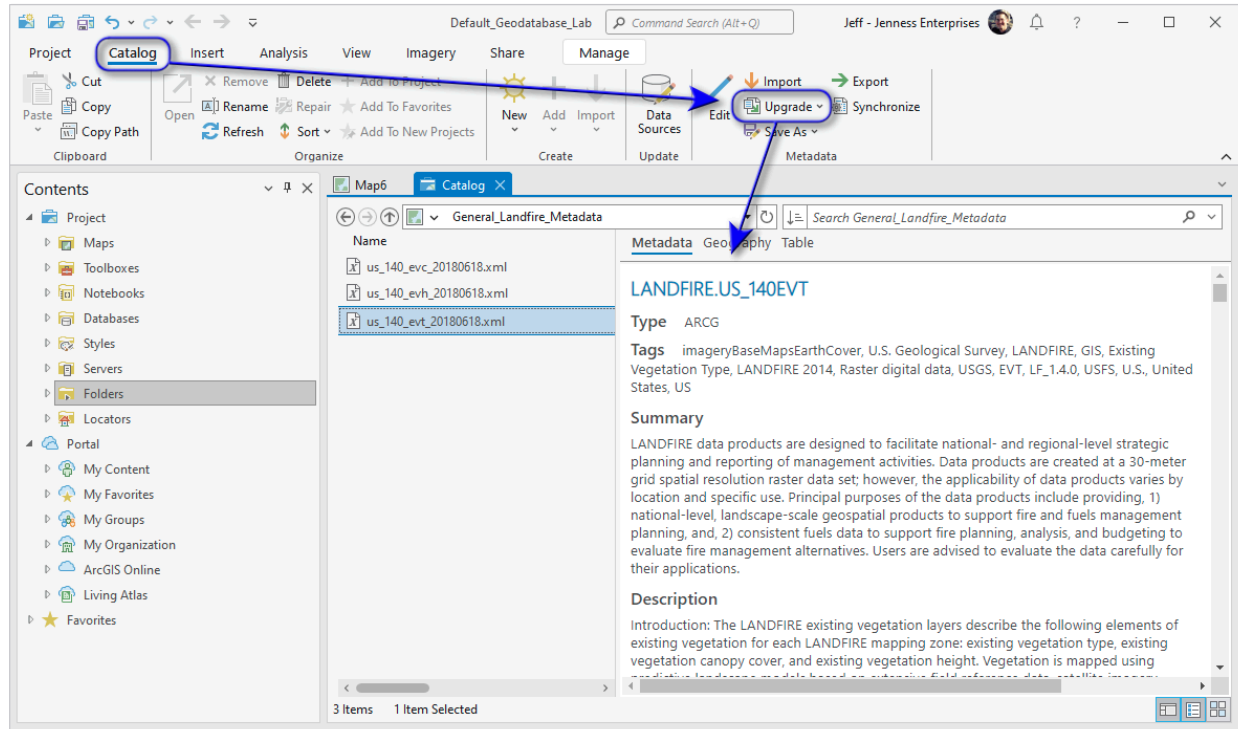
- **Southern Rocky Mountain Ponderosa Pine Woodland**, according to the attribute field "CLASSNAME".
- **Tr Southern Rocky Mountain Ponderosa Pine Woodland**, according to the attribute field "EVT_FUEL_N".
- **Tree**, according to the attribute field "EVT_LF"
- **Conifer**, according to the attribute field "EVT_PHYS"
- **Ponderosa Pine Forest, Woodland and Savanna**, according to the attribute field "EVT_GP_N"

- **SAF 237: Interior Ponderosa Pine**, according to the attribute field “SAF_SRM” (**Note:** These classifications come from the Society of American Foresters and the Society for Range Management).
- **Tree-dominated**, according to the attribute field “EVT_ORDER”
- **Open tree canopy**, according to the attribute field “EVT_CLASS”
- **Evergreen open tree canopy**, according to the attribute field “EVT_SBCLS”

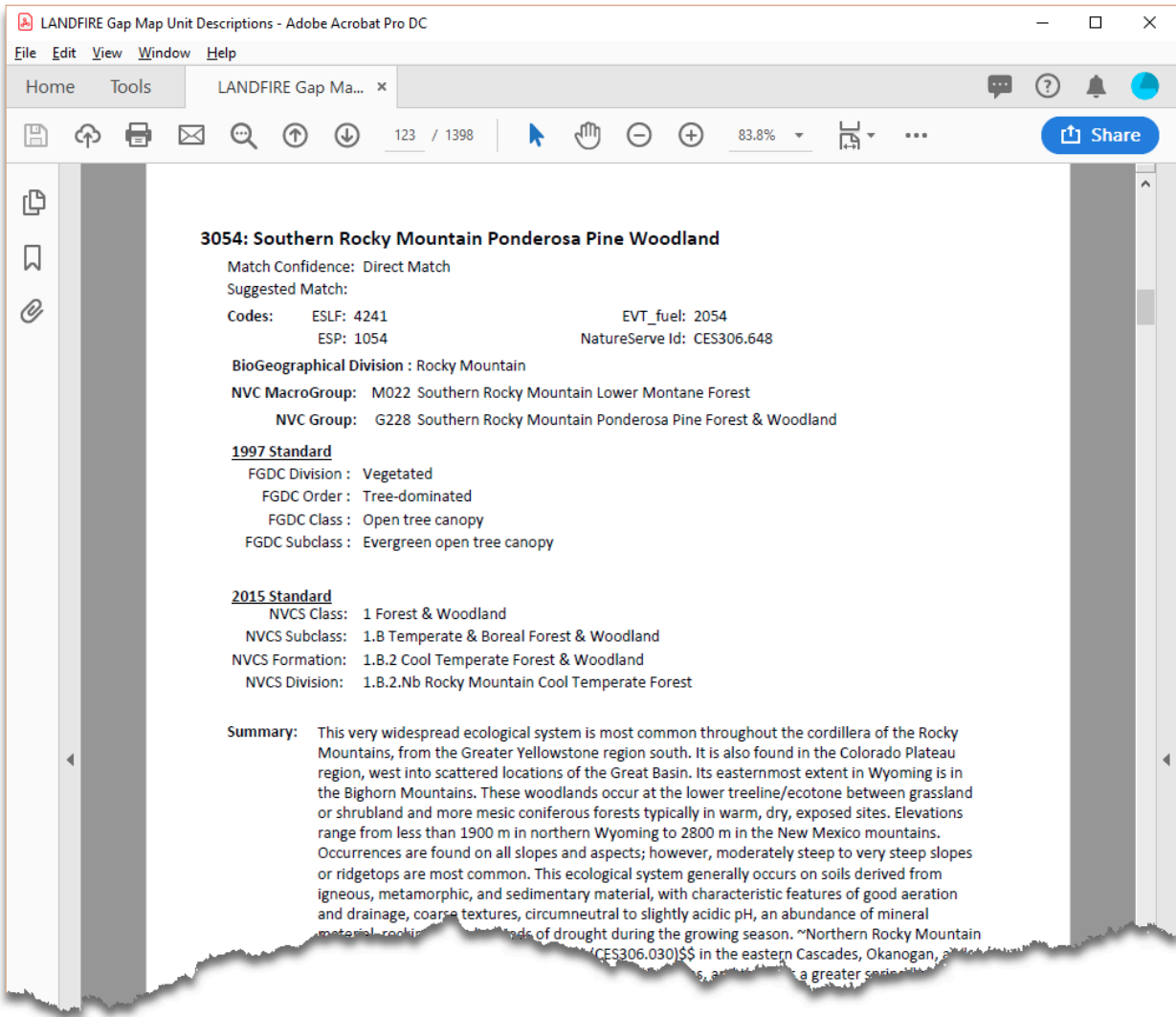
Note: You may notice that the Landfire feature class does **not** have good metadata describing what all these attribute fields are. If you are curious (and you should be if you ever think you might want to use this data), you can find brief descriptions of them in the folder “General_Landfire_Metadata” (located in your folder `..\Class_GIS_Data\RockPrj_CF_2014\..`). Use your Catalog view to find the file, then click on the “Description” tab to see the metadata:



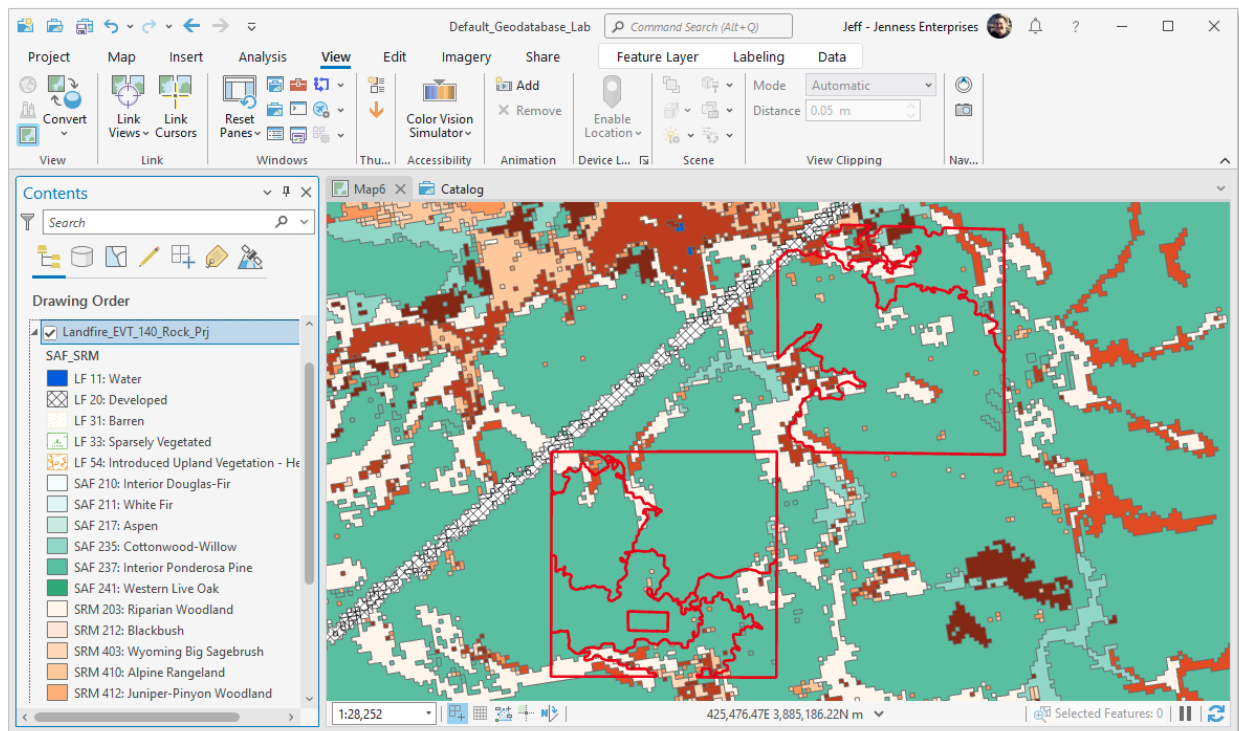
Tip: Since ArcGIS Pro is unable to view the metadata in its native format, you can use the “Upgrade” tool in the Metadata tab to upgrade it. Your choice.



There is a lot more information available on the links at <https://www.landfire.gov/documents/LF-GAPMapUnitDescriptions.pdf>, but it takes some searching to find good descriptions for each classification type. To get info on **Southern Rocky Mountain Ponderosa Pine Woodland**, you would search the PDF for this value (or, more simply, for the corresponding Code value 3054):



- ☐ Try symbolizing by the attribute field **SAF_SRM**. Try playing around some with the symbols for specific categories. I tried symbolizing the tree classes by ramping through shades of green, and the grass/shrub classes through shades of red, and then I set the border width for all classes to be 0. **Note:** You don't have to try to duplicate the image below! Just make something that looks useful to you.

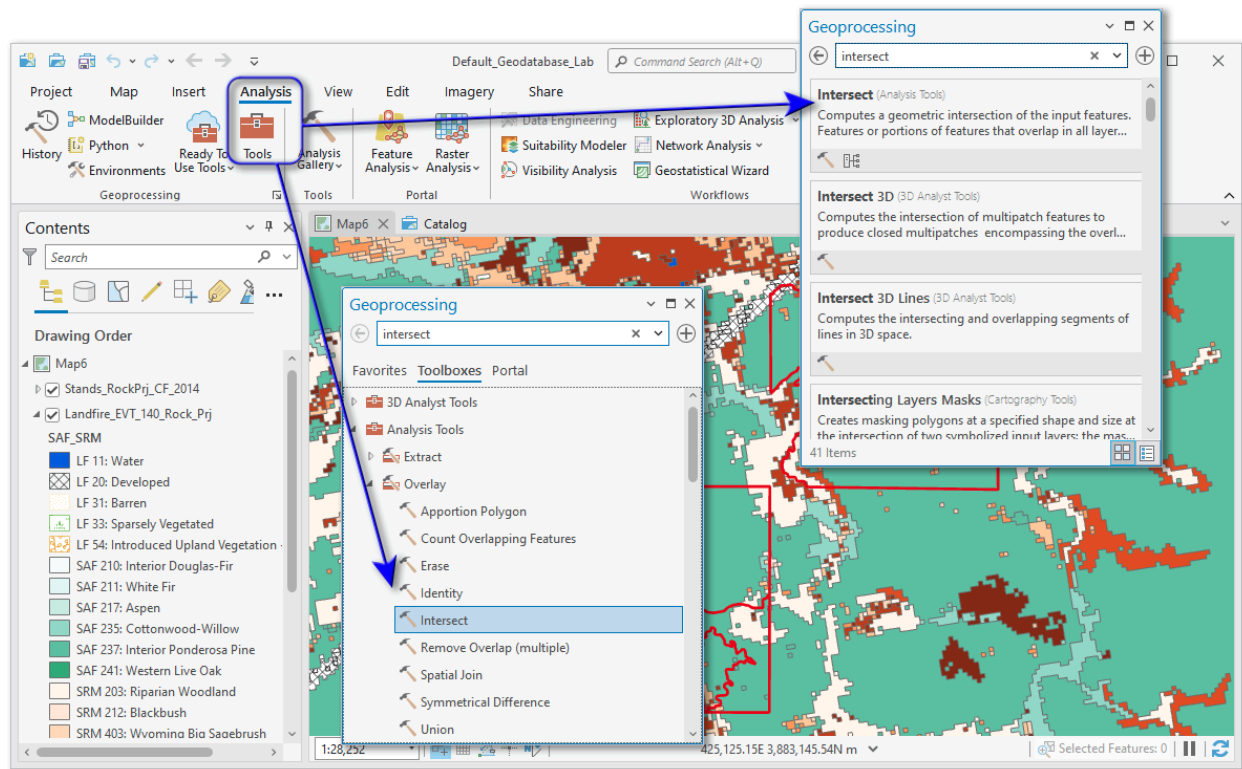


- ❑ What about if you want to just look at Landfire classes inside your project stand boundaries?
You have two options that I can think of:

- You could select all Landfire polygons that intersect the stands, then just run the statistics tool on them. **However**, this will give you a lot of information on areas outside your project area, because the Landfire polygons do **not** line up nicely with your stand boundaries.
- **The better solution** is to run the geoprocessing **intersect** tool to intersect your stand boundaries with the Landfire polygons. This essentially cuts all the Landfire polygons to your stand boundary edges.

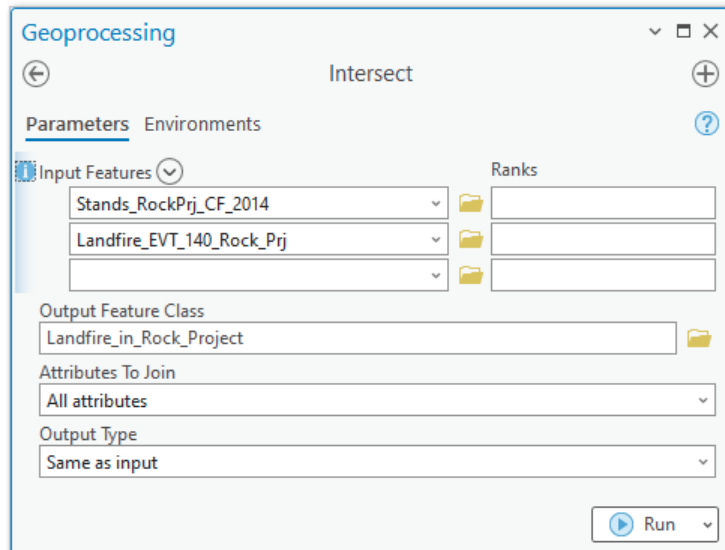
Run the Intersect Tool:

- ❑ Open your **Intersect** tool by clicking the **Analysis** tab, and **Tools**, and then choosing **Intersect**:
- You can find the **Intersect** tool by either searching for it directly, or by open up the **Toolboxes**, then **Analysis Tools**, then **Overlay**, then **Intersect**.

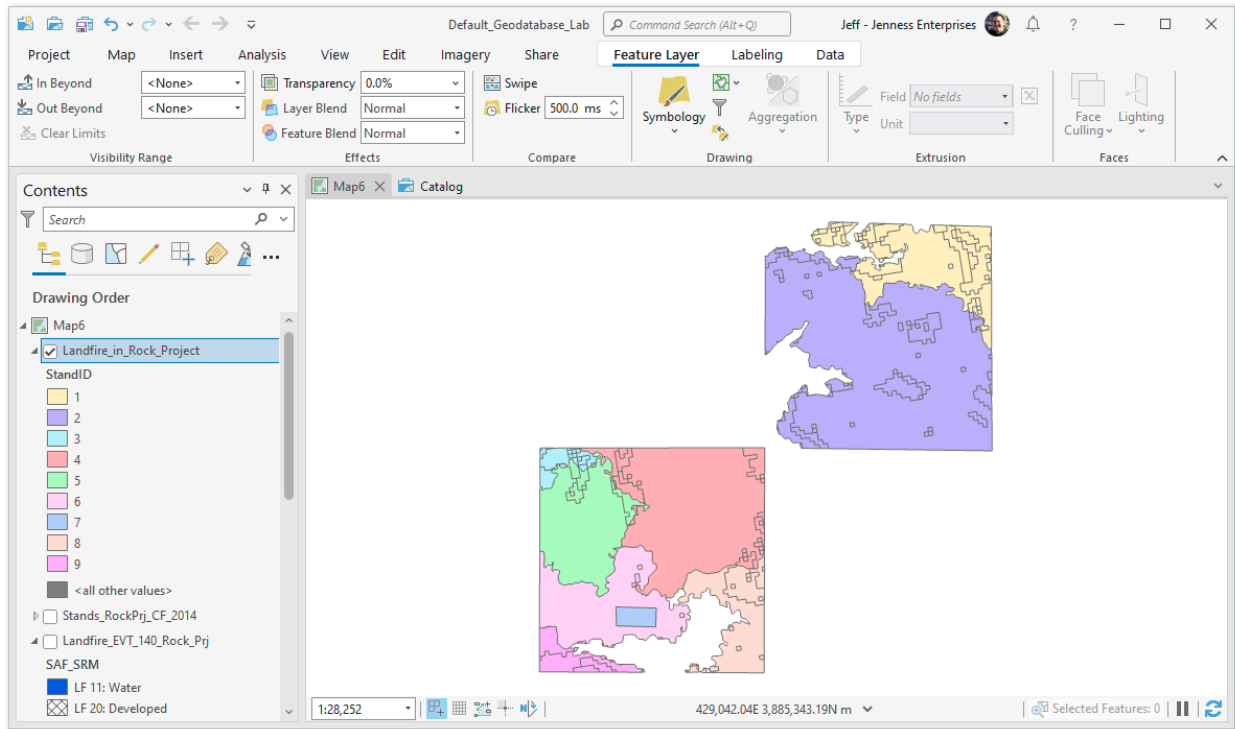


❑ Fill out the Intersect parameters as follows:

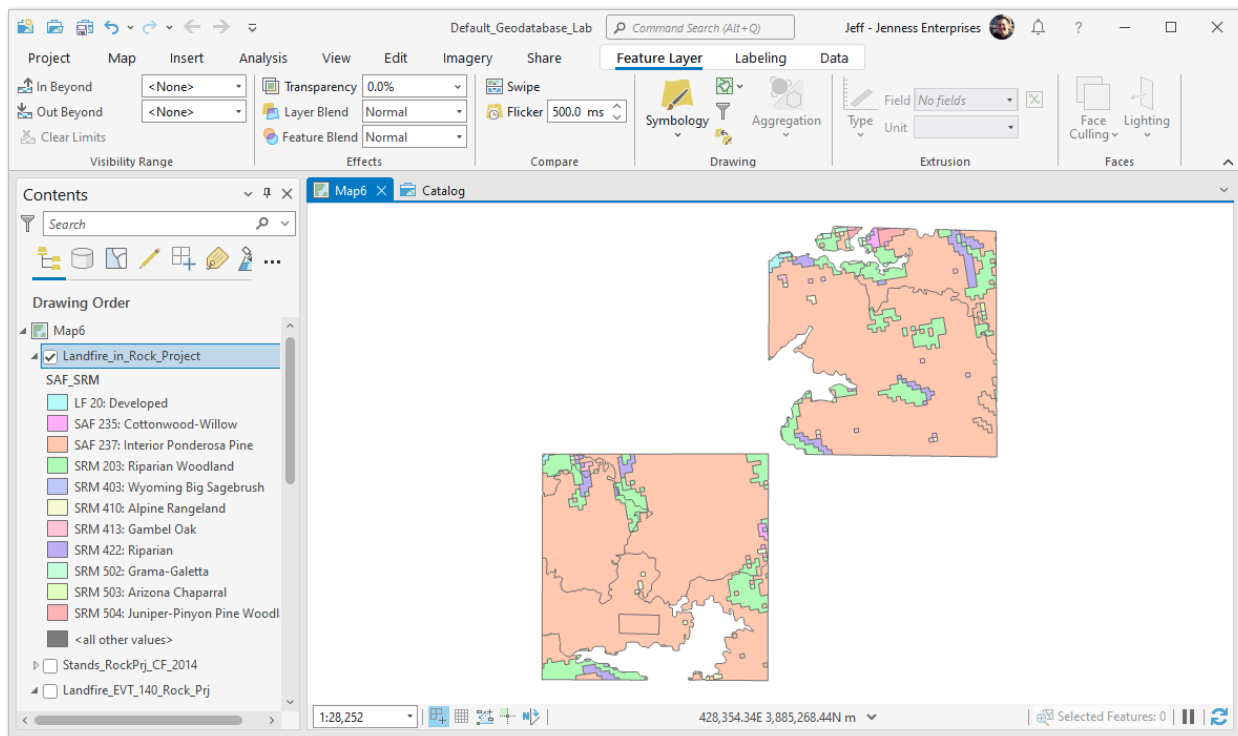
- For **Input Features**, add your **Standings_RockPrj_CF_2014** and **Landfire_EVT_140_Rock_Prj** feature classes.
- For **Output Feature Class**: Just save it somewhere you will remember. **You will not have to turn this feature class in!** Just remember the location and name so you can find it later if you need to.
- For **Attributes to Join**, select **ALL**
- For **Output Type**, set it as **Same as Input**.



- ☐ Hit Run and wait for it to finish. The tool **should** add the new feature class automatically to your map. If it does not, then just load it manually like you do any other dataset.
- ☐ Notice that your new feature class includes **only** those regions that existed in both **Stands_RockPrj_CF_2014** and **Landfire_EVT_140_Rock_Prj**. Notice also that all polygon boundaries that existed in both original feature classes are included in this new one. This means that the new feature class has all your stand boundaries (and all the stand attributes), plus all the LandFire boundaries and attributes.
- ☐ Try symbolizing by the attribute **StandID** and you will see that all the original stand edges and attributes are there:



- Try symbolizing by the attribute **SAF_SRM** and you will see that all the LandFire classes are there as well.



- ☐ So now we have a vegetation type dataset that is cut to our stand boundaries. We can now easily calculate statistics on it. This example will show how to calculate the acreages of all the SAF_SRM classes in your project area, but you should be able to easily adapt this to any other classification field.

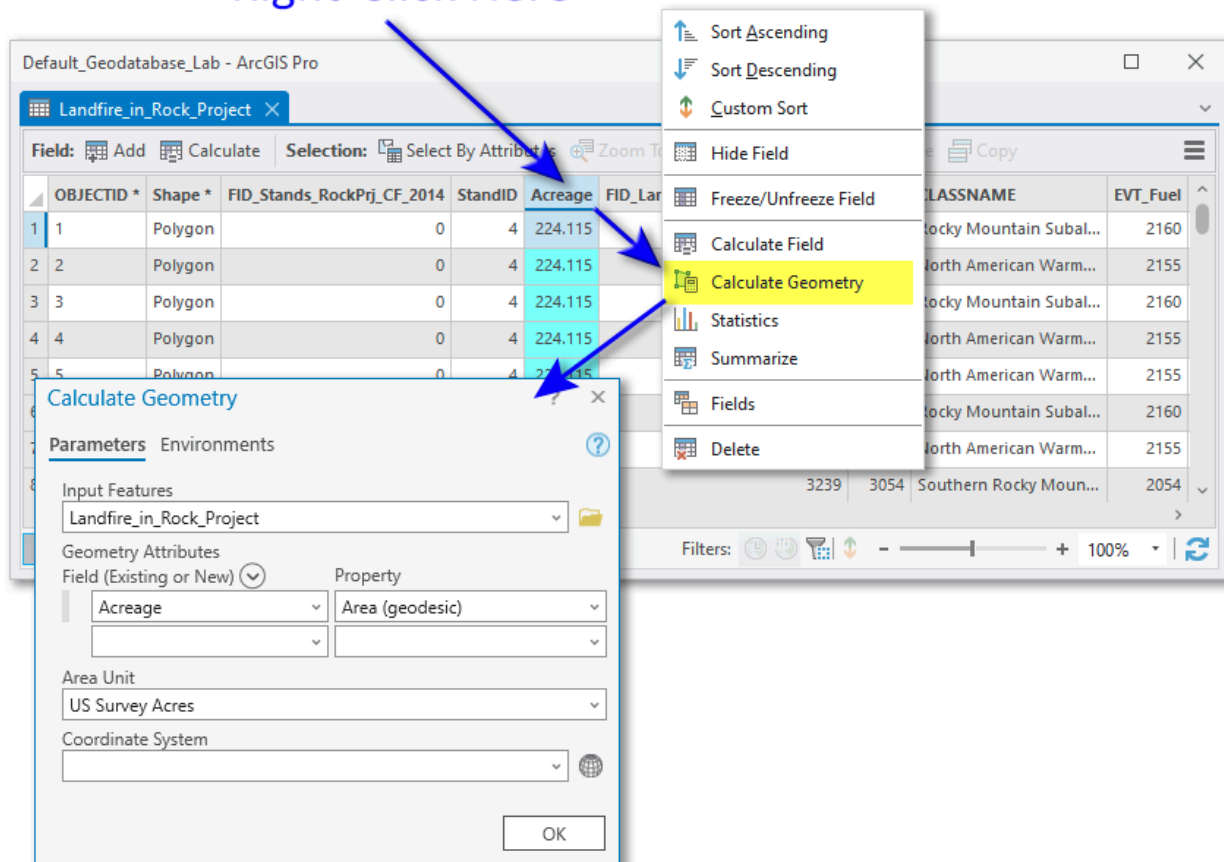
Use Summarize tool to Calculate Acreages of Each SAF_SRM Type in Project Area

- ☐ Open the attribute table for your new feature class and find the attribute field **Acreage**.

	OBJECTID *	Shape *	FID_Stands_RockPrj_CF_2014	StandID	Acreage	FID_Landfire_EVT_140_Rock_Prj	VALUE	CLASSNAME	EVT_Fuel
1	1	Polygon	0	4	224.115	3057	3160	Rocky Mountain Subal...	2160
2	2	Polygon	0	4	224.115	3113	3155	North American Warm...	2155
3	3	Polygon	0	4	224.115	3154	3160	Rocky Mountain Subal...	2160
4	4	Polygon	0	4	224.115	3177	3155	North American Warm...	2155
5	5	Polygon	0	4	224.115	3193	3258	North American Warm...	2155
6	6	Polygon	0	4	224.115	3206	3160	Rocky Mountain Subal...	2160
7	7	Polygon	0	4	224.115	3207	3155	North American Warm...	2155
8	8	Polygon	0	4	224.115	3239	3054	Southern Rocky Moun...	2054

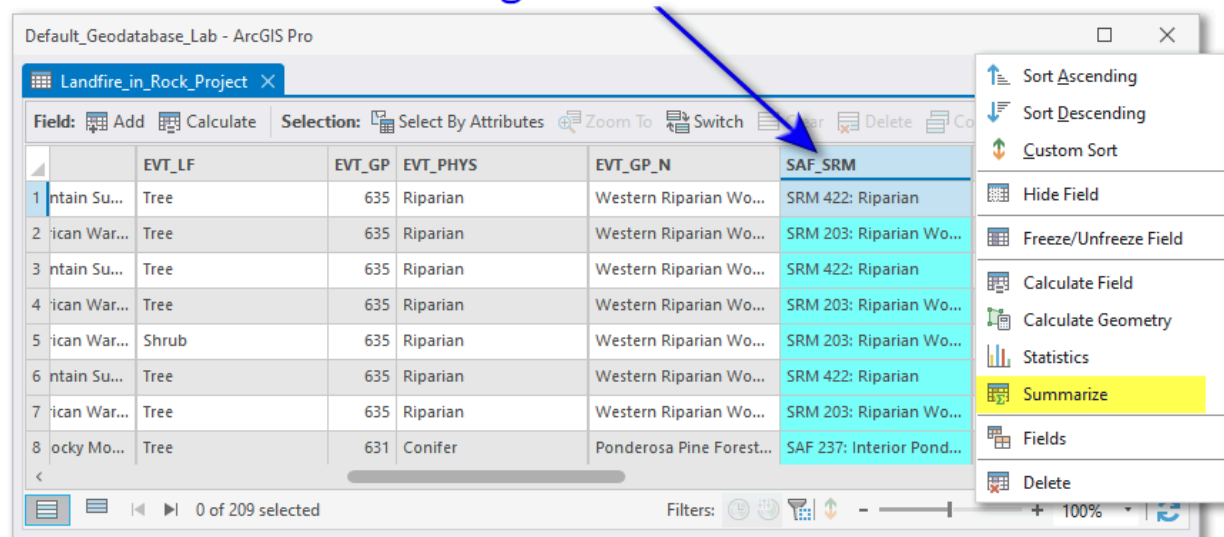
- ☐ Observe that there are already acreage values for all the records. **Question:** Are these values correct? Why or why not?
 - The correct answer is **no**, because they were simply attribute values that got transferred over like all the other attributes. There is nothing special about this attribute field to tell ArcGIS Pro that it reflects **area** values or that it should be automatically changed if you modify the data. ArcGIS sees this attribute field as no different than the others, and it doesn't change these values any more than it changes any of the others.
- ☐ Recalculate the acreages: Right-click on the attribute field and choose "Calculate Geometry", then tell it to calculate **Area (geodesic)** in **US Survey Acres**:

Right-Click Here



- Now the acreages are correct. Next, find the attribute field **SAF_SRM**, right-click on it and choose **Summarize**:

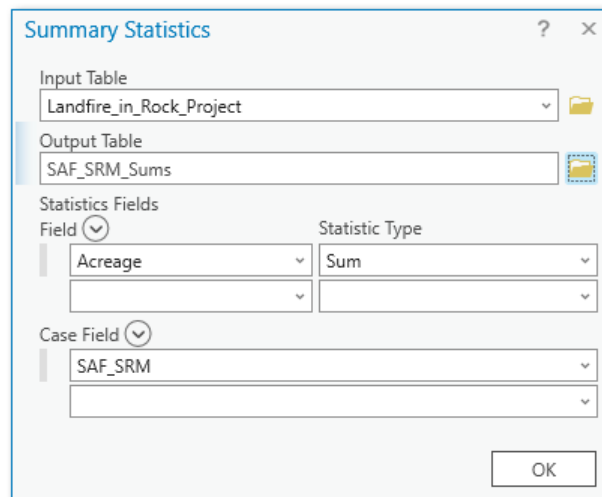
Right-Click Here



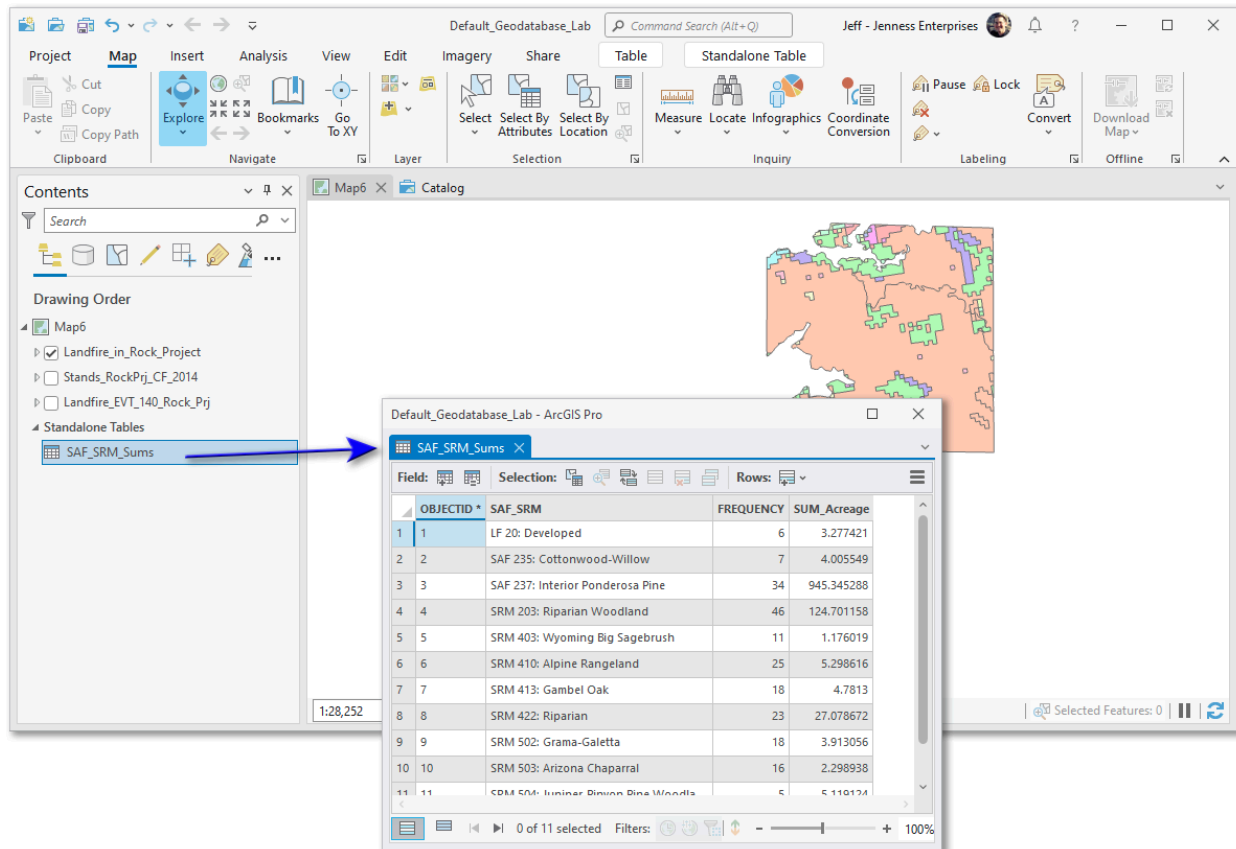
- ❑ **What does Summarize do?** First, it returns a table telling you how many times it found each unique value in the attribute field you are summarizing. Therefore the new Summary table always has an attribute field holding the original values (except there will only be one record per value), plus another attribute field named “Frequency”.

Fortunately, we can also tell the Summarize tool to calculate statistics on other attribute fields. Here it will generate a separate set of statistics for each unique **SAF_SRM** value.

- ❑ We want to calculate the **sum** statistic for the attribute field **Acreage**, so that we will get the total acreage for each **SAF_SRM** class in the project area. In the **Summarize** dialog, simply set the *Field* to be “Acreage”, and the *Statistic Type* to be “Sum”. Leave the *Case Field* as **SAF_SRM**.



- ❑ Specify your output table and remember where you put it! In this example, I am saving the table with the name **SAF_SRM_Sums**.
- ❑ Hit OK and ArcGIS Pro will generate a statistics table and add it to your **Standalone Tables** section. Open your new table and observe that you have a separate row for each unique **SAF_SRM** value, a “Count” field, and a “Sum_Acreage” field”



Optional:

If you would like to classify based on one of the other Landfire classifications, please feel free to do so. I will be happy to help if you run into any problems.

If it would help other Forestry projects, then I have also added the LandFire Existing Vegetation Height (EVH) and Existing Vegetation Cover (EVC) data to the file geodatabase **Rock_Project_Landfire.gdb**. You can use the methods we just ran through to calculate statistics on these other datasets if you like.

Optional II:

If you would like to run a similar Landfire-based analysis anywhere on the Coconino, then I have posted EVT, EVH and EVC datasets on BBLearn in the "Extra Data [Optional]" folder. These datasets include both raster and vector versions of the data. The geodatabase is large (475 MB after extracting it from the zip file), so make sure you put it on a drive that has some space.

Note: The attribute tables are not included with the feature classes, so you will have to join them before you will be able to run analyses like those illustrated above. If you do not remember how to do this, then please refer to Lab Exercise 5 of the *Selections, Joins and Relates* lecture. Again, I will be happy to help if you run into any problems.

Optional III:

The Forest Service GIS web portal has a few vegetation datasets for the Coconino and for Region 3 in general. The site goes down periodically, but you can usually reach the Region 3 GIS site at:

<https://www.fs.usda.gov/detailfull/r3/landmanagement/gis/?cid=stelprdb5201889&width=full>

From there, you can scroll through the site and review the offerings.