Geocoding & Such

GIS 5222

Jake K. Carr

Week 6

Where Do Shapefiles Come From?

The magic Shapefile stork?

Probably Not.

Where Do Shapefiles Come From?

Sometimes we'll be really lucky and someone will make them for us (thanks ESRI and/or Census Bureau).

Other times, we'll have our very own, brand new spatial data. Data that hasn't been 'shapefiled' by ESRI or the Census.

That means we will have to 'digitize' the data ourselves in order to use it in map form.

Where Do Shapefiles Come From?

How we digitize spatial data depends on the geometry type.

Lines and polygons are made up of sequences of points. If there aren't that many line or polygon features in the data, then typically digitization can be done by hand.

Point features can be digitized in one of two ways:

- Using coordinate information in the data (NICE)
- Geocoding

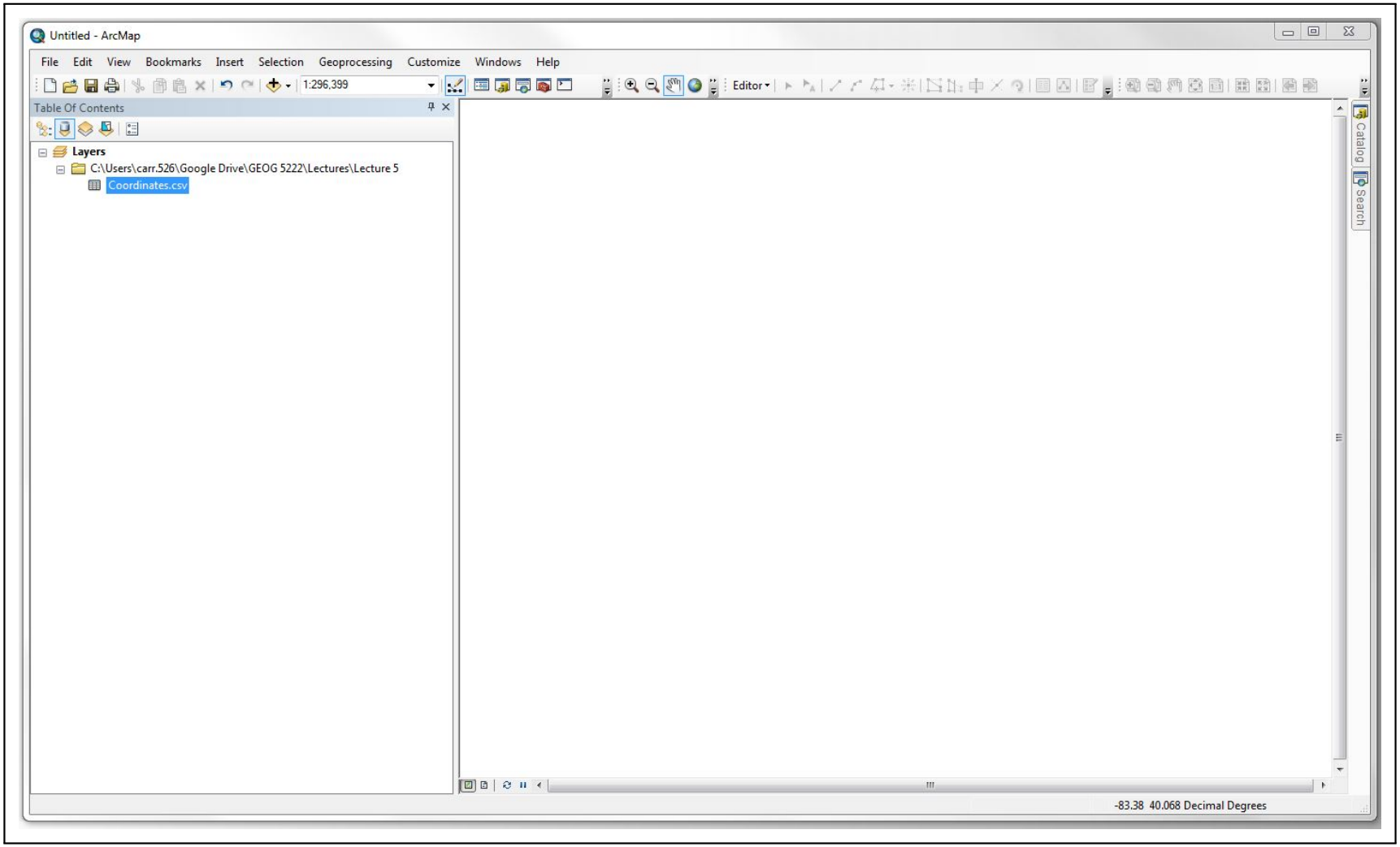
Using Provided Coordinate Information

Suppose someone gives us a table (like **Coordinates.csv**) of point features and wanted us to make a map for them.

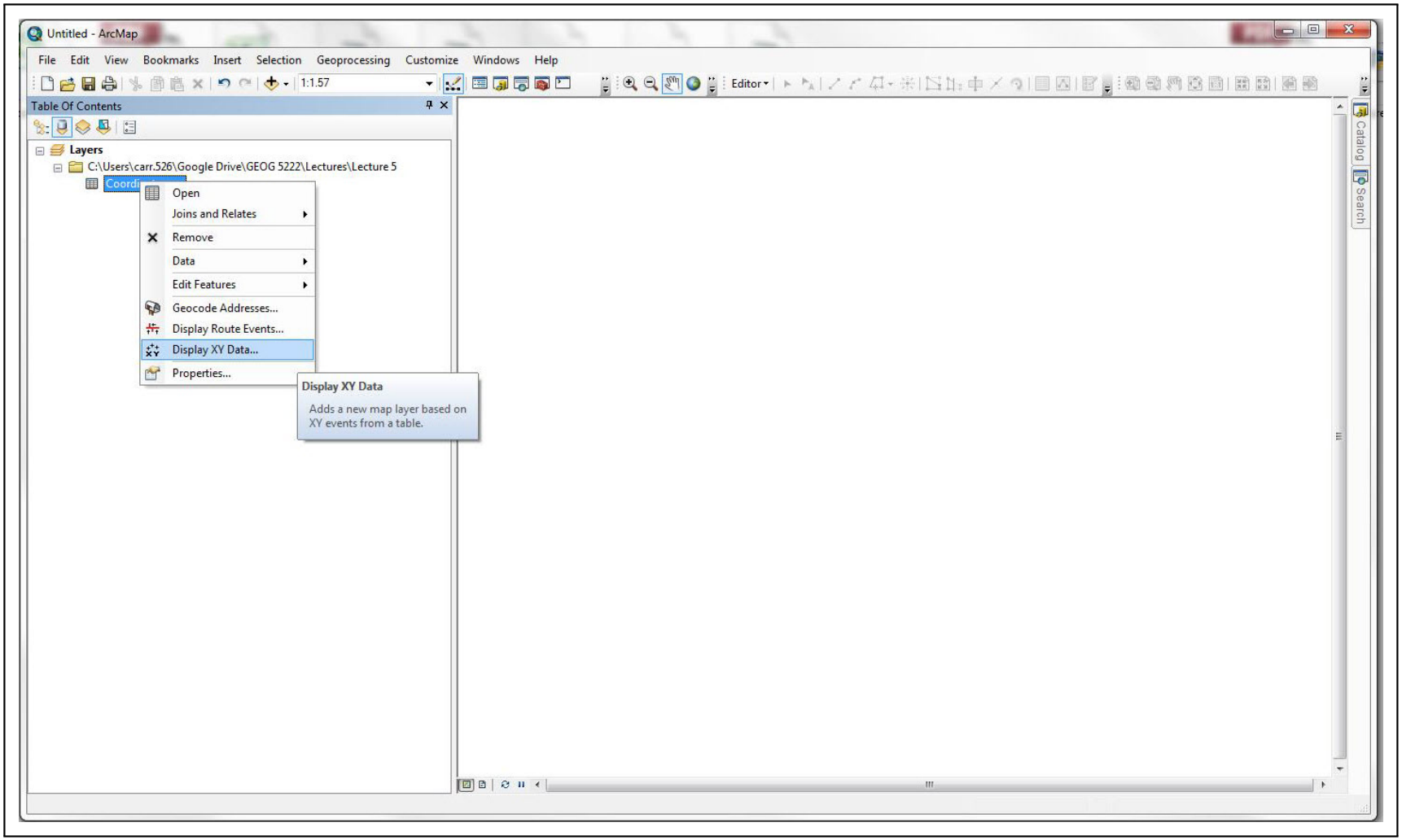
Also assume that the table has fields containing the longitude and latitude of each location.

What do we do?

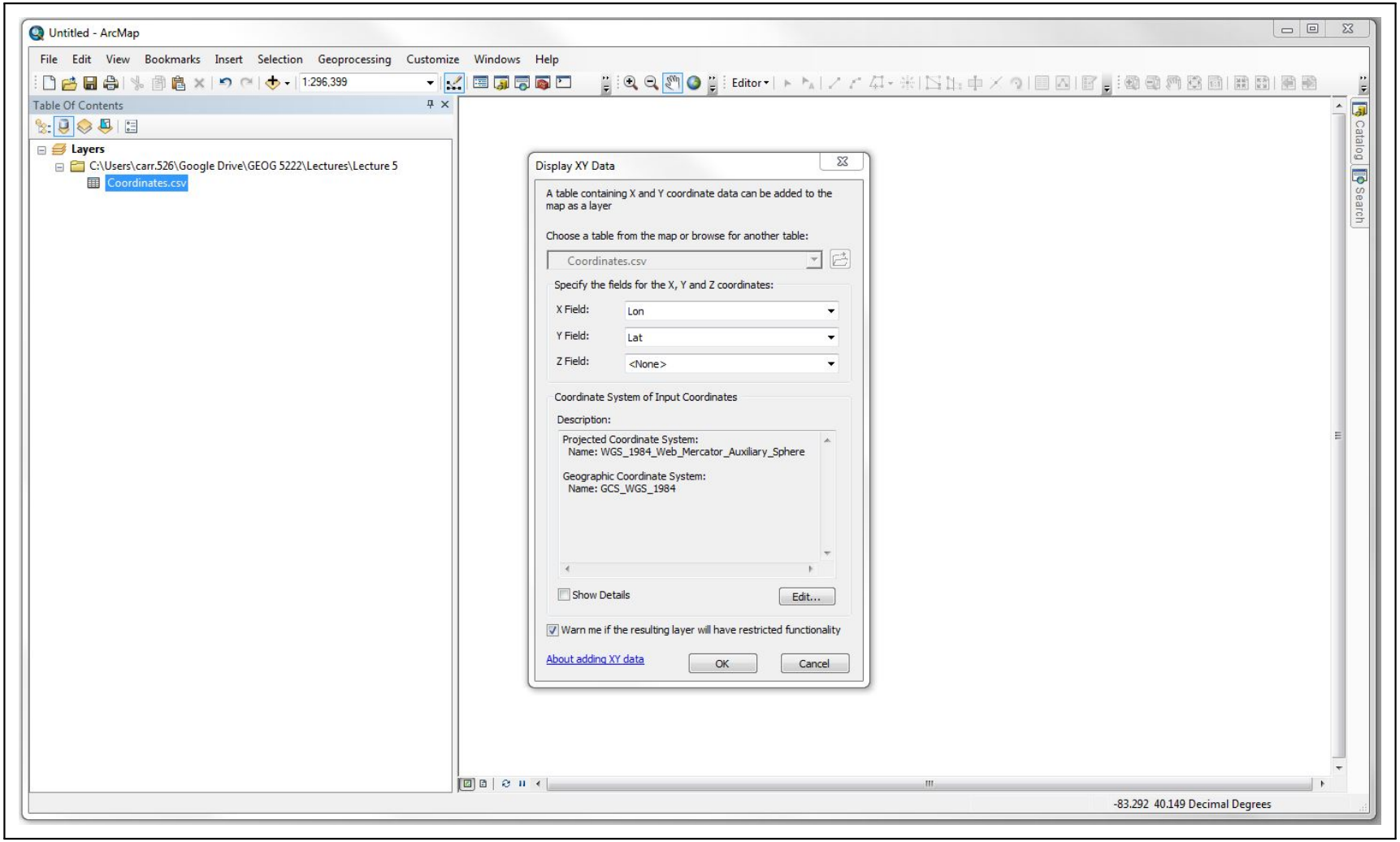
Add Table (**.csv**) to a Map



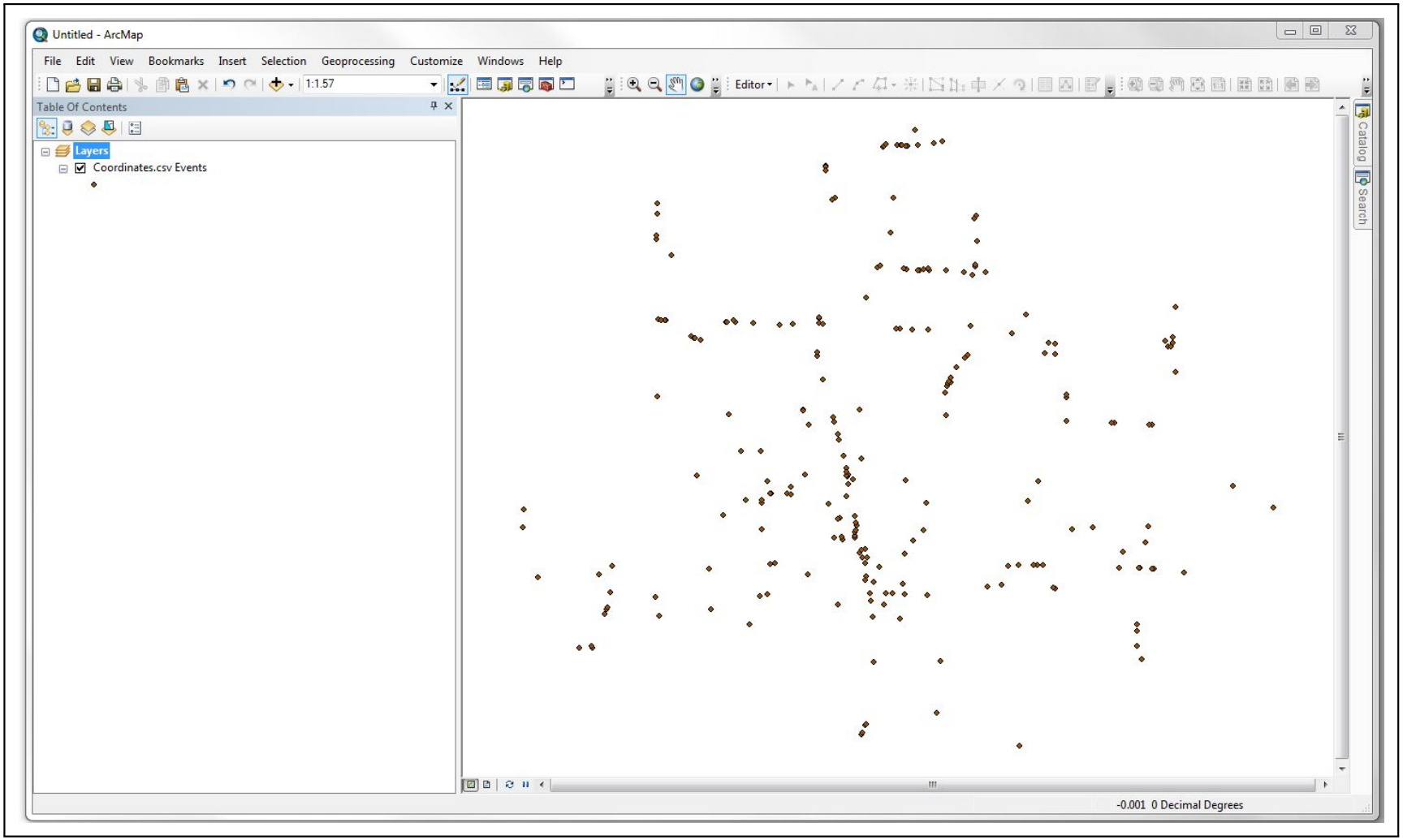
Display XY Data



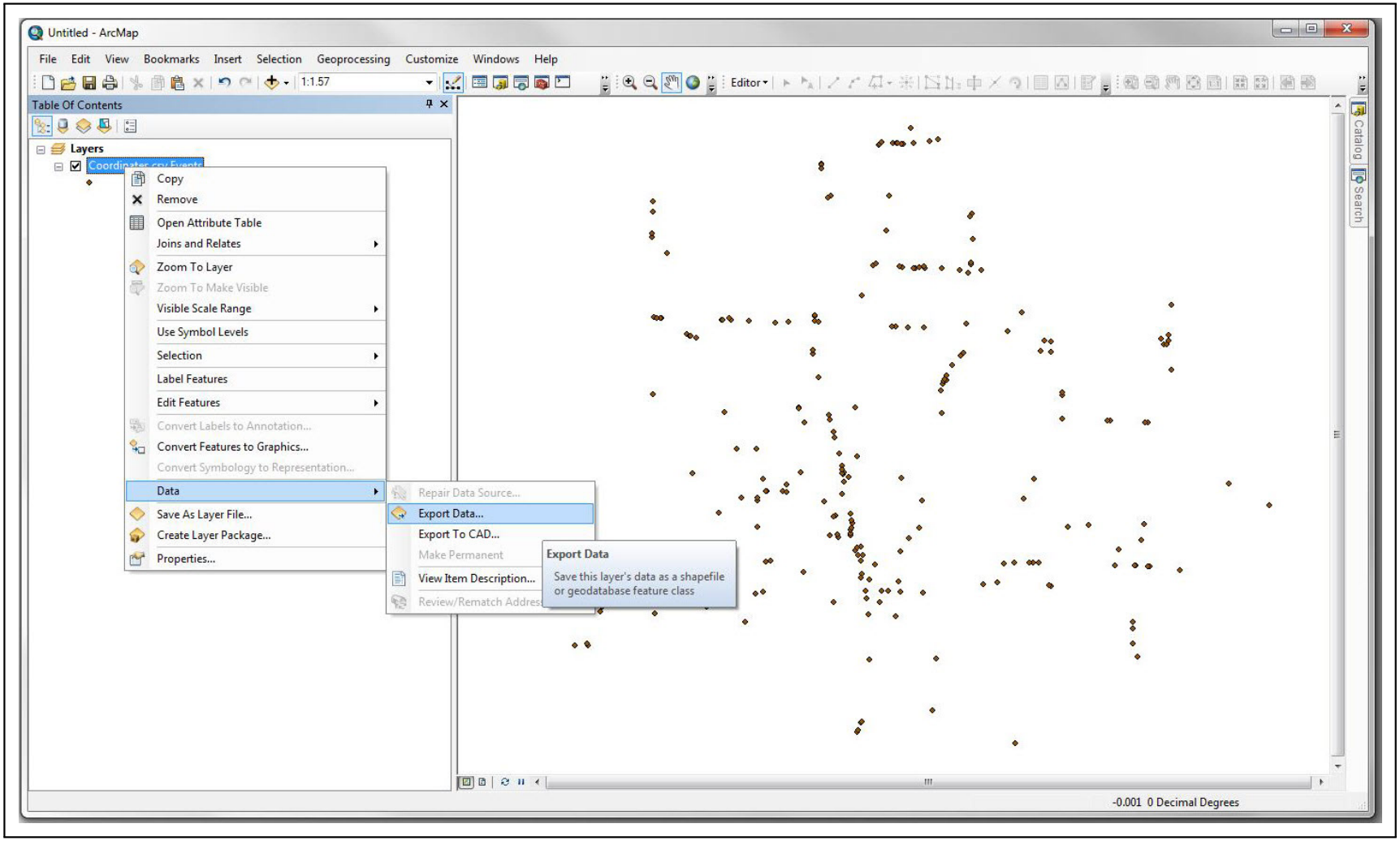
Display XY Data Dialog



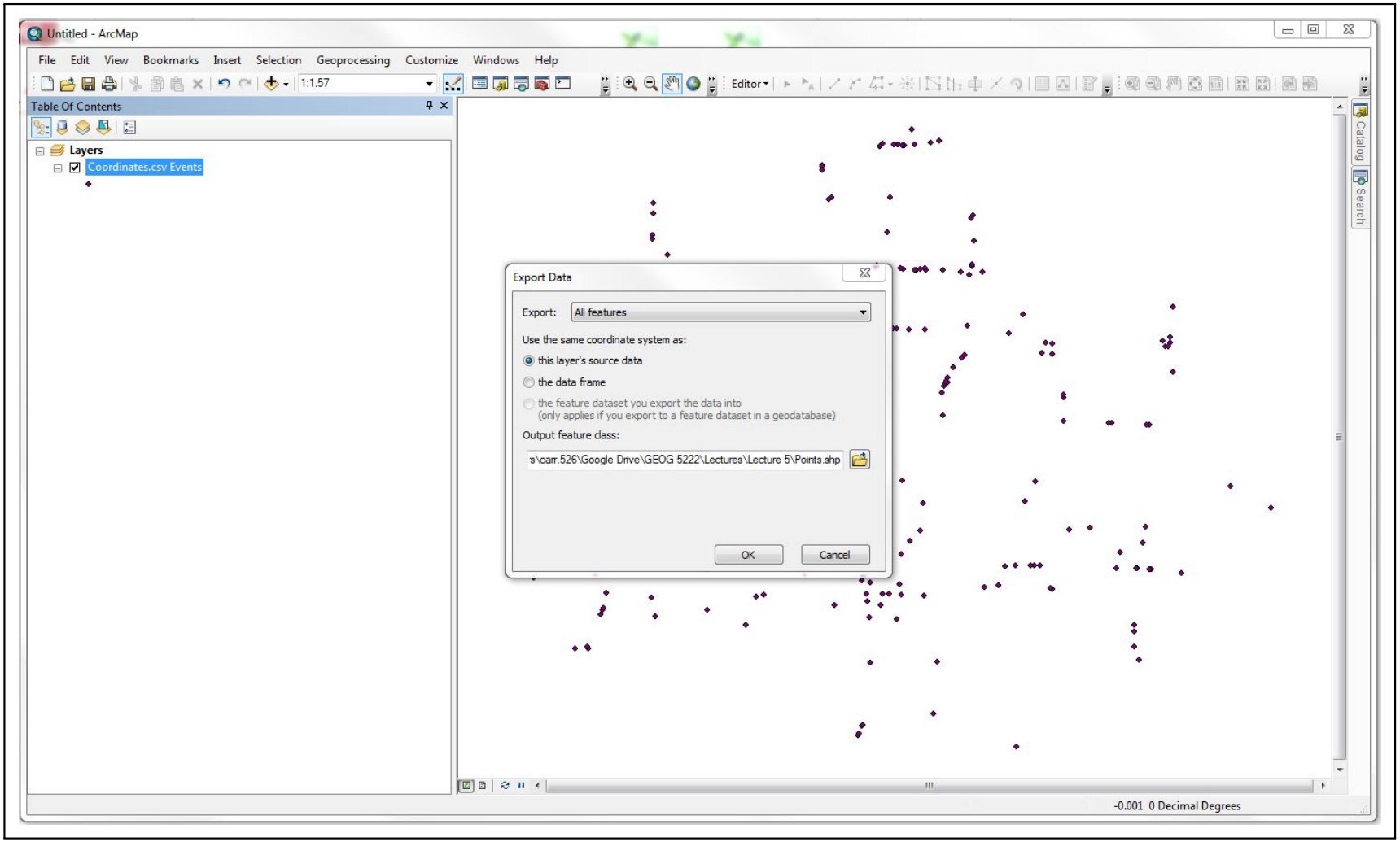
Event Map


Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:1.57
Editor
Table Of Contents
Layers
Coordinates.csv Events
Catalog
Search
-0.001 0 Decimal Degrees

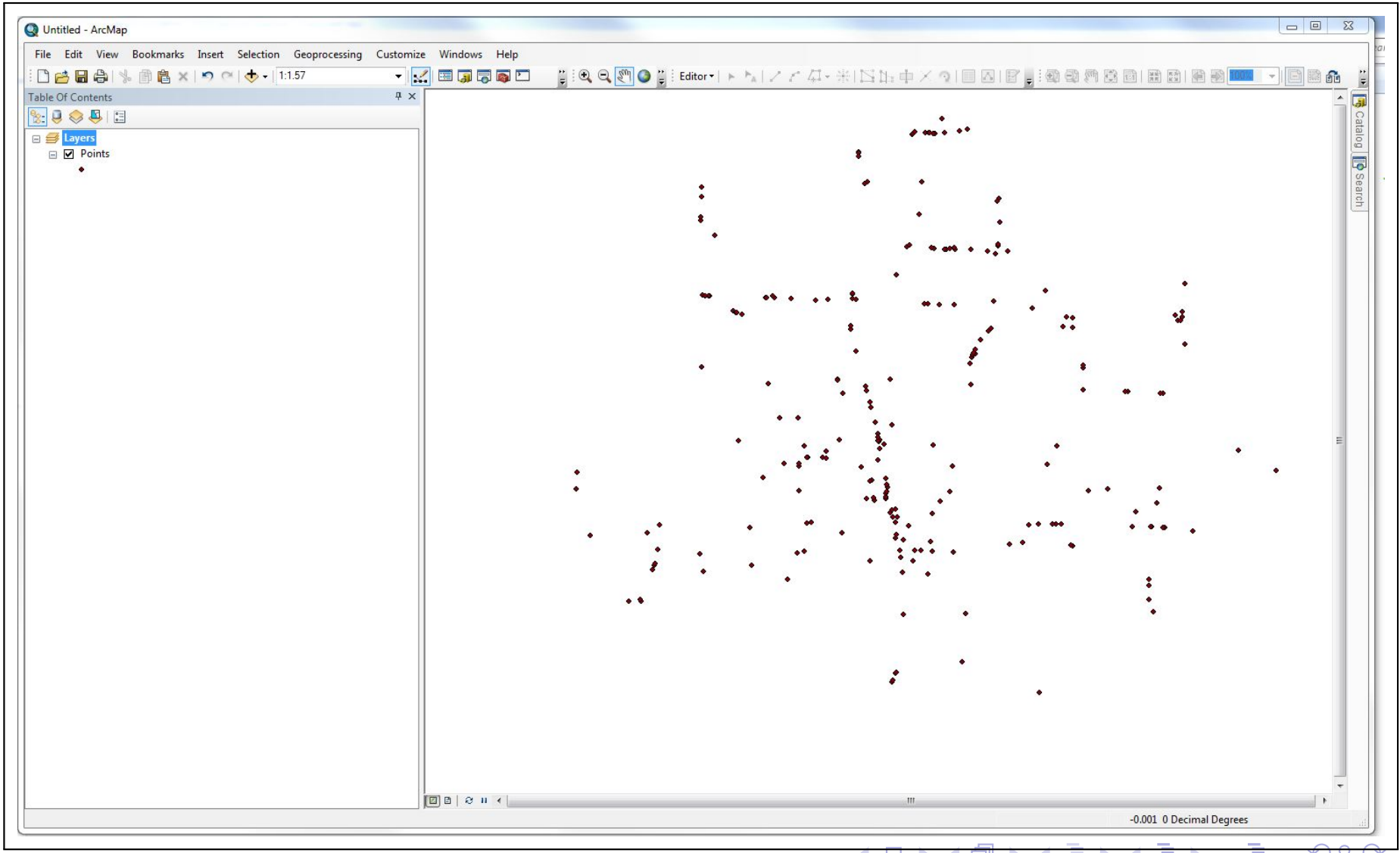
Export Data


Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:1.57
Editor
Table Of Contents
Layers
Copy
Remove
Open Attribute Table
Joins and Relates
Zoom To Layer
Zoom To Make Visible
Visible Scale Range
Use Symbol Levels
Selection
Label Features
Edit Features
Convert Labels to Annotation...
Convert Features to Graphics...
Convert Symbology to Representation...
Data
Save As Layer File...
Create Layer Package...
Properties...
Repair Data Source...
Export Data...
Export To CAD...
Make Permanent
View Item Description...
Review/Rematch Addres
Export Data
Save this layer's data as a shapefile or geodatabase feature class
Catalog
Search
-0.001 0 Decimal Degrees

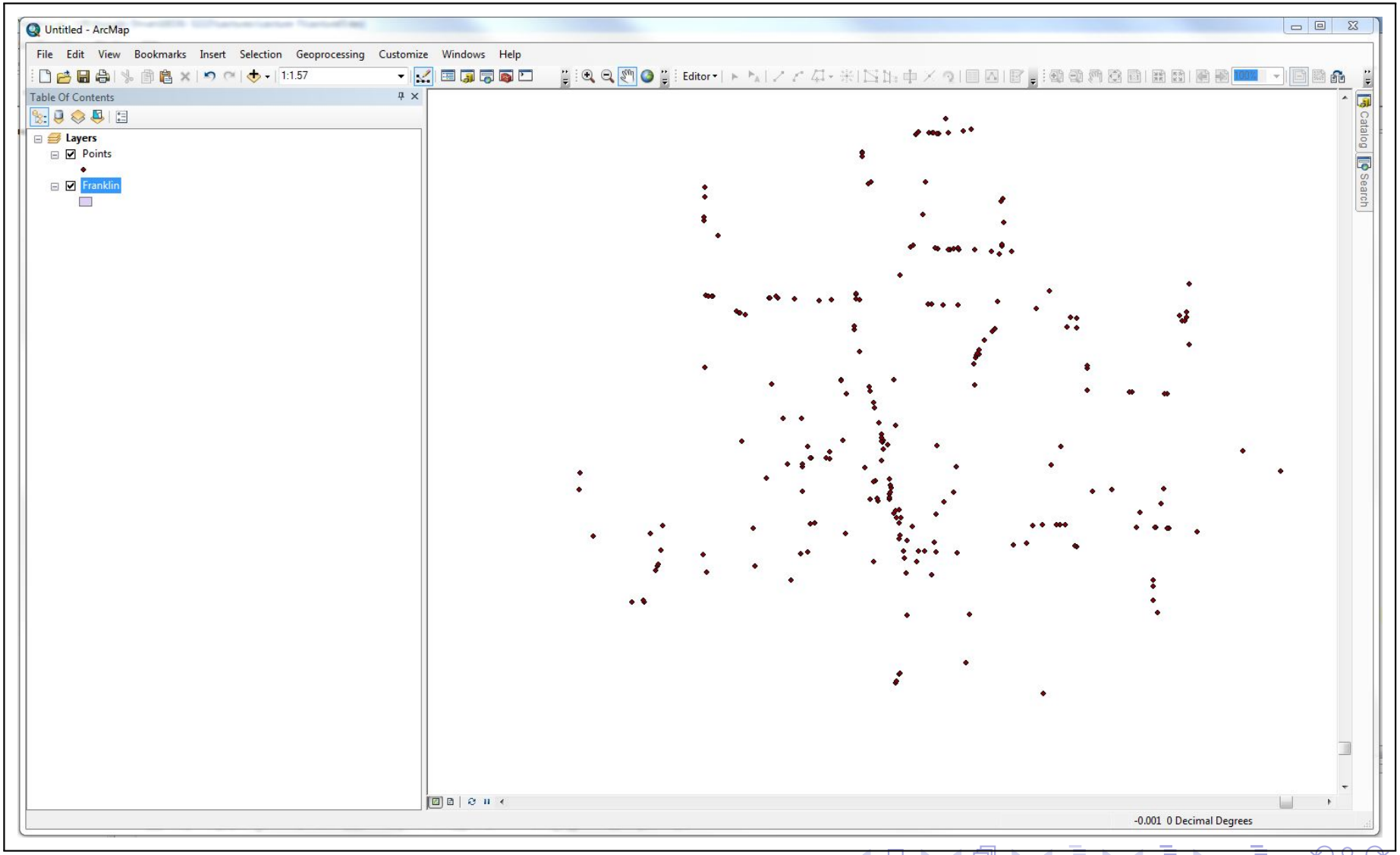
Export Data Dialog



Columbus Restaurants: **Points.shp**



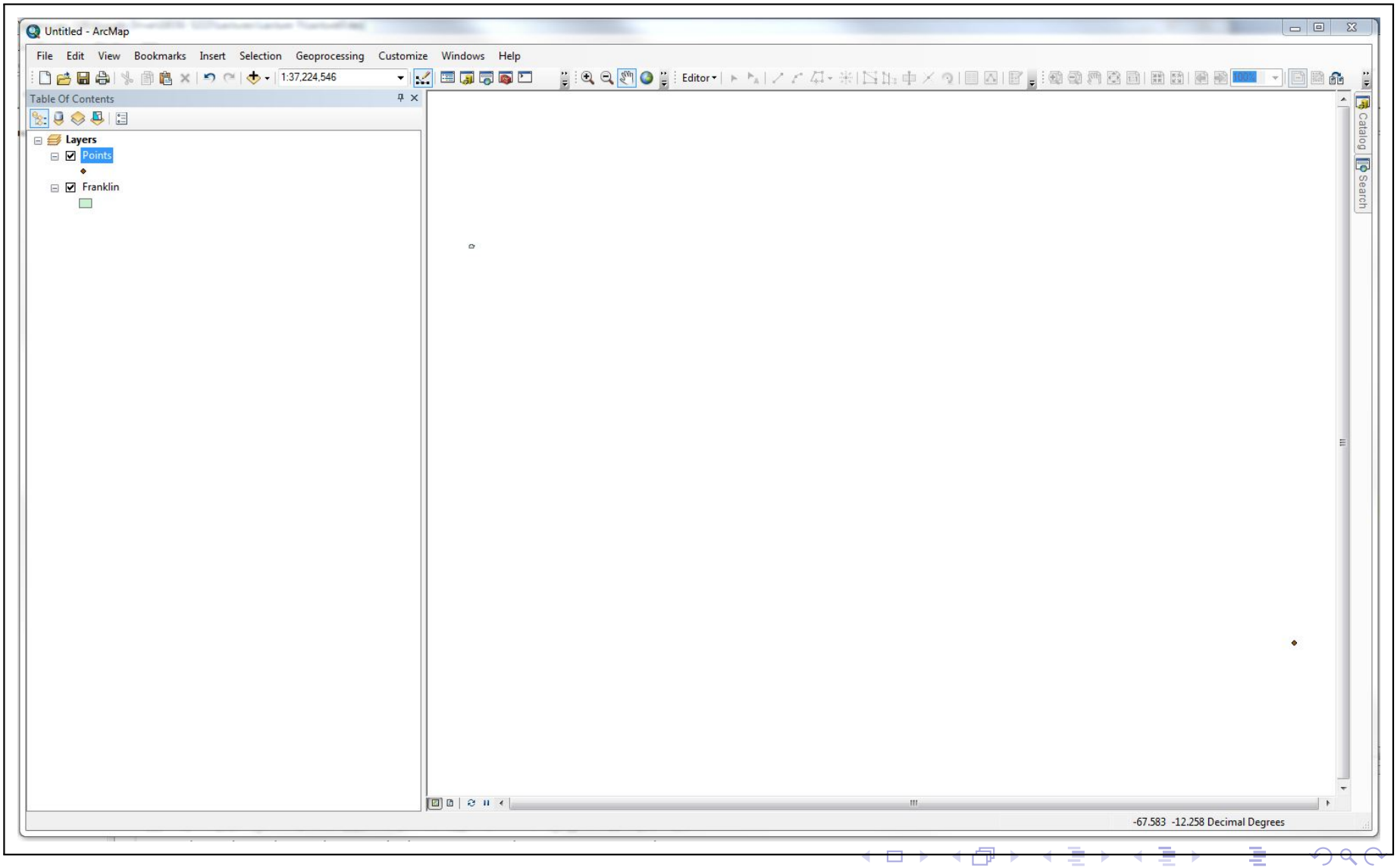
Add **Franklin.shp**



What Happened?

Where is the Franklin County polygon?

There it is


Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:37,224,546
Editor
Table Of Contents
Layers
Points
Franklin
Catalog
Search
-67.583 -12.258 Decimal Degrees

Why did that Happen?

That's right, the two shapefiles have different coordinate systems (and/or projections).

How do I know that?

Hint: Look at their properties.

What do we do about it?

There are two solutions[1]:

- Python (obvious)
- Mess with the **.prj** file

Actually, there are other solutions to this problem. We'll ignore those alternatives for a good reason (which you'll soon see).

[1] Because of course there are two solutions

Mess with the **.prj** file

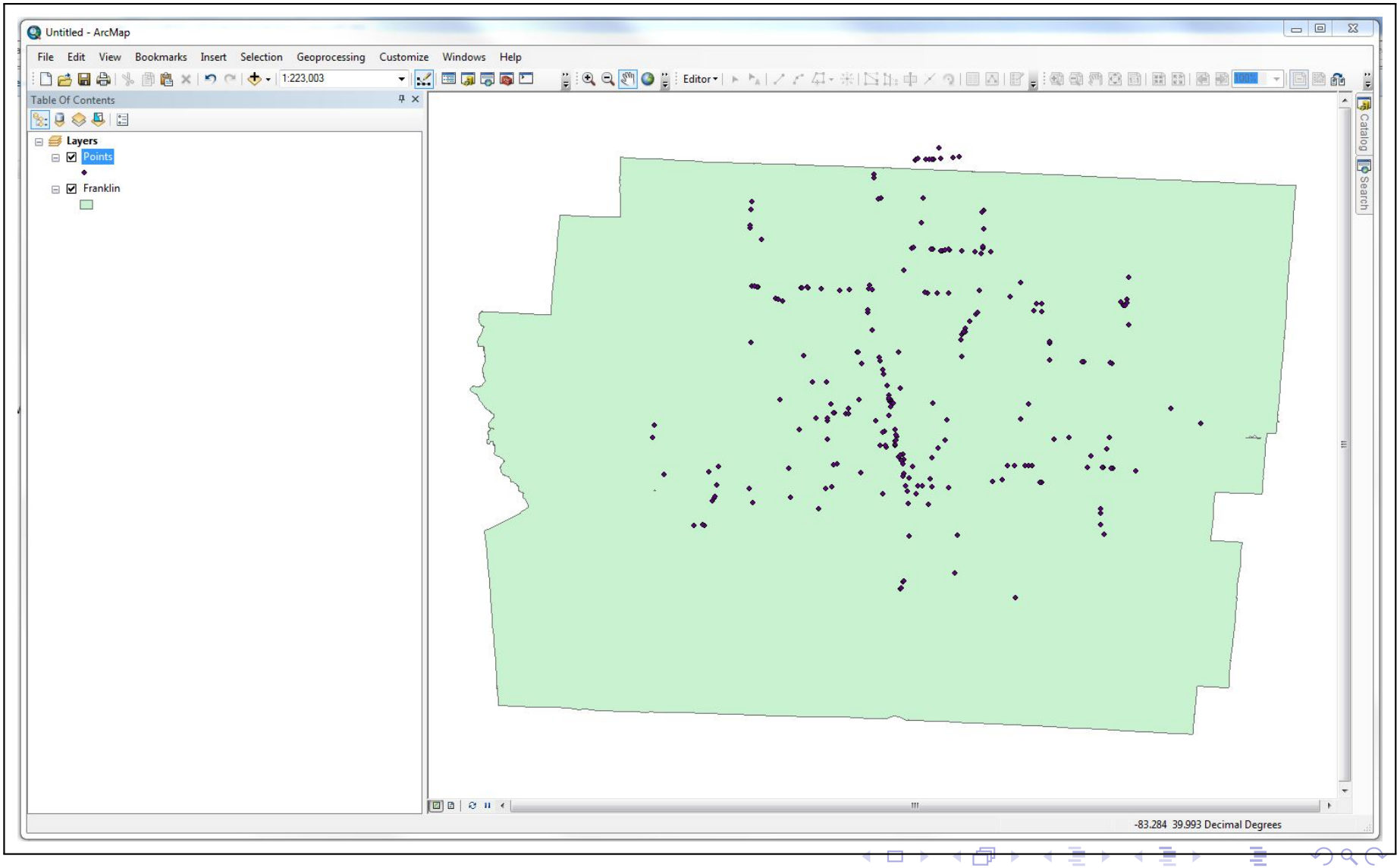
Open **Franklin.prj** and **Points.prj** (they should be two text files - See slide 22 of *Lecture 3*)

Copy entire contents of **Franklin.prj** and paste over all contents of **Points.prj**

Save and close both **.prj** files. Add both shapefiles to an empty **.mxd** file.

Did you notice that we didn't have to know what the coordinate system/projection for **Franklin.prj** was? Nice right?

There it is


Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:223,003
Editor
Table Of Contents
Layers
Points
Franklin
Catalog
Search
-83.284 39.993 Decimal Degrees

Python Solution

Assume that **Points.shp** is back to its original (poorly defined) self. To fix the misalignment issue in Python we need to do the following two things:

First, pull the spatial reference from **Franklin.shp**

- (What function, in what module, accomplishes this?)

Second, assign that spatial reference to **Points.shp**

- There is a *DefineProjection* function in the *management* toolbox in ArcMap
- Ask the TA about it.

Python Solution

```
# Jake K. Carr
# The Ohio State University
# Lecture 5: Misalignment Correction

# Import ArcPy and set Workspace
import arcpy
arcpy.env.workspace = r"C:\GEOG 5222\Lectures\Lecture 5\Maps"

# Create spatialReference object from Franklin.shp
sr = arcpy.Describe("Franklin.shp").spatialReference

# Assign projection from Franklin.shp onto Points.shp
arcpy.management.DefineProjection("Points.shp", sr)
```

Did you notice that we didn't have to know what the coordinate system/projection for **Franklin.prj** was? Nice right?

Ultimate Python Solution

At the end of the day there is no need to open ArcMap, load the (**.csv**) table into the map document, and then use the Properties, Display XY Data technique to convert our coordinate data into a shapefile/feature class.

As with everything else we've seen, there's an ArcPy function for that:

The *MakeXYEventLayer* tool from the *management* toolbox converts a table of data into a TEMPORARY layer file.

We can save the TEMPORARY layer file as a shapefile/feature class with the *CopyFeatures* tool from the *management* toolbox.

Ultimate Python Solution

```
# Jake K. Carr
# The Ohio State University
# Lecture 5: Create a point shapefile from table data

# Import ArcPy and set Workspace
import arcpy
arcpy.env.workspace = r"C:\Users\carr5_000\Google Drive\GEOG 5222\

# Create spatialReference object from Franklin.shp
sr = arcpy.Describe("Franklin.shp").spatialReference

# Make the temporary XY event layer...
arcpy.management.MakeXYEventLayer(r"C:\Users\carr5_000\Google Driv
                                  "Lon", "Lat", "Coords", sr)

# Save to a layer file
arcpy.management.CopyFeatures("Coords", "Restaurants.shp")
```

Geocoding

The process of using the information about the location of a geographic object and finding the 'exact' location of that object on the map.

Typically we will work with locations for which we don't have the Long/Lat coordinates (unlike the example from above), but have information on the street address of these locations.

Ultimately, we will match these addresses to a corresponding street line segment from a 'good' street level feature class/shapefile.

If the match is 'good enough' we'll say that we've found the point location of that particular location.

Geocoding Workflow

- Obtain address data for locations we want to 'pin-point'
- Obtain street network reference data for area under consideration
- Use street network reference to build an address locator
- Use the address locator from the Geocoding Toolbar to locate address from address data table
- Re-match unmatched address

Geocoding Process: How it works

For example, suppose we wanted to locate the point on the map corresponding to the location of the building with address:

380 New York Street, Redlands, CA 92373

What information are we given here?

The building is in CA. Should we look in a map of NJ for this building then? (Hint: Nope)

Geocoding Process: How it works

We also know that the building is in Redlands, CA. So, we should probably not waste our time looking in San Diego, or San Francisco.

If we had a zip code map we could look for the zip code polygon corresponding to 92373.

We could then find all of the streets that intersect the zip code polygon and look for any street segments called ‘New York Street’.

Finally we would pick the segment called ‘New York Street’ for which 380 falls in the interval of address numbers.

What We Will Need

The first step to finding something on a map is to have the right map.

It would be impossible to find your way to 380 New York Street in Redlands, California if you only have a map of New Jersey.

Also, you won't be able to pinpoint the street address very well if your map only shows highways and/or major cities.

Therefore your map must have enough detail of the area in question to pinpoint the location for which you are searching.

What We Will Need

ArcMap goes through the same exact process to geocode a set of addresses.

Therefore, the layers that we will use for creating an address locator (more on that in a bit) need to have an adequate set of details to locate the point that we want to find.

When looking for addresses, the primary reference data usually consists of a street network, but a parcel map can sometimes be used as well.

Again, the important thing to remember is that the data has to have a level of detail commensurate with what you want to find.

Address Locator

The address locator is what makes geocoding possible.

An address locator is created from a street network feature class - it connects to the street network and allows us to search the network for specific locations.

Once created, an address locator contains a snapshot of the address attributes in the reference data - like address format, zip code field, etc.

The address locator also contains a set of address parsing and matching rules that directs the geocoding engine to perform address standardization and matching.

Address Locator

When you enter an address into an address locator, the geocoding engine converts the input address into pieces, such as the number, street name, and street type, based on the parsing rules defined in the address locator.

These pieces are known as address elements.

The geocoding engine may generate multiple interpretations of the same address, as some values in the input address can be considered in more than one element.

For example, the word park can be both a street name and a street type.

Address Locator

Each combination of the address elements will be searched in the address locator.

The goal is to find all the possible matching candidates.

Once possible candidates are identified, each individual variable in the candidate is compared with each corresponding address element.

A score is generated indicating how well the address is matched.

Finally, the address locator presents the best matches based on the score and the location of the address being matched. We pick the 'best' one and say **There it is**!

Address Data

The first step to any geocoding job is to look at the data you want to geocode!

Here, we will use the data (so nicely provided) in **Addresses.csv**.

We will want an address locator that expects all address elements in a single field (called Addr in **Addresses.csv**), and possibly some other information, like zip code.

That means that we need to pick the appropriate Address Locator Style!

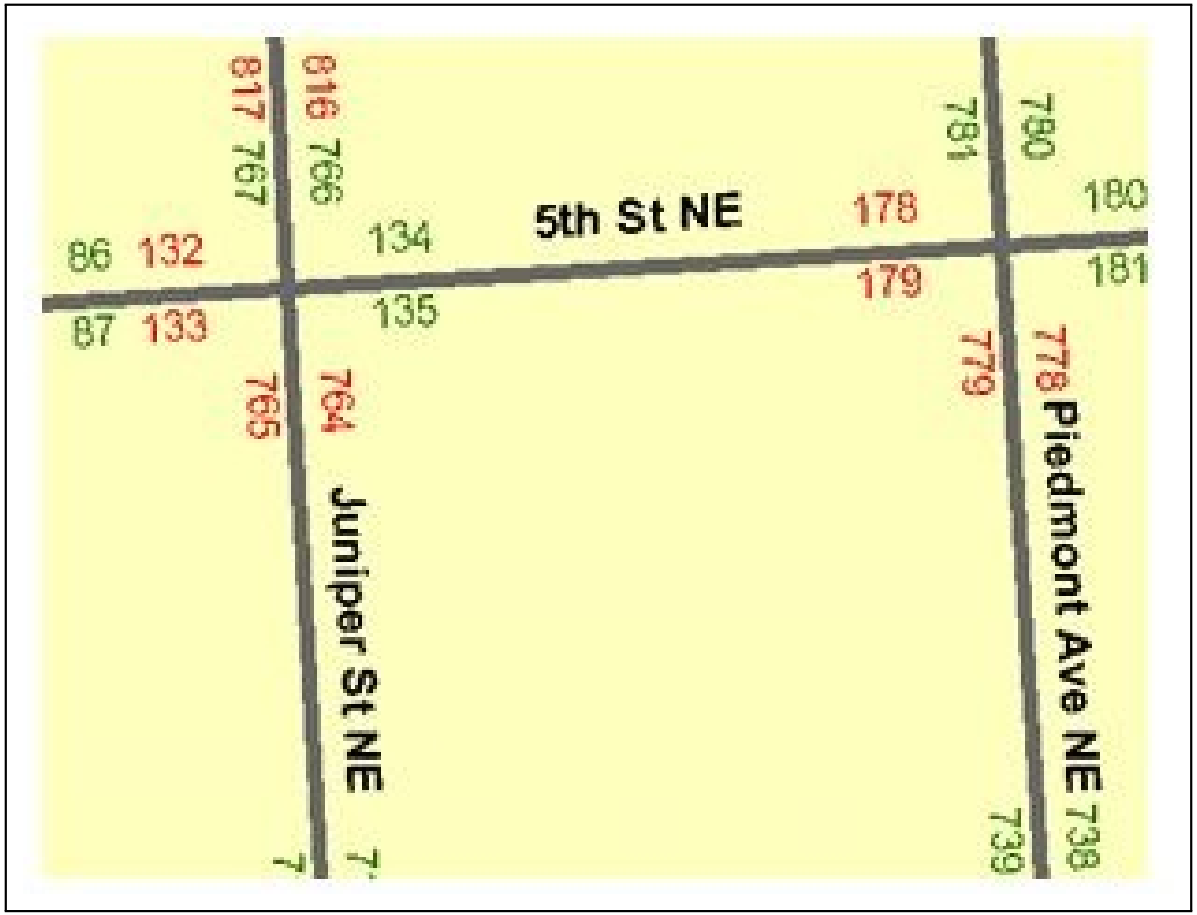
Address Locator Styles

Ask the TA what it knows about address locator styles.

It probably makes sense to use the 'US Address – Dual Ranges' address locator style, since:

- we have 'All address elements in a single field' in **Address.csv**
- we are interested in 'Finding a house on a specific side of the street'

What does Dual Ranges Mean?



Reference Data

Reference Data refers to the base data that the Address Locator is built off of.

Recall what the TA showed us about Address Locators, specifically the 'US Address - Dual Ranges' locator style.

It says that the Typical reference dataset geometry is **Lines**

As mentioned above, we will use a street network to reference our address against.

Reference Data

Where do we get the streets network shapefile that we need for geocoding purposes?

Luckily for us, the U.S. Census Bureau maintains high quality, up-to-date street network data.

As usual, lets ask the TA about Census street data:

NOTE: Census Shapefile data also goes by the name TIGER or TIGER/Line, which stands for: Topologically Integrated Geographic Encoding and Referencing.

Census TIGER/Line

The Census Bureau makes their data available by county[2].

Look at the first link on the webpage in the footnote.

What does **tl_2014_01001_addrfeat.zip** mean?

Obviously it references a TIGER/Line (tl) shapefile referencing 2014 data, containing a set of address range features (addrfeat).

What's with the weird five digit number?

[2] ftp://ftp2.census.gov/geo/tiger/TIGER2014/ADDRFEAT/

FIPS Codes

It turns out that those (weird) five digit numbers refer to Federal Information Processing Standards (FIPS) codes used to identify each state and county in the US.

The first two digits correspond to the State, the last three correspond to the county.

The state FIPS code for OH is 39, the county FIPS code for Franklin County is 049 (the leading 0 is important!)

That means we want to download the **.zip** file called **tl_2014_39049_addrfeat.zip**

Streets File

Download the streets file and unzip the shapefile to the same folder with **Franklin.shp**.

Add both to an empty ArcMap **.mxd** file.

Open the attribute table to examine what fields are available.

We'll pay particular attention to the fields listed from **FULLNAME** to **ZIPR**.

Streets File

Recall that we are going to build an address locator based on address ranges for both sides of the street segment.

The fields in this shapefile contain the data needed to build such an address locator.

FULLNAME contains the full streetname of each of the 64,065 street line segments in Franklin County.

Streets File

The fields **LFROMHN** and **LTOHN** contain the beginning (FROM) and end (TO) of the range of *House Numbers* (HN) on the 'LEFT' side of that street segment.

The fields **RFROMHN** and **RTOHN** contain the beginning (FROM) and end (TO) of the range of *House Numbers* (HN) on 'RIGHT' side of that street segment.

The fields **ZIPL** and **ZIPR** tell us what zip code the street line segment begins in and the zip code the street line segment ends in.

Get To Work

So far we've examined our address table (**Address.csv**) and found a quality street reference file (**tl_2014_39049_addrfeat.shp**).

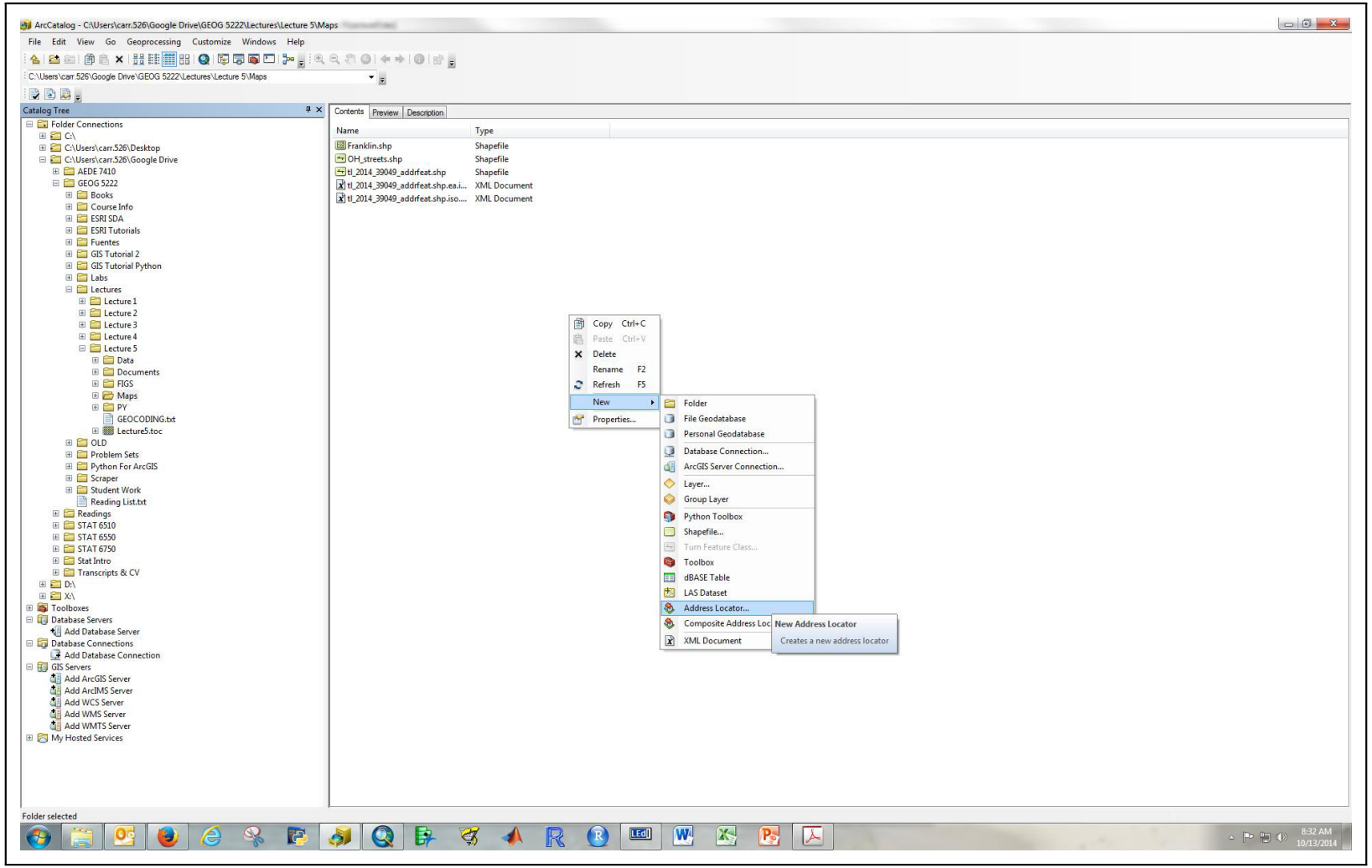
The next step is to create the address locator!

Simple: Open ArcCatalog and navigate to the folder (or geodatabase) with the street network data[3].

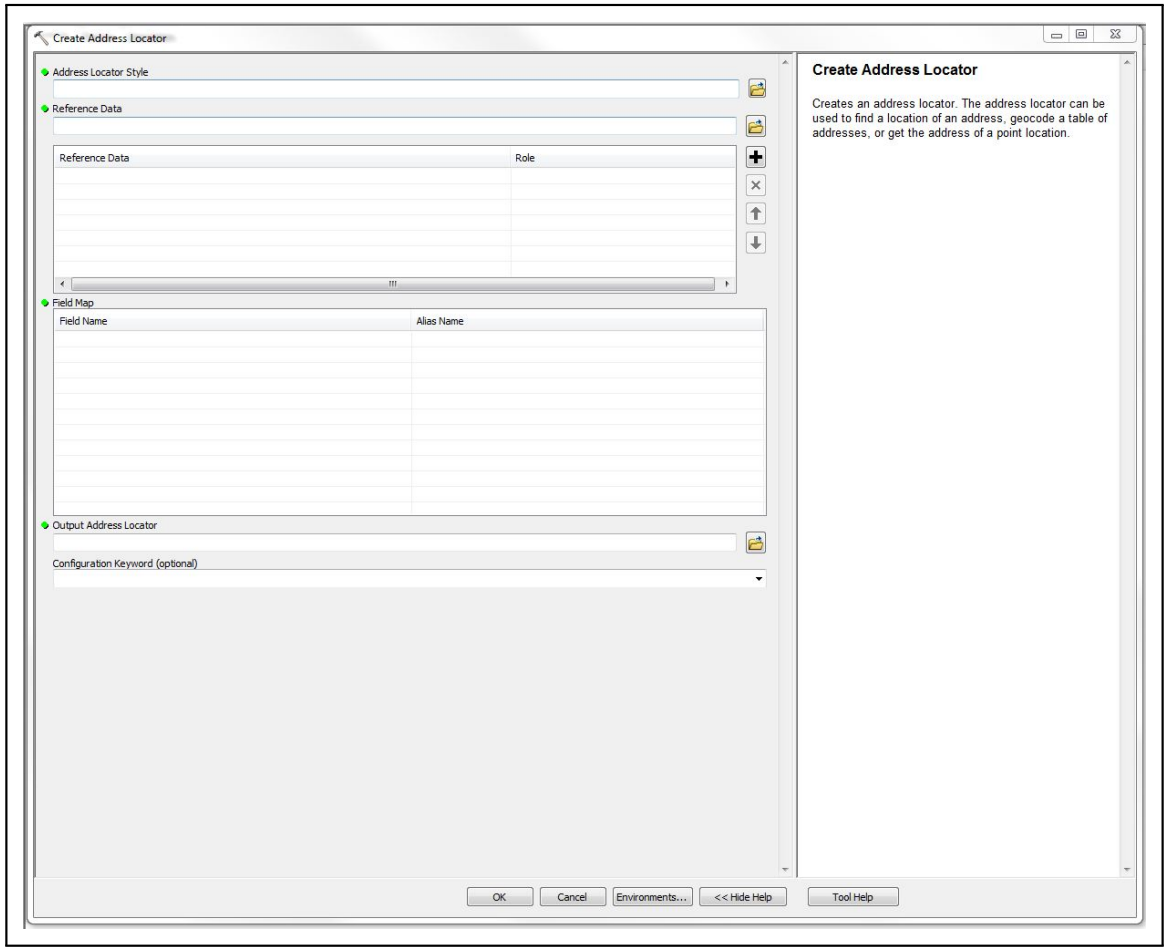
You know the routine: we'll use the Right-click and select method.

[3] For me thats at C:\Users\carr.526\Google Drive\GEOG 5222\Lectures\Lecture 5\Maps

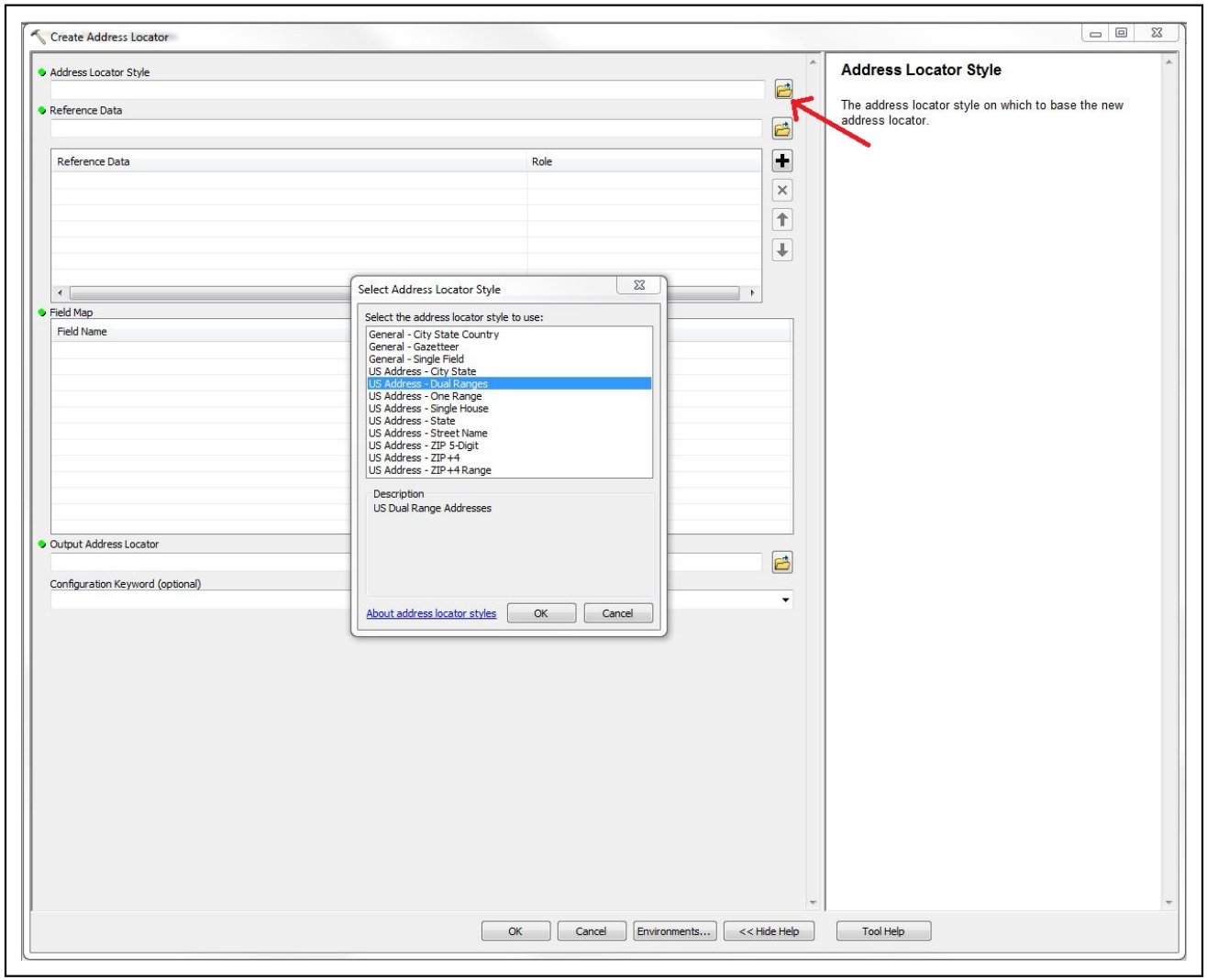
Create Address Locator



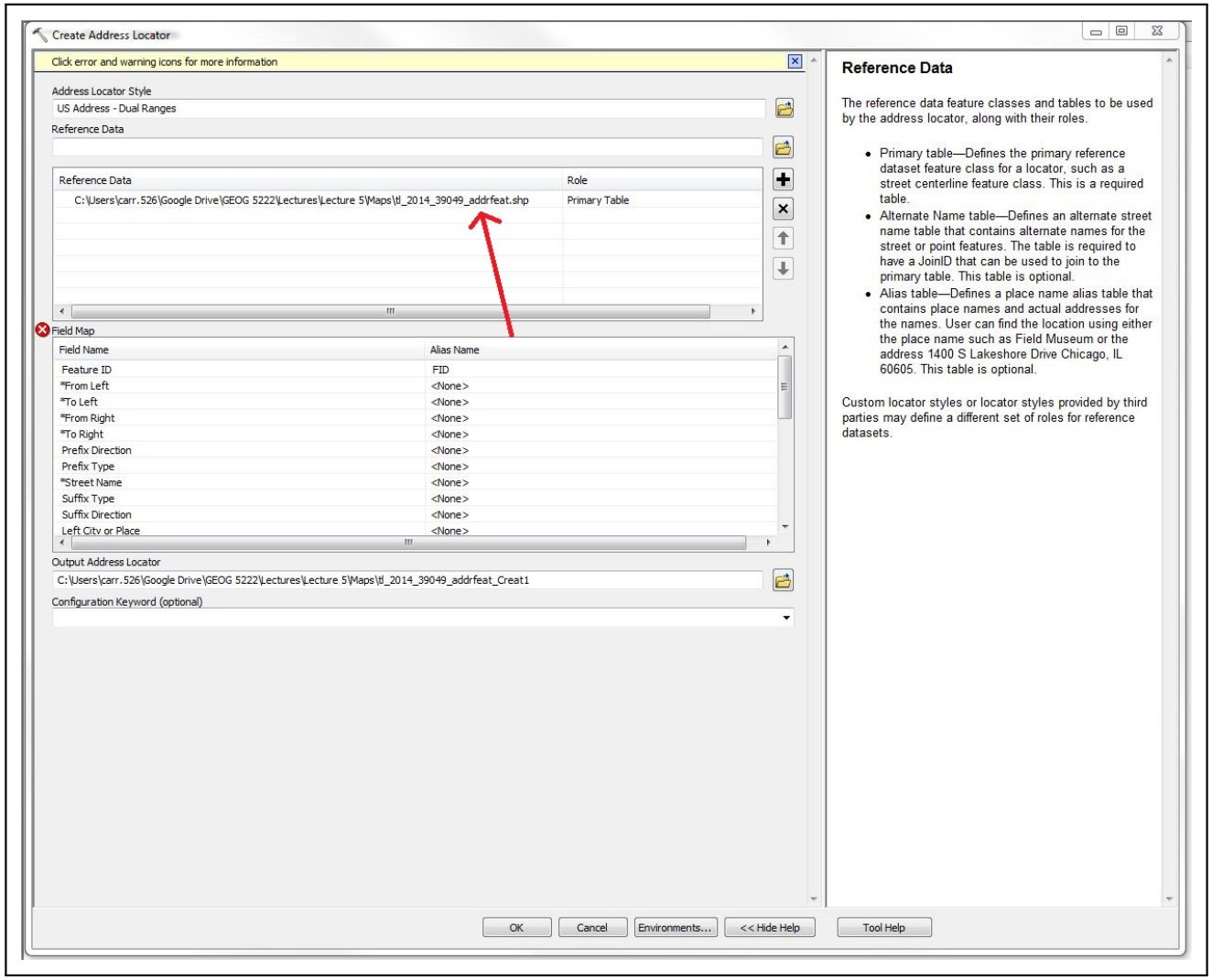
Address Locator Dialog Box



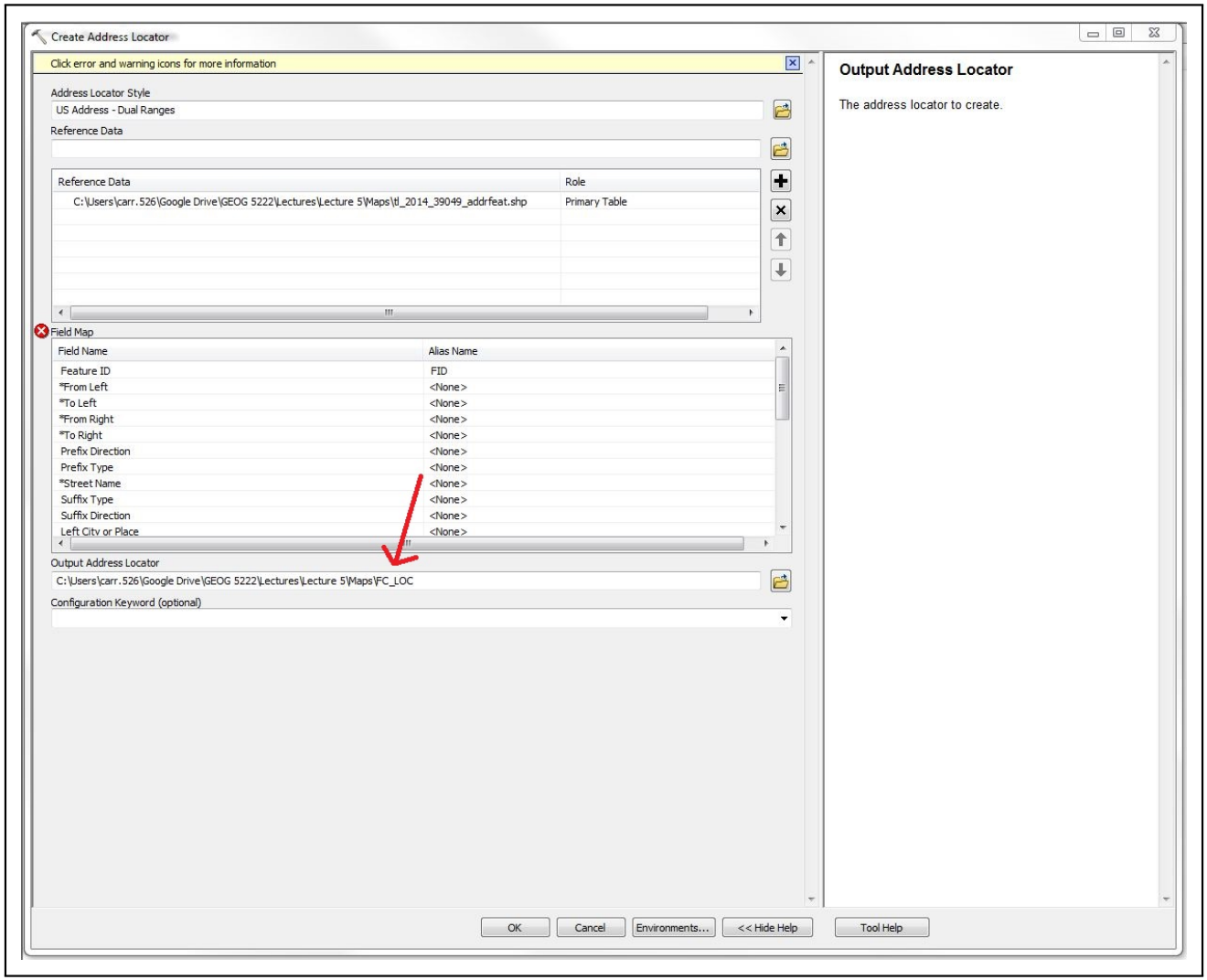
Set Address Locator Style



Set Reference Data = Street Network Shapefile



Set Output Location



Set Field Map

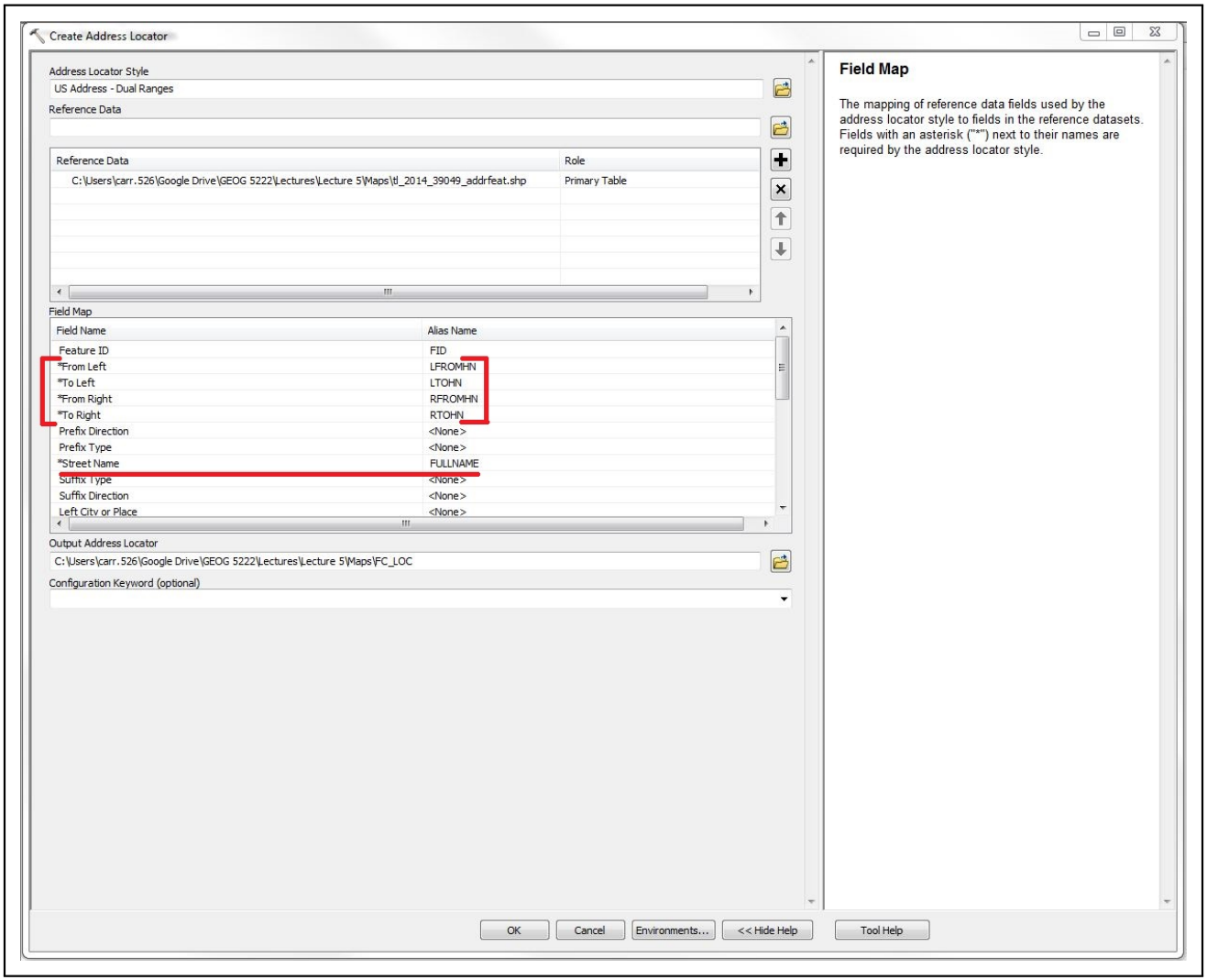
The field map is the most important part of this process.

This is where we tell ArcMap which fields in our street network shapefile (**tl_2014_39049_addrfeat.shp**) contain the street names, address ranges, and other important street segment information we'll need to match addresses to locations.

That's why we took note about which fields were in **tl_2014_39049_addrfeat.shp**, and what those fields represented.

Fields with an asterix (*) are required!

Set Field Map



Create Address Locator

Click OK, and ArcMap will begin compiling the address locator.

It will take a minute or two to run.

Once finished there will now be a file called **FC_LOC.loc** in the folder[4].

And once it is created, this address locator can be used to geocode any set of addresses in Franklin County - we can use it more than once!.

[4] Note that like shapefiles, loc files consist of three seperate files

Let the fun begin!

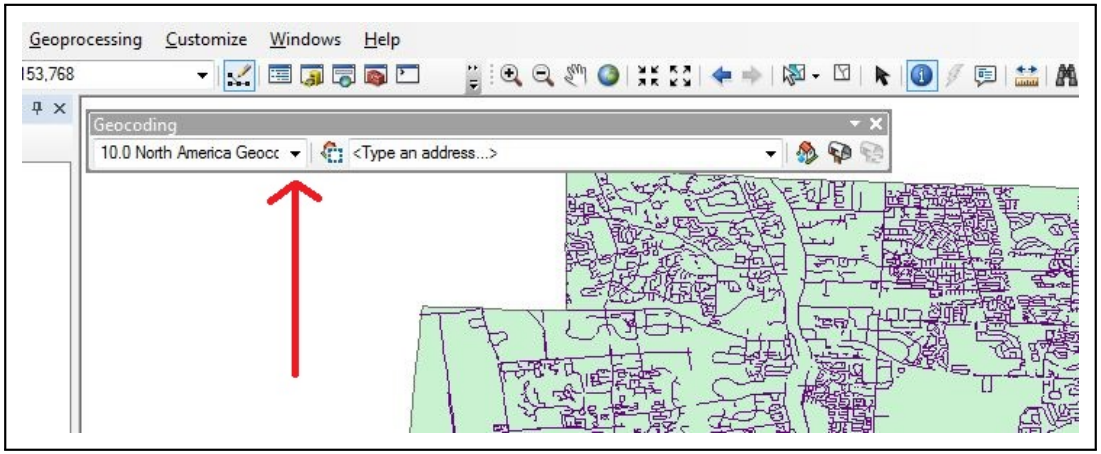
Go back into ArcMap (where we had **Franklin.shp** and **tl_2014_39049_addrfeat.shp** open).

Add the Geocoding toolbar by clicking Customize, Toolbars, Geocoding.

It's just that easy!

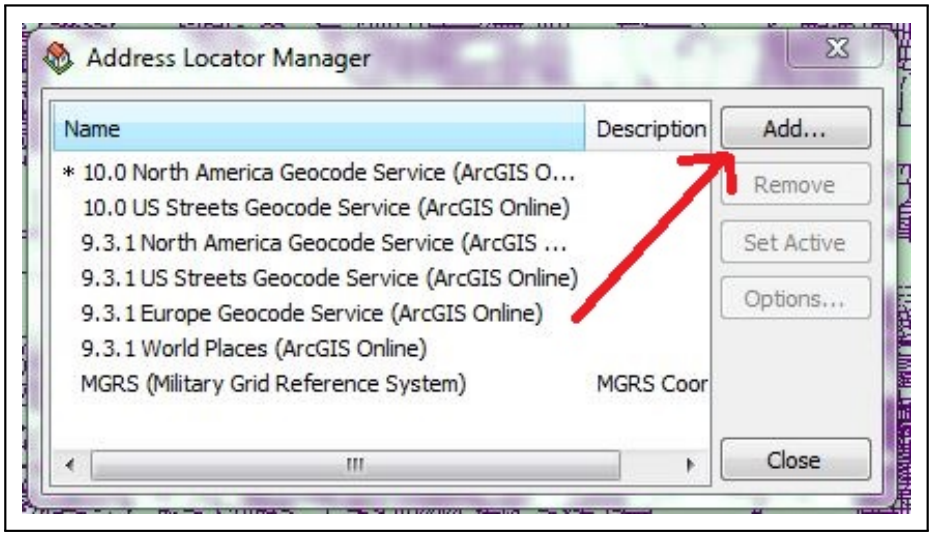
Geocoding Toolbar

Click on the left-most drop-down menu on the Geocoding Toolbar and select 'Manage Address Locators...'.



Address Locator Manager

In the Address Locator Manager dialog box, click the Add button.

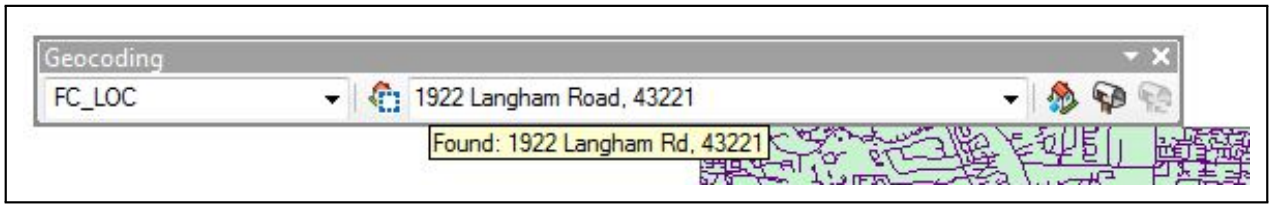


Add **FC_LOC.loc** and click Close.

Now we are all set to start geocoding!

Try to Find Your Location

Try it out by entering your street address into the second window in the Geocoding Toolbar.



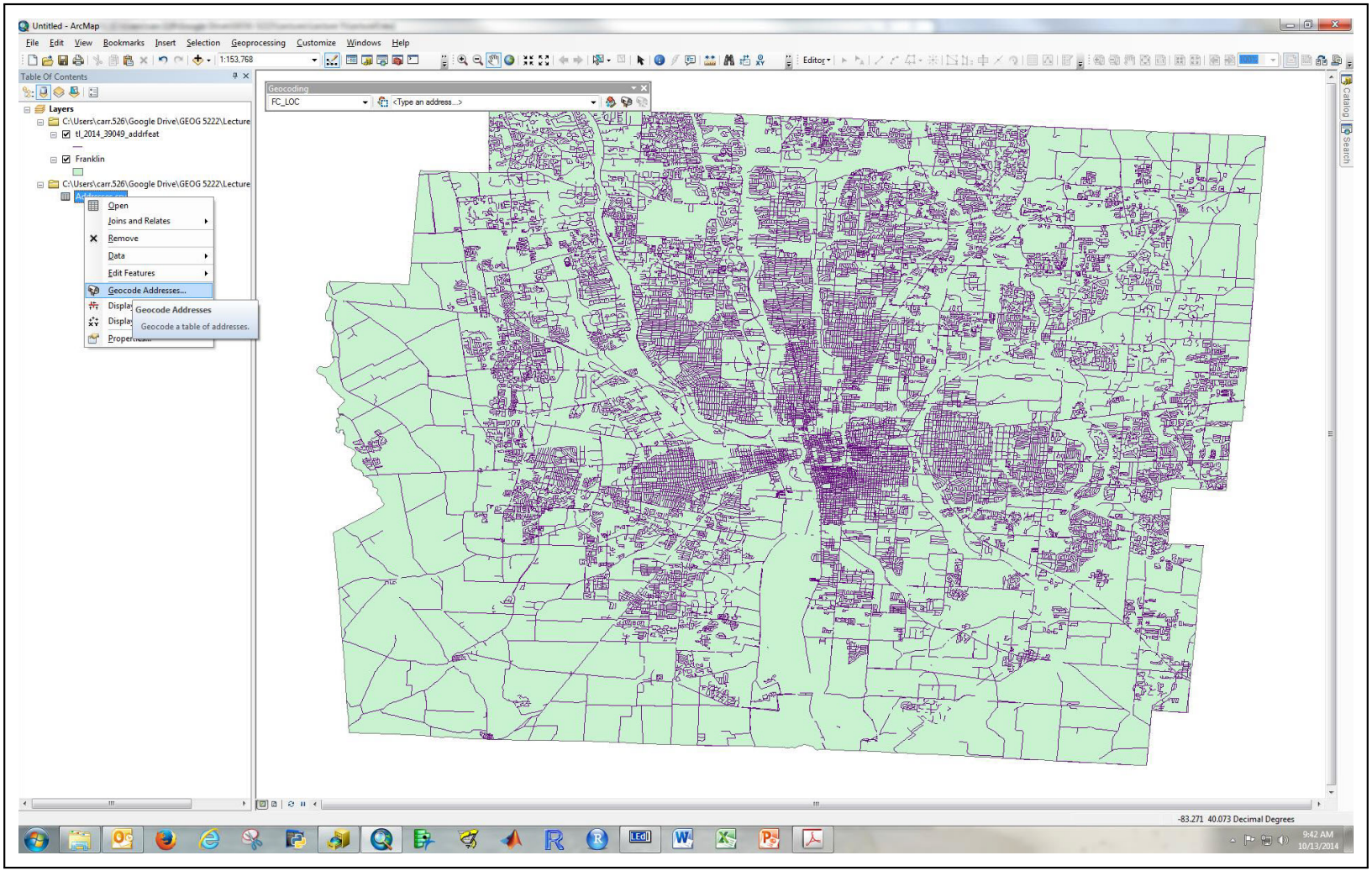
If you Right-click the address in the second window you can select 'Add Labeled Point' to add a point event to the map, labeled with the address you entered.

Geocoding a Table of Addresses

Add **Addresses.csv** to the ArcMap **.mxd** document.

Right-click the Address.csv table on the List by Source view of the Table of Contents and click Geocode Addresses in the context menu.

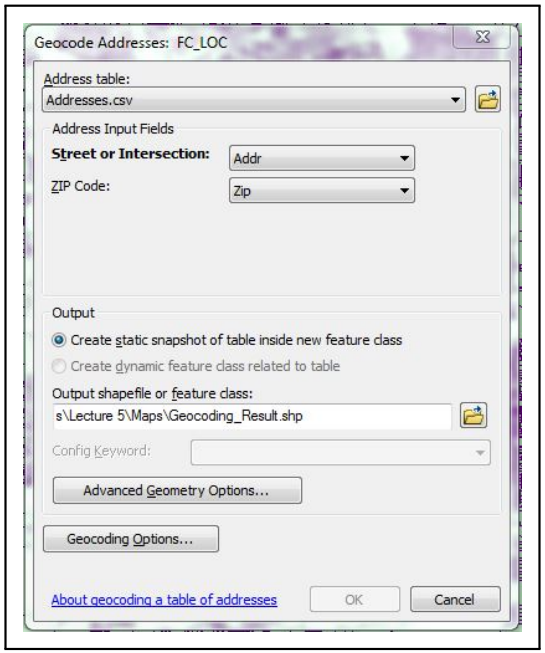
Geocoding a Table of Addresses



Geocoding a Table of Addresses

Choose FC_LOC as the Address Locator to use.

The Geocode Addresses dialog box appears:



Geocoding a Table of Addresses

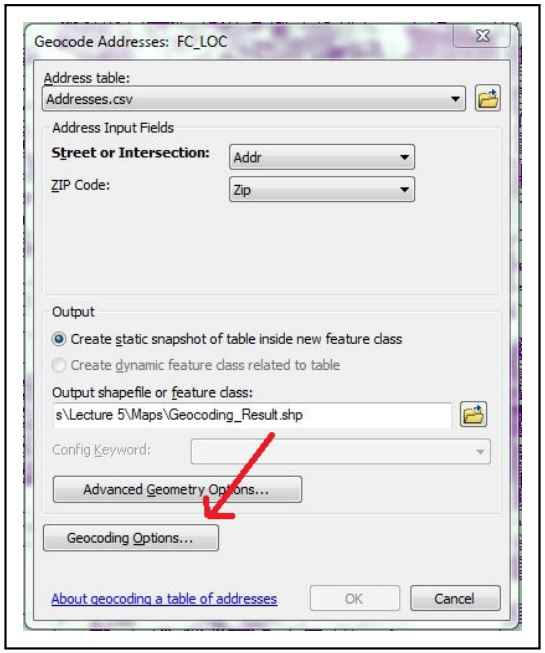
Set the appropriate Street Address field from **Addresses.csv**: recall that the street address data is found in the column called *Addr*.

Set the right zip code field from **Addresses.csv**: recall that the zip code data is found in the column called *Zip*.

Save the output shapefile/feature class to the same directory as the **Franklin.shp.**.

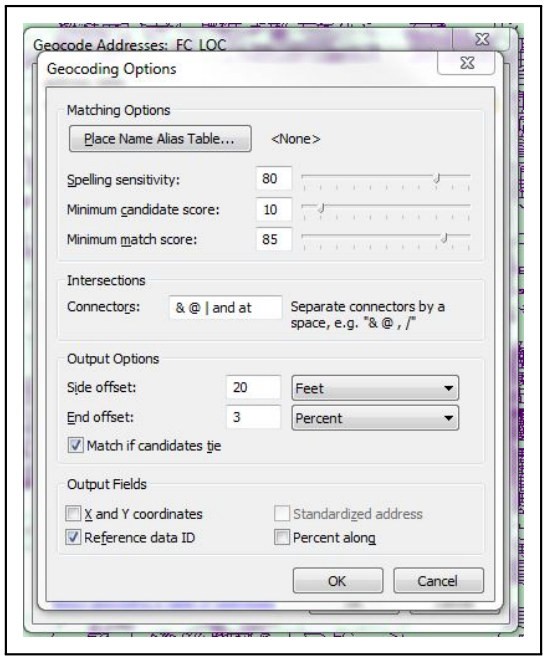
Click on the Geocoding Options button.

Geocoding a Table of Addresses



Geocoding a Table of Addresses

Additional options can be set here, like spelling sensitivity scores, spatial offsets, and Intersection Connectors.

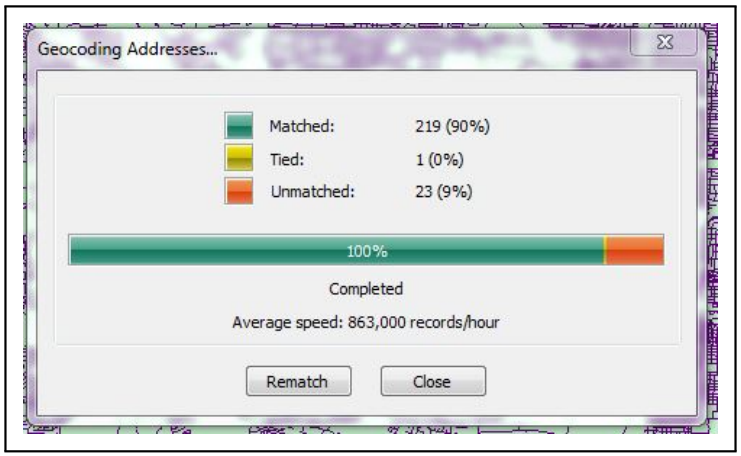


It's typically alright to just accept the defaults.

Click OK, and then OK again: $\Rightarrow$ We're GEOCODING!

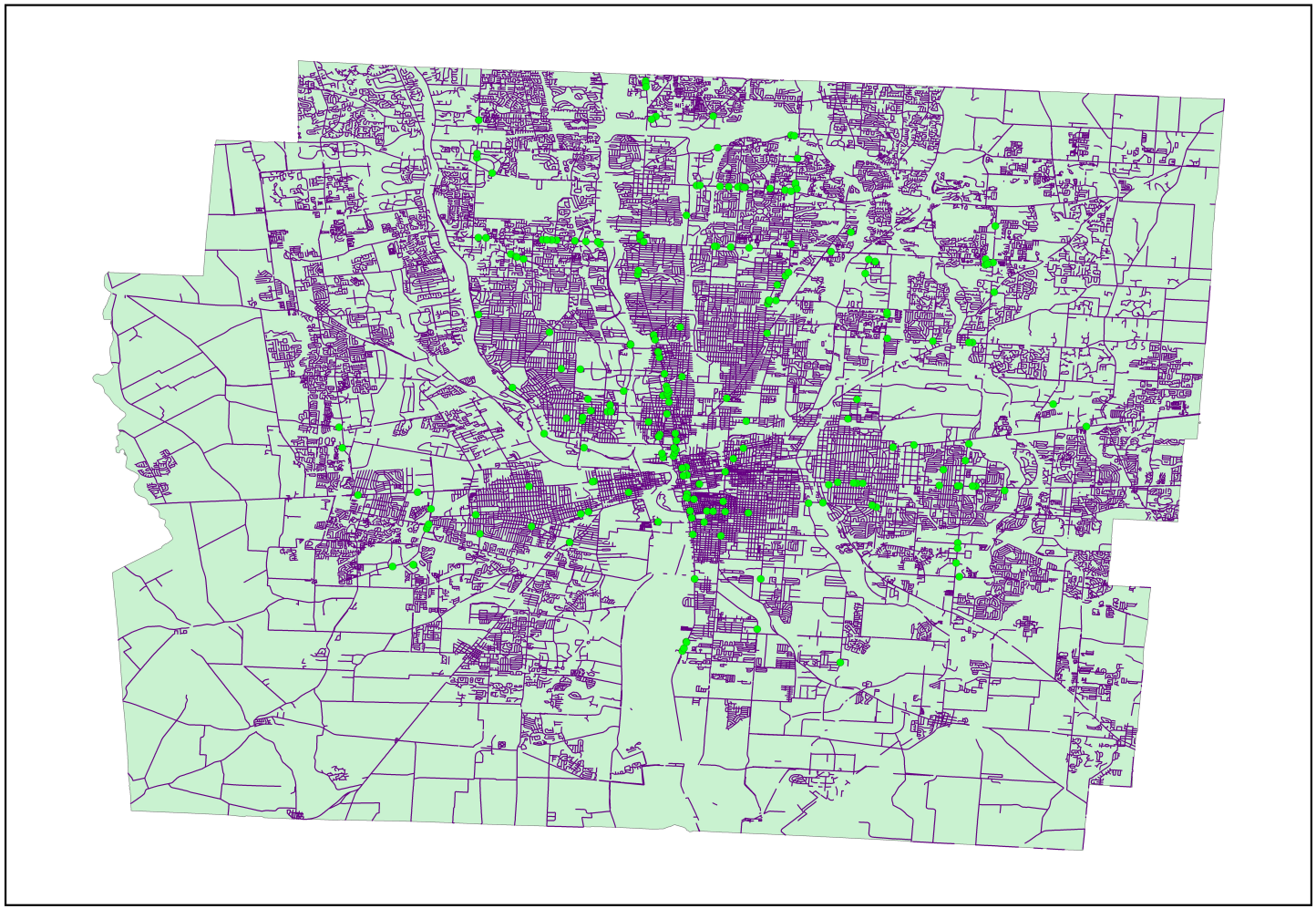
DONE!

When the process is complete the message box will tell us how successful we were at matching addresses from **Addresses.csv** to known locations in Franklin County.



About 90% of addresses matched a location, around 9% failed to match, and 1 observation (from **Addresses.csv**) is 'tied' for a match. Click the Close button to exit the Geocoding dialog.

The Results of All That Hard Work



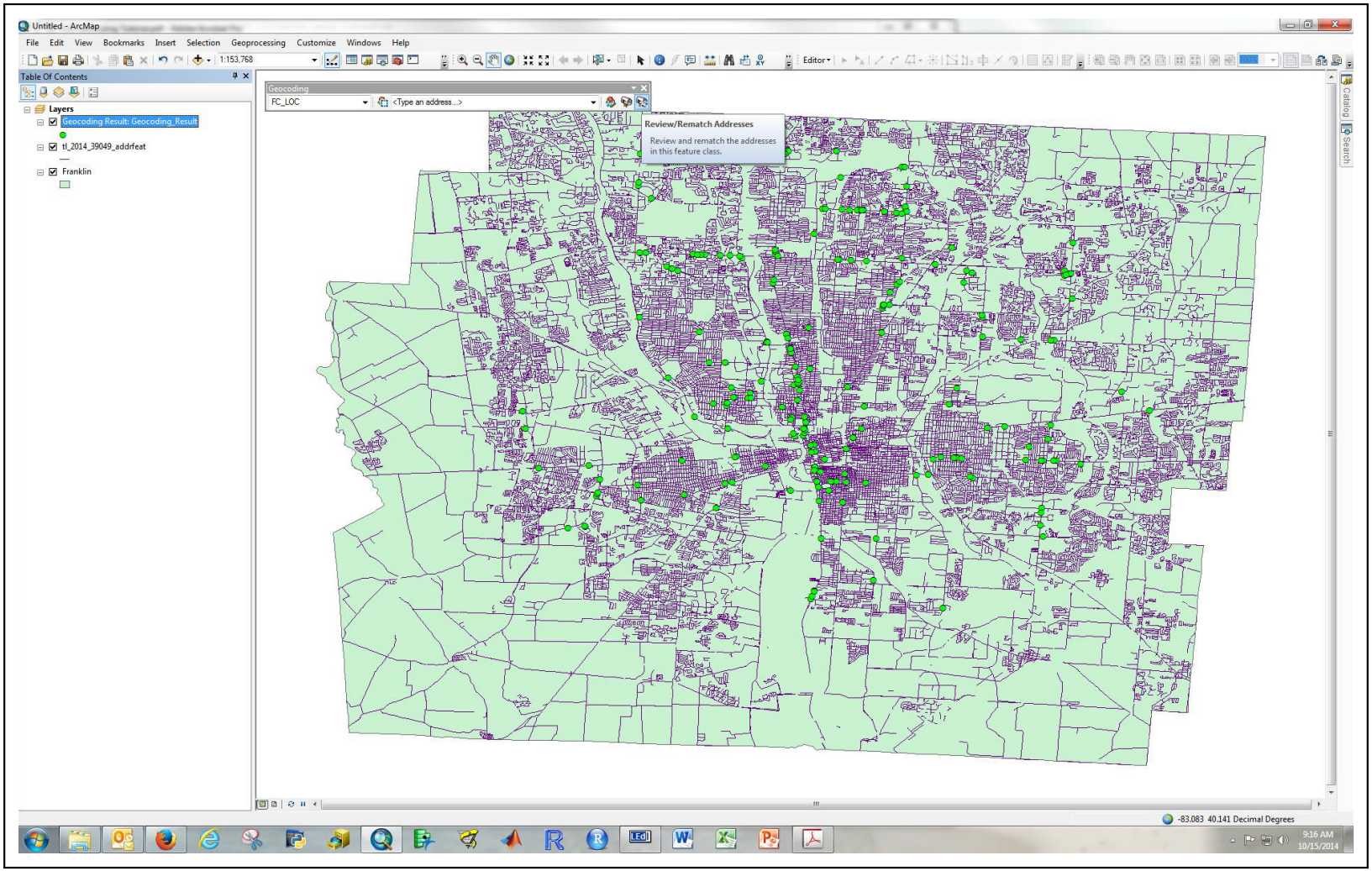
I Thought We Were DONE

Not so fast my friends?

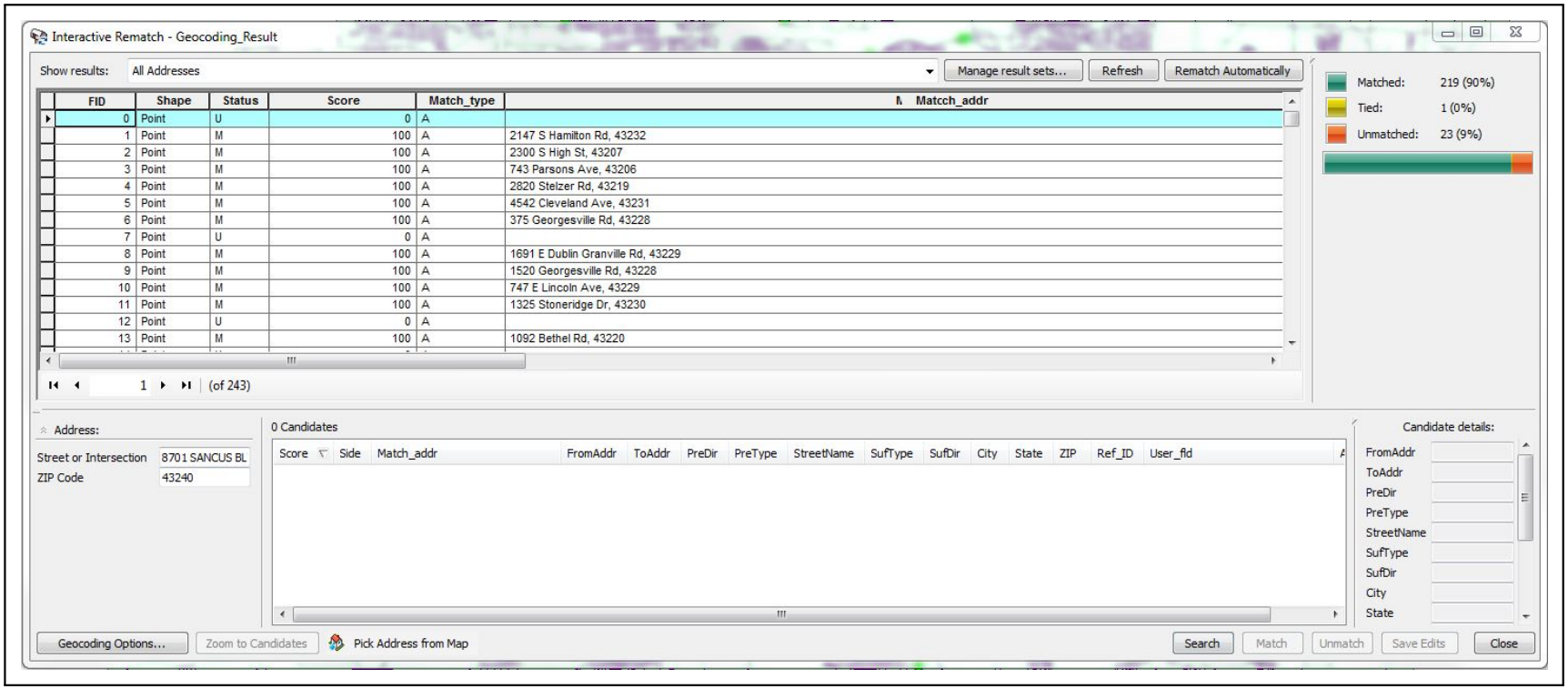
What about those 9% that failed to match? What didn't those address match to a location?

We can investigate these cases further by opening the Review/Rematch Dialog from the Geocoding toolbar.

Review Process



Review Process



Review Process

Set the 'Show results:' drop-down to 'Unmatched Addresses'.

The table just below that will show all of the 23 unmatched addresses.

By selecting an individual row in that table, we can see what possible matches the Address Locator found for that particular observation.

- For some reason *House of Japan* and *Red Lobster* don't have any possible matches.
- What's the deal?

Review Process

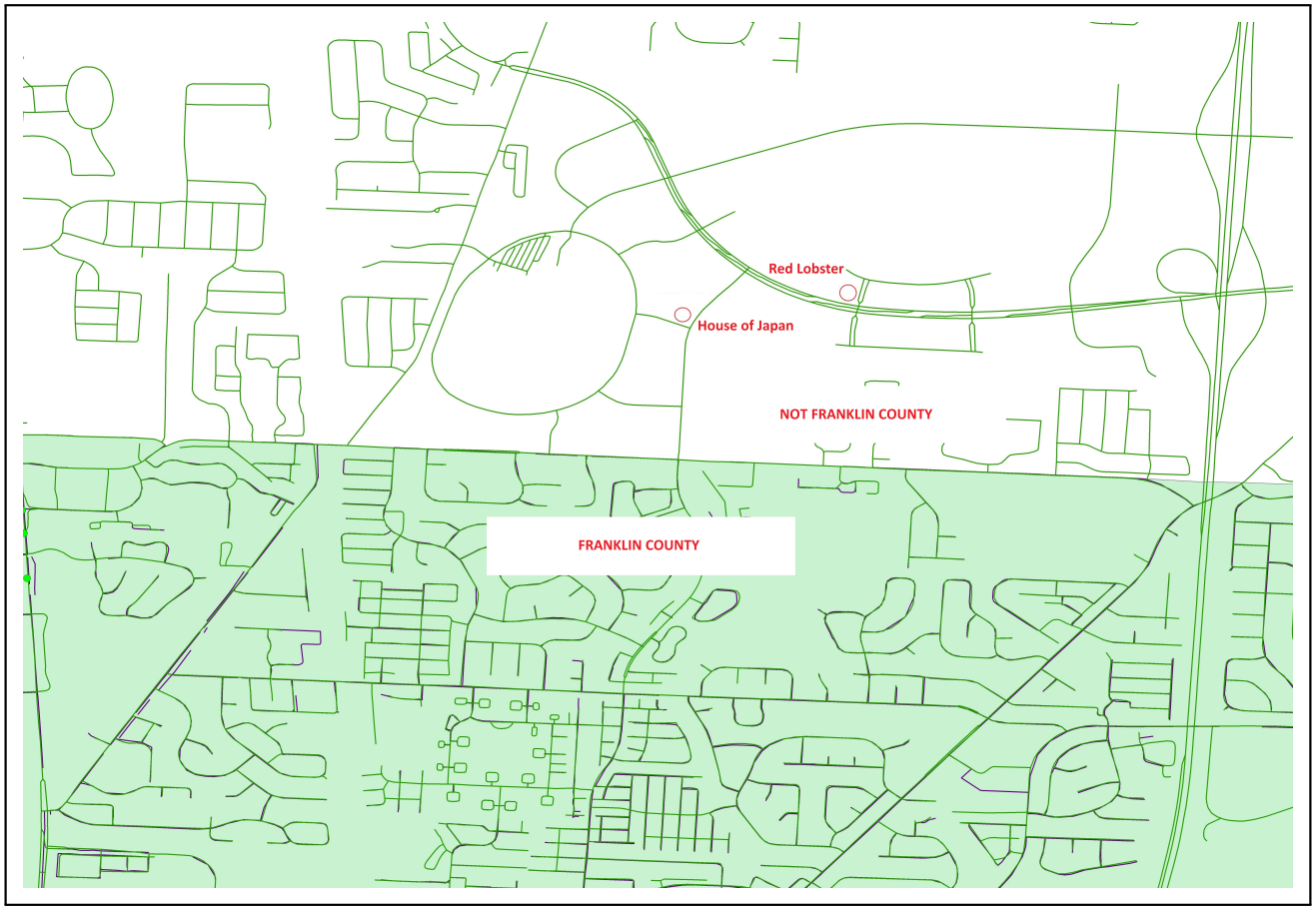
Ask the TA about 8701 SANCUS BLVD

It's a place that actually exists, so we know that the address information is legitimate.

Red Lobster is right in the same neighborhood!

Another thought - add **OH_streets.shp** to the map.

Columbus/Franklin County Boundary Issues



Columbus/Franklin County Boundary Issues

For this particular set of addresses, is it OK to assume that the 14 addresses with no matches fall outside of Franklin County, and hence are not referenceable from our street network?

Note that there are 13 restaurants in **Point.shp** that fall outside of Franklin County - we saw that a week ago.

That explains 13 of the 14 no-match restaurants, what about the remaining one?

- The remaining one is called Sammy's New York Bagels with address at 162 CITY CENTER DR

Sammy's Bagels

A TA search of the YP address for Sammy's Bagels reveals a location downtown - near the Commons[5].

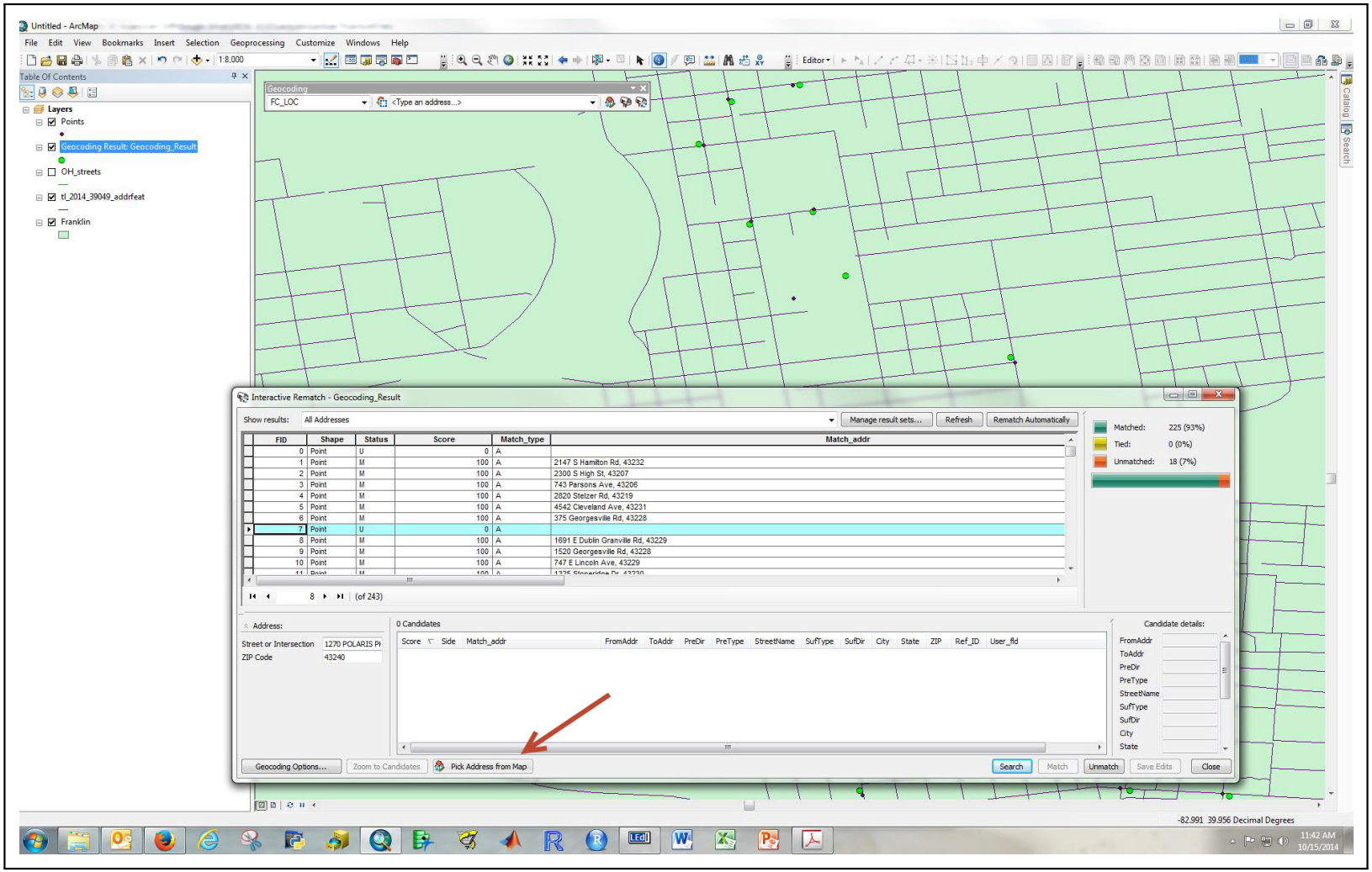
So we'll have to 'estimate' a location based on an estimated location from the web, and place a point on the map ourselves.

Navigate to the area on the map you want to place the point, and open the Review/Rematch Dialog from the Geocoding toolbar.

Select the 'Pick Address from Map' button.

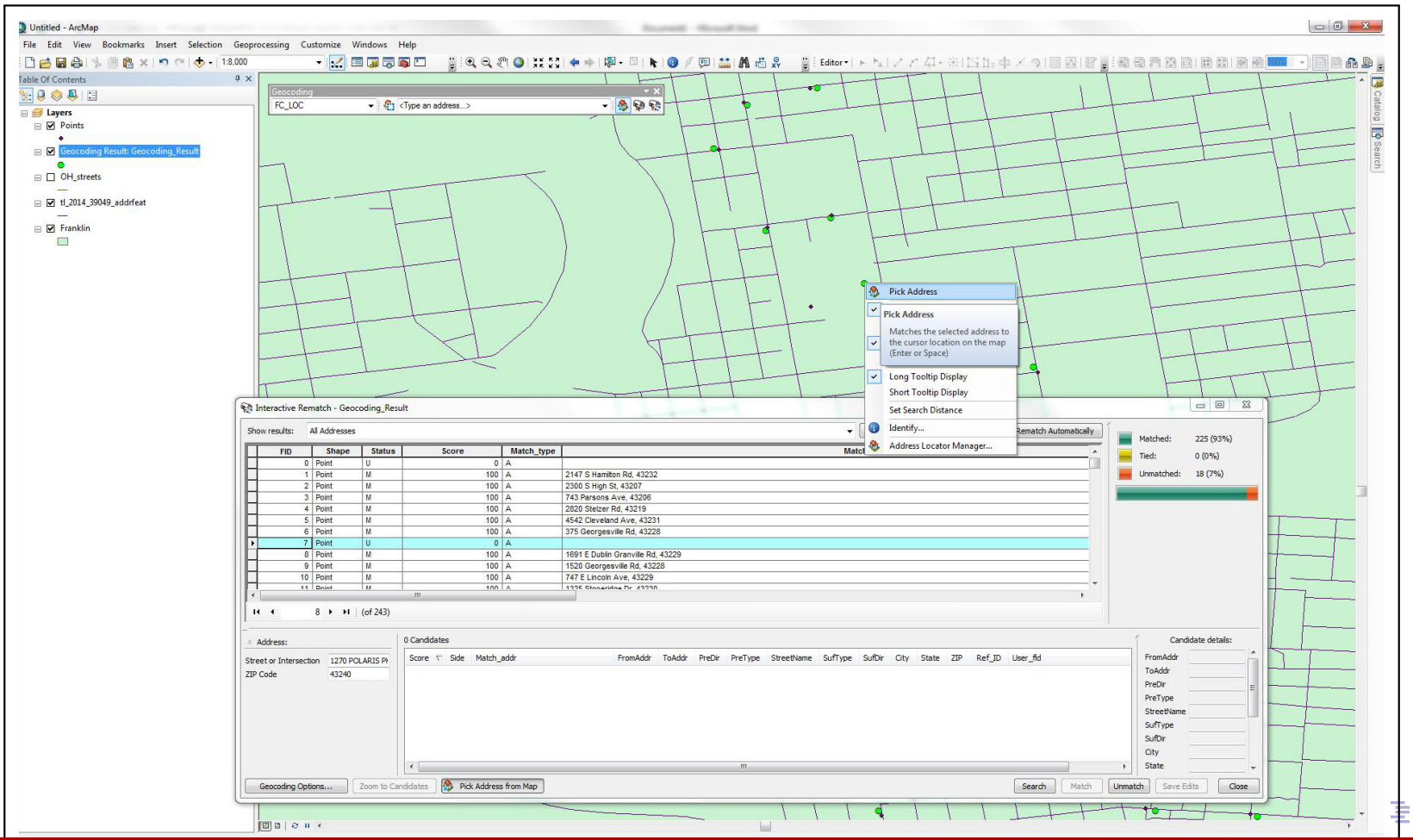
[5] Thanks mapquest.

Sammy's Bagels



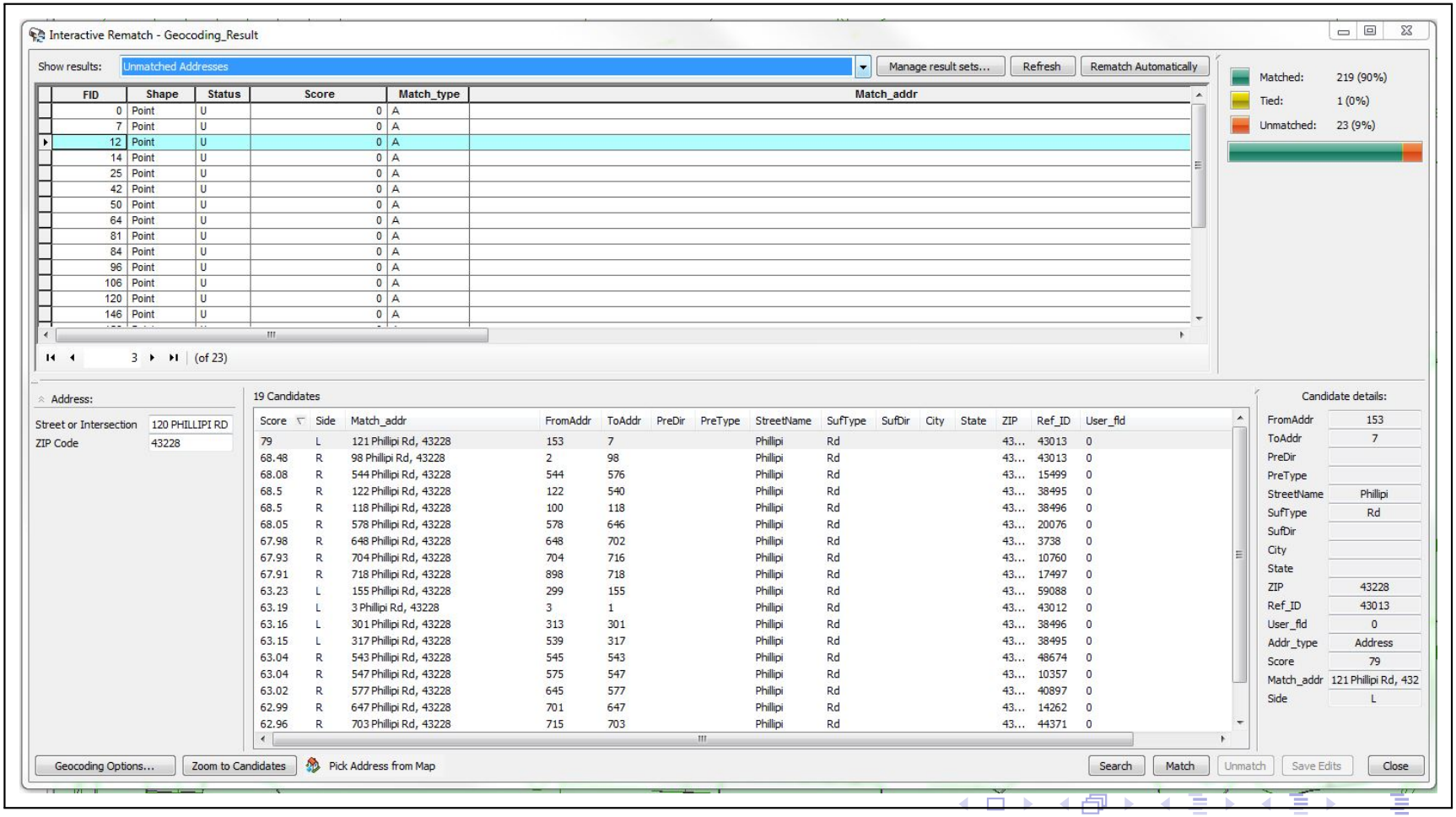
Sammy's Bagels

Hover to the location on the map you want to place the point.
Right-click and select **Pick Address**



Picking a 'Good' Match for the Remaining Unmatched

Use some intuition.



Ties

Continue to use some intuition.