Workflow to perform land use analysis on the City of Bothell's proposed surface water monitoring basins

This guide will describe the workflow used to create the analysis spreadsheet for land use in the City of Bothell's proposed surface water monitoring basins. The goal is for this analysis to *assist in* prioritization of surface water infrastructure improvements among these monitoring basins. This guide covers the derivation of analysis layers from Bothell's core GIS data using ArcGIS Desktop 10.6.1 (advanced license), but does NOT include the derivation of the basin boundaries or the land use feature class itself.

It's also important to note that this analysis and workflow are based on GIS data at coarse scale and are NOT intended to represent a scientific or survey-grade study of drainage or other phenomena and that:

GIS data (maps) are produced by the City of Bothell for internal purposes. No representation or guarantee is made concerning the accuracy, currency, or completeness of the information provided.

Accompanying this guide should be several file geodatabases and other files to help with this workflow.

Name	Date modified	Type	Size
Buildings_2019_11_21.gdb	12/16/2019 3:11 PM	File folder	
Impervious_2019_11_21.gdb	12/16/2019 3:11 PM	File folder	
Streets_2019_11_21.gdb	12/16/2019 3:11 PM	File folder	
Surfacewater_2019_11_21.gdb	12/16/2019 3:11 PM	File folder	
WatershedPrioritization_Analysis.gdb	12/16/2019 3:11 PM	File folder	
WatershedPrioritization_Analysis_Prep.gdb	12/16/2019 3:11 PM	File folder	
WatershedPrioritization_Analysis_Staged.gdb	12/16/2019 3:11 PM	File folder	
Land Use and Land Cover Analysis_V2.docx	12/16/2019 3:11 PM	Microsoft Word D...	57 KB
PopulationFieldCalculatorExpression.cal	12/16/2019 3:11 PM	CAL File	3 KB
tabulate_storm_layers.py	12/16/2019 3:11 PM	Python File	3 KB
WatershedPrioritization_Analysis_Blank.xlsx	12/16/2019 3:11 PM	Microsoft Excel W...	22 KB
WatershedPrioritization_Analysis_BothellResults.xlsx	12/16/2019 3:11 PM	Microsoft Excel W...	31 KB
WatershedPrioritization_Analysis_DataDictionary.docx	12/16/2019 3:11 PM	Microsoft Word D...	40 KB
WatershedPrioritization_Analysis_DataDictionary.pdf	12/16/2019 3:11 PM	Adobe Acrobat D...	890 KB
WatershedPrioritization_Analysis_Workflow.docx	12/16/2019 3:11 PM	Microsoft Word D...	2,940 KB
WatershedPrioritization_Analysis_Workflow.pdf	12/16/2019 3:11 PM	Adobe Acrobat D...	1,612 KB
WatershedPrioritization_PresentDelineation.pdf	12/16/2019 3:11 PM	Adobe Acrobat D...	3,146 KB

The following page has a table of contents that should help you approach replicating the workflow in detail. The first four sections cover preliminary preparation of data layers. If you prefer to skip to the final analysis steps, you can use the file geodatabase **WatershedPrioritization_Analysis_Staged.gdb** to start at **Part 6 – Final Analysis**. If you have any questions about this process or anything in this guide, please don't hesitate to contact John Ritzman at john.ritzman@bothellwa.gov.

Table of Contents

Workflow to perform land use analysis on the City of Bothell's proposed surface water monitoring basins 1

Part 1. Land Use Preparation 3

Part 2. Impervious Surface Preparation 7

Part 3. Physical Geography Preparation 13

Part 4. Storm Layers Preparation 14

Part 5. Population Density Estimation Preparation 16

Part 6. Final Analysis 23

- **Final Part 1.** Simple Overlap Analysis 24
- **Final Part 2.** Complex Overlap Analysis 26
- **Final Part 3.** Storm Layer Analysis 28
- **Final Part 4.** Final Wrap-Up 29

Appendix 1. Python Code for Storm Layer Analysis 31

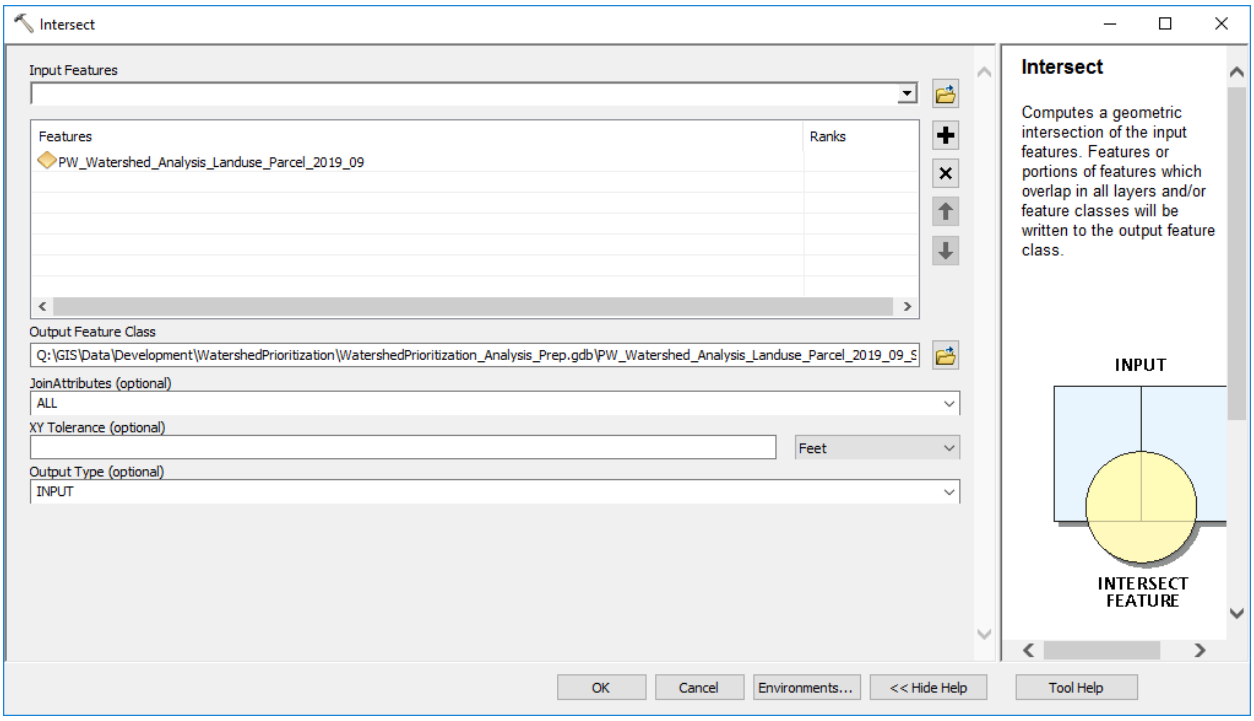
Part 1. Land Use Preparation

1.1. Start with the file geodatabase **WatershedPrioritization_Analysis.gdb**.

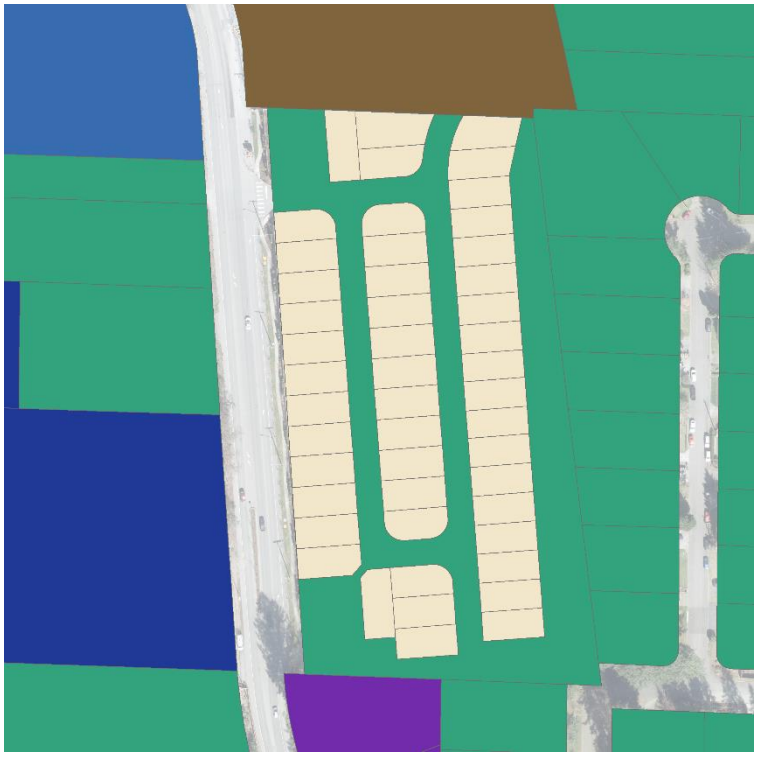
1.2. Copy the **PW_Watershed_Analysis_Landuse_Parcel_2019_09** feature class to **WatershedPrioritization_Analysis_Prep.gdb** so we can edit it without changing the original.

1.3. In ArcMap, add the *newly copy/pasted* feature class **PW_Watershed_Analysis_Landuse_Parcel_2019_09**.

1.4. Perform a self-intersect to identify any areas of overlap.



1.5. Identify any areas of significant overlap.

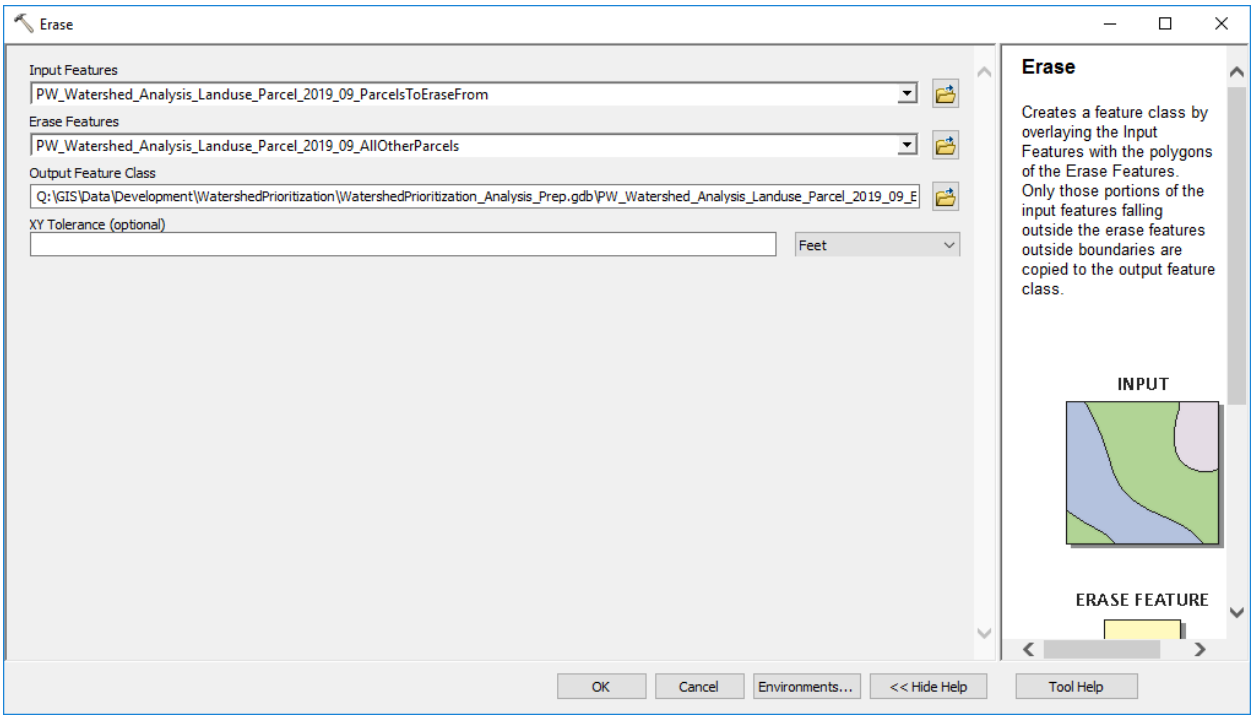


1.6. Six areas of significant overlap were found in the land use feature class. All had one large parcel covering many others and occasionally filling up additional space in between or around them. The six large, overlapping parcels are as follows:

- **a.** PIN = '00519500200405'
- **b.** PIN = '27043600414500'
- **c.** PIN = '1679500000'
- **d.** PIN = '7922690000'
- **e.** PIN = '8678800000'
- **f.** PIN = '9540500000' *(3 parcels with this same PIN)*
- **g.** SQL query to select all eight: **PIN IN ('00519500200405', '27043600414500', '1679500000', '7922690000', '8678800000', '9540500000')**

1.7. We'll now deal with the overlapping parcels. Copy and paste the land use layer into the ArcMap Table of Contents to create two of them. Rename them as needed to keep track of which is which. In one layer, select just the eight large, overlapping parcels. In the other, select the opposite: **PIN IS NULL OR PIN NOT IN ('00519500200405', '27043600414500', '1679500000', '7922690000', '8678800000', '9540500000')**

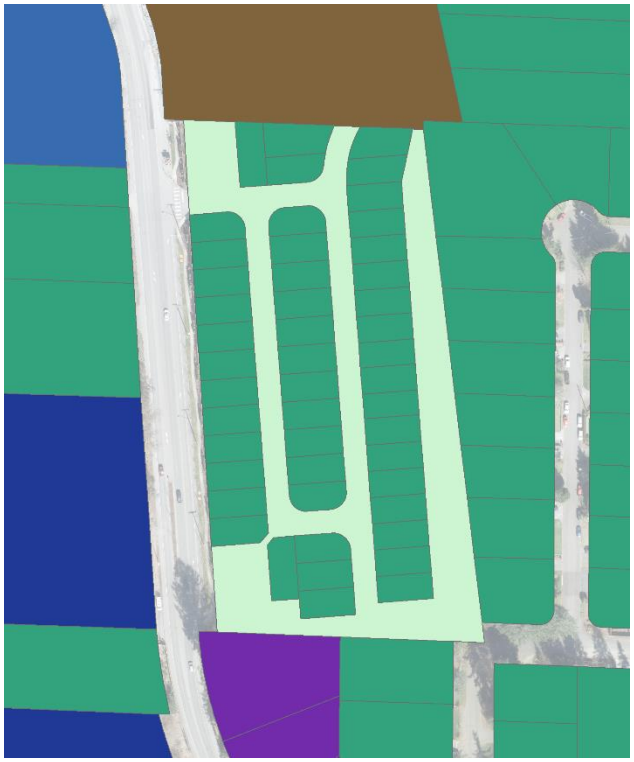
1.8. Use the Erase tool to erase *from* the eight large, overlapping parcels (place them as the Input Layer). Place the output in the file geodatabase **WatershedPrioritization_Analysis_Prep.gdb**.



1.9. Unselect all features and remove the second "assisting" layer used in the Erase tool.

1.10. Start editing the original land use layer. Select the eight large, overlapping parcels one more time and delete them. Select and copy their "erased versions" from the result layer of the Erase tool and paste them back into the land use layer you're editing.

a. PIN = '00519500200405' was entirely covered by smaller parcels, so it was completely erased during the Erase tool and so didn't have a replacement after being deleted.



1.11. Save your edits and stop editing.

1.12. Copy the **PW_Watershed_Analysis_Landuse_Parcel_2019_09** feature class back into the analysis (non-prep) geodatabase and name it **PW_Watershed_Analysis_Landuse_Parcel_2019_09_NoOverlap**. Your land use layer should be ready to use in the final analysis.

Part 2. Impervious Surface Preparation

2.1. Start with the file geodatabase **Impervious_2019_11_21.gdb**.

2.2. Use the Select tool to extract just impervious features from all six feature classes using the following SQL query: **Impervious IS NULL OR Impervious <> 2**. Place the outputs in the file geodatabase **WatershedPrioritization_Analysis_Prep.gdb** and name the new feature classes **Impervious_[featureclass]**.

2.3. Open the file geodatabase **Buildings_2019_11_21.gdb**.

2.4. Copy the **Buildings** feature class to **WatershedPrioritization_Analysis_Prep.gdb**. All buildings are considered impervious, so you can rename the feature class **Impervious_Buildings** after copying as well.

2.5. Open the file geodatabase **Streets_2019_11_21.gdb**.

2.6. Copy the **Sidewalks** feature class to **WatershedPrioritization_Analysis_Prep.gdb**. All sidewalks are considered impervious, so you can rename the feature class **Impervious_Sidewalks** after copying as well.

2.7. Use the Select tool to extract just impervious features from **EdgeOfPavement** using the following SQL query: **Pervious IS NULL OR Pervious <> 1**. Place the output in the file geodatabase **WatershedPrioritization_Analysis_Prep.gdb** and name the new feature class **Impervious_EdgeOfPavement**.

2.8. Add the following layers to ArcMap:

- **a.** From **Streets_2019_11_21.gdb**, add **StreetNetwork**
- **b.** From **WatershedPrioritization_Analysis.gdb**, add **CityLimit**.
- **c.** From **WatershedPrioritization_Analysis_Prep.gdb**, add **Impervious_EdgeOfPavement**.

2.9. Symbolize **StreetNetwork** by the field *Map_Carto_Class*. It helps to choose different colors for each value AND a size gradient, with "4" being the largest. This will help you later during editing.

2.10. Add a text field to **Impervious_EdgeOfPavement** called *AnalysisCategory* (length = 50 is fine).

We'll use the *AnalysisCategory* field to categorize road pavement into three classes of roads: Limited Access, Arterials, and Local Roads.

2.11. Start editing the **Impervious_EdgeOfPavement** layer.

2.12. Use Select By Location to select all **Impervious_EdgeOfPavement** features that intersect the **CityLimit**. Switch selection so only features *outside* the city are selected and calculate the *AnalysisCategory* field to "Outside Bothell".

2.13. Clear your selection.

The process from this point has a significant manual component that is somewhat subjective. Generally, we will follow the *Map_Carto_Class* field in **StreetNetwork**, which has five categories. To put pavement into only three categories, we'll run a series of Select By Locations, followed by manual corrections.

2.14. Use Select By Attributes to select all *potentially* limited access roads in **StreetNetwork** using the following SQL query: **Map_Carto_Class IN (3, 4)**

2.15. Use Select By Location to select all **Impervious_EdgeOfPavement** features that intersect those selected roads in **StreetNetwork**.

2.16. Use Select By Attributes to unselect any **Impervious_EdgeOfPavement** features that already have a value in *AnalysisCategory*. Calculate *AnalysisCategory* in the remaining features to "Limited Access".

2.17. Clear your selection.

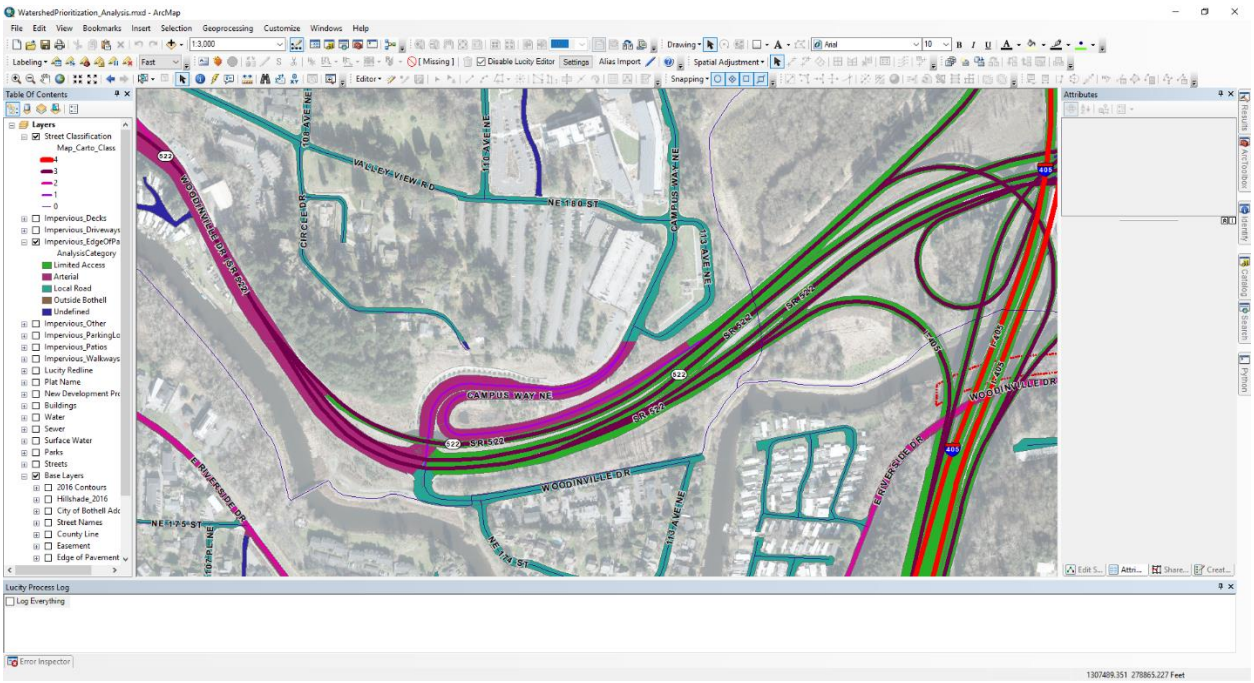
2.18. Symbolize **Impervious_EdgeOfPavement** by the field *AnalysisCategory*. Add three additional categories to the list of symbols already in there to get the following:

- **a.** Limited Access
- **b.** Arterial
- **c.** Local Road
- **d.** Outside Bothell
- **e.** Undefined

2.19. Looking at the features now labeled "Limited Access", identify any features that appear to be along *non-limited access* highways and edit their *AnalysisCategory* to "Arterial".

- **a.** A good place to start is the north-south running Bothell Way, Bothell-Everett Highway, and SR 527 and Woodinville Dr (SR 522) WEST of the I-405 interchange (pictured below).

- **b.** Overpasses and underpasses are another situation where sometimes roads can be misidentified. Just do your best with these as sometimes the pavement features themselves aren't split nicely along road type boundaries.



Woodinville Dr (SR 522) to the WEST of the I-405 interchange is NOT limited access, but the ramps and the portion EAST of the interchange are. This screenshot also includes some additional categories defined later in this section.

2.20. Save your edits.

2.21. Use Select By Attributes to select all arterial roads in **StreetNetwork** using the following SQL query: **Map_Carto_Class IN (1, 2, 3)**

2.22. Use Select By Location to select all **Impervious_EdgeOfPavement** features that intersect those selected roads in **StreetNetwork**.

2.23. Use Select By Attributes to <u>unselect</u> any **Impervious_EdgeOfPavement** features that already have a value in *AnalysisCategory*. Calculate *AnalysisCategory* in the remaining features to "Arterial".

2.24. Clear your selection.

2.25. Use Select By Attributes to select all local roads in **StreetNetwork** using the following SQL query: **Map_Carto_Class = 0**

2.26. Use Select By Location to select all **Impervious_EdgeOfPavement** features that intersect those selected roads in **StreetNetwork**.

2.27. Use Select By Attributes to <u>unselect</u> any **Impervious_EdgeOfPavement** features that already have a value in *AnalysisCategory*. Calculate *AnalysisCategory* in the remaining features to "Local Road".

2.28. Clear your selection.

2.29. Use Select By Attributes to select any **Impervious_EdgeOfPavement** features that don't have a value in *AnalysisCategory*. Calculate *AnalysisCategory* in these features to "Undefined".

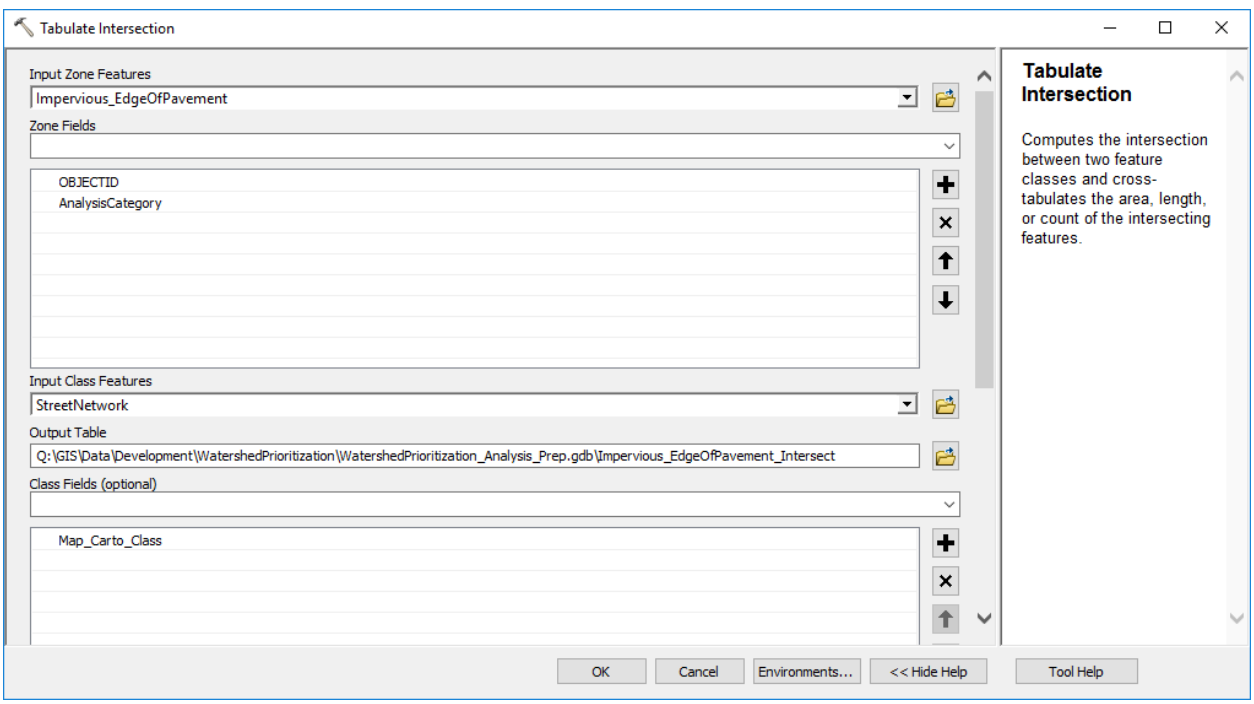
2.30. Clear your selection.

2.31. Save your edits.

Finally, we'll manually edit **Impervious_EdgeOfPavement** features, mainly to deal with cases where roads of a different class clipped the edge of pavement features, typically at intersections, causing them to be misidentified. The most common case is "Local Roads" being misidentified as "Arterials".

2.32. To help guide your search for misidentified pavement features, use the Tabulate Intersection tool with the following parameters:

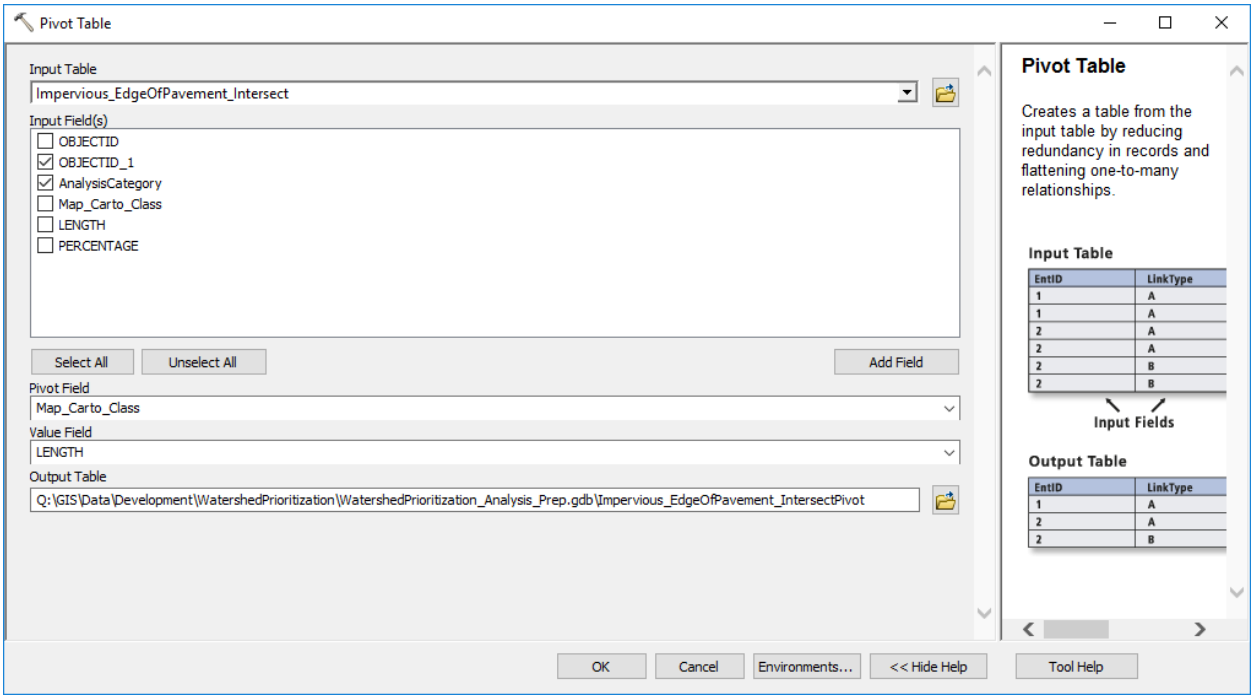
a. **Input Zone Features:** Impervious_EdgeOfPavement
b. **Zone Fields:** OBJECTID, AnalysisCategory
c. **Input Class Features:** StreetNetwork
d. **Output Table:** *[Put in analysis prep geodatabase]*
e. **Class Fields:** Map_Carto_Class



2.33. This got us partway there. To simplify searching for misidentified pavement features, use the Pivot Table tool with the following parameters:

a. **Input Table:** *[The table you just created]*
b. **Input Fields:** OBJECTID_1 *["_1" is the original source OID]*, AnalysisCategory

c. **Pivot Fields:** Map_Carto_Class
d. **Value Field:** LENGTH
e. **Output Table:** *[Put in analysis prep geodatabase]*



Use this table to help you identify where pavement features might be misidentified. It can be cumbersome to keep bouncing back and forth grabbing OBJECTIDs to search (remember to use the second OBJECTID column in the Pivot Table). It's quicker (and perhaps advisable in addition) to do a quick sweep through the City and identify misidentified pavement visually through your symbology.

2.34. To the best of your ability, manually edit **Impervious_EdgeOfPavement** until all pavement features are the correct *AnalysisCategory*.

a. You might also encounter pavement features that could be split if they include large portions that should fall in a different *AnalysisCategory* or "Undefined" (e.g. driveways). I did this in a couple places.

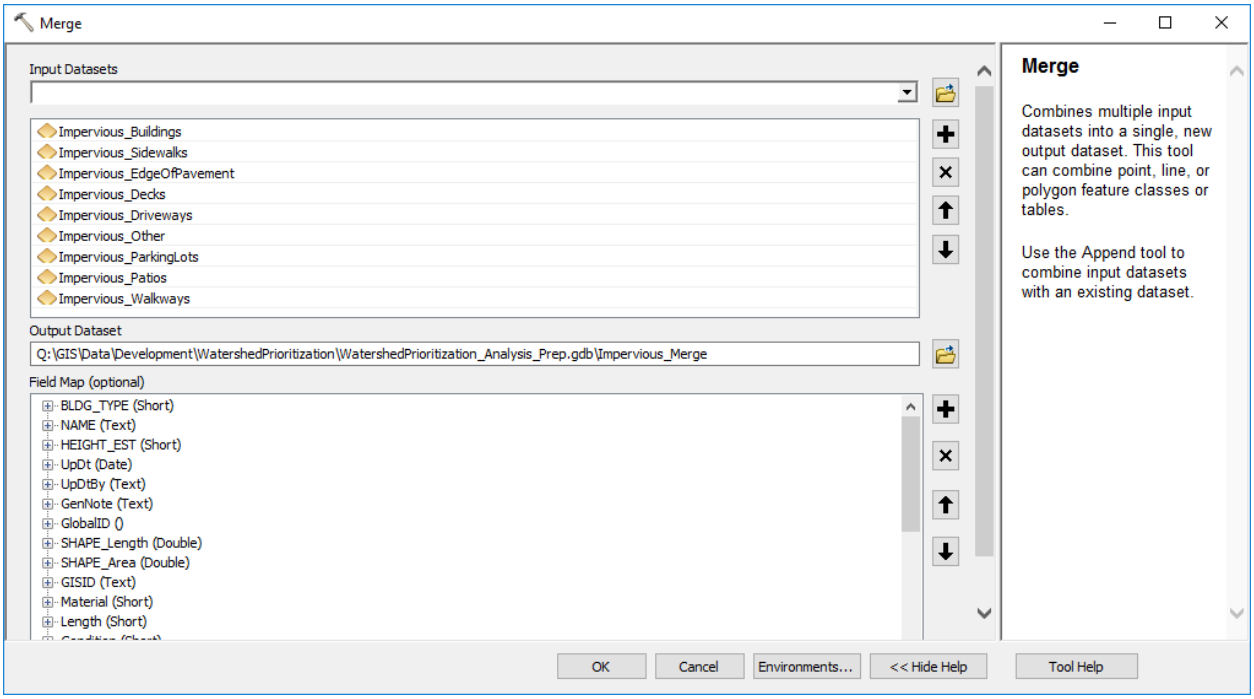
b. You can also self-intersect **Impervious_EdgeOfPavement** if you'd like to resolve some minor cases of pavement feature overlap.

2.35. When you're done, save your edits and stop editing.

Finally, we'll merge all impervious surface feature classes to get a total impervious surface.

2.36. Use the Merge tool to merge the following feature classes (place the output in the analysis prep geodatabase):

a. Impervious_Buildings
b. Impervious_Sidewalks
c. Impervious_EdgeOfPavement *[after you're done editing]*
d. Impervious_Decks
e. Impervious_Driveways
f. Impervious_Other
g. Impervious_ParkingLots
h. Impervious_Patios
i. Impervious_Walkways



2.37. Use the Dissolve tool on the output feature class to merge all features (gets rid of any overlap). Place the output in the analysis prep geodatabase and call it **Impervious_Dissolve**.

2.38. Copy **Impervious_EdgeOfPavement** and **Impervious_Dissolve** to the analysis (non-prep) geodatabase. Your impervious surface layers should now be ready to use in the final analysis.

Part 3. Physical Geography Preparation

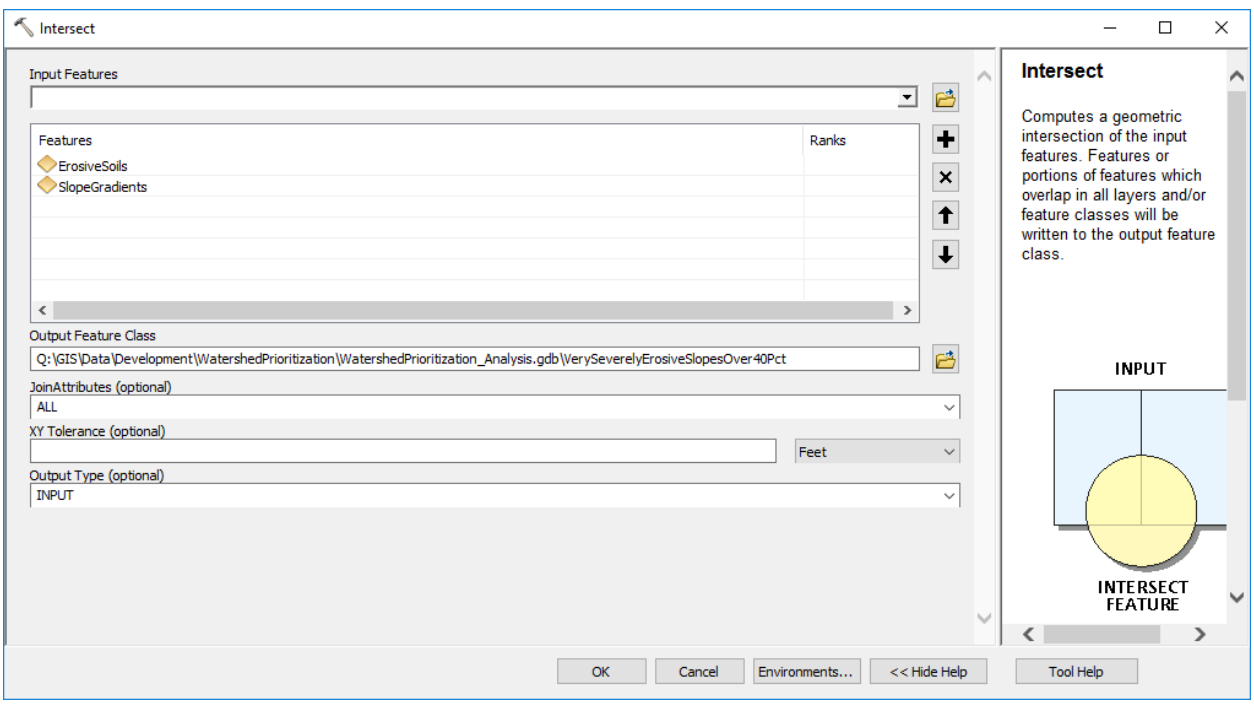
3.1. Start with the file geodatabase **WatershedPrioritization_Analysis.gdb** (no prep).

3.2. Add the **ErosiveSoils** and **SlopeGradients** feature classes to your ArcMap document.

3.3. Use Select By Attributes to select the most highly erosive soils in **ErosiveSoils** using the following SQL query: **EROSIVITY = 'Class V'**

3.4. Use Select By Attributes to select the steepest slopes in **SlopeGradients** using the following SQL query: **SlopePercent = '40%+'**

3.5. *With both selections still active*, use the Intersect tool to intersect **ErosiveSoils** and **SlopeGradients**. This will give "very severely erosive soils on slopes over 40%", so we'll name it **VerySeverelyErosiveSlopesOver40Pct**. Since this is the only preparation we will need to do on these layers, you can place the output in the file geodatabase **WatershedPrioritization_Analysis.gdb** (no prep).



3.6. Clear your selections. Your physical geography layer should now be ready to use in the final analysis.

Part 4. Storm Layers Preparation

4.1. Start with the file geodatabase **Surfacewater_2019_11_21.gdb**.

4.2. Use the Select tool to extract active assets from the following feature classes using the following SQL query: **Status IS NULL OR Status < 950**. I put them in the analysis prep geodatabase and called them **Active_[featureclass]**.

- **a.** dBioretention
- **b.** dBioSwales
- **c.** dCatchBasins
- **d.** dCulverts
- **e.** dDetentionPonds
- **f.** dDitches
- **g.** dPerviousPavement
- **h.** dPipes
- **i.** dRivers_and_Streams
- **j.** dTreeBoxes
- **k.** dTrenches
- **l.** dVaults

4.3. Also copy the **dWetlands** and **dWetlandBuffer** feature classes to the analysis prep geodatabase. All are considered active, so you can rename them **Active_[featureclass]** after copying as well.

From now on, work ONLY with the copies you've made in the analysis prep geodatabase.

4.4. In ArcMap, add **Active_dCatchBasins** (if needed) and Select By Attributes features with the following SQL query: **((QualityControl < 830 AND QualityControl NOT IN (120, 140, 180)) OR FilterType < 830) AND (FilterCount IS NULL OR FilterCount < 1)**

4.5. Calculate *FilterCount* for all selected catch basins to 1. Clear your selection.

4.6. In **Active_dVaults**, select features with the following SQL query: **FilterType < 830 AND (FilterCount IS NULL OR FilterCount < 1)**

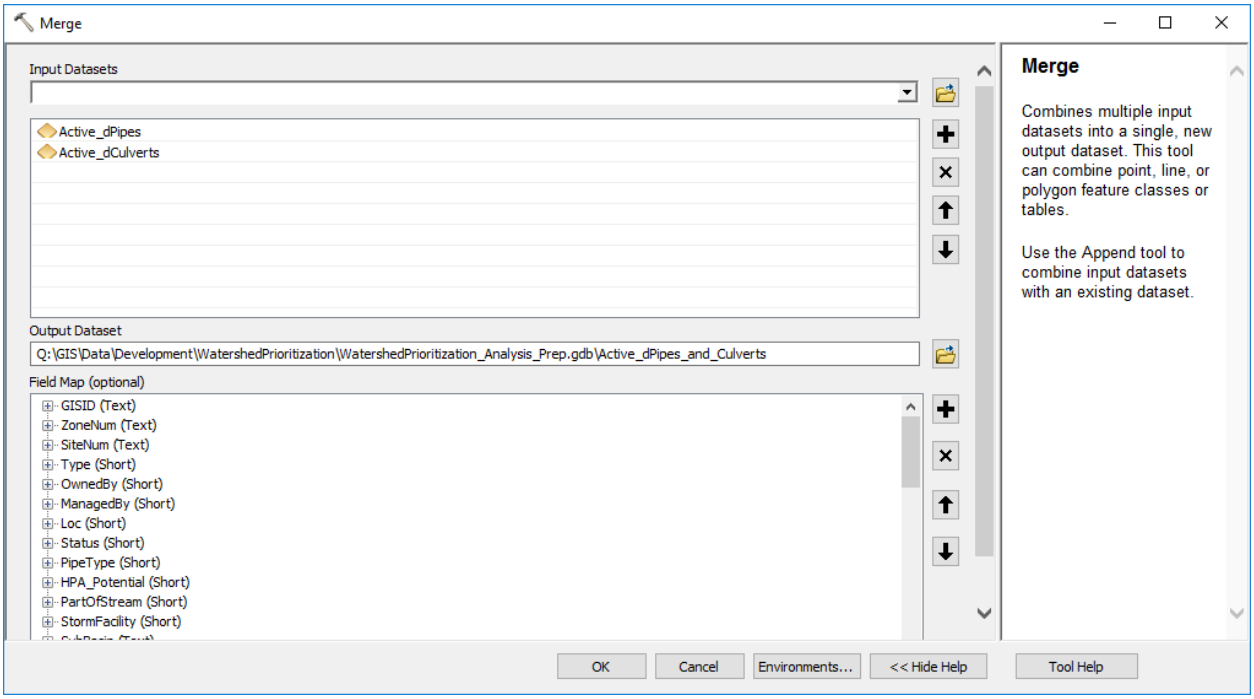
4.7. Calculate *FilterCount* for all selected vaults to 1. Clear your selection. *Save edits if editing!*

4.8. Use the Select tool to extract control structures from **Active_dCatchBasins** using the following SQL query: **FlowControl = 100**. Call it **Active_dControlStructures**.

4.9. Start editing **Active_dCatchBasins** and delete the control structures you just exported out of this table (using same SQL query as the previous step). Clear your selection and stop editing.

4.10. Use the Select tool to extract detention pipes from **Active_dPipes** using the following SQL query: **PipeType = 161**. Call it **Active_dDetentionPipes**. (We won't delete these because they're counted in a different category.)

4.11. Use the Merge tool to combine **Active_dPipes** and **Active_dCulverts** into **Active_dPipes_and_Culverts**.



4.12. Use the Merge tool to combine **Active_dWetlands** and **Active_dWetlandBuffer** into **Active_dWetlands_and_Buffers**.

4.13. Copy *all* of the **Active_...** feature classes into the analysis (non-prep) geodatabase EXCEPT:

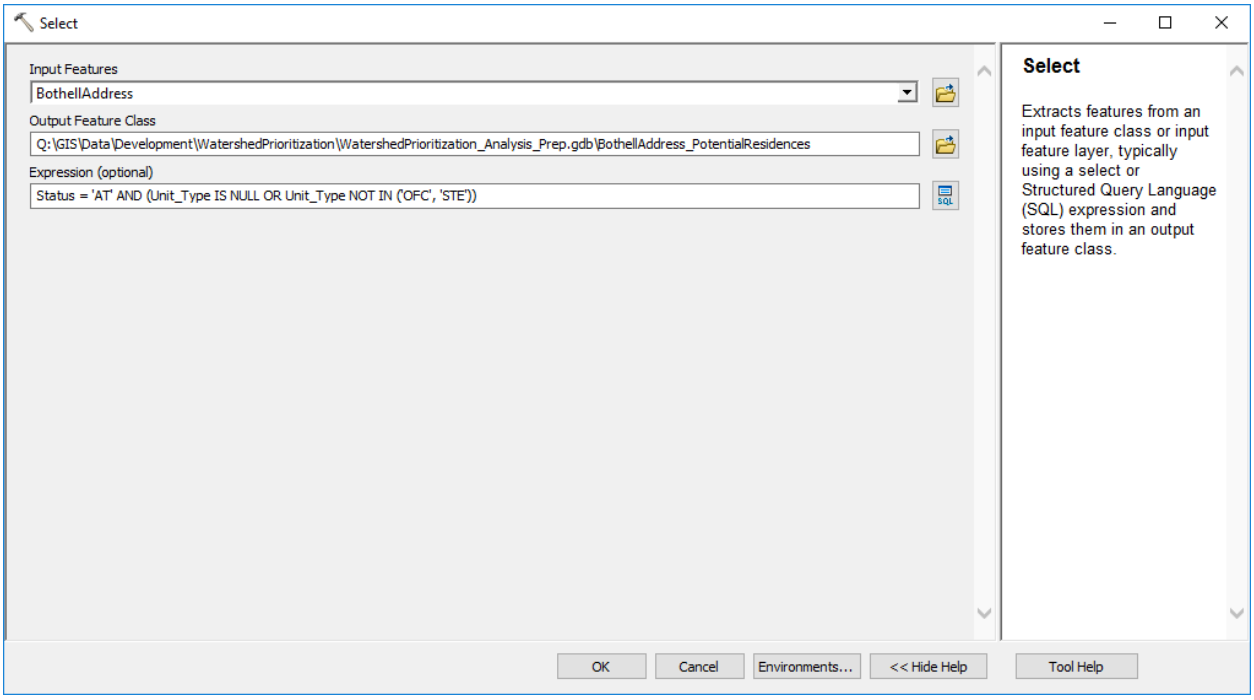
- **a.** Active_dCulverts
- **b.** Active_dPipes
- **c.** Active_dWetlandBuffer

4.14. Your storm layers should now be ready to use in the final analysis.

Part 5. Population Density Estimation Preparation

5.1. Start with the file geodatabase **WatershedPrioritization_Analysis.gdb** (no prep).

5.2. Use the Select tool to extract potential residences from **BothellAddress** using the following SQL query: **Status = 'AT' AND (Unit_Type IS NULL OR Unit_Type NOT IN ('OFC', 'STE'))**. Place the output in **WatershedPrioritization_Analysis_Prep.gdb** and name the new feature class **BothellAddress_PotentialResidences**.

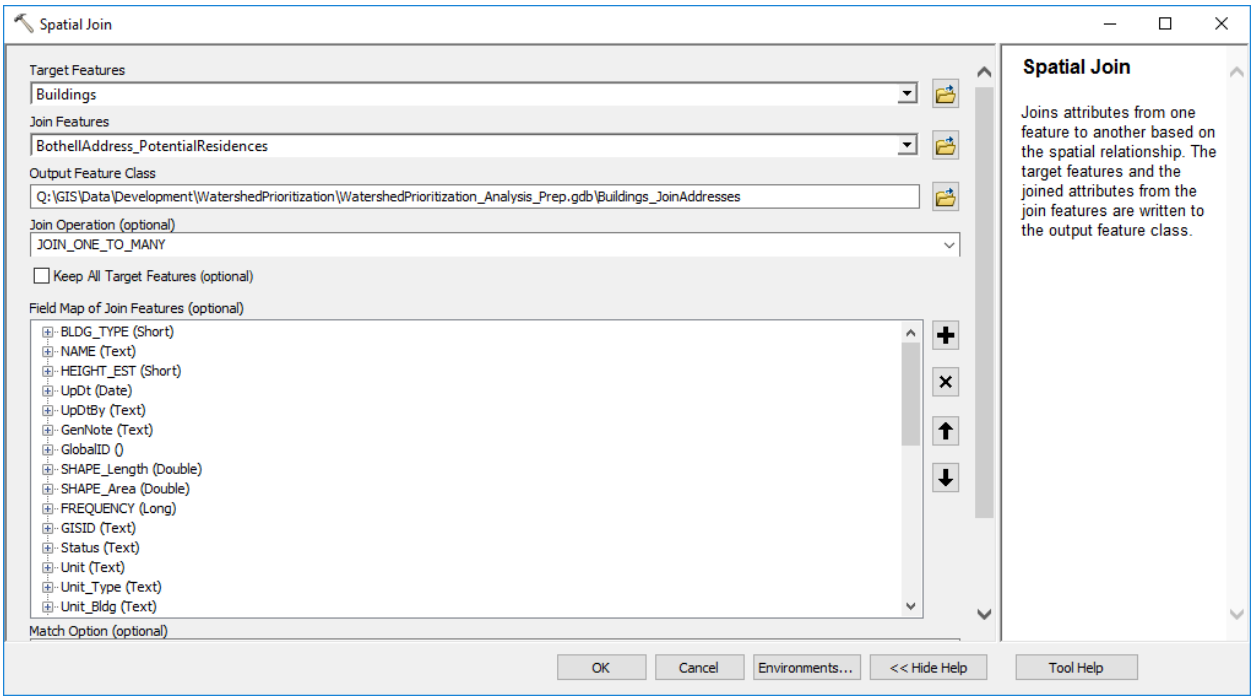


5.3. Open the file geodatabase **Buildings_2019_11_21.gdb**.

5.4. Copy the **Buildings** feature class and paste it into **WatershedPrioritization_Analysis_Prep.gdb**. Add this newly created copy to your ArcMap.

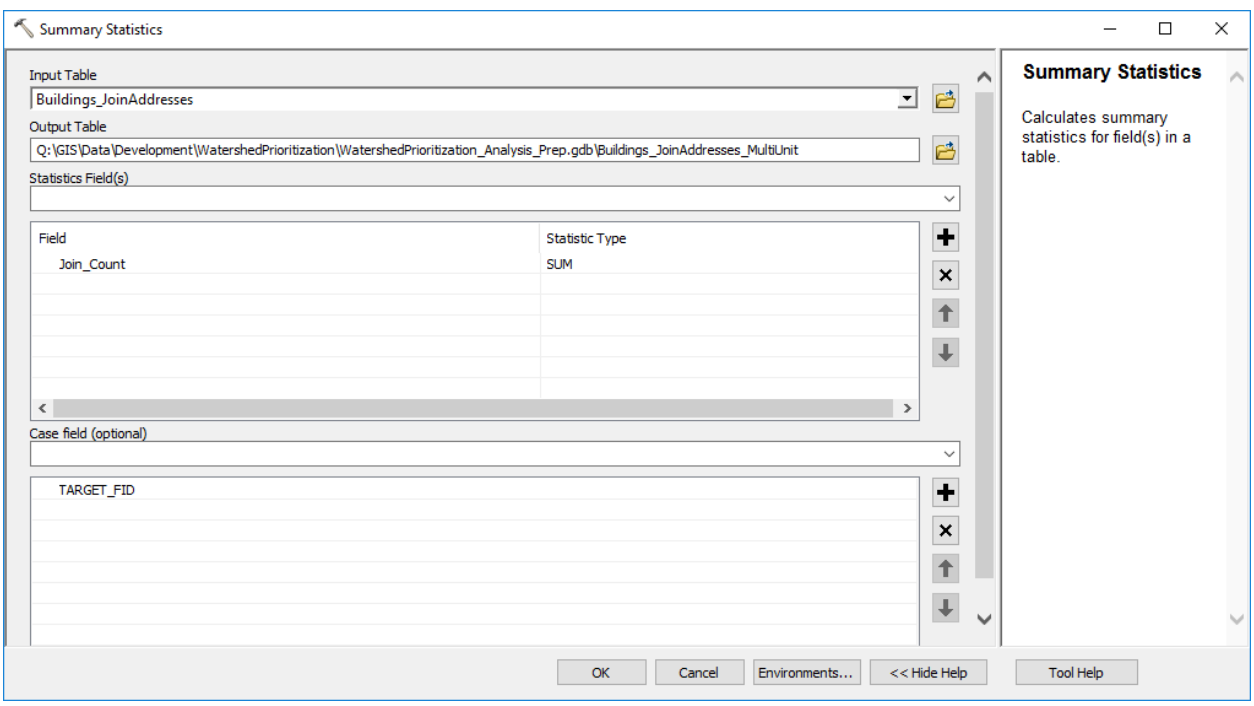
5.5. Use the Spatial Join tool on **Buildings** to join all fields from the **BothellAddress_PotentialResidences** feature class to any building that <u>intersects</u> an address. Place the output in **WatershedPrioritization_Analysis_Prep.gdb** and name the new feature class **Buildings_JoinAddresses**. Use the following parameters.

- **a. Target Features:** Buildings
- **b. Join Features:** BothellAddress_PotentialResidences
- **c. Output Feature Class:** *[Put in analysis prep geodatabase]*
- **d. Join Operation:** JOIN_ONE_TO_MANY
- **e. Keep All Target Features:** *[Unchecked]*
- **f.** *[Leave the rest of the parameters as defaults. E.g. Match Option = 'INTERSECT']*



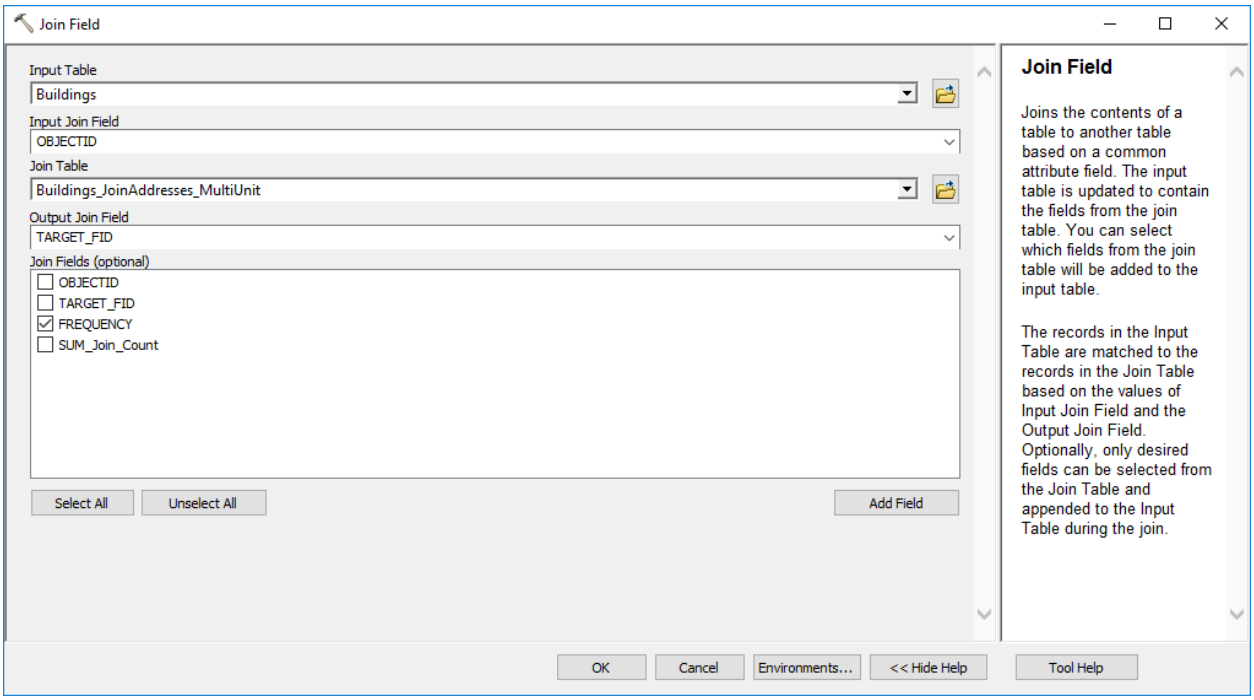
5.6. Use the Summary Statistics tool on **Buildings_JoinAddresses** with the following parameters:

- **a. Input Table:** Buildings_JoinAddresses
- **b. Output Table:** *[Put in analysis prep geodatabase]*
- **c. Statistics Field(s):** Join_Count [SUM]
- **d. Case field:** TARGET_FID



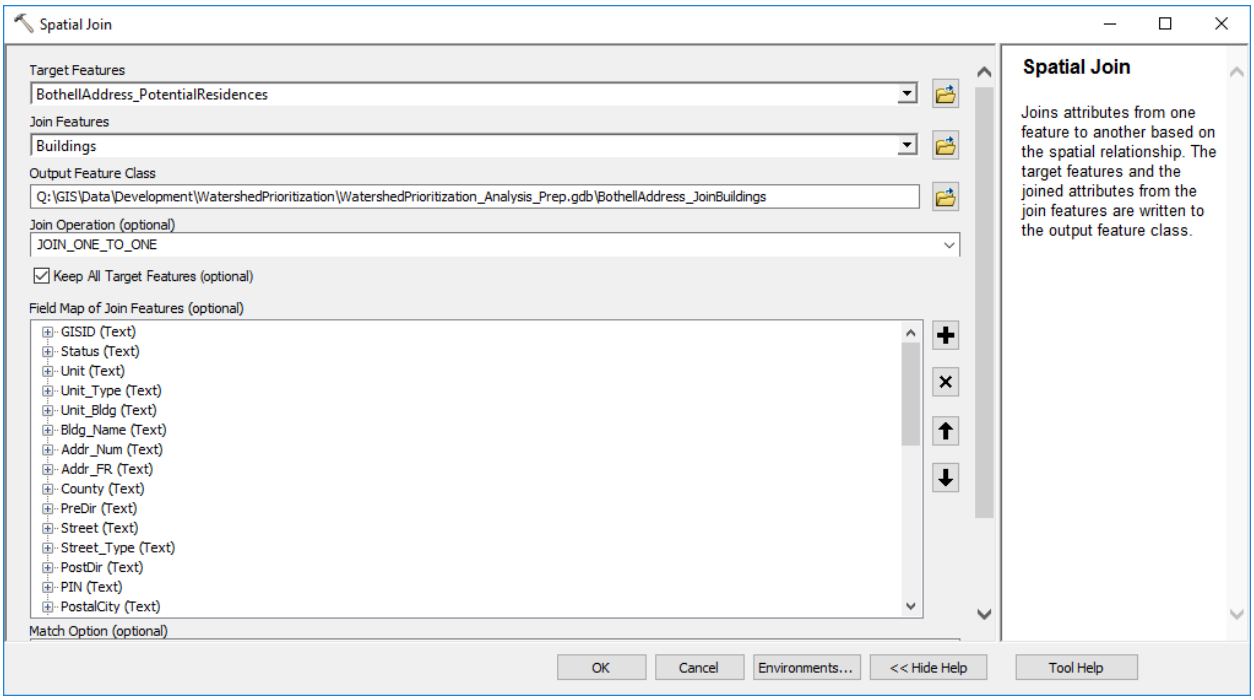
5.7. Use the Join Field tool to join the FREQUENCY column back to **Buildings** *[in analysis prep]* with the following parameters:

- **a. Input Table:** Buildings *[the one in the analysis prep geodatabase]*
- **b. Input Join Field:** OBJECTID
- **c. Join Table:** [the summary table you just created]
- **d. Output Join Field:** TARGET_FID
- **e. Join Fields:** FREQUENCY



5.8. Use the Spatial Join tool on **BothellAddress_PotentialResidences** to join all fields from the **Buildings** feature class (that you just joined FREQUENCY to) to any address that <u>intersects</u> a building. Place the output in **WatershedPrioritization_Analysis_Prep.gdb** and name the new feature class **BothellAddress_JoinBuildings**. Use the following parameters.

- **a. Target Features:** BothellAddress_PotentialResidences
- **b. Join Features:** Buildings *[the one you just joined FREQUENCY to]*
- **c. Output Feature Class:** *[Put in analysis prep geodatabase]*
- **d. Join Operation:** JOIN_ONE_TO_ONE
- **e. Keep All Target Features:** *[Checked on]*
- **f.** *[Leave the rest of the parameters as defaults. E.g. Match Option = 'INTERSECT']*



Building footprints are typically added later than addresses. For newer addresses that don't have building footprints in GIS yet, we'll use the landuse layer and other information to designate a few more likely residences before continuing.

5.9. Use Select By Attributes on **BothellAddress_JoinBuildings** with the following SQL query: **Bldg_Name = 'LAKE PLEASENT RV PARK'**. Use Calculate Field on these 189 records to set BLDG_TYPE to 210 [Mobile Home] and FREQUENCY to 1. Clear your selection.

5.10. Open the file geodatabase **WatershedPrioritization_Analysis.gdb** (no prep) and add the **PW_Watershed_Analysis_Landuse_Parcel_2019_09** feature class to your ArcMap. (You can alternatively add the "_NoOverlap" version from analysis prep if you've created it already.)

5.11. Use Select By Attributes to select any land use features that match the SQL query: **Landuse = 'Single Family Residential (1a)'**.

5.12. Use Select By Location to select any **BothellAddress_JoinBuildings** features that intersect the *selected features* of **PW_Watershed_Analysis_Landuse_Parcel_2019_09**.

5.13. Use Select By Attributes to remove from the selection on **PW_Watershed_Analysis_Landuse_Parcel_2019_09** any features that already have a BLDG_TYPE.

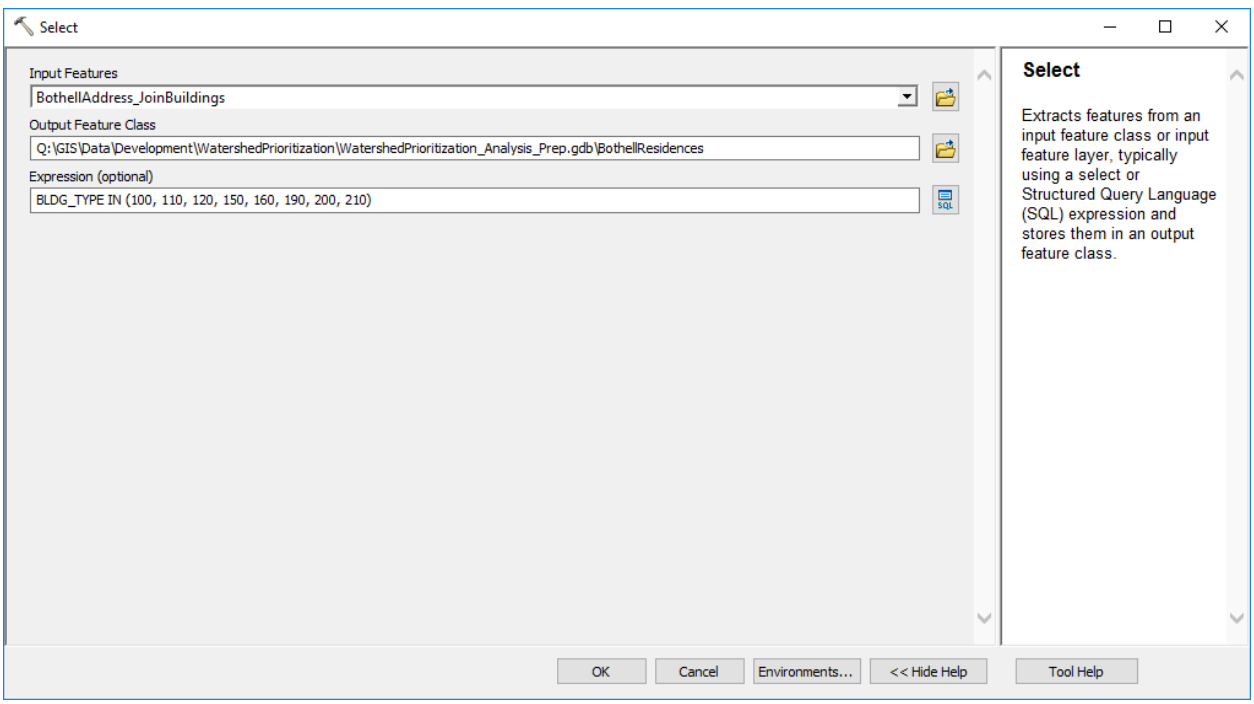
5.14. Use Calculate Field on these 284 records to set BLDG_TYPE to 100 [Single Family] and FREQUENCY to 1. Clear your selections.

5.15. Repeat the previous four steps to define two more cases of likely residences.

a. **Landuse = 'Multi-Family Residential (1b)'**
   This should select 334 records in **BothellAddress_JoinBuildings** at the end.
   Calculate BLDG_TYPE to 120 [Apartment] and FREQUENCY to 5.
b. **Landuse = 'Mixed Use (3a)'**
   This should select 0 records in **BothellAddress_JoinBuildings** at the end, so no need to run any field calculations. If there are any, it would be:
   Calculate BLDG_TYPE to 200 [Mixed Use] and FREQUENCY to 5.

5.16. Clear your selections!

5.17. Use the Select tool to extract likely residences from **BothellAddress_JoinBuildings** using the following SQL query: **BLDG_TYPE IN (100, 110, 120, 150, 160, 190, 200, 210)**. Place the output in **WatershedPrioritization_Analysis_Prep.gdb** and name the new feature class **BothellResidences**.



5.18. Add a double field to **BothellResidences** called "Population".

5.19. Use Calculate Field to calculate Population for every feature in **BothellResidences** using one of the following methods.

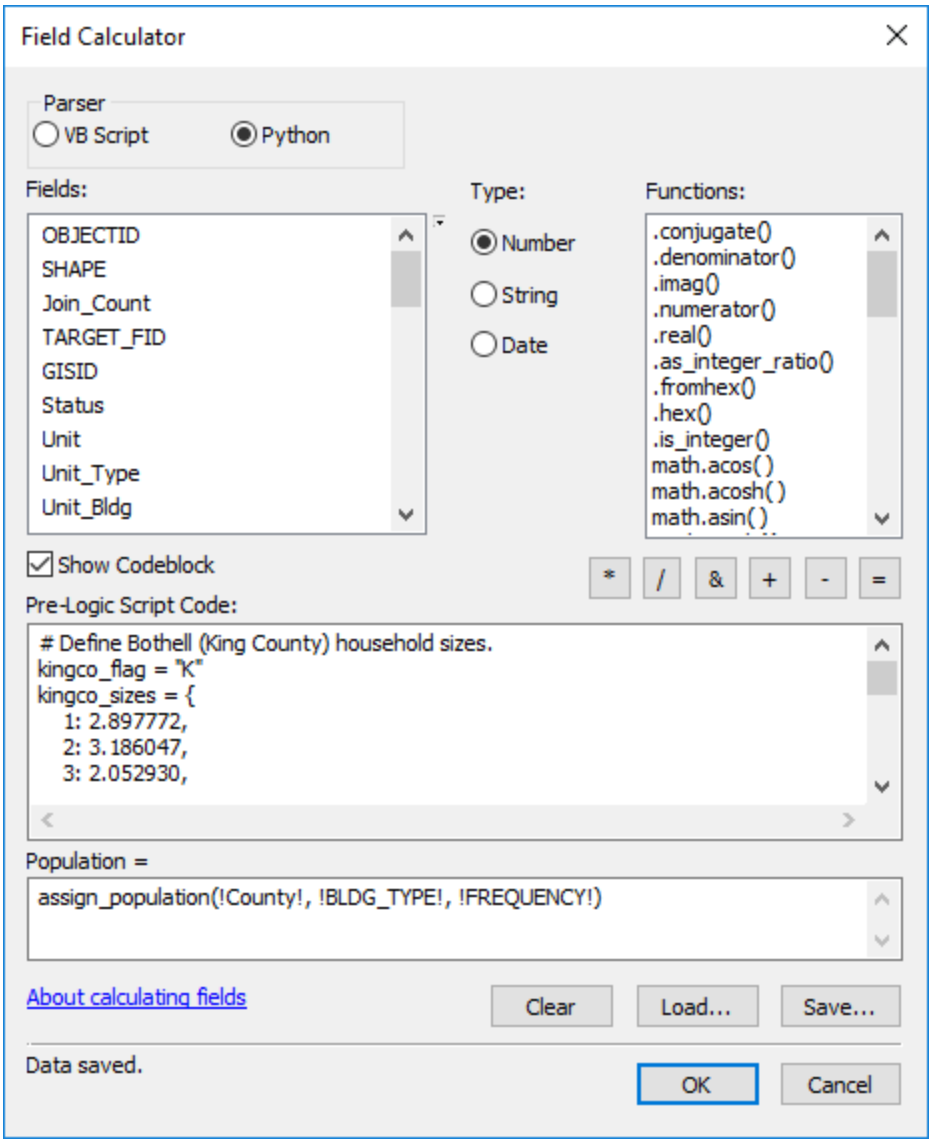
a. Import the code from PopulationFieldCalculatorExpression.cal. *Make sure* to switch the Parser to "Python" (9.3 if it asks).
b. You can copy directly from Word (not PDF) the following code pieces (next page). *Make sure* to still switch the Parser to "Python" (9.3 if it asks):

Code Block (can directly copy from Word, not PDF):

```
# Define Bothell (King County) household sizes.
kingco_flag = "K"
kingco_sizes = {
    1: 2.897772,
    2: 3.186047,
    3: 2.052930,
    4: 2.052930,
    }
kingco_5plus = 1.980000
kingco_manuf = 1.934804

# Define Bothell (Snohomish County) household sizes.
snoco_flag = "S"
snoco_sizes = {
    1: 2.960115,
    2: 2.048193,
    3: 2.004425,
    4: 2.004425,
    }
snoco_5plus = 1.681847
snoco_manuf = 1.783163

# Define manufactured/mobile home type codes.
manuf_types = {190, 210}


# Define Field Calculator function.


def assign_population(county, bldg_type, unit_count):
    """Return average household size of address with certain characteristics."""
    if county == kingco_flag:
        if bldg_type in manuf_types:
            return kingco_manuf
        else:
            return kingco_sizes.get(unit_count, kingco_5plus)
    elif county == snoco_flag:
        if bldg_type in manuf_types:
            return snoco_manuf
        else:
            return snoco_sizes.get(unit_count, snoco_5plus)
    else:
        return None
```

Expression:

```
assign_population(!County!, !BLDG_TYPE!, !FREQUENCY!)
```



5.20. Once the Field Calculator has completed and you've verified that Population values have been added, save edits (if in an edit session).

5.21. Copy the **BothellResidences** feature class to the analysis (non-prep) geodatabase. Your population density estimation layer should now be ready to use in the final analysis.

Part 6. Final Analysis

6.1. Start with the file geodatabase **WatershedPrioritization_Analysis.gdb** (no prep). It should look identical to the following screenshot with **30 total feature classes**. If it doesn't, the differences should indicate which of the four previous sections you should review or repeat. You can also create a copy of **WatershedPrioritization_Analysis_Staged.gdb** to skip the preparation steps and jump in right here.

Contents | Preview | Description

Name	Type
Active_dBioretention	File Geodatabase Feature Class
Active_dBioSwales	File Geodatabase Feature Class
Active_dCatchBasins	File Geodatabase Feature Class
Active_dControlStructures	File Geodatabase Feature Class
Active_dDetentionPipes	File Geodatabase Feature Class
Active_dDetentionPonds	File Geodatabase Feature Class
Active_dDitches	File Geodatabase Feature Class
Active_dPerviousPavement	File Geodatabase Feature Class
Active_dPipes_and_Culverts	File Geodatabase Feature Class
Active_dRivers_and_Streams	File Geodatabase Feature Class
Active_dTreeBoxes	File Geodatabase Feature Class
Active_dTrenches	File Geodatabase Feature Class
Active_dVaults	File Geodatabase Feature Class
Active_dWetlands	File Geodatabase Feature Class
Active_dWetlands_and_Buffers	File Geodatabase Feature Class
Bothell_Trees	File Geodatabase Feature Class
BothellAddress	File Geodatabase Feature Class
BothellResidences	File Geodatabase Feature Class
CityLimit	File Geodatabase Feature Class
dMonitoringBasins	File Geodatabase Feature Class
dMonitoringBasins_BothellCity	File Geodatabase Feature Class
dMonitoringBasins_BothellMUGA	File Geodatabase Feature Class
dMonitoringBasins_TotalCalc	File Geodatabase Feature Class
dSubBasins_New	File Geodatabase Feature Class
Impervious_Dissolve	File Geodatabase Feature Class
Impervious_EdgeOfPavement	File Geodatabase Feature Class
PW_Watershed_Analysis_Landuse_Parcel_2019_09	File Geodatabase Feature Class
PW_Watershed_Analysis_Landuse_Parcel_2019_09_NoOverlap	File Geodatabase Feature Class
RightOfWay	File Geodatabase Feature Class
VerySeverelyErosiveSlopesOver40Pct	File Geodatabase Feature Class

6.2. Open a fresh ArcMap document and add **dMonitoringBasins_BothellCity** to your map. Make sure you select the correct one of these!

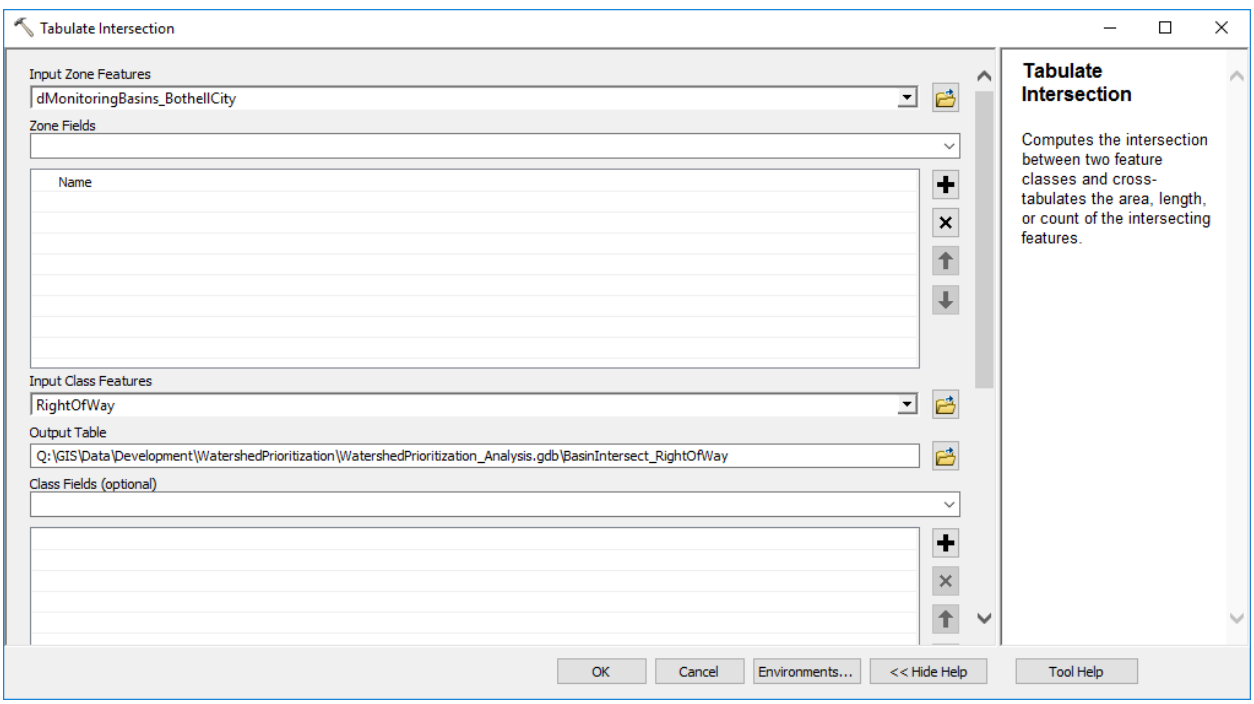
Final Part 1. Simple Overlap Analysis

6.3. Add the following feature classes from your analysis geodatabase to your map. You may want to pause drawing since the map display isn't particularly relevant now.

- **a.** Bothell_Trees
- **b.** BothellResidences
- **c.** CriticalDrainage_Parcels
- **d.** Impervious_Dissolve
- **e.** RightOfWay
- **f.** VerySeverelyErosiveSlopesOver40Pct

6.4. For each feature class, use the Tabulate Intersection tool to determine overlap with each monitoring basin.

- **a.** **Input Zone Features:** dMonitoringBasins_BothellCity
- **b.** **Zone Fields:** Name
- **c.** **Input Class Features:** *[current layer]*
- **d.** **Class Fields:** *[blank]*
- **e.** **Sum Fields:**
  - i. BothellResidences ➔ Population
  - ii. *[All Other Layers]* ➔ *[blank]*
- **f.** **Output Table:** *[Place in the analysis geodatabase you're currently working in]*



BasinIntersect_RightOfWay

OBJECTID *	Name	AREA	PERCENTAGE
1	Horse Creek	3510948.562227	12.98667
2	Juanita Creek	3614761.072284	17.32108
3	Little Bear Creek	1076298.658177	19.240975
4	Little Swamp Creek	3124082.863965	11.355145
5	Lower North Creek	9282318.123047	13.89592
6	Lower Sammamish River	6157325.020863	11.937388
7	Parr Creek	3618179.324951	11.495059
8	Perry Creek	5328378.636706	17.409533
9	Swamp Creek	1595776.272035	11.534497
10	Upper North Creek	9851883.46323	18.060729
11	Upper Sammamish River	12007832.463896	23.315289

6.5. Select all rows in the output table and copy them to a blank Excel spreadsheet.

6.6. Immediately copy the set of cells you just pasted and paste them again (somewhere not overlapping) with the Transpose option. This will turn the rows into columns and vice versa.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	OBJECTID	Name	AREA	PERCENTAGE			OBJECTID	1	2	3	4	5	6	7	8	9	10	11		
2	1	Horse Cre	3510949	12.98667			Name	Horse Cre	Juanita Cr	Little Bear	Little Swa	Lower Nor	Lower San	Parr Creek	Perry Cree	Swamp Cr	Upper Nor	Upper Sammamish River		
3	2	Juanita Cr	3614761	17.32108			AREA	3510949	3614761	1076299	3124083	9282318	6157325	3618179	5328379	1595776	9851883	12007832		
4	3	Little Bear	1076299	19.24098			PERCENTA	12.98667	17.32108	19.24098	11.35515	13.89592	11.93739	11.49506	17.40953	11.5345	18.06073	23.31529		
5	4	Little Swa	3124083	11.35515																
6	5	Lower Nor	9282318	13.89592																
7	6	Lower San	6157325	11.93739																
8	7	Parr Creek	3618179	11.49506																
9	8	Perry Cree	5328379	17.40953																
10	9	Swamp Cr	1595776	11.5345																
11	10	Upper Nor	9851883	18.06073																
12	11	Upper San	12007832	23.31529																
13																				

6.7. For all layers *EXCEPT* BothellResidences, copy the **PERCENTAGE** row and paste *JUST THE VALUES* into the corresponding row in your main analysis spreadsheet.

- **a.** Bothell_Trees ➔ Tree Canopy
- **b.** CriticalDrainage_Parcels ➔ Critical Drainage
- **c.** Impervious_Dissolve ➔ Impervious
- **d.** RightOfWay ➔ Right-of-Way
- **e.** VerySeverelyErosiveSlopesOver40Pct
  ➔ Very Severely Erosive Soils (Class V) on Slopes over 40%

6.8. For BothellResidences, go back to Excel and copy the **Bothell City Portion (sqmi)** row from the main analysis spreadsheet into your "scratch" spreadsheet and *make sure* the columns line up with your values. In a new row, divide each basin's **Population** by its square miles to get "people per square mile". Then, copy this new row as values (not formulas) into the corresponding row (**Population Density**) back in your main analysis spreadsheet.

NOTE: The columns (basin names) *should* line up with those in the analysis spreadsheet **IF** you haven't reordered the columns **AND** all basins had overlap with the particular layer. If some basins did NOT have overlap, then the *order* of the columns will still be correct, but you'll need to insert "0"s for any columns that are missing a value (triple-check the column names in this case!).

6.9. Repeat for each layer in the first step in this subsection.

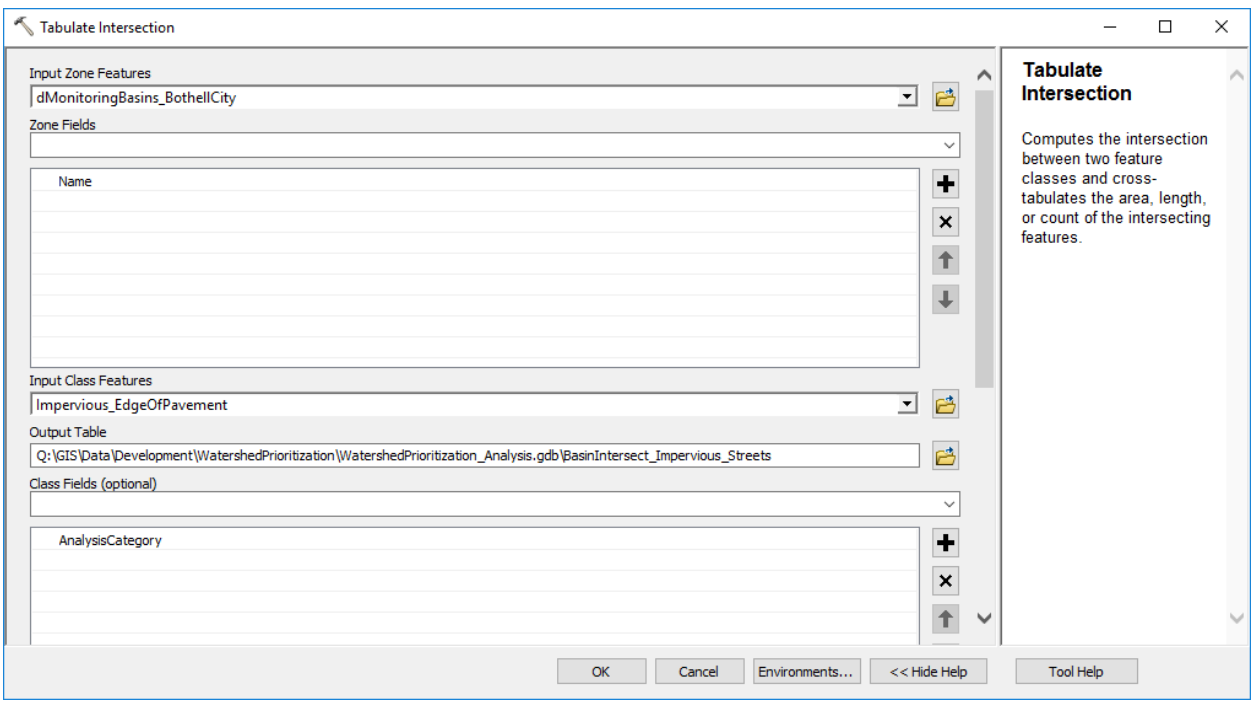
Final Part 2. Complex Overlap Analysis

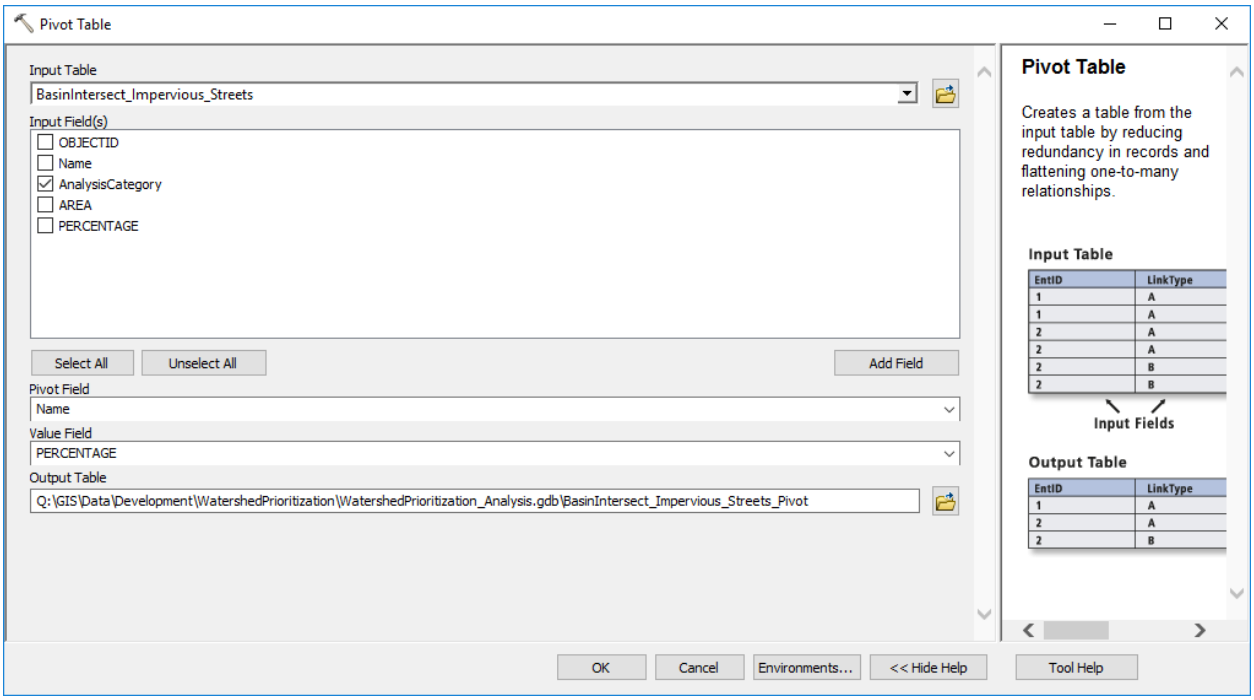
6.10. Add the following feature classes from your analysis geodatabase to your map. You may want to pause drawing since the map display isn't particularly relevant now.

- **a.** Impervious_EdgeOfPavement
- **b.** PW_Watershed_Analysis_Landuse_Parcel_2019_09**_NoOverlap**

6.11. For each feature class, use the Tabulate Intersection tool to determine overlap with each monitoring basin.

- **a. Input Zone Features:** dMonitoringBasins_BothellCity
- **b. Zone Fields:** Name
- **c. Input Class Features:** *[current layer]*
- **d. Class Fields:** *[Different for each layer]*
  - i. Impervious_EdgeOfPavement ➔ AnalysisCategory
  - ii. PW_Watershed_Analysis_Landuse_Parcel_2019_09_NoOverlap ➔ Landuse
- **e. Sum Fields:** *[blank]*
- **f. Output Table:** *[Place in the analysis geodatabase you're currently working in]*





6.12. Next, use the Pivot Table tool on the output table to convert the basins into columns.

- **a. Input Table:** *[Output table from previous step]*
- **b. Input Fields:** *[The unique field for each layer from the previous step (e.g. Landuse)]*
- **c. Pivot Field:** Name *[basin name]*
- **d. Value Field:** PERCENTAGE

6.13. Select all rows in the output table and copy them to a blank Excel spreadsheet (transpose is NOT needed this time).

BasinIntersect_Impervious_Streets_Pivot

OBJECTID *	AnalysisCategory	Horse Cree	Juanita Cre	Little Bear C	Little Swam	Lower Nort	Lower Sam	Parr Creek	Perry Creek	Swamp Cre	Upper North	Upper Sam
1	Arterial	3.211606	2.340799	3.97159	1.998881	1.989065	2.391256	3.550922	4.163802	1.119425	3.208567	3.861229
2	Limited Access	0	3.099527	0	0	3.162008	0	1.139735	1.350647	0	2.105872	4.014983
3	Local Road	4.964414	6.694123	9.267353	5.334134	3.663652	4.96063	3.220707	5.465045	4.851003	4.695235	6.075516
4	Outside Bothell	0	0	0	0	0	0.000014	0	0	0	0	0.000044
5	Undefined	0.821728	0.494543	0.198618	0.54532	0.276864	0.385216	0.38047	0.386963	0.305878	0.317141	0.46366

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	OBJECTID	AnalysisC	Horse Cre	Juanita Cr	Little Bear	Little Swa	Lower Nor	Lower San	Parr Creek	Perry Cree	Swamp Cr	Upper Nor	Upper Sam	mamish River	
2	2	Limited A	0	3.099527	0	0	3.162008	0	1.139735	1.350647	0	2.105872	4.014983		
3	1	Arterial	3.211606	2.340799	3.97159	1.998881	1.989065	2.391256	3.550922	4.163802	1.119425	3.208567	3.861229		
4	3	Local Roa	4.964414	6.694123	9.267353	5.334134	3.663652	4.96063	3.220707	5.465045	4.851003	4.695235	6.075516		
5	4	Outside B	0	0	0	0	0	0.000014	0	0	0	0	0.000044		
6	5	Undefine	0.821728	0.494543	0.198618	0.54532	0.276864	0.385216	0.38047	0.386963	0.305878	0.317141	0.46366		
7															

6.14. Copy the relevant "subtype" rows and paste *JUST THE VALUES* into the corresponding rows in your main analysis spreadsheet.

- **a.** You may want to go row-by-row to be safe. If copying multiple rows at a time, *make sure* they're in the correct order (e.g. for Streets we had to switch rows 1 and 2).

NOTE: The conditions at the end of the last subsection pertaining to how columns *should* line up apply here. Make sure to add "0"s in the appropriate gaps (if present) after checking your column headers.

6.15. Repeat for each layer in the first step in this subsection.

Final Part 3. Storm Layer Analysis

This subsection is dependent on your having named your storm layers exactly according to the template **"Active_d…"** (e.g. Active_dCatchBasins). If you didn't, you may want to use the staged analysis geodatabase instead for this part, or you can just rename the layers after you've added the feature classes to ArcMap in the second step below (see the staged geodatabase for the layer names).

6.16. Make sure there are NO storm layers in your ArcMap document (other than the monitoring basins). The script we'll run later will look for the expected storm layer names in ArcMap.

6.17. Add all feature classes in your analysis geodatabase that start with **"Active_d"** to your map. You may want to pause drawing since the map display isn't particularly relevant now.

6.18. Find the Python script **tabulate_storm_layers.py** in the files included for this project. Right-click it and **Edit with IDLE** *[NOT ArcGIS Pro]***.**

- **a.** If you're missing this file, check the end of this document for the text you can copy. You'll need to type or paste your geodatabase path (described in the next step) *after* pasting into the ArcMap Python interactive window, which can be a little tricky.

6.19. Near the top of the file, type or paste the full file path of the analysis geodatabase you're currently working with in between the two quotation marks on the line: **out_gdb = r""**

6.20. Select <u>all</u> of the code in this Python file and copy.

6.21. In your ArcMap document, open the Python interactive window.

6.22. Paste the Python code into the first line of the Python window and press Enter.

Let the Python code run. It will create Tabulate Intersection output tables as it goes. You'll know it's complete when you see **"Done!"** print in the Python interactive window (you might have to scroll down in the Python window to see this).

6.23. After you've confirmed that the code is finished running, go back to your main analysis spreadsheet and look for the first storm layer (Wetland Area, under Natural Areas). Complete the following steps for each layer as you go down the rows in your spreadsheet.

6.24. In ArcMap, open the intersection output table (**"BasinIntersect_..."**) for the layer corresponding to the row you're on.

- **a.** **Filters** is a sum of values from three tables: **Catch Basins**, **Control Structures**, **Vaults**.

6.25. Select all rows in the output table and copy them to a blank Excel spreadsheet.

6.26. Immediately copy the set of cells you just pasted and paste them again (somewhere not overlapping) with the Transpose option. This will turn the rows into columns and vice versa.

6.27. Copy the **Bothell City Portion (sqmi)** row from the main analysis spreadsheet into your "scratch" spreadsheet and *make sure* the columns line up with your values.

6.28. Look back at the current row you're trying to fill in your main analysis spreadsheet. The units (second column) for that row will tell you which row in your "scratch" spreadsheet to use in the following step:

- **a.** **%** ➔ **PERCENTAGE**
- **b.** **mi/sqmi** ➔ **LENGTH**
- **c.** **#/sqmi** ➔ **PNT_Count (if present) or InLucity**
  - i. If you need to use **InLucity** (the count for non-point layers), change any value of "Yes" to "1" in this row before proceeding.
- **d.** For the row **Filters**, use **FilterCount**. You'll need to add together the FilterCounts from **Catch Basins**, **Control Structures**, and **Vaults** into a single row before proceeding. Keep close track of the FilterCount rows as you work with each table.

6.29. If the field is **PERCENTAGE**, skip to the next step and just copy the **PERCENTAGE** row values. Otherwise, in a new row, divide the values in the row just you identified in the previous step by the square miles row to get "[units] per square mile".

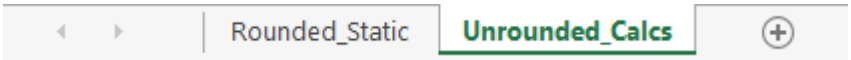
6.30. Finally, copy the "per square mile" row you just created as values (not formulas) into the corresponding row in your main analysis spreadsheet. Repeat these steps for each Storm row until the spreadsheet is complete.

NOTE: The conditions at the end of the first subsection pertaining to how columns *should* line up apply here. Make sure to add "0"s in the appropriate gaps (if present) after checking your column headers.

Final Part 4. Final Wrap-Up

Provided you were copying values and not formulas, and that you kept all columns lined up, your main analysis spreadsheet should now be complete. Any of the meta-rows (all which are sums) should have also auto-calculated as you were going.

You may want to create a “static copy” of this worksheet by duplicating it and repasting the values in as values to get rid of formulas. You can also consider rounding values in this “static copy”. If you started from **WatershedPrioritization_Analysis_Blank.xlsx**, you should have two worksheets already created for this purpose. You’ll also find both versions in the final Excel file from the City of Bothell to compare if needed.



Appendix 1. Python Code for Storm Layer Analysis

The following code can be copied (from Word; PDF does not work) into the ArcMap Python interactive window during step 6.22. You'll need to type or paste the full file path of the analysis geodatabase you're currently working with in between the two quotation marks on the line: **out_gdb = r""**

```
if True:
    import os
    out_gdb = r""
    if not os.path.exists(out_gdb):
        err_msg = "output geodatabase does NOT exist at {0}".format(out_gdb)
        raise ValueError(err_msg)
    try:
        mxd = arcpy.mapping.MapDocument("CURRENT")
        layers = arcpy.mapping.ListLayers(mxd)
        wetland_layers = {
            "Active_dWetlands",
            "Active_dWetlands_and_Buffers",
            }
        filter_layers = {
            "Active_dCatchBasins",
            "Active_dControlStructures",
            "Active_dVaults",
            }
        line_layers = {
            "Active_dBioSwales",
            "Active_dDetentionPipes",
            "Active_dDitches",
            "Active_dPipes_and_Culverts",
            "Active_dRivers_and_Streams",
            "Active_dTrenches",
            }
        for layer in layers:
            if layer.name.startswith("Active_d"):
                print(layer.name)
                if layer.name in filter_layers:
                    sum_fields = ["InLucity", "FilterCount"]
                elif layer.name in wetland_layers:
                    sum_fields = None
                else:
                    sum_fields = ["InLucity"]
                if layer.name in line_layers:
                    arcpy.TabulateIntersection_analysis(
                        "dMonitoringBasins_BothellCity",
                        "Name",
                        layer.name,
                        os.path.join(out_gdb,
                                     "BasinIntersect_" + layer.name[7:]),
                        sum_fields=sum_fields,
                        out_units="MILES",
                        )
                else:
                    arcpy.TabulateIntersection_analysis(
                        "dMonitoringBasins_BothellCity",
                        "Name",
                        layer.name,
                        os.path.join(out_gdb,
                                     "BasinIntersect_" + layer.name[7:]),
                        sum_fields=sum_fields,
                        )
    finally:
        try:
            del layer
        except:
            pass
        try:
            del layers
        except:
            pass
        try:
            del mxd
        except:
            pass
    print("Done!")
```