

Introduction to ModelBuilder

Aims and Objectives

In this practical you will be introduced to the ModelBuilder tool in ArcGIS, and will use it to build two geoprocessing workflows.

By the end of this class you should be able to:

1. Translate conceptual models of workflows into ModelBuilder
2. Use a range of ArcGIS tools to build complex models
3. Save models as custom toolboxes
4. Load and edit saved toolboxes

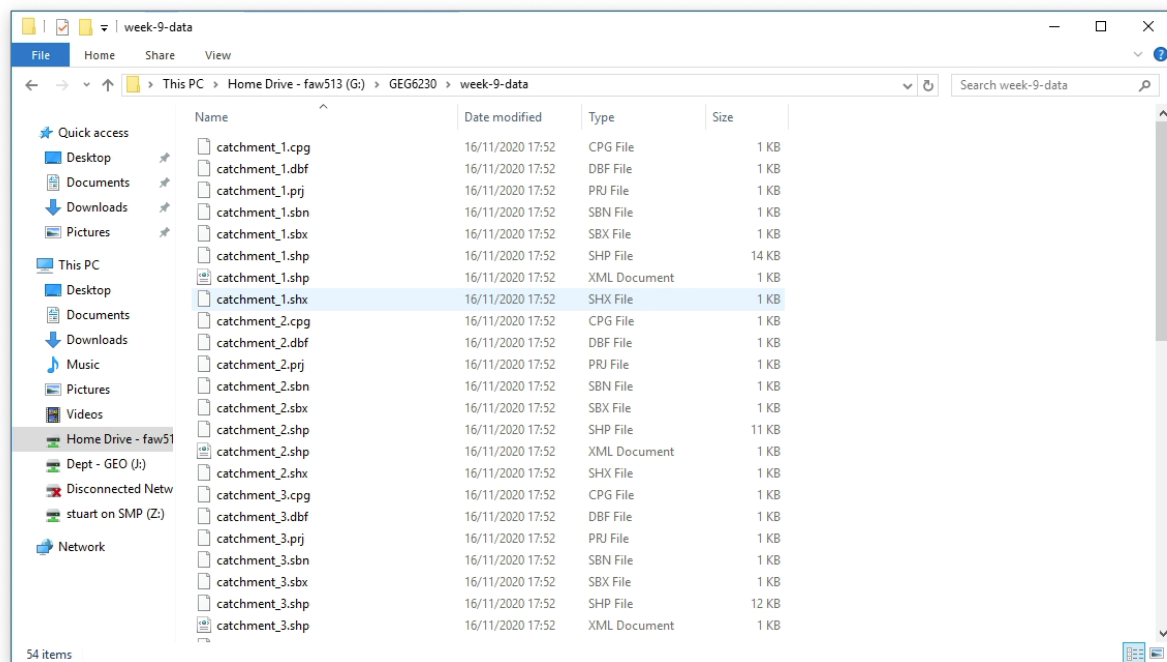
Downloading the data

This practical, and the work in the next two weeks all use the same data, which can be downloaded from the [Week 8](#) section on QMplus.

Extracting the data

Once the zip file has been downloaded and copied to a folder in your [G: Drive](#), you can extract it by right clicking on the downloaded zip file and selecting [Extract All...](#)

This will complete after a few seconds, and you will be able to see all of your data files in the folder that has been created:



Remember to put your data in a sensible folder that does not have any spaces in the file name.

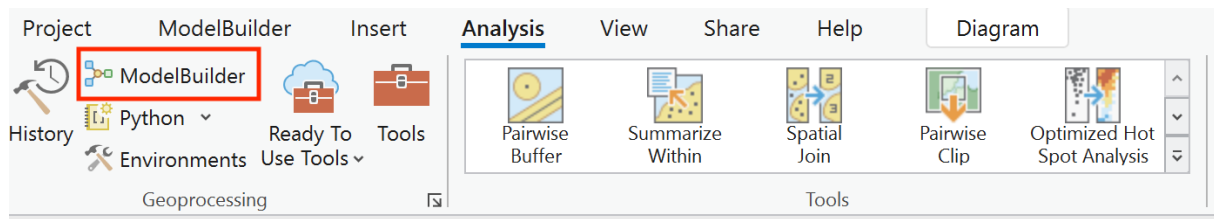
Building our first model

The first model we will be building will be a solution to our in-class exercise of identifying trees near a road and lake. To recap our requirements, we need to identify all trees that are within 300 meters of the lake and 400 meters of the road. For this model, we will be using the following files:

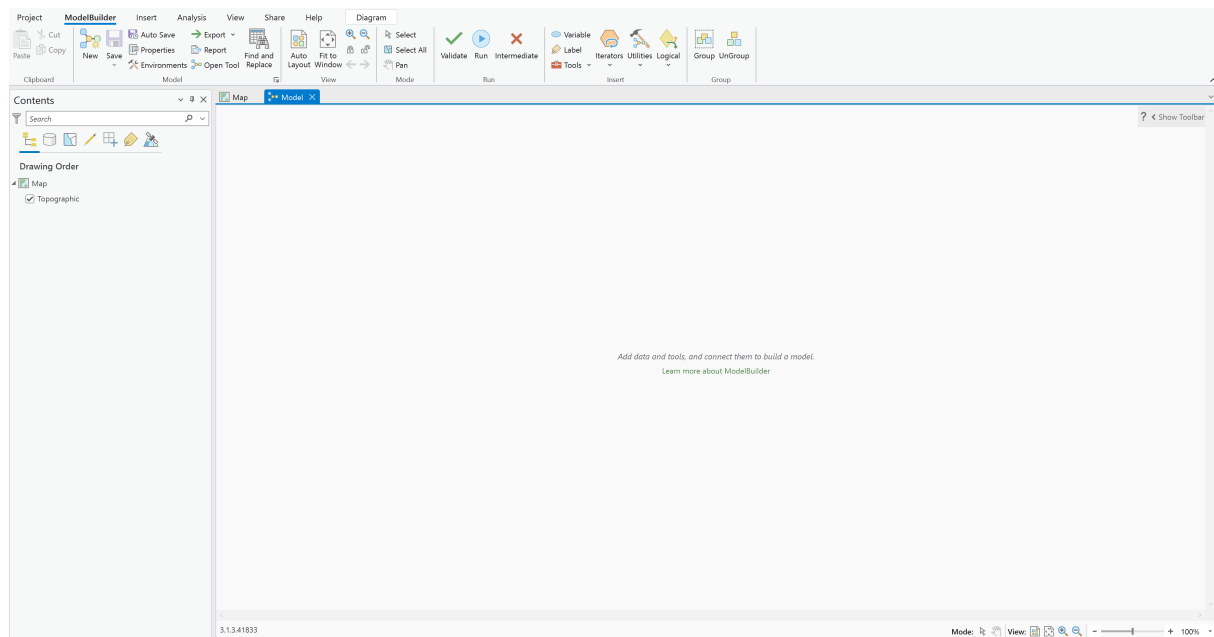
- `road.shp`
- `trees.shp`
- `field.shp`
- `lake.shp`

Loading ModelBuilder

Load ArcGIS as normal, and create a new blank map document. Launch the ModelBuilder by opening the **Analysis** ribbon at the top of the screen. On the left hand side there is a button called **ModelBuilder**, click it to open the modelbuilder window.



After a few seconds, a window looking something like this will open up:



This is our workspace to create a data processing workflow. Note that we can toggle between our model view and our map view using the tabs at the top of the window. Take a little bit of time to explore the various buttons and options within this window, you'll see some familiar things and some new stuff - don't worry if the new stuff is unfamiliar at this stage, we'll explore the key tools through the rest of this session.

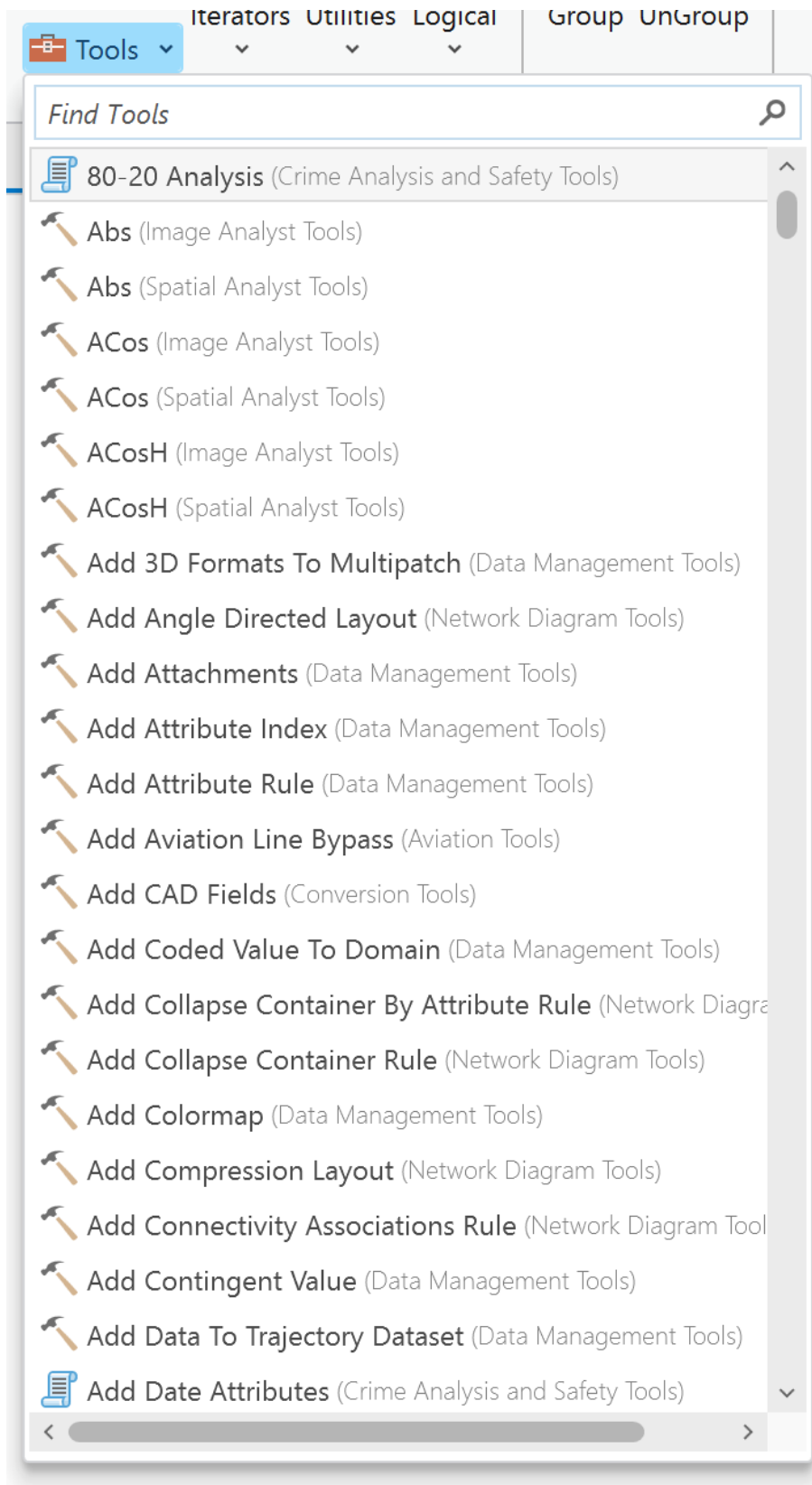
Building our model

Now that we have familiarised ourselves with the options available to us, let's add some data to our map window. Using the normal **Add Data** button in the map ribbon, add the `road.shp`, `trees.shp` and `lake.shp` files that we downloaded earlier. Now switch back to the ModelBuilder tab using the button at the top of the window.

When we are designing a workflow, we often start with the data that we are processing. To add data to the ModelBuilder window, click and drag a layer from the table of contents into the ModelBuilder window. After a few seconds it will appear as a blue oval - this is how data is represented. Do this for

the three datasets we have added.

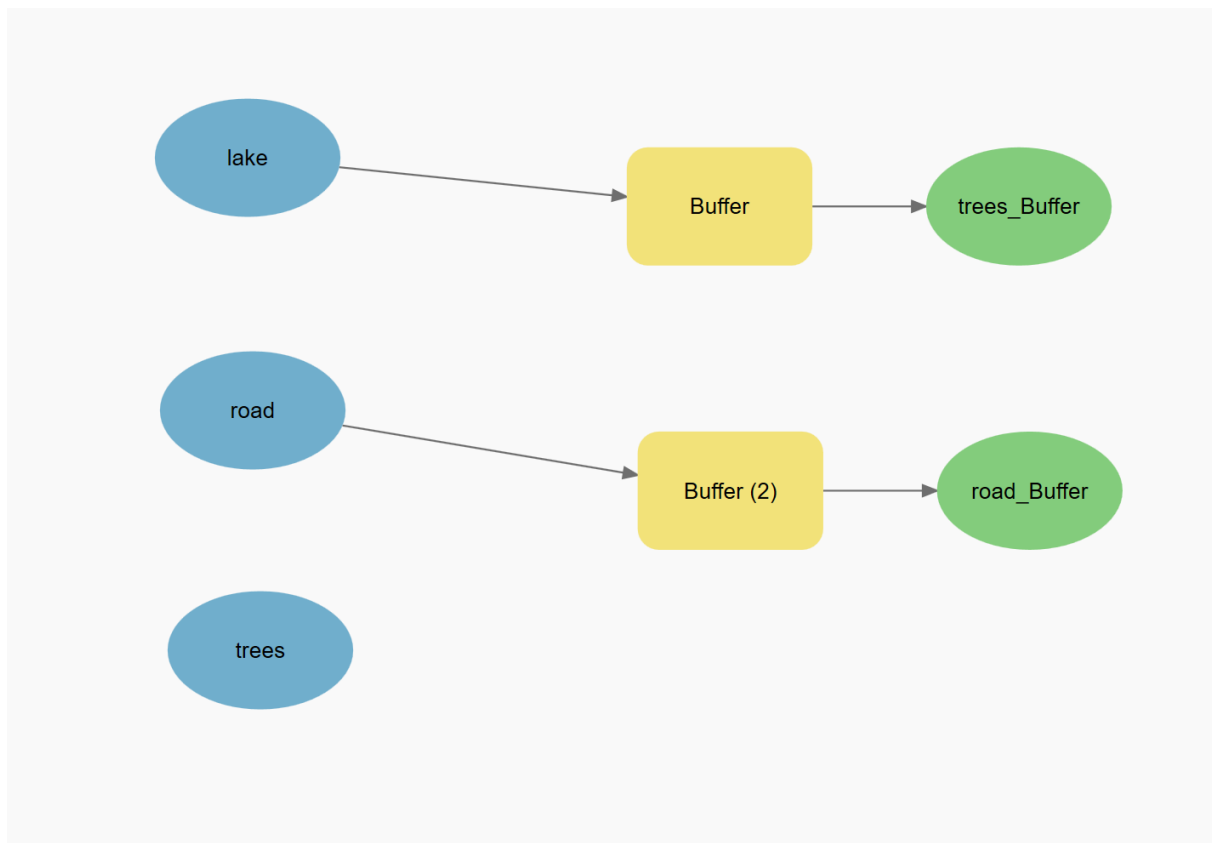
We can now add our first GIS tool from the **Tools** dropdown inside the ModelBuilder ribbon. Click on the small down arrow to open up the search window.



We need to find all of the trees within a distance of an object, so let's create a buffer. Search for **Buffer** (**Analysis Tools**) and click and drag the tool into the ModelBuilder window. Initially it will load as a grey rectangle connected to a grey oval. Remember that our input data are blue ovals, so that means that ovals are used to represent data, and rectangles represent tools.

Once we have added a tool it needs to be connected to some data. Click and drag on the **road** oval to connect it to the **Buffer** tool and selecting **Input Features** when prompted. Now double click on the buffer tool, choose a sensible place to store the output, and set the distance to 400 **meters**. Click OK, and when you return to the main ModelBuilder window, the tool will have turned yellow, and its output green.

Repeat this process with a second buffer tool, this time connected to **lake.shp**, with a buffer distance of 300 **meters**. Your model should now look similar to this:



We can use the **Intersect** (**Analysis Tools**) tool to find the intersection between the two buffered polygons: this will give us the area that is within the two distances of our features of interest. Again, search for this tool and drag and drop it into the ModelBuilder window and connect the two outputs from the buffer tools as inputs into the **Intersect** tool.

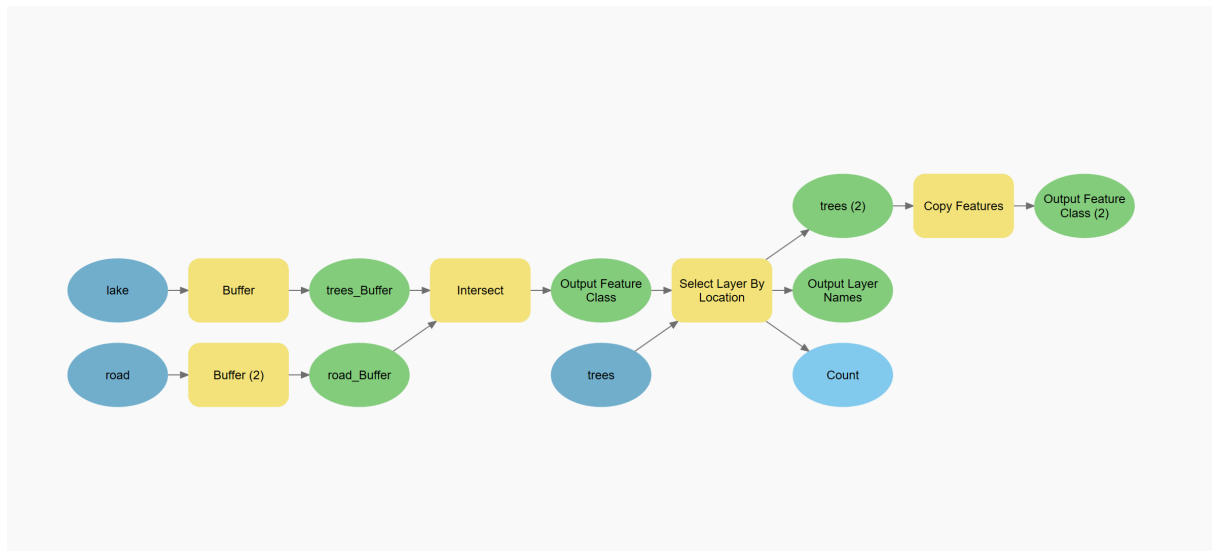
The final stage is then to select all of the trees within this intersection and create a new shapefile containing only the selected trees. Add in the **Select Layer by Location** (**Data Management**

Tools) tool by dragging and dropping it into the model. Don't connect anything to it directly, instead double click on it and choose the following options:

- Input Feature Layer: **trees** which will be listed under **Model Variables**
- Relationship: **WITHIN**
- Selecting Features: The output from the intersect tool, **Output Feature Class** if you didn't rename it

Leave everything else as its default and click **OK**. It should now be connected up and coloured yellow.

Now add the **Copy Features** (**Data Management Tools**) tool to the model, connect the output of the **Select Layer by Location** tool to it, double click on it and set a sensible place for the final result of the model to be saved. Your finished model should look something like:



Running a model

To run a model, first click **Validate** to check for any common errors (missing files, broken connections) and then click on the **Run** button, this will launch a window showing the progress of your model, once it has completed, you can return to the ArcGIS map window and load the results of the model to see if it has worked.

Tidying up

Once you have done all this, your model may be a bit messy, click the **Auto Layout** button to tidy things up. You should also make sure that all of the input and output names make sense, you will need to change at least some of them by right clicking on an element of the model and selecting **Rename**.

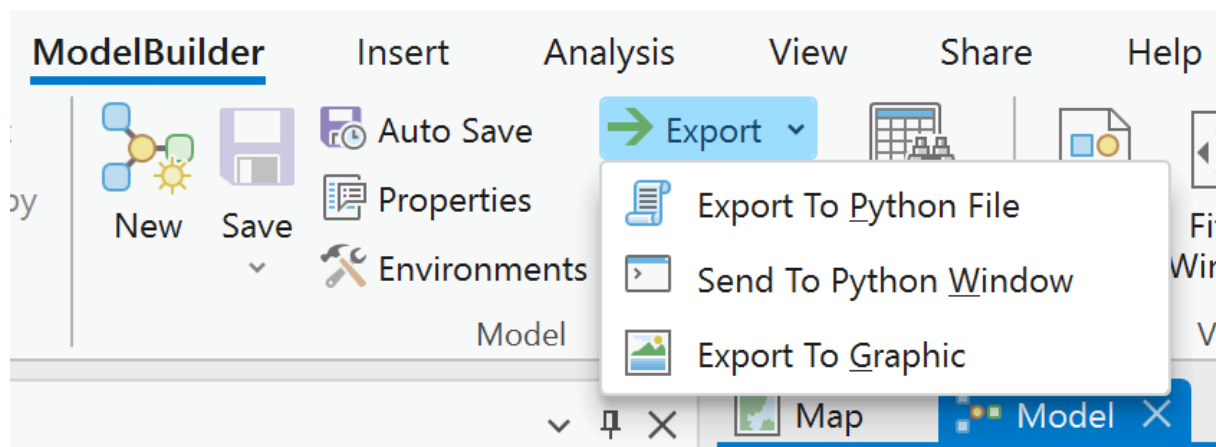
You can also add labels to the model using the [Label](#) button, found just above the [Tools](#) button we have been using. **This is very important in helping others understand your model.**

Saving a model

Now that we have made a model, we will want to save it so that we can use it again and share it with others. To do this, click the save button, which will automatically save the model to a Toolbox associated with your current project.

Exporting a model image

It is often useful to be able to export an image of our model to include in a report. This can be done with the [Export](#) dropdown menu:



Select [Export](#) > [To Graphic](#) and choose a place to save the file. We can choose a [.pdf](#) file if we just need a quick image, but if we want to be able to edit the file easily in Inkscape, we need to select [Scalable Vector Graphic \(SVG\)](#), as this can be loaded by Inkscape.

Loading a model

If we wish to return to a previously saved model, we need to load the toolbox we saved it in. To do this, click on the [Toolbox](#) button inside the [Insert](#) ribbon and select [Add Toolbox](#), and navigate to where you saved your project and select the toolbox file. Now we can open the [Catalog](#) pane and expand the [Toolboxes](#) dropdown to see our newly added toolbox. Expand the toolbox to see your model. To reopen your model in ModelBuilder, right click on the model and click [Edit](#).

A more complex model

Now that we have figured out the basics of building and using a model, we are going to build a new model to calculate the stream power at the outlet of a series of catchments. To do this we will use the following data, downloaded earlier in this practical:

- `catchment_1.shp` through `catchment_4.shp`
- `Slope.tif`

Do this work in a new ArcGIS project, as we are working in a different coordinate system to the last example and things can get confusing otherwise.

The aim of our new model is to calculate the stream power, E , which is an approximate measure of the rate of fluvial erosion at a point in a landscape. It is calculated as:

$$E = KA^mS^n$$

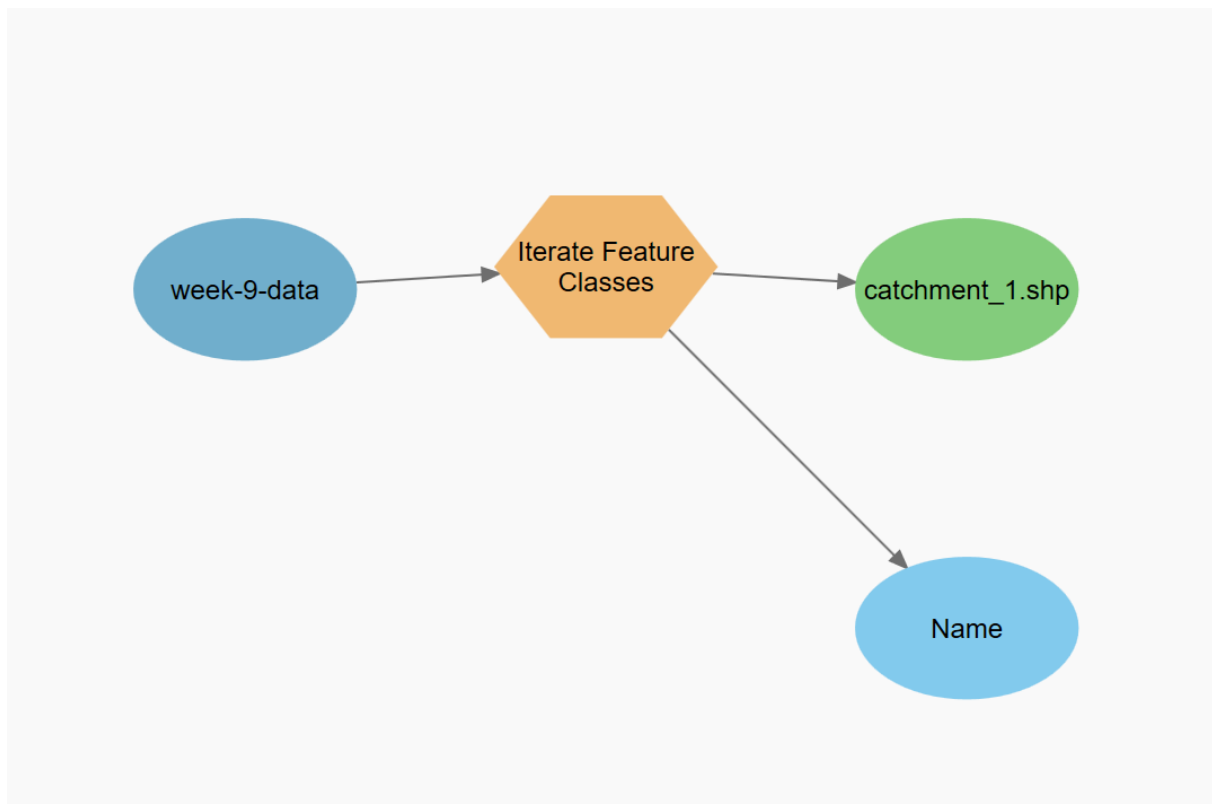
Where K is an erosion coefficient, m and n are constants, S is the slope and A is the basin area. Note that typically we would use the channel slope for this calculation, but for simplicity we will use the basin slope here as an approximation. The result of this calculation will be the erosion rate at the catchment outlet in meters per year.

Iterators

In this exercise we have multiple different catchment files which need to be processed. The aim is to add a new field to each of their attribute tables containing their erosion rate. ModelBuilder has some special tools for cases like this, called **iterators**, we can think of this as like the for loops we have been learning in Python.

The iterator we are going to use here is called `Iterate Feature Classes`, it is found in the `Iterators` dropdown menu. Double click on this tool to open its settings, and under `Workspace or Feature Dataset` choose the folder that the four `catchment_*.shp` files are in. Finally, select `Polygon` as the feature type. Click OK, and the tool will become coloured in. We have created a series of inputs, one for each of our catchment shapefiles, that we can now process automatically.

Your ModelBuilder window should look something like this:



In this example, the data folder may have more than just the `catchment_*.shp` files in it. To ensure that our iterator only iterates over the catchment files, we can use the **Wildcard** option in the same way we use `glob` in Python, adding in the text: `catchment_*` so that only files that match that pattern are iterated over.

If you include the file extension as part of the wildcard (e.g., `catchment_*.shp`), it will not work.

Zonal statistics

We need to calculate the mean basin slope for each of our catchments, we can do this using the **Zonal Statistics as Table (Spatial Analyst Tools)** tool. First, add the slope data to your map using the **Add Data** button and drag it across into the ModelBuilder window, then drag and drop the **Zonal Statistics as Table (Spatial Analyst Tools)** tool into the window and connect it to the slope raster, as the **Input Value Raster**. Now double click on the **Zonal Statistics as Table** tool and configure the following options:

- Input raster or feature zone data: `Catchment_1.shp` which will be listed under **Model Variables**
- Statistics type: **MEAN**

Click OK, and the tool should now be connected up to the green output from the iterator.

Gathering Values

We now have all of the values we need to calculate stream power. In order to perform calculations in ModelBuilder we need to use the **Get Field Value (Model Tools)** tool. Connect this new tool to the output of the **Zonal Statistics as Table** tool and double click on it to open its settings. Here you need to select the Field that we want to get, in this case this is **Mean**. You also have to specify the **Data type**. As we are working with numerical data, select **Double**, which is similar to the **float** data type we have seen in Python. This will ensure that calculations can be performed using these values.

This tool effectively creates a variable that we can use in calculations within a model run, so we should give it a meaningful name such as **slope**, by right clicking on the green output and choosing **Rename**.

We now need to repeat this process to get the K value and basin area from each shapefile's attribute table. Add another **Get Field Value** tool, and this time connect it to the green output from the iterator. Select either **K** or **area** in the settings, and rename the output as before.

Repeat this once more so that we have 3 outputs from **Get Field Value** tools, one for each of Area, K and Slope.

Just like in Python, all of these names are case-sensitive, so be careful!

Adding a new field

We need to create a new field in each of our catchment shapefiles to store the stream power values. Drag and drop the **Add Field (Data Management Tools)** into the ModelBuilder window. Connect it to the green output from the iterator, double click on it and modify the following settings, leaving all the others as their defaults:

- Field Name: **E** You will have to type this, as we are creating a new column in the attribute table.
- Field Type: **Double**

Calculating stream power

The next tool we have to add to our model is the one that will do the calculation, using the values we have identified using the **Get Field Value** tool. Drag and drop the **Calculate Value (Model Tools)** tool into the window. This is a really useful tool in ModelBuilder as it allows us to write small bits of Python to perform calculations.

Double click on this tool and select the following options:

- Expression Type: `Python`
- Expression: `stream_power(%Slope%, %Area%, %K%)` - the % symbols are used to denote an ArcMap variable name in a way that Python can understand.
- Code Block: See Below

The code block we need to add is a small Python function which will calculate the stream power for a given set of input values:

```
1 def stream_power(slope, area, K):  
2     return K * (area ** 0.3) * (slope ** 0.6)
```

The completed window for the `Calculate Field` tool will look something like this:

Calculate Value

Parameters Environments Properties

Expression Type
Python

Expression

Variables Functions

catchment_1.shp
Name
week-9-data
Slope
Area
K
ZonalSt_catchme1

.as_integer_ratio()
.capitalize()
.center()
.conjugate()
.count()
.decode()
.denominator()

Insert Values

Expression
stream_power(%Slope%, %Area%, %K%)

Code Block

```
def stream_power(slope, area, k):  
    return k * (area **0.3) * (slope **0.3)
```

Data type
Variant

OK

Finally, we are going to store the result of our calculations in the new column in the attribute table we created. Drag and drop the **Calculate Field (Data Management Tools)** tool into the window.

Double click on the tool and select the following outputs:

- Input Table: This is the output of the **Add Field** tool
- Field Name: This should be the name of the new field added by the **Add Field** tool, which will be **E** in our case
- Expression Type: **Python**
- E = : %E% - This might give an error, but is correct!

The completed window for the **Calculate Field** tool will look something like this:

Calculate Field [X]

i This tool modifies the Input Table [X]

Parameters Environments Properties [?]

Input Table
catchment_1.shp (2) [v] [Folder Icon]

Field Name (Existing or New)
E [v] [Gear Icon]

Expression Type
Python [v]

* Expression

Fields [v] [Filter Icon] Helpers [Filter Icon]

FID	.as_integer_ratio()
Shape	.capitalize()
Id	.center()
gridcode	.conjugate()
K	.count()
Area	.decode()
E	.denominator()

Insert Values [v] * / + - =

E =
%E%

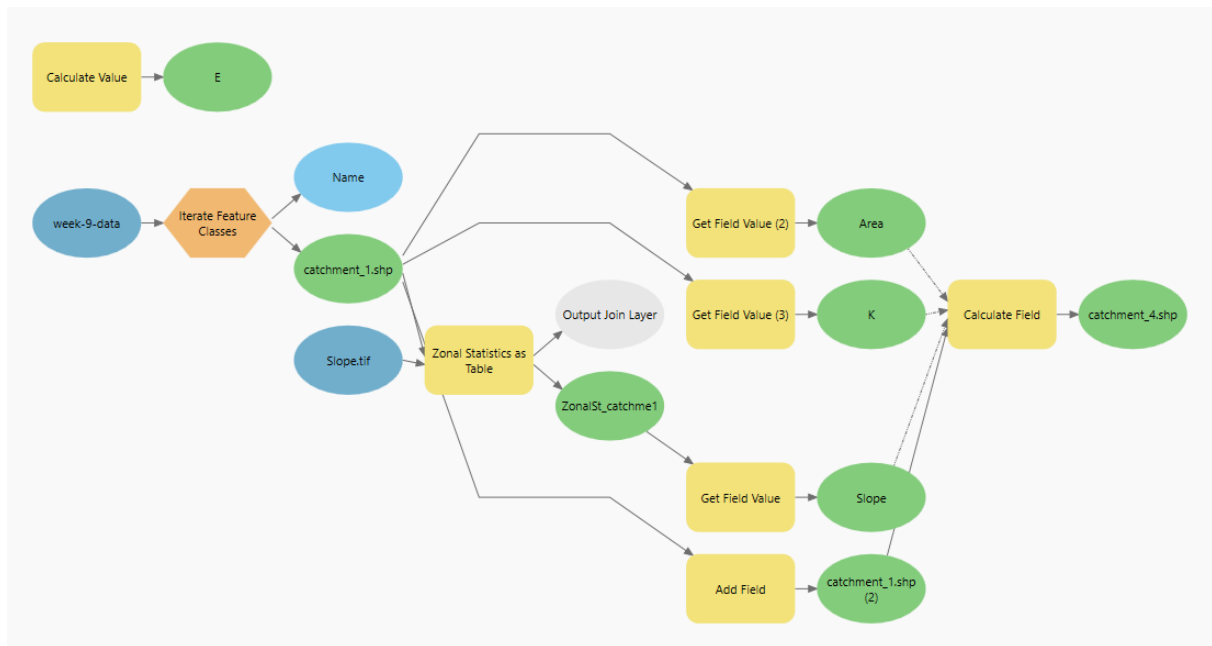
Code Block

Recent [v] [Refresh Icon] [Delete Icon] [Checkmark Icon] [Folder Icon] [Next Icon]

☐ Enforce Domains

OK

Having filled all of this in, click OK to return to your model. The final thing we need to do is add **Preconditions** to the model, so that it does not run things in the wrong order. To do this, connect each of the outputs (K, slope, Area) to the **Calculate Field** tool, selecting the **precondition** option when prompted. This ensures that the calculation will only be performed when the variables are all ready. The final model should look something like this:



Once all that is done, we are ready to run our model! As before, click validate to check for any errors, and if none are found, click run. It might take a bit of time for all for all of the steps to complete, but now it is automated you can walk away from your machine while it works for you.

Debugging and finishing up

You can now add the 4 catchment files to the regular ArcGIS window and check to see the results of your calculation. Just as before, remember to save this model in a toolbox, you can use the same one you created earlier, so that you can return to this in the future.