ABM – Week 6 – WORKING WITH GIS

***GOAL OF THIS LESSON***

Be comfortable with loading and using GIS data in NetLogo

**INTRODUCTION**

1. Review the foreach command – for example, try using it to print out each of the elements of a list.

E.g. foreach [1 2 3] [x -> show x]

1. Ask each patch to sprout some random number of turtles on itself.

Useful primitives: sprout

1. Load the GIS extension into NetLogo

Useful primitives: extensions

**LOADING AND BASIC VISUALISATION**

The functions you will need are available here:

<https://github.com/NetLogo/GIS-Extension#primitives>

1. GIS data always has a projection which tells the computer (or viewer) how the dataset maps to the real, physical world. Shapefiles should always come with a .prj file to define their projection. First, load a projection.

Hint: the primitive gis:load-coordinate-system will be helpful here, together with the names of the projection files.

1. Practice loading a basic dataset into NetLogo. You don’t need to display the dataset – just load it. Remember: the dataset is given in the \*.shp files, and the projections in the \*.prj files.

Useful primitives: gis:load-dataset

1. Try to display the shapefile. Practice changing the colour in which the geometries are drawn.

Useful primitives: gis:set-drawing-color and gis:fill

1. Make sure that you’re comfortable setting up the envelope which determines what is shown in the image.

Useful primitives: gis:set-world-envelope and gis:envelope-of.

**WORKING WITH THE DATA**

1. Try to use the console to print out the “NAME\_2” attribute associated with each

geometry.

Useful primitives: gis:property-value, gis:feature-list-of

1. See if you can change the colour of the polygons to be based on the “POP2017” attribute.

Useful primitives: gis:property-value, scale-color

1. It can be a little tricky to output point data, because the GIS extension works with vertices. Very carefully, see if you can use the gis:location-of function to create a turtle which shows the name of the borough at the center of that borough.

Useful primitives: gis:location-of, gis:property-value, possibly gis:centroid-of

**CONVERTING VECTOR TO RASTER**

1. See if you can use the gis:apply-coverage to turn the vector data into raster data.
2. Sprout turtles from the patches based on the population data (**CAUTION**: you should probably only sprout 1 turtle for every 10000 people!)

**EXTENDED CHALLENGES**

If you found all that challenging but reasonable, consider undertaking the following:

1. Make turtles move around the environment in response to the density of other turtles in the polygon.
2. Sprout turtles from within polygons rather than patches.
3. Download road data for London and see if you can display it in the same model.