Locating the optimal position of wind turbines

Scenario

Wind turbines are seen as a good source of clean energy. After the initial cost of installation a turbine can be expected to last between 20 – 25 years. Modern turbines are typically between 25 – 80 metres high with a rotor diameter of up to 65m. They are large structures which can be seen for many miles.

For the best performance, wind turbines are typically placed in windy places such as the top of valleys, ridges and offshore.

Some people see wind turbines as “blots” on the landscape, considering them as noisy and unattractive structures. Despite being green energy they are not welcomed in areas of outstanding natural beauty such as national parks and due their immense size they can potentially be very hazardous for low flying aircraft and migratory birds.

With so many objections to consider, GIS is an ideal tool for helping a company to locate an optimal site which maximises the energy output whilst considering the wishes of local people and other organisations.

In this exercise you are a manager of a wind farm company looking to site three new wind farms. You will use a mixture of data to locate windy sites and a DEM to calculate a viewshed. A viewshed is a spatial layer indicating which areas are visible to a known location. From this you can identify which towns are able to see a new wind farm and with this knowledge you can tailor any advice given to these populations.

Data used in Exercise

This exercise is supplied with several datasets and you will use these to create a viewshed and answer some basic question about visibility. The supplied data are:

- DEM (a raster binary file)
- Wind speed data (a raster binary file)
- Towns (a shape file)
- Wilderness area boundary (a shape file)

All of these map layers have co-ordinates measured in unprojected degrees of latitude and longitude, based on the WGS1984 datum.

Setup

Start up ArcGIS Pro and create a new map from the template for this. When you have started up the software, head for the *project* menu, then *licencing*, and then press ********** to activate the Spatial Analyst extension.

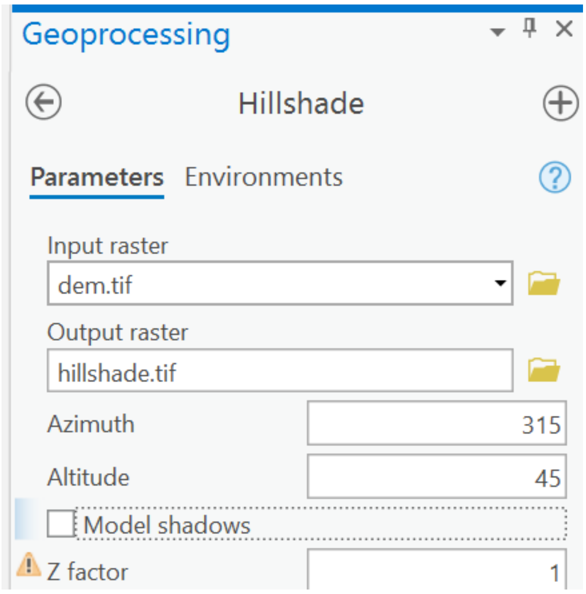
Set up a connection to your working folder with the data for this exercise and display and explore the map layers provided.

Exercise

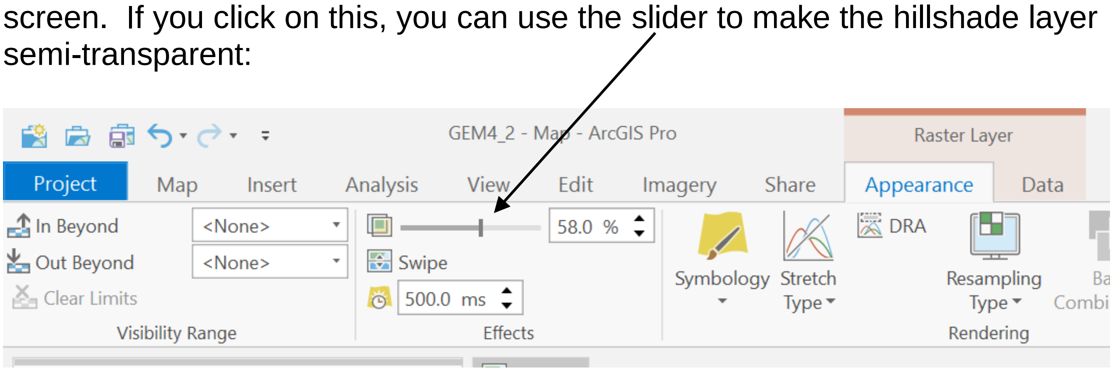
1.1 Create a hill shade of your DEM

We will begin by creating a hillshading image of our DEM. To do this:

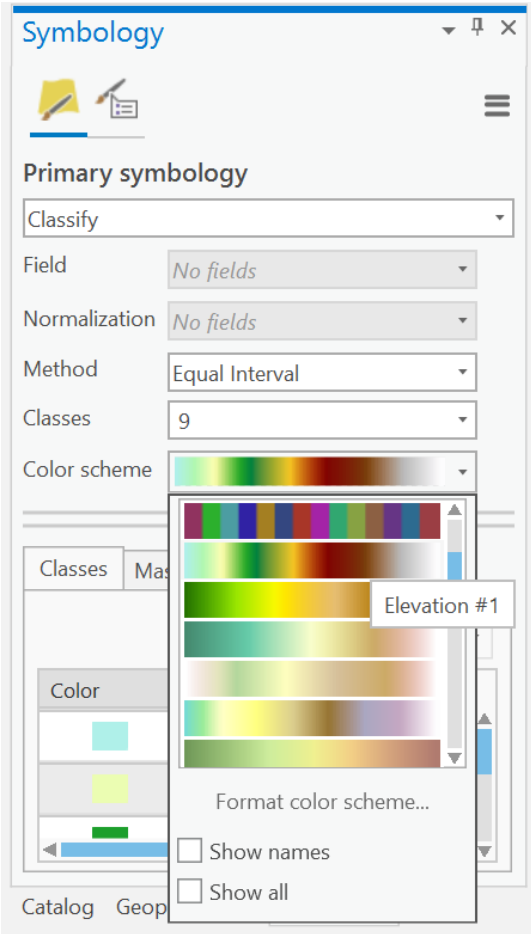
- go to the analysis menu and choose *tools* to bring the geoprocessing panel, then search for 'hillshade', choosing the *hillshade* tool from the list of search results.
- Under *input raster*, select your **dem** raster map layer and provide an appropriate *output raster* name, such as **hillshade** [I typically save raster files into folders rather than geodatabases, using the geotiff format. You can do this by adding a .tif to a file name. I do this because ArcGIS Pro is less 'fussy' about file names and folder pathways for raster geotiffs. Other users have different ways of working of course]
- To calculate hillshade, you need to specify the location of the sun. This is normally specified as two angles. The *altitude* indicates the sun's position in degrees above the horizon, whilst the *azimuth* indicates the sun's position as a compass bearing. The default settings have the sun at 45^0 above the horizon and at a bearing 315^0 (i.e. to the northwest). In most circumstances, these settings provide an appropriate cartographic display.
- The *z factor* enables the effect of terrain to be exaggerated in areas of gentle topography. In effect, the *z factor* scales altitude relative to horizontal distance, so a *z factor* of 3 will triple height differences relative to reality. For now, leave this set to *1*.
- By default, *hillshade* only considers the orientation (slope and aspect) of the terrain relative to the sun. It does not take into account the effect that a landform can have on its surroundings, such as the casting of shadows in a valley bottom. The *model shadows* option enables such effects to be modelled. This might be used in remote sensing studies of shadowing, but is typically not used for cartographic purposes, so leave this option unchecked.



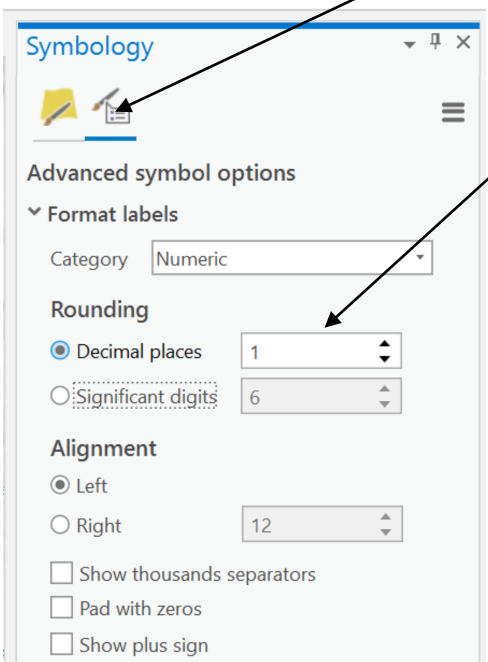
Having set up a new hillshading layer, we can then select this layer in the left-hand contents panel, so that the *Appearance* menu appears at the top of the screen. If you click on this, you can use the slider to make the hillshade layer semi-transparent:



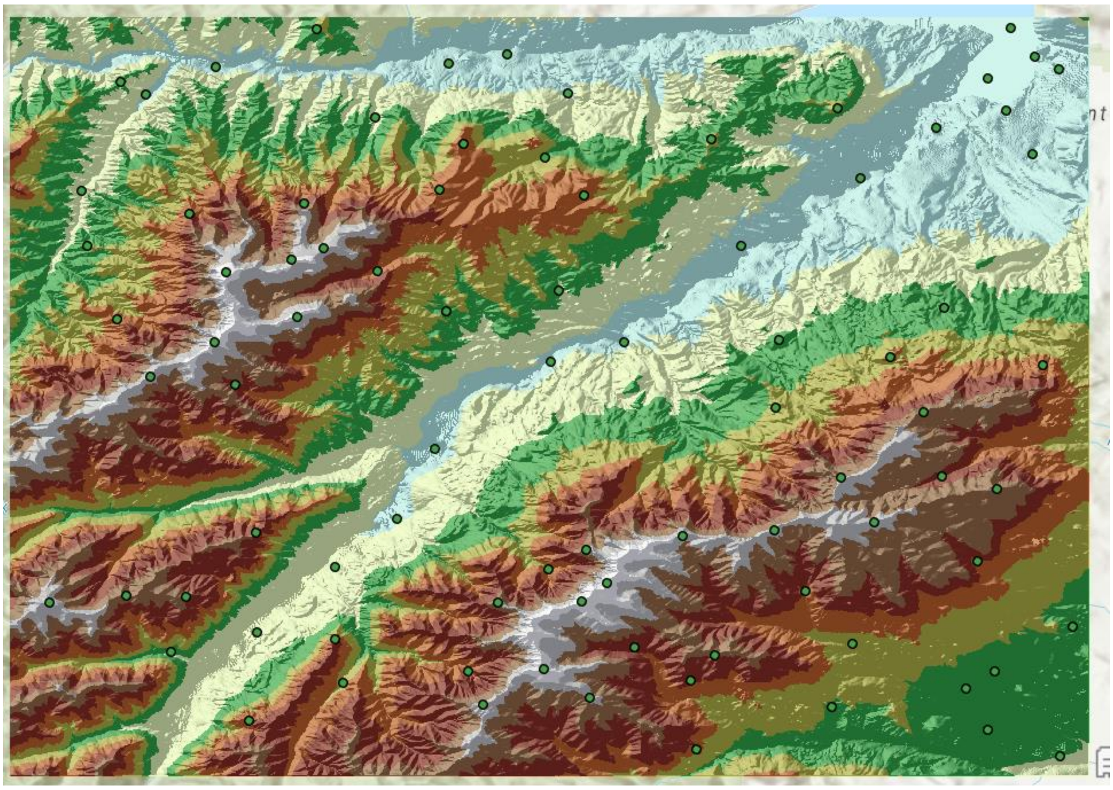
You might also want to change the display of the DEM layer by right-clicking on it in the left-hand table of contents and choosing *symbology*. Click on the *classify* option and under *classes* on the right, select a large number of classes, e.g. 9. You can also set the method of classification to be *equal interval* (we assume here that you are already familiar with the use of colour in ArcGIS). The *color ramp* shown below is particularly effective at displaying elevation data.



Similarly, you might want to switch to the 'advanced' symbol options and reduce the number of decimal places in the legend to **1** (since the elevation layer is very unlikely to be accurate to 6 decimal places!):



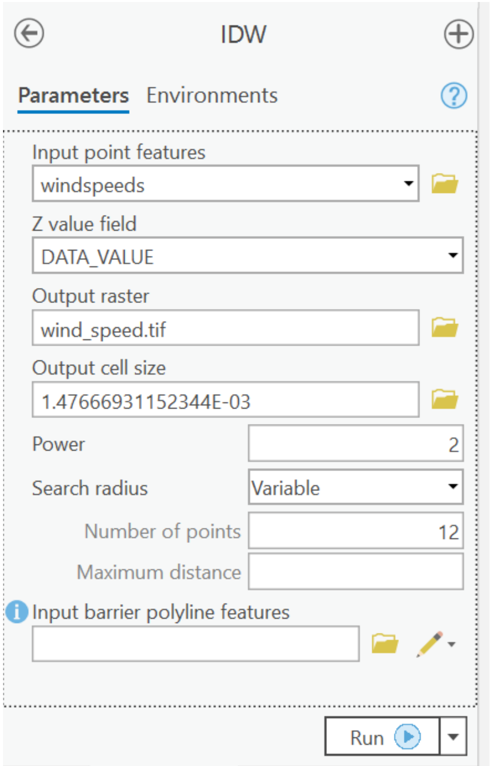
You should then see a map display that looks something like this:



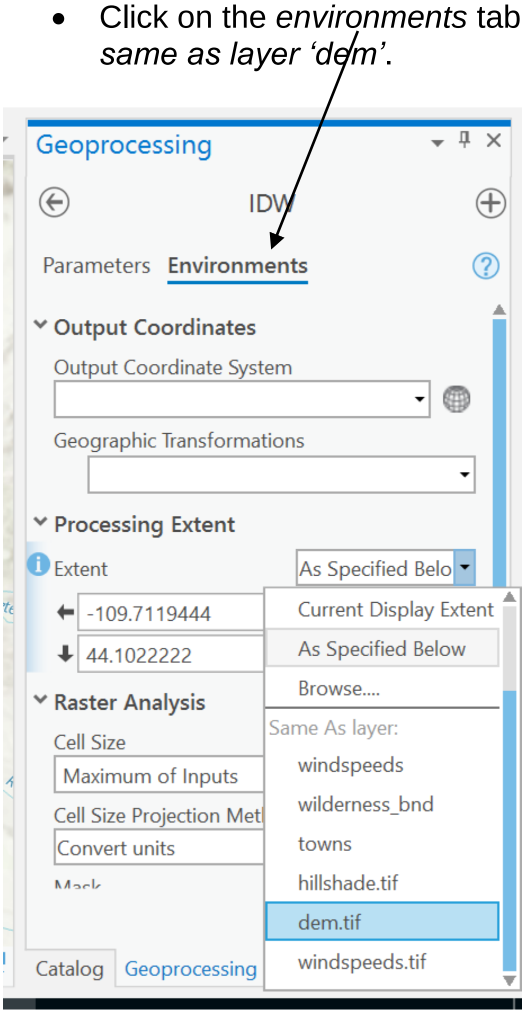
1.2 Interpolating a wind speed layer

We now need to identify those areas with high wind speeds. To do this, we will need to interpolate from the point measurements of average wind speed in the file **windspeeds**. To do this:

- Add the **windspeeds** shape file to your map display.
- Head for the *analysis* menu and then press the *tools* button on the ribbon to open up the geoprocessing panel, then search for “IDW”. Then run the *IDW* tool. IDW uses the interpolation technique called inverse distance weighting (IDW).
- Choose **windspeeds** as the *input point file* and select *data_value* as the *z_value_field* (i.e. the field that contains the wind speed estimates that will form the basis for the interpolation).
- IDW works by calculating a local average wind speed for each output grid cell based on nearby measured wind speeds. Under *search radius settings*, the *number of points* indicates how many of the points in **windspeeds** will be used to calculate this average (**12).** The *search radius* is set to *variable* because ArcGIS will look over a different area for nearby points depending on the density of measurement points (in areas with a dense network of points, the *search radius* will be smaller compared to areas with a sparser network of points).
- Type in an appropriate name for the *output raster*, e.g. **wind_speed**.



- Click on the *environments* tab and then set the *processing extent* to be *same as layer 'dem'*.



Click on OK and ArcGIS Pro should create a new raster grid depicting wind speeds across the study area. On top of this, add in the **towns** and **wilderness** map layers.

1.3 Locating the wind turbines

1. You are now ready to locate the three turbine sites. Each point you identify represents a single wind farm. A single farm can have more than one turbine but as a first pass, you are simply interested in identifying the general farm location.

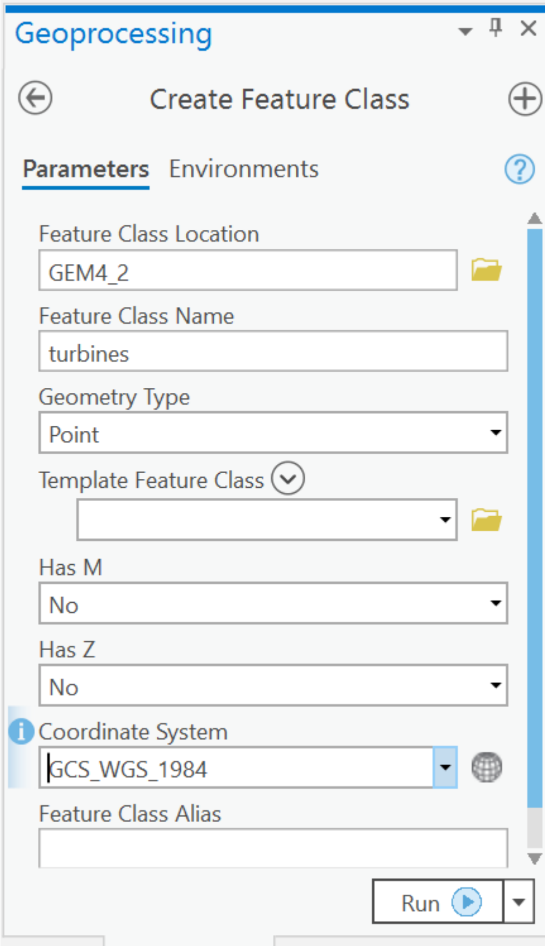
2. Each wind farm you identify must be:
    - Outside the wilderness area
    - Located in the windiest place
    - Not near another wind farm

Let us create a new point map layer to store these locations. To do this, right click in the right-hand catalog panel. Then click on the *New => shapefile* (or if you prefer, you could use the geopackage or file geodatabase options, creating a point feature class within your geodatabase if you choose the latter option).

A brief note about file formats for those converting to ArcGIS Pro from ArcGIS Desktop:

- **Geopackage**: If you are unfamiliar with it, this is an open geospatial data format based around SQLite (a relational database file format), used in open source packages such as QGIS as well as ESRI software. This format, gaining popularity, is used more widely in Pro than in Desktop.
- ArcGIS Pro no longer supports the **personal geodatabase** (a .mdb or Microsoft Access file format for storing spatial data, used in ArcGIS Desktop). Only the file geodatabase has been retained.

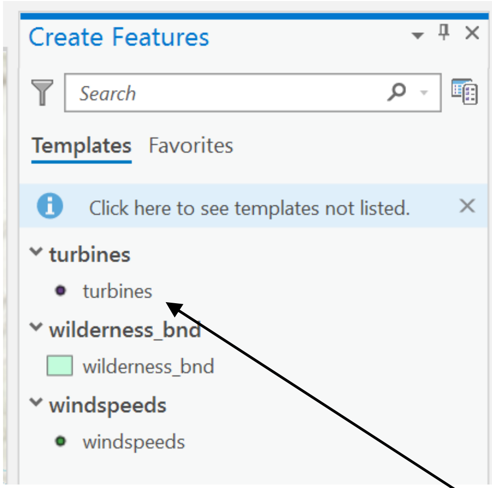
- If you are using a shape file or geopackage format, choose a suitable folder as the *output location* and enter an appropriate name for the *output feature class* (e.g. **turbines**)
- Set the *geometry type* to *point* and choose *OK.*



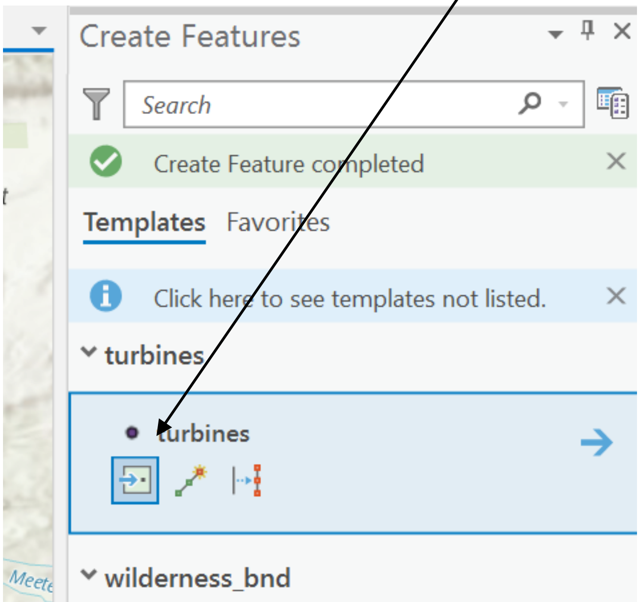
You can set the *coordinate system* to be the same as one of your existing map layers, e.g. **DEM**. This should then select GCS_WGS_1984 (i.e. latitude and longitude referenced to the WGS 1984 datum). The '*Has Z'* option is for enabling 3D shapefiles, which have Z (height) coordinates as well as X and Y coordinates – leave this set to 'No'. Similarly, '*Has M'* enables what are called 'measure coordinates' (distances along a polyline or perimeter), a specialist and seldom-used form of coordinate system. Leave this set to 'No' too and press *run.*

You should now see your new point feature class appear on the left-hand side of the ArcMap screen. We now need to add some points to this new feature class. To do this, we need to start editing the **turbines** map layer:

- Head for the *Edit* menu, then click on *Create:*



You should now be able to add points representing turbines by selecting this tool and digitising on screen



When you have digitised your three turbines, press *Save* on the ribbon to save your edits.

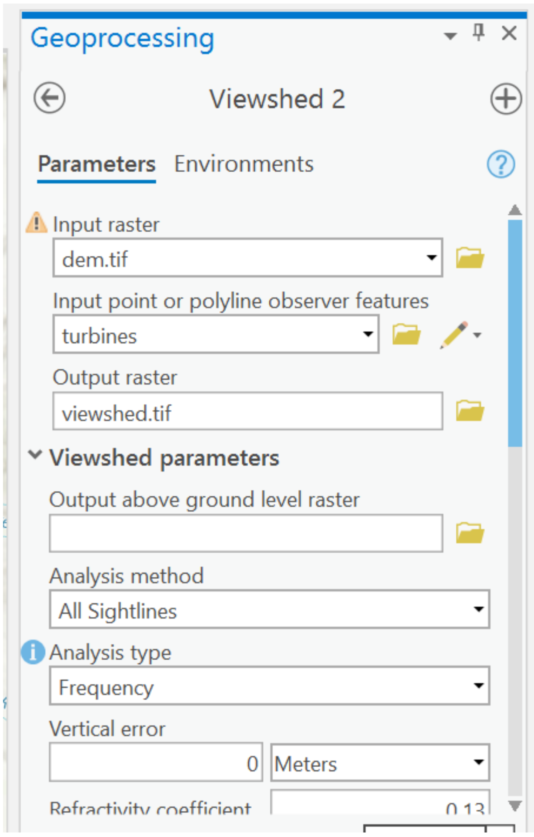
You may find that the last point you digitised is still selected, so to clear the selection, you can right-click on the layer in the left-hand contents panel, choose *selection* and then *clear selection* (if you do not clear the selection, then if you run any tools subsequently, only the selected point will be used for analysis).

1.4 Construct a viewshed

We are now in a position to calculate the area from which the new turbines will be visible. ArcGIS Pro has several tools for such calculations. For calculating viewsheds, there is the older *viewshed* and the newer *viewshed 2* tool. *Viewshed 2* uses a more sophisticated algorithm and can incorporate uncertainty in elevation values into its calculations too, so we will use it here.

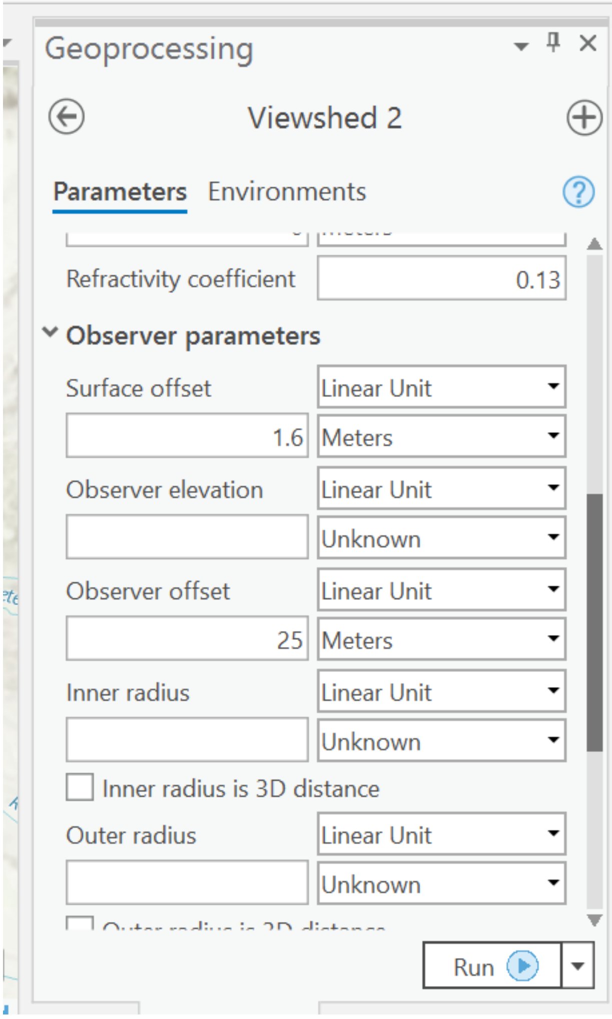
To do this:

- Bring up the geoprocessing panel again (e.g. by clicking on *Analysis* menu and then *tools)*, then search for *viewshed*. Then run *Viewshed 2*.
- Select **dem** as your input raster and choose your **turbines** as your *input point or polyline observer features*.
- Call the *output raster* an appropriate name, such as **viewshed**.
- The fairly daunting looking *refractivity coefficient* is used in specialist applications involving non-visible bits of the spectrum, such as siting mobile phone masts. We can ignore it here.
- The *Analysis method* gives two options: all sightlines, the default, which we will use here, or *perimeter sightlines*, a much quicker but less accurate calculation that works out which pixels along the study area perimeter are visible, radiating these back to observer features.
- The *Frequency* setting means that the number of visible observer points will be calculated.
- Notice also the *vertical error* setting. If we knew the Root Mean Square Error (RMSE) of the elevation values in our DEM, it could be incorporated into viewshed calculations via this setting.



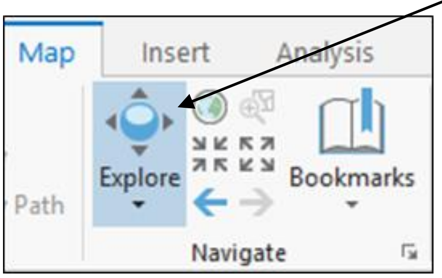
Observer parameters allows us to set a couple of further parameters:

- *Surface offset* will 'raise' the surface of the DEM by a given amount. Enter **1.6** for this as an approximate height in metres of somebody standing in the landscape.
- *Observer offset* raises the surface where the wind turbines are present. Since our turbines are 25 metres tall, enter **25** here (yes – it is a bit confusing that the software terms our turbines 'observer points'!). Note that *Observer elevation* enables us to set the height of our turbines as an absolute elevation value above sea level – not what we want to do here.
- The *Radii* settings can be used to limit the viewshed calculations to just part of a study area, but we will not use them here.



Hit *Run* and you should now see the areas from which your proposed turbines are visible (note that this may take some time, as the calculations are fairly computer-intensive).

After running the tool, you can use the 'explore' tool on the map menu ribbon to investigate the output



The resultant raster grid indicates how many of the input features (wind turbines) are visible from each grid cell. A value of 'null' means that the grid cell cannot be seen by any of the turbines, a value of one means that the grid cell is visible from one turbine, two means that the grid cell is visible from two turbines, and so on.

Note: If you use 'old' *Viewshed* (not *Viewshed 2*), it is possible to take into account the height of both an observer and an object superimposed on a landscape via the attribute table of the observer points layer. You can add new fields to the attributes of the **turbines** that will be used in the calculation (by opening up the map layer's attribute table, and then selecting 'add field' from the button in the top left of this screen). A numeric field called OFFSETA can be added to include an observer height in the calculations (e.g. 1.6 metres, the typical height of a person), whilst a numeric field called OFFSETB can be added to include an object height (e.g. for the turbines, which can vary in height depending on capacity and circumstances). Other fields can be added to the attribute table to control other aspects of the calculation, such as the maximum distance or radius over which calculations are made for each feature.

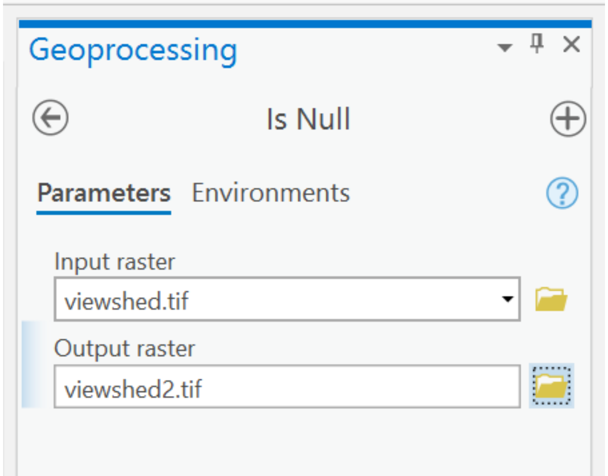
Somewhat harder to handle are obstacles, such as buildings, vegetation and so forth. Although Digital Surface Models (as derived using LiDAR) will include object heights, some 'bare earth' elevation products – particularly those coarser spatial resolution products used over wider areas - will not include these objects. One solution is to 'extrude' a bare earth elevation product by using tools like the 'math / trigonometric / plus' tool within Spatial Analyst to add on approximate object heights for buildings, tree stands, and so forth.

Which towns can see the wind farms and how much?

In the final part of this exercise you will introduce a layer representing towns in the region. You will identify which towns can see the wind farms and how much of the town is affected. *Be aware that your results will differ in detail* *from this example as your viewshed depends entirely upon where you located your wind farms.*

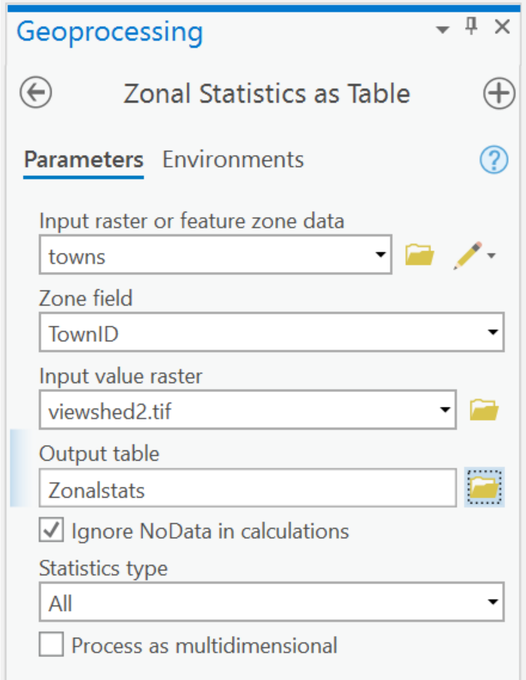
Have a think about how you might work out which towns can see the wind turbines before turning the page.

We have a slight complication to deal with, in that areas where no wind turbines can be seen are coded as 'Null' values (no data). One way of tackling this issue is to first run the *Is Null* tool. Head for the *geoprocessing panel* again, and this time, search for *'null',* then run the *Is Null* tool. This will replace null values with 1 and non-null values (everything else) with 0. Use your viewshed layer as the input and enter a suitable name for the output:



Having created this gridded surface where 1=turbines not visible and 0=turbines visible, one way of finding out the towns from which the wind turbines are visible is to use ArcGIS Pro's 'zonal statistics' feature:

- in the *geoprocessing* panel, search for '*zonal'* and choose *zonal statistics as table*.
- Choose **towns** as the *input raster or feature zone data*. These are the zones for which you will summarise values from a raster image.
- **TownID** – the only integer (whole number) field in this layer's attributes – should automatically be selected as the *zone field*.
- Select your raster output from the '*is null'* tool that you have just created as the *input value raster* and choose an appropriate name for the *output table*, such as **turbine_vis**

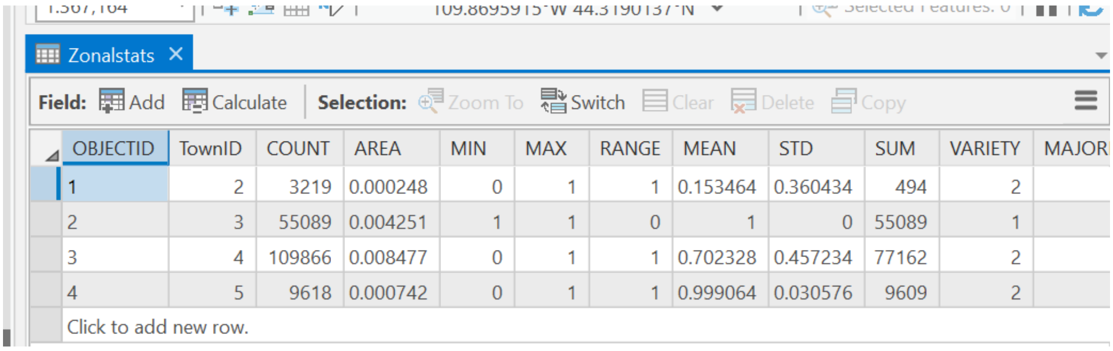


Click on OK and ArcGIS Pro will overlay the town boundaries onto the *Is Null* output raster grid and calculate summary statistics, based on the grid cell values in the viewshed raster. The results will be stored in a new table.

To view this table, you will need to open it up (under *Standalone tables* in the left-hand contents panel) and then right-click on it and choose *open*.

What does the information in the table mean?

Each row in this table represents a different town (from our 'feature zone data' in the input screen). The first column, **OID** (see illustration below), is simply a unique serial number for each town (town 0, town 1, town 2, and so on). The second column, **value**, contains the values of **TownID**, copied over from our original map layer of towns that we fed into Viewshed when we specified a 'zone field'. Based on this logic, the row with Value = 2 represents the town in the southeast corner of our towns map layer with **TownID** = 2, whilst the row with Value = 5 represents the town up in the northwest part of the study site.



OBJECTID	TownID	COUNT	AREA	MIN	MAX	RANGE	MEAN	STD	SUM	VARIETY	MAJOR
1	2	3219	0.000248	0	1	1	0.153464	0.360434	494	2	
2	3	55089	0.004251	1	1	0	1	0	55089	1	
3	4	109866	0.008477	0	1	1	0.702328	0.457234	77162	2	
4	5	9618	0.000742	0	1	1	0.999064	0.030576	9609	2	

Moving along the columns, **count** tells us how many grid cells in our viewshed raster grid lay within the boundary of that particular town. The **area** column is an estimate of the area of the town, though in our case the numbers here are areas in squared degrees of latitude and longitude, so should not be considered as meaningful (this is because our original DEM's reference system was measured in degrees of latitude and longitude).

The relevant information for us comes in the following set of columns. These are a summary of the numbers stored in the raster viewshed for each town. '**Min'** is the smallest number stored in any of the grid cells that lay inside the boundary of a particular town, whilst '**Max'** is the largest number in any of these grid cells. Looking at the first row in the example shown above, you should be able to see that the town with **townid** = 2 (i.e. the one in the southeast corner) had a value for both '**Min'** and '**Max'** of 1. Since 1 means 'turbines not visible', this means that *none* of the turbines could be seen from anywhere in this town. In contrast, looking at the town with **townid** = 4 (i.e. the one up in the northwest of the study area), the column 'Min' contains a 0. This means that in part of the town, at least one of the turbines was visible. In the remaining two towns, the **min** field is 0. The **mean** field of 0.70 tells us that the turbines were not visible in 70% of the town's area. N.B. If your table looks different to the example above, do not be alarmed. You are likely to have placed your wind turbines in slightly different positions when drawing them on-screen.

The other columns of information – **range**, **sum**, and so on – contain other summary measures of the values of our viewshed raster for each town. However, for our purposes here, they are less useful than the **mean, max** and **min** columns.

This final calculation may seem excessively complicated. A simpler alternative, for example, might be to overlay the towns map layer on top of our viewshed raster and simply look at how visible each town is by eye. The advantage of using 'zonal statistics', however, would come if we were looking at many different locations (e.g. several hundred houses, not just 4 towns) and wished to automatically calculate inter-visibility for all of them using ArcGIS Pro.

Note:

- There are other approaches to working out which turbines are visible from where that would also work – the approach above is just one of several possible solutions. You may be able to think of other ways of approaching the same calculation.

Additional background

RenewableUK
Web link: http://www.renewableuk.com/