

Night Lights and ArcGIS: A Brief Guide*

Matt Lowe

January 12, 2014

1 Intro

Economists have recently begun making more use of high-resolution data on light density measured by satellites at night (Bleakley and Lin (2012), Henderson et al. (2012)¹, Michalopoulos and Papaioannou (2013), Lowe (2014), Storeygard (2012), Pinkovskiy (2013)). The data has been shown, by and large, to proxy well for local economic activity, and to correlate fairly strongly with other welfare proxies. It lends itself particularly well to projects in which a policy with highly localised effects is being evaluated, or to policy evaluation in countries with poor or non-existent sub-national GDP data. For these reasons, it is likely that once the mystery behind the data clears it will be used more and more in empirical work.

This brief guide has two primary motivations: the first is that I am returning to work using the lights data, and writing this guide gives me a good means of fully documenting my dealings with the data. The second motivation is the frustration I found when searching for help using the lights data and ArcGIS more generally. Not much exists specifically for economists (though Dell's excellent guide here is an exception: http://scholar.harvard.edu/files/dell/files/090110combined_gis_notes.pdf as is Kudamatsu's course here (which I only recently found): http://people.su.se/~mkuda/gis_lecture.html²), and though I got there in the end, I took the scenic route.

This guide then aims to take the reader step-by-step from a point of wanting to use the data, to having a dataset ready for analysis. I will be sparse on general ArcGIS tips, and focus on clear steps to getting the lights data ready for analysis. Hopefully the guide will show that the data is easy to use once you know how, though some basic skills required in ArcGIS will be illuminated along the way. If you plan to use ArcGIS a lot, head straight to Section 4 first for an overview of projections and scripting.

*Comments, suggestions and corrections welcome to mlope@mit.edu.

¹Replication data and code available here: http://www.econ.brown.edu/faculty/henderson/lights_hsw_data.html.

²I cannot recommend these two guides highly enough. Kudamatsu's in particular includes lots of paper replication with the necessary data posted. In addition, a remote-sensing expert's guide to the lights data is here: http://sedac.ciesin.columbia.edu/binaries/web/sedac/thematic-guides/ciesin_nl_tg.pdf.

2 Lights Data

2.1 Downloading

The lights data can be downloaded here: <http://ngdc.noaa.gov/eog/dmsp/downloadV4composites.html>. The data is available for 1992-2010, and for some years, data was collected by two satellites, so there are two files that can be downloaded. Good practice is to download both, and in the analysis use an average of the two.

There are two sets of zipped files for each satellite-year: “Average Visible, Stable Lights, & Cloud Free Coverages” and “Average Lights x Pct”. The latter multiplies each pixel by the percent frequency of light detection. The former is probably what you want - the latter is used to infer gas flaring volumes.

Each file has the name of the satellite and the year (e.g. F182010 is the file created from DMSP (Defense Meteorological Satellite Program) satellite number F18 for the year 2010). Once you download the data, it needs to be unzipped. These are large files - once all unzipped, the data from F182010 is around 2GB (though you can probably get away with only keeping one-third of the files). All the data for 1992-2010 (including both satellites for a given year when available) will be around 60GB.

Once unzipped, you get three choices - each with a tiff and tfw file. The tiff file is an image file you can open in ArcGIS; the tfw file is a text file associated with the tiff file. It stores X and Y pixel size, rotational information and world coordinates. From the README in the zipped file³, these are the details of each choice:

- F1?YYYY_v4c_cf_cvg.tif: Cloud-free coverages tally the total number of observations that went into each 30 arcsecond grid cell. This image can be used to identify areas with low numbers of observations where the quality is reduced. In some years there are areas with zero cloud-free observations in certain locations.
- F1?YYYY_v4c_avg_vis.tif: Raw avg_vis contains the average of the visible band digital number values with no further filtering. Data values range from 0-63. Areas with zero cloud-free observations are represented by the value 255.
- F1?YYYY_v4c_stable_lights.avg_vis.tif: The cleaned up avg_vis contains the lights from cities, towns, and other sites with persistent lighting, including gas flares. Ephemeral events, such as fires have been discarded. Then the background noise was identified and replaced with values of zero. Data values range from 1-63. Areas with zero cloud-free observations are represented by the value 255.

³The README also notes that the digital number (DN) values assigned to pixels are not strictly comparable from one year to the next since the sensors used have no on-board calibration. My guess is that this is not a problem provided we use year dummy variables in subsequent regressions.

The third tiff file is the one we will usually be interested in. You can see by now that these composite images of light density across the world combine only cloud-free observations, and each pixel is given a value up to 63 (or 255 if there were no cloud-free observations at all), reflecting the brightness of the lights. Some cleaning of the composite still needs to be done, but first an aside.

2.2 Data types in ArcGIS

By now you may have worked out that there are a few main categories of data that be used within ArcGIS. The lights data is a type of raster file (tiff is a type of raster image file), whereas two other main types are feature files (file extension .shp) and TIN files.

Raster data is comprised of pixels, with the size of the pixel reflecting the resolution of the raster. An example (lights data):

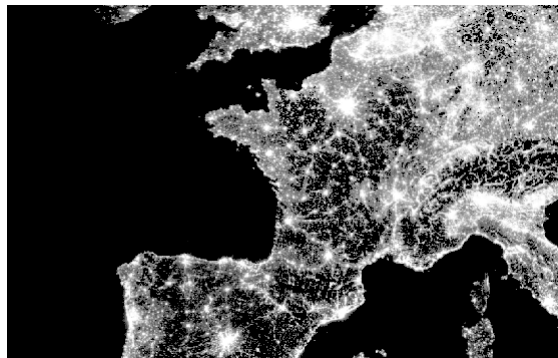


Figure 1: Lights in Europe

Feature data includes point, lines and polygons - examples could be country borders (each country is a polygon), roads (lines) or data on natural resource deposits, with each point reflecting the location of the deposit. A shapefile of country borders:

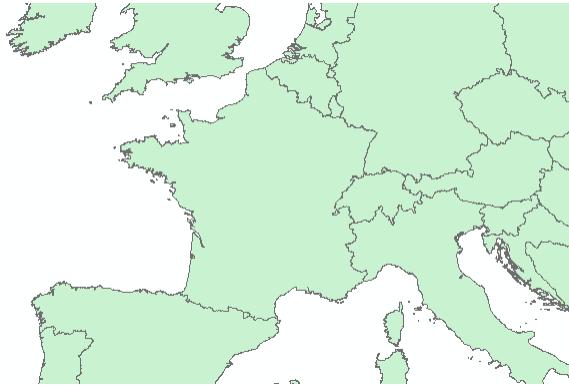


Figure 2: Europe Country Borders

TIN files contain three-dimensional data - an example would be elevation data. It looks like this:

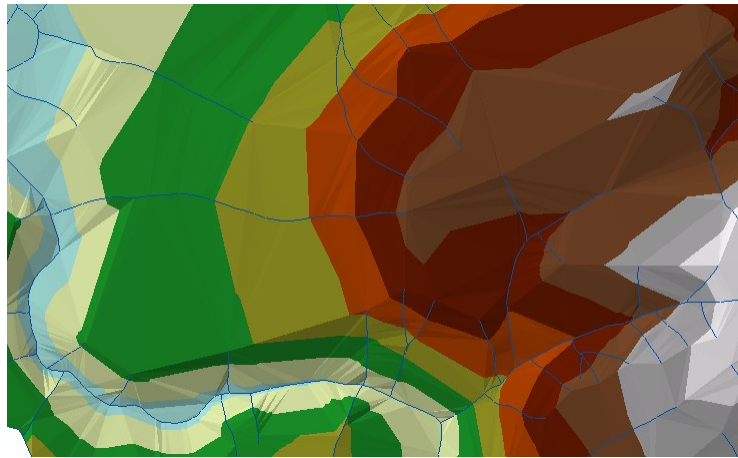



Figure 3: TIN Data

For now we will worry only about the light raster data.

2.3 Getting into ArcGIS

ArcGIS for Desktop 10.1 (current version) is a suite of software. For our purposes, we need only concern ourselves with ArcMap, within which we can visualise and process data. There is also ArcCatalog - crucial for managing your GIS data, and better than managing within usual file explorers which don't collect the many files associated with one shapefile (for example) together, but Catalog can be accessed from within ArcMap (click Catalog - ). ArcGIS for Desktop is expensive, though a free trial is available here: <http://www.esri.com/software/arcgis/arcgis-for-desktop>, and many economics departments will have a license⁴.

Once ArcMap is loaded, it is ultimately helpful to know how to use the Model Builder and also basic Python programming to process GIS data efficiently (particularly when loops are required), but I will focus initially on the basic menu commands. Once in ArcMap, click “OK” when prompted to load a blank map, and this is what you see.

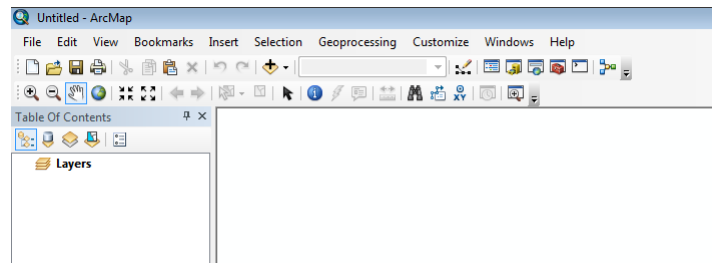



Figure 4: ArcMap

I show only the top-left, since there is not much else to see right now. The empty space is the workspace where maps will appear. In general, you proceed by loading ‘layers’ of maps, with the list of ‘layers’ loaded showing on the left. You can then switch them on and off by clicking a tick box, and move them above or below each other - this affects the order in which they are stacked, and thus what you see in the workspace.

The first task is to load the lights raster. To do so, click “Add Data” () , browse to the correct tiff file and open. Arc may ask to “build pyramids” - say yes⁵. I downloaded the F152006 data, so this is what I see:

⁴At MIT, you can get ArcGIS on your own laptop from here: <http://ist.mit.edu/arcgis/10/win>, though it is only operational when you are connected to MIT servers. You can still access Arc off-campus by logging into the MIT VPN (using Vmware View Client for example).

⁵This just allows the data to load faster by only showing the raster at a lower-resolution when it can get away with it. More info here: http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=Raster_pyramids.

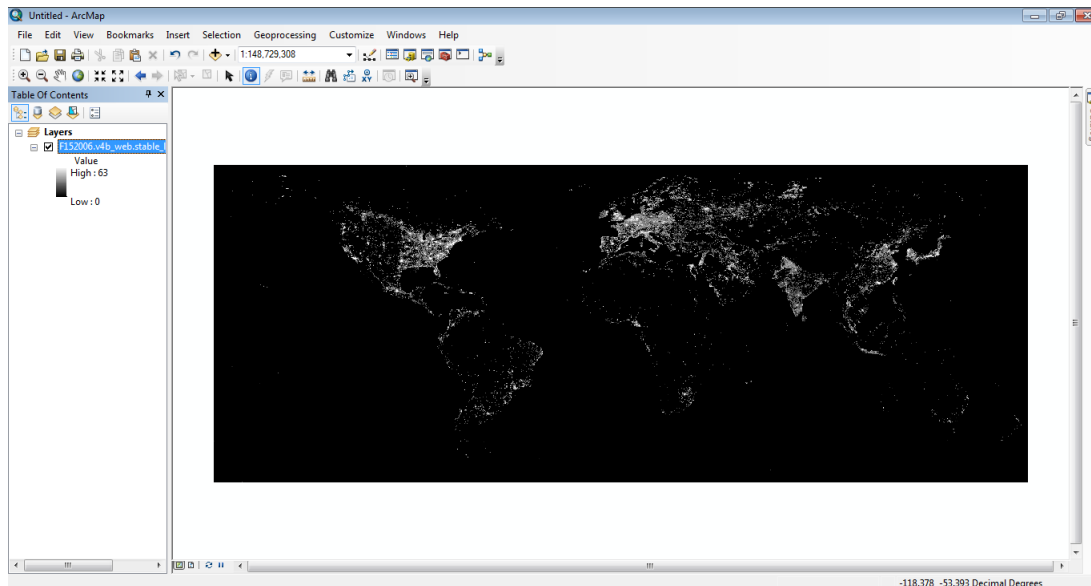


Figure 5: Lights in ArcMap

You can see on the left that the highest pixel value in this raster is 63 and the lowest is 0 - no pixels have the value 255, so as far as I can see, in the 2010 data there were no pixels with zero cloud-free observations. It is easy to zoom in and out on the data using the zoom buttons in the top-left.

Now the data is loaded, let us start cleaning.

2.4 Light Data Cleaning

2.4.1 Converting

I have seen it mentioned elsewhere (e.g. Kudamatsu’s course) that it may be desirable to convert the light rasters to another raster format, ESRI grid, for faster processing⁶. We can use the “Raster To Other Format” tool for this. First open ArcToolbox (🗂️). You will see a menu appear with an array of categories of tools - these will be used again and again in order to manipulate and analyse the GIS data⁷. The tool we want is in **Conversion Tools→To Raster→Raster To Other Format**. Simply specify the input raster, output filepath and the raster format as **“GRID”**.

⁶I’m not sure whether the difference is noticeable though. I tried Clipping both rasters, and this operation at least took almost exactly the same time.

⁷Though since there are so many of them, it may be quicker to use the Search tool to find the tool you want. Click the icon by ArcToolbox (🔍).

Tools like these run in the background⁸, allowing you to keep using Arc while the tool is running. When the process is done, a pop-up box will appear in the bottom-right.

2.4.2 Re-classifying

Though in this case we don't need to re-classify a pixel value to missing, it is useful to know how. The relevant category here is Spatial Analyst Tools⁹→Reclass→Reclassify¹⁰. This is simple enough: I choose F152006.v4b_web.stable_lights.avg_vis.tif as the input raster to be re-classified, and Value as the field of interest (the Count variable contains number of pixels with a given value). Click "Unique" and a list will populate with all the unique pixel-values found in the raster. If it were necessary, we would click "Add Entry" and type "255" in the Old values column, and "NoData" in the New values column. Then choose a name and location for the output raster, and click OK.

You may notice that in the bottom right of the tool window is the option to "Show Help". This can be incredibly useful when stuck with how to use a tool. You may also have noticed that the default output raster filepath ends with something like "\Default.gdb\Reclass.tif2". The .gdb extension is that of a geodatabase. A good way of organising a collection of spatial data (e.g. all that associated with one project) is within a geodatabase like this. You can create these using ArcCatalog¹¹.

2.4.3 Averaging

For years where there are two composites available (from two satellites), after re-classifying (if necessary), you may want to create a new raster where the pixel value is an average of the two. To do this, add both composites (e.g. F152006 and F162006) as layers in ArcMap. We can then go to Spatial Analyst→Raster Calculator. Using the line: $(\text{Float}(\text{raster1}) + \text{Float}(\text{raster2}))/2$ (where you double-click the layer names to get raster1 and raster2 and you can also double-click to get Float and other operations) we get the desired output. The calculation takes a couple of minutes.

2.4.4 Gas Flares

A couple of cleaning problems remain with the stable_lights tiff file. Firstly, light from gas flares still show up in the data, which will be problematic for certain countries (to the extent that we don't think of gas flares as proxying for economic activity). It may be wise to report regression


⁸Though this can be changed on the main toolbar by clicking Geoprocessing→Geoprocessing Options, then uncheck Enable.


⁹Note for these tools you require the Spatial Analyst extension. Go to the toolbar, click Customize→Extensions, and make sure that Spatial Analyst is ticked. Otherwise, this may help if you are having trouble: <http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//00030000000110000000.htm>.

¹⁰Alternatively you can use Spatial Analyst Tools→Map Algebra→Raster Calculator.

¹¹I won't add details here, but for the interested reader, see <http://ocw.tufts.edu/data/54/639533.pdf>, and distinctions between types of geodatabases here: http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=Types_of_geodatabases.

results using first the raw lights data, and then for robustness, again with the gas flare-adjusted data.

Pixels with gas flares can be excluded from the analysis using shapefiles available here¹²: http://ngdc.noaa.gov/eog/interest/gas_flares_countries_shapefiles.html. Download all the shapefiles - again you have to unzip, and you will notice that each shapefile (.shp extension) has associated with it other file types: .dbf, .prj, .shx. If you look at the data using Catalog (click ) all these files are considered together. Otherwise, these files should be kept in the same directory and not be deleted separately.

Within ArcMap, you should click  again to load in these gas flare shapefiles (only the .shp files will be options). An error message appears¹³: Click OK. I think this relates to problems that three of the gas flare shapefiles have incorrect spatial reference data. We will see this in a second.

Zoomed in here¹⁴, you can see the location of gas flares in North Africa, and also some in the sea (these are less important for us since we would have dropped that data anyway when considering policies within countries):

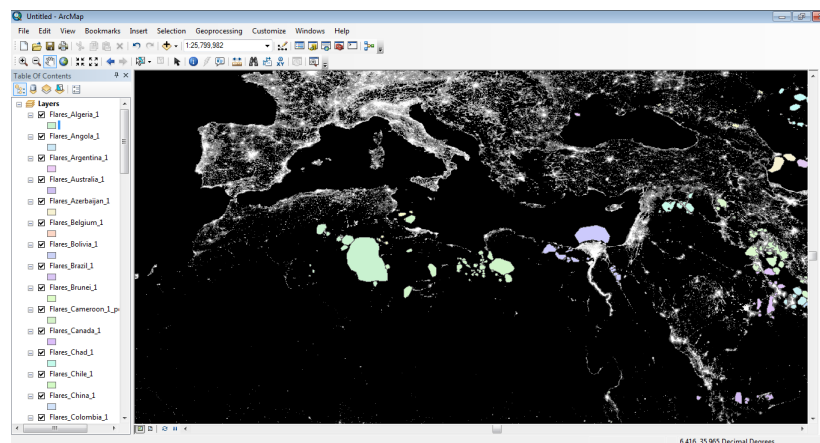


Figure 6: Gas Flare Polygons

Zooming in a little more, you can see what the gas flare looks like in the data, before and after it is covered by the polygon (this is for Algeria):

¹²This approach is not ideal - we only have a cross-section of where gas flares are located, rather than a panel - but probably the best we can do for now.

¹³The message reads: “Warning, inconsistent extent! One or more of the added layers has an extent that is not consistent with the associated spatial reference information. Re-projecting the data in such a layer may lead to unexpected behavior.”

¹⁴These figures use the 2010 data (F182010).

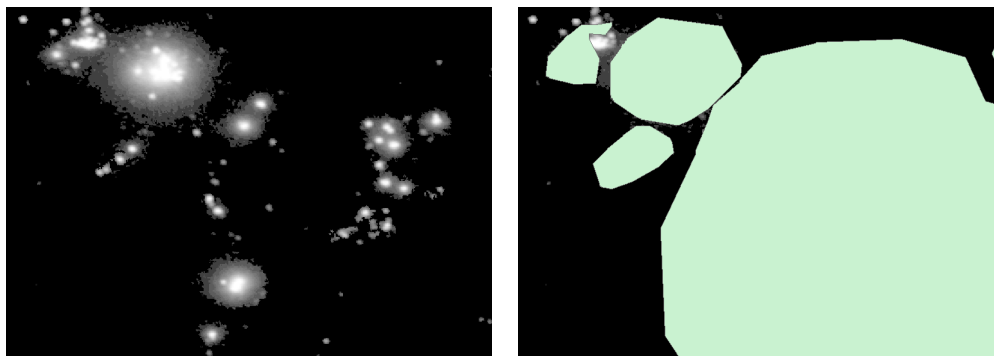


Figure 7: Gas Flares in Algeria

The polygon doesn't seem to perfectly cover the gas flare, but it is the best we have as far as I know.

Now note that if you click “Full Extent” in the top-left (🌐), Arc will zoom out until every layer you have added is completely visible. We can see now that a number of the gas flare polygons have incorrect geo-referencing¹⁵ - they are located outside of the world. Clicking “Identify” (🔍) we can click on the rogue polygons to determine which country they should be associated with. With a bit of sleuthing, we find them to be Ghana, Mauritania and Côte d'Ivoire. Without a better option right now, we will just omit to gas flare adjust in these countries. Right-click these three layers on the left and click “Remove”. By doing this we lose them from the workspace¹⁶.

It makes sense now to join all the country gas flare polygons into one shapefile - we will then use this shapefile of global gas flares to determine which pixels in the lights raster to set to missing. We use ArcToolbox again, and this time go to Analysis Tools→Overlay→Union. We need to specify the Input Features as all the separate gas flare shapefiles, give a name to the output shapefile, and keep the default options (including allowing gaps, which is crucial here). The tool executes and the unified shapefile is added to the workspace. We can now remove all the separate flare files (highlight, right-click, Remove). The layer we are left with, if we switch the lights off, is:

¹⁵I did check at some stage whether the mistake was obvious, e.g. the coordinates multiplied by 10, but it didn't seem so.

¹⁶Though we do not delete them. We could do that using Catalog.



Figure 8: Gas Flare Layer

Since we only want to analyse non-gas flare light density within countries, the next step is to bring in country boundaries which we will clip down to. You can get these from here: <http://www.diva-gis.org/Data>. Add the country boundaries shapefile into ArcMap. We want to clip the lights raster down to exclude gas flares and the sea¹⁷.

First we use **Analysis Tools→Overlay→Erase**. Choose the **country shapefile as Input Features**, and the global **gas flare shapefile** as the **Erase Features**. You are left with a countries shapefile with gaps where there are gas flares. It should look something like this, zoomed in on South America:

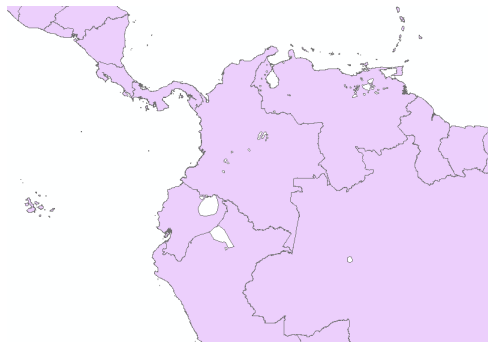


Figure 9: Gas Flares Erased

¹⁷Note: it should be possible to clip the lights raster directly to exclude gas flares without needing to first bring in the country data, but for some reason I could only clip the lights data down to where there *are* gas flares as opposed to where there *aren't*. We need the country data anyway later, so this doesn't matter so much.

Now go to **Data Management Tools→Raster→Raster Processing→Clip**. Choose the **lights raster** as the **input raster**, and the **countries less gas flares shapefile** as the **output extent**. Tick the box for **“Use Input Features as Clipping Geometry”** - this ensures we clip the lights down to the shape of the countries, not to a rectangle which encloses the countries. This clipping may take some time (it actually took two hours for me).

The result is a lights raster with data only for countries and missing data where we suspect there are gas flares.

2.4.5 Country Clipping

The next thing we might like to do is restrict the lights data to a subset of countries. Suppose here that we are only interested in China. We want to Select only that part of the country boundaries layer we are interested in - go to the main toolbar, **Select→Select by Attributes**. Pick the countries layer at the top, double-click **“NAME”** and **“=”** and then click **“Get Unique Values”**. You can now scroll all the different countries in the data. (I hadn’t said earlier, but each layer has a database associated with it - to see this, you can always right-click the layer on the left and click **“Open Attribute Table”**. For the countries shapefile the unit is the polygon, and the data includes country name, continent, size etc.). Double-click **‘China’** and you will have populated the text box with **“NAME”=‘China’**. Click **OK**. Going back to the map you will see that the border of China is now lit up in turquoise (unless the colour settings have been changed).

Now we want to clip the lights down to the border. For this, we do the same as earlier: **Data Management Tools→Raster→Raster Processing→Clip**. We choose the countries shapefile as the output extent - ArcGIS remembers that we selected China, and only considers the selection. And the result:



Figure 10: China Clipped

2.4.6 Gridding and Exporting

We now have gas flare removed light data for China (in 2006, in my case). Going to the Attribute Table, I can right-click the Count heading and click Statistics to find that there are 13,624,457 pixels remaining. 11,283,800 are equal to zero - in China, in 2006, dropping gas-flare areas, there is 83% bottom-coding¹⁸. The countries Attribute Table tells us that China spans 9.4 million square kilometres, which makes sense, since each pixel is 30x30 arcseconds¹⁹ (or around 0.86 km squared at the equator).

A pixel-level dataset would have a ton of observations. Otherwise we could make a county- or a grid-level dataset. I will go through the latter here. Let's suppose we want to grid up to 12x12 pixels, i.e. 6x6 arcminutes²⁰. We go to **Data Management Tools**→**Feature Class**→**Create Fishnet**. **Choose Template Extent** to be the same as the **China lights layer**. **Cell size width and height are in degrees here, so type in 0.1 for both**. We can set **Number of Rows and Number of Columns to zero** - they will be calculated automatically given the specified width and height. You probably want to **uncheck "Create Label Points"**. Finally, specify the **Geometry Type as Polygon** (we are creating a grid of polygons and will then find the mean lights within each polygon).

You can see from this, zoomed in, that the grid is aligned with the pixels as we wanted:

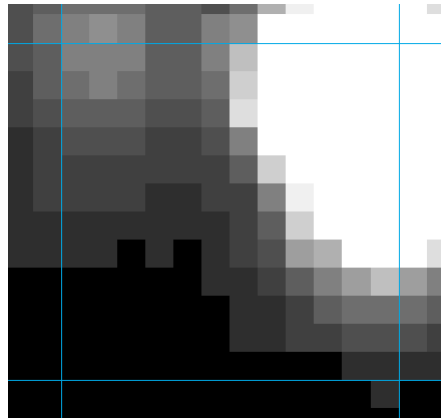


Figure 11: Aligned Pixels

¹⁸With future analysis in mind, it may be desirable to drop areas with no population - we can do this by using Gridded Population of the World data here: <http://sedac.ciesin.columbia.edu/data/collection/gpw-v3>.

¹⁹There are 60 arcseconds in an arcminute, and 60 arcminutes in a degree. A pixel is then 1/120 x 1/120 degrees.

²⁰If you try making a pixel-level grid you will likely get an error that the output shapefile exceeds the 2GB limit. You can get around this by saving to a file geodatabase, where the file size limit is 1TB. When using very large datasets then, it makes sense to use file geodatabases.

The grid though is a rectangle - we want to clip it to the size of the lights, as usual. Since we can only Clip down to the size and shape of a polygon, not to that of a raster (I think), we first have to convert the lights raster to a polygon. For this we go to Conversion Tools→From Raster→Raster to Polygon. Choose the China lights raster as the input, and uncheck “Simplify polygons” - we want the output polygon to have the exact perimeter of the China lights.

Now that we have the right-sized polygon, we go to Analysis Tools→Extract→Clip. Choose the grid as input features, and the newly created polygon as the clip features. Now we get a grid cut down to cover the lights exactly, with appropriately blocky borders, as can be seen here:

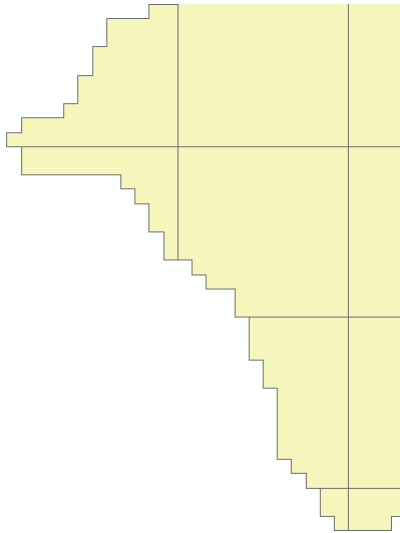


Figure 12: Blocky Borders

Some of the grid squares are now only partial, but that doesn’t matter - we will take unweighted averages of light pixel value within each grid cell, and we could easily later control for cell size in regressions (or use cell fixed effects).

If you right-click the grid layer and open the attribute table, you will see we have around 100,000 grid cells, each with a unique ID (column FID). For each grid cell we want to generate a new variable containing mean luminosity. For this we go to Spatial Analyst Tools→Zonal→Zonal Statistics as Table. First enter the layer that defines the zones by which we want to calculate statistics - i.e. the grid. Choose FID as the zone field, the China lights as the input raster and MEAN as the statistics type. Enter .dbf as the file extension - this means the table can be read directly into Stata - and

remember the name and location of this .dbf file for later. Click OK and a table will appear on the left at the bottom of the other layers - right-click, Open, and you will see that we now have a grid-cell level dataset with variables FID, COUNT, AREA and MEAN. The first is the ID of the grid-cell (we can use this to merge with the grid attribute table we already have), the second is the number of pixels, the third is area in decimal degrees (though this doesn't mean much because of distance distortions - more on this later) and the final variable is the mean light pixel value.

Our work in ArcGIS is done - we can read this table, and others, directly into Stata and do cleaning/merging there²¹. Before exiting Arc, you should save the Arc map document (extension .mxd) by File→Save. Note here that you are not saving the feature class/raster data itself, only the way you have layered and organised them within ArcMap. The map document just contains the details of where the layer files in use are located and how you put them together²².

2.4.7 Once in Stata

You can read in the .dbf file using the “odbc load” command - provided the .dbf file has a name which is 5 characters or less (no idea why this is so). Specify the working directory to where the grid data is located using “cd...” then type

```
odbc load, table("grid.dbf") dsn("dBASE Files") lowercase clear
```

Lowercase ensures variables names are all in lower case. You can then save the dataset as a .dta type. Alternatively there is the command “shp2dta”²³ for if you went the longer route in Arc by joining the table to the grid and creating a new shapefile (these details were in an earlier footnote). Try typing

```
shp2dta using "grid.shp", data("grid_data") coor("grid_coor")
```

This will create two Stata datasets - one containing the main data, the other containing the lat-long coordinates of the corners of every grid-cell. One advantage of this over “odbc” is that you can use the options “gcentroids” along with “genid” to have Stata generate variables containing

²¹It is probably to be preferred to do this in Stata, but we could also do the merging in Arc: we can merge the newly created table with the grid attribute table by right-clicking the grid layer, then Joins and Relates→Joins. Set the field to join on as FID, the table to join as the one just created, set field as FID again and Keep all records. If prompted, say Yes to indexing. If you open the attribute table now, you will see the variables have been joined, and though some are unnecessary (e.g. FID is repeated now), it is easiest to drop these later in Stata. Once we are done joining, we need to export the layer as a new shapefile. Right-click the grid layer, Data→Export Data. The default options are fine, just choose shapefile as the output type of the output feature class. This new grid layer has a .dbf file associated with it which can read into Stata.

²²Another point here: Arc has some bug that I encountered once where the .mxd file can get larger and larger every time you open it. You'll find some answers to this particular bug here: <http://support.esri.com/en/knowledgebase/techarticles/detail/20872>.

²³If you don't have it, type “findit shp2dta” and you can download and install it.

the coordinates of the centroid of each polygon. This is probably a more efficient way than doing the equivalent within Arc.

3 Calculating Distances

This section isn't technically related to the lights data (feel free to skip). In this section I bring in some railroads data, and show how you can calculate the distance to the nearest railroad from the centroid of each grid cell.

First download railroad data for China here: <http://www.diva-gis.org/gdata> (the source is Digital Chart of the World). Add it as a layer in ArcMap. It seems that the best way to get accurate distance measures is to get the latitude and longitude of the nearest railway in Arc, and then use the “globdist” ado in Stata to calculate the distance from each grid cell to the nearest railway²⁴. This method means we don't need to bother defining new projections for our layers in ArcGIS (whereas if we wanted to calculate distances within Arc we would need to do this). In Arc, go to Analysis Tools→Proximity→Near²⁵. Choose the grid as the input features and the railroads and the near features. Be sure to check Location - this ensures the X and Y coordinates of the nearest railroad are stored, in addition to the distance (which will be of no use as it will be in degrees). This calculation may take a while (it took me around 20 mins²⁶). Next we need to add fields to the grid attribute table containing the coordinates of the centroid of each cell²⁷. Open the attribute table of the grid layer. Click Table Options (☰) and Add Field - add two new fields, one for latitude and one for longitude, both as Float variables. You can now right-click the headings of these new (empty) fields and click Calculate Geometry. Click Yes after being warned that your changes will be irreversible²⁸. Choose X coordinate of centroid and Y coordinate of centroid for the two new fields. Stick with the default coordinate system - it should be WGS 1984.

We can now read the .dbf file into Stata as we did earlier (or can use the shp2dta command to read in both the .shp and .dbf). We now use globdist to calculate distances:

(syntax) globdist newvar [if] [in] , lat0(#|...) lon0(#|...) [options]

(example) globdist distrail, lat0(Y) lon0(X) latvar(NEAR_Y) lonvar(NEAR_X)

²⁴VINCENTY is another Stata module that does this, though I do not know what the differences are.

²⁵In this case, the distance calculated by Arc is distance from the (nearest) edge of the polygon (i.e. the grid cell), as opposed to e.g. the centroid.

²⁶And thirty-two hours for a pixel-level dataset...

²⁷An alternative to the steps that follow is to use the “gencentroids” option together with “genid” when using “shp2dta” in Stata (as mentioned earlier).

²⁸You can do this instead within an Edit session and easily undo any results. Click Customize→Toolbars→Editor if the Editor toolbar is not showing.

This command generates a new variable (practically instantly, which is a nice change from Arc) named “distrail” containing the distance in km (can specify miles instead if preferred) from the coordinate specified in lat0, lon0 (the reference coordinates) to that in latvar, lonvar. The latvar and lonvar options are superfluous if the variables NEAR_Y and NEAR_X were named “lat” and “lon”. The distances calculated here are from the centroid of each grid cell, not from the nearest edge, as would have been calculated using the Near tool in Arc.

Should we want to do the distance the calculations in Arc instead, we should first re-project the needed layers (grid and railroads) to a Projected Coordinate System (i.e. one with location in metres rather than degrees). We can use the Project tool for this (Data Management Tools→Projections and Transformations→Feature→Project). The key with this is to choose a Projection most suited to the task in hand - i.e. measuring distances between points within China. In general, using the appropriate UTM projection for distances within small regions will lead to accurate measurement. For China as a whole (and large areas in general), I’m not sure what projection is best. More on this coming up.

4 General ArcGIS

There are plenty of other important aspects to ArcGIS that I have ignored so far. Here are some.

4.1 Projections

This has come up a little, but I have skirted over the details so far. A good understanding of projections is fairly fundamental - mis-understanding can lead to inaccuracies when calculating distances and areas. I will explain why.

The earth is roughly spherical, but our analysis is of two-dimensional maps, so we have to somehow ‘project’ the earth’s surface onto a plane. There are many different ways of doing this, and each generates its own distortions along different dimensions - that means the type of projection that is appropriate will depend on what you are trying to calculate. An example of a distortion is this: degrees of latitude and longitude are commonly used as a location’s ‘coordinates’. But you will realise from looking at a globe (as opposed to a two-dimensional map) that one degree of longitude is much greater in terms of distance at the equator than at the poles. Using this projection for calculating (and comparing) distances near to the equator and far from the equator will lead to large distortions²⁹.

There are two types of coordinate system (within which there are many different choices of projections): Geographic and Projected. The former has locations coded in terms of degrees,

²⁹On the other hand, it is probably a good approach to use the Near tool on this data to get coordinates, and then use the globdist module in Stata to calculate distances. This way we get distance measures without ever having to change the projection.

minutes and seconds (and the most common coordinate system you will find is one of these: WGS 1984). The latter has locations in terms of metres from some reference point. It follows that distance calculations using Geographic and Projected coordinate systems will be in degrees and metres respectively.

In ArcGIS, each layer we introduce will (tend to) have a projection associated with it. For example, the railroads data we used earlier uses WGS 1984. Using tools in Arc we can re-project - changing the projection - if it we want to do calculations that will be more accurate using a different projection.


In general, there are two rules to follow:

1. Choose the right projection for purpose. If a layer does not have a projection (no .prj file associated with it), use the tool Define Projection. If it already has one, use the tool Project/Project Raster. One thing to note is that changing the projection changes the way the map looks, but to see this, you will have to open a new map document and add the newly projected layer first. The look of the workspace follows the projection of the first layer added. To see more on what projections to use, read Dell's Section 1.4 here³⁰. A general rule is that if the analysis is only for a small region (e.g. one Indian state), the best option is probably to opt for the associated UTM projection, which has very little distortion in any dimension. Again, Dell has more details.
2. Make sure layers use the same projection before carrying out calculations using those layers.

You can check the current projection of a layer by right-clicking and clicking Properties.

4.2 Model Builder

There are three main advantages of using the Model Builder over menus: (i) we can loop over commands with Model Builder; (ii) we can save the Models so that others can easily replicate; and (iii) we can generate python scripts easily by exporting the results to python code.

The Model Builder works like this: click the icon , then drag and drop the tool you want to use into the empty space. Double-click the block that appears with the tool name, and use the same menu as usual to change the tool settings, then click OK. The Model Builder will have put some blocks together automatically: input features, tool, output features. We can add more processes to the model (and edit existing ones by double-clicking the blocks) and when ready to run, click Model→Run.

³⁰Be thorough in choosing projections - I opted for equidistant cylindrical for my analysis of distances to railway lines on the African continent, thinking the distance distortion was minimal, but it seems not to be the case. I re-did the analysis using WGS 1984 to get coordinates, and then used globdist in Stata. (It is interesting to note that my new, more accurate, distance measure had a correlation of 0.94 with the old one, so in this case getting the projection wrong did not affect my regression results).

So far, nothing new. But you can save your model by clicking Model→Save As. Models have to be saved within toolboxes, so create one to contain it: click New Toolbox (📁) in the top-right of the Save window³¹. Now save the model. If you want to run the model later, you can find it in Catalog and double-click. If you want to edit, right-click the model in Catalog and click Edit. To export to python, click Model (on the Model Builder toolbar) then Export→Export to Python Script. You then get a .py file that you can edit and run from within Python (just go to Windows Start→ArcGIS→Python 2.7→IDLE (Python GUI)). In principle, running the script should do exactly the same as running the Model would. In practice, sometimes the exported code is not perfect - so you may need to check and edit from within Python before you run.

4.3 Python Scripting

Python scripts are again useful for automation and later replication. They can be unwieldy though - some statements are long and not the kind you would memorise. The best bet here is to use the Model Builder to generate the code, then make minor edits within the script.

A short example script is here³²:

³¹You can also add toolboxes by right-clicking within ArcToolbox where all the tools are listed, then click Add Toolbox.

³²A longer template geared towards cleaning the lights data is here: <http://economics.mit.edu/grad/mlowe/papers>.

```

# Import arcpy module
import arcpy
from arcpy import env
# Set working directory ("cd" in Stata). Always use / or \\ for filepath
# \ is for line continuation
env.workspace = "C:/ArcFiles"

# Variables
fact = "factory_2011.shp"
# Use print to display text
print "Cleaning raw data."

# Try-except for error messages. If problem with any indented message following try, will get message.
try:

# Example tool: delete NumWorkForce from factory_2011 data
arcpy.DeleteField_management(fact,"NumWorkForce")

# For loop example
vars = ['workforce', 'investment', 'isic']
for i in vars:
arcpy.DeleteField_management(fact, i)

# Example tool: select from factory_2011 (syntax takes input, output and clause)
arcpy.Select_analysis(fact,
                      "factorylocation", "[location] = 'factory'")
# Note: can continue over multiple lines if within brackets

# Example of while loop and concatenation:
x = 1985
while x<2011:
print str(x)
arcpy.Select_analysis("factorylocation", "fact"+str(x)+".shp", '[ApproYear] = '+str(x))
x = x+1


# Try-except for error messages.
except:
print arcpy.GetMessages()

## Finish ##
print "(END) Job done."

```

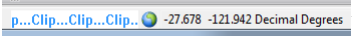
Figure 13:

You will notice that # is used for comments (Stata uses *). To run the script, click Run→Run Module (or hit F5).

You can also write python code to execute tools within ArcGIS: click the Python icon () by ArcToolbox. You will notice that as you type into the command line here Arc helps a little by giving you command options to choose from and reminding you of the appropriate syntax.

For more on scripting, see Dell's guide and Kudamatsu's course.

5 Other Tips

- You can check the status and details of your operations by clicking the moving name of the operation in the bottom right (i.e. the “...Clip...” part of this: ). This loads the “Results” tab on the left where you can see how long the current operation has taken, and how long past ones took, among other things.
- ArcGIS is slow, and has plenty of bugs. Try not to do things in ArcGIS unless you absolutely have to (I once spent days spatially joining Thai data to different administrative levels. The administrative ID was in the attribute table - I could have just done this in Stata in 20 minutes...)

References

- [1] Bleakley, H. and J. Lin (2012) “Portage and Path Dependence”, *Quarterly Journal of Economics*, 127, pp.587-644.
- [2] Henderson, J. V., A. Storeygard and D. Weil (2012) “Measuring Growth from Outer Space”, *American Economic Review*, 102(2), pp.994-1028.
- [3] Lowe, M. (2014) “The Privatization of African Rail”, Working Paper.
- [4] Michalopoulos, S. and E. Papaioannou (2013) “Pre-Colonial Ethnic Institutions and Contemporary African Development”, *Econometrica*, 81(1), pp.113-152.
- [5] Pinkovskiy, M. (2013) “Economic Discontinuities at Borders: Evidence from Satellite Data on Lights at Night”, Working Paper.
- [6] Storeygard, A. (2012) “Farther on down the road: transport costs, trade and urban growth in sub-Saharan Africa”, JMP.