Applied GIS in the Workplace Practical: An intro into GIS, programming and visualisation

The purpose of this practical is to use programming as a key tool for data analysis within GIS. Whilst not strictly part of your Applied GIS in the Workplace module, programming is increasingly being used within the GIS field to automate processing and conduct analysis on greater scales. Within this practical, we will be using the Python programming language as it is used within multiple GIS programs (ArcGIS and QGIS) and is increasingly the language of choice for those conducting analysis on large datasets. Another common language is R, which was first built primarily as a statistical analysis software but also increasingly used for mapping.

After this practical, you will achieve the following:

- Download and clean your own dataset, ready for processing and analysis.
- Use a jupyter notebook to run code and produce outputs.
- Learn the basics of how to edit python code.
- Understand the benefits of using scripting to automate processing and increase efficiency with GIS work.

You will need to think through the data presented to you, restructure files and use your own initiative to complete many aspects of the task. You will be provided with the code necessary to complete the processing as well as instructions to edit the code appropriately. You will have instructions but these will not be 'step-by-step', nor will you have screenshots, so pay attention to what you are doing.

Everything required is provided in the zipfile, ready for download on blackboard. ~~I would suggest you move the unzipped folder into your normal practical folder - and keep everything together within this folder.~~ Due to issues with the university network, you need to move the folder to your local C drive - and place it in the documents there. If you are unsure of how to access your local C drive (instead of the network filestore) please ask. However I recommend making a copy of the folder, once you finished the data cleaning outlined in these instructions, and placing it in your Applied GIS Practical folder for future use.

-It is essential that you do this for the practical to work. In addition, you need to read the instructions very carefully - unfortunately with coding, one error can create a bug in your script which will result in your code breaking - or worse, running but creating errors!

Mapping Obesity Change in England, 2011 - 2016

In Practical 2, you used Excel and ArcGIS to map the relationship between obesity in Year 6 students and deprivation in the England, creating tertiles for each of these variables and creating a bivariate map. Today, we're going to focus solely on mapping obesity - and in particular, how obesity rates have changed in Year 6 students over the last six years.

The work you're doing has critical relevance to several recently published studies. A Cancer Research UK article at the end of February highlighted that 'millenials' i.e. people born between the mid-eighties to mid-nineties resulting in ages of approximately 20 - 32, are likely to be the fattest generation. It is predicted that based on population trends, more than seven in every 10 millenials will be obese by the time they reach middle-aged (see: http://www.bbc.co.uk/news/health-43195977). The report also showed that obesity was the second cause of cancer, after smoking; an increase in obesity across the population is therefore likely to lead to an increase in cancer as well. There are also other health problems associated with obesity, such as heart issues, immobility and even mental health issues. Obesity is therefore a critical health issue for the country and needs to tackled through policy and prevention. To help with this, research is vital to ensure that policy designed and the funds that are provided are appropriate, including allocating the funds where the need is the most. As you saw in Practical 2, there is stark differences in the level of obesity across the country and as a result, any funds should target those areas with the highest levels. But we also need to account for the changing rates of obesity as well, to understand the general trend in these areas to ensure we are not just seeing a 'blip' in the data and that again the funds are used wisely.

We are therefore going to start mapping the changing obesity rates across the country and across a time-series to see if we can spot a similar pattern for those within 'Generation Z'. The aim is to determine whether obesity is increasing and where these rates are highest. The idea is that this analysis you are doing could actually be used in a future research paper that could, like the Cancer Research UK study, be published and promoted to help educate those designing Obesity policy[1].

We are however going to do this analysis all within a Jupyter Notebook using Python code - and not use ArcGIS. **Welcome to Geo-Data Science 101!** You'll be learning the basics of how to process data and produce several outputs that could be used in publications and promotion materials, such as new articles.

The two outputs from the analysis are: (1) a set of shapefiles that you can then use to map in ArcGIS (if you'd like to make some slightly more aesthetically pleasing maps!) or later export to CSVs and statistically analyse; and (2) a GIF of basic maps you'll produce using python. The latter could be used in a blog post or news article to provide a snazzy visualisation of your work!

Let's get started!

- If you haven't already, download the zip folder from blackboard, unzip it and move it to your Documents within the local C drive.

[1] One caveat! I have edited the data in one of the spreadsheets provided to make it easier for our analysis, so we couldn't release this actual analysis until I reversed those changes. If you'd like to know why, just come and ask - but it was too much to account for in this one practical!

- As you'll see we've got quite a few files to potentially look through and check. First, we're going to structure our folder to help everything make sense - and we need to do this for our code in the notebook to run without too much editing.

- We need to structure our folder as follows, where [F] = folder and file. (Don't include the [F] in your folder name!). You'll need to create all the folders yourself:

```
                              PRACTICAL_5 [F]
                       (i.e. the unzipped folder practical)
                 ____________________|____________________
                 |                            |              |
             Data [F]                     Maps [F]    Obesity_Mapping.ipynb
      ___________|________                                 (empty)
      |                  |
Obesity [F]        England_lad_2011.shp
   ____|___________
   |              |
Shapefiles [F]   RAW[F]
```

- In your RAW folder, store the original NHS Digital CSVs, which start with 'nati-chil-meas'..

- In your Obesity folder, store the cleaned CSVs I have provided, which are all titled 'YEAR_LA_OBESITY'.csv.

- Your Shapefile folder will be *empty.*

- And for ease of processing, we will move the LAD file into the Data folder, but leave it within this folder.

- Our python notebook, denoted as a .ipynb file and named 'Obesity_Mapping', can stay in the main folder.

- As you can see, none of our files or folders contain spaces in the name. This is super important when programming, as just one space within a 'path' can break an entire script!

- As stated in the blurb, we're looking to conduct the analysis from 2011 until 2016. This means six years - and as a result six CSVs.

- After structuring your folders and moving your data into the right place, you might have noticed that we are missing one of the CSVs required for analysis.

- You'll need to find and download this last dataset yourself. The data is from a programme - what programme is this and what organisation runs this programme? I've provided a hint in these instructions where you might find the necessary CSV - and just another hint, we're looking for 2016 - 2017.

- Finding data is a necessary skill for GIS, particularly in the workplace. You won't always be given the required data to analyse, so being able to find out about different sources of data is essential. Furthermore, you might end up with missing data, as in this case. It's not fun when you've got a lot of work to do!

- Go ahead and try and find our missing CSV. Let one of us know if you really do get stuck here.

- Once downloaded, you need to copy or save this data over into your RAW folder. We've then got quite a bit of data cleaning to do!

- The data is provided in a large excel workbook with multiple tables. It also uses 'Freeze rows' which means you can scroll through the data but the top rows won't move!

- As you can see, the data is also not structured in a suitable way for analysis - we need the data to be in its simplest format, with one set of headers/field names. If you look at the already processed CSVs, we can see what structure and field names we're looking for; the latter of which is extremely important as the code you'll be using is based on using the same field names across as CSVs.

- To get the data we want, we will open a new spreadsheet in Excel (i.e. a standalone document and not a new sheet within the raw dataset) and then copy over the data we need.

- Open a new spreadsheet in Excel - don't worry about adding any field names yet.

- Go back to the downloaded spreadsheet and navigate using the tabs on the bottom to our Table of Interest - 'Table 3b: Prevalence of underweight, healthy weight, overweight and obese children, by region and local authority (based on the postcode of the child)'.

- From this worksheet, we need to extract only THREE columns of data. Check through the already cleaned CSVs to see what columns we need.

  - You might see that the columns are named slightly differently, but check the contents to find what you need.

- For each column, I'd recommend to start copying from Row 13 to 380 I.e. <u>don't</u> include England or the row below it in your selection. Make sure you copy over just these rows for column, and you can leave in the formatting (e.g. bold names). You can use the formatting to check that when copying over the second and third column, you are matching the data correctly. An alternative approach is to save a copy of the worksheet and then delete what you don't need - this is likely to result in fewer mistakes, but due to all the formatting in the original spreadsheet, will be slower.

- Once you have your three columns ready in a single sheet, we need to remove the data we don't need. We are conducting the analysis at the LA level, which requires 326 LAs. (An easy way to confirm this would be to check the LAD shapefile in ArcGIS to see how many features are within the shapefile, but for now you can trust these instructions!.)

- As you can see, we have 367 rows of observations. This is because we have data grouped at a higher spatial level within our sheet e.g. the rows in bold, such as 'North East', which is a region not an LA.

- Whilst we could go through each row and delete the observations we don't want, this would take quite a long time - particularly if you had to do this for the other five years. Instead we can use simple logic within our data to remove the entries we don't want.

- If you closely at the LA Codes, we can see a pattern.
    - All regions have an EA code starting with E12.
    - All counties have an EA code starting with E10. A county groups multiple LAs, hence the original two-step formatting in the raw spreadsheet.
    - All LAs have an EA code starting with E06, E07, E08 or E09.

- If we sort our data by EA code, we can stack the counties and regions together for easy selection and deletion. To do this, select the EA code column, find the sort tool (in the data menu or on the toolbar) and sort - it doesn't matter whether you sort ascending or descending, BUT MAKE SURE TO EXPAND YOUR SELECTION in order to move the EA names with the code at the same time!

- You then need to scroll to the counties/regions stack and delete all the rows together. You may find that the 'North East' region remains on the first row (considered by Excel to be the field names) so don't forget to delete this!

- After deletion you should find that you have 324 rows or LAs left in the data. But wait, 324? We need 326! So why are we missing two extra LAs?

- To find out why, we need to go back to our original NHS Digital RAW data. Can you scroll through the table and see why we are missing two LAs? A hint: find the 'Notes!'.

- For the purpose of this tutorial, we're going to manually add in our missing two LAs and assign them with same data value as the LAs they've been merged with. This ensures that when mapping, the two LAs provide 'data' rather than 'no data', which in my opinion is a more truthful representation.

- We can then always provide a note similar to that found in the spreadsheet about the data with our maps and reports! This is one of the key aspects with GIS and GIS work - you'll often not get perfect data, so it's up to you to decide how best to use it in order to get the most truthful representation - and always make people aware of these adjustments by noting your decisions, particularly if your data and maps are to be used by other people.

- To add in the missing LAs, we can find exactly how to enter the LA Name and LA Code by looking them up using our LAD shapefile in ArcGIS or by using the already cleaned spreadsheets. For now, we'll use our one of our cleaned spreadsheets.

- Open up '2011_LA_Obesity.csv' and find the two missing LAs. You can then copy over the Name and Code into your spreadsheet (you may need to do this manually rather than using the copy and paste function). For the data value, copy over the value of the corresponding LA with which the missing LA was merged. Make sure to use COPY and PASTE rather than just type in the number yourself as you need all of the decimal places for the percentile function in our script to run.

- You should now have 326 LAs - great! Now create a new row at the top and add in the field names, if you haven't already. You can again copy and paste these over from the 2011 spreadsheet to ensure that we don't have any mistakes.

- Make sure you save your CSV now! It needs to be saved in the Obesity Folder, and follow the same name structure as the other files. Each CSV will be named in the same convention - for this tutorial, we are using the first year of the year range to denote the year of the CSV i.e. 2011-2012 becomes 2011.

- Just to reiterate - you need to ensure that all files are in the same location (Data/Obesity/) and named in the same way: 201X_LA_OBESITY . Make sure to save as a csv (you can use the drop down button on the Save menu) and not the default Excel worksheet.

- So we have our 326 LAs, structured under the same three field names as the other tables - and should be ready to start processing our data...right?!

- Alas no... and welcome to the true workflow of GIS. We still have some more data cleaning to do.

- The thing about data cleaning is that, often, you don't know that you need to do it, until something goes wrong during your processing and/or analysis. And then you have to figure out why something has gone wrong and then what you need to do about it. It's definitely a learning curve - and you only start to 'know' by using data, and using it more than once.

- Fortunately, we've used the LAD and Obesity data before so we can do a few checks before we start processing.

- One of the first things to check is name spelling. In your third practical, we found out that sometimes there is an error with County Durham and Durham. We should check that the name is consistent between our Obesity data and the shapefile; to do so, we'll again use the 2011 CSV which has been cleaned to match the LA shapefile data. Have a look at the cleaned 2011 CSV and your current CSV and make sure you've caught some of these errors, as you may end up with empty data rows if you don't - or even worse break your code.

  - *The other error that you might not have spotted is to do with the consistency of using a . after St.*

- We also saw when looking for our two missing LAs that the original workbook used superscript to denote footnotes - and these numbers copied over to our new spreadsheet. We therefore need to check our that our LA names are clear of any additional numbers/superscript. We particularly want to look at the two LAs that were merged with the missing LAs.

- Once we're confident that our data is looking 'clean', we can save our CSV again and close the sheet and Excel. It's time to move on and begin our analysis! At this point, it's now time to switch over to the Jupyter notebook.

- Click the Start button and navigate to: All Programs → Programming Languages → Anaconda3 → Jupyter Notebook. *(Make sure you are using the Anaconda 3 option!).*

- A black box should appear, and then if you wait a few seconds a browser window should load up with what looks like a file manager/directory in front of you.

- Click to navigate to your folder containing the Practical data. (i.e. Click Documents → AG_Practical).

- Click on the Obesity Mapping Python notebook (Obesity_Mapping.ipynb) - and we’re ready to start with our analysis.