GEOG0030: Geocomputation

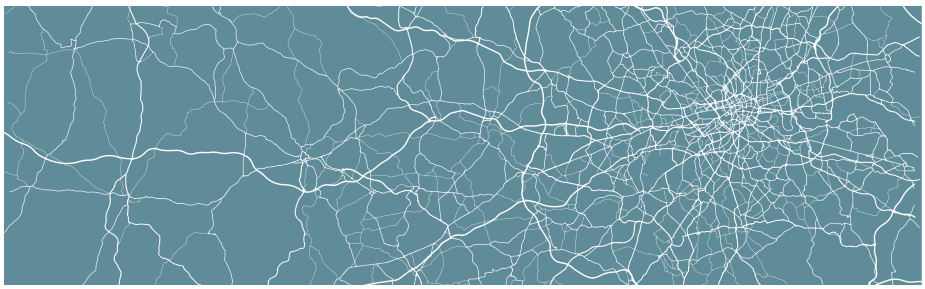
Justin van Dijk

Last modified: 2022-11-25

Contents

Module Overview

Module Introduction



Welcome

Welcome to **Geocomputation**. This module will introduce you both to the principles of spatial analysis as well as provide you with a comprehensive introduction to the use of programming. Over the next ten weeks, you will learn about the theory, methods and tools of spatial analysis through relevant case studies. We will start by using QGIS before moving to the R programming language. You will learn how to find, manage and clean spatial, demographic and socioeconomic datasets, and then analyse them using core spatial and statistical analysis techniques.

Moodle

Moodle is the central point of contact for GEOG0030 and it is where all important information will be communicated such as key module and assessment information. This workbook contains links to all reading material as well as the content of all computer tutorials

Module overview

The topics covered over the next ten weeks are:

Week	Section	Topic
1	Foundational Concepts	Geocomputation: An Introduction
2	Foundational Concepts	GIScience and GIS software
3	Foundational Concepts	Cartography and Visualisation
4	Foundational Concepts	Programming for Data Analysis
5	Foundational Concepts	Programming for Spatial Analysis
	Reading week	**Reading week**
6	Core Spatial Analysis	Analysing Spatial Patterns I: Geometric Operations and Spatial Queries
7	Core Spatial Analysis	Analysing Spatial Patterns II: Spatial Autocorrelation
8	Core Spatial Analysis	Analysing Spatial Patterns III: Point Pattern Analysis
9	Advanced Spatial Analysis	Rasters, Zonal Statistics and Interpolation
10	Advanced Spatial Analysis	Transport Network Analysis

Troubleshooting

Spatial analysis can yield fascinating insights into geographical relationships, albeit at times it can be challenging, particularly when we combine this with learning how to program at the same time. You will most likely encounter many error messages, experience software crashes, and spend hours to identify bugs in your code. However, the rewards of learning how to programmatically solve complex spatial problems will be very much worth it in the end.

If you need specific assistance with this course please:

- Ask a question at the end of a lecture or during the computer practical.
- Attend the Department's **Coding Therapy sessions** that are run on a weekly basis.
- Check the Moodle assessment tab for queries relating to this module's assessment.

If after pursuing all these avenues you still need help, you can book into our office hours. You can use an office hour to discuss a geographical concept in relation to the material, assessment or for any personal matters relevant to the completion of the module.

Acknowledgements

This year's workbook is updated and compiled using:

- The GEOG0030: Geocomputation 2021-2021 workbook as created and compiled by Dr Jo Wilkin.
- The GEOG0030: Geocomputation 2021-2022 workbook.

The datasets used in this workbook contain:

- Crime data obtained from data.police.uk (Open Government Licence)
- National Statistics data © Crown copyright and database right [2015] (Open Government Licence)
- Ordnance Survey data © Crown copyright and database right [2015]
- Public Health England © Crown copyright 2021

Foundational Concepts

Chapter 1

Geocomputation: An Introduction

This week's lecture provided you with a thorough introduction on Geocomputation, outlining how and why it is different to a traditional GIScience course. We set the scene for the remainder of the module and explained how the foundational concepts that you will learn in the first half of term sit within the overall module. This week we start easy by setting up our work environment and set up the software that we will need over the coming weeks.

1.1 Reading list

Essential readings

- Brundson, C. and Comber, A. 2020. Opening practice: Supporting reproducibility and critical spatial data science. *Journal of Geographical Systems* 23: 477–496. [Link]
- Longley, P. *et al.* 2015. Geographic Information Science & Systems, **Chapter 1**: *Geographic Information: Science, Systems, and Society.* [Link]
- Singleton, A. and Arribas-Bel, D. 2019. Geographic Data Science. *Geographical Analysis.* [Link]

Suggested readings

- Miller, H. and Goodchild, M. 2015. Data-driven geography. *GeoJournal* 80: 449–461. [Link]

- Goodchild, M. 2009. Geographic information systems and science: Today and tomorrow. *Annals of GIS* 15(1): 3-9. [Link]

1.2 Getting started

Over the next few weeks, we will be taking a closer look at many of the foundational concepts that will ultimately enable you to confidently and competently analyse spatial data using both programming and GIS software. You will further learn how to plan, structure and conduct your own spatial analysis using programming – whilst making decisions on how to best present your work, which is a crucial aspect of any type of investigation but of particular relevance to your dissertation.

To help with this, we highly recommend that you try to stay organised with your work, including taking notes and making yourself a coding handbook. We would also suggest to list the different datasets you come across - and importantly, the scales and different projections you use them at - more on this over the next weeks. Finally, you should also make notes about the different spatial analysis techniques you come across, including the different properties they assess and parameters they require to run.

1.3 Software

This course primarily uses the R programming language, although we start by using QGIS in the next two weeks to give you a basic foundation in the principles of spatial analysis.

Note Please follow the instructions below to install both R and QGIS onto your own personal computer. If you cannot install the software on your personal computer or you are not planning to bring your own laptop to the computer practicals, please refer to the UCL Desktop and RStudio Server section below. Please make sure that you have access to a working installation of QGIS and R (including relevant packages) **before** the first hands-on practical session next week.

1.3.1 QGIS Installation

QGIS is an open-source graphic user interface GIS with many community developed add-on packages (or plugins) that provide additional functionality to the software. You can download and install QGIS on your personal machine by going to the QGIS website: [Link].

Note We recommend installing the **Long Term Release** (*QGIS 3.22 LTR*) as this version should be the most stable version. For Windows users: the QGIS installation may be a little slow.

After installation, start QGIS to see if the installation was successful and no errors are shown after start up.

1.3.2 R and RStudio Installation

R is both a programming language and software environment - in the form of RStudio- originally designed for statistical computing and graphics. R's great strength is that it is open-source, can be used on any computer operating system, and is free for anyone to use and contribute to. Because of this, it is rapidly becoming the statistical language of choice for many academics and has a very large user community with people constantly contributing new packages to carry out all manner of statistical, graphical, and importantly for us, geographical tasks.

Installing R takes a few relatively simple steps involving two programmes. First there is the R programme itself. Follow these steps to get it installed on your computer:

1. Navigate in your browser to your nearest CRAN mirror: [Link]
2. If you use a Windows computer, click on *Download R for Windows*. Then click on *base*. Download and install **R 4.2.x for Windows**. If you use a Mac computer, click on *Download R for macOS* and download and install **R-4.2.x.pkg**

That is it! You now have installed the latest version of R on your own machine. However, to make working with R a little bit easier we also need to install something called an Integrated Development Environment (IDE). We will use RStudio:

1. Navigate to the official webpage of RStudio: [Link]
2. Download and install RStudio Desktop on your computer (**free version!**)

After this, start RStudio to see if the installation was successful and no errors are shown after start up.

1.3.3 UCL Desktop and RStudio Server

As an alternative to installing QGIS and R with RStudio onto your personal device, there are some other options. Firstly, both programmes are available through Desktop@UCL Anywhere as well as all UCL computers on campus. In

case of R, there is also an RStudio server version available which you can access through your web browser: [Link]

You should be able to log in with your normal UCL username and password. After logging in, you should see the RStudio interface appear.

Note If it is the first time you log on to RStudio server you may only see the RStudio interface appear once you have clicked on the *start a new session* button. More importantly: if you are not on campus, RStudio server will only work with an active Virtual Private Network (VPN) connection that links your personal computer into UCL's network. Details on setting up a VPN connection can be found in UCL's VPN connection guides: [Link]

1.3.4 R package installation

Now we have installed or have access to QGIS and R, we need to customise R. Many useful R function come in packages, these are free libraries of code written and made available by other by R users. This includes packages specifically developed for data cleaning, data wrangling, visualisation, mapping, and spatial analysis. To save us some time, we will install all R packages that we will need over the next ten weeks in one go. Without going into detail on the RStudio (Server) interface, copy and paste the following code into the *console*. You can execute the code by hitting **Enter**. This may take a while.

```
# install all packages that we need
install.packages(c("tidyverse", "sf", "tmap", "osmdata", "RColorBrewer", "janitor",
    "spdep", "dbscan", "raster", "spatstat", "gstat", "dodgr"))
```

Once you have installed the packages, we need to check whether we can in fact load them into our R session. Copy and paste the following code into the **console**, and executed by hitting **Enter** again.

```
# load all packages
library(tidyverse)
library(sf)
library(tmap)
library(osmdata)
library(RColorBrewer)
library(janitor)
library(spdep)
library(dbscan)
library(raster)
library(spatstat)
library(gstat)
library(dodgr)
```

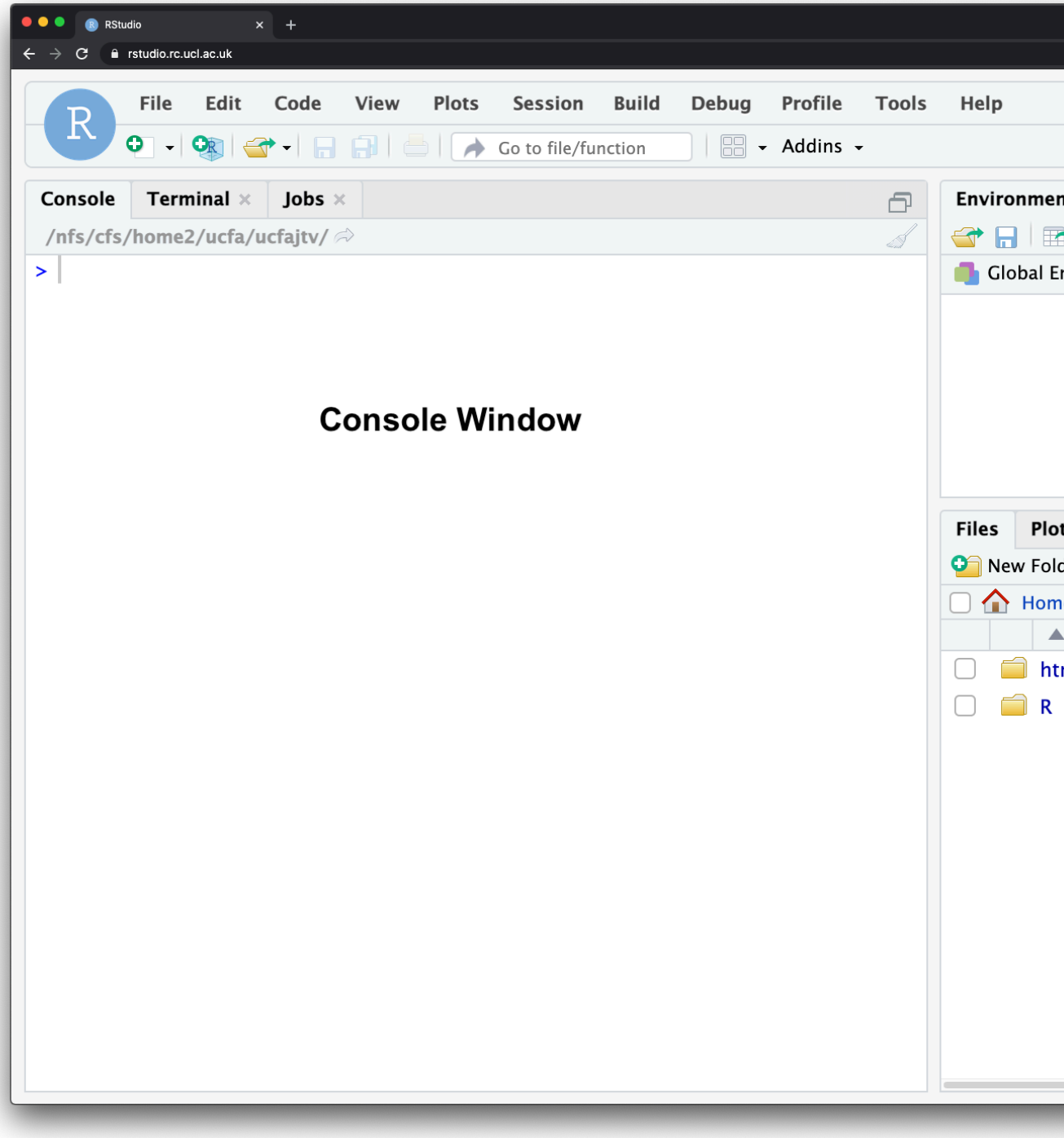


Figure 1.1: The RStudio Server interface.

You will see some information printed to your console but as long as you do not get a message that is similar to `Error: package or namespace load failed for <packagename>` or `Error: package '<packagename' could not be loaded` all should be fine.

Note Even if you have used R or RStudio Server before and already installed some of the packages in the above list, do re-install all packages to make sure you have the latest versions. Legacy installations that have not been updated may lay lead to problems when going through the tutorials.

1.3.5 A note on ArcGIS

ArcGIS Pro (previously ArcMap) is the main commercial GIS software that you may have already used - or seen/heard about through other modules or even job adverts. We do not use ArcGIS Pro in our Practicals for several reasons:

- Computing requirements for ArcGIS Pro are substantial and it **only** operates on the Windows Operating System. For Mac users, using ArcGIS Pro (and ArcMap) would require using either a Virtual Machine or running a copy of Windows OS on a separate partition of your hard drive.
- It is **proprietary** software, which means you need a license to use the software. For those of us in education, the University covers the cost of this license, but when you leave, you will need to pay for a personal license (around £100 for non-commercial use) to continue using the software and repeat any analysis you have used the software for.
- Whilst ArcPro can use pure Python (and even R) as a programming language within it through scripts and notebooks, it primarily relies on its own **ArcPy** and **ArcGIS API for Python** packages to run the in-built tools and analytical functions. To use these packages, you still need a license which makes it difficult to share your code with others *if* they do not have their own ArcGIS license.

Recent developments in the ArcPro software, however, does make it an attractive tool for spatial data science and quantitative geography - it has cross-user functionality, from data analysts who like to use a tool called Notebooks for their code development, to those focused more on cartography and visualisation with in-built bridges to Adobe's Creative Suite. We therefore do not want to put you off looking into ArcGIS in the future, but for this course, we want to ensure the reproducibility of your work.

Note This also means that the analysis you will be doing for your coursework assignment must be completed in R and QGIS. Specific guidance on the coursework assignment and permitted software will be made available at the end of Reading Week.

1.4 Before you leave

You should now be all ready to go with the computer practicals the coming week. That is it for this week!

Chapter 2

GIScience and GIS software

This week's lecture introduced you to foundational concepts associated with GIScience and GIS software, with particular emphasis on the representation of spatial data and sample design. Out of all our foundational concepts you will come across in the next four weeks, this is probably the most substantial to get to grips with and has both significant theoretical and practical aspects to its learning. The practical component of the week puts some of these learnings into practice, starting with a short digitisation excercise followed by a simple visualisation of London's population over time.

2.1 Reading list

Essential readings

- Longley, P. *et al.* 2015. Geographic Information Science & Systems, **Chapter 2**: *The Nature of Geographic Data*. [Link]
- Longley, P. *et al.* 2015. Geographic Information Science & Systems, **Chapter 3**: *Representing Geography*. [Link]
- Longley, P. *et al.* 2015. Geographic Information Science & Systems, **Chapter 7**: *Geographic Data Modeling*. [Link]

Suggested readings

- Goodchild, M. and Haining, R. 2005. GIS and spatial data analysis: Converging perspectives. *Papers in Regional Science* 83(1): 363–385. [Link]
- Schurr, C., Müller, M. and Imhof, N. 2020. Who makes geographical knowledge? The gender of Geography's gatekeepers. *The Professional Geographer* 72(3): 317-331. [Link]

- Yuan, M. 2001. Representing complex geographic phenomena in GIS. *Cartography and Geographic Information Science* 28(2): 83-96. [Link]

2.2 Simple digitisation of spatial features

To get spatial features in a digital form, they need to be digitised. Let's take what should be a straight-forward example of digitising the River Thames in London.

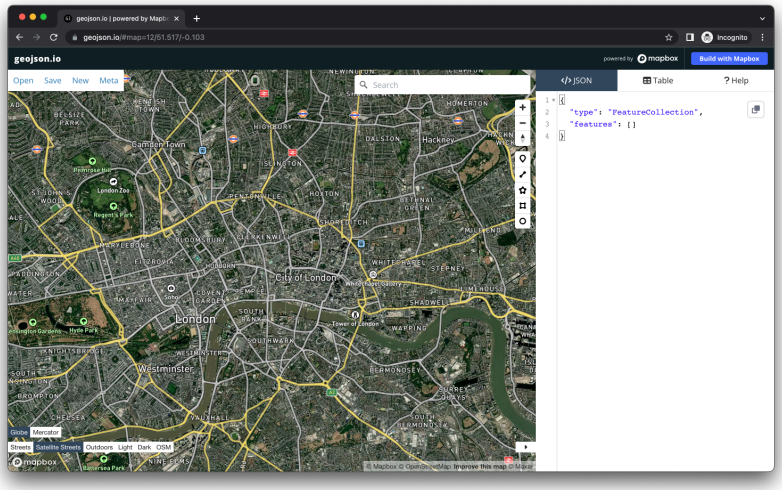


Figure 2.1: The Thames.

We are going to use a very simple online tool that allows us to create digital data and export the data we create as raw files.

1. Head to geojson.io.
2. In the bottom left-hand corner, select *Satellite Streets* as your map option.
3. Next, click on the `Draw Linestring` tool which you can find on the right hand side of the screen. You can hover over the icons to get the names of each tool.
4. Now digitise the river Thames. Simply click from a starting point on the left- or right-hand side of the map, and digitise the river.
5. Once you are done, double-click your final point to end your line.
6. You can click on the line and select *Info* in the pop-up screen to find out how long the line is.
7. You can export your data using the *Save* menu.

Questions

- How easy did you find it to digitise the data and what decisions did you make in your own "sample scheme"?
- How close together are your clicks between lines?
- Did you sacrifice detail over expediency or did you spend perhaps a little too long trying to capture ever small bend in the river?
- How well do you think your line represents the River Thames?

2.3 Population change in London

The second part of this practical will introduces you to **attribute joins** followed by creating a choropleth map. You will be using different types of *joins* throughout this module, and probably the rest of your career, so it is incredibly important that you understand how they work.

Note The datasets you will create in this practical will be used in next week's practical, so make sure to follow every step and save your data carefully.

When using spatial data, there is generally a very specific workflow that you will need to go through and, believe it or not, the majority of this is not actually focused on analysing your data. Along with the idea that 80% of data is geographic data, the second most often-quoted GIS-related unreferenced 'fact' is that anyone working with spatial data will spend 80% of their time simply finding, retrieving, managing and processing the data before any analysis can be done.

One of the reasons behind this need for a substantial amount of processing is that the data you often need to use is almost never in the format that you require for analysis. For example, for our investigation, there is not a 'ready-made' spatial population dataset (i.e. population `shapefile`) we can download to explore population change across England:

Instead, we need to go and find the raw datasets and create the data layers that we want. As a result, before beginning any spatial analysis project, it is best-practice to think through what end product you will ultimately need for your analysis.

A typical spatial analysis workflow usually looks something like this:

- **Identify** the data you need to complete your analysis i.e. answer your research questions. This includes thinking through the scale, coverage and currency of your dataset.
- **Find** the data that matches your requirements, e.g. is it openly and easily available?
- **Download** the data and store it in the correct location.

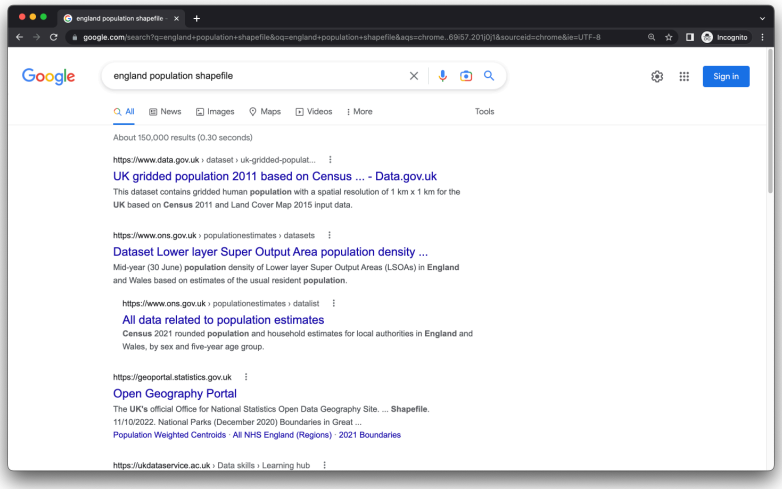


Figure 2.2: Alas a quick Google search shows that finding a shapefile of England's population is not straightforward.

- **Clean** the data. This may be done before or after ingesting your data into your chosen software programme.
- **Load** the data into your chosen software programme.
- **Transform and process** the data. This may require re-projection, creating joins between datasets, calculating new fields and applying selections.
- **Analyse** your data using appropriate methods.
- **Visualise** your data and results with graphs and maps.
- **Communicate** your results.

As you can see, the analysis and visualisation part comes quite late in the overall spatial analysis workflow - and instead, the workflow is very top-heavy with data management. However, very often in GIS-related courses you will be given pre-processed datasets. Because data management is an essential part of your workflow, we are clean (the majority of) our data from the get-go. This will help you understand the processes that you will need to go through in the future as you search for and download your own data, as well as deal with the data first-hand before loading it into our GIS software.

2.3.1 Setting the scene

For this practical, we will investigate how the population in London has changed over time. Understanding population change - over time and space - is spatial analysis at its most fundamental. We can understand a lot just from where population is growing or decreasing, including thinking through the impacts

of these changes on the provision of housing, education, health and transport infrastructure.

We can also see first-hand the impact of wider socio-economic processes, such as urbanisation. Today we will look at population in London in 2011, 2015, and 2019 at the *ward* scale that we can use within our future analysis projects, starting next week.

Note We will use the population dataset to *normalise* other datasets. Why? When we record events created by humans, there is often a population bias: simply, more people in an area will by probability lead to a higher occurrence of said event, such as crime. We will look at this in greater detail next week.

2.3.2 Finding data

In the UK, finding authoritative data on population and *Administrative Geography* boundaries is increasingly straight-forward. Over the last decade, the UK government has opened up many of its datasets as part of an **Open Data** precedent that began in 2010 with the creation of data.gov.uk and the Open Government Licence (the terms and conditions for using data).

Data.gov.uk is the UK government's central database that contains open data that the central government, local authorities and public bodies publish. This includes, for example, aggregated census and health data – and even government spending. In addition to this central database, there are other authoritative databases run by the government and/or respective public bodies that contain either a specific type of data (e.g. census data, crime data) or a specific collection of datasets (e.g. health data from the NHS, data about London). Some portals are less up-to-date than others, so it is wise to double-check with the 'originators' of the data to see if there are more recent versions.

For our practical, we will access data from two portals:

1. For our administrative boundaries, we will download the **spatial** data from the *London Datastore* (which is exactly what it sounds like).
2. For population, we will download **attribute** data from the *Office of National Statistics (ONS)*.

2.3.3 Housekeeping

Before we download our data, it is important to establish an organised file systems that we will use throughout the module:

1. Create a `GEOG0030` folder in your `Documents` folder on your computer.
2. Within your `GEOG0030` folder, create the following subfolders:

Folder name	Purpose
`data`	To store both raw data sets and final outputs.
`maps`	To save the maps you produce during your tutorials.

3. Within your `data` folder, create the following subfolders:

Folder name	Purpose
`raw`	To store all your raw data files that have not yet been processed.
`output`	To store all your final data files that have been processed and analysed, potentially ready to be mapped.

2.3.4 Downloading data

We will start by downloading the administrative geography boundaries:

1. Navigate to the relevant page on the London Datastore: [Link].
2. Download all three zipfiles to your computer: `statistical-gis-boundaries-london.zip`, `London-wards-2014.zip` and `London-wards-2018.zip`.

The first dataset contains all levels of London's administrative boundaries. In descending size order: borough, Ward, Middle layer Super Output Area (MSOA), Lower layer Super Output Area (LSOA), and Output Area (OA) based on the 2011 Census. The second dataset contains an *updated* version of the Ward boundaries, as of 2014. The third dataset contains yet another *updated* version of the Ward boundaries, as of 2018. As we will be looking at population data for 2015 and 2019, it is best practice to use those boundaries that are most reflective of the 'geography' at the time; therefore, we will use these 2014 / 2018 ward boundaries for our 2015 / 2019 population dataset, respectively.

Note Once downloaded, you will need to unzip all files before you can use them. To unzip the file, you can use the built-in functionality of your computer's operating system. For Windows: right click on the zip file, select **Extract All**, and then follow the instructions. For Mac OS: double-click on the the zip file and it should unzip automatically.

Once unzipped, you will find two folders: *Esri* and *MapInfo.* These folders contain the same data but in different data formats: **Esri shapefile** and **MapInfo TAB**.

Note MapInfo is another proprietary GIS software, which has historically been used in public sectors services in the UK and abroad, although has generally been replaced by either Esri's ecosystem or open-source software GIS.

Now open your `GEOG0030/data/raw/` folder and create a new folder called `boundaries`. Within this folder, create three new folders: `2011`, `2014` and `2018`. Copy the entire contents of `Esri` folder of each year into their respective year folder.

We do not want to add the additional `Esri` folder as a step in our filesystem, i.e. your file paths should read: `GEOG0030/data/raw/boundaries/2011` for the 2011 boundaries, `GEOG0030/data/raw/boundaries/2014` for the 2014 boundaries, and `GEOG0030/data/raw/boundaries/2018` for the 2018 boundaries.

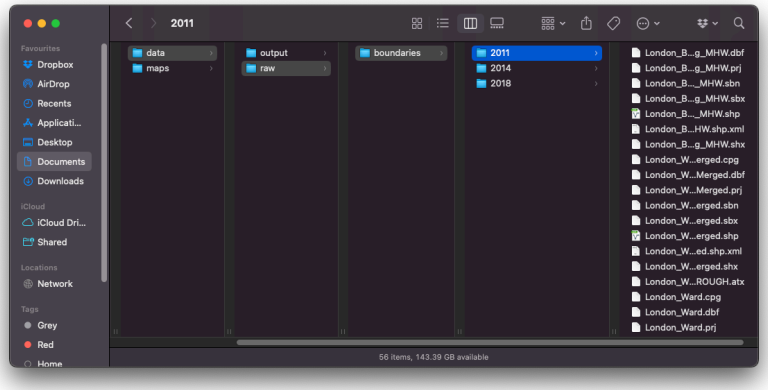


Figure 2.3: Your setup should look something like this.

We now have our administrative geography files ready for use.

Note Administrative geographies are a way of dividing the country into smaller sub-divisions or areas that correspond with the area of responsibility of local authorities and government bodies. These administrative sub-divisions and their associated geography have several important uses, including assigning electoral constituencies, defining jurisdiction of courts, planning public healthcare provision, as well as what we are concerned with: used as a mechanism for collecting census data and assigning the resulting datasets to a specific administrative unit. These geographies are updated as populations evolve and as a result, the boundaries of the administrative geographies are subject to either periodic or occasional change. The UK has quite a complex administrative geography, particularly due to having several countries within one overriding administration and then multiple ways of dividing the countries according to specific applications. More details on the administrative geographies of the UK can be found on the website of the Office for National Statistics.

For our population datasets, we will use the ONS mid-year estimates (MYE). These population datasets are estimates that are based on the 2011 census count and then updated with estimated population growth. They are released once

a year, with a delay of a year. Today we will use the data for 2011, 2015, and 2019.

1. Navigate to the *ward* level datasets: [Link]
2. When you navigate to this page, you will find multiple choices of data to download. We will need to download the estimates for **2011, 2015** and **2019**. Click to download each of the zipfiles. Choose the **revised** versions for 2015 and the (Census-based) 2011 wards edition for 2011.
3. In your `GEOG0030/data/raw/` folder, create a new folder called `population`, unzip your downloaded files, and copy the three spreadsheets to the newly created `population` folder.
4. Rename the files you donwloaded to: `MYE_ward_2011.xls`, `MYE_ward_2015.xls`, and `MYE_ward_2019.xlsx`.

Now it is time to do some quite extensive data cleaning and preparation.

2.3.5 Cleaning data

When you open up any of the ward spreadsheets in Excel =, you will notice that there are several worksheets contained in this workbook. However, we are only interested in the total population tab. We therefore need to copy over the data from the 2011, 2015 and 2019 datasets into separate `csv` files.

2.3.5.1 London population in 2011

1. Open the 2011 ward spreadsheet in Excel.
2. Click on the `Mid-2011 Persons` tab and have a look at the data. As you should be able to see, we have a set of different fields (e.g. `Ward Code`, `Ward Name`), including population counts. Because we do not need all the data in the spreadsheet, we will extract only the data we need for our analysis. This means we need the total population (`All Ages`) data, alongside some identifying information that distinguishes each record from one another. Here we can see that both `Ward Code` and `Ward Name` suit this requirement. We can also think that the `Local Authority` column might be of use, so we also keep this information.
3. Create a new Excel spreadsheet Excel and from the `Mid-2011 Persons` spreadsheet, copy over all cells from columns **A** to **D** and rows **4 to 636** into this new spreadsheet. Row 636 denotes the end of the *Greater London* wards (i.e. the end of the *Westminster Local Authority*) which are kept (in most scenarios) at the top of the spreadsheet as their **Ward Codes** are the first in sequential order.
4. Before we go any further, we need to format our data. First, we want to rename our fields to remove the spaces and superscript formatting. Re-title the fields as follows: `ward_code`, `ward_name`, `local_authority` and `pop2011`.

5. One further bit of formatting that you must do before saving your data is to format our population field. At the moment, you will see that there are commas separating the thousands within our values. If we leave these commas in our values, QGIS will read them as decimal points, creating decimal values of our population. There are many points at which we could solve this issue, but the easiest point is now - we will strip our population values of the commas and set them to integer (whole numbers) values. To format the `pop2011` column, select the entire column and right-click on the `D` cell. Click on **Format Cells** and set the Cells to **Number** with **0** decimal places. You should see that the commas are now removed from your population values.
6. Save your spreadsheet into your `output` folder as `ward_population_2011.csv`.

2.3.5.2 London population in 2015

1. Open the 2015 ward spreadsheet in Excel.
2. As you will see again, there are plenty of worksheets available and we want to select the `Mid-2015 Persons` tab. We now need to copy over the data from our 2015 dataset to a new spreadsheet again. However, at first instance, you will notice that the City of London (CoL) wards are missing from this dataset. Then if you scroll to the end of the London Local Authorities, i.e. to the bottom of Westminster, what you should notice is that the final row for the Westminster data is in fact row 575 - this suggests we are missing the data fror some Local Authorities (LAs). We need to determine which ones are missing and try to find them in the 2015 spreadsheet. With this in mind, start by copying over all cells from columns **A** to **D** and rows **5 to 575** into a new spreadsheet.
3. If you were to compare the names of the London Boroughs that we have now copied with the full list, you would notice that we are missing *City of London*, *Hackney*, *Kensington and Chelsea*, and *Tower Hamlets*. If we head back to the original 2015 raw dataset, we can actually find this data (as well as the City of London) further down in the spreadsheet. It seems like these LAs had their codes revised in the 2014 revision and are no longer in the same order as the 2011 dataset.
4. Locate the data for the *City of London*, *Hackney*, *Kensington and Chelsea* and *Tower Hamlets* and copy this over into our new spreadsheet. Double-check that you now have in total **637** wards within your dataset.
5. Remember to rename the fields as above, but change your population field to **pop2015**. Also, remember to reformat the values in your `pop2015` column.
6. Once complete, save your spreadsheet into your `output` folder as `ward_population_2015.csv`.

2.3.5.3 London population in 2019

1. Open the 2019 ward spreadsheet in Excel. This time we are interested in the `Mid-2019 Persons` tab.
2. This time the data that we are interested in can be found in columns `A`, `B`, `D` and `G`. Because the columns that we want are not positioned next to one antoher, start by hiding columns `C`, `E` and `F`. You can do this by right-clicking on the colums you want to hide and selecting **Hide**.
3. Next, copy the data from **row 5 to the final row for the Westminster data** for columns `A`, `B`, `D` and `G` over into a new spreadsheet.
4. If you look at the total rows that we have copied over, we have even fewer wards than the 2015 dataset. This time we are not only missing data for *City of London*, *Hackney*, *Kensington and Chelsea*, *Tower Hamlets* but also for *Bexley*, *Croydon*, *Redbridge*, and *Southwark*.
5. Copy over the remaining wards for these Local Authorities/boroughs.
6. Once you've copied them over - you should now have **640** wards. Delete columns `C`, `E` and `F` and rename the remaining fields as you have done previously. Also, remember to reformat the values in your `pop2019` column.
7. Once complete, save your spreadsheet into your `output` folder as `ward_population_2019.csv`.

You should now have your three population `csv` datasets in your `output` folder. We are now (finally) ready to start using our data within QGIS.

2.3.6 Using QGIS to map our population data

2.3.6.1 Setting up a project

We will now use QGIS to create population maps for the wards in London across our three time periods. To achieve this, we need to **join our table data to our spatial datasets** and then map our populations for our visual analysis.

Because, as we have seen above, we have issues with the number of wards and changes in boundaries across our three years, we will not (for now) complete any quantitative analysis of these population changes - this would require significant additional processing that we do not have time for today.

Note Data interoperability is a key issue that you will face in spatial analysis, particularly when it comes to Administrative Geographies.

1. Start **QGIS**. Let's start a new project.
2. Click on **Project** -> **New**. Save your project into your `qgis` folder as `w2-pop-analysis`. Remember to save your work throughout the practical.
3. Before we get started with adding data, we will first set the Coordinate Reference System of our Project. Click on **Project** -> **Properties** –

CRS. In the Filter box, type **British National Grid**. Select **OSGB 1936 / British National Grid - EPSG:27700** and click Apply. Click **OK**.

Note We will explain CRSs and using CRSs in GIS software in more detail next week.

2.3.6.2 Adding layers

We will first focus on loading and joining the 2011 datasets.

5. Click on **Layer** -> **Add Layer** -> **Add Vector Layer**.
6. With **File** select as your source type, click on the small three dots button and navigate to your 2011 boundary files.
7. Here, we will select the `London_Ward.shp` dataset. Click on the `.shp` file of this dataset and click **Open**. Then click **Add**. You may need to close the box after adding the layer.

We can take a moment just to look at our ward data - and recognise the shape of London. Can you see the City of London in the dataset? It has the smallest wards in the entire London area. With the dataset loaded, we can now explore it in a little more detail. We want to check out two things about our data: first, its **Properties** and secondly, its **Attribute Table**.

8. Right-click on the `London_Ward` layer and open the **Attribute Table** and look at how the attributes are stored and presented in the table. Explore the different buttons in the Attribute Table and see if you can figure out what they mean. Once done, close the Attribute Table.
9. Right-click on the `London_Ward` layer and select **Properties**. Click through the different tabs and see what they contain. Keep the **Properties** box open.

Before adding our population data, we can make a quick map of the wards in London - we can add labels and change the *symbolisation* of our wards.

10. In the **Properties** box, click on the **Symbology** tab - this is where we can change how our data layer looks. For example, here we can change the line and fill colour of our wards utilising either the default options available or clicking on **Simple Fill** and changing these properties directly. Keep the overall **styling** to a **Single Symbol** for now - we will get back to this once we have added the population data. You can also click on the **Labels** tab - and set the Labels option to **Single labels**.

11. QGIS will default to the **NAME** column within our data. You can change the properties of these labels using the options available. Change the font to **Futura** and size **8** and under the add a small buffer to the labels by selecting **Draw text bufer** under the **Buffer** tab. You can click **Apply** to see what your labels look like. Please note that the background colour may differ.

12. Click **OK** once you are done changing the Symbology and Label style of your data to return to the main window.

Note The main strength of a GUI GIS system is that is really helps us understand how we can visualise spatial data. Even with just these two shapefiles loaded, we can understand two key concepts of using spatial data within GIS.

The first, and this is only really relevant to GUI GIS systems, is that each layer can either be turned on or off, to make it visible or not (try clicking the tick box to the left of each layer). This is probably a feature you are used to working with if you have played with interactive web mapping applications before!

The second concept is the order in which your layers are drawn – and this is relevant for both GUI GIS and when using plotting libraries such as `ggplot2` or `tmap` in RStudio. Your layers will be drawn depending on the order in which your layers are either tabled (as in a GUI GIS) or 'called' in your function in code.

Being aware of this need for 'order' is important when we shift to using RStudio and `tmap` to plot our maps, as if you do not layer your data correctly in your code, your map will end up not looking as you hoped!

For us using QGIS right now, the layers will be drawn from bottom to top. At the moment, we only have one layer loaded, so we do not need to worry about our order right now - but as we add in our 2015 and 2018 ward files, it is useful to know about this order as we will need to display them individually to export them at the end.

2.3.6.3 Conducting an attribute join

We are now going to join our 2011 population data to our 2011 shapefile. First, we need to add the 2011 population data to our project.

13. Click on **Layer** -> **Add Layer** -> **Add Delimited Text Layer**.

14. Click on the three dots button again and navigate to your **2011 population data** in your `working` folder. Your file format should be set to `csv`. You should have the following boxes clicked under the **Record and Field options** menu: *Decimal separator is comma; First record has field*

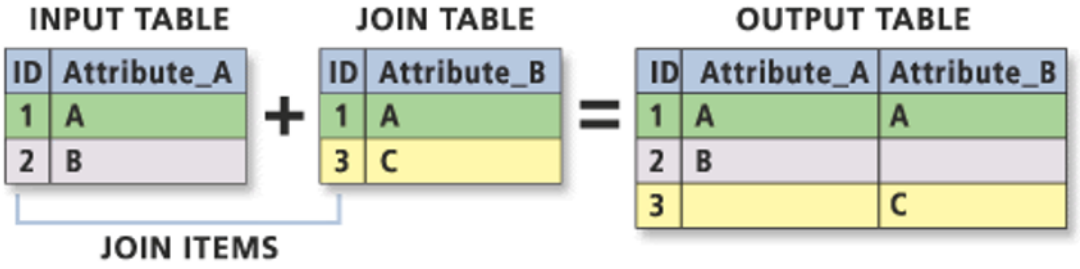
Figure 2.4: It looks incredibly busy.

names; Detect field types; Discard empty fields. QGIS does many of these by default, but do double-check!

15. Set the Geometry to *No geometry (attribute only table)* under the **Geometry Definition** menu. Then click **Add** and **Close**. You should now see a table added to your `Layers` box.

We can now join this table data to our spatial data using an **Attribute Join**.

Note An attribute join is one of two types of data joins you will use in spatial analysis (the other is a spatial join, which we will look at later on in the module). An attribute join essentially allows you to join two datasets together, as long as they share a common attribute to facilitate the 'matching' of rows:



Essentially you need a **single identifying ID** field for your records within both datasets: this can be a code, a name or any other string of information. In spatial analysis, we always **join our table data to our shape data** (One way to think about it as attaching the table data to each shape).

As a result, your target layer is always the shapefile (or spatial data) whereas your join layer is the table data. These are known as the left- and right-side tables when working with code.

To make a join work, you need to make sure your ID field is correct across both datasets, i.e. no typos or spelling mistakes. Computers can only follow instructions, so they do not know that *St. Thomas* in one dataset is that same as *St Thomas* in another, or even *Saint Thomas*! It will be looking for an exact match!

As a result, whilst in our datasets we have kept both the name and code for both the boundary data and the population data, **when creating the join, we will always prefer to use the CODE over their names**. Unlike names, codes reduce the likelihood of error and mismatch because they do not rely on understanding spelling!

Common errors, such as adding in spaces or using `0` instead `O` (and vice versa) can still happen – but it is less likely.

To make our join work, we need to check that we have a matching **UID** across both our datasets. We therefore need to look at the tables of both datasets and check what attributes we have that could be used for this possible match.

16. Open up the Attribute Tables of each layer and check what fields we have that could be used for the join. We can see that both our respective "code" fields have the same codes (`ward_code` and `GSS_code`) so we can use these to create our joins.

17. Right-click on your `London_Ward` layer -> **Properties** and then click on the **Joins** tab.

- Click on the + button. Make sure the **Join Layer** is set to `ward_population_2011`.
- Set the **Join field** to `ward_code`.
- Set the **Target field** to `GSS_code`.
- Click the **Joined Fields** box and click to only select the `pop2011` field.
- Click on the **Custom Field Name Prefix** and **remove** the pre-entered text to leave it blank.
- Click on **OK**.
- Click on **Apply** in the main Join tab and then click **OK** to return to the main QGIS window.

We can now check to see if our join has worked by opening up our `London_Ward` **Attribute Table** and looking to see if our wards now have a **Population** field attached to it.

18. Right-click on the `London_Ward` layer and open the **Attribute Table** and check that the population data column has been added to the table.

As long as it has joined, you can move forward with the next steps. If your join has not worked, try the steps again - and if you are still struggling, do let us know.

Note Now, the join that you have created between your ward and population datasets in only held in QGIS's memory. If you were to close the programme now, you would lose this join and have to repeat it the next time you opened QGIS. To prevent this from happening, we need to export our dataset to a new shapefile - and then re-add this to the map.

Let's do this now:

19. Right-click on your `London_Ward` shapefile and click **Export** -> **Save Features As...**. The format should be set to an ESRI shapefile.

- Then click on the three dots buttons and navigate to your `final` folder and enter: `ward_population_2011` as your file name.
- Check that the **CRS** is **British National Grid**.
- Leave the remaining fields as selected, but check that the **Add saved file to map** is checked. Click **OK**.

You should now see our new shapefile add itself to our map. You can now remove the original `London_Ward` and `ward_population_2011` datasets from our Layers box (Right-click on the layers and opt for **Remove Layer...**).

The final thing we would like to do with this dataset is to style our dataset by our newly added population field to show population distribution around London.

20. To do this, again right-click on the **Layer** -> **Properties** -> **Symbology**.

- This time, we want to style our data using a **Graduated** symbology.
- Change this option in the tab and then choose `pop2011` as your column.
- We can then change the color ramp to suit our aesthetic preferences - *Viridis* seems to be the cool colour scheme at the moment, and we will choose to invert our ramp as well.
- The final thing we need to do is **classify** our data - what this simply means is to decide how to group the values in our dataset together to create the graduated representation.
- We will be looking at this in later weeks, but for now, we will use the **Natural Breaks** option.
- Click on the drop-down next to Mode, select **Natural Breaks**, change it to **7** classes and then click **Classify**.
- Finally click **Apply** to style your dataset.

Note Understanding what classification is appropriate to visualise your data is an important step within spatial analysis and visualisation, and something you will learn more about in the following weeks. Overall, they should be determined by understanding your data's distribution and match your visualisation accordingly.

Feel free to explore using the different options with your dataset at the moment – the results are almost instantaneous using QGIS, which makes it a good playground to see how certain parameters or settings can change your output.

You should now be looking at something like this:

You will be able to see that we have **some** missing data - and this is for several wards within the City of London. This is because census data is **only recorded for 8 out of the 25 wards** and therefore we have no data for the remaining wards. As a result, these wards are left blank, i.e. white, to represent a `NODATA` value.

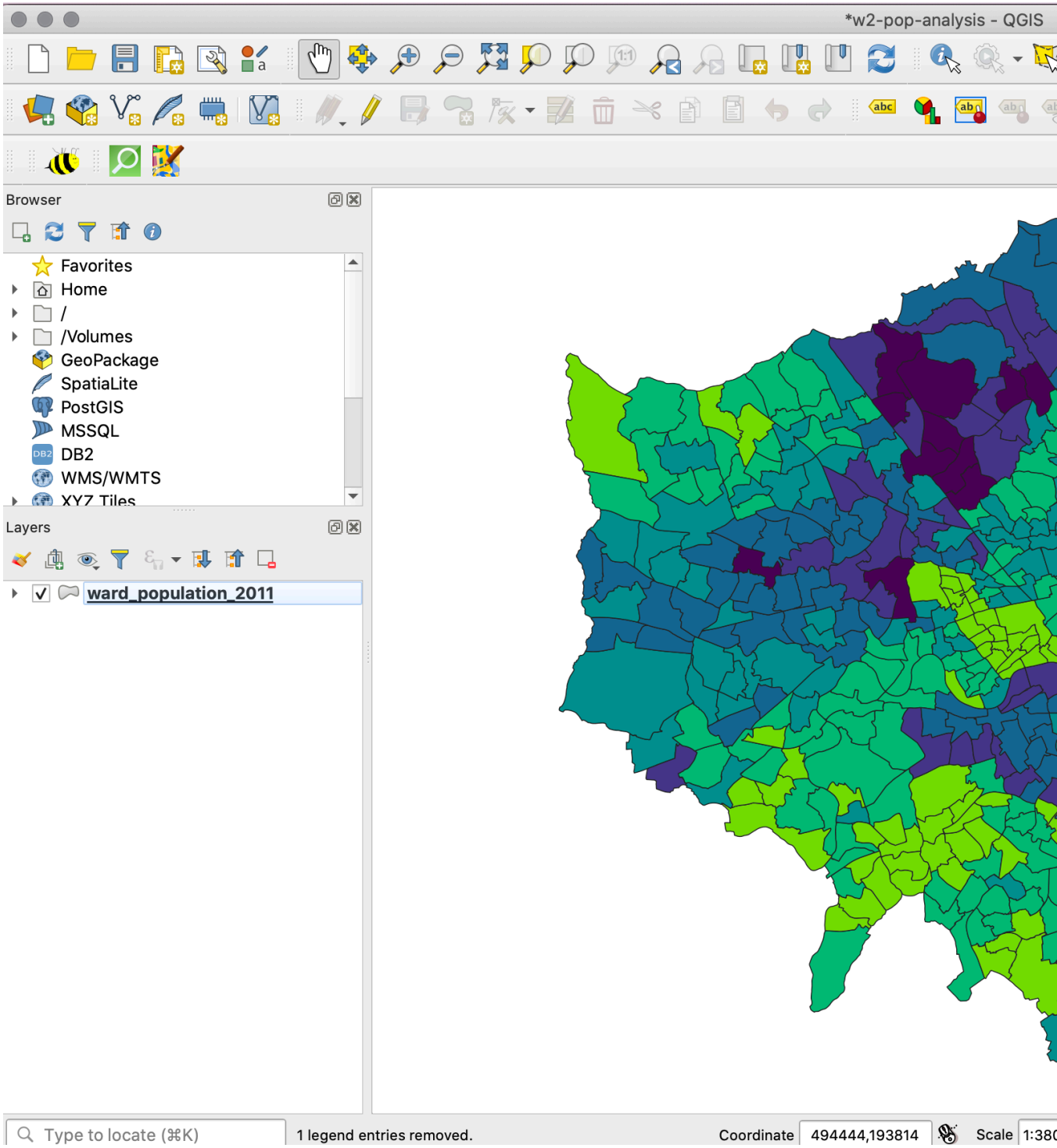


Figure 2.5: Your result.

One thing to flag is that `NODATA` means no data - whereas `0`, particularly in a scenario like this, would be an actual numeric value. It is important to remember this when processing and visualising data, to make sure you do not represent a `NODATA` value incorrectly.

2.3.7 Exporting map for visual analysis

To export your map (as is): - Select only the map layers you want to export and then opt for **Project** -> **Import/Export** -> **Export to Image** and save your final map in your `maps` folder. You may want to create a folder for these maps titled `w02`.

Next week, we will look at how to style our maps using the main map conventions (adding North Arrows, Scale Bars and Legends) but for now a simple picture will do.

2.4 Assignment

You now need to **repeat this whole process** for your 2015 and 2019 datasets. Remember, you need to:

- Load the respective Ward dataset as a Vector Layer.
- Load the respective Population dataset as a Delimited Text File Layer (remember the settings!).
- Join the two datasets together using the Join tool in the Ward dataset Properties box.
- Export your joined dataset into a new dataset within your `final` folder.
- Style your data appropriately.
- Export your maps as an image to your `maps` folder.

To make visual comparisons against our three datasets, theoretically we would need to standardise the breaks at which our classification schemes are set at. To set all three datasets to the same breaks, you can do the following:

- Right-click on the `ward_population_2019` dataset and navigate to the `Symbology` tab. Double-click on the Values for the smallest classification group and set the Lower value to 141 (this is the lowest figure across our datasets, found in the 2015 data). Click **OK**, then click **Apply**, then click **OK** to return to the main QGIS screen.
- Right-click again on the `ward_population_2019` dataset but this time, click on **Styles** -> **Copy Styles** -> **Symbology**.

- Now right-click on the `ward_population_2015` file, but this time after clicking on **Styles** -> **Paste Style** -> **Symbology**. You should now see the classification breaks in the 2015 dataset change to match those in the 2019 data.
- Repeat this for the 2011 dataset as well.
- The final thing you need to do is to now change the classification column in the `Symbology` tab for the 2015 and 2011 datasets back to their original columns and press **Apply**. You will see when you first load up their Symbology options this is set to *pop2019*, which of course does not exist within this dataset.

2.5 Before you leave

Finally, that is it for this week! Remember to save your project!

Additional Resources

Chapter 3

Data Sources

Below you will find a list of resources that you might want to explore when sourcing data for your coursework assignment or your dissertation. This is by no means an exhaustive list, but simply contains some suggestions of websites that you may want to use.

Note You are **not limited** to using these datasets for your coursework assignment or your dissertation.

3.1 Open Data

The following websites contain Open Data or link to Open Data from several respectable data providers:

- AirBnB Data
- Bike Docking Data (ready for R)
- Camden Air Action
- Consumer Data Research Centre
- DEFRA
- DIVA-GIS
- Edina (e.g. OS mastermap)
- EU Tourism Data
- Eurostat
- Geofabrik (OSM data)
- Global Weather Data
- Google Dataset Search
- Johns Hopkins COVID19 Data (ready for R
- King's College Data on Air Pollution
- London Data Store

- London Tube PM2.5 Levels
- NASA EARTHDATA
- NASA SocioEconomic Data and Applications Center (SEDAC)
- NHS Data (ready for R)
- nomis Official Census and Labour Market Statistics
- Office for National Statistics Geoportal
- Office for National Statistics
- Open Topography
- Tesco Store Data (London)
- TfL Cycling Data
- TfL Open Data
- Tidy Tuesday Data (not exclusively spatial data)
- Uber Travel Time Data
- UK COVID19 Data
- UK Data Service
- US Census Data
- US City Open Data Census
- USGS Earth Explorer
- WorldPop GitHub
- WorldPop

Some other websites that could be helpful:

- Awesome Public Datasets; general collection of datasets, although not limited to spatial data.
- Free GIS data; long list with lots of GIS datasets on many different topics and covering many different areas.

3.2 Safeguarded Data

Undergraduate students can also apply for a **Safeguarded** dataset held by the Consumer Data Research Centre. There is a process to access these **Safeguarded** datasets, which is detailed on the CDRC website. Please be aware that it normally takes 4-5 weeks for your application to be processed.

As part of the process, you will need to say in your application why you want that specific dataset and what you are planning to do with it. You will also need to have at least thought about the ethical implications of using that data and provide this with your data application (alongside your standard ethics application).

Some of the datasets held by the CDRC that you can apply for are:

- Bicycle Sharing System Docking Station Observations

- CDRC Modelled Ethnicity Proportions - LSOA Geography
- FCA Financial Lives Survey
- Local Data Company - SmartStreetSensor Footfall Data – Research Aggregated data
- NHS Hospital Admission Rates by Ethnic Group and other Characteristics
- Speedchecker Broadband Internet Speed Tests

Note Given that the application can take several weeks, the Safeguarded CDRC datasets may be useful for your dissertation work but probably not for the GEOG0030 coursework assignment. However, any of the CDRC datasets that are marked as **Open Data** do not require this application process and you can download these datasets directly after registering on the website.