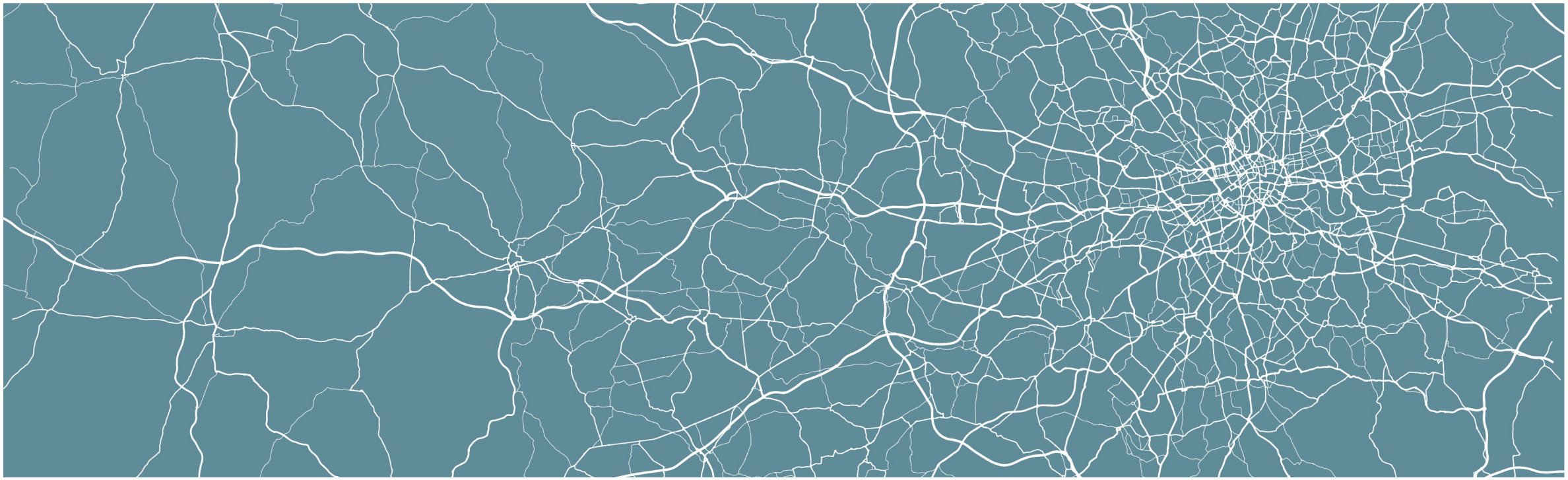
Geocomputation

GIScience and GIS software



UCL

Where are we at?

Part I: Foundational Concepts

W1 Geocomputation: An Introduction

W2 **GIScience and GIS Software**

W3 Cartography and Visualisation

] QGIS

W4 Programming for Data Analysis

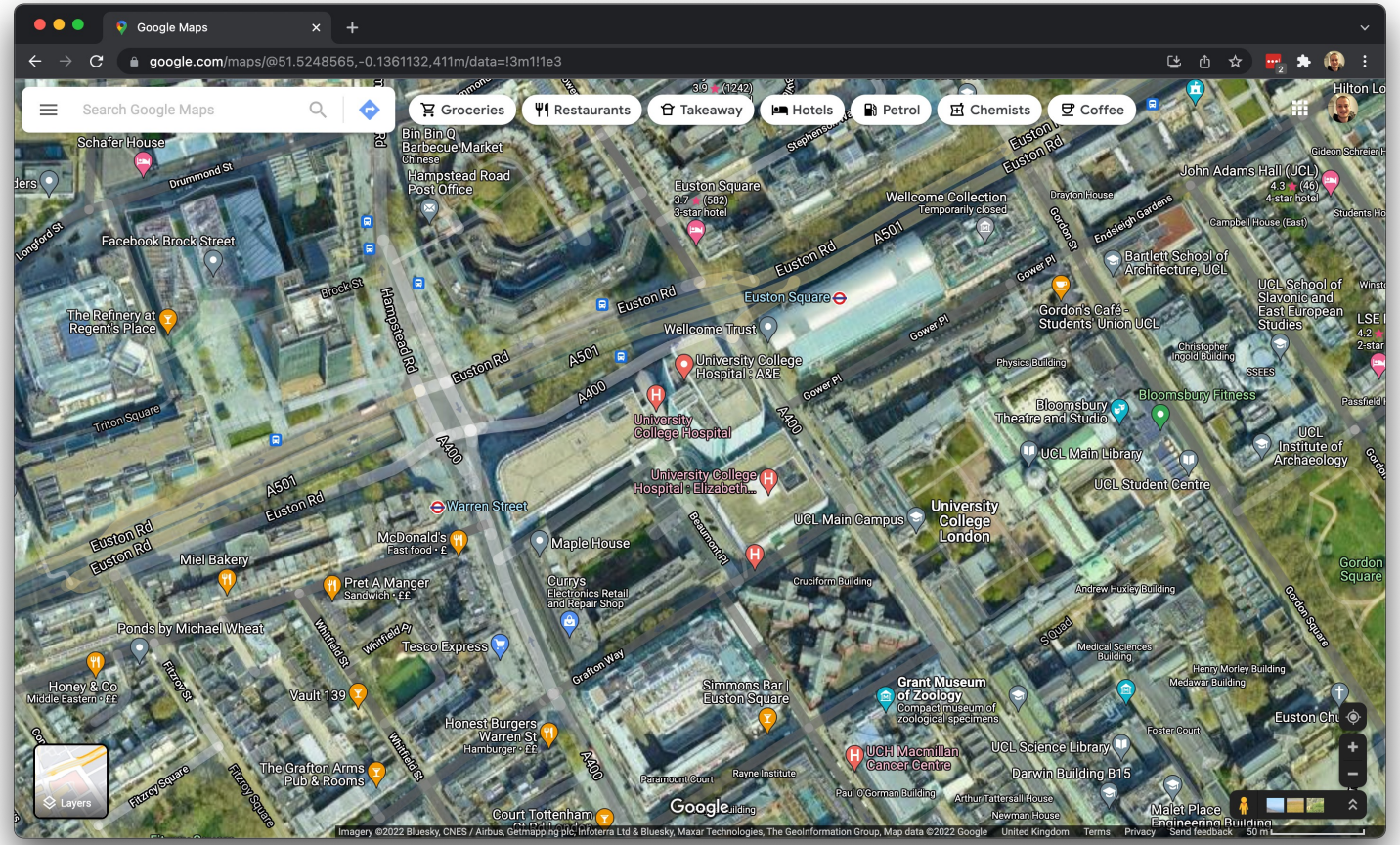
W5 Programming for Spatial Analysis

] R

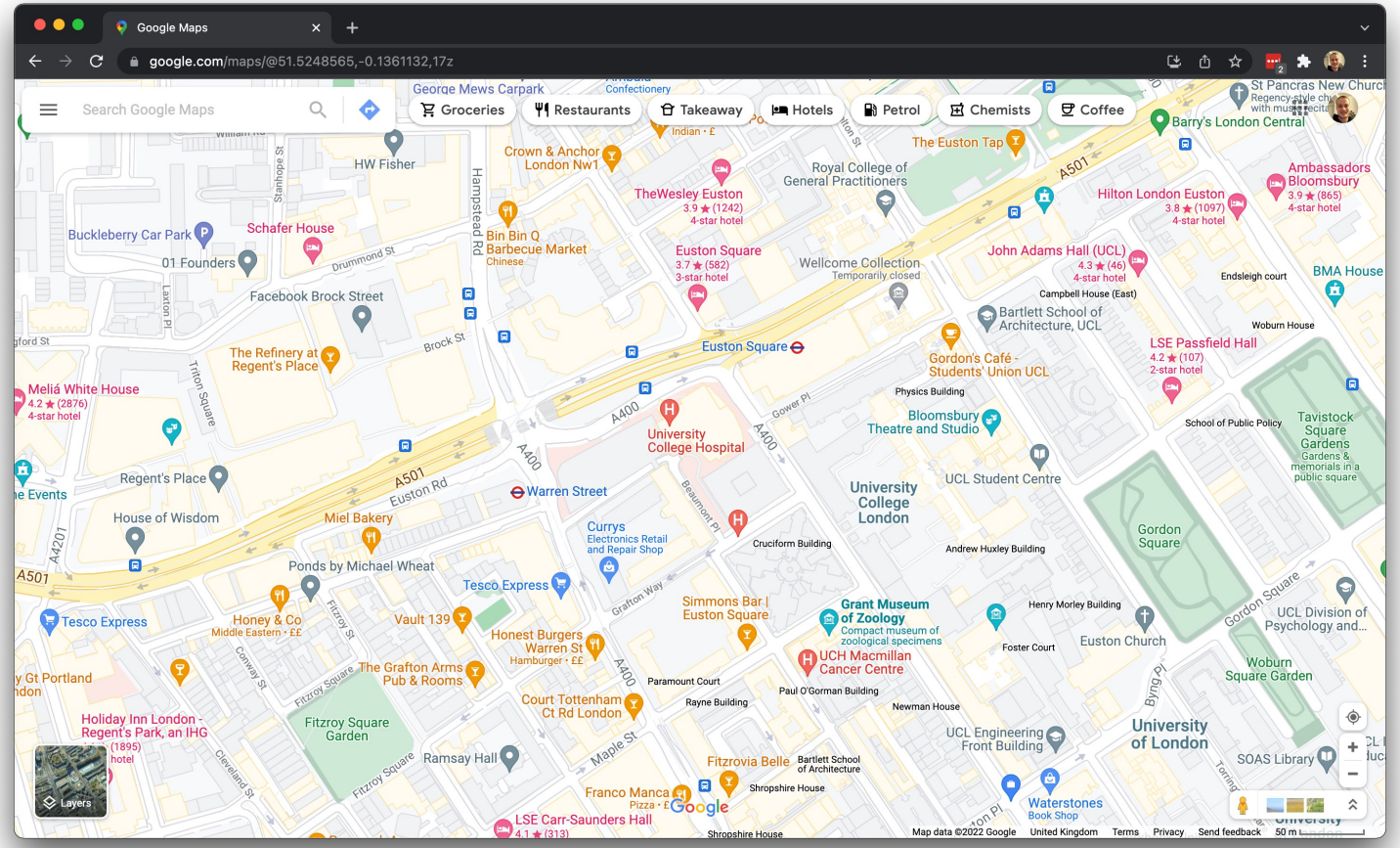
This week

- Digitally representing spatial data within GIS software
  - *Vector* data
  - *Raster* data
- GIS software and file types

Spatial modelling and digital representation



Spatial modelling and digital representation



Father of GIS

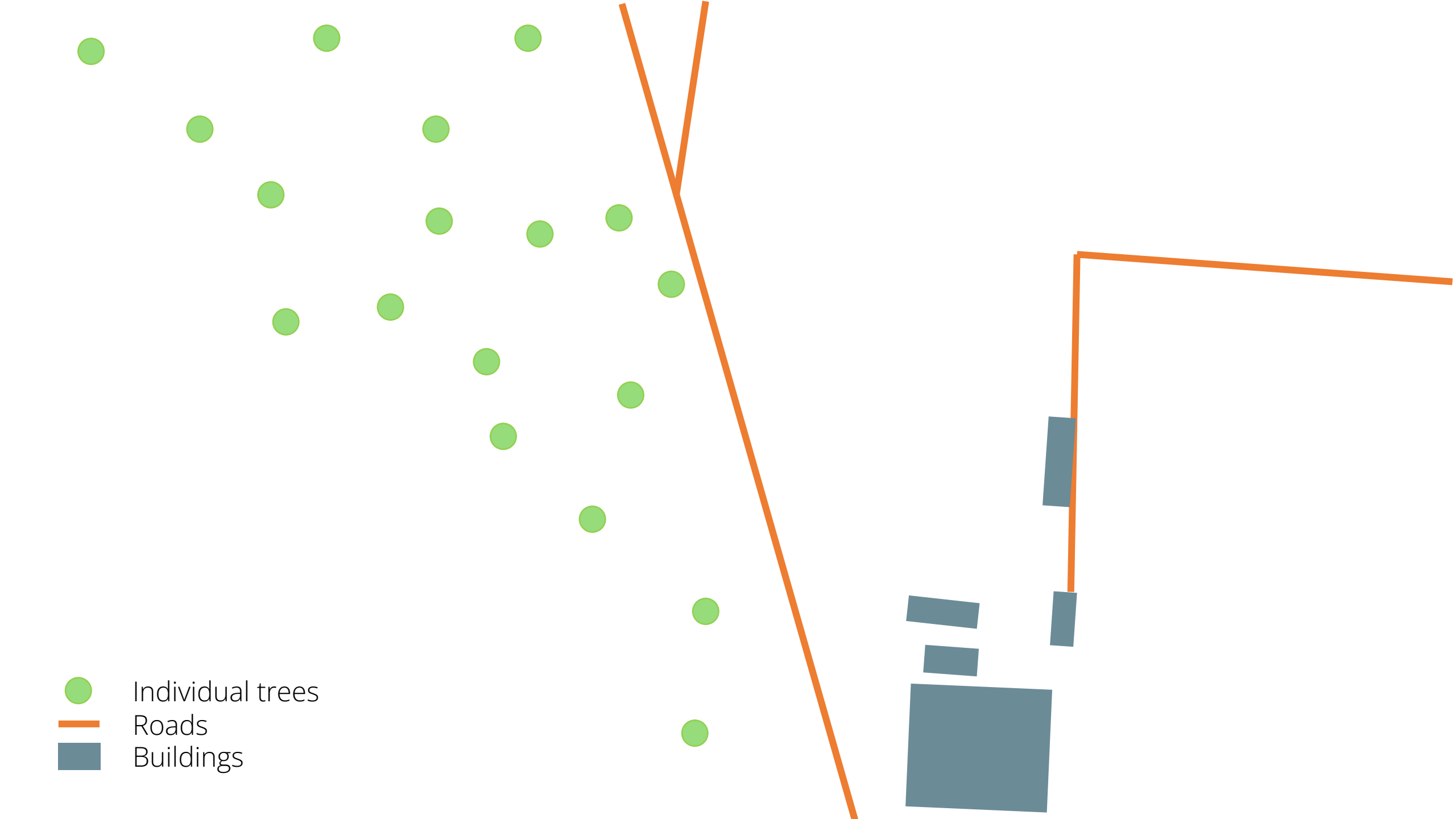
- Roger F. Tomlinson (1933-2014)
- Ph.D. dissertation: "*The application of electronic computing methods and techniques to the storage, compilation, and assessment of mapped data*" (1962, UCL).
- Conceived the idea of analysing multiple layers of spatial data within a single environment as well how to represent such spatial data in a digital format.

GIS data models

- GIScience requires spatial information to be represented in a digital format
- Traditionally, geographic information is represented in two ways:

  vector: a finite set of discrete geometric objects

  raster: images representing a surface (values, colours)


Individual trees
Roads
Buildings

Features

The objects created are called features. A feature can be described according to its characteristics which is termed an attribute in GIS. The attribute of a feature can be a numeric or text observation.

Types of features

- Individual tree: type of tree, height, width. Point feature.
- Roads: type of road, length of road, speed limit. Polyline feature.
- Buildings: type of building (commercial, residential), number of people living in the building, number of stories. Polygon feature.

Vector data

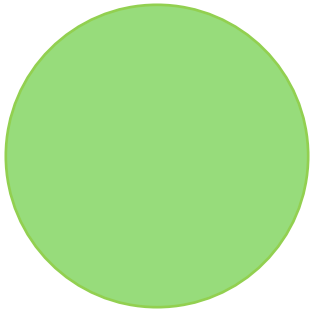
- The objects that we created of discrete entities are called vector data.
- Three types of vector data: point vectors, polylines or line vectors, and polygon vectors.

Point vector

Characteristics of a point vector in a GIS data model:

- Single XY location (coordinate)
- Has no area
- Has no length
- Geometry consists of a single node or vertex
- Used for: discrete features or ‘events’

Point vector



Point vector

Point vector

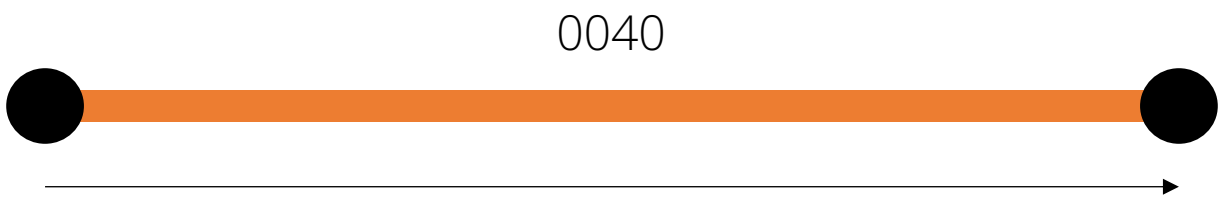
FeatureID	Type	Height
0030	Ent	500

Polyline vector

Characteristics of a polyline vector in a GIS data model:

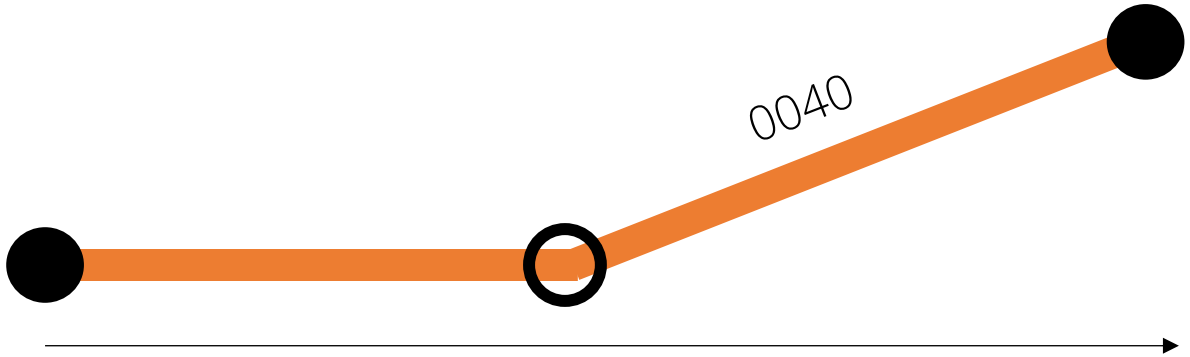
- Series of XY locations (coordinates) that form a line
- Has no area
- Has a length
- Has a direction (importance when it comes to roads, rivers, etc.)
- Can be connected to other polyline vectors to form a network
- Geometry consists of 2 nodes (start node and end node) and can have one or more vertices
- Used for: features without an area but with a length

Polyline vector



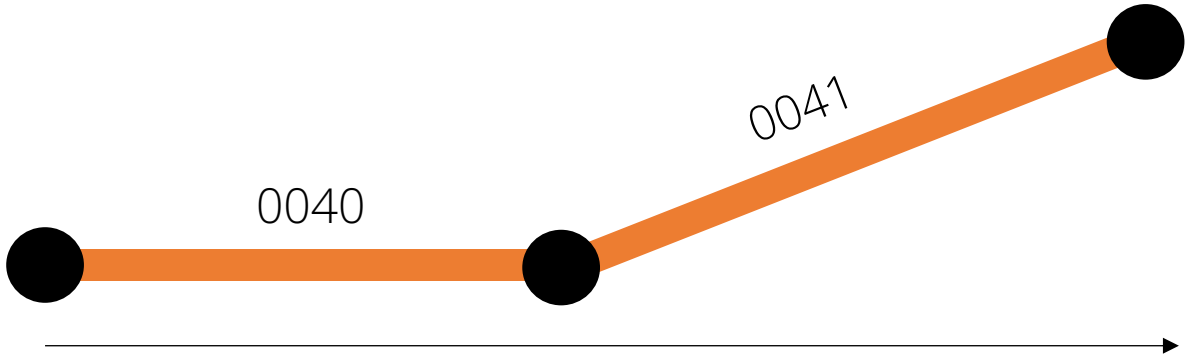
FeatureID	Type	Length
0040	Bicycle lane	1,500

Polyline vector



FeatureID	Type	Length
0040	Bicycle lane	1,650

Polyline vector



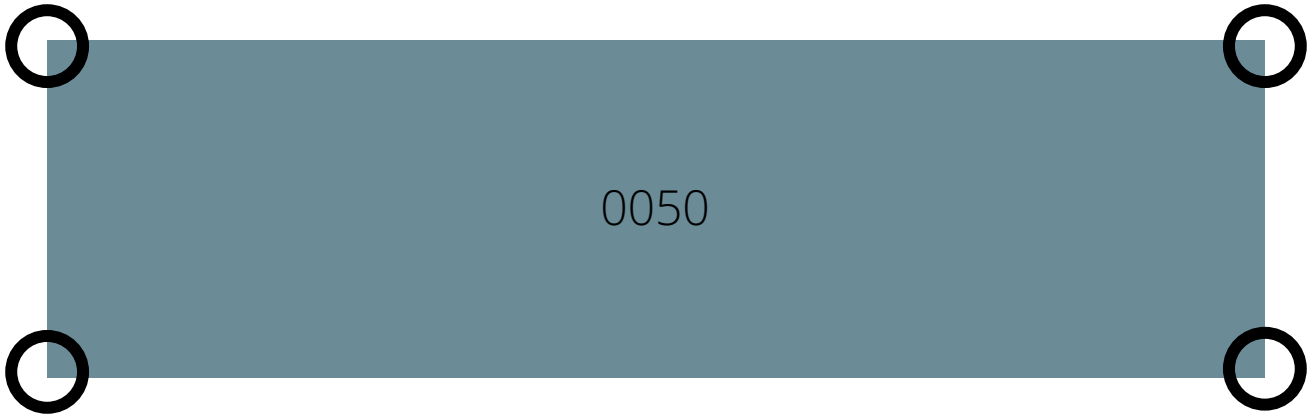
FeatureID	Type	Length
0040	Bicycle lane	600
0041	Bicycle lane	1,050

Polygon vector

Characteristics of a polyline vector in a GIS data model:

- Series of XY locations (coordinates) to form an enclosed region
- Has an area
- Has no length, but does have a perimeter
- Geometry consists of at least 3 nodes or vertices whereby the first node or vertex connects with the last one.
- Used for: features with enclosed regions such as buildings and administrative areas

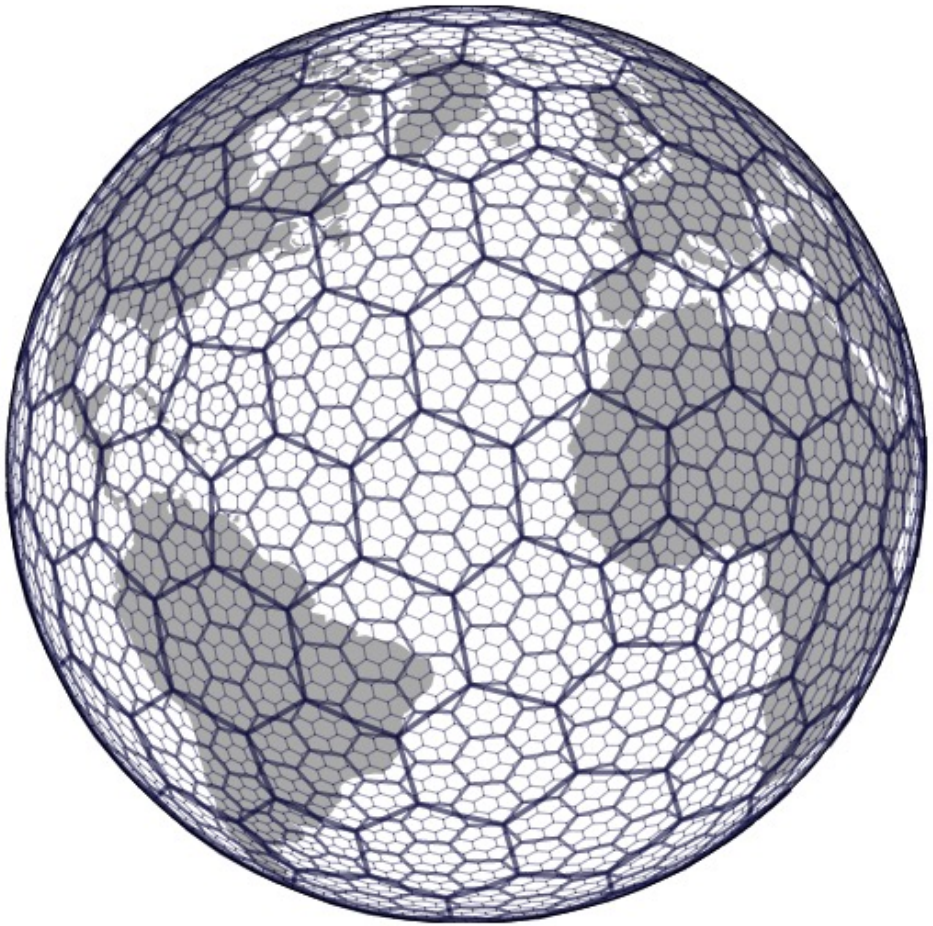
Polygon vector



FeatureID	Type	Area
0050	University building	2000

Vector

Vector



Uber. 2018. *H3: Uber's Hexagonal Hierarchical Spatial Index*. [online] https://eng.uber.com/h3/

Attribute joins

GeoID	Population
GEO0030	540
GEO0031	320

Attribute joins

FeatureID	GeoID
0050	GEO0030

Attribute joins

FeatureID	GeoID
0050	GEO0030

GeoID	Population
GEO0030	540
GEO0031	320

Attribute joins

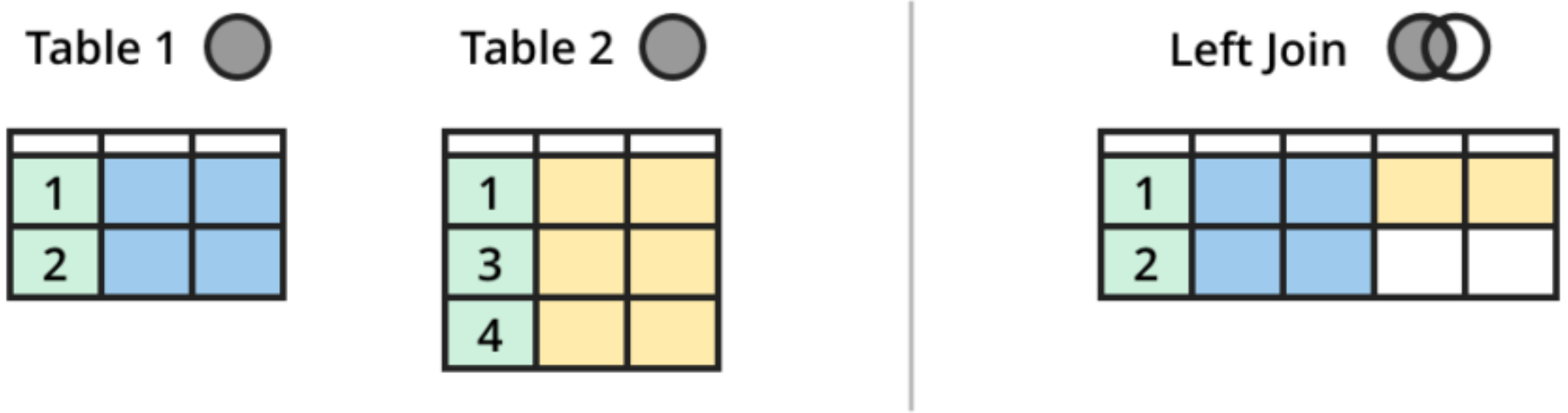
FeatureID	GeoID
0050	GEO0030

GeoID	Population
GEO0030	540
GEO0031	320

Attribute joins

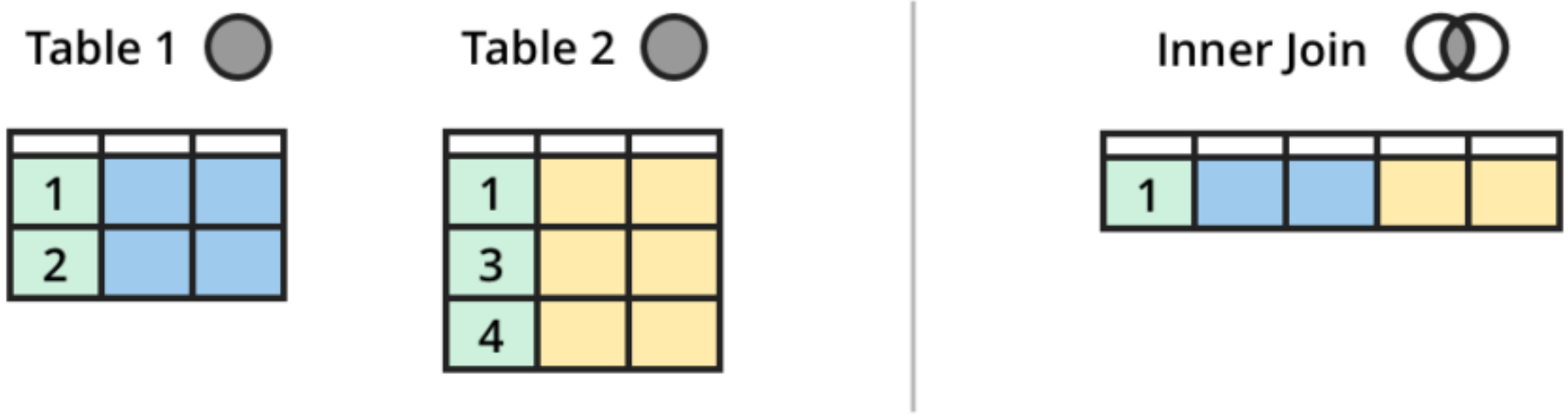
FeatureID	GeoID	Population
0050	GEO0030	540

Left joins



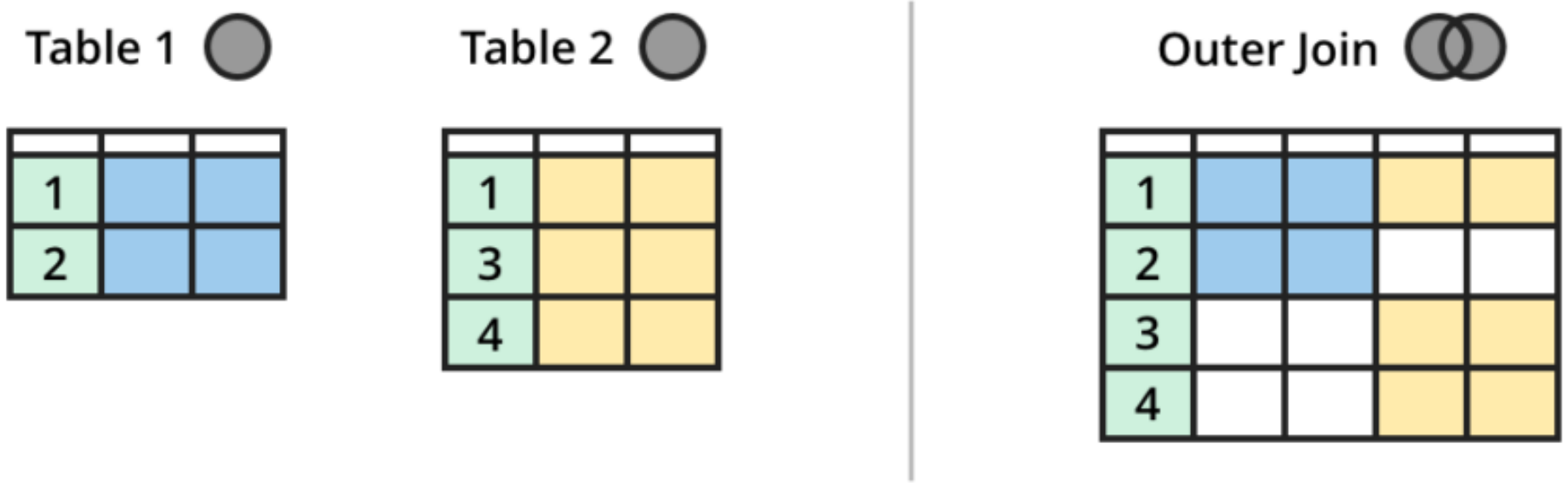
The Data School. 2019. *SQL Join Types Explained Visually*.

Inner joins



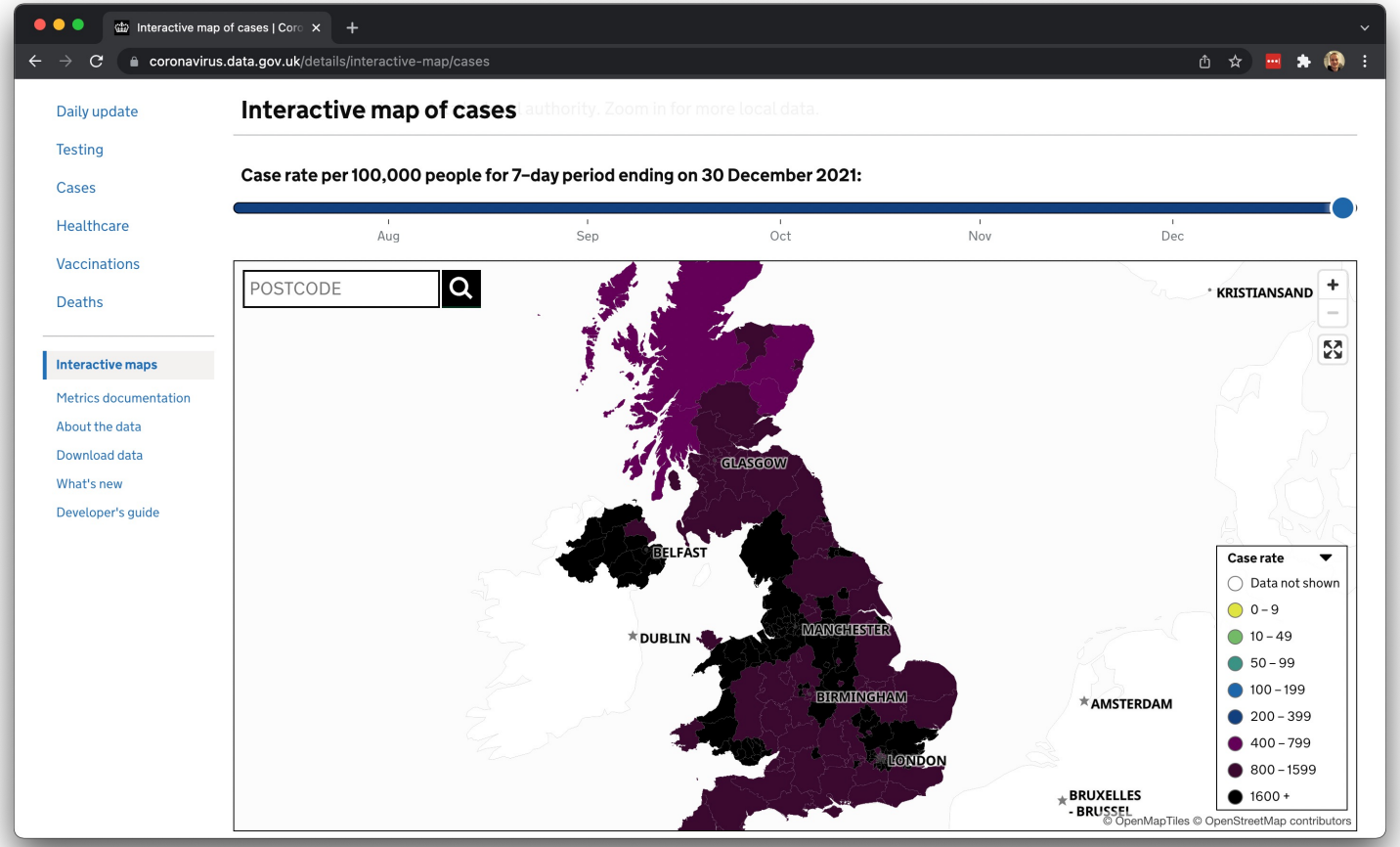
The Data School. 2019. *SQL Join Types Explained Visually*.

Outer joins



The Data School. 2019. *SQL Join Types Explained Visually*.

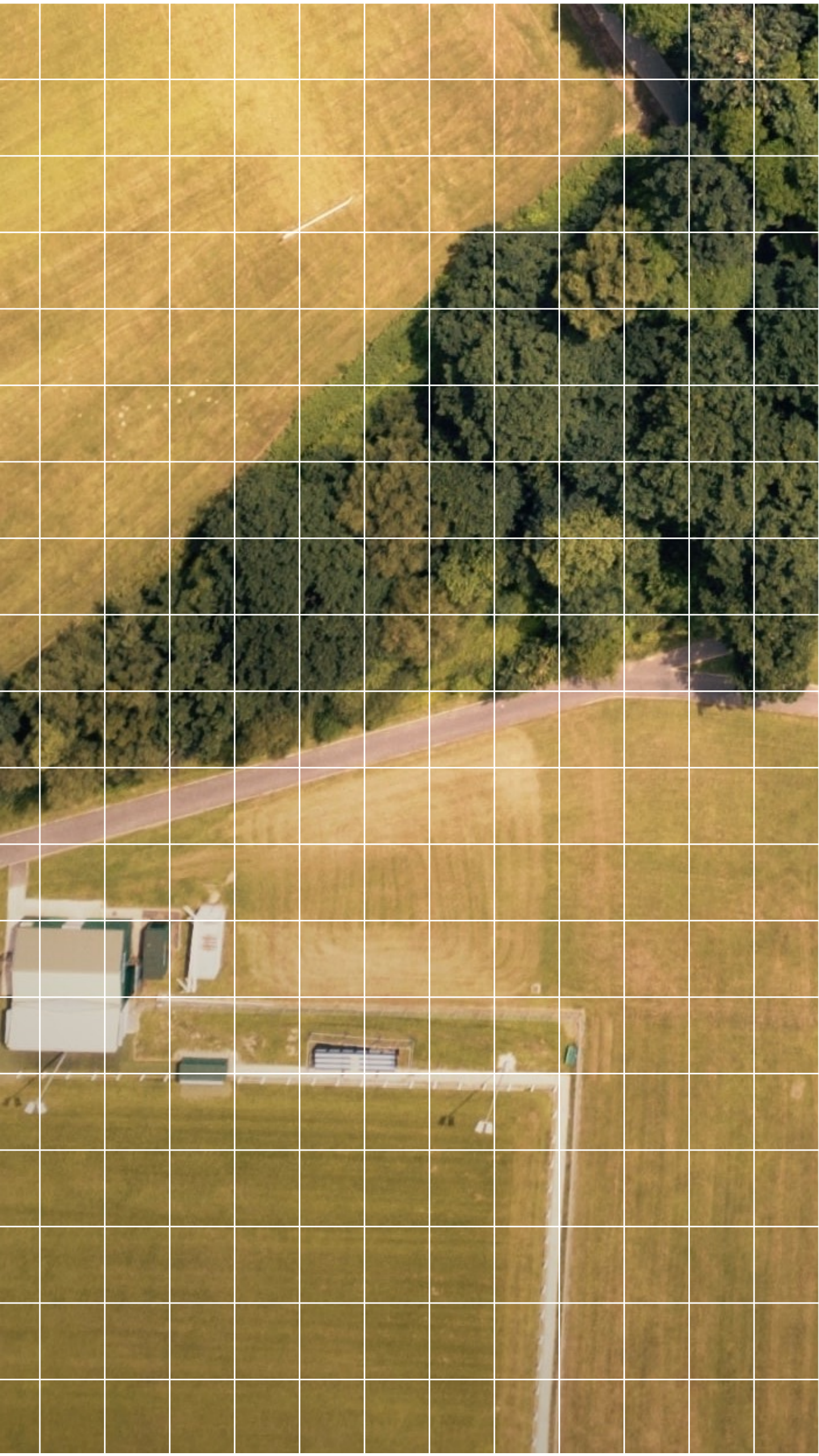
Covid cases in the UK



GIS data models

Traditionally, geographic information is represented in two ways:

- vector: a finite set of geometric objects
- raster: images representing a surface (values, colours)



8	9	9	10	0	10	10	10	0	0	0	7	5	3	0	0	0	0	1
8	9	9	10	10	0	10	9	9	0	0	5	3	0	0	0	0	0	0
8	8	9	9	10	0	0	9	8	7	5	0	0	0	1	0	0	0	0
5	8	8	9	10	10	0	9	7	5	0	0	5	5	5	0	0	0	1
3	5	8	9	9	10	0	0	3	0	0	0	5	0	0	1	0	0	2
2	5	8	8	9	9	10	0	0	0	1	5	0	0	0	0	0	0	1
2	4	6	8	8	9	0	0	0	1	5	0	0	5	5	5	0	0	1
0	3	6	8	8	0	0	0	0	5	0	5	5	5	5	5	0	0	0
2	2	5	8	0	0	0	0	0	0	5	5	5	5	5	5	3	0	0
0	2	5	0	0	1	2	3	4	4	4	4	4	4	4	5	0	0	0
0	0	0	0	1	1	1	1	4	4	4	4	4	4	4	5	0	0	0
0	0	1	1	2	2	2	2	3	3	3	3	3	3	3	4	0	3	0
1	1	1	1	2	2	3	3	3	3	1	1	1	1	1	2	3	4	3

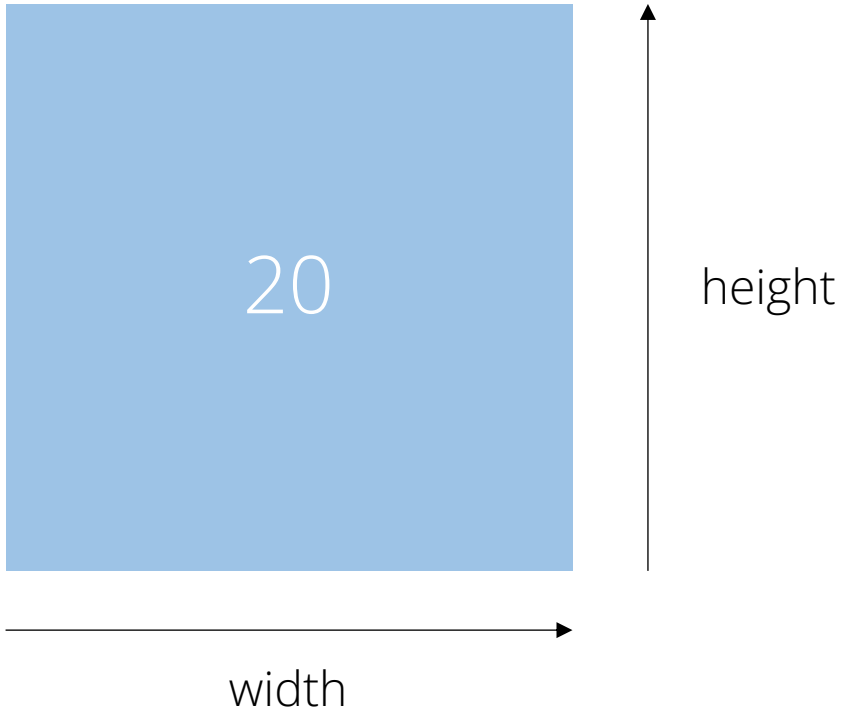
Raster data

- Unlike the vector data. The above feature describes rainfall levels across the surface of the landscape – the feature is is not measured discretely but on a continuous surface to show gradient in changes.
- This non-discrete feature is classed a raster data.

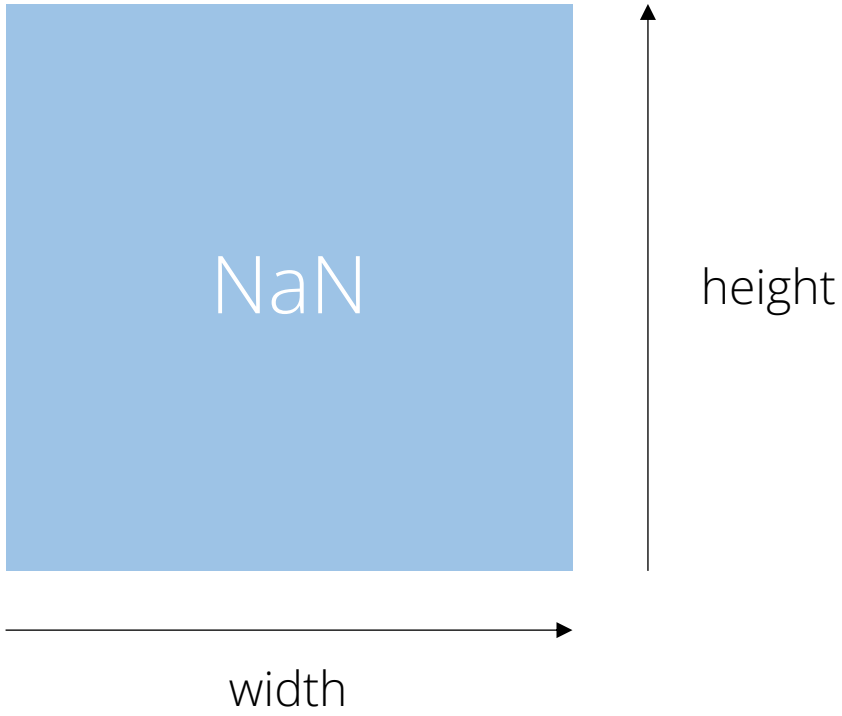
Raster data

- Raster data is represented by a matrix of pixels or grid-cells that contains a numeric or text value for a feature its representing.
- It is composed of rows and columns.
- Each pixel or grid-cell has a resolution (or size for height and width).

Raster data



Raster data



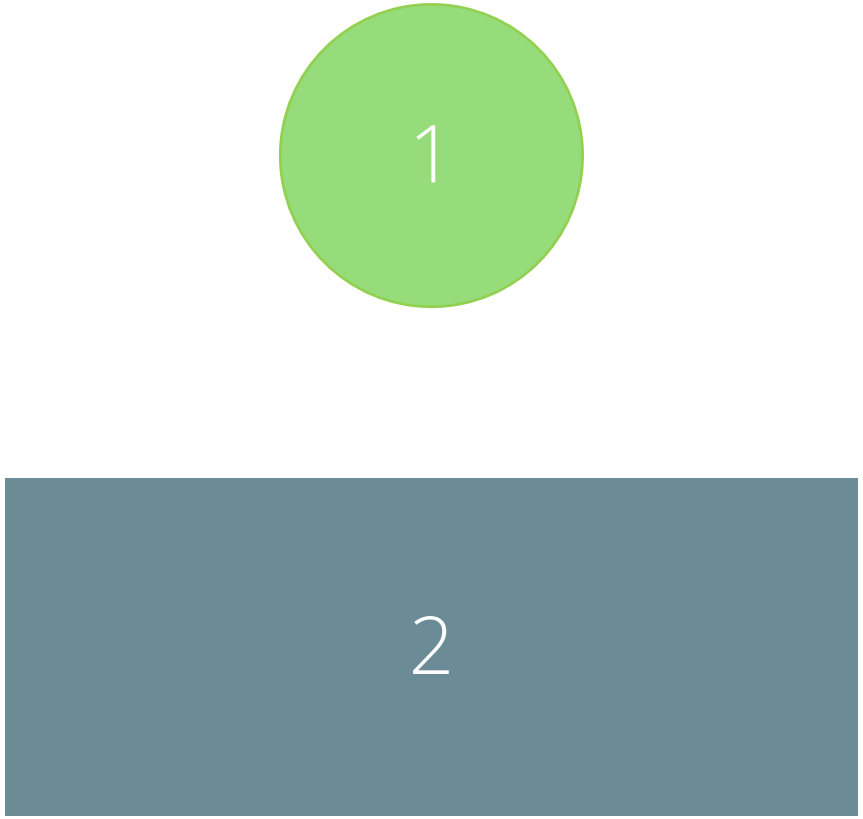
Raster data

Sample scheme

- Both for the vector and raster data model real life features are ‘sampled’ or represented in a certain way. How to represent the spatial information? With which level of detail?
- It must be fine enough to provide general consistency in our feature or field as well as accurately represent its distribution.
- It must be fine enough also to capture the important changes in our feature or field, e.g. a turn in a road, or a certain measurement change in a variable, such as temperature or rainfall.
- But we must also not over sample – we need to consider efficiency and efficacy of our sampling as we collect the data and store it digitally.

Sample scheme

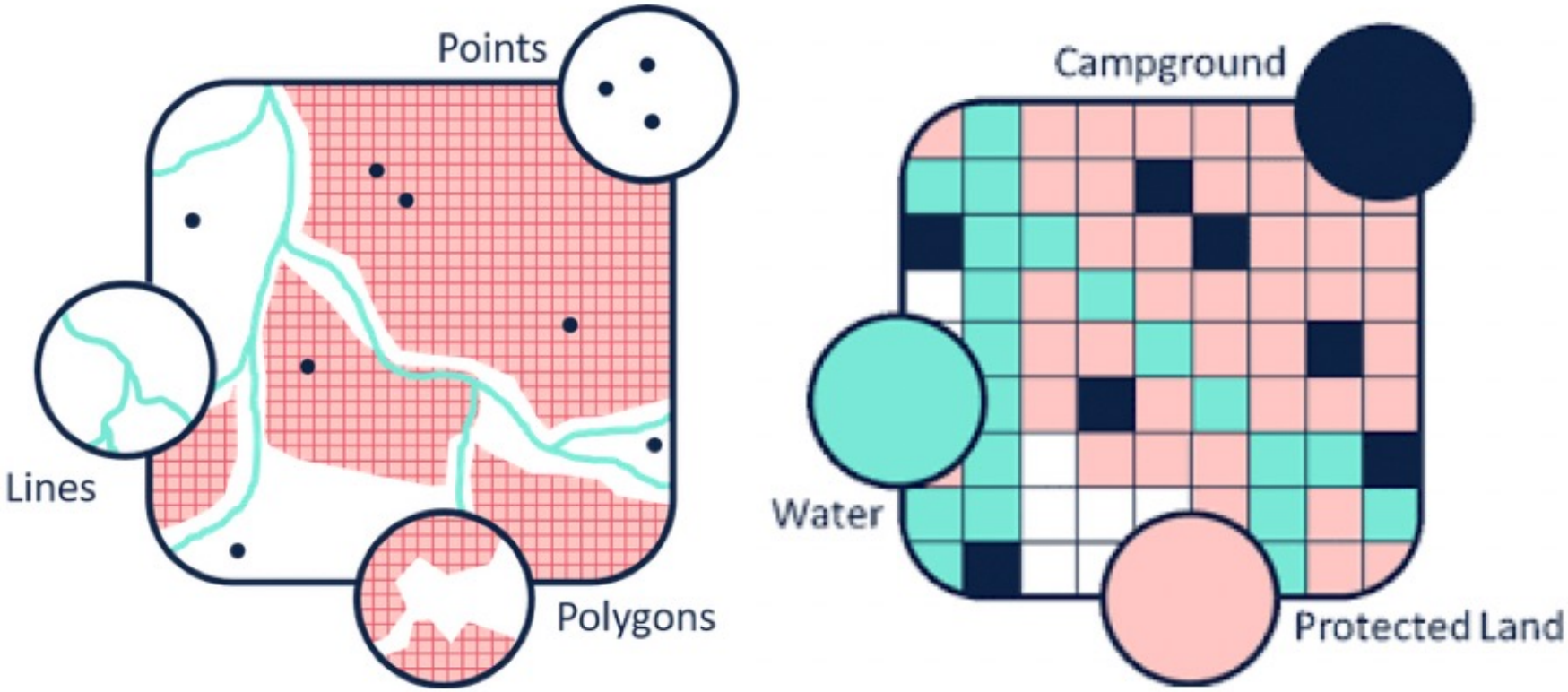
Representing Torrington Place 1-19



Vector versus raster

- Why not vectorise everything? Can add as many attributes as we like?
- ”Raster is faster but vector is corrector”
- Depends on your application and intended analysis.
- Vector and raster data both have their advantages and disadvantages.

Vector versus raster



Vector versus raster

	Advantages	Disadvantages
Raster	Map Algebra with raster data is usually quick and easy to perform	Linear features and paths are difficult to display
	Some specific use cases can only be achieved with raster data (e.g. modeling water flow over the land surface)	Subject to a pixelated look and feel
		Datasets can become very large because they record values for each cell
Vector	Graphical output is generally more aesthetically-pleasing	Continuous data is poorly stored and displayed
	Higher geographic accuracy because data isn't dependent on grid size	Needs a lot of work and maintenance to ensure that it is accurate and reliable

Spatial data formats

- There are a number of commonly used file formats that store vector and raster data, some of which you will come across during this course and it is important to understand what they are, how they represent data and how you can use them.
- Different file formats for vector data and raster data.
- Common vector formats: shapefile, GeoJSON, GeoPackage
- Common raster formats: GeoTIFF, GeoPackage

Shapefiles

- Perhaps the most (in)famous file format.
- Widely used, despite being outdated, especially limitations of `.dbf` format.
- A shapefile is not a single file, but a collection of files of which at least three are needed for the data to be displayed in GIS software.

Shapefiles


Shapefile
@shapefile
"The length of shapefile field names is limited which diminishes the format's usability, you know."
#mappymeme

Shapefiles

- `.shp` contains the feature geometry. *Mandatory.*
- `.shx` index file which stores the position of the feature's ID in the `.shp` file. *Mandatory.*
- `.dbf` stores alle attribute information associated with the records. *Mandatory.*
- `.prj` contains the coordinate system information and projection. *Optional but not really.*
- `.xml` general metadata. *Optional.*
- `.cpg` encoding information. *Optional.*
- `.sbn` optimisation file for spatial queries. *Optional.*

Shapefiles

GeoJSON

- GeoJSON (Geospatial Data Interchange format for JavaScript Object Notation) is becoming an increasingly popular spatial data file, particularly for web-based mapping as it is based on JavaScript Object Notation.
- Unlike a shapefile in a GeoJSON, the attributes, boundaries and projection information are all contained in the same file.
- How to spot in the wild: `.json` or .geojson

GeoJSON

- Point
```
{"type": "Point", "coordinates": [30.0, 10.0]}
```
- LineString
```
{"type": "LineString", "coordinates": [[30.0, 10.0], [10.0, 30.0], [40.0, 40.0] ]}
```
- Polygon
```
{"type": "Polygon", "coordinates": [[[30.0, 10.0], [40.0, 40.0], [20.0, 40.0], [10.0, 20.0], [30.0, 10.0]]]}
```

GeoPackage

- A GeoPackage is an open, standards-based, platform-independent, portable, self-describing, compact format for transferring geospatial data.
- It stores spatial data layer as a single file, based upon an SQLite database.
- How to spot in the wild: .gpkg

GeoTIFF

- Geostationary Earth Orbit Tagged Image File Format.
- Created by NASA and is a standard public domain format.
- All necessary information to establish the location of the data on Earth’s surface is embedded into the image. This includes all details on map projection.
- How to spot in the wild: .tiff

GI file formats and GI systems

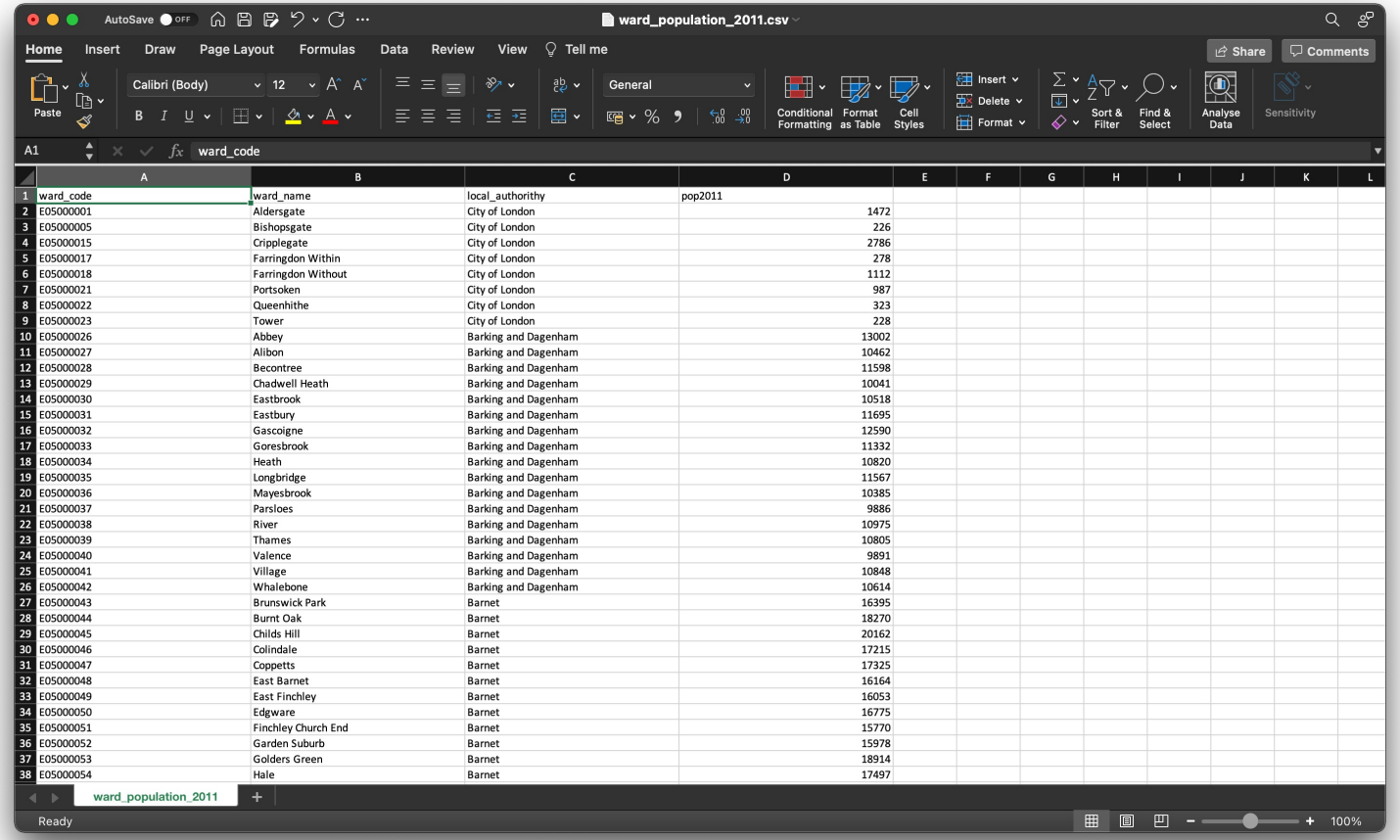
- Standard GI systems can read most file types without any problems or need for conversion.
- When using a programming language it is sometimes necessary to use a dedicated function to read in the data – so important to know which format you are dealing with (but libraries / packages exist to do just that).

On the topic of file formats: to join attribute data to your spatial files: .csv

CSV files

- “Comma Separated Values” or “Character Separated Values”.
- Format to store tabular data in rows and columns.
- Plain text rather than a binary file (e.g. Microsoft Excel).
- No limits on number of rows, columns, cells.

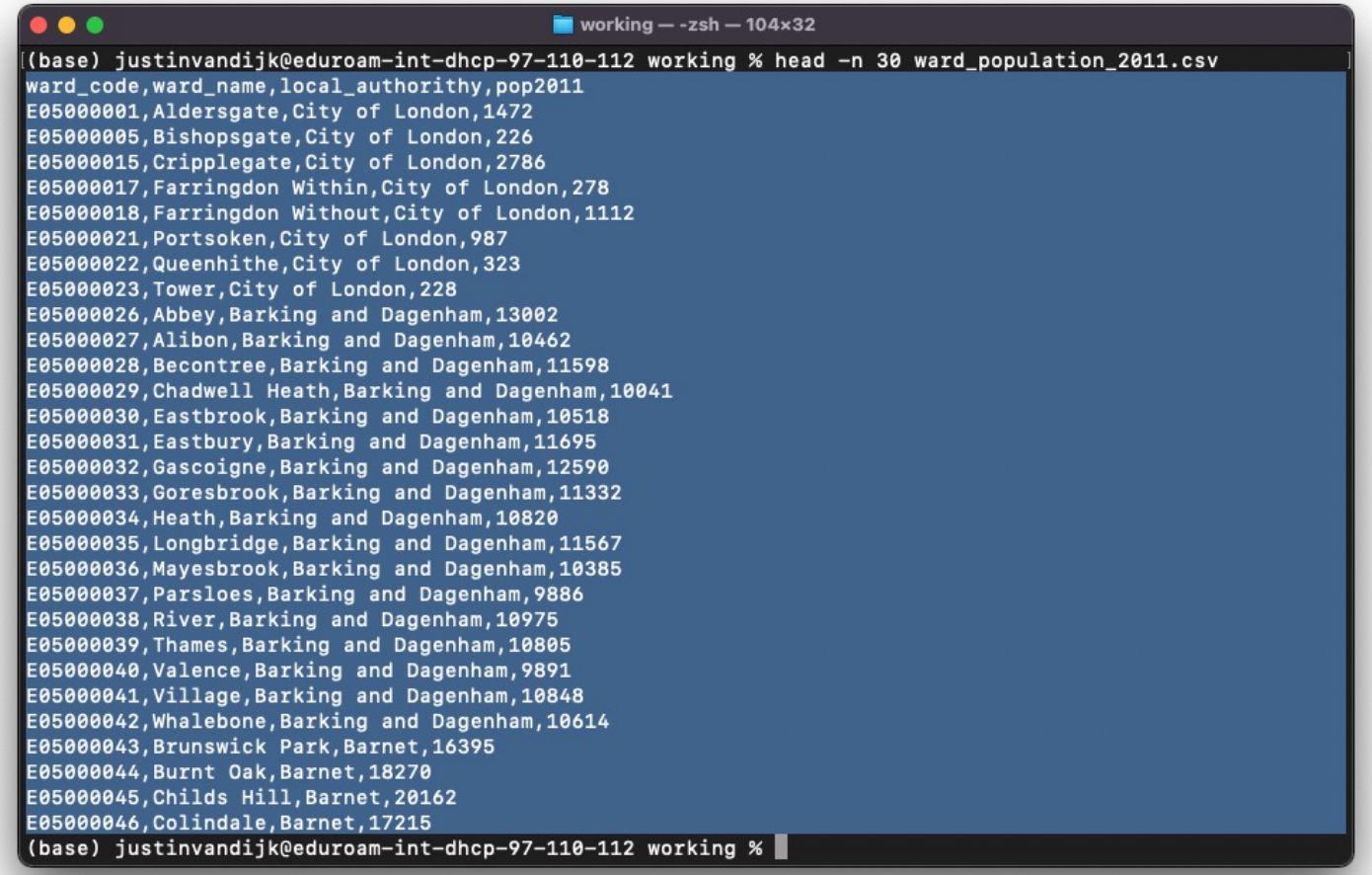
CSV files



ward_population_2011.csv

	A	B	C	D
1	ward_code	ward_name	local_authorithy	pop2011
2	E05000001	Aldersgate	City of London	1472
3	E05000005	Bishopsgate	City of London	226
4	E05000015	Cripplegate	City of London	2786
5	E05000017	Farringdon Within	City of London	278
6	E05000018	Farringdon Without	City of London	1112
7	E05000021	Portsoken	City of London	987
8	E05000022	Queenhithe	City of London	323
9	E05000023	Tower	City of London	228
10	E05000026	Abbey	Barking and Dagenham	13002
11	E05000027	Alibon	Barking and Dagenham	10462
12	E05000028	Becontree	Barking and Dagenham	11598
13	E05000029	Chadwell Heath	Barking and Dagenham	10041
14	E05000030	Eastbrook	Barking and Dagenham	10518
15	E05000031	Eastbury	Barking and Dagenham	11695
16	E05000032	Gascoigne	Barking and Dagenham	12590
17	E05000033	Goresbrook	Barking and Dagenham	11332
18	E05000034	Heath	Barking and Dagenham	10820
19	E05000035	Longbridge	Barking and Dagenham	11567
20	E05000036	Mayesbrook	Barking and Dagenham	10385
21	E05000037	Parsloes	Barking and Dagenham	9886
22	E05000038	River	Barking and Dagenham	10975
23	E05000039	Thames	Barking and Dagenham	10805
24	E05000040	Valence	Barking and Dagenham	9891
25	E05000041	Village	Barking and Dagenham	10848
26	E05000042	Whalebone	Barking and Dagenham	10614
27	E05000043	Brunswick Park	Barnet	16395
28	E05000044	Burnt Oak	Barnet	18270
29	E05000045	Childs Hill	Barnet	20162
30	E05000046	Colindale	Barnet	17215
31	E05000047	Coppetts	Barnet	17325
32	E05000048	East Barnet	Barnet	16164
33	E05000049	East Finchley	Barnet	16053
34	E05000050	Edgware	Barnet	16775
35	E05000051	Finchley Church End	Barnet	15770
36	E05000052	Garden Suburb	Barnet	15978
37	E05000053	Golders Green	Barnet	18914
38	E05000054	Hale	Barnet	17497

CSV files


working — -zsh — 104×32
(base) justinvandijk@eduroam-int-dhcp-97-110-112 working % head -n 30 ward_population_2011.csv
ward_code,ward_name,local_authorithy,pop2011
E05000001,Aldersgate,City of London,1472
E05000005,Bishopsgate,City of London,226
E05000015,Cripplegate,City of London,2786
E05000017,Farringdon Within,City of London,278
E05000018,Farringdon Without,City of London,1112
E05000021,Portsoken,City of London,987
E05000022,Queenhithe,City of London,323
E05000023,Tower,City of London,228
E05000026,Abbey,Barking and Dagenham,13002
E05000027,Alibon,Barking and Dagenham,10462
E05000028,Becontree,Barking and Dagenham,11598
E05000029,Chadwell Heath,Barking and Dagenham,10041
E05000030,Eastbrook,Barking and Dagenham,10518
E05000031,Eastbury,Barking and Dagenham,11695
E05000032,Gascoigne,Barking and Dagenham,12590
E05000033,Goresbrook,Barking and Dagenham,11332
E05000034,Heath,Barking and Dagenham,10820
E05000035,Longbridge,Barking and Dagenham,11567
E05000036,Mayesbrook,Barking and Dagenham,10385
E05000037,Parsloes,Barking and Dagenham,9886
E05000038,River,Barking and Dagenham,10975
E05000039,Thames,Barking and Dagenham,10805
E05000040,Valence,Barking and Dagenham,9891
E05000041,Village,Barking and Dagenham,10848
E05000042,Whalebone,Barking and Dagenham,10614
E05000043,Brunswick Park,Barnet,16395
E05000044,Burnt Oak,Barnet,18270
E05000045,Childs Hill,Barnet,20162
E05000046,Colindale,Barnet,17215
(base) justinvandijk@eduroam-int-dhcp-97-110-112 working %

Conclusion

- Two GIS data models: the vector data model and the raster data model.
- The vector model uses points, line, and polygon segments to identify locations on the earth while the raster model uses a series of cells to represent locations on the earth.
- Both GIS data models accommodate attributes: the qualitative or quantitative descriptions of the feature.
- Per definition any data model is an incomplete representation of reality.
- GI systems have been designed to work with a variety of different file types.

Computer tutorial

Questions

Justin van Dijk

j.t.vandijk@ucl.ac.uk