

Lecture 1: Introduction & Geometric objects

Instructor: Ane Vierø

Jan 30, 2023

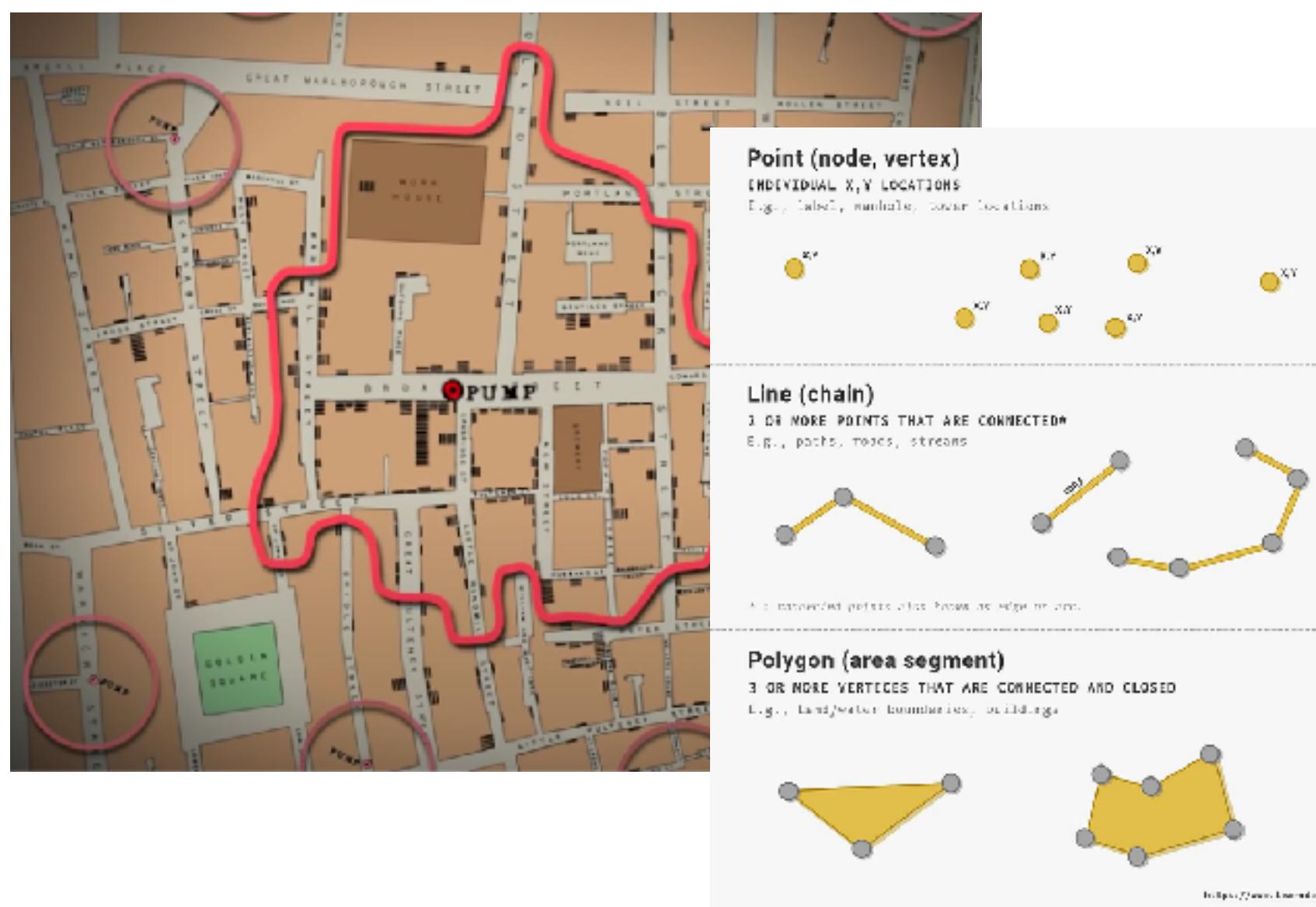


Today you will learn about GDS, John Snow and shapely

What is Geospatial Data Science?



John Snow & shapely



Course Overview & Logistics

Number	Date	Topic	Lecturer	Exercise
1	Jan 30	Intro	Ane	Env install
2	Feb 6	Spatial Data: Vector	Ane	Vector
3	Feb 13	Choropleth Mapping	Ane	Choropleth
4	Feb 20	Spatial Weights	Ane	Weights
5	Feb 27	Spatial Autocorrelation	Ane	Autocorrelation
6	March 6	Spatial Clustering	Ane	Clustering
7	March 13	Point Pattern Analysis	Marina	Point Patterns
8	March 20	OpenStreetMap	Ane	OSM
9	March 27	Spatial Networks	Ane	Networks
10	April 3	Spatial Data: Raster	Marina	Raster
11	April 17	GeoAI	Marina	GeoAI
12	April 24	Mobility	Ane	Mobility
13	May 1	Big Spatial Data	Ane	Project Questions
14	May 8	Spatial Data Viz	Ane	Project Questions

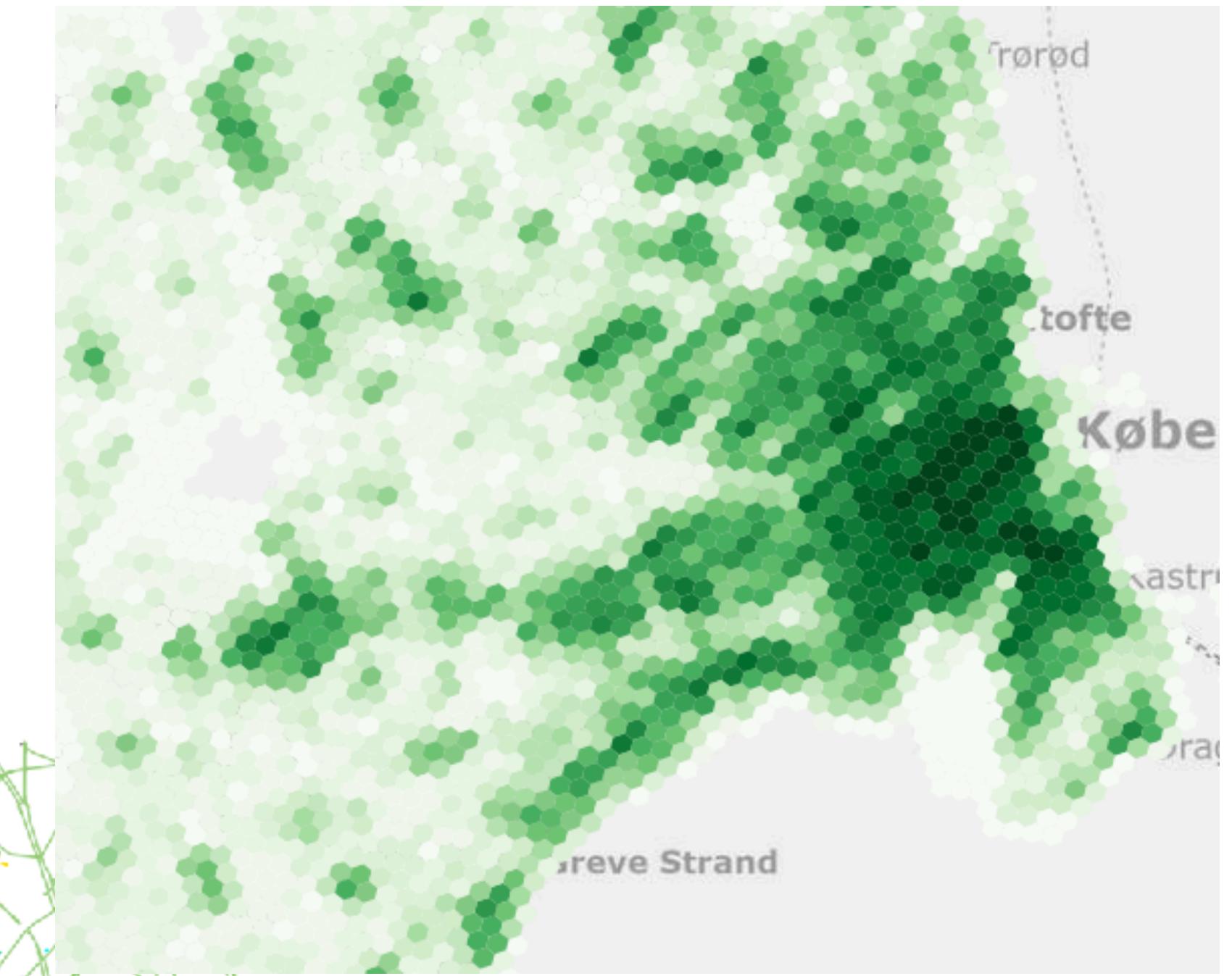
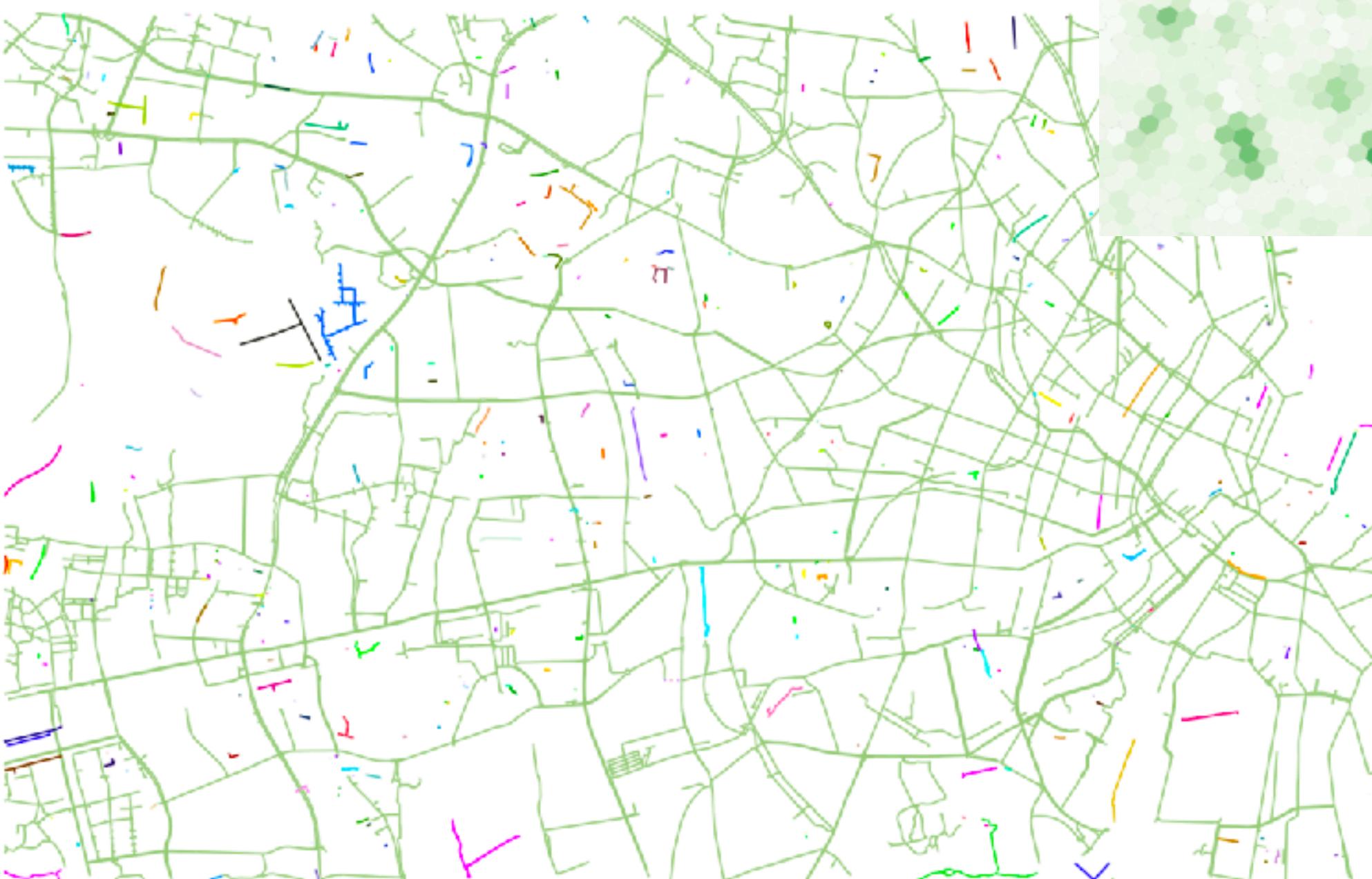
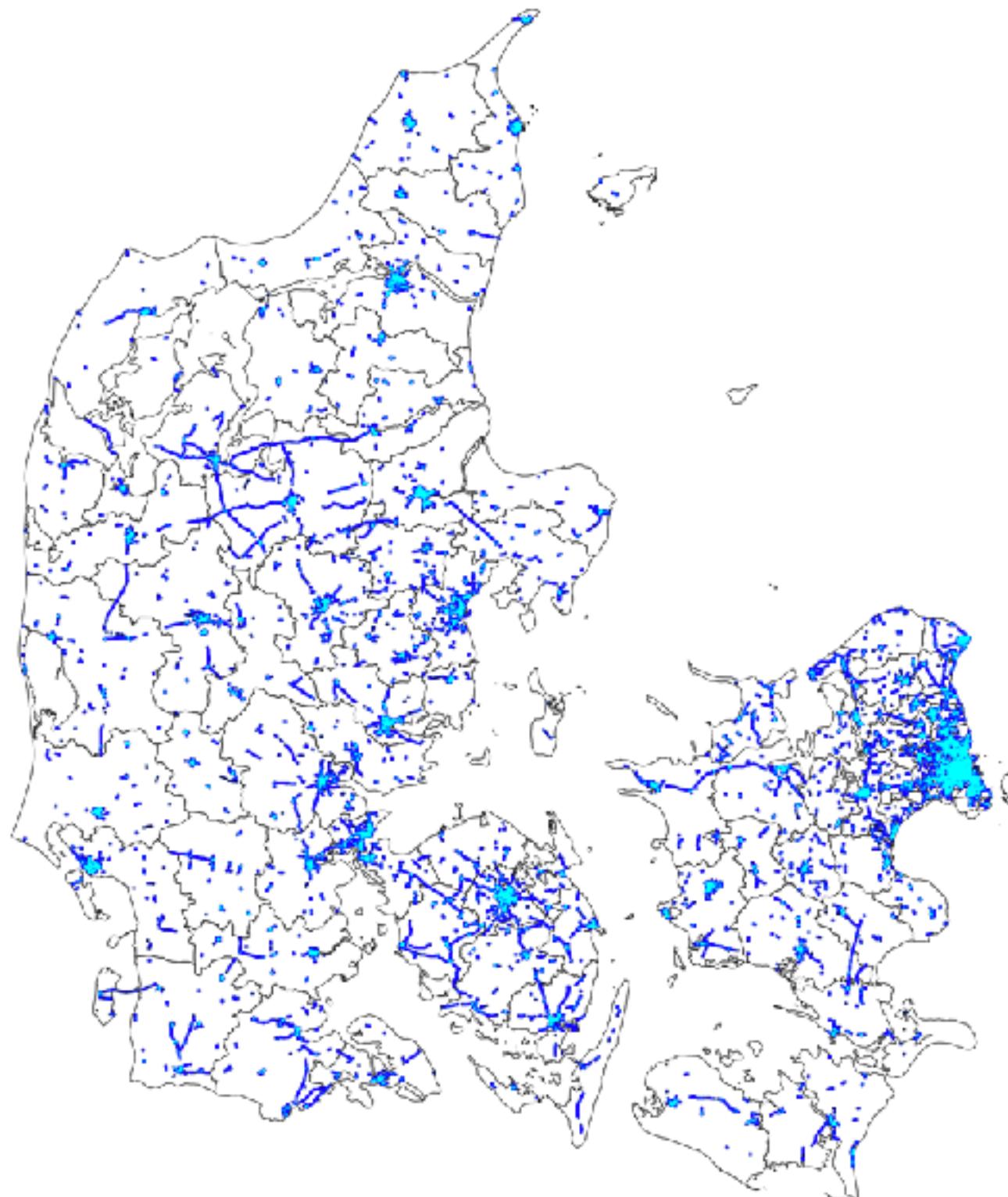
Ane Rahbek Vierø

Phd student at NERDS, CS, ITU (2022 - present)

Msc in Geographic Information Science, Lund Uni (2020)

Msc in Urban Planning & Human Geography, RUC (2019)

Bachelor in Urban Planning, RUC (2016)



anev@itu.dk

<https://anerv.github.io/>

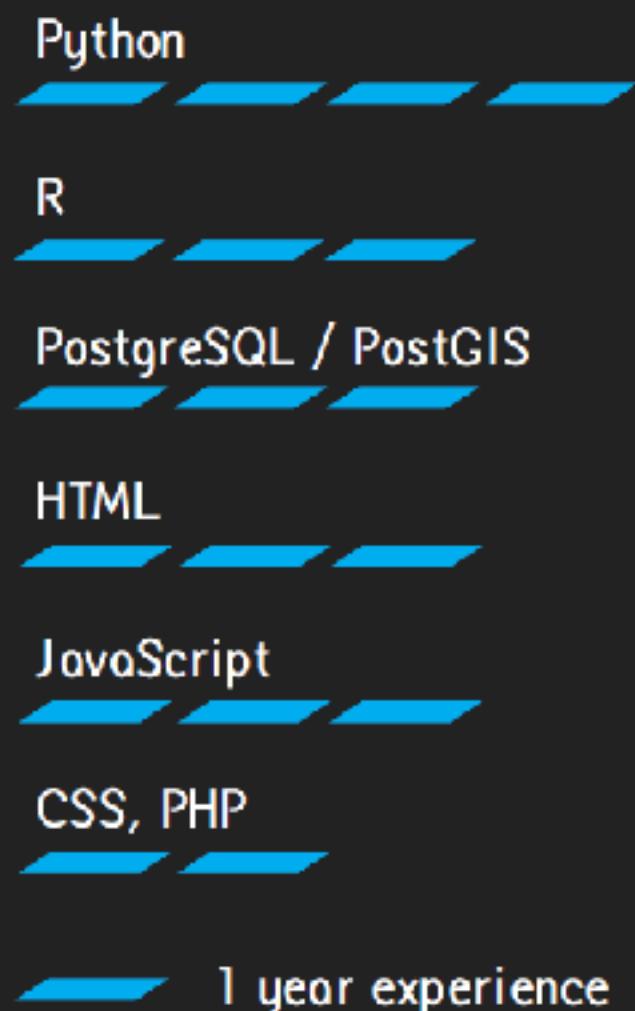
<https://github.com/anerv>



Marina Georgati
PhD Fellow

T: +45 52812505
E: geor@itu.dk
github.com/mgeorgati/

Programming skills



Profile

PhD fellow interested in working at the intersection of socio-economic studies and computer sciences. Currently exploring Machine Learning solutions for cohort-based spatial population projections. Passionate with urban analysis, demography and programming, I focus on migration patterns and their spatio-temporal variations at local level in European cities within the scope of the H2020 FUME Project.

Education

Feb. 2020 - Present PhD Fellow, Aalborg University Copenhagen

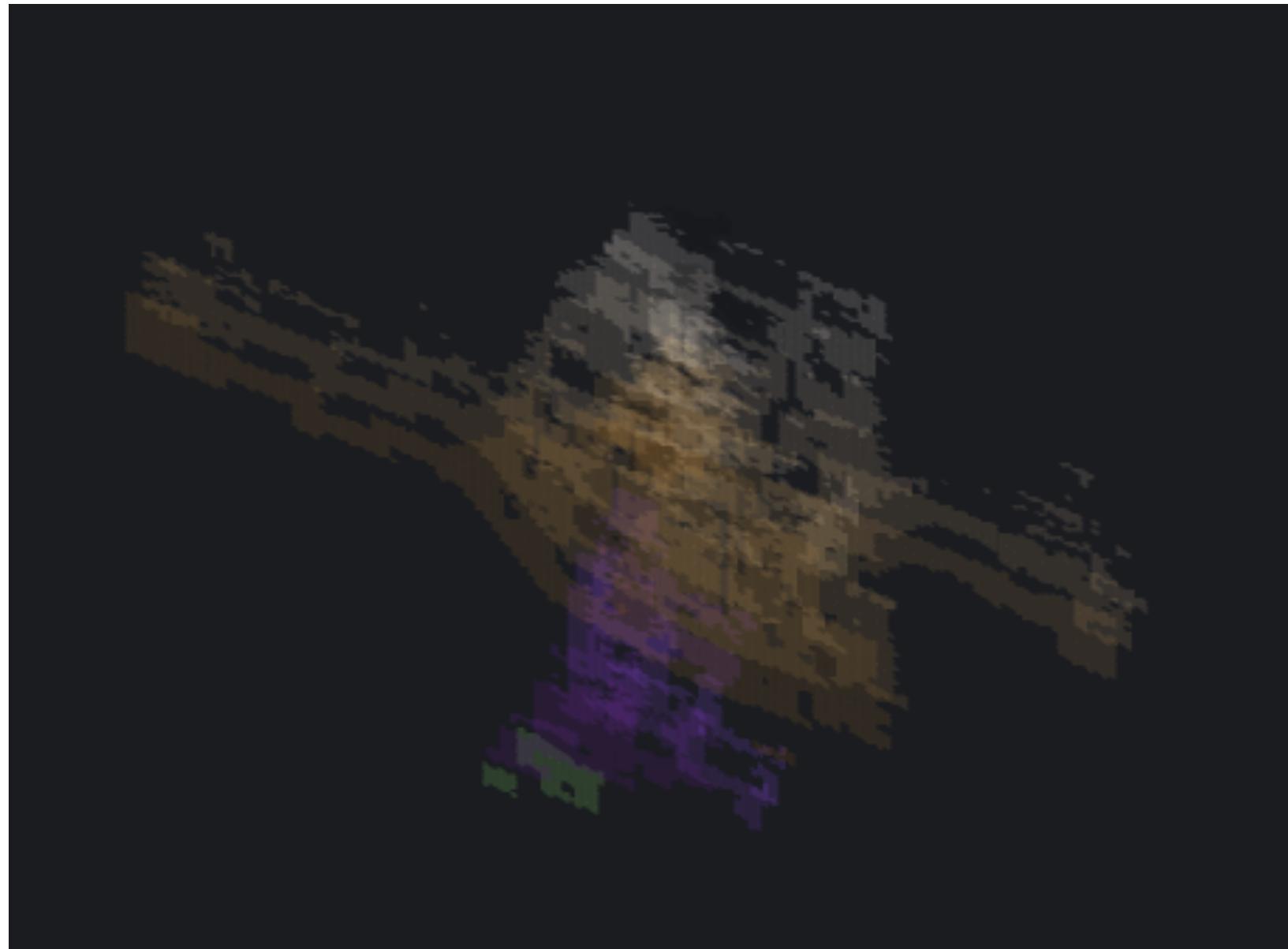
Sept. 2017 - Jun. 2019 Geoinformatics (Msc), Aalborg University Copenhagen

Sept. 2010 - Jun. 2017 Architecture (Msc), National Technical University of Athens

GIS Passions

Urban structures, space & development – Population Distribution – Socio-economic changes and causation – Spatial injustice – Urban regeneration & planning – 3D Visualizations – Routing services – WebGIS

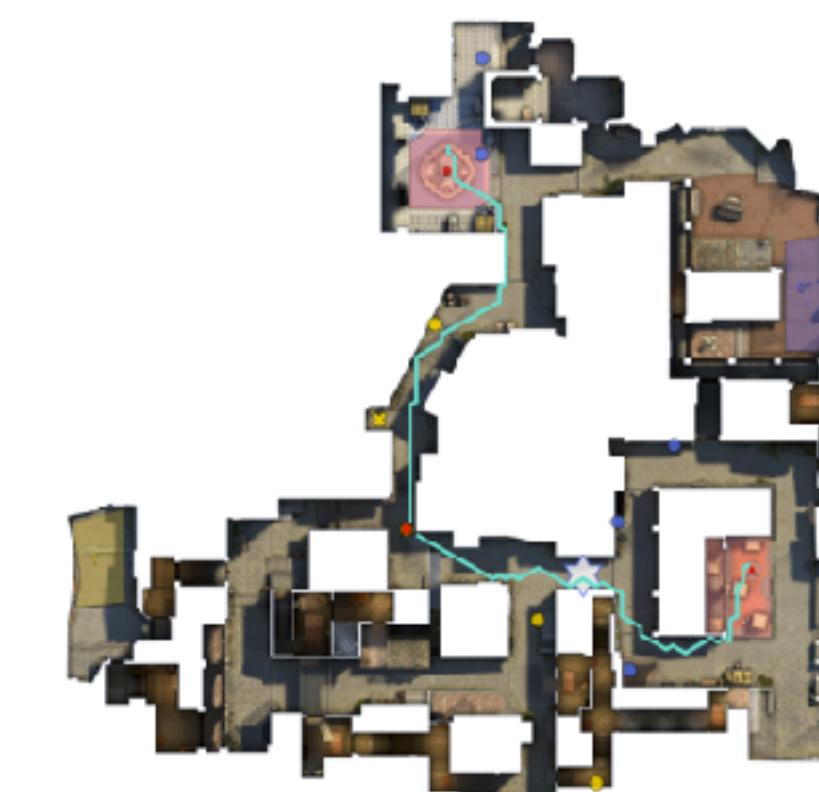
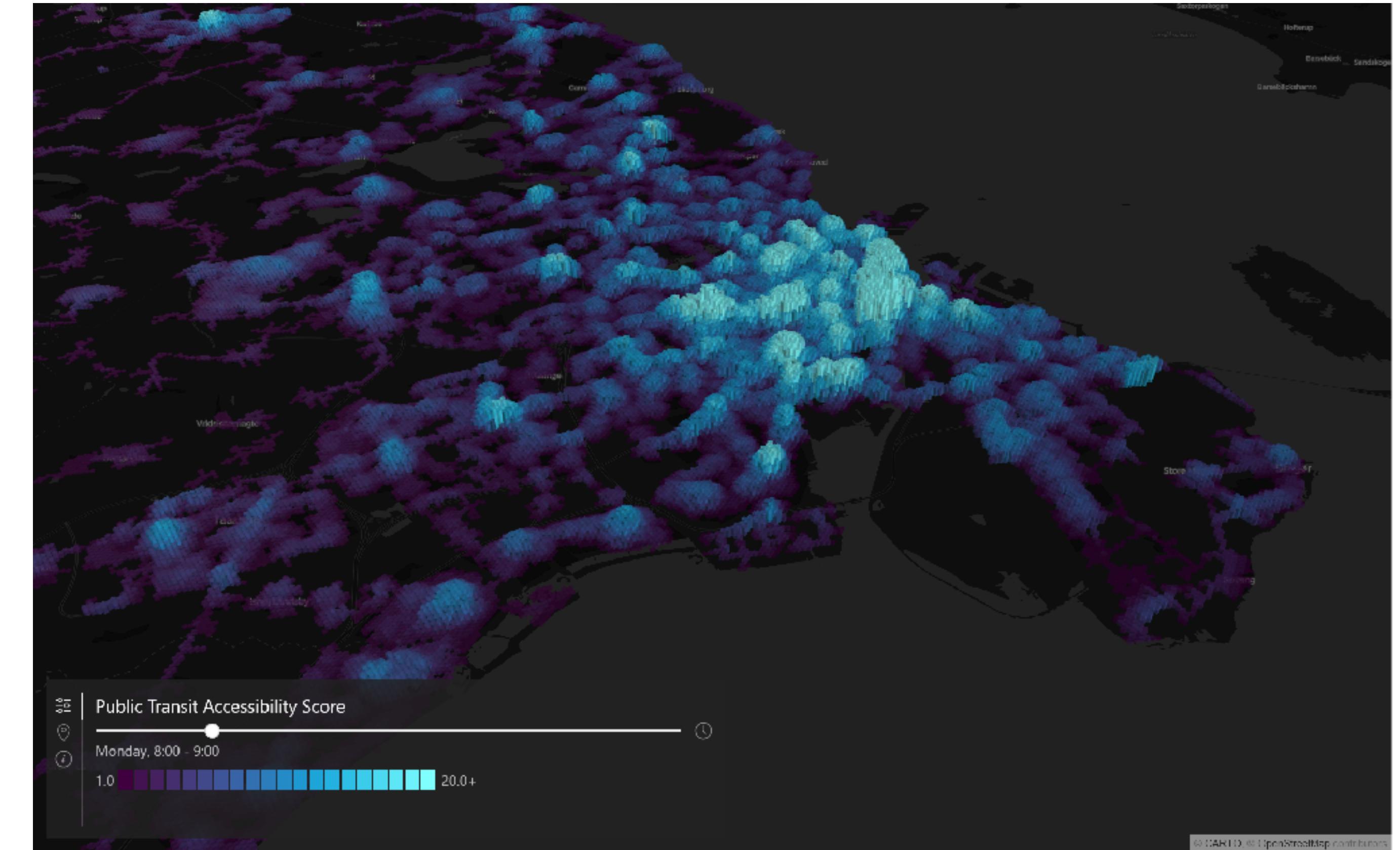
Jan Schelhaas



Geospatial project: Analyzing transit accessibility in Denmark

Research / Thesis: Multimodal routing to evaluate progress towards the 15-minute city

Work: Spatial DS in Counter-Strike and Valorant



@jan@datasci.social

jasch@itu.dk

What is Geospatial Data Science?

<https://www.menti.com/alh64wbaejxf>

www.menti.com 1813 7782

“Geospatial Data Science is the branch of data science, that encompasses **locational analytics, satellite imagery, remote sensing, analysis of projection systems, raster and vector data**.”

“Geography is the study of places and the relationships between people and their environments.

Geographers explore both the physical properties of Earth's surface and the human societies spread across it.”

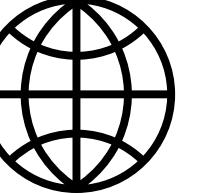
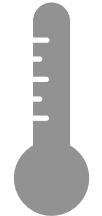
Geospatial Data Science is all the things that exist in ‘regular’ data science - but with a focus on **space** and **location**

What is Geospatial Data?

Geospatial data is information that describes objects, events or other features with a location on or near the surface of the earth.

What is Geospatial Data?

Geospatial data is information that describes objects, events or other features with a location on or near the surface of the earth.

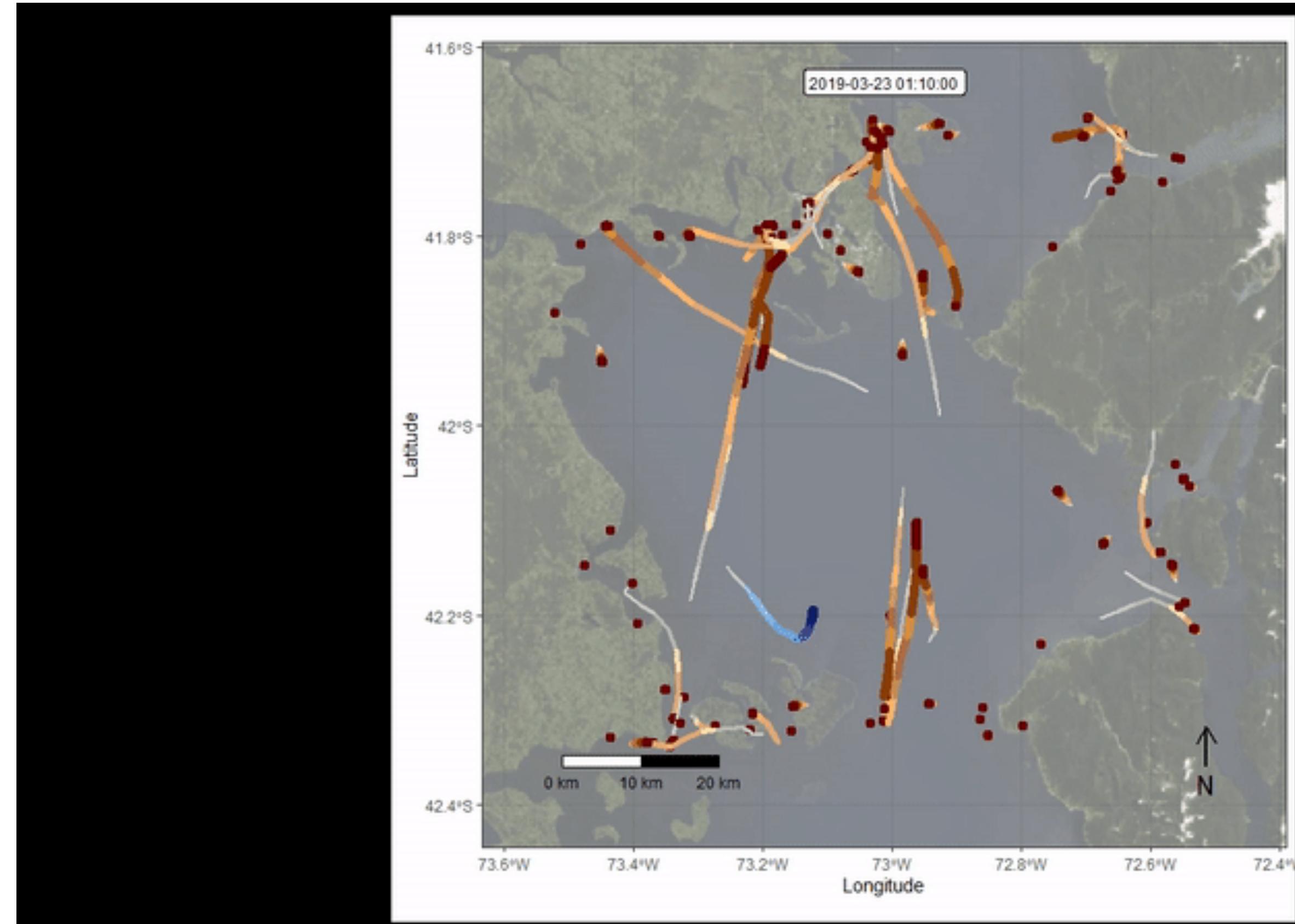
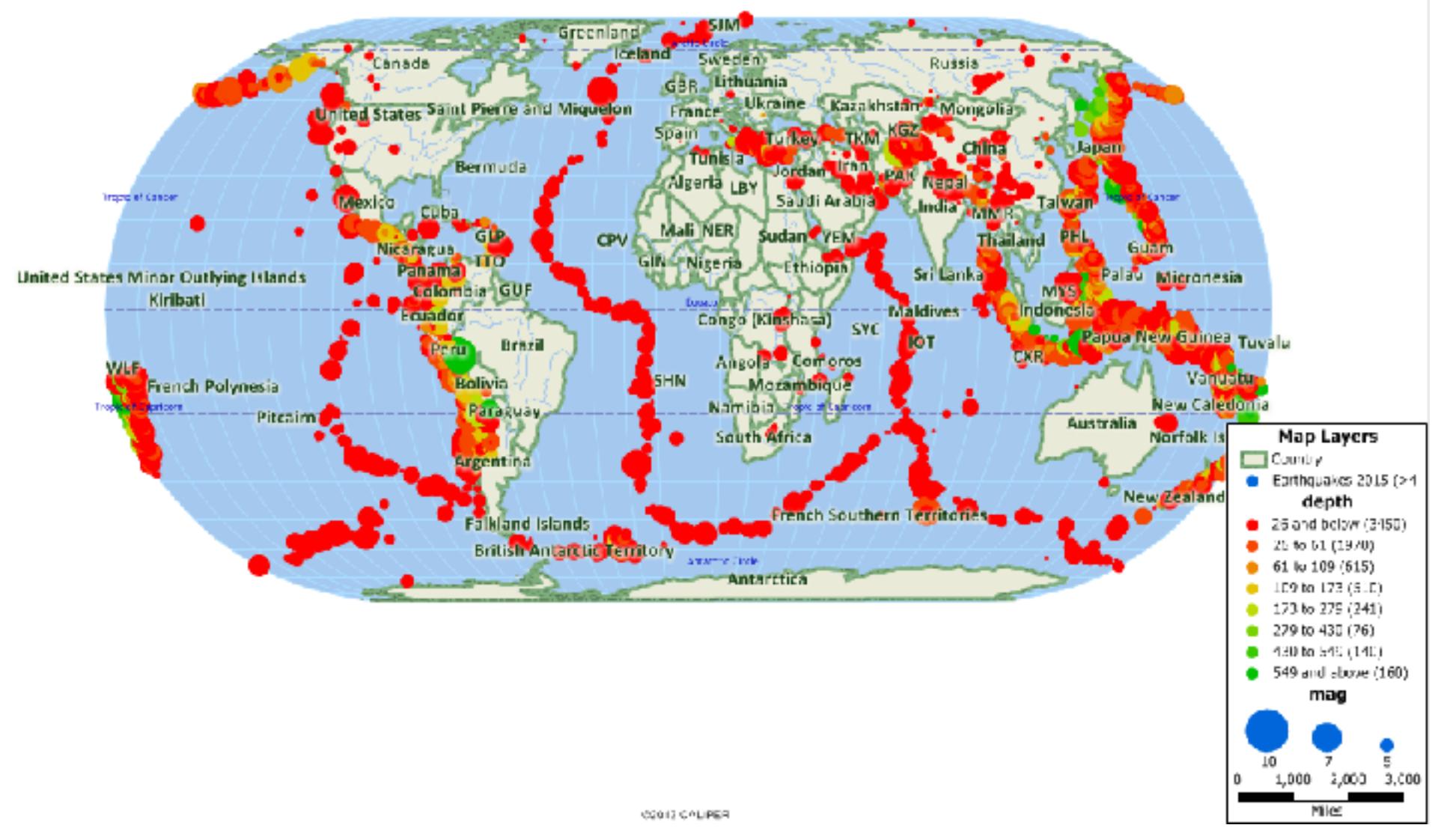
- Coordinates 
- Attributes 
- Temporal information 

“80% of data are geographic”



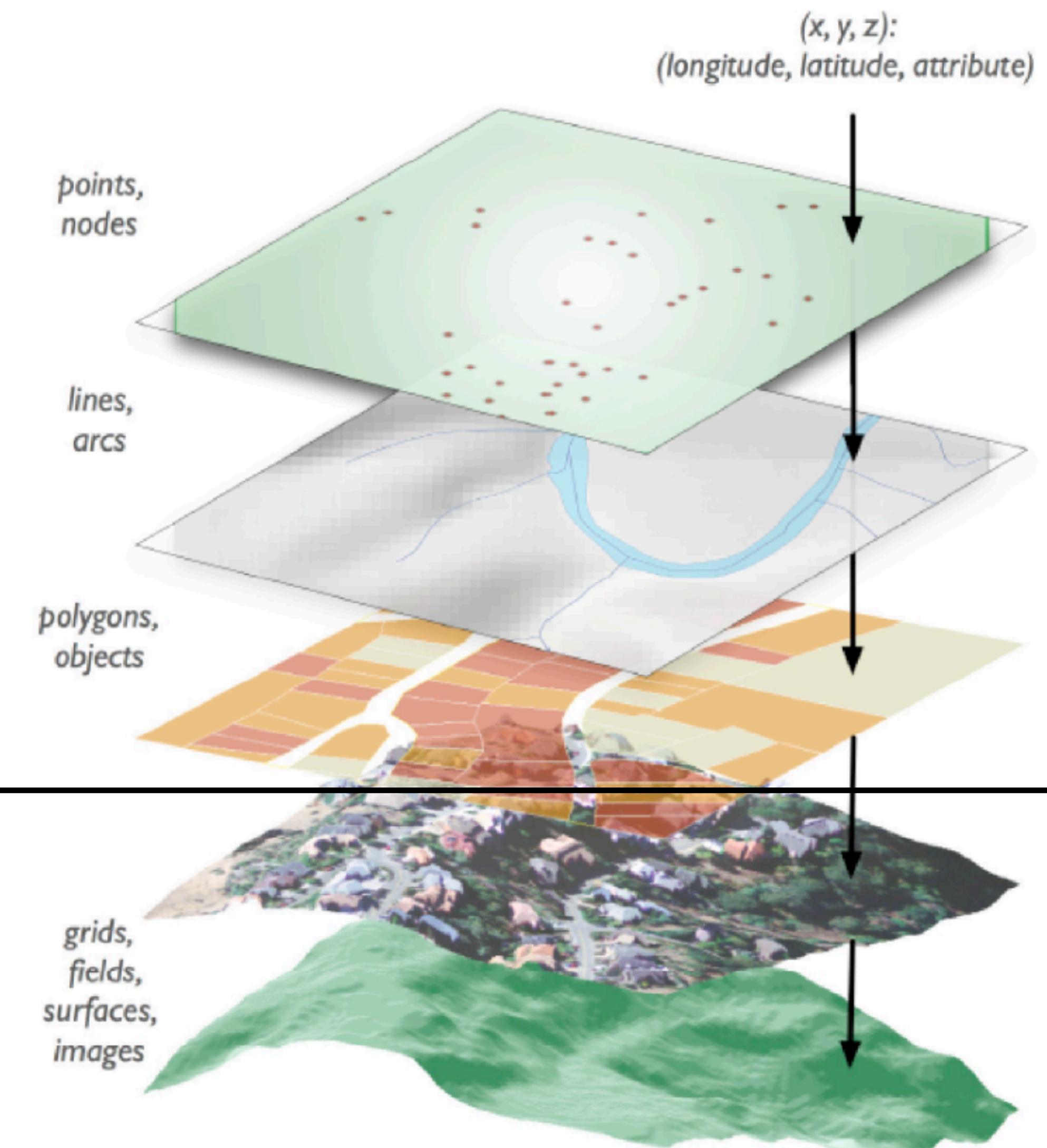
...maybe not, but almost everything has a **spatial component**

There is static and dynamic data



There is **vector** and **raster** data

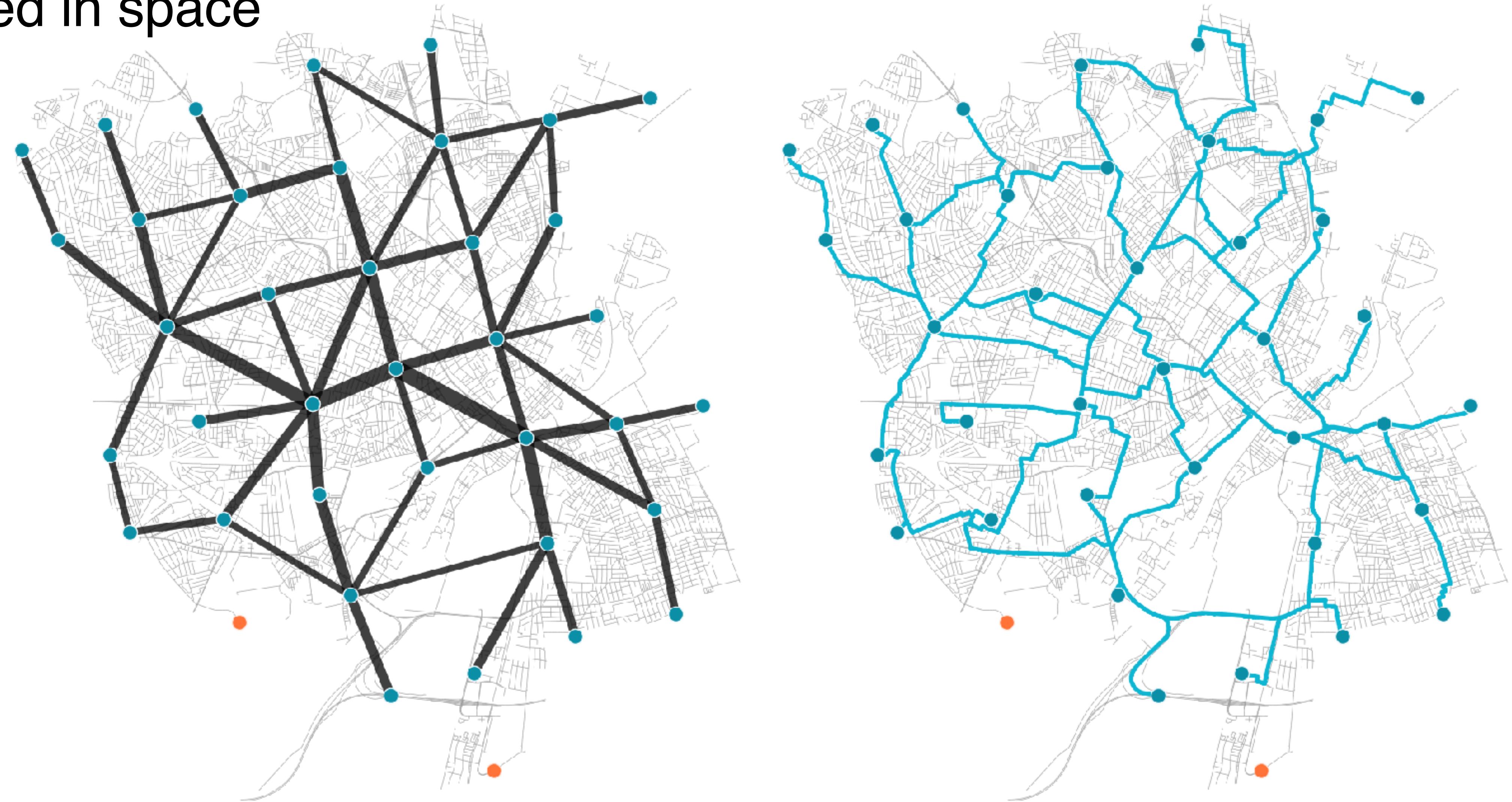
Vector: Geometric objects
.gpkg, .shp, .svg, geojson



Raster: Grid of pixels
.tif, .jpg, .png, .bmp

There is **network** data

Structure (topology),
embedded in space



GDS asks: How do things relate in space?

Everything is related to everything else,
but near things are more related than
distant things.

Tobler's 1st law of geography

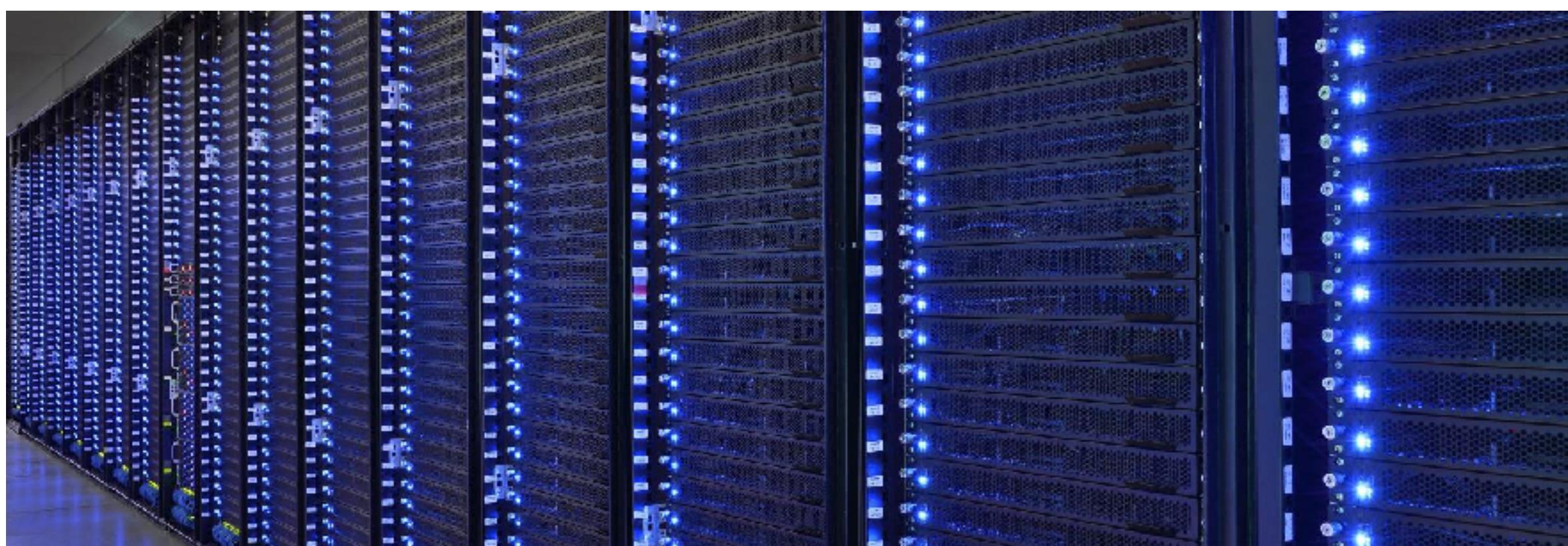
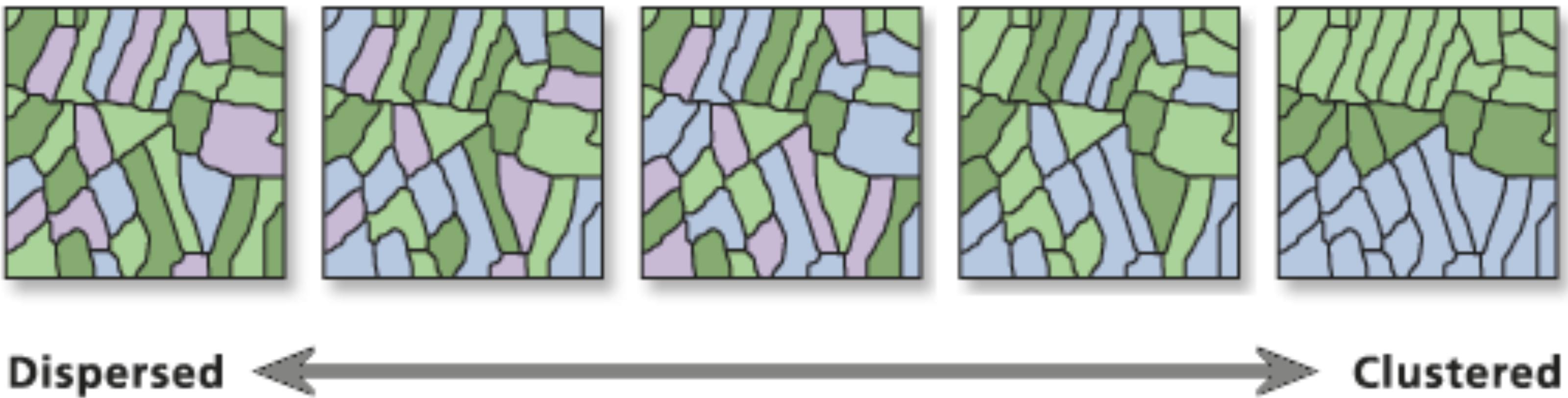
GDS asks: How do things relate in space?

How can we **formalize** this question?

How can we **operationalize** this question?

GDS is mathematical, visual, and computational

$$I = \frac{N}{W} \frac{\sum_i \sum_j w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2}$$



Geospatial Data Science stands on the shoulder of giants

	Most proponents come from	Since	Goal
Urban/regional planning	Urban planning, Architecture	3000 BC	City/Regional development
Human geography	Social sciences	18th century	Socio-cultural observation/critique
Geoinformatics	Geography, Computer Science	1980s	Spatial analysis
Urban computing	Computer science	2000s	Urban Big Data and technology analysis
Complex spatial systems	Physics	2000s	Parsimonious models
Geospatial data science	Geography, Computer science, Geoinformatics, Physics	2010s	Quantitative understanding

This course will teach you:

- Data structures and principles of GDS
- Gathering and preprocessing large-scale geospatial data
- State-of-the-art computational tools for GDS
- Spatial network analysis
- Real world applications of these techniques in an applied context

We use the Geographic Data Science platform gds_env

The screenshot shows the homepage of the gds_env website. At the top left is the gds_env logo. To its right is a search bar with the placeholder "Search gds_env". Further to the right is a link to "gds_env on GitHub". On the far left is a vertical navigation menu with links: Home (which is highlighted in light blue), Stacks, Guides, Contributing, and FAQ. The main content area features the text "A containerised platform for Geographic Data Science: gds_env". Below this text is a "launch binder" button, which includes a small icon of a computer monitor with a circular arrow. At the bottom of the content area is a list item: "• Dani Arribas-Bel [[@darribas](#)]".



https://darribas.org/gds_env/

Course overview

Mondays 8.15 - 12

Lecture 08:15 - 10:00

Break 10:00 - 10:15

Exercise 10:15 - 12:00

Room: 3A52

Course overview

Mondays 8.15 - 12

Lecture 08:15 - 10:00

Break 10:00 - 10:15

Exercise 10:15 - 12:00

Room: 3A52

Important dates

Deadline for project proposals:
March 31

Deadline for project approvals:
April 11

Exam hand-in:
May 26

Course overview

Number	Date	Topic	Lecturer	Exercise
1	Jan 30	Intro	Ane	Env install
2	Feb 6	Spatial Data: Vector	Ane	Vector
3	Feb 13	Choropleth	Ane	Choropleth
4	Feb 20	Spatial Weights	Ane	Weights
5	Feb 27	Spatial Autocorrelation	Ane	Autocorrelation
6	March 6	Spatial Clustering	Ane	Clustering
7	March 13	Point Pattern Analysis	Marina	Point Patterns
8	March 20	OpenStreetMap	Ane	OSM
9	March 27	Spatial Networks	Ane	Networks
10	April 3	Spatial Data: Raster	Marina	Raster
11	April 17	GeoAI	Marina	GeoAI
12	April 24	Mobility	Ane	Mobility
13	May 1	Big Spatial Data	Ane	Project Questions
14	May 8	Spatial Data Viz	Ane	Project Questions

After this course you will be able to:

- Use Python libraries programmatically for geospatial data analysis.
- Reflect on the motivation and inner workings of the main methodological approaches of GDS
- Critically evaluate the suitability of a specific GDS technique, what it can offer and how it can help answer questions of interest.
- Explain how to interpret results, in a process of turning data into insights.

In case of issues, contact:

Student Affairs and Programmes (SAP) sap@itu.dk

Ane: anev@itu.dk (No office hours, but always happy to set up a meeting!)

Marina: marinag@plan.aau.dk

Use Q&A forum on LearnIT

Anonymous feedback or questions? → Link to feedback form on LearnIT

We can only cover a small part of all tools

The typical GDS workflow:

Set Up Your Environment

- Virtual machine environments (Linux-based)
- Spatial databases (PostgreSQL/PostGIS) with multi-user editing and versioning (GeoGig)

Wrangle Data

- APIs (Google Maps, OpenStreetMap)
- Modern data formats and tools (GeoJSON, GDAL)

Analyze Data

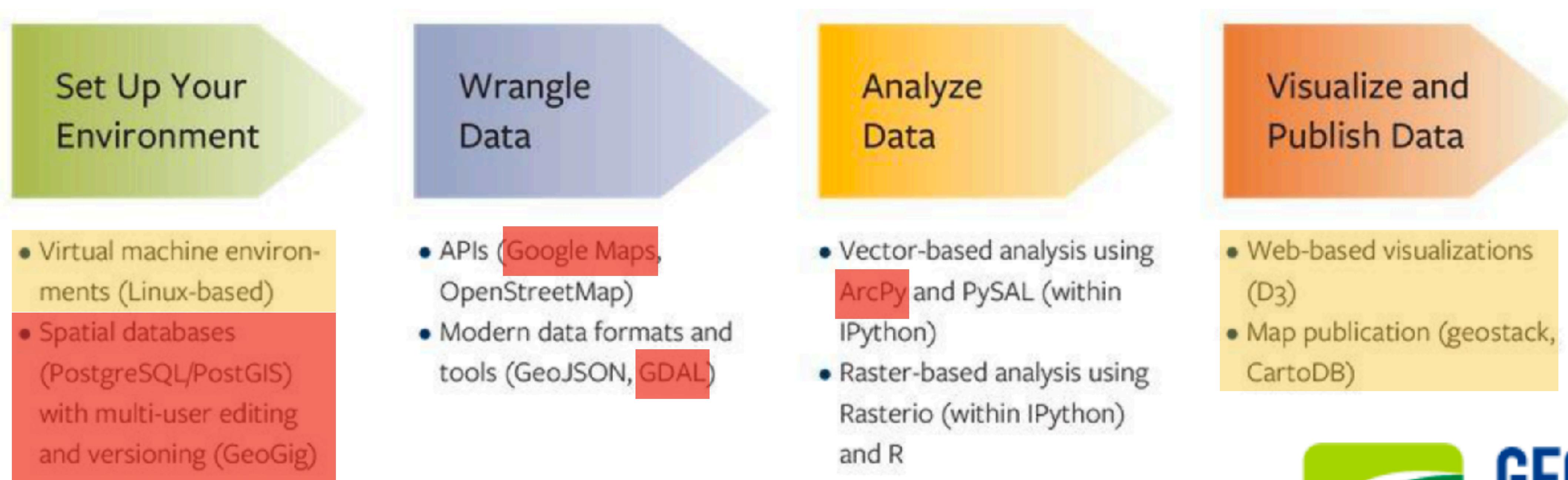
- Vector-based analysis using ArcPy and PySAL (within IPython)
- Raster-based analysis using Rasterio (within IPython) and R

Visualize and Publish Data

- Web-based visualizations (D3)
- Map publication (geostack, CartoDB)

We can only cover a small part of all tools

The typical GDS workflow:



Cutting-Edge Mapping Technology at UC Berkeley

We will only cover the procedural part of GDS (in Python)

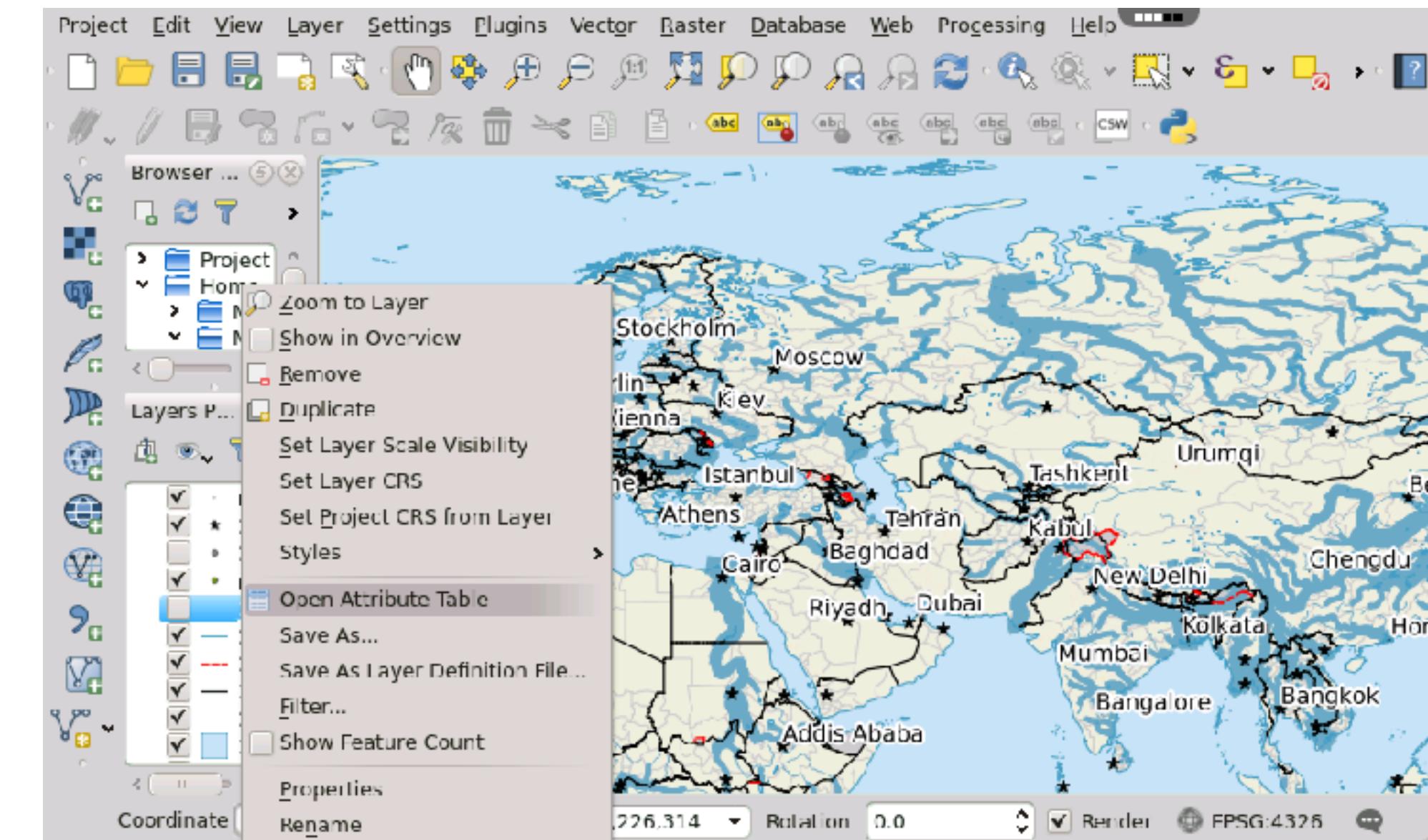
DB handling



Procedural

```
31     def __init__(self, file=None, fingerprints=None, logduplicates=False, debug=False, logger=logging.getLogger(__name__)):
32         self.file = file
33         self.fingerprints = set()
34         self.logduplicates = logduplicates
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(path, 'ab')
39             self.file.seek(0)
40             self.fingerprints.update(fingerprint for fingerprint in self.file)
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('supersecret.debug')
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

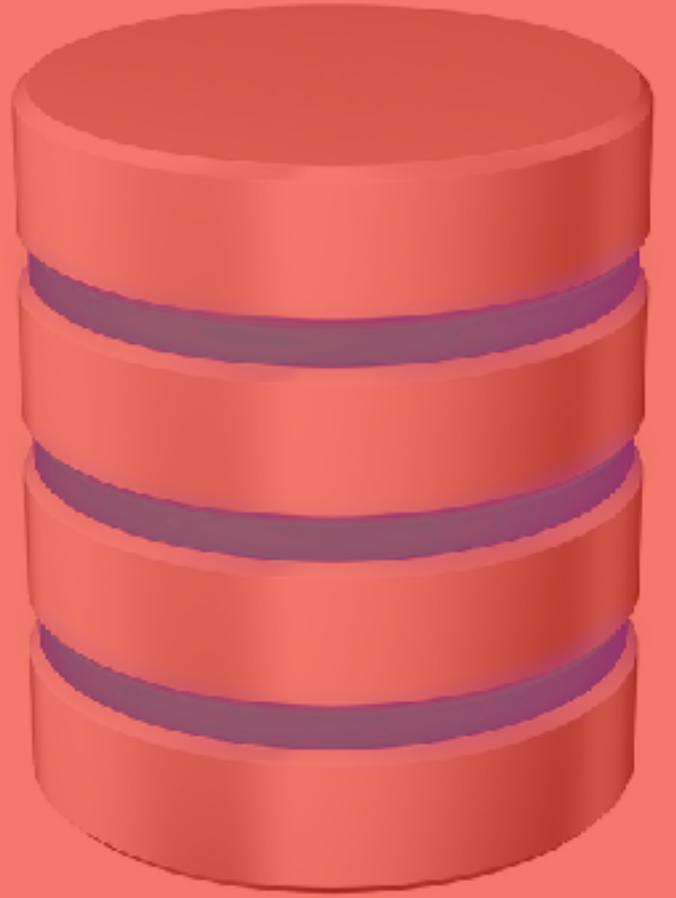
Point & Click GIS



ArcGIS

We will only cover the procedural part of GDS (in Python)

DB handling

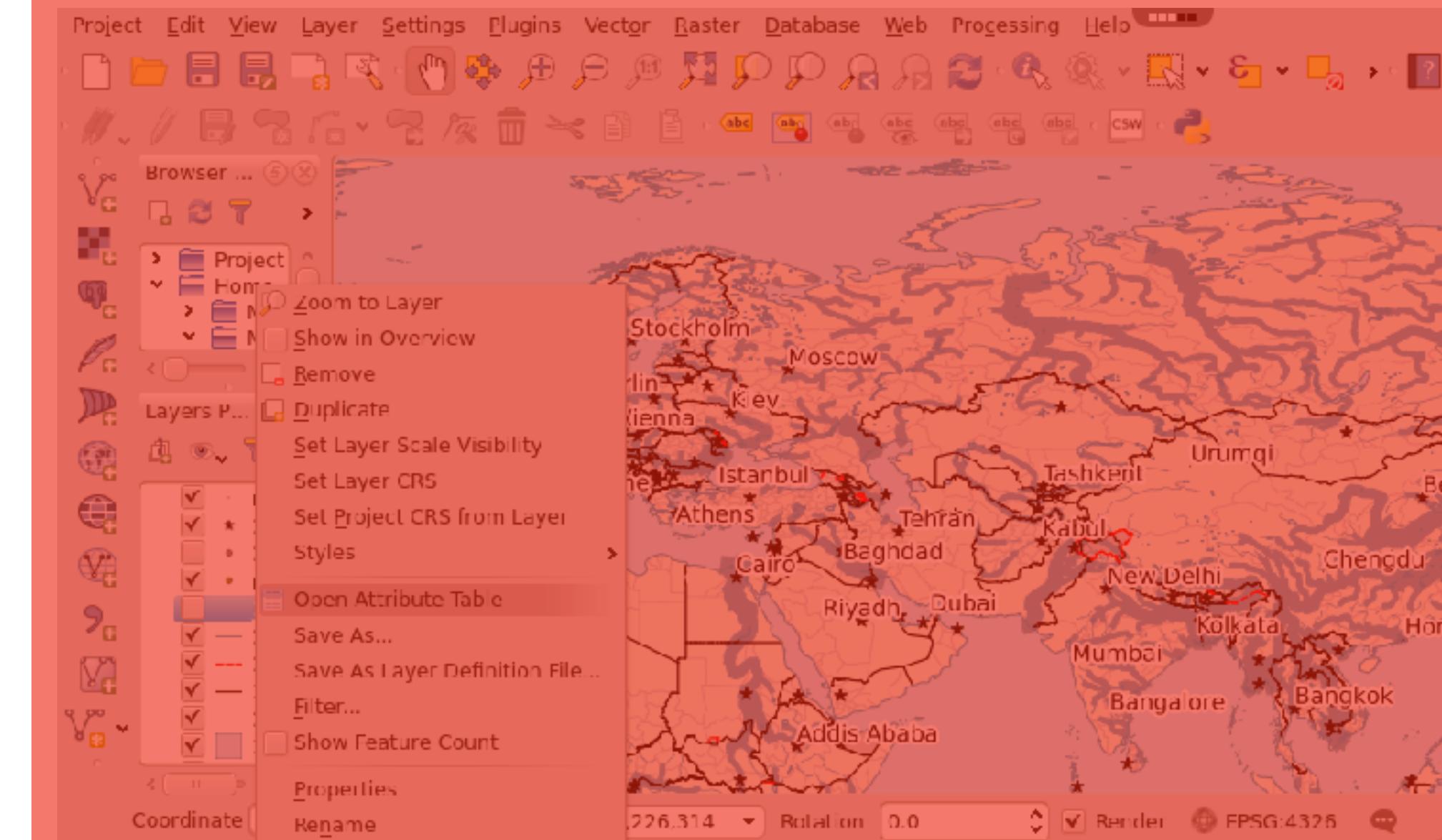


Procedural

```
31     def __init__(self, settings):
32         self.file = None
33         self.fingerprints = set()
34         self.logduplicates = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(path, 'ab')
39             self.file.seek(0)
40             self.fingerprints.update(self._read_file())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('superuser.debug')
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```



Point & Click GIS

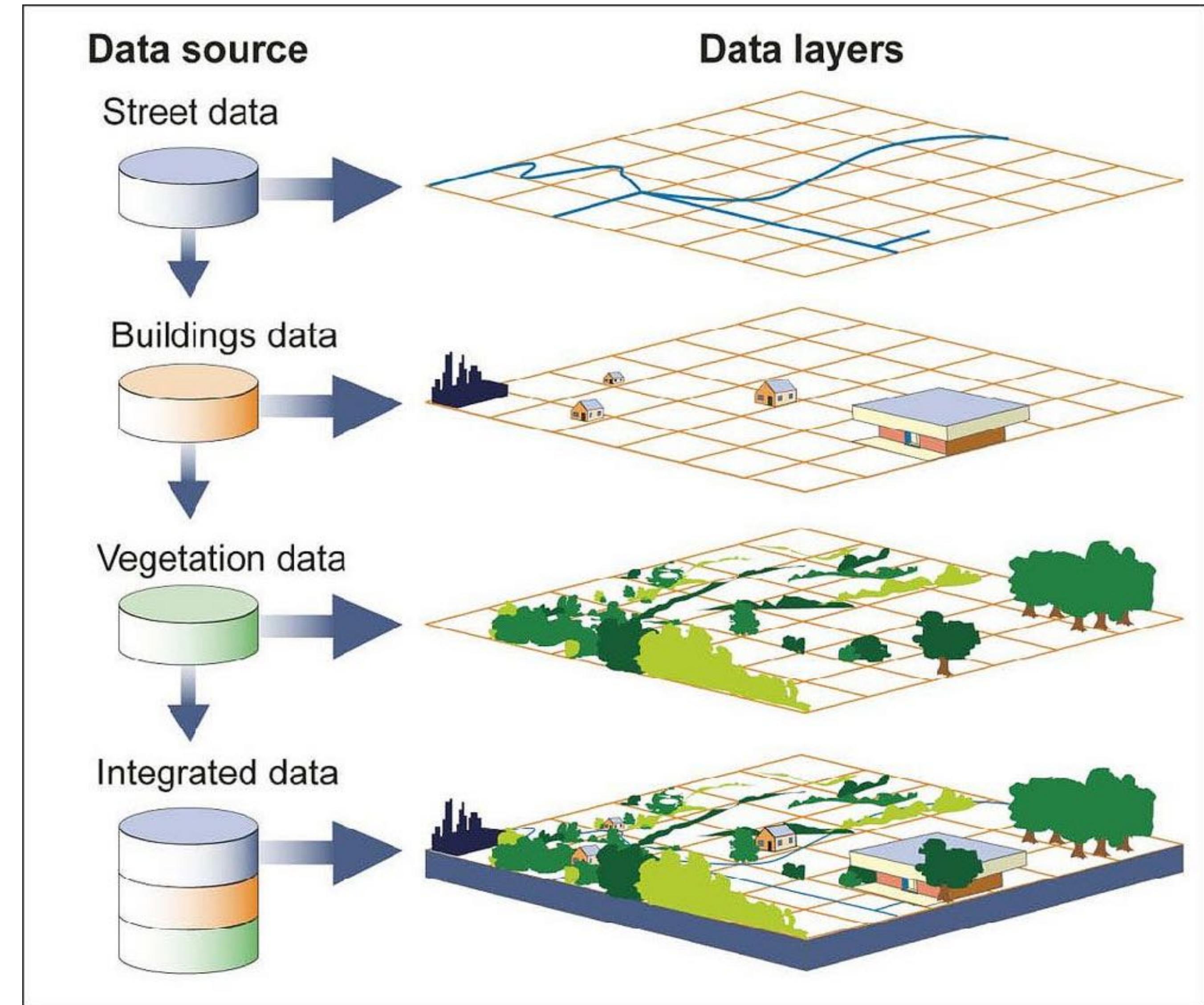


ArcGIS

A **Geographic information system (GIS)** is a system for capturing, storing, checking, and displaying geospatial data.

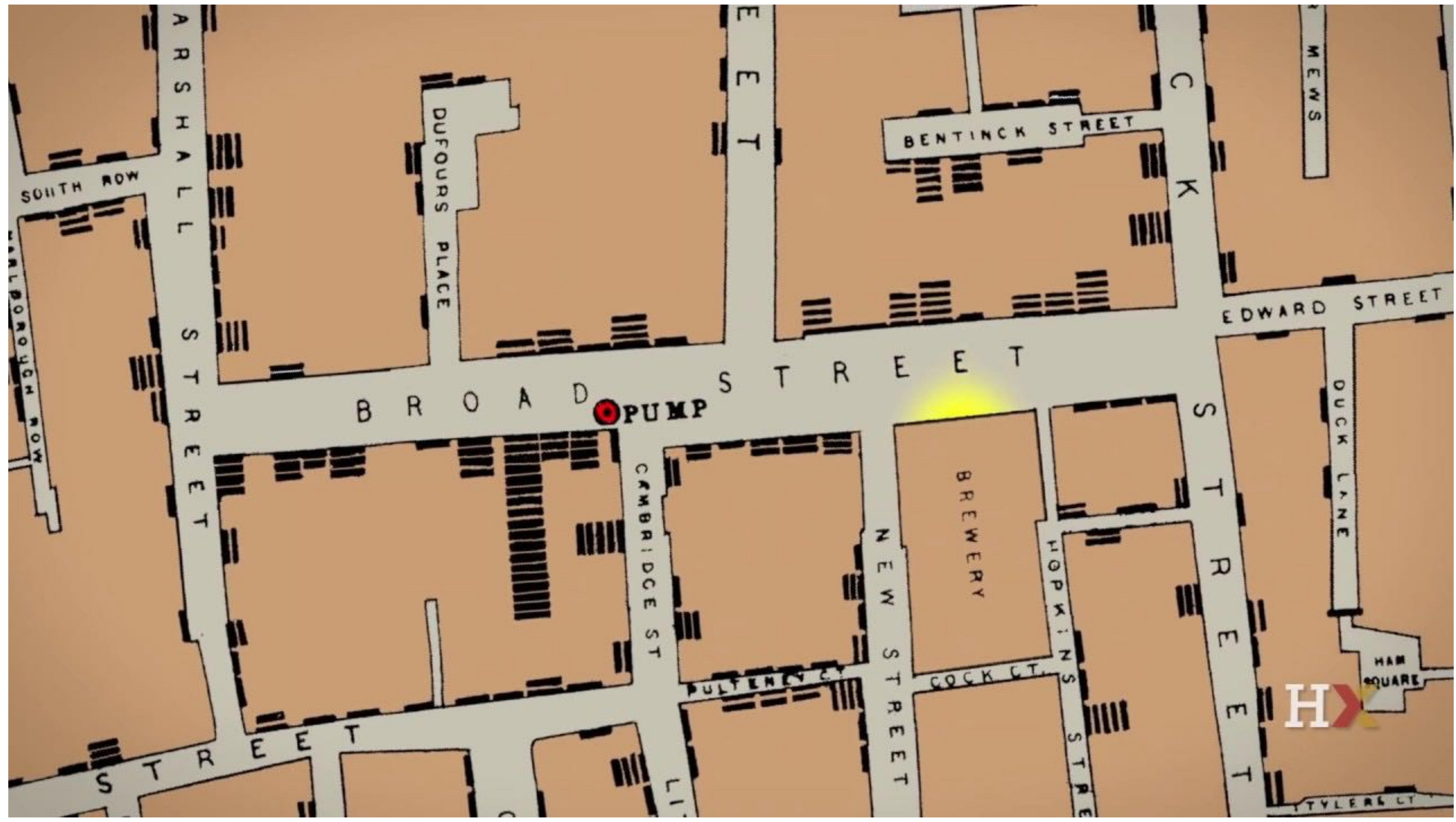
Can refer to:

- concrete software
- abstract concept



Source: GAO.

Who is John Snow?



Jupyter