The ingredients for a geographical modelling project

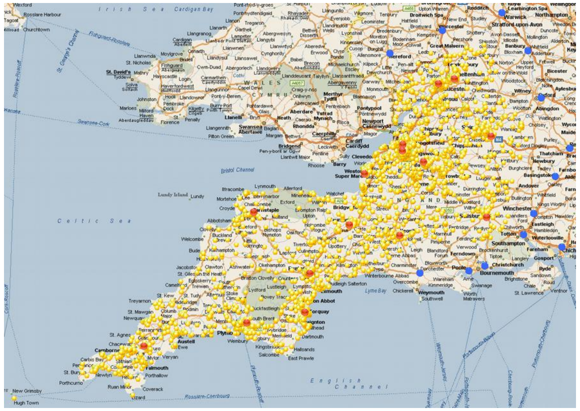
1) **Scope**: Specify the scope of the problem (national, regional, local)
2) **Data**: Location of population, location of services, admissions
3) **Calculate**: Travel time between each population location and service location
4) **Outputs**: Performance criteria. Any fixed contraints?
5) **Scenarios**: Which options to explore (a specific number of service locations, or explore the trade-offs)
6) **Model**: Algorithm to find good solution/s
7) **Communicate findings:** Based on scenarios (show one solution, a range of solutions, or raise awareness of the trade-offs between conflicting performance criteria?)
8) **Visualisation:** People do love a good map.
We use an open source mapping software, QGIS.

1. **Scope: Specify the scope of the problem (national, regional, local)**

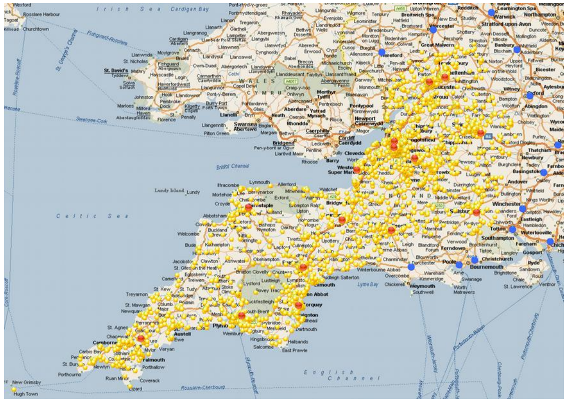
This will be determined by the problem owner before any modelling work starts. Be clear that this is <u>not</u> a flexible parameter to change at a later date.

This defines the modelled geographical boundary & is a surprisingly important step.

- Which service locations are in the scope
- Which service locations out of scope are offering the same service
- Which population are included in the model
- Do all population currently served within scope need to be served by an in scope service in the future? Or can they cross the boundary (impact on another region)



2. **Data**



1. Service locations
In-scope (red dots): provide the service within the modelled region.
Out-of-scope (blue dots): provide the service outside of the modelled region.

2. Population locations (yellow dots)
Those that live closest to an in-scope service location while all service locations (in & out of scope) are open.
Per LSOA, postcode sector...

3. Admissions to each service
The owner of the project problem may own this data.
Or is it collected and held regionally/nationally?
Model based on population (demographic data as an estimate)

3. **Calculate: Travel time between each patient and service location**

Method 1: Straight line distance calculations

This takes the distance between two points as being the distance of the shortest theoretical path (straight line) between them.

+ Very easy to calculate
- Straight line paths are not (usually) practical travel routes

3. **Calculate: Travel time between each patient and service location**

Method 2: Straight line distance converted to road travel time

Use a polynomial equation to estimate the average speed from the straight line distance between two points.

Convert straight line to time travelled using the average speed

+ Easy to calculate
+ More realistic than straight line
- More realistic, but still includes some simplification

Speed = $25.9364 + 0.740692\,d - 0.00537274\,d^2 + 1.9121\text{e-}05\,d^3 - 3.19161\text{e-}08\,d^4 + 1.99508\text{e-}11\,d^5$

Using d = 38.4 miles

Average speed = 47.5 mph

Travel time = 48 minutes

3. **Calculate: Travel time between each patient and service location**

Method 3: Route travel distance calculations

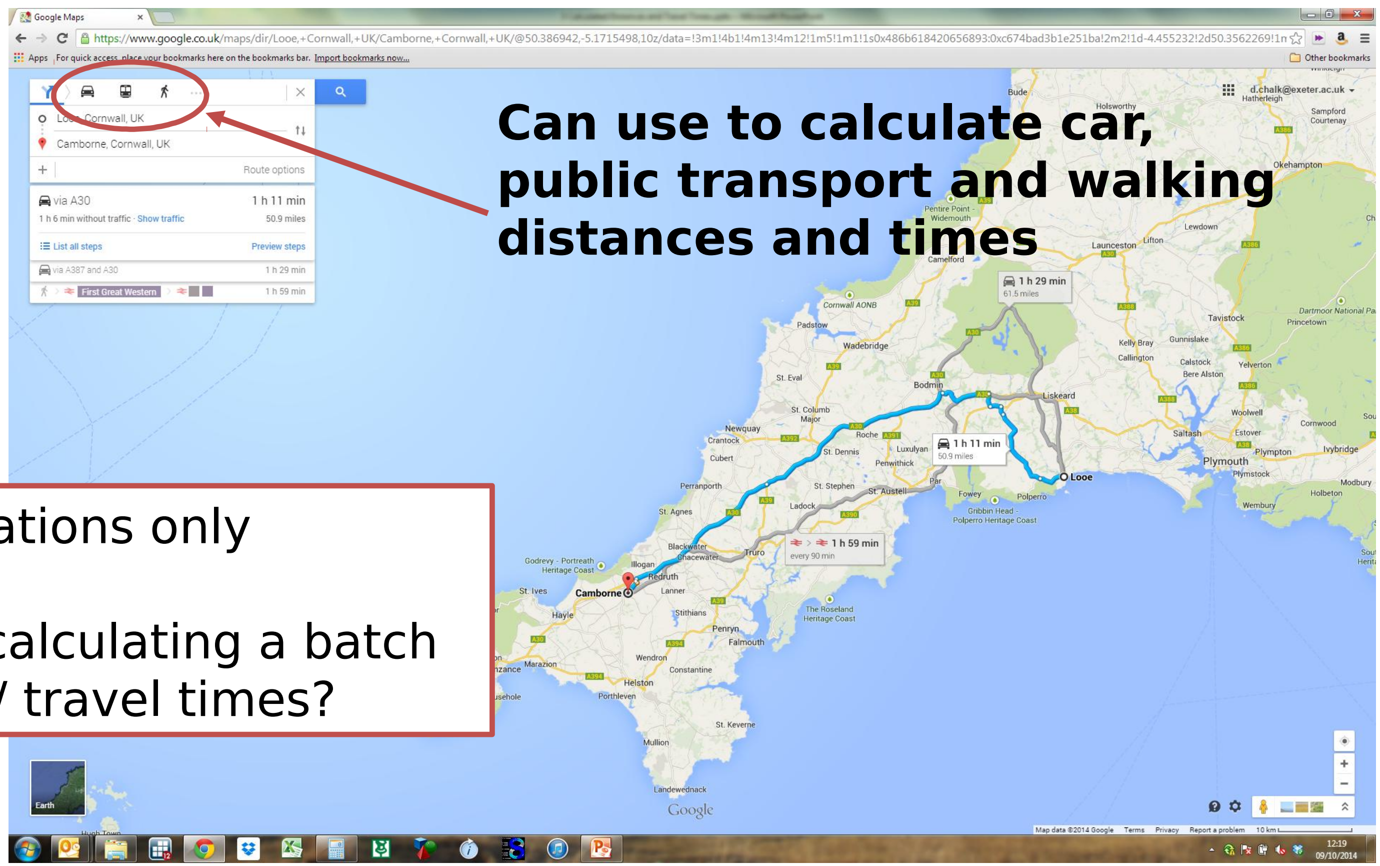
This calculates the shortest travel between two points taking only traversable paths (such as roads).

+ More realistic
+ Software available (Routino)
- Harder (and longer) to calculate

3. **Calculate: Travel time between each patient and service location**

Software and Services to Calculate Distance and Travel Times

Google Maps (http://maps.google.co.uk)



Can use to calculate car, public transport and walking distances and times

Single calculations only

What about calculating a batch of distances / travel times?

3. **Calculate: Travel time between each patient and service location**

Paid for option:
Maptitude for mappoint

Open source option:
Mike’s adapted an existing routine called Routino.
Needs Linux or Windows10 (with autumn 2017 update)

4. **Outputs: The performance criteria. Any fixed constraints?**

Involve the problem owner.

Examples of performance criteria

- Number of service locations: MINIMISE
- Average travel time: MINIMISE
  → Clinical benefit: MAXIMISE
- Maximum travel time: MINIMISE
- Maximum admissions to any one service location: MINIMISE
- Minimum admissions to any one service location: MAXIMISE
- Max/Min admissions ratio: MINIMISE
- Proportion population travelling within 80 mins: MAXIMISE
- Proportion of service locations that have >1,000 admission: MAXIMISE
- Proportion of population within 30 mins: MAXIMISE
- Proportion of population attending service location with > 1,000 admissions: MAXIMISE
  Or combine two into one:
- Proportion of population within 30 mins AND >1,000 admissions: MAXIMISE

… If too many criteria, check out https://en.wikipedia.org/wiki/Dimensionality_reduction

Examples of fixed constraints

- All service locations need > 1,000 admissions
- Population not travel further than 1 hour

5. **Model: Algorithm to find good solution/s**

These are mathematical optimisational techniques to find good solutions (“high points on the fitness landscape”).

Choice depends on problem type & number of locations.

Facility location	**Travelling salesman problem**	**Shortest path**
Brute force	Brute force	Brute force
Greedy	Hill climbing	Dijkstras
Hill climbing	Genetic algorithm	A*
Genetic algorithm	Simulated annealling	

6. **Scenarios: Which options to explore**

Informed by the problem holder.

Are they looking to know how many to open/close (in order to meet a criteria)

Or are they looking to know which 1, 2 or 3 to open or close

7. **Communicate findings**

Distil your findings and communicate the important points based on the project aims (present recommended configurations, or performance trade-offs to inform discussions).

Use percentiles as well as min, ave, max.

We’ve always been asked for the range of admissions to each service location and where the most affected population live.

Maps are a great way to display a lot of geographical information...

8. **Visualisation**

QGIS (mapping software)