Learning about *walkability* in Wellington with Python and open data

Shrividya Ravi (Shriv)

What can you expect?

The slides contain

- An introduction to doing spatial data science
- A smattering of Python code
- Lots of graphs and figures
- Some insights about walkability in Wellington

Motivation

The global open data movement

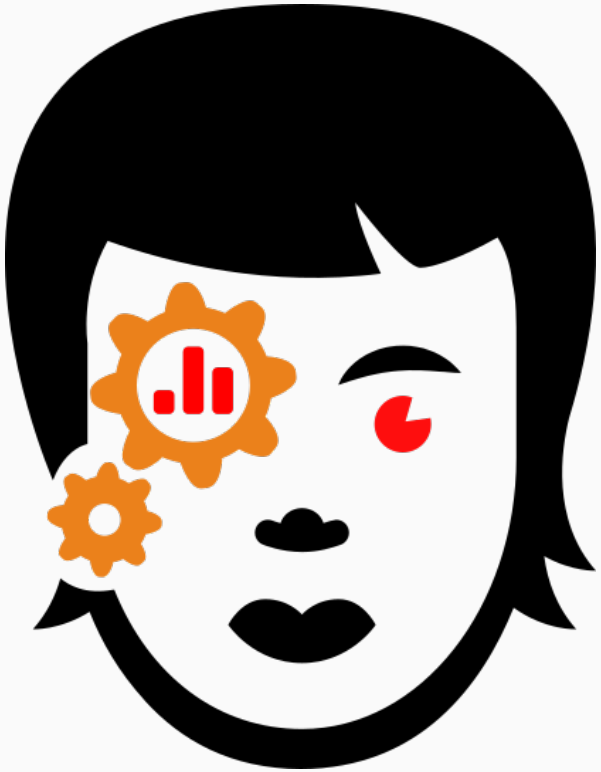
- Data sources
  - OpenStreetMap
  - Data.gov*: data.gov, data.govt.nz, data.gov.uk, data.gov.in
- Inspiring people generating tools / insights with open data
  - Kuan Butts
  - Geoff Boeing

Limited analysis of NZ

I want to learn about the coolest little capital in the world

- How are we being served?
  - housing
  - public transport
  - amenities
  - infrastructure
- Can things be better?

.. but I'm still just a Data Scientist



Audiens Cave

Walkability

Importance

Reducing car reliance and encouraging more transport-related physical activity are now recognised as beneficial objectives from health, social and environmental perspectives. Evidence is accumulating that a number of built environment attributes are associated with the likelihood of residents using active transport.

– Measuring neighbourhood walkability in NZ cities

Need

Unlike cars, pedestrians are sensitive to their environment; changes to it can impact the *walking experience* or the *decision to walk*.

Task

We'll explore the impact of hilly terrain on walkability Specifically, on walkability to council playgrounds - an amenity that should be locally accessible on foot.

Python package set up

Main packages

```
# Geoprocessing
import osmnx as ox                          # Processing OSM as graphs
import networkx as nx                       # Graph structure processing
import pandana as pa                        # Efficient accessibility computing
import geopandas                            # Processing geodataframes
from shapely.geometry import \
Point, Polygon, LineString                  # Core geometric objects
```

Supporting packages

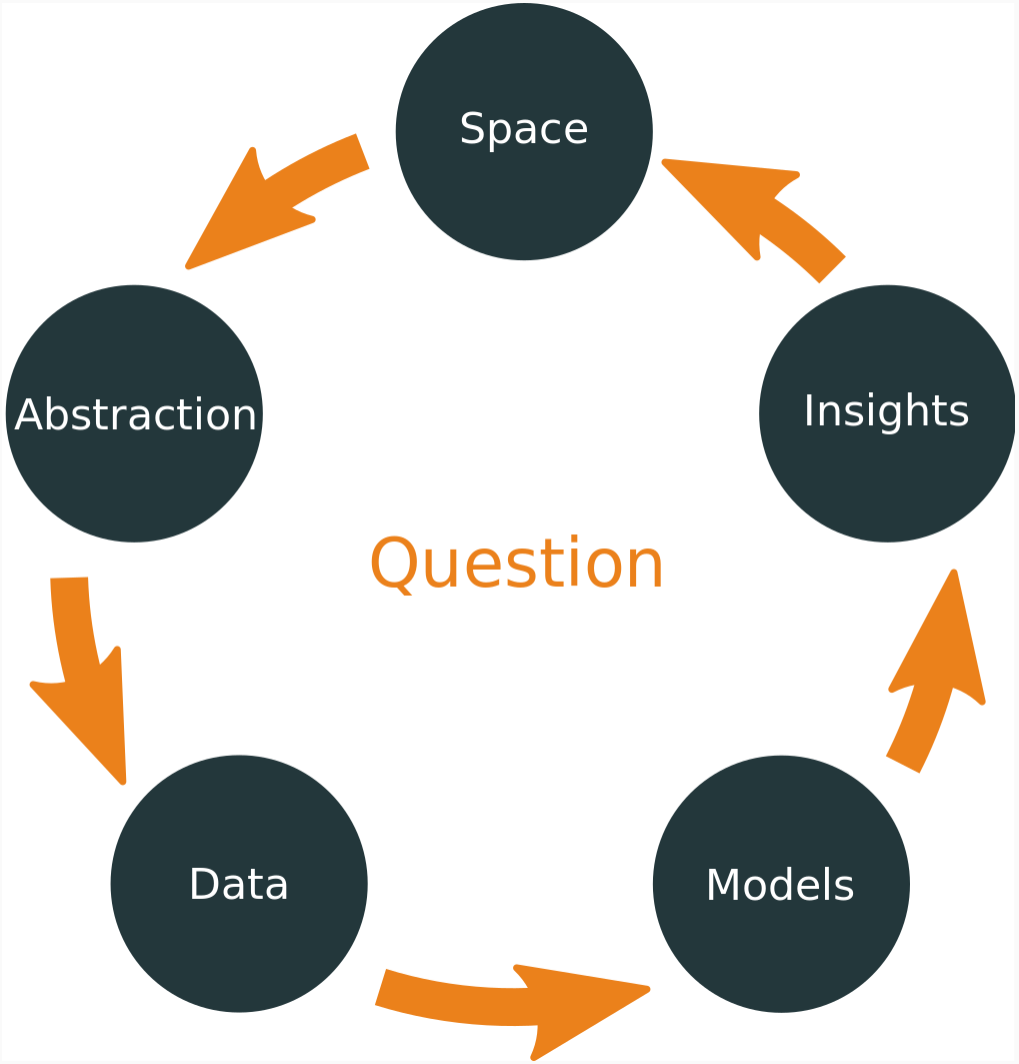
```
# Plotting
import matplotlib.pyplot as plt        # Classic Python plotting package
import seaborn as sns                  # Ggplot2-like plotting in Python
import folium                          # Interactive, web-ready maps
from descartes import PolygonPatch     # Geometric patches for matplotlib

# General utilities
import yaml                            # Reading stored API keys
import numpy as np                     # Processing arrays and matrices
import pandas as import pd             # Processing dataframes

# Bayesian analysis
import pystan                          # Running Bayesian models
```

Spatial data science

Overview



Question

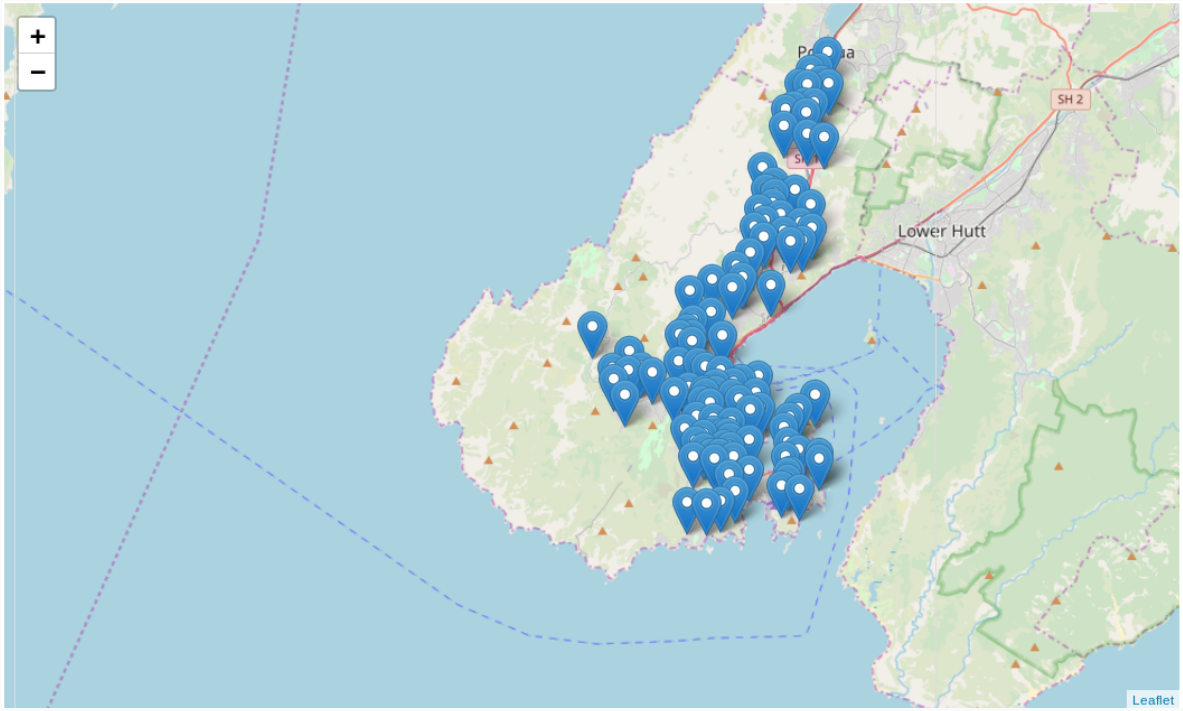
What is the impact of hills on walkability to playgrounds in Wellington?

Space

Wellington

Abstraction: spatial primitives

Points

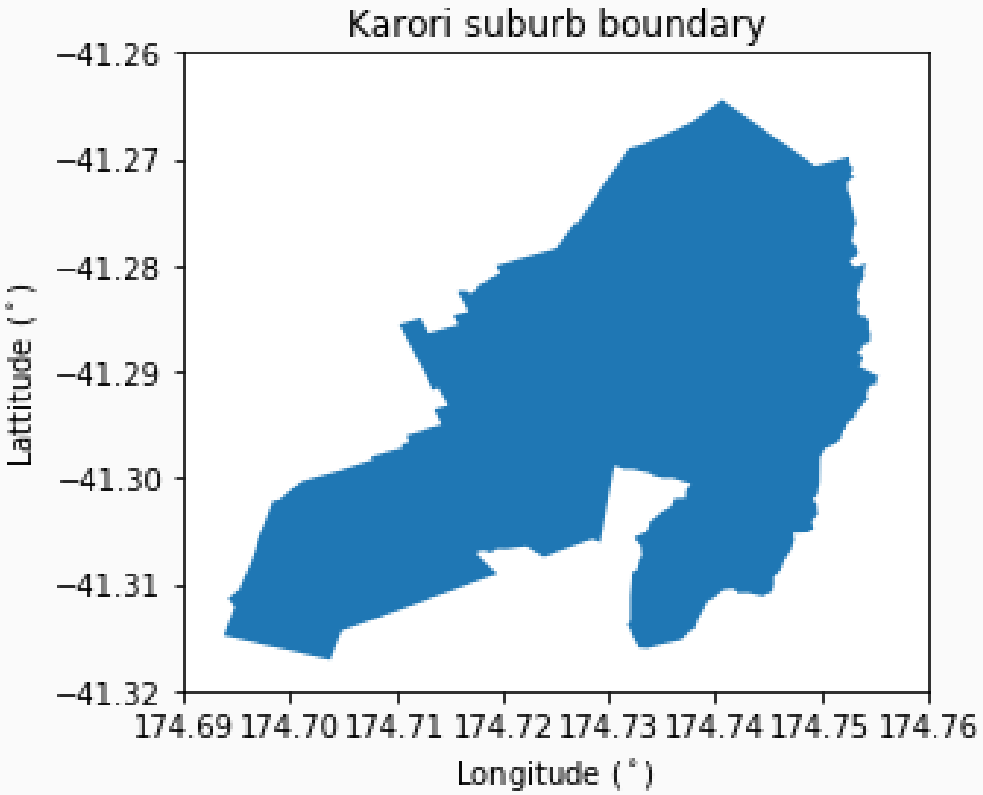


- Point coordinates of playgrounds
- *Overlaid* on map of Wellington

Lines

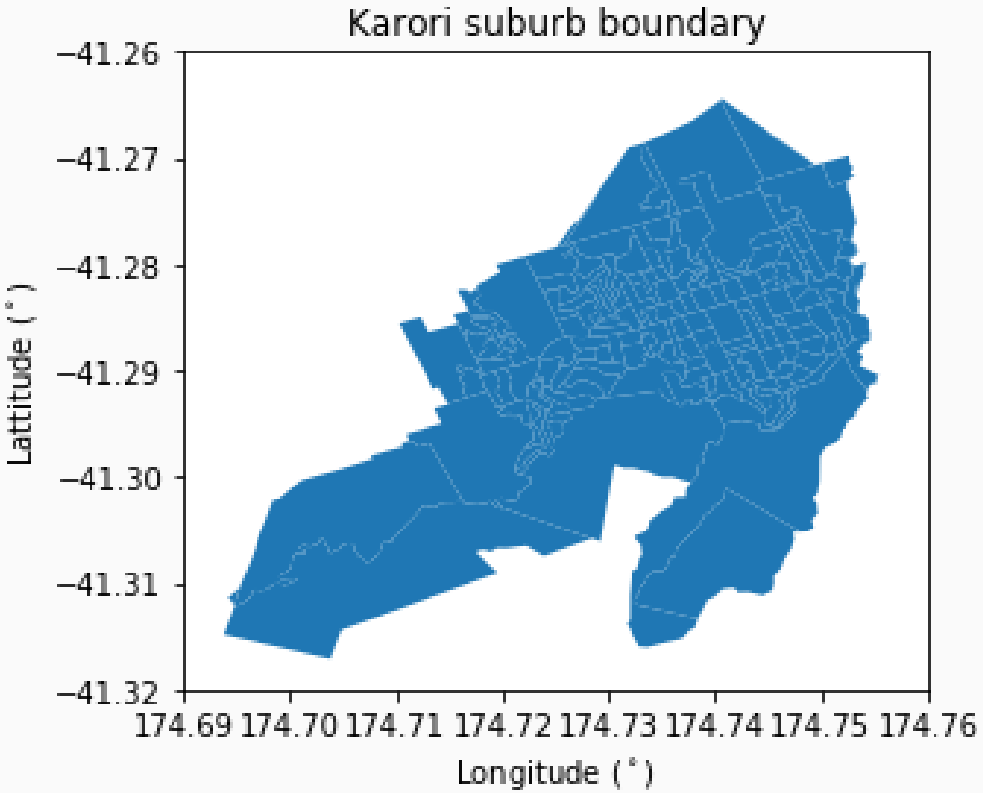
- Line segments that define a street or in this case, a route.

Polygons



- Poylgon boundary of suburb

Complex abstractions



- Poylgon boundary of suburb
- Polygon boundaries of meshblocks *within* suburb

Abstraction: map to graph

Creating a street graph

```
G_sub = ox.graph_from_point((wcc_playgrounds.ix[32]['lat'],
                             wcc_playgrounds.ix[32]['lon']),
                             distance=1500,
                             network_type='walk')
ox.plot_graph(G_sub, fig_height=8)
```

Map represented as street edges with intersections as nodes

Data: spatial primitives

Spatial entities in geodataframes

```
(wlg_meshblock_suburbs_ov
.query('suburb == "Karori"')[['suburb', 'postcode', 'MB2019_V1_00', 'geometry']]
.head())
```

	suburb	postcode	MB2019_V1_00	geometry
917	Karori	6012.0	2104100	POLYGON ((174.7527410567269 -41.27200381936174...
934	Karori	6012.0	2106100	POLYGON ((174.7533828776446 -41.28300381209932...
942	Karori	6012.0	2105503	(POLYGON ((174.7530386517898 -41.2800340600980...
944	Karori	6012.0	2149500	POLYGON ((174.7535317353109 -41.28234654562732...
947	Karori	6012.0	2149400	(POLYGON ((174.754264330901 -41.27988740170357...

Points, lines and polygons can all be compressed in a geodataframe.

Data: graphs

Graph nodes in geodataframes

```
nodes_gdfs, edges_gdfs = ox.graph_to_gdfs(G_sub)
nodes_gdfs.head()
```

	elevation	highway	osmid	x	y	geometry
1259077823	196.755	NaN	1259077823	174.792882	-41.227920	POINT (174.7928822 -41.22792)
1259077824	218.696	NaN	1259077824	174.791983	-41.229385	POINT (174.7919835 -41.2293852)
1259077827	163.804	NaN	1259077827	174.805433	-41.213698	POINT (174.8054327 -41.2136978)
3619684648	12.692	NaN	3619684648	174.780604	-41.276563	POINT (174.7806038 -41.2765628)
3619684652	12.344	NaN	3619684652	174.781234	-41.276037	POINT (174.7812341 -41.2760368)

Graph edges in geodataframes

```
edges_gdfs[['name', 'grade', 'osmid', 'maxspeed']].head()
```

		name	grade	length	osmid	maxspeed
1259077823	**1259072929**	Truscott Avenue	0.1319	66.800	110175609	50
	1259072943	Truscott Avenue	-0.0475	65.443	110175609	50
	6083853567	John Sims Drive	-0.1116	177.292	110176112	50
1259077824	**6083853567**	John Sims Drive	0.1650	13.022	110176112	50
1259077827	**465611807**	Cambrian Street	0.0396	71.272	107284021	50

Data used for analysis

- Street graph: with street gradient attribute for edges
- WCC playgrounds represented as points
- Suburb boundaries defined by WCC as polygons

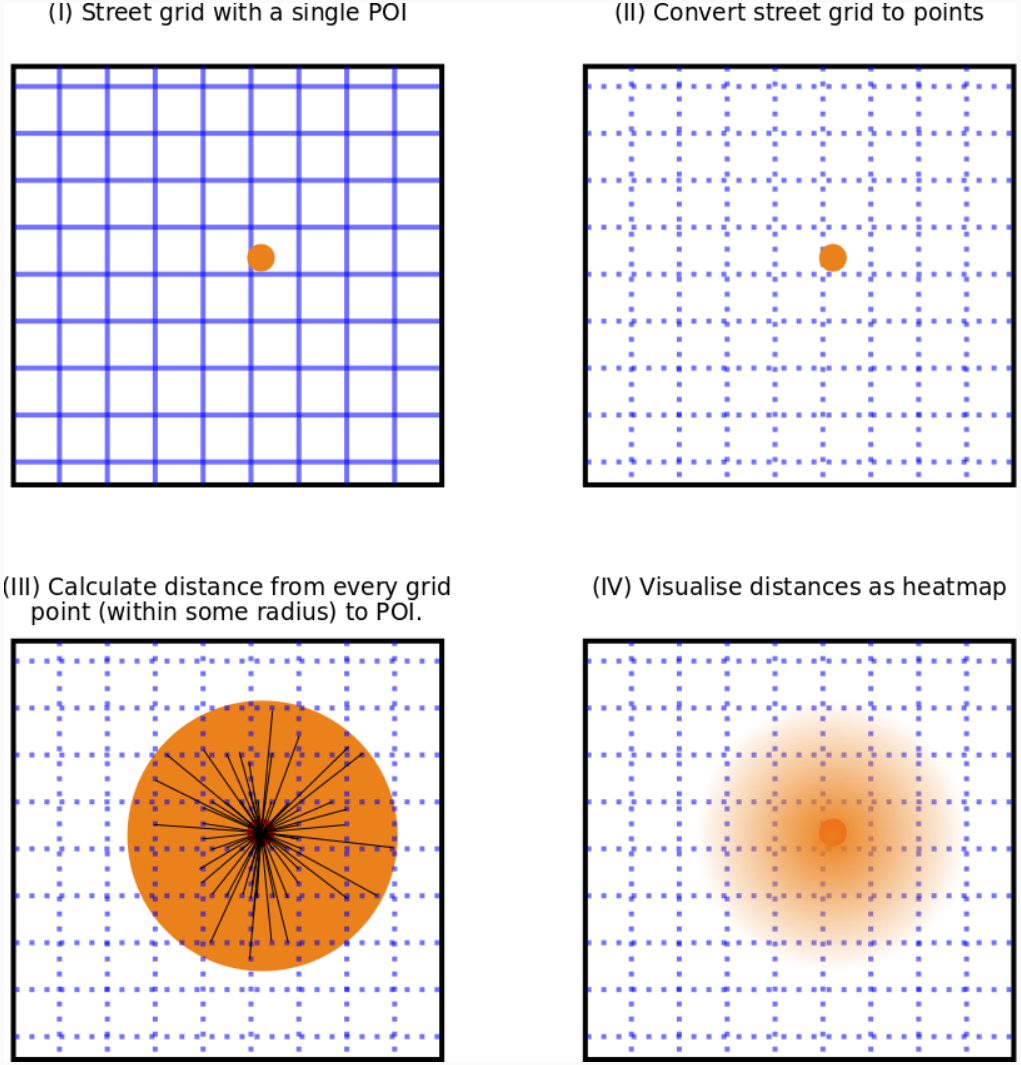
Model 1

Approximating walkability as accessibility

Just to make life confusing, there are several definitions of accessibility. For the following analyses, accessibility is:

- an objective metric
- calculated with a street graph and points of interest (POIs)
  - e.g. Wellington street graph and playground locations
- calculated with a specific unit of interest
  - e.g. distance, travel time, total travel time etc.
- limited to *nearest* POI

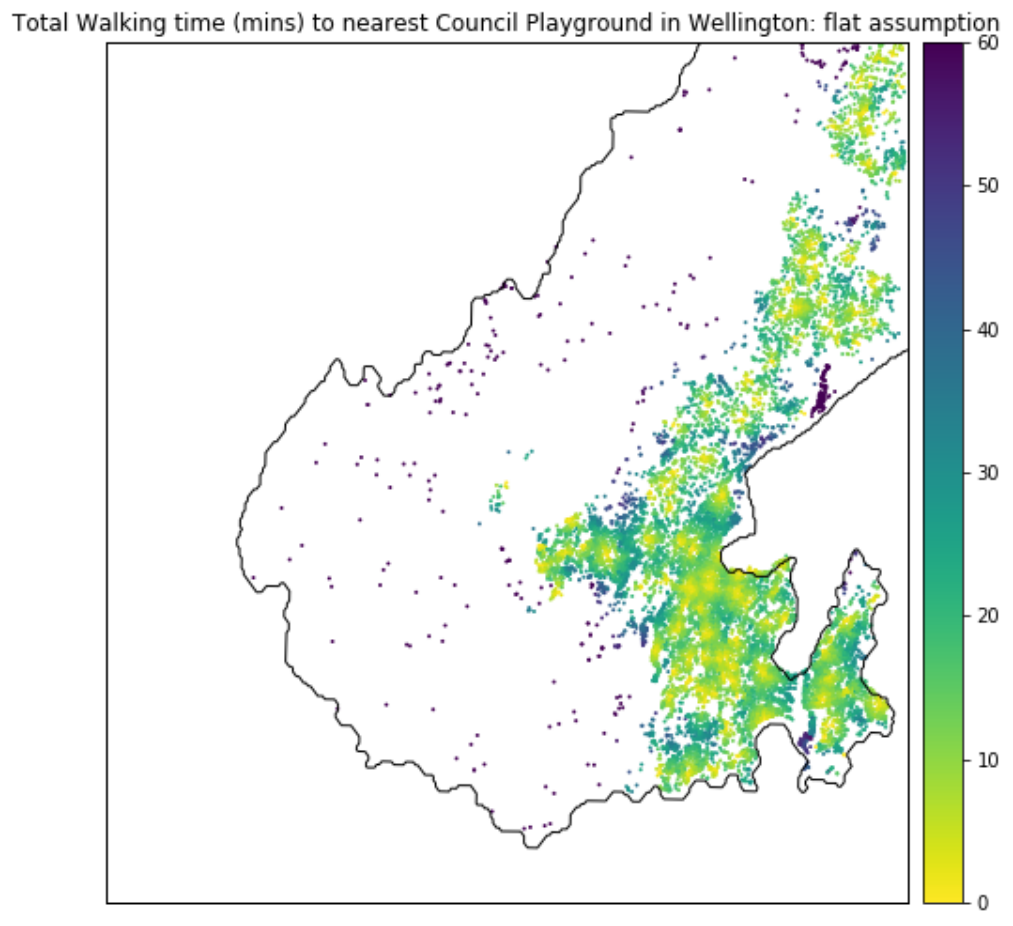
How to calculate accessibility



Accessibility on streets

- Find closest street graph nodes to: start and park
- Find shortest part between start and park nodes
- Sum edge weights of shortest path

Efficient accessibility with Pandana

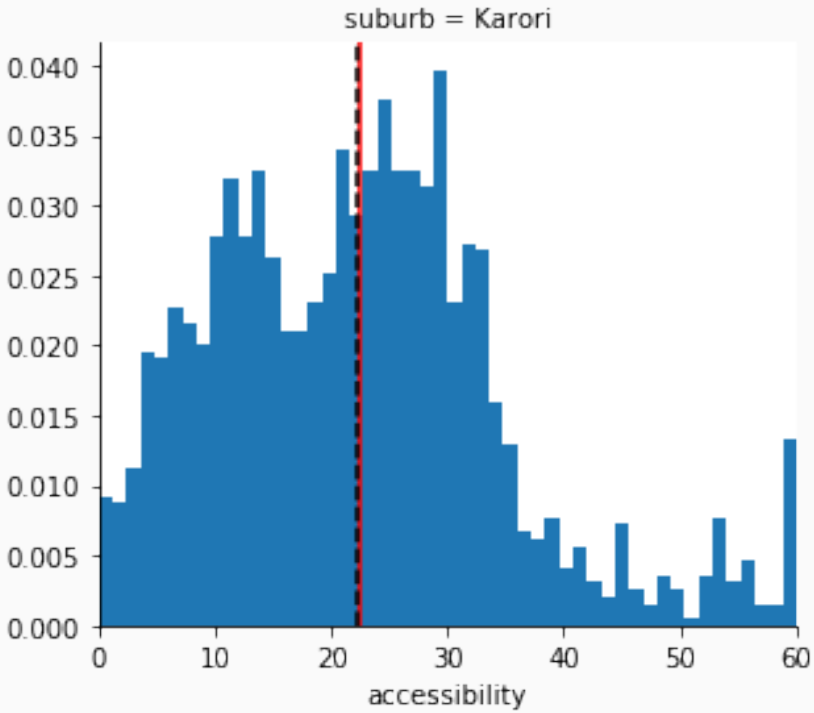


Model 2

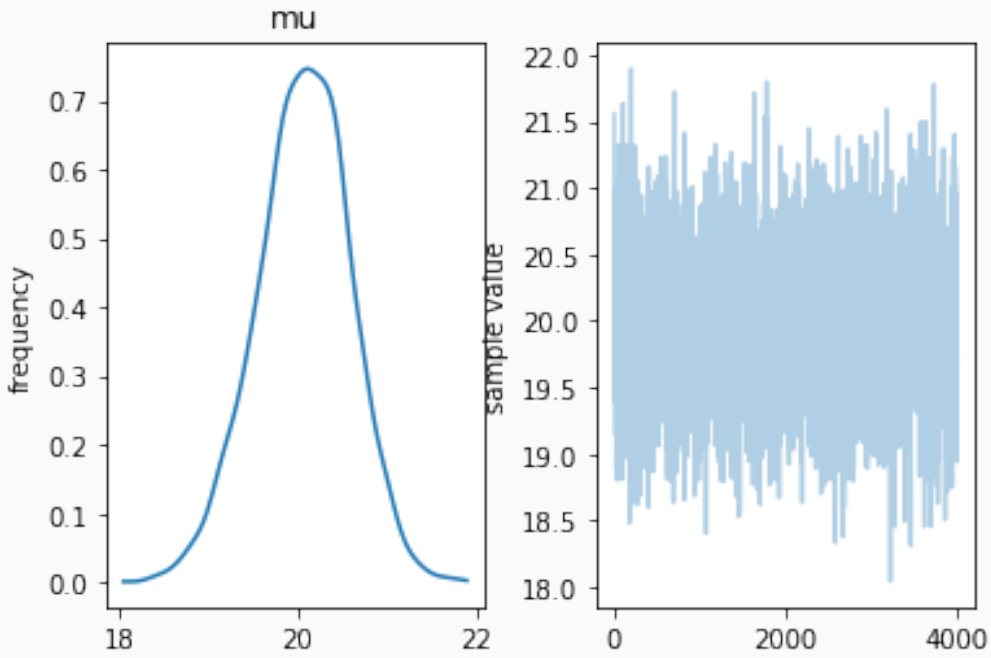
Bayesian modelling

From the observed accessibility data, what is the average accessibility to a playground across the different Wellington suburbs?

Set up Bayesian model



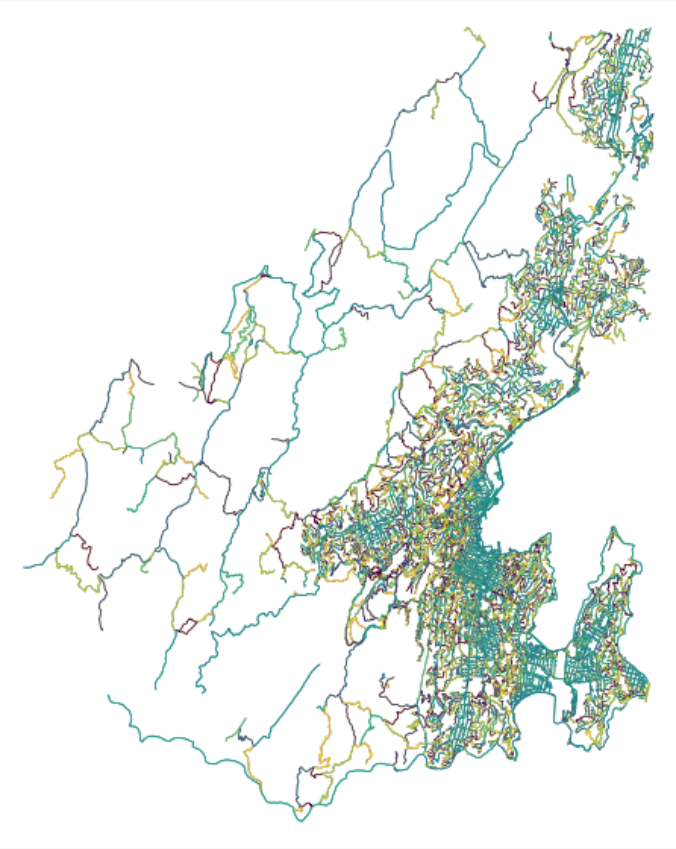
- Model individual suburb accessibility (A_s) as a lower value truncated normal distribution.
- Normal distribution: $A_s \sim N(\mu, \sigma)$
- Truncation condition: $A_s \in [0, \text{inf}]$



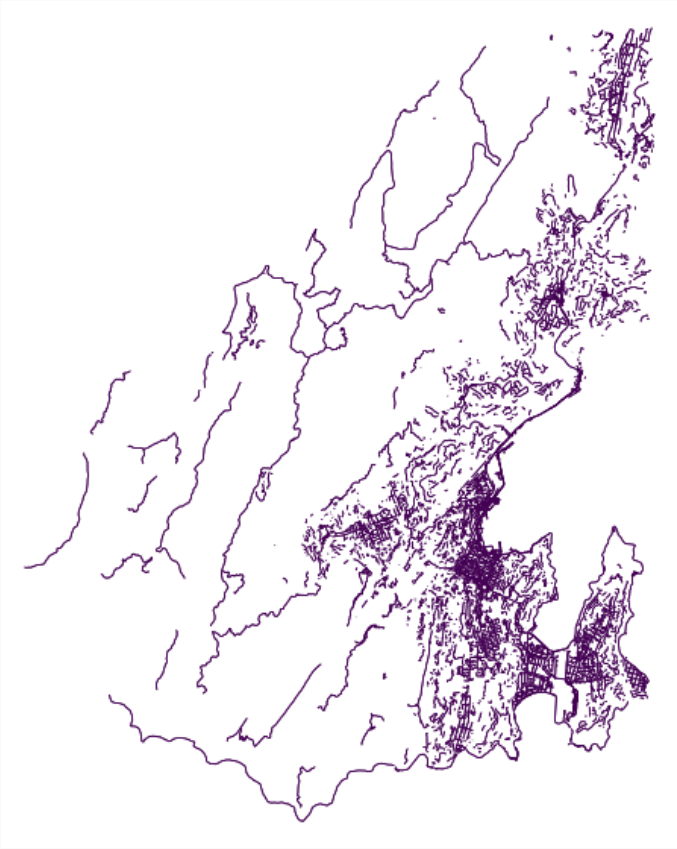
- Stan model output for μ (labelled as mu)
- Samples of μ drawn by Stan

Insights (mostly visual)

Street graph with gradients

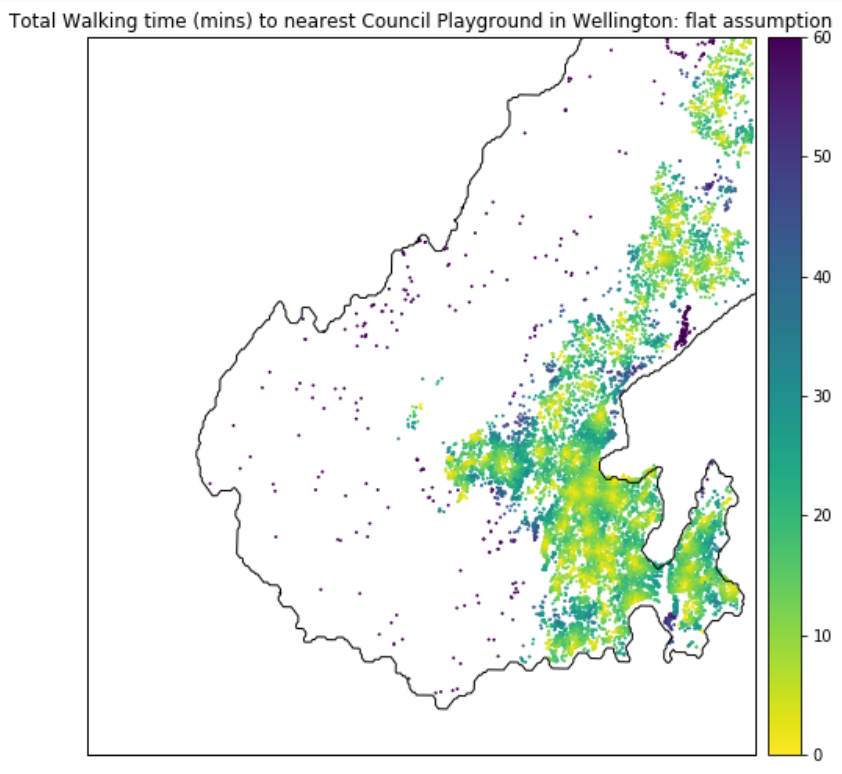


- All edges (green ~ flat gradient)

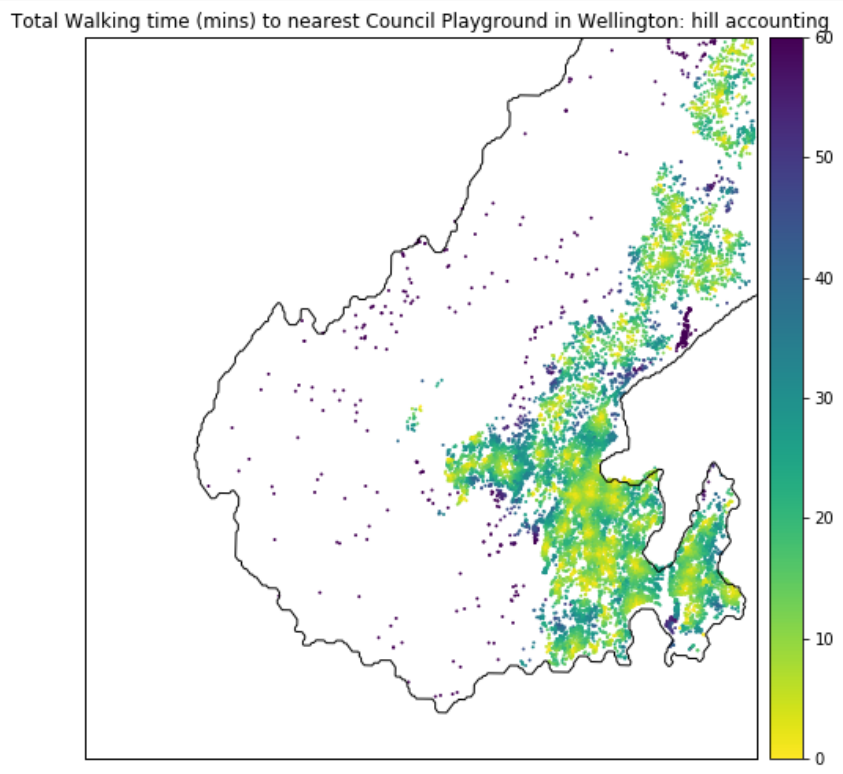


- Edges within 5% absolute gradient

Hills vs. flat land

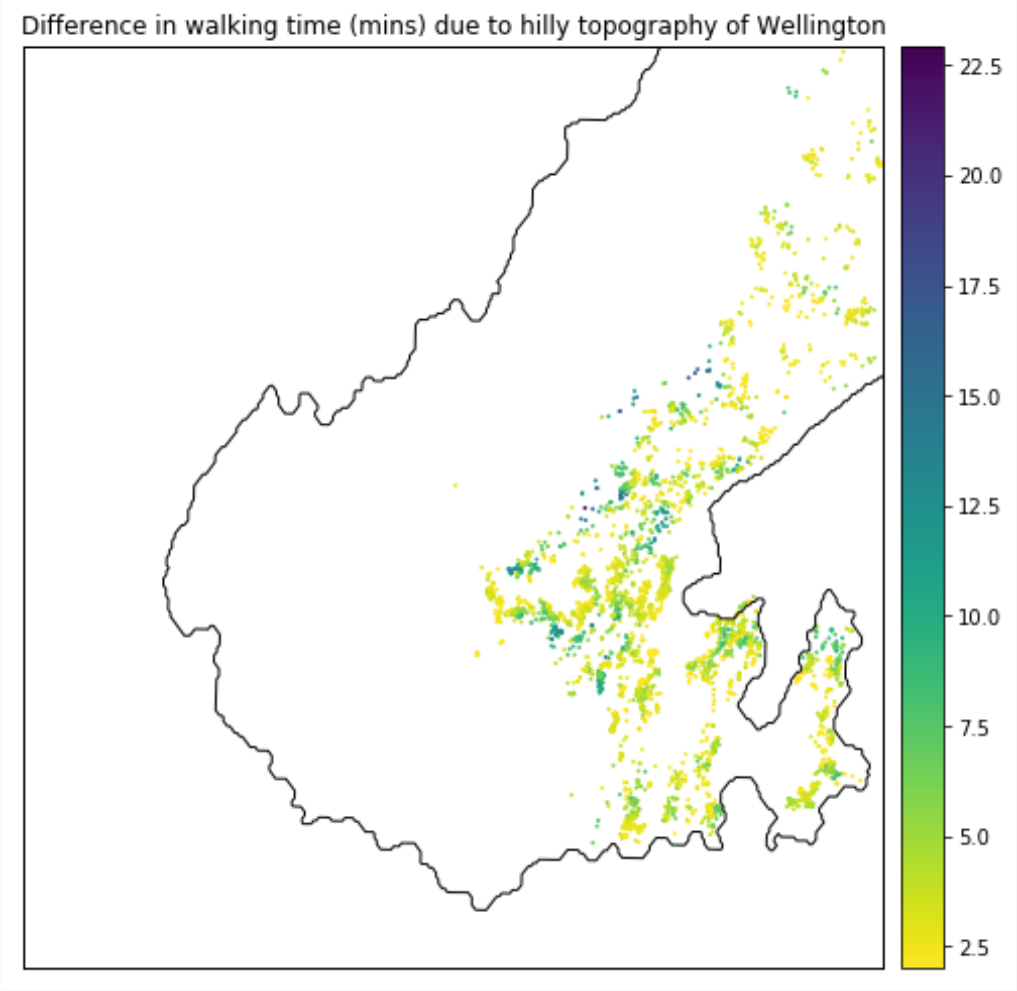


- Assuming single speed

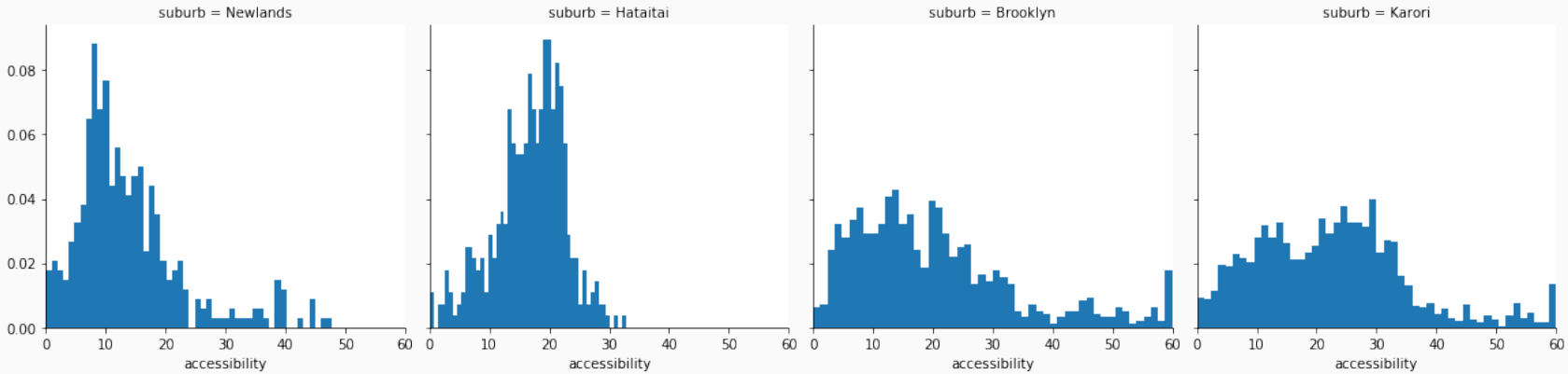


- Accounting for speed variability due to hills

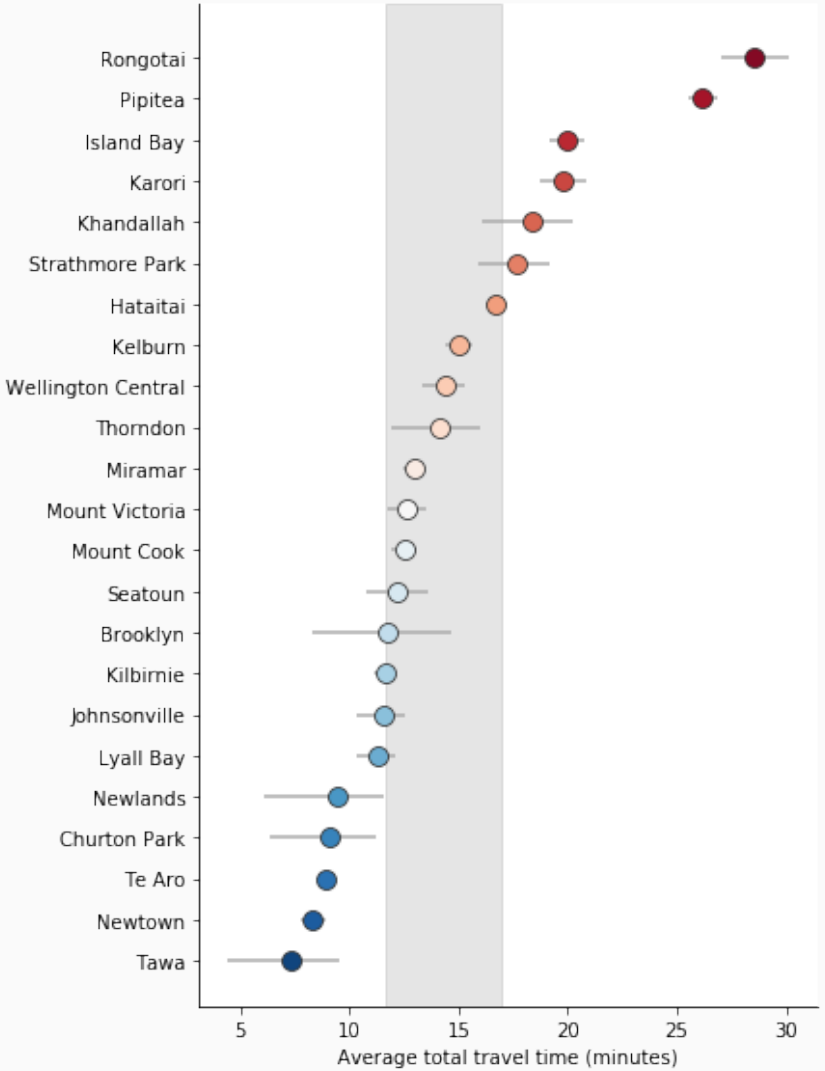
Impact of hills on playground accessibility



Accessibility by suburb



Modelling playground accessibility by suburb



What have we learned?

- Hills have an impact on total travel time. Differences in total travel time can be up to 20 minutes.
- Wellington suburbs average 12-17 minutes in total travel time to nearest playground.
- But, there is a large variation *within* suburbs.

Lots more work to do!

- Making graphs better
- Removing the 60 minute spikes from the Bayesian modelling
- Exploring the heterogeneity in accessibility *within* suburbs
- Impact of including school playgrounds in the analysis
- Impact of adding a new council playground (e.g. Berhampore playground coming in ~2020)

Resources

- Write up on https://shriv.github.io
- Code in https://github.com/shriv/accessibility-series/

Image Credits

- Created by Thibault Geffroy for NounProject
- Created by Thuy Nguyen for NounProject
- Created by Christopher Smith for NounProject
- Created by ProSymbols for NounProject