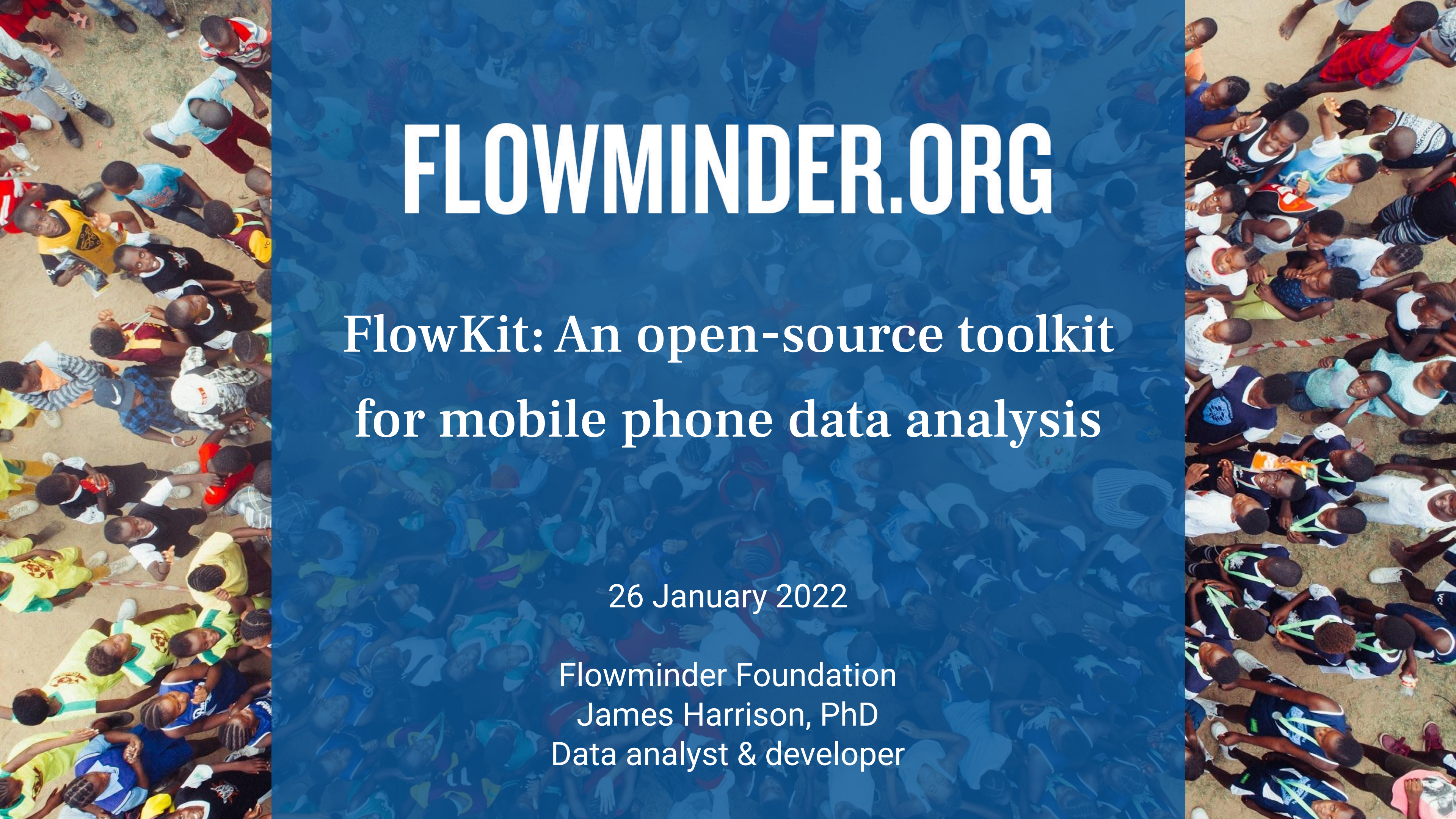


FLOWMINDER.ORG

# FlowKit: An open-source toolkit for mobile phone data analysis

26 January 2022

Flowminder Foundation
James Harrison, PhD
Data analyst & developer

*Some of our supporters:*

JIPS informing solutions to internal displacement

GSMA

Hewlett Foundation

IDB Inter-American Development Bank

Gavi The Vaccine Alliance

dial Digital Impact Alliance

IOM UN MIGRATION

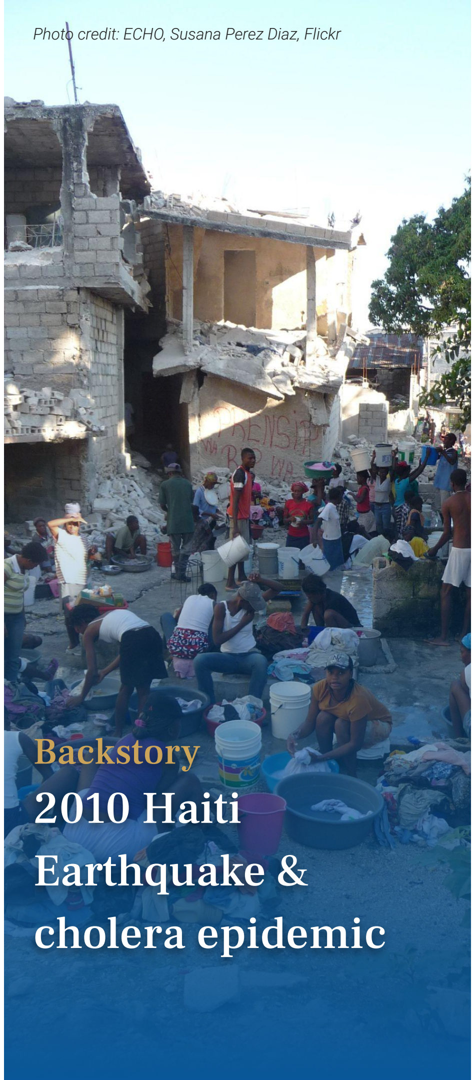
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of Foreign Affairs FDFA
**State Secretariat STS-FDFA**
Peace and Human Rights

## Our supporters

Non-profit funded by & supporting key actors

*Photo credit: ECHO, Susana Perez Diaz, Flickr*



## Backstory

2010 Haiti Earthquake & cholera epidemic

ARTICLES
Past and future spread of th
Aedes aegypti and Aedes alb
The New York Times
Le Monde
edictability of population displacement after th
rthquake
nature publishing group
stract
nat
Population Movements with Mobile Pho

## Science & Innovation

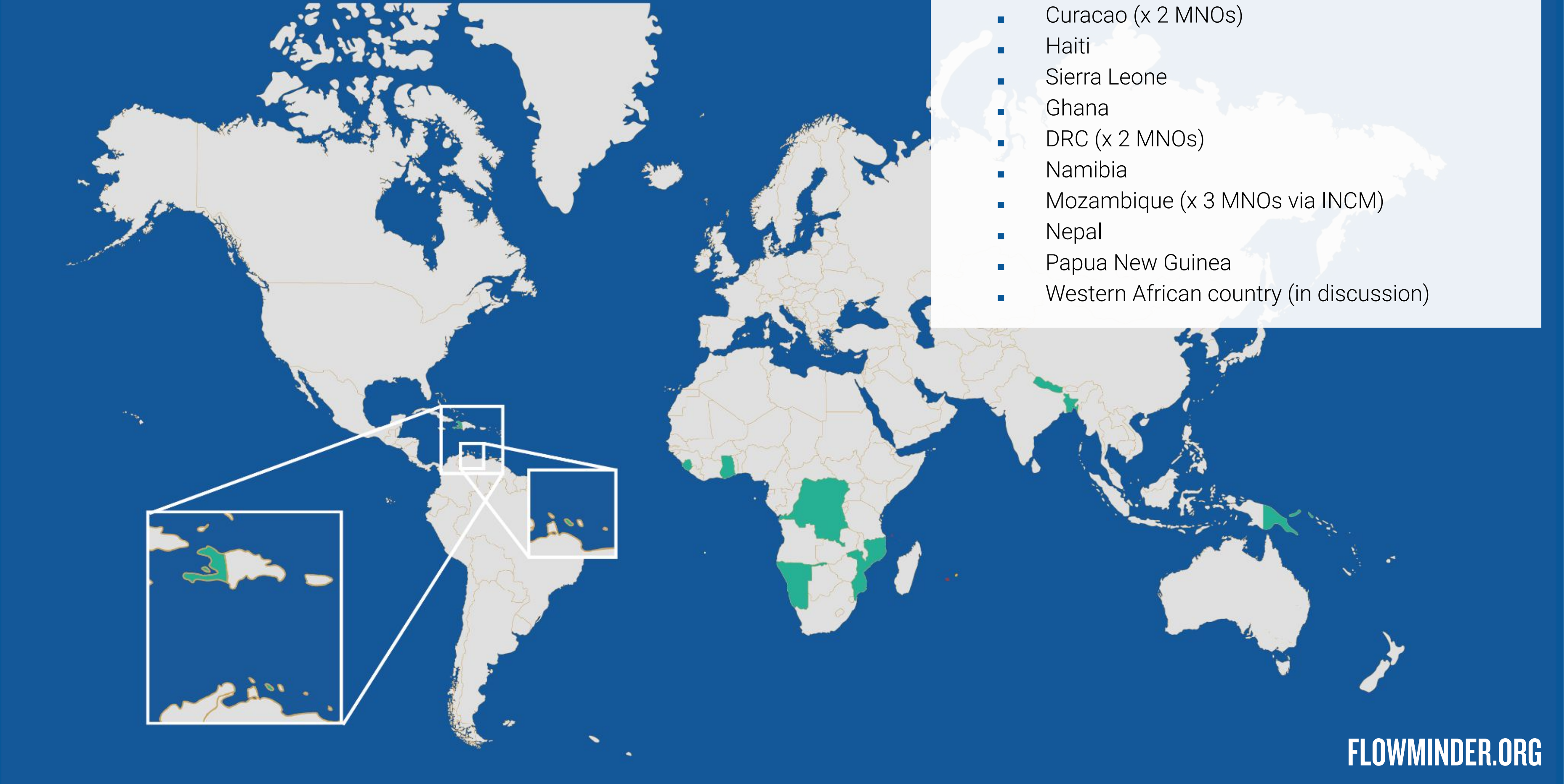
Solid academic research. 50+ peer reviewed publications

## Our team & work

38 staff to enable data driven decision support for LMICs

FLOWMINDER.ORG

# Flowminder MNO collaborations to date



**Countries where Flowminder has collaboration with MNOs (present and past):**

- Curacao (x 2 MNOs)
- Haiti
- Sierra Leone
- Ghana
- DRC (x 2 MNOs)
- Namibia
- Mozambique (x 3 MNOs via INCM)
- Nepal
- Papua New Guinea
- Western African country (in discussion)

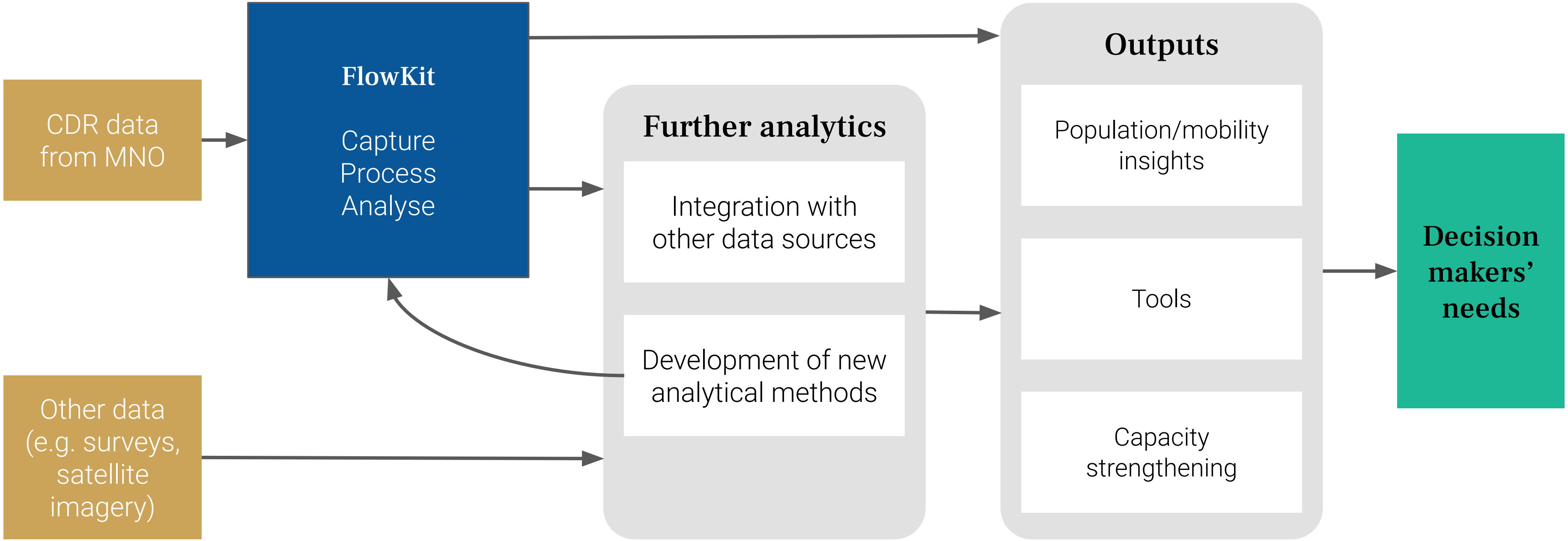
FLOWMINDER.ORG

# What is FlowKit?

- Open-source (MPLv2) software suite
- Enables secure processing, analysis and granular access control to CDR data for humanitarian and development purposes
- Designed to be installed within mobile network operator’s firewall
- Containerised to simplify deployment

-> https://flowkit.xyz

# FlowKit in the wider context



CDR data from MNO

**FlowKit**

Capture
Process
Analyse

Other data (e.g. surveys, satellite imagery)

**Further analytics**

Integration with other data sources

Development of new analytical methods

**Outputs**

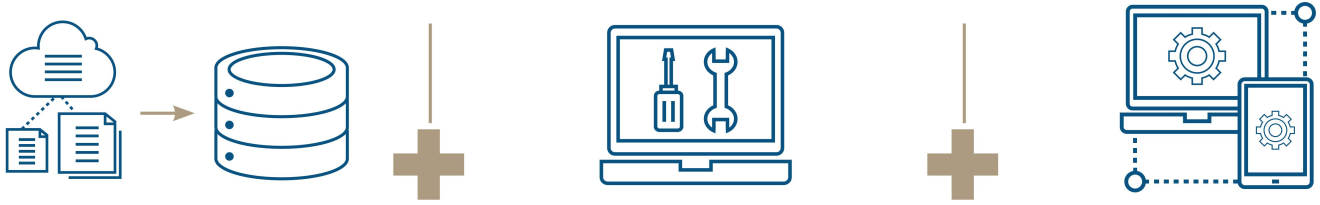
Population/mobility insights

Tools

Capacity strengthening

**Decision makers' needs**

# FlowKit components



Mobile data

Database

Ingestion and QA

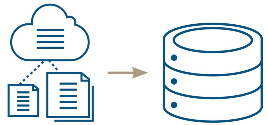
Tools

Processing

API

Authentication and controlled access

# FlowKit components: Ingestion and QA



Ingestion and QA

Purpose: Reduce effort to get data “analysis-ready”

## Data ingestion

Connection made during installation

New data ingested automatically

Reduces effort prior to processing data

## Quality assurance

QA checks run automatically on new data

Identify data issues which may affect outputs

## Data model

Utility tables (events, cells, geography)

Familiar structure to underlie queries

# FlowKit components: Processing

Tools

Processing

Purpose: Method use, re-use and creation

## Code and methods

Python library for constructing SQL queries

Mobility, characteristics, subsetting

Re-use tested methods

## PostgreSQL database

Caching: break down large queries, re-use intermediate results

PostGIS for geospatial analysis

Efficient performance on a single server

## Flexibility

May already provide all that is needed

Extensible, modular design

Designed to grow

# FlowKit components: Authentication, access control

API

Authentication and controlled access

Purpose: Permissions and integration

## Data security

Outputs are anonymised

Grant flexible access to aggregated results without requiring access to individual-level data

## Interface (API)

Trigger queries and retrieve results

Protect users from internal changes

Facilitates integration with other tools

## Authentication

Granular permissions

All transactions logged for auditing

Reduces risk of data leak

CDR-derived insights should never permit the identification of individual subscribers.

# Processing sensitive data

FlowKit protects the privacy of individual subscribers in the following ways:

- All processing of individual-level data occurs within MNO's premises
- API enables flexible querying of data, while only allowing anonymised outputs to be retrieved
- Outputs from API are aggregated over subscribers, with any rows corresponding to 15 or fewer subscribers redacted (k-anonymity)

Outputs from FlowKit’s API are **aggregated**, hence protecting the privacy of subscribers.

# Processing sensitive data

(cont’d)

- Granular access control facilitates data minimisation
- Audit logs
- Direct identifiers (phone numbers etc.) are pseudonymised before ingestion into FlowKit, to reduce risk of re-identification in the event of a data leak

# Applications

# Integrating mobile operator data into official statistics

Integrating mobile operator data into official statistics for the wellbeing of all in Ghana
Helping Parliament to improve quality of life in Ghana through evidence, using data to oversee progress towards SDGs
BELAQUA
DELL

FLOWMINDER.ORG

# Official statistics: Ghana

Flowminder has partnered with Ghana Statistical Service (GSS) and Vodafone Ghana.

Flowminder is supporting GSS to use CDR data from Vodafone Ghana to produce information on population mobility and characteristics.

The aim is to strengthen humanitarian and development decision-making (e.g. for public health, disaster preparedness or transportation planning).

# Official statistics: Ghana

## How FlowKit helps:

- Vodafone Ghana can provide on-demand access to CDR-derived aggregates without sharing individual-level data
- GSS can use methods implemented in FlowKit (e.g. home location estimation) to produce data products that support national statistics

This public-private collaboration is first of its kind in Ghana, and one of the first in Africa.

# Public Health

FLOWMINDER.ORG

# Mapping for Health in DRC

**Role of mobility data** *(data provided by Vodacom RDC):*

- Estimates of hard-to-reach / highly mobile / displaced populations
- Monthly updates on population movements, to aid resource supply planning
- Alert health planners when sudden large-scale displacements occur
- Evaluate the scope for updating population density estimates and projections using mobile phone usage data

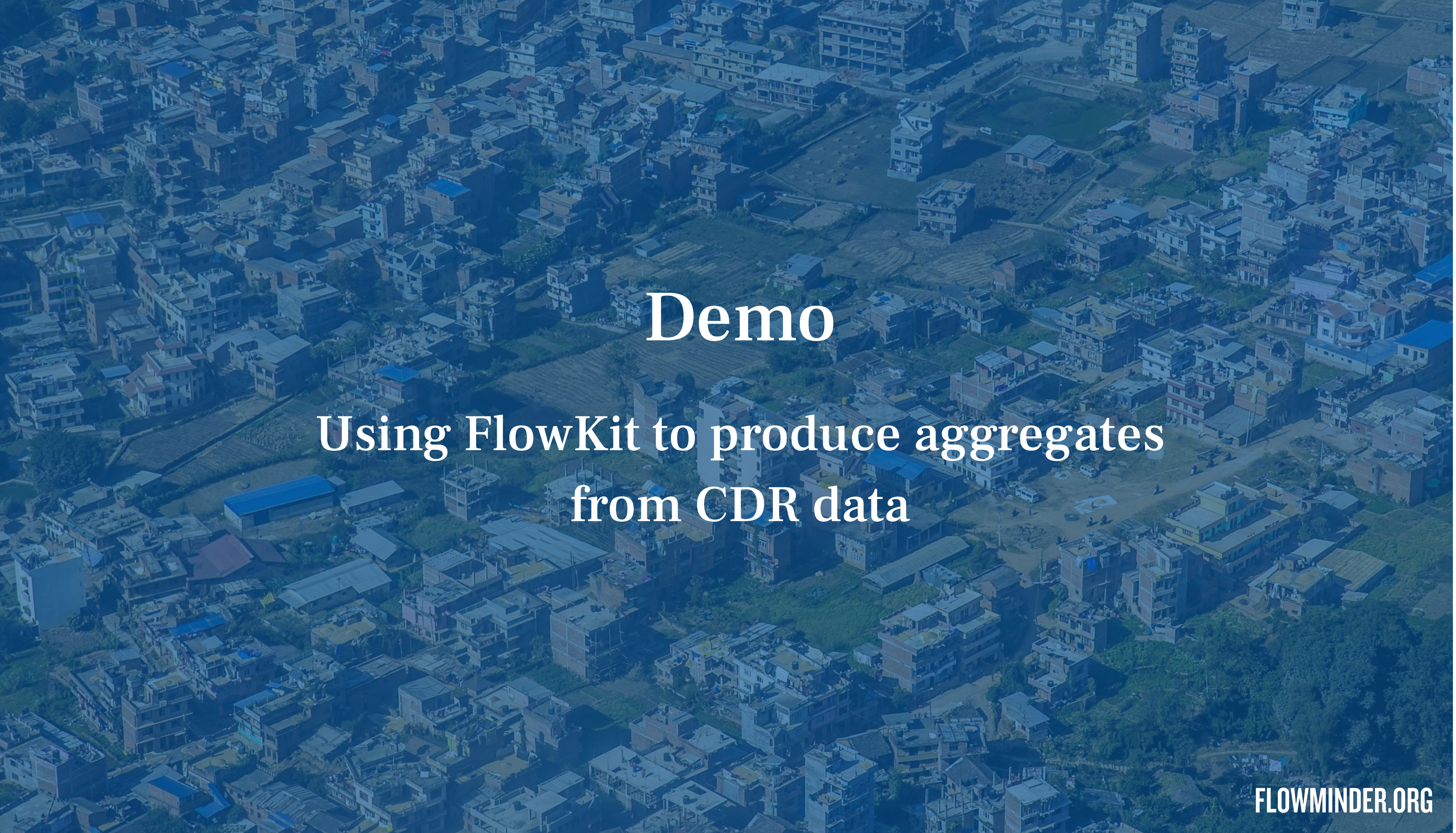
GRID3
MAPPING FOR HEALTH
FLOWMINDER.ORG Center for International Earth Science Information Network
EARTH INSTITUTE | COLUMBIA UNIVERSITY

Initiative to improve the **effectiveness** and **equity of vaccination interventions** in the Democratic Republic of the Congo (DRC).

# Mapping for Health in DRC

## How FlowKit helps:

- Automated tools built on top of the FlowKit API to regularly produce and disseminate data outputs
- Individual-level data never leave Vodacom's system
- Convenient platform for prototyping new methods
- Methods developed are built into FlowKit for re-use in future projects



# Demo

## Using FlowKit to produce aggregates from CDR data

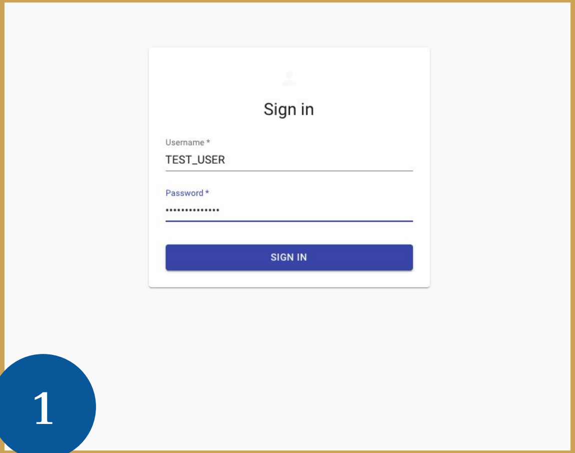
# Prerequisites

- FlowKit installed on a server within MNO premises
- Data (CDR, cell tower locations, geographic boundaries) ingested for the period of interest
- Permissions to access the required aggregates granted by a system administrator
- FlowKit API exposed at an accessible URL
- User is familiar with python data analysis ecosystem

More information at https://flowkit.xyz

# Get an access token

1



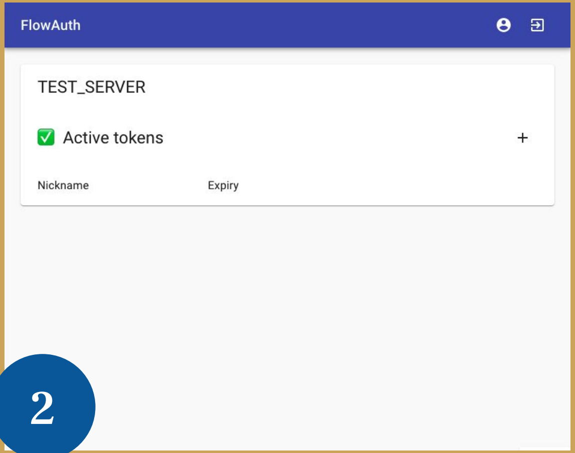
Sign in

Username *
TEST_USER

Password *
••••••••••••••

SIGN IN

2



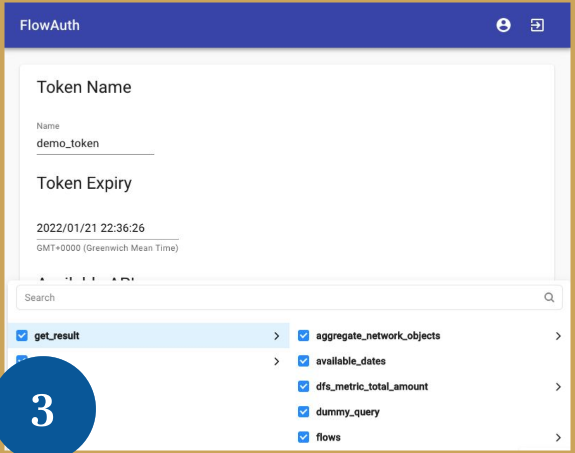
FlowAuth

TEST_SERVER

✅ Active tokens +

| Nickname | Expiry |
|---|---|

3



FlowAuth

Token Name

Name
demo_token

Token Expiry

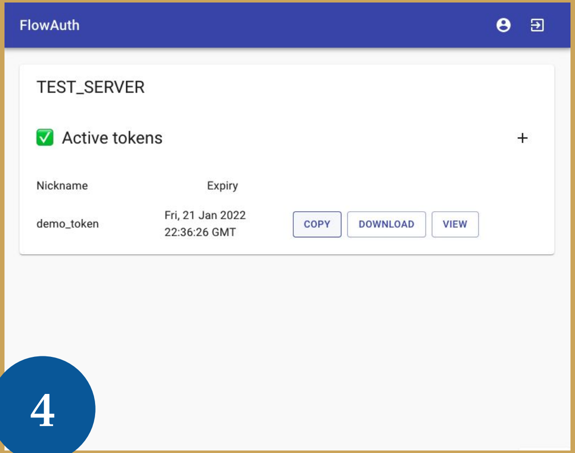
2022/01/21 22:36:26
GMT+0000 (Greenwich Mean Time)

Search

☑ get_result >
☑ >

☑ aggregate_network_objects >
☑ available_dates
☑ dfs_metric_total_amount >
☑ dummy_query
☑ flows >

4



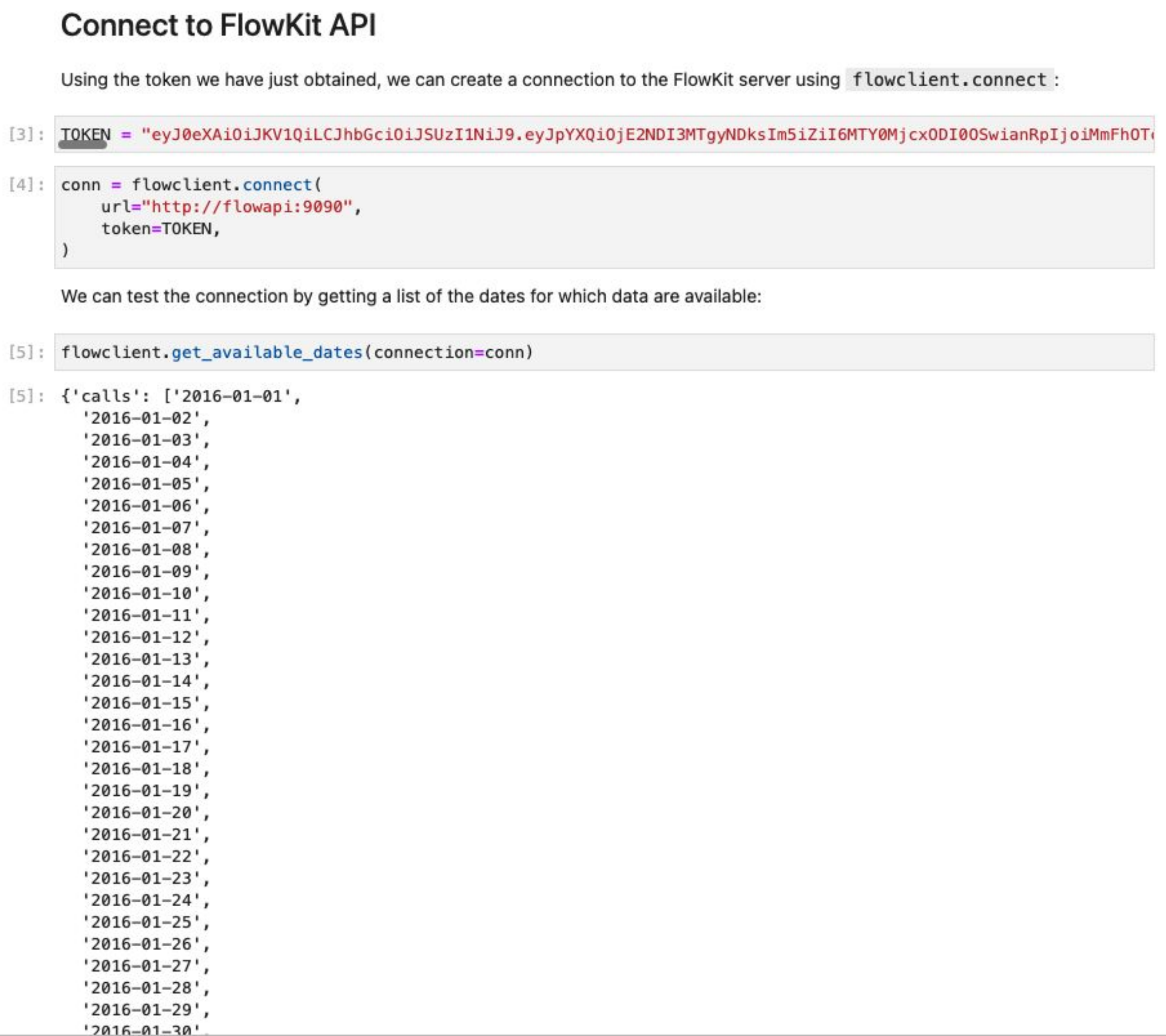
FlowAuth

TEST_SERVER

✅ Active tokens +

| Nickname | Expiry | |
|---|---|---|
| demo_token | Fri, 21 Jan 2022<br>22:36:26 GMT | COPY DOWNLOAD VIEW |

FLOWMINDER.ORG



## Connect to FlowKit API

Using the token we have just obtained, we can create a connection to the FlowKit server using `flowclient.connect`:

```
[3]: TOKEN = "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpYXQiOjE2NDI3MTgyNDksIm5iZiI6MTY0MjcxODI0OSwianRpIjoiMmFhOT
```

```
[4]: conn = flowclient.connect(
         url="http://flowapi:9090",
         token=TOKEN,
     )
```

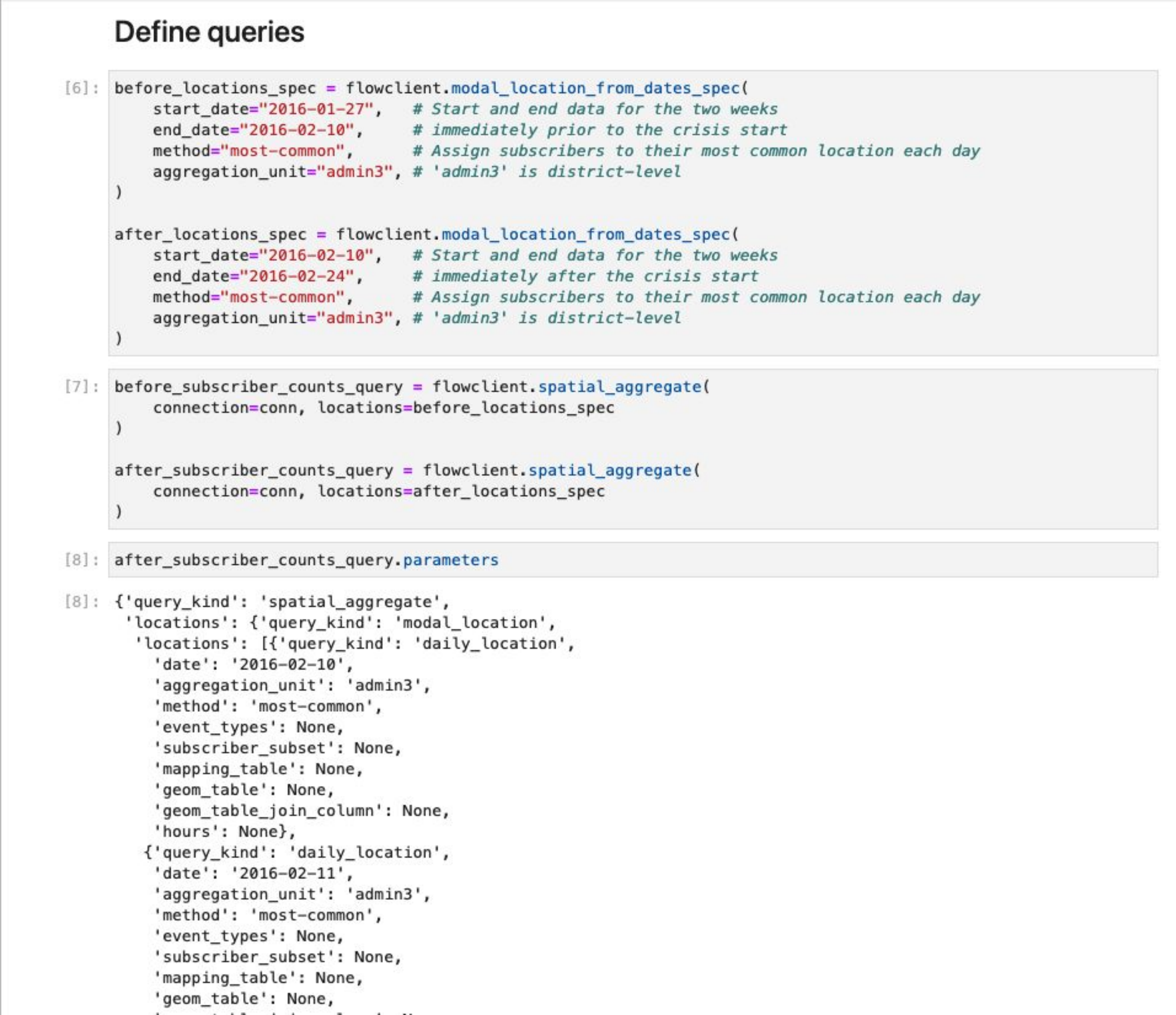
We can test the connection by getting a list of the dates for which data are available:

```
[5]: flowclient.get_available_dates(connection=conn)
```

```
[5]: {'calls': ['2016-01-01',
       '2016-01-02',
       '2016-01-03',
       '2016-01-04',
       '2016-01-05',
       '2016-01-06',
       '2016-01-07',
       '2016-01-08',
       '2016-01-09',
       '2016-01-10',
       '2016-01-11',
       '2016-01-12',
       '2016-01-13',
       '2016-01-14',
       '2016-01-15',
       '2016-01-16',
       '2016-01-17',
       '2016-01-18',
       '2016-01-19',
       '2016-01-20',
       '2016-01-21',
       '2016-01-22',
       '2016-01-23',
       '2016-01-24',
       '2016-01-25',
       '2016-01-26',
       '2016-01-27',
       '2016-01-28',
       '2016-01-29',
       '2016-01-30',
```

# Connect to FlowAPI

FLOWMINDER.ORG



## Define queries

[6]:

```
before_locations_spec = flowclient.modal_location_from_dates_spec(
    start_date="2016-01-27",   # Start and end data for the two weeks
    end_date="2016-02-10",     # immediately prior to the crisis start
    method="most-common",      # Assign subscribers to their most common location each day
    aggregation_unit="admin3", # 'admin3' is district-level
)

after_locations_spec = flowclient.modal_location_from_dates_spec(
    start_date="2016-02-10",   # Start and end data for the two weeks
    end_date="2016-02-24",     # immediately after the crisis start
    method="most-common",      # Assign subscribers to their most common location each day
    aggregation_unit="admin3", # 'admin3' is district-level
)
```

[7]:

```
before_subscriber_counts_query = flowclient.spatial_aggregate(
    connection=conn, locations=before_locations_spec
)

after_subscriber_counts_query = flowclient.spatial_aggregate(
    connection=conn, locations=after_locations_spec
)
```

[8]:

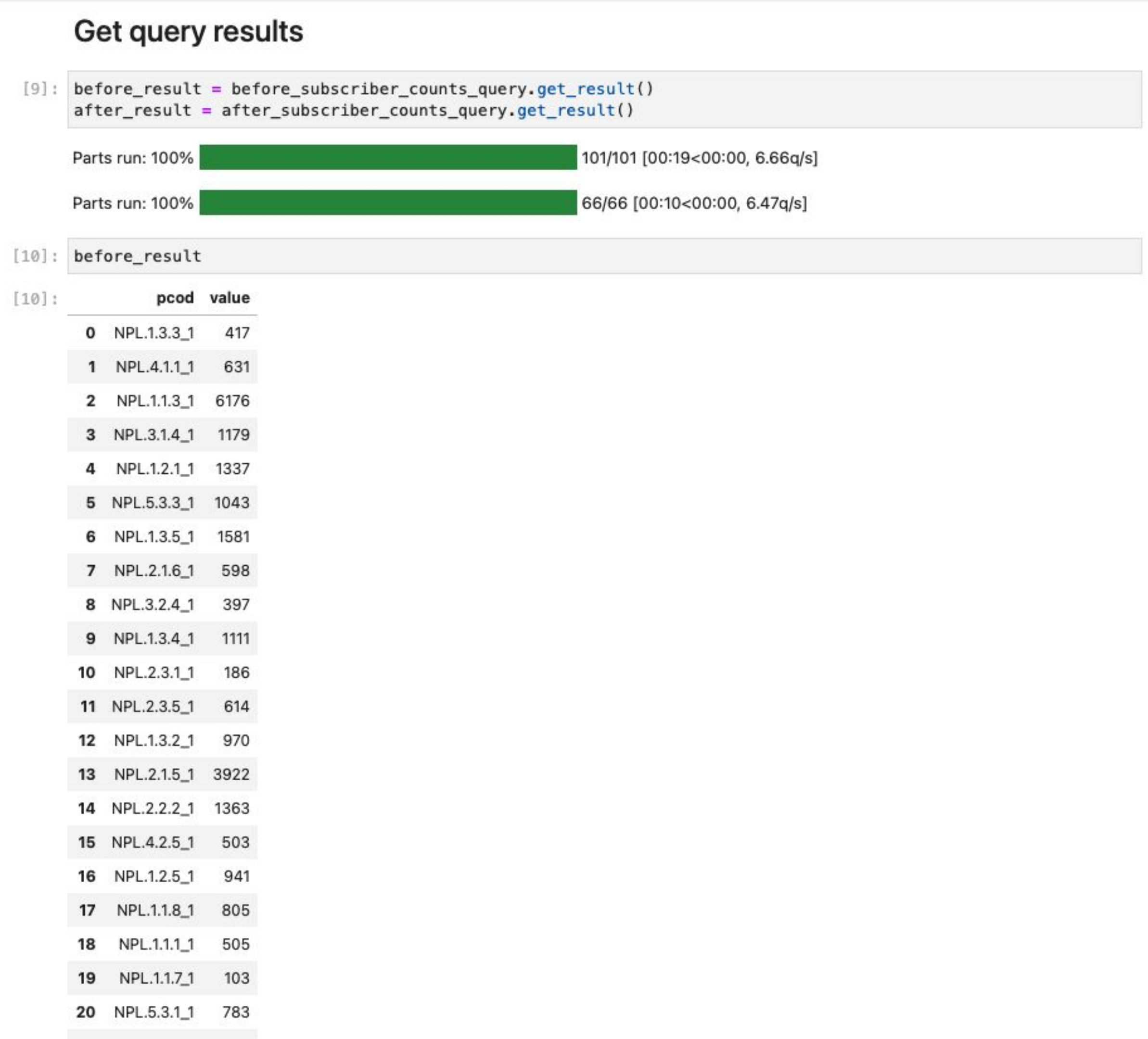
```
after_subscriber_counts_query.parameters
```

[8]:

```
{'query_kind': 'spatial_aggregate',
 'locations': {'query_kind': 'modal_location',
  'locations': [{'query_kind': 'daily_location',
    'date': '2016-02-10',
    'aggregation_unit': 'admin3',
    'method': 'most-common',
    'event_types': None,
    'subscriber_subset': None,
    'mapping_table': None,
    'geom_table': None,
    'geom_table_join_column': None,
    'hours': None},
   {'query_kind': 'daily_location',
    'date': '2016-02-11',
    'aggregation_unit': 'admin3',
    'method': 'most-common',
    'event_types': None,
    'subscriber_subset': None,
    'mapping_table': None,
    'geom_table': None,
```

# Define queries

FLOWMINDER.ORG



## Get query results

```
[9]: before_result = before_subscriber_counts_query.get_result()
     after_result = after_subscriber_counts_query.get_result()
```

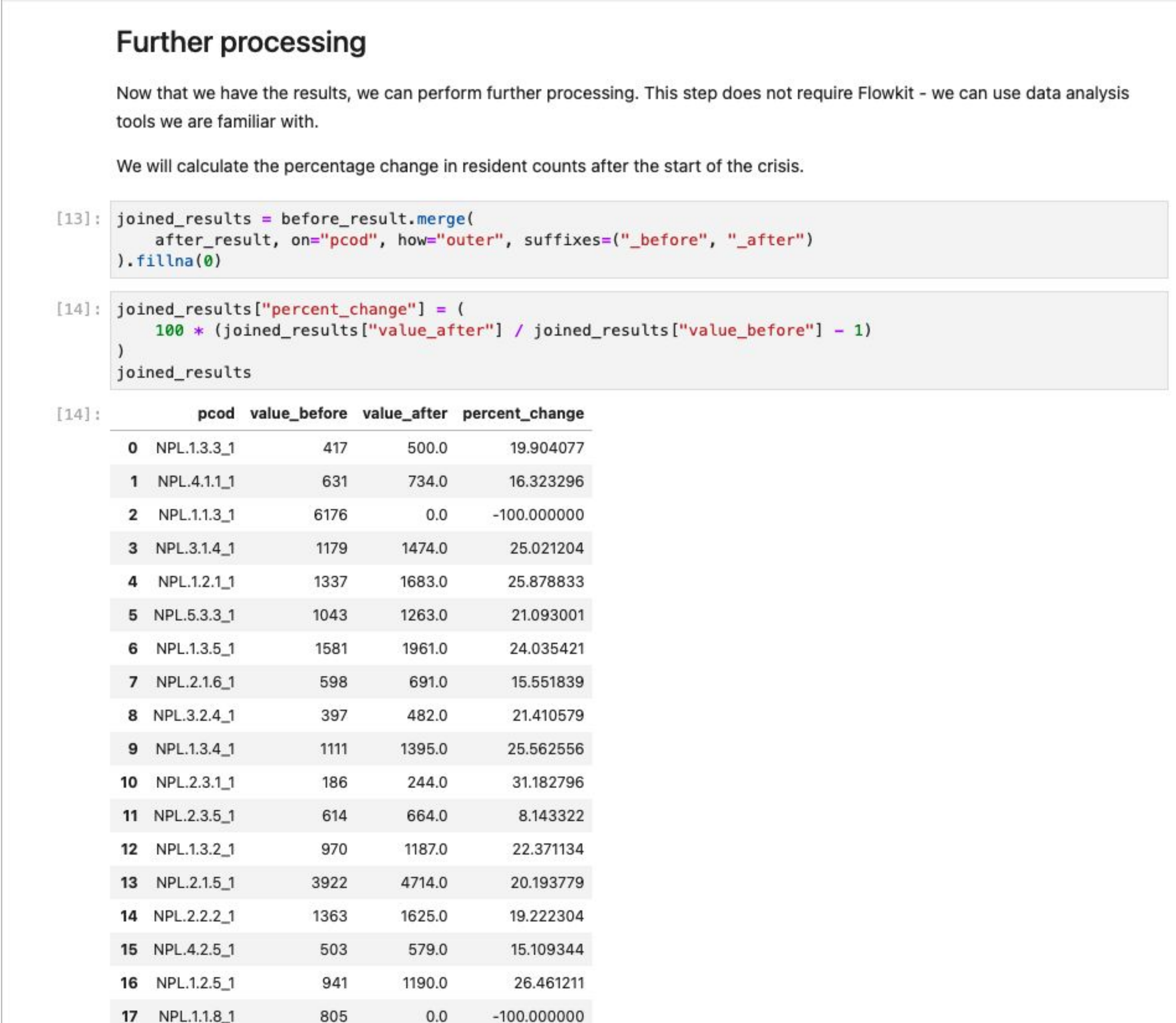
```
Parts run: 100% ████████████████████ 101/101 [00:19<00:00, 6.66q/s]

Parts run: 100% ████████████████████ 66/66 [00:10<00:00, 6.47q/s]
```

```
[10]: before_result
```

[10]:

| | pcod | value |
|---|---|---|
| 0 | NPL.1.3.3_1 | 417 |
| 1 | NPL.4.1.1_1 | 631 |
| 2 | NPL.1.1.3_1 | 6176 |
| 3 | NPL.3.1.4_1 | 1179 |
| 4 | NPL.1.2.1_1 | 1337 |
| 5 | NPL.5.3.3_1 | 1043 |
| 6 | NPL.1.3.5_1 | 1581 |
| 7 | NPL.2.1.6_1 | 598 |
| 8 | NPL.3.2.4_1 | 397 |
| 9 | NPL.1.3.4_1 | 1111 |
| 10 | NPL.2.3.1_1 | 186 |
| 11 | NPL.2.3.5_1 | 614 |
| 12 | NPL.1.3.2_1 | 970 |
| 13 | NPL.2.1.5_1 | 3922 |
| 14 | NPL.2.2.2_1 | 1363 |
| 15 | NPL.4.2.5_1 | 503 |
| 16 | NPL.1.2.5_1 | 941 |
| 17 | NPL.1.1.8_1 | 805 |
| 18 | NPL.1.1.1_1 | 505 |
| 19 | NPL.1.1.7_1 | 103 |
| 20 | NPL.5.3.1_1 | 783 |

# Get query results



## Further processing

Now that we have the results, we can perform further processing. This step does not require Flowkit - we can use data analysis tools we are familiar with.

We will calculate the percentage change in resident counts after the start of the crisis.

```
[13]: joined_results = before_result.merge(
          after_result, on="pcod", how="outer", suffixes=("_before", "_after")
      ).fillna(0)
```

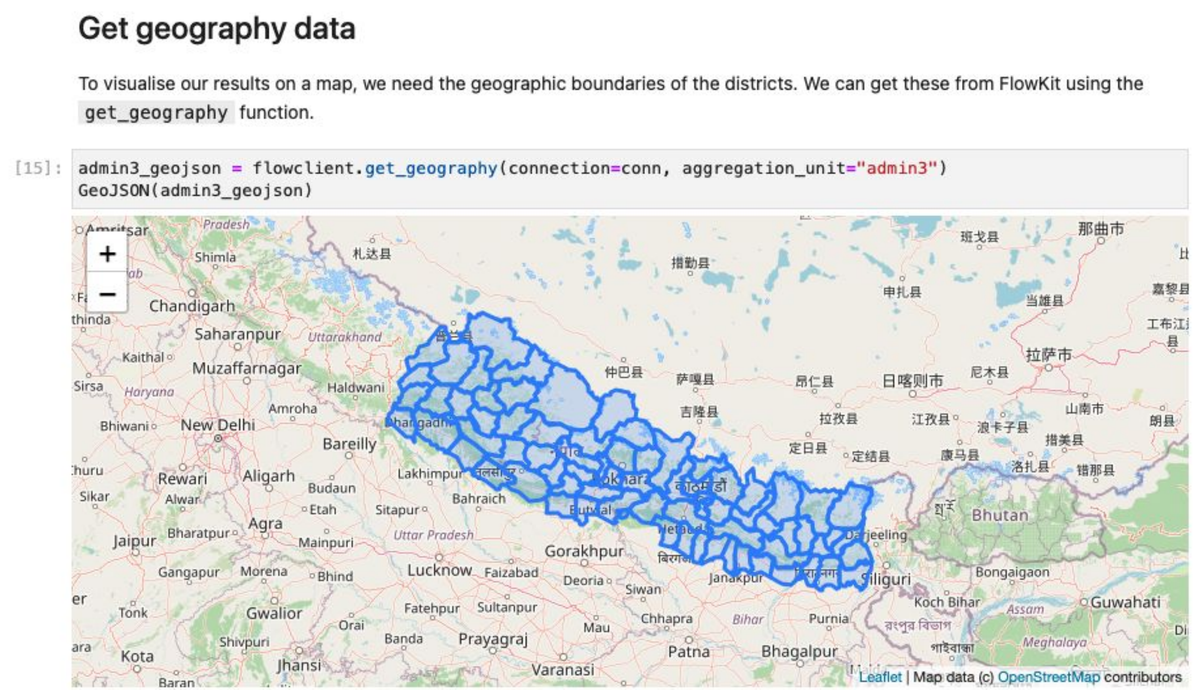
```
[14]: joined_results["percent_change"] = (
          100 * (joined_results["value_after"] / joined_results["value_before"] - 1)
      )
      joined_results
```

[14]:

| | pcod | value_before | value_after | percent_change |
|---|---|---|---|---|
| **0** | NPL.1.3.3_1 | 417 | 500.0 | 19.904077 |
| **1** | NPL.4.1.1_1 | 631 | 734.0 | 16.323296 |
| **2** | NPL.1.1.3_1 | 6176 | 0.0 | -100.000000 |
| **3** | NPL.3.1.4_1 | 1179 | 1474.0 | 25.021204 |
| **4** | NPL.1.2.1_1 | 1337 | 1683.0 | 25.878833 |
| **5** | NPL.5.3.3_1 | 1043 | 1263.0 | 21.093001 |
| **6** | NPL.1.3.5_1 | 1581 | 1961.0 | 24.035421 |
| **7** | NPL.2.1.6_1 | 598 | 691.0 | 15.551839 |
| **8** | NPL.3.2.4_1 | 397 | 482.0 | 21.410579 |
| **9** | NPL.1.3.4_1 | 1111 | 1395.0 | 25.562556 |
| **10** | NPL.2.3.1_1 | 186 | 244.0 | 31.182796 |
| **11** | NPL.2.3.5_1 | 614 | 664.0 | 8.143322 |
| **12** | NPL.1.3.2_1 | 970 | 1187.0 | 22.371134 |
| **13** | NPL.2.1.5_1 | 3922 | 4714.0 | 20.193779 |
| **14** | NPL.2.2.2_1 | 1363 | 1625.0 | 19.222304 |
| **15** | NPL.4.2.5_1 | 503 | 579.0 | 15.109344 |
| **16** | NPL.1.2.5_1 | 941 | 1190.0 | 26.461211 |
| **17** | NPL.1.1.8_1 | 805 | 0.0 | -100.000000 |

# Further processing

FLOWMINDER.ORG

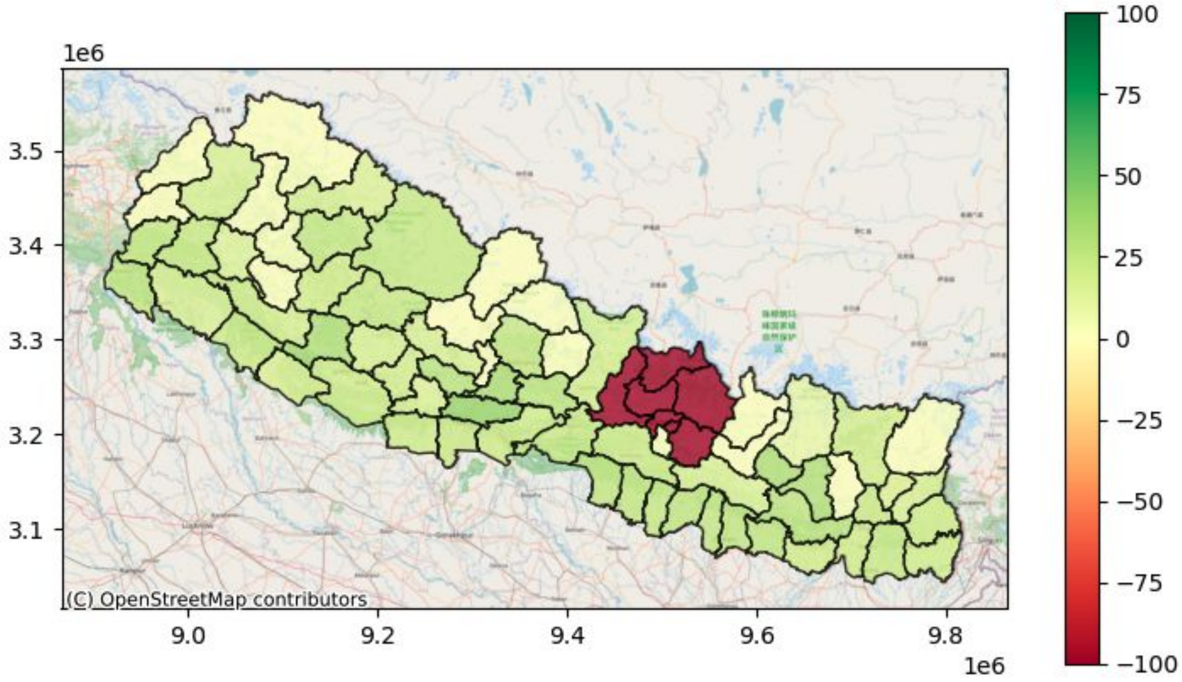


## Get geography data

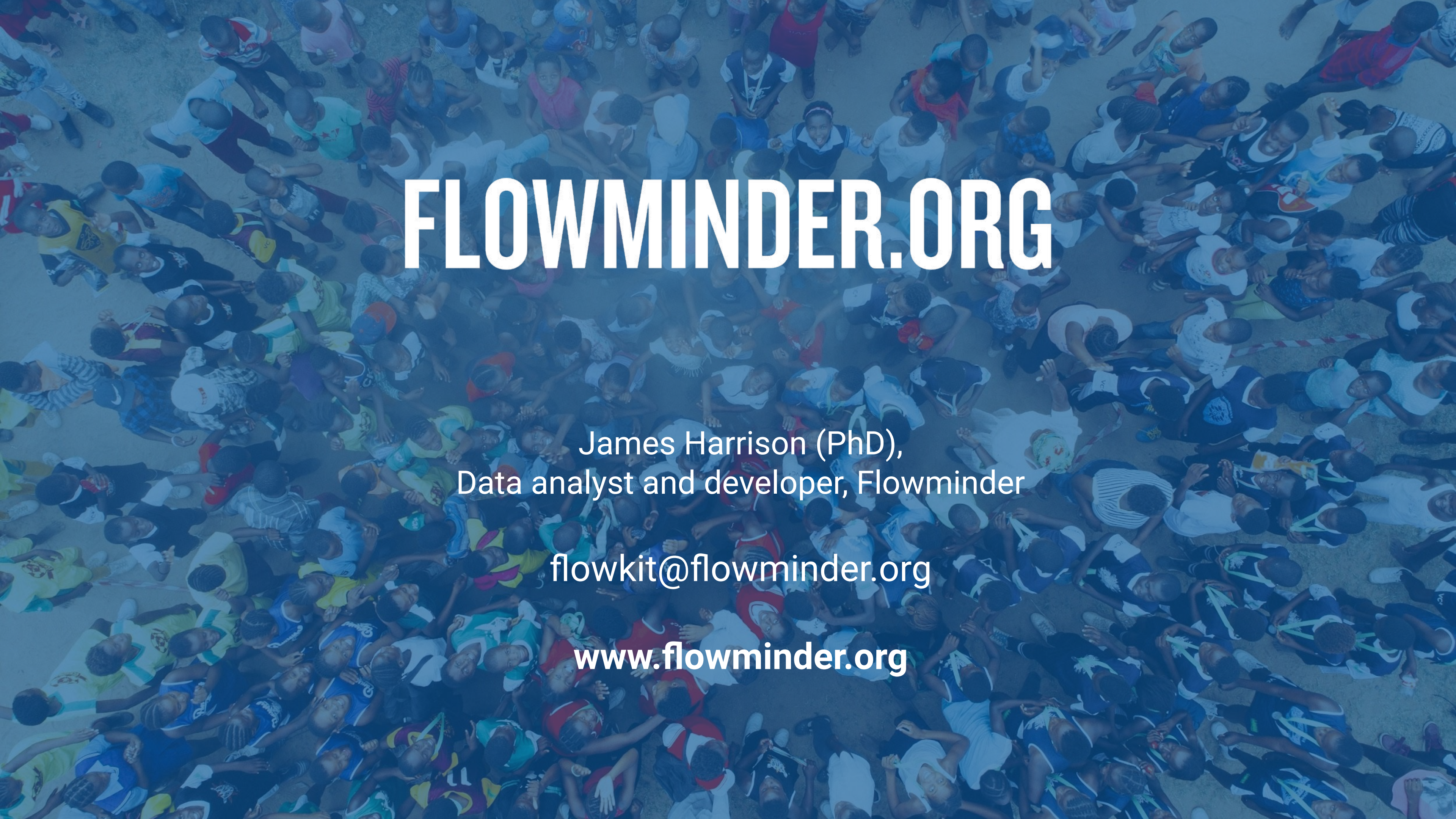
To visualise our results on a map, we need the geographic boundaries of the districts. We can get these from FlowKit using the `get_geography` function.

```
[15]: admin3_geojson = flowclient.get_geography(connection=conn, aggregation_unit="admin3")
      GeoJSON(admin3_geojson)
```

# Get geography data



# Visualise results



# FLOWMINDER.ORG

James Harrison (PhD),
Data analyst and developer, Flowminder

flowkit@flowminder.org

**www.flowminder.org**