Utrecht University

How to write Python scripts for PORT

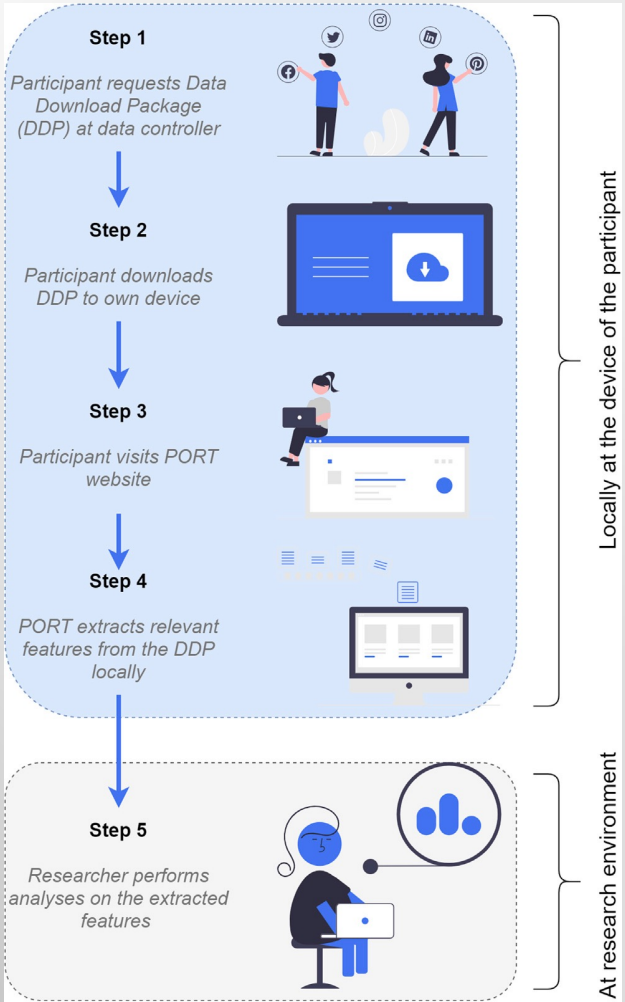
Dr. Haili Hu
Research Engineering team
Utrecht Universiy

Research Engineering

Overview

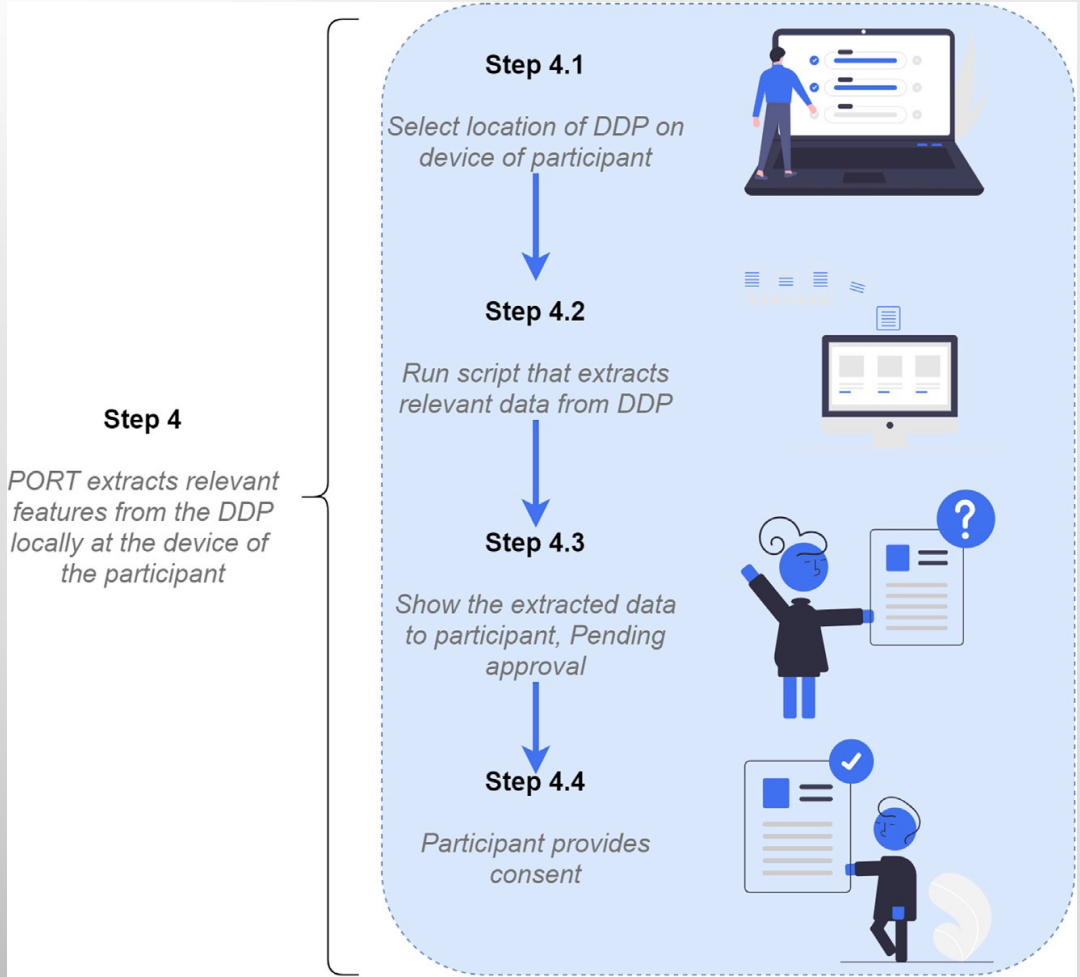
- Requirements of data extraction script
- Writing your script
- Testing your script
- Example: Google Semantic Location History

PORT Workflow



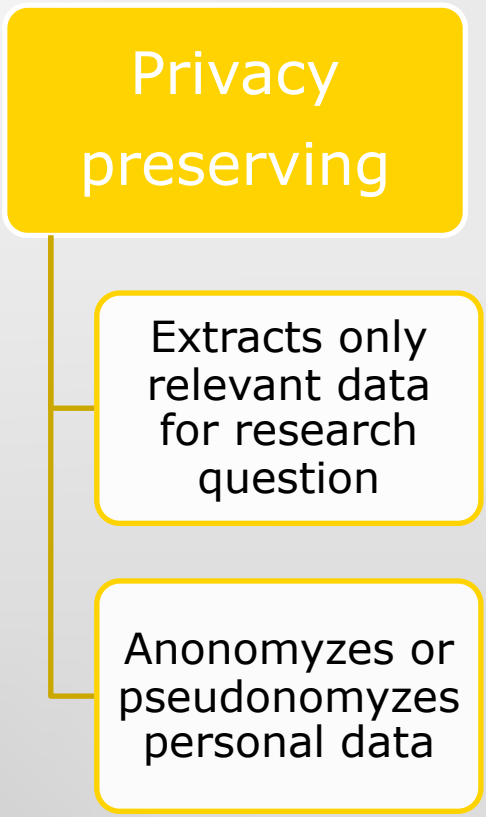
Boeschoten et al. 2022

PORT Data Extraction



Boeschoten et al. 2022

Requirements of data extraction script



Requirements of data extraction script

Technical

Written in **Python:**

- Support for **standard Python libs**
- Support for **Pandas**

Must have function called "**process**"

- Takes as input a **File Object**

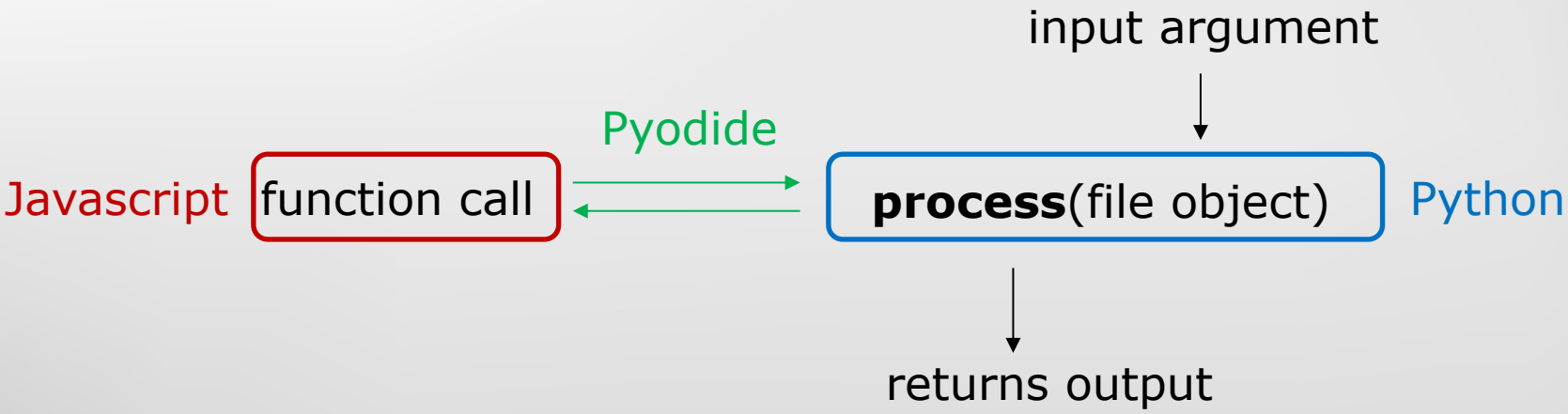
Returns as output an **Array of Dictionaries,** each containing the following keys:

- **title** (string)
- **data_frame** (Pandas DataFrame)

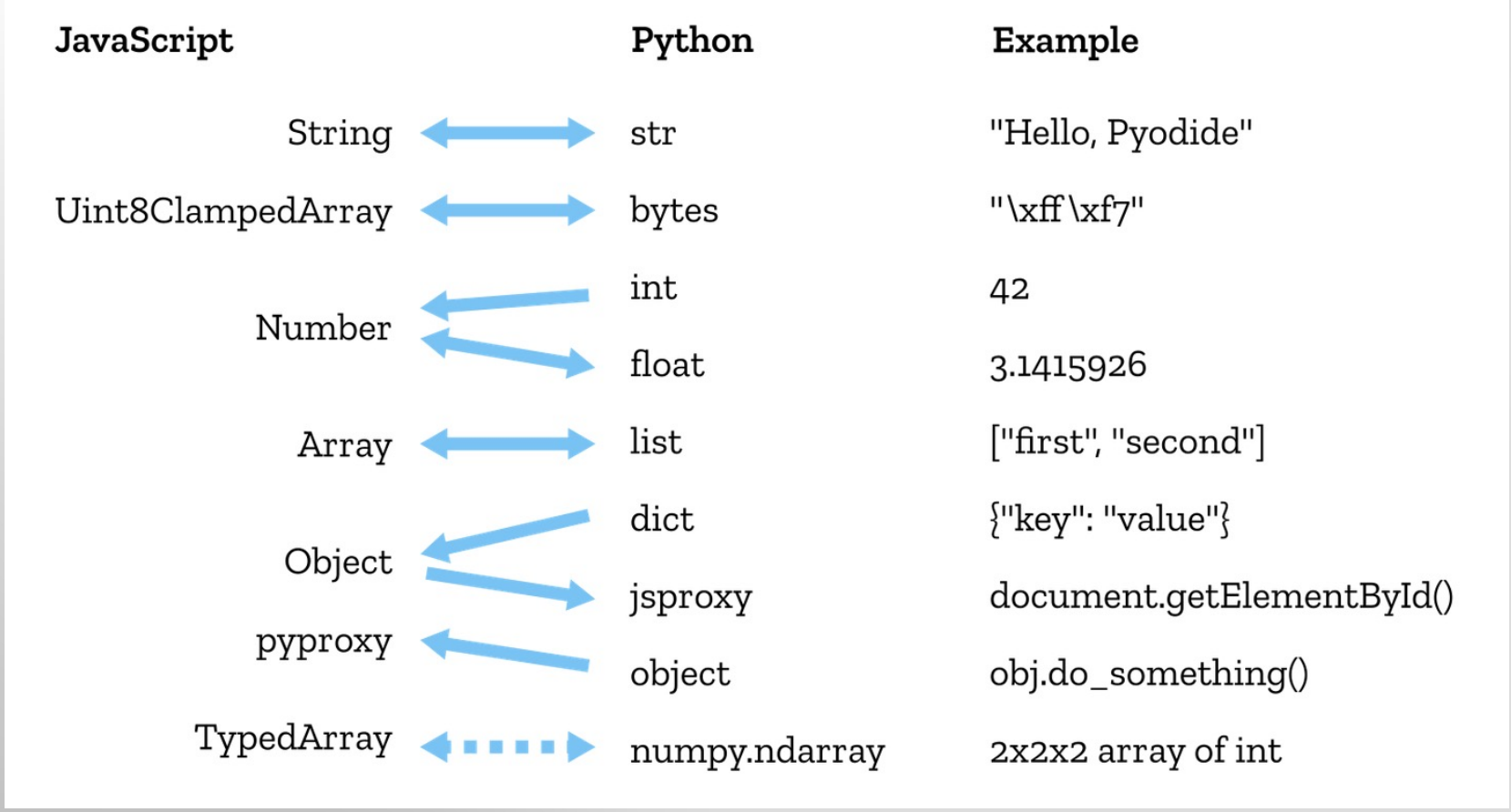
Requirements of data extraction script

```python
import zipfile
import pandas as pd

def process(file_data):
    names = []
    zfile = zipfile.ZipFile(file_data)
    data = []
    for name in zfile.namelist():
        names.append(name)
        info = zfile.getinfo(name)
        data.append((name, info.compress_size, info.file_size))

    return [
        {
            "title": "Files in the ZIP",
            "data_frame": pd.DataFrame(
                data, columns=["filename", "compressed size", "size"])
        }
    ]
```

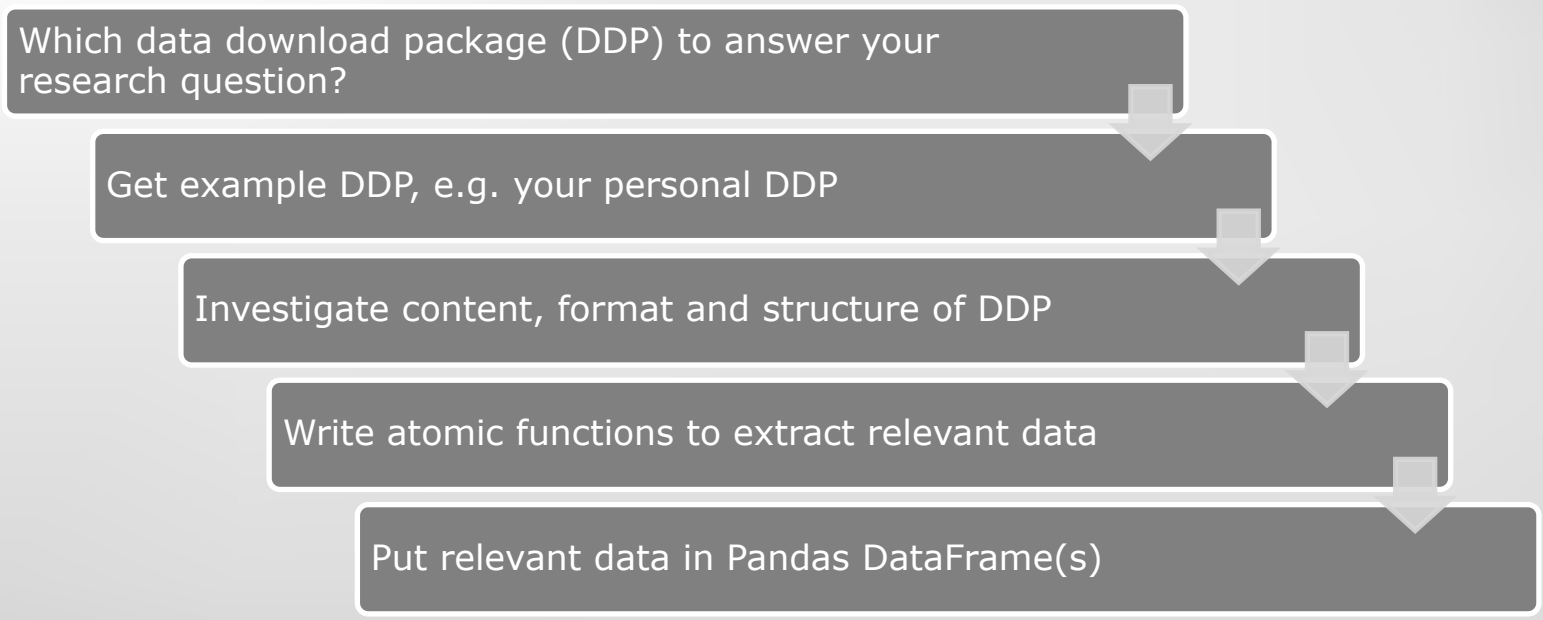
Requirements of data extraction script



Pyodide converts Python to WebAssembly -> bridge between Javascript and Python



Writing your script



Testing your script

WRITE TEST PROGRAM THAT CALLS THE “PROCESS” FUNCTION

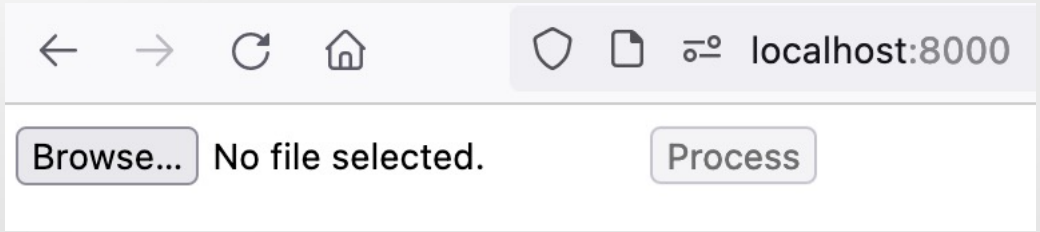
WRITE UNIT TEST FOR EACH FUNCTION, E.G. WITH PYTEST

CREATE FAKE DATA FOR YOUR TESTS, E.G. WITH FAKER

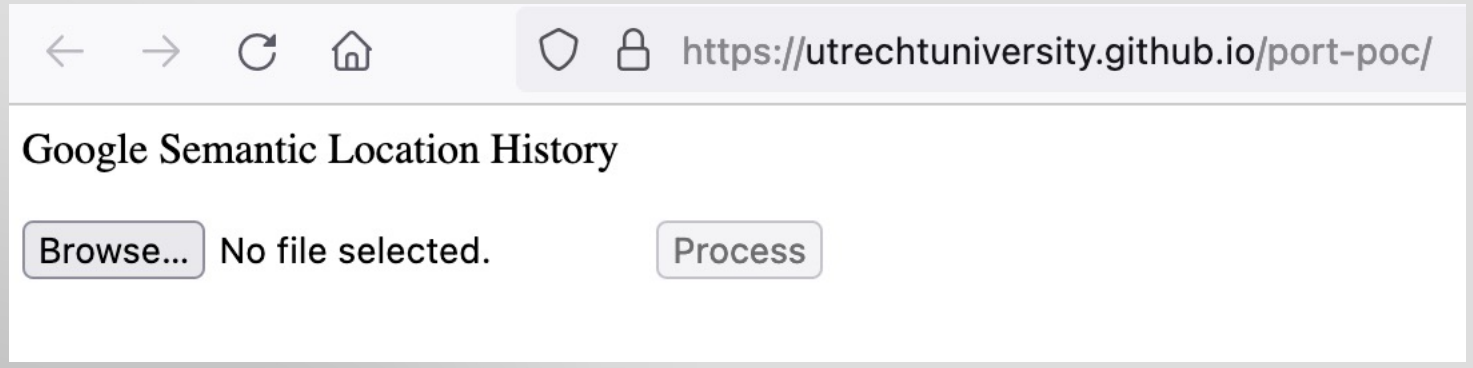
TEST SCRIPT ON DDPS FROM OTHER PEOPLE

Testing your script

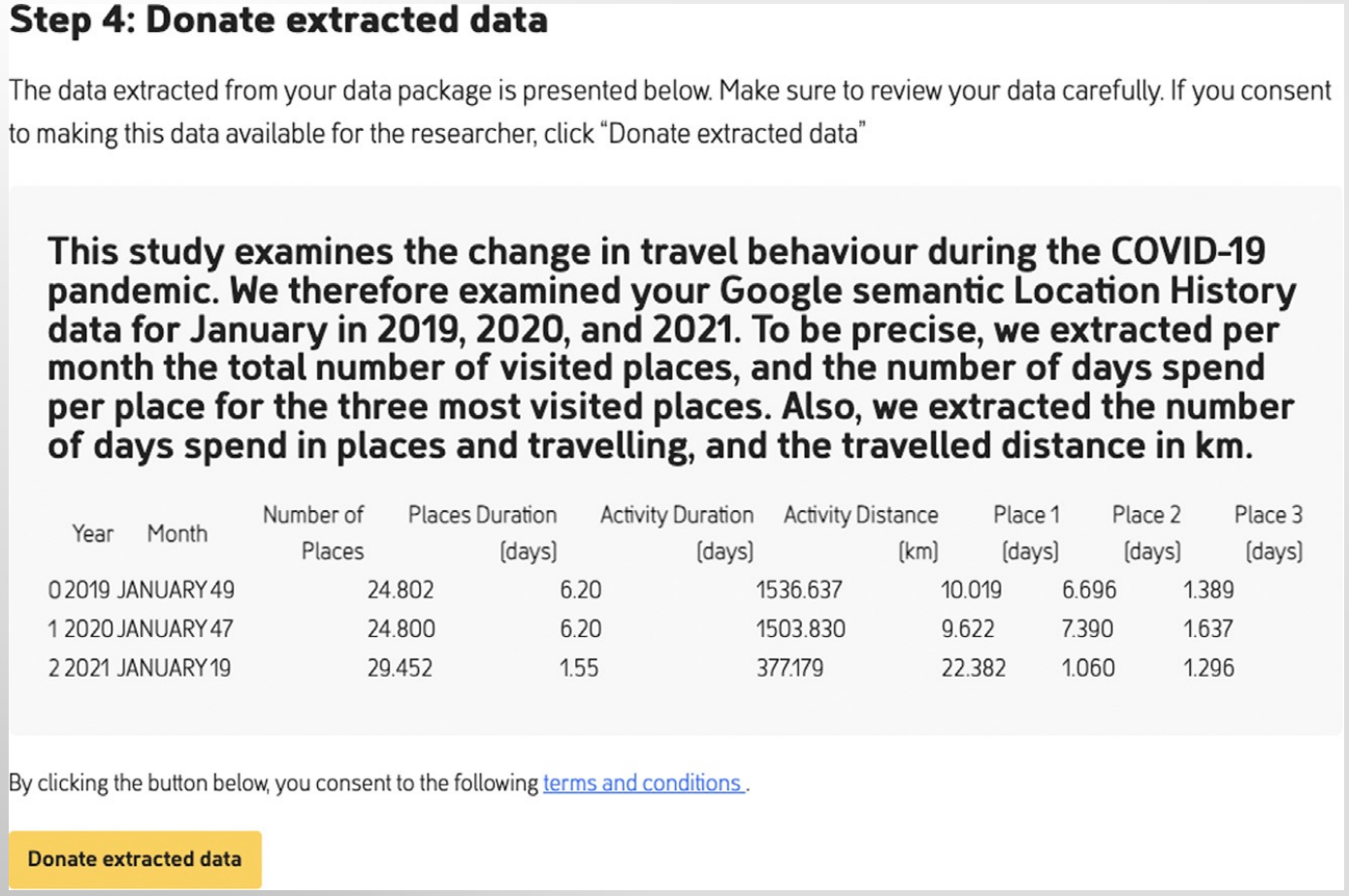
- Test locally with PORT-POC (github.com/eyra/port-poc):



- Test online with GitHub Pages:



Example: Google Semantic Location History



Step 4: Donate extracted data

The data extracted from your data package is presented below. Make sure to review your data carefully. If you consent to making this data available for the researcher, click "Donate extracted data"

This study examines the change in travel behaviour during the COVID-19 pandemic. We therefore examined your Google semantic Location History data for January in 2019, 2020, and 2021. To be precise, we extracted per month the total number of visited places, and the number of days spend per place for the three most visited places. Also, we extracted the number of days spend in places and travelling, and the travelled distance in km.

	Year	Month	Number of Places	Places Duration (days)	Activity Duration (days)	Activity Distance (km)	Place 1 (days)	Place 2 (days)	Place 3 (days)
0	2019	JANUARY	49	24.802	6.20	1536.637	10.019	6.696	1.389
1	2020	JANUARY	47	24.800	6.20	1503.830	9.622	7.390	1.637
2	2021	JANUARY	19	29.452	1.55	377.179	22.382	1.060	1.296

By clicking the button below, you consent to the following terms and conditions .

Donate extracted data

Boeschoten et al. 2022

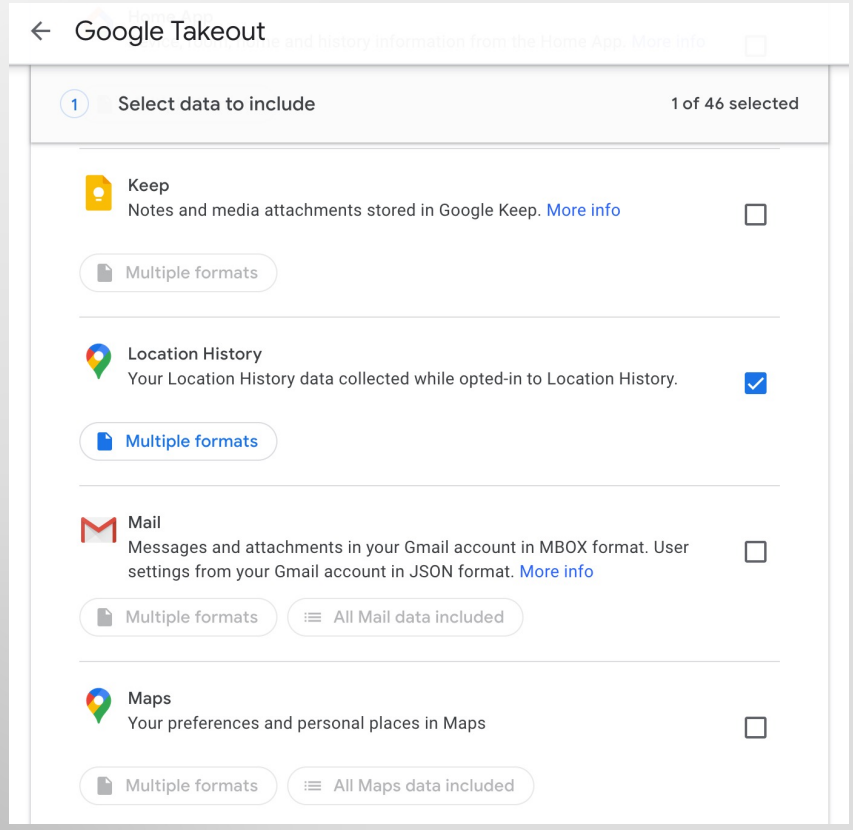
Which data download package (DDP) to answer your research question?

Research question: *How does travel behavior change in times of a Covid-19 lockdown?*

DDP: *Google Semantic Location History, Google's interpretation of your location data, with inferred place visits and activity segments*

Get example DDP, e.g. your personal DDP

Go to Google Takeout and request Location History:



Investigate format and structure of DDP

- A ZIP file containing monthly JSON files
- The JSON files have a complex nested structure, with *activivitySegment* describing journeys and *placeVisit* describing places


```
{
    "timelineObjects": [{
        "activitySegment": {
            "startLocation": {
                "latitudeE7": 521471860,
                "longitudeE7": 56329130
            },
            "endLocation": {
                "latitudeE7": 521561930,
                "longitudeE7": 521561930
            },
            "duration": {
                "startTimestampMs": "1609465777920",
                "endTimestampMs": "1609466313600",
                "activityType": "WALKING"
            },
            "distance": 3091,
            "activityType": "KNxDHIcTwQsRGbsUlqVP",
            "confidence": "lNLcMuyBupHtMdagvTKB",
            "activities": [{
                "activityType": "nZakdIjrWuWtUVXDdINR",
                "probability": -143797059664.65
            }],
```

```
"placeVisit": {
    "location": {
        "latitudeE7": 521471860,
        "longitudeE7": 56329130,
        "placeId": "Z1-6789188e",
        "address": "Wesleyhof 0\n1864TT\nRijs",
        "name": "van den Assem & Pastoors",
        "sourceInfo": {
            "deviceTag": 5809
        },
        "locationConfidence": 124906.302276313,
        "semanticType": "FMwKBABEpJMnwHnHUJzr"
    },
    "duration": {
        "startTimestampMs": "1609455600000",
        "endTimestampMs": "1609465777920"
    },
```

Example: Google Semantic Location History

Write atomic functions to extract relevant data

For example, separate functions to calculate:

distance of activities

duration of activities

duration of visits

Example: Google Semantic Location History

Put relevant data in Pandas dataframes

- Use understandable column names
- Anonomyze personal data, e.g. replace names and addresses
- Also possible to put metadata in Dataframe, e.g. logging and errors, for development purposes

Possible challenges & solutions

- **Format and content of DDPs vary (in time, per person, etc.)**
  - *Do not make script too specific, generalize if possible*
  - *Test on a variety of DDPs from different people*
  - *Do a pilot study berore actual study*

- **Script crashes for unknown reasons**
  - *Include error handling*
  - *Gather metadata (logging, errors, etc.) in DataFrame and return as part of output*

- **Python script works locally but not in PORT platform**
  - *Test first in PORT-POC (Javascript and Pyodide)*

Possible challenges & solutions

- **A custom Python script needed for every study**
  - *Share all scripts used in PORT (Open Science)?*
  - *Library of common functions researchers can reuse (FAIR)?*

- **Researcher's programming skills are not adequate**
  - *Documentation on using PORT needed: clear technical requirements and instructions for researchers*
  - *Follow training (level of Python depends on complexity of DDP)*
  - *Work together with a research engineer* 😀