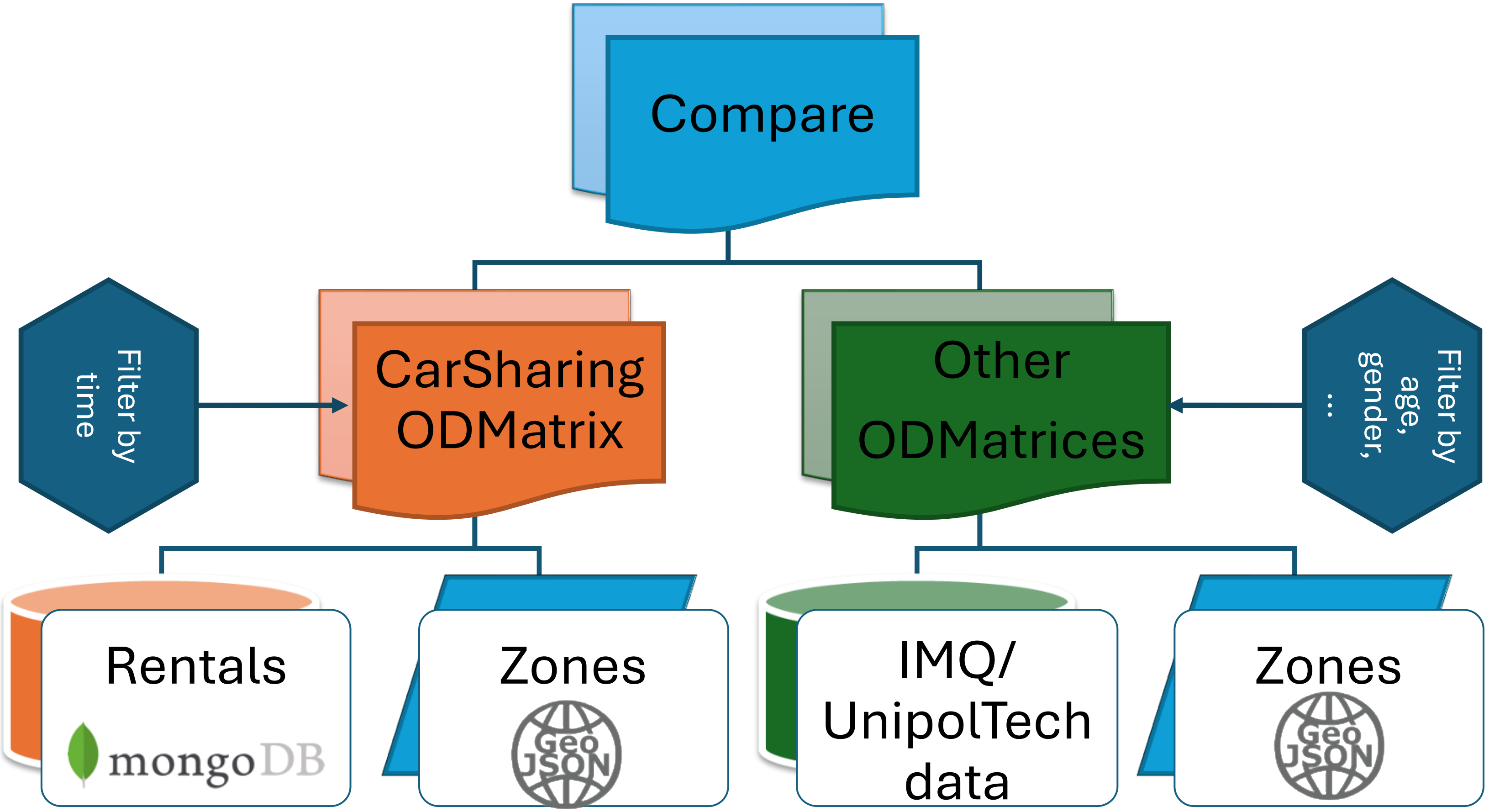
LAB 2
OD Matrices analytics

Goal

- This laboratory builds on the data analyzed during the first lab
- We want to find out which are the most likely class of users of car sharing system
- For this, we compare the OD matrices obtained from different sources
- Use metadata to filter
  - Based on gender, age, goals of trip, etc.
- And obtain the most similar OD matrix as the one from car sharing users

Overall view

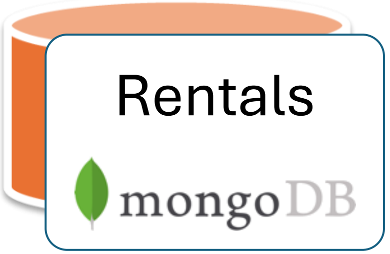


Datasets

3 Datasets

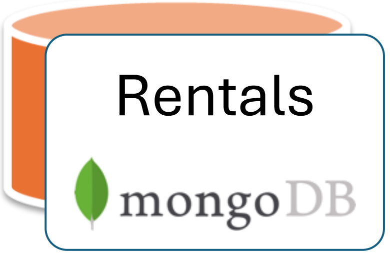
- Car Sharing rentals to obtain OD matrix of car sharing users
  - Car rentals for Enjoy and Car2Go
  - 2 months of data, with Origin/Destination (indexed for running geospatial queries)
  - Stored on MongoDB
- Open data from IMQ 2013 (Indagine Mobilità e Qualità)
  - https://mtm.torino.it/it/dati-statistiche/indagini/matrici-od-imq-2013/
- Data from UnipolTech collected in 2024
- DO NOT SHARE carsharing and UnipolTech data

Carsharing dataset



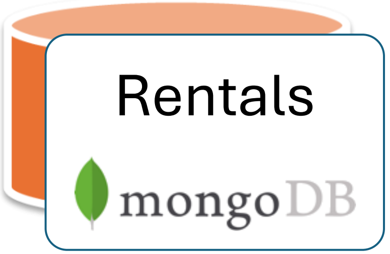
- Rentals are available on MongoDB
  - **ictts_PermanentBookings**
  - **ictts_enjoy_PermanentBookings**
- There are 3 indexes
  - `init_loc` and `final_loc` with 2D coordinates to support geospatial queries
  - `init_time` for filtering over time

Extracting data from rentals



- We can filter
  - By time
  - By position
- Useful **time operators** in `$aggregate` pipeline
  - `$hours` – extract the hour in UTC from a ISODate
    - E.g., filter on rentals between midnight and midday
  - `$dayOfWeek` – Returns the day of the week for a date as a number
    - 1 (Sunday) and 7 (Saturday)
    - E.g., filter on rentals on Monday, Tuesday,..., Friday only

Example



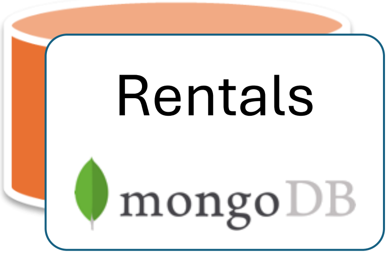

```
pb_coll.aggregate([
      {'$project’: {
          [redacted]
          'init_loc': 1,
          'final_loc’: 1
      }},
      {'$match’: {
          [redacted]
      }},
      {'$count': 'tot’}
  ])
```

Transform the init_date
to get the hour and day

Filter on morning
and weekdays (1=Sunday, ...)

Then count

Example




```
pb_coll.aggregate([
      {'$project’: {
          'hour': {'$hour': '$init_date’},
          'day': {'$dayOfWeek': '$init_date’},
          'init_loc': 1,
          'final_loc’: 1
      }},
      {'$match’: {
          'hour': {'$gte': 0, '$lt':12},
          'day': {'$gte': 2, '$lt':7},
      }},
      {'$count': 'tot’}
  ])
```

Transform the init_date to get the hour and day

Filter on morning and weekdays (1=Sunday, ...)

Then count

Geospatial queries

- MongoDB's geospatial indexing allows you to efficiently execute spatial queries on a collection that contains geospatial shapes and points
- The geometry data in the location field must follow the **GeoJSON format**

```
<field>: { type: <GeoJSON type> , coordinates: <coordinates> }
```

- `type` can be
  - Point: `{type:"Point",coordinates:[40,5]}`
  - Linestring: `{type:"LineString",coordinates:[[40,5],[41,6]]}`
  - Polygon: `{type:"Polygon",coordinates:[[[0,0],[3,6],[6,1],[0,0]]]}`
  - …

Geospatial queries

- MongoDB support geopatial indexes that allow efficient queries
- A `2dsphere` index supports queries that calculate geometries on an earth-like sphere

Possible queries

- `$geoWithin`: query for location data found within a GeoJSON polygon
  - Location data must be stored in GeoJSON format

```
<collection>.find( { <location field>.coordinates :
{ '$geoWithin' :
  { '$geometry' :
   { 'type' : 'Polygon',
     'coordinates' : [ <coordinates> ]
}}}})
```

Geospatial queries

Other possible queries:

- `$geoIntersects` to select all indexed points and shapes that intersect with the polygon defined by the coordinates array
- `$near` operator or `$geoNear` return the points closest to the defined point and sorts the results by distance

```
<collection>.find( { <location field> :
 { '$near' :
  {'$geometry' :
   {'type' : 'Point' ,
    'coordinates' : [ <longitude> , <latitude> ]
   },
   '$maxDistance' : <distance in meters>
}}})
```

Example of geospatial query

- Find all cars available in a region of 1000m from Politecnico di Torino entrance

```
pb_coll.find({
     'init_loc’: {
          '$near’: {
               '$geometry’: {
                    'type': 'Point’,
                    'coordinates': [7.660, 45.0645]
               },
               ‘$maxDistance': 1000,
          }
     }
})
```

Warning
Cost of geoqueries is not negligible

Extracting data from rentals

- Useful **geospatial operators** in `$aggregate` pipeline
  - `$geoWithin`: data that exists entirely within a specified shape documents with geospatial
  - The shape can be
    - GeoJSON Polygon
    - GeoJSON MultiPolygon (Polygon with "holes")
  - The `$geoWithin` operator uses the `$geometry` operator to specify the GeoJSON object

```
res = pb_coll.aggregate([
    {'$match': {
      'init_loc': {
        '$geoWithin': {
          '$geometry': {
            'type': 'Polygon',
            'coordinates':[…]

    }}}}}])
```

Extracting the O/D matrix

ICT4SS
Information and Communications Technologies for Smart Societies

Rentals
mongoDB

```
orig_zone = …  # put here the arrays describing the zone
dest_zone = … # put here the arrays describing the zone
pb_coll.aggregate([
        { '$match': {
                'init_loc': {'$geoWithin' :
                        {'$geometry': {
                                'type': 'MultiPolygon',
                                'coordinates': orig_zone
                                }
                        }
                },
                'final_loc': {'$geoWithin' :
                        {'$geometry': {
                                'type': 'MultiPolygon',
                                'coordinates': dest_zone
                                }
                        }
                }
        }
        },
        {'$count': 'tot'
        }
  ])
```

Filter on init_loc that falls within the orig_zone

and final_loc that falls within the dest_zone

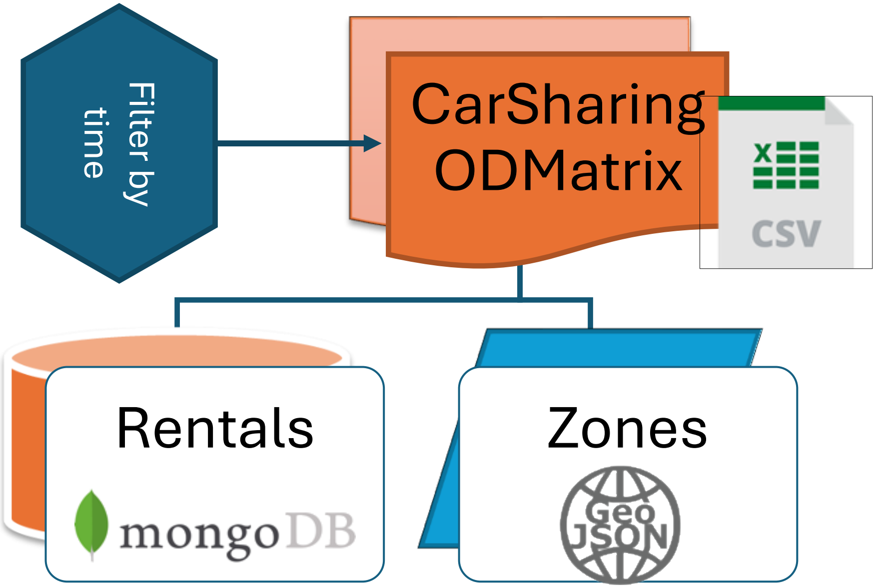
Then count

ICT4SS
Information and Communications Technologies for Smart Societies

Putting everything together

Prepare a script with the program to run

- For each orig_zone
  - For each dest_zone
    - Count rentals in ODMatrix
      - From orig_zone
      - To dest_zone
      - In a given timeslot
- Derive different ODMatrixes
  - For Enjoy, Car2Go
  - Considering
    - All rentals
    - Rentals in weekends or weekdays only
    - Mornings and afternoons
  - …
- Save in a CSV file



UnipolTech dataset

- Data collected by company – Anonymized
  - One week in September 2024 in Torino
- Data in 2 .csv files, link them by vehicle ID (`id_veicolo`)
- Information of users/vehicles (`Info_TO.csv`)
  - In total about 38k users
  - Gender, age, whether for commercial use
- Information of trips (`Trips_OD_TO.csv`)
  - Origin/Destination location (latitude and longitude)
  - Departure/arrival time
  - In total 545k trip records

Dataset – IMQ

- IMQ 2013 contains data collected with phone interviews
  - 52,120 completed interviews
  - 105,099 trips reported
  - Trips from Monday to Friday
  - Covers 185 zones of Piedmont
    - **23 zones in Torino, we will use them**
- Interviews are stratified by
  - Gender (male/female)
  - Age (11-19, 20-49, 50-64, 65+)
  - Motivation (go to work, got to school, visiting parents, ...)

Dataset – IMQ

- 183 zones
- Defined in a shape file
  - Resource: **Zone_IMQ_TO.rar**
- Available as geojson shape
  - Resource: **zoneBeauty.geojson**
- 23 zones in Torino
  - Resource: **TorinoZonescol.geojson**
  - Resource: **TorinoZonesArray.geojson**

Q001	TORINO - CENTRO
Q002	TORINO - S.SALVARIO
Q003	TORINO - CROCETTA
Q004	TORINO - S.PAOLO
Q005	TORINO - CENISIA
Q006	TORINO - S.DONATO
Q007	TORINO - AURORA
Q008	TORINO - VANCHIGLIA
Q009	TORINO - NIZZA-MILLEFONTI
Q010	TORINO - LINGOTTO
Q011	TORINO - S.RITA
Q012	TORINO - MIRAFIORI NORD
Q013	TORINO - POZZO STRADA
Q014	TORINO - PARELLA
Q015	TORINO - VALLETTE
Q016	TORINO - MADONNA CAMPAGNA
Q017	TORINO - B.TA VITTORIA
Q018	TORINO - B.RA MILANO
Q019	TORINO - FALCHERA
Q020	TORINO - REGIO PARCO
Q021	TORINO - MADONNA PILONE
Q022	TORINO - CAVORETTO
Q023	TORINO - MIRAFIORI SUD

IMQ data

- Phone interviews
  - From Tuesday to Saturday - 9.30 and 21.30
- Asked about movements done the day before
  - **Only obtained trips during weekdays**
  - Hour, departure place, arrival place, reason for the trip, modes,…
  - http://mtm.torino.it/it/dati-statistiche/imq-alle-fasi-finali
  - Sampling specifications:
    - 3% in the Turin metropolitanea area
    - At least 60 interviews per zone
- Results were available as open data
  - http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/base-dati-imq-2013 [not available anymore]

IMQ data

- Raw data in Microsoft Access from http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/base-dati-imq-2013/IMQ2013_opendata.zip [not available anymore]
- There are several tables
  - Spostamenti: information about the movements done in the previous day
  - Interviste: information about the person
  - Tab_*: tables explaining the mapping for each category
- We have extracted the subset of movements done in the Torino area
  - From zone Q* to zone Q*
  - 16,567 trips in total

IMQ data sample

ID_INT	PROGR_USC	PROGR_SPOST	SESSO	FASCIA_ETA	ZONA_RES	SCOPO	ZONA_PAR	PROV_PAR	ORA_PAR	ZONA_ARR	PROV_ARR	ORA_ARR
8139335	1	2	1	2	C006	5	Q005	1	02/02/17 10:00	Q023	1	02/02/17 10:20
8139641	1	2	2	4	C009	4	Q002	1	02/02/17 16:30	Q023	1	02/02/17 16:50
8139830	1	2	1	2	C004	4	Q019	1	02/02/17 14:30	Q019	1	02/02/17 14:50
8140616	1	2	1	2	C003	9	Q003	1	02/02/17 20:00	Q005	1	02/02/17 20:30
8140678	1	2	2	3	C011	4	Q009	1	02/02/17 09:00	Q009	1	02/02/17 09:10
8140745	1	2	2	3	C004	4	Q015	1	02/02/17 18:00	Q021	1	02/02/17 18:15
8140799	1	2	2	3	C006	4	Q010	1	02/02/17 10:45	Q020	1	02/02/17 11:00
8141314	1	2	2	3	C009	4	Q022	1	02/02/17 18:15	Q022	1	02/02/17 18:30
8141976	2	2	1	4	C003	9	Q001	1	02/02/17 15:40	Q007	1	02/02/17 16:10
8142800	1	2	2	2	C003	7	Q012	1	02/02/17 19:30	Q019	1	02/02/17 19:45
8143087	1	2	2	3	C003	4	Q006	1	02/02/17 15:00	Q006	1	02/02/17 15:10
8143375	1	2	1	4	C004	9	Q001	1	02/02/17 17:00	Q018	1	02/02/17 17:15
8146677	2	2	2	3	C023	7	Q008	1	02/02/17 19:20	Q002	1	02/02/17 19:40
8147093	1	2	1	1	C016	2	Q001	1	02/02/17 14:00	Q014	1	02/02/17 14:35

ICT4SS
Information and Communications Technologies for Smart Societies

IMQ data sample

ID_INT	PROGR_USC	PROGR_SPOST	SESSO	FASCIA_ETA	ZONA_RES	SCOPO	ZONA_PAR	PROV_PAR	ORA_PAR	ZONA_ARR	PROV_ARR	ORA_ARR
8139335	1	2	1	2	C006	5	Q005	1	02/02/17 10:00	Q023	1	02/02/17 10:20
8139641	1	2	2	4	C009	4	Q002	1	02/02/17 16:30	Q023	1	02/02/17 16:50
8139830	1	2	1	2							1	02/02/17 14:50
8140616	1	2	1	2							1	02/02/17 20:30
8140678	1	2	2	3							1	02/02/17 09:10
8140745	1	2	2	3	C004	4	Q015	1	02/02/17 18:00	Q021	1	02/02/17 18:15
8140799	1	2	2	3	C006	4	Q010	1	02/02/17 10:45	Q020	1	02/02/17 11:00
8141314	1	2	2	3	C009	4	Q022	1	02/02/17 18:15	Q022	1	02/02/17 18:30
8141976	2	2	1	4	C003	9	Q001	1	02/02/17 15:40	Q007	1	02/02/17 16:10
8142800	1	2	2	2	C003	7	Q012	1	02/02/17 19:30	Q019	1	02/02/17 19:45
8143087	1	2	2	3	C003	4	Q006	1	02/02/17 15:00	Q006	1	02/02/17 15:10
8143375	1	2	1	4	C004	9	Q001	1	02/02/17 17:00	Q018	1	02/02/17 17:15
8146677	2	2	2	3	C023	7	Q008	1	02/02/17 19:20	Q002	1	02/02/17 19:40
8147093	1	2	1	1	C016	2	Q001	1	02/02/17 14:00	Q014	1	02/02/17 14:35

Gender	
1	Male
2	Female

ICT4SS
Information and Communications Technologies for Smart Societies

IMQ data sample

ID_INT	PROGR_USC	PROGR_SPOST	SESSO	FASCIA_ETA	ZONA_RES	SCOPO	ZONA_PAR	PROV_PAR	ORA_PAR	ZONA_ARR	PROV_ARR	ORA_ARR
8139335	1	2	1	2	C006	5	Q005	1	02/02/17 10:00	Q023	1	02/02/17 10:20
8139641	1	2	2	4	C009	4	Q002	1	02/02/17 16:30	Q023	1	02/02/17 16:50
8139830	1	2	1	2	C						1	02/02/17 14:50
8140616	1	2	1	2	C						1	02/02/17 20:30
8140678	1	2	2	3	C						1	02/02/17 09:10
8140745	1	2	2	3	C						1	02/02/17 18:15
8140799	1	2	2	3	C						1	02/02/17 11:00
8141314	1	2	2	3	C009	4	Q022	1	02/02/17 18:15	Q022	1	02/02/17 18:30
8141976	2	2	1	4	C003	9	Q001	1	02/02/17 15:40	Q007	1	02/02/17 16:10
8142800	1	2	2	2	C003	7	Q012	1	02/02/17 19:30	Q019	1	02/02/17 19:45
8143087	1	2	2	3	C003	4	Q006	1	02/02/17 15:00	Q006	1	02/02/17 15:10
8143375	1	2	1	4	C004	9	Q001	1	02/02/17 17:00	Q018	1	02/02/17 17:15
8146677	2	2	2	3	C023	7	Q008	1	02/02/17 19:20	Q002	1	02/02/17 19:40
8147093	1	2	1	1	C016	2	Q001	1	02/02/17 14:00	Q014	1	02/02/17 14:35

Age	
1	From 11 to 19
2	From 20 to 49
3	From 50 to 64
4	65+

ICT4SS
Information and Communications Technologies for Smart Societies

IMQ data sample

ID_INT	PROGR_USC	PROGR_SPOST	SESSO	FASCIA_ETA	ZONA_RES	SCOPO	ZON					
8139335	1	2	1	2	C006	5	Q					
8139641	1	2	2	4	C009	4	Q					
8139830	1	2	1	2	C004	4	Q					
8140616	1	2	1	2	C003	9	Q					
8140678	1	2	2	3	C011	4	Q					
8140745	1	2	2	3	C004	4	Q					
8140799	1	2	2	3	C006	4	Q					
8141314	1	2	2	3	C009	4	Q					
8141976	2	2	1	4	C003	9	Q					
8142800	1	2	2	2	C003	7	Q					
8143087	1	2	2	3	C003	4	Q					
8143375	1	2	1	4	C004	9	Q001	1	02/02/17 17:00	Q018	1	02/02/17 17:15
8146677	2	2	2	3	C023	7	Q008	1	02/02/17 19:20	Q002	1	02/02/17 19:40
8147093	1	2	1	1	C016	2	Q001	1	02/02/17 14:00	Q014	1	02/02/17 14:35

Motivation	
1	Go to work
2	Working reason
3	Study
4	Shopping
5	Bring someone
6	Cures or medical visits
7	Sport or leisure
8	Going back home
9	Visiting relatives or friends
10	Other
11	Going back home on the day of the interview

IMQ data sample

ID_INT	PROGR_USC	PROGR_SPOST	SESSO	FASCIA_ETA	ZONA_RES	SCOPO	ZONA_PAR	PROV_PAR	ORA_P
8139335	1	2	1	2	C006	5	Q005	1	02/02/17
8139641	1	2	2	4	C009	4	Q002	1	02/02/17
8139830	1	2	1	2	C004	4	Q019	1	02/02/17
8140616	1	2	1	2	C003	9	Q003	1	02/02/17
8140678	1	2	2	3	C011	4	Q009	1	02/02/17
8140745	1	2	2	3	C004	4	Q015	1	02/02/17
8140799	1	2	2	3	C006	4	Q010	1	02/02/17
8141314	1	2	2	3	C009	4	Q022	1	02/02/17
8141976	2	2	1	4	C003	9	Q001	1	02/02/17
8142800	1	2	2	2	C003	7	Q012	1	02/02/17
8143087	1	2	2	3	C003	4	Q006	1	02/02/17
8143375	1	2	1	4	C004	9	Q001	1	02/02/17
8146677	2	2	2	3	C023	7	Q008	1	02/02/17
8147093	1	2	1	1	C016	2	Q001	1	02/02/17

ZONE	
Q001	TORINO - CENTRO
Q002	TORINO - S.SALVARIO
Q003	TORINO - CROCETTA
Q004	TORINO - S.PAOLO
Q005	TORINO - CENISIA
Q006	TORINO - S.DONATO
Q007	TORINO - AURORA
Q008	TORINO - VANCHIGLIA
Q009	TORINO - NIZZA-MILLEFONTI
Q010	TORINO - LINGOTTO
Q011	TORINO - S.RITA
Q012	TORINO - MIRAFIORI NORD
Q013	TORINO - POZZO STRADA
Q014	TORINO - PARELLA
Q015	TORINO - VALLETTE
Q016	TORINO - MADONNA CAMPAGNA
Q017	TORINO - B.TA VITTORIA
Q018	TORINO - B.RA MILANO
Q019	TORINO - FALCHERA
Q020	TORINO - REGIO PARCO
Q021	TORINO - MADONNA PILONE
Q022	TORINO - CAVORETTO
Q023	TORINO - MIRAFIORI SUD

Zone Description

Extracting the zones

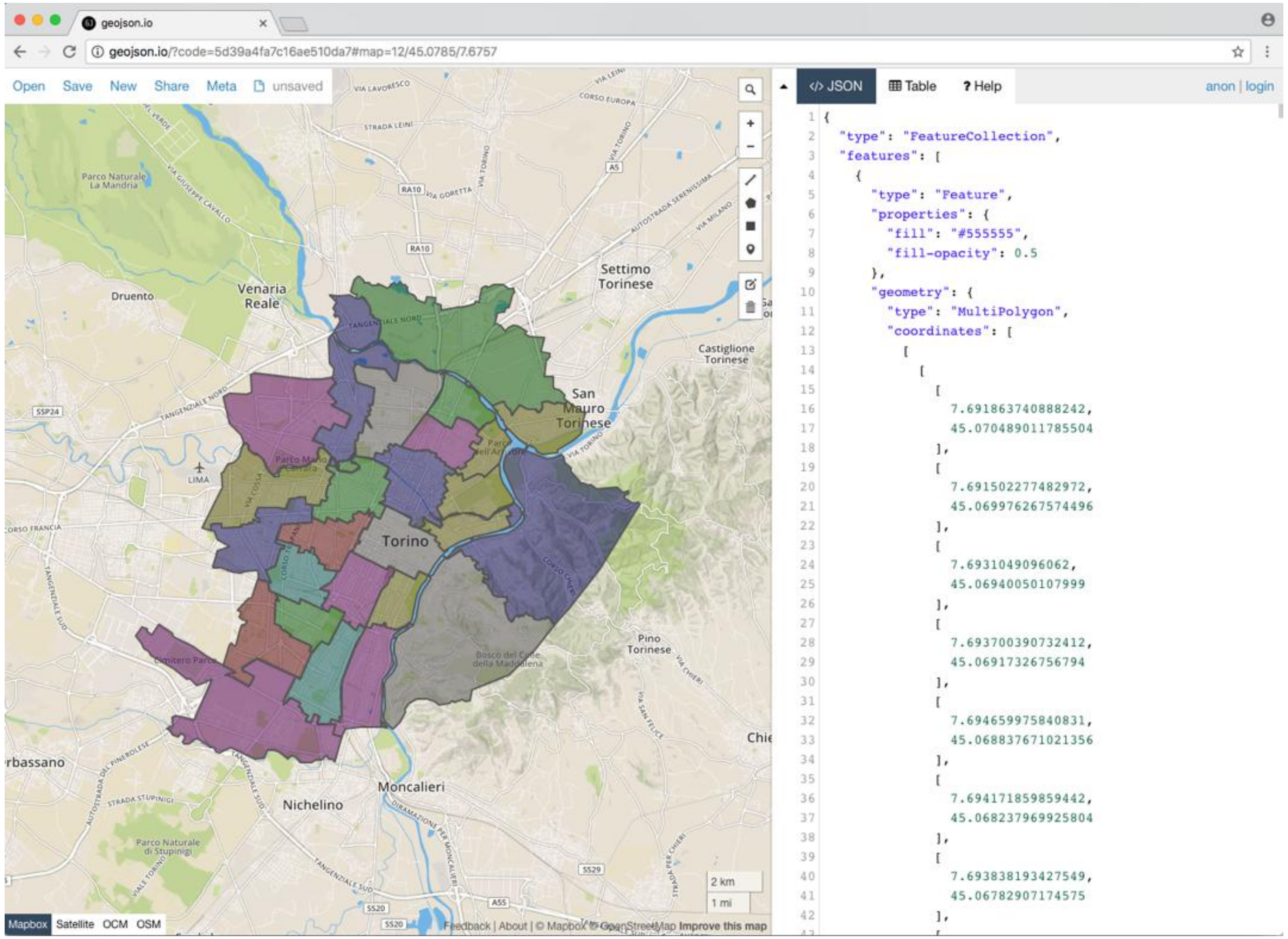
- There are 23 zones in Torino
  - `Q001 – Q023`
- These are defined in
  - **TorinoZonesArray.geojson**
    - An array ready for use
  - **TorinoZonescol.geojson**
    - Original geojson shape
    - Play with it in http://geojson.io/

Zones

Q001	TORINO - CENTRO
Q002	TORINO - S.SALVARIO
Q003	TORINO - CROCETTA
Q004	TORINO - S.PAOLO
Q005	TORINO - CENISIA
Q006	TORINO - S.DONATO
Q007	TORINO - AURORA
Q008	TORINO - VANCHIGLIA
Q009	TORINO - NIZZA-MILLEFONTI
Q010	TORINO - LINGOTTO
Q011	TORINO - S.RITA
Q012	TORINO - MIRAFIORI NORD
Q013	TORINO - POZZO STRADA
Q014	TORINO - PARELLA
Q015	TORINO - VALLETTE
Q016	TORINO - MADONNA CAMPAGNA
Q017	TORINO - B.TA VITTORIA

Zones in Torino

- Check the map at http://geojson.io/



Analyzing OD matrix

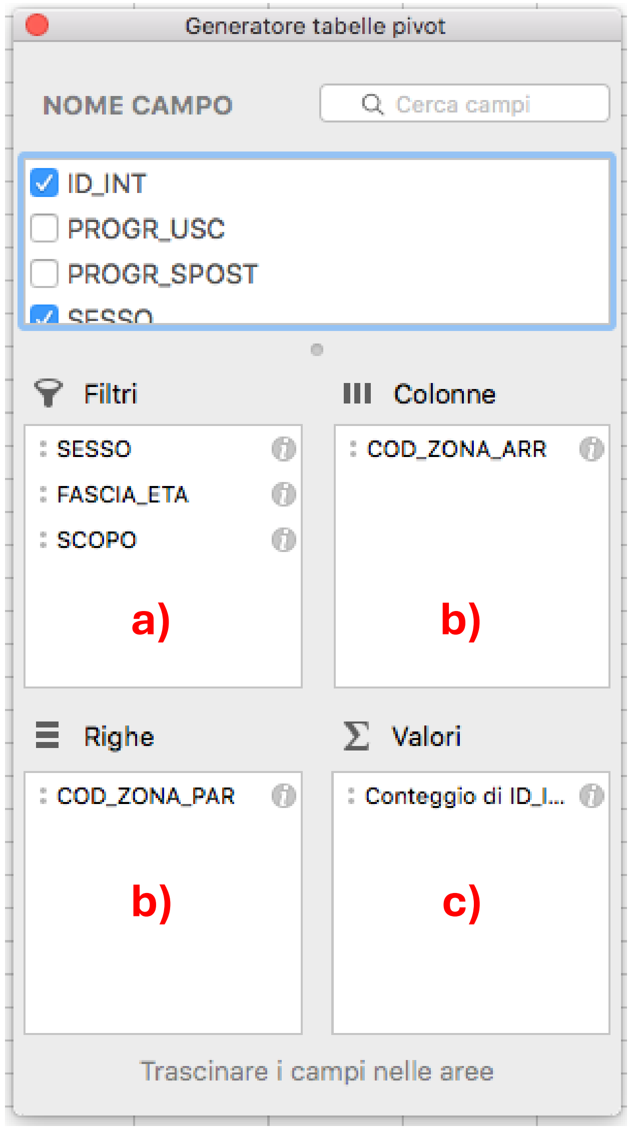
Goal: extract OD matrix

- OD matrix[i][j]: fraction of trips from zone i to zone j
- The simplest way to extract the OD matrix is using **pivot tables**
  - Pivot tables are simple ways to summarize data
  - They allow to process and transform data
  - And filter data
- You can do it with pandas dataframe
  - https://pandas.pydata.org/docs/reference/api/pandas.pivot_table.html

Create a pivot table with Excel

[Example in Excel – but any spreadsheet allow you to do it]

1. open a new spreadsheet and import the CVS file
2. Add a pivot table
   [menu->data->add pivot table]
3. Drag elements
   a) In filters: sesso, scopo, eta, …
   b) Rows and columns: zona_par, zona_arr
   c) Value: any field
      [we are going to ”count” occurrences]



Information and Communications Technologies for Smart Societies

Result

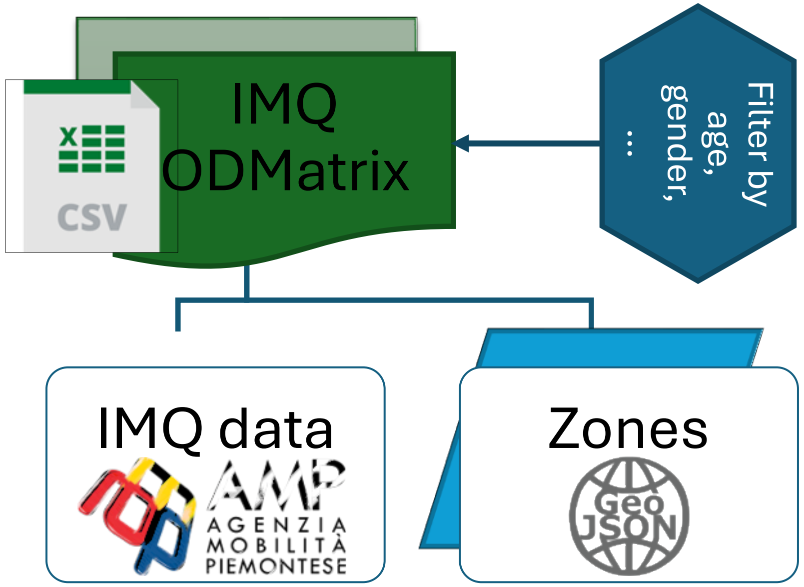
You can get different ODmatrixes by selecting different filters

FASCIA_ETA	(più elementi)
SCOPO	(più elementi)
SESSO	(Tutto)
ORA_ARR	(Tutto)

Conteggio di PR	Etichette di col																						
Etichette di r	Q001	Q002	Q003	Q004	Q005	Q006	Q007	Q008	Q009	Q010	Q011	Q012	Q013	Q014	Q015	Q016	Q017	Q018	Q019	Q020	Q021	Q022	Q023	Totale com
Q001	102	11	25	1	18	15	10	19	11	11	3	6	9	12	3	3	4	3	1	1	9	8	8	293
Q002	37	27	12	1	5	1	6	6	16	2	1	4	3	3	1	2	1	1	1	3	2	4	5	144
Q003	35	7	41	7	11	8	5	4	9	9	6	2	2	1	2		3		2			2	1	157
Q004	18	8	7	11	16	8	4	3	6	2	8	8	7	3	1	1	2	1			1		5	120
Q005	28	7	17	6	33	14	4	4	5	2	5	6	21	8	3		5		1			5	5	179
Q006	50	7	14	3	14	70	12	15	13	8	2	4	6	13	6	7	14		3	2	2	4	5	274
Q007	44	5	11	2	10	18	36	15	5	3	3	7	4	8	2	9	8	7	17	5	2	3	2	226
Q008	44	6	1	5	3	3	12	32	3	2	3	2	1	2	1	4	3	4	3	5	4	7		150
Q009	27	14	6	2	5	2	10	4	46	14	2	1	1	4	1	2	4	2	1	1	1	2	2	154
Q010	21	5	11	2	3	3	4	6	14	29	7	7	7	2		3	5		1	3	1	1	15	150
Q011	15	8	8	7	6	4	7	2	4	8	27	14	2	1	3	4	1			1			4	126
Q012	19	4	8	3	10	11	4	2	15	14	13	27	9	2		3	2	1	2		2	2	16	169
Q013	34	4	15	10	9	11	3	8	8	4	4	7	46	28	5	4	3		4	2	1	1	6	217
Q014	28	6	6	4	6	23	5	3	4	4	5	6	15	41	1	2	6		3	1			2	171
Q015	17	3	3	2	1	11	1	6	6	3	1	2	2	7	20	9	4		5		1			104
Q016	30	2	9	4	9	15	6	7	6	7		2	4	5	12	22	17	5	10	4	1		8	185
Q017	35	6	7	2	5	18	11	6	3	8	4	2	3	11	8	12	57	3	7	3	1		5	217
Q018	25	6	5	2	3	10	25	7	2	2	2	1	1	2	3	8	9	35	25	16	2	3	5	199
Q019	19	4	5	1	1	4	6	2	1			1		1	2	2	6	10	25	5	1		1	97
Q020	13	2	4			2	3	5	3						1	2	2	2	5	14			2	60
Q021	17	1		3	5	4	5	15	1	4		1	1		1	1	2	3	1		11	6	2	84
Q022	16	12	1		1	4	2	6	3	1	1	1	1	3	1					1	2	16	4	76
Q023	26	4	6	2	5	3	1	7	12	22	2	6	1	1			2	1	1	2	2	3	27	136
Totale compless	**700**	**159**	**222**	**80**	**179**	**262**	**182**	**184**	**196**	**159**	**99**	**117**	**146**	**158**	**77**	**100**	**160**	**78**	**118**	**69**	**46**	**67**	**130**	**3688**

Results

- You can compute different OD matrixes
  - For different age
  - For different sex
  - For different trip reason
  - …



Last step: evaluate similarity

- Given matrices $\mathbf{A}=[a_{ij}]$ and $\mathbf{B}=[b_{ij}]$ similarity can be defined as the sum of "distance" between each element

$$d_1(\mathbf{A},\mathbf{B}) = \sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij} - b_{ij}|$$

$$d_2(\mathbf{A},\mathbf{B}) = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} (a_{ij} - b_{ij})^2}$$

$$d_\infty(\mathbf{A},\mathbf{B}) = \max_{1\le i\le n}\max_{1\le j\le n} |a_{ij} - b_{ij}|$$

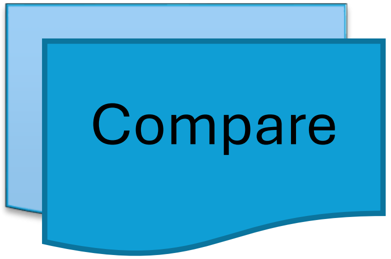
$$d_m(\mathbf{A},\mathbf{B}) = \max\{\|(\mathbf{A}-\mathbf{B})\mathbf{x}\| : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| = 1\}$$

Compare

Last step: compare similarity

To better appreciate the distance computation, compare

- Two uniform random matrices
  - $a_{ij} = rnd(0,1)$ *and* $b_{ij} = rnd(0,1)$
- Two OD matrices extracted from the same dataset
  - From car sharing data: A = OD(first week), B = OD(second week)
  - From car sharing data: A = OD(enjoy), B = OD(Car2Go)
  - From IMQ: A= OD(all), B = OD(males), …
- A uniform random matrix and a matrix from the data



Warning

1. Normalize elements, e.g., $\Sigma_{ij}(a_{ij})=1$
2. Compute distance of each element
   $d_{ij}=(a_{ij} - b_{ij})^n$ similarity
3. Sum all distances
   $D_n(\mathbf{A},\mathbf{B})=\Sigma_{ij}(d_{ij})$