

POLITECNICO DI TORINO



CORSO DI LAUREA MAGISTRALE  
IN INGEGNERIA CIVILE

TESI DI LAUREA MAGISTRALE

**USO DEL FORMATO OMX  
PER LA CODIFICA E LA MANIPOLAZIONE  
DI MATRICI ORIGINE/DESTINAZIONE**

**Relatori**

Prof. Marco DIANA

.....

Prof. Francesco DEFLORIO

.....

**Candidato**

Michela CORRADIN

.....

Settembre 2018

# **USO DEL FORMATO OMX PER LA CODIFICA E LA MANIPOLAZIONE DI MATRICI ORIGINE/DESTINAZIONE**

Nel 2013 un gruppo di professionisti attivi nel settore dei trasporti, creò un nuovo formato libero per l'archiviazione di dati di tipo matriciale con il nome Open Matrix (OMX). Questa specifica, che riprende molto delle caratteristiche del già affermato formato HDF5, permette di memorizzare, attraverso una struttura definita "ad albero", più matrici in unico file, a patto che abbiano tutte la stessa dimensione, accompagnate dai relativi metadati e da una "mappa dell'indice", per l'associazione dei codici di zona alle righe/colonne. La sua implementazione è risultata utile, agli occhi dei suoi sviluppatori, in quanto i formati sino a quel momento esistenti presentavano tutti delle grosse carenze quando utilizzati per il salvataggio delle matrici di trasporto. Ai file OMX è possibile accedere attraverso alcuni software di pianificazione dei trasporti, che ne supportano il formato, oppure grazie ai più comuni linguaggi di programmazione. Esiste, inoltre, un pratico visualizzatore, dal nome "OMX Viewer", che permette di aprire i file OMX e di esplorarne il contenuto.

Il presente lavoro ha, quindi, l'obiettivo di analizzare questo formato libero, valutarne gli aspetti positivi e negativi a confronto con un formato dalle opposte caratteristiche come il MAT, formato proprietario di MATLAB. Questo sia per quanto riguarda il salvataggio dei dati, determinando lo spazio di memoria occupato nel calcolatore, sia per quanto riguarda l'accessibilità e la manipolazione dei dati immagazzinati, valutando l'onere computazionale e i tempi di esecuzione dei listati di comando. I risultati pubblicati sul sito dell'Agenzia della Mobilità Piemontese (AMP), riguardanti l'Indagine sulla Mobilità delle persone e sulla Qualità dei trasporti (IMQ) effettuata sul territorio piemontese nel 2013,

hanno permesso di avere a disposizione una quantità abbastanza rilevante di informazioni da memorizzare e manipolare. Mentre, tra i diversi linguaggi di programmazione supportati da OMX, si è scelto di utilizzare Python per le elaborazioni proposte.

Attraverso degli script Python sono, quindi, stati creati i file OMX per le matrici origine/destinazione dell'indagine IMQ. In particolare, essendo redatte sulla base di due diverse zonizzazioni (una per le 208 zone campionarie più una zona definita "esterno" e una per 5 macro-aree), necessariamente i file OMX prodotti sono risultati due: il primo contenente le matrici 5x5; il secondo quelle 209x209. Analogamente, con un due listati MATLAB, si sono creati i corrispondenti file MAT di confronto. Le successive manipolazioni hanno interessato solamente i due file con la suddivisione per zone campionarie, poiché di maggiori dimensioni. Nello specifico, le elaborazioni proposte hanno riguardato: la determinazione del flusso tra una coppia di zone; il calcolo del flusso totale prodotto/attratto da una zona; l'estrazione di parte della matrice; l'aggregazione dei risultati sulla base di un attributo territoriale; l'ottenimento di informazioni di sintesi, come la percentuale di scelta del mezzo pubblico. Il software QGIS è stato impiegato come interfaccia per la visualizzazione grafica di alcuni risultati ottenuti.

# **USE OF THE OMX FORMAT TO CODE AND TO WORK ON ORIGIN/DESTINATION MATRICES**

In 2013 a group of practitioners working in the transportation field, created a new free format for storing matrix data with the name Open Matrix (OMX). This specification, which takes a lot of characteristics of the successful HFD5 format, allows to save, through a “tree structure”, multiple matrices in a single file, with the provision that they have all the same size, accompanied by their respective metadata and by an “index map”, for the association of area codes to rows/columns. Its implementation was useful, in the eyes of its developers, because all the existing formats occurred great lacks when used for transportation matrix storage. OMX files can be accessed through some transport planning software, which supports the format, or thanks to the most common programming languages. There is also a practical viewer, called "OMX Viewer", which allows to open OMX files and explore their contents.

The present work has, therefore, the objective of analyzing this free format, evaluating the positive and negative aspects compared to a format with opposing characteristics such as MAT, the vendor format of MATLAB. This both for data storage, determining the memory space occupied in the computer, and the accessibility and manipulation of the stored data, evaluating the computational burden and the execution times of the command lists. The results published on the piedmont mobility agency (AMP) website, concerning the people mobility and transport quality survey (IMQ) carried out in the piedmont area in 2013, have made available a quite relevant amount of information to be memorized and manipulated. While, among the different programming languages supported by OMX, it

was decided to use Python for the proposed elaborations.

Therefore, through Python scripts, the OMX files were created for the storage of the origin/destination matrix of the IMQ survey. Specifically, because they are drawn up based on two different zonings (one for the 208 sample areas plus one defined as "external" and one for 5 macro-areas), the OMX files produced were necessarily two: the first containing the 5x5 matrices; the second the 209x209 ones. Similarly, with two MATLAB lists, the corresponding comparison MAT files were created. The subsequent manipulations have affected only the two files with the subdivision by sample areas, because they were larger dimensions than other ones. Specifically, the proposed elaborations concerned: the determination of the flow between a couple of zones; the calculation of the total flow produced/attracted by a specific zone; the extraction of part of the matrix; aggregation of results based on a territorial attribute; obtaining summary information, such as the percentage of public transport choice. The QGIS software has been used as an interface for the graphic display of some results obtained.

# INDICE

INTRODUZIONE	1
<b>1. FORMATO OPEN MATRIX</b>	<b>4</b>
1.1 Caratteristiche del formato OMX	7
1.1.1 <i>Struttura dei dati</i>	8
1.1.2 <i>Visualizzazione della struttura in OMX Viewer</i>	11
1.1.3 <i>Specifiche del formato (versione 0.3)</i>	12
1.1.4 <i>Compressione dei file OMX</i>	16
1.1.5 <i>Software che ne supportano il formato</i>	18
1.2 Interfacce di programmazione	24
1.2.1 <i>Caratteristiche del linguaggio di programmazione Python</i>	26
1.3 Applicazioni del formato OMX	28
1.4 Conclusioni	30
<b>2. SOLUZIONI ESISTENTI</b>	<b>32</b>
2.1 Hierarchical Data Format: HDF	32
2.1.1 <i>Applicazioni nel campo dei trasporti</i>	37
2.2 Network Common Data Form: NetCDF	38
2.3 Structured Query Lite: SQLite	41
2.4 Binary Javascript Object Notation: BSON	43
2.5 Zip Matrix: ZMX	44
2.6 Formati proprietari	45
2.7 Formati di testo	47
2.8 Formati a confronto	47

<b>3. DATABASE IMQ 2013</b>	<b>50</b>
3.1 Cenni storici sulle IMQ	52
3.2 Indagine conoscitiva del 2013	54
3.2.1 <i>Campionamento</i>	55
3.2.2 <i>Questionario</i>	62
3.3 Dati elementari della IMQ	65
3.4 Elaborazioni già a disposizione dai dati IMQ	70
3.5 Opportunità di ulteriori elaborazioni	82
<b>4. MATRICI IMQ 2013 IN FORMATO OMX</b>	<b>83</b>
4.1 Ambiente di lavoro in Python	84
4.1.1 <i>Come ottenere Python</i>	84
4.1.2 <i>Interprete interattivo e IDLE Python</i>	85
4.1.3 <i>Miniconda</i>	88
4.2 Moduli e comandi principali	90
4.2.1 <i>Openmatrix</i>	91
4.2.2 <i>Openpyxl</i>	93
4.2.3 <i>Altri comandi utili</i>	94
4.3 Redazione ed esecuzione degli script per la generazione delle matrici	94
4.3.1 <i>Matrici per macro-aree</i>	95
4.3.2 <i>Matrici per zone campionarie</i>	99
4.4 Verifica del codice, risultati ed osservazioni	102
<b>5. ELABORAZIONI SULLE MATRICI OMX</b>	<b>108</b>
5.1 Estrazione del flusso tra due zone definite	110
5.2 Determinazione del flusso totale prodotto/attratto da una zona	114
5.3 Estrazione di parte di matrice	121
5.4 Aggregazione di una matrice in base agli attributi	126
5.5 Ottenimento di informazioni di sintesi	131
5.6 Confronto con MATLAB	137

## INDICE

<i>5.6.1 Estrazione del singolo flusso</i>	140
<i>5.6.2 Determinazione del flusso totale prodotto/attratto</i>	142
<i>5.6.3 Estrazione di parte della matrice</i>	145
<i>5.6.4 Aggregazione della matrice</i>	147
<i>5.6.5 Ottenimento informazioni di sintesi</i>	151
<i>5.6.6 Commenti sui risultati del confronto</i>	153
<b>6. CONCLUSIONI</b>	<b>157</b>
<b>ALLEGATO A</b>	<b>161</b>
<b>ALLEGATO B</b>	<b>163</b>
<b>ALLEGATO C</b>	<b>175</b>
<b>ALLEGATO D</b>	<b>178</b>
<b>ALLEGATO E</b>	<b>181</b>
<b>ALLEGATO F</b>	<b>187</b>
<b>FONTI BIBLIOGRAFICHE</b>	<b>189</b>



# INTRODUZIONE

La pianificazione dei trasporti è una disciplina che permette di determinare quali interventi possono essere attuati, al fine di migliorare le condizioni di mobilità osservate o previste in determinati scenari. La prevedibile complessità del processo richiede l'elaborazione di approcci e metodi che siano, d'altro canto, il più possibile semplici e chiari. Il continuo sviluppo di strumenti di analisi e di simulazione per i calcolatori, quali software e applicativi, rende più facile e gestibile lo studio della grande quantità di dati e di variabili a cui è associato tale processo.

Il presente lavoro ha come oggetto l'analisi di un recente formato di file per la memorizzazione dei dati associati alle matrici Origine/Destinazione, frutto dei processi conoscitivi degli spostamenti di interesse per l'area di studio. Tale estensione, il cui debutto risale al 2013 con l'acronimo di OMX (Open Modeling Matrix), prende spunto da formati simili già affermati e in commercio alla data di creazione. L'obiettivo è dunque quello di esplorarne le potenzialità applicative e di metterne in risalto i punti di forza e di evidenziarne i punti deboli a confronto con le soluzioni sinora proposte.

A tale scopo, si è reso necessario l'utilizzo di un linguaggio di programmazione che permettesse di elaborare dei dati di mobilità con questa specifica. Tra i vari linguaggi supportati dal formato OMX, la scelta è ricaduta su Python. Per quanto riguarda i dati di mobilità, la recente pubblicazione dei risultati dell'indagine conoscitiva piemontese IMQ (Indagini sulla Mobilità delle Persone e sulla Qualità dei Trasporti) del 2013, e dei relativi database, ha permesso l'ottenimento di un set di dati adatti sui quali operare.

Il lavoro è stato suddiviso in cinque sezioni. La prima parte è dedicata alla conoscenza

## INTRODUZIONE

approfondita del formato OMX; vengono illustrate quindi le sue caratteristiche principali, la sua struttura interna e il suo funzionamento, le interfacce di programmazione disponibili (API), i software che ne supportano il formato e infine un suo esempio di applicazione ad un database riguardante una raccolta di flussi di merci negli Stati Uniti.

La seconda parte del lavoro è dedicata invece alla definizione del panorama delle soluzioni esistenti alternative al formato OMX. Vengono presentati i principali formati di archiviazione di grandi quantità di dati, con particolare attenzione a quelli che possono essere utilizzati per memorizzare i dati relativi al settore dei trasporti. Tra questi viene presentato anche il formato HDF5, sulla traccia del quale è stato strutturato l'Open Matrix.

Nella terza parte vengono presentate le indagini IMQ. Viene fornita una rassegna degli studi compiuti nel comune di Torino e, più in generale, nella regione Piemonte negli ultimi anni. La trattazione si focalizza quindi su quella che fornirà la base dati per le elaborazioni, ovvero l'Indagine del 2013. Ne vengono descritte le modalità e presentati i risultati prodotti dall'Agenzia Metropolitana Piemontese, responsabile dell'esecuzione e dell'analisi.

La quarta parte riporta dettagliatamente tutti i passaggi eseguiti per la creazione dei codici Python di conversione delle matrici nel formato OMX. Vengono indicati i passaggi da eseguire per l'ottenimento di tutti gli strumenti necessari allo scopo; come l'installazione di Python e di altri strumenti accessori indispensabili alla programmazione (Wing e Anaconda) e vengono illustrati i comandi principali utilizzati nello *script* di conversione dei file.

Infine, l'ultima parte del lavoro, mostra come possono essere risolte alcune operazioni sulle matrici O/D mediante il formato OMX. A tal scopo si sono analizzate alcune situazioni abbastanza ricorrenti nella pratica professionale e nella ricerca accademica: l'estrazione di un flusso tra due zone ben definite di una matrice O/D (ovvero la ricerca di un particolare valore memorizzato in una specifica cella della tabella); la determinazione del flusso totale in uscita e in entrata da una particolare zona (ovvero l'assegnazione

## INTRODUZIONE

rispettivamente dei totali di riga e di colonna); l'estrazione di parte di una matrice in base alla zona di indagine; il raggruppamento dei flussi forniti a livello di zona campionaria in base ad una caratteristica della zona (come possono esserlo la provincia o la macro-zona di appartenenza); l'ottenimento di informazioni di sintesi sull'andamento generale di un fenomeno (come la percentuale di utilizzo di trasporto pubblico sugli spostamenti totali). Queste stesse operazioni sono state, in ultimo, riproposte nell'ambiente di MATLAB per confrontare i vantaggi e svantaggi nell'utilizzo del formato OMX.

# 1. FORMATO OPEN MATRIX

Nei primi mesi del 2013 una serie di aziende, consulenti e ricercatori, attivi nel settore dei trasporti, formarono un gruppo di lavoro con l'obiettivo di implementare un nuovo formato *open source*<sup>1</sup> per l'archiviazione dei file di tipo matriciale [84]. Lo scopo era quello di creare un formato standard aperto, e comune ai diversi software di progettazione, per permettere la memorizzazione delle matrici Origine/Destinazione, ma soprattutto per facilitare lo scambio dei dati tra i diversi programmi [25]. Infatti, fino a quel momento, il passaggio di informazioni tra software diversi era molto complicato. Per questo motivo aziende, agenzie, professionisti e studenti, si trovavano in difficoltà quanto risultava necessario ottenere dati da lavori svolti con software diversi.

Da questi presupposti venne sviluppato il formato Open Modeling Matrix, o brevemente Open Matrix (OMX); basato sulla tecnologia di archiviazione già utilizzata nel formato HDF5 (si veda il Capitolo 2.1 - Hierarchical Data Format: HDF); tale formato risulta utile per qualunque scopo legato alla gestione di un gran numero di dati con struttura a matrice. Questa specifica, permette di memorizzare più matrici in un solo file, a patto che queste siano tutte della stessa dimensione, come dati binari compressi. Consente di includere inoltre delle "legende" per gli indici (molto utili nel caso si rappresentino le

---

<sup>1</sup> Con il termine *open source* si indica una caratteristica di alcuni formati di salvataggio di file, i quali sono gestiti da enti di standardizzazione che operano in ottica non proprietaria. Il principale vantaggio di questi formati è che non sono legati a uno specifico software per la lettura. Inoltre, poiché indipendenti dall'applicazione con cui sono maneggiati, e quindi anche dai successivi aggiornamenti, ne è garantito l'accesso e la "leggibilità" nel tempo.

matrici O/D per identificare la zona da cui parte, o arriva, il flusso di spostamenti rappresentato nelle relative celle), degli attributi e dei metadati associati definiti dall'utente.

Il merito di questo formato, come anticipato, va attribuito a un gruppo di lavoro, capeggiato da Benjamin Stabler [30] (ai tempi dello sviluppo del formato impiegato nella multinazionale Parsons Brinchkerhoff, attualmente invece direttore del Resource Systems Group) e composto da diversi membri, quali: Billy Charlton (Puget Sound Regional Council); Brian J. Gregor (Oregon Systems Analytics); Christopher Frazier (Parsons Brinchkerhoff); Elizabeth Sall (San Francisco County Transportation Authority); Jeremy Raw (Federal Highway Administration); John E. Abraham (HBA Specto Incorporated); e Suzanne Childress (Puget Sound Regional Council). Successivamente ulteriori collaboratori e aziende si sono uniti al gruppo per coltivare lo sviluppo di questo formato; tra questi: Peter Schmiedeskamp (University of Washington); Andrew Rohne (OKI Regional Council of Government); INRO (società sviluppatrice del software Emme, uno tra i primi applicativi per la previsione del traffico); PTV Group (società sviluppatrice degli omonimi software di pianificazione dei trasporti); Caliper (società sviluppatrice di TransCAD Transportation Planning Software); e Citilabs (società sviluppatrice di Cube, una piattaforma di modellazione che copre tutti gli aspetti associati alla pianificazione dei trasporti).

Nell'aprile del 2014, il gruppo originario illustrò il progetto dell'OMX, con una presentazione dal titolo "*Collaborative Development of an Open Matrix Standard*", nella quinta edizione della Transportation Research Board Conference on Innovations in Travel Modeling<sup>2</sup> del Transportation Research Board<sup>3</sup> a Baltimora [85].

---

<sup>2</sup> Si tratta di una serie di conferenze, tenute con cadenza biennale, il cui scopo è quello di creare un forum per la condivisione degli ultimi sviluppi in fatto di modellazione nel settore dei trasporti. Gli incontri sono divisi in sessioni, e affrontano una vasta gamma di argomenti relativi alle più recenti innovazioni. Queste  
(segue)

Inizialmente il formato OMX era stato implementato per alcuni linguaggi di programmazione, denominati API (Application Programming Interface), nello specifico per Python (si veda il Paragrafo 1.2.1 - Caratteristiche del linguaggio di programmazione Python), Java, R e C# (si veda il Capitolo 1.2 - Interfacce di programmazione), e supportato da alcuni software di pianificazione dei trasporti come Emme (si veda il Paragrafo 1.1.5 - Software che ne supportano il formato) e PECAS<sup>4</sup>. Ma il continuo lavoro del gruppo al progetto, ha portato a un aumento dei linguaggi supportati, aggiungendo ai precedenti Ruby e C++. Per quanto riguarda invece i software di pianificazione che permettono il salvataggio nel formato OMX, a quelli citati in precedenza, si sono aggiunti TransCAD, Cube, Visum e Aimsun.

Parallelamente al nuovo formato, il gruppo capeggiato da Stabler, creò anche un visualizzatore per i file OMX, denominato OMX Viewer [27]. Questa applicazione *freeware*<sup>5</sup>, il cui download è disponibile on line nella pagina del progetto OMX<sup>6</sup> per Windows, Linus e Macintosh, permette l'apertura e l'esplorazione delle matrici salvate, e prende spunto dall'esistente visualizzatore HDF5 ViTables.

---

occasioni permettono inoltre l'incontro tra i professionisti, che così possono indentificare le diverse problematiche di settore, individuare le possibili soluzioni e i campi in cui è auspicabile svolgere delle ricerche.

<sup>3</sup> La Transportation Research Board (TRB) è una delle sei divisioni del National Research Council degli Stati Uniti [80] (insieme a: Behavioral and Social Sciences and Education; Earth and Life Studies; Engineering and Physical Sciences; Health and Medicine; Policy and Global Affairs; e Gulf Research Program [81]). Fornisce consulenza al governo federale per quanto riguarda le questioni scientifiche e tecniche di importanza nazionale e sostiene l'innovazione e il progresso nel settore dei trasporti. È supportata dai dipartimenti dei trasporti statali, dalle agenzie federali e da aziende e soggetti appartenenti al settore.

<sup>4</sup> PECAS è un software per la modellazione del trasporto su terra; è stato sviluppato da HBA Specto Incorporated, un'azienda fondata nel 2000 con sede in Canada [50].

<sup>5</sup> Il termine *freeware* indica una categoria di software, proprietari o *open source*, di cui è concesso l'ottenimento e l'utilizzo in modo gratuito.

<sup>6</sup> Benjamin Stabler, *OMX Viewer*, <https://github.com/osPlanning/omx/wiki/OMX-Viewer>, data di consultazione: 20/06/2017.

Il formato OMX non ha ancora raggiunto la sua versione definitiva e continui miglioramenti, sviluppi e modifiche, da parte del gruppo di lavoro, sono all'ordine del giorno. L'obiettivo è quello di rendere tale formato sempre più "aperto" verso tipi di dati e formati differenti da quelli finora supportati. Per questo motivo, ogni evoluzione è identificata dalla relativa versione (attualmente è in corso la 0.3, rilasciata il 6 aprile 2018). Ciò rende possibile il recupero delle specifiche di formato appropriate al file di dati che si vuole aprire o manipolare. In aggiunta, altre implementazioni sono focalizzate sulle dimensioni dei file. Attualmente è possibile salvare dei documenti già abbastanza grandi, fino a molti gigabyte di memoria; l'obiettivo per il futuro è quello di superare il terabyte.

## 1.1 CARATTERISTICHE DEL FORMATO OMX

Sostanzialmente, un file matriciale OMX (la cui estensione è banalmente ".omx"), è una collezione strutturata di oggetti, detti *array*, bidimensionali e relativi metadati associati. Nel campo dell'informatica, con il termine *array* si indica un "vettore" strutturato, che permette di memorizzare e organizzare i dati secondo un indice. Per semplificarne il concetto, questi vettori possono essere visti come una serie contigua di celle vuote, nelle quali possono essere memorizzati i dati; a questi ultimi è possibile accedere individuando l'indice della relativa cella, ovvero il numero d'ordine della stessa, considerando che la numerazione solitamente parte da zero e non da uno (la prima cella ha quindi indice 0, la seconda 1, e così via). Sono definiti "dati strutturati" poiché sono ottenuti mediante associazione di altri dati, che possono essere anch'essi strutturati o semplici.

La struttura di ogni file OMX, per poter funzionare in modo ottimale, deve rispondere a due requisiti fondamentali [28]:

- \* la completezza e la coerenza delle informazioni contenute nelle matrici;
- \* il recupero efficiente e corretto dei dati salvati.

La creazione e la manipolazione di un file OMX è possibile attraverso i software di pianificazione che ne supportano il formato, gli strumenti dell'HDF5, e le API viste in precedenza. L'utilizzo di un supporto, piuttosto che di un altro, è indifferente poiché l'importante è che il file risultante sia conforme ai due requisiti descritti sopra. Detto ciò è preferibile realizzare e modificare i file di questo formato, utilizzando direttamente gli *script* dei linguaggi di programmazione supportati; questo perché con tale metodo si ha un maggiore controllo della coerenza dei parametri associati alle matrici da memorizzare.

### **1.1.1 Struttura dei dati**

La struttura dei dati, in un formato di file, rappresenta la logica con la quale sono memorizzati i dati al suo interno. Quella dei file OMX è, utilizzando la terminologia tipica del campo dell'informatica, del tipo "ad albero"; come quella del formato HDF5 da cui trae spunto. È caratterizzata da due elementi principali [29], identificabili come due cartelle distinte chiamate "*cartella dati*" e "*cartella di ricerca*", alloggiate in una *directory* comune denominata "*radice*".

#### 1.1.1.1 ROOT NODE

Il *Root Node*, (letteralmente il "nodo radice") è simile a una *directory*. Contiene tutte le cartelle e i dati del file, identifica la versione dello standard OMX impiegato e la **dimensione** delle matrici ospitate, che deve essere **fissa**, e caratterizzata da due numeri interi che ne identificano rispettivamente il numero di righe e di colonne. Un file infatti non può contenere tabelle di dimensioni differenti, ma può tranquillamente ospitare matrici che includono tipologie di dati differenti. Per questo motivo, per ogni matrice salvata si dovrà specificare la tipologia di dati contenuti:

- \* *int* nel caso si tratti di numeri interi;
- \* *float* e *double* nel caso di numeri con virgola mobile.



Attualmente solo questi tre tipi di dati sono supportati dal formato. Si fa inoltre presente che il tipo di dato può differire da matrice a matrice, ma non può cambiare all'interno della stessa; tutte le celle che la compongono devono necessariamente ospitare lo stesso tipo di dato.

#### 1.1.1.2 DATA FOLDER

La *Data Folder*, è una cartella di “dati”. È quella che contiene tutte le matrici del file e in cui non è consentita la creazione di ulteriori sottocartelle. Le tabelle sono memorizzate, come accade nel formato HDF5, ordinando i dati per righe, e identificate attraverso un nome univoco. Non possono essere presenti, nello stesso file, due matrici con la stessa denominazione; questo allo scopo di garantire il corretto accesso e recupero delle informazioni contenute. Al nome può anche essere associato un attributo descrittivo del sistema, il quale riporta una breve descrizione di quello che contiene la matrice in oggetto. In realtà le possono essere associati un numero illimitato di attributi aggiuntivi definiti dall'utente (diversificati dagli attributi del sistema), alcuni esempi possono essere: l'anno a cui è associata la matrice in oggetto, il modo utilizzato per gli spostamenti rappresentati e molti altri.

#### 1.1.1.3 LOOKUP FOLDER

La *Lookup Folder*, è una cartella di “ricerca” che contiene le “mappe degli indici” delle righe e delle colonne delle matrici. Spesso è utile associare agli indici di vettori e matrici, delle informazioni che ne permettono di identificare meglio il contenuto. Nelle matrici O/D questa caratteristica è essenziale. È indispensabile infatti associare agli indici, le informazioni sulla zonizzazione dell'area di studio, al fine di individuare le rispettive TAZ (Traffic

Analysis Zone)<sup>7</sup> che originano, o accolgono, il flusso rappresentato nella rispettiva cella della matrice. Nello specifico si tratta di un vettore, di lunghezza pari a una delle due dimensioni della matrice, che contiene ordinatamente, per ciascun indice, il codice identificativo della relativa TAZ. Nel caso la matrice fosse quadrata (come accade per le matrici O/D) è possibile definirne uno solo, valido per entrambi gli indici di righe e colonne; nel caso in cui invece la matrice non lo fosse è possibile specificare due mappature, una per ogni dimensione. Come accade per le matrici, e per gli stessi motivi, anche queste “mappe degli indici” devono essere salvate con un nome che le identifichi univocamente. È possibile specificare inoltre, attraverso un attributo del sistema, se la mappatura in oggetto è correlata alla prima o alla seconda dimensione della matrice; assegnando a tale campo rispettivamente il valore 0 o il valore 1. Nel caso in cui la correlazione fosse per entrambe, righe e colonne, questo attributo invece non deve essere specificato.

Questi tre elementi rappresentano per intero la struttura ad albero del formato OMX. Tutte le matrici salvate, e i relativi metadati, sono quindi reperibili nella sottocartella *data*, mentre le indicazioni sugli indici, e i relativi attributi, sono memorizzati nella sottocartella *lookup*. Queste costituiscono i due, e unici, rami del nodo rappresentato dalla cartella denominata *root node*, che le contiene entrambe. Da quello che si può vedere, la struttura ad albero impiegata da questo formato risulta dunque molto semplice.

---

<sup>7</sup> Le Traffic Analysis Zones, letteralmente le “zone di analisi del traffico”, sono le unità territoriali nelle quali viene suddivisa l’area di studio durante il processo di pianificazione. Sono individuate a seguito della zonizzazione del territorio, e possono essere più o meno grandi in base al dettaglio che si vuole raggiungere nell’indagine. Generalmente il loro limite è definito seguendo i confini amministrativi dei comuni o delle circoscrizioni di quartiere. Spesso sono identificate attraverso un codice che ovviamente differisce dall’indice dell’oggetto *array* che ne memorizza il flusso in uscita o in ingresso.

### 1.1.2 Visualizzazione della struttura in OMX Viewer

È possibile osservare in maniera esplicita questa struttura di memorizzazione dei dati nell'interfaccia del visualizzatore OMX Viewer. In FIGURA 1.1.1 è riportata la schermata che si presenta all'apertura di un file con questa applicazione.



FIGURA 1.1.1 Schermata all'apertura di un file con OMX Viewer. Lo script per la realizzazione del file esemplificativo è fornito nell'ALLEGATO A.

Come si può notare sulla sinistra della schermata è presente un elenco a cascata, che riproduce la struttura dei dati, dove è possibile individuare il nome del file come una *directory* e le due cartelle in cui sono alloggiati i dati memorizzati (*data* e *lookup*). Nello specifico, in questo file di esempio sono memorizzate tre matrici, una per gli spostamenti privati (dal nome "*PRIV*"), una per quelli pubblici (dal nome "*PUBB*") e una per quelli totali (dal nome "*TOT*"), e una mappa dell'indice (dal nome "*mappatura*"), per relazionare la posizione dei dati delle matrici alle zone in cui è stata suddivisa l'area di studio.

Nella FIGURA 1.1.2 è evidente come per ogni matrice sia possibile salvare delle informazioni utili alla sua identificazione (reperibili nella scheda "*proprietà*" della matrice), come l'anno del sondaggio su cui è basata, l'intervallo temporale di osservazione degli

spostamenti, e molti altri. Lo spazio a tale scopo è illimitato per questo formato. Un'altra osservazione, riguarda la mappatura; come si può osservare sempre nella FIGURA 1.1.2, gli indici di riga e colonna sono stati manipolati per rendere più immediata la lettura della matrice. Infatti, figurano i codici assegnati alle zone, piuttosto che la tipica indicizzazione degli *array* (numerica, sequenziale e con base 0).

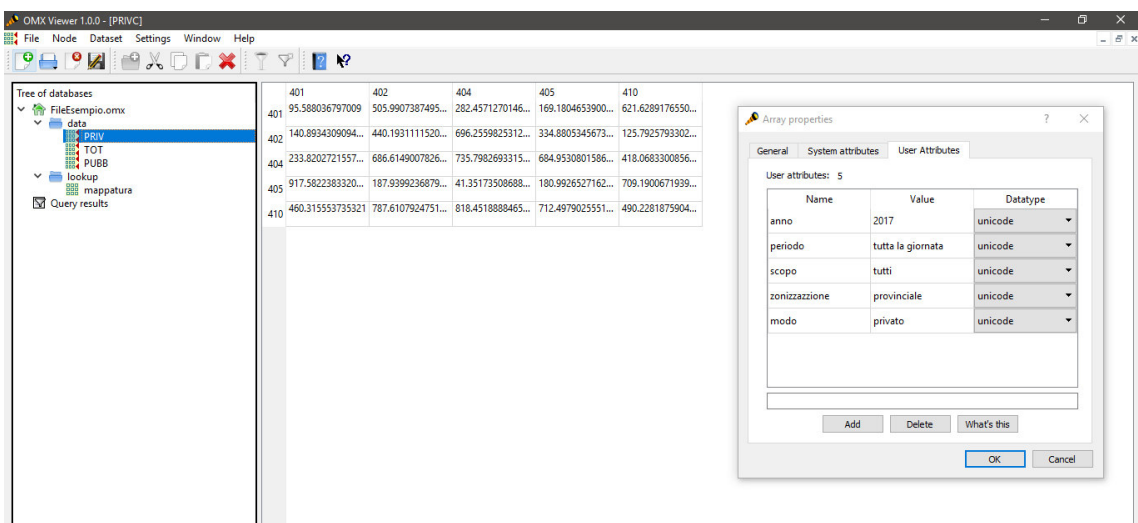


FIGURA 1.1.2 Visualizzazione, con OMX Viewer di una delle tre matrici salvate e dei relativi metadati ad essa associati.

### 1.1.3 Specifiche del formato (versione 0.3)

La memorizzazione delle matrici con il formato OMX avviene attraverso la definizione del *chunk*, la cui struttura è descritta attraverso una serie di informazioni rappresentate dai metadati. Il *chunk* è un blocco di memoria<sup>8</sup> che risiede nell'*heap*, un tipo di struttura di dati

<sup>8</sup> Un blocco di memoria, in informatica, è una parte della memoria fisica (o memoria reale), ottenuta a seguito di un processo di gestione della memoria definito "segmentazione", che ne prevede la suddivisione in blocchi di lunghezza fissa o variabile detti "segmenti". Questo processo non influisce assolutamente sui dati salvati, ma ha il vantaggio di diminuire i tempi di I/O da disco poiché suddivide il programma in moduli.

basato sugli alberi. È composto da un *header*, cioè una specie di intestazione, di dimensione costante pari a 8 byte e da uno spazio per la memorizzazione dei dati, di dimensione variabile maggiore o uguale a 8 byte. Il formato OMX non richiede di specificare la dimensione del *chunk* nel momento della sua definizione, cosa che invece accade per il formato HDF5 da cui trae spunto. Questa viene gestita direttamente dall'API impiegata, che ne imposta un valore di default che nella maggior parte delle applicazioni lavora bene.

Alcuni dei metadati che descrivono la struttura del blocco di memoria risultano obbligatori, come la dimensione della matrice o il tipo di dati contenuto, altri invece sono facoltativi e possono essere legati ad esempio direttamente alla matrice, alla sua dimensione o ancora ai dati contenuti [22]. Poiché questa caratteristica potrebbe cambiare con le versioni successive del formato, è utile specificare che le informazioni riportate di seguito fanno riferimento alla versione 0.3 del formato, ovvero la più recente sviluppata.

Un file OMX consiste, come si è visto nel paragrafo precedente, in un elenco di tabelle bidimensionali identificate univocamente attraverso un nome. In aggiunta è possibile, ma non necessario, allegare a queste matrici delle “mappe degli indici” o *lookup*; cioè un elenco opzionale di vettori, a una dimensione, utilizzati per includere delle informazioni aggiuntive sugli indici delle tabelle. Infine, è possibile completare il file con una quantità arbitraria di metadati, memorizzati sotto forma di coppie chiave-valore.

Nella visione più semplice del problema, una matrice è, sostanzialmente, una tabella a due dimensioni, composta da una serie di celle che ospitano un determinato tipo di dati. Il numero delle celle è ovviamente legato alla dimensione della matrice e ognuna di esse è individuabile attraverso una coppia di numeri interi. Questi identificano i due indici che le corrispondono, il primo correlato alla riga e il secondo alla colonna di appartenenza. Nel formato OMX è possibile definire gli indici degli *array* con una certa libertà. Infatti, è possibile utilizzare, come base di partenza per il conteggio delle celle occupate dai dati, il numero 0, come avviene nella maggior parte dei linguaggi di programmazione, l'1 o un

numero  $N$  intero definito arbitrariamente. Questo, sia 0, 1 o  $N$ , rappresenta il valore più basso riscontrabile tra gli indici, mentre quello più alto si ottiene aggiungendogli la dimensione della matrice ridotta di un'unità.

Ogni matrice salvata accoglie un tipo di dati ben noto nelle sue celle, come possono esserlo degli interi, delle stringhe, ecc. L'aspetto importante è che tali dati devono poter essere convertiti dall'API in un formato nativo. L'API deve quindi fare da intermediario diretto tra il tipo di dati nel formato di archiviazione e il tipo di dato nativo equivalente. Se una cella della matrice dovesse rimanere vuota, deve essere predisposto l'uso di un indicatore coerente con il tipo di dati introdotti; alcuni esempi possono essere: "NaN" (Not a Number); "NA" (Not Available); "999999"; o ancora un qualsiasi numero negativo.

I *lookup*, ovvero le "mappe degli indici", sono vettori, anch'essi identificati univocamente da un nome, di lunghezza pari a una delle due dimensioni delle matrici a cui si riferiscono. Possono accogliere solamente numeri interi, ma non necessariamente consecutivi. Ogni elemento del *lookup* viene salvato utilizzando gli stessi indici impiegati nelle matrici, questo per mantenere la correlazione tra i dati. L'utilità di questi vettori è rappresentata dal fatto che, considerando la sua applicazione al campo dei trasporti e in particolar modo alle matrici O/D, è possibile salvare il codice che identifica una zona di origine o destinazione del flusso (solitamente identificata attraverso un codice quasi mai sequenziale e con base 0) in relazione agli indici impiegati nella matrice. Attraverso i vettori di *lookup*, dal codice della zona si ricava il relativo indice della matrice, e con questo è possibile accedere alle rispettive celle di interesse.

Per quanto riguarda invece i metadati, l'unica restrizione imposta è che questi siano convertibili dall'API in oggetti memorizzabili, nel formato di archiviazione HDF5, come

attributi<sup>9</sup>. Di seguito è riportato un elenco di metadati, alcuni obbligatori, altri facoltativi, che spesso si trovano allegati alle matrici OMX.

- \* Formato di archiviazione dati utilizzato. È facoltativo, ma se non specificato diversamente viene considerato di default l'HDF5. Rappresenta un vantaggio poiché permette di conoscere il formato di archiviazione senza effettuare ipotesi o inutili tentativi.
- \* Schema di compressione utilizzato. Spesso risulta utile comprimere file molto grandi, con questo attributo opzionale è possibile specificare quale filtro di compressione è stato impiegato (si veda il Paragrafo 1.1.4 Compressione dei file OMX).
- \* Versione. Si tratta di un metadato obbligatorio che specifica la versione OMX con la quale è stata codificata la matrice.
- \* Tipo di dato. Anche questo è un attributo obbligatorio, e serve per indicare il tipo di dati che la matrice accoglie. Tra i formati supportati si può scegliere: int8<sup>10</sup>, int16, int32, int64, uint8<sup>11</sup>, uint16, uint32, uint64, float32<sup>12</sup>, float64,

---

<sup>9</sup> Gli attributi del formato HDF5 sono l'equivalente dei metadati nel formato OMX. Sono composti anch'essi da un *header* e da uno spazio dedicato al salvataggio dei dati, la cui dimensione viene definita dall'utente. In questi è possibile salvare qualsiasi tipo di dato purché abbia un formato valido per l'HDF5 [52].

<sup>10</sup> Il formato di dati "int" permette di rappresentare i numeri interi, sia negativi che positivi. Il numero associato (8, 16, 32 e 64) rappresenta i bits di memoria riservati alla loro rappresentazione. Questo si traduce con una limitazione nel campo dei numeri che si possono salvare; ad esempio con int8 si possono salvare i numeri compresi nell'intervallo [-128, 127] e con int16 invece [-32768, 32767].

<sup>11</sup> Il formato di dati "uint" permette di rappresentare solo i numeri interi positivi. In questo caso, l'intervallo di dati rappresentabile in base ai bits di memoria dedicati, si sposta dal lato positivo. Ad esempio, con uint8 si rappresentano i numeri compresi nell'intervallo [0, 255] e con uint16 invece [0, 65535].

<sup>12</sup> Il formato di dati "float" permette di rappresentare i numeri reali, detti anche numeri in virgola mobile. La differenza tra i bits riservati, in questo caso, incide sia sul range di numeri che è possibile rappresentare sia sul numero di cifre dopo la virgola che si possono memorizzare.

complex64<sup>13</sup>, complex128, bool<sup>14</sup>, string<sup>15</sup> e unicode<sup>16</sup>.

- \* Dimensione della matrice. È un attributo facoltativo che, se non specificato, attribuisce alla tabella, di default, due dimensioni. In realtà, con il formato OMX, è possibile salvare anche tabelle con più di due dimensioni. Quando accade che una serie di matrici abbiano uguale forma, dimensione e tipo di dati, ma soprattutto analogia tra le informazioni memorizzate, è possibile creare delle matrici con più di due dimensioni. In tal caso risulta doveroso indicizzare le dimensioni aggiuntive in maniera opportuna ed efficiente. In ogni caso questo procedimento non è obbligatorio e può essere facilmente aggirato creando una raccolta di matrici bidimensionali.

A questo breve elenco è possibile aggiungere ulteriori metadati, a discrezione dell'utente, sia per quanto riguarda la quantità che per il contenuto accolto.

#### ***1.1.4 Compressione dei file OMX***

Come anticipato spesso può risultare vantaggioso comprimere un file OMX; riducendone le dimensioni infatti è possibile diminuirne lo spazio di occupazione sul disco di salvataggio. La compressione è sostanzialmente un'operazione che sfrutta degli algoritmi che, basandosi su dei metodi statistici, permettono di eliminare da un file di qualsiasi tipo, le informazioni ridondanti. Questi algoritmi sono chiamati "filtri di compressione" e ne esistono molteplici, basati su altrettanti metodi, già implementati e pronti all'utilizzo. Si dividono

---

<sup>13</sup> Il formato di dati "complex" permette di rappresentare i numeri complessi; tenendo separata la parte reale da quella immaginaria.

<sup>14</sup> Il formato di dati "bool" memorizza le variabili booleane, cioè vero/falso, true/false o ancora 1/0.

<sup>15</sup> Il formato di dati "string" permette di rappresentare delle sequenze di caratteri alfanumerici.

<sup>16</sup> Il formato di dati "unicode" permette di memorizzare ogni carattere alfabetico attraverso un numero.



principalmente in due macro-categorie: quelli che ammettono una perdita di informazioni, e in tal caso si parla di compressione *lossy*, dalla quale non è più possibile recuperare il file nativo; e quelli che non ammettono perdite, chiamati *lossless*, che permettono di “ricostruire” il file nella sua interezza originaria.

Per comprimere i file OMX sono disponibili diversi filtri, ovvio che per l'importanza, e il tipo, di dati da comprimere, gli algoritmi da utilizzare dovranno permettere il recupero per intero del file originario, e quindi appartenere alla seconda categoria di filtri appena descritti. Comunque gli sviluppatori del formato, consigliano di comprimere i file OMX con il filtro di compressione “*zlib con livello di compressione 1*”. Zlib è una libreria *free* di compressione disponibile per qualunque hardware e sistema operativo [49]. È stata sviluppata nel 1995 da Jean-loup Gailly e Mark Adler. I suoi algoritmi sono richiamabili attraverso un linguaggio di programmazione, come può esserlo Python, mediante delle funzioni specifiche. Ha nove livelli di compressione, indicati dai relativi numeri arabi: il livello 1 comporta una compressione minore, ma più veloce; il livello 9 permette la massima compressione che però avviene molto lentamente; tra questi estremi si collocano i restanti sette livelli intermedi. Tale filtro viene consigliato, ed è effettivamente l'unico supportato, dai creatori dell'OMX, perché è il solo disponibile in tutte le implementazioni del formato HDF5. D'altronde, se venisse utilizzato un filtro diverso, si potrebbe ottenere un file non leggibile; e in tal caso verrebbe a mancare il vantaggio principale del formato OMX, nonché lo scopo della sua implementazione, quale universalità di lettura.

Il filtro di compressione *zlib* con livello 1 è veloce e il suo impiego, per quanto visto, non compromette la trasferibilità del file matriciale. L'uso sembrerebbe conveniente per qualsiasi file, al fine di ridurre le dimensioni per occupare quanto meno spazio sul disco di salvataggio. Questa affermazione è solo in parte corretta, poiché la compressione implica anche uno svantaggio: la lentezza di scrittura. È consigliabile dunque valutare di volta in volta se la compressione possa portare a un reale vantaggio. Ad esempio, con i file di grosse

dimensioni, sicuramente risulta conveniente, mentre con file più piccoli talvolta può risultare più vantaggioso avere una scrittura agevole a discapito di qualche byte di memoria.

### ***1.1.5 Software che ne supportano il formato***

L'obiettivo del formato OMX è di essere quanto più aperto possibile; per agevolare al massimo la condivisione dei dati matriciali inerenti al settore dei trasporti. Ciò non sarebbe stato realizzabile se le aziende produttrici di software per la pianificazione, avessero continuato a supportare i soli formati proprietari. Fortunatamente, le finalità di azione del gruppo capeggiato da Stabler, furono condivise anche da alcune di queste aziende, le quali, addirittura, contribuirono significativamente all'implementazione del formato OMX. Di seguito sono riportati i principali software che attualmente supportano questo formato per la memorizzazione dei dati di tipo matriciale.

#### **1.1.5.1 EMME**

È un software per la modellazione della domanda e per la previsione degli spostamenti a livello urbano, regionale e nazionale [56]. Integra diversi strumenti tra di loro, come modelli di previsione e sistemi GIS (Geographic Information System), per poter ricreare degli scenari da confrontare. Permette di fare previsioni sulla domanda di trasporto; è possibile rappresentare la zonizzazione scelta per l'area di studio e implementare modelli personalizzati di generazione e di distribuzione degli spostamenti. È utile, ad esempio, per la pianificazione dei trasporti pubblici, poiché è possibile quantificare l'effetto che alcune modifiche apporterebbero alla rete, come la frequenza dei passaggi, di conoscere l'affollamento sui veicoli e nelle stazioni e di valutarne l'accessibilità e l'integrazione tra modi differenti. È possibile eseguire analisi di valutazione di tipo finanziaria ed economica, quantificare gli impatti, come ad esempio i livelli di emissione, per determinare la qualità del servizio, e compiere delle stime di tipo costi/benefici. Con questo software, inoltre, è

possibile applicare modelli predittivi di più recente sviluppo, come i modelli basati sulle attività.

È commercializzato per Windows e Linux da INRO, un'azienda con sede a Montreal in Canada, che opera in questo settore da più di 30 anni [55]. Oltre a Emme, propone ai professionisti altri due software: il primo è Dynameq, che permette di simulare dinamicamente il traffico in modo avanzato basandosi sul singolo veicolo, ciò consente di modellare in maniera molto precisa le congestioni infrastrutturali; il secondo è CityPhi, che crea visualizzazioni e animazioni interattive dei risultati ottenuti.

Attualmente Emme è alla versione 4.3, ma il supporto al formato OMX era presente già a partire dalla versione 4.1.5 [23]. Attraverso due strumenti del software è possibile importare le matrici OMX, scegliendo il file corretto e la relativa mappatura, ed esportarle creando i relativi metadati, come il nome, la descrizione della matrice, ecc. e associargli la mappa degli indici secondo la zonizzazione effettuata nel programma.

#### 1.1.5.2 PTV VISUM

È un software dedicato all'analisi della domanda e dei flussi di traffico attesi in un sistema GIS [83]. Copre diversi campi di applicazione, anche se non è completo come il precedente software Emme. È possibile creare o importare i grafi che rappresentano la rete delle infrastrutture, prevedere la domanda attraverso il noto modello a quattro stadi<sup>17</sup> e valutare

---

<sup>17</sup> Il modello a quattro stadi è una semplificazione, che permette di suddividere la modellazione dei flussi di traffico in un'area di studio, per l'appunto in quattro "step". Difatti, pensare di costruire un unico modello che, a partire dai dati raccolti, fornisca direttamente i flussi su ogni infrastruttura rappresentata nel grafo è troppo complesso. Il problema viene quindi diviso in quattro fasi, a cui corrispondono i relativi modelli, che sono: generazione, da cui si ricavano quanti spostamenti vengono generati o accolti da una zona (ovvero i totali di riga e di colonna delle matrici O/D); distribuzione, da cui si ottiene la suddivisione dei precedenti nelle singole zone (si determinano dunque le singole celle delle matrici O/D); scelta modale, che permette lo smistamento dei precedenti nei vari modi considerati (ottenendo così delle matrici O/D unimodali); assegnazione, che prevede l'attribuzione degli spostamenti ai vari archi del grafo.

gli scenari riprodotti. È possibile inoltre monitorare il rapporto costi/benefici dei servizi offerti, ma anche effettuare valutazioni microscopiche, come la progettazione delle intersezioni [82]. Consente di effettuare delle analisi economiche, che comprendono: la definizione dell'offerta e della domanda; l'ottimizzazione delle linee pubbliche; la generazione di dati per valutazioni macroeconomiche; la creazione di un database per le analisi di fattibilità economica e sostenibilità ecologica.

La sigla PTV (Planung Transport Verkehr) raggruppa in realtà una serie di software, tutti sviluppati dall'omonimo gruppo fondato nel 1979 in Germania [65], che coprono diversi campi del settore dei trasporti, ad esempio: la modellazione degli spostamenti effettuati a piedi con PTV Viswalk; la simulazione del traffico con PTV Vissim; la gestione in tempo reale degli spostamenti con PTV Optima; e molti altri ancora.

Il supporto al formato OMX, in questo programma, avviene attraverso due componenti aggiuntivi [32], denominati “*OMX import*” e “*OMX export*”, rispettivamente per importare ed esportare qualunque matrice o parte di essa. Questi due strumenti permettono inoltre di archiviare anche i metadati, le mappature degli indici e qualsiasi altro attributo associato. Attualmente viene rilasciata la versione 16 del software, ma questi due strumenti erano incorporati già nelle precedenti versioni, più precisamente dalla 14.00-08.

### 1.1.5.3 TRANSCAD

È un software progettato specificatamente per l'uso professionale. Serve per archiviare, visualizzare, gestire e analizzare dati di trasporto, a qualsiasi scala o livello di dettaglio. È un programma che permette di affrontare diverse problematiche di settore, come lo possono essere ad esempio l'analisi della rete, la pianificazione e la modellazione della domanda, l'analisi del transito e la gestione del territorio [35]. Per svolgere tali compiti, integra strumenti GIS, con estensioni specifiche per il settore dei trasporti, strumenti di mappatura, visualizzazione e analisi e moduli applicativi per la previsione della domanda, la logistica e la

gestione del territorio [36]. Con questo software è possibile analizzare reti e matrici senza aver limiti nelle dimensioni. È molto aperto per quanto riguarda i formati supportati; infatti nell'elenco si possono contare oltre 50 tipi di file e oltre 100 formati GIS e CAD (tra cui i file SHP e DXF).

È stato sviluppato nel 1988 da Caliper Corporation, un'azienda fondata nel 1983 con sede a Newton negli Stati Uniti [33]. TransCAD non è l'unico software proposto dall'azienda: gli si affiancano anche Maptitude, un programma di mappatura completo progettato per la visualizzazione e l'analisi geografica dei dati (in questo caso anche non necessariamente legati al settore dei trasporti) e TransModeler, un pacchetto per la simulazione del trasporto pubblico e del traffico di auto e camion per tutti i tipi di reti stradali, dalla rete urbana alle autostrade.

TransCAD è disponibile solo per Windows e garantisce la lettura e la scrittura dei file OMX dall'ultima versione rilasciata da Caliper, ovvero la 7.0 [31]. Questo è possibile mediante il GISDK (Geographic Information System Developer's Kit) [34], che raccoglie degli strumenti per gli sviluppatori di sistemi GIS. Con questo è possibile richiamare più di 700 funzioni, precompilate e pronte all'uso, o crearne delle altre attraverso il Caliper Script, che consiste in un linguaggio di programmazione dedicato alla progettazione delle *macro*. Queste sono delle procedure, ovvero una successione di comandi, simili a delle funzioni, che se richiamate, permettono di compiere determinate operazioni al programma. Con il Caliper Script è possibile scrivere e modificare il codice di calcolo, ovvero le istruzioni contenute nelle *macro*, attraverso un qualunque editor di testo. Riassumendo, il GISDK non solo consente di abilitare la scrittura e la lettura delle matrici OMX, ma permette, più in generale, di estendere e di personalizzare il software TransCAD attraverso la scrittura di brevi codici di programmazione.

#### 1.1.5.4 CUBE

Si tratta di un software per la modellazione di molti aspetti relativi alla pianificazione, all'ingegneria dei trasporti e alla gestione del territorio. Integra diversi strumenti che permettono di prevedere la domanda, simulare il traffico e rappresentare con sistemi GIS quanto prodotto [75]. Più precisamente, permette di gestire i dati di input, nella loro creazione, modifica e visualizzazione e gestire gli scenari, documentando le ipotesi di input e producendo i relativi risultati. Offre soluzioni per diverse scale, è possibile eseguire analisi di dettaglio come riprodurre intere reti regionali. Con questo software è possibile osservare gli sviluppi e le conseguenze a seguito di una modifica alla rete di trasporto, alla gestione del territorio e alla popolazione [38]. Risulta molto simile a Emme. È composto da una serie di moduli, ciascuno dei quali risponde a una specifica problematica: Cube Voyager, per l'implementazione di modelli macroscopici su scala regionale e per lo studio delle esigenze di viaggio su larga scala; Cube Avenue, per la simulazione macroscopica delle città e per la modellazione delle dinamiche della congestione durante le ore di punta; Cube Dynasim, per la simulazione microscopica del traffico e delle operazioni di parcheggio; Cube Land, per la previsione dei cambiamenti sul territorio a seguito di modifiche al sistema di trasporto; Cube Cargo, per la modellazione del movimento delle merci nelle città o su scala regionale; Cube Analyst, per la stima diretta delle matrici di trasporto; e infine Cube Cloud, per la condivisione sicura dei modelli creati e dei risultati ottenuti.

Cube, e i relativi moduli appena descritti, sono stati tutti sviluppati da Citilabs Ltd, un'azienda operante da quarant'anni nel settore della mobilità, con sede a Sacramento in California ed uffici sparsi in diverse città del Mondo come Atlanta in Georgia, Tallahassee in Florida, Abu Dhabi negli Emirati Arabi e Milano [37]. Oltre a Cube produce altri software, sempre legati al settore dei trasporti, come Sugar, che rappresenta una soluzione per la gestione, l'analisi e la visualizzazione delle reti di trasporto, Flow, che è utile per il monitoraggio e la previsione delle prestazioni, delle operazioni e dei budget, e Streetlytics,

che serve a ricavare informazioni in tempo reale, attraverso la tecnologia GPS installata su cellulari e veicoli, per ricostruire un quadro completo e quanto più fedele dei movimenti reali della popolazione.

Solo con l'ultima versione sviluppata di Cube [20], ovvero la 6.4.3 rilasciata solo recentemente (ottobre 2017), è supportata la lettura e la scrittura diretta dei file OMX. Prima di questa, ciò era consentito attraverso un'applicazione denominata "*Cube OMX Converter*" [21], la quale rappresentava solo una soluzione provvisoria fino al momento del supporto nativo.

#### 1.1.5.5 AIMSUN NEXT

Si tratta di un software integrato utile nella pianificazione strategica delle reti di trasporto e nella gestione della mobilità. Le sue applicazioni più comuni comprendono [16]: la valutazione e l'ottimizzazione degli schemi TSP<sup>18</sup> e BRT<sup>19</sup>; gli studi di fattibilità per corsie HOV<sup>20</sup> e HOT<sup>21</sup>; le analisi di impatto a seguito di modifiche alla rete; le analisi di impatto ambientale; la valutazione dei costi di viaggio; la valutazione delle strategie di gestione della domanda; le analisi di sicurezza; la valutazione degli ITS<sup>22</sup>; e molto altro.

---

<sup>18</sup> Con la sigla TSP (Transit Signal Priority) si rappresenta un insieme di tecnologie, che mirano a ridurre il tempo di sosta negli incroci stradali regolati da semaforo.

<sup>19</sup> Con la sigla BRT (Bus Rapid Transit) si rappresentano una serie di sistemi di trasporto pubblico, che offrono un servizio più veloce rispetto al classico autobus metropolitano.

<sup>20</sup> Con il termine HOV (High-Occupancy Vehicle) si indica una corsia, o più, dedicata al transito di veicoli che presentano più passeggeri a bordo. Questo sistema dovrebbe incentivare le persone a condividere l'auto privata, al fine di ridurre le congestioni delle ore di punta.

<sup>21</sup> Con il termine HOT (High-Occupancy Toll) si indica l'equivalente delle corsie HOV, applicate ai pedaggi stradali. In questo caso i veicoli condivisi tra più passeggeri possono godere di tariffe agevolate ai caselli.

<sup>22</sup> Con il termine ITS (Intelligent Transportation System) si designano degli strumenti, che integrano le conoscenze nel campo dell'elettronica, delle telecomunicazioni e dell'informatica con l'ingegneria dei trasporti. Lo scopo di tale fusione è il miglioramento della sicurezza delle persone e dei veicoli (o delle merci).

Aimsun Next viene commercializzato dall'omonima azienda; nata in un laboratorio di ricerca dell'Università Politecnica della Catalogna, attorno ai primi anni '90 [14], con lo sviluppo di un prototipo di micro-simulatore denominato AIMSUN2 (Advanced Interactive Microscopic Simulator). Insieme ad Aimsun Next, fornisce un ulteriore software chiamato Aimsun Live [15], che permette di elaborare dati in tempo reale, come suggerisce il nome, simulando il movimento dei veicoli all'interno della rete stradale. È possibile scaricare dal sito ufficiale dell'azienda anche una versione di prova del programma Aimsun Next<sup>23</sup>.

Il supporto al formato OMX da parte di questo software è avvenuto con l'ultima versione rilasciata dall'azienda, la 8.2, che risale all'aprile del 2017 [17]. Con questa variante del programma è possibile infatti leggere e scrivere delle matrici Origine/Destinazione direttamente nel formato *open*.

Questi rappresentano i primi esempi di sviluppo del formato OMX. Si presuppone che in futuro altri software aprano le proprie porte a questo formato di salvataggio; magari in occorrenza dello sviluppo di una loro nuova versione. Questo perché, quanto più un software è aperto, tanto più è facilitato lo scambio di informazioni e dati per l'utenza e tanto più è appetibile il loro acquisto da parte dei professionisti di settore.

## 1.2 INTERFACCE DI PROGRAMMAZIONE

È possibile effettuare delle operazioni sulle matrici memorizzate con il formato OMX, senza passare attraverso uno dei software descritti nel capitolo precedente; ciò è garantito dalle APIs supportate da tale formato. Queste sono semplicemente delle interfacce di

---

<sup>23</sup> Aimsun, *Aimsun Next free trial version*, <https://www.aimsun.com/aimsun-next/free-trial/>, data di consultazione: 17/01/2018.



programmazione, ovvero dei linguaggi, con i quali è possibile manipolare in modo diretto le matrici OMX. Permettono di compiere delle operazioni essenziali sulle tabelle, esprimendo i comandi con l'idioma proprio del linguaggio scelto. È possibile creare e riempire una matrice, modificarla, eliminarla o semplicemente accedere ai dati contenuti. Le stesse operazioni possono essere effettuate anche sui metadati opzionali associati e sugli attributi di sistema del file matriciale.

Il formato OMX supporta diversi linguaggi di programmazione, tra cui C++, C#, Java, Python, R e Ruby, di seguito brevemente descritti:

- \* Il C++ è un linguaggio di programmazione molto utilizzato; è stato sviluppato, inizialmente con il nome di *C with classes*, nei primi anni '80 da Bjarne Stroustrup in collaborazione con i laboratori di ricerca e sviluppo Bell [92];
- \* Il C#, ovvero il *C sharp*, è un linguaggio di programmazione orientato agli oggetti che accoglie sia le strutture dati, che le procedure che operano su di esse, per creare un software autonomo dotato di dati e metodi. È stato sviluppato nel 2000 da Microsoft [91];
- \* Il Java è un linguaggio di programmazione, nonché una piattaforma di elaborazione, attualmente molto diffuso che è stato sviluppato nel 1995 da Sun Microsystems, un'azienda con sede a San Francisco [58];
- \* L'R è un linguaggio di programmazione nato per il trattamento e l'analisi di grandi moli di dati. È stato sviluppato negli anni '90 da Robert Gentleman e Ross Ihaka, come versione *open source* di un precedente linguaggio proprietario denominato S (sviluppato negli anni '70) [69].
- \* Il Ruby è un linguaggio di programmazione *open source* orientato alla semplicità di scrittura. È stato sviluppato nel 1993 in Giappone da Yukihiro Matsumoto, e rappresenta il primo linguaggio nato in oriente [70].

Python, descritto ampiamente nel seguente paragrafo, è il formato scelto per le applicazioni richieste dal presente lavoro di tesi, e per questo motivo è quello a cui si dedica più spazio.

### **1.2.1 Caratteristiche del linguaggio di programmazione Python**

Python è un linguaggio di programmazione, semplice ma potente, presentato nel 1999 dal ricercatore informatico olandese Guido van Rossum (ma nasce agli inizi degli anni '80 con il nome "ABC" [51] nel Centrum Wiskunde & Informatica<sup>24</sup> di Amsterdam), che ha riscosso grande successo grazie alla sua adattabilità. Gli ambiti di applicazione sono molteplici: dalla semplice elaborazione di testi, alla creazione di web browser interattivi, allo sviluppo di applicazioni desktop, alla realizzazione di interfacce grafiche, al calcolo numerico e scientifico, alla gestione di database, giochi e grafica 3D, e molto altro [74]. È un linguaggio minimalista di agevole scrittura e lettura, è facile da imparare grazie alla sua sintassi semplificata che permette di concentrarsi sulla risoluzione del problema piuttosto che sulla struttura della lingua impiegata. Tra gli impieghi più noti si possono nominare NASA, Yahoo e Google.

È un linguaggio *free* ed *open source*, ovvero FLOSS (Free/Libre and Open Source Software), che si basa sul principio della libera condivisione delle informazioni. Infatti, è possibile distribuire copie del software, accedere al codice sorgente, modificarlo e copiare parti di codice per utilizzarle nella scrittura di altri programmi. Python viene definito un linguaggio di alto livello, questo significa che, durante la scrittura di un programma, è possibile ignorare la gestione delle risorse di basso livello, come può esserlo il controllo dello spazio di memoria usata dal programma stesso. Inoltre, è un linguaggio orientato agli oggetti. Esistono infatti due tipologie di programmazione: la prima è chiamata

---

<sup>24</sup> Il Centrum Wiskunde & Informatica (CWI) è il principale istituto di ricerca nazionale per la matematica e l'informatica, fondato nel 1946 nei Paesi Bassi [43].

“procedurale”, in tal caso il programma è costruito mediante delle funzioni, ovvero delle procedure, estrapolate da altri programmi; e un’altra tipologia, definita “orientata agli oggetti”, nel quale il programma è sviluppato mediante degli oggetti, detti *array*, costituiti da variabili e vettori, combinati tra di loro attraverso delle funzioni. Python, inoltre, riesce automaticamente a gestire la memoria, ciò permette di usare liberamente le variabili senza doversi preoccupare di dichiararne il tipo e di definirne la dimensione dello spazio di memoria dedicato (cosa che invece accade in altri tipi di linguaggi come il C o il C++).

Un’altra caratteristica di Python è che non necessita di produrre un codice binario per poter essere eseguito dal computer, cosa che invece accade con i programmi scritti con altri linguaggi come il C o il C++. In pratica il programma viene eseguito richiamando il relativo codice sorgente, questo viene quindi convertito automaticamente dal calcolatore in una forma intermedia chiamata *bytecode* (anche Java opera in tale modo), la quale, a sua volta, viene infine tramutata in un linguaggio macchina utile all’esecuzione da parte del computer. La portabilità di Python ne ha permesso un grande sviluppo; il linguaggio è stato infatti modificato diverse volte, nelle varie versioni che si sono succedute, per poter girare su molte piattaforme informatiche. Attualmente è possibile utilizzare questo linguaggio di programmazione in Linux, Windows, Macintosh, ma anche nei meno popolari FreeBSD, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, PalmOS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE e PocketPC.

L’ultima caratteristica degna di nota, del linguaggio di programmazione Python, è la sua estensibilità. Possiede infatti un’enorme libreria standard di funzioni che coprono molteplici e differenti aspetti. A questa si affiancano altrettante librerie disponibili apertamente, consultabili nel *repository* PyPI (Python Package Index) e installabili in modo molto semplice attraverso un *tool* chiamato “pip”. Senza contare il fatto che è comunque sempre possibile scrivere dei moduli aggiuntivi per ulteriori funzionalità specifiche. Quest’ultimi, inoltre, possono essere redatti con il linguaggio di Python o con altri

linguaggi, come il C o il C++, questo per avere una maggiore rapidità nell'esecuzione di quella parte di codice (non dovendo passare attraverso il *bytecode*) o per rendere inaccessibile l'algoritmo stesso, e richiamati nello *script* di Python con delle semplici sintassi.

Al momento della scrittura del presente lavoro, le versioni correnti di Python in commercio sono due: la 2.7 e la più recente 3.6 [68]. Viene consentito il download della versione 2, ormai sorpassata, semplicemente perché dei pacchetti software di terze parti potrebbero non essere ancora compatibili con Python 3. Entrambe sono scaricabili dal sito ufficiale di Python nella sezione *Download*<sup>25</sup>. Maggiori dettagli sull'uso pratico di questo linguaggio di programmazione sono forniti nel Capitolo 4.

### 1.3 APPLICAZIONI DEL FORMATO OMX

Nonostante la grande pubblicizzazione del formato OMX, operata dal suo sviluppatore Ben Stabler, inizialmente attraverso la quinta conferenza “*Innovations in Travel Modeling*” del TRB, avvenuta nel 2014, ma anche su web, attraverso la pagina ufficiale del formato<sup>26</sup>, non ne sono purtroppo pubblicizzati i suoi impieghi più rilevanti. È noto, attraverso i precedenti capitoli, che comunque diverse aziende sviluppatrici di software per la pianificazione, hanno adottato, o hanno in previsione di farlo nelle versioni future, l'OMX come formato di salvataggio dei dati matriciali prodotti. I fatti, dunque, indicherebbero una consistente diffusione di questa specifica; altrimenti non si spiegherebbe questa popolarità tra la concorrenza proprietaria.

L'unico caso di impiego del formato OMX, abbastanza documentato, che si è reperito, è quello del FAF, acronimo di Freight Analysis Framework. Il FAF3 consiste in un database

---

<sup>25</sup> Python, *Scarica Python*, <https://www.python.it/download/>, data di consultazione: 22/06/2017.

<sup>26</sup> Ben Stabler, *Home*, <https://github.com/osPlanning/omx/wiki>, data di consultazione: 20/06/2017.

(il 3 ne indica il numero dell'edizione) di flussi di merci tra gli stati e le diverse aree metropolitane degli Stati Uniti, ed eventualmente tiene in conto anche quelli da e verso regioni straniere, che avvengono con tutti i modi di trasporto [44]. È sviluppato attraverso la collaborazione di due agenzie del governo federale statunitense: la BTS (Bureau of Transportation Statistics) e la FHWA (Federal Highway Administration). Generalmente questo tipo di dati non viene fornito come matrice, ma in forma tabellare, ma ciò non ha frenato l'ambizione di voler provare ad utilizzare il formato OMX per questi dati raccolti.

Nello specifico, sono rappresentati i movimenti delle merci tra 123 zone interne all'area di studio e 8 zone esterne, per un totale, dunque, di 131 zone. Abitualmente questi dati sono organizzati in una serie di documenti (di Microsoft Access), che riportano i flussi tra coppie di zone, utilizzando tre parametri come unità per la quantificazione: il valore delle merci trasportate; le tonnellate; e le miglia effettuate dalle stesse. I movimenti sono distinti sulla base del tipo di merce, del tipo di commercio (se import, export o interno), del modo di trasporto e della modalità di importazione. L'analisi dei flussi è stata eseguita in diversi anni, inizialmente nel 1997, 2002, 2007 e dal 2010 in modo regolare ogni cinque anni, con la previsione di coprire l'intervallo temporale fino almeno al 2040. Da queste informazioni è possibile quantificare la mole di dati raccolta da questo database; che per un solo anno corrisponde a: 3 sistemi di misura (valore, tonnellate e miglia), per 131 zone, per 43 diverse classi merceologiche, per 8 modalità di trasporto.

Le informazioni riportate nel database comprendono: la zona di origine e il modo di importazione (se il bene proviene dall'estero); la zona di destinazione e il modo di esportazione (se il bene è destinato al mercato estero); la zona di origine interna e la zona di destinazione interna del bene; il modo con cui è stato trasportato nelle zone interne; il tipo di merce; il tipo di commercio (di importazione, esportazione o interno). Si tratta dunque di una base dati che può tranquillamente essere convertita sotto forma di matrici OMX.

Questa operazione è stata eseguita attraverso uno *script* in R, che ha letto i dati del

FAF3 e ha creato un file OMX con una matrice per ognuna delle tre unità di misura. Compiuta questa operazione, BTS ed FHWA, hanno testato l'efficacia di questo formato valutandolo attraverso una serie di aspetti [19]. Una prima osservazione ha riguardato lo spreco di spazio: infatti, rappresentando i flussi con una matrice, le celle riguardanti gli spostamenti tra due zone estere sono risultate essere vuote, poiché tali dati non erano oggetto di indagine. Un secondo aspetto negativo riguarda l'impossibilità di maneggiare i dati per poterne ricavare le informazioni necessarie. Con il database a disposizione invece è possibile aggregare con facilità i risultati. Inoltre, nel database, i valori che corrispondono alle tre diverse unità di misura sono racchiusi nella stessa cella, mentre nel file OMX, come è stato anticipato, è necessario creare tre diverse matrici. Unica nota positiva che è stata riscontrata è che, con tale formato, il file risulta essere di minori dimensioni rispetto al formato di testo originario, nello specifico il file OMX ha occupato uno spazio di memoria pari al 23% del precedente (più precisamente 49 MB contro 213 MB).

## 1.4 CONCLUSIONI

Il formato OMX rappresenta dunque un grande vantaggio per la memorizzazione di dati in formato matriciale e, più precisamente, per quanto ne riguarda lo scambio e lo spazio di memoria occupato. Attualmente è supportato da diversi software commerciali, e si prevede che in futuro, a questo elenco, se ne aggiungano molti altri. L'ottica non proprietaria del formato ne costituisce di fatto l'innovazione e il punto di forza; concretizzando quello che probabilmente era il sogno di molti professionisti del settore.

Per quanto è deducibile dalla presentazione del formato Open Matrix da parte dei suoi proponenti, sembrerebbe che si possano dare solo giudizi positivi. Per quanto visto invece nel precedente paragrafo, si può dire che non sempre questa specifica può rappresentare la scelta migliore tra i vari formati di archiviazione. Ovviamente, per poterlo analizzare in

maniera critica, è necessario provare a creare e maneggiare in prima persona delle matrici OMX. Per questo motivo ogni commento legato alla sua praticità viene rimandato al Capitolo 4 del presente lavoro.

## 2. SOLUZIONI ESISTENTI

Parallelamente allo sviluppo dei sistemi informatici, si è resa ovviamente sempre più necessaria la creazione di sistemi di archiviazione dati. Col passare degli anni, inoltre, la quantità di informazioni da immagazzinare si è fatta sempre più importante. Ciò ha contribuito alla nascita di formati di salvataggio sempre più specifici.

Di seguito sono riportati alcuni formati che possono rappresentare una soluzione al problema dell'archiviazione di grandi quantità di dati, con particolare attenzione alle matrici/tabelle e a quelli più comunemente utilizzati nel settore dei trasporti. L'elenco di formati esplicitati nell'omonima sezione web del progetto OMX<sup>27</sup>, ha fornito la base di partenza per lo sviluppo di questa rassegna.

### 2.1 HIERARCHICAL DATA FORMAT: HDF

Lo Hierarchical Data Format (HDF) è una suite tecnologica per la gestione di database grandi e complessi. È realizzato attraverso una serie di software, tutti sviluppati dallo stesso produttore, The HDF Group, che rispondono a esigenze diverse di un comune ambito lavorativo [78]; ovvero la memorizzazione, l'organizzazione, l'analisi, la condivisione, e più in generale la gestione, di grandi quantità di dati, anche **eterogenei** e in continua

---

<sup>27</sup> Benjamin Stabler, *OMX – Existing Solution*, <https://github.com/osPlanning/omx/wiki/Existing-Solution>, data di consultazione: 20/06/2017.



evoluzione.

Tale formato di dati era stato originariamente sviluppato presso il National Center for Supercomputing Applications (NCSA); un centro di ricerca transdisciplinare fondato nel 1986 nell'University of Illinois [61]. Tutt'ora rappresenta un punto di incontro tra vari collaboratori, che si uniscono per affrontare ricerche legate al campo dell'informatica. Ma attualmente, il formato HDF è supportato dall'omonimo gruppo, una società creata verso la fine degli anni '80 e composta da circa 30 collaboratori provenienti da ambiti scientifici diversi [76]. Con sede a Champaign, in Illinois, presso l'University of Illinois Research Park; è una società senza scopo di lucro, la cui unica missione è garantire il continuo sviluppo delle tecnologie HDF e l'accessibilità continua dei dati memorizzati con tale formato.

Nel 1987 la Graphics Foundations Task Force (GFTF), appartenente all'NCSA, decise di creare una libreria di software e un formato di file indipendente, con lo scopo di agevolare lo scambio e la trasmissione dei dati scientifici tra le varie piattaforme informatiche utilizzate proprio all'interno dell'NCSA [96]. Gli obiettivi di tale formato erano: la memorizzazione e l'accesso efficiente a grandi banche dati; la registrazione, in un unico contenitore, di molti dati anche di diverso tipo; la capacità di accogliere i metadati degli oggetti memorizzati; e infine l'accesso a tali dati attraverso un linguaggio di programmazione (inizialmente erano stati scelti il C<sup>28</sup> e il Fortran<sup>29</sup>). Dapprima tale formato prese il nome di AEHOO (All Enhanced Hierarchical Object Oriented format), fu

---

<sup>28</sup> Il Linguaggio C è un linguaggio di programmazione progettato nel 1972 da D.M. Ritchie, presso i laboratori AT&T Bell in New Jersey (USA) ed è stato riconosciuto come standard dall'American National Standards Institute (ANSI C) nel 1983 [90].

<sup>29</sup> Il Fortran (da Formula Translator) è un linguaggio di programmazione pubblicato nel 1957 da un gruppo di lavoro guidato da John Backus, un informatico della IBM (International Business Machines Corporation), la nota azienda statunitense che opera nel settore informatico [54].

sviluppato in modalità *open source* e distribuito gratuitamente dall'Università dell'Illinois. L'idea originaria combinò le caratteristiche di una serie di formati già esistenti, tra cui TIFF<sup>30</sup>, CGM<sup>31</sup>, FITS<sup>32</sup> e PICT<sup>33</sup>. Al momento della creazione, la NCSA produsse anche una serie di strumenti accessori, ad esempio per la visualizzazione e la presentazione dei dati memorizzati.

La grande utilità portò tale formato alla popolarità e a una rapida diffusione in tutti gli Stati Uniti [96]; tanto che nel 1990 e nel 1992 la National Science Foundation<sup>34</sup> fornì delle sovvenzioni allo scopo di migliorare ancora di più la qualità complessiva dell'HDF. Questa fu la svolta che portò il formato al di fuori degli utilizzi meramente accademici. All'inizio degli anni '90, la NASA (National Aeronautics and Space Administration), tra 15 diversi formati di memorizzazione dati, scelse proprio l'HDF per il progetto EOS (Earth

---

<sup>30</sup> Il TIFF (Tagged Image File Format) è un formato di archiviazione di immagini *raster* (immagini assimilabili a delle "griglie" dove a ogni cella, definita *pixel*, viene associato, attraverso un codice numerico, un colore). È stato sviluppato nel 1986 dalla Aldus Corporation, ora assorbita dalla Adobe Systems Incorporated, una società pubblica con sede a San Jose in California [18].

<sup>31</sup> Il CGM (Computer Graphics Metafile) è un formato di archiviazione di immagini *raster* e immagini vettoriali (immagini definite attraverso elementi geometrici come punti, linee e poligoni ai quali possono essere attribuiti colori e sfumature) che, per lo scopo, utilizza un *metafile*, ovvero un file di testo, che ne contiene le informazioni di base e le "istruzioni" per la lettura da parte del calcolatore. È stato sviluppato nel 1999 dal W3C (World Wide Web Consortium), un'organizzazione non governativa internazionale con sede a Cambridge. [87]

<sup>32</sup> Il FITS (Flexible Image Transport System) è un formato di archiviazione di immagini, soprattutto a scopo scientifico, che permette di includere informazioni aggiuntive al file, come indicazioni fotometriche (utili nel campo dell'astronomia) e altro. È stato sviluppato nel 1981 da IAU FITS Working Group, un'autorità internazionale di controllo composta da diversi membri provenienti da altrettante nazioni [53].

<sup>33</sup> Il PICT (Macintosh Picture) è un formato di archiviazione di file grafici (*raster* e vettoriali), che si avvale anch'esso di un *metafile* per lo scopo. È stato sviluppato nel 1984 da Apple in California, e introdotto nel primo calcolatore Apple Macintosh come formato standard per lo scambio di file tra le diverse applicazioni [46].

<sup>34</sup> La National Science Foundation (NSF) è un'agenzia federale statunitense indipendente fondata nel 1950, con lo scopo di sostenere e di promuovere il progresso in tutti i settori scientifici e ingegneristici (escluse le scienze mediche) [62].

Observing System); il quale avrebbe dovuto raccogliere informazioni utili allo studio del cambiamento climatico globale, per un totale di tre terabyte di dati a giorno in decine di formati diversi.

Il progetto originale fu revisionato diverse volte, sino ad arrivare nella sua quarta versione, denominata HDF4, tuttora attivamente supportata dal The HDF Group. Tale formato ha però diverse limitazioni, che sono state superate nel 1998 con il più recente HDF5 (originariamente chiamato Big HDF). Nel 2002 l'HDF5 permise al gruppo di vincere il *Research and Development 100 Award*, un premio che lo etichettò come “*uno dei 100 nuovi prodotti tecnologicamente significativi del 2002*”.

Mentre l'HDF diventava popolare, come ci si poteva aspettare, numerosi strumenti commerciali e non, vennero estesi per leggere e scrivere dati in tale formato. Ad esempio, in MATLAB<sup>35</sup> venne aggiunto un supporto per la lettura e la scrittura in HDF4 e HDF5. Inoltre, dal 2006, per questo programma, l'HDF5 diventò il formato standard per il salvataggio dei dati di dimensioni superiori ai 2 gigabyte.

Secondo quanto riportato nel sito ufficiale dell'HDF5 “*la suite dell'HDF5 include:*

- \* *un modello di dati versatile, che può rappresentare dati anche molto complessi e i relativi metadati;*
- \* *un formato di file portatile e illimitato per tipo e quantità; una libreria software, che funziona su diverse piattaforme computazionali;*
- \* *un insieme di funzionalità e prestazioni integrate, che ottimizzano i tempi di accesso e lo spazio di archiviazione;*

---

<sup>35</sup> MATLAB (Matrix Laboratory) è un software per il calcolo numerico, molto utilizzato nel campo scientifico e ingegneristico, che consente di operare con funzioni e algoritmi. Questo avviene attraverso un semplice linguaggio di programmazione (tipo il C) che permette di esprimere le operazioni con matrici e vettori in modo diretto. È stato sviluppato nel 1984 da MathWorks in Massachusetts (USA) [60].

\* *delle applicazioni per la gestione, la manipolazione, la visualizzazione e l'analisi dei dati della raccolta;*

e inoltre permette lo sviluppo di driver aggiuntivi da parte dell'utente che ne permettono un'implementazione personalizzata” [79]. Tutti questi strumenti sono distribuiti dal Gruppo HDF apertamente e senza costi, con licenza di tipo BSD<sup>36</sup>; attualmente le versioni scaricabili sono due [77], l'HDF5-1.8 e il più recente HDF5-1.10<sup>37</sup>. Questi due formati, sono entrambi supportati da molte piattaforme commerciali e linguaggi di programmazione, come ad esempio: Java, MATLAB, Scilab<sup>38</sup>, Octave<sup>39</sup>, Mathematica<sup>40</sup>, IDL<sup>41</sup>, Python, R, C, C++ e Julia<sup>42</sup>.

L'HDF5, in pratica, semplifica la struttura dei file per includere solo due tipi di oggetti principali: i *dataset*, che sono i vettori multidimensionali, ovvero i “contenitori”, che raccolgono i dati di tipo omogeneo; e i gruppi, che possono contenere diversi *dataset* e altri

---

<sup>36</sup> Le licenze BSD, delineate dalla Berkeley Software Distribution, sono una famiglia di licenze permissive per software senza *copyleft*; cioè un modello di gestione dei diritti d'autore, che permette di usare, diffondere e modificare liberamente l'opera in oggetto.

<sup>37</sup> The HDF Group – Support, *Downloads*, <https://support.hdfgroup.org/downloads/>, data di consultazione: 09/11/2017.

<sup>38</sup> Scilab è un software per il calcolo numerico in ambito ingegneristico e scientifico. Sviluppato in Francia a partire dagli inizi degli anni '80, sotto il nome originario di Blaise da INRIA (Institut National de Recherche en Informatique et en Automatique), un istituto francese per la ricerca matematica e informatica. A partire dai primi anni '90 ha subito anche il contributo dell'ENPC (École Nationale des Ponts et Chaussées), la più famosa istituzione francese per la formazione di ingegneri [72].

<sup>39</sup> Octave è un software, con annesso linguaggio di programmazione, destinato principalmente al calcolo numerico, che sia esso lineare o meno. È stato sviluppato nel 1988 negli Stati Uniti da John W. Eaton, con il contributo di altri collaboratori; mentre la prima versione ufficiale è stata pubblicata soltanto nel 1993 [48].

<sup>40</sup> Mathematica è un ambiente di calcolo, ideato nel 1988 da Stephen Wolfram, che utilizza un linguaggio di programmazione proprio, che riporta il nome del suo ideatore [103].

<sup>41</sup> L'IDL (Interactive Data Language) è un linguaggio di programmazione specializzato nel trattamento di dati astronomici, è stato sviluppato nel 1977 da David Stern [97].

<sup>42</sup> Julia è un recente linguaggio di programmazione per il calcolo numerico. È stato sviluppato nel 2012 da un team di lavoro composto da: Jeff Bezanson; Stefan Karpinski; Viral B. Shah; e Alan Edelman [58].

gruppi. È dunque un formato di archiviazione dati di tipo gerarchico. I metadati invece sono memorizzati sotto forma di attributi sia dei gruppi che dei *dataset*.

In conclusione, si può affermare che l'HDF5 è un formato molto noto e ben supportato, che dispone inoltre di un visualizzatore, il quale permette di visionare il contenuto anche a chi non dispone del pacchetto software (come accade per l'OMX). Esistono inoltre numerosi convertitori già implementati che trasformano diversi tipi di file nell'HDF5. Probabilmente, la grande popolarità e l'apprezzamento che questo formato ha avuto dagli esordi fino ad oggi, sono stati i fattori che hanno spinto il gruppo di Stabler a sceglierlo come fonte di ispirazione per il formato che desideravano realizzare.

### **2.1.1 Applicazioni nel campo dei trasporti**

L'HDF5 è sostanzialmente un “contenitore vuoto” in cui è possibile aggiungere tabelle di dati in modo gerarchico, secondo la struttura: dataset; gruppi di dataset. Ogni contenitore ha un nome e può avere una quantità illimitata di metadati a esso collegati. Tale semplice struttura ne ha favorito l'uso anche nel campo dei trasporti, infatti è attualmente utilizzato in diverse applicazioni legate a tale settore; di seguito ne sono illustrati alcuni esempi, tratti dal sito web del progetto OMX<sup>43</sup>.

1. **SFCTA.** La San Francisco County Transportation Authority (SFCTA) è un'agenzia di pianificazione dei trasporti nell'omonima contea [71]. Attualmente utilizza il formato HDF5 per la memorizzazione di tutto l'archivio delle matrici di trasporto. In realtà gli spostamenti vengono analizzati con Cube, e solo successivamente convertiti in formato HDF;

---

<sup>43</sup> Benjamin Stabler, *HDF5*, <https://github.com/osPlanning/omx/wiki/HDF5>, data di consultazione: 20/06/2017.

questo per sfruttarne la compressibilità al fine di ottenere dei file di dimensioni ridotte. Questo richiede un tempo di elaborazione aggiuntivo poiché con Cube non è possibile leggere e scrivere matrici direttamente con il formato HDF (cosa che si ricorda è invece attuabile con il formato OMX nelle ultime versioni del programma, si veda il Paragrafo 1.1.5.4 Cube).

2. **PSRC.** La Puget Sound Regional Council (PSRC) è un'agenzia di pianificazione regionale dei trasporti e dello sviluppo economico, con sede a Seattle negli Stati Uniti [66]. Utilizza il formato HDF5 per lo scambio di tutti i dati raccolti, anche non necessariamente legati a questo specifico settore. In particolare, per l'archiviazione dei dati di mobilità, l'agenzia utilizza il software Emme che permette di salvare i file con tale formato.

Questi sono due esempi di come il formato HDF5 è attualmente utilizzato da diverse agenzie e software per memorizzare e scambiare i dati legati al settore dei trasporti e non. Il vantaggio di tale formato sta proprio nel suo largo impiego, il che lo rende molto popolare; grazie anche al fatto che è disponibile in maniera *freeware* ed *open source*. Un punto di debolezza invece potrebbe essere proprio rappresentato dalla sua grande versatilità; un formato *ad hoc* per la rappresentazione dei flussi di traffico, attraverso le matrici di trasporto, universalmente riconosciuto, potrebbe risultare più vantaggioso.

## 2.2 NETWORK COMMON DATA FORM: NETCDF

Il Network Common Data Format, o in breve NetCDF, rappresenta un'interfaccia per una libreria di funzioni di accesso a dei dati memorizzati sotto forma di array. Gli utenti, o i software, che accedono a questi dati possono visualizzarli, elaborarli e analizzarli. È utilizzato soprattutto per memorizzare dati di tipo scientifico; è infatti comunemente usato in applicazioni climatologiche, meteorologiche e oceanografiche, e in tutti quei campi dove

usualmente sono impiegate delle applicazioni GIS [98]. Infatti, ne rappresenta anche un possibile formato di *input/output*.

NetCDF è stato sviluppato da Unidata; una comunità, nata all'inizio degli anni '80, composta da ricercatori e istituti d'istruzione, con il comune obiettivo di trovare delle soluzioni alla condivisione di dati scientifici e di fornirne gli strumenti per l'accesso e la visualizzazione [86]. È finanziata principalmente dalla National Science Foundation (NSF); un'agenzia governativa degli Stati Uniti che sostiene la ricerca e la formazione in tutti i campi della scienza e dell'ingegneria, ad esclusione della medicina (che possiede un'agenzia a parte). Più precisamente, Unidata rappresenta uno dei programmi comunitari della University Corporation for Atmospheric Research (UCAR). Il progetto del formato NetCDF è iniziato nel 1989, ed è ancora attivamente sostenuto dalla UCAR. È stato originariamente basato sul modello del Common Data Format sviluppato dalla NASA, ma con gli il passare delle versioni, si è scostato da esso, ed attualmente i due formati non sono nemmeno compatibili. Al momento, le versioni del NetCDF supportate sono due: la versione 3, rilasciata nel 1997, ma aggiornata recentemente nel 2012; e la versione 4, rilasciata nel 2008, che rappresenta quella con più novità rispetto alle precedenti. Unidata ha comunque garantito che l'accesso a tutte le versioni passate del formato sarà comunque sempre supportato sia dalle versioni correnti che future. L'ultima versione rilasciata in questo momento, è rappresentata dalla 4.5.0.

NetCDF è un formato auto-descrittivo [64], cioè contiene le informazioni sui dati che accoglie. Vi è un'intestazione che descrive tutto ciò che riguarda i dati (formato e dimensione), nonché i metadati associati a essi, all'interno dello stesso file. Grazie a questa caratteristica è possibile individuare e visualizzare in maniera più rapida le informazioni ricercate. È possibile assegnare alle variabili dei nomi, come attributi, per rendere più chiaro il file, e le relative unità di misura, in una forma che permette il loro uso anche nei calcoli. Questa intestazione accoglie anche informazioni sulla versione del formato che si sta

utilizzando; questo poiché, come verrà spiegato più avanti nel Grafo, tra le diverse versioni vi sono notevoli differenze e incompatibilità. Nonostante l'intestazione obbligatoria dei file NetCDF, risultano comunque compatti, richiedendo pochissimo spazio per la memorizzazione di questi dati ausiliari che li rendono auto-descrittivi.

Il formato NetCDF è anche portatile, ciò significa che un file può essere letto da piattaforme differenti, anche con diverse modalità di memorizzazione dei dati (come numeri interi, a virgola mobile, ecc.). Si basa su uno standard simile all'XDR<sup>44</sup>, per la descrizione dei dati memorizzati, che permette di codificare i dati in sequenze di *bit* indipendenti dalla macchina. Le librerie fornite da Unidata permettono quindi l'accesso ai dati dei file NetCDF, sia in lettura che in scrittura, attraverso la codifica e decodifica degli *array* e dei metadati. Tutte le operazioni per accedere e manipolare i dati, devono utilizzare, però, solo il set di funzioni fornite dall'interfaccia. La libreria di base è scritta in C, ma sono disponibili anche per C++, Fortran, Java, R, Perl<sup>45</sup>, Python, Ruby, Haskell<sup>46</sup>, Mathematica, MATLAB, IDL e Octave.

Come anticipato, è molto importante specificare nell'intestazione, al momento della creazione del file, il formato a cui si fa riferimento; anche se è possibile modificarlo in un secondo momento. Le librerie NetCDF infatti supportano tre diversi formati binari [63]:

- \* il formato definito "classico", utilizzato sin dalla nascita di questo standard; ne rappresenta il formato predefinito, poiché è quello con la massima portabilità.

---

<sup>44</sup> L'External Data Representation (XDR) è un formato standard di serializzazione dei dati, sviluppato a metà degli anni '80 da Sun Microsystems, che permette di trasferire i dati tra sistemi informatici differenti. La conversione, dalla rappresentazione locale a questo formato, è chiamata "codifica", il processo inverso "decodifica". È implementato come una libreria di funzioni, portatile tra i diversi sistemi operativi [94].

<sup>45</sup> Il Perl, anche detto Practical Extraction and Report Language, è un linguaggio di programmazione ad alto livello, ispirato al linguaggio C, creato nel 1987 da Larry Wall [99].

<sup>46</sup> L'Haskell è un linguaggio di programmazione funzionale, creato nel 1990 da un comitato, i cui due principali fondatori sono Simon Peyton Jones e Paul Hudak [95].



È identificato attraverso la dicitura “CDF \ 001” all’inizio del file.

- \* il formato binario a 64-bit, introdotto nella versione 3.6.0 (quelle precedenti quindi non lo possono decodificare); anche se molto simile al precedente, permette il salvataggio ulteriori tipi di dati e la creazione di file di dimensioni maggiori. È identificato attraverso la dicitura “CDF \ 002” all’inizio del file.
- \* il formato NetCDF-4 / HDF5, introdotto dalla versione 4.0, utilizza lo standard dell’HDF5, anche se con alcune limitazioni. Gli utenti del formato NetCDF possono dunque creare dei file HDF5, sfruttando i vantaggi non disponibili con il primo formato, come: la possibilità di creare gruppi di dati; tipi composti; matrici di lunghezze variabili; variabili illimitate; nuovi tipi di dati (ad esempio il “VLEN” che permette delle lunghezze variabili); l’accesso in input/output parallelo e molto altro. Non da meno è la possibilità di creare file di dimensioni maggiori. I file NetCDF-4, infine, possono essere letti direttamente dal formato HDF5, se si utilizza la versione 1.8.0 o successive.

Una vasta gamma di software e applicativi utilizzano e/o leggono i file NetCDF. Ne sono stati sviluppati anche diversi visualizzatori; per citarne qualcuno: NcBrowse; Ncview; Panoply. Tra i software comunemente utilizzati dagli operatori del settore dei trasporti si trova ArcGIS; le versioni successive alla 9.2 possono leggere i file NetCDF semplicemente scaricando il relativo *plugin*.

## 2.3 STRUCTURED QUERY LITE: SQLITE

SQLite è un sistema *open source* di gestione di database (abbreviato in DBMS – Database Management System), caratterizzato da una libreria di programmazione in linguaggio C. È incorporato in varie applicazioni per l’archiviazione di molti tipi differenti di dati. Un database SQLite è costituito da un file binario accessibile direttamente dalla macchina; si

può creare una raccolta di dati, come variabili, tabelle e vettori, in un unico file *cross-platform*<sup>47</sup> su una macchina *host*<sup>48</sup> [73]. Il codice di SQLite è di pubblico dominio, e per tanto può essere utilizzato per qualsiasi scopo, commerciale o privato. La libreria è molto compatta; e occupando poco spazio, è spesso impiegata nella gestione dei database su strumenti con memoria limitata come i telefoni cellulari.

La libreria di SQLite è supportata, oggi, da una società di ingegneria chiamata Hwaci, ma è stata pubblicata per la prima volta nel 2000, da D. Richard Hipp. L'idea di questo formato nacque mentre progettava un software per la Marina Militare degli Stati Uniti da utilizzare a bordo di un cacciatorpediniere. L'obiettivo fondamentale dell'SQLite doveva essere quello di consentire il funzionamento di questo programma, senza richiedere l'installazione di un ulteriore sistema di gestione dei database. Hipp strutturò la sintassi di questa libreria prendendo spunto da PostgreSQL<sup>49</sup>.

SQLite, quindi, non è un sistema autonomo, ma è necessariamente incorporato in un altro programma, di cui ne diventa parte integrante attraverso delle semplici chiamate di funzioni. È stato sviluppato, come già accennato, con linguaggio C, ma negli anni è stato implementato anche per molti altri linguaggi, come ad esempio il C++. Include un programma tipo *prompt*, presenta un'interfaccia a linea di comando con la quale è possibile eseguire dei comandi da tastiera, che consente di accedere al database anche manualmente.

---

<sup>47</sup> Nel settore dell'informatica con il termine *cross-platform* si indica un software multiplatforma, o indipendente dalla piattaforma. Per possedere tale caratteristica sono possibili due strade: una consiste nel compilare lo script per due o più piattaforme; l'altra consiste nella scrittura attraverso un linguaggio interpretato comune come può esserlo il *bytecode*. Ad esempio, un'applicazione *cross-platform* può essere eseguita sia in Windows x86, che in Linux x86 e in MacOS senza restrizioni [93].

<sup>48</sup> Con il termine *host*, in informatica, si identifica un dispositivo collegato ad una rete (come può esserlo la rete internet). Lo sono, ad esempio, computer, cellulari, tablet e web TV.

<sup>49</sup> PostgreSQL è anch'esso un DBMS con licenza libera. È stato implementato dall'omonimo gruppo di lavoro, il PostgreSQL Global Development Group, ma è nato nel 1982, con il nome di INGRES, come progetto dell'Università di Berkeley in California [100].

Con questo strumento è possibile creare un database, definire delle tabelle, inserire e modificare righe e colonne.

Un database è costituito da un unico file, che può raccogliere più tabelle e indici; inoltre l'indicizzazione è consentita sia per righe che per colonne. Una caratteristica particolare di SQLite è la sua gestione flessibile dei tipi di dati; infatti non solo si possono avere tabelle con dati differenti in un unico file, ma anche all'interno della stessa, ogni campo può contenere qualsiasi tipo di informazione. È una qualità insolita per un database; normalmente, infatti, si assegna un preciso tipo a una colonna, mentre in questo caso i tipi vengono assegnati ai singoli valori. In termini informatici, questa caratteristica viene indicata con il termine "digitazione dinamica". In realtà, nel caso si volessero tenere sotto controllo i dati inseriti, sembrerebbe una proprietà poco vantaggiosa.

Esistono delle applicazioni multiplatforma, che consentono di gestire in modo semplificato i database memorizzati, con delle interfacce grafiche appositamente sviluppate per SQLite; ne rappresentano alcuni esempi SQLite Database Browser, SQLiteManager e SQLiteDBManager.

SQLite viene utilizzato in moltissime applicazioni [101], come: nei browser Chrome e Mozilla Firefox dove è impiegato per memorizzare varie informazioni di configurazione (cronologia, segnalibri, cookie, contatti, ecc.); in Skype; nelle applicazioni di Adobe come file di formato per Adobe Photoshop Lightroom, Adobe AIR e Adobe Reader. Date le ridotte dimensioni, è inoltre impiegato per la gestione di SMS e MMS, del calendario, della cronologia delle chiamate e per la memorizzazione della rubrica sui sistemi iOS e MacOS.

## **2.4 BINARY JAVASCRIPT OBJECT NOTATION: BSON**

Con il termine BSON si identifica un formato binario di scambio di dati informatici,

utilizzato principalmente per la memorizzazione di database. Permette di rappresentare strutture di dati semplici e *array* associativi. I documenti BSON, altro non sono che liste ordinate di elementi, dei quali si deve definire il nome, il tipo e il valore. I tipi di dati che è possibile memorizzare sono molteplici, si affiancano ai classici interi, double e stringhe, altri formati come: date; *byte array* (dati binari); oggetti BSON; *array* BSON [89]. Non esiste invece un tipo numerico “universale”, che accetti l’inserimento sia di numeri interi che con virgola. Grazie all’utilizzo di formati riconosciuti dal linguaggio C, la codifica e la decodifica dei file BSON, avviene in maniera diretta e rapida nella maggior parte dei linguaggi di programmazione [41]. Il nome BSON prende spunto da JSON e ne rappresenta la versione binaria. Quest’ultimo è anch’esso un formato per lo scambio di dati, basato su un sottoinsieme del linguaggio di programmazione Java.

Sia il JSON che il BSON sono formati di trasferimento nel database MongoDB. Si tratta di un DBMS orientato ai documenti; differente dai classici database basati su tabelle. Si tratta infatti di una raccolta di documenti e *array*, che possono essere incorporati in altri documenti e *array*. È stato sviluppato nel 2007 dall’omonima società di software, e presenta licenza libera e *open source*.

Rispetto al JSON, il BSON è progettato per occupare meno spazio di memoria. Ma in realtà, dedicando uno spazio a dimensione fissa per la memorizzazione delle informazioni ausiliarie (al fine di velocizzare la ricerca nel documento), può capitare che per piccoli database, il formato JSON occupi meno spazio di memorizzazione rispetto alla corrispondente versione binaria.

## 2.5 ZIP MATRIX: ZMX

Il Zip Matrix è un formato di file utilizzato per memorizzare le matrici in forma compressa. È stato sviluppato dalla multinazionale di ingegneria e design Parsons Brinckerhoff, la stessa

azienda con cui collaborava Benjamin Stabler al momento della nascita del formato Open Matrix.

Secondo quanto riportato proprio da quest'ultimo *“questo formato utilizza i file zip come contenitori; i documenti non vengono “smontati”, ma alloggiati per intero nelle cartelle. In questo modo però l'accesso ai dati avviene in maniera abbastanza lenta, poiché non è possibile approcciarsi direttamente ai contenuti. Per poter aprire un file ZMX, è solamente necessario disporre di una libreria che legga i file compressi zip. Le matrici che tale formato è in grado di memorizzare, possono avere da 0 a n dimensioni. Non si pone quindi un limite all'estensione delle tabelle. L'unica restrizione imposta è che però tutte queste abbiano la stessa forma. Infatti, come accade per le matrici dei file OMX, queste devono avere tutte la stessa dimensione. I dati accolti invece possono anche essere di diverso tipo tra una matrice e un'altra. Gli indici, che permettono di identificare univocamente ogni cella di una tabella, possono essere anch'essi memorizzati all'interno della cartella compressa che le accoglie. È possibile infine assegnare dei nomi alle tabelle, in modo da identificarle e definirle univocamente, e associare una quantità illimitata di metadati, come coppie nome-valore”* [26].

## 2.6 FORMATI PROPRIETARI

Un'importante soluzione al problema del salvataggio dei dati matriciali di trasporto è sicuramente data dai produttori stessi di software. Certamente ogni software possiede però il proprio formato proprietario, che rende il file salvato leggibile solamente a chi possiede una licenza per lo stesso programma di origine dei dati. Nonostante la poca portabilità, questi formati rappresentano comunque una rapida soluzione al problema del salvataggio delle matrici. Di seguito sono riportati alcuni esempi di software, e relativi formati di salvataggio.

- \* **Cube** memorizza le matrici in un formato binario. Vengono scritte e lette

attraverso una libreria del linguaggio C++. Ogni file può contenere più matrici, ognuna delle quali è identificata attraverso un nome e un numero.

- \* In **Emme** viene utilizzato invece il formato proprietario EMX. Ogni file con questa estensione accoglie una singola matrice. L'input e l'output sono gestiti attraverso l'interfaccia di programmazione denominata Emme Modeller Python. Solo con le ultime versioni è inoltre possibile salvare le matrici a livello binario.
- \* **TransCAD** memorizza anch'esso le matrici avvalendosi di un formato binario, la cui scrittura e lettura avviene attraverso una libreria C++ come in Cube. Anche qui ogni file può contenere più matrici, ciascuna identificata attraverso un nome o un numero.
- \* In **Visum** esiste il formato a matrici binarie compresse, identificato dalle sigle \$BI o \$BK. Ogni file con queste estensioni contiene una sola matrice di dati. La compressione avviene attraverso il noto filtro zlib, e le matrici possono essere lette e scritte utilizzando la libreria di Python VisumPy. In questo specifico caso però, per accedere alle matrici, non è necessario disporre di una licenza valida del software, ma si può utilizzare la libreria per interagirvi direttamente attraverso uno script.
- \* In **MATLAB**, infine, esiste un formato binario per il salvataggio delle variabili create in un listato, chiamato MAT, accessibile esclusivamente attraverso l'uso del programma stesso. In un file con tale estensione è possibile memorizzare qualsiasi tipologia di variabile, anche con dimensioni differenti tra di loro (variabili singole, vettori e matrici), e contenente qualsiasi tipologia di dato (interi, stringhe, float, ecc.). Risulta quindi un formato molto flessibile ma d'altro canto anche molto chiuso.

## 2.7 FORMATI DI TESTO

Esiste infine un'ultima possibilità per rappresentare le matrici, che consiste in dei semplici file di testo. Per utilizzare questa tecnica di memorizzazione, le dimensioni, le celle e i metadati associati a ogni matrice devono necessariamente essere codificati come ASCII<sup>50</sup> o Unicode. Per mettere in pratica questa possibilità esistono diverse tecniche [26].

La soluzione più banale è quella che impiega dei formati simili a dei fogli di calcolo. La matrice, in questo caso, viene strutturata come una tabella di dati e organizzata in righe e colonne, che possono essere tabulate o separate con uno specifico carattere (con ad esempio “,” o con “;” o con una semplice spaziatura). L'unico problema di questa rappresentazione è la difficile lettura e gestione delle informazioni contenute. Inoltre, i metadati possono essere impliciti o espliciti; ad esempio, la dimensione della matrice può essere contenuta implicitamente nel numero di righe o di colonne presenti (metadati impliciti), oppure può essere dichiarata all'inizio del file di testo (metadati espliciti).

Questo tipo di file viene comunemente utilizzato per il semplice passaggio di dati; infatti non è possibile operarvi in maniera agevole. Il vantaggio che si ha trasformando una matrice in un file di testo, è la continuità di lettura, la quale è garantita nel tempo e non legata a una specifica versione di un qualche software proprietario. Inoltre, i file testuali, anche se non compressi, occupano poco spazio nella memoria.

## 2.8 FORMATI A CONFRONTO

I formati elencati nel precedente paragrafo posseggono tutti delle caratteristiche che li

---

<sup>50</sup> L'ASCII (American Standard Code for Information Interchange) consiste in un codice standardizzato per la codifica dei caratteri, proposto per la prima volta dall'ingegnere dell'IBM Bob Bemer, nel 1961 [88]. In pratica ogni lettera dell'alfabeto (maiuscola e minuscola), ogni numero e diversi simboli corrispondono a un numero equivalente nel codice digitale.

rendono ottimali per i diversi scopi via via presentati. Grazie ad alcune delle loro qualità, però, si prestano relativamente bene anche al salvataggio dei dati matriciali, e per questo motivo se ne è dato spazio in questo capitolo. Non essendo nati per questo scopo però, presentano alcuni svantaggi, talvolta rappresentati proprio da quelli che in realtà sono i loro punti di forza. Nella TABELLA 2.8.1 ne è riportato uno schema riassuntivo.

TABELLA 2.8.1 Pregi e difetti dei formati di salvataggio dati analizzati.

Formato	Vantaggi	Svantaggi
HDF	<i>Freeware, open source</i> , largamente diffuso, di grande versatilità, con grande capacità, eterogeneità di dati, possibilità di salvataggio di metadati, supportato da molte piattaforme commerciali e linguaggi di programmazione.	Poco specifico.
NetCDF	Autodescrittivo, permette un rapido accesso ai dati memorizzati.	Struttura rigida, necessita di intestazione per riconoscere il contenuto, poca portabilità anche tra versioni differenti dello stesso.
SQLite	Necessita di poco spazio di memoria per il salvataggio di grandi database, è molto flessibile per quanto riguarda i tipi di dati da memorizzare.	Non è un sistema autonomo ma necessita di un programma per accedere al suo contenuto, non permette nessun controllo sui tipi di dati introdotti.
BSON	Il salvataggio di dati con formati “standard” permette la rapida codifica e decodifica, necessita di poco spazio di memoria.	Studiato per una struttura da database e non da matrice, è più orientato alla raccolta di documenti che di dati numerici.
ZMX	Le tabelle non sono rimaneggiate ma solo memorizzate e compresse, permette dimensioni illimitate delle matrici.	Accesso ai dati molto lento, necessita di una libreria dedicata per leggere i file compressi.
Formati proprietari	Comodi perché studiati “su misura” per ogni software proprietario.	Non consentono portabilità tra software di diverse case proprietarie.
Formati di testo	Universalmente leggibili, continuità di lettura nel tempo, necessitano di poco spazio di memoria.	Di difficile comprensione e gestione delle informazioni raccolte.

Come è stato anticipato, ciascun formato di quelli elencati sopra riporta delle caratteristiche negative per quanto riguarda lo scopo che si sta ricercando in essi. Il formato OMX, quindi,



semberebbe posizionarsi in maniera più privilegiata rispetto a questi, poiché rappresenta un formato studiato *ad hoc* per il salvataggio dei dati matriciali legati al settore dei trasporti. Inoltre, gli sviluppatori, essendo dei professionisti attivi nel settore, avevano ben note le caratteristiche negative dei formati sino a quel tempo utilizzati, come d'altronde quelle da cui prendere spunto per l'implementazione.

Con queste premesse è possibile affermare che il formato OMX, essendo stato creato per la maggiore sullo spunto del formato HDF, ne riprende tutte le caratteristiche positive: è infatti un formato *open source* il cui scopo è quello di essere il più leggibile possibile, sia per quanto riguarda la situazione attuale, caratterizzata da molteplici piattaforme commerciali e moltissimi linguaggi di programmazione disponibili, che quella futura, garantendo l'uso di un formato che sarà sempre leggibile nel tempo. D'altro canto, l'OMX è una soluzione studiata con uno scopo ben specifico, cosa che per l'HDF non è stato, poiché implementato con l'obiettivo di salvare dati di diversi settori scientifici. In questa ottica, gli sviluppatori dell'OMX hanno così sopperito a quella grossa lacuna che il formato HDF presentava per il loro settore di impiego.

# 3. DATABASE IMQ 2013

Le IMQ (Indagini sulla Mobilità delle persone e sulla Qualità dei trasporti) sono dei sondaggi, effettuati su un campione di popolazione, con lo scopo di conoscere la domanda di trasporto e rilevare i giudizi sulla qualità percepita dei servizi offerti, sia pubblici che privati. Vengono effettuate nella provincia di Torino a intervalli temporali regolari, ogni due o tre anni, in modo da evidenziare i cambiamenti comportamentali dei cittadini al trascorrere del tempo e in risposta ai nuovi sviluppi del settore [6].

Sono indagini specifiche finalizzate alla stima diretta della domanda di trasporto; si collocano quindi nella fase conoscitiva del processo di pianificazione. Durante questa fase è necessario raccogliere tutti i dati indispensabili a descrivere lo stato dello scenario di indagine; sia per quanto riguarda la domanda che per l'offerta. A tale scopo queste indagini devono essere attentamente studiate e preparate.

Se si pensasse di rilevare tutti gli spostamenti di tutti i residenti di una regione ci si renderebbe subito conto dell'impossibilità di tale operazione. È dunque necessario ricorrere alla statistica e affidarsi a un'indagine eseguita su un campione rappresentativo della popolazione. Per una più accurata analisi risulta spesso utile suddividere la popolazione in classi, sulla base di alcuni parametri che si pensi possano influenzare l'oggetto dell'indagine; in tal caso si parla di campionamento stratificato. Per ciascuna delle classi individuate viene scelto quindi un tasso di campionamento, che può essere costante o meno, nel primo caso allora il campionamento stratificato è detto equiprobabile. Si definisce così la grandezza del campione, ovvero il numero di persone che risponderanno all'intervista. Il criterio di scelta di queste ultime deve essere del tutto casuale. Gli operatori addetti alla somministrazione

delle domande e alla raccolta delle risposte devono essere formati, in modo che il loro lavoro sia svolto in modo corretto. Le informazioni così raccolte sono inserite nei database, i cui dati possono infine essere elaborati allo scopo di estendere i risultati ottenuti all'intera popolazione.

Il processo appena descritto è parte del lavoro necessario per la definizione delle matrici Origine/Destinazione, le quali rappresentano in sintesi la domanda di trasporto dell'area di studio. Ogni cella di una matrice simboleggia un flusso, ossia la quantità di spostamenti che avvengono tra la rispettiva zona di partenza e di arrivo. Le matrici O/D possono essere diversificate sulla base del mezzo utilizzato, delle motivazioni che hanno generato gli spostamenti ed anche sulla base temporale. Possono infatti rappresentare il flusso giornaliero, annuale o quello di un intervallo orario specifico, e in tal caso l'interesse spesso ricade sull'ora di punta. Per poter ottenere tutti questi risultati, è necessario che l'indagine sia eseguita in maniera completa ed esaustiva. Le domande poste devono essere studiate e indirizzate sugli obiettivi dello studio, per permettere la raccolta di tutti i dati necessari.

L'IMQ 2013 rappresenta l'undicesima edizione, la prima di queste indagini risale al 1991. Effettuata dall'Agenzia della Mobilità Piemontese<sup>51</sup>, ha la peculiarità di essere stata eseguita, per la seconda volta nel corso di questi anni (la prima nel 2004), sull'intera regione Piemonte, uscendo dunque dai confini provinciali. Tra tutte le IMQ risulta la più estesa, in termini di campione intervistato, conta infatti più di 52 mila individui sottoposti al sondaggio.

---

<sup>51</sup> AMP (Agenzia della Mobilità Piemontese) è un consorzio tra enti territoriali nato nel 2003 con il compito di migliorare e ottimizzare il trasporto pubblico nell'area metropolitana di Torino [1]. Partecipano a tale consorzio Regione Piemonte, Comune di Torino, Provincia di Torino e 31 Comuni dell'area metropolitana (Grugliasco, Collegno, Venaria Reale, Borgaro Torinese, Settimo Torinese, San Mauro Torinese, Pino Torinese, Moncalieri, Pecetto Torinese, Nichelino, Candiolo, Beinasco, Orbassano, Rivalta di Torino, Rivoli, Alpignano, Pianezza, Druento, Leini, Chieri, Trofarello, Cambiano, Santena, Caselle Torinese, Volpiano, Baldissero Torinese, La Loggia, Carignano, Vinovo, Piobesi Torinese, Piosasco).

### 3.1 CENNI STORICI SULLE IMQ

Come anticipato, le IMQ sono state effettuate sin dai primi anni novanta a intervalli temporali regolari. La prima indagine, del 1991, e le sei successive, eseguite dal 1994 al 2004 con cadenza biennale, sono frutto del lavoro di cooperazione tra ATM<sup>52</sup>, Regione Piemonte, Provincia di Torino, Comune di Torino, Trenitalia<sup>53</sup> e SATTI<sup>54</sup>. Nel 2000 però, la *Legge Regionale n. 1* [40] ha richiesto l'istituzione di un consorzio, da cui è nata nel 2003 l'Agenzia per la Mobilità Metropolitana di Torino, ora Agenzia della Mobilità Piemontese, alla quale sarebbero stati affidati tutti i compiti in materia di trasporto pubblico locale, prima assegnati ai singoli enti aderenti. Per tanto, a partire dal 2006, le IMQ sono state, e sono tuttora, un impegno dell'AMP, quale ente responsabile per la Pubblica Amministrazione della mobilità collettiva nel territorio piemontese.

A partire dalla prima IMQ, che è stata eseguita su un campione di grande dimensione, le indagini successive sono state un'alternanza tra indagini di grandi (25-30 mila persone intervistate) e di piccole dimensioni (solo 5-7 mila persone intervistate) [3]. L'IMQ del 2013 risulta in questi termini la più estesa, in quanto ha coinvolto esattamente 52.119 individui residenti in tutta la regione Piemonte.

La maggior parte delle IMQ sono focalizzate sulla mobilità all'interno del comune di Torino e nella sua cintura. Le interviste infatti coinvolgevano i soli residenti nella provincia del capoluogo piemontese. Eccezione a queste sono l'IMQ del 2004 e quella del 2013, le

---

<sup>52</sup> ATM (Azienda Torinese per la Mobilità) è stata l'azienda principale per il trasporto pubblico locale del comune di Torino fino al 2003, data in cui confluì nel Gruppo Torinese Trasporti insieme alla SATTI.

<sup>53</sup> Trenitalia S.p.A. è la principale società italiana per il trasporto passeggeri e merci ferroviario, fa parte del gruppo Ferrovie dello Stato Italiane S.p.A. dal 2000 [45].

<sup>54</sup> SATTI (Società per Azioni Torinese Trasporti Intercomunali) è stata una società di trasporto pubblico, in servizio maggiormente nella provincia di Torino fino al 2003, data in cui venne fusa con l'ATM a formare il Gruppo Torinese Trasporti.

quali hanno riguardato la mobilità dell'intera regione, estendendo il sondaggio a tutti i residenti del Piemonte.

Nello specifico le indagini hanno permesso la raccolta di una serie di informazioni: quali dati personali sul soggetto intervistato; sugli spostamenti che questo ha effettuato in un normale giorno feriale e le opinioni e i giudizi sulla qualità di tali spostamenti. Le interviste sono state realizzate telefonicamente, ad eccezione dell'indagine del 2008 che è stata eseguita personalmente dagli intervistatori, questo con lo scopo di raggiungere anche le famiglie prive di linea telefonica fissa.

Il compito principale dell'AMP è stato quello di analizzare i dati così raccolti e di produrre degli elaborati finali, come rapporti di sintesi e matrici Origine/Destinazione. L'Agenzia ne ha poi diffuso il contenuto tra i soggetti interessati, principalmente enti consorziati e aziende di trasporto. Le banche dati elementari, nella loro interezza, sono state mantenute, fino all'indagine del 2010 compresa, ad uso esclusivo dell'Agenzia. Ciò, per quanto asserito dall'AMP nelle *Linee Guida per la diffusione dei dati sulla Mobilità* [13], per evitarne l'uso improprio dal punto di vista statistico. Si trattava infatti di indagini campionarie che, secondo l'Agenzia, non dovevano essere confuse con le indagini censuarie. Negli anni, solo le informazioni quantitative e qualitative sulla mobilità, elaborate nei rapporti finali, sono state pubblicate. Queste hanno compreso anche un numero variabile, in base alla dimensione del campione intervistato, di matrici Origine/Destinazione. Nello specifico si trattava di quattro matrici, differenziate per modo (pubblico o privato) e scopo (tutti gli scopi o tutti gli scopi esclusi i ritorni), nel caso in cui fossero basate su un campione di interviste di piccola dimensione, e sei, differenziate per modo, scopo e periodo (ora di punta o intera giornata), nel caso di grande dimensione. Eccezionalmente, e per fini esclusivamente didattici, i dati elementari relativi a vecchie indagini, potevano comunque essere accessibili per Scuole e Università. Anche per questo motivo l'IMQ del 2013 rappresenta un'eccezione. Infatti, per la prima volta, l'AMP ha reso in parte pubblici due,

dei tre database raccolti. Nello specifico si tratta di quello riguardante le informazioni personali sui soggetti intervistati e quello sugli spostamenti di quest'ultimi; continua a essere ad essere invece un'esclusiva dell'Agenzia il terzo database, quello sulla qualità.

Al momento, l'IMQ del 2013 risulta essere l'ultima effettuata. Ovviamente, l'AMP ha in previsione di effettuare delle ulteriori indagini, la loro prospettiva è quella di pubblicare un bando di gara entro il 2018. Infatti, per effettuare il sondaggio, è necessario indire un bando ad evidenza pubblica, a livello europeo, per individuare la società addetta alle interviste sul campo. Presumo quindi che la prossima campagna di interviste possa avere luogo tra l'autunno del 2018 e la primavera del 2019. I risultati di questa eventuale indagine, non potranno dunque essere disponibili prima della fine del 2019<sup>55</sup>.

### **3.2 INDAGINE CONOSCITIVA DEL 2013**

L'IMQ del 2013 è un sondaggio indirizzato sugli spostamenti delle persone residenti in Piemonte. È un'indagine che si definisce di tipo RP (Revealed Preferences), poiché si focalizza sui comportamenti reali degli intervistati. Indaga gli spostamenti eseguiti dal soggetto in un normale giorno feriale. Sono indagini alle famiglie, i soggetti intervistati sono scelti casualmente all'interno della zona di studio. L'IMQ del 2013 è definibile un'indagine di tipo CATI (Computer Assisted Telephone Interviewing), il questionario infatti è stato somministrato unicamente per via telefonica, mentre l'intervistatore, provvisto di videoterminale, ha registrato le risposte mediante un apposito software.

Le telefonate sono state eseguite fra le ore 9:30 e le 21:30, dal martedì al sabato, a partire da aprile 2013 per terminare a dicembre dello stesso anno. La fase di elaborazione,

---

<sup>55</sup> Informazioni ottenute contattando direttamente l'ufficio *Pianificazione e Marketing* dell'Agenzia della Mobilità Piemontese (AMP).

di completamento e di verifica delle informazioni raccolte ha occupato diversi mesi [2], tanto che i database non sono stati disponibili per le elaborazioni statistiche finali dell'Agenzia fino a ottobre 2014.

### **3.2.1 Campionamento**

L'IMQ del 2013 ha avuto come scopo l'individuazione dei flussi di traffico all'interno della regione nella sua interezza. La popolazione, da cui è stato estratto il campione da intervistare, ha compreso tutti i residenti del Piemonte con un'età superiore ai 10 anni. Al 31 dicembre 2012, si contavano 3.992.491 individui con tali caratteristiche, circa il 90% dell'intera popolazione regionale. Nel caso specifico gli intervistati sono stati estratti dagli elenchi telefonici dei distretti del Piemonte; i quali rappresentano quindi la base di campionamento. Tale procedura è la più diffusa, ma non la più accurata: infatti sono così esclusi tutti gli spostamenti effettuati dai residenti al di fuori della regione Piemonte, che però vi si recano quotidianamente ad esempio per motivi di studio o di lavoro; ma anche tutti i soggetti privi di linea telefonica fissa.

Poiché si è optato per un campionamento stratificato, la popolazione è stata suddivisa per: sesso (2 classi); età (4 classi); e zona di residenza (208 classi). La suddivisione sulla base dell'età ha previsto le seguenti quattro fasce: dagli 11 ai 19 anni; dai 20 ai 49 anni; dai 50 ai 64 anni; e dai 65 anni in su. Il territorio regionale è invece stato suddiviso in 208 unità, dette zone di campionamento. Per comodità di esposizione le zone campionarie sono state aggregate qui, come nel *Rapporto conclusivo* [5] redatto dall'Agenzia della Mobilità Piemontese, in 10 macro-aree, come mostrato nella FIGURA 3.2.1 e FIGURA 3.2.2 e riassunto in TABELLA 3.2.1, (si veda l'ALLEGATO B per i dettagli). Quest'ultime sono costituite dalle 7 ex province del Piemonte con l'esclusione di Torino, che è stata ulteriormente suddivisa in tre parti: comune, cintura e resto della provincia [7]. Nella TABELLA 3.2.2, TABELLA 3.2.3 e TABELLA 3.2.4 è riportata la sintesi della

suddivisione della popolazione nelle varie stratificazioni individuate.

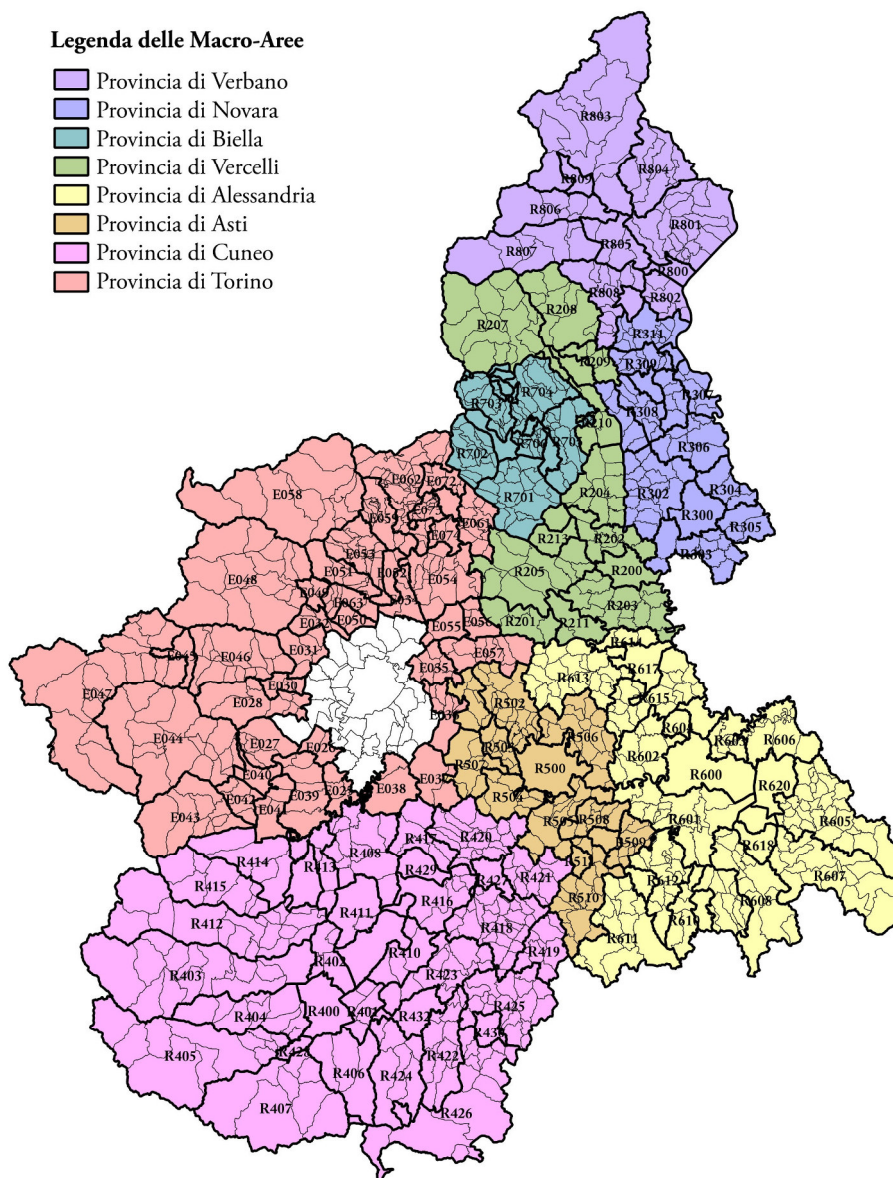


FIGURA 3.2.1 Aggregazione dei comuni piemontesi nelle zone campionarie e nelle macro-aree (ad eccezione di Torino e cintura). Elaborazione propria in QGIS basata sui dati dell'IMQ 2013 [11]<sup>56</sup>.

<sup>56</sup> Gli shapefile per la creazione della mappa in QGIS sono stati ricavati dal sito internet dell'Istat [20],  
(segue)



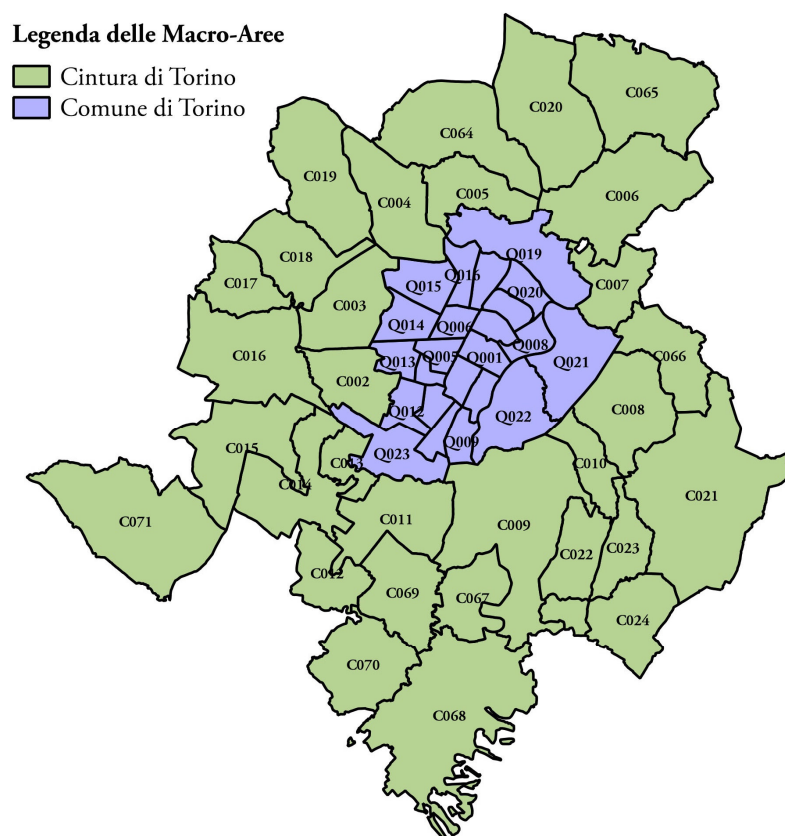


FIGURA 3.2.2 Suddivisione del comune di Torino nelle zone campionarie e individuazione dei comuni appartenenti alla cintura. Elaborazione propria in QGIS basata sui dati dell'IMQ 2013 [11].

---

riportante i confini dei comuni piemontesi nel 2013, e da quello del Comune di Torino [47], riportante la suddivisione dei quartieri torinesi.

TABELLA 3.2.1 Numero di zone campionarie per ogni macro-area individuata. Elaborazione propria basata sui dati dell'IMQ 2013 confrontabile con quanto riportato nel *Rapporto Regionale* [5] Tabella 1.

MACRO-AREA	N° ZONE CAMPIONARIE
Comune di Torino	23
Cintura di Torino	31
Resto provincia di Torino	40
Provincia di Vercelli	13
Provincia di Novara	14
Provincia di Cuneo	33
Provincia di Asti	13
Provincia di Alessandria	22
Provincia di Biella	8
Provincia di Verbano-Cusio-Ossola	11
TOTALE	208

TABELLA 3.2.2 Suddivisione della popolazione nelle stratificazioni campionarie per età e macro-area di residenza, senza distinzione di sesso. Elaborazione propria basata sui dati dell'IMQ 2013.

MACRO-AREA	11-19 anni	20-49 anni	50-64 anni	+65 anni	TOTALE
Comune di Torino	63.219	365.703	175.204	223.345	827.471
Cintura di Torino	52.135	251.901	132.929	141.419	578.384
Resto provincia di Torino	58.364	284.411	153.337	169.079	665.191
Provincia di Vercelli	12.852	66.084	37.307	44.628	160.871
Provincia di Novara	28.465	147.439	74.246	80.169	330.319
Provincia di Cuneo	49.284	230.342	116.780	133.066	529.472
Provincia di Asti	16.663	82.338	45.004	53.287	197.292
Provincia di Alessandria	29.661	157.960	89.623	114.110	391.354
Provincia di Biella	13.447	66.308	38.329	47.936	166.020
Provincia di Verbano	12.134	60.780	33.840	39.363	146.117
TOTALE	336.224	1.713.266	896.599	1.046.402	3.992.491

TABELLA 3.2.3 Suddivisione della popolazione MASCHILE nelle stratificazioni campionarie per età e macro-area di residenza. Elaborazione propria basata sui dati dell'IMQ 2013.

MACRO-AREA	11-19 anni	20-49 anni	50-64 anni	+65 anni	TOTALE
Comune di Torino	32.524	183.570	83.241	92.277	391.612
Cintura di Torino	26.943	124.862	63.605	62.899	278.309
Resto provincia di Torino	30.056	142.750	75.885	73.646	322.337
Provincia di Vercelli	6.633	33.476	18.375	18.410	76.894
Provincia di Novara	14.706	74.495	36.479	32.927	158.607
Provincia di Cuneo	25.255	116.592	58.457	58.229	258.533
Provincia di Asti	8.673	41.196	22.327	22.773	94.969
Provincia di Alessandria	15.264	79.256	44.270	48.055	186.845
Provincia di Biella	6.824	33.327	18.731	19.749	78.631
Provincia di Verbanò	6.249	30.667	16.697	16.394	70.007
TOTALE	173.127	860.191	438.067	445.359	1.916.744

TABELLA 3.2.4 Suddivisione della popolazione FEMMINILE nelle stratificazioni campionarie per età e macro-area di residenza. Elaborazione propria basata sui dati dell'IMQ 2013.

MACRO-AREA	11-19 anni	20-49 anni	50-64 anni	+65 anni	TOTALE
Comune di Torino	30.695	182.133	91.963	131.068	435.859
Cintura di Torino	25.192	127.039	69.324	78.520	300.075
Resto provincia di Torino	28.308	141.661	77.452	95.433	342.854
Provincia di Vercelli	6.219	32.608	18.932	26.218	83.977
Provincia di Novara	13.759	72.944	37.767	47.242	171.712
Provincia di Cuneo	24.029	113.750	58.323	74.837	270.939
Provincia di Asti	7.990	41.142	22.677	30.514	102.323
Provincia di Alessandria	14.397	78.704	45.353	66.055	204.509
Provincia di Biella	6.623	32.981	19.598	28.187	87.389
Provincia di Verbanò	5.885	30.113	17.143	22.969	76.110
TOTALE	163.097	853.075	458.532	601.043	2.075.747

Il tasso di campionamento è stato scelto pari al 3% nella cintura di Torino e dell'1% nel resto della regione. Questo per ottenere ampiezze degli intervalli di confidenza sulla mobilità pubblica simili, nonostante la diversa ripartizione modale attesa nelle macro-aree. In ogni caso è stato garantito un minimo di 60 interviste per ogni zona campionaria. La

dimensione del campione ottenuta è riportata nelle TABELLA 3.2.5, TABELLA 3.2.6 e TABELLA 3.2.7.

TABELLA 3.2.5 Dimensione del campione stratificato per età e macro-area di residenza, senza distinzione di sesso. Elaborazione propria basata sui dati dell'IMQ 2013.

MACRO-AREA	11-19 anni	20-49 anni	50-64 anni	+65 anni	TOTALE
Comune di Torino	622	3.655	1.750	2.213	8.240
Cintura di Torino	1.560	7.846	4.060	4.034	17.500
Resto provincia di Torino	577	2.961	1.538	1.644	6.720
Provincia di Vercelli	134	707	381	446	1.668
Provincia di Novara	277	1.533	741	780	3.331
Provincia di Cuneo	495	2.379	1.170	1.321	5.365
Provincia di Asti	161	869	468	540	2.038
Provincia di Alessandria	305	1.711	926	1.147	4.089
Provincia di Biella	137	695	386	471	1.689
Provincia di Verbano	119	638	341	381	1.479
TOTALE	4.387	22.994	11.761	12.977	52.119

TABELLA 3.2.6 Dimensione del campione MASCHILE stratificato per età e macro-area di residenza. Elaborazione propria basata sui dati dell'IMQ 2013.

MACRO-AREA	11-19 anni	20-49 anni	50-64 anni	+65 anni	TOTALE
Comune di Torino	323	1.837	829	912	3.901
Cintura di Torino	809	3.908	1.956	1.796	8.469
Resto provincia di Torino	301	1.493	768	710	3.272
Provincia di Vercelli	69	358	190	182	799
Provincia di Novara	142	782	362	317	1.603
Provincia di Cuneo	254	1.207	590	572	2.623
Provincia di Asti	83	437	233	229	982
Provincia di Alessandria	157	864	460	478	1.959
Provincia di Biella	70	352	188	191	801
Provincia di Verbano	62	323	168	156	709
TOTALE	2.270	11.561	5.744	5.543	25.118

TABELLA 3.2.7 Dimensione del campione FEMMINILE stratificato per età e macro-area di residenza.  
Elaborazione propria basata sui dati dell'IMQ 2013.

MACRO-AREA	11-19 anni	20-49 anni	50-64 anni	+65 anni	TOTALE
Comune di Torino	299	1.818	921	1.301	4.339
Cintura di Torino	751	3.938	2.104	2.238	9.031
Resto provincia di Torino	276	1.468	770	934	3.448
Provincia di Vercelli	65	349	191	264	869
Provincia di Novara	135	751	379	463	1.728
Provincia di Cuneo	241	1.172	580	749	2.742
Provincia di Asti	78	432	235	311	1.056
Provincia di Alessandria	148	847	466	669	2.130
Provincia di Biella	67	343	198	280	888
Provincia di Verbanò	57	315	173	225	770
TOTALE	2.117	11.433	6.017	7.434	27.001

Sulla base della dimensione della popolazione e di quella del campione, è stato possibile infine definire il tasso di campionamento effettivo, riportato nella TABELLA 3.2.8.

TABELLA 3.2.8 Tasso di campionamento delle stratificazioni campionarie per età e macro-area di residenza.  
Elaborazione propria basata sui dati dell'IMQ 2013.

MACRO-AREA	11-19 anni	20-49 anni	50-64 anni	+65 anni	TOTALE
Comune di Torino	0,98%	1,00%	1,00%	0,99%	1,00%
Cintura di Torino	2,99%	3,11%	3,05%	2,85%	3,03%
Resto provincia di Torino	0,99%	1,04%	1,00%	0,97%	1,01%
Provincia di Vercelli	1,04%	1,07%	1,02%	1,00%	1,04%
Provincia di Novara	0,97%	1,04%	1,00%	0,97%	1,01%
Provincia di Cuneo	1,00%	1,03%	1,00%	0,99%	1,01%
Provincia di Asti	0,97%	1,06%	1,04%	1,01%	1,03%
Provincia di Alessandria	1,03%	1,08%	1,03%	1,01%	1,04%
Provincia di Biella	1,02%	1,05%	1,01%	0,98%	1,02%
Provincia di Verbanò	0,98%	1,05%	1,01%	0,97%	1,01%
TOTALE	1,30%	1,34%	1,31%	1,24%	1,31%

NOTA: il tasso di campionamento per ogni stratificazione individuata è lo stesso per entrambi i sessi.

### **3.2.2 Questionario**

Il questionario sottoposto alle persone intervistate, contenuto nel *Capitolato Tecnico di affidamento del servizio* [12], è stato articolato in più parti. Le domande sono state suddivise in cinque sezioni.

1. **Sezione introduttiva.** Vengono riportati alcuni dati utili alla registrazione delle risposte al questionario: numero dell'intervistatore; numero dell'intervista; giorno e ora della chiamata; e numero di telefono. Viene quindi rilevata la classe a cui appartiene il soggetto intervistato: codice della zona campionaria da cui risponde; sesso; età; giorno in cui questo ha effettuato l'ultimo spostamento. Questo poiché se nella giornata precedente all'intervista il soggetto non avesse effettuato alcuno spostamento, sarebbe stato sottoposto alla sola sezione relativa ai dati personali.
2. Sezione relativa alla **mobilità**. In questa parte vengono registrate il numero di uscite effettuate, con i rispettivi indirizzi di partenza e di arrivo, nonché gli orari, per un massimo di 10 uscite. Sono stati esclusi dai conteggi gli spostamenti effettuati a piedi con durata inferiore a 5 minuti. Inoltre, nel caso in cui l'intervistato avesse svolto un'attività lavorativa di trasporto di persone o beni, gli spostamenti a scopo professionale non sarebbero stati considerati validi. Viene riportato il motivo di ogni spostamento, se sono stati usati mezzi e quali fino a un massimo di tre diversi modi, e in tal caso i luoghi e gli orari di cambio mezzo. Sono stati registrati anche ulteriori dati: il numero di passeggeri, nel caso fosse stata usata l'auto come conducente; il numero delle linee pubbliche (fino a un massimo di quattro), nel caso fosse stato usato un

bus urbano, suburbano o un tram GTT<sup>57</sup>; nel caso in cui fosse stata usata la metropolitana, da quanto tempo questa venisse utilizzata e in sostituzione di quale mezzo. Tutte queste informazioni sono state raccolte per tutti gli spostamenti effettuati.

3. Sezione relativa alla possibilità di **scelta del mezzo**. In questa parte del questionario, vengono indagati i motivi di scelta di un mezzo piuttosto che di un altro. Viene chiesto, nel caso in cui fosse stato utilizzato il trasporto pubblico in alternativa all'auto privata per gli spostamenti, se il soggetto intervistato avesse avuto comunque a disposizione un'auto. In caso contrario, viene chiesto anche se la scelta modale sarebbe stata la stessa, cioè di preferire il trasporto pubblico a quello privato. Viene registrato il motivo per il quale la scelta è ricaduta su un mezzo pubblico e il titolo di viaggio utilizzato, se biglietto singolo o abbonamento. Nel caso in cui fosse stato utilizzato il sistema di Bike Sharing in alternativa al trasporto pubblico, viene chiesto se l'intervistato avesse utilizzato un abbonamento personale o meno per compiere lo spostamento, da quanto tempo venisse sfruttato tale servizio e in sostituzione di quale mezzo. Nel caso in cui fosse stata preferita l'auto privata ai mezzi pubblici, viene chiesto il motivo di tale decisione, se fosse stato possibile sostituire, se non tutto almeno in parte, tale percorso con dei mezzi pubblici e il motivo per cui questa scelta non fosse stata effettuata. Se fossero stati usati in combinazione mezzi pubblici e privati per uno stesso spostamento, viene chiesto per quale motivo non fosse stato utilizzato un unico mezzo. Nel caso in cui fosse stato usato un mezzo del sistema di Bike

---

<sup>57</sup> GTT (Gruppo Torinese Trasporti) è un'azienda di trasporto pubblico nata nel gennaio 2003 dalla fusione di ATM e SATTI, attiva nel trasporto pubblico locale della provincia di Torino [39].

Sharing, viene chiesto se l'intervistato avesse comunque a disposizione una bicicletta personale, da quanto tempo usufruisse di tale servizio e in sostituzione a quale mezzo. Se fosse stata usata infine una bicicletta personale, viene chiesto per quale motivo la scelta fosse ricaduta su tale tipo di servizio piuttosto che sui mezzi pubblici sull'auto.

4. Sezione relativa alla **qualità del sistema**. Vengono registrate una serie di votazioni, da 1 (totalmente insoddisfacente) a 10 (pienamente soddisfacente), per diversi modi, quali: mezzi pubblici, bicicletta e auto privata. Per ciascuno di questi viene richiesto: un voto generale iniziale, una serie di voti su degli aspetti specifici, e un voto finale globale, dato alla luce di quanto affermato per i singoli aspetti. Viene chiesto se sono stati notati dei miglioramenti nel livello di sicurezza stradale e di descrivere l'iniziativa giudicata più significativa. Vengono chiesti dei voti per i parcheggi pubblici custoditi e per quelli a pagamento "a raso" della città in cui sono stati più frequentemente usati dal soggetto, ed anche quale aspetto di questi potrebbe essere migliorato. Infine, viene chiesto quali spostamenti si sarebbero potuti evitare grazie all'uso della tecnologia, ad esempio lavorando da casa grazie alla condivisione di file e cartelle, e con quale beneficio percepito.
5. Sezione relativa ai **dati personali** dell'intervistato. In questa sezione vengono registrati: il numero di auto possedute dal nucleo familiare; l'attività professionale del soggetto intervistato; se studente, quale tipo di scuola frequenta; se lavora o si trova momentaneamente in cassa integrazione; il settore lavorativo; il titolo di studio; e infine se la residenza anagrafica coincide con l'indirizzo di registrazione della linea telefonica fissa utilizzata per l'intervista.



Tutti questi dati sono stati raccolti durante lo svolgimento della telefonata grazie all'uso di uno specifico software. Sono stati quindi completati e verificati e infine organizzati nelle banche dati.

### 3.3 DATI ELEMENTARI DELLA IMQ

I dati elementari raccolti con i questionari sono organizzati in tre database. Il primo raccoglie tutte le informazioni personali dell'intervistato, il secondo i dati riguardanti gli spostamenti e il terzo le opinioni sulla qualità dei sistemi di trasporto. Come già anticipato questi prodotti diretti sono stati fino a qualche anno fa ad uso esclusivo dell'Agenzia della Mobilità Piemontese. Solo con l'ultima indagine del 2013, l'Agenzia ha deciso di rendere pubblici i primi due database. Tali dati sono reperibili online sul sito ufficiale dell'AMP in formato MDB (Microsoft Access Database) nella sezione *Dati e Statistiche – Rapporti IMQ 2013*<sup>58</sup>, insieme ai rapporti finali e ad altre elaborazioni (trattate nel Paragrafo 3.4 Elaborazioni già a disposizione dai dati IMQ) [9].

Si noti che, nonostante i dati vengano spesso definiti “elementari”, sono comunque il frutto di una fase di codifica, di verifica della coerenza e di *cleaning* delle informazioni raccolte attraverso i sondaggi. Questo al fine di ottenere dati completi, verificati, validati ed elaborabili.

Il database delle interviste contiene i dati raccolti nella sezione introduttiva dell'intervista, come sesso, età, zona campionaria di appartenenza, e quelli raccolti nella sezione relativa ai dati personali, come numero di auto possedute, attività lavorativa ecc., per ogni soggetto del campione intervistato. Il database degli spostamenti contiene tutti

---

<sup>58</sup> Agenzia della Mobilità Piemontese, *Rapporti IMQ 2013*, <http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/rapporti-imq-2013>, data di consultazione: 13/05/2017.

quei dati raccolti nella sezione relativa alla mobilità dell'intervista per ogni spostamento valido effettuato. Il database sulla qualità raccoglie infine tutte quelle informazioni ottenute nella sezione omonima dell'intervista per ogni soggetto del campione. Per comodità, le informazioni personali che identificano il soggetto nel rispettivo strato di appartenenza, insieme al tasso di campionamento, sono ripetute anche nei database degli spostamenti e della qualità.

Nello specifico i due database forniti al pubblico dall'Agenzia della Mobilità Piemontese, riportano i dati descritti nelle TABELLA 3.3.1 e TABELLA 3.3.2. Si noti che in realtà i dati reperibili online sono solo una parte dei dati raccolti attraverso le interviste.

TABELLA 3.3.1 Informazioni contenute nel *database* delle INTERVISTE (soli dati elementari forniti al pubblico). Elaborazione propria basata sui dati dell'IMQ 2013.

CAMPO	DESCRIZIONE CAMPO	FORMATO
ID_INT	Codice identificativo dell'intervista	
GIORNO	Giorno di effettuazione dell'intervista	gg/mm/aaaa
COD_ZONA_RES	Zona di residenza. Appartenenza del recapito telefonico contattato	Vedere Allegato B
RESIDENZA	Macro-area di residenza. Appartenenza del recapito telefonico contattato	Vedere Allegato B
SESSO	Sesso	1: maschile 2: femminile
FASCIA_ETA	Fascia di età	1: da 11 a 19 anni 2: da 20 a 49 anni 3: da 50 a 64 anni 4: oltre i 65 anni
TRASP	Attività professionale che consiste nel trasporto di persone o cose	1: sì 2: no null: non è uscito
Q_USC	Numero totale di uscite di casa nella giornata	
N_SPOST	Numero totale di spostamenti effettuati nella giornata.	
N_AUTO	Numero di auto disponibili in famiglia	

(segue)

(segue) TABELLA 3.3.1 Informazioni contenute nel *database* delle INTERVISTE (soli dati elementari forniti al pubblico). Elaborazione propria basata sui dati dell'IMQ 2013.

ATTIVITÀ	Attività o condizione professionale	1: imprenditore e libero professionista 2: funzionario e dirigente 3: impiegato e addetto al commercio 4: operaio 5: insegnante 6: rappresentante 7: artigiano e commerciante 8: studente 9: casalinga 10: pensionato 11: attesa del 1° impiego 12: disoccupato 13: altro
TIPO_STUD	Tipo di scuola frequentata	1: media inferiore 2: media superiore 3: corsi parascolastici regionali 4: altri corsi parascolastici 5: università e politecnico 6: corsi post-laurea null: non è studente
LAVORA	Lavoratore in attività	1: sì 2: no (è in cassa integrazione) null: non è lavoratore
SETTORE_LAV	Settore di attività	1: agricoltura 2: industria 3: commercio e artigianato 4: servizi 5: pubblica amministrazione 6: altro null: non è lavoratore
TITOLO_STUDIO	Titolo di studio	1: nessuno 2: licenza elementare 3: licenza media inferiore 4: diploma media superiore 5: laurea 6: corsi di specializzazione post-laurea

(segue)

*(segue)* TABELLA 3.3.1 Informazioni contenute nel *database* delle INTERVISTE (soli dati elementari forniti al pubblico). Elaborazione propria basata sui dati dell'IMQ 2013.

RESID_ANAGR	Residenza anagrafica	1: presso l'abitazione da cui risponde 2: in altro indirizzo ma nello stesso comune 3: in altro comune nella regione 4: altrove
TASSO_CAMP	Tasso percentuale di campionamento dello strato cui appartiene l'intervistato	

TABELLA 3.3.2 Informazioni contenute nel *database* degli SPOSTAMENTI (soli dati elementari forniti al pubblico). Elaborazione propria basata sui dati dell'IMQ 2013.

CAMPO	DESCRIZIONE CAMPO	FORMATO
ID_INT	Codice identificativo dell'intervista	
PROGR_USC	Numero progressivo dell'uscita nell'intervista	
PROGR_SPOST	Numero progressivo dello spostamento nell'uscita	
SESSO	Sesso	1: maschile 2: femminile
FASCIA_ETA	Fascia di età	1: da 11 a 19 anni 2: da 20 a 49 anni 3: da 50 a 64 anni 4: oltre i 65 anni
COD_ZONA_RES	Zona di residenza.	Vedere Allegato B
SCOPO	Scopo dello spostamento	01: recarsi al lavoro 02: motivi di lavoro 03: studio 04: acquisti e commissioni 05: accompagnamento 06: cure e visite mediche 07: sport e svago 08: ritorno a casa 09: visite a parenti e amici 10: altro 11: ritorno a casa nella giornata di intervista
COD_ZONA_PAR	Zona di partenza	Vedere Allegato B null: fuori Regione Piemonte

*(segue)*

(segue) TABELLA 3.3.2 Informazioni contenute nel *database* degli SPOSTAMENTI (soli dati elementari forniti al pubblico). Elaborazione propria basata sui dati dell'IMQ 2013.

PROV_PAR	Provincia o Stato estero di partenza	Vedere Allegato B
ORA_PAR	Ora di partenza	hh:mm
COD_ZONA_ARR	Zona di arrivo	Vedere Allegato B null: fuori Regione Piemonte
PROV_ARR	Provincia o Stato estero di arrivo	Vedere Allegato B
ORA_ARR	Ora di arrivo	hh:mm
USO_MEZZ	Uso di mezzi di trasporto nello spostamento	1: sì 2: no solo piedi
MEZZO1	Primo tipo di mezzo usato nello spostamento	01: piedi 02: ciclomotore 03: autoveicolo come conducente 04: autoveicolo come passeggero 05: taxi 06: bus urbano, suburbano, tram GTT (TO e cintura) 07: bus extraurbano GTT 08: trasporto urbano di altri comuni e altre città 09: bus extraurbano non GTT 10: bus aziendale 11: scuolabus 12: ferrovia Trenitalia 13: ferrovia GTT 15: altro 16: bicicletta personale 20: bicicletta del Bike Sharing 19: metropolitana GTT 21: Autoveicolo del Car Sharing come conducente 22: ferrovia altro vettore 23: altro mezzo pubblico terrestre 24: altro mezzo pubblico su acqua
MEZZO2	Secondo tipo di mezzo usato nello spostamento	Come sopra
MEZZO3	Terzo tipo di mezzo usato nello spostamento	Come sopra
PAX_AUTO	Numero di passeggeri in auto	null: non usata auto come conducente

(segue)

(segue) TABELLA 3.3.2 Informazioni contenute nel *database* degli SPOSTAMENTI (soli dati elementari forniti al pubblico). Elaborazione propria basata sui dati dell'IMQ 2013.

NUM_LIN_URB	Numero di linee urbane e suburbane GTT utilizzate nell'area di Torino	null: non usato TPL
COD_ALTROURB	Codice ISTAT del comune nel quale si è utilizzato un sistema TPL urbano o suburbano diverso da quello dell'area di Torino	
COD_ZONA_2M	Zona dove è stato preso il secondo tipo di mezzo	Vedere Allegato B
PROV_2M	Provincia dove è stato preso il secondo tipo di mezzo	Vedere Allegato B
ORA_2MEZZO	Ora nella quale è stato preso il secondo tipo di mezzo	hh:mm
COD_ZONA_3M	Zona dove è stato preso il terzo tipo di mezzo	Vedere Allegato B
PROV_3M	Provincia dove è stato preso il terzo tipo di mezzo	Vedere Allegato B
ORA_3MEZZO	Ora nella quale è stato preso il terzo tipo di mezzo	hh:mm
TASSO_CAMP	Tasso percentuale di campionamento dello strato cui appartiene l'intervistato	

### 3.4 ELABORAZIONI GIÀ A DISPOSIZIONE DAI DATI IMQ

I database vengono analizzati dall'Agenzia della Mobilità Piemontese al fine di ricavarne delle informazioni di sintesi da distribuire agli enti interessati e alla collettività. Queste vengono principalmente riassunte in alcuni rapporti che sono disponibili online sul sito dell'Agenzia stessa nella sezione *Dati e Statistiche - Rapporti IMQ 2013*<sup>59</sup>. Oltre a questi vengono definite delle matrici O/D, in numero variabile in base alla dimensione del

<sup>59</sup> Agenzia della Mobilità Piemontese, *Rapporti IMQ 2013*, <http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/rapporti-imq-2013>, data di consultazione: 13/05/2017.

campione di interviste.

Per l'IMQ del 2013 sono stati redatti diversi rapporti di sintesi: uno riguardante l'intera regione Piemonte; uno riguardante la sola area metropolitana di Torino; e infine un rapporto per ogni comune dell'area metropolitana. Tutte queste sintesi si trovano in formato PDF (Portable Document Format) e sono scaricabili apertamente dal sito dell'Agenzia nella già citata pagina web. Il *Rapporto sulla Regione Piemonte* [5], come è intuibile dal titolo, è un documento che presenta un quadro di sintesi sulla mobilità e sulla qualità dei servizi di trasporto percepita dai residenti in Piemonte. Fa riferimento all'indagine del 2004, poiché l'unica altra effettuata sull'intera regione, come confronto per i dati ottenuti nel 2013. I dati sono aggregati a livello territoriale nelle otto province (Torino, Alessandria, Asti, Biella, Cuneo, Novara, Verbano-Cusio-Ossola, Vercelli). Il rapporto può essere idealmente suddiviso in due parti. Nella prima parte è riportata la procedura di indagine, la descrizione dell'area di studio, della popolazione e del campione, del territorio coinvolto e della motorizzazione privata. Nella seconda parte vengono riportati invece i risultati delle analisi eseguite sui dati elementari. Questi comprendono la descrizione della mobilità dei residenti: quanto, quando e perché questi si spostano e con quali mezzi. Vengono conteggiati gli spostamenti interni alla regione, in uscita e in entrata. Infine, sono riportati i risultati relativi all'indagine sulla qualità dei trasporti: la soddisfazione percepita e i motivi di scelta di un modo rispetto un altro.

Il *Rapporto di Sintesi sull'area metropolitana* [4] riguarda invece la mobilità delle persone residenti appunto nell'area metropolitana di Torino. Questa comprende, oltre al comune di Torino, 31 comuni della cintura, aderenti al Consorzio dell'AMP (Grugliasco, Collegno, Venaria Reale, Borgaro Torinese, Settimo Torinese, San Mauro Torinese, Pino Torinese, Moncalieri, Pecetto Torinese, Nichelino, Candiolo, Beinasco, Orbassano, Rivalta di Torino, Rivoli, Alpignano, Pianezza, Druento, Leinì, Chieri, Trofarello, Cambiano, Santena, Caselle Torinese, Volpiano, Baldissero Torinese, La Loggia, Carignano, Vinovo,

Piobesi Torinese, Piossasco). Anche per questa area più ristretta vengono riportate le stesse sintesi e deduzioni del rapporto completo redatto per l'intera regione.

Infine, sono forniti 23 rapporti di sintesi [10] sui comuni di Torino e della cintura<sup>60</sup>. Questi sono in forma più breve dei due precedenti e riportano solo i grafici e le tabelle di sintesi ottenute dall'analisi dei soli dati che coinvolgono il comune in questione.

Insieme ai rapporti l'Agenzia fornisce, come anticipato, delle matrici O/D<sup>61</sup>. Nel caso dell'indagine del 2013, sono state redatte sei matrici; distinte in base al modo utilizzato (mezzo privato o pubblico), allo scopo dello spostamento (tutti gli scopi o tutti meno il ritorno a casa) e al periodo (intera giornata o ora di punta). Alcune di queste sono riportate, sotto forma di matrici, nei rapporti di sintesi appena descritti. In ogni caso, tutte queste, sono fornite dall'Agenzia sotto forma di elenco in formato XLS; hanno dunque bisogno di una elaborazione, seppur minima, da parte dell'utente per essere riassunte sotto forma di matrice vera e propria. Le sei matrici sono costruite sia tenendo conto di tutte le interviste, e quindi considerando i residenti di tutta la regione, sia considerando solo i quelli della provincia di Torino, questo per poter confrontare i risultati ottenuti con le indagini precedenti che non siano quella del 2004. Inoltre, sono distinte anche sulla base dell'aggregazione territoriale; sono fornite sia a livello di zone campionarie (quindi 208 zone di partenza e arrivo) e sia a livello di quattro macro-aree (comune di Torino, cintura, resto della provincia e resto della regione). Nel rapporto invece i dati delle matrici O/D sono aggregati nelle otto province.

Per quanto riguarda la distinzione in base al modo utilizzato, l'Agenzia ha classificato

---

<sup>60</sup>Agenzia della Mobilità Piemontese, *Report dei comuni dell'area metropolitana*, <http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/report-dei-comuni-dellarea-metropolitana>, data di consultazione: 15/07/2017.

<sup>61</sup>Agenzia della Mobilità Piemontese, *Matrici Origine/Destinazione indagine IMQ 2013*, <http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/matrici-od-imq-2013>, data di consultazione: 14/05/2017.



gli spostamenti in tre gruppi sulla base delle seguenti regole: pubblico, se è stato utilizzato almeno un mezzo pubblico, anche in combinazione con altri tipi di mezzi (ad esempio auto); privato, se è stata utilizzata l'auto come conducente o passeggero, anche in combinazione con altri tipi di mezzi, che però non rientrino nella precedente definizione di mezzo pubblico; altro, per i restanti mezzi (ad esempio piedi, bicicletta, taxi, moto, ecc.). La mobilità motorizzata invece è semplicemente definita come la somma della mobilità pubblica e della mobilità privata.

Per quanto riguarda invece la distinzione in base al periodo, il picco di mobilità mattutina è stato individuato tra le 08:01 e le 09:00, mentre quello serale, in realtà meno netto del primo, tra le 17:01 e le 18:00, si osservi a tal scopo la FIGURA 3.4.1 ottenuta elaborando il database dell'IMQ del 2013. Nello specifico, per la redazione delle matrici O/D riguardanti l'ora di punta, in accordo con quanto fatto nelle elaborazioni dell'agenzia<sup>62</sup>, si sono considerati tutti gli spostamenti con l'ora di arrivo compresa tra le 07:43 e le 08:42 [8].

Dalla TABELLA 3.4.1 alla TABELLA 3.4.7 sono riportate le matrici O/D ricavate elaborando il database dell'IMQ del 2013. Per ottenerle è bastato semplicemente creare delle Tabelle Pivot dove sulle righe si sono posizionate le origini, sulle colonne le destinazioni e infine selezionando gli opportuni filtri per creare la distinzione di modo, scopo e periodo. Queste però sono anche ottenibili in maniera meno complessa dai singoli file Excel disponibili sul sito dell'AMP. La zonizzazione considerata è quella che suddivide il territorio nelle province, la stessa dei rapporti. Non si è scelta quella più accurata che considera le 208 zone campionarie poiché di difficile rappresentazione nel capitolo. Come ci si poteva aspettare, le matrici ottenute elaborando direttamente il database sono risultate

---

<sup>62</sup> Agenzia Mobilità Piemontese, *Ora di punta*, <http://mtm.torino.it/it/dati-statistiche/faq-domande-frequenti-indagini/spostamenti/ora-di-punta>, data di consultazione: 14/05/2017.

essere identiche a quelle ricavate dai file Excel messi a disposizione nel sito, nonché a quelle riportate nel *Rapporto sulla Regione Piemonte* [5].

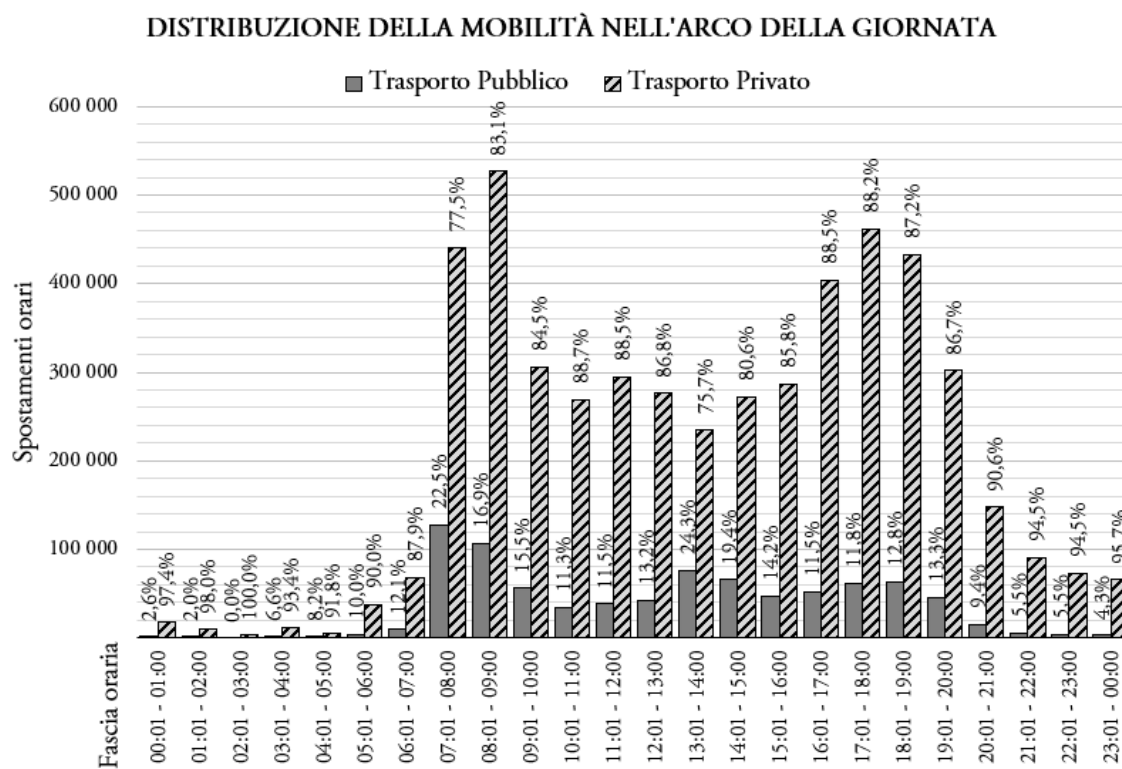


FIGURA 3.4.1 Distribuzione della mobilità pubblica e privata della regione Piemonte nell'arco della giornata. Elaborazione propria basata sui dati dell'IMQ 2013, confrontabile con quanto riportato nel *Rapporto Regionale* [5] Figura 24.

TABELLA 3.4.1 Matrice O/D degli spostamenti effettuati con MEZZO MOTORIZZATO per qualunque scopo in tutta la giornata. Elaborazione propria basata sui dati IMQ 2013, confrontabile con quanto riportato nel *Rapporto Regionale* [5] Tabella 8.

ORIGINI	DESTINAZIONI									TOTALE
	Torino	Alessandria	Asti	Biella	Cuneo	Novara	Verbano	Vercelli	Esterno	
Torino	2.999.138	6.462	14.846	4.926	31.592	3.361	290	9.953	14.862	3.085.431
Alessandria	6.462	440.807	9.023	682	1.351	1.554	97	4.933	16.752	481.661
Asti	14.747	9.023	192.514	97	8.133	100	297	197	1.443	226.551
Biella	4.926	682	97	227.797	309	1.748	297	11.639	1.764	249.260
Cuneo	31.526	1.351	8.232	309	804.356	378	107	191	4.666	851.116
Novara	3.361	1.554	100	1.748	378	452.605	6.048	14.706	22.821	503.321
Verbano	290	97	297	297	107	6.048	188.977	158	4.886	201.158
Vercelli	10.055	4.933	95	11.639	191	14.706	158	183.361	2.576	227.713
Esterno	14.862	16.848	1.443	1.764	4.795	22.821	5.003	2.576	1.063	71.175
TOTALE	3.085.367	481.757	226.648	249.260	851.213	503.321	201.275	227.713	70.833	5.897.387

TABELLA 3.4.2 Matrice O/D degli spostamenti effettuati con MEZZO PRIVATO per qualunque scopo in tutta la giornata. Elaborazione propria basata sui dati IMQ 2013, confrontabile con quanto riportato nel *Rapporto Regionale* [5] Tabella 10.

ORIGINI	DESTINAZIONI									TOTALE
	Torino	Alessandria	Asti	Biella	Cuneo	Novara	Verbano	Vercelli	Esterno	
Torino	2.357.050	4.740	10.578	4.134	22.675	2.361	290	7.053	12.574	2.421.457
Alessandria	4.740	417.213	8.588	682	1.155	1.358	97	4.289	11.663	449.786
Asti	10.464	8.402	176.085	97	7.632	100	297	197	1.443	204.718
Biella	4.037	682	97	224.030	309	1.656	297	10.740	1.165	243.014
Cuneo	22.396	1.249	7.644	309	751.200	378	107	191	4.278	787.752
Novara	2.361	1.358	100	1.656	378	427.900	5.756	12.816	17.033	469.358
Verbano	290	97	297	297	107	5.756	180.506	158	3.417	190.926
Vercelli	7.254	4.289	95	10.740	191	12.523	158	177.192	1.723	214.165
Esterno	12.530	11.759	1.443	1.061	4.407	17.246	3.535	1.723	872	54.576
TOTALE	2.421.125	449.790	204.927	243.007	788.055	469.278	191.044	214.359	54.168	5.035.752

TABELLA 3.4.3 Matrice O/D degli spostamenti effettuati con MEZZO PRIVATO esclusi i ritorni a casa in tutta la giornata. Elaborazione propria basata sui dati IMQ 2013, confrontabile con quanto riportato nel *Rapporto Regionale* [5] Tabella 14.

ORIGINI	DESTINAZIONI									TOTALE
	Torino	Alessandria	Asti	Biella	Cuneo	Novara	Verbano	Vercelli	Esterno	
Torino	1.240.306	2.263	3.995	1.761	10.690	1.287	200	3.313	12.574	1.276.390
Alessandria	2.565	213.392	2.476	188	366	480	0	2.408	11.663	233.539
Asti	6.907	6.502	90.472	0	5.452	0	105	197	1.443	111.076
Biella	2.473	494	97	118.401	309	1.161	191	6.449	1.165	130.740
Cuneo	12.298	789	2.497	0	389.964	285	0	98	4.278	410.208
Novara	1.074	878	100	495	94	222.643	3.779	7.595	17.033	253.693
Verbano	90	97	192	105	107	2.265	92.439	100	3.417	98.814
Vercelli	4.040	2.163	0	4.982	93	6.478	58	96.968	1.723	116.505
Esterno	785	194	0	0	130	384	0	292	872	2.656
TOTALE	1.270.537	226.772	99.830	125.933	407.204	234.982	96.773	117.420	54.168	2.633.620

TABELLA 3.4.4 Matrice O/D degli spostamenti effettuati con MEZZO PRIVATO per qualunque scopo nell'ora di punta (ora di arrivo compresa tra le 07:43 e le 08:42). Elaborazione propria basata sui dati IMQ 2013.

ORIGINI	DESTINAZIONI									
	Torino	Alessandria	Asti	Biella	Cuneo	Novara	Verbano	Vercelli	Esterno	TOTALE
Torino	298.369	373	1.132	479	2.519	161	0	1.067	3.313	307.414
Alessandria	1.048	52.332	672	87	0	87	0	808	2.866	57.901
Asti	1.595	1.187	22.826	0	1.326	0	0	0	193	27.128
Biella	294	99	0	24.396	0	97	0	1.168	185	26.239
Cuneo	2.996	296	401	0	100.268	189	0	0	685	104.835
Novara	100	95	0	96	94	54.700	489	1.878	4.919	62.370
Verbano	0	0	0	0	0	689	21.349	0	287	22.325
Vercelli	1.392	96	0	1.197	93	1.244	0	23.812	662	28.496
Esterno	0	0	0	0	98	0	0	0	0	98
<b>TOTALE</b>	<b>305.794</b>	<b>54.479</b>	<b>25.031</b>	<b>26.255</b>	<b>104.398</b>	<b>57.167</b>	<b>21.837</b>	<b>28.735</b>	<b>13.111</b>	<b>636.805</b>

TABELLA 3.4.5 Matrice O/D degli spostamenti effettuati con MEZZO PUBBLICO per qualunque scopo in tutta la giornata. Elaborazione propria basata sui dati IMQ 2013, confrontabile con quanto riportato nel *Rapporto Regionale* [5] Tabella 9.

ORIGINI	DESTINAZIONI									TOTALE
	Torino	Alessandria	Asti	Biella	Cuneo	Novara	Verbano	Vercelli	Esterno	
Torino	642.087	1.722	4.268	792	8.917	1.000	0	2.899	2.289	663.974
Alessandria	1.722	23.594	435	0	196	196	0	644	5.089	31.876
Asti	4.283	620	16.429	0	501	0	0	0	0	21.833
Biella	889	0	0	3.768	0	92	0	899	599	6.247
Cuneo	9.129	102	589	0	53.156	0	0	0	388	63.364
Novara	1.000	196	0	92	0	24.705	293	1.889	5.788	33.963
Verbano	0	0	0	0	0	293	8.471	0	1.468	10.232
Vercelli	2.800	644	0	899	0	2.183	0	6.169	853	13.548
Esterno	2.332	5.089	0	703	388	5.575	1.468	853	191	16.599
<b>TOTALE</b>	<b>664.242</b>	<b>31.967</b>	<b>21.721</b>	<b>6.253</b>	<b>63.158</b>	<b>34.043</b>	<b>10.232</b>	<b>13.354</b>	<b>16.665</b>	<b>861.635</b>

TABELLA 3.4.6 Matrice O/D degli spostamenti effettuati con MEZZO PUBBLICO esclusi i ritorni a casa in tutta la giornata. Elaborazione propria basata sui dati IMQ 2013, confrontabile con quanto riportato nel *Rapporto Regionale* [5] Tabella 13.

ORIGINI	DESTINAZIONI									TOTALE
	Torino	Alessandria	Asti	Biella	Cuneo	Novara	Verbano	Vercelli	Esterno	
Torino	338.801	302	65	0	1.452	435	0	860	2.289	344.203
Alessandria	1.420	11.500	97	0	94	0	0	369	5.089	18.568
Asti	4.217	430	8.314	0	501	0	0	0	0	13.463
Biella	889	0	0	1.747	0	92	0	295	599	3.621
Cuneo	7.968	102	195	0	26.403	0	0	0	388	35.056
Novara	565	196	0	0	0	12.655	0	522	5.788	19.726
Verbano	0	0	0	0	0	293	4.434	0	1.468	6.195
Vercelli	2.040	275	0	604	0	1.552	0	3.335	853	8.660
Esterno	0	214	0	0	0	0	0	0	191	405
<b>TOTALE</b>	<b>355.900</b>	<b>13.019</b>	<b>8.671</b>	<b>2.351</b>	<b>28.450</b>	<b>15.026</b>	<b>4.434</b>	<b>5.380</b>	<b>16.665</b>	<b>449.896</b>



TABELLA 3.4.7 Matrice O/D degli spostamenti effettuati con MEZZO PUBBLICO per qualunque scopo nell'ora di punta (ora di arrivo compresa tra le 07:43 e le 08:42). Elaborazione propria basata sui dati IMQ 2013.

ORIGINI	DESTINAZIONI									TOTALE
	Torino	Alessandria	Asti	Biella	Cuneo	Novara	Verbano	Vercelli	Esterno	
Torino	102.945	101	65	0	289	142	0	426	701	104.669
Alessandria	678	3.672	0	0	0	0	0	181	3.138	7.669
Asti	1.557	75	3.427	0	198	0	0	0	0	5.257
Biella	480	0	0	667	0	0	0	186	101	1.435
Cuneo	3.743	0	195	0	11.734	0	0	0	97	15.769
Novara	281	196	0	0	0	4.431	0	320	2.645	7.874
Verbano	0	0	0	0	0	97	2.090	0	176	2.364
Vercelli	778	82	0	99	0	388	0	1.323	287	2.957
Esterno	0	0	0	0	0	0	0	0	0	0
<b>TOTALE</b>	<b>110.462</b>	<b>4.126</b>	<b>3.687</b>	<b>766</b>	<b>12.221</b>	<b>5.058</b>	<b>2.090</b>	<b>2.436</b>	<b>7.145</b>	<b>147.992</b>

### **3.5 OPPORTUNITÀ DI ULTERIORI ELABORAZIONI**

Nei due capitoli precedenti sono stati mostrati nel dettaglio i dati reperibili per quanto riguarda l'indagine conoscitiva del 2013; si ricorda che per le versioni passate le uniche informazioni disponibili sono invece deducibili dai soli rapporti finali e dalle matrici O/D. L'Agenzia ha sempre elaborato personalmente i dati raccolti, e diffuso quanto ne ha ricavato secondo alcuni principi: mantenere a uso esclusivo proprio i dati grezzi nella loro interezza; segmentare i dati elementari a cui dare accesso agli enti ed aziende del settore; effettuare elaborazioni finali a disposizione della collettività.

Alla luce di tali osservazioni è possibile affermare che le elaborazioni già effettuate dall'agenzia sono utili per esigenze di tipo conoscitivo; come nel caso in cui si voglia informare in generale gli enti e le aziende settoriali, ma anche l'opinione pubblica, della situazione rilevata. L'utilità si riassume in due vantaggi: i dati sono già pronti a una lettura critica, non necessitano infatti di ulteriori elaborazioni, e di conseguenza non possono essere mal interpretati; sono "leggibili" da chiunque, non richiedono infatti conoscenze specifiche di settore.

D'altro canto, i dati elementari sono essenziali, non tanto per il generico cittadino, quanto più per enti e aziende coinvolte, per poter estrarre da questi, informazioni di loro competenza. Sono infatti necessari in caso di controllo di uno scenario di mobilità o in caso di intervento sull'offerta di servizio, poiché sono "adattabili" alla particolare situazione. È possibile infatti partire da questi e focalizzarsi su un aspetto particolare, magari omesso nelle elaborazioni effettuate dall'Agenzia.

## 4. MATRICI IMQ 2013 IN FORMATO OMX

In questo capitolo viene descritto dettagliatamente come mettere in pratica gli obiettivi di questa tesi. La base dati è stata scaricata dal sito dell’Agenzia della Mobilità Piemontese<sup>63</sup> e consiste in 24 matrici O/D prodotte sulla base dei dati dell’IMQ del 2013 (si veda l’ALLEGATO C). Queste sono differenziate per la popolazione rappresentata (residenti del Piemonte o residenti nella Provincia di Torino), per il modo utilizzato (pubblico o privato), per il periodo (ora di punta o intera giornata) e per lo scopo (tutti o escluso il ritorno<sup>64</sup>) dello spostamento. Si presentano come file di Excel (XLS) in cui i dati non sono già in forma di matrice, bensì di elenco, per ogni coppia origine-destinazione; e per questo motivo sarà necessario compiere qualche operazione in più al momento della compilazione dello *script*. Come già riportato nel Paragrafo 3.2.1 Campionamento, queste sono diversificate per macro-aree e zone campionarie. Questa differenziazione, implica ovviamente una diversità nelle dimensioni delle matrici stesse: infatti, quelle per macro-aree avranno dimensione 5x5 (Torino, cintura, resto della provincia, resto della regione ed esterno), mentre quelle per zone campionarie 209x209 (ovvero 208 zone campionarie più una zona definita “esterno” per rappresentare origini o destinazioni al di fuori del Piemonte). Poiché ogni file OMX può contenere più di una matrice, purché abbiano tutte la stessa dimensione,

---

<sup>63</sup> Agenzia Mobilità Piemontese, *Matrici Origine/Destinazione indagine IMQ 2013*, <http://mtm.torino.it/dati-statistiche/indagine-imq-2013/matrici-od-imq-2013>, data di consultazione: 14/05/2017.

<sup>64</sup> Gli spostamenti in cui è escluso il ritorno come scopo sono valutati solo per l’intera giornata.

per rappresentare entrambe le casistiche sarà necessario creare due file distinti, ciascuno di quali conterrà dunque 12 matrici.

## 4.1 AMBIENTE DI LAVORO IN PYTHON

Come è già stato anticipato nei capitoli precedenti, tra le varie API supportate dal formato OMX, la scelta del linguaggio di programmazione, per la compilazione del programma di conversione delle matrici, è ricaduta su Python. È già stato presentato nel Paragrafo 1.2.1 Caratteristiche del linguaggio di programmazione Python e ne sono già state descritte le sue caratteristiche principali, per cui, nei paragrafi seguenti, ci si dedicherà solo all'aspetto operativo di questa interfaccia.

### 4.1.1 Come ottenere Python

Il primo passo da compiere è installare Python sulla piattaforma di lavoro. Il linguaggio è scaricabile liberamente dal sito ufficiale nella pagina dei *download*<sup>65</sup>. In particolare, tra le due versioni al momento disponibili, la 2.7 e la 3.6, si è scelta l'ultima sviluppata. Poiché entrambe sono supportate per la scrittura di file OMX, non è stato il caso di basarsi su una vecchia variante.

Nella pagina dei *download* è possibile trovare i file EXE per Windows a 32 e 64-bit e per Mac OS X, anche se in quest'ultimo, come in Linux, Python dovrebbe essere già presente. Nello specifico, sulla piattaforma di lavoro utilizzata, è stata installata la versione 3.6.0 (rilasciata il 23 dicembre 2016) per Windows a 64-bit. Una volta determinato il file di installazione corretto in base alla propria piattaforma, è possibile scaricarlo, eseguirlo ed installare il prodotto seguendo la procedura guidata. Conclusa questa prima fase Python

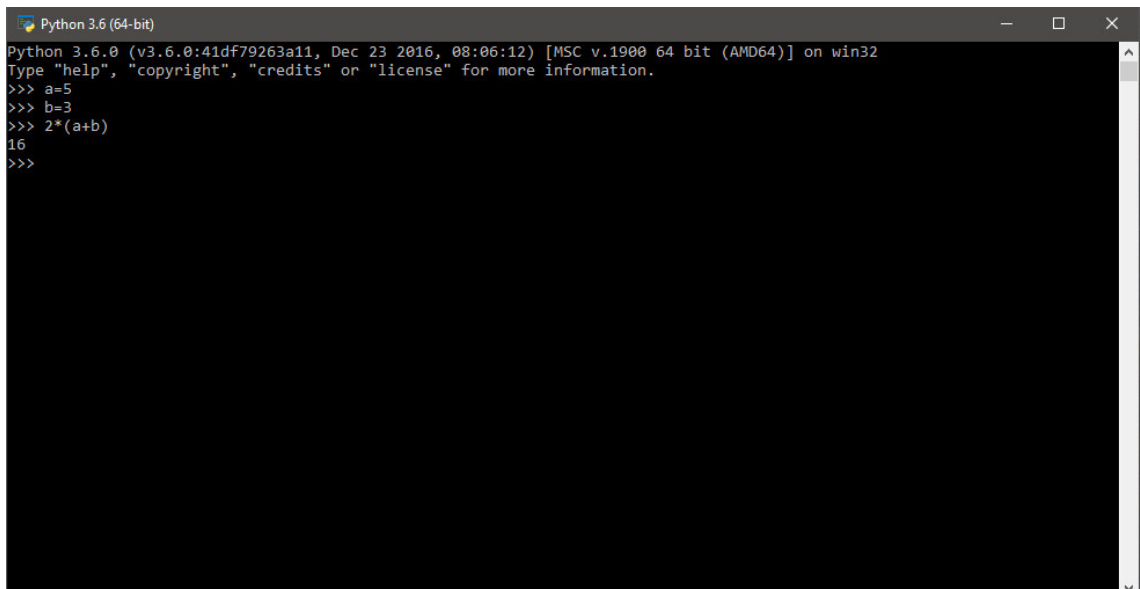
---

<sup>65</sup> Python, *Scarica Python*, <https://www.python.it/download/>, data di consultazione: 22/06/2017.

risulta pronto all'uso. Il pacchetto che si installa comprende, oltre a Python, anche l'IDLE Python Shell e il Python Module Docs, le cui funzioni saranno illustrate nei prossimi paragrafi.

### 4.1.2 Interprete interattivo e IDLE Python

Quando l'interfaccia di Python viene aperta, si presenta una schermata simile a una console, o *prompt* dei comandi, a sfondo nero, con alcune righe precompilate. Sulla prima è riportata la versione installata e la relativa data di rilascio. Sulla seconda riga, invece, sono indicati dei comandi base, da inserire da tastiera, in caso si vogliano ottenere particolari informazioni sul prodotto o aiuto (i comandi sono quattro e sono rispettivamente: "help", "copyright", "credits" e "license"). La terza riga invece si presenta con una stringa particolare, caratterizzata da tre caratteri di maggiore ">>>", che identifica un ambiente di lavoro denominato interprete interattivo. La schermata appena descritta è riportata nella FIGURA 4.1.1.



```
Python 3.6 (64-bit)
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a=5
>>> b=3
>>> 2*(a+b)
16
>>>
```

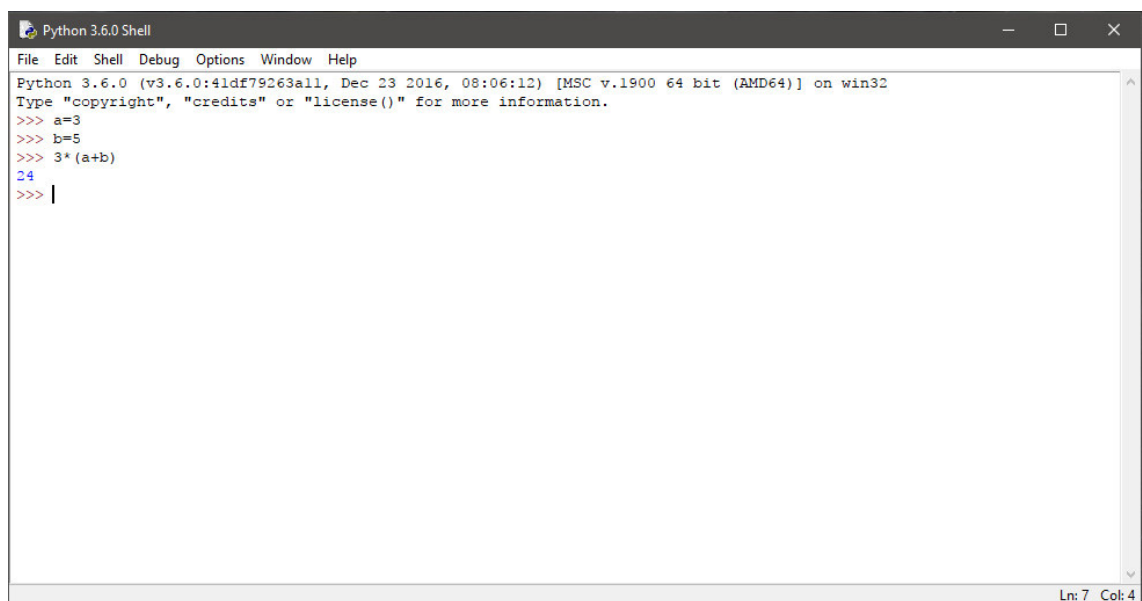
FIGURA 4.1.1 Interprete interattivo di Python ed esempio di esecuzione da riga di comando.

L'**interprete interattivo** di Python permette di eseguire in modo molto rapido delle funzioni (come quella rappresentata in FIGURA 4.1.1), attraverso l'uso del codice sequenziale. Ovvero è possibile utilizzare una serie di operazioni, il cui risultato viene fornito man mano, senza la possibilità di crearne delle altre più complesse o compiere dei salti particolari. Questo tipo di programmazione viene denominato **esecuzione da riga di comando**. Da questo ambiente è anche possibile eseguire *script* contenenti sequenze di istruzioni di Python, ovvero eseguire dei programmi già compilati. La potenzialità di questa tipologia di programmazione sta nella velocità di esecuzione di piccole istruzioni, poiché la risposta viene fornita immediatamente. Per altri scopi, che magari richiedono una programmazione più complessa, questo ambiente risulta poco adatto ed è consigliabile affidarsi alla programmazione assistita.

A tal scopo può essere utilizzato uno degli "accessori" forniti con il *download* di Python: l'**IDLE** (Integrated Development and Learning Environment). All'apertura appare esattamente come l'ambiente appena descritto, però su sfondo bianco. Anche le righe precompilate riportano le stesse diciture, e anche qui si presenta l'interprete interattivo. In FIGURA 4.1.2 è riportata la schermata dell'IDLE Python.

Il vantaggio che offre questo strumento, in realtà, è caratterizzato dalla possibilità di scrivere un codice in un file di estensione PY e di eseguirlo attraverso un comando denominato "*run*" (non ha più quindi l'esecuzione automatica col tasto "Invio"). Ciò è possibile semplicemente andando a creare un nuovo file (digitando "Ctrl+N"), o aprendone uno già esistente (digitando "Ctrl+O"). Da questo ambiente, caratterizzato come un qualunque editor di testo, è possibile modificare, eseguire e fare il *debug* degli *script* (ovvero individuare e correggere errori nella scrittura), da un'unica interfaccia. L'IDLE, inoltre, può anche essere personalizzato scegliendo dimensione e carattere del font e i colori delle parole chiave. L'output sarà, infine, visibile nel Python Shell e sarà necessario utilizzare delle funzioni di stampa a video (come "*print()*") per visualizzare i risultati attesi. In realtà,

questo sistema composto da editor di testo, per la compilazione, e *prompt*, per l'esecuzione, non è molto diffuso tra i programmatori. Per velocizzare l'esecuzione e la scrittura del codice, spesso risulta utile affidarsi a degli ambienti integrati più strutturati, come può esserlo Wing.

A screenshot of a Python 3.6.0 Shell window. The window title is "Python 3.6.0 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area shows the following text:

```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a=3
>>> b=5
>>> 3*(a+b)
24
>>> |
```

The status bar at the bottom right indicates "Ln: 7 Col: 4".

FIGURA 4.1.2 IDLE Python Shell ed esempio di esecuzione da riga di comando.

#### 4.1.2.1 WING IDE

Wing è un IDE (Integrated Development Environment), ovvero un ambiente di sviluppo integrato proprietario ma *free* (nella versione “Personal”), per il linguaggio Python. Si tratta di un software che, in fase di programmazione, aiuta il professionista nella scrittura del codice sorgente di un programma. È stato sviluppato da Wingware nel 2000 inizialmente per Linux [102]. L’interfaccia che si presenta all’utente è quella riportata in FIGURA 4.1.3, e come si può osservare, è più strutturata rispetto a quella degli ambienti presentati sinora. Consiste infatti di più componenti: un editor di codice sorgente; un compilatore; degli strumenti “*tool*”; e infine un *debugger*.

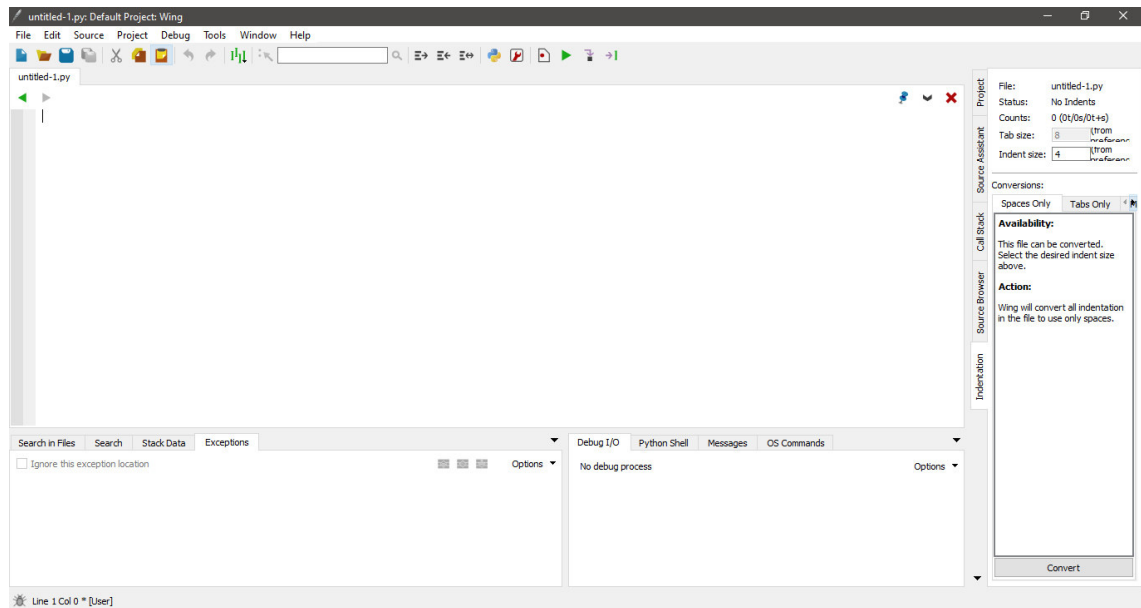


FIGURA 4.1.3 Interfaccia dell'IDE Wing.

In realtà non è necessario scaricare Wing per produrre il codice di scrittura dei file OMX, ma risulta comodo per controllare eventuali errori. Per questo motivo, anche questo software è stato installato sulla piattaforma di lavoro prima di iniziare la programmazione. Il download è disponibile nella pagina ufficiale Wingware<sup>66</sup>, e qui, in realtà, non è necessario scegliere la versione da scaricare perché ne è disponibile una sola, per Windows, indifferente se il sistema è a 32 o 64-bit, Linux e OS X. Anche in questo caso, l'installazione è piuttosto semplice, in quanto basterà eseguire passo passo la procedura guidata che partirà al lancio del file EXE.

### 4.1.3 Miniconda

A questo punto si dovrebbero avere tutti gli strumenti per programmare in Python in

---

<sup>66</sup> Wingware, *Wing python IDE Download*, <https://wingware.com/downloads/wing-personal>, data di consultazione: 12/07/2017.



maniera agevole. Ma in realtà, per compilare il codice sorgente del programma, che trasformerà le matrici in file OMX, è necessario installare alcuni **moduli** (approfonditi nel Paragrafo 4.2 Moduli e comandi principali). Uno di questi è il fondamentale OpenMatrix.

Il modo più facile per ottenerli è installare Anaconda, o la sua versione ridotta Miniconda. Questi sono semplicemente dei *prompt* che gestiscono l'ottenimento dei pacchetti aggiuntivi disponibili per Python, attraverso il sistema di gestione *conda*. Il file di installazione di Miniconda è disponibile nella pagina ufficiale Conda<sup>67</sup>; anche in questo caso è necessario prestare attenzione alla versione corretta da installare sulla macchina. Infatti, è disponibile per Windows (32 e 64-bit), Mac OS X e Linux (32 e 64-bit), per Python 3.6 e per la versione precedente 2.7. Nello specifico, sulla piattaforma di lavoro utilizzata, è stata scaricata la versione per Windows a 64-bit e Python 3.6. A questo punto è possibile eseguire l'installazione di Miniconda attraverso la procedura guidata. Alcune accortezze vanno poste in questa fase, poiché il programma deve trovare e scaricare i dati correttamente nella memoria del computer. Ad esempio, la cartella di destinazione va posta in C:\Anaconda (o l'equivalente se si usa un Mac) e, inoltre, bisogna porre due spunte nella finestra di installazione per aggiungere Anaconda al percorso di sistema e per registrarlo come predefinito per il linguaggio Python.

Al termine della procedura di installazione, però, è fondamentale compiere ancora alcune operazioni per completare definitivamente la configurazione. Dal *prompt* di Windows (o analogo in Mac OS X e Linux) è necessario eseguire questi due semplici comandi:

```
conda config --add channels conda-forge
conda update --all --yes
```

---

<sup>67</sup> Conda, *Miniconda*, <https://conda.io/miniconda.html>, data di consultazione: 19/01/2018.

Questa coppia di istruzioni aggiunge il canale *conda-forge*<sup>68</sup> e aggiorna i pacchetti preinstallati alle ultime versioni rilasciate. Ora è possibile installare i moduli di Python nel *prompt* di Anaconda digitando semplicemente: `pip install nome_modulo`.

## 4.2 MODULI E COMANDI PRINCIPALI

I moduli, conosciuti con il nome di “librerie” negli altri linguaggi di programmazione, sono semplicemente delle raccolte di funzioni di Python relative a uno stesso ambito. Consentono di semplificare il lavoro utilizzando delle istruzioni già compilate da altri programmatori, al fine di risolvere determinati compiti standard di un ambiente specifico. Python possiede già una lista di moltissimi moduli ufficiali (ad esempio, al momento della scrittura di questa tesi sono ben 129497 [67]), conosciuti anche con il nome di “*standard library*”. In realtà è anche possibile ottenerne di altri “non ufficiali” o di definirne di nuovi.

In linea di massima, coloro i quali hanno sviluppato un pacchetto per Python, nella maggioranza dei casi, lo hanno caricato su PyPI: il Python Package Index<sup>69</sup>. Si tratta del *repository* ufficiale di Python, in cui è possibile, mediante una veloce ricerca attraverso il *browser*, sfogliare la libreria standard, individuare il modulo che risponde al particolare ambito a cui si è interessati, e individuarne la corretta dicitura per l’ottenimento. Infatti, i pacchetti non vengono scaricati direttamente dal sito web della libreria, ma come si è anticipato nel paragrafo precedente, il modo più immediato per ottenerli è quello di usare `pip` dal *prompt* di Anaconda.

Pip è il *package manager* di default di Python, indispensabile per scaricare i moduli

---

<sup>68</sup> Conda-forge è un’organizzazione il cui scopo è quello di scrivere, mantenere, aggiornare, modificare e correggere moltissime “*conda recipes*”, ovvero una serie di moduli utili alla programmazione.

<sup>69</sup> Python, *PyPI – the Python Package Index*, <https://pypi.python.org/pypi>, data di consultazione: 14/02/2018.

sulla piattaforma in uso. Dovrebbe essere già disponibile se si sta utilizzando una versione successiva alla 3.4 o alla 2.7.9, altrimenti è necessario installarlo manualmente seguendo le istruzioni riportate sul sito ufficiale di Python<sup>70</sup>. In pratica, quando si installa un qualsiasi pacchetto, Pip si collega a PyPI (deve quindi essere presente una connessione attiva alla rete Internet), cerca e scarica il modulo richiesto, il tutto in modo autonomo e senza nessun intervento, o procedura particolare da seguire, da parte dell'utilizzatore.

A questo punto il modulo risulta installato sulla piattaforma e pronto all'utilizzo. Prima però di poter accedere ai suoi contenuti è necessario importarlo nel proprio progetto. A tal scopo bisogna utilizzare il costrutto `import nome_modulo`. Questa è la forma standard per accedere alle funzioni contenute nel pacchetto, che nel codice vengono richiamate applicando la sintassi `nome_modulo.nome_funzione`, però, in realtà esistono anche altre possibilità, che sono: `from nome_modulo import nome_funzione`, in tal caso non verrà importato tutto il pacchetto ma solo la funzione specificata (utile nel caso servano solo poche funzioni del modulo); e `import nome_modulo as nuovo_nome`, utile quando i nomi dei pacchetti sono ambigui o particolarmente lunghi poiché permette di richiamare il modulo con in nuovo nome fornitogli.

Nei paragrafi seguenti sono riportati i due moduli fondamentali per la costruzione dello *script* di interesse: il primo, Open Matrix riguarda le funzioni legate proprio al formato OMX; il secondo, denominato `openpyxl`, è necessario nel caso specifico in quanto permette di leggere i file Excel che contengono la base dati.

### **4.2.1 Openmatrix**

Per ottenere il pacchetto Open Matrix è necessario aprire l'ambiente *prompt* di Anaconda e digitare: `pip install openmatrix`. Al momento dell'invio, automaticamente partirà il

---

<sup>70</sup> Python, *Pip – Installation*, <https://pip.pypa.io/en/latest/installing/>, data di consultazione: 14/02/2018.

*download* del modulo. Per richiamarlo invece nel file di lavoro, è più comodo utilizzare il costrutto `import openmatrix as omx`, in tal modo, in tutte le funzioni richiamate, sarà possibile abbreviarne il nome e rendere la scrittura del programma più veloce. Si illustrano ora alcune funzioni che saranno utili alla scrittura del file.

- \* `OMXfile = omx.open_file('NomeFile.omx','w')` apre, o crea se non esiste nella stessa directory di salvataggio del file `.py`, il file OMX con il nome specificato, e collega alla variabile “OMXfile” il documento. L’attributo “w” indica la modalità di scrittura (*writing*), altrimenti è possibile anche aprire un file esistente in modalità di modifica sostituendolo con “a” (*append/edit*). Bisogna prestare attenzione a questo aspetto poiché se un file esistente viene aperto in modalità di scrittura, i dati verranno sovrascritti ai precedenti. Se invece non viene specificato alcun attributo, il file verrà aperto, di default, in modalità lettura (*reading*).
- \* `OMXfile['nome_matrice'] = matriceOD` in questo modo l’array a due dimensioni chiamato “matriceOD” viene inserito nel file OMX, aperto con la funzione precedente, con il nome inserito tra le parentesi quadre. Si ricorda che tutte le matrici incluse nello stesso file devono avere la stessa dimensione.
- \* `OMXfile['nome_matrice'].attrs.nome_attributo = valore_attributo` con questa sintassi vengono definiti gli attributi, ovvero i metadati, associati alla specifica matrice definita con la funzione precedente. I metadati possono essere delle sia stringhe che dei numeri (interi o con virgola).
- \* `OMXfile.create_mapping('nome_mappatura',vettoreMAP)` in questo modo il vettore chiamato “vettoreMAP” viene salvato nel file `.omx` come mappatura dell’indice, con il nome fornito nel costrutto.
- \* `OMXfile.close()` chiude il file OMX aperto all’inizio dello script.

### 4.2.2 *Openpyxl*

Come già anticipato, oltre all'indispensabile pacchetto OpenMatrix, si è dovuto ottenere anche un altro modulo che permettesse di importare e leggere i dati su cui operare. Nel caso delle matrici O/D dell'IMQ 2013, il pacchetto da installare è Openpyxl, poiché sono fornite attraverso file Excel. Altri tipi di dati necessariamente richiederebbero altri moduli: ad esempio, la libreria per accedere ai database di Access è denominata Pyodbc.

Per ottenere il pacchetto Openpyxl è necessario, come per il precedente, aprire l'ambiente *prompt* di Anaconda e digitare: `pip install openpyxl`. Anche in questo caso, al momento dell'invio, parte autonomamente il *download* del modulo. Invece, per richiamarlo nel file di lavoro, si usa il costrutto `import openpyxl`. Si illustrano ora alcune funzioni di questa libreria che saranno utili alla scrittura del programma di conversione.

- \* `ExcelFile = openpyxl.load_workbook('NomeFile.xls')` in questo modo il file specificato viene aperto e caricato nella variabile denominata "ExcelFile". Si fa presente che i file devono essere memorizzati nella stessa *directory* del file .py che si sta scrivendo.
- \* `nsheet = ExcelFile.sheetnames` con questa funzione è possibile "leggere" i nomi dei fogli contenuti nel file di Excel che si è caricato. Restituisce una lista in cui ciascun nome viene memorizzato come una variabile di tipo stringa.
- \* `sheet = ExcelFile['nome_foglio']` con questa istruzione è possibile caricare nella variabile "sheet" il foglio del file indicati nella dicitura.
- \* `rowmax = sheet.max_row` con questa funzione è possibile determinare quante righe del foglio "sheet" accolgono dei dati.
- \* `VAL = sheet.cell(row=n,column=m).value` con questo costrutto è possibile estrarre il valore contenuto in una cella del foglio "sheet", individuandola attraverso il numero della riga "n" e il numero (non il carattere alfabetico)

della colonna “m”.

### **4.2.3 Altri comandi utili**

Un altro modulo utile allo *script* è chiamato Numpy. È di aiuto nella gestione degli *array* multidimensionali, come matrici e vettori. Anch'esso è ottenibile e richiamabile nel lavoro allo stesso modo dei due precedenti, e cioè attraverso l'uso di `pip` e `import`. In particolare, le sue funzioni sono servite per inizializzare, con una serie di zeri, vettori e matrici da completare all'interno di cicli, utilizzando il costrutto: `nome_array = np.zeros(dim)`.

Per inizializzare un vettore di stringhe è stato utilizzato invece il comando `vett_stringa = list()`, il quale restituisce un vettore di stringhe vuoto. Con il comando `indice = nome_lista.index("valore_da_cercare")`, infine, si ricerca nella lista chiamata `nome_lista`, il valore “valore\_da\_cercare”, e ne si ottiene l'indice che ne identifica la posizione nell'*array*. Queste ultime due funzioni presentate non fanno parte di alcun pacchetto, ma sono parte della libreria standard di Python che è già inclusa.

## **4.3 REDAZIONE ED ESECUZIONE DEGLI SCRIPT PER LA GENERAZIONE DELLE MATRICI**

A questo punto, si posseggono tutti gli strumenti e tutte le nozioni utili per redigere lo *script*. Una prima accortezza da porre è che i 12 file Excel (12 file XLS sono redatti per macro-aree e 12 per zone campionarie), contenenti i dati delle matrici O/D da convertire, siano memorizzati tutti all'interno della stessa cartella. In questa, inoltre, verrà salvato anche il file di Python che produrrà il file OMX. Una seconda operazione preliminare da effettuare riguarda il formato dei file Excel: i documenti scaricati dal sito dell'AMP sono in

formato XLS<sup>71</sup> mentre il modulo Openpyxl consente di aprire i file in formato XLSX<sup>72</sup>. Quindi è stato necessario aprire singolarmente i file con Microsoft Excel e convertirli (a tal scopo esiste un'icona predefinita, dal nome “*Converti*” che si trova seguendo il percorso “*File*”-“*Informazioni*”-“*Converti*”). Se non si è in possesso del pacchetto Office, eventualmente, ci si può affidare a diversi convertitori reperibili in rete.

Di seguito sono riportati i due script redatti per le due diverse tipologie di matrici trattate: quelle per macro-aree e quelle per zone campionarie. Entrambi seguono un iter abbastanza semplice; in linea di massima il procedimento è il seguente: richiamo dei moduli; creazione del file OMX; apertura dei file Excel; costruzione della matrice O/D; salvataggio della matrice nel file OMX; associazione dei metadati; abbinamento della mappatura dell'indice; chiusura del file OMX. Ovviamente, in entrambi i casi, i passaggi dall'apertura del file di Excel all'associazione dei metadati, devono essere ripetuti per ben 12 volte (tante quante i file da aprire).

#### **4.3.1 Matrici per macro-aree**

In questo caso le matrici sono di dimensione contenuta; infatti sono solo delle 5x5. Per questo motivo il vettore “*zone*”, contenente le indicazioni della zonizzazione effettuata, è stato definito manualmente. Dunque, con qualche ciclo (si veda lo *script* seguente da riga 14 a 18) si sono generati in modo automatico i nomi dei file Excel da aprire (per maggior chiarimenti si veda l'ALLEGATO C dove sono descritti tutti i file da analizzare). Questa procedura poteva essere effettuata anche manualmente, ma dovendo gestire 12 file, è

---

<sup>71</sup> Il formato XLS è stato il formato predefinito da quando è stato creato Microsoft Excel fino al 2003. È basato sul BIFF (Binary Interchange File Format), per cui memorizza i dati direttamente in un formato binario proprietario.

<sup>72</sup> Il formato XLSX identifica invece il nuovo formato predefinito di Microsoft Excel, attivo dalla versione di Office 2007 in poi. Non è leggibile quindi dalle versioni precedenti del software. È basato sul formato open XML, il quale memorizza i dati in un file di testo.

risultato più utile e sbrigativo creare un ciclo che ne controllasse l'accesso sulla base del nome a esso associato. In tal modo è stato anche possibile generare automaticamente i metadati relativi alla matrice. Ad esempio, il file Excel "*IMQ2013\_MPrIG\_ResTOsr.xlsx*" fa riferimento alla matrice O/D redatta considerando gli spostamenti effettuati con mezzo privato (MPr), nell'intera giornata (IG), dai residenti della sola provincia di Torino (ResTO), per tutti gli scopi escluso il ritorno (sr). Mentre un file dal nome "*IMQ2013\_MPuOP\_ResPIE.xlsx*" fa riferimento alla matrice O/D redatta considerando gli spostamenti effettuati con mezzo pubblico (MPu), nell'ora di punta (OP), dai residenti dell'intero Piemonte (ResPIE), per tutti gli scopi.

Una volta aperti i file Excel, si sono estratte delle informazioni generali sui fogli presenti, per poter localizzare al suo interno i dati di interesse: come il nome dei fogli contenuti e il numero di righe compilate. A questo punto si è potuto procedere con la definizione della relativa matrice O/D, che avviene con un ciclo *for* (si veda lo *script* seguente da riga 33 a 36), nel quale si percorrono singolarmente le righe compilate del foglio Excel. Si noti (riga 33 dello *script*) che il ciclo comincia dalla 9 e termina una riga prima dell'ultima compilata. Questo perché nelle prime 8 righe del file Excel è riportata un'intestazione fissa che identifica il contenuto della matrice, come numerosità del campione, modo utilizzato, ecc., mentre l'ultima riga, anch'essa da escludere, riporta la somma di tutti gli spostamenti.

Nello specifico, per ogni riga ripercorsa nel foglio Excel, si confronta la cella contenente la zona di origine (presente nella colonna 1) e la cella contenente la zona di destinazione (presente nella colonna 3 del file) con il vettore della zonizzazione (denominato *zone*). Si estrae dunque il relativo indice all'interno di questo vettore e lo si utilizza per compilare la matrice O/D con il valore dello spostamento presente nella colonna 5 del file Excel. Fatta questa operazione per tutti gli spostamenti registrati, la matrice così ottenuta viene salvata nel file OMX e le vengono attribuiti i relativi metadati



(si veda lo *script* da riga 40 a 64); controllando come anticipato il nome del file, è possibile determinare gli attributi corretti da associare in modo automatico. Ripetute queste operazioni per i 12 documenti, il file OMX può essere chiuso.

Di seguito è riportato lo *script* in oggetto, denominato “*Macroaree.py*”.

```
01 import openpyxl
02 import numpy as np
03 import openmatrix as omx
04
05 #Creazione del file OMX (w=writing)
06 OMXfile = omx.open_file('IMQ2013_macroaree.omx','w')73
07
08 #Zonizzazione
09 zone = ['Torino','Cintura','Resto Provincia TO',
10         'Resto Piemonte','Esterno']
11
12 #Per ogni casistica popolazione/modo/periodo/scopo si determina il
13 relativo file Excel
14 for pop in ['TO','PIE']:
15     for mez in ['Pr','Pu']:
16         for per in ['IG','OP']:
17             for sco in ['sr','']:
18                 if not (sco=='sr' and per=='OP'):
19                     #Costruzione del nome del file in maniera automatica
20                     namedoc = 'IMQ2013_M'+mez+per+'_'+'Res'+pop+sco+'.xlsx'
21                     #Apertura del documento Excel
22                     excel_document = openpyxl.load_workbook(namedoc)
23                     #Recupero dei nomi dei fogli presenti nel file
24                     nsheet = excel_document.sheetnames
25                     #Caricamento del foglio di interesse (il primo)
26                     sheet = excel_document[nsheet[0]]
27                     #Determinazione del numero di righe compilate
28                     rowmax = sheet.max_row
29
30                     #Inizializzazione della matrice O/D
31                     matrOD = np.zeros((5,5))
32                     #Compilazione della matrice O/D con i dati del file
33                     for i in range(9,rowmax):
```

---

<sup>73</sup> Si ricorda che l'apertura di un file OMX con attributo “w” permette la creazione del file con il nome fornito (cioè “IMQ2013\_macroaree.omx”) nel caso in cui nella cartella di esecuzione del file Python non sia presente alcun documento con lo stesso nome. In caso contrario avviene una sovrascrittura dello stesso, con conseguente cancellazione delle informazioni precedentemente memorizzate.

```
34         matrOD[zone.index(sheet.cell(row=i, column=1).value)]
35             [zone.index(sheet.cell(row=I, column=3).value)]=
36             sheet.cell(row=i, column=5).value
37 #Scrittura della matrice sul file OMX
38 OMXfile['Res'+pop+'m'+mez+per+sco] = matrOD
39
40 #Scrittura dei metadati complementari
41 OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Indagine = 'IMQ 2013'
42 if pop=='TO':
43     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Popolazione =
44         'Provincia di Torino'
45     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.NumCampione = 32460
46 else:
47     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Popolazione =
48         'Regione Piemonte'
49     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.NumCampione = 52119
50 if mez=='Pr':
51     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Modo = 'Privato'
52 else:
53     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Modo = 'Pubblico'
54 if per=='IG':
55     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Periodo =
56         'Intera giornata'
57 else:
58     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Periodo = 'Ora di
59         punta (arrivo 07:43-08:42)'
60 if sco=='sr':
61     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Scopo = 'Tutti
62         escluso ritorno'
63 else:
64     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Scopo = 'Tutti'
65
66 #Chiusura del file creato
67 OMXfile.close()74
```

Si noti che in questo caso non è stata creata una mappatura dell'indice, poiché per i file OMX, questa deve essere definita attraverso dei numeri interi, e nel caso specifico, invece, è composta da stringhe. Si è lasciato quindi l'indice di *default* che identifica le zone nel seguente modo: 1 - Torino; 2 - Cintura; 3 - Resto Provincia TO; 4 - Resto Piemonte; 5 -

---

<sup>74</sup> È importante ricordarsi di chiudere sempre un file OMX al termine delle operazioni per garantirne l'integrità della memorizzazione.

Esterno.

### **4.3.2 Matrici per zone campionarie**

In questo secondo caso le matrici presentano dimensioni maggiori, ma comunque contenute considerandone il loro campo di applicazione, pari a 209x209 (208 zone campionarie più una zona definita “*Esterno*”). Per questo motivo il vettore della zonizzazione è stato costruito a partire da un file Excel riportante l’elenco delle 209 zone campionarie<sup>75</sup> (si veda lo *script* alla riga 09), e non manualmente come nel precedente caso. In questo caso si sono dovuti creare due vettori per la mappatura dell’indice: il primo, denominato “*zoneA*”, accoglie i codici di zona assegnati dall’Agenzia della Mobilità Piemontese alle zone campionarie, in formato alfanumerico (da qui la lettera “A” a suffisso); il secondo, chiamato “*zoneN*”, raccoglie gli stessi codici ma in formato numerico (da qui il suffisso “N”). La differenza tra questi due vettori è nelle lettere a prefisso nei codici di zona (ad esempio E036), che identificano le macro-zone di appartenenza; queste, infatti, sono state sostituite nel secondo da dei numeri (ad esempio 3036). Più precisamente:

- \* Comune di Torino – il prefisso identificativo Q è stato sostituito dal numero 1 (ad esempio: Q001 – 1001 Torino Centro);
- \* Cintura di Torino – il prefisso identificativo C è stato sostituito dal numero 2 (ad esempio: C002 – 2002 Grugliasco);
- \* Resto della provincia di Torino – il prefisso identificativo E è stato sostituito

---

<sup>75</sup> Il file denominato “*ZoneCampionarie2013.xlsx*”, ottenuto da una elaborazione propria in Excel a partire dal documento fornito dall’Agenzia della Mobilità Piemontese [11], riporta nella prima colonna il CODICE NUMERICO di identificazione della zona campionaria, nella seconda colonna il relativo CODICE ALFANUMERICO assegnato dall’Agenzia, nella terza il nome della zona, nella quarta la descrizione dei comuni coinvolti, nella quinta la provincia di appartenenza; nella sesta colonna la macro-area e nelle ultime due (la settima e la ottava) rispettivamente la longitudine e latitudine del centroide della stessa.

dal numero 3 (ad esempio: E025 – 3025 Castagnole Piemonte);

- \* Resto del Piemonte – il prefisso identificativo R è stato sostituito con il numero 4 (ad esempio: R400 – 4400 Cuneo; R500 – 4500 Asti; R600 – 4600 Alessandria; R200 – 4200 Vercelli; R700 – 4700 Biella; R300 – 4300 Novara; R800 – 4800 Verbania);
- \* Esterno – è stato associato al numero 9999.

La definizione di questo secondo vettore di zona, riportante solamente numeri interi, è stata necessaria per poter associare alla matrice una mappatura, che per dimensioni così elevate risulta indispensabile alla comprensione dei dati contenuti. Si ricorda, infatti (si veda il Paragrafo 1.1.3 Specifiche del formato (versione 0.3)), che come mappa dell'indice, non è possibile considerare un vettore di stringhe, ma solo di numeri interi.

Per il resto lo *script*, nominato “*ZoneCampionarie.py*” e riportato di seguito, si sviluppa come il precedente caso basato sulle macro-aree.

```
01 import openpyxl
02 import numpy as np
03 import openmatrix as omx
04
05 #Creazione del file OMX (w=writing)
06 OMXfile = omx.open_file('IMQ2013_zonecampionarie.omx','w')
07
08 #Apertura del documento di Excel contenente la zonizzazione
09 excel_document = openpyxl.load_workbook('ZoneCampionarie2013.xlsx')
10 #Recupero dei nomi dei fogli presenti
11 nsheet = excel_document.sheetnames
12 #Caricamento del foglio di interesse (il primo)
13 sheet = excel_document[nsheet[0]]
14 #Inizializzazione dei vettori delle zone
15 zoneA = list()
16 zoneN = np.zeros(209)
17
18 #Recupero dei codici delle zone (sia alfanumerici che interi)
19 for j in range(0,209):
20     zoneA.append(sheet.cell(row=j+2,column=2).value)
21     zoneN[j] = sheet.cell(row=j+2,column=1).value
22
23 #Per ogni casistica popolazione/modo/periodo/scopo si determina il
```

```
24 relativo file Excel
25 for pop in ['TO','PIE']:
26     for mez in ['Pr','Pu']:
27         for per in ['IG','OP']:
28             for sco in ['sr','']:
29                 if not (sco=='sr' and per=='OP'):
30                     #Costruzione del nome del file in maniera automatica
31                     namedoc = 'IMQ2013_M'+mez+per+'Res'+pop+sco+'.xlsx'
32                     #Apertura del documento Excel
33                     excel_document = openpyxl.load_workbook(namedoc)
34                     #Recupero dei nomi dei fogli presenti nel file
35                     nsheet = excel_document.sheetnames
36                     #Caricamento del foglio di interesse (il primo)
37                     sheet = excel_document[nsheet[0]]
38                     #Determinazione del numero di righe compilate
39                     rowmax = sheet.max_row
40
41                     #Inizializzazione della matrice O/D
42                     matrOD = np.zeros((209,209))
43                     #Compilazione della matrice O/D con i dati del file
44                     for i in range(9,rowmax+1):
45                         matrOD[zoneA.index(sheet.cell(row=i,column=1).value)]
46                             [zoneA.index(sheet.cell(row=i,column=3).value)] =
47                             Sheet.cell(row=i,column=5).value
48                     if not sco=='sr':
49                         sco = "t"
50
51                     #Scrittura della matrice sul file OMX
52                     OMXfile['Res'+pop+'m'+mez+per+sco] = matrOD
53
54                     #Scrittura dei metadati complementari
55                     OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Indagine = 'IMQ 2013'
56                     if pop=='TO':
57                         OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Popolazione =
58                             'Provincia di Torino'
59                         OMXfile['Res'+pop+'m'+mez+per+sco].attrs.NumCampione = 32460
60                     else:
61                         OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Popolazione =
62                             'Regione Piemonte'
63                         OMXfile['Res'+pop+'m'+mez+per+sco].attrs.NumCampione = 52119
64                     if mez=='Pr':
65                         OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Modo = 'Privato'
66                     else:
67                         OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Modo = 'Pubblico'
68                     if per=='IG':
69                         OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Periodo =
70                             'Intera giornata'
71                     else:
72                         OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Periodo = 'Ora di
73                             punta (arrivo 07:43-08:42)'
```

```

74         if sco=='sr':
75             OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Scopo = 'Tutti
76                                                         escluso ritorno'
77         else:
78             OMXfile['Res'+pop+'m'+mez+per+sco].attrs.Scopo = 'Tutti'
79
80 #Associazione della mappatura dell'indice
81 OMXfile.create_mapping('Mappatura',zoneN)
82
83 #Chiusura del file creato
84 OMXfile.close()
    
```

A titolo esemplificativo si riporta nella FIGURA 4.3.1 la schermata dell'OMX Viewer con aperto il file appena creato, anche se le matrici sono di dimensioni troppo elevate per poterne osservare interamente il contenuto.

	1001	1002	1003	1004	1005	1006	1007	1008	1009
1001	4371.03	2770.78	4282.45	1093.52	3364.05	3129.53	3987.19	2743.93	2502.08
1002	1956.85	2212.9	1518.7	492.49	402.49	912.84	1105.34	706.41	2133.85
1003	4004.11	1527.01	6372.43	1880.93	2155.05	1406.64	1113.43	295.03	1350.54
1004	1281.38	391.97	1982.48	3072.29	2320.99	1185.35	798.91	493.87	763.64
1005	3266.98	597.7	2349.62	2309.92	2689.6	1906.66	1209.06	802.78	689.78
1006	3230.23	1213.78	1409.81	890.56	2043.17	6601.51	2402.98	1624.54	1203.65
1007	3267.6	802.68	1011.18	892.9	908.85	2661.05	2824.49	2013.33	633.78
1008	2738.18	444.13	396.58	694.19	704.85	1416.74	1899.76	3355.26	498.6
1009	2849.67	1817.21	1422.62	697.39	886.12	1300.48	860.3	901.84	1658.99
1010	2010.5	1197.33	1797.16	604.58	705.27	1315.36	507.14	600.48	2503.46
1011	1396.11	99.7	1595.99	1413.9	1113.16	924.35	603.16	510.46	940.42
1012	1723.49	503.56	1218.97	1490.93	1469.13	1593.34	509.36	102.26	1924.92
1013	2172.42	91.92	1684.07	2691.75	3672.48	2069.23	510.07	782.17	493.68
1014	1936.84	391.37	795.6	1398.04	2102.21	3012.09	1038.73	405.06	610.8
1015	1002.52	202.13	196.67	132.53	896.1	1401.61	301.23	196.3	511.53
1016	1426.46	133.97	263.02	171.53	486.65	1863.71	1769.78	807.94	408.57

FIGURA 4.3.1 Apertura del file “IMQ2013\_zonecampionarie.omx” con il visualizzatore OMX Viewer.

## 4.4 VERIFICA DEL CODICE, RISULTATI ED OSSERVAZIONI

Nel caso delle matrici redatte per macro-aree (si veda il Paragrafo 4.3.1 Matrici per macro-aree), essendo di dimensione contenuta (12 matrici O/D 5x5), è possibile effettuare un

confronto diretto visivo con i corrispondenti dati Excel per confermare la bontà del codice scritto. Per accedere alle matrici OMX è stato utilizzato l'OMX Viewer, mentre per avere una lettura più immediata dei dati contenuti nei fogli di calcolo, sono state create delle Tabelle Pivot. Nelle seguenti figure e tabelle sono riportate a titolo di esempio due matrici scelte casualmente tra quelle disponibili.

La prima matrice confrontata accoglie i dati riguardanti gli spostamenti effettuati con mezzo privato (mPr), nell'intera giornata (IG), per qualunque scopo escluso il ritorno (sr) da tutti i residenti del Piemonte (ResPIE). Nella FIGURA 4.4.3 è riportata la matrice per come si presenta nell'interfaccia dell'OMX Viewer. Mentre nella TABELLA 4.4.1 è riportata la matrice ottenuta dai dati originari nel documento di Excel, utilizzando il comando "Tabella Pivot". Come si può osservare le due matrici coincidono perfettamente.

	1	2	3	4	5
1	334124.04	78882.4	17155.94	6778.95	3596.75
2	96640.19	197420.37	20773.16	3684.84	2477.8
3	51807.73	47914.2	395587.94	13045.88	6499.1
4	13065.17	4616.97	11764.38	1284404.76	40722.68
5	32.44	167.12	585.08	999.8	871.58

FIGURA 4.4.1 Matrice denominata "ResPIEmPrIGsr" contenuta nel file OMX "IMQ2013\_macroaree.omx" visualizzata attraverso l'OMX Viewer.

Di seguito è riportato un secondo esempio; in questo caso le matrici confrontate raccolgono i dati riguardanti gli spostamenti effettuati con mezzo pubblico (mPu), nell'ora di punta (OP), per qualunque scopo dai soli residenti della Provincia di Torino (ResTO). Nella FIGURA 4.4.2 è riportata la matrice per come si presenta nell'interfaccia dell'OMX Viewer,

TABELLA 4.4.1 Matrice ricavata dal file Excel “*IMQ2013\_MPrIG\_ResPIEsr.xlsx*” attraverso elaborazione propria con il comando “Tabella Pivot”.

ORIGINI	DESTINAZIONI					TOTALE
	Torino	Cintura	Provincia TO	Resto Piemonte	Esterno	
Torino	334124	78882	17156	6779	3597	440538
Cintura	96640	197420	20773	3685	2478	320996
Provincia TO	51808	47914	395588	13046	6499	514855
Resto Piemonte	13065	4617	11764	1284405	40723	13545474
Esterno	32	167	585	1000	872	2656
TOTALE	495670	329001	445867	1308914	54168	2633619

mentre nella TABELLA 4.4.2 è riportata la matrice ottenuta dai dati del documento di Excel. Come si può osservare, anche in questo secondo esempio, le due matrici coincidono perfettamente.



FIGURA 4.4.2 Matrice denominata “*ResTOmPuOP*” contenuta nel file OMX “*IMQ2013\_macroaree.omx*” visualizzata attraverso l’OMX Viewer.



TABELLA 4.4.2 Matrice ricavata dal file Excel “*IMQ2013\_MPuOP\_ResTO.xlsx*” attraverso elaborazione propria con il comando “Tabella Pivot”.

ORIGINI	DESTINAZIONI					TOTALE
	Torino	Cintura	Provincia TO	Resto Piemonte	Esterno	
Torino	54982	2892	533	398	207	59011
Cintura	12665	3814	930	132	33	17575
Provincia TO	11748	1589	13792	492	461	28083
Resto Piemonte	0	0	0	0	0	0
Esterno	0	0	0	0	0	0
TOTALE	79395	8294	15256	1023	701	104669

Nel secondo caso analizzato (si veda il Paragrafo 4.3.2 Matrici per zone campionarie), in cui la zonizzazione è effettuata per zone campionarie, la dimensione delle matrici ottenute (209x209) non ne ha permesso la presentazione in maniera completa nel presente lavoro. Anche in questo caso, però, si è proceduto a confrontare le matrici OMX con i corrispondenti dati Excel. Nello specifico, per effettuare questo controllo, è stato redatto uno *script* Python (riportato e commentato nell'ALLEGATO D) in cui le matrici O/D, sotto forma di Tabelle Pivot di Excel, sono state confrontate con le corrispondenti matrici contenute nel file OMX “*IMQ2013\_zonecampionarie.omx*”, calcolandone, per ciascuna delle 12 coppie, il RMSE<sup>76</sup>. Poiché in questo caso il RMSE viene applicato per verificare l'uguaglianza delle matrici confrontate, ci si aspetta che sia in tutti i casi pari a zero, e questo è effettivamente ciò che si è ottenuto. Si riporta di seguito, a testimonianza di quanto affermato, l'*output* prodotto a video dallo *script* di controllo.

<sup>76</sup> Il RMSE (Root Mean Square Error) è traducibile in italiano come la “radice dell'errore quadratico medio”. Rappresenta un indice statistico che permette di determinare quanto dei valori, solitamente riferiti a dei dati osservati e a dei dati stimati, si discostano tra di loro. Viene calcolato con la seguente formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{n}}$$

## - VERIFICA DEL CODICE DI SCRITTURA DEL FILE OMX PER ZONE CAMPIONARIE -

Tabella Pivot - Matrice OMX:	RMSE
MPrIGResPIEsr - ResPIEmPrIGsr	0.0
MPrIGResPIE - ResPIEmPrIGt	0.0
MPrOPResPIE - ResPIEmPrOPT	0.0
MPuIGResPIEsr - ResPIEmPuIGsr	0.0
MPuIGResPIE - ResPIEmPuIGt	0.0
MPuOPResPIE - ResPIEmPuOPT	0.0
MPrIGResTOsr - ResTOmPrIGsr	0.0
MPrIGResTO - ResTOmPrIGt	0.0
MPrOPResTO - ResTOmPrOPT	0.0
MPuIGResTOsr - ResTOmPuIGsr	0.0
MPuIGResTO - ResTOmPuIGt	0.0
MPuOPResTO - ResTOmPuOPT	0.0

L'elenco completo con la descrizione di tutte le matrici generate dagli *script* e raccolte nei due file OMX, il primo relativo alle macro-aree e il secondo relativo alle zone campionarie, è riportato nell'ALLEGATO E.

Per quanto riguarda, invece, lo spazio di memoria occupato dai nuovi file OMX, rispetto ai corrispondenti file Excel, si può dire che rappresentino un grosso vantaggio. I 24 file XLS sono raggruppati in soli 2 file OMX, quindi si ha un vantaggio anche di tipo pratico: se si vogliono osservare in generale tutti i dati, i documenti da aprire nel secondo caso sono molti meno. Inoltre, i file OMX, come anticipato, occupano moltissimo spazio in meno rispetto ai corrispondenti XLS. Questo nonostante la memoria "sprecata" dai file OMX che, per struttura, dedicano uno spazio anche se tra due zone non avviene nessun flusso (nei file Excel invece queste informazioni sono state eliminate; non sono infatti presenti coppie origine-destinazione con flusso registrato pari a 0). Di seguito, nella FIGURA 4.4.3, è riportata una sintesi della memoria occupata dai vari file nei due formati.

Si osservi come in entrambi i casi i formati OMX occupino circa il 60% di memoria in meno rispetto ai corrispondenti dati salvati con il formato dei fogli di calcolo di Excel.

Riguardo i tempi di esecuzione dei due programmi scritti, si può affermare che entrambi gli *script* sono terminati in pochi secondi, pari al massimo a una decina. La

percentuale massima di utilizzo del processore<sup>77</sup> registrata durante l'esecuzione è intorno al 20%, nel caso dello *script* per macro-aree, e intorno al 35%, nel caso dello *script* per zone campionarie.

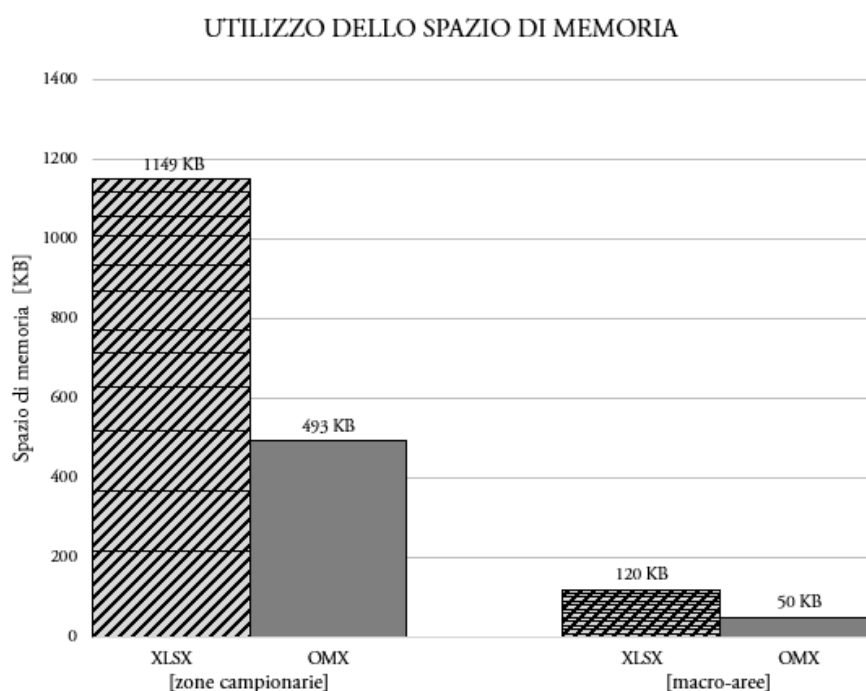


FIGURA 4.4.3 Memoria occupata dai 12 file Excel (XLSX) e dal corrispondente file OMX, sia nel caso di zonizzazione per zone campionarie che in quello per macro-aree.

<sup>77</sup> Specifiche tecniche della macchina: Notebook HP, processore Intel® Core™ i5-6200U CPU @ 2.30-2.40 GHz; RAM installata 8.00 GB; sistema operativo a 64 bit, processore basato su x64.

# 5. ELABORAZIONI SULLE MATRICI OMX

Nel precedente Capitolo 4 viene descritta la procedura che consente di creare delle matrici OMX a partire da delle rilevazioni effettuate sul territorio come quelle dell'indagine IMQ del 2013. È stata mostrata, quindi, una metodologia che permette solamente di archiviare in maniera differente i dati raccolti. In quest'ultima parte del lavoro, invece, si vuole mostrare in che modo è possibile estrarre da questi delle informazioni utili.

I seguenti paragrafi vogliono essere degli esempi, che ripercorrono, in pratica, le operazioni da effettuare sui dati memorizzati, accedendovi direttamente attraverso il già noto linguaggio di programmazione Python. Vengono quindi redatti alcuni *script* che illustrano i passaggi fondamentali per lo svolgimento di alcune operazioni che possono occorrere nella pratica professionale.

Come strumento di interazione con l'utente è stato scelto il software *free ed open source* Quantum GIS, o più brevemente **QGIS**<sup>78</sup> (versione 2.6.1-Brighton). Questa applicazione è

---

<sup>78</sup> Come suggerisce anche il nome, QGIS è un software di tipo GIS (Geographic Information System) multiplatforma: è sviluppato per i sistemi MacOS, Linux, Unix e Microsoft Windows. È utilizzato nelle analisi di tipo territoriale; permette di creare, modificare, visualizzare e analizzare informazioni di tipo geospaziale. Con questo software è possibile, infatti, creare mappe personalizzate a rappresentazione dei risultati ottenuti a seguito delle elaborazioni sui dati di trasporto. Supporta numerosissimi formati vettoriali e raster, ma soprattutto possiede un'interfaccia di programmazione in Python (chiamata *Python Console*) che permette di ampliare, a propria necessità, le funzionalità base del programma. Nato come progetto ufficiale della Open Source Geospatial Foundation (OSGeo), è supportato da un *team* di volontari che periodicamente ne ampliano i contenuti [42].

stata utilizzata per l'acquisizione dei dati di *input* e per mostrare in forma grafica alcuni risultati delle elaborazioni proposte. I risultati che hanno fornito delle nuove matrici di *output* sono stati osservati con il già noto visualizzatore **OMX Viewer**. Con questo strumento è possibile aprire i file OMX ed esplorarne il contenuto, sia per quanto riguarda le matrici, che la mappatura dell'indice, che i metadati associati. Permette anche di compiere delle operazioni, seppur limitate, sui file aperti, come: creare dei nuovi gruppi di matrici; rinominarle; tagliarle, copiarle e incollarle in altri file; ed eliminarle. Qualche possibilità di estrazione di informazioni più elaborate forse è possibile attraverso il comando "*Query...*" che si trova nel menu denominato "*Dataset*", ma in tutti i file creati nel presente lavoro, questa opzione risultava sempre disabilitata.

Infine, le operazioni effettuate negli esempi applicativi, sono state eseguite anche con il programma di calcolo numerico MATLAB, al fine di avere un termine di paragone in merito all'efficienza computazionale. Mentre, le matrici prodotte, sono state memorizzate attraverso il relativo formato proprietario MAT (si veda il Paragrafo 2.6 Formati proprietari) per confrontarne la memoria occupata dai file.

Si fa presente che tutte le elaborazioni che seguono sono state effettuate sul file OMX contenente le matrici per zone campionarie, dal nome "*IMQ2013\_zonecampionarie.omx*" (si veda l'ALLEGATO E per maggiori dettagli).

In particolare, gli esempi proposti riguardano:

1. La determinazione del flusso da una zona campionaria specifica ad un'altra, e viceversa, sulla base delle condizioni specificate (modo, periodo e scopo);
2. Il calcolo del flusso totale prodotto e attratto da una zona scelta, sempre sulla base delle condizioni specificate, con la possibilità di ottenere delle mappe personalizzate per mostrare i risultati ottenuti anche in forma grafica;
3. L'estrazione di parte delle matrici individuando delle zone su cui focalizzare

l'attenzione e lo studio degli spostamenti;

4. L'aggregazione dei flussi contenuti nelle matrici sulla base di un attributo legato alle zone campionarie;
5. L'ottenimento e la visualizzazione di informazioni di sintesi come, ad esempio, la determinazione della percentuale di scelta del mezzo pubblico sugli spostamenti totali.

## 5.1 ESTRAZIONE DEL FLUSSO TRA DUE ZONE DEFINITE

Il seguente esempio mostra la procedura da mettere in pratica nel caso si voglia conoscere in particolare il flusso tra due determinate zone. Nello specifico, si è voluto indagare sugli spostamenti effettuati tra due comuni della Cintura di Torino: il primo scelto è Chieri, il secondo è Santena. Grazie alla mappa delle zone campionarie creata in QGIS (si veda il Paragrafo 3.2.1 Campionamento, e in particolare la FIGURA 3.2.1 e la FIGURA 3.2.2), è possibile individuare geograficamente i due comuni e ricavarne i codici identificativi, come mostrato nella FIGURA 5.1.1.

Lo *script* redatto allo scopo di estrarre questo tipo di informazioni dalla matrice, è stato nominato “*EstrazioneSingoloValore.py*”, ed è riportato di seguito.

```

01 import openmatrix as omx
02 import openpyxl
03 import numpy as np
04
05 #Apertura del file OMX
06 OMXfile = omx.open_file('IMQ2013_zonecampionarie.omx','r')
07 print("File OMX di riferimento: ",OMXfile.filename,"\n")
08
09 #Caricamento delle informazioni sulla mappatura (conversione cod.
10 alfanumerico - cod. numerico)
11 excel_document = openpyxl.load_workbook('ZoneCampionarie2013.xlsx')
12 nsheet = excel_document.sheetnames

```

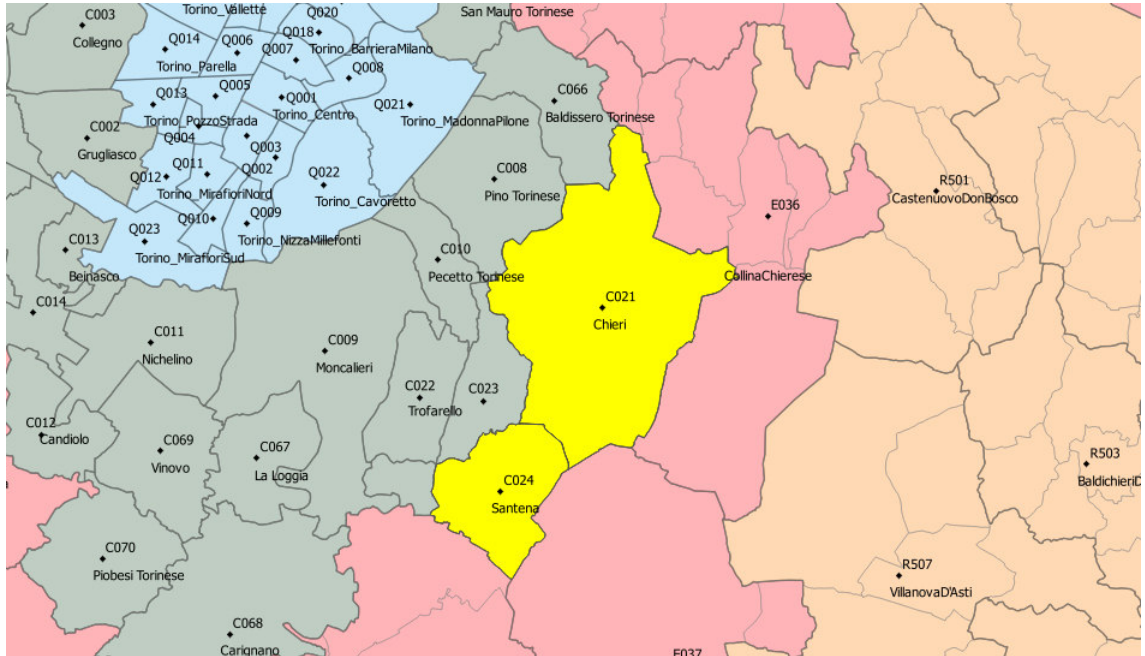


FIGURA 5.1.1 Individuazione del codice delle zone campionarie di interesse attraverso l'uso di una mappa elaborata in QGIS. I diversi colori rappresentano le diverse macro-aree di appartenenza. Le zone evidenziate di giallo sono quelle selezionate per lo sviluppo dell'esempio.

```

13 sheet = excel_document[nsheet[0]]
14 zonaA = list()
15 zoneN = np.zeros(209)
16 for j in range(0,209):
17     zonaA.append(sheet.cell(row=j+2,column=2).value)
18     zoneN[j] = sheet.cell(row=j+2,column=1).value
19
20 #Definizione della funzione di ricerca del valore nella matrice
21 def estrazioneValore(matrice,Z1,Z2):
22     S1to2 = OMXfile[matrice][zonaA.index(Z1)][zonaA.index(Z2)]
23     S2to1 = OMXfile[matrice][zonaA.index(Z2)][zonaA.index(Z1)]
24     return (S1to2,S2to1)
25
26 #Ottenimento dei dati di ingresso
27 zona1 = input("Inserire codice prima zona:\t")
28 zona2 = input("Inserire codice seconda zona:\t")
29 modo = input("Inserire il modo (Pr-privato, Pu-pubblico, T-totali): ")
30 per = input("Periodo (IG-intera giornata, OP-ora di punta): ")
31 if per == "IG":
32     scopo = input("Scopo (t-tutti, sr-tutti escluso ritorno): ")
33 else:
34     scopo = 't'
35

```

```

36 #Ricerca delle informazioni e stampa a video dei risultati
37 if modo == "T":
38     SPOST1to2 = 0
39     SPOST2to1 = 0
40     for md in ["Pr","Pu"]:
41         (V1to2,V2to1) = estrazioneValore("ResPIEm"+md+per+scopo,zona1,zona2)
42         SPOST1to2 = SPOST1to2+V1to2
43         SPOST2to1 = SPOST2to1+V2to1
44     print("\n- SPOSTAMENTI REGISTRATI TRA LE DUE ZONE -")
45     print(zona1,"-->",zona2," = ",SPOST1to2)
46     print(zona2,"-->",zona1," = ",SPOST2to1)
47     print("Matrici di riferimento: ", "ResPIEmPu"+per+scopo, "/
48                                             ResPIEmPr"+per+scopo)
49 else:
50     (SPOST1to2,SPOST2to1) =
51         estrazioneValore("ResPIEm"+modo+per+scopo,zona1,zona2)
52     print("\n-SPOSTAMENTI REGISTRATI TRA LE DUE ZONE-")
53     print(zona1,"-->",zona2," = ",SPOST1to2)
54     print(zona2,"-->",zona1," = ",SPOST2to1)
55     print("Matrice di riferimento: ", "ResPIEm"+modo+per+scopo)
56
57 #Chiusura del file aperto
58 OMXfile.close()

```

Come di consueto la prima parte dello *script* è dedicata al caricamento delle librerie. Il file OMX di interesse viene dunque aperto con la funzione già illustrata nel precedente Paragrafo 4.2.1 Openmatrix e la modalità utilizzata è quella di default in sola lettura. Successivamente viene riaperto e riutilizzato il documento di Excel adoperato per l'ottenimento delle informazioni sulla conversione dei codici delle zone (si veda la Nota 75 a Pagina 99). Anche se attraverso il comando dalla sintassi `NomeVariabile = File.mapping('NomeMappa')` è possibile memorizzare la mappatura dell'indice presente nel file OMX aperto, nello specifico caso, non è risultato sufficiente in quanto si necessitava della conversione codice alfanumerico – codice numerico, non presente nel file contenente le matrici. Successivamente viene definita una funzione, dal nome “estrazioneValore” che ricevuto in *input* il nome della matrice, e il codice delle due zone, ne restituisce gli spostamenti nei due sensi (dalla prima alla seconda e viceversa). Questa definizione è utile per evitare il ripetersi di operazioni che dovranno essere effettuate più volte nello *script*.



Si passa quindi all'ottenimento delle informazioni utili all'individuazione del flusso ricercato. Il programma stampa a video le richieste (si veda lo *script* alle righe 26-34), che comprendono: il codice alfanumerico delle due zone interessate; il modo utilizzato (pubblico, privato o qualunque); il periodo (intera giornata oppure ora di punta); e lo scopo (tutti o escluso il ritorno). I dati introdotti permettono quindi di determinare univocamente il nome della matrice. A questo punto è possibile passare queste informazioni alla funzione "estrazioneValore" ed ottenere i risultati ricercati. Come ultima operazione si ricorda che è sempre necessario chiudere i file OMX aperti, per evitare malfunzionamenti degli stessi. Di seguito è riportato un esempio di *output* ottenuto a video.

```
File OMX di riferimento:  IMQ2013_zonecampionarie.omx
```

```
Inserire codice prima zona:      C021
Inserire codice seconda zona:    C024
Inserire il modo (Pr-privato, Pu-pubblico, T-totali): Pr
Periodo (IG-intera giornata, OP-ora di punta): OP
```

```
- SPOSTAMENTI REGISTRATI TRA LE DUE ZONE -
C021 --> C024 = 64.93
C024 --> C021 = 33.97
Matrice di riferimento:  ResPIEmPrOPT
```

Dunque, per far funzionare correttamente il presente programma è necessario eseguire i seguenti passaggi: aprire la mappa delle zone campionarie in QGIS; individuare le zone di interesse (come riportato nella FIGURA 5.1.1) e ricavarne i corrispondenti codici alfanumerici (questi primi due passaggi possono essere evitati se i codici sono già noti); richiamare lo *script* di Python "*EstrazioneSingoloValore.py*" in un *prompt* di comandi o lanciare l'esecuzione nell'ambiente di Wing; inserire i dati richiesti (codici delle due zone, modo, periodo ed eventualmente scopo). A seguito della procedura, verranno restituiti, a video, i due valori di flusso scambiati tra le due zone e l'indicazione con il nome della matrice da cui sono state estratte le informazioni.

## 5.2 DETERMINAZIONE DEL FLUSSO TOTALE PRODOTTO/ATTRATTO DA UNA ZONA

Il secondo *script* riportato qui di seguito ripercorre, invece, la procedura da eseguire nel caso si voglia estrarre da una matrice O/D, in formato OMX, il flusso totale prodotto e attratto da una specifica zona. Nell'esempio, si sono voluti ricavare gli spostamenti in entrata e in uscita dal quartiere torinese denominato "Lingotto". Anche in questo caso, si è fatto ricorso alla mappa in QGIS per individuare geograficamente la zona e ottenere il codice della zona campionaria, come rappresentato in FIGURA 5.2.1.

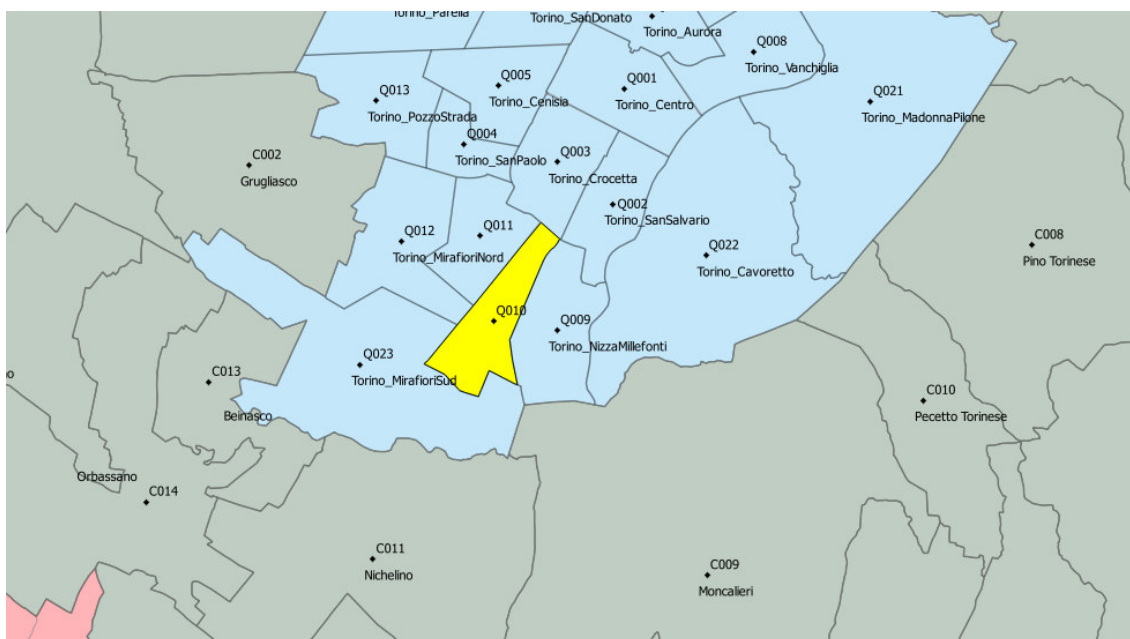


FIGURA 5.2.1 Individuazione del codice della zona campionaria di interesse attraverso l'uso di una mappa elaborata in QGIS. I diversi colori rappresentano le diverse macro-aree di appartenenza. La zona evidenziata di giallo è quella selezionata per lo sviluppo dell'esempio.

Lo *script* redatto allo scopo di estrarre questo tipo di informazioni dalla matrice, è stato nominato "*TotaleProdottoAttratto.py*", ed è riportato di seguito.

## ELABORAZIONI SULLE MATRICI OMX Determinazione del flusso totale prodotto/attratto da una zona

```
01 import openmatrix as omx
02 import openpyxl
03 import numpy as np
04
05 #Apertura del file OMX
06 OMXfile = omx.open_file('IMQ2013_zonecampionarie.omx')
07 print("File OMX di riferimento: ",OMXfile.filename,"\n")
08
09 #Caricamento delle informazioni sulla mappatura (per conversione cod.
10 alfanumerico - cod. numerico)
11 excel_document = openpyxl.load_workbook('ZoneCampionarie2013.xlsx')
12 nsheet = excel_document.sheetnames
13 sheet = excel_document[nsheet[0]]
14 zoneA = list()
15 zoneN = np.zeros(209)
16 for j in range(0,209):
17     zoneA.append(sheet.cell(row=j+2,column=2).value)
18     zoneN[j] = sheet.cell(row=j+2,column=1).value
19
20 #Ottemimento dei dati di ingresso
21 zona = input("Inserire codice zona: ")
22 modo = input("Inserire il modo (Pr-privato, Pu-pubblico, T-totali): ")
23 per = input("Periodo (IG-intera giornata, OP-ora di punta): ")
24 if per == "IG":
25     scopo = input("Scopo (t-tutti, sr-tutti escluso ritorno): ")
26 else:
27     scopo = 't'
28
29 #Apertura del file di scrittura dei risultati
30 filePROD = open("SpostProd_"+zona+".txt","w")
31 filePROD.write('COD_ZONA;SPOSTAMENTI;coord_X;coord_Y\n')
32 fileATTR = open("SpostAttr_"+zona+".txt","w")
33 fileATTR.write('COD_ZONA;SPOSTAMENTI;coord_X;coord_Y\n')
34
35 #Determinazione del flusso prodotto/attratto
36 FProd = 0
37 FAttr = 0
38 if modo == "T":
39     for md in ['Pu','Pr']:
40         for i in range(0,209):
41             FProd = FProd +
42                 OMXfile['ResPIEm'+md+per+scopo][zoneA.index(zona)][i]
43             FAttr = FAttr +
44                 OMXfile['ResPIEm'+md+per+scopo][i][zoneA.index(zona)]
45 #Stampa a video dei risultati
46 print("\nSPOSTAMENTI TOTALI IN USCITA: ",FProd)
47 print("SPOSTAMENTI TOTALI IN ENTRATA: ",FAttr)
48 print("Matrici di riferimento: ","ResPIEmPu"+per+scopo,"/
49                                     ResPIEmPr"+per+scopo)
50 #Stampa su file dei risultati
```

## ELABORAZIONI SULLE MATRICI OMX Determinazione del flusso totale prodotto/attratto da una zona

```
51     for i in range(0,208):
52         if not i == zoneA.index(zona):
53             cod = sheet.cell(row=i+2,column=2).value
54             P = str(OMXfile['ResPIEmPu'+per+scopo][zoneA.index(zona)][i]+
55                     OMXfile['ResPIEmPr'+per+scopo][zoneA.index(zona)][i])
56             A = str(OMXfile['ResPIEmPu'+per+scopo][i][zoneA.index(zona)]+
57                     OMXfile['ResPIEmPr'+per+scopo][i][zoneA.index(zona)])
58             cX = str(sheet.cell(row=i+2,column=7).value)
59             cY = str(sheet.cell(row=i+2,column=8).value)
60             filePROD.write((cod+';'+P+';'+cX+';'+cY)); filePROD.write('\n')
61             fileATTR.write((cod+';'+A+';'+cX+';'+cY)); fileATTR.write('\n')
62     else:
63         for i in range(0,209):
64             FProd = FProd +
65                     OMXfile['ResPIEm'+modo+per+scopo][zoneA.index(zona)][i]
66             FAttr = FAttr +
67                     OMXfile['ResPIEm'+modo+per+scopo][i][zoneA.index(zona)]
68     #Stampa a video dei risultati
69     print("\nSPOSTAMENTI TOTALI IN USCITA: \t",FProd)
70     print("SPOSTAMENTI TOTALI IN ENTRATA: \t",FAttr)
71     print("Matrici di riferimento: ", "ResPIEm"+modo+per+scopo)
72     #Stampa su file dei risultati
73     for i in range(0,208):
74         if not i == zoneA.index(zona):
75             cod = sheet.cell(row=i+2,column=2).value
76             P = str(OMXfile['ResPIEm'+modo+per+scopo][zoneA.index(zona)][i])
77             A = str(OMXfile['ResPIEm'+modo+per+scopo][i][zoneA.index(zona)])
78             cX = str(sheet.cell(row=i+2,column=7).value)
79             cY = str(sheet.cell(row=i+2,column=8).value)
80             filePROD.write((cod+';'+P+';'+cX+';'+cY)); filePROD.write('\n')
81             fileATTR.write((cod+';'+A+';'+cX+';'+cY)); fileATTR.write('\n')
82
83     print("\nGenerati i file: ",filePROD.name,"/",fileATTR.name)
84
85     #Chiusura dei file aperti
86     filePROD.close()
87     fileATTR.close()
88     OMXfile.close()
```

Anche in questo caso le operazioni preliminari da effettuare sono simili a quelle affrontate nello *script* precedente: caricamento delle librerie; apertura del file OMX; caricamento delle mappature dal file Excel; e ottenimento dei dati di ingresso.

In questo caso, oltre alla stampa a video dei dati di sintesi ottenuti, si è scelto di memorizzare su due file di testo TXT gli spostamenti attratti e prodotti dalla zona

**ELABORAZIONI SULLE MATRICI OMX** Determinazione del flusso totale prodotto/attratto da una zona

interessata, da e verso, le restanti zone campionarie. I due totali di flusso stampati a video sono stati ottenuti con dei semplici cicli *for* sulle colonne e sulle righe della matrice O/D (si veda lo *script* alle righe 40-44 e 63-67). Mentre i flussi da e verso le altre zone sono stati semplicemente estratti dalla matrice e salvati ordinatamente in un file di testo così strutturato: codice alfanumerico della zona campionaria; spostamento prodotto o attratto; longitudine del centroide della zona; latitudine del centroide della zona. Queste ultime due informazioni sono contenute nel già noto file Excel prodotto per la conversione codice alfanumerico – codice numerico dal nome “*ZoneCampionarie2013.xlsx*” (si veda la Nota 75 a Pagina 99), e saranno utili per la rappresentazione dei risultati nell’ambiente di QGIS.

Di seguito è riportato l’*output* ottenuto a video.

File OMX di riferimento: IMQ2013\_zonecampionarie.omx

Inserire codice zona: Q010

Inserire il modo (Pr-privato, Pu-pubblico, T-totali): Pr

Periodo (IG-intera giornata, OP-ora di punta): OP

SPOSTAMENTI TOTALI IN USCITA: 4787.910000000001

SPOSTAMENTI TOTALI IN ENTRATA: 6196.400000000002

Matrice di riferimento: ResPIEmPrOPT

Generati i file: SpostProd\_Q010.txt / SpostAttr\_Q010.txt

Di seguito, a titolo esemplificativo, è riportato un estratto del file di testo prodotto, dal nome “*SpostProd\_Q010.txt*”.

```
COD_ZONA;SPOSTAMENTI;coord_X;coord_Y
Q001;301.24;396178.49496;4991495.20846
Q002;100.78;395977.85622;4989356.8478
Q003;197.55;394944.20973;4990130.57686
Q004;0.0;393210.20324;4990461.3596
Q005;98.78;393840.65741;4991550.94728
Q006;0.0;394598.76099;4993107.50811
Q007;0.0;396701.59293;4992847.56862
Q008;0.0;398583.68409;4992179.17396
[...]
```

## ELABORAZIONI SULLE MATRICI OMX Determinazione del flusso totale prodotto/attratto da una zona

Per ottenere questi risultati è necessario, quindi, eseguire i seguenti passaggi: aprire la mappa delle zone campionarie in QGIS; individuare la zona di interesse (come riportato nella FIGURA 5.2.1) e ricavarne il corrispondente codice alfanumerico (questi primi due passaggi possono essere evitati se il codice è già noto); richiamare lo *script* di Python “*TotaleProdottoAttratto.py*” in un *prompt* di comandi o lanciare l’esecuzione nell’ambiente di Wing; inserire i dati richiesti (codice della zona, modo, periodo ed eventualmente scopo). A seguito della procedura, verranno restituiti, a video, i due valori di flusso totali in uscita e in entrata dalla zona e l’indicazione con il nome della matrice da cui sono state estratte le informazioni. Inoltre, sono riportati i nomi dei due file di testo creati, i quali sono memorizzati nella stessa cartella dove è alloggiato il file Python.

È possibile aprire questi due file TXT in QGIS per creare una mappa personalizzata. L’operazione è effettuabile attraverso il comando “*Aggiungi layer testo delimitato...*”, che si trova sotto il menu a tendina “*Layer*” - “*Aggiungi vettore*”, impostando come separatore di campi il carattere “;” e come coordinata x e y le ultime due colonne del file. Ripetendo l’operazione per entrambi i file si ottengono due nuovi vettori di “tipo punto”, posizionati nei centroidi delle zone, che accolgono in una colonna della tabella attributi gli spostamenti registrati tra la zona specifica e quella scelta originariamente (Q010 - Lingotto). A questo punto è possibile personalizzare la rappresentazione a proprio piacimento; nell’esempio si è scelto di assegnare il colore VERDE agli spostamenti prodotti e il ROSSO a quelli attratti, e di attribuire una dimensione dell’indicatore proporzionale al flusso registrato. La FIGURA 5.2.2, la FIGURA 5.2.3 e la FIGURA 5.2.4 mostrano questa rappresentazione.

Come si può osservare, è possibile ottenere subito un’idea generale dell’andamento del fenomeno osservato. I flussi in entrata ed in uscita dalle zone limitrofe sono più o meno equamente distribuiti (FIGURA 5.2.2 e FIGURA 5.2.3); si registrano, invece, numerosi spostamenti in entrata nella zona di Lingotto da zone più distanti (si veda la FIGURA 5.2.4 e nello specifico le zone E047 – Alta Val di Susa e R422 – Vicoforte). Si ricorda che questa

**ELABORAZIONI SULLE MATRICI OMX** Determinazione del flusso totale prodotto/attratto da una zona

La rappresentazione, secondo i dati immessi al momento dell'esecuzione dello *script*, riporta gli spostamenti effettuati con mezzo privato nell'ora di punta (arrivi tra le 07:43 e le 08:42).

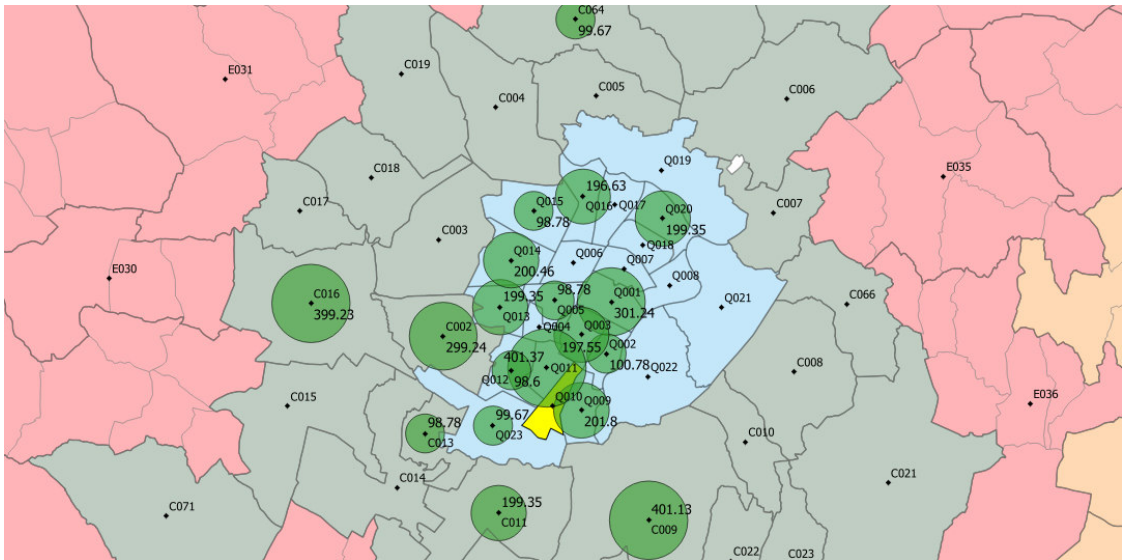


FIGURA 5.2.2 Rappresentazione delle informazioni estratte con lo *script* “*TotaleProdottoAttratto.py*” dal file OMX “*IMQ2013\_zonecampionarie.omx*”. In verde gli spostamenti prodotti dalla zona Q010. Visuale completa di tutti gli spostamenti registrati in uscita. I numeri riportati indicano l’entità dello spostamento.

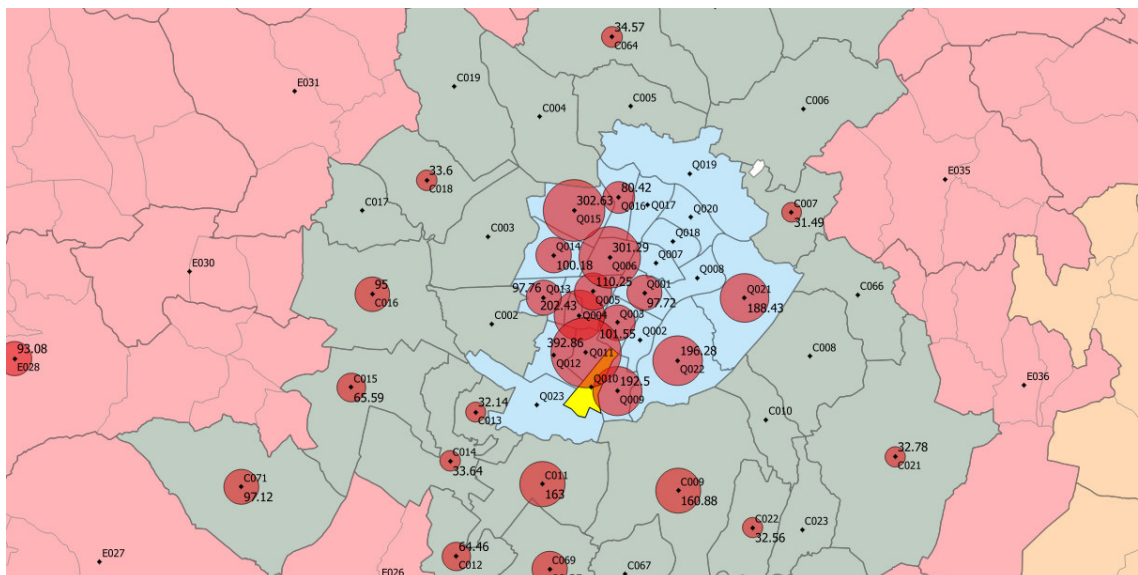


FIGURA 5.2.3 Rappresentazione delle informazioni estratte con lo *script* “*TotaleProdottoAttratto.py*” dal file OMX “*IMQ2013\_zonecampionarie.omx*”. In rosso gli spostamenti attratti dalla zona Q010. Dettaglio delle zone limitrofe. I numeri riportati indicano l’entità dello spostamento.

ELABORAZIONI SULLE MATRICI OMX Determinazione del flusso totale prodotto/attratto da una zona

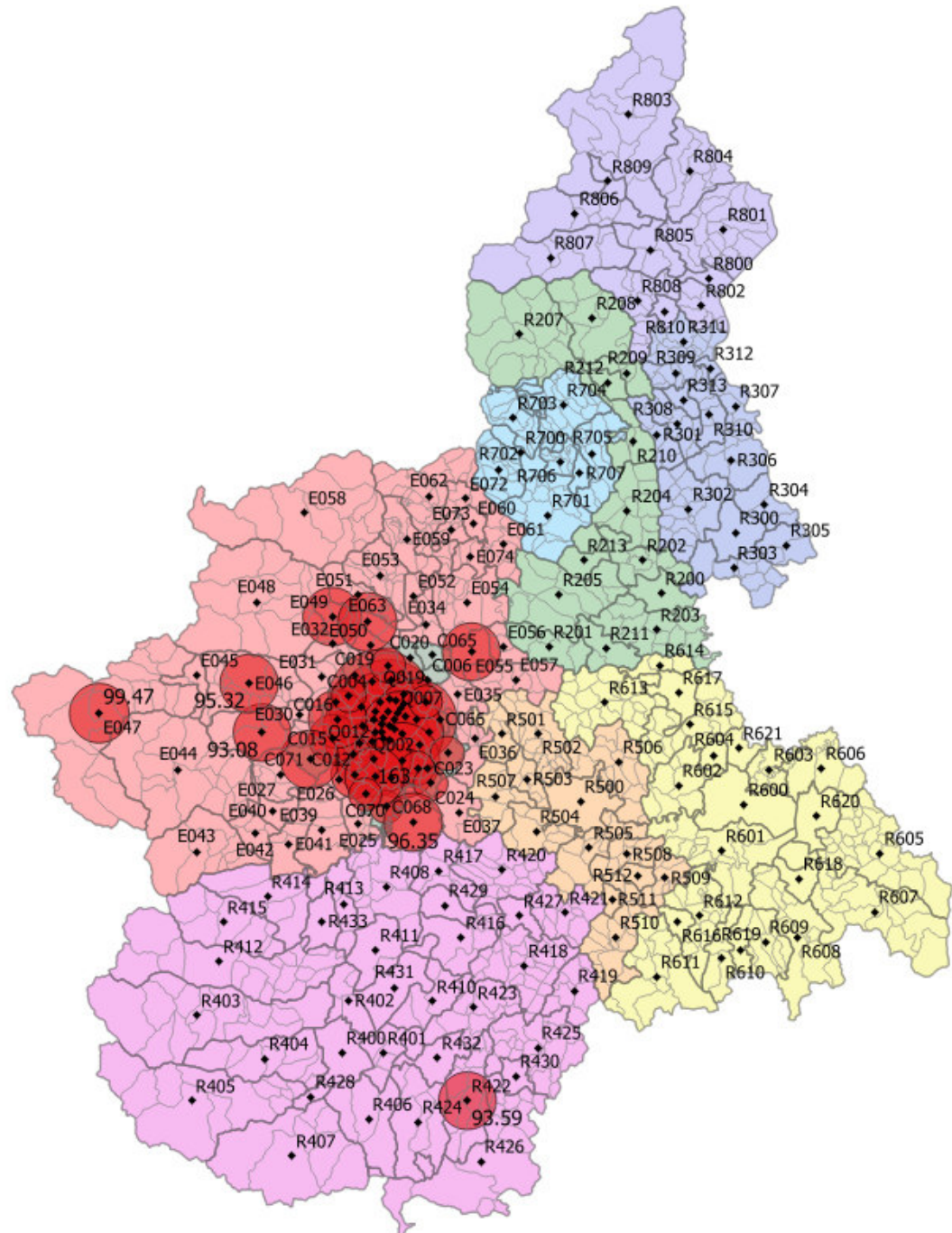


FIGURA 5.2.4 Rappresentazione delle informazioni estratte con lo script "TotaleProdottoAttratto.py" dal file OMX "IMQ2013\_zonecampionarie.omx". In rosso gli spostamenti attratti dalla zona Q010. Visuale generale del fenomeno. I numeri riportati indicano l'entità dello spostamento.



### 5.3 ESTRAZIONE DI PARTE DI MATRICE

Questo *script* permette l'estrazione delle informazioni contenute nel file "IMQ2013\_zonecampionarie.omx" sulla base della localizzazione delle zone campionarie. Si pensi ad esempio di voler ottenere delle matrici, come quelle redatte per l'indagine IMQ del 2013, ma anziché per l'intero territorio piemontese, focalizzate sugli spostamenti delle zone della Val di Susa verso il centro del capoluogo regionale. Ricorrendo alla mappa in QGIS è possibile selezionare le zone campionarie interessate ed esportarne gli attributi. La FIGURA 5.3.1 mostra questo specifico esempio.

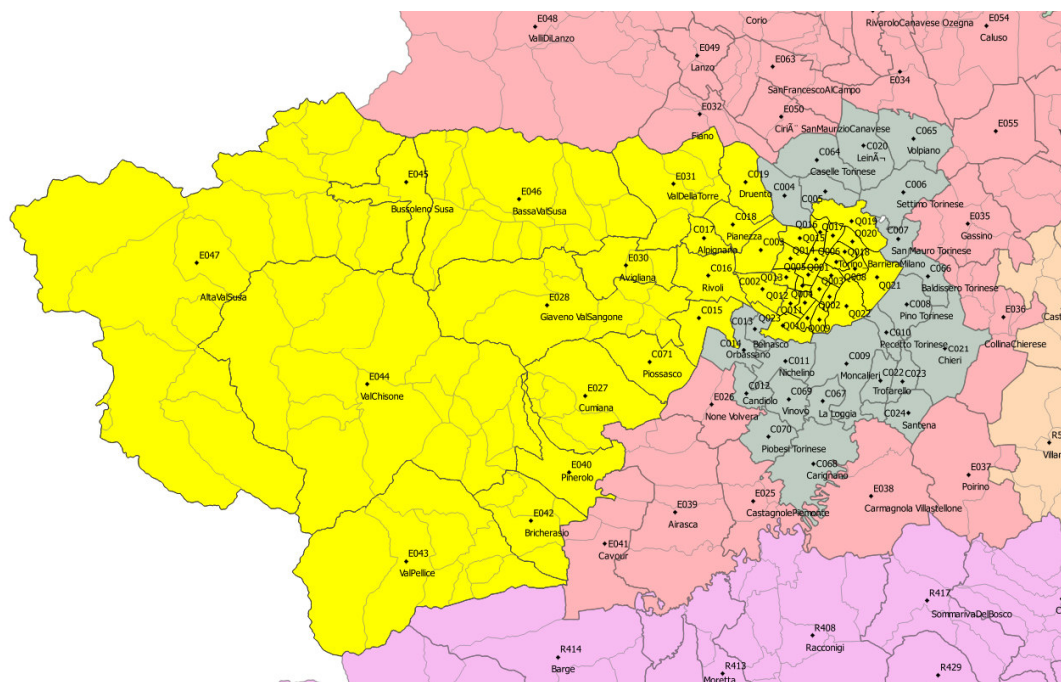


FIGURA 5.3.1 Individuazione delle zone campionarie da estrarre attraverso l'uso di una mappa elaborata in QGIS. I diversi colori rappresentano le diverse macro-aree di appartenenza. La zona evidenziata di giallo è quella selezionata per lo sviluppo dell'esempio.

L'operazione di esportazione è effettuabile, a seguito della selezione delle zone, attraverso il comando QGIS "Salva con nome...". Le caratteristiche del salvataggio devono essere le

seguenti: si salvano solo le geometrie selezionate; si utilizza il formato di testo delimitato CSV<sup>79</sup>; e si salva il nuovo file con il nome “ESTRAZIONE.csv”.

Lo *script* redatto allo scopo di estrarre questo tipo di informazioni dal file OMX, è stato nominato “*EstrazioneParteMatrice.py*”, ed è riportato di seguito.

```

01 import openmatrix as omx
02 import openpyxl
03 import numpy as np
04 import csv
05
06 #Apertura del file OMX di input
07 OMXfileI = omx.open_file('IMQ2013_zonecampionarie.omx')
08 print("File OMX di riferimento: ",OMXfileI.filename)
09 #Apertura del file OMX di output
10 OMXfileO = omx.open_file('IMQ2013_estrazione.omx','w')
11
12 #Caricamento delle informazioni sulla mappatura (per conversione cod.
13 alfanumerico - cod. numerico)
14 excel_document = openpyxl.load_workbook('ZoneCampionarie2013.xlsx')
15 nsheet = excel_document.sheetnames
16 sheet = excel_document[nsheet[0]]
17 zoneA = list()
18 zoneN = np.zeros(209)
19 for j in range(0,209):
20     zoneA.append(sheet.cell(row=j+2,column=2).value)
21     zoneN[j] = sheet.cell(row=j+2,column=1).value
22
23 #Apertura del file contenente il nuovo elenco delle zone
24 INfile = open('ESTRAZIONE.csv','r')
25 reader = csv.reader(INfile, delimiter=',')
26 NEWzone = list()
27 i=0
28 for row in reader:
29     NEWzone.append(row[0])
30     i=i+1
31 del NEWzone[0]
32 dim = len(NEWzone)
33
34 #Estrazione delle matrici
35 NEWmatrOD = np.zeros((dim,dim))

```

---

<sup>79</sup> Il formato CSV (Comma-Separated-Values) è utilizzato soprattutto come file di interscambio di tabelle di dati. Ogni riga di una tabella viene riportata in una riga di testo, e i valori nelle varie celle vengono divisi tra di loro da uno specifico carattere separatore.

```

36 for pop in ['TO','PIE']:
37     for mez in ['Pr','Pu']:
38         for per in ['IG','OP']:
39             for sco in ['sr','t']:
40                 if not (sco=='sr' and per=='OP'):
41                     for i in range(0,dim):
42                         for j in range(0,dim):
43                             NEWmatrOD[i][j] = OMXfileI['Res'+pop+'m'+mez+per+sco]
44                                 [zoneA.index(NEWzone[i])][zoneA.index(NEWzone[j])]
45 OMXfileO['Res'+pop+'m'+mez+per+sco] = NEWmatrOD
46 #Scrittura dei metadati complementari
47 OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Indagine = 'IMQ 2013'
48 if pop=='TO':
49     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Popolazione =
50         'Provincia di Torino'
51     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.NumCampione = 32460
52 else:
53     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Popolazione =
54         'Regione Piemonte'
55     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.NumCampione = 52119
56 if mez=='Pr':
57     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Modo = 'Privato'
58 else:
59     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Modo = 'Pubblico'
60 if per=='IG':
61     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Periodo =
62         'Intera giornata'
63 else:
64     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Periodo = 'Ora di
65         punta (arrivo 07:43-08:42)'
66 if sco=='sr':
67     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Scopo = 'Tutti
68         escluso ritorno'
69 else:
70     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Scopo = 'Tutti'
71
72 #Associazione della mappatura dell'indice
73 for i in range(0,dim):
74     NEWzone[i] = zoneN[zoneA.index(NEWzone[i])]
75 OMXfileO.create_mapping('Mappatura',NEWzone)
76 print("FILE OMX CREATO: ",OMXfileO.filename)
77
78 #Chiusura dei file aperti
79 OMXfileI.close()
80 OMXfileO.close()
81 INfile.close()

```

La prima operazione da far compiere allo *script* è sempre il caricamento delle librerie. In

questo caso è stato necessario utilizzare anche il modulo denominato “csv”, utile per la gestione degli omonimi file. I file OMX da aprire qui sono due: il primo è quello da cui ricavare le matrici di *input*; il secondo è quello che accoglierà le nuove matrici estratte dalle prime. Successivamente, come di consueto, viene caricato il file Excel per la definizione dei due vettori delle mappe (si veda lo *script* alle righe 12-21).

A questo punto si dispongono di tutti gli strumenti utili all'esecuzione delle operazioni richieste. Il file “*ESTRAZIONE.csv*”, creato nell'ambiente di QGIS in precedenza, viene aperto, e, attraverso un breve ciclo *for* (si veda lo *script* alle righe 28-31), i dati contenuti all'inizio di ogni riga, ovvero i codici delle zone campionarie selezionate, vengono memorizzati in un vettore denominato “*NEWzone*”. Il programma, quindi, per ogni matrice memorizzata nel file OMX di *input*, ricerca i valori di spostamento da inserire nelle nuove matrici ridotte, eseguendo una ricerca basata sugli indici (si veda lo *script* alle righe 43-44). Nello specifico, per ogni cella della nuova matrice, si determina l'indice relativo a quella zona campionaria, ricercandolo nel vettore “*zoneA*” (che ha lo stesso ordine delle matrici O/D di *input*). Si utilizza quindi la posizione ricavata per accedere direttamente alla cella interessata e memorizzarne il contenuto. Una volta estratta tutta la matrice, viene salvata nel nuovo file OMX e le vengono attribuiti i relativi metadati, come già visto nel Paragrafo 4.3.2 Matrici per zone campionarie. L'ultima operazione da compiere, prima della chiusura di tutti i file utilizzati nello *script*, è l'attribuzione della mappatura dell'indice al nuovo file. Non è possibile semplicemente utilizzare il vettore “*NEWzone*” poiché questo contiene i codici delle zone nella loro versione alfanumerica e, come ormai noto, la mappatura OMX non lo consente. A tal scopo sono stati ricercati nel vettore “*zoneN*” i corrispondenti codici numerici.

L'*output* fornito a video dal programma è il seguente.

```
File OMX di riferimento: IMQ2013_zonecampionarie.omx  
FILE OMX CREATO: IMQ2013_estrazione.omx
```

Ricapitolando, per far eseguire correttamente lo *script* al calcolatore, si devono seguire i seguenti passaggi: aprire la mappa delle zone campionarie in QGIS; selezionare le zone di interesse (come riportato nella FIGURA 5.3.1); salvare la selezione con il comando “*Salva con nome...*” (provvedendo a impostare il salvataggio delle sole geometrie selezionate e utilizzando il formato CSV); memorizzare il file con il nome “*ESTRAZIONE.csv*” e salvarlo nella stessa cartella del file Python; richiamare lo *script* di Python “*EstrazioneParteMatrice.py*” in un *prompt* di comandi o lanciare l’esecuzione nell’ambiente di Wing. A seguito della procedura, verrà restituito, a video, il nuovo nome del file OMX creato, il quale, come i precedenti, è memorizzato nella stessa cartella dove è alloggiato il file Python.

Il nuovo file OMX creato è esplorabile, nel suo contenuto, attraverso il visualizzatore, come mostrato nella FIGURA 5.3.2.



	3040	3028	3027	3031	3030	3043	3042	3044
3040	23290.22	93.08	3841.22	193.69	189.34	2600.97	3748.62	3088.14
3028	198.61	28128.29	95.0	0.0	1957.97	0.0	0.0	93.08
3027	2675.73	95.0	10113.1	0.0	0.0	96.5	198.52	204.02
3031	193.69	0.0	0.0	5271.2	284.49	0.0	0.0	0.0
3030	351.2	1957.97	0.0	382.31	16122.83	0.0	0.0	0.0
3043	2603.69	0.0	96.5	0.0	0.0	15364.28	1558.76	390.94
3042	3648.22	0.0	198.52	0.0	0.0	1662.47	6116.08	485.44
3044	3283.97	93.08	204.02	0.0	0.0	390.94	485.44	15158.14
3046	0.0	768.54	0.0	387.77	4243.21	0.0	0.0	0.0
3045	0.0	93.08	0.0	105.13	488.28	0.0	0.0	0.0
3047	0.0	197.99	0.0	0.0	204.47	0.0	0.0	0.0
2017	129.98	490.01	0.0	850.92	459.82	0.0	0.0	0.0
2003	0.0	265.32	0.0	494.76	254.85	0.0	0.0	0.0
2019	0.0	33.0	0.0	291.65	0.0	0.0	0.0	0.0
2002	188.3	735.67	103.39	197.88	576.23	0.0	0.0	93.66
2018	0.0	160.73	0.0	943.92	165.66	0.0	0.0	0.0
	740.50	757.04	460.73	0.0	170.17	0.0	0.0	160.14

FIGURA 5.3.2 Apertura del nuovo file OMX creato, dal nome “*IMQ2013\_estrazione.omx*”, e visualizzazione dei dati contenuti.

## 5.4 AGGREGAZIONE DI UNA MATRICE IN BASE AGLI ATTRIBUTI

Utilizzando lo stesso procedimento appena visto, è possibile estrarre da un file CSV con Python, non solo i nomi delle zone, ma tutti i campi della tabella attributi di un vettore o di una selezione in QGIS. In questo specifico caso si è scelto di estrarre le informazioni di tutte le zone campionarie del Piemonte. È stato creato così il file dal nome “*AGGREGAZIONE.csv*”, il quale contiene una copia per intero della tabella attributi del *layer* rappresentante le zone campionarie. Utilizzando quindi un campo della tabella, lo scopo è stato quello di aggregare le matrici O/D sulla base di questo attributo.

Come riportato nella FIGURA 5.4.1, la tabella legata al *layer* delle zone campionarie permette solamente l’aggregazione o sulla base della provincia o sulla base della macro-area di appartenenza della zona, poiché questi due attributi sono i soli disponibili.

	COD_ZONA	PROVINCIA	MACRO_AREA	NOME
0	R312	NO	ProvNovara	Arona
1	R313	NO	ProvNovara	Borgomanero
2	R300	NO	ProvNovara	Novara
3	R305	NO	ProvNovara	Treccate
4	R304	NO	ProvNovara	Galliate
5	R303	NO	ProvNovara	Vespolate
6	R302	NO	ProvNovara	Caltignaga
7	R301	NO	ProvNovara	Grignasco
8	R311	NO	ProvNovara	SanMaurizioD'Op...
9	R310	NO	ProvNovara	Gattico
10	R309	NO	ProvNovara	Gozzano

FIGURA 5.4.1 Porzione della tabella attributi, del *layer* QGIS contenente le informazioni sulle zone campionarie del Piemonte dell’indagine IMQ 2013.

Lo *script* utile per compiere questa operazione sul file OMX è riportato di seguito, ed è stato nominato “*AggregazioneMatrice.py*”.

```
001 import openmatrix as omx
```

```
002 import openpyxl
003 import numpy as np
004 import csv
005
006 def eliminadoppi(lista):
007     for i in lista:
008         if lista.count(i)>1:
009             lista.remove(i)
010             eliminadoppi(lista)
011     return lista
012
013 #Apertura del file OMX di input
014 OMXfileI = omx.open_file('IMQ2013_zonecampionarie.omx')
015 print("File OMX di riferimento: ",OMXfileI.filename,'\n')
016 #Apertura del file OMX di output
017 OMXfileO = omx.open_file('IMQ2013_aggregazione.omx','w')
018
019 #Caricamento delle informazioni sulla mappatura (per conversione cod.
020 alfanumerico - cod. numerico)
021 excel_document = openpyxl.load_workbook('ZoneCampionarie2013.xlsx')
022 nsheet = excel_document.sheetnames
023 sheet = excel_document[nsheet[0]]
024 zoneA = list()
025 zoneN = np.zeros(209)
026 for j in range(0,209):
027     zoneA.append(sheet.cell(row=j+2,column=2).value)
028     zoneN[j] = sheet.cell(row=j+2,column=1).value
029
030 #Ottenimento dei dati di input
031 sel = int(input("Inserire criterio di aggregazione (1-per province,
032                                                         2-per macro-aree): "))
033
034 #Apertura del file contenente le zone e il criterio di selezione
035 INfile = open('AGGREGAZIONE.csv','r')
036 reader = csv.reader(INfile, delimiter=',')
037 zoneORD = list()
038 CRIT = list()
039 criterio = list()
040 for row in reader:
041     zoneORD.append(row[0])
042     CRIT.append(row[sel])
043     criterio.append(row[sel])
044 del zoneORD[0]
045 zoneORD.append("Esterno")
046 del CRIT[0]
047 CRIT.append("EXT")
048 del criterio[0]
049 criterio.append("EXT")
050 criterio = eliminadoppi(criterio)
051 dim = len(criterio)
```

```

052
053 #Estrazione delle matrici
054 for pop in ['TO','PIE']:
055     for mez in ['Pr','Pu']:
056         for per in ['IG','OP']:
057             for sco in ['sr','t']:
058                 if not (sco=='sr' and per=='OP'):
059                     NEWmatrOD = np.zeros((dim,dim))
060                     for i in range(0,209):
061                         indexO = zoneORD.index(zoneA[i])
062                         indO = criterio.index(CRIT[indexO])
063                         for j in range(0,209):
064                             indexD = zoneORD.index(zoneA[j])
065                             indD = criterio.index(CRIT[indexD])
066                             NEWmatrOD[indO][indD] = NEWmatrOD[indO][indD]+
067                                 OMXfileI['Res'+pop+'m'+mez+per+sco][i][j]
068 OMXfileO['Res'+pop+'m'+mez+per+sco] = NEWmatrOD
069 #Scrittura dei metadati complementari
070 OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Indagine = 'IMQ2013'
071 if pop=='TO':
072     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Popolazione =
073         'Provincia di Torino'
074     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.NumCampione = 32460
075 else:
076     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Popolazione =
077         'Regione Piemonte'
078     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.NumCampione = 52119
079 if mez=='Pr':
080     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Modo = 'Privato'
081 else:
082     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Modo = 'Pubblico'
083 if per=='IG':
084     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Periodo =
085         'Intera giornata'
086 else:
087     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Periodo = 'Ora di
088         punta (arrivo 07:43-08:42)'
089 if sco=='sr':
090     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Scopo = 'Tutti
091         escluso ritorno'
092 else:
093     OMXfileO['Res'+pop+'m'+mez+per+sco].attrs.Scopo = 'Tutti'
094
095 #Associazione di un file .txt contenente la mappatura
096 fileMAP = open("mappatura_AGGREGAZIONE.txt","w")
097 i=1
098 for element in criterio:
099     fileMAP.write((str(i)+';'+element)); fileMAP.write('\n')
100     i = i+1
101 print("\nFILE OMX CREATO: IMQ2013_aggregazione.omx")

```



```
102 print("FILE TXT CREATO: mappatura_AGGREGAZIONE.txt")
103
104 #Chiusura dei file aperti
105 OMXfileI.close()
106 OMXfileO.close()
107 INfile.close()
108 fileMAP.close()
```

Il presente *script* ripercorre più o meno le stesse operazioni di quello del precedente Paragrafo 5.3 Estrazione di parte di matrice, a meno di qualche piccola differenza dovuta allo specifico caso. In particolare, viene definita una funzione, chiama “*eliminadoppi*”, che serve ad eliminare da un vettore gli elementi che si ripetono nella lista, e che sarà utile più avanti nello script a determinare la dimensione delle nuove matrici e a individuare la nuova zonizzazione.

Il file CSV viene dunque aperto, e caricato l’elenco delle zone nel vettore “*zoneORD*” (in realtà si possiede già l’elenco delle zone, nel vettore “*zoneA*”, ma nel file CSV non hanno lo stesso ordine, per questo motivo è necessario disporre di un riferimento alla zona). Viene caricato dal file anche la colonna con l’attributo scelto per l’aggregazione (la scelta viene effettuata dall’utente, si veda lo *script* alle righe 31-32). A questi vettori va aggiunta infine una cella che conterrà gli spostamenti da, e verso, l’esterno del Piemonte. Viene quindi messo in pratica il consueto procedimento per la scrittura delle matrici e dei metadati nel nuovo file OMX. In particolare, passando in rassegna le celle delle matrici originarie una per volta, l’aggregazione avviene ricercando nel vettore “*zoneORD*” l’indice relativo alla zona campionaria osservata. Con questo indice si accede al vettore degli attributi e si determina la nuova zona campionaria di appartenenza.

Poiché, come già ribadito più volte, la mappatura nel formato OMX non accetta formati di dati diversi dai numeri interi, e non avendo parametri “standard” per effettuare una conversione tra codice alfanumerico – codice numerico (come invece è stato effettuato in occasione della redazione delle matrici OMX per zone campionarie, si veda il Paragrafo

4.3.2 Matrici per zone campionarie), è stato scelto di generare un file TXT contenente i riferimenti alle nuove zone (si veda lo *script* alle righe 96-100).

L'*output* prodotto a video dallo *script* è il seguente.

```
File OMX di riferimento:  IMQ2013_zonecampionarie.omx
```

```
Inserire criterio di aggregazione (1-per province, 2-per macro-aree): 1
```

```
FILE OMX CREATO:  IMQ2013_aggregazione.omx
```

```
FILE TXT CREATO: mappatura_AGGREGAZIONE.txt
```

Ricapitolando, per far eseguire correttamente lo *script* al calcolatore, si devono seguire i seguenti passaggi: aprire la mappa delle zone campionarie in QGIS; selezionare le zone di interesse; salvare la selezione con il comando “*Salva con nome...*” (provvedendo a impostare il salvataggio delle sole geometrie selezionate e utilizzando il formato CSV); memorizzare il file con il nome “*AGGREGAZIONE.csv*” e salvarlo nella stessa cartella del file Python; richiamare lo *script* di Python “*AggregazioneMatrice.py*” in un *prompt* di comandi o lanciare l'esecuzione nell'ambiente di Wing; specificare, come *input* da tastiera, il campo della tabella attributi sulla base del quale effettuare l'aggregazione. A seguito della procedura verranno restituiti due file, alloggiati entrambi nella cartella dove è presente il file Python: il primo è un file OMX contenente le matrici aggregate, mentre il secondo è un file di testo contenente la nuova mappatura delle zone.

Il nuovo file OMX creato è esplorabile, nel suo contenuto, attraverso il visualizzatore, come mostrato nella FIGURA 5.4.2.

Il file di testo “*mappatura\_AGGREGAZIONE.txt*”, contenente i riferimenti per la mappatura, è riportato di seguito.

```
1;NO  
2;CN  
3;VB  
4;AT  
5;AL
```

6;BI  
7;VC  
8;TO  
9;EXT

	1	2	3	4	5	6	7	8	9
1	12655.31999999...	0.0	0.0	0.0	195.95	0.0	521.79	565.03	5787.81999999...
2	0.0	26403.01000000...	0.0	194.93	102.26	0.0	0.0	7967.8500000000...	387.96
3	292.73	0.0	4434.04	0.0	0.0	0.0	0.0	0.0	1468.16
4	0.0	501.1799999999...	0.0	8314.42	429.7999999999...	0.0	0.0	4217.43	0.0
5	0.0	93.83	0.0	96.82	11499.64999999...	0.0	368.61	1419.86	5089.17
6	91.75	0.0	0.0	0.0	0.0	1746.76	294.78	888.9699999999...	599.02
7	1551.82	0.0	0.0	0.0	275.21	604.41	3335.16	2039.76	853.1800000000...
8	434.68	1451.67999999...	0.0	65.2	301.96	0.0	859.5000000000...	338800.84000000...	2288.77999999...
9	0.0	0.0	0.0	0.0	213.94	0.0	0.0	0.0	191.15

FIGURA 5.4.2 Apertura del nuovo file OMX creato, dal nome “*IMQ2013\_aggregazione.omx*”, e visualizzazione dei dati contenuti.

## 5.5 OTTENIMENTO DI INFORMAZIONI DI SINTESI

In questo ultimo esempio si vuole mostrare come effettuare un'elaborazione dei dati contenuti nel file OMX. Si è scelto, a scopo esemplificativo, di determinare la percentuale di scelta del mezzo pubblico sugli spostamenti totali effettuati in uscita da ogni zona campionaria. Lo *script* che permette questo tipo di elaborazione, chiamato “*SceltaMezzoPubblico.py*”, è riportato di seguito.

```
01 import openmatrix as omx
02 import numpy as np
03 import openpyxl
04
05 #Apertura del file OMX di input
06 OMXfileI = omx.open_file('IMQ2013_zonecampionarie.omx')
07 print("File OMX di riferimento: ",OMXfileI.filename,'\n')
08 #Apertura del file OMX di output
```

```

09 OMXfile0 = omx.open_file('IMQ2013_scltamezzopubblico.omx','w')
10 #Apretura del file di scrittura dei risultati
11 filex100 = open("SceltaMezzoPubblico.txt","w")
12 filex100.write('COD_ZONA;UTILIZZOX100;coord_X;coord_Y\n')
13
14 #Caricamento delle informazioni sulla mappatura
15 excel_document = openpyxl.load_workbook('ZoneCampionarie2013.xlsx')
16 nsheet = excel_document.sheetnames
17 sheet = excel_document[nsheet[0]]
18 zoneA = list()
19 zoneN = np.zeros(209)
20 for j in range(0,209):
21     zoneA.append(sheet.cell(row=j+2,column=2).value)
22     zoneN[j] = sheet.cell(row=j+2,column=1).value
23
24 #Estrazione delle matrici
25 for pop in ['TO','PIE']:
26     for per in ['IG','OP']:
27         for sco in ['sr','t']:
28             if not (sco=='sr' and per=='OP'):
29                 NEWmatr = np.zeros((209,209))
30                 for i in range(0,209):
31                     for j in range(0,209):
32                         if not OMXfileI['Res'+pop+'mPu'+per+sco][i][j]+
33                             OMXfileI['Res'+pop+'mPr'+per+sco][i][j] == 0:
34                             NEWmatr[i][j] = OMXfileI['Res'+pop+'mPu'+per+sco][i][j]/
35                                 (OMXfileI['Res'+pop+'mPu'+per+sco][i][j]+
36                                 OMXfileI['Res'+pop+'mPr'+per+sco][i][j])*100
37                 OMXfileO['Res'+pop+'x100mPu'+per+sco] = NEWmatr
38                 #Scrittura dei metadati complementari
39                 OMXfileO['Res'+pop+'x100mPu'+per+sco].attrs.Indagine = 'IMQ 2013'
40                 OMXfileO['Res'+pop+'x100mPu'+per+sco].attrs.Modo = '% di utilizzo
41                                     del mezzo pubblico'
42                 if pop=='TO':
43                     OMXfileO['Res'+pop+'x100mPu'+per+sco].attrs.Popolazione =
44                                     'Provincia di Torino'
45                     OMXfileO['Res'+pop+'x100mPu'+per+sco].attrs.NumCampione = 32460
46                 else:
47                     OMXfileO['Res'+pop+'x100mPu'+per+sco].attrs.Popolazione =
48                                     'Regione Piemonte'
49                     OMXfileO['Res'+pop+'x100mPu'+per+sco].attrs.NumCampione = 52119
50                 if per=='IG':
51                     OMXfileO['Res'+pop+'x100mPu'+per+sco].attrs.Periodo = 'Intera
52                                     giornata'
53                 else:
54                     OMXfileO['Res'+pop+'x100mPu'+per+sco].attrs.Periodo = 'Ora di
55                                     punta (arrivo 07:43-08:42)'
56                 if sco=='sr':
57                     OMXfileO['Res'+pop+'x100mPu'+per+sco].attrs.Scopo = 'Tutti
58                                     escluso ritorno'

```

```

59         else:
60             OMXfile0['Res'+pop+'x100mPu'+per+sco].attrs.Scopo = 'Tutti'
61
62 #Associazione della mappatura dell'indice
63 OMXfile0.create_mapping('Mappatura',zoneN)
64 print("FILE OMX CREATO:  IMQ2013_scltamezzopubblico.omx")
65
66 #Scrittura dati su file di testo
67 for i in range(0,209):
68     FProdPU = 0
69     FProdPR = 0
70     for j in range(0,209):
71         FProdPU = FProdPU + OMXfileI['ResPIEmPuOPT'][i][j]
72         FProdPR = FProdPR + OMXfileI['ResPIEmPrOPT'][i][j]
73     PERC = str(FProdPU/(FProdPU+FProdPR)*100)
74     cod = sheet.cell(row=i+2,column=2).value
75     cX = str(sheet.cell(row=i+2,column=7).value)
76     cY = str(sheet.cell(row=i+2,column=8).value)
77     filex100.write((cod+';'+PERC+';'+cX+';'+cY)); filex100.write('\n')
78 print("FILE TXT CREATO:  SceltaMezzoPubblico.txt")
79
80 #Chiusura dei file aperti
81 OMXfileI.close()
82 OMXfile0.close()
83 filex100.close()

```

Nello specifico, il programma apre il file OMX contenente le matrici per zone campionarie, quindi, per ogni casistica di popolazione, periodo e scopo, calcola la percentuale di scelta del mezzo pubblico sugli spostamenti effettuati con qualunque mezzo (si veda lo *script* alle righe 34-36). Quindi le salva, insieme ai relativi metadati e alla zonizzazione, in un nuovo file OMX, chiamato “*IMQ2013\_scltamezzopubblico.omx*”.

Insieme alla nuova raccolta di matrici, viene generato anche un file di testo, dal nome “*SceltaMezzoPubblico.txt*”, il quale riporta, per ogni zona campionaria, il relativo codice alfanumerico di identificazione, la percentuale di utilizzo del mezzo pubblico sugli spostamenti totali in uscita dalla zona, nell’ora di punta e per qualunque scopo (si veda lo *script* alle righe 71-72), e le due coordinate geografiche del centroide per la localizzazione in QGIS. L’*output* ottenuto a video è il seguente.

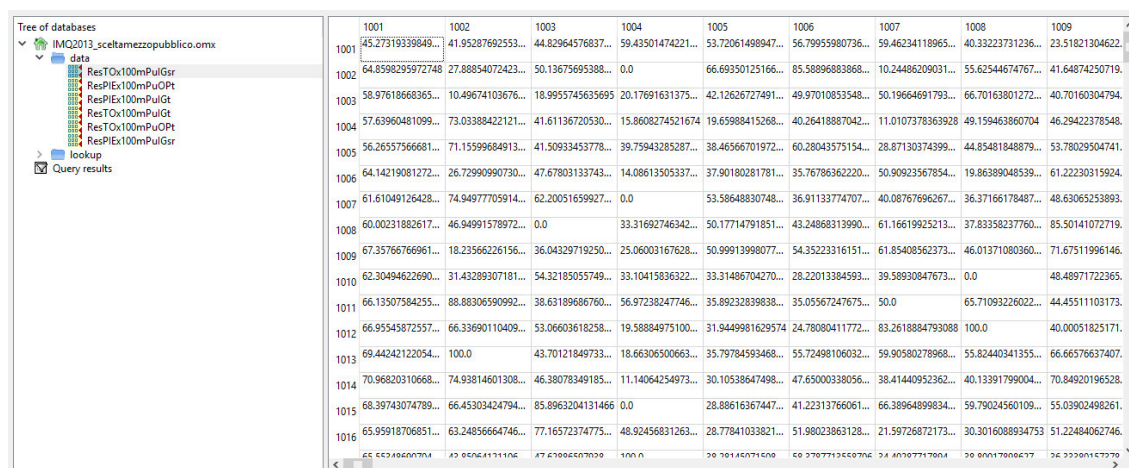
File OMX di riferimento: IMQ2013\_zonecampionarie.omx

FILE OMX CREATO: IMQ2013\_scltamezzopubblico.omx

FILE TXT CREATO: SceltaMezzoPubblico.txt

In questo caso, per eseguire correttamente lo *script* di Python “*SceltaMezzoPubblico.py*”, basta semplicemente richiamarlo in un *prompt* di comandi o lanciare l’esecuzione nell’ambiente di Wing. A seguito della procedura, verranno restituiti, a video, i nomi del file OMX e del file di testo creati, i quali, come i precedenti, sono memorizzati nella stessa cartella dove è alloggiato il file Python.

Al file OMX creato è possibile accedere, come di consueto, con il visualizzatore, come mostrato nella FIGURA 5.5.1.



	1001	1002	1003	1004	1005	1006	1007	1008	1009
1001	45.27319339849...	41.95287692553...	44.82964576837...	59.43501474221...	53.72061498947...	56.79955980736...	59.46234118965...	40.33223731236...	23.51821304622...
1002	64.8598295972748	27.88854072423...	50.13675695388...	0.0	66.69350125166...	85.58896883868...	10.24486209031...	55.62544674767...	41.64874250719.
1003	58.97618668365...	10.49674103676...	18.9955745635695	20.17691631375...	42.12626727491...	49.97010853548...	50.19664691793...	66.70163801272...	40.70160304794.
1004	57.63960481099...	73.03388422121...	41.61136720530...	15.8608274521674	19.65988415268...	40.26418887042...	11.0107378363928	49.159463860704	46.29422378548.
1005	56.26557566681...	71.15599684913...	41.50933453778...	39.75943285287...	38.46566701972...	60.28043575154...	28.87130374399...	44.85481848879...	53.78029504741.
1006	64.14219081272...	26.72999097030...	47.67803133743...	14.08613505337...	37.90180281781...	35.76786362220...	50.90923567854...	19.86389048539...	61.22230315924.
1007	61.61049126428...	74.94977705914...	62.20051659927...	0.0	53.58648830748...	36.91133774707...	40.08767696267...	36.37166178487...	48.63065253893.
1008	60.00231882617...	46.94991578972...	0.0	33.31692746342...	50.17714791851...	43.24868313990...	61.16619925213...	37.83358237760...	85.50141072719.
1009	67.35766766961...	18.23566226156...	36.04329719250...	25.06003167628...	50.99913998077...	54.35223316151...	61.85408562373...	46.01371080360...	71.67511996146.
1010	62.30494622690...	31.43289307181...	54.32185055749...	33.10415836322...	33.31486704270...	28.22013384593...	39.58930847673...	0.0	48.48971722365.
1011	66.13507584255...	88.88306590992...	38.63189686760...	56.97238247746...	35.89232839838...	35.05567247675...	50.0	65.71093226022...	44.45511103173.
1012	66.95545872557...	66.33690110409...	53.06603618258...	19.58884975100...	31.9449981629574	24.78080411772...	83.2618884793088	100.0	40.00051825171.
1013	69.44242122054...	100.0	43.70121849733...	18.66306500663...	35.79784593468...	55.72498106032...	59.9580278968...	55.82440341355...	66.66576637407.
1014	70.96820310668...	74.93814601308...	46.38078349185...	11.14064254973...	30.10538647498...	47.65000338056...	38.41440952362...	40.13391799004...	70.84920196528.
1015	68.39743074789...	66.45303424794...	85.8963204131466	0.0	28.88616367447...	41.22313766061...	66.38964898934...	59.79024560109...	55.03902498261.
1016	65.95918706851...	63.24856664746...	77.16572374775...	48.92456831263...	28.77841033821...	51.98023863128...	21.59726872173...	30.3016088934753	51.22484062746.

FIGURA 5.5.1 Apertura del nuovo file OMX creato, dal nome “*IMQ2013\_scltamezzopubblico.omx*”, e visualizzazione dei dati contenuti.

Si osservi che in questo esempio il numero delle matrici si è dimezzato, in quanto non è più presente la differenziazione per il mezzo utilizzato.

Di seguito, invece, è riportato un estratto del file di testo prodotto, dal nome “*SceltaMezzoPubblico.txt*”.

```
COD_ZONA;UTILIZZOx100;coord_X;coord_Y
Q001;28.250990195838728;396178.49496;4991495.20846
Q002;36.71350943788674;395977.85622;4989356.8478
Q003;39.75051066171206;394944.20973;4990130.57686
Q004;26.562770998729977;393210.20324;4990461.3596
Q005;44.03905975767699;393840.65741;4991550.94728
Q006;43.365894639426685;394598.76099;4993107.50811
Q007;42.741659000205175;396701.59293;4992847.56862
Q008;26.829644220244365;398583.68409;4992179.17396
[...]
```

Questo documento può essere importato nell'ambiente di QGIS, come effettuato nel Paragrafo 5.2 Determinazione del flusso totale prodotto/attratto da una zona, sotto forma di *layer* di punti grazie al riferimento delle coordinate del centroide della zona. A questo punto è possibile personalizzare la visualizzazione dei dati. Si è scelto, per esempio, di unire il *layer* contenente le zone campionarie a quello appena caricato, attraverso il comando “*Unisci attributi per posizione...*”, che si trova nel menu a tendina sotto “*Vettore*” – “*Strumenti di gestione dati*”. In questo modo si è creato un nuovo vettore di aree, sulla base dello *shapefile* delle zone campionarie, la cui tabella attributi contiene anche le informazioni sulla percentuale di scelta del mezzo pubblico. Si sono, dunque, evidenziate le aree in cui la percentuale risulta più alta, dalle altre, grazie a un colore più intenso, come riportato nella FIGURA 5.5.2 e nella FIGURA 5.5.3.

Anche in questo caso, la visualizzazione dei risultati in forma grafica, permette di captare l'andamento del fenomeno senza entrare nei dettagli dei singoli casi, e, come si poteva prevedere, le zone relative al centro della città di Torino, sono quelle in cui il trasporto pubblico viene scelto con più frequenza.

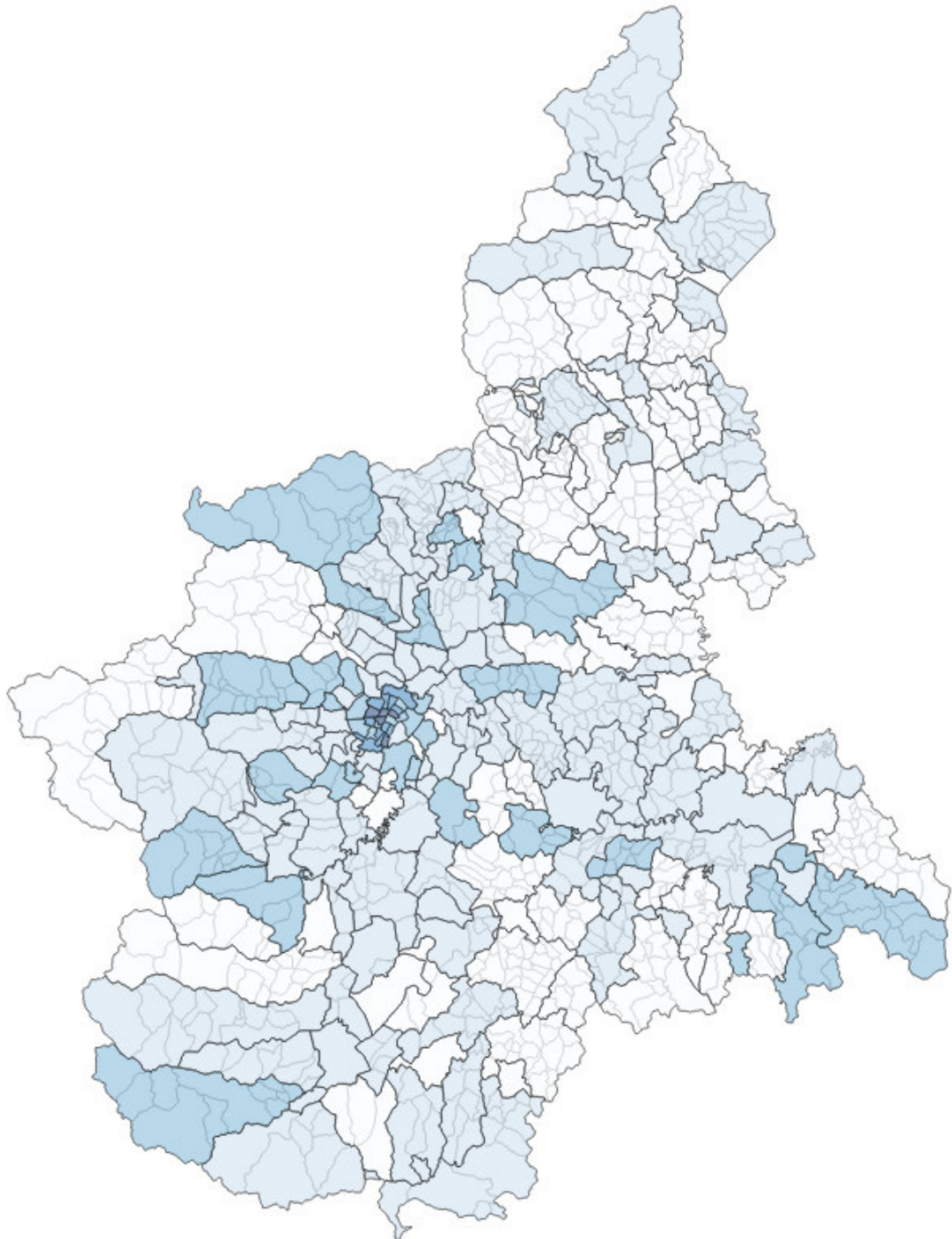


FIGURA 5.5.2 Rappresentazione delle informazioni estratte con lo script “*SceltaMezzoPubblico.py*” dal file OMX “*IMQ2013\_zonecampionarie.omx*”. In blu più scuro le aree in cui si è scelto maggiormente il trasporto pubblico a discapito di quello privato, per gli spostamenti totali effettuati in uscita dalla zona nell’ora di punta e per qualunque scopo. Visuale generale del fenomeno.



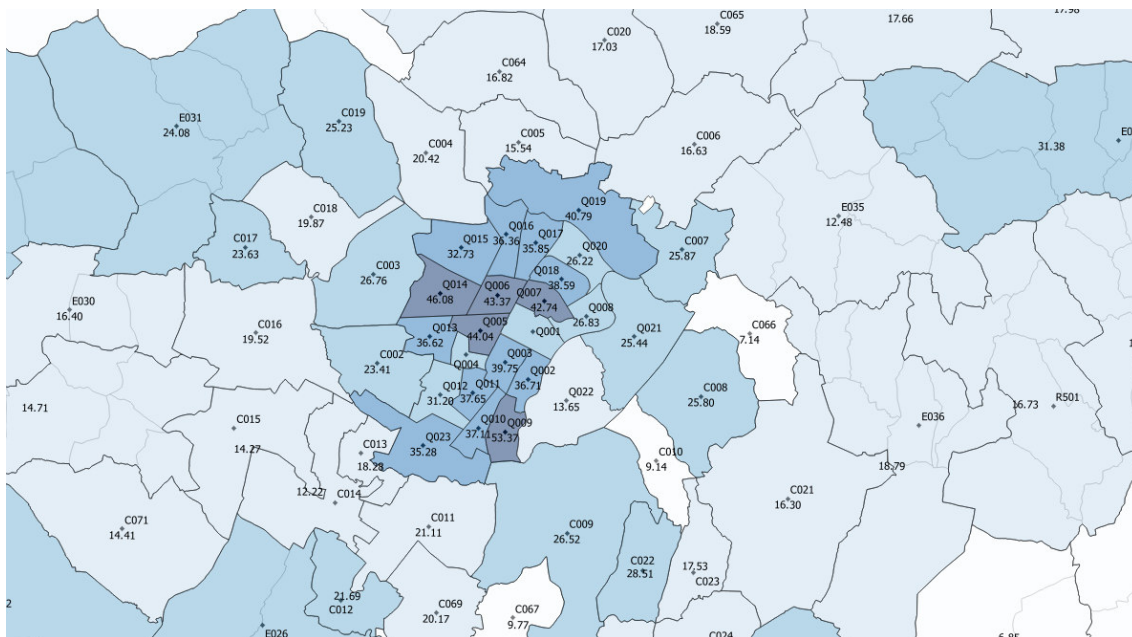


FIGURA 5.5.3 Rappresentazione delle informazioni estratte con lo script “*SceltaMezzoPubblico.py*” dal file OMX “*IMQ2013\_zonecampionarie.omx*”. In blu più scuro le aree in cui si è scelto maggiormente il trasporto pubblico a discapito di quello privato, per gli spostamenti totali effettuati in uscita dalla zona nell’ora di punta e per qualunque scopo. Visuale di dettaglio delle zone centrali del capoluogo piemontese. I numeri riportati accanto al codice della zona indicano la percentuale.

## 5.6 CONFRONTO CON MATLAB

In questo paragrafo si vogliono illustrare i procedimenti necessari ad effettuare le stesse operazioni viste nei precedenti, ma nell’ambiente MATLAB (versione R2016b), programma specifico per il calcolo matriciale (per maggiori dettagli si veda la Nota 35 a pagina 35). In questo caso le matrici di trasporto sono state salvate utilizzando un formato proprio del programma, denominato MAT. Si tratta di un formato binario leggibile **solo** da MATLAB che permette, con una sintassi di linguaggio molto semplice, di salvare una lista di variabili utilizzate in un file, in un documento esterno, che può quindi essere caricato in un altro *script*.

Il primo listato creato, dal nome “*ZoneCampionarie.m*”, ha riguardato la compilazione

delle matrici O/D sulla base dei dati forniti dall'Agenzia della Mobilità Piemontese nei file Excel, in modo analogo a quanto effettuato nel Paragrafo 4.3.2 Matrici per zone campionarie con lo *script* di Python, ed è riportato di seguito.

```

01 close all;
02 clear all;
03 clc;
04
05 %Caricamento delle informazioni sulla zonizzazione
06 [zoneN,zoneA] = xlsread('ZoneCampionarie2013.xlsx','A2:B210');
07 zoneA = string(zoneA);
08 coordX = xlsread('ZoneCampionarie2013.xlsx','G2:G210');
09 coordY = xlsread('ZoneCampionarie2013.xlsx','H2:H210');
10
11 %Creazione delle matrici O/D
12 [ResPIEmPrIGt] = estrazionematrice('IMQ2013_MPrIGResPIE.xlsx',zoneA);
13 [ResPIEmPrIGsr] = estrazionematrice('IMQ2013_MPrIGResPIEsr.xlsx',zoneA);
14 [ResTOMPrIGt] = estrazionematrice('IMQ2013_MPrIGRestO.xlsx',zoneA);
15 [ResTOMPrIGsr] = estrazionematrice('IMQ2013_MPrIGRestOsr.xlsx',zoneA);
16 [ResPIEmPrOPt] = estrazionematrice('IMQ2013_MPrOPResPIE.xlsx',zoneA);
17 [ResTOMPrOPt] = estrazionematrice('IMQ2013_MPrOPRestO.xlsx',zoneA);
18 [ResPIEmPuIGt] = estrazionematrice('IMQ2013_MPuIGResPIE.xlsx',zoneA);
19 [ResPIEmPuIGsr] = estrazionematrice('IMQ2013_MPuIGResPIEsr.xlsx',zoneA);
20 [ResTOMPuIGt] = estrazionematrice('IMQ2013_MPuIGRestO.xlsx',zoneA);
21 [ResTOMPuIGsr] = estrazionematrice('IMQ2013_MPuIGRestOsr.xlsx',zoneA);
22 [ResPIEmPuOPt] = estrazionematrice('IMQ2013_MPuOPResPIE.xlsx',zoneA);
23 [ResTOMPuOPt] = estrazionematrice('IMQ2013_MPuOPRestO.xlsx',zoneA);
24
25 %Salvataggio delle matrici create nel file .mat
26 save IMQ2013_zonecampionarie zoneA coordX coordY ResPIEmPrIGt
27     ResPIEmPrIGsr ResTOMPrIGt ResTOMPrIGsr ResPIEmPrOPt ResTOMPrOPt
28     ResPIEmPuIGt ResPIEmPuIGsr ResTOMPuIGt ResTOMPuIGsr ResPIEmPuOPt
29     ResTOMPuOPt
30
31 %Definizione della funzione di compilazione della matrice O/D
32 function [matr] = estrazionematrice(nomefile,zone)
33 matr = zeros(length(zone),length(zone));
34 [SPOST,OeD] = xlsread(nomefile,'A9:E6884');
35 ORIG = string(OeD(:,1));
36 DEST = string(OeD(:,3));
37 for i = 1:length(SPOST)
38     matr(find(zone==ORIG(i)),find(zone==DEST(i))) = SPOST(i);
39 end
40 end

```

Si noti che in questo caso, a differenza di quanto fatto con lo *script* di Python (si veda il

Paragrafo 4.3.2 Matrici per zone campionarie), non si è potuto operare con dei cicli *for* per “ricostruire” il nome del file Excel e delle relative matrici create, ma è stato necessario effettuare l’operazione di lettura dei file XLSX ben 12 volte, tante quante i documenti. La definizione di una funzione che, dati in ingresso il nome del file e della zonizzazione, ha restituito la matrice compilata, ha comunque reso più compatto lo *script*.

A questo punto nel file “*IMQ2013\_zonecampionarie.mat*”, analogamente all’omonimo file OMX, sono raccolte le 12 matrici O/D prodotte e il relativo vettore di associazione alle zone campionarie contenente i codici **alfanumerici**. Infatti, a differenza del formato OMX, il formato MAT permette il salvataggio di qualunque tipo di dato. Si sono persi, invece, i metadati associati alle singole matrici, che in quest’ultimo formato non possono essere accolti. Nel file si memorizzano anche le coordinate dei centroidi delle zone, in modo da avere a disposizione i dati nel momento in cui si stamperanno i file TXT di *output* per l’interazione con QGIS. Si fa notare, inoltre, che il file MAT ha dimensione pari a 214 KB, mentre il file OMX 493 KB; questo formato proprietario di MATLAB, dunque, occupa circa il 60% in meno di memoria.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	4.3710e+03	2.7708e+03	4.2825e+03	1.0935e+03	3.3641e+03	3.1295e+03	3.9872e+03	2.7439e+03	2.5021e+03	2.0046e+03	1302	1.7149e+03	2.5688e+03	2.0111e+03	1.0364e+03	1.4998e+03	2.7437e+03
2	1.9569e+03	2.2129e+03	1.5187e+03	492.4900	402.4900	912.8400	1.1053e+03	706.4100	2.1339e+03	1.1063e+03	412.1100	798.6500	123.7600	293.5500	302.0300	234.4900	502.0000
3	4.0041e+03	1.5270e+03	6.3724e+03	1.8809e+03	2.1551e+03	1.4066e+03	1.1134e+03	295.0300	1.3505e+03	1.7961e+03	2.0360e+03	1.3513e+03	1.9109e+03	794.1800	32.9600	263.0200	637.0000
4	1.2814e+03	391.9700	1.9825e+03	3.0723e+03	2.3210e+03	1.1854e+03	798.9100	493.8700	763.6400	604.5800	1.2104e+03	1.8841e+03	2.5051e+03	1.2968e+03	234.7800	67.4400	391.0000
5	3.2670e+03	597.7000	2.3496e+03	2.3099e+03	2.6896e+03	1.9067e+03	1.2091e+03	802.7800	689.7800	606.4900	1.0473e+03	1.4672e+03	3.5814e+03	1.9927e+03	899.3100	427.0500	983.0000
6	3.2302e+03	1.2138e+03	1.4098e+03	890.5600	2.0432e+03	6.6015e+03	2.4030e+03	1.6245e+03	1.2037e+03	1.0105e+03	1.0249e+03	1.4909e+03	2.1588e+03	3.4189e+03	1.4623e+03	1.9661e+03	2.1743e+03
7	3.2676e+03	802.6800	1.0112e+03	892.9000	908.8500	2.6611e+03	2.8245e+03	2.0133e+03	633.7800	537.1300	603.1600	609.2600	510.0700	1.1057e+03	497.5900	1.9442e+03	2.2562e+03
8	2.7382e+03	444.1300	396.5900	694.1900	704.8500	1.4167e+03	1.8998e+03	3.3553e+03	498.6000	701.2600	510.4600	102.2600	597.7100	302.7200	297.0800	807.9400	1.3035e+03
9	2.8497e+03	1.8172e+03	1.4225e+03	697.3900	886.1200	1.3005e+03	860.3000	901.8400	1.6590e+03	2.5368e+03	808.0800	1.9798e+03	593.8600	641.0200	543.9600	307.7000	675.0000
10	2.0105e+03	1.1973e+03	1.7972e+03	604.5800	705.2700	1.3154e+03	507.1400	600.4800	2.5035e+03	6.3641e+03	2.2423e+03	2.8328e+03	1.4571e+03	400.9600	500.1900	930.5400	781.0000
11	1.3961e+03	99.7000	1.5960e+03	1.4139e+03	1.1132e+03	924.3500	603.1600	510.4600	940.4200	1.5812e+03	3.5155e+03	3.5899e+03	889.0600	808.4500	199.2800	362.9800	815.0000
12	1.7235e+03	503.5600	2.2190e+03	1.4909e+03	1.4691e+03	1.5933e+03	509.3600	102.2600	1.9249e+03	2.9982e+03	3.4771e+03	5.9438e+03	1.6284e+03	964.6800	306.2400	335.8900	587.0000
13	2.1724e+03	91.9200	1.6841e+03	2.6918e+03	3.6725e+03	2.0692e+03	510.0700	782.1700	493.6800	1.6263e+03	988.9300	1.8200e+03	6.3272e+03	5.2919e+03	655.7300	791.8700	676.0000
14	1.9368e+03	391.3700	795.6000	1.3980e+03	2.1022e+03	3.0121e+03	1.0387e+03	405.0600	610.8000	400.9600	710.6300	928.7100	4.7502e+03	6.4207e+03	1.5071e+03	1.4375e+03	845.0000
15	1.0025e+03	202.1300	196.6700	132.5300	896.1000	1.4016e+03	301.2300	196.3000	511.5300	705.5300	199.2800	405.9200	686.5100	1.4069e+03	4.4750e+03	2.4322e+03	1.8063e+03
16	1.4265e+03	133.9700	263.0200	171.5300	486.6500	1.8637e+03	1.7698e+03	807.9400	408.5700	824.3500	362.9800	506.5100	623.4200	1.2634e+03	2.1066e+03	2.4534e+03	3.6227e+03
17	2.6473e+03	666.5900	902.7100	295.4900	1.0798e+03	2.2346e+03	2.3094e+03	1.3036e+03	709.6700	614.7000	685.1800	329.0500	483.8200	1.2359e+03	1.8445e+03	3.5185e+03	9.6238e+03
18	1.4244e+03	398.2300	300.6900	512.2600	102.4900	1.0124e+03	2.3191e+03	904.9600	204.0300	31.9300	203.9600	199.9300	102.4900	403.1100	600.7800	881.0300	1.2036e+03
19	1.1313e+03	298.6800	396.9000	316.7200	208.1100	497.0800	2.7033e+03	675.9200	314.5300	0	207.7200	306.2300	697.7300	497.5000	1.1255e+03	972.9600	1.7848e+03
20	1.2035e+03	308.4100	404.0900	0	194.5700	498.2100	406.5300	1.2169e+03	594.8000	302.3500	97.9500	0	394.7500	0	98.6200	364.9700	586.0000
21	1.9876e+03	594.0500	197.8300	275.8400	502.7900	702.5200	887.2900	1.7914e+03	695.7300	490.8500	0	201.9600	291.5200	302.1700	100.8800	191.1200	405.0000
22	3.0938e+03	1.1332e+03	903.9300	197.0600	410.4900	594.6500	300.3500	988.4700	1.1121e+03	394.4700	399.3500	499.8500	490.1400	502.5500	196.6000	98.5000	209.0000
23	1.9708e+03	810.5900	1.0988e+03	1.0393e+03	1.0314e+03	905.2300	308.0600	201.6900	1.6930e+03	4.3217e+03	1.0237e+03	2.1846e+03	687.3500	1.0013e+03	104.3100	401.8000	842.0000

FIGURA 5.6.1 Apertura di una matrice salvata nel file “*IMQ2013\_zonecampionarie.mat*” nell’ambiente di esplorazione delle variabili di MATLAB.

Nella FIGURA 5.6.1 è riportata, a titolo esemplificativo, una matrice prodotta. In particolare, si è scelto di rappresentare la stessa matrice della FIGURA 4.3.1 in modo da poter confrontare i risultati ottenuti con i due metodi: OMX e MAT. Si noti la coincidenza dei valori di flusso riportati.

Si vede ora nel dettaglio come accedere a questi dati in formato MAT, ripercorrendo gli esempi già presentati per il formato OMX precedentemente nel capitolo.

### **5.6.1 Estrazione del singolo flusso**

Seguendo quanto già visto nel Paragrafo 5.1 Estrazione del flusso tra due zone definite, si vogliono determinare gli spostamenti effettuati tra due comuni della Cintura di Torino: Chieri e Santena. Attraverso la mappa di QGIS, si sono determinati i due codici identificativi delle due zone, rispettivamente C021 e C024. Lo *script* dal nome “*EstrazioneSingoloValore.m*”, riportato di seguito, permette questo tipo di elaborazione.

```

01 clear all;
02 close all;
03 clc;
04
05 %Caricamento delle matrici O/D e della mappatura
06 load 'IMQ2013_zonecampionarie';
07 fprintf('File MAT di riferimento:  IMQ2013_zonecampionarie.mat\n\n');
08
09 %Ottenimento dei dati di ingresso
10 zona1 = input('Inserire codice prima zona: ');
11 zona2 = input('Inserire codice seconda zona: ');
12 modo = input('Inserire il modo (Pr-privato, Pu-pubblico, T-totali): ');
13 per = input('Periodo (IG-intera giornata, OP-ora di punta): ');
14 if per == 'IG';
15     scopo = input('Scopo (t-tutti, sr-tutti escluso ritorno): ');
16 else;
17     scopo = 't';
18 end;
19
20 %Estrazione dei due flussi (andata/ritorno) in base all'input fornito
21 fprintf('\n- SPOSTAMENTI REGISTRATI TRA LE DUE ZONE -');
22 if modo=='T';
23     nomeMATR1 = strcat('ResPIEmPu',per,scopo);
24     nomeMATR2 = strcat('ResPIEmPr',per,scopo);

```

```

25   if per=='IG',scopo=='sr';
26       [SPOST1to2,SPOST2to1] = singval(ResPIEmPuIGsr,zona1,zona2,zonaA)+
27           singval(ResPIEmPrIGsr,zona1,zona2,zonaA);
28   end; if per=='IG',scopo=='t';
29       [SPOST1to2,SPOST2to1] = singval(ResPIEmPuIGt,zona1,zona2,zonaA)+
30           singval(ResPIEmPrIGt,zona1,zona2,zonaA);
31   end; if per=='OP',scopo=='t';
32       [SPOST1to2,SPOST2to1] = singval(ResPIEmPuOPt,zona1,zona2,zonaA)+
33           singval(ResPIEmPrOPt,zona1,zona2,zonaA);
34   end;
35   fprintf('\nMatrici di riferimento: ');fprintf(nomeMATR1);
36           fprintf(' / ');fprintf(nomeMATR2);fprintf('\n');
37   else;
38       nomeMATR = strcat('ResPIEm',modo,per,scopo);
39       if modo=='Pu',per=='IG',scopo=='sr';
40           [SPOST1to2,SPOST2to1] = singval(ResPIEmPuIGsr,zona1,zona2,zonaA);
41       end; if modo=='Pu',per=='IG',scopo=='t';
42           [SPOST1to2,SPOST2to1] = singval(ResPIEmPuIGt,zona1,zona2,zonaA);
43       end; if modo=='Pu',per=='OP',scopo=='t';
44           [SPOST1to2,SPOST2to1] = singval(ResPIEmPuOPt,zona1,zona2,zonaA);
45       end; if modo=='Pr',per=='IG',scopo=='sr';
46           [SPOST1to2,SPOST2to1] = singval(ResPIEmPrIGsr,zona1,zona2,zonaA);
47       end; if modo=='Pr',per=='IG',scopo=='t';
48           [SPOST1to2,SPOST2to1] = singval(ResPIEmPrIGt,zona1,zona2,zonaA);
49       end; if modo=='Pr',per=='OP',scopo=='t';
50           [SPOST1to2,SPOST2to1] = singval(ResPIEmPrOPt,zona1,zona2,zonaA);
51       end;
52       fprintf('\nMatrice di riferimento: ');fprintf(nomeMATR);fprintf('\n');
53   end;
54
55   %Stampa a video dei risultati
56   fprintf(zona1);fprintf(' --> ');fprintf(zona2);fprintf(' = ');
57           fprintf('%.2f',SPOST1to2);fprintf('\n');
58   fprintf(zona2);fprintf(' --> ');fprintf(zona1);fprintf(' = ');
59           fprintf('%.2f',SPOST2to1);fprintf('\n');
60
61   %Definizione della funzione di estrazione dei valori
62   function [s1to2,s2to1] = singval(matrice,Z1,Z2,zona);
63   s1to2 = matrice(find(zona==Z1),find(zona==Z2));
64   s2to1 = matrice(find(zona==Z2),find(zona==Z1));
65   end

```

L'idea di base è la stessa del corrispondente file Python, solo che in MATLAB, come già accennato, non è possibile “ricostruire” il nome di una variabile; quindi, ottenute le informazioni dall'*input* è stato necessario impostare una serie di dichiarazioni condizionali *if* per determinare la corretta matrice da cui estrarre i dati ricercati. Anche in questo caso, ci si

è avvalsi della definizione di una funzione per l'esecuzione delle operazioni di estrazione (si veda lo *script* alle righe 61-65). L'*output* ottenuto a video è quello rappresentato di seguito.

File MAT di riferimento: IMQ2013\_zonecampionarie.mat

```
Inserire codice prima zona: 'C021'
Inserire codice seconda zona: 'C024'
Inserire il modo (Pr-privato, Pu-pubblico, T-totali): 'Pr'
Periodo (IG-intera giornata, OP-ora di punta): 'OP'
```

```
Matrice di riferimento: ResPIEmPrOPT
- SPOSTAMENTI REGISTRATI TRA LE DUE ZONE -
C021 --> C024 = 64.93
C024 --> C021 = 33.97
```

Come si può osservare, il risultato ottenuto è lo stesso del Paragrafo 5.1 Estrazione del flusso tra due zone definite.

### ***5.6.2 Determinazione del flusso totale prodotto/attratto***

Il secondo *script*, invece, riguarda la determinazione del flusso totale prodotto e attratto da una specifica zona. Nell'esempio riportato nel Paragrafo 5.2 Determinazione del flusso totale prodotto/attratto da una zona, sono stati ricavati gli spostamenti in entrata e in uscita dal quartiere torinese denominato “Lingotto”, codice zona campionaria Q010. Lo *script* MATLAB che permette questo tipo di operazione, dal nome “*TotaleProdottoAttratto.m*”, è riportato di seguito.

```
001 clear all;
002 close all;
003 clc;
004
005 %Caricamento delle matrici O/D e della mappatura
006 load 'IMQ2013_zonecampionarie';
007 fprintf('File MAT di riferimento: IMQ2013_zonecampionarie.mat\n\n');
008
009 %Ottenimento dei dati di ingresso
010 zona = input('Inserire codice zona: ');
011 modo = input('Inserire il modo (Pr-privato, Pu-pubblico, T-totali): ');
012 per = input('Periodo (IG-intera giornata, OP-ora di punta): ');
013 if per == 'IG';
```

```

014   scopo = input('Scopo (t-tutti, sr-tutti escluso ritorno): ');
015   else;
016   scopo = 't';
017   end;
018
019   %Determinazione dei flussi totali in uscita e in entrata
020   if modo=='T';
021       nomeMATR1 = strcat('ResPIEmPu',per,scopo);
022       nomeMATR2 = strcat('ResPIEmPr',per,scopo);
023       if per=='IG',scopo=='sr';
024           [FProd,FAttr] = flussototale(ResPIEmPrIGsr,zoneA,zona)+
025                                   flussototale(ResPIEmPuIGsr,zoneA,zona);
026           stampafileT(ResPIEmPrIGsr,ResPIEmPuIGsr,zoneA,zoneA,coordX,coordY);
027       end; if per=='IG',scopo=='t';
028           [FProd,FAttr] = flussototale(ResPIEmPrIGt,zoneA,zoneA)+
029                                   flussototale(ResPIEmPuIGt,zoneA,zoneA);
030           stampafileT(ResPIEmPrIGt,ResPIEmPuIGt,zoneA,zoneA,coordX,coordY);
031       end; if per=='OP',scopo=='t';
032           [FProd,FAttr] = flussototale(ResPIEmPrOPT,zoneA,zoneA)+
033                                   flussototale(ResPIEmPuOPT,zoneA,zoneA);
034           stampafileT(ResPIEmPrOPT,ResPIEmPuOPT,zoneA,zoneA,coordX,coordY);
035       end;
036       fprintf('\nMatrici di riferimento: ');fprintf(nomeMATR1);
037                                   fprintf(' / ');fprintf(nomeMATR2);
038   else;
039       nomeMATR = strcat('ResPIEm',modo,per,scopo);
040       if modo=='Pu',per=='IG',scopo=='sr';
041           [FProd,FAttr] = flussototale(ResPIEmPuIGsr,zoneA,zoneA);
042           stampafile(ResPIEmPuIGsr,zoneA,zoneA,coordX,coordY);
043       end; if modo=='Pu',per=='IG',scopo=='t';
044           [FProd,FAttr] = flussototale(ResPIEmPuIGt,zoneA,zoneA);
045           stampafile(ResPIEmPuIGt,zoneA,zoneA,coordX,coordY);
046       end; if modo=='Pu',per=='OP',scopo=='t';
047           [FProd,FAttr] = flussototale(ResPIEmPuOPT,zoneA,zoneA);
048           stampafile(ResPIEmPuOPT,zoneA,zoneA,coordX,coordY);
049       end; if modo=='Pr',per=='IG',scopo=='sr';
050           [FProd,FAttr] = flussototale(ResPIEmPrIGsr,zoneA,zoneA);
051           stampafile(ResPIEmPrIGsr,zoneA,zoneA,coordX,coordY);
052       end; if modo=='Pr',per=='IG',scopo=='t';
053           [FProd,FAttr] = flussototale(ResPIEmPrIGt,zoneA,zoneA);
054           stampafile(ResPIEmPrIGt,zoneA,zoneA,coordX,coordY);
055       end; if modo=='Pr',per=='OP',scopo=='t';
056           [FProd,FAttr] = flussototale(ResPIEmPrOPT,zoneA,zoneA);
057           stampafile(ResPIEmPrOPT,zoneA,zoneA,coordX,coordY);
058       end;
059       fprintf('\nMatrice di riferimento: ');fprintf(nomeMATR);
060   end;
061
062   %Stampa a video dei risultati
063   fprintf('\nSPOSTAMENTI TOTALI IN USCITA: ');fprintf('%0.12f',FProd);

```

```

064 fprintf('\nSPOSTAMENTI TOTALI IN ENTRATA:  ');fprintf('%.12f',FAttr);
065 fprintf('\n\nGenerati i file:  SpostProd.txt / SpostAttr.txt\n');
066
067 %Definizione della funzione di somma degli spostamenti
068 function [FP,FA] = flussototale(matrice,zone,Z);
069 FP = sum(matrice(find(zone==Z),:));
070 FA = sum(matrice(:,find(zone==Z)));
071 end
072
073 %Definizione delle funzioni di stampa su file
074 function stampafile(matrice,Z,zone,CX,CY);
075 filePROD = fopen('SpostProd.txt','w');
076 fileATTR = fopen('SpostAttr.txt','w');
077 zone(209)=[]; matrice(209,:)=[];matrice(:,209)=[];
078 SP = matrice(find(zone==Z),:);
079 stampaP = [zone SP' CX CY]; stampaP(find(zone==Z),:)=[];
080 stampaP=stampaP';
081 fprintf(filePROD,'%s %f %f %f\r\n',stampaP);
082 SA = matrice(:,find(zone==Z));
083 stampaA = [zone SA CX CY]; stampaA(find(zone==Z))=[]; stampaA=stampaA';
084 fprintf(fileATTR,'%s %f %f %f\r\n',stampaA);
085 fclose('all');
086 end
087
088 function stampafileT(matricePR,matricePU,Z,zone,CX,CY);
089 filePROD = fopen('SpostProd.txt','w');
090 fileATTR = fopen('SpostAttr.txt','w');
091 zone(209)=[];
092 matricePR(209,:)=[];matricePR(:,209)=[];
093 matricePU(209,:)=[];matricePU(:,209)=[];
094 SP = matricePU(find(zone==Z),:)+matricePR(find(zone==Z),:);
095 stampaP = [zone SP' CX CY]; stampaP(find(zone==Z),:)=[];
096 stampaP=stampaP';
097 fprintf(filePROD,'%s %f %f %f\r\n',stampaP);
098 SA = matricePU(:,find(zone==Z))+matricePR(:,find(zone==Z));
099 stampaA = [zone SA CX CY]; stampaA(find(zone==Z))=[]; stampaA=stampaA';
100 fprintf(fileATTR,'%s %f %f %f\r\n',stampaA);
101 fclose('all');
102 end

```

L'*output* prodotto a video in MATLAB è il seguente.

File MAT di riferimento: IMQ2013\_zonecampionarie.mat

Inserire codice zona: 'Q010'

Inserire il modo (Pr-privato, Pu-pubblico, T-totali): 'Pr'

Periodo (IG-intera giornata, OP-ora di punta): 'OP'



Matrice di riferimento: ResPIEmPrOPt  
 SPOSTAMENTI TOTALI IN USCITA: 4787.910000000001  
 SPOSTAMENTI TOTALI IN ENTRATA: 6196.400000000002

Generati i file: SpostProd.txt / SpostAttr.txt

Come si può osservare i risultati ottenuti sono gli stessi del Paragrafo 5.2 Determinazione del flusso totale prodotto/attratto da una zona. I due file di testo generati possono essere aperti, come i corrispondenti ottenuti in Python, in QGIS, importati come vettori di punti, e personalizzati nella visualizzazione analogamente a quanto già visto. Un estratto del file “*SpostProd.txt*” è riportato di seguito.

```
Q001 301.240000 396178.495000 4991495.208500
Q002 100.780000 395977.856200 4989356.847800
Q003 197.550000 394944.209700 4990130.576900
Q004 0.000000 393210.203200 4990461.359600
Q005 98.780000 393840.657400 4991550.947300
Q006 0.000000 394598.761000 4993107.508100
Q007 0.000000 396701.592900 4992847.568600
Q008 0.000000 398583.684100 4992179.174000
[...]
```

Si noti che in questo caso non è presente la prima riga di intestazione e che i campi sono delimitati da spazi anziché da “;”. Queste differenze non rappresentano un grosso problema perché dalla finestra del comando “*Aggiungi layer testo delimitato...*” in QGIS è possibile personalizzare l’importazione dei dati a seconda dello specifico caso.

### **5.6.3 Estrazione di parte della matrice**

Come eseguito nel Paragrafo 5.3 Estrazione di parte di matrice, con il seguente *script*, dal nome ‘*EstrazioneParteMatrice.m*’, è possibile separare dei dati di interesse dal resto delle informazioni contenute in una matrice O/D. Nello specifico esempio si erano estratti i flussi registrati tra la zona della Val di Susa e il centro della città di Torino. L’elenco delle zone da considerare nella nuova matrice era stato ricavato attraverso il software QGIS,

salvando nel file “*ESTRAZIONE.csv*” la selezione delle aree. Di seguito è dunque riportato il listato che permette queste operazioni.

```

01 clear all;
02 close all;
03 clc;
04
05 %Caricamento delle matrici O/D e della mappatura
06 load 'IMQ2013_zonecampionarie';
07 fprintf('File MAT di riferimento:  IMQ2013_zonecampionarie.mat\n');
08
09 %Ottenimento del nuovo elenco delle zone
10 [empty,NEWzone] = xlsread('ESTRAZIONE.csv','A2:A210');
11 NEWzone = string(NEWzone);
12
13 %Estrazione delle nuove matrici
14 [ESTR_ResPIEmPrIGt] = estrazione(ResPIEmPrIGt,NEWzone,zoneA);
15 [ESTR_ResPIEmPrIGsr] = estrazione(ResPIEmPrIGsr,NEWzone,zoneA);
16 [ESTR_ResTOmPrIGt] = estrazione(ResTOmPrIGt,NEWzone,zoneA);
17 [ESTR_ResTOmPrIGsr] = estrazione(ResTOmPrIGsr,NEWzone,zoneA);
18 [ESTR_ResPIEmPrOPt] = estrazione(ResPIEmPrOPt,NEWzone,zoneA);
19 [ESTR_ResTOmPrOPt] = estrazione(ResTOmPrOPt,NEWzone,zoneA);
20 [ESTR_ResPIEmPuIGt] = estrazione(ResPIEmPuIGt,NEWzone,zoneA);
21 [ESTR_ResPIEmPuIGsr] = estrazione(ResPIEmPuIGsr,NEWzone,zoneA);
22 [ESTR_ResTOmPuIGt] = estrazione(ResTOmPuIGt,NEWzone,zoneA);
23 [ESTR_ResTOmPuIGsr] = estrazione(ResTOmPuIGsr,NEWzone,zoneA);
24 [ESTR_ResPIEmPuOPt] = estrazione(ResPIEmPuOPt,NEWzone,zoneA);
25 [ESTR_ResTOmPuOPt] = estrazione(ResTOmPuOPt,NEWzone,zoneA);
26
27 %Salvataggio delle nuove matrici
28 save IMQ2013_estrazione NEWzone ESTR_ResPIEmPrIGt ESTR_ResPIEmPrIGsr
29     ESTR_ResTOmPrIGt ESTR_ResTOmPrIGsr ESTR_ResPIEmPrOPt ESTR_ResTOmPrOPt
30     ESTR_ResPIEmPuIGt ESTR_ResPIEmPuIGsr ESTR_ResTOmPuIGt ESTR_ResTOmPuIGsr
31     ESTR_ResPIEmPuOPt ESTR_ResTOmPuOPt;
32 fprintf('FILE MAT CREATO:  IMQ2013_estrazione.mat\n');
33
34 %Definizione della funzione di estrazione della matrice
35 function [newMATR] = estrazione(matrORIG,Nzone,Ozone);
36 for i=1:length(Nzone);
37     for j=1:length(Nzone);
38         newMATR(i,j) = matrORIG(find(Ozone==Nzone(i)),find(Ozone==Nzone(j)));
39     end;
40 end;
41 end

```

La FIGURA 5.6.2 riporta, a titolo di esempio, una delle nuove matrici create con il

precedente *script* e salvate nel file “*IMQ2013\_estrazione.mat*”. Si noti che si è scelto di riportare la stessa matrice mostrata nella FIGURA 5.3.2 in modo da poter confrontare i risultati ottenuti con i due diversi programmi. I flussi riportati nelle due immagini, infatti, sono coincidenti. L’*output* prodotto a video, invece, è riportato di seguito.

File MAT di riferimento: *IMQ2013\_zoncampionarie.mat*

FILE MAT CREATO: *IMQ2013\_estrazione.mat*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2.3290e+04	93.0800	3.8412e+03	193.6900	189.3400	2.6010e+03	3.7486e+03	3.0881e+03	0	0	0	129.9800	97.2600	0	188.3000	0	717.2
2	198.6100	2.8128e+04	95	0	1.9580e+03	0	0	93.0800	768.5400	93.0800	197.9900	490.0100	265.3200	33	629.8200	160.7300	757.9
3	3.6757e+03	95	1.0113e+04	0	0	96.5000	198.5200	204.0200	0	0	0	0	0	0	103.3900	0	460.7
4	193.6900	0	0	5.2712e+03	284.4900	0	0	0	485.5900	105.1300	0	850.9200	389.6300	291.6500	197.8800	943.9200	0
5	351.2000	1.9580e+03	0	382.3100	1.6123e+04	0	0	0	4.0446e+03	390.1800	105	459.8200	157.5800	0	576.2300	165.6600	129.1
6	2.6037e+03	0	96.5000	0	0	1.5364e+04	1.5588e+03	390.9400	0	0	0	0	0	0	0	0	0
7	3.6482e+03	0	198.5200	0	0	1.6625e+03	6.1161e+03	485.4400	0	0	0	0	0	0	0	0	0
8	3.2840e+03	93.0800	204.0200	0	0	390.9400	485.4400	1.5158e+04	0	0	0	0	0	0	93.6600	0	160.1
9	0	768.5400	0	387.7700	4.2432e+03	0	0	0	2.9735e+04	3.6464e+03	506.5700	102.5000	330.2700	31.3600	100.8100	402.6000	32.2
10	0	93.0800	0	105.1300	488.2800	0	0	0	3.4497e+03	1.0682e+04	4.0305e+03	31.6600	0	0	0	33.6000	0
11	0	197.9900	0	0	204.4700	0	0	0	506.5700	4.0305e+03	1.0391e+04	0	67.0700	0	0	0	0
12	129.9800	490.0100	0	850.9200	459.8200	0	0	0	102.5000	31.6600	0	5.8429e+03	642.1300	134.5800	258.9600	770.7300	32.3
13	0	265.3200	0	494.7600	254.8500	0	0	0	229.4600	0	67.0700	577.7600	1.4689e+04	257.2200	2.4109e+03	573.1100	32.2
14	0	33	0	291.6500	0	0	0	0	31.3600	0	0	134.5800	289.8600	3.1201e+03	65.1300	433.9400	0
15	188.3000	735.6700	103.3900	197.8800	576.2300	0	0	93.6600	100.8100	0	0	258.9600	2.3149e+03	32.3300	1.3417e+04	404.8200	298.3
16	0	160.7300	0	943.9200	165.6600	0	0	0	402.6000	33.6000	0	770.7300	640.3200	435.0100	406.0500	6.2693e+03	0
17	749.5900	757.9400	460.7300	0	129.1700	0	0	160.1400	32.2300	0	0	32.3900	32.2300	0	298.3200	0	9.7652e+
18	130.9600	1.9339e+03	32.2000	438.9500	2.1802e+03	100.5800	0	0	780.6600	130.9600	64.0200	1.7196e+03	2.8638e+03	98.1200	1.7452e+03	1.0936e+03	159.7
19	291.6000	1.0788e+03	229.2300	0	32.0400	94.0300	0	0	0	0	0	63.7000	192.9700	0	462.9500	134.6100	687.9
20	91.3300	0	0	0	97.6200	0	0	0	100.8200	0	0	99.9000	560.2000	67.5200	432.2600	168.8200	164.5
21	391.9300	1365	215.8200	823.3900	392.8900	190.3800	294.9100	299.7500	403.6100	100.9900	505.9200	327.6100	1.5003e+03	230.7700	1.1160e+03	237.6000	428.5
22	182.3100	97.8500	100.3800	293.7600	186.9700	0	0	0	96.1700	0	0	0	483.4900	33	1.4972e+03	296.8900	96.8
23	91.1500	192.6500	321.3500	95.8800	188.3500	0	0	0	193.4800	0	99.4700	96.0300	623.3500	131.4100	574.6700	129.7700	258.8

FIGURA 5.6.2 Apertura di una matrice salvata nel file “*IMQ2013\_estrazione.mat*” nell’ambiente di esplorazione delle variabili di MATLAB.

### 5.6.4 Aggregazione della matrice

Nel corrispondente esempio riportato nel Paragrafo 5.4 Aggregazione di una matrice in base agli attributi, si erano volute aggregare le matrici IMQ sulla base di un campo della tabella attributi delle zone campionarie in QGIS. Nello specifico, si era scelto di compattare le matrici O/D sulla base della provincia di appartenenza delle singole zone campionarie. Per fare ciò, la tabella attributi del relativo *shapefile* era stata esportata interamente in un file dal nome “*AGGREGAZIONE.csv*”. Lo *script* che permette di effettuare le stesse operazioni, ma nell’ambiente di MATLAB, è chiamato “*AggregazioneMatrice.m*”, ed è riportato di

seguito.

```

01 clear all;
02 close all;
03 clc;
04
05 %Caricamento delle matrici O/D e della mappatura
06 load 'IMQ2013_zonecampionarie';
07 fprintf('File MAT di riferimento:  IMQ2013_zonecampionarie.mat\n');
08
09 %Ottenimento delle informazioni sul criterio di aggregazione
10 [empty,file] = xlsread('AGGREGAZIONE.csv','A2:C210');
11 zoneORD = string(file(:,1)); zoneORD = [zoneORD;'Esterno'];
12 PROV = string(file(:,2));
13 MACR = string(file(:,3));
14 CRIT = input('\nInserire criterio di aggregazione (1-per province,
15                                                    2-per macro-aree): ');
16
17 %Aggregazione delle matrici
18 if CRIT==1;
19     NEWzone = unique(PROV);
20     PROV = [PROV;'EXT']; NEWzone = [NEWzone;'EXT'];
21     [AGGR_ResPIEmPrIGt] =
22         aggregazione(ResPIEmPrIGt,NEWzone,PROV,zoneORD,zoneA);
23     [AGGR_ResPIEmPrIGsr] =
24         aggregazione(ResPIEmPrIGsr,NEWzone,PROV,zoneORD,zoneA);
25     [AGGR_RestOmPrIGt] =
26         aggregazione(RestOmPrIGt,NEWzone,PROV,zoneORD,zoneA);
27     [AGGR_RestOmPrIGsr] =
28         aggregazione(RestOmPrIGsr,NEWzone,PROV,zoneORD,zoneA);
29     [AGGR_ResPIEmPrOPT] =
30         aggregazione(ResPIEmPrOPT,NEWzone,PROV,zoneORD,zoneA);
31     [AGGR_RestOmPrOPT] =
32         aggregazione(RestOmPrOPT,NEWzone,PROV,zoneORD,zoneA);
33     [AGGR_ResPIEmPuIGt] =
34         aggregazione(ResPIEmPuIGt,NEWzone,PROV,zoneORD,zoneA);
35     [AGGR_ResPIEmPuIGsr] =
36         aggregazione(ResPIEmPuIGsr,NEWzone,PROV,zoneORD,zoneA);
37     [AGGR_RestOmPuIGt] =
38         aggregazione(RestOmPuIGt,NEWzone,PROV,zoneORD,zoneA);
39     [AGGR_RestOmPuIGsr] =
40         aggregazione(RestOmPuIGsr,NEWzone,PROV,zoneORD,zoneA);
41     [AGGR_ResPIEmPuOPT] =
42         aggregazione(ResPIEmPuOPT,NEWzone,PROV,zoneORD,zoneA);
43     [AGGR_RestOmPuOPT] =
44         aggregazione(RestOmPuOPT,NEWzone,PROV,zoneORD,zoneA);
45 else if CRIT==2;
46     NEWzone = unique(MACR);
47     MACR = [MACR;'EXT']; NEWzone = [NEWzone;'EXT'];

```

```

48     [AGGR_ResPIEmPrIGt] =
49         aggregazione(ResPIEmPrIGt,NEWzone,MACR,zoneORD,zoneA);
50     [AGGR_ResPIEmPrIGsr] =
51         aggregazione(ResPIEmPrIGsr,NEWzone,MACR,zoneORD,zoneA);
52     [AGGR_ResTOMPrIGt] =
53         aggregazione(ResTOMPrIGt,NEWzone,MACR,zoneORD,zoneA);
54     [AGGR_ResTOMPrIGsr] =
55         aggregazione(ResTOMPrIGsr,NEWzone,MACR,zoneORD,zoneA);
56     [AGGR_ResPIEmPrOPt] =
57         aggregazione(ResPIEmPrOPt,NEWzone,MACR,zoneORD,zoneA);
58     [AGGR_ResTOMPrOPt] =
59         aggregazione(ResTOMPrOPt,NEWzone,MACR,zoneORD,zoneA);
60     [AGGR_ResPIEmPuIGt] =
61         aggregazione(ResPIEmPuIGt,NEWzone,MACR,zoneORD,zoneA);
62     [AGGR_ResPIEmPuIGsr] =
63         aggregazione(ResPIEmPuIGsr,NEWzone,MACR,zoneORD,zoneA);
64     [AGGR_ResTOMPuIGt] =
65         aggregazione(ResTOMPuIGt,NEWzone,MACR,zoneORD,zoneA);
66     [AGGR_ResTOMPuIGsr] =
67         aggregazione(ResTOMPuIGsr,NEWzone,MACR,zoneORD,zoneA);
68     [AGGR_ResPIEmPuOPt] =
69         aggregazione(ResPIEmPuOPt,NEWzone,MACR,zoneORD,zoneA);
70     [AGGR_ResTOMPuOPt] =
71         aggregazione(ResTOMPuOPt,NEWzone,MACR,zoneORD,zoneA);
72     end;
73 end;
74
75 %Salvataggio delle nuove matrici
76 save IMQ2013_estrazione NEWzone AGGR_ResPIEmPrIGt AGGR_ResPIEmPrIGsr
77     AGGR_ResTOMPrIGt AGGR_ResTOMPrIGsr AGGR_ResPIEmPrOPt AGGR_ResTOMPrOPt
78     AGGR_ResPIEmPuIGt AGGR_ResPIEmPuIGsr AGGR_ResTOMPuIGt AGGR_ResTOMPuIGsr
79         AGGR_ResPIEmPuOPt AGGR_ResTOMPuOPt;
80 fprintf('\nFILE MAT CREATO:  IMQ2013_aggregazione.mat\n');
81
82 %Definizione della funzione di aggregazione
83 function [newMATR] = aggregazione(matrORIG,Nzone,criterio,zORDIN,zORIG);
84 newMATR = zeros(length(Nzone),length(Nzone));
85 for i=1:209;
86     index0 = find(zORDIN==zORIG(i));
87     i0 = find(Nzone==criterio(index0));
88     for j=1:209;
89         indexD = find(zORDIN==zORIG(j));
90         iD = find(Nzone==criterio(indexD));
91         newMATR(i0,iD) = newMATR(i0,iD)+matrORIG(i,j);
92     end;
93 end;
94 end

```

L'*output* prodotto a video da questo listato è riportato di seguito.

File MAT di riferimento: IMQ2013\_zonecampionarie.mat

Inserire criterio di aggregazione (1-per province, 2-per macro-aree): 1

FILE MAT CREATO: IMQ2013\_aggregazione.mat

La FIGURA 5.6.3 riporta una delle nuove matrici, salvate nel file “*IMQ2013\_aggregazione.mat*”. Si noti che, come nei precedenti casi, si è scelto di riportare la stessa matrice mostrata nella FIGURA 5.4.2 in modo da poter confrontare i risultati ottenuti con i due diversi programmi. Come ci si poteva aspettare, anche in questo caso, i flussi riportati nelle due immagini sono gli stessi, anche se l’ordine del salvataggio delle nuove zone è differente. Nella FIGURA 5.6.4 è riportato il vettore contenente i nomi identificativi delle nuove zone.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1.1500e+04	96.8200	0	93.8300	0	1.4199e+03	0	368.6100	5.0892e+03	0	0	0	0	0
2	429.8000	8.3144e+03	0	501.1800	0	4.2174e+03	0	0	0	0	0	0	0	0
3	0	0	1.7468e+03	0	91.7500	888.9700	0	294.7800	599.0200	0	0	0	0	0
4	102.2600	194.9300	0	2.6403e+04	0	7.9679e+03	0	0	387.9600	0	0	0	0	0
5	195.9500	0	0	0	1.2655e+04	565.0300	0	521.7900	5.7878e+03	0	0	0	0	0
6	301.9600	65.2000	0	1.4517e+03	434.6800	3.3880e+05	0	859.5000	2.2888e+03	0	0	0	0	0
7	0	0	0	0	292.7300	0	4.4340e+03	0	1.4682e+03	0	0	0	0	0
8	275.2100	0	604.4100	0	1.5518e+03	2.0398e+03	0	3.3352e+03	853.1800	0	0	0	0	0
9	213.9400	0	0	0	0	0	0	0	191.1500	0	0	0	0	0
10														
11														
12														
13														

FIGURA 5.6.3 Apertura di una matrice salvata nel file “*IMQ2013\_aggregazione.mat*” nell’ambiente di esplorazione delle variabili di MATLAB.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	AL													
2	AT													
3	BI													
4	CN													
5	NO													
6	TO													
7	VB													
8	VC													
9	EXT													
10														
11														

FIGURA 5.6.4 Apertura del vettore della zonizzazione, salvato nel file “*IMQ2013\_aggregazione.mat*”, nell’ambiente di esplorazione delle variabili di MATLAB.

### 5.6.5 Ottenimento informazioni di sintesi

Nell’ultimo esempio, riportato nel Paragrafo 5.5 Ottenimento di informazioni di sintesi, si era scelto di determinare la percentuale di scelta del mezzo pubblico sugli spostamenti totali effettuati. Il listato di MATLAB che permette questo tipo di elaborazione, è stato chiamato “*SceltaMezzoPubblico.m*”, ed è riportato di seguito.

```

01 clear all;
02 close all;
03 clc;
04
05 %Caricamento delle matrici O/D e della mappatura
06 load 'IMQ2013_zonecampionarie';
07 fprintf('File MAT di riferimento:  IMQ2013_zonecampionarie.mat\n\n');
08
09 %Creazione del nuvo file con le nuove matrici
10 SPUB_ResPIEx100mPuIGt = ResPIEmPuIGt./(ResPIEmPuIGt+ResPIEmPrIGt)*100;
11 SPUB_ResPIEx100mPuIGsr =ResPIEmPuIGsr./(ResPIEmPuIGsr+ResPIEmPrIGsr)*100;
13 SPUB_ResTOx100mPuIGt = RestOmPuIGt./(RestOmPuIGt+RestOmPrIGt)*100;
14 SPUB_ResTOx100mPuIGsr = RestOmPuIGsr./(RestOmPuIGsr+RestOmPrIGsr)*100;
15 SPUB_ResPIEx100mPuOPT = ResPIEmPuOPT./(ResPIEmPuOPT+ResPIEmPrOPT)*100;
16 SPUB_ResTOx100mPuOPT = RestOmPuOPT./(RestOmPuOPT+RestOmPrOPT)*100;
17 save IMQ2013_scltamezzopubblico zoneA coordX coordY
18     SPUB_ResPIEx100mPuIGt SPUB_ResPIEx100mPuIGsr SPUB_ResTOx100mPuIGt
19     SPUB_ResTOx100mPuIGsr SPUB_ResPIEx100mPuOPT SPUB_ResTOx100mPuOPT;
20 fprintf('FILE MAT CREATO:  IMQ2013_scltamezzopubblico.mat\n');
21
22 %Scrittura dati sul file di testo

```

```

23 for i=1:208;
24     SPRODpubb(i) = sum(ResPIEmPuOPt(i,:));
25     SPRODpriv(i) = sum(ResPIEmPrOPt(i,:));
26 end
27 PERC = SPRODpubb./(SPRODpubb+SPRODpriv)*100;
28 file = fopen('SceltaMezzoPubblico.txt','w');
29 zoneA(209)=[];
30 stampa = [zoneA PERC' coordX coordY]; stampa=stampa';
31 fprintf(file,'%s %f %f %f\r\n',stampa);
32 fclose('all');
33 fprintf('FILE TXT CREATO: SceltaMezzoPubblico.txt\n');

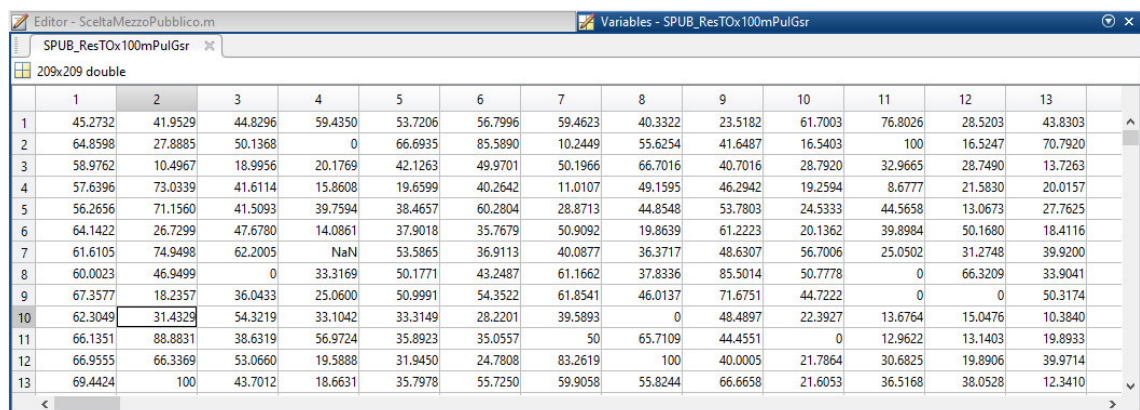
```

Di seguito è riportato anche l'*output* prodotto a video e, a titolo di esempio, una matrice di quelle appena prodotte. Si noti che, per corrispondenza con il relativo esempio in Python, la FIGURA 5.6.5 riporta la stessa matrice rappresentata nella precedente FIGURA 5.5.1.

File MAT di riferimento: IMQ2013\_zonecampionarie.mat

FILE MAT CREATO: IMQ2013\_scltamezzopubblico.mat

FILE TXT CREATO: SceltaMezzoPubblico.txt



	1	2	3	4	5	6	7	8	9	10	11	12	13
1	45.2732	41.9529	44.8296	59.4350	53.7206	56.7996	59.4623	40.3322	23.5182	61.7003	76.8026	28.5203	43.8303
2	64.8598	27.8885	50.1368	0	66.6935	85.5890	10.2449	55.6254	41.6487	16.5403	100	16.5247	70.7920
3	58.9762	10.4967	18.9956	20.1769	42.1263	49.9701	50.1966	66.7016	40.7016	28.7920	32.9665	28.7490	13.7263
4	57.6396	73.0339	41.6114	15.8608	19.6599	40.2642	11.0107	49.1595	46.2942	19.2594	8.6777	21.5830	20.0157
5	56.2656	71.1560	41.5093	39.7594	38.4657	60.2804	28.8713	44.8548	53.7803	24.5333	44.5658	13.0673	27.7625
6	64.1422	26.7299	47.6780	14.0861	37.9018	35.7679	50.9092	19.8639	61.2223	20.1362	39.8984	50.1680	18.4116
7	61.6105	74.9498	62.2005	NaN	53.5865	36.9113	40.0877	36.3717	48.6307	56.7006	25.0502	31.2748	39.9200
8	60.0023	46.9499	0	33.3169	50.1771	43.2487	61.1662	37.8336	85.5014	50.7778	0	66.3209	33.9041
9	67.3577	18.2357	36.0433	25.0600	50.9991	54.3522	61.8541	46.0137	71.6751	44.7222	0	0	50.3174
10	62.3049	31.4329	54.3219	33.1042	33.3149	28.2201	39.5893	0	48.4897	22.3927	13.6764	15.0476	10.3840
11	66.1351	88.8831	38.6319	56.9724	35.8923	35.0557	50	65.7109	44.4551	0	12.9622	13.1403	19.8933
12	66.9555	66.3369	53.0660	19.5888	31.9450	24.7808	83.2619	100	40.0005	21.7864	30.6825	19.8906	39.9714
13	69.4424	100	43.7012	18.6631	35.7978	55.7250	59.9058	55.8244	66.6658	21.6053	36.5168	38.0528	12.3410

FIGURA 5.6.5 Apertura di una matrice salvata nel file “*IMQ2013\_scltamezzopubblico.mat*” nell’ambiente di esplorazione delle variabili di MATLAB.

Si è creato anche qui un file di testo, dal nome “*SceltaMezzoPubblico.txt*”, contenente per ogni zona la percentuale di scelta del mezzo pubblico sugli spostamenti totali effettuati in uscita dalla zona stessa nell’ora di punta per qualunque scopo. Questo, di cui se ne riporta



un estratto di seguito, può essere aperto come il corrispondente del Paragrafo 5.5 Ottenimento di informazioni di sintesi, in QGIS per ottenere una mappa personalizzata della zona di studio sulla base di questa informazione estratta.

```
Q001 28.251000 396178.495000 4991495.208500
Q002 36.713500 395977.856200 4989356.847800
Q003 39.750500 394944.209700 4990130.576900
Q004 26.562800 393210.203200 4990461.359600
Q005 44.039100 393840.657400 4991550.947300
Q006 43.365900 394598.761000 4993107.508100
Q007 42.741700 396701.592900 4992847.568600
Q008 26.829600 398583.684100 4992179.174000
[...]
```

### ***5.6.6 Commenti sui risultati del confronto***

Confrontando in generale i due formati di salvataggio, OMX e MAT (formato predefinito di MATLAB), si possono notare alcune grosse differenze; nonostante siano entrambi formati per la memorizzazione di dati matriciali. OMX è studiato appositamente per matrici di trasporto, e quindi è pensato per poter accogliere i metadati associati e la mappatura della zonizzazione. Anche se una grossa limitazione è data dal fatto che nella “mappa dell’indice” non è possibile memorizzare dati che non siano interi. Come è accaduto nel Paragrafo 5.4 Aggregazione di una matrice in base agli attributi, non potendo associare in modo logico dei numeri alle diverse province, si è dovuto ricorrere a un file di testo per tenere traccia della conversione indice-provincia.

Il MAT, invece, non essendo un formato specifico, consente il salvataggio di qualunque tipo di variabile definita nell’ambiente di calcolo di MATLAB. Grazie a questa sua caratteristica, si è potuto sempre associare la mappatura, e non è risultato necessario ogni volta accedere al file di Excel (si veda la nota 75 a pagina 99) contenente la conversione codice numerico – codice alfanumerico. Inoltre, non imponendo nessuna restrizione sulla dimensione degli elementi salvati, con il MAT, volendo, si sarebbero potute

salvare in un unico file sia le matrici O/D per zone campionarie, sia quelle redatte per macro-aree. L'OMX impone invece che le matrici contenute abbiano tutte la stessa dimensione.

Infine, con il formato di MATLAB, si è potuto memorizzare nel file anche le informazioni di geolocalizzazione dei centroidi (le due coordinate X e Y), in modo da avere sempre a disposizione i dati per creare dei file TXT da utilizzare in QGIS per un *output* di più immediata comprensione. Una grossa limitazione è invece l'impossibilità di accedere ai dati salvati senza una valida licenza del software MATLAB. Infatti, questo formato è leggibile **solo** attraverso questo programma.

Per quanto riguarda il linguaggio di per sé, si può affermare che in Python si ha il vantaggio di poter “ricostruire” i nomi delle variabili da richiamare come sequenze di stringhe. In questo modo si è potuto evitare di compiere la stessa operazione per tutte le matrici salvate, o per tutti i file da aprire.

La TABELLA 5.6.1, invece, riporta lo spazio della memoria di massa occupato dai diversi file creati nel Capitolo 5, nei due formati analizzati. Come si può osservare, il formato MAT occupa molta meno memoria del corrispondente OMX. Questo perché, sostanzialmente, quest'ultimo formato è più strutturato e inoltre accoglie più dati del primo. Infatti, oltre alle matrici e alla mappatura, comuni in entrambi i casi, nel formato Open Matrix sono presenti anche un certo numero di metadati. Il risparmio della memoria, però, diminuisce di entità man mano che le dimensioni, e la quantità, delle matrici salvate aumentano. Dunque, più sono “grandi” i file da memorizzare, meno vantaggio si avrebbe nell'utilizzare il formato MAT. Al contrario, per quanto riguarda la memoria centrale (o RAM<sup>80</sup>), le due applicazioni utilizzate per l'accesso ai dati, Wingware nel caso delle matrici

---

<sup>80</sup> La memoria RAM (*Random Access Memory*), chiamata anche memoria centrale, primaria o principale, accoglie i dati in fase di elaborazione. Possiede una grande velocità di accesso ed è definita “volatile” poiché, al  
(segue)

OMX e MATLAB nel caso delle matrici MAT, hanno occupato rispettivamente circa 100 MB e 550 MB. Nel primo caso, il caricamento delle matrici OMX nella memoria della macchina ha comportato un aumento compreso tra il 20-30% (nello specifico i file “*IMQ2013\_scltamezzopubblico.omx*” e “*IMQ2013\_estrazione.omx*” circa 30 MB, “*IMQ2013\_zonecampionarie.omx*” e “*IMQ2013\_aggregazione.omx*” circa 25 MB e infine “*IMQ2013\_macroaree.omx*” circa 20 MB). Mentre nel caso delle matrici MAT il caricamento dei dati non ha prodotto un significativo aumento della memoria occupata, di per sé già elevata.

Riguardo invece l’utilizzo del calcolatore e i tempi di esecuzione dei programmi si veda la TABELLA 5.6.2. Si ricorda che le caratteristiche tecniche della macchina sono: Notebook HP, processore Intel® Core™ i5-6200U CPU @ 2.30-2.40 GHz; RAM installata 8.00 GB; sistema operativo a 64 bit, processore basato su x64. Come si può osservare i listati MATLAB sono leggermente “meno impegnativi” da eseguire per la macchina: richiedono, infatti, meno tempo e meno carico di lavoro. In particolare, nella prima riga della TABELLA 5.6.2 sono riportati i dati relativi allo *script* di Python che permette l’elaborazione delle matrici OMX per macro-aree; nella seconda, invece, il corrispondente listato MATLAB. Nella terza e nella quarta riga, infine, i dati riguardanti rispettivamente lo *script* Python e il listato MATLAB redatti per la creazione delle matrici per zone campionarie. Come era presumibile, le differenze maggiori si apprezzano nel caso in cui le matrici hanno dimensioni maggiori.

---

momento dello spegnimento della macchina, ogni informazione immagazzinata o processo in corso viene perso. Si contrappone alla memoria di massa, chiamata anche secondaria o ausiliaria, la quale invece permette la memorizzazione di grandi quantità di dati in maniera permanente (o almeno fino al momento della loro eliminazione da parte dell’utente).

TABELLA 5.6.1 Dimensioni su disco (memoria di massa) di alcuni file nel formato OMX e MAT.

NOME FILE	NUMERO MATRICI	DIMENSIONI MATRICI	OMX	MAT	
<i>IMQ2013_macroaree</i>	12	5x5	50 KB	3 KB	- 94 %
<i>IMQ2013_zonecampionarie</i>	12	209x209	493 KB	214 KB	- 57 %
<i>IMQ2013_scltamezzopubblico</i>	6	209x209	169 KB	113 KB	- 33 %
<i>IMQ2013_estrazione</i>	12	42x42	136 KB	45 KB	- 67 %
<i>IMQ2013_aggregazione</i>	12	9x9	53 KB	4 KB	- 92 %

TABELLA 5.6.2 Percentuale massima di utilizzo della macchina e tempi di esecuzione di alcuni *script*.

NOME FILE	% MASSIMA DI UTILIZZO DELLA MACCHINA	TEMPO DI ESECUZIONE
<i>Macroaree.py</i>	20 %	5 s
<i>Macroaree.m</i> <sup>81</sup>	10 %	4 s
<i>ZoneCampionarie.py</i>	35 %	10 s
<i>ZoneCampionarie.m</i>	20 %	8 s

<sup>81</sup> Il listato MATLAB dal nome "*IMQ2013\_macroaree.m*" permette la compilazione delle matrici O/D, redatte per macro-aree, sulla base dei dati dell'indagine IMQ2013, ed è riportato nell'ALLEGATO F.

## 6. CONCLUSIONI

Lo scopo del presente lavoro era quello di provare ad utilizzare un modo alternativo all'archiviazione e gestione dei dati di tipo matriciale, derivanti dalle operazioni di analisi della domanda di mobilità nell'ambito della pianificazione dei trasporti. Questo perché i formati proprietari, adottati dagli strumenti di simulazione delle reti di trasporto non sono completamente soddisfacenti per due principali motivi: sono leggibili solo da chi possiede lo stesso software e una licenza valida; sono soggetti a problemi di obsolescenza da parte delle nuove versioni che spesso rendono illeggibili i file più datati. A tal scopo si è scelto di testare il formato Open Matrix, che si pone come formato standard nel settore, andando a colmare le lacune di quelli proposti da singoli strumenti di analisi commerciali. Il formato con cui si è posto a paragone è il MAT; formato proprietario del software MATLAB specifico per il calcolo matriciale.

Un primo aspetto da migliorare, che è emerso durante l'elaborazione delle matrici OMX, riguarda l'impossibilità di memorizzare, nel vettore contenente la "mappa dell'indice", dei dati di tipo stringa. Questa è risultata essere la più grossa limitazione riscontrata; a causa di ciò, infatti, nei file OMX si è perso un po' il riscontro diretto dei risultati. Le zone campionarie, disponevano di un codice identificativo alfanumerico (una lettera seguita da tre numeri), che è stato facilmente tramutato in codice numerico (si veda il Paragrafo 4.3.2 Matrici per zone campionarie); ma ciò non è stato possibile con altri casi visti. Ad esempio, quando si sono aggregate le matrici sulla base della provincia di appartenenza (si veda il Paragrafo 5.4 Aggregazione di una matrice in base agli attributi), all'apertura del file OMX non si disponeva di alcun riferimento utile all'identificazione

## CONCLUSIONI

dell'origine e della destinazione del flusso riportato in ogni cella. In questo modo le matrici salvate perdevano il loro significato. Come soluzione provvisoria, si era scelto di accompagnare al file OMX un file di testo contenente la mappatura. Ma questa soluzione non è ideale, in quanto tale informazione non sarebbe archiviata nel file OMX, che quindi non sarebbe più “autosufficiente”.

Nel caso delle zone campionarie, sebbene sia stato possibile creare una relazione logica tra codici numerici ed alfanumerici, la conversione ha fornito delle matrici O/D non relazionabili in maniera immediata con gli altri documenti redatti in occasione dell'indagine (ad esempio con il database originario).

Considerando invece la possibilità di accedere ai dati salvati, si può affermare che il formato OMX sia il migliore tra quelli visti. L'Agenzia della Mobilità Piemontese ha scelto, come formato di pubblicazione delle matrici O/D, l'XLS di Excel. In questo caso, però, si necessita di una licenza di Office attiva per poterne visualizzarne il contenuto. Anche se il problema può essere facilmente aggirato scaricando la versione *free* di Open Office Calc. Il MAT invece, testato nel Paragrafo 5.6 Confronto con MATLAB, è risultato essere ancora più “chiuso” di questo. Infatti, l'unica possibilità di accesso a questi file è attraverso il software MATLAB, che si ricorda avere una licenza non libera. Per accedere ai dati in OMX, invece, è possibile scaricare apertamente il visualizzatore OMX Viewer ed aprire qualsiasi documento con questo formato.

Riguardo lo spazio occupato nella memoria di massa si può dire che i file OMX rappresentino un grosso vantaggio rispetto agli analoghi file Excel. I primi occupano moltissimo spazio in meno rispetto ai corrispondenti XLS; questo nonostante la memoria “spreca” dai file OMX che, per struttura, dedicano uno spazio anche se tra due zone non avviene nessun flusso (nei file Excel invece queste informazioni sono state eliminate; non sono infatti presenti coppie origine-destinazione con flusso registrato pari a 0, ma semplicemente non compaiono nell'elenco). I file MAT sono risultati anch'essi molto

## CONCLUSIONI

piccoli. Anche se si è visto che il risparmio, in termini di spazio di memoria, risulta sempre più basso al crescere della dimensione dei file stessi (si veda il Paragrafo 5.6.6 Commenti). I file MAT, inoltre, non ponendo alcun tipo di vincolo sulle variabili memorizzate, possono accogliere in un unico file anche matrici di dimensioni differenti. Non solo; possono accogliere anche altri dati non direttamente legati agli spostamenti, come è successo negli esempi riportati nel Paragrafo 5.6 Confronto con MATLAB, dove si è scelto di inserire nei file di salvataggio MAT anche i vettori contenenti le coordinate geografiche dei centroidi delle zone campionarie. Concentrandosi invece sulla memoria RAM, per caricare le matrici OMX si occupano circa 125 MB (con delle differenze di più o meno 5 MB in base allo specifico file), a dispetto dei circa 550 MB che servono per i corrispondenti file MAT. Ciò significa che nel caso delle matrici OMX è possibile gestire più applicazioni prima di congestionare la memoria della macchina.

Per quanto riguarda invece gli *script*, l'API Python è risultato aderente alla sua presentazione. È un linguaggio abbastanza semplice che può essere padroneggiato, nelle sue funzioni basilari come quelle utilizzate per questo scopo, anche da autodidatti. In rete inoltre si trovano moltissimi *tutorial* che ne spiegano i primi rudimenti passo per passo. Dunque, non serve essere esperti informatici per comprenderne il funzionamento e mettere in pratica degli *script* del tipo di quelli riportati in questo lavoro. Riguardo i tempi di esecuzione dei programmi scritti si può dire che siano abbastanza brevi in tutti i casi considerati. L'esecuzione dei listati MATLAB è risultata leggermente più veloce di quella degli *script* di Python, ma comunque senza variare di troppo (si veda il Paragrafo 5.6.6 Commenti). L'esecuzione più lunga è quella che ha riguardato l'aggregazione delle matrici per macro-aree. In quel caso la macchina ha impiegato circa un minuto per portare a termine le operazioni richieste.

Si ricorda infine che, nell'esempio di impiego del formato OMX riportato nel Paragrafo 1.3 Applicazioni del formato OMX per il *database* del FAF, a questo formato gli

## CONCLUSIONI

venisse imputata l'impossibilità di aggregazione dei dati. Come si è mostrato in questa tesi, invece, non è per nulla impossibile.

Per concludere, il formato OMX sembra essere un discreto strumento per la memorizzazione dei dati bidimensionali. Permette di memorizzare le informazioni secondo una logica molto semplice: le matrici attinenti si possono raggruppare in uno stesso file, sotto la cartella denominata "*data*", identificandole in questa con un nome caratteristico. Una seconda cartella, dal nome "*lookup*", ospita invece le mappe degli indici. La lettura dei dati è immediata ed accessibile a chiunque. Lo spazio occupato nella memoria per l'archiviazione è esiguo. Gli attributi sono memorizzati, per ogni matrice, e possono essere recuperati accedendo alle proprietà della stessa. Ma, come già accennato, per essere ancora più di immediata comprensione, dovrebbe permettere alle mappe degli indici di poter accogliere anche dei dati di tipo stringa.

Per quanto riguarda la "maneggiabilità" dei dati contenuti, si è visto che per le operazioni più semplici non è molto difficile redigere uno *script* allo scopo. Ovviamente, se il formato OMX venisse usato come formato di salvataggio non proprietario all'interno di un software di pianificazione, queste operazioni sarebbero effettuabili in maniera forse più immediata agendo direttamente dall'interfaccia dello stesso con comandi già implementati. Un suggerimento, quindi, potrebbe essere proprio quello di implementare delle nuove funzioni, creando ad esempio un nuovo modulo Python, per mettere in relazione le matrici salvate con il formato OMX con un software GIS, come potrebbe esserlo QGIS, in modo da compiere le principali operazioni e interrogazioni sui dati direttamente da questo, attraverso il richiamo di un comando.



# ALLEGATO A

## ***SCRIPT DI ESEMPIO PER LA CREAZIONE DI UN FILE OMX CON ATTRIBUTI E MAPPATURA DELL'INDICE***

Di seguito viene riportato uno *script* di Python con cui è possibile memorizzare una serie di matrici all'interno di un unico file OMX. Nel caso specifico si è scelto di utilizzare due matrici di numeri casuali, utilizzate rispettivamente per rappresentare degli spostamenti con due diversi modi (privato e pubblico). Una terza è determinata dalla somma delle due precedenti (spostamenti totali). La dimensione delle tre matrici è 5x5.

Alla matrice che rappresenta gli spostamenti privati sono stati assegnati, sempre a scopo esemplificativo, una serie di metadati.

Infine, è stata associata una mappatura degli indici delle tre matrici memorizzate.

Si veda il Paragrafo 1.1.1 *Struttura dei dati* per l'apertura del file così generato con il visualizzatore OMX Viewer.

```
01 #Caricamento delle librerie
02 import openmatrix as omx
03 import numpy as np
04
05 #Creazione del file di esempio
06 esempio = omx.open_file('FileEsempio.omx','w')
07
08 #Generazione dei dati - creo tre matrici utilizzando dei numeri casuali
09 priv = np.random.random(size=(5,5))*1000
10 pubb = np.random.random(size=(5,5))*1000
11 tot = priv+pubb
12
13 #Scrittura delle matrici sul file
14 esempio['PRIV'] = priv
15 esempio['PUBB'] = pubb
16 esempio['TOT'] = tot
17
18 #Assegnazione dei metadati (passaggio facoltativo)
```

**ALLEGATO A**      Script di esempio per la creazione di un file OMX con attributi e mappatura dell'indice

```
19 esempio['PRIV'].attrs.periodo = 'tutta la giornata'
20 esempio['PRIV'].attrs.scopo = 'tutti'
21 esempio['PRIV'].attrs.modo = 'privato'
22 esempio['PRIV'].attrs.zonizzazione = 'provinciale'
23 esempio['PRIV'].attrs.anno = '2017'
24
25 #Associazione della mappatura (passaggio facoltativo)
26 map = [401, 402, 404, 405, 410]
27 esempio.create_mapping('mappatura',map)
28
29 #Chiusura del file
30 esempio.close()
```

# ALLEGATO B

## ***ZONE CAMPIONARIE IMQ 2013 E MACRO-AREE DI APPARTENENZA***

Nella TABELLA B.0.1 sono elencate le 208 zone campionarie dell'IMQ del 2013, i relativi codici per l'identificazione nei database, la descrizione delle zone o dei comuni coinvolti e la provincia per ogni macro-area di appartenenza.

TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte. Elaborazione propria basata sui dati dell'IMQ 2013.

CODICE	NOME ZONA	DESCRIZIONE DEI COMUNI COINVOLTI
<b>Macro-area Comune di Torino</b>		
Q001	Centro	Torino quartiere Centro
Q002	San Salvario	Torino quartiere San Salvario
Q003	Crocetta	Torino quartiere Crocetta
Q004	San Paolo	Torino quartiere San Paolo
Q005	Cenisia	Torino quartieri Cenisia e Cit Turin
Q006	San Donato	Torino quartieri Campidoglio e San Donato
Q007	Aurora	Torino quartiere Aurora
Q008	Vanchiglia	Torino quartiere Vanchiglia
Q009	Nizza e Millefonti	Torino quartiere Millefonti e Nizza
Q010	Lingotto	Torino quartieri Filadelfia e Lingotto
Q011	Santa Rita	Torino quartiere Santa Rita
Q012	Mirafiori Nord	Torino quartiere Mirafiori Nord
Q013	Pozzo Strada	Torino quartieri Borgata Lesna e Pozzo Strada
Q014	Parella	Torino quartiere Parella
Q015	Valette	Torino quartieri Lucento e Vallette
Q016	Madonna di Campagna	Torino quartiere Madonna di Campagna
Q017	Borgata Vittoria	Torino quartiere Borgata Vittoria
Q018	Barriera di Milano	Torino quartiere Barriera di Milano
Q019	Falchera	Torino quartieri Barca, Bertolla, Falchera e Villaretto
Q020	Regio Parco	Torino quartieri Rebaudengo e Regio Parco
Q021	Madonna del Pilone	Torino quartieri Madonna del Pilone e Sassi

*(segue)*

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

Q022	Cavoretto	Torino quartieri Borgo Po e Cavoretto
Q023	Mirafiori Sud	Torino quartiere Mirafiori Sud
<b>Macro-area Cintura di Torino</b>		
C002	Grugliasco	Grugliasco
C003	Collegno	Collegno
C004	Venaria	Venaria Reale
C005	Borgaro Torinese	Borgaro Torinese
C006	Settimo Torinese	Settimo Torinese
C007	San Mauro Torinese	San Mauro Torinese
C008	Pino Torinese	Pino Torinese
C009	Moncalieri	Moncalieri
C010	Pecetto Torinese	Pecetto Torinese
C011	Nichelino	Nichelino
C012	Candiolo	Candiolo
C013	Beinasco	Beinasco
C014	Orbassano	Orbassano
C015	Rivalta Di Torino	Rivalta Di Torino
C016	Rivoli	Rivoli
C017	Alpignano	Alpignano
C018	Pianezza	Pianezza
C019	Druento	Druento
C020	Leini	Leini
C021	Chieri	Chieri
C022	Trofarello	Trofarello
C023	Cambiano	Cambiano
C024	Santena	Santena
C064	Caselle Torinese	Caselle Torinese
C065	Volpiano	Volpiano
C066	Baldissero Torinese	Baldissero Torinese
C067	La Loggia	La Loggia
C068	Carignano	Carignano
C069	Vinovo	Vinovo
C070	Piobesi Torinese	Piobesi Torinese
C071	Piossasco	Piossasco
<b>Macro-area Provincia di Torino</b>		
E025	Castagnole Piemonte	Castagnole Piemonte, Lombriasco, Osasio, Pancalieri e Virle Piemonte

(segue)

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

E026	None e Volvera	None e Volvera
E027	Cumiana	Cantalupa, Cumiana, Frossasco e Roletto
E028	Giaveno e Val Sangone	Bruino, Coazze, Giaveno, Reano, Sangano, Trana, Valgioie e Villarbasse
E030	Avigliana	Avigliana, Buttigliera Alta e Rosta
E031	Val della Torre	Caslette, Givoletto, La Cassa, S. Gillio, Val della Torre
E032	Fiano	Fiano, Robassomero, Vallo Torinese e Varisella
E034	Feletto	Bosconero, Feletto, Lombardore, Lusigliè, S. Benigno
E035	Gassino	Castiglione Torinese, Cinzano, Gassino Torinese, Rivalba, San Raffaele Cimena e Sciolze
E036	Collina Chierese	Andezeno, Arignano, Marentino, Mombello di Torino, Montaldo Torinese, Moriondo Torinese, Pavarolo e Riva Presso Chieri
E037	Poirino	Isolabella, Poirino e Pralormo
E038	Carmagnola e Villastellone	Carmagnola e Villastellone
E039	Airasca	Airasca, Cercenasco, Piscina, Scalenghe, Vigone e Villafranca Piemonte
E040	Pinerolo	Pinerolo
E041	Cavour	Buriasco, Cavour, Garzigliana e Macello
E042	Bricherasio	Bricherasio, Campiglione Fenile, Osasco, Prarostino e San Secondo di Pinerolo
E043	Val Pellice	Angrogna, Bibiana, Bobbio Pellice, Luserna San Giovanni, Lusernetta, Rorà, Torre Pellice e Villar Pellice
E044	Val Chisone	Fenestrelle, Inverso Pinasca, Massello, Perosa Argentina, Perrero, Pinasca, Pomaretto, Porte, Prigelato, Prali, Pramollo, Roreto Chisone, Roure, Salza di Pinerolo, San Germano Chisone, San Pietro Val Lemina, Usseaux e Villar Perosa
E045	Bussoleno e Susa	Bussoleno e Susa
E046	Bassa Val Susa	Almese, Borgone Susa, Bruzolo, Caprie, Chianocco, Chiusa di San Michele, Condove, Rubiana, San Didero, San Giorgio di Susa, Sant'Ambrogio di Torino, Sant'Antonino di Susa, Vaie, Villar Dora e Villar Focchiardo
E047	Alta Val Susa	Bardonecchia, Cesana Torinese, Chiomonte, Claviere, Exilles, Giaglione, Graverè, Mattie, Meana di Susa, Mompantero, Moncenisio, Novalesa, Oulx, Salbertrand, Sauze d'Oulx, Sauze di Cesana, Sestriere e Venaus

(segue)

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

E048	Valli di Lanzo	Ala di Stura, Balme, Cantoiria, Ceres, Chialamberto, Coassolo Torinese, Germagnano, Groscavallo, Lemie, Mezzenile, Monastero di Lanzo, Pessinetto, Traves, Usseglio e Viù
E049	Lanzo	Balangero, Cafasse e Lanzo Torinese
E050	Cirié e S. Maurizio C.se	Cirié e San Maurizio Canavese
E051	Corio	Barbania, Corio, Front, Levone, Rocca Canavese e Vauda Canavese
E052	Rivarolo Canavese e Ozegna	Ozegna e Rivarolo Canavese
E053	Cuorgné	Busano, Canischio, Cuorgné, Favria, Forno Canavese, Oglianico, Pertusio, Prascorsano, Pratiglione, Rivara, Rivarossa, Salassa, San Colombano Belmonte, San Ponso e Valperga
E054	Caluso	Barone Canavese, Caluso, Candia Canavese, Ciconio, Foglizzo, Mazzè, Mercenasco, Montalenghe, Montanaro, Orio Canavese, San Giorgio Canavese, San Giusto Canavese Villareggia e Vische
E055	Chivasso e Brandizzo	Brandizzo e Chivasso
E056	Verolengo	Rondissone, Torrazza Piemonte e Verolengo
E057	Cavagnolo	Brozolo, Brusasco, Casalborgone, Castagneto Po, Cavagnolo, Lauriano, Monteu da Po, San Sebastiano da Po e Verrua Savoia
E058	Valli Orco e Soana	Alpette, Ceresole Reale, Frassinetto, Ingria, Locana, Noasca, Pont Canavese, Ribordone, Ronco Canavese, Sparone e Valprato Soana
E059	Castellamonte	Aglié, Bairo, Borgiallo, Castellamonte, Castelnuovo Nigra, Chiesanuova, Cintano e Collettero Castelnuovo
E060	Ivrea	Ivrea
E061	Ivrea Est	Albiano d'Ivrea, Azeglio, Bollengo, Borgomasino, Caravino, Cossano Canavese, Maglione, Palazzo Canavese, Piverone, Settimo Rottaro e Vestignè
E062	Settimo Vittone	Alice Superiore, Andrate, Baldissero Canavese, Brosso, Carema, Cuceglio, Issiglio, Lugnacco, Meugliano, Nomaglio, Pecco, Perosa Canavese, Quincinetto, Rueglio, San Martino Canavese, Settimo Vittone, Strambinello, Tavagnasco, Torre Canavese, Trausella, Traversella, Vialfrè, Vico Canavese, Vidracco e Vistrorio
E063	San Francesco al Campo	Grosso, Mathi, Nole, San Carlo Canavese, San Francesco al Campo e Villanova Canavese

(segue)

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

E072	Ivrea Nord	Borgofranco d'Ivrea, Burolo, Cascinette d'Ivrea, Chiaverano, Lessolo, Montaldo Dora e Quassolo
E073	Ivrea Ovest	Banchette, Collettero Giacosa, Fiorano Canavese, Loranze, Parella, Pavone Canavese, Quagliuzzo, Salerano Canavese e Samone
E074	Strambino	Romano Canavese, Scarmagno e Strambino
<b>Macro-area Provincia di Cuneo</b>		
R400	Cuneo	Cuneo
R401	Beinette	Beinette, Castelletto Stura, Margarita, Montanera e Morozzo
R402	Centallo	Centallo, Tarantasca, Villafalletto e Vottignasco
R403	Busca e Valle Maira	Acceglio, Busca, Canosio, Cartignano, Celle di Macra, Dronero, Elva, Macra, Marmora, Prazzo, Roccabruna, San Damiano Macra, Stroppio e Villar San Costanzo
R404	Caraglio e Valle Grana	Bernezzo, Caraglio, Castelmagno, Cervasca, Montemale di Cuneo, Montegrosso Grana, Pradleves, Valgrana e Vignolo
R405	Valle Stura di Demonte	Aisone, Argentera, Demonte, Gaiola, Moiola, Pietraporzio, Rittana, Roccasparvera, Sambuco, Valloriate e Vinadio
R406	Boves	Boves, Chiusa di Pesio e Peveragno
R407	Valli Gesso e Vermenagna	Entracque, Limone Piemonte, Roaschia, Robilante, Roccavione, Valdieri e Vernante
R408	Racconigi	Caramagna Piemonte, Casalgrasso, Cavallerleone, Cavallermaggiore, Faule, Marene, Monasterolo di Savigliano, Murello, Polonghera, Racconigi, Ruffia e Villanova Solaro
R410	Bene Vagienna	Bene Vagienna, Cervere, Salmour, Sant'Albano Stura e Trinità
R411	Savigliano e Genola	Genola e Savigliano
R412	Verzuolo e Valle Varaita	Bellino, Brondello, Brossasco, Casteldelfino, Castellar, Costigliole Saluzzo, Frassinio, Isasca, Manta, Melle, Pagno, Piasco, Pontechianale, Rossana, Sampeyre, Valmala, Venasca e Verzuolo
R413	Moretta	Cardè, Lagnasco, Moretta, Scarnafigi e Torre S. Giorgio
R414	Barge	Bagnolo Piemonte, Barge, Envie e Revello
R415	Paesana e Valle Po	Crissolo, Gambaasca, Martiniana Po, Oncino, Ostanta, Paesana, Rifreddo e Sanfront

(segue)

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

R416	Cherasco	Cherasco, La Morra, Narzole, Roddi, Santa Vittoria d'Alba e Verduno
R417	Sommariva del Bosco	Ceresole d'Alba, Pocapaglia, Sanfrè, Sommariva del Bosco e Sommariva Perno
R418	Diano d'Alba	Albaretto della Torre, Arguello, Barolo, Benevello, Borgomale, Bosia, Bossolasco, Castiglione Falletto, Castino, Cerretto Langhe, Cissone, Cravanzana, Diano d'Alba, Feisoglio, Grinzane Cavour, Lequio Berria, Monchiero, Monforte d'Alba, Montelupo Albese, Niella Belbo, Novello, Roddino, Rodello, San Benedetto Belbo, Serralunga d'Alba, Serravalle Langhe e Sinio
R419	Cortemilia	Bergolo, Castelletto Uzzone, Cortemilia, Gorzegno, Levice, Perletto, Pezzolo Valle Uzzone e Torre Bormida
R420	Canale	Baldissero d'Alba, Canale, Castagnito, Castellinaldo, Corneliano d'Alba, Govone, Guarene, Magliano Alfieri, Montà, Montaldo Roero, Monteu Roero, Monticello d'Alba, Piobesi d'Alba, Priocca, Santo Stefano Roero e Vezza d'Alba
R421	Santo Stefano Belbo	Barbaresco, Camo, Castiglione Tinella, Cossano Belbo, Mango, Neive, Neviglie, Rocchetta Belbo, Santo Stefano Belbo, Treiso e Trezzo Tinella
R422	Vicoforte	Briaglia, Ciglié, Monasterolo Casotto, Montaldo di Mondovì, Niella Tanaro, Pamparato, Roburent, Rocca Ciglié, San Michele Mondovì, Torre Mondovì e Vicoforte
R423	Dogliani	Bastia Mondovì, Belvedere Langhe, Bonvicino, Carrù, Clavesana, Dogliani, Farigliano, Lequio Tanaro, Magliano Alpi, Piozzo, Rocca de Baldi e Somano
R424	Villanova di Mondovì	Frabosa Soprana, Frabosa Sottana, Monastero di Vasco, Pianfei, Roccaforte e Villanova Mondovì
R425	Saliceto	Camerana, Castellino Tanaro, Castelnuovo di Ceva, Gottasecca, Igliano, Lesegno, Marsaglia, Mombarcaro, Monesiglio, Montezemolo, Murazzano, Paroldo, Priero, Prunetto, Roascio, Sale delle Langhe, Sale San Giovanni, Saliceto e Torresina
R426	Valle Tanaro	Alto, Bagnasco, Battifollo, Briga Alta, Caprauna, Garessio, Lisio, Mombasiglio, Nucetto, Ormea, Perlo, Priola, Scagnello e Viola
R427	Alba	Alba

(segue)



(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

R428	Borgo San Dalmazzo	Borgo San Dalmazzo
R429	Bra	Bra
R430	Ceva	Ceva
R431	Fossano	Fossano
R432	Mondovì	Mondovì
R433	Saluzzo	Saluzzo
<b>Macro-area Provincia di Asti</b>		
R500	Asti	Asti
R501	Castelnuovo Don Bosco	Albugnano, Berzano di San Pietro, Buttigliera d'Asti, Capriglio, Castelnuovo Don Bosco, Moncucco Torinese e Pino d'Asti
R502	Montiglio Monferrato	Aramengo, Camerano Casasco, Cerretto d'Asti, Chiusano d'Asti, Cinaglio, Cocconato, Corsione, Cortanze, Cossombrato, Cunico, Montechiaro d'Asti, Montiglio Monferrato, Moransengo, Passerano Marmorito, Piea, Piovà Massaia, Robella, Settime, Soglio, Tonengo, Viale d'Asti e Villa San Secondo
R503	Baldichieri d'Asti	Baldichieri d'Asti, Castellero, Cortandone, Cortazzone, Mareto, Monale, Montafia e Roatto
R504	San Damiano d'Asti	Antignano, Cantarana, Celle Enomondo, Cisterna d'Asti, Revigliasco d'Asti, San Damiano d'Asti, San Martino Alfieri e Tigliole
R505	Costigliole d'Asti	Azzano d'Asti, Calosso, Castagnole delle Lanze, Coazzolo, Costigliole d'Asti, Isola d'Asti, Mongardino, Montaldo Scarampi, Montegrosso d'Asti, Rocca d'Arazzo, Rocchetta Tanaro e Vigliano d'Asti
R506	Moncalvo	Calliano, Casorzo, Castagnole Monferrato, Castell'Alfero, Castello di Annone, Cerro Tanaro, Frinco, Grana, Grazzano Badoglio, Moncalvo, Montemagno, Penango, Portacomaro, Refrancore, Scurzolengo, Tonco e Viarigi
R507	Villanova d'Asti	Cellarengo, Dusino San Michele, Ferrere, San Paolo Solbrito, Valfenera, Villafranca e Villanova d'Asti
R508	Mombercelli	Agliano Terme, Belveglio, Castelnuovo Calcea, Cortiglione, Incisa Scapaccino, Mombercelli, Vaglio Serra e Vinchio
R509	Mombaruzzo	Bruno, Castel Boglione, Castel Rocchero, Castelletto Molina, Castelnuovo Belbo, Fontanile, Maranzana, Mombaruzzo e Quaranti

(segue)

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

R510	Calamandrana	Bubbio, Calamandrana, Cassinasco, Cessole, Loazzolo, Moasca, Mombaldone, Monastero Bormida, Montabone, Olmo Gentile, Roccaverano, Rocchetta Palafea, San Giorgio Scarampi, San Marzano Oliveto, Serole, Sessame e Vesime
R511	Canelli	Canelli
R512	Nizza Monferrato	Nizza Monferrato
<b>Macro-area Provincia di Alessandria</b>		
R600	Alessandria	Alessandria
R601	Castellazzo Bormida	Bergamasco, Borgoratto Alessandrino, Bosco Marengo, Carentino, Casal Cermelli, Castellazzo Bormida, Castelspina, Frascaro, Frugarolo, Gamalero, Masio, Oviglio, Predosa e Sezzadio
R602	Felizzano	Castelletto Monferrato, Cuccaro Monferrato, Felizzano, Fubine, Lu, Quargnento, Quattordio e Solero
R603	Bassignana	Bassignana, Montecastello, Pecetto di Valenza, Pietra Marazzi, Piovera e Rivarone
R604	S. Salvatore Monferrato	San Salvatore Monferrato
R605	Viguzzolo	Avolasca, Berzano di Tortona, Brignano Frascata, Carbonara Scrivia, Carezzano, Casalnoceto, Casasco, Castellana, Castellar Guidobono, Cerreto Grue, Costa Vescovato, Dernice, Fabbrica Curone, Garbagna, Gremiasco, Momperone, Monleale, Montacuto, Montegioco, Montemarzino, Paderna, Pozzol Groppo, San Sebastiano Curone, Sant'Agata Fossili, Sarezzano, Spineto Scrivia, Viguzzolo, Villalvernia, Villaromagnano, Volpedo e Volpeglino
R606	Castelnuovo Scrivia	Alluvioni Cambiò, Alzano Scrivia, Castelnuovo Scriva, Guazzora, Isola Sant'Antonio, Molino dei Torti, Pontecurone e Sale
R607	Serravalle Scrivia	Albera Ligure, Arquata Scrivia, Borghetto di Borbera, Cabella Ligure, Cantalupo Ligure, Carrega Ligure, Cassano Spinola, Gavazzana, Grondona, Mongiardino Ligure, Pozzolo Formigaro, Roccaforte Ligure, Rocchetta Ligure, Sardigliano, Serravalle Scrivia, Stazzano e Vignole Borbera

(segue)

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

R608	Gavi	Basaluzzo, Bosio, Capriata d'Orba, Carrosio, Fraconalto, Francavilla Bisio, Fresonara, Gavi, Parodi Ligure, Pasturana, San Cristoforo, Tassarolo e Voltaggio
R609	Castelletto d'Orba	Belforte Monferrato, Casaleggio Boiro, Castelletto d'Orba, Lerma, Montaldeo, Mornese, Rocca Grimalda, Silvano d'Orba e Tagliolo Monferrato
R610	Molare	Carpeneto, Cassinelle, Cremolino, Molare e Trisobbio
R611	Bistagno	Bistagno, Cartosio, Castelletto d'Erro, Cavatore, Denice, Malvicino, Melazzo, Merana, Montechiaro d'Acqui, Pareto, Ponti, Ponzzone, Spigno Monferrato e Terzo
R612	Cassine	Alice Bel Colle, Cassine, Castelnuovo Bormida, Grogna, Montaldo Bormida, Morbello, Morsasco, Orsara Bormida, Prasco, Ricaldone, Rivalta Bormida, Strevi e Visone
R613	Cerrina	Alfiano Natta, Camino, Castelletto Merli, Cereseto, Cerrina Monferrato, Coniolo, Gabiano, Mombello Monferrato, Moncestino, Murisengo, Odalengo Grande, Odalengo Piccolo, Ottiglio, Ozzano Monferrato, Pontestura, Ponzano Monferrato, Sala Monferrato, San Giorgio Monferrato, Serralunga di Crea, Solonghello, Treville, Villadeati e Villamiroglio
R614	Villanova Monferrato	Balzola, Morano sul Po e Villanova Monferrato
R615	Rosignano Monferrato	Altavilla Monferrato, Borgo San Martino, Bozzole, Camagna Monferrato, Cella Monte, Conzano, Frassinello Monferrato, Frassineto Po, Giarole, Mirabello Monferrato, Occimiano, Olivola, Pomaro Monferrato, Rosignano Monferrato, Terruggia, Ticineto, Valmacca e Vignale Monferrato
R616	Acqui Terme	Acqui Terme
R617	Casale Monferrato	Casale Monferrato
R618	Novi Ligure	Novi Ligure
R619	Ovada	Ovada
R620	Tortona	Tortona
R621	Valenza	Valenza
<b>Macro-area Provincia di Vercelli</b>		
R200	Vercelli	Vercelli
R201	Crescentino	Crescentino, Fontanetto Po, Lamporo e Saluggia

(segue)

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

R202	Borgo Vercelli	Borgo Vercelli, Caresanablot, Casanova Elvo, Collobiano, Formigliana, Olcenengo, Oldenico, Quinto Vercellese e Villata
R203	Asigliano Vercellese	Agliano Vercellese, Caresana, Costanzana, Desana, Lignana, Motta de' Conti, Pertengo, Pezzana, Prarolo, Rive, Ronsecco, Sali Vercellese, Stroppiana e Tricerro
R204	Rovasenda	Albano Vercellese, Arborio, Balocco, Buronzo, Carisio, Ghislarengo, Greggio, Lenta, Rovasenda, San Giacomo Vercellese e Villarboit
R205	Livorno Ferraris	Alice Castello, Bianzè, Borgo d'Ale, Cigliano, Crova, Livorno Ferraris, Moncrivello, Salasco, San Germano Vercellese e Tronzano Vercellese
R207	Val Sesia	Alagna Valsesia, Balmuccia, Boccioleto, Campertogno, Carcoforo, Mollia, Pila, Piode, Postua, Rassa, Rima San Giuseppe, Rimasco, Riva Valdobbia, Rossa, Scopa e Scopello
R208	Varallo	Cervatto, Civiasco, Cravagliana, Fobello, Rimella, Sabbia, Varallo e Vocca
R209	Quarona	Breia, Cellio, Quarona e Valduggia
R210	Gattinara	Gattinara, Guardabosone, Lozzolo, Roasio e Serravalle Sesia
R211	Trino Palazzolo V.	Palazzo Vercellese e Trino
R212	Borgosesia	Borgosesia
R213	Santhià	Santhià
<b>Macro-area Provincia di Biella</b>		
R700	Biella	Biella
R701	Candelo	Benna, Borriana, Candelo, Cavaglià, Cerrione, Dorzano, Gaglianico, Magnano, Massazza, Ponderano, Roppolo, Salussola, Sandigliano, Verrone, Villanova Biellese, Viverone, Zimone e Zubiena
R702	Occhieppo Inferiore	Camburzano, Donato, Graglia, Mongrando, Muzzano, Netro, Occhieppo Inferiore, Occhieppo Superiore, Pollone, Sala Biellese, Sordevolo e Torrazzo
R703	Andorno Micca	Andorno Micca, Campiglia Cervo, Miagliano, Piedicavallo, Pralungo, Quittengo, Rosazza, Sagliano Micca, San Paolo Cervo, Tavigliano e Tollegno
R704	Trivero	Ailoche, Callabiana, Camandona, Caprile, Coggiola, Crevacuore, Mosso, Pettinengo, Portula, Pray, Selve Marcone, Trivero, Valle Mosso e Veglio

(segue)

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

R705	Lessona	Bioglio, Brusnengo, Casapinta, Castelletto Cervo, Crosa, Curino, Giffenga, Lessona, Masserano, Mezzana Mortigliengo, Mottalciata, Soprana, Sostegno, Strona e Villa del Bosco
R706	Vigliano Biellese	Cerreto Castello, Piatto, Quaregna, Ronco Biellese, Ternengo, Valdengo, Vallanzengo, Valle San Nicolao, Vigliano Biellese e Zumaglia
R707	Cossato	Cossato
<b>Macro-area Provincia di Novara</b>		
R300	Novara	Novara
R301	Grignasco	Ghemme, Grignasco, Prato Sesia, Romagnano Sesia e Sizzano
R302	Caltignaga	Biandrate, Briona, Caltignaga, Carpignano Sesia, Casalbeltrame, Casaleggio Novara, Casalvolone, Castellazzo Novarese, Fara Novarese, Landiona, Mandello Vitta, Recetto, San Nazzaro Sesia, San Pietro Mosezzo, Sillavengo e Vicolungo
R303	Vespolate	Borgolavezzaro, Casalino, Garbagna Novarese, Granozzo con Monticello, Nibbiola, Terdobbiate, Tornaco, Vespolate e Vinzaglio
R304	Galliate	Cameri, Galliate e Romentino
R305	Treccate	Cerano, Sozzago e Treccate
R306	Oleggio	Barengo, Bellinzago Novarese, Marano Ticino, Mezzomerico, Momo, Oleggio e Vaprio d'Agogna
R307	Castelletto Sopra Ticino	Borgo Ticino, Castelletto Sopra Ticino, Comignago, Divignano, Dormelletto, Pombia e Varallo Pombia
R308	Fontaneto d'Agogna	Boca, Cavaglietto, Cavaglio d'Agogna, Cavallirio, Cressa, Cureggio, Fontaneto d'Agogna e Maggiore
R309	Gozzano	Bolzano Novarese, Briga Novarese, Gargallo, Gozzano, Inorio, Oleggio Castello, Paruzzaro, Pugno e Soriso
R310	Gattico	Agrate Conturbia, Bogogno, Gattico, Suno e Veruno
R311	San Maurizio d'Opaglio	Ameno, Armeno, Colazza, Lesa, Massino Visconti, Meina, Miasino, Nebbiuno, Orta San Giulio, Pella, Pettenasco, Pisano e San Maurizio d'Opaglio
R312	Arona	Arona
R313	Borgomanero	Borgomanero
<b>Macro-area Provincia di Verbano</b>		
R800	Verbania	Verbania

(segue)

(segue) TABELLA B.0.1 Descrizione delle zone campionarie in cui è stato suddiviso il Piemonte.  
Elaborazione propria basata sui dati dell'IMQ 2013.

R801	Cannobio	Arizzano, Aurano, Bee, Cambiasca, Cannero Riviera, Cannobio, Caprezzo, Cavaglio Spocchia, Cossogno, Cursolo Orasso, Falmenta, Ghiffa, Gurro, Intragna, Miazzina, Oggebbio, Premeno, San Bernardino Verbano, Trarego Viggiona e Vignone
R802	Stresa	Baveno, Belgirate, Brovello Carpugnino, Gignese, Stresa
R803	Crevoladossola	Baceno, Bognanco, Crevoladossola, Crodo, Formazza, Masera, Montecrestese, Premia, Trasquera, Trontano e Varzo
R804	Malesco	Craveggia, Druogno, Malesco, Re, Santa Maria Maggiore, Toceno e Villette
R805	Ornavasso	Anzola d'Ossola, Mergozzo, Ornavasso, Premosello Chiovena e Vogogna
R806	Villadossola	Antrona Schieranco, Beura Cardezza, Montescheno, Pallanzeno, Seppiana, Viganella e Villadossola
R807	Pieve Vergonte	Bannio Anzino, Calasca Castiglione, Ceppo Morelli, Macugnaga, Piedimulera, Pieve Vergote, Vanzone con San Carlo
R808	Gravellona Toce	Arola, Casale Corte Cerro, Cesara, Germagno, Gravellona Toce, Loreglia, Madonna del Sasso, Massiola, Nonio, Quarna Sopra, Quarna Sotto e Valstrona
R809	Domodossola	Domodossola
R810	Omegna	Omegna

# ALLEGATO C

## **MATRICI ORIGINE/DESTINAZIONE PRODOTTE DALL'AMP SULLA BASE DEI DATI IMQ 2013**

Nell'elenco seguente sono riportate le 24 matrici redatte dall'Agenzia della Mobilità Piemontese (AMP) in occasione dell'Indagine sulla Mobilità e sulla Qualità dei trasporti del 2013. Si ricorda che tali matrici, utilizzate come base di dati per le elaborazioni del presente lavoro, sono reperibili *on-line* sul sito ufficiale dell'Agenzia stessa, nella sezione "Dati e Statistiche"<sup>82</sup>, con formato di divulgazione XLS. A seguito della descrizione del contenuto di ogni matrice è riportato il relativo nome del *file* Excel; la stessa codifica sarà utilizzata per assegnare il nome alle singole matrici nei *file* OMX prodotti.

\* IMQ2013 per macro-aree, popolazione residente nella provincia di Torino:

1. Mezzo privato – intera giornata – con ritorni a casa  
*IMQ2013\_MPrIG\_ResTO.xls*
2. Mezzo privato – intera giornata – senza ritorni a casa  
*IMQ2013\_MPrIG\_ResTOsr.xls*
3. Mezzo privato – ora di punta (7:43-8:42) – con ritorni a casa  
*IMQ2013\_MPrOP\_ResTO.xls*
4. Mezzo pubblico – intera giornata – con ritorni a casa  
*IMQ2013\_MPuIG\_ResTO.xls*
5. Mezzo pubblico – intera giornata – senza ritorni a casa  
*IMQ2013\_MPuIG\_ResTOsr.xls*

---

<sup>82</sup> Agenzia della Mobilità Piemontese, *Matrici Origine/Destinazione indagine IMQ 2013*, <http://mtm.torino.it/dati-statistiche/indagine-imq-2013/matrici-od-imq-2013>, data di consultazione: 14/05/2017.

6. Mezzo pubblico – ora di punta (7:43-8:42) – con ritorni a casa  
*IMQ2013\_MPuOP\_ResTO.xls*

\* IMQ2013 per macro-aree, popolazione residente nella regione:

7. Mezzo privato – intera giornata – con ritorni a casa  
*IMQ2013\_MPrIG\_ResPIE.xls*

8. Mezzo privato – intera giornata – senza ritorni a casa  
*IMQ2013\_MPrIG\_ResPIEsr.xls*

9. Mezzo privato – ora di punta (7:43-8:42) – con ritorni a casa  
*IMQ2013\_MPrOP\_ResPIE.xls*

10. Mezzo pubblico – intera giornata – con ritorni a casa  
*IMQ2013\_MPuIG\_ResPIE.xls*

11. Mezzo pubblico – intera giornata – senza ritorni a casa  
*IMQ2013\_MPuIG\_ResPIEsr.xls*

12. Mezzo pubblico – ora di punta (7:43-8:42) – con ritorni a casa  
*IMQ2013\_MPuOP\_ResPIE.xls*

\* IMQ2013 per zone campionarie, popolazione residente nella provincia di Torino:

13. Mezzo privato – intera giornata – con ritorni a casa  
*IMQ2013\_MPrIG\_ResTO.xls*

14. Mezzo privato – intera giornata – senza ritorni a casa  
*IMQ2013\_MPrIG\_ResTOsr.xls*

15. Mezzo privato – ora di punta (7:43-8:42) – con ritorni a casa  
*IMQ2013\_MPrOP\_ResTO.xls*

16. Mezzo pubblico – intera giornata – con ritorni a casa  
*IMQ2013\_MPuIG\_ResTO.xls*

17. Mezzo pubblico – intera giornata – senza ritorni a casa  
*IMQ2013\_MPuIG\_ResTOsr.xls*

18. Mezzo pubblico – ora di punta (7:43-8:42) – con ritorni a casa  
*IMQ2013\_MPuOP\_ResTO.xls*



## \* IMQ2013 per zone campionarie, popolazione residente nella regione:

19. Mezzo privato – intera giornata – con ritorni a casa  
*IMQ2013\_MPrIG\_ResPIE.xls*
20. Mezzo privato – intera giornata – senza ritorni a casa  
*IMQ2013\_MPrIG\_ResPIEsr.xls*
21. Mezzo privato – ora di punta (7:43-8:42) – con ritorni a casa  
*IMQ2013\_MPrOP\_ResPIE.xls*
22. Mezzo pubblico – intera giornata – con ritorni a casa  
*IMQ2013\_MPuIG\_ResPIE.xls*
23. Mezzo pubblico – intera giornata – senza ritorni a casa  
*IMQ2013\_MPuIG\_ResPIEsr.xls*
24. Mezzo pubblico – ora di punta (7:43-8:42) – con ritorni a casa  
*IMQ2013\_MPuOP\_ResPIE.xls*

# ALLEGATO D

## ***SCRIPT PYTHON PER LA VERIFICA DEL CODICE DI CREAZIONE DEL FILE OMX PER ZONE CAMPIONARIE***

Di seguito è riportato il codice Python dal nome “*VerificaCodice.py*” utilizzato per provare la correttezza delle matrici OMX di dimensioni 209x209 redatte per zone campionarie.

```
01 import openpyxl
02 import openmatrix as omx
03
04 #Caricamento delle matrici O/D dal file OMX
05 OMXfile = omx.open_file('IMQ2013_zonecampionarie.omx',"r")
06 matriciOMX = OMXfile.list_matrices()
07
08 #Caricamento dei vettori delle zone
09 zone_exceldoc = openpyxl.load_workbook('ZoneCampionarie2013.xlsx')
10 zone_nsheets = zone_exceldoc.sheetnames
11 zone_sheet = zone_exceldoc[zone_nsheets[0]]
12 zoneA = list()
13 for j in range(0,209):
14     zoneA.append(zone_sheet.cell(row=j+2,column=2).value)
15
16 #Apertura del file Excel contenente le tabelle pivot
17 PIV_exceldoc = openpyxl.load_workbook('IMQ2013_TabellaePivot.xlsx')
18 PIV_nsheets = PIV_exceldoc.sheetnames
19
20 #Calcolo del RMSE
21 print('- VERIFICA DEL CODICE DI SCRITTURA DEL FILE OMX
22                                     PER ZONE CAMPIONARIE-')
23 print('\nTabella Pivot - Matrice OMX:\tRMSE')
24 for mat in range(0,12):
25     PIV = PIV_exceldoc[PIV_nsheets[mat]]
26     somma = 0
27     for i in range(0,209):
28         for j in range(0,209):
29             fPIV = PIV.cell(row=i+2,column=j+2).value
30             try:
31                 float(fPIV)
32             except:
33                 try:
```

```

34         int(fPIV)
35     except:
36         fPIV = 0.0
37     OR = PIV.cell(row=i+2,column=1).value
38     DE = PIV.cell(row=1,column=j+2).value
39     fOMX = OMXfile[matriciOMX[mat]][zoneA.index(OR)][zoneA.index(DE)]
40     somma = somma+(fPIV-fOMX)**2
41     RMSE = (somma/(209*209))**(0.5)
42     print(PIV_nsheet[mat], '- ',matriciOMX[mat], '\t',RMSE)
43
44 #Chiusura del file aperto
45     OMXfile.close()

```

Le prime operazioni compiute dal codice sono quelle che permettono l’apertura del file OMX contenente le matrici O/D (dal nome “*IMQ2013\_zonecampionarie.omx*”, ottenuto con lo *script* del Paragrafo 4.3.2 Matrici per zone campionarie e il cui contenuto è riportato nell’ALLEGATO E), l’apertura del file Excel contenente l’elenco delle zone campionarie in forma alfanumerica (dal nome “*ZoneCampionarie2013.xlsx*” si veda la Nota 75 a Pagina 99) e infine il file Excel contenente le Tabelle Pivot<sup>83</sup>. Entrando nel merito del codice, le operazioni compiute sono le seguenti: per ogni cella della matrice Excel si memorizza il corrispondente flusso nella variabile “fPIV” (si veda lo *script* alla riga 29); si eseguono quindi dei controlli sul tipo di variabile memorizzata (si veda lo *script* alle righe 30-36) in quanto le celle vuote, altrimenti, sarebbero del tipo “NoneType” e non permetterebbero le operazioni aritmetiche successive, nel caso si assegna manualmente il valore 0 alla variabile; si memorizzano quindi nelle variabili “OR” e “DE” rispettivamente il codice alfanumerico di origine e di destinazione del flusso (questo risulta necessario poiché le due matrici da confrontare non hanno lo stesso ordine di mappatura); grazie al vettore delle zone “zoneA” si ricava l’indice corretto per l’ingresso nelle matrici OMX da cui si estrae il flusso “fOMX”

---

<sup>83</sup> Documento Excel dal nome “*IMQ2013\_TabellePivot.xlsx*”, di propria elaborazione, che accoglie le 12 Tabelle Pivot create sulla base degli altrettanti file forniti dall’AMP, contenenti le matrici O/D sotto forma di elenco (si veda l’ALLEGATO C). Nello specifico ogni foglio del documento accoglie una matrice O/D, i nomi di questi ultimi richiamano i file da cui sono state prese le informazioni per la compilazione.

**ALLEGATO D**      Script Python per la verifica del codice di creazione del file OMX per zone campionarie

(si veda lo *script* alla riga 39). A questo punto è possibile applicare la formula per il calcolo del RMSE e stampare a video il risultato ottenuto. Queste operazioni vengono compiute 12 volte, tante quante le coppie di matrici da confrontare (si veda lo *script* alla riga 24).

# ALLEGATO E

## ***MATRICI ORIGINE/DESTINAZIONE PRODOTTE NEGLI SCRIPT PYTHON***

Il file per macro-aree “*IMQ2013\_macroaree.omx*”, generato nel codice riportato nel Paragrafo 4.3.1 Matrici per macro-aree, comprende 12 matrici O/D, i cui nomi e attributi sono elencati di seguito.

\* *ResPIEmPrIGsr*

Indagine: IMQ2013  
Periodo: intera giornata  
Numerosità del campione: 52119  
Popolazione: regione Piemonte  
Scopo: tutti escluso il ritorno  
Modo: privato

\* *ResPIEmPrIG*

Indagine: IMQ2013  
Periodo: intera giornata  
Numerosità del campione: 52119  
Popolazione: regione Piemonte  
Scopo: tutti  
Modo: privato

\* *ResTOmPrOP*

Indagine: IMQ2013  
Periodo: ora di punta (arrivo 07:43-08:42)  
Numerosità del campione: 32460  
Popolazione: provincia di Torino  
Scopo: tutti  
Modo: privato

- \* ResPIEmPrOP
  - Indagine: IMQ2013
  - Periodo: ora di punta (arrivo 07:43-08:42)
  - Numerosità del campione: 52119
  - Popolazione: regione Piemonte
  - Scopo: tutti
  - Modo: privato
  
- \* ResTOMPuIG
  - Indagine: IMQ2013
  - Periodo: intera giornata
  - Numerosità del campione: 32460
  - Popolazione: provincia di Torino
  - Scopo: tutti
  - Modo: pubblico
  
- \* ResTOMPuIGsr
  - Indagine: IMQ2013
  - Periodo: intera giornata
  - Numerosità del campione: 32460
  - Popolazione: provincia di Torino
  - Scopo: tutti escluso il ritorno
  - Modo: pubblico
  
- \* ResPIEmPuIG
  - Indagine: IMQ2013
  - Periodo: intera giornata
  - Numerosità del campione: 52119
  - Popolazione: regione Piemonte
  - Scopo: tutti
  - Modo: pubblico
  
- \* ResTOMPuOP
  - Indagine: IMQ2013
  - Periodo: ora di punta (arrivo 07:43-08:42)
  - Numerosità del campione: 32460
  - Popolazione: provincia di Torino
  - Scopo: tutti
  - Modo: pubblico

- \* ResPIEmPuIGsr
  - Indagine: IMQ2013
  - Periodo: intera giornata
  - Numerosità del campione: 52119
  - Popolazione: regione Piemonte
  - Scopo: tutti escluso il ritorno
  - Modo: pubblico
  
- \* ResTOMPrIG
  - Indagine: IMQ2013
  - Periodo: intera giornata
  - Numerosità del campione: 32460
  - Popolazione: provincia di Torino
  - Scopo: tutti
  - Modo: privato
  
- \* ResTOMPrIGsr
  - Indagine: IMQ2013
  - Periodo: intera giornata
  - Numerosità del campione: 32460
  - Popolazione: provincia di Torino
  - Scopo: tutti escluso il ritorno
  - Modo: privato
  
- \* ResPIEmPuOP
  - Indagine: IMQ2013
  - Periodo: ora di punta (arrivo 07:43-08:42)
  - Numerosità del campione: 52119
  - Popolazione: regione Piemonte
  - Scopo: tutti
  - Modo: pubblico

Il secondo file per zone campionarie, denominato “*IMQ2013\_zonecampionarie.omx*”, creato con lo *script* riportato nel Paragrafo 4.3.2 Matrici per zone campionarie, comprende altrettante matrici, i cui nomi e attributi sono riportati, invece, nel seguente elenco.

- \* ResPIEmPrIGsr
  - Indagine: IMQ2013
  - Periodo: intera giornata
  - Numerosità del campione: 52119
  - Popolazione: regione Piemonte
  - Scopo: tutti escluso il ritorno
  - Modo: privato
  
- \* ResPIEmPrIGt
  - Indagine: IMQ2013
  - Periodo: intera giornata
  - Numerosità del campione: 52119
  - Popolazione: regione Piemonte
  - Scopo: tutti
  - Modo: privato
  
- \* ResTOmPrOPt
  - Indagine: IMQ2013
  - Periodo: ora di punta (arrivo 07:43-08:42)
  - Numerosità del campione: 32460
  - Popolazione: provincia di Torino
  - Scopo: tutti
  - Modo: privato
  
- \* ResPIEmPrOPt
  - Indagine: IMQ2013
  - Periodo: ora di punta (arrivo 07:43-08:42)
  - Numerosità del campione: 52119
  - Popolazione: regione Piemonte
  - Scopo: tutti
  - Modo: privato
  
- \* ResTOmPuIGt
  - Indagine: IMQ2013
  - Periodo: intera giornata
  - Numerosità del campione: 32460
  - Popolazione: provincia di Torino
  - Scopo: tutti
  - Modo: pubblico



- \* ResTOMPuIGsr  
Indagine: IMQ2013  
Periodo: intera giornata  
Numerosità del campione: 32460  
Popolazione: provincia di Torino  
Scopo: tutti escluso il ritorno  
Modo: pubblico
- \* ResPIEmPuIGt  
Indagine: IMQ2013  
Periodo: intera giornata  
Numerosità del campione: 52119  
Popolazione: regione Piemonte  
Scopo: tutti  
Modo: pubblico
- \* ResTOMPuOPt  
Indagine: IMQ2013  
Periodo: ora di punta (arrivo 07:43-08:42)  
Numerosità del campione: 32460  
Popolazione: provincia di Torino  
Scopo: tutti  
Modo: pubblico
- \* ResPIEmPuIGsr  
Indagine: IMQ2013  
Periodo: intera giornata  
Numerosità del campione: 52119  
Popolazione: regione Piemonte  
Scopo: tutti escluso il ritorno  
Modo: pubblico
- \* ResTOMPrIGt  
Indagine: IMQ2013  
Periodo: intera giornata  
Numerosità del campione: 32460  
Popolazione: provincia di Torino  
Scopo: tutti

Modo: privato

\* ResTOMPrIGsr

Indagine: IMQ2013

Periodo: intera giornata

Numerosità del campione: 32460

Popolazione: provincia di Torino

Scopo: tutti escluso il ritorno

Modo: privato

\* ResPIEmPuOPt

Indagine: IMQ2013

Periodo: ora di punta (arrivo 07:43-08:42)

Numerosità del campione: 52119

Popolazione: regione Piemonte

Scopo: tutti

Modo: pubblico

Si noti come, essendo due file OMX diversi, si è potuto assegnare lo stesso nome alle 12 matrici, che sostanzialmente, contengono gli stessi dati ma con due zonizzazioni differenti.

# ALLEGATO F

## **LISTATO MATLAB PER LA CREAZIONE DELLE MATRICI MAT PER MACRO-AREE**

Di seguito è riportato il codice MATLAB, dal nome “*Macroaree.m*”, che consente di creare li file MAT “*IMQ2013\_macroaree.mat*” contenente le matrici O/D di dimensione 5x5, redatte sulla base dei dati dell’indagine IMQ del 2013 per macro-aree.

```
01 close all;
02 clear all;
03 clc;
04
05 %Caricamento delle informazioni sulla zonizzazione
06 zone = {'Torino','Cintura','Resto Provincia TO',
07         'Resto Piemonte','Esterno'};
08 zone = string(zone);
09
10 %Creazione delle matrici O/D
11 [ResPIEmPrIGt] = estrazionematrice('IMQ2013_MPrIG_ResPIE.xlsx',zone);
12 [ResPIEmPrIGsr] = estrazionematrice('IMQ2013_MPrIG_ResPIEsr.xlsx',zone);
13 [ResTOMPrIGt] = estrazionematrice('IMQ2013_MPrIG_RestO.xlsx',zone);
14 [ResTOMPrIGsr] = estrazionematrice('IMQ2013_MPrIG_RestOsr.xlsx',zone);
15 [ResPIEmProPt] = estrazionematrice('IMQ2013_MPrOP_ResPIE.xlsx',zone);
16 [ResTOMProPt] = estrazionematrice('IMQ2013_MPrOP_RestO.xlsx',zone);
17 [ResPIEmPuIGt] = estrazionematrice('IMQ2013_MPuIG_ResPIE.xlsx',zone);
18 [ResPIEmPuIGsr] = estrazionematrice('IMQ2013_MPuIG_ResPIEsr.xlsx',zone);
19 [ResTOMPuIGt] = estrazionematrice('IMQ2013_MPuIG_RestO.xlsx',zone);
20 [ResTOMPuIGsr] = estrazionematrice('IMQ2013_MPuIG_RestOsr.xlsx',zone);
21 [ResPIEmPuOPt] = estrazionematrice('IMQ2013_MPuOP_ResPIE.xlsx',zone);
22 [ResTOMPuOPt] = estrazionematrice('IMQ2013_MPuOP_RestO.xlsx',zone);
23
24 %Salvataggio delle matrici create nel file .mat
25 save IMQ2013_macroaree zone ResPIEmPrIGt ResPIEmPrIGsr ResTOMPrIGt
26     RestOMPrIGsr ResPIEmProPt RestOMProPt ResPIEmPuIGt ResPIEmPuIGsr
27     RestOMPuIGt RestOMPuIGsr ResPIEmPuOPt RestOMPuOPt;
28
29 %Definizione della funzione di compilazione della matrice O/D
30 function [matr] = estrazionematrice(nomefile,zone);
31 matr = zeros(length(zone),length(zone));
```

```
32 [SPOST,OeD] = xlsread(nomefile,'A9:E33');
33 ORIG = string(OeD(:,1));
34 DEST = string(OeD(:,3));
35 for i = 1:length(SPOST);
36     matr(find(zone==ORIG(i)),find(zone==DEST(i))) = SPOST(i);
37 end;
38 end
```

# FONTI BIBLIOGRAFICHE

1. Agenzia della Mobilità Piemontese, *Consorzio*  
<http://mtm.torino.it/it/agenzia/chi-siamo> (data di consultazione 13/05/17)
2. Agenzia della Mobilità Piemontese, *Cronistoria dell'indagine IMQ 2013*  
<http://mtm.torino.it/it/dati-statistiche/imq-alle-fasi-finali> (data di consultazione 13/05/17)
3. Agenzia della Mobilità Piemontese, *Dimensione del campione*  
<http://mtm.torino.it/it/dati-statistiche/faq-domande-frequenti-indagini/campionamento/dimensione-del-campione> (data di consultazione 14/05/17)
4. Agenzia della Mobilità Piemontese, *IMQ 2013. Indagine sulla Mobilità delle Persone e sulla Qualità dei Trasporti. Rapporto di sintesi sull'area metropolitana*, 2015  
<http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/pdf-1/IMQ2013RapportoSintesi.pdf/view>  
(data di consultazione 13/05/17)
5. Agenzia della Mobilità Piemontese, *IMQ 2013. Indagine sulla Mobilità delle Persone e sulla Qualità dei Trasporti. Rapporto sulla Regione Piemonte*, 2015  
[http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/pdf-1/IMQ2013\\_RapportoPiemonte.pdf](http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/pdf-1/IMQ2013_RapportoPiemonte.pdf)  
(data di consultazione 13/05/17)
6. Agenzia della Mobilità Piemontese, *Le indagini sulla mobilità delle persone sulla qualità dei trasporti*  
<http://mtm.torino.it/it/dati-statistiche/indagini> (data di consultazione 13/05/17)
7. Agenzia della Mobilità Piemontese, *Matrici Origine/Destinazione indagine IMQ 2013*  
<http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/matrici-od-imq-2013> (data di consultazione 14/05/17)
8. Agenzia della Mobilità Piemontese, *Ora di punta*  
<http://mtm.torino.it/it/dati-statistiche/faq-domande-frequenti-indagini/spostamenti/ora-di-punta>  
(data di consultazione 14/05/17)
9. Agenzia della Mobilità Piemontese, *Rapporti IMQ 2013*,  
<http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/rapporti-imq-2013> (data cons. 13/05/17)
10. Agenzia della Mobilità Piemontese, *Report dei comuni dell'area metropolitana*,  
<http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/report-dei-comuni-dellarea-metropolitana>  
(data di consultazione 15/07/17)

## FONTI BIBLIOGRAFICHE

11. Agenzia della Mobilità Piemontese, *Tabella associazione comuni alle zone campionarie*  
[http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/matrici-od-imq-2013/tab\\_comuni.xls/view](http://mtm.torino.it/it/dati-statistiche/indagine-imq-2013/matrici-od-imq-2013/tab_comuni.xls/view)  
(data di consultazione 14/05/17)
12. Agenzia per la Mobilità Metropolitana, *Indagine sulla Mobilità delle persone e sulla Qualità dei trasporti nella Regione Piemonte IMQ-Piemonte. Capitolato tecnico. Procedura per l'affidamento del servizio IMQ Piemonte, 2012*  
<http://mtm.torino.it/it/amministrazione-trasparente/bandi-di-gara-e-contratti/Beni-e-servizi/imq-2012/capitolato%20tecnico%20finale.pdf/view> (data di consultazione 18/07/17)
13. Agenzia per la Mobilità Metropolitana, *Linee Guida per la diffusione dei dati sulla Mobilità e sulla Qualità raccolti con le IMQ, 2008*  
[http://mtm.torino.it/it/agenzia/normativa/atti-amministrativi/deliberazioni-consiglio-di-amministrazione/cda\\_delibere\\_2008/09\\_2\\_linee\\_guida\\_diff\\_dati\\_mob.pdf](http://mtm.torino.it/it/agenzia/normativa/atti-amministrativi/deliberazioni-consiglio-di-amministrazione/cda_delibere_2008/09_2_linee_guida_diff_dati_mob.pdf) (data di consultazione 14/05/17)
14. Aimsun, *About Aimsun*  
<https://www.aimsun.com/about-aimsun/> (data di consultazione 17/01/18)
15. Aimsun, *Aimsun Live*  
<https://www.aimsun.com/aimsun-live/> (data di consultazione 17/01/18)
16. Aimsun, *Aimsun Next mobility modeling software*  
<https://www.aimsun.com/aimsun-next/> (data di consultazione 16/01/18)
17. Aimsun, *New features in Aimsun Next*  
<https://www.aimsun.com/aimsun-next/new-features/> (data di consultazione 16/01/18)
18. Aldus Developers Desk, *TIFF (1992)*  
<https://www.itu.int/itudoc/itu-t/com16/tiff-fx/docs/tiff6.pdf> (data di consultazione 07/11/17)
19. Benjamin Stabler, *Case Studies*  
<https://github.com/osPlanning/omx/wiki/Case-Studies> (data di consultazione 08/02/18)
20. Benjamin Stabler, *Cube*  
<https://github.com/osPlanning/omx/wiki/Cube> (data di consultazione 14/12/17)
21. Benjamin Stabler, *Cube OMX Converter*  
<https://github.com/osPlanning/omx/wiki/Cube-OMX-Converter> (data di consultazione 20/06/17)
22. Benjamin Stabler, *Data Structure 0.2*  
<https://github.com/osPlanning/omx/wiki/Data-Structure-0.2> (data di consultazione 20/06/17)
23. Benjamin Stabler, *Emme*  
<https://github.com/osPlanning/omx/wiki/EMME> (data di consultazione 07/12/17)
24. Benjamin Stabler, *HDF5*  
<https://github.com/osPlanning/omx/wiki/HDF5> (data di consultazione 20/06/17)
25. Benjamin Stabler, *Home*  
<https://github.com/osPlanning/omx/wiki> (data di consultazione 20/06/17)

## FONTI BIBLIOGRAFICHE

26. Benjamin Stabler, *OMX - Existing Solution*  
<https://github.com/osPlanning/omx/wiki/Existing-Solution> (data di consultazione 20/06/17)
27. Benjamin Stabler, *OMX Viewer*  
<https://github.com/osPlanning/omx/wiki/OMX-Viewer> (data di consultazione 20/06/17)
28. Benjamin Stabler, *Open matrix requirements*  
<https://github.com/osPlanning/omx/wiki/Open-matrix-requirements> (data di cons. 20/06/17)
29. Benjamin Stabler, *Specification*  
<https://github.com/osPlanning/omx/wiki/Specifications> (data di consultazione 20/06/17)
30. Benjamin Stabler, *Team*  
<https://github.com/osPlanning/omx/wiki/Team> (data di consultazione 22/06/17)
31. Benjamin Stabler, *TransCAD*  
<https://github.com/osPlanning/omx/wiki/TransCAD> (data di consultazione 12/12/17)
32. Benjamin Stabler, *Visum*  
<https://github.com/osPlanning/omx/wiki/Visum> (data di consultazione 07/12/17)
33. Caliper, *About Caliper*  
<https://www.caliper.com/ovuabout.htm> (data di consultazione 14/12/17)
34. Caliper, *TransCAD GISDK Add-Ins*  
<https://www.caliper.com/transcad/addins.htm> (data di consultazione 12/12/17)
35. Caliper, *TransCAD Transportation Planning Software. Overview*  
<http://www.caliper.com/tcovu.htm> (data di consultazione 13/11/17)
36. Caliper, *TransCAD Transportation Planning Software. Transportation Analysis Tools*  
<https://www.caliper.com/transcad/applicationmodules.htm> (data di consultazione 12/12/17)
37. Citilabs, *About Citilabs*  
<http://www.citilabs.com/about/> (data di consultazione 14/12/17)
38. Citilabs, *CUBE. Transportation & Land-use Modeling*  
<http://www.citilabs.com/software/cube/> (data di consultazione 16/11/17)
39. Città di Torino, *GTT Gruppo Torinese Trasporti (ex ATM Azienda Torinese Mobilità)*  
<http://www.comune.torino.it/servizionline/schede/userTorinoE.php?context=torinoE&submitAction=homeIndice&cid=824&cidRoot=134&refLanguage=it> (data di consultazione 11/07/17)
40. Consiglio Regionale del Piemonte, L.R. 4 gennaio 200, n. 1 *norme in materia di "Trasporto pubblico locale"*, in attuazione del D.L. 19 novembre 1997, n. 422. Art. 8, Consorzio per la mobilità nell'ambito metropolitano torinese
41. Creative Commons, *BSON*  
<http://bsonspec.org/> (data di consultazione 16/11/17)
42. Creative Commons, *QGIS*

## FONTI BIBLIOGRAFICHE

- <https://www.qgis.org/it/site/about/index.html> (data di consultazione 16/06/18)
43. CWI, *About CWI*  
<https://www.cwi.nl/about> (data di consultazione 13/02/18)
  44. Federal Highway Administration, *Freight Analysis Framework*  
[https://ops.fhwa.dot.gov/freight/freight\\_analysis/faf/](https://ops.fhwa.dot.gov/freight/freight_analysis/faf/) (data di consultazione 08/02/18)
  45. Ferrovie dello Stato Italiane, *Trenitalia*  
<https://www.fsitaliane.it/content/fsitaliane/it/il-gruppo-fs/societa-del-gruppo/trenitalia.html> (data di consultazione 11/07/17)
  46. FILExt, *.PICT File*  
<http://filext.com/file-extension/pict> (data di consultazione 07/11/17)
  47. Geoportale del Comune di Torino, *Azzonamenti statistici*  
SHAPEFILE DELLA CARTOGRAFIA DEL COMUNE DI TORINO  
[http://www.comune.torino.it/geoportale/azz\\_statistici.htm](http://www.comune.torino.it/geoportale/azz_statistici.htm) (data di consultazione 06/11/17)
  48. GNU Octave, *About/History*  
<https://gnu.org/software/octave/about.html> (data di consultazione 09/11/17)
  49. Greg Roelofs, *Zlib*  
<http://www.zlib.net/> (data di consultazione 29/11/17)
  50. HBA Specto Incorporated, *PECAS*  
<http://www.hbaspecto.com/pecas/> (data di consultazione 20/11/17)
  51. HDDEV.net, *Corso Python: Impara a programmare in Python*  
<https://hddev.net/corso-python-impara-a-programmare-in-python/> (data di consultazione 16/11/17)
  52. HDF Group, *HDF5 Attributes*  
[https://support.hdfgroup.org/HDF5/doc1.6/UG/13\\_Attributes.html](https://support.hdfgroup.org/HDF5/doc1.6/UG/13_Attributes.html) (data di consultazione 06/12/17)
  53. IAU, *IAU FITS Working Group*  
<https://fits.gsfc.nasa.gov/iaufwg/> (data di consultazione 07/11/17)
  54. IBM100, *FORTRAN. The Pioneering Programming Language*  
<http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/fortran/> (data di consultazione 09/11/17)
  55. INRO, *About INRO*  
<https://www.inrosoft.com/en/about-inro/> (data di consultazione 07/12/17)
  56. INRO, *Emme*  
<https://www.inrosoft.com/en/products/emme/> (data di consultazione 13/11/17)
  57. Istat, *Confini delle unità amministrative a fini statistici*  
SHAPEFILE DELLA CARTOGRAFIA DEI COMUNI ITALIANI  
<https://www.istat.it/it/archivio/124086> (data di consultazione 06/11/17)
  58. Java, *Che cos'è la tecnologia Java e a cosa serve?*  
[https://www.java.com/it/download/faq/whatis\\_java.xml](https://www.java.com/it/download/faq/whatis_java.xml) (data di consultazione 09/11/17)



## FONTI BIBLIOGRAFICHE

59. Julia Org, *Julia – Home*  
<https://julialang.org/> (data di consultazione 09/11/17)
60. MathWorks, *MATLAB – Presentazione*  
<https://it.mathworks.com/products/matlab.html> (data di consultazione 08/11/17)
61. National Centre for Supercomputing Applications, *About NCSA*  
<http://www.ncsa.illinois.edu/about> (data di consultazione 28/07/17)
62. National Science Foundation, *About the National Science Foundation*  
<https://nsf.gov/about/> (data di consultazione 09/11/17)
63. NetCDF, *An Introduction to NetCDF*  
[https://www.unidata.ucar.edu/software/netcdf/docs/netcdf\\_introduction.html](https://www.unidata.ucar.edu/software/netcdf/docs/netcdf_introduction.html) (data cons. 18/11/17)
64. NetCDF, *FAQ*  
<https://www.unidata.ucar.edu/software/netcdf/docs/faq.html> (data di consultazione 18/11/17)
65. PTV Group, *Our Story*  
<http://www.ptvgroup.com/en/our-story/> (data di consultazione 16/11/17)
66. Puget Sound Regional Council, *Home*  
<https://www.psrc.org/> (data di consultazione 09/11/17)
67. Python, *PyPI – the Python Package Index*  
<https://pypi.python.org/pypi> (data di consultazione 14/02/18)
68. Python, *Scarica Python*  
<https://www.python.it/download/> (data di consultazione 22/06/17)
69. R Development Core Team, *What is R?*  
<https://www.r-project.org/about.html> (data di consultazione 09/11/17)
70. Ruby, *Ruby è...*  
<https://www.ruby-lang.org/it/> (data di consultazione 20/11/17)
71. San Francisco County Transportation Authority, *About the Transportation Authority*  
<http://www.sfcta.org/about-transportation-aauthority> (data di consultazione 09/11/17)
72. Scilab, *History*  
<https://www.scilab.org/en/scilab/history> (data di consultazione 09/11/17)
73. SQLite, *About SQLite*  
<https://www.sqlite.org/about.html> (data di consultazione 19/11/17)
74. Swaroop C H, *A Byte of Python*, 25/01/2013
75. Systematica, *CUBE*  
[http://www.systematica.net/?page\\_id=3067](http://www.systematica.net/?page_id=3067) (data di consultazione 13/11/17)
76. The HDF Group, *About Us – Our Company*  
<https://www.hdfgroup.org/about-us/> (data di consultazione 26/07/17)

## FONTI BIBLIOGRAFICHE

77. The HDF Group – Support, *Downloads*  
<https://support.hdfgroup.org/downloads/> (data di consultazione 09/11/17)
78. The HDF Group – Support, *HDF5 Technologies*  
[https://support.hdfgroup.org/about/hdf\\_technologies.html](https://support.hdfgroup.org/about/hdf_technologies.html) (data di consultazione 08/11/17)
79. The HDF Group – Support, *What is HDF5?*  
<https://support.hdfgroup.org/HDF5/whatishdf5.html> (data di consultazione 26/07/17)
80. The National Academies of Sciences, Engineering and Medicine, *Get Involved With TRB*  
<http://www.trb.org/GetInvolvedwithTRB/GetInvolvedwithTRB.aspx> (data di cons. 17/11/17)
81. The National Academies of Sciences, Engineering, and Medicine, *Our Program Units*  
<http://www.nationalacademies.org/nasem/> (data di consultazione 17/11/17)
82. TPS, *Campi di applicazione*  
[http://www.tpsitalia.it/software/ingegneria\\_traffico/visum\\_campi\\_applicazione.php](http://www.tpsitalia.it/software/ingegneria_traffico/visum_campi_applicazione.php) (data di consultazione 07/12/17)
83. TPS, *Introduzione*  
[http://www.tpsitalia.it/software/ingegneria\\_traffico/visum\\_introduzione.php](http://www.tpsitalia.it/software/ingegneria_traffico/visum_introduzione.php) (data di cons. 07/12/17)
84. Travel Forecasting Resource, *Open Matrix Format*  
[http://tfresource.org/Open\\_Matrix\\_Format](http://tfresource.org/Open_Matrix_Format) (data di consultazione 22/06/17)
85. TRB Transportation Research Board, *Innovation in Travel Modelling*  
<http://www.cvent.com/events/innovations-in-travel-modelling/event-summary-477f7b8596454859ac120f90d9ebc56c.aspx> (data di consultazione 22/06/17)
86. Unidata, *What is Unidata?*  
<https://www.unidata.ucar.edu/about/tour/> (data di consultazione 22/01/18)
87. W3C, *WebCGM Profile*  
<https://www.w3.org/Graphics/WebCGM/> (data di consultazione 07/11/17)
88. Wikipedia, *ASCII*  
<https://it.wikipedia.org/wiki/ASCII> (data di consultazione 30/01/18)
89. Wikipedia, *BSON*  
<https://en.wikipedia.org/wiki/BSON> (data di consultazione 16/11/17)
90. Wikipedia, *C (linguaggio)*  
[https://it.wikipedia.org/wiki/C\\_\(linguaggio\)](https://it.wikipedia.org/wiki/C_(linguaggio)) (data di consultazione 09/11/17)
91. Wikipedia, *C#*  
[https://it.wikipedia.org/wiki/C\\_sharp](https://it.wikipedia.org/wiki/C_sharp) (data di consultazione 09/11/17)
92. Wikipedia, *C++*  
<https://it.wikipedia.org/wiki/C%2B%2B> (data di consultazione 09/11/17)

## FONTI BIBLIOGRAFICHE

93. Wikipedia, *Cross-platform*  
<https://en.wikipedia.org/wiki/Cross-platform> (data di consultazione 27/01/18)
94. Wikipedia, *External Data Representation*  
[https://en.wikipedia.org/wiki/External\\_Data\\_Representation](https://en.wikipedia.org/wiki/External_Data_Representation) (data di consultazione 23/01/18)
95. Wikipedia, *Haskell (linguaggio)*  
[https://it.wikipedia.org/wiki/Haskell\\_\(linguaggio\)#cite\\_note-1](https://it.wikipedia.org/wiki/Haskell_(linguaggio)#cite_note-1) (data di consultazione 23/01/18)
96. Wikipedia, *Hierarchical Data Format*  
[https://en.wikipedia.org/wiki/Hierarchical\\_Data\\_Format](https://en.wikipedia.org/wiki/Hierarchical_Data_Format) (data di consultazione 26/07/17)
97. Wikipedia, *Interactive Data Language*  
[https://it.wikipedia.org/wiki/Interactive\\_Data\\_Language](https://it.wikipedia.org/wiki/Interactive_Data_Language) (data di consultazione 09/11/17)
98. Wikipedia, *NetCDF*  
<https://en.wikipedia.org/wiki/NetCDF> (data di consultazione 18/11/18)
99. Wikipedia, *Perl*  
<https://it.wikipedia.org/wiki/Perl> (data di consultazione 23/01/18)
100. Wikipedia, *PostgreSQL*  
<https://it.wikipedia.org/wiki/PostgreSQL> (data di consultazione 27/01/18)
101. Wikipedia, *SQLite*  
<https://it.wikipedia.org/wiki/SQLite> (data di consultazione 19/11/17)
102. Wikipedia, *Wing IDE*  
[https://en.wikipedia.org/wiki/Wing\\_IDE](https://en.wikipedia.org/wiki/Wing_IDE) (data di consultazione 13/02/18)
103. Wolfram, *Mathematica*  
<https://www.wolfram.com/mathematica/> (data di consultazione 09/11/17)