Migration visualization

Oerlemans, G.G.

Award date:
2012

Statistics Netherlands
Divisie Methodologie en Kwaliteit
Sector Methodologie Den Haag

P.O. Box 24500
2490 HA The Hague
The Netherlands

Migration Visualization

Giel Oerlemans

Remarks:
The views expressed in this paper are those of the author and do not necessarily reflect the policies of Statistics Netherlands.

Project number:
BPA number:
Date: *April 24, 2012*

MIGRATION VISUALIZATION

Abstract: Demographic experts at CBS are interested in migration. Currently, analysis of migration data is a time-consuming process. The migration data set contains both information about the regions between which people migrate, and the migrants themselves. A combination of existing, easy to interpret visualizations is used to get insight in the data and speed up the exploration and analysis of migration data. Aggregation of the regions and filtering of the people are used to cope with the scale of the data. The user is enabled to interactively explore the data by using a familiar representation of a country: a map. Different graphs and a scatterplot show additional information about the inspected regions and the people moving in or out. A clustering algorithm is employed to seek for patterns in the data, by grouping similar regions based on attributes like age or income distribution, or by grouping them based on the migration data. A matrix visualization is used to analyze or validate the clustering. The implementation is demonstrated using migrations between municipalities in The Netherlands. Other types of flow data can be supported using minor adjustments. The targeted audience are experts in migration data analysis and demographics, to help with the analysis of migration data, but also any interested novice to the field.

Contents

Chapter 1

Introduction

Each year, a lot of people move to a new place, a phenomenon referred to as *migration*. All around the world, for a variety of reasons, people change homes. With migration, not only people are moving out of some area and settle in some other area. A lot comes with them, like income and knowledge. Therefore, migration has a direct and significant effect on the prosperity and development of the regions people migrate between. This makes migration an interesting field of study for demographics experts.

However, there is a lot of data to process. We will focus on the Netherlands, in which around 10% of the population migrates each year, that is around one and a half million people. Most of them do not move very far and stay within the same municipality. The rest, around 600,000 people, migrate between the 400+ municipalities. Statistics Netherlands (CBS) is interested in analyzing migration and its impact on different parts of the country.

The dataset is provided by the Statistics Bureau and consists of a list of all Dutch inhabitants from 2005 to 2009. For each person a lot of data is available, like the place they lived for the past years, gender, date of birth, ethnicity, monthly income, etc. Studies so far have been quite time-consuming as regions often have to be statistically analyzed one by one. Also, because the great number of flows (between around 60,000 pairs of municipalities), migrants are considered as one big bag of people moving out of or in to some area. Their origin or destination is not considered.

To help the expert analysts at CBS to explore and analyze the migration data, a visualization tool is build. In cooperation with the social geographers and demographers at the Statistics Bureau, a set requirements is formed. These will be presented in Chapter 2, together with some background knowledge of the Netherlands. Next, existing solutions to the problem of migration visualization are summarized in Chapter 3.

Migration can be considered as flow, in this case of people moving between cities, towns, municipalities, provinces and countries. But the ideas and techniques described in the remainder of this text may be applied to a broader spectrum of problems. Therefore, the problem is formalized in Chapter 4, in which also the data model is described.

Having a formal model to work with, the solution is presented in Chapter 5. The tool that has been build is presented, some general principles used are given, and the individual visualizations that are used are elaborated on, discussing both the semantics of the visualization as well as implementation details.

Chapters 6 and 7 provide more detail about some elements of the implementation, namely aggregation and filtering, and clustering. In Chapter 8, some previous studies are replayed and evaluation of the tool by experts is described. Some cases are described and different images produced by the tool confirm the hypotheses of the cases.

Because the problem is complex, there is a lot of future work that can be done. This is described in Chapter 9. Finally, we conclude in Chapter 10. In the appendices, some implementation details, and details about the used libraries can be found.

Chapter 2

Problem

In this chapter, the goal of the project is formulated (Section 2.1), and translated to a list of functional requirements in Section 2.2. Section 2.3 provides some background information on the topography of the Netherlands.

2.1 Goal

The project was carried out at the Statistics Netherlands, CBS (Centraal Bureau voor de Statistiek). CBS is the Dutch national institute for gathering, analyzing and publishing statistics for the governments, science and business. Data is gathered about inhabitants, institutions and companies.

Demographers are studying the size, structure and spread of the population over a country and are interested in the way these things change over time. Besides births, deaths and aging, also migration causes these changes to occur. Groups of young people tend to migrate quite a lot to get educated for instance. This causes shifts in the age distribution in both the place of departure and the place of establishment. After finishing their study, students leave their dorm and move back to the countryside, or some other city to find a job.

Studies are performed on the entire society, or groups of people based on certain criteria such as education, nationality, religion, etc. The students from the previous example form such an interesting group. But demographers are interested in all kinds of social groups. The spread of different ethnic groups over the country is an ever returning topic.

Social demographers interpret statistics, find explanations for a certain process and try to oversee the consequences for society. To do that properly, they need to perform all kinds of statistical analysis on the migration data available. From the raw data, that states where someone lived per moment of sampling, aggregates are calculated per region. Next, regions are compared one by one. To cope with huge amounts of data, the analysis is performed on large aggregated areas. Extra steps are required to make images from the calculated results.

To speed up the process of finding interesting patterns in the migration data and answering questions about the data, visualization can be of great aid. The goal of the project is to develop a visualization tool to help analyze migration in the Netherlands.

2.2 Requirements

Requirements have been formulated in collaboration with CBS' demographic and migration experts. A total of 7 experts have collaborated in the early stages of the project during various group meetings. Another 3 have helped in latter stages to improve usability. The resulting list of tasks the user should be enabled to perform, and requirements on the tool to be developed, are presented here.

The tool should support users in answering both qualitative and quantitative questions, but also enable the user to explore the dataset. The tool is intended to be used by both experts in the field of demographics (and migration in particular) and people with less affinity for this field. This means that the tool should be easy to use and offer comprehensible but powerful visualizations. Experts should be enabled to extract the image they 'want to see' by manipulating all kinds of settings and parameters.

There are three main categories of requirements ([16]) with respect to the information exposed by the tool. The user should be enabled to:

1. See the data (quantitative)
2. See the data in combination with other variables (qualitative)
3. Discover patterns in the data (discovery)

To help accomplish the three tasks denoted above, the following requirements should be met:

1. Show the growth/shrinkage, inflow and outflow per region.
2. Show all migrations between all pairs of regions at the same time.
3. Show for a particular region the net flow, inflow and outflow from and to all other regions.
4. Show the *most important* neighbor of a certain region in terms of migration.
5. Filter the set of persons, i.e. filter on a certain attribute or combination of attributes of a person.
6. Show information about regions, including the size of the population, age distribution, income distribution and other (demographic) statistics.
7. Show the *effect* of migration on a region's properties such as age-distribution.
8. Discover the main migration pattern or direction, if one exists, using the region hierarchy.
9. Find the region with the highest growth/shrinkage.
10. Find groups of nodes with similar migration patterns.
11. Relate growth to a property of a region.

12. Support the multiple regional divisions of the Netherlands, often used by experts at the CBS.

A brief look at the requirements indicates that the *visualization mantra* of Shneiderman [14]: "overview, zoom and filter, details-on-demand" is applicable. We must be able to show an *overview* of the data, which is captured by requirements 1,2 and 8.

To *zoom* into the data, the user must be given the opportunity to filter the data records and look at a strict subset. This can be done in different directions.

As some requirements already suggest (3, 4), it should be possible to look at a region, or set of regions, in isolation. We refer to this process as *selecting*. Also, we may choose to look at the data differently, by using different visualizations or by displaying different values in the same visualization.

Another possibility is to look at a subset of all persons. This can be done by filtering persons on their attributes. For instance, we could only include people between the age of 65+, which is the group of retired persons in the Netherlands. Restricting the group of people looked at is referred to as *filtering*.

Our final type of zooming is *aggregation*. Aggregation exploits the hierarchy that exists between the different region types. Municipalities' migration data can be aggregated for their parent region (the COROP), reducing the amount of regions by a factor ten. Attributes concerning people and municipalities may be aggregated as well. A classical example is aggregating age information in *bins* of 5 or 10 years.

Algorithmic support Requirement 10 states that the tool should allow the user to find groups of nodes with similar behavior. What that exactly means is discussed later on in Chapter 7. The requirement suggests that some algorithmic support must be provided to perform this grouping, or *clustering*.

Scalability As with many visualization problems, the solution has to scale well in terms of performance. In our particular example, we should at least be able to analyze all migrations between two sample moments in an interactive way. This boils down to roughly 10 percent of the entire population, per year, including internal migrations, i.e., around one and a half million movements. If the solution is tested on bigger countries or even continents or the entire world, scalability would become even more important. To keep things manageable, interactivity should be accomplished for the provided dataset only.

Another aspect of scalability is that for the given amount of data, the visualization should be visually readable and understandable. In the next chapter we therefore consider existing solutions for the visualization of such data.

2.3 Topography of the Netherlands

So far, we have only considered *regions* in general. The Netherlands has multiple levels of regional divisions however, most of which are also used by demographic

experts to study certain phenomena (including migration). So far, most studies have performed on a level that divides the country in 40 regions, or at a more fine-grained level for an isolated part of the country. One of the requirements of the visualization is to be able to study nation-wide migration on a finer level. The coarser regional divisions must also be available to provide natural options of aggregation, which helps in identifying trends and structure in the data [1]. We now look at the topography of the Netherlands and the multiple regional divisions that are widely used across all sections of CBS:

1. *gemeentes* (municipalities) 400,
2. *COROP-areas* or *COROP*s 40,
3. *provincies* (provinces) 12.

The first and third are trivial, the middle layer needs some explanation. COROP stands for "Coördinatie Commissie Regionaal OnderzoeksProgramma" in Dutch, which are groups of municipalities. There are 40 of them and they mostly consist of a central core, which is a city, combined with the surrounding *service area*. The layout of COROP areas is based on, for example commuters traffic. In some cases, this principle is violated because the COROP's borders have to match the provincial borders. To be able to make consistent statistics over the year, the COROP grouping has not changed since 1971. Municipalities have been merged and split ever since. Since 1985, the number of provinces in the Netherlands is 12, divided into a total of 40 COROP areas, which are again divided into a total number of 415 municipalities at January the first, 2012. These numbers do not take the Caribbean Islands of Bonaire, Sint Eustatius and Saba into account.

All regions have a unique code and a name, which is not necessarily unique. There are municipalities in different provinces with the same name and there are municipalities, COROP-areas and provinces that carry the same name. The codes of municipalities were originally assigned in series within provinces. With the assignment of new codes for new municipalities, this is not the case any more. The code consists of 4 digits with leading zeros. When a municipality disappears after a reclassification, it keeps its code. When a municipality keeps its name after a reclassification, the code is also maintained. This has to be considered when working with real data which span multiple years, and thus, multiple regional divisions.

Hierarchical relation between regions

There exists a hierarchical relation between the three levels. Each municipality lays in exactly 1 COROP-area and each COROP-area lays in exactly 1 province. This hierarchy can be captured using a tree structure, the *region hierarchy*.

Each level of regional division adds to the *height* of the tree, which in our case is 3 (municipalities at level 0, COROP-regions at level 1 and provinces at level 2).

The *root* of the tree is the object of interest, in our case The Netherlands. Municipalities form the lowest level of the tree, the *leaves*.

Figure 2.1: The Netherlands, divided into three layers of regions, the municipalities, COROP-areas and provinces

The terms *parent* and *child* are used to indicate the relations between regions on different levels of regional division. Regions on the same level are called *peers*. Regions with the same parent are referred to as *siblings*.

The data describe residences in terms of municipalities. We are also interested in other levels in the region hierarchy, which is exploited by the use of partitions, which is explained in Section 4.2.

Chapter 3

Related work

There exist some methods to visualize flow data or origin-destination data. This chapter shows some of those techniques that might be applied for the provided case, but that do have several shortcomings. We start with more general models, such as the node-link diagram, and end with more specific visualizations that were developed to explicitly visualize migration data.

3.1 Node link diagram

Migration data can be considered as a graph, with the regions being nodes and the migrations being directed edges. A natural way to visualize a graph is by drawing a node-link diagram. Due to the scale of the dataset, this would result in highly cluttered and unreadable images, as can be seen in Figure 3.1. Clutter reduction techniques, such as force-directed graph layouts are not sufficient to cope with around sixty thousand edges between more than four hundred nodes. Besides, they would not preserve the geographical layout of the nodes, which makes it harder to relate a 'node' to a region on the map.

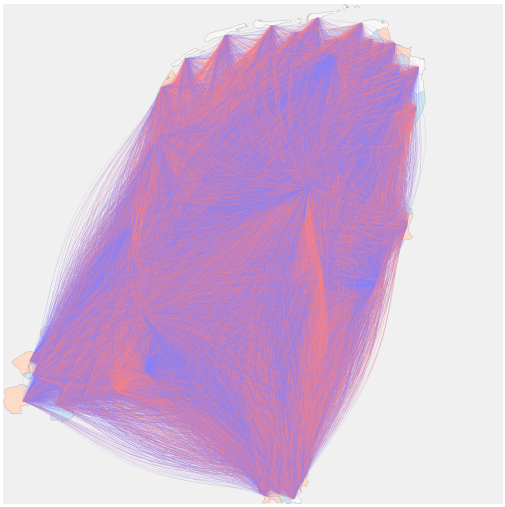


Figure 3.1: Visualizing the dataset as node-link diagram would result in highly cluttered images. Here, around 16,000 edges (10% of all migrations) are drawn over the map of the Netherlands.

3.1.1 Edge bundling

Another way to reduce clutter is by the use of edge-bundling using the approach of Holten and van Wijk [8]. We could try to bundle the edges of Figure 3.1 while preserving the map-layout. This would result in an image like that of Figure 3.2. As one can see, important hubs are easy to distinguish, especially in Figure 3.2 d.

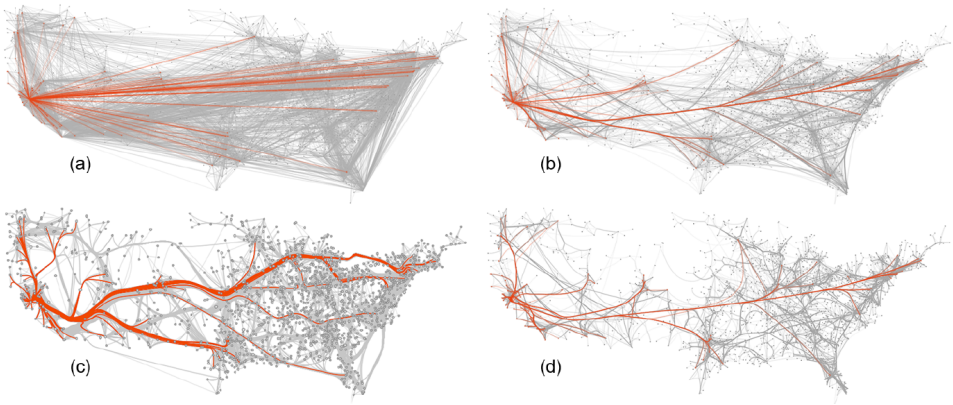


Figure 3.2: Edge bundling applied to migration data of the US (1715 nodes and 9780 edges). From Holten [8].

Although variants b, c and d are already way clearer than image a, some difficulties still remain. First of all, it is hard to capture the bi-directionality of the graph. Capturing direction in a node-link diagram is not trivial, as is shown by Holten and van Wijk [9]. Also, not all edges contain the same number of migrants. Bundling edges makes it very hard to use the thickness of an edge to represent a variable. The other alternative, coloring, could conflict with the goal of showing direction. Another question to ask is whether this method would scale well to over 50,000 edges.

3.2 Spiral trees

In a recent paper, Verbeek, Buchin and Speckmann [4] suggest the use of *spiral trees* to visualize flow data on maps. The technique generates nice images that capture the main direction and magnitude of a flow, which are very useful for presentation purposes. It is however limited to focusing on one or a couple of areas, for which the flows from or to other regions are shown. Instead of drawing difficult spiral trees, which take some time to compute, simply coloring the map's regions provides the user with the same information in a more basic visualization, that can be rendered immediately, thus allowing the user to quickly switch the focus to another area.

3.3 Chloropleth map

The chloropleth map is a map in which each area is colored according to some color map and scalar value. It has the same limitation as Spiral Trees, namely that it is not possible to show all migration data at the same time, but need to lay focus on some area and color the rest. Besides that, the chloropleth map can be used to display other

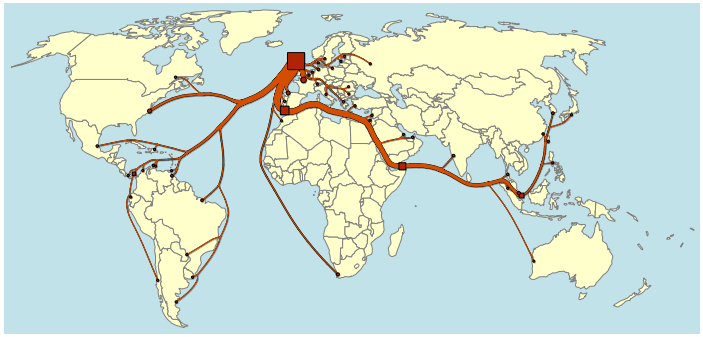


Figure 3.3: Spiral tree showing the whisky export from Scotland. Taken from [4].

(scalar) variables per region. Rendering a chloropleth map is cheap and can therefore be used for interactive visualizations.

Because the chloropleth map is only capable of showing one variable at a time and can only show migration data with respect to some selected area at a time, it needs to be accompanied by other (linked) visualizations to offer more information.

3.4 Flowstrates

Demographic experts mentioned in interviews that they prefer the nodes layout to resemble the real-life geographic coordinates. Therefore it seems obvious that a map should be included in the visualization. A visualization that does that are the Flowstrates of Boyandin et al. [2]. The idea is to put two maps (which may or may not be identical) besides each other and draw arrows from one map, containing all origins, to the other, containing all destinations. These arrows are not straight lines but are interrupted between the two maps. For each arrow, a row is available on which a temporal variable is shown. In the paper of Boyandin, this variable varies over time, but it might as well contain another quantity.

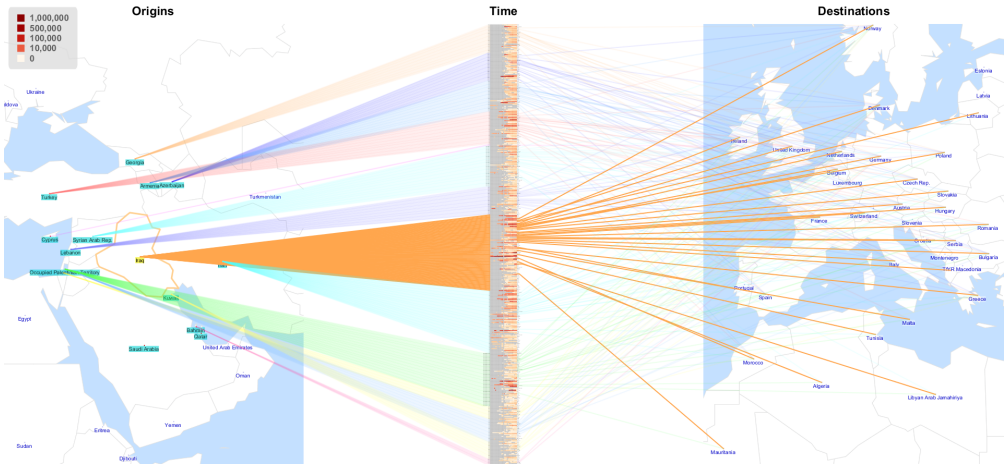


Figure 3.4: An example of Flowstrates, from [2], showing the flow of people from the Middle-East to Europe. Flows from Iran are highlighted in orange.

Although the idea of the two maps may be practical to overcome the problem of arrow

directions, it does not resolve the problem of showing edge-weights. We could show this on the rows between the two maps as an alternative. But for this view to be readable, only a limited amount of edges should be shown, for instance all edges for only one origin or for one destination. This is due to the fact that each edge is displayed on a single row and 60.000 edges simply do not fit on a screen. As can be seen in Figure 3.4, focusing on 15 origins and 35 destinations (maximum of 525 edges) gets the visualization running out of screen space already.

3.5 OD Maps

Another existing method we have looked at are the OD Maps from Wood, Dykes and Slingsby [19]. OD stands for Origins and Destinations, and the OD maps are meant to visualize flows. The technique is suited for large numbers of edges (up to 10^6) and does take flow quantities into account. The technique rasterizes the area into rectangles and prints the origins as such a raster with in each cell the complete raster again, depicting the destinations, see Figure 3.5.

For the scale of COROP-areas, this may be very suited, but in case of municipalities, rasterizing will cause a lot of trouble. The municipalities vary greatly in size and they can also have strange shapes, holes and span multiple disconnected areas. Maintaining a balance between geographic familiarity, cell interpretability and space efficiency, important for the understandability of the OD Maps, is thus very difficult at the scale of municipalities.

Another idea that is used in an alternative OD map visualization, is to show the deviation from a certain expected value per edge, instead of the quantity. This approach is adopted and is explained in Chapter 5.

3.6 Matrix visualization

Another approach is to consider the migration data as a table, where each cell denotes the flows from one region to another, i.e., a Migration Matrix. This matrix can be shown directly. Color values are assigned according to the value of the cells. For smaller numbers of areas, such as the 40 COROP-areas, this is a good way to start exploring the data. Figure 3.6 shows an example of such a visualization. One can immediately see that the greatest flows are along the diagonal, hence between municipalities within the COROP-areas. Apparently people tend to migrate to nearby municipalities. The matrix falls short in many ways however. It is hard to see the net result between two different areas as the symmetric fields' colors have to be compared, which is (depending on the used color map) hard to do. Also, we are unable to see whether regions are growing or shrinking. And even if we would be able to distinguish the values in the columns and rows, the picture is greatly influenced by the used normalization. Finally, the geographic layout of the nodes, shown in Figure 2.1 is not captured by the matrix.

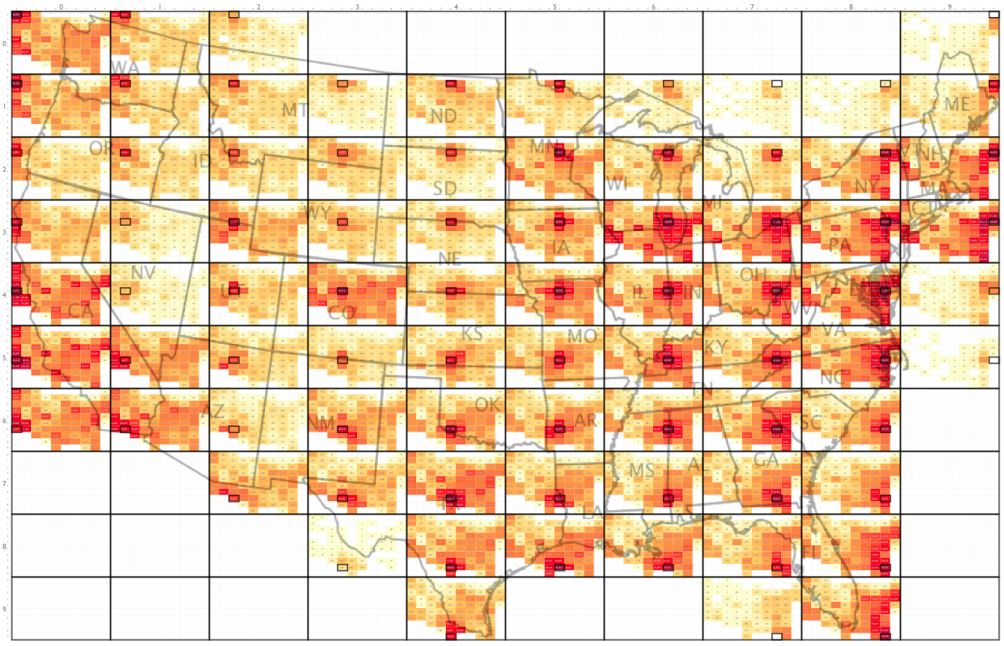


Figure 3.5: US county-county migration vectors (721,432) shown as an OD map, taken from [19]. Each large grid cell is an origin with all destinations drawn in it as smaller grid cells.

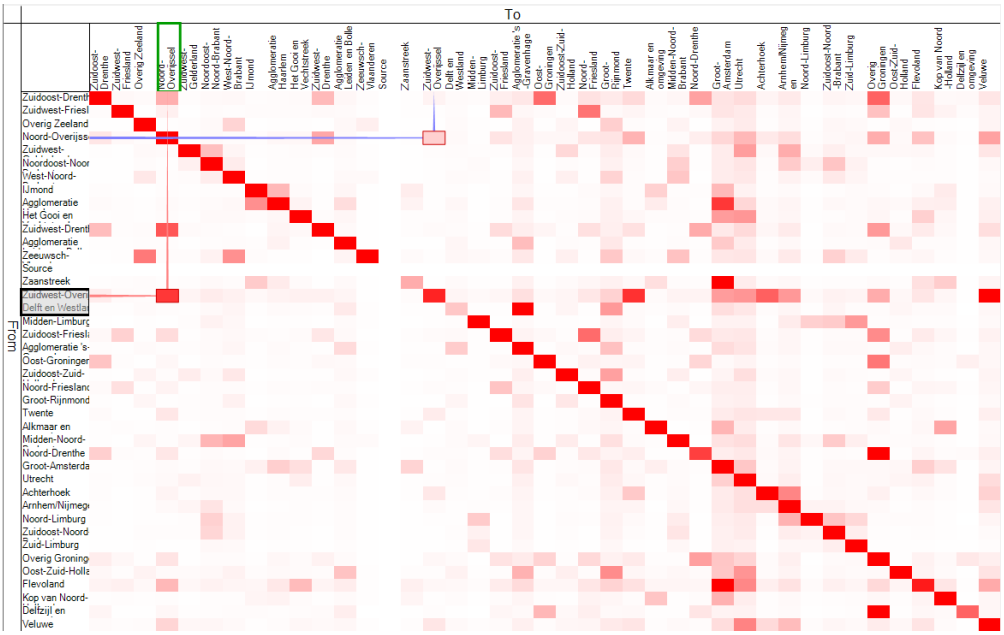


Figure 3.6: A matrix visualisation, showing the underlying 'raw' data.

Chapter 4

Data model

In this chapter, the dataset is elaborated on and a formal data model and definitions are given to describe the migration data.

4.1 The dataset

The dataset used for the project contains all Dutch inhabitants in the period from 2005 to 2009. Each line represents one person. The set of persons in the dataset is defined as follows:

Definition 4.1.1 (Inhabitants)**.**
$\mathscr{P}_{\text{NL}}$ is the set of all persons appearing in the dataset.

For each person, the residence municipality is registered at a certain reference date for each year. Due to the yearly residence information, it is possible to tell whether or not people have migrated by looking for differences from year to year. The residence municipality may be empty for a certain year due to births, deaths and immigration and emigration to foreign countries. Instead of looking at the residence per year, we chose to look at just one pair of years (two moments of sampling), to show the net effect of migration. From all these sampled residences, we can form a set of sampled regions.

Definition 4.1.2 (Set of sampled regions)**.**
$\mathscr{R} = \{a_1, a_2, \ldots, a_n\}$ is the set of all regions appearing in the dataset for the two selected years of sampling. Empty regions are included and are referred to as the *source* or *sink* region.

Besides the place of residence, a whole range of other data is available for each inhabitant. Each person in the dataset has a gender, age and annual income, is part of some type of household, belongs to a certain ethnic group, etc. Having this information available for all people living in a certain region or migrating between two regions opens a lot of opportunities to look at migration from a different perspective. Instead of looking at the number of people moving from a to b, we could look at the summed annual income, or the average age of people migrating. Another advantage of having this information available per person, is the possibility to filter on people's attributes to look at certain groups of people in isolation.

4.1.1 Human migration

People migrate from one region to another region if there is some change of residence. The regions are geographical entities, such as streets, cities, municipalities, provinces, states or countries.

People who move into a region are called *immigrants*, while they are called *emigrants* in the region they depart from. We refer to people moving within a certain region as *internal migrants* as they are both *immigrants* and *emigrants* in the same region at the same time. People who move between different regions are referred to as *external migrants*.

Because we consider various levels of regional divisions, an immigrant at one level, may be an internal migrant at another.

External migrants for the whole country are people that move abroad or come from another country. Due to the lack of information about these people's future or previous residence, they are collected into one artificial region that represents both a source and sink. In the remainder, people moving from or to other countries are referred to as foreign migrants.

Detecting migration

To detect a migration, the person must have changed his or her residence from one region to another region in between two moments of sampling, t_1 and t_2. Such a detection method has some limitations:

1. We cannot detect a person moving from a to b and back to a within our time-frame $[t_1, t_2]$, as both residences reported will be a.
2. In the current dataset, we can not detect a person moving within a region. We are bound to the level of detail of the data.

The first issue is hard to overcome and outside the scope of this project. The second limitation can be overcome by coupling more data sources. Also, this issue is somewhat limited to the lowest level of regional division. If we aggregate some of these sample points, there is *internal* migration possible within the group of sample points, between the sample points of the group.

As in this project only two moments of sampling are used, there are two residences per inhabitant. These are defined as follows:

Definition 4.1.3 (Sampled residences)**.**
For a *Person* $p \in \mathscr{P}_{\text{NL}}$, and a set of municipalities $\mathscr{R}$:

$r_1(p) \in \mathscr{R}$ is the region in which the person lived on the first moment of sampling.

$r_2(p) \in \mathscr{R}$ is the region in which the person lived on the second moment of sampling.

4.1.2 Preprocessing

The dataset needs to be preprocessed first to fit the tool. First of all, two reference dates are chosen. In the preprocessing step, two years are chosen by selecting two columns representing the residence. Furthermore, the columns representing age, income, gender need to be identified.

The preprocessed dataset still contains *all* inhabitants. When the preprocessed dataset is read in for the first time by the tool, another preprocessing step is performed. A new file of persons is made, in the same format, but with only people who migrate (where the first sampled residence is not equal to the second sampled residence). The rest of the people (the stayers) are aggregated and stored as region information in separate files. This leads to the following definition for the set of people used in the tool:

Definition 4.1.4 (Set of all persons)**.**
$\mathscr{P} \subseteq \mathscr{P}_{\text{NL}}$ is the set of all persons who were registered in the Netherlands for which $r_1(p) \neq r_2(p)$.

A preprocessed input line typically looks like this:

```
00484535;1680;0118;35010;37865;44;0
```

The first column is a unique person identifier, the second and third columns contain codes of regions. This particular person (00484535) migrated from region 1680 to region 0118. This man (last column indicates this is a male) of age 44 had an annual income of 35010 in the first year and 37865 in the second year.

4.2 The Data Model

The creation of the data model from the raw input data is summarized in Figure 4.1. The input is the dataset provided, containing data for several years. The preprocessing step selects the two years as defined in Definition 4.1.3. The dataset resulting from this preprocessing step is read in by the visualization tool. Lists of persons and regions are extracted. Filters are applied on the set of persons and regions are aggregated by creating a partition. From the filtered set of persons and the partition, the data model is generated, the *migration matrix*.

First, the method that is used to create set partitions to represent groups of areas is described. This can be used to represent the COROP and provinces regional divisions, but also user defined, custom aggregations of areas.

4.2.1 Set Partitioning

To group sampled entities into geographic areas, set partitions are used. Set partitions are defined as follows:

Definition 4.2.1 (Set Partition)**.**
A *partition* $B = \{B_1, B_2, \ldots, B_m\}$ of the set A, is a set of *non-empty* subsets of A such

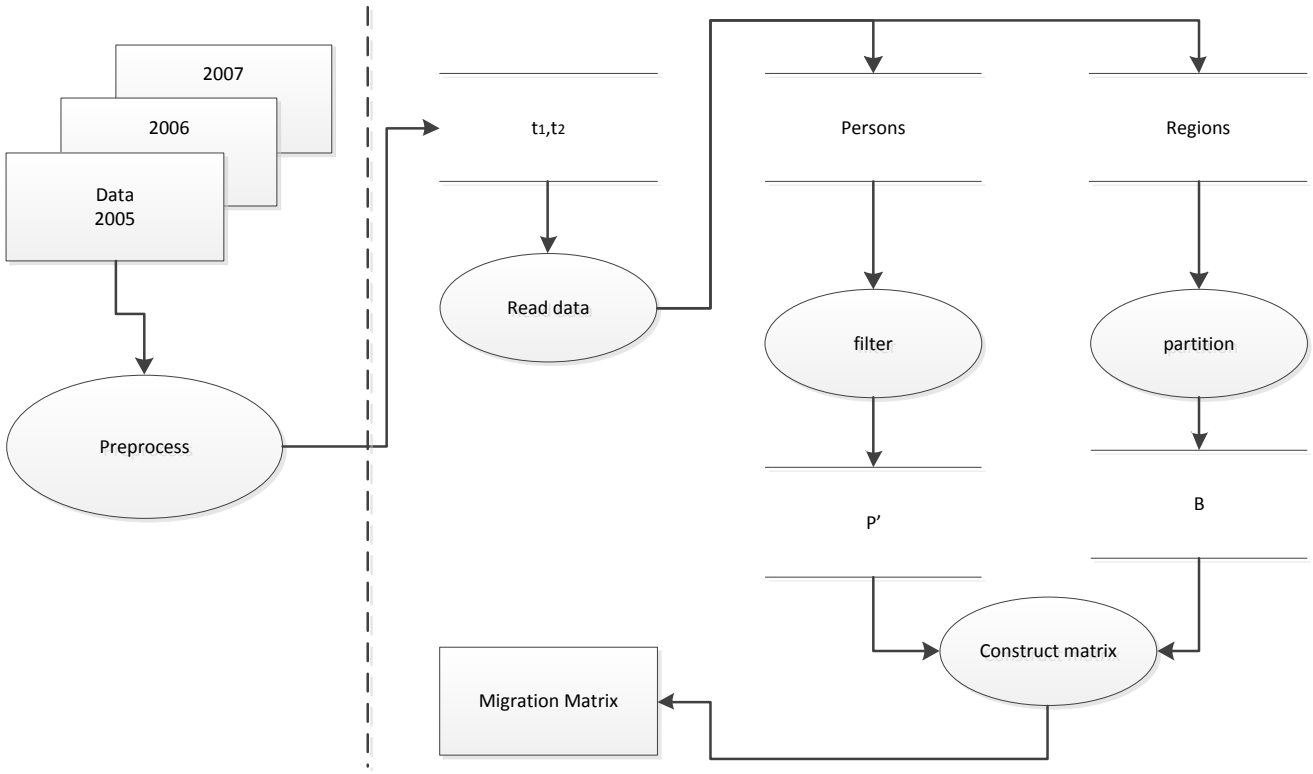


Figure 4.1: Schema describing the migration matrix construction

that:

$$\bigcup_{i=1\ldots m} B_i = A$$
$$i \neq j \Rightarrow B_i \cap B_j = \emptyset$$

The elements B_i of a partition B are called blocks, cells or parts. For the remainder of this document we also refer to them as areas or regions. If the sampled regions are meant (the municipalities, appearing in the raw dataset), this is explicitly mentioned.

For the sampled areas, certain properties, such as a population, age distribution, income distribution, are available. Each block of the partition has the same properties available. These are defined as the union of the properties of the elements of the block.

Definition 4.2.2 (Properties of partition' blocks)**.**
The *population* of block B_i:

$$S_i = \sum_{a \in B_i} (\text{population}(a))$$

Definition 4.2.3 (Partition refinement)**.**
Any partition C of A is a *refinement* of partition B of A, if $(\forall_i : (\exists_j : C_i \subseteq B_j))$. Or, every element of C is a subset of some element of B.
We say that C is *finer than* B, B is *coarser* than C, or $C \leq B$.

As defined in Definition 4.1.2, $\mathscr{R}$ is the set of all sampled regions. Every partition B of set $\mathscr{R}$ is thus a grouping of the sampled regions, in our case municipalities.

We already have several of these partitions at our disposal, namely the sets of municipalities, COROP-areas and provinces. These default partitions are defined as follows:

Definition 4.2.4 (Default partitions)**.**
$G = \{\{a_1\}, \{a_2\}, \ldots, \{a_n\}\}$ is the set of singletons of sampled regions (the lowest level of geographic areas), in this case the municipalities;
C contains sets of municipalities per COROP;
P contains sets of municipalities per province.

We can also make the partition $N = \{\{A\}\}$, which would represent the *country* as a whole.

Note that these default partitions are refinements of each other, thus $M \leq C \leq P \leq N$, which follows directly from the tree structure in the region hierarchy, mentioned in Section 2.3.

As we are working with partitions, it is easy to find out what partitions people are moving from and to, by checking whether their sampled residence is present in a certain cell of the partition.

Thus, we can divide a group of people according to the defined partition.

Definition 4.2.5. Given a partition B of $\mathscr{R}$ and a set of people $P' \subseteq \mathscr{P}$, the sets P_{ij} are defined as follows:

$$P_{ij} = \{p \in P' : r_1(p) \in B_i \wedge r_2(p) \in B_j\}$$

Note that the persons are not taken from the entire set $\mathscr{P}$ of migrants, but instead from a subset P'. This provides us with the opportunity to focus on different groups of people, or simply take a sample set of the original data. Also, people who do not migrate are not included in the sets P_{ij}.

At COROP and province levels, sets P_{ii} are not necessarily empty as B_i might be equal to B_j while $r_1(p) \neq r_2(p)$. In the case of COROP-areas, the diagonal is filled with people migrating from a municipality in the COROP to a different municipality within the same COROP.

4.2.2 The Migration Matrix

All people moving from one region to a different region ($r_1(p) \neq r_2(p)$) make up our set of *migration data*.

Migration data can be modeled as a graph with regions as nodes. The nodes correspond to the regions from the region hierarchy. Directed edges exist between the nodes for each person moving from one node to another. This movement is one single migration. Of course, multiple people can move between the same pair of nodes, leading to multiple edges with the same source and target.

For conciseness, the migration data is modeled as a matrix instead of a graph. Each cell of the matrix contains information about migrants moving from an area indicated by the row index, to an area indicated by the column index.

The definition is as follows:

Definition 4.2.6 (Migration Matrix)**.**
Given a partition B of $\mathscr{R}$ and a set of people $P' \subseteq \mathscr{P}$, the *Migration Matrix* $M(B,P')$ is an $m \times m$ matrix, in which $m = |B|$, such that:
$M_{ij} = f(P_{ij})$

The function $f : \mathscr{P} \to \mathbb{R}$ that is used to calculate M_{ij} may differ. The size of P_{ij} represents the number of people migrating. As there is more information available about these people, other aggregates are possible, such as the summed income of all people. But also, some normalization can be applied.

Data schema

The data schema shown in Figure 4.2 results from the previous definitions. The building blocks for the Migration Matrix, which consists of exactly $n \times n$ cells, are the Persons and Blocks. Persons are related to Regions by the two sampled regions. Blocks contain at least one region. Information is stored per Region. This region information is aggregated for each Block. A Block is a generalization of the parts of the default partitions: G, C and P.

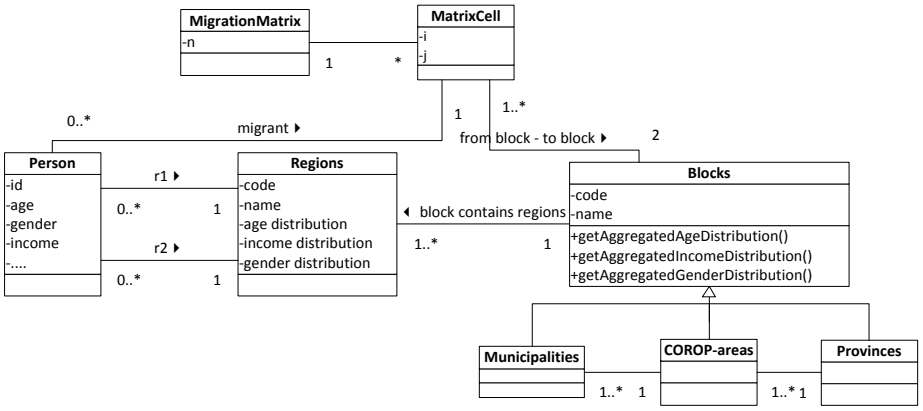


Figure 4.2: Data schema describing relations between the data items

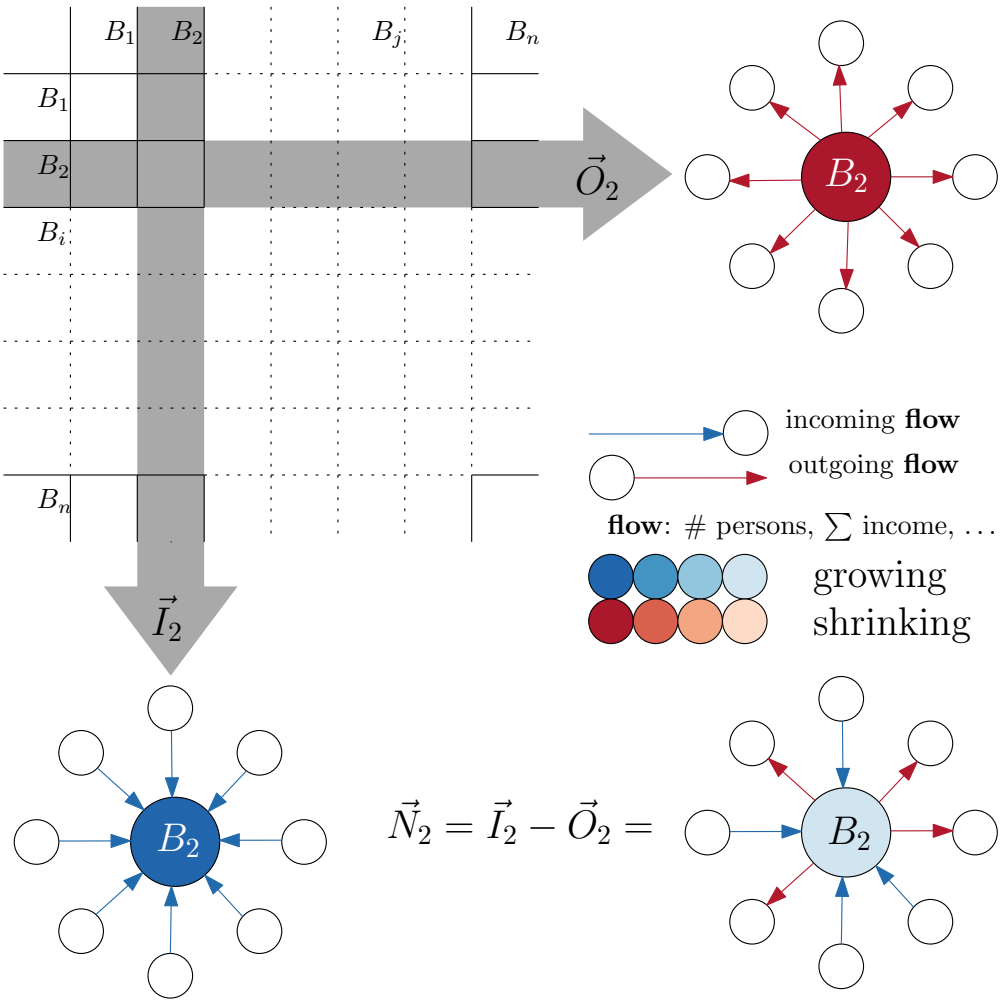


Figure 4.3: Migration matrix and different extracted views

Extracting migration matrix information

The schema in Figure 4.3 gives an overview of the data that can easily be extracted from the matrix, by inspecting cells, rows or columns.

Using the Migration Matrix, some metrics can be defined for the partition's blocks.

Definition 4.2.7 (In-vector, Out-vector)**.**
The *In-vector* of B_i:$\vec{I}_i =$ the i'th column of $M(B,P')$
The *Out-vector* of B_i: $\vec{O}_i =$ the i'th row of $M(B,P')$
The *Net-vector* of B_i: $\vec{N}_i = \vec{I}_i - \vec{O}_i$

Definition 4.2.8 (Inflow, outflow, net flow)**.**
For some Migration Matrix $M(B,P)$ of $n \times n$

$$\text{The } \textit{inflow} \text{ into } B_i: \quad F_i^{\text{in}} = \sum_{k=1}^{n} (M_{ki}) - M_{ii}$$

$$\text{The } \textit{outflow} \text{ from } B_i: \quad F_i^{\text{out}} = \sum_{k=1}^{n} (M_{ik}) - M_{ii}$$

$$\text{The } \textit{net flow} \text{ of a } B_i: \quad F_i^{\text{net}} = \sum_{k=1}^{n} (M_{ik} - M_{ki}) = F_i^{\text{in}} - F_i^{\text{out}}$$

From the definition it can be seen that *internal* flow is not taken into account when calculating the in- or outflow.

Definition 4.2.9 (Growing, stable, shrinking)**.**
If $F_i^{\text{net}} > 0$, B_i is *growing*.
If $F_i^{\text{net}} = 0$, B_i is *stable*.
If $F_i^{\text{net}} < 0$, B_i is *shrinking*.

Chapter 5

Solution

This chapter first gives an overview of the tool, followed by a descriptions of the general principles that are used for all visualizations, and finally a description of the individual visualizations: map, matrix, scatterplot, age pyramid, top flow bar chart. The development of the tool has been an iterative process in which the demographic and migration experts were consulted on a regular basis. Their input has been used to improve the usability and understandability of the tool. The goal was to make the visualization powerful, but simple enough to keep things intuitive. As the targeted audience is broad, the tool should be operable without having expert knowledge.

5.1 Overview

The main interface of the tool on startup is shown in Figure 5.1. Not all visualizations are active from the start. Instead, the basic visualizations that can be used to answer the more quantitative questions are displayed. Via the View menu in the top menu bar, the other visualizations may be switched on. Figure 5.2 shows the tool with all visualizations switched on.

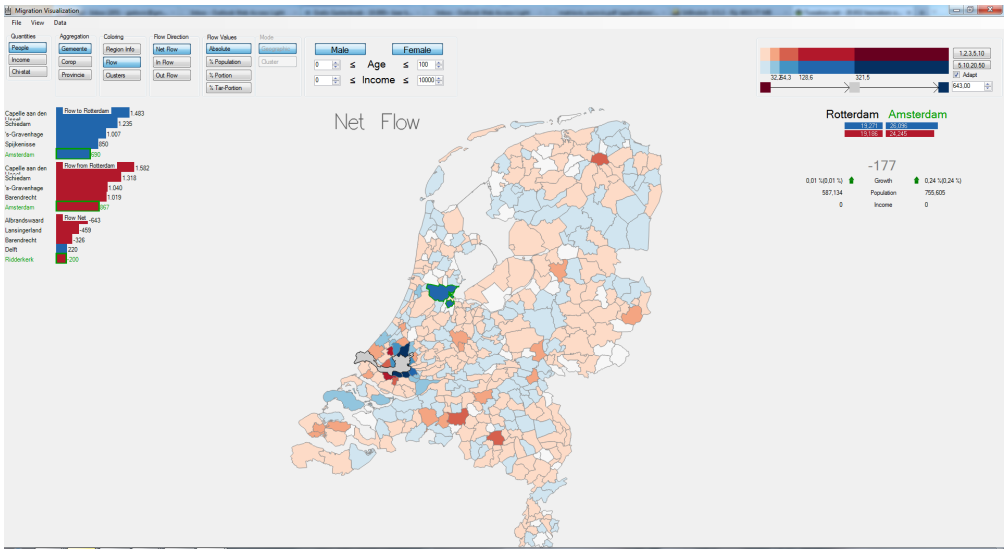


Figure 5.1: The complete tool with the initially enabled visualizations and after loading a dataset and selecting Rotterdam.

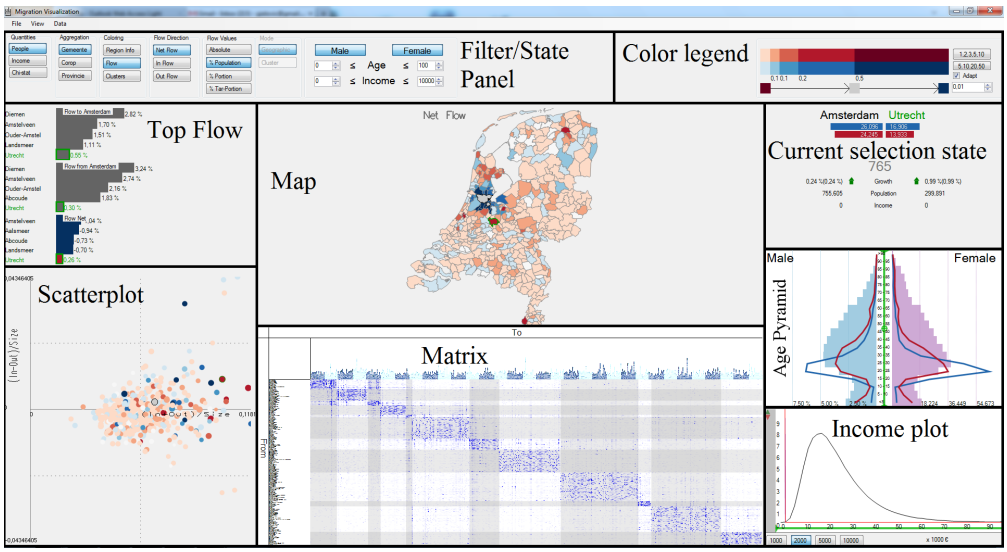


Figure 5.2: The complete tool with all visualizations expanded.

5.2 General principles

The following general principles and guidelines are used in all different visualizations.

5.2.1 Simple visualizations

We selected visualizations that are rather simple and intuitive, instead of using algorithmically heavy or sophisticated novel graphical encodings, also for the sake of an easier implementation as well as for the sake of a better understandable tool for the user. Also non-expert users should be enabled to use the tool and interpret the visualizations.

5.2.2 Linking and brushing of simple visualizations

Using different visualizations simultaneously is the second principle used. Making a single perfect picture has so far been impossible, and most likely will remain to be impossible. Therefore the approach of *multiple views* is chosen: Multiple, linked and interactive visualizations working together to give the user insight into the data. The visualizations not only serve as information displays, they are also used to perform selections, apply filters and interact with the data.

The coupling of different views makes it possible to combine all kinds of variables and data for the different regions and migrations. In general, if a region is selected and highlighted in one view, it is also highlighted in other views. Furthermore, where possible, the same colors are used in different views. The user can select which visualizations are activated, as mentioned in Section 5.1.

5.2.3 Selections

Two types of selections that are important for the user have to be supported:

1. Selection of a set of regions
2. Selection of a pair of regions, inspecting the migration between the regions.

In the first case, one or multiple areas are selected at once. Such a selection indicates there is an interest in this aggregated set of regions and visualizations should thus be displaying information about the selected set. Only regions of the current partition may be selected.

In the second case, the user is interested in a certain cell of the Migration Matrix, or an edge between two regions. So, specific details about the interaction between two different regions should be displayed.

If one or multiple regions are selected, the partition used is updated. Suppose $S_B = \{B_i \in B | B_i \text{ is selected } \}$ is the set of selected regions of partition B of $\mathscr{R}$. The partition is updated as follows: $B' = B \setminus S_B \cup \{a \in B_i | B_i \in S_B\}$. Thus, the currently selected regions are removed from their partition and its elements are added as one big region. This region is referred to as the *selected region* or selected area.

5.2.4 Filtering

Integrate filter specifications and visualizations. The goal is to provide a very natural way of setting filters, also for a non-expert user. Therefore, the choice has been made to integrate the filter controls and the visualizations that are about the same information. Age filters are set in an age distribution diagram (or age pyramid) for instance.

Visual cues within the visualizations are also stronger than just a simple notion of a filter setting in text.

5.2.5 Real time state overview

To prevent the user from loosing track of what filters are set (visualizations may be hidden) and what selections are made, the current *state* of the visualization should be visible at all time. Therefore, we added an option to show the currently set filters, including the possibility to adapt these.

5.2.6 Aggregation

The data model that is used is very well suited to make aggregations of the data. All visualizations are able to deal with these aggregations. To keep things consistent, visualizations should be aware of the current state of aggregation and should display data for the current set of partition's cells. So, if the COROP's partition is used, an aggregated age distribution for the (selected) COROP should be displayed.

5.2.7 Foreign migrants and isolation

As mentioned in Section 4.1.1, we deal with foreign migrants (people migrating from or to other countries) by putting them all in artificial regions called the *Source* for

entrants and *Sink* for leavers. These regions are merged in one Source/Sink region (source in the remainder of this document). Unfortunately, due to the nature of the dataset, we can not separate foreign migrants from births and deaths. Extra information would be necessary to be able to do that. Births may be filtered out using the age filter, but this would also filter out the youngest of the foreign migrants. The foreign migration plus births and deaths exceed the migration figures in most age categories for almost all municipalities. Therefore, the option is provided not to take foreign migrants into account.

A lot of previous studies regarding migration focused on some particular region, like a province. To accommodate that, we have build an option to *isolate* a certain group of municipalities. This option filters out all $a \in \mathscr{R}$, that are not part of the current selected region. The new partition and matrix are build using only the regions that where present in the selection, the regions that were filtered out are considered foreign and are included in the source. A typical use of isolation is to investigate what is happening inside a certain province.

5.2.8 Displayed information

Three types of information are discriminated:

1. *Region info*. For instance, population, growth, or any other loaded variable, that can be expressed as some variable X_i on area B_i.
2. *Flow info*, matrix cell's contents (M_{ij}).
3. The *partition* a region is in.

The first, region info, contains information per region, of the form X_i, thus containing only one index (i). Certain migration data is also considered as region info. Examples are the total inflow, outflow and net flow of a region, which are in fact, aggregates of the migration matrix columns and rows. Other examples of region info are population size, total area, age and gender distribution and income distribution. Various other scalar variables can be read in later on, using the regions variable load functionality.

Flow information has to do with the contents of the migration matrix' cells. In case of flow information, the variable M_{ij} uses two indexes, namely i and j. This indicates that we are interested in couples of regions. Flow info can also have different forms, dependent on the function f that is chosen in $M_{ij} = f(P_{ij})$.

The third type of information to be shown is the distribution of municipalities over a partition's blocks.

Flow information

As defined in 4.2.6, M_{ij} is a function f over the group of people P_{ij}. This provides the possibility to show more statistics than simply the number of people. Dependent on what should be displayed, M_{ij} denotes:

Total number of migrants: $|P_{ij}|$

Summed income of migrants: $\sum_{p \in P_{ij}} p.\text{income}$

Chi-statistic: $\frac{|P_{ij}| - E_{ij}}{\sqrt{E_{ij}}}$

The total number of migrants represents the number of people moving from B_i to B_j, the next is the summed income of all migrants moving from B_i to B_j. The final one needs some more explanation. It is the signed Chi statistic that is also used by Wood et al. [19]. The observed number of people $|P_{ij}|$ is corrected for the expected value, based on the population sizes of the origin and destination areas. If a flow is smaller than expected, a negative value is the result, if the flow is bigger than expected, the value of the Chi-statistic is positive.

The expected number of people to migrate between B_i and B_j, E_{ij}, is defined as:

$$E_{ij} = \frac{|\mathscr{P}|}{\sum_{k=1}^{n} S_k} \cdot \frac{S_i + S_j}{2(n-1)}$$

The first factor $\frac{|\mathscr{P}|}{\sum_{k=1}^{n} S_k}$ represents the probability someone migrates (total number of migrants divided by the total population). Then both populations (of B_i and B_j) are divided by $(n-1)$, the number of regions left, to simulate a uniform distribution of migration over the remaining regions. Finally, the result is divided by 2 to correct for the bi-directionality. Thus E_{ij} is the expected number of migrants between B_i and B_j, based on the size of the population of both regions.

Actually, the Chi-statistic can be considered as a normalization step, which is discussed in Section 5.3. Due to its explicit definition, we decided to consider it flow information, and present it as such in the tool.

5.2.9 Coloring

The use of colors is a classic way to encode data, and these are present throughout the entire tool. Different classes of color maps are used to support the different types of information. Provided are:

1. Sequential color maps
2. Diverging color maps
3. Qualitative color maps

For scalar variables with values in $\mathbb{N}$ or $\mathbb{R}_{\geq 0}$, a sequential color map is used to display the magnitude of the variable per region. If the values lie in $\mathbb{Z}$ or $\mathbb{R}$, a diverging color map is used to display sign and magnitude. The qualitative color map is used for displaying categorical variables or partitions, and consists of easy-to-distinguish colors.

To display region info(1), both the sequential or diverging color maps may be used. For flow (2), a diverging color map is required as the sign can be used to distinguish between in- and out-flow.

High quality color maps were obtained from ColorBrewer 2.0 [3].

5.3 Normalization and scaling

As in most visualizations, normalization and scaling plays an important role when it comes to creating 'good', meaningful images. Instead of looking at absolute numbers, we might consider these numbers as percentages of a certain other amount. For instance, looking at absolute inflow (number of people flowing into an area) will definitely accentuate the major municipalities containing big cities with large populations. If the inflow is divided by the population, this effect diminishes.

5.3.1 Normalizing region info

Region info variables are of the form X_i. Normalizing and scaling these is straightforward. As $X_i \in \mathbb{R}$, we can divide by a non-zero constant c or a variable Y_i.

Dividing by a constant c is a global normalization step, where $c = 1$ means: 'no scaling' and 'no normalization'. Division by the total size of the population gives migration info as a fraction of this quantity. Dividing by another variable Y_i is a region dependent normalization step. A common variable that is used for normalization in demographics is the size of the population S_i. Also in migration normalizing by population size is a natural choice. Regions with a big population probably have a lot of inflow and outflow, assuming a uniform distribution of migrants over the country (which we are investigating actually). If growth is considered, percentages are often more interesting than absolute numbers. A net migration of $+100$ persons is more significant for a small town than it is for a big city. Therefore, the net flow F_i^{net} is often normalized by the population S_i.

A consideration that could be made is whether or not to use filters in determining the population. If we are looking for instance to the net flow of people between the age of 20 and 30, we could be interested in how much percent this particular group is growing, instead of looking at the percentage growth of the entire population of the region. The tool does not support this option, as we think it does not contribute to the understandability of the visualizations.

5.3.2 Normalizing flow info

The use of a normalization for flow alters the content of a Migration Matrix' cell M_{ij}. The case of dividing by some constant c is, just as in the case of region info, trivial.

Population correction It becomes more difficult when we try to normalize by a region or matrix dependent quantity. Just as in the case of region info normalization, population correction is a plausible step when analyzing migration data. But, as there are two regions involved, a choice should be made how to combine the populations of regions B_i and B_j. A generic approach is to use:

$$M'_{ij} = \frac{M_{ij}}{a \cdot S_i + b \cdot S_j}$$

where a and b are to be chosen dependent on purpose.

If all cells need normalization, we choose $a = b = 1$. If a set of areas is selected however, we are often merely interested in the in-vector and out-vector of the selected region (note that this is a single block in our partition, as mentioned in 5.2.3). When considering the in-vector $\vec{I}_j$, the population of B_j is the same for each cell to normalize, and thus we leave it out by setting $b = 0$ and $a = 1$. The case considering $\vec{O}_i$ is symmetric, thus $a = 0$ and $b = 1$.

Of course, just like the case of region info normalization, every variable Y_i might be used for normalization, instead of S_i.

Flow ratio A special kind of property of regions B_i and B_j to use for normalizing cell M_{ij}, is a flow-related property that is derived from the migration matrix itself. This option is named flow ratio as we are normalizing by some flow quantity, extracted from the (flow) matrix. Such normalization may not reduce the dominating effect of large regions (with large populations), but does provide us with a much more narrow range of values, depending on the normalizing variable chosen.

Just as in case of population correction, two scenarios are distinguished, namely the case where all flows are considered, and the case in which the focus lies on some selected region and we are interested in its in- and out-vectors.

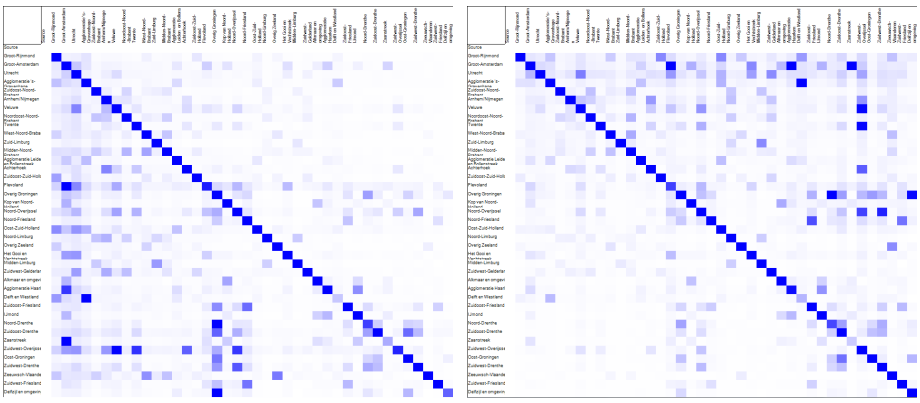
When considering a selected region B_s, we are interested in $\vec{I}_s$ and $\vec{O}_s$. Instead of looking at the absolute values of the elements of these vectors, we can consider *ratios* of the total inflow F_s^{in} or total outflow F_s^{out}.

$$\begin{aligned} M'_{si} &= M_{si}/F_s^{\text{out}}, \text{ for } 1 \leq i \leq n \wedge i \neq s \\ M'_{is} &= M_{is}/F_s^{\text{in}}, \text{ for } 1 \leq i \leq n \wedge i \neq s \\ M'_{ss} &= \text{undefined} \end{aligned}$$

Without a fixed area B_s, it is not so obvious what quantity to use for normalization. As in the population correction, we could combine the in- and outflow: $M'_{ij} = \frac{M_{ij}}{a \cdot F_i^{\text{out}} + b \cdot F_j^{\text{in}}}$, but this is difficult to interpret.

As we are deriving the normalizing property from the matrix, we might as well define the normalization in terms of the matrix. Instead of using the previous formula, the cell's contents can be divided by the row or column maximum or sum. This leads to a highly unbalanced matrix in which one side of the diagonal will be favored over the other in terms of high intensity coloring. An example is shown in Figure 5.3.

A final option would be to change the cells content in such a way that all rows and columns sum up to 1. For this, a linear programming algorithm may be used. This would slightly change the values in the cells however, making it somewhat inaccurate.



(a) Row Max (b) Col Max

Figure 5.3: Normalizing using only row or column metrics leads to an unbalanced matrix visualization.

5.3.3 Color map scaling

Finally, the values are scaled to fit nicely on a range that suits the color map. A color map assigns a color to some scalar value. First, the value is mapped to the range $[-1 \ldots 0 \ldots 1]$ or $[0 \ldots 1]$, then a color is assigned.

Mapping absolute numbers to this range is non-trivial as outliers can easily ruin the distribution. Also, assigning a color to a value can be done either discretely or in a continuous way. When using a discrete color map, such as the one in Figure 5.4, the color range boundaries may be adapted to obtain 'better' images, or to get easy to interpret, meaningful scales.

Figure 5.4: In a discrete colormap, the range boundaries can be shifted obtain a more effective color distribution.

Information types per visualization

We implemented seven types of visualizations. Table 5.1 shows these and what information is visualized. An *S* indicates scalar values can be shown per region, a *D* indicates the possibility to display distributions. A circle indicates *all* regions' values are shown at once in case of Region info and Partition, and *all* flows in case of Flow info. Gray shaded areas indicate the favorable visualizations per information type.

5.4 Map

The *map* is the central visualization of the tool. It contains the entire geographical map of the region of interest. In our case, a map of The Netherlands to start with.

Table 5.1: Information types per visualization.

x	Region Info	Flow	Partition
Scatterplot	Ⓢ	S	◯
Matrix		Ⓢ	◯
Map	Ⓢ	S	◯
Age Pyramid	D	D	
Income plot	D	D	
Cluster view	D/S		
Top 5		S	

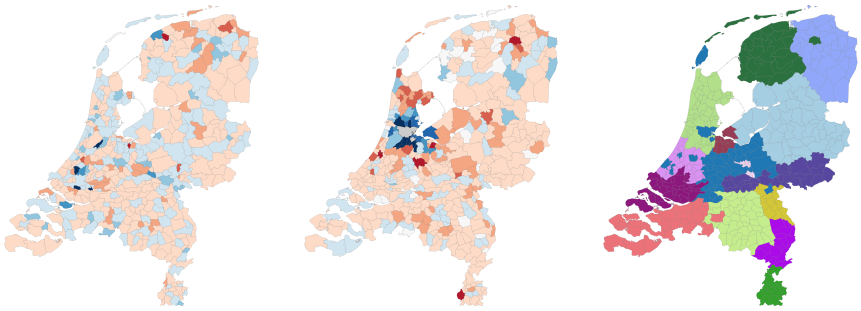
5.4.1 Displayed information

A map is a common visual representation of space. But as we are bound to the existing geographical layout of a country, we are limited in ways to represent *information* about the different parts of the map. There are two well-known methods, namely coloring the individual areas on the map, known as a chloropleth map, and using symbols or glyphs to depict some variable. The first one is a very natural way of representing information, although one must be careful in choosing the color map that is used to determine each region's color. Different types of data require different color maps, as mentioned in Section 5.2.9.

The second one, the use of symbols, is also a very powerful mechanism as instead of displaying a single variable, more information can be shown at the same time. Think of an arrow per region in which the direction represents some ratio and the length represents a certain quantity. Also, entire distributions might be put into the symbol map. We choose not to include symbols in the map, as symbols tend to clutter an image. We are exploiting the power of linking and brushing by using 'simple' visualizations, rather than putting as much information in a single picture as possible. Also, all information that can be put in symbols now comes available by the use of interaction. The layout of the municipalities in The Netherlands also does not help, as there are many irregularly shaped, small areas, in which the symbols would not fit anyway. This results in an even more messy image.

Table 5.1 already shows that the map is suited to display all types of information required: region info, flow info and the partition. As region info, scalar values can be displayed by coloring each region according to the variable's value and some color map. Not all flows can be displayed at once. There has to be a selected region in the partition, which is highlighted as 'the selection' (Amsterdam in Figure 5.5b). The rest of the partition's regions receive one color each, and their elements (which correspond to areas on the map) are filled using that color. For the default partitions, separate maps are available. So instead of coloring the partitions, as is done in Figure 5.5, other maps with different boundaries are loaded, as can be seen in Figure 5.6.

The partition can be shown by filling elements (areas on the map) that are together in a partition's block with the same color. Although this seems trivial, it can be difficult to assign distinct colors to each different block. In case of the default partitions, in which the partition consists of blocks of municipalities that are geographically connected, the borders of the partition can be highlighted/thickened.



(a) Region info (b) Flow info (c) Partition

Figure 5.5: Displaying the three types of information in the chloropleth map

(a) Municipalities (b) COROP-areas (c) Provinces

Figure 5.6: Default partitions have their own map

Optionally, arrows can be drawn between a hovered and selected region. When hovering over the selected region, all areas coming into and going out of the selected area are drawn on the map. This is shown in Figure 5.7. The chloropleth coloring is removed in the left image.

5.4.2 Implementation details

The map data was provided by CBS in the form of a shapefile [6] containing one or more polygons per region. For each level in the region hierarchy, a separate map is used. The map data is not ready for use immediately. The polygons are not all convex, which causes a problem for the OpenGL polygon-fill routines. A vector method for splitting concave polygons, described in [7], Chapter 3, is used to split the concave polygons in sets of convex polygons.

Per region (municipality, COROP, province), an OpenGL display list is generated to be executed during runtime. Borders, all having the same color anyway, are put together in one single display-list. This way, rendering the map is in no way a limiting factor.

Users must be enabled to select regions. This is implemented via an inside-outside test using the *odd-even* rule. A line is drawn from the point of interest to a position outside the scene and the number of crossed segments for each polygon is determined. If this number is odd, the point of interest lies within the polygon.

To draw the arrows, Bézier-splines are implemented using the OpenGL Bézier-Spline

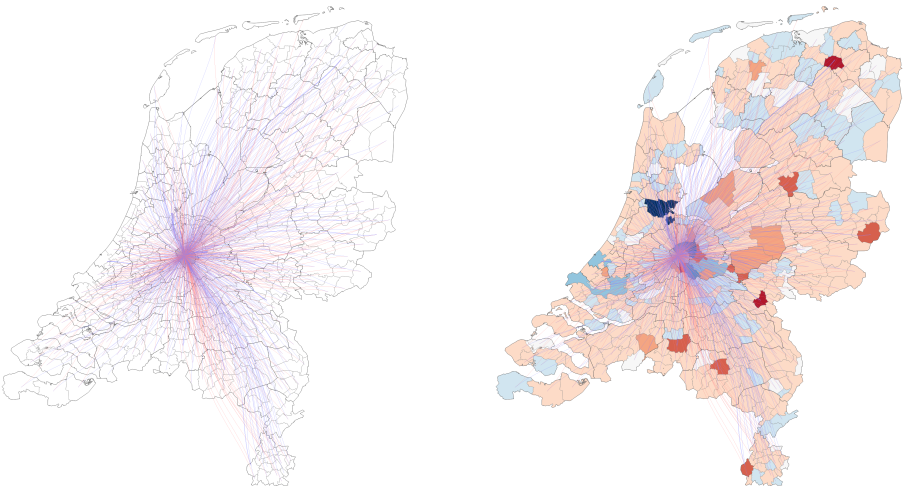


Figure 5.7: Arrows shown during a hover over the selected are. Arrow's directions are from fat to thin.

Curve functions. For this, a set of *control points* has to be defined. Several combinations have been tried, but as we are focused only on arrows from and to the selection, a simple approach is taken. Three control points are added between the start and end point for each arrow, two at $1/4$'th and $3/4$'th between the start and the end. The center point is placed with a perpendicular offset from a point halfway between the start and the end points. The control points and resulting pair of arrows between two points are shown in Figure 5.8.

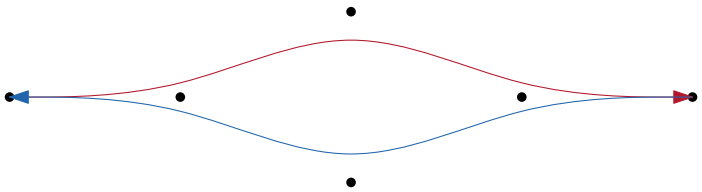
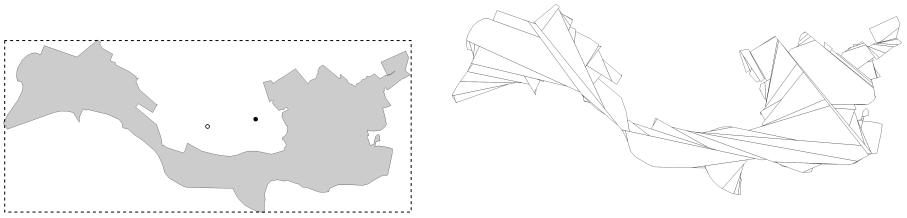


Figure 5.8: Drawing both direction arrows between a pair of region centroids (left and right most points).

A problem occurs when defining the start and endpoints of the arrows. These should lie inside the boundary of the (multi-)polygon describing the region. The irregular shapes make it difficult to define a centroid that lies inside the area. For instance, Rotterdam, show in Figure 5.9a, is an example of a municipality for which the centroid of the bounding box does not lie in the polygon (circle). Also, taking the average of all vertices does not result in a vertex lying inside the polygon (disk). Computing a good centroid for a concave, irregular polygon containing disconnected sections and holes is an interesting computational geometry problem.

A simple approach is chosen. Having split the concave polygon in multiple convex ones, the latter can be used to quickly find a point that lies inside the concave one. We can not say anything about the closeness to the border using this technique. We could try to find the convex polygon centroid that lies closest to the average vertex. In case of a shape like the one in Figure 5.9a, this method fails horribly as it will certainly pick the centroid of the convex polygon closest to the border. If this is a thin polygon (we

are not applying the Delaunay triangulation algorithm so thin polygons do appear), the chosen centroid will lie very close to the border.



(a) Centroid (b) Convex

Figure 5.9: The centroid of the bounding box (circle) and the average vertex (disk) of Rotterdam do not lie inside the polygon. The right image shows the concave polygon, split in a set of convex polygons.

5.4.3 Interaction

The map has normal interactions available to view the map like zooming (with the mouse wheel) and panning (by clicking and dragging). The region that is hovered over is highlighted by recoloring its boundary. If a region is hovered, it can be clicked upon to select it.

On a right-click, a context menu opens, which provides options to turn certain visual features on or off. Also, default colors can be set, the color map can be changed and the image can be saved to a file on a higher resolution. There are also two options regarding the selected area. First the option to *invert* the selection, which is a quick way to select a large number or regions. Invert takes the complement of the currently selected partition(s).

The second option is to isolate the selected area, a process that is described in Section 5.2.7.

5.5 Matrix

With a matrix as underlying data model, it is rather straightforward to use a flow matrix visualization. In Chapter 3, we already discussed limitations of the flow matrix visualization. Nonetheless, a flow matrix can greatly contribute to obtain insight in global patterns. The disadvantage of being unable to capture the geographical layout, which the map does offer, comes with the advantage of being able to re-order the matrix's rows and columns.

5.5.1 Displayed information

Unlike the map, which only shows migration data with respect to a certain selection, the matrix displays *all* flows between regions in one image. The asymmetry (along the diagonal) indicates a net flow in a particular direction, but this is very hard to see.

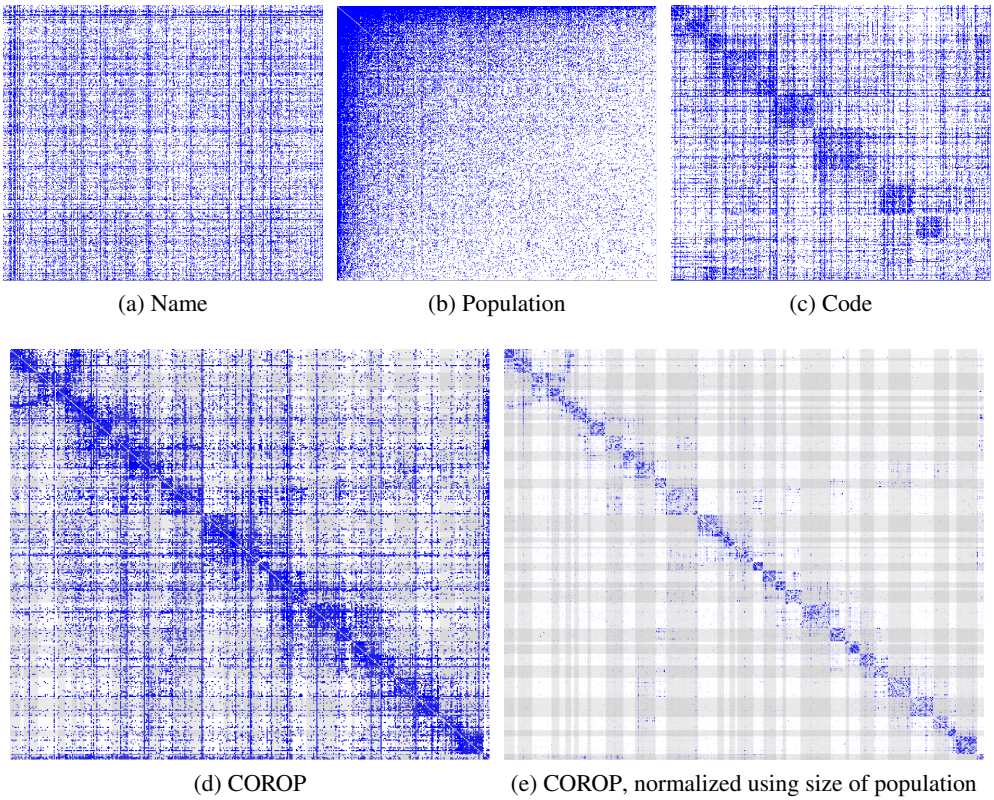
Ordering

Initially, the matrix often looks like a scattered mess. Although this might indicate the absence of any global pattern in the data, it probably has to do with bad row/column ordering. If the regions (parts of our partition) are for instance alphabetically ordered on their name, a bad image is to be expected (Figure 5.10a). Sorting the municipalities by population (Figure 5.10b) clearly shows that the biggest flows can be found between the regions with the biggest populations. A good idea is to order the areas based on their geographic position. This can be achieved fairly easy as the municipality codes are ordered according to the geographic layout of the country (Figure 5.10c). Unfortunately, changes in the regional division over the years have caused irregularities in this pattern.

Also, we can take advantage of the region hierarchy. Municipalities can be sorted for COROP-region, and these can be sorted for provinces. The same holds for sorting based on P (provinces).

To stress the boundaries of these coarser partition's cells, columns and rows with equal parent cells are shaded alternately, introducing a checkerboard pattern. Figures 5.10d and 5.10e show examples of this banding.

Besides the geographic pattern, it is also interesting to see if global patterns arise when another coarser partition is used for the sorting, for instance: clusters of municipalities having similar age distributions. In Chapter 8, an example of such a case is shown.



(a) Name (b) Population (c) Code

(d) COROP (e) COROP, normalized using size of population

Figure 5.10: Ordering the regions has great impact on the resulting image of the matrix. Normalization also plays an important role in the construction of readable and meaningful images.

Coloring

Each cell has a positive value, and thus a sequential color map is used.

5.5.2 Implementation details

When using the partition M, there are 400+ blocks in the partition available. Having one row and column for each partition-block, the matrix already contains at least $400^2 = 160{,}000$ cells. For each cell, the number of migrants has to be evaluated, according to the set of filters applied. Also a cell has to be plotted with a certain color. A small optimization does not plot the cells that are empty and will receive the white background color anyway. For the test-data set, this approach still takes roughly a couple of seconds. On each change of filters, these calculations have to be made. For other types of interaction, like selection, it would be very inefficient to recalculate the cells' values and redraw them. Therefore, a bitmap is made on a filter change, which is later used to be drawn on in case of other interaction that does not change the data values. This way, interactive frame rates are still possible.

5.5.3 Interaction

It is possible to zoom into a certain group of cells in the matrix by holding shift and dragging a box around the cells. This is particularly useful in case of a great number of parts in the partition (and thus many columns/rows). Zooming back to the original visualization is done by shift clicking anywhere on the matrix.

The matrix can be used to select and hover over areas. The hovered and selected area are also highlighted in the map, which makes it easy to relate a certain row or column in the matrix to the corresponding geographic area on the map. It is not possible to select groups of areas using the matrix.

As the labels surrounding the matrix are unreadable small in case of a high-ranked partition, they are enlarged during the hover action.

The right mouse button opens a context menu that enables the user to switch to a different color map, change the color scaling and changing the sort function.

5.6 The Scatterplot

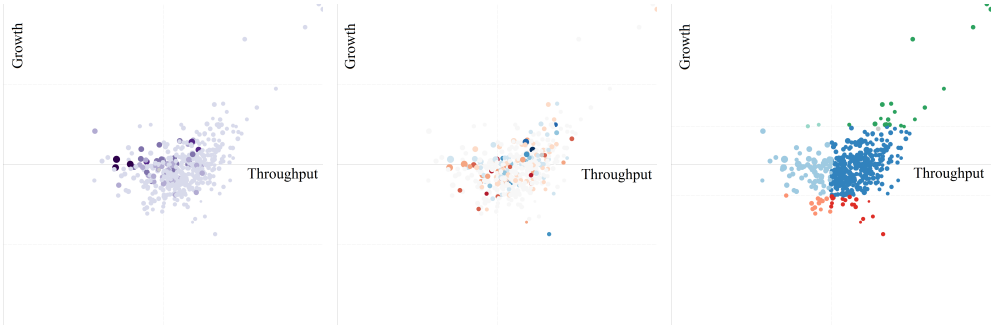
A scatterplot is a standard visualization for showing relations between two attributes of sets of items. We included a scatterplot to enable experts to study for instance the relation between growth and the size of regions.

5.6.1 Displayed information

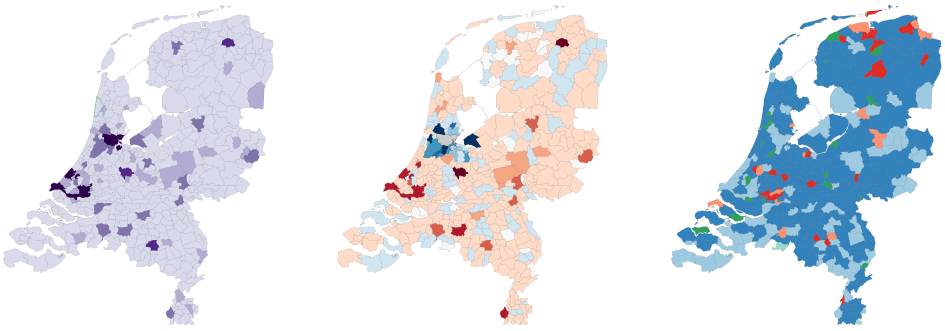
The scatterplot can be used in several ways. In general, it shows information about all the areas in the current geographic aggregation. For both axes, a variable can be chosen to be displayed. By choosing growth on the vertical axis (may be positive or negative) and any other variable on the horizontal axis (can only be positive), it is

easy to show the effect of certain variables, such as the average income, the number of agricultural companies, the number of new houses build, etc., on the attractiveness of an area. The radius of a dot indicates the population of the area.

By default, the color of the dots in the scatterplot is the same as the coloring in the map. This makes it easier to relate dots to the areas they represent on the map. The selected areas are highlighted by the same gray color and black outline and the hovered area is also highlighted as it is in the map. Although the position of the dots can only represent region info, the color can also represent flow info between regions. When displaying flow information in the map (by selecting some area), the same colors are applied to the dots in the scatterplot, showing the content of cells M_{ij} and M_{ji} for a fixed (selected) i and $0 \leq j < n$. When displaying region info in the map, for instance



(a) Region Info (b) Flow Info (c) Partitioning



(d) (e) (f)

Figure 5.11: The scatterplot enables the user to relate region information, flow information and partitioning to growth and throughput.

the population per municipality, the color map is also applied for the same variable (population) on the scatterplot. This way, one is able to relate all kinds of variables, including custom loaded variables, to growth and throughput (or any other variable currently loaded on the horizontal or vertical axis).

By default, the color is used to relate the areas on the map, to the dots in the scatterplot. It is however possible to display other scalar variables using a sequential color map. The scatterplot is also used for selection and hovering, which is especially useful to relate dots to areas on the map.

The scatterplot contains stippled lines which divide the total area in six rectangles. These rectangles can be used to color the dots, and also the areas in the map, according to their position in the scatterplot. The choice for six rectangles is based on the default

categories often used by demographic experts. These are: growing regions, stable regions and shrinking regions on one axis; and regions with low and high throughput on the other axis. Moving the stippled lines causes re-clustering of the areas into six groups, according to their position, such that the user is able to adapt the definition of a strongly growing/shrinking region and see the effect on the map.

5.6.2 Implementation details

The scatterplot is, just like the map, build using the OpenGL graphics library. Dots are plotted using the `GL_POINTS` primitive. In case of a selection, where the border of the dot is highlighted with a black line, a circle is used, as the OpenGL point does not provide the option to color the border with a different color. Another option would be to under-plot a slightly bigger black dot.

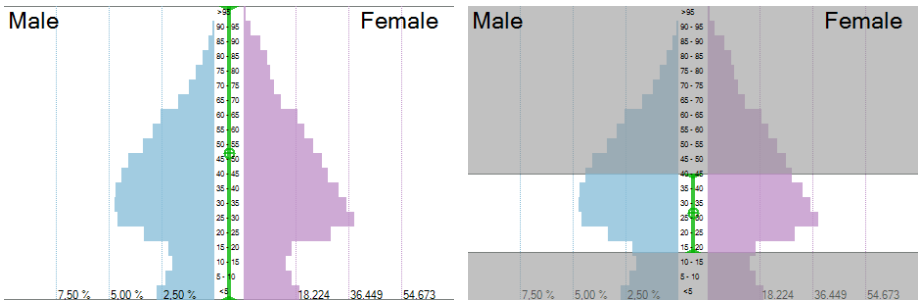
5.7 People's attributes plot

The age pyramid and income plot are typical examples of visualizations dedicated to a certain variable related to people. In the dataset, each person $p \in \mathscr{P}$ has a range of attributes, such as age, gender, annual income, education, household situation, ethnicity, etc. Each of these require slightly different visualizations. Due to time constraints, not all are implemented. Age, gender and income are chosen after discussion with the demographic experts, who pointed out that these variables greatly influence the (economic) position of regions. It is interesting to see where young (high potential) people go and where the money is going. Education is also a great indicator but is left out for practical reasons. According to the demographic experts, education is poorly documented for elder people, and in some occasions, an ordinal relation is hard to define.

Ethnicity is (unfortunately) also of great influence on the prosperity of a region, but on the other hand a controversial topic. Nonetheless, demographic experts indicated it would be very interesting to see where ethnic minorities are moving to and what the effect is of such a migration. Due to time constraints, this is also not implemented. It would however make a nice third category of attribute plots, as a histogram is a typical representation to represent such categorical/unordered data.

5.7.1 Age Pyramid

The age pyramid displays information about both the age distribution and gender distribution of the people in a region or set of regions. In the latter case, distributions of the individual regions in the set are aggregated. The age pyramid is a specialistic visualization, and is the most common way to display the age and gender distribution of a region. By general agreement, males are displayed on the left and females are displayed on the right. Between a certain range of ages, we use 0 - 100 years, people are put into bins. We choose for 20 bins of 5 years. The length of the bin is determined by the fraction of the total population the bin represents. An example is given

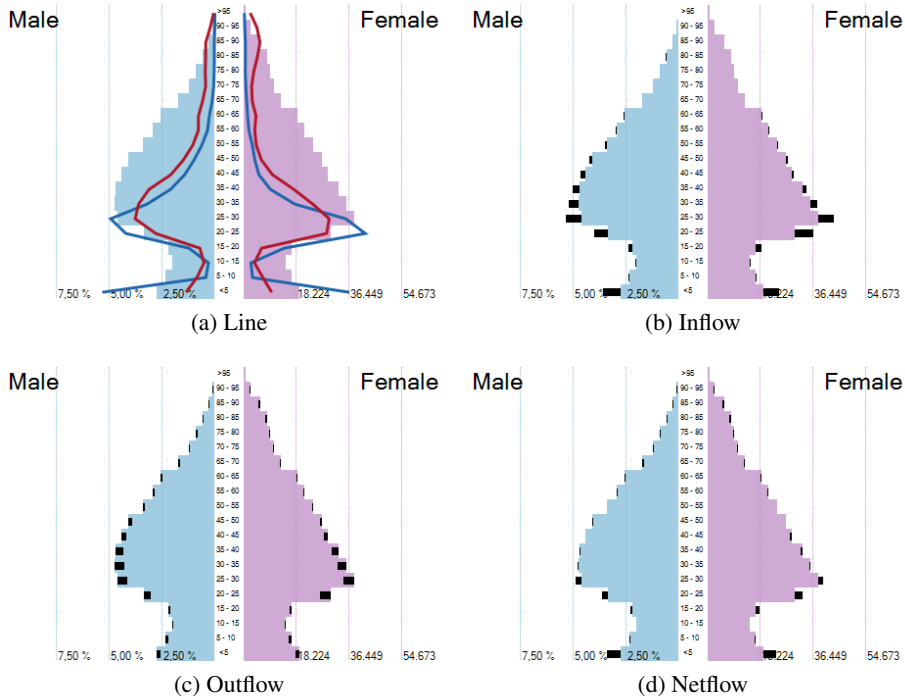


(a) (b)

Figure 5.12: A typical age-pyramid. This one is showing the age distribution of the municipality of Amsterdam.

in Figure 5.12. Demographic experts will immediately recognize a typical big-city-like pattern. Also, this particular example region shows a surplus of 20-30 year old females, compared to the males.

The green handles in Figure 5.12a can be used to set age ranges for filtering. The result of dragging these handles can be seen in Figure 5.12b.



(a) Line (b) Inflow

(c) Outflow (d) Netflow

Figure 5.13: The age distribution of immigrants and emigrants can be shown in different ways.

Besides being able to show the age distribution of the population of a region, we are also interested in the age distribution of migrants. In terms of region info, the profile of all people moving in and the profile of all people moving out is interesting, as these can cause a shift in the age distribution of a region. There are several ways to show the effect of migration on a region's age distribution. Two are implemented.

In Figure 5.13a, a line plot for the migrants' age distribution is shown. In this line plot, it is easy to see the ratio between inflow and outflow for each range of ages per

gender. For this particular case, we see a higher inflow into Amsterdam in that age range of 15 to 30 years, for both males and females. Also, there is a higher inflow of females, compared to males. What cannot be derived from this line plot is the effect of migration on the current population, as both the lines and the underlying shaded bins are scaled to fit the visualization nicely.

For that, the second type of plot is used, shown in Figures 5.13b,5.13c and 5.13d. Here the immigrants are added to the bins, while the emigrants are 'removed' from it using a darker shaded rectangle. The net flow gives insight what age categories grew due to migration. As can been seen in Figure 5.13d, this effect is rather small in most bins.

5.7.2 Income plot

The income plot is shown in Figure 5.14 and has similar features as the age pyramid. Besides looking at the contents of a region and its migrants, the income filter can be set using the visualization. The only limitation is that males and females are not split. This is due to the fact that income information is not that individual as age information. A lot of households have some kind of shared income, and a lot of different other scenario's are possible. Therefore, no distinction is made between males and females.

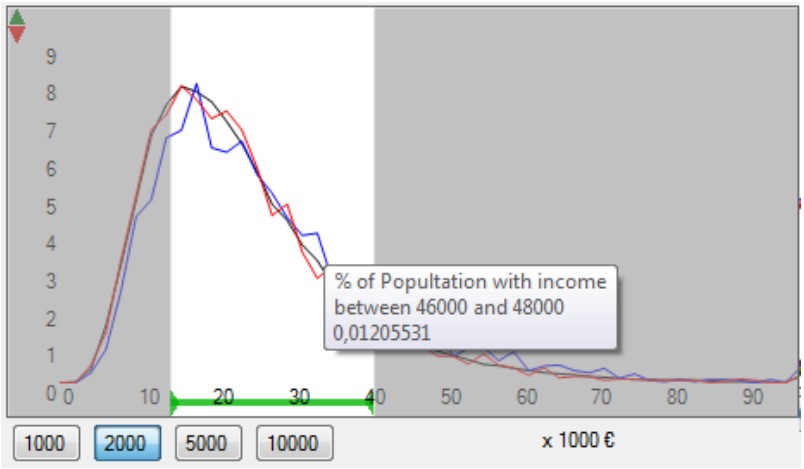


Figure 5.14: The income plot panel showing the income distribution of the selected region in black. The red line is the income distribution of the emigrants, the blue line is the income distribution of the immigrants.

5.8 The Top-Flow boxes

To show the top origins and destinations for the selected region, bar-charts are used. The in-vector and out-vector of the selected region B_s, $\vec{I}_s$ and $\vec{O}_s$ are sorted in descending order. If the default metric (number of people $M_{ij} = |P_{ij}|$) is used, the regions where most people are going to and are coming from are in the front of the vector. The first k origins and destinations are displayed in separate bar charts.

Color is removed from the in- and outflow bar charts as it is perceived as misleading by several experts who helped testing. Instead, a neutral gray shade is used.

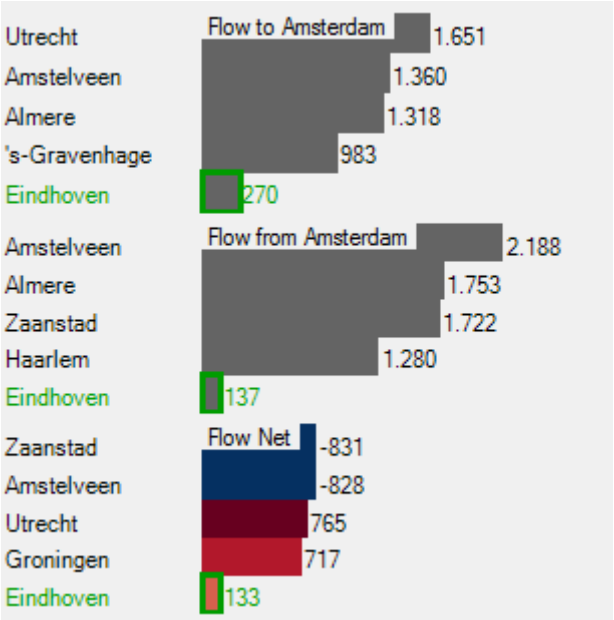


Figure 5.15: Top flow boxes showing the top inflow, outflow and net flow municipalities normalized by population to and from Amsterdam. The fifth position is taken by the hovered region, Eindhoven in this case.

Additionally, one bar chart shows the biggest net effects. The absolute value is used for the sorting. The color of the bars match the colors on the map when showing flow information, thereby indicating the sign of the value.

5.8.1 Interaction

The top-flow boxes react on a hover action. Note that the boxes are only shown if a certain region is selected in the map (or elsewhere). To locate the regions that are displayed in the top-flow boxes, the user can hover over the bars, after which the region is highlighted in the map and on any other visualization that can show a hover.

If an area is hovered in a different visualization, it will also be highlighted in the top-flow boxes. If the area is not part of the top-k currently visible, the last bar is replaced by that of the hovered area. This way, the user can inspect the value of the flow between each pair of regions (each cell M_{ij}).

5.9 The State panel

The *state panel* both shows the current state of the visualization and at the same time is used to oversee and change the filters. To keep things simple, settings are displayed next to each other and all possible options are displayed beneath one another. This way, it is easy to observe the state by a single horizontal scan over the panel. The panel is shown in Figure 5.16.

From left to right, the following options are displayed:

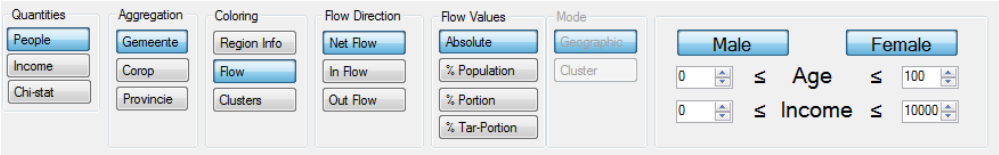


Figure 5.16: The state panel displaying the current state of the visualization

Quantities	Allows the user to choose between different types of flow information, as described in Section 5.2.8. The number of *people*, the summed *income*, or the *Chi*-statistic may be chosen.
Aggregation	The user can switch between the default partitions at any time. The behavior depends on the selection that is currently active.
Coloring	Enables the user to choose between the three types of information that can be shown in the various visualizations. Although 'information type' might be a more accurate caption, coloring is chosen as the choice greatly influences the way most variables are colored.
Flow Direction	Lets the user switch between net flow, inflow and outflow. In case of both area info and flow info this leads to different views.
Flow Values	Here, the normalization applied may be chosen. The four variants for the flow info normalization are present in case we are looking at flow information. In the matrix, only no and population normalization is available. The same holds for the area info normalization.
Mode	Mode is only used when a non-default partitioning is created by clustering the regions. In case of selecting the individual regions in the map by clicking, the clusters can be selected at once. Also, another matrix is generated based on the partition parts instead of the partitions contents.

Also the current settings of (age, gender and income) filters are shown and an interactive color-legend is displayed. The color-legend is shown in Figure 5.17. The ranges

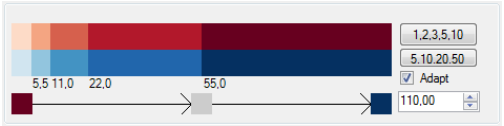


Figure 5.17: The interactive color legend

can be changed by dragging the borders horizontally. The *adapt* option automatically chooses a maximum value for the color map to which all the values are scaled. This maximum can also be set manually. The two buttons on the right provide two presets for percentages. The three colored squares connected by arrows guide the user when looking at the flow from and to the selection. In this particular case, people move from a red region to the gray selection, to the blue region. Thus, red regions loose to the selection and blue regions win from the selection.

Chapter 6

Filtering & Aggregation

This chapter focuses on the implementation of the *Migration Matrix* and especially on the effect of *aggregation switches* and *filter* on the matrix. Filters are essential to sample from $\mathscr{P}$ and generate a new migration matrix with this subset of people, and are discussed in Section 6.1. Aggregation switches can help the user to get a better overview of what is going on as effects on the small scale of municipalities may vary a lot within certain higher level regions, such as the COROP-areas and provinces. Aggregation itself is discussed in Section 6.2. In Section 6.3 the creation of higher level migration matrices and mixed matrices is discussed.

6.1 Filtering

Filters are applied on the set of people $\mathscr{P}$, to create a subset P', from which the migration matrix is build using definition 4.2.6. A Filter has type `Person`→`Bool`, which indicates whether the person should be included (result is `true`) or not (result is `false`). To keep track of what filters are currently set by the user in the user interface, a *state* is kept. The state contains the following fields:

- Minimum Age $\in \mathbb{N}$
- Maximum Age $\in \mathbb{N}$
- Gender $\subseteq$ { male, female}
- Minimum Income $\in \mathbb{N}$
- Maximum Income $\in \mathbb{N}$

The prototype comes with the following set of predefined filters.

- `agefilter`(p) = Minimum Age $\leq p$.age $\leq$ Maximum Age
- `genderfilter`(p) = p.gender $\in$ Gender
- `incomefilter`(p) = Minimum Income $\leq p$.income $\leq$ Maximum Income

Of course, there are numerous other filters possible for all four types of variables (nominal, ordinal, discrete and continuous). Another possibility would be to not limit for instance the age filter to one range only. Demographic experts indicate that they did not need more than one age interval.

The user is able to change the state using the different visualizations. The minimum and maximum age may be set in the Age Pyramid visualization, or in the filter settings panel, which is displayed in Figure 6.1. The same holds for the gender field. Minimum and maximum income can be changed using the income plot, or again the filter settings panel. The latter one displays the state at all time, so that the user can easily see what filters are currently active, even if the corresponding visualizations are hidden. Despite that, we believe that displaying filters in the visualizations provide the user with a much stronger cue to recognize the activation of certain filters.

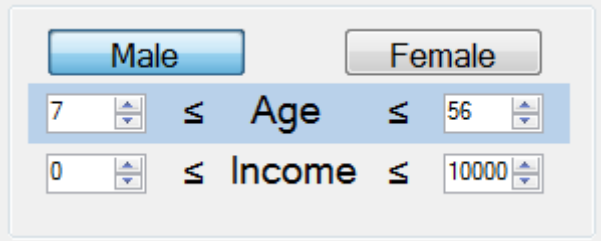


Figure 6.1: The part of the filtersettings showing the person filters set.

6.2 Aggregation

The large number of municipalities present in The Netherlands can cause problems in different ways. Practical problems arise. There is for instance too little screen space to plot a 441×441 matrix on a screen, including labels. Each cell is only a few pixels wide and high, which makes it hard to inspect the cell's contents and relate a cell to a pair of regions. These problems are however not only related to visualization. Also on an administrative level, it is hard to control such a large number of areas. Therefore, the other regional divisions were introduced, like the provinces. The COROP-areas are introduced to help with data-gathering and statistical analysis. For our tool, we reuse these areas to overcome the problem of a large number of areas. Figure 6.2 shows an example of the use of aggregation. The growth as a percentage of the population for each municipality, COROP and province is shown from left to right.

Instead of looking at the flows between municipalities, the flows between COROP-areas can be considered. For the case of provinces, a node-link diagram might be even feasible due to the limited number of arrows.

Also, perceptual problems arise. According to requirement 8, we are interested in finding main patterns and directions in the migration data. This might mean that a lot of people move from one part of the country to another part of the country. Figure 6.3 shows two examples of interactions between a single municipality, selected and gray-shaded, and the municipalities lying in a nearby COROP-area. Red areas are loosing people to the selection, blue areas are gaining people from the selection. In the left

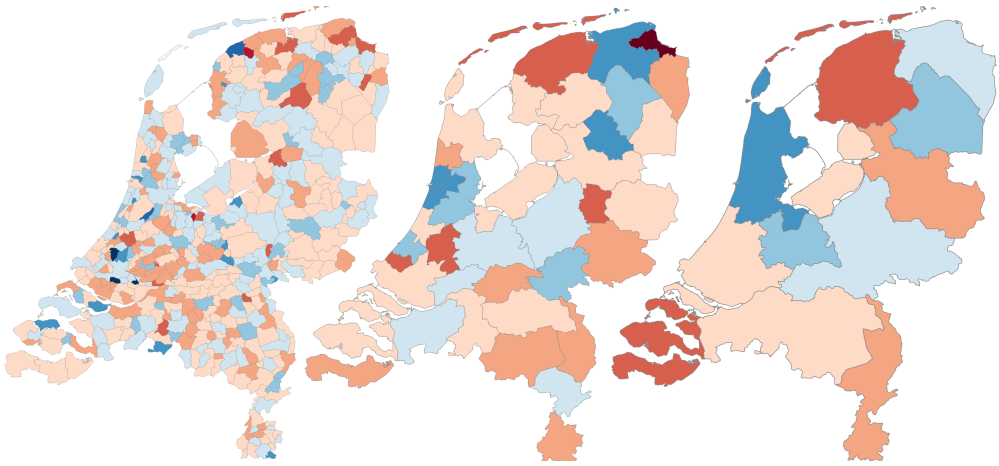


Figure 6.2: The growth due to internal migration as a percentage of the population, for municipalities, COROP-areas and provinces.

picture, the COROP-area as a whole is loosing, but due to the strong-colored blue center, one might think otherwise. In the right picture, the net effect is close to 0, while again the blue color combined with the size of the municipality, might give the user the idea the selected municipality is loosing people.

If we are not interested in these local differences between municipalities, it is better to aggregate all flows from and to the COROP and shade the area using one, homogeneous color.

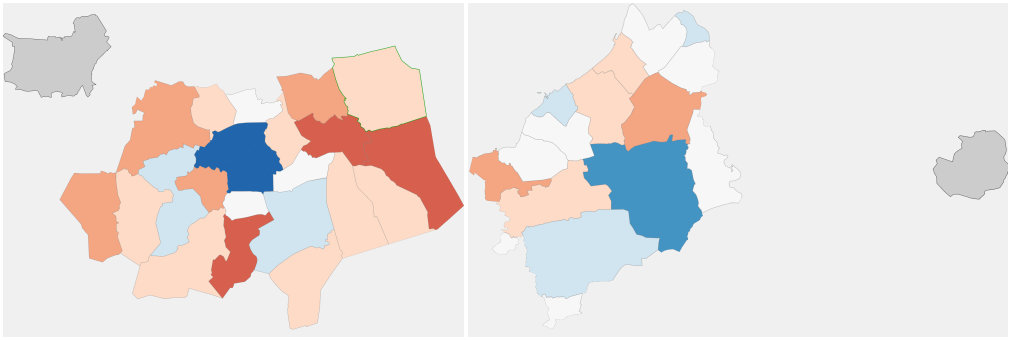


Figure 6.3: Interaction between a single municipality and the municipalities in a single COROP-area. On the left, Tilburg vs. Zuidoost-Noord-Brabant (COROP-area), on the right, Enschede vs. Veluwe (COROP-area). Blue regions grow, red regions shrink.

6.3 Higher level migration matrices

As described in Section 4.2.2, a Migration Matrix is build using a certain partition of sampled municipalities. Also, some default partitions are provided, extracted from the existing regional division of the Netherlands. This *region hierarchy* can be considered as a tree. When creating a partition on a different level, all the leaves are put in the same partition. Figure 6.4 gives an schematic example of a partition if the aggregation level of provinces is selected. In this case, the five provinces all form a block in

the partition which elements are the municipalities that lie in the province. From the regional division, these municipalities are all geographically connected. This need not be the case for custom partitions.

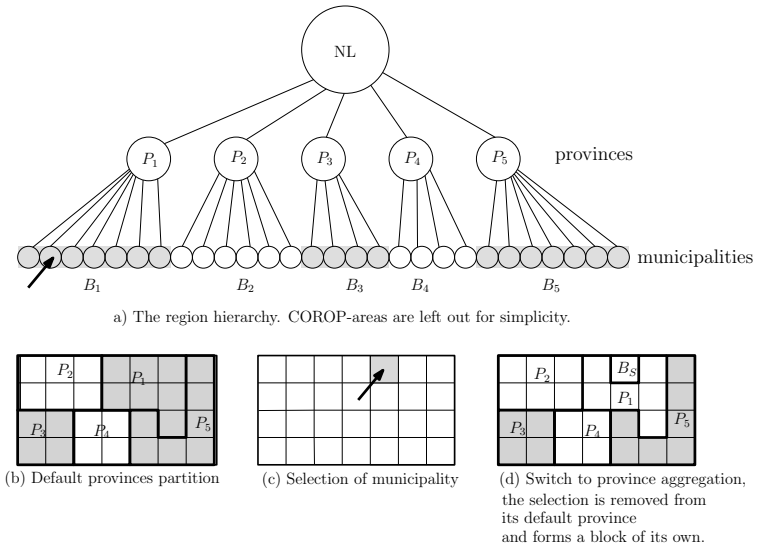


Figure 6.4: Partitioning of sampled regions using the region hierarchy

6.3.1 Mixed mode migration matrices

When a switch of aggregation level is made, a new partition is generated, and a new matrix is created out of that partition. If no region is selected, the predefined (default) partition is used. If some region was selected however, something different happens. In section 5.2.3, the effects of a selection on the current partition are described. That is, currently selected regions are removed from their current block and the selected regions are put into one big block, which is the *selected region*.

$$B' = B \setminus \{B_i \in B | B_i \text{ is selected } \} \cup B_S, \text{ for } B_S = \{a \in B_i | B_i \in B \wedge B_i \text{ is selected } \}$$

If a switch to a higher level is made, the new partition also contains this selected block S_B. From the default partition that corresponds to the level of aggregation, the regions in the selected block are removed. Thus, from an existing (default) partition C a new partition C' is created as follows:

$$C' = \{C'_i : C'_i = C_i \setminus B_S \wedge C_i \in C\} \cup B_S$$

Figure 6.4 (b-d) illustrates this. Five provinces, with 28 municipalities are used. In Figure 6.4 c, the municipality level is shown, and one of the regions is selected. Figure 6.4 d shows the partition when a higher level partition is chosen. The single region in B_S is subtracted from P_1, and is considered as a separate block.

For efficiency reasons, only the in- and out-vectors of the selection are calculated and are shown in the map. The matrix does not show this partition C', but instead shows C.

Chapter 7

Clustering

So far, only the default partitions have been used to group municipalities to analyze migration at a higher level. All of these partitions are based on the geographic location of the municipalities. Although the geographic location seems to be heavily correlated with the migration pattern (discussed in Chapter 8), the user might be interested in other partitions. Instead of looking at groups of municipalities that are close together in geographical sense, we might want to group these based on some region characteristic, like age distribution, income distribution, urbanization degree, etc.

To group similar regions, the same clustering approach as in [17] is used. Clustering can be performed on the most detailed level (municipalities), but also on other levels in the region hierarchy. It actually adds one layer above the layer that is clustered, which can be selected as being a partition.

Besides being able to group regions with identical characteristics, the clustering can also be based on the migration behavior of inhabitants of a region. For that, the in- and out-vectors $\vec{I}_i$ and $\vec{O}_i$ can be fed to the clustering algorithm. Regions with similar migration behavior are grouped together. If some smaller municipalities lie close to a municipality containing a large city, each of them might have a lot of interaction with the big city. These municipalities are probably going to form a cluster of their own.

In the remainder of this chapter the clustering algorithm used is explained (Section 7.1), with all its options and parameters. The migration clustering is described in Section 7.2. Then the interface showing the clusters is shown and finally some results of clustering with various settings and some discussion is presented in Section 7.4.

7.1 Hierarchical clustering

The cluster method used is that of *agglomerative hierarchical clustering* [15], which produces a *hierarchy* of clusters. The hierarchy can be used to quickly change the number of clusters, which greatly enhances interactivity. Agglomerative clustering is quite slow with complexity $O(n^3)$, so creating the cluster hierarchy will take some time.

7.1.1 Metrics and linkage

In agglomerative clustering, each region starts in its own cluster. Pairs of clusters are merged until only one cluster is left, the *root* of the cluster hierarchy. To decide which clusters to merge, a measure of dissimilarity of two sets of regions is required. This measure is defined by a *metric* which measures the distance between individual regions, and a *linkage* function that tells the algorithm how to calculate the distance between sets of regions. Regions, or clusters of regions, that are most similar are merged.

To provide enough flexibility, several metrics and linkage functions are implemented being, for two vectors u and v:

1. Euclidean distance: $||u-v||_2 = \sqrt{\sum_i (u_i - v_i)^2}$
2. Squared Euclidean distance: $||u-v||_2^2 = \sum_i (u_i - v_i)^2$
3. Manhattan distance: $||u-v||_1 = \sum_i |u_i - v_i|$
4. Chebyshev distance: $||u-v||_\infty = \max_i |u_i - v_i|$
5. Cosine similarity: $\frac{u \cdot v}{||u||\ ||v||}$

For two sets of regions A and B, the following linkage functions are implemented.

1. Average linkage: $\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} \text{distance}(a,b)$
2. Minimum linkage (single-linkage): $\min\{\text{distance}(a,b) | a \in A \wedge b \in B\}$
3. Maximum linkage (complete linkage): $\max\{\text{distance}(a,b) | a \in A \wedge b \in B\}$

Distances are calculated on vectors defined per region. An example is a vector of the number of people per age, with each component of the vector representing the number of people at a certain age.

7.1.2 Normalization

As in the visualizations, normalization plays an important role in clustering. If absolute numbers are considered, huge differences will exist between the major cities (large quantities) and smaller areas. This may lead to the choice of normalizing the vectors by dividing all its components by the size of the population of the related region, to correct for differences in population. This normalization is not used explicitly in practice however, as we prefer to define the normalization in terms of the elements of the vectors themselves. Also, when clustering on the age distribution of a region for instance, the size of the population is captured in the sum of all the vectors' components.

Two other normalization methods are also implemented:

- Maximum normalization: Divide by the maximum component of a vector $\vec{u}$: $\max \vec{u} = \max_i \vec{u}_i$
- Sum normalization: Divide by the sum of all components of a vector $\vec{u}$: $\sum \vec{u} = \sum_i \vec{u}_i$

Division by the maximum component of a vector is used to emphasize the shape of the vector. Van Wijk and van Selow [17] also use this normalization step in their time-series clustering. Normalizing by the maximum component is very sensitive to outliers. One very large value can significantly diminish all other components' values. Also, vectors are harder to compare as the sum of all components after normalization is not necessarily equal. We only know that $\sum_i (\vec{u}_i / \max \vec{u}) \geq 1$.

Dividing by the sum of all components results in a (discrete) probability distribution. In case of clustering on the age distribution of a region, the variable is `AGE`, and for each age x, the probability is equal to: $P(\texttt{AGE} = x) = \vec{u}_x / \sum \vec{u}$. This nicely sums up to 1 for each normalized vector. The method is less sensitive to outliers than normalizing using the maximum component and is still able to capture the shape of the vector.

Not all metrics require a normalization step however. The cosine similarity by itself normalizes the vectors by dividing by the (Euclidean) length of the vector.

7.1.3 Dendrogram

The output of the clustering is a hierarchy of clusters that can best be represented by a dendrogram, as in Figure 7.1.

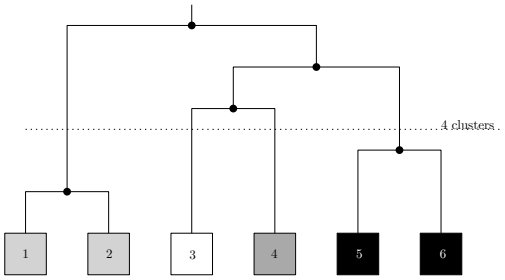


Figure 7.1: Example of a dendrogram for 6 regions. The colors are assigned based on the choice of displaying 4 clusters (dotted line).

7.1.4 Color assignment

Section 7.3 gives an example of how the map is colored according to the clusters (partition coloring). Assigning a unique color to each cluster so that clusters can be easily distinguished is a non-trivial problem. Especially if the regions within a cluster are not close to each other, colors should be easy to distinguish. In a first attempt, the dendrogram was used to assign colors to clusters, using a rainbow color map. A default tree traversal algorithm can be used to assign a color to each cluster. Problem is that

the dendrogram might be very out-balanced and we can not really trust its structure to assign good to distinguish colors for the first levels.

The approach that is chosen is the following. The start situation is always that of four clusters. These are assigned distinctive colors, from Color Brewer [3]. Each time a cluster is split, the largest child obtains its parent's color. The other child obtains a its parent's color with a random amount added to each of the three color components (R, G and B). This turns out to be sufficient to see which cluster is split in a step. Afterwards, the user may change the color manually.

7.2 Migration clustering

Clustering similar areas based on some demographic property and inspecting the major flows between groups of areas is useful, but clustering based on the migration pattern might be far more powerful to provide insight into the migration data. For this, the in- and out-vectors of a region are used ($\vec{I}_i$ and $\vec{O}_i$). An example from a clustering based on $\vec{I}_i$ is shown in Figures 7.2a and 7.2b. As normalization, the sum of the element of the vector is chosen, the point distance metric is Manhattan and the linkage function is complete linkage.

Filters are applied before extracting the in- and out-vectors and are therefore also applied on migration clustering. The results of clustering without any filters set and clustering with an age filter of 16 to 23 years set differ significantly as can be seen in Figure 7.2. Because the clustering algorithm only needs the migration vectors per municipality once during computations, filters may be removed after the clustering is done to inspect all flows from and to a certain cluster.

The matrix can be used to validate the clustering. As the regions in a cluster should have similar in- or out-flow patterns, sorting on clusters should form groups of rows and/or columns that are alike. Figures 7.2c and 7.2d give examples of the accompanying matrices in which the municipalities are sorted by cluster. The values are normalized using the summed sizes of the populations of both regions.

As the clustering is performed on the in-vector per region, a better approach to validate the clustering would be to normalize the matrix' cells in the same way as the vectors were normalized before the clustering. This can be seen in Figures 7.2e and 7.2f. These clusterings seem to be correct as the regions in the clusters are much alike (most of the migration happens within the clusters).

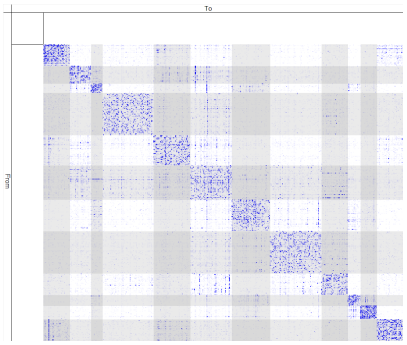
7.3 The Cluster panel

The cluster panel is dedicated to the clustering feature of the tool. The cluster panel shows some of the 'content' of each cluster (Figure 7.3). The map in Figure 7.2a gives insight which municipalities are located in what cluster.

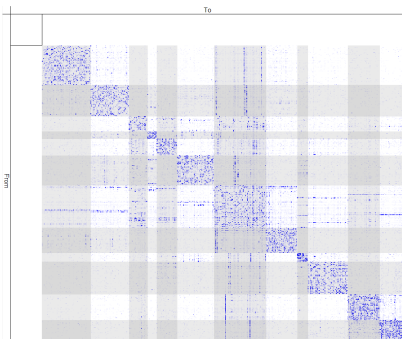
For each cluster, the total inflow, outflow and internal flow are displayed and the number of regions in the cluster, the summed population and age pyramid are given. The user has the option to split a certain cluster or let the algorithm decide which cluster to split.

(a) No Filters

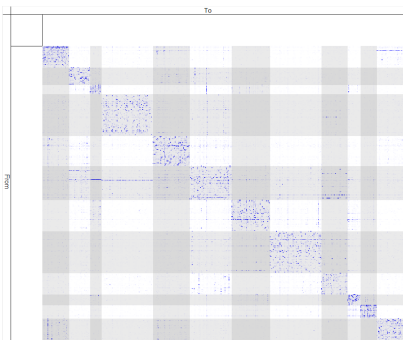
(b) Agefilter: $16-23$



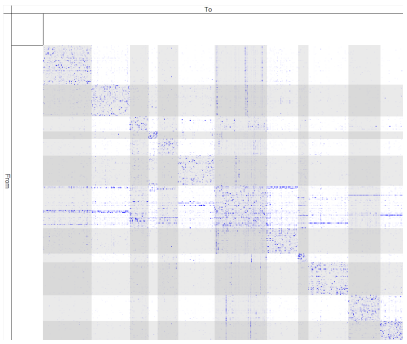
(c) Matrix, population normalized



(d) Matrix Agefilter: $16-23$, population normalized



(e) Matrix, column normalization



(f) Matrix Agefilter: $16-23$, column normalization

Figure 7.2: The matrix used to validate the output of the migration clustering

The list of clusters might become long after several split operations. Therefore, it can be sorted on attributes, such as number of municipalities, total population, total flow and total flow per municipality. Sorting on name is also possible, but as names are randomly assigned, this is not too useful.

Finally, the color of the cluster (which corresponds to the shade color in the map) can be changed. This way, clusters may be easier identified on the map.

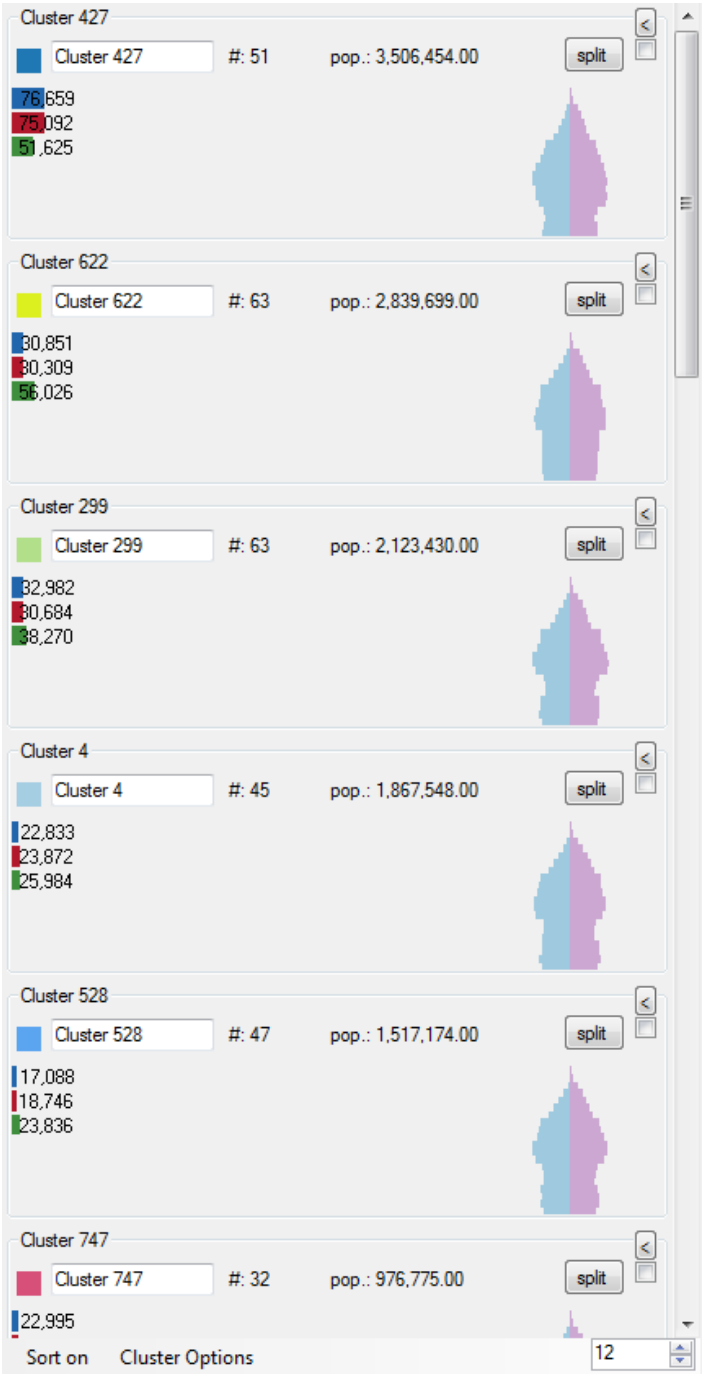


Figure 7.3: The cluster panel gives an overview of the clusters and their contents.

7.4 Results & Discussion

Although clustering can be a very powerful tool, the overall results are not always that convincing yet. It may require an expert to see certain patterns appearing in some cases however. Here, some more interesting partitions based on hierarchical clustering are shown.

7.4.1 General

There are many options and parameters that can be set to influence the clustering algorithm itself. As mentioned in Section 7.2, the filters are also part of the options, if we are clustering on migration data, that is. Figure 7.4 shows an overview of the options to be selected before starting the clustering algorithm. There are many combinations possible, and some give better results than other combinations. The first three columns provide options to select what has to be clustered and what attribute is clustered on. The last three columns contain options that greatly influence the result of the clustering. The effect of different options is described below. Figure 7.5 gives an overview of nine combinations of distance metric and linkage function.

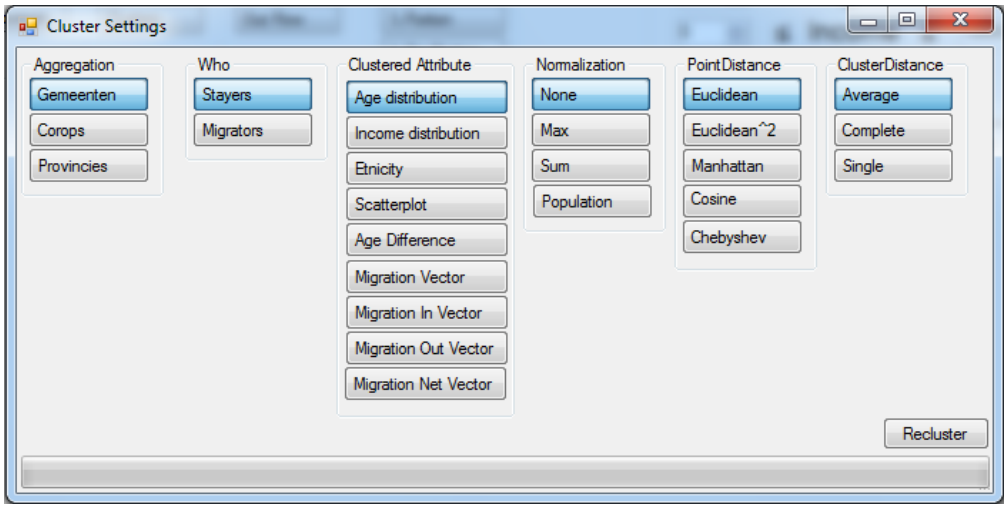


Figure 7.4: The cluster options panel

Normalization

The normalization of vectors is already discussed in Section 7.1.2. For the remainder of this chapter, the used normalization method is the sum normalization, as it is less sensitive to outliers than maximum normalization.

Linkage function

The three possible linkage functions: single (minimum), complete (maximum) and average linkage have great impact on the form of the dendrogram and the clusters resulting from it. In Figure 7.5, the linkage function is varied from top to bottom. Single linkage seems to work very bad and results in a very unbalanced dendrogram in which single municipalities are added to one large cluster one by one (remember we are clustering bottom-up). As the distance from a point to a cluster is defined by the minimum distance to any point in the cluster, the probability the point is closest to a large cluster is pretty big.

Complete linkage does the opposite and postpones the merging of large clusters. Average distance linking is somewhere in between and seems slightly more sensitive to outliers than complete linkage.

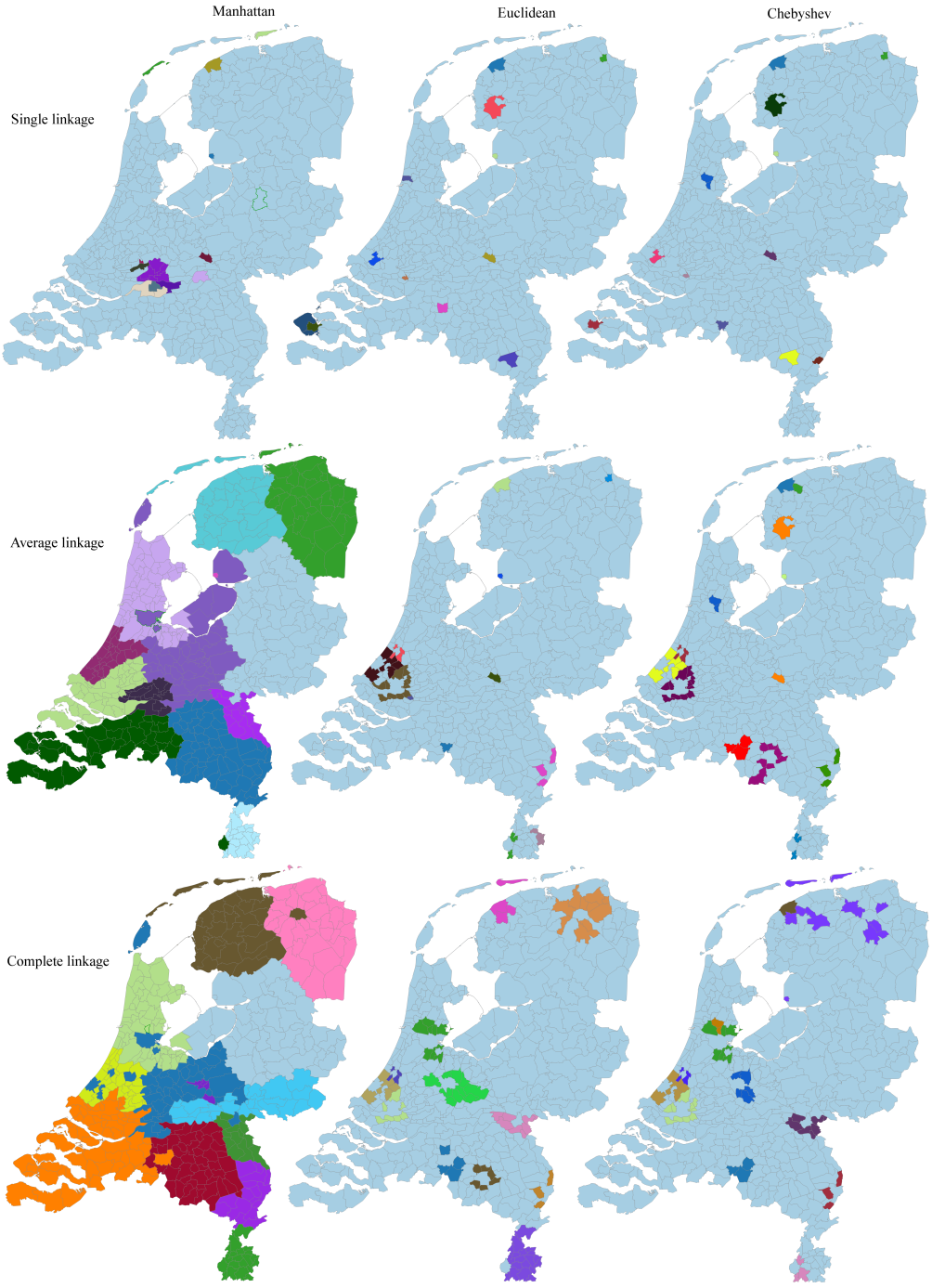
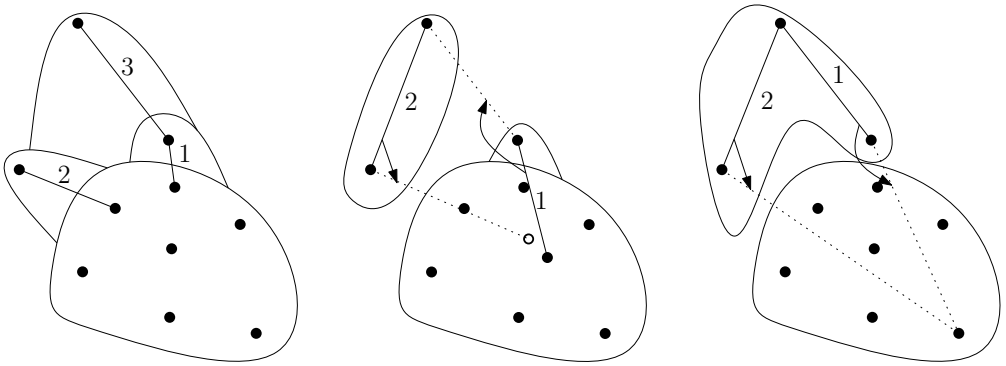


Figure 7.5: Different variations of distance metric and linkage function. The clustering is based on the in-vector of a region. The normalization used is sum normalization, the number of clusters is 12.

Figure 7.6 gives a schematic overview of the differences between the three functions.



(a) Single linkage (b) Average linkage (c) Complete linkage

Figure 7.6: Different linkage functions delivers different results. The single linkage usually forms a single large cluster of which outliers are split one by one (top down). The complete linkage tends to form larger groups which are merged towards the top of the dendrogram. Average linkage is, as expected, somewhere in between. The circle in the middle picture is the updated average of the cluster after adding the first point. The points are added in the order of the number that is next to the line connecting the points, which is the smallest distance and is a solid line. Solid lines competed with dotted lines to which an arrow points.

Point distance

It is difficult to define a good distance measure between the regions themselves. In the first place because different variables can be used for the clustering. Most of them however, are vectors of integers, or reals (after normalization). Therefore, standard (Euclidean) distance measures seem obvious. The question is whether these lists of quantities of people (in a certain age, or migrating to some region) can be regarded as vectors (containing magnitude and direction) in n-space. And thus, whether a Euclidean distance measure really 'means' something.

Nonetheless, a plausible explanation of the effects of choosing different distance measures can be given. In Figure 7.5, three variations of the Minkowski distance measures are compared, 1-norm (Manhattan distance), 2-norm (Euclidean distance) and ∞-norm (Chebyshev distance).

In Figure 7.5, the distance metric is varied from the left to the right. For single linkage, there is not a big difference between the various point distance metrics. As we discussed, it is very sensitive to outliers, and thus forms solely clusters of one or two municipalities. In case of average and complete linkage, clusters are getting smaller and more focused around the larger cities if we move from left to right. We will analyze the most extreme case, complete linkage with the Chebyshev distance metric, and try to find an explanation for the existence of these small focused groups and one large remaining cluster.

It seems like municipalities that have a very large part of their inflow (we are clustering on the in-vector) coming from a single other municipality are grouped together for the higher order Minkowski distances, as all clusters are concentrated around big cities. In

some cases, these municipalities have over 50% of their inflow coming from a single place. This is exactly the number that is present in the vector, as we are using the sum normalization, resulting in a probability distribution.

Lets first divide all municipalities into two sets. In set Y, we put municipalities which have a peak at some index. Lets say a vector has a peak if some component is bigger than a certain value α. A reasonable value for α would be between 0.25 and 0.50 . The rest of the municipalities goes into set Z.

We consider a group of municipalities $Y_i \subseteq Y$, sharing a huge peak at the same index i in their vectors. As we are using the Chebyshev distance metric, the distance from $x \in Y_i$ to any other point $y \in Z$ is defined by $|x_i - y_i|$. We have $x_i \geq \alpha$. We assume no special relation of y to the municipality at index i and thus, we expect y_i to be around $1/441$ (uniform distribution). Thus $|x_i - y_i| \approx x_i \geq \alpha$. We are a bit sloppy here, which is fine as the equation will hold for most of the $y \in Z$ due to the scale and distribution of the data. If it is not true, y and x are more alike than we expected and they might end up in the same cluster after all, which is no problem.

The distance from $x \in Y_i$ to some $y \in Y_j$ (peaking at index $j \neq i$), is expected to be $\max(x_i, y_j) \geq \alpha$, following the same reasoning.

For pairs of municipalities $(x, y) \in Z$, there is no index i such that $x_i \geq \alpha$ or $y_i \geq \alpha$. Thus $|x_i - y_i| < \alpha$. This strict bound is quite pessimistic, the distance is probably way smaller than α.

The distance between municipalities $x \in Y_i$ and $y \in Y_i$ is probably not defined by their peak values at index i, but by some other index $j \neq i$. We know that these peak indices represent large cities and from inspection we have seen that per municipality there is often only one such peak per vector. Thus, it is safe to assume that all other components of the vector x_j and y_j are quite small, especially compared to the other distances. We assume $|x_j - y_j| \approx \varepsilon$.

Thus, what will happen in agglomerative clustering with max linkage? First, all municipalities in groups Y_i are merged into (small) clusters. The distance of municipalities in Z to these clusters is bigger than α. Thus, the municipalities of Z are merged into one big cluster (all having a distance smaller than α between them, even for the pair with the maximum distance). Finally, the small clusters, defined by the Y_i's are merged one by one into this big cluster of Z.

Of course, the peak values are also influencing the clustering when using Manhattan distance. But, in case of Manhattan distance, the rest of the $n - 2 = 439$ entries also contribute equally to the distance, thereby compensating for the missing peak values. The Euclidean distance is in between both, but judging from the results in Figure 7.5, peak values seem to dominate this distance metric also.

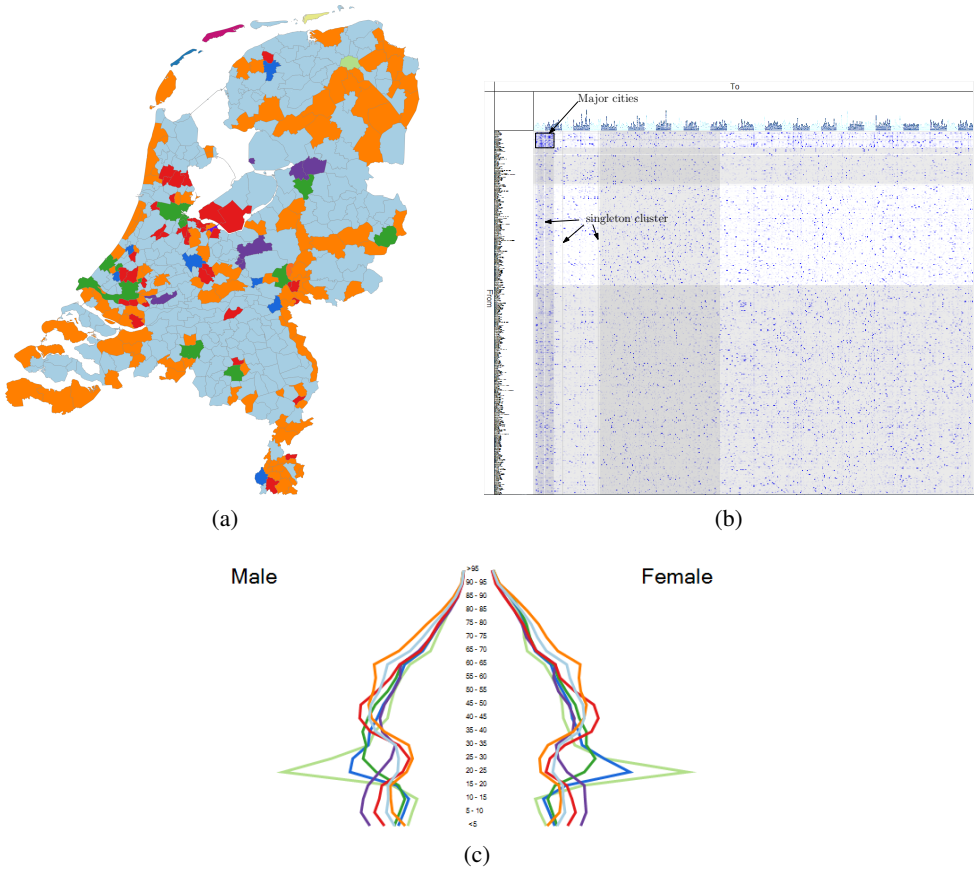
The fourth distance measure is the cosine similarity. This turns out to be not working. It is also very difficult to interpret an 'angle' between the migration vectors or age distribution vectors.

Depending on the goal of the clustering, which may be finding outliers or finding geographically connected, equally sized regions, a different point distance metric can be chosen.

7.4.2 Age clustering

Clustering the regions based on their age and gender distribution might show a strong correlation between the age distribution of a region and its migration pattern. Regions with similar age distributions are expected to be grouped together in clusters. Figure 7.7 shows the result of showing 12 clusters on the map, of which the ones containing more than one municipality are plotted together in an age pyramid. The matrix is sorted on the clusters.

Just as the matrix is used to validate the migration clustering, the age pyramid is used to validate the age distribution clustering. Clearly, the biggest deviations are in the age categories of 0 to 45 years. The experts often emphasized the lack of youth at the borders of the country (yellow line). Municipalities containing large cities (light green) have an abundance of young people on the other hand. So, the clustering algorithm produced a reasonable result.



(a) (b)

(c)

Figure 7.7: The output of an age distribution clustering, showing 12 clusters. Settings used: sum normalization, Chebyshev distance, complete linkage.

After multiple runs of the algorithm with different settings, the age distribution clustering seems to be vulnerable to outliers. Especially the municipalities with a very small population, as a consequence having capricious age pyramids, tend to form little clusters on their own because of their irregular shapes. Maximum linkage is therefore the best choice for cluster distance, as it is less sensitive to outliers. Another problem is the existence of a large group of similar municipalities. Although this is not a bad

thing in general, it makes the matrix useless due to the scale. Groups of just a couple municipalities are barely visible in an overview and are dominated by the large groups.

From the matrix in Figure 7.7 we can not derive a correlation between the age distribution of a region and the migration pattern. In the top left corner a denser square is visible. This is the migration within the cluster containing all the major and student cities. As the cell's values are corrected for the size of the population, this is notable. The rest of the matrix does not show much of a pattern unfortunately.

Chapter 8

Evaluation

To be able to judge the value of the tool, opinions of demographic experts are taken into account. All experts who have seen the tool or worked with it were quite enthusiastic and saw clear potential in the use of visualization for analyzing migration (and other flow) data. Especially being able to add attributes and relating region information to migration data are powerful mechanisms. The demographers have a lot of presumptions about the migration data at the moment and are therefore especially interested in being able to quickly inspect the data. Using clustering and the matrix representation to discover different patterns could not count on the same amount of interest. Nonetheless, some do see the potential of it. If we would have had better results, that did not fall in the category 'obvious', it may have been a different story.

In the remainder of this chapter some previous studies of demographic experts are used to validate the effectiveness and efficiency of the tool. Although we do not possess the exact same dataset, a similar analysis can be made using our dataset. We assume that the situation has not changed dramatically. The conclusions do not have to match exactly, it is more important a similar analysis can be performed. One issue with the tool is that time is not included, while most analyses do provide information year by year.

8.1 Student cities

Demographic experts clearly state that student cities attract a lot of young males and females, as they are migrating mainly because of their study in one of those cities. Figure 8.1 shows the growth as a percentage of the total population per municipality. In Figure 8.1, the municipalities containing universities are surrounded by an ellipse. All of these are having positive growth. A city like Groningen (the most northern one) is an extreme example where the group of 18 to 25 year old persons represent a growth of 2.27% (from 2008 to 2009) with respect to the entire population, while the total growth is only 1.52%. This means that in other age categories, Groningen is loosing people.

To classify the student cities according to their growth rate, we can use the scatterplot. Instead of using hierarchical clustering, the clusters can be created based on the position of a point representing the municipalities in the scatterplot and the positioning of the split lines in the scatterplot. Figure 8.3 gives an example. The map in Figure 8.3b

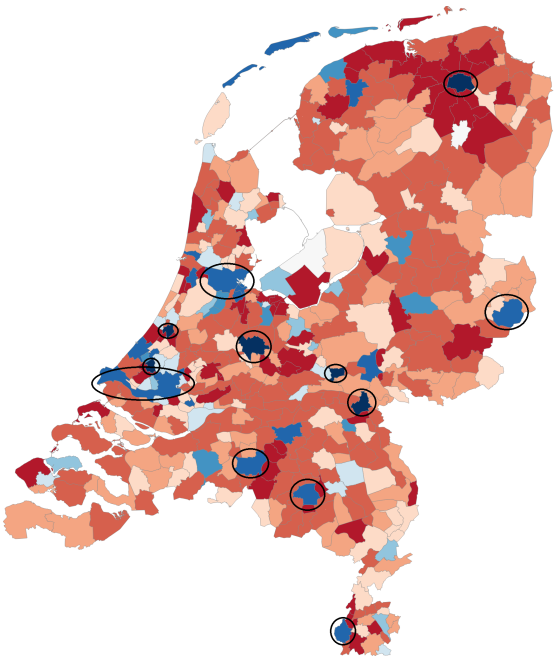


Figure 8.1: Student cities are growers in the age range of 18-25. Growing municipalities are assigned a blue color, shrinking municipalities are colored red. Student cities are highlighted by an ellipse.

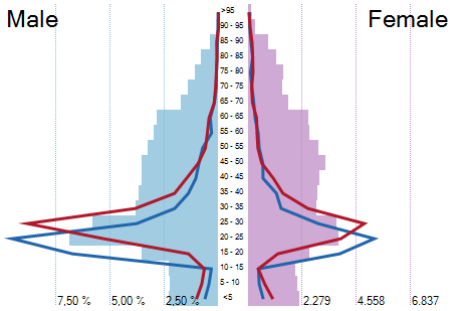


Figure 8.2: The age pyramid of Delft, clearly showing an over-representation of males in the age of 15-30 years.

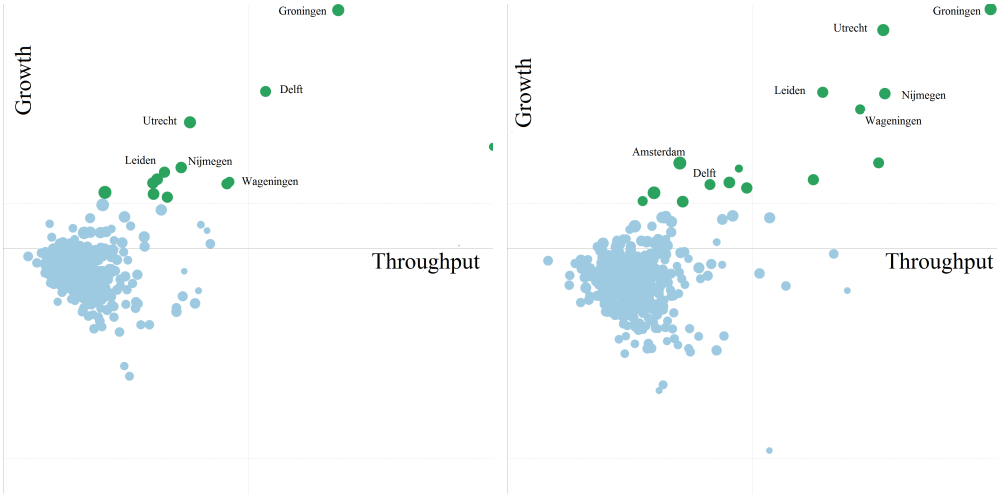
clearly shows the student cities as growers. The green color represents the highest growth category, as can be seen in Figure 8.3a.

Instead of looking at overall growth, we would also like to verify the claim that some cities are more attractive to females, while others are more attractive to males, probably due to the types of studies that are offered by certain university. If we filter on males or females, new scatterplots are generated. The results are shown in Figures 8.3c and 8.3d. We can see that some cities (like Delft) drop in vertical direction when going from the male to the female plots. Obviously, Delft has a bigger inflow of males (1,06% growth) than of females (0,46% growth). Delft is especially known for its technical studies and is clearly a 'male' city. The age pyramid supports this statement (Figure 8.2).

Other universities attract more females, such as Nijmegen, Wageningen and Leiden. This can also be seen in Figure 8.3.

(a) Males & Females

(b) Accompanying map



(c) Males

(d) Females

Figure 8.3: Using the scatterplot, the regions are colored according to their growth and throughput rate. The positions take the filters into account, and thus we can generate scatterplots for the male and female migrants. It is clear that student cities are the biggest growers in the age range of 18-25.

8.2 The city and province Groningen

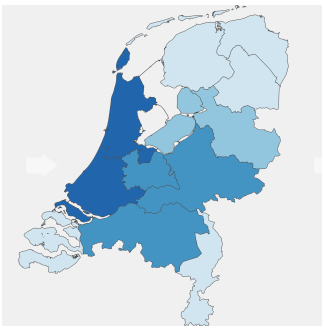
In the north of the Netherlands lies the province of Groningen, with region capital Groningen. In the second quarter of 2008, Latten et al. [11] published a report about the city Groningen, that is fulfilling the role of escalator of the Northern Netherlands. The first thing that is remarkable is that there is no map included in the entire paper.

Some of the conclusions were as follows:

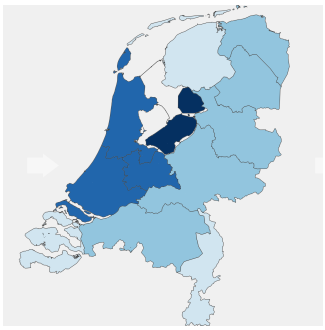
1. The northern part of the country failed to attract a lot of foreign and domestic migrants. The region overall seems to be quite stable in terms of migrations.
2. In the north, the city of Groningen plays an important role in migration.
3. A lot of young people in the north migrate to the city of Groningen.

4. Their stay is not too long. Within 4 to 5 years, more than half of the immigrants moved out again.

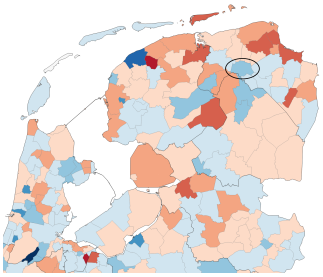
5. A lot move to the western part of the country.



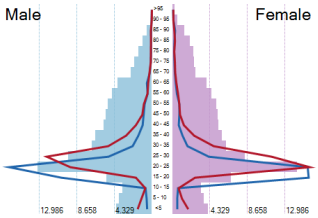
(a) Absolute growth



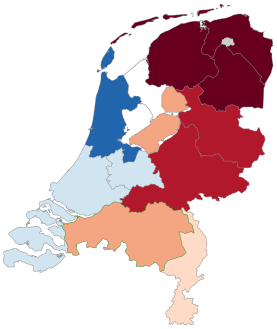
(b) Relative growth



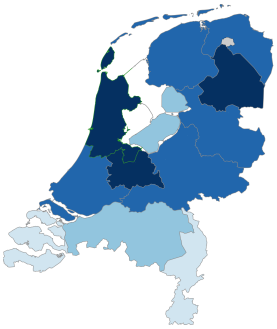
(c) Growth per region due to domestic migration



(d) Age pyramid of Groningen



(e) Groningen to/from provinces: 15 to 25 years old



(f) Groningen to/from provinces: 25 to 35 years old

Figure 8.4: The north of the Netherlands does not grow as quickly as other parts of the country, as can be seen in (a) and (b). These figures include foreign migrations and births and deaths. Figure (c) shows that the city of Groningen is not a region with strong growth coming from migration within municipalities only. Figure (d) shows the age pyramid of Groningen, showing a large inflow of younger people and a large outflow of people who are $\pm$ 5 years older. This is confirmed by figures (e) and (f), where the direction of the flow is reversed.

To start with conclusion 1, we are interested in the absolute and relative growth of the three northern provinces. Unfortunately, due to the nature of the data set, we can not filter out deaths and births when looking at foreign migrants. Nonetheless, Figures 8.4a and 8.4b indicate that the northern provinces are less attractive than other parts of the country. The fact that the city of Groningen plays an important role in migrations in the north is a bit subjective. Furthermore, as conclusions 3 and 4 indicate, the city seems to attract young people, most of which emigrate within 4 to 5 years. Therefore, the total effect might be quite stable. Figure 8.4c does not show for instance a high growth percentage for the municipality of Groningen, compared to other municipalities in the north.

If we take a look at the age pyramid of the city in Figure 8.4d, we can see high peaks for inflow and outflow with a slight age difference between them. If we focus on the younger group, of 15 to 25 year olds (Figure 8.4e), we indeed can confirm conclusion 3, as the red areas loose a lot to the selected region, which is the city of Groningen.

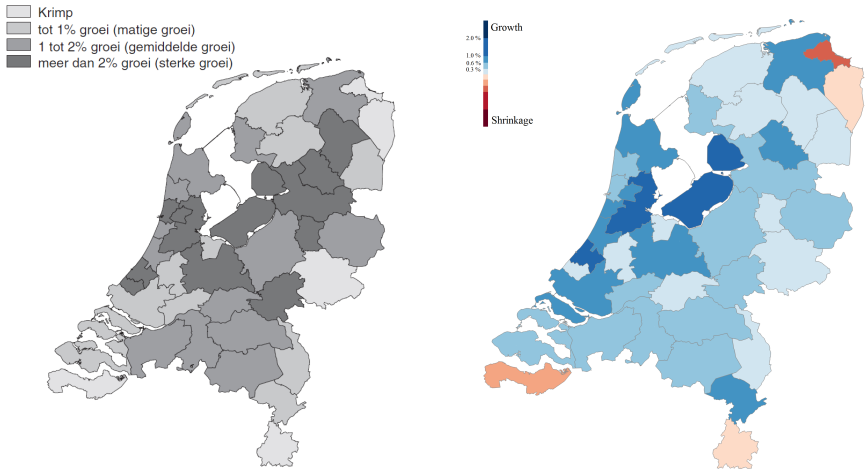
Although the article focuses on a group of people moving into the city for studying and moving away within 4 to 5 years, we can partly confirm this effect by showing that people in the age of 25 to 35 are moving out of Groningen. This is shown in Figure 8.4f. In this image, conclusion 5 is also confirmed.

8.3 Growth

In 2011, Latten and Kooiman [12] published an article about the attractiveness of regions and the demographic consequences of that. The study is performed on the COROP-areas in the Netherlands and is especially focused on the attractiveness (expressed in terms of growth and throughput) in combination with age. Both of these are easy accessible by the tool. Again, we are not in possession of the original dataset, but we should be able to replicate most of the images or create similar images at least.

Figure 8.5a shows an image from [12]. The lightest areas are shrinking while the darker ones are all growing. Not only domestic migration is taken into account, but also births, deaths and foreign migration. A similar picture can be made using our dataset for the year 2009. It is featured in Figure 8.5b. Instead of using 5 years, only 1 year is displayed. Therefore, we can not expect the same growth percentages.

If we take a closer look at Figure 8.5, we can clearly see the same regions being marked as 'shrinking'. In the left, these are the lightest regions, on the right, these regions are shaded red. Differences in the darkness (growth) of the other regions can be explained by differences in the dataset.



(a) Latten and Kooiman (b) FlowVis

Figure 8.5: Comparison between the study of Latten and Kooiman [12] and output of the tool. Although the left image is based on data from 2005 to 2010, a similar pattern appears in the right image with data from 2008 - 2009 only.

Some other interesting conclusions can also be investigated using our tool. The article states that most people migrate within a COROP-area. For this, we should have to include statistics on people migrating within the municipalities. Furthermore, the article states that certain cities have very large inflows of young people. We already saw this in Sections 8.1 and 8.2.

Another interesting observation is that the distance people migrate over decreases with age. Although we do not have absolute distance measures included (we have no real-life distances included between municipalities), we can underestimate migrants exceeding COROP-borders or province-borders.

Chapter 9

Future work

Due to the scale of the project and the limited amount of time, there are a lot of open questions and a lot of ideas to give a try. Here, the most promising ones are summarized.

9.1 Attributes and filters

As time has been a limiting factor in the project, a lot remains to be done. To fully exploit the potential of the dataset, more persons' attributes can be taken into account, such as ethnicity, household situation and education. This way, demographics get more attributes and filters to play with.

Filtering is only performed on the migrants themselves, not on the regions they migrate between. This is both a choice to improve implementation simplicity, as to improve the understandability of the tool. However, more advanced users might be interested on the effect of migration for a certain part of the population.

To be able to do this, the way information is stored about regions needs to be changed. Either we can choose to store all people resident in a region, or large cross-tables should be constructed. For the current situation, in which gender, age and income are taken into account, the latter might be doable. As soon as we start to add extra attributes, things will soon start to be problematic as the tables will get sparser and sparser while the memory usage grows exponentially.

9.2 Expectancy

The *expectancy matrix*, introduced by plugging in the Chi statistic into the migration matrix is a too much simplified model. Basically, what is assumed is that migration is purely dependent on the size of the population of both the origin and destination. When looking at previous studies, the visualizations, and by using merely common sense, this is of course an oversimplification. People tend to stay close to their old residence. Distance is of great influence on the probability someone migrates and can be taken into account when trying to improve the prediction. Instead of absolute (euclidean) distance, another distance metric might be used, for instance whether two municipalities lie in the same COROP-area.

Not only for the prediction used in the expectancy matrix the distance over which someone migrates is important. The average distance over which people migrate is characteristic for certain regions in the country. In the north, people tend to migrate further away than in the western part of the country.

Also, the age distribution of the population in a region has effect on the migration pattern. Young people form the biggest group of migrants, and this could be taken into account.

9.3 Time

A great addition to the tool would be the dimension of time. Time is included already as the changes (migration) between two moments in time are investigated. Being able to add more time steps would enable the user to discover changes in migration patterns over the years. Also, the user would be able to regard different spans of years. A choice that has to be made in the data preprocessing step right now.

Besides being able to look at the development over the years, the CBS data is suited to follow persons during their entire life's. So instead of regarding every migration as a unique person moving from some region to some other region, we could track where people went in the past. This is important if we are trying to replicate the study from Latten et al. [11]. Another type of filter could be created like: Filter on the people who migrated towards the city of Groningen in 2005. Then, in 2010, we can see if indeed 50% is moving away again. We could even see if they are moving back to the same place they came from when moving to Groningen 5 years earlier.

Mimicking this kind of behavior is not totally impossible in the current tool. As an input dataset, we could take the exact group of people who moved to Groningen in 2005 and see where they live in 2010. This is rather impractical however, and also breaks with the intention of the tool, namely: not having to regard each region one by one.

Thus, adding time would offer a wide range of possibilities. Care must be taken to prevent the tool to become too complex however.

9.4 Other applications

Although the tool and this document is devoted to migration data and its visualization, the used techniques are applicable on all kinds of flow data. The regions can be replaced by other types of entities and the flow units can also be replaced by something else. To stay with migration, instead of using geographic regions and a map of the Netherlands (or any other country), one can think of investigating the migrations between different types of residencies, like flats, apartments, either rented or bought, etc.

The regions can also be replaced by companies, industries or business sectors and the migrants can be replaced by transactions.

9.5 Cluster distance measures

As mentioned in Chapter 7, there is an open question regarding the distance metrics used to compare different regions, which are represented by vectors of length n. Euclidean space is more or less assumed, but is not necessarily the right way to regard these vectors.

Chapter 10

Conclusion

We have formalized the problem of migration visualization, which can be reused for other applications of flow visualization. Existing techniques were not sufficient to provide the desired insight. Therefore, the choice was made to combine simple, existing techniques to create a tool that both provides overview, but also enables the user to explore the data in different directions. A simple chloropleth map has proven to be an effective tool to display all kinds of information, but is unable to provide a view of the entire dataset. The matrix can give this overview, and can also be used to discover patterns in the data and validate clustering algorithms, that cluster regions based on their flow patterns.

Having worked with a group of migration and demographic experts has really helped to find out what the key features of the visualization should be. Also, they helped with design decisions, of which the project has been full. All kinds of normalizations have been discussed, but also the color choices are based on interviews with the experts.

To be able to do more than answer quantitative questions about the migration data, the opportunity to load arbitrary variables and the possibility to cluster the regions based on their properties or migration behavior are build in. These options enable the user to relate growth due to migration to virtually any (scalar) variable thinkable for the regions. Also the clustering allows groupings of regions, other than the geographic coherent divisions.

Unfortunately, there have been some speed bumps on the way. Due to privacy regulations at CBS, it was difficult to work with the original data set. Also, not all techniques worked that well. The clustering algorithm did a good job at clustering similar regions, but no real correlation was detected during the development of the tool or the writing of this document.

Nonetheless, a good first step towards a full-featured migration analysis tool has been made, with a lot of viable ideas still open for research.

Appendix A Data structures

Although the data model is build on the usage of a migration matrix that is generated from a subset of all persons and a partition of the municipalities, the implementation is done using a graph. On the lowest level, the Migration Graph contains Areas (which represents the municipalities) as vertices and Migration Edges as edges. A Migration Edge consists of a source and target Area, and a list of Persons migrating 'over the edge'.

Instead of filtering the entire set of persons and constructing a new graph for each filter change, filters are applied on the list of persons on the edges. This way, only persons of the currently required edges need to be scanned.

The graph is implemented using the Quickgraph library [13] as a bidirectional graph. Internally, two dictionaries are contained, with a mapping from vertices to lists of edges. One dictionary contains the in-edges and the other one holds references to the out-edges.

The graph model is chosen over a matrix data structure as it is more efficient in scanning all in- or out-edges. Checking for an edge would be cheaper using a matrix, but as most operations require a scan of all edges, the graph is the best choice overall.

Appendix B Reading in large datafiles

Reading in and constructing the initial data structures is quite a time-consuming process. Therefore some standard techniques are used to speed up the process. When a dataset is loaded for the first time, a copy is made in which only the migrants are stored. Given the fact that around 10% of the people migrate, this reduces both space and scan time by a factor 10. The copy is stored in sorted order on the first year's municipality and then second year's municipality.

When reading in this 'reduced' dataset, the edges can be read in one by one, instead of having to do a lookup whether or not an edge already exists and add the person, or if a new edge is to be created. This reduces the graph's construction time by a factor $\mathcal{O}(|E|)$, in practice it seems to be 3 times faster (17 to 5 seconds) for building the graph between 441 vertices based on around 1.1 million migrants.

Appendix C Data preprocessing tool

A special tool was build to preprocess the data. The format of the original dataset was quite inconvenient to work with. Not only was the original dataset in fixed-width format, it contained a lot of derived variables to aid the analysis it was assembled for. As the visualization tool works with a more or less fixed set of attributes per person and region, a more structured input file would be a lot faster and easier to process. The input file was transformed to a comma-separated- value file by using external tools.

The data preprocessing tool, shown in Figure 10.1 and from now on referred to as 'tool', reads in an input file's first line and assumes it to be the list of column headings separated by a certain separation symbol. The headings appear in the list view on the

left of the form. From there, the column headings can be dragged and dropped in one of the boxes on the Persons attributes or Yearly data sections. The persons attributes are a fixed set of attributes that are assumed constant. Age is of course not a constant, but some choice has to be made when investigating a timespan of more than one year.

Yearly data contains the attributes that change over time and of which we are also interested in the change over time. For instance, if the two municipalities a person lived in are unequal, this person migrated. To know what the effect of the migration is on the income, income data on a yearly basis is also a requirement.

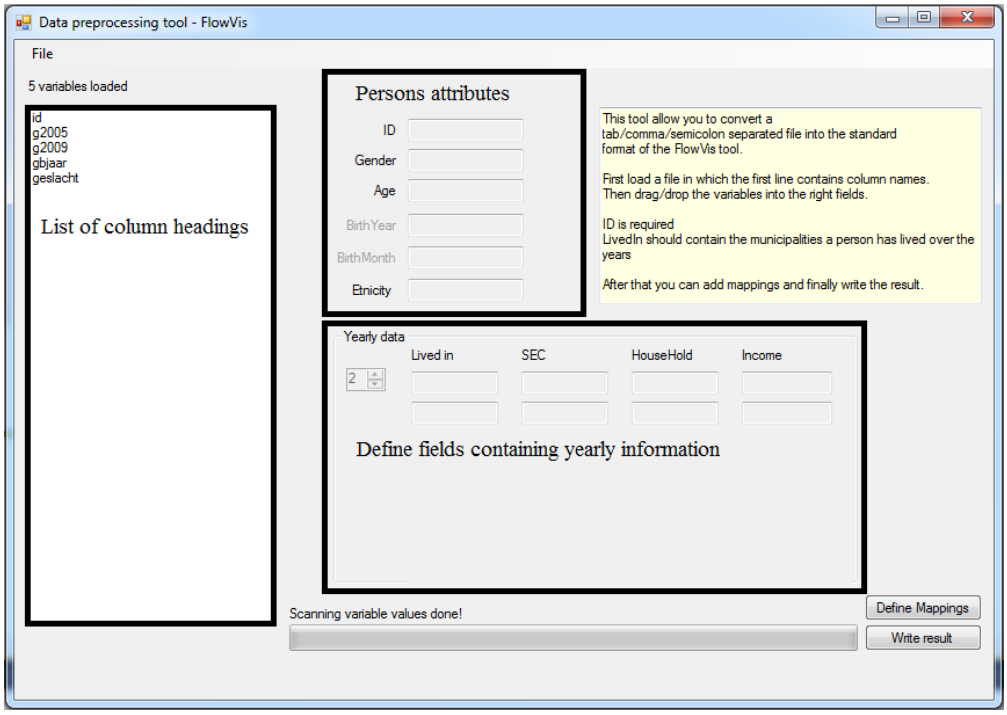


Figure 10.1: A snapshot of the preprocessing tool.

While the headings are distributed over the different fields, the rest of the data is scanned, and all unique values are determined per variable.

After filling in as much fields as required, mappings may be defined. These are meant to change the representation of certain variables to fit with the tool. As the tool is aware of all unique values per variable, a mapping can easily be constructed. For instance, genders are stored as 0 (male) or 1 (female). If the input file uses 1 and 2, a mapping can resolve this mismatch. The mapping is also used to deal with reclassifications in the regional division. By default, the first year's municipality codes have to be translated into the codes of the second year of sampling.

Finally, the result file can be written to disk.

Appendix D Implementation

The software is written in Visual C#, using Microsoft Visual Studio 2010 Professional edition. The Tao OpenGL and FreeGlut library are used to provide the OpenGL bind-

ing for the .NET framework. The user interface is implemented using Windows Forms that ships with .NET.

Bibliography

[1] Gennady Andrienko and Natalia Andrienko. Spatiotemporal aggregation for visual analysis of movements. In *In Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST 2008), IEEE Computer*, pages 51–58. Society Press, 2008.

[2] Ilya Boyandin, Enrico Bertini, Peter Bak, and Denis Lalanne. Flowstrates: An approach for visual exploration of temporal origin-destination data. *Computer Graphics Forum*, 30(3):971–980, 2011.

[3] Cynthia A. Brewer. Colors from www.colorbrewer.org, 2012.

[4] Kevin Buchin, Bettina Speckmann, and Kevin Verbeek. Flow Map Layout via Spiral Trees. *IEEE Transactions on Visualization and Computer Graphics*, 17:2536–2544, 2011.

[5] Arie de Graaf. Gezinnen in beweging. *Bevolkingstrends, 2e kwartaal 2011*, 2011.

[6] ESRI. Esri shapefile technical description, 6 1988.

[7] D.D. Hearn, P. Baker, and W. Carithers. *Computer Graphics with OpenGL*. Prentice Hall, 2010.

[8] Danny Holten and Jarke J Van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983990, 2009.

[9] Danny Holten and Jarke J. van Wijk. A user study on visualizing directed edges in graphs. In *27th SIGCHI Conference on Human Factors in Computing Systems*, pages 2299–2308, 2009.

[10] Han Nicolaas en Bas Hamers Jan Latten. De prijs van migratie: Selectieve verhuisstromen van de vier grote steden. *Bevolkingstrends, 1e kwartaal 2006*, 2006.

[11] Marjolijn Das Jan Latten and Katja Chkalova. De stad groningen als roltrap van noord-nederland. *Bevolkingstrends, 2e kwartaal 2008*, 2008.

[12] Niels Kooiman Jan Latten. Aantrekkingskracht van regio's en demografische gevolgen. *Bevolkingstrends, 2e kwartaal 2011*, 2011.

[13] CodePlex open source community. Quickgraph, graph datastructures and algorithms for .net.

[14] Ben Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*, 0(UMCP-CSD CS-TR-3665):336–343, 1996.

[15] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[16] Alexandru Telea. *Data visualization principles and practice*. A K Peters, Ltd., Wellesley, Massachusetts, 2008.

[17] Jarke J. van Wijk and Edward R. van Selow. Cluster and calendar based visualization of time series data. In *INFOVIS*, pages 4–9, 1999.

[18] Hans Verbraeken. Bevolking trekt steeds meer weg uit grensgebieden en zoekt stad op. *Het Financieele Dagblad*, December 2011.

[19] Jo Wood, Jason Dykes, and Aidan Slingsby. Visualisation of origins, destinations and flows with od maps. *Cartographic Journal, The*, 47(2):117–129, 2010-05-01T00:00:00.