Handling, Measuring, Estimating and Visualizing Migration Data in R

Guy J. Abel, James Raymer, Ellen Kraly

2023-05-19

Contents

Chapter 1

Introduction

This manual covers a range of methods for handling, measuring, estimating, and visualizing migration data in R. These methods are based on several authoritative sources, including the UN DESA manuals on *Methods of measuring internal migration* and *Preparing migration data for subnational population projections*, as well as the migration chapters of the IUSSP *Tools for Demographic Estimation*. Additionally, we will cover many more recent developments in the field. By the end of this manual, you will have a comprehensive understanding of the various methods available for working with migration data in R, and how to apply them to your own research.

To make the most of this manual, we assume that you have some knowledge of using R, especially the *tidyverse* set of packages. If you're not familiar with R or need a refresher, we recommend working your way through an online course before diving into this manual. Some good resources for learning R and the tidyverse include:

- *R for Data Science*, a comprehensive guide to data science in R, covering data import and cleaning, data visualization, and statistical modeling.
- *DataCamp*, an online learning platform that offers interactive courses on R programming and data science topics.
- *R Bootcamp*, a free online course that covers the basics of R programming and the tidyverse.
- *Tidyverse.org*, a website dedicated to the tidyverse packages with tutorials, articles, and other resources for learning and using them.
- *Swirl*, an interactive learning platform within R that teaches you how to use R in a hands-on manner.

These resources provide a comprehensive introduction to R programming and the tidyverse, which will be useful throughout this manual and in your future data analysis work.

The manual is organized into nine chapters, each covering a different aspect of handling, measuring, estimating, and visualizing migration data in R. Chapter 1 provides an introduction to migration data and concepts, while Chapter 2 focuses on handling migration data in R. Chapter 3 covers summary migration indices, while Chapter 4 is dedicated to estimating net migration. Chapter 5 focuses on describing and estimating migration age structure, and Chapter 6 covers describing bilateral migration data. Chapter 7 is dedicated to estimating bilateral migration, while Chapters 8 and 9 cover different methods for visualizing bilateral migration, including chord diagrams and Sankey plots. In each chapter, we provide code and data that will allow you to replicate the outputs we present, as well as exercises that allow you to practice the concepts covered in that chapter on your own. We also provide solutions to these exercises, so you can check your work and ensure that you have a firm grasp of the material before moving on to the next chapter. By following the examples and completing the exercises in this manual, you will gain a deep understanding of how to handle, measure, estimate, and visualize migration data in R.

Chapter 2

Migration Concepts

Migration is a complex phenomenon that involves a change in place of abode, or place of "usual" residence. As defined by the United Nations, migration refers to "the movement of a person or a group of persons, either across an international border or within a state" (**?**). It can refer to various demographic units, such as a person, a family, or a household. However, the definition of migration typically excludes certain groups, such as nomads who do not have a fixed place of residence, or seasonal migrants who live in two or more places during the course of a year.

Both a spatial (place) and temporal (change) dimension are required in the definition of migration. Spatially, migration involves moving from one place to another, whether it be across a border or within a state. Temporally, migration involves a change over time, such as taking up life in a new or different place. Unlike other demographic processes, migration is not a one-time event, but rather a process that can involve multiple moves over the course of a lifetime

2.1 Spatial

2.1.1 Usual Residence

Central to the spatial dimension of defining a migration event is the concept of place of residence, used to determine the migrants origin and destination locations. The *Principles and Recommendations for Population and Housing Censuses* (UN Statistics Division 2008: 102, para. 1.463) defines usual residence as follows:

"It is recommended that countries apply a threshold of 12 months when considering place of usual residence according to one of the following two criteria:

1. The place at which the person has lived continuously for most of the last

12 months (that is, for at least six months and one day), not including temporary absences for holidays or work assignments, or intends to live for at least six months

2. The place at which the person has lived continuously for at least the last 12 months, not including temporary absences for holidays or work assignments, or intends to live for at least 12 months."

The use of two alternative criteria leaves some area of ambiguity, where the subtle differences in each could have significant implications for the measurement of migration. Further, within either of these criteria they might be additional ambiguity. For example, persons on long work assignments might have intentions to stay for only a few months that might turn into many months.

When defining migration, typically no restrictions are placed on the distance involved in a relocation **?**, it could involve a move from one apartment to another in the same building or a move to another country. In the past, some researchers have distinguished between moves between local communities, such as cities or labor markets, and moves within local communities, often labeled as "migration" and "local mobility." However, many have argued that this distinction is problematic and that no spatial constraints on the definition of migration should be used. This is because such a distinction could be arbitrary and context-dependent.

When analyzing migration patterns, it is useful to have information on the distance involved in a relocation. If address information on points of origin and destination is available, it is possible to tabulate moves by the distance covered. However, in many countries without population registers, this is not possible, and it may be time-consuming and of little policy relevance.

Census or survey results are usually tabulated for the administrative or political units into which the country is divided. Therefore, migration is operationally defined as a change of residence from one civil division to another, and the volume of migration is then function of the size of areas chosen for compilation, where typically larger administrative units, such as a state or country, have a higher volume of migration compared to smaller units like cities or counties. Most countries typically have hierarchies in their administrative units and might provide migration data to reflect one or more of these geographies. Using larger geographic units can result in a loss of detail and accuracy in migration data. For instance, if a state is used as the unit of compilation, it may not capture migration patterns between different cities within the state, which could be important for local policy-making. On the other hand, using smaller geographic units such as census tracts may result in too much detail, making it difficult to draw broader conclusions about migration trends at the regional or national level.

2.1.2 Origin

When analyzing migration patterns, it is important to consider the different types of data that can be used. Most migration data can be categorized into different types based on the origin of migrants is defined. The two most commonly used types are migration event data, which is based on the previous place of residence, and migrant transition data, which is based on the place of residence a fixed number of years ago. Other types of migration data are occasionally collected that do not involve the definition of a migrants origin, including the duration at residence, number of moves over a given interval, and country of citizenship.

Lifetime migration data is another common type of migration data. It can be considered as a form of transition data, where the number of years changes based on the age of each individual. Migrant stock data is an aggregation over all persons' lifetime migration flow and is given at a specific point in time without an interval. The migration data literature often distinguishes between stock and flow data.

2.1.3 Migrant Transition Data

Migrant transition data are typically collected in national censuses, which identify migrants by comparing their place of usual residence at the time of enumeration (t) with that at a specified earlier date ($t-n$). This type of data provides information on the movements of migrants over a given time period, which is usually either 1 year (e.g. UK) or 5 years (e.g. USA). Some countries have time periods that correspond to the interval between the current and last census or significant time points in the country's history. Transition data have some limitation that become more prominent for migration measured using longer time periods. For example, migration transition data fail to identify multiple and return moves, which can lead to an underestimation of the true level of migration. Additionally, migrant transition data do not capture migrants who are born or who die during the measurement period, which can also impact the accuracy of the data.

Migrant transition data provide counts of migrants, where a migrant is defined as a person who has experienced one or more migrations during the specified prior period. It is important to note that persons who moved during the measurement interval and subsequently died before its end should technically be counted as migrants and their moves as migrations. However, in practice, such cases are usually excluded as information on migrants is usually obtained after the end of the interval and with reference to persons still living at that time. This exclusion can potentially lead to an underestimation of the true number of migrant transitions.

2.1.4 Migation Event Data

Event data records every move made by an individual, including multiple and return migrations, newborn moves, and moves immediately before death. Population registers typically collect these data and provide a more complete record of migration over time. However, the geographical units for which data are available are generally coarser, and registers often fail to capture information on within-region moves. Additionally, less information about the characteristics of migrants is usually available, and some groups may be omitted altogether, such as prisoners and military personnel.

There are important distinctions between the play (migration) and the actor (migrant). For a given migration interval, the number of migrants is rarely, if ever, as large as the number of migrations. Unless the interval is very short, such as a day or perhaps a week, some persons are certain to move more than once. The longer the migration interval, the more the count of migrants will understate the amount of migration. Conversely, the shorter the migration interval, the count of migrants will approach the number of migrations **?**.

2.2 Temporal

2.2.1 Migration Interval

Migration is a continuous process that occurs over time, and to study its incidence, data must be compiled with reference to specific periods of time. These time periods can be either definite or indefinite. Definite interval data is typically collected over fixed-term periods such as one year, five years, ten years, or the intercensal period. Indefinate interval data such as lifetime migration measures or data based on place of last residence lack a definite time reference as age or time at the current residence varies by each individual migrant.

The comparability of migration data with different definite time intervals can be prohibitively complicated. Commonly described as the one-year five-year problem, observed migration data consistently shows the number of migrants recorded over a five-year interval is far less than five times the number recorded over a one-year interval. In addition, the ratio of migrants between a five-year period and a one-year period is not constant, where variations occur depending on multiple factors such as the intensity and type of migration both over time and in each origin and destination. Consequently, there is no straightforward algebraic solution to comparing one-year and five-year migration probabilities **?**.

2.3 Migration Measures

Migration measures are used to quantify the magnitude and direction of population movements between places or regions. These measures can provide important insights into the demographic and social dynamics of populations. There are several different types of migration measures that are commonly used in research and policy analysis, each with its own strengths and limitations.

One of the most common migration measures is the migration rate, which is defined as the number of migrants divided by the total population at risk. The migration rate can be calculated for different migration types discussed above, such as one year or five years, and can be used to compare migration across different places or regions.

Other migration measures include the count of the number of migrants, the migration intensity, which is the number of migrants per unit of population, and the migration propensity, which is the proportion of the population that have migrated. These measures can be useful for identifying patterns in migration behavior, such as the prevalence of long-distance migration or the likelihood of migration among certain demographic groups.

It is important to note that migration measures can be affected by data quality issues, such as underreporting of migrants or errors in place of residence information. Additionally, different measures may be more appropriate for different research questions or policy applications. For example, the migration rate may be more useful for understanding the overall magnitude of migration in the population, while the count of migrants can provide a basic understanding of the scale of migration patterns over time and between different spatial units.

Migration measures can be defined at different levels of detail, ranging from region-to-region measures, to region totals, to system totals or index measures. Region-to-region measures capture the flow of migrants between two specific regions, while region totals capture the total number of migrants coming in or going out of a specific region. System totals, or index measures, provide an overall picture of migration within a given system or country, which we will discuss in more detail in the next chapter.

2.3.1 Region-to-region

Region-to-region migration measures also known as *bilateral* migration, migration *streams* or *origin-destination* migration, refer to a migration measure that cross-classified by region of origin and region of destination, forming a matrix of $n \times (n-1)$ streams along each origin-destination combination, where n represents the number of regions. The set of region-to-region migration measures can be represented by m_{ij}, where the sub-scripts i and j represent the same set of regions for each origin-destination combination. The set of bilateral migration flows provide a basis to asses the compartive volumnes and directions of migration between a set of regions.

The *gross interchange* represents the total number of migrants moving between a particular pair of regions, i.e. $m_{ij} + m_{ji}$. The *net migration steam* or *bilateral net migration* represents the difference in migration between a pair of region i.e. $m_{ij} - m_{ji}$. For a pair of streams that are of unequal size, where the net migration stream is not close to zero, there exists a *dominant stream* which is far large than the *reverse* or *counter* stream in the opposite direction.

2.3.2 Region Totals

Every migration event can be considered an out-migration in relation to the region of origin and an in-migration in relation to the region of destination. When migration events involve changes of countries, migration events are typically described as emigration and immigration, rather than out-migration and in-migration. Totals on in- or out-migration for each region are typically used to evaluate the volume of migration to or from a particular set of regions. In some countries, data is collected or aggregated without reference to the place of origin for in-migration totals or destination for out-migration totals. Consequently the migration totals provide the most detailed measure of regional migration but with little information on the direction of the migration flows between each region. A summary of the common terms for migration totals are shown in Table 1. The in-migration (or immigration) totals can be represented by replacing the origin i index with a $+$; m_{+j}. Similarly, the out-migration (or emigration) totals can be represented by replacing the destination j index with $+$; m_{+j}.

Scale	Area	Event Term	Migrant Term
Internal	Origin	out-migration	out-migrant
	Destination	in-migration	in-migrant
International	Origin	emigration	emigrant
	Destination	immigration	immigrant

The sum of the in-migration and out-migration totals ($m_{i+} + m_{+j}$) provides the *turnover* of each region. Net migration totals provides a balance of movements in opposing directions from the difference between in-migration and out-migration ($m_{+j} - m_{i+}$). Net migration measures are more typically obtained via demographic accounting, as a residual from the differences in population change, births and deaths over a period in each region. As this calculation does not require expensive migration data collection systems, net migration measures are one of the most common forms of migration measures. However, net migration measures have a number of notable drawbacks, as highlighted by **?**. In particular, net migration does not enumerate migrants themselves, but instead follows a residual of in-migrants and out-migrants. Consequently, the dynamics related to the observed migration patterns can be missed. For example, an net migration of -100 might involve a region receiving no in-migrants and sending 100 out-migrants or receiving 1,000,000 migrants and sending 1,000,100 out-migrants. Further migration dynamics are also missed when looking at net migration rates (discussed in the next section) and regularities in age profiles of migration (discussed in Chapter X) are often preculded when using age-specicifc net migration measures.

2.3.3 Rate measures

Migration rates are important indicators for understanding the dynamics of population movement. Out-migration or emigration rates are calculated by dividing the number of out-migrants or emigrants during a specific period by the population exposed to the likelihood of migration. This is represented by the formula:

$$e^{[t,t+1]} = \frac{E^{[t,t+1]}}{P} k$$

Here, $e^{[t,t+1]}$ represents the out or emigration rate, E is the number of out-migrants or emigrants during the period, P is the population exposed to the likelihood of migration, and k is a constant, often set as 1000. The exposure population can be the population at the mid-interval, assuming migration is evenly distributed, or the population at the start or end of the interval if migration has a negligible effect on population change. Additionally, out-migration rates can be further decomposed by subsets of the population, such as age or sex:

$$e_i^{[t,t+1]} = \frac{E_i^{[t,t+1]}}{P_i} k$$

On the other hand, in-migration or immigration rates are calculated by dividing the number of in-migrants or immigrants by the population not exposed to the risk of migrating into the region. The formula for in-migration rate is:

$$i^{[t,t+1]} = \frac{I^{[t,t+1]}}{P} k$$

Similarly, net migration rates are calculated by dividing the net migration (difference between in-migration and out-migration) by the population not exposed to migration risk:

$$m^{[t,t+1]} = \frac{M^{[t,t+1]}}{P} k$$

It is worth noting that in-migration and net migration rates are different from other demographic rates because they use the resident population (population not exposed to risk) in the denominator. This approach satisfies the needs of the demographic balancing equation, as rates of gain and loss are measured relative to the same population. The demographic balancing equation is expressed as:

$$P^{t+1} = P^t \left(1 + b^{[t,t+1]} - d^{[t,t+1]} + i^{[t,t+1]} - e^{[t,t+1]}\right)$$

where P^{t+1} is the population at the next time point, $b^{[t,t+1]}$ and $d^{[t,t+1]}$ represent births and deaths during the period, and $i^{[t,t+1]}$ and $e^{[t,t+1]}$ denote in-migration and out-migration rates. Net migration $(M^{[t,t+1]})$ can be substituted with the difference between in-migration and out-migration $(I^{[t,t+1]} - O^{[t,t+1]})$. The equation can be simplified as:

$$P^{t+1} = P^t \left(1 + b^{[t,t+1]} - d^{[t,t+1]} + i^{[t,t+1]} - o^{[t,t+1]}\right)$$

This formulation allows for the analysis of population change considering the effects of births, deaths, in-migration, and out-migration.

2.4 References

Chapter 3

Handling Migration Data in R

The R statistical language provides many powerful tools to carry out data analysis. In this chapter we highlight some useful R functions to manipulate migration data into specific formats that might be required for more advanced functions to analysis or visualization.

3.1 Contingency Table

Bilateral migration data is often organized and represented in square tables, commonly referred to as migration matrices. These tables or matrices are form of contingency tables, otherwise know as cross-tabulations or frequency tables, often used in data analysis and statistics. Migration tables provide a structured way to organize and summarize origin-destination migration data, often with table rows containing the region of origin and the table columns based on the regions of destinations. The cells in the table typically capture the counts of migration from one region of origin to another region of destination, where each non-diagonal cell in the table represents the number of migrants moving between a specific pair of regions. The inspection of migration tables themselves often provides valuable first insights into the magnitude and direction of migration flows between different areas.

Origin

Destination

A

B

C

D

Sum

A

100

30

70

200

B

50

45

5

100

C

60

35

40

135

D

20

25

20

65

Sum

130

160

95

115

500

The diagonal cells in migration tables, when provided, typically represent populations that either do not migrate or move within the same region. These values are often not presented or are assigned a specific value to indicate a non-moving or within-region population for allow for a clearer comparisons of the migration patterns in the non-diagonal cells.

3.2 Data Creation

In R, we may create migration tables directly using the `matrix()` or `array()` functions. Both functions output create `array` type objects, are sometimes a pre-requisite for more complicated functions used for describing, estimating or visualising bilateral migration data.

The `matrix()` function allows users to specify the dimensions of the matrix and populate it with desired values. It can be used to create matrices of any size, and supports various options for filling the matrix, such as using a sequence of numbers, replicating values, or using external data sources. Data is provided via a vector passed to the `data` argument. By default the data populates the matrix from the first column on, which can be altered by setting `byrow = FALSE`.

```
m0 <- matrix(data = c(0, 100, 30, 70, 50, 0, 45, 5, 60, 35, 0, 40, 20, 25, 20, 0),
             nrow = 4, ncol = 4, byrow = TRUE)
m0
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0  100   30   70
## [2,]   50    0   45    5
## [3,]   60   35    0   40
## [4,]   20   25   20    0
```

It is often valuable to supply a vector of character strings for the origin and destination names to the `matrix` data object. These can be provided to an existing `matrix` object using the `dimnames()` via the `dimnames` argument or can be set via the `dimnames` argument within the `matrix()`. The corresponding `rownames()` and `colnames()` functions can be used to assign or display individual dimension names.

```
# create region labels
r <- LETTERS[1:4]
r
```

```
## [1] "A" "B" "C" "D"
```

```
# check dimension names
dimnames(m0)
```

```
## NULL
```

```
# add dimension names
dimnames(m0) <- list(orig = r, dest = r)
m0
```

```
##     dest
```

```
## orig  A   B  C  D
##    A  0 100 30 70
##    B 50   0 45  5
##    C 60  35  0 40
##    D 20  25 20  0
```

```
# create matrix with dimension names directly
# m0 <- matrix(data = c(0, 100, 30, 70, 50, 0, 45, 5, 60, 35, 0, 40, 20, 25, 20, 0),
#               nrow = 4, ncol = 4, byrow = TRUE,
#               dimnames = list(orig = r, dest = r))
```

In R, the `array()` function is used to create multidimensional arrays, which can have more than two dimensions. While the `matrix()` function creates two-dimensional structures, the `array()` function extends this capability to higher dimensions.

Similar to the `matrix()` function, the `array()` function allows users to define the dimensions of the array and populate it with desired values. However, in the `array()` function, the dimensions are specified as a vector to the `dim` argument, indicating the size of each dimension. The `array()` function can be seen as a generalization of the `matrix()` function, as matrices are a specific type of two-dimensional arrays. By using the `array()` function, users can work with data that requires more complex organization and analysis, such as migration data cross classified by origin, destination and additional variables such as sex, age or education.

```
m1 <- array(data = sample(x = 1:100, size = 32),
            dim = c(4, 4, 2),
            dimnames = list(orig = r, dest = r, sex = c("female", "male")))
m1
```

```
## , , sex = female
##
##     dest
## orig  A   B  C  D
##    A 68 100  5 89
##    B  9  78 96 84
##    C 25  74 47 98
##    D 37  40 55 85
##
## , , sex = male
##
##     dest
## orig  A  B  C  D
##    A 48 58 90 77
##    B 28 59 44 63
##    C 88  8 91 18
##    D 20  6 51 95
```

3.3 Data Manipulation

Statistical offices, government agencies, and international organizations collect and disseminate migration data in different formats to accommodate the needs of users and researchers. The format of the data may not necessarily be in square matrices that can be read directly into R and converted into a `matrix` object. However, there are useful functions in R that can be employed to convert data into appropriate formats for migration analysis.

The `xtabs()` function is particularly helpful as it enables the conversion of data frames in a tidy format **?** into matrices or arrays. It requires a `formula` argument that specifies the column names in the data frame that will be used to construct and populate the `matrix` or `array`. The formula consists of the left-hand side representing the column name with the data to fill, the `~` symbol to separate the left and right-hand sides, and the right-hand side representing the columns used for cross-classifying the left-hand variable (separated by `+`). The `data` argument specifies the data object where the data in a tidy format with the variables to be used in the formula.

```
## # A tibble: 16 x 3
##    orig  dest   flow
##    <chr> <chr> <int>
##  1 A     A         1
##  2 A     B         2
##  3 A     C         3
##  4 A     D         4
##  5 B     A         5
##  6 B     B         6
##  7 B     C         7
##  8 B     D         8
##  9 C     A         9
## 10 C     B        10
## 11 C     C        11
## 12 C     D        12
## 13 D     A        13
## 14 D     B        14
## 15 D     C        15
## 16 D     D        16
```

```
##     dest
## orig  A  B  C  D
##    A  1  2  3  4
##    B  5  6  7  8
##    C  9 10 11 12
##    D 13 14 15 16
```

The `as.data.frame.table()` function provides an inverse of the data manipulation of the `xtabs()` function, whereby it takes a `matrix` or `array` and converts it into a data frame based on the array dimension names. The `responseName` argument can be used to set the column name of the values in the cells of the `matrix` or `array`.

```
# convert previous matrix back to tibble
m2 %>%
  as.data.frame.table(responseName = "migration") %>%
  as_tibble()
```

```
## # A tibble: 16 x 3
##    orig  dest  migration
##    <fct> <fct>     <int>
##  1 A     A             1
##  2 B     A             5
##  3 C     A             9
##  4 D     A            13
##  5 A     B             2
##  6 B     B             6
##  7 C     B            10
##  8 D     B            14
##  9 A     C             3
## 10 B     C             7
## 11 C     C            11
## 12 D     C            15
## 13 A     D             4
## 14 B     D             8
## 15 C     D            12
## 16 D     D            16
```

Note, above we use `as_tibble()` to convert the `data.frame` object returned from `as.data.frame.table()` to the more user friendly `tibble` object type (insert citation) and use the pipe line function `%>%` to combine together a sequence of R functions.

```
# convert array to tibble
d1 <- m1 %>%
  as.data.frame.table(responseName = "flow") %>%
  as_tibble()
d1
```

```
## # A tibble: 32 x 4
##    orig  dest  sex     flow
##    <fct> <fct> <fct>  <int>
##  1 A     A     female    68
##  2 B     A     female     9
##  3 C     A     female    25
##  4 D     A     female    37
##  5 A     B     female   100
```

```
##   6 B     B     female    78
##   7 C     B     female    74
##   8 D     B     female    40
##   9 A     C     female     5
##  10 B     C     female    96
## # ... with 22 more rows
```

3.4 Matrix Operations

When working with `matrix` objects based on migration data in R there are additional functions that are useful for further formatting and data exploration. The `addmargins()` function is a useful tool for adding row and column margin totals to a `matrix` or `array` object.

```
addmargins(A = m0)
```

```
##      dest
## orig    A   B  C   D Sum
##   A     0 100 30  70 200
##   B    50   0 45   5 100
##   C    60  35  0  40 135
##   D    20  25 20   0  65
##   Sum 130 160 95 115 500
```

When working with migration matrices in R, it can sometimes be challenging to effectively view and analyze the data due to various factors such as lengthy dimension names and large unit sizes. Additionally, the inclusion of diagonal terms, which are often not of interest in migration analysis, can further complicate the interpretation of the matrix. However, R provides several helpful functions that can assist in adapting migration matrix objects for easier viewing and analysis. To illustrate their application, we will use the `uar_1960` object from the **migest** package, which represents a lifetime migration matrix for the Governorates of the United Arab Republic in 1960 as documented in the United Nations manual by **?**. Notice how by default the object is difficult to completely view due to the forementioned issues:

```
library(migest)
uar_1960
```

```
##             dest
## orig           Cairo Alexandria Port-Said Ismailia Kalyubia Gharbia Menoufia
##   Cairo      2079434      31049      5293     9813    23837   10034     7038
##   Alexandria   47220    1085602      2641     2625     2135    4921     1505
##   Port-Said     9464       2562    168046     6461      496     817      323
##   Ismailia      9518       1395      3490   171297      718     910      306
##   Kalyubia     90668       4730       758     3182   886464    3727     3523
##   Gharbia      99179      39953      1742     3347     7870 1604851     6313
```

```
##   Menoufia    216764      46781      1640      3338      2918    29580   1308283
##   Giza         64584       4899       513      2013      2887     1503      2161
##   Assyiut     100305      25497      1738      2522       122     2245       636
##   Souhag      100100      63712     12087      9436       295     2791      1095
##   All others  456464     177476     43898     66973     49816    47315     12179
##             dest
## orig            Giza Assyiut  Souhag All others
##   Cairo        88543    4951    2569      58476
##   Alexandria    6910    1355    1467      29534
##   Port-Said     1505     326     454      11184
##   Ismailia      1593     319     263      10269
##   Kalyubia     10279     340     128      18076
##   Gharbia      14529     848     491      64140
##   Menoufia     30915     567     401      47843
##   Giza       1040179     540     433      13518
##   Assyiut      13153 1290255    5955      35157
##   Souhag       17958   11608 1540020      53224
##   All others   94577   14690   22375   11900302
```

When working with names or labels that are lengthy or contain unnecessary details, the `abbreviate()` function can be helpful. The function applies an algorithm to shorten the names while still retaining their essential information.

```
dimnames(uar_1960)
```

```
## $orig
##  [1] "Cairo"      "Alexandria" "Port-Said"  "Ismailia"   "Kalyubia"
##  [6] "Gharbia"    "Menoufia"   "Giza"       "Assyiut"    "Souhag"
## [11] "All others"
##
## $dest
##  [1] "Cairo"      "Alexandria" "Port-Said"  "Ismailia"   "Kalyubia"
##  [6] "Gharbia"    "Menoufia"   "Giza"       "Assyiut"    "Souhag"
## [11] "All others"
```

```
# make a copy
u0 <- uar_1960
# new abbreviated region names
r <- list(orig = uar_1960 %>%
            rownames() %>%
            abbreviate(),
          dest = uar_1960 %>%
            colnames() %>%
            abbreviate())
r
```

```
## $orig
##      Cairo Alexandria  Port-Said   Ismailia   Kalyubia    Gharbia   Menoufia
```

```
##     "Cair"      "Alxn"      "Pr-S"      "Isml"      "Klyb"      "Ghrb"      "Menf"
##       Giza     Assyiut      Souhag All others
##     "Giza"      "Assy"      "Sohg"      "Allo"
##
## $dest
##      Cairo Alexandria  Port-Said   Ismailia   Kalyubia    Gharbia   Menoufia
##     "Cair"      "Alxn"      "Pr-S"      "Isml"      "Klyb"      "Ghrb"      "Menf"
##       Giza     Assyiut      Souhag All others
##     "Giza"      "Assy"      "Sohg"      "Allo"
```

```
# apply the abbreviated region names
dimnames(u0) <- r

u0
```

```
##       dest
## orig      Cair    Alxn    Pr-S    Isml    Klyb    Ghrb    Menf    Giza    Assy
##   Cair 2079434   31049    5293    9813   23837   10034    7038   88543    4951
##   Alxn   47220 1085602    2641    2625    2135    4921    1505    6910    1355
##   Pr-S    9464    2562  168046    6461     496     817     323    1505     326
##   Isml    9518    1395    3490  171297     718     910     306    1593     319
##   Klyb   90668    4730     758    3182  886464    3727    3523   10279     340
##   Ghrb   99179   39953    1742    3347    7870 1604851    6313   14529     848
##   Menf  216764   46781    1640    3338    2918   29580 1308283   30915     567
##   Giza   64584    4899     513    2013    2887    1503    2161 1040179     540
##   Assy  100305   25497    1738    2522     122    2245     636   13153 1290255
##   Sohg  100100   63712   12087    9436     295    2791    1095   17958   11608
##   Allo  456464  177476   43898   66973   49816   47315   12179   94577   14690
##       dest
## orig      Sohg     Allo
##   Cair    2569    58476
##   Alxn    1467    29534
##   Pr-S     454    11184
##   Isml     263    10269
##   Klyb     128    18076
##   Ghrb     491    64140
##   Menf     401    47843
##   Giza     433    13518
##   Assy    5955    35157
##   Sohg 1540020    53224
##   Allo   22375 11900302
```

Basic arithmetic operators can be employed to scale the data to an appropriate level, such as dividing the values by a common factor or multiplying them to achieve a desired magnitude. This can be useful when working with migration matrices to adjust the values and make them more interpretable or comparable. The `round()` function which allows users to specify the precision of numbers in your migration data, which can be handy when working with migration rates or other quantitative measures.

```
u1 <- round(x = u0/1000, digits = 1)
u1
```

```
##       dest
## orig     Cair   Alxn   Pr-S   Isml   Klyb   Ghrb   Menf   Giza   Assy   Sohg
##   Cair 2079.4   31.0    5.3    9.8   23.8   10.0    7.0   88.5    5.0    2.6
##   Alxn   47.2 1085.6    2.6    2.6    2.1    4.9    1.5    6.9    1.4    1.5
##   Pr-S    9.5    2.6  168.0    6.5    0.5    0.8    0.3    1.5    0.3    0.5
##   Isml    9.5    1.4    3.5  171.3    0.7    0.9    0.3    1.6    0.3    0.3
##   Klyb   90.7    4.7    0.8    3.2  886.5    3.7    3.5   10.3    0.3    0.1
##   Ghrb   99.2   40.0    1.7    3.3    7.9 1604.9    6.3   14.5    0.8    0.5
##   Menf  216.8   46.8    1.6    3.3    2.9   29.6 1308.3   30.9    0.6    0.4
##   Giza   64.6    4.9    0.5    2.0    2.9    1.5    2.2 1040.2    0.5    0.4
##   Assy  100.3   25.5    1.7    2.5    0.1    2.2    0.6   13.2 1290.3    6.0
##   Sohg  100.1   63.7   12.1    9.4    0.3    2.8    1.1   18.0   11.6 1540.0
##   Allo  456.5  177.5   43.9   67.0   49.8   47.3   12.2   94.6   14.7   22.4
##       dest
## orig      Allo
##   Cair    58.5
##   Alxn    29.5
##   Pr-S    11.2
##   Isml    10.3
##   Klyb    18.1
##   Ghrb    64.1
##   Menf    47.8
##   Giza    13.5
##   Assy    35.2
##   Sohg    53.2
##   Allo 11900.3
```

The `diag()` function allows users to manipulate the diagonal terms of a matrix, which often represent non-moving individuals or populations within a region which can be many orders of magnatude larger than the counts of persons migrating in the non-diagonal cells. The `diag()` function takes a matrix as input and returns a new matrix with the same values, except that the diagonal elements are modified according to the specified rule. In the context of migration data, setting the diagonal terms to zero effectively removes the non-moving populations from the matrix, making it easier to analyze the migration flows between regions of interest.

```
u2 <- u0
diag(u2) <- 0
u2
```

```
##       dest
```

```
## orig     Cair   Alxn  Pr-S  Isml  Klyb  Ghrb  Menf  Giza  Assy  Sohg  Allo
##   Cair      0  31049  5293  9813 23837 10034  7038 88543  4951  2569 58476
##   Alxn  47220      0  2641  2625  2135  4921  1505  6910  1355  1467 29534
##   Pr-S   9464   2562     0  6461   496   817   323  1505   326   454 11184
##   Isml   9518   1395  3490     0   718   910   306  1593   319   263 10269
##   Klyb  90668   4730   758  3182     0  3727  3523 10279   340   128 18076
##   Ghrb  99179  39953  1742  3347  7870     0  6313 14529   848   491 64140
##   Menf 216764  46781  1640  3338  2918 29580     0 30915   567   401 47843
##   Giza  64584   4899   513  2013  2887  1503  2161     0   540   433 13518
##   Assy 100305  25497  1738  2522   122  2245   636 13153     0  5955 35157
##   Sohg 100100  63712 12087  9436   295  2791  1095 17958 11608     0 53224
##   Allo 456464 177476 43898 66973 49816 47315 12179 94577 14690 22375     0
```

3.5 Summaries

3.5.1 Bilateral measures

the *migest* package offers several useful functions for generating summaries of origin-destination migration data. One such function is *sum_bilat()*, which allows you to calculate the counter flow, net flow and interchange for all migration pairs. This function can accept either a `matrix`, `array` or a `data.frame` (or `tibble`) as input.

```
sum_bilat(m0)
```

```
## # A tibble: 12 x 8
##    orig  dest  corridor pair   flow counter_flow net_flow interchange
##    <chr> <chr> <chr>    <chr> <dbl>        <dbl>    <dbl>       <dbl>
##  1 B     A     B -> A   A - B    50          100      -50         150
##  2 C     A     C -> A   A - C    60           30       30          90
##  3 D     A     D -> A   A - D    20           70      -50          90
##  4 A     B     A -> B   A - B   100           50       50         150
##  5 C     B     C -> B   B - C    35           45      -10          80
##  6 D     B     D -> B   B - D    25            5       20          30
##  7 A     C     A -> C   A - C    30           60      -30          90
##  8 B     C     B -> C   B - C    45           35       10          80
##  9 D     C     D -> C   C - D    20           40      -20          60
## 10 A     D     A -> D   A - D    70           20       50          90
## 11 B     D     B -> D   B - D     5           25      -20          30
## 12 C     D     C -> D   C - D    40           20       20          60
```

3.5.2 Total Measures

Another useful function in the *migest* package is `sum_region()`, which allows you to generate comprehensive summaries of in-migration, out-migration, net migration, and turnover totals for each region in your migration data. Similar to the `sum_bilat()` function, `sum_region()` also accepts either a `matrix` or a `data.frame` (or `tibble`) as input, providing flexibility in working with different data formats.

By using the `sum_region()` function, you can obtain valuable information about migration flows at the regional level. It calculates the total number of migrants moving into each region (in-migration), the total number of migrants moving out of each region (out-migration), the net migration balance (in-migration minus out-migration), and the turnover (sum of in-migration and out-migration) for each region. These summaries offer a comprehensive picture of migration levels for each region, allowing for further analysis and interpretation.

```
sum_region(m0)
```

```
## # A tibble: 4 x 5
##   region out_mig in_mig  turn   net
##   <chr>    <dbl>  <dbl> <dbl> <dbl>
## 1 A          200    130   330   -70
## 2 B          100    160   260    60
## 3 C          135     95   230   -40
## 4 D           65    115   180    50
```

Note, when the data provided to the `sum_region()` is a data frame, the origin and destination regions names are assumed to be in variables named `orig` and `dest`. In addition, the migration data re assumed by default to be in variable named `flow`. If the corresponding column names differ, the user can supply these to the `orig_col`, `dest_col` and `flow_col` argumenets in the `sum_region()` function.

The `sum_country()` function provides the same calculations, but provides summary variables names corresponding to the equivalent terms used for international migration data. When the input data for either the `sum_region()` or `sum_country()` functions are over more than two dimensions, beyond the standard origin and destination dimensions, the `group_by` function from the *dplyr* package should be used to allow for specific calculations beyond the origin and destination dimensions. To demonstrate we will used the international flow estimates of **?** which can be downloaded and read directly into R from the online CSV file.

```
# read data from web depository
f <- read_csv("https://ndownloader.figshare.com/files/26239945")
f
```

```
## # A tibble: 235,236 x 9
##    year0 orig  dest  sd_drop_neg sd_rev_neg mig_rate da_min_open da_mi~1 da_pb~2
##    <dbl> <chr> <chr>       <dbl>      <dbl>    <dbl>       <dbl>   <dbl>   <dbl>
##  1  1990 BDI   BDI             0          0        0           0       0       0
##  2  1990 COM   BDI             0          0        0           0       0       0
##  3  1990 DJI   BDI             0          0        0           0       0       0
```

```
## 4  1990 ERI   BDI            0          0        0            0       0      90
## 5  1990 ETH   BDI            0          0        0            0       0       2
## 6  1990 KEN   BDI           30         30       69           45      29      87
## 7  1990 MDG   BDI            0          0        0            0       0       0
## 8  1990 MWI   BDI            0          0        0            0       0     125
## 9  1990 MUS   BDI            0          0        0            0       1       1
## 10 1990 MYT   BDI            0          0        0            0       0       0
## # ... with 235,226 more rows, and abbreviated variable names 1: da_min_closed,
## #   2: da_pb_closed
```

```r
# single period (1990-1995)
f %>%
  filter(year0 == 1990) %>%
  sum_country(flow_col = "da_pb_closed")
```

```
## # A tibble: 197 x 5
##    country    emi     imm    turn     net
##    <chr>    <dbl>   <dbl>   <dbl>   <dbl>
##  1 ABW       1662   15874   17536   14212
##  2 AFG     345255 3421712 3766967 3076457
##  3 AGO      82775  225637  308412  142862
##  4 ALB     464693   21479  486172 -443214
##  5 ARE     272648  640784  913432  368136
##  6 ARG     444239  339393  783632 -104846
##  7 ARM     648202  151937  800139 -496265
##  8 ATG       6153    8387   14540    2234
##  9 AUS     691618 1042781 1734399  351163
## 10 AUT     154853  382724  537577  227871
## # ... with 187 more rows
```

```r
# all periods using group_by
f %>%
  group_by(year0) %>%
  sum_country(flow_col = "da_pb_closed") %>%
  arrange(country)
```

```
## # A tibble: 1,188 x 6
## # Groups:   year0 [6]
##    year0 country     emi     imm    turn      net
##    <dbl> <chr>     <dbl>   <dbl>   <dbl>    <dbl>
##  1  1990 ABW        1662   15874   17536    14212
##  2  1995 ABW        4007   10945   14952     6938
##  3  2000 ABW        3814   10064   13878     6250
##  4  2005 ABW        7544    7124   14668     -420
##  5  2010 ABW        8654    9910   18564     1256
##  6  2015 ABW       16306   17316   33622     1010
##  7  1990 AFG      345255 3421712 3766967  3076457
```

```
## 8  1995 AFG     1286436  418906 1705342  -867530
## 9  2000 AFG      434706 1178865 1613571   744159
## 10  2005 AFG     1500149  457339 1957488 -1042810
## # ... with 1,178 more rows
```