Patias, Nikos

Article

Exploring long-term youth unemployment in Europe using sequence analysis: A reproducible notebook approach

REGION

Provided in Cooperation with:
European Regional Science Association (ERSA)

Suggested Citation: Patias, Nikos (2019) : Exploring long-term youth unemployment in Europe using sequence analysis: A reproducible notebook approach, REGION, ISSN 2409-5370, European Regional Science Association (ERSA), Louvain-la-Neuve, Vol. 6, Iss. 3, pp. 53-69, https://doi.org/10.18335/region.v6i3.277

This Version is available at:
https://hdl.handle.net/10419/235805

REGION
The Journal of ERSA
Powered by WU

Volume 6, Number 3, 2019, 53–69
DOI: 10.18335/region.v6i3.277

journal homepage: region.ersa.org

Exploring long-term youth unemployment in Europe using sequence analysis: A reproducible notebook approach*

Nikos Patias[1]

[1] University of Liverpool, Liverpool, UK

Received: 28 August 2019/Accepted: 17 January 2020

Abstract. Youth unemployment is an important factor influencing the lifetime earnings and future job prospects of individuals, often resulting in deterioration in their health and well-being. Youth unemployment in Europe has been affected by the financial crisis of 2008. However, the magnitude of these effects varied across European countries. The objective of this notebook is to identify representative trajectories of youth unemployment change in Europe from 2008 to 2018. This notebook provides a self-contained research workflow that is fully reproducible and transparent. My findings suggest that northern Europe has high concentration of regions with stable low youth unemployment while southern Europe has high concentration of regions with stable high youth unemployment. Identifying key patterns of youth unemployment change among European countries can provide useful insights that help to understand migration patterns originating from the more "disadvantaged" regions to more "advantaged" ones, or beyond. Finally, I hope that data and regional scientists can benefit by the functionalities offered in this notebook and use it as a complementary guide for analysing their own data.

Key words: sequence analysis, unemployment, Europe, regional inequalities, reproducible research

1 Introduction

Youth unemployment is an important factor influencing the lifetime earnings and future job prospects of individuals, often resulting in deterioration in their health and well-being (Bell, Blanchflower 2011, O'Reilly et al. 2015). The effects of financial crisis of 2008 on youth unemployment were prominent and varied across European countries and regions. The European average for youth unemployment culminated in 2012 to more than 20%, but there were countries that scored much higher (i.e. more than 50% in Greece and more that 30% in Bulgaria and Italy) (Dietrich 2012). However, regional variations can help to contextualise and analyse patterns of youth unemployment more effectively (Pop et al. 2019). Understanding and tracking the evolution of regions that faced high levels of unemployment can help in planning future policies, as today's youth are going to be in the workforce for the next 50 years. Finally, the trajectories of youth unemployment change across regions (i.e. whether they have successfully recovered or not) and can be

*This paper is available as computational notebook on the REGION webpage.

linked to patterns of regional resilience against economic crises. In this notebook, I use a sequence analysis approach to identify representative trajectories of youth unemployment change by NUTS 2 regions in Europe from 2008 to 2018.

The idea of using reproducible analyses in computational research has a growing number of advocates (Peng 2011, Sandve et al. 2013, Rule et al. 2019). The development of computational notebooks such as R and Jupyter notebooks, allow scientists to incorporate code, documentation, graphs and text in a single document. Consequently, more than ever before, computational research has become more open, transparent and fully replicable. Peng (2011) developed a reproducibility spectrum to highlight the importance of incorporating publication standards text with linked code and data to achieve the "gold standard of reproducibility". The spectrum begins from the traditional static publications, which are not reproducible. They become more reproducible when code and data are incorporated in the publication. Finally, the full replication is achieved when linked and executable code and data are included. As Peng (2011) highlights, data is an integral part of reproducible research and should be clearly documented within the workflow. However, researchers often neglect to provide adequate information on the datasets used. As a result, other researchers have difficulties on replicating this piece of research. Linked datasets through direct web-links or Application Programming Interfaces (APIs) when available, contribute to the transparency of the research, by explicitly pointing the end-user to the source of information described in a research project.

The objective of this notebook is to identify key representative trajectories of youth unemployment change in Europe from 2008 to 2018. In the present notebook I provide a self-contained research workflow that is fully reproducible and transparent. Moreover, I make use of the functionalities offered by computational notebooks written in R markdown such as direct access to online tabular/spatial datasets, manipulation and linkage between these datasets as well as interactive plots/maps. Finally, this notebook aims to provide the sufficient tools that a data or regional scientist needs to perform similar types of analysis.

2 Packages and Dependencies

This section is used to report all the packages and dependencies required to run this notebook which are vital components of reproducible research. By reporting the R version under which I created this notebook and the packages used will ensure its replicability.

Firstly, is important to report the R version used in this notebook by running the following line of code.

[1]:
```
# to get the version of R used in the notebook
paste("The R Version used in this notebook is", getRversion())
```

[1]:
```
## [1] "The R Version used in this notebook is 3.5.1"
```

I then specify the CRAN repository where the packages have been downloaded from.

[2]:
```
# Define the CRAN repository for this session
r_rep = getOption("repos")
r_rep["CRAN"] = "http://cran.us.r-project.org"
options(repos = r_rep)
```

And install/load the packages required to run this notebook. Please note that the installation stage is required only the first time you run this notebook.

[3]:
```
# These are the packages required to run this notebook
# First should be installed
# install.packages("eurostat")
# install.packages("rvest")
# install.packages("knitr")
# install.packages("rgdal")
# install.packages("countrycode")
# install.packages("dplyr")
# install.packages("reshape2")
# install.packages("ggplot2")
# install.packages("TraMineR")
# install.packages("cluster")
```

```
# install.packages("factoextra")
# install.packages("RColorBrewer")
# install.packages("leaflet")
# install.packages("plotly")
# And then should be loaded
library(eurostat)
library(rvest)
library(knitr)
library(rgdal)
library(countrycode)
library(dplyr)
library(reshape2)
library(ggplot2)
library(TraMineR)
library(cluster)
library(factoextra)
library(RColorBrewer)
library(leaflet)
library(plotly)
```

Finally, I create a list of the available packages in my R environment and report the version of each package used.

[4]:
```
# Create a list with all the available packages in my R environment
pkg_used <- available.packages()
```

[5]:
```
# For every package print the version of the package, the version of R that depends
# on and the packages that imports
paste("eurostat Version is:", pkg_used["eurostat", "Version"])
paste("rvest Version is:", pkg_used["rvest", "Version"])
paste("knitr Version is:", pkg_used["knitr", "Version"])
paste("rgdal Version is:", pkg_used["rgdal", "Version"])
paste("countrycode Version is:", pkg_used["countrycode", "Version"])
paste("dplyr Version is:", pkg_used["dplyr", "Version"])
paste("reshape2 Version is:", pkg_used["reshape2", "Version"])
paste("ggplot2 Version is:", pkg_used["ggplot2", "Version"])
paste("TraMineR Version is:", pkg_used["TraMineR", "Version"])
paste("cluster Version is:", pkg_used["cluster", "Version"])
paste("factoextra Version is:", pkg_used["factoextra", "Version"])
paste("RColorBrewer Version is:", pkg_used["RColorBrewer", "Version"])
paste("leaflet Version is:", pkg_used["leaflet", "Version"])
paste("plotly Version is:", pkg_used["plotly", "Version"])
```

[5]:
```
## [1] "eurostat Version is: 3.4.20002"
## [1] "rvest Version is: 0.3.5"
## [1] "knitr Version is: 1.27"
## [1] "rgdal Version is: 1.4-8"
## [1] "countrycode Version is: 1.1.0"
## [1] "dplyr Version is: 0.8.3"
## [1] "reshape2 Version is: 1.4.3"
## [1] "ggplot2 Version is: 3.2.1"
## [1] "TraMineR Version is: 2.0-14"
## [1] "cluster Version is: 2.1.0"
## [1] "factoextra Version is: 1.0.6"
## [1] "RColorBrewer Version is: 1.1-2"
## [1] "leaflet Version is: 2.0.3"
## [1] "plotly Version is: 4.9.1"
```

In this notebook, I have installed the latest versions of the packages used. I understand that the analysis can be run by using previous versions too. However, using the versions of the packages as reported here ensures the reader that this notebook will run without any errors.

Now that all the packages have been correctly installed it is useful to provide a brief overview of the main functionalities of each package.

`eurostat` This package allows access to Eurostat data through their API.

`rvest` This package is used to scrape data from web pages.

`knitr` This package provides better visualisation of the results within the notebook (i.e. table formatting).

`rgdal` This package is used to read, merge and manipulate geospatial datasets.

`countrycode` This package is used to convert country codes (i.e. ISO 3166) to country names.

`dplyr` This package is used for more effective dataframes' manipulation.

`reshape2` This package is used to reshape tables from wide to long format and vice versa.

`ggplot2` This package is used for creating plots.

`TraMineR` This is the package is used to perform sequence analysis.

`cluster` This package is used to perform cluster analysis.

`factoextra` This package provides the functionalities to assess the optimal number of clusters.

`RColorBrewer` This package provides colour palettes to be used in maps.

`leaflet` This package is used for interactive mapping.

`plotly` This package is used for interactive plotting in conjuction with `ggplot2`.

3 Data and Methods

3.1 Data

Eurostat (https://ec.europa.eu/eurostat/data/database) has a large database providing a wide range of available datasets in varying geographies and time frames. In this notebook, I analyse youth unemployment in Europe from 2008 to 2018. Eurostat captures youth unemployment by measuring the percentage of "Young people neither in employment nor in education and training (NEET rates)". This dataset is available from 2000 to 2018 at NUTS 2 regions (more information on NUTS classification can be found at https://ec.europa.eu/eurostat/web/nuts/background). Eurostat defines youth unemployment either people aged 15-24 or 18-24 and are unemployed. In this study, I consider the percentage of people aged between 18 and 24. The original dataset used in this notebook can be accessed following https://ec.europa.eu/eurostat/web/products-datasets/-/edat_lfse_22. Eurostat has created its own R package (https://cran.r-project.org/web/packages/eurostat/index.html) to allow access in the database through an API with a comprehensive documentation (https://ropengov.github.io/eurostat/index.html) and tutorial (http://ropengov.github.io/eurostat/articles/eurostat_tutorial.html). In order to make the most of the functionalities offered by a notebook as well as enable researchers to replicate this approach I make use of Eurostat's API to access the dataset analysed in this notebook.

I first search for the datasets referring to young unemployed people.

[6]:
```
# search information about the datasets that are related to young unemployed people
kable(head(search_eurostat("Young people neither in employment")))
```

[6]:
```
Output in Table 1
```

Of the available datasets, I am interested in the first on this list with code `edat_lfse_22`. I specified my request to 11 time periods. Starting from the most current year (i.e. 2018) and going back to 2008. I also specified that I want total percentages (i.e. both male and female - `sex = "T"`) and finally the age group that covers people aged from 18 to 24.

[7]:
```
# Specify the ID of the dataset required
id <- "edat_lfse_22"
# Request of the dataset
young_unempl <- get_eurostat(id, filters = list(lastTimePeriod=11, sex = "T",
                     age = "Y18-24"), time_format = "num")
```

Table 1: Available datasets from Eurostat related to youth unemployment

title	code	type	last update of data	last table structure change	data start	data end	values
Young people neither in employment nor in education and training by sex and NUTS 2 regions (NEET rates)	edat_lfse_22	dataset	01.07.2019	13.08.2019	2000	2018	NA
Young people neither in employment nor in education and training by sex, age and degree of urbanisation (NEET rates)	edat_lfse_29	dataset	01.07.2019	13.08.2019	2000	2018	NA
Young people neither in employment nor in education and training by type of disability, sex and age	hlth_de030	dataset	21.03.2019	21.03.2019	2011	2011	NA
Young people neither in employment nor in education and training by sex, age and labour status (NEET rates)	edat_lfse_20	dataset	01.07.2019	13.08.2019	2000	2018	NA
Young people neither in employment nor in education and training by sex, age and citizenship (NEET rates)	edat_lfse_23	dataset	25.04.2019	13.08.2019	2004	2018	NA
Young people neither in employment nor in education and training by sex, age and country of birth (NEET rates)	edat_lfse_28	dataset	25.04.2019	13.08.2019	2004	2018	NA

I also make use of a spatial dataset to enable use of an interactive map to facilitate better presentation of results. To achieve that, I have downloaded NUTS 2 regions shapefile from the second version of Eurostat's spatial database (https://ec.europa.eu/eurostat/cache/GISCO/distribution/v2/). Eurostat provides the spatial data as a bulk download, so I first unzip the file and then select the shapefile that matches the tabular data that I have already downloaded. The name of the dataset "NUTS_RG_60M_2016_4326_LEVL_2.shp.zip" is self-explanatory showing that the spatial data is projected in the World Geodetic System of 1984 (i.e. WGS84 or EPSG:4326) for NUTS 2 regions in 2016.

[8]:
```
# Specify the url that links to the zipped spatial datasets
url = "http://ec.europa.eu/eurostat/cache/GISCO/distribution/v2/nuts/download/
...ref-nuts-2016-60m.shp.zip"
# Download the file
download.file(url, basename(url))
# Unzip the bulk file
unzip(basename(url))
# Unzip the specific shapefile needed
unzip(paste0(getwd(), "/NUTS_RG_60M_2016_4326_LEVL_2.shp.zip"))
# Read in the shapefile
geodata <- readOGR(dsn = getwd(), layer = "NUTS_RG_60M_2016_4326_LEVL_2")
```

Eurostat provides only country codes, which is not always helpful when presenting results. For this reason, I used the R package `countrycode` (https://cran.r-project.org/web/packages/countrycode/index.html) to convert country codes to country names. While in general, the European commission uses ISO 3166-1 alpha-2 codes, there are two exceptions. Greece is reported as "EL" (rather than "GR") and United Kingdom as "UK" (rather than "GB"). Thus, I recoded these two countries manually.

[9]:
```
# Create a new column for country names
geodata@data$cntr_name <- countrycode(geodata@data$CNTR_CODE, "iso2c", "country.name")
# Because European commission uses EL for Greece (in ISO 3166-1 alpha-2 codes is GR)
# and UK for United Kingdom (in ISO 3166-1 alpha-2 codes is GB) I should replace these
# two countries manually
geodata@data$cntr_name <- ifelse(geodata@data$CNTR_CODE=="EL","Greece",
ifelse(geodata@data$CNTR_CODE=="UK", "United Kingdom", geodata@data$cntr_name))
```

3.2 Methods

This notebook aims to identify representative trajectories of youth unemployment change across NUTS 2 regions in Europe. The methodological workflow followed here is similar to Patias et al. (2020) that analyses trajectories of neighbourhood change in Great Britain. Sequence analysis is a method that analyses sequences of categorical variables, and extracts information on their structure and evolution. Sequence analysis has its origins in biology, where it is used to analyse DNA sequences (Sanger et al. 1977). It can also be applied to analyse longitudinal individual-level family, migration and career trajectories (Brzinsky-Fay 2007, Rowe et al. 2017a,b). This method is also used on neighbourhood trajectory mining in the United States to identify patterns of socioeconomic change over

a period of time (Delmelle 2016). The key component of sequence analysis method is the optimal matching analysis which is used to measure pairwise dissimilarities between sequences and identifies "types of sequence patterns" (Studer, Ritschard 2016). In this notebook sequence analysis is used in a spatio-temporal concept assessing how youth unemployment in European regions (i.e. spatial) has changed from 2008 to 2018 (i.e. temporal). Sequence analysis has been used as it is a method capable of capturing multiple dimensions of spatio-temporal processes namely incidence, duration, timing and sequencing. The youth unemployment data downloaded from Eurostat is expressed in percentages (by NUTS 2 regions). Sequence analysis can be applied to categorical data, hence I had to classify the regions' youth unemployment percentages into quintiles so that they can be treated as categories. In this way, the multiple dimensions of spatio-temporal processes of youth unemployment change can be systematically measured. Thus, it explicitly captures for each region:

- The number of times in a particular quintile (i.e. incidence);
- The time span in a particular quintile (i.e. duration);
- The year at occurrence of change from one quintile to another (i.e. timing); and
- The chronological order of transitions between quintiles (i.e. sequencing).

The key stages followed in this notebook are described below with links to the particular sub-sections which provide some further technical clarifications:

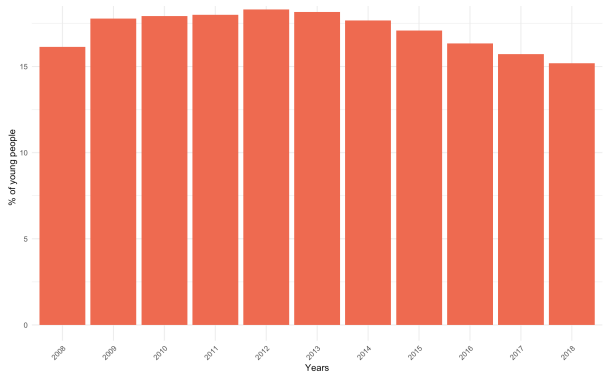
1. *Data pre-processing*, by classifying NUTS 2 regions into quintiles based on the percentage of youth unemployment in each year from 2008 to 2018 (regions with the lowest % youth unemployment belong to the 1st quintile and regions with the highest % youth unemployment belong to the 5th quintile).
2. Create a *sequence object* based on the quintile each region belongs to in every year (i.e. from 2008 to 2018).
3. *Measuring sequence dissimilarity* based on substitution costs which is the probability of transitioning from one quintile to another (i.e. higher transition rate from 1st quintile to 5th quintile rather than from 2nd quintile to 1st quintile). The substitution costs between quintiles i and j are calculated based on Equation (1).
4. Using the substitution costs calculated in the previous stage, I built a *dissimilarity matrix* including every pair of sequences. In this notebook I have used the Optimal Matching (OM) algorithm. The algorithm substitutes the elements of each sequence based on their substitution costs which in turn is the OM distance between each pair of sequences.
5. In the last stage I produce I typology of youth unemployment trajectories using the resulting dissimilarity matrix from stage 4. Partitioning Around Medoids (PAM) clustering algorithm is used for the *classification of sequences*

$$SubsCosts_{i,j} = 2 - p(i|j) - p(j|i) \tag{1}$$

where $p(i|j)$ is the transition rate between quintiles i and j. For the sequence analysis I have used R package `TraMineR` (https://cran.r-project.org/web/packages/TraMineR/index.html) which provides all the required functionalities.

4 Data Analysis

As shown in Figure 1, the average percentage of young people who are neither in employment nor in education and training in Europe increased from 16% in 2008 to 18.5% in 2012, followed by a decrease in 2018 (i.e. at around 15%). The results suggest that on average, regions show patterns of resilience against financial crises and that policies



Notes: Data from Eurostat, calculations by the author

Figure 1: European % average of young people neither in employment nor in education and training

targeting the decrease of youth unemployment have proven efficient. However, not all regions follow the same patterns.

This section of the notebook aims to provide an understanding on long-term youth unemployment patterns in NUTS 2 regions in Europe but also to guide the reader on the analytical process of sequence analysis and the functionalities offered by computational notebooks. Each of the following five sub-sections will present in detail each of the steps followed for the production of the results.

[10]:
```
# I change the years from numbers to characters so to be recongised as categorical
# rather than continuous variable
young_unempl$time <- as.character(young_unempl$time)
# Create a plot by showing the European % average of young people neither in employment
# nor in education and training
young_unempl %>%
  group_by(time) %>%
  summarise_all(mean, na.rm = TRUE) %>%
  ggplot()
  geom_bar(aes(x = time, y = values), stat = "identity", fill = "coral2")
  labs(title = "European average % of young unemployed people",
     x = "Years",
     y = "% of young people",
     caption = "Data downloaded from Eurostat\ncalculations made by the author")
  theme_minimal()
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

[10]:
```
Output in Figure 1
```

4.1 Data pre-processing

While the datasets provided by Eurostat are in clean format, they almost always require some data pre-processing. Here, there are three main tasks required to bring the data in the required format to proceed with the analysis. First, to subset the dataset to have only the NUTS 2 regions (the original dataset also includes country and NUTS 1 data). Second, to calculate the quintiles that every NUTS 2 region belongs to in every year. Third, to re-format the dataset from a long (each region represented in multiple rows – one for every year) to a wide format (each region represented by a single row and there are multiple columns containing the yearly young unemployment rates).

In coding terms, the first task is to calculate the number of characters of all geography codes and store them in a new column. I then create a subset of the dataset with only NUTS 2 regions which are those that contain four characters (i.e. the first two represent the country name followed by two numbers represent the NUTS 2 region).

Table 2: Preview of the data that will be used in sequence analysis

	sex	age	training	wstatus	unit	geo	n_char	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
2	T	Y18-24	NO_FE_NO_NFE	NEMP	PC	AT12	4	2	2	1	2	1	1	2	1	1	1	1
3	T	Y18-24	NO_FE_NO_NFE	NEMP	PC	AT13	4	2	2	2	2	2	2	2	3	3	2	3
5	T	Y18-24	NO_FE_NO_NFE	NEMP	PC	AT22	4	1	2	1	1	1	1	1	1	1	1	1
6	T	Y18-24	NO_FE_NO_NFE	NEMP	PC	AT31	4	1	1	1	1	1	1	1	1	1	1	1
8	T	Y18-24	NO_FE_NO_NFE	NEMP	PC	AT33	4	1	1	1	1	1	1	1	1	1	1	1
10	T	Y18-24	NO_FE_NO_NFE	NEMP	PC	BE10	4	5	4	4	4	4	4	4	4	4	4	4

[11]:
```
# Create a new column to store the number of characters of the geography
young_unempl$n_char <- nchar(as.character(young_unempl$geo))
# Subset only the NUTS 2 regions - their geography code contains 4 characters
young_unempl_NUTS2 <- young_unempl %>%
  filter(n_char == 4)
```

The next task is to calculate quintiles by year for each region based on their % of youth unemployment. This was done by looping through each year.

[12]:
```
# Calculate quintiles by year
# It is good to specify  the filter function to be used from dplyr function to avoid
# error messages
quant_data <- NULL
for (var in unique(young_unempl_NUTS2$time)) {
    young_unempl_NUTS2_temp <- young_unempl_NUTS2 %>%
      dplyr::filter(time == var) %>%
      mutate(quintiles = ntile(values, 5) )
    quant_data <- rbind(quant_data,young_unempl_NUTS2_temp)
}
```

The final task is to keep only the column containing the quintiles (the actual percentages will not be used in the rest of this notebook). The dataset will then be re-formatted to a wide format where each region will be represented by a single row. For each region there are multiple columns containing the corresponding yearly young unemployment quintiles. Finally, I delete all the rows that contain missing values. This is important as there are regions that have "gaps" in their data availability, meaning that there is no data in at least one year between 2008 and 2018. These regions are ignored in this analysis to speed up computational time and to have consistency across sequences.

[13]:
```
# I delete the column inlcuding the % as I will use the quintiles from now on in the
# analysis
quant_data <- subset(quant_data, select = -values)
# Re-format the data from long to wide format
# This means that every row will represent a region and every column represents a year
quant_data_wide <-
  dcast(quant_data,
  sex  age  training  wstatus  unit  geo  n_char ~ time,
  value.var = 'quintiles')
# We remove rows that  do not have values in at least one year so we have consistency
# between sequences
quant_data_wide <- na.omit(quant_data_wide)
# Have a look at the dataset
kable(head(quant_data_wide))
```

[5]:
```
Output in Table 2
```

4.2 Sequence object

Creating a sequence object is the initial point of sequence analysis. In this notebook I pass a subset of the columns of the dataset that contain the quintile values. Practically, it means that I create a sequence of quintiles from 2008 to 2018 which are from the 8th to 18th column for each region. As I have already mentioned, I have used the R package `TraMineR` (https://cran.r-project.org/web/packages/TraMineR/index.html). For more detailed information on sequence analysis and all the functionalities, please refer to the user guide (http://mephisto.unige.ch/pub/TraMineR/doc/TraMineR-Users-Guide.pdf) which provides detailed information on all the functionalities of the package.

Table 3: Substitution costs between quintiles

	1	2	3	4	5
1	0.000000	1.694604	1.985533	2.000000	2.000000
2	1.694604	0.000000	1.591372	1.960797	2.000000
3	1.985533	1.591372	0.000000	1.654430	2.000000
4	2.000000	1.960797	1.654430	0.000000	1.752492
5	2.000000	2.000000	2.000000	1.752492	0.000000

[14]:
```
# Create the sequence object using only the quintiles that every region belongs
seq_obj <- seqdef(quant_data_wide[,8:18])
```

4.3 Measuring sequence dissimilarity

A key element of sequence analysis is to calculate "distances" between each pair of sequences that can be used later for the Optimal Matching analysis. These distances are a measure based on how similar two sequences are. There are two components related to these distances. First is the insertion/deletion (indel) cost which is used when the length of sequences is not the same. This is the cost of deleting or inserting a state in a sequence so all the sequences have the same length. On this notebook, the time period covered is the same for every region (i.e. from 2008 to 2018) so the sequence length is fixed, an 11-state long sequence. Hence this step is not required.

The second component for calculating "distances" between each pair of sequences is to calculate the substitution costs for transforming one state (i.e. one quintile group) to another. Substitution costs can be theory-driven or empirically-driven (Salmela-Aro et al. 2011). Theory-driven costs are usually used when researchers define costs based on pre-determined concepts. Thus, the costs between states are solely dependent on the researchers' choices (i.e. how "far" is one state from another). On the other hand, empirically-driven costs are based on the observed transitions between states. Hence, two states are closer when there are more observed transitions between them. In this notebook I follow the empirically-driven approach as I intend to explicitly consider the observed transitions between states (i.e. quintile groups here).

By following the empirically-driven approach, there are two options to calculate substitution costs. The first is to assign a constant value for substituting sequence states (i.e. quintiles). The second option is to calculate transition rates which are the probabilities of transitioning from one state to another (between quintiles in this notebook). These transition rates are then used to calculate the substitution costs as shown in Equation (1). In this analysis, I have used transition rates (i.e. `method = "TRATE"`) because it is important to capture the higher probability of transitioning between 1st and 2nd quintile compared to 1st and 5th quintile. By assigning a constant value, this information would have been missed. For more detailed information on substitution costs please refer again to the `TraMineR` user guide (http://mephisto.unige.ch/pub/TraMineR/doc/TraMineR-Users-Guide.pdf).

Table 3 shows the substitution costs of this study. It is clear from the table that the probability of transitioning between 1st and 2nd quintile is higher than transitioning from 1st to 5th quintile. Hence, the substitution cost from 1st to 2nd is lower than the 1st to 5th. This information will then be used in the next step – the Optimal Matching.

[15]:
```
# Calculate substistution costs
subs_costs <- seqsubm(seq_obj, method = "TRATE")
# Print the substitution costs
kable(subs_costs)
```

[5]:
```
Output in Table 3
```

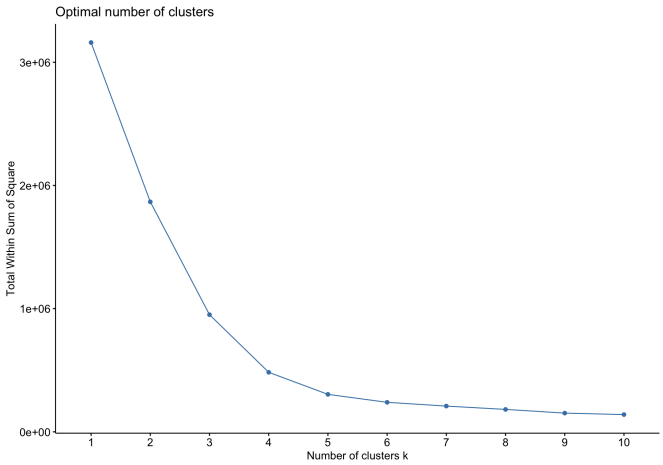


Figure 2: Within sum of squares to access optimal clustering solution

4.4 Dissimilarity matrix

The dissimilarity matrix is a symmetric matrix between all the regions. The matrix is populated by calculating the difference of the sequences between every pair of regions and is symmetrical because each row and column represents a NUTS 2 region (as it happens in any distance matrix). Each region consists of an 11-state long sequence (i.e. from 2008 to 2018). As in the previous stages there are different options to compare sequences. In this notebook I have used the simple Optimal Matching algorithm which uses the substitution costs between the quintiles and aggregates them for every pair of sequences. 'Dynamic Hamming' distance is an alternative method of Optimal Matching which calculates different substitution costs for every time period, assuming that the probabilites of transitioning between different states significantly change over time. Another variant of Optimal Matching is the 'Optimal Matching of Transition Sequences' that accounts for the sequencing of states by explicitly considering their order. In this notebook I have used the simple Optimal Matching algorithm as the aim of the notebook is to present how sequence analysis can be applied to the context of exploring youth unemployment change without making any assumptions that the timing or the ordering of states is considered more important which other Optimal Matching methods can explicitly capture. Studer, Ritschard (2016) provide a good review of different variants of Optimal Matching.

Hence, using the Optimal Matching method a dissimilarity matrix between all NUTS 2 regions has been built based on the substitution costs shown in Table 3. Lower costs mean that sequences are more similar, while larger costs mean that they are different. Hence, it is an abstract distance matrix, showing how "close" two sequences are.

[16]:
```
# Calculate the distance matrix
seq.OM <- seqdist(seq_obj, method = "OM", sm = subs_costs)
```

4.5 Classification of sequences

The final analytical step is to classify the sequences based on their similarities. There is a wide range of clustering algorithms to choose from, when it comes to object classification. Here, the Partitioning Around Medoids (PAM) clustering method was selected for classifying sequences. The PAM algorithm is similar to k-means, but is considered more robust (Kaufman, Rousseuw 1991). A dissimilarity matrix can be used as an index. The algorithm iterates to minimize the sum of dissimilarities within clusters, compared to k-means that aims to minimize the sum of squared Euclidean distances. PAM is based on finding k representative objects or medoids among the observations and then k clusters (that should be defined as in k-means) are created to assign each observation to its nearest medoid. There are different fit statistics to assess optimal clustering solutions.

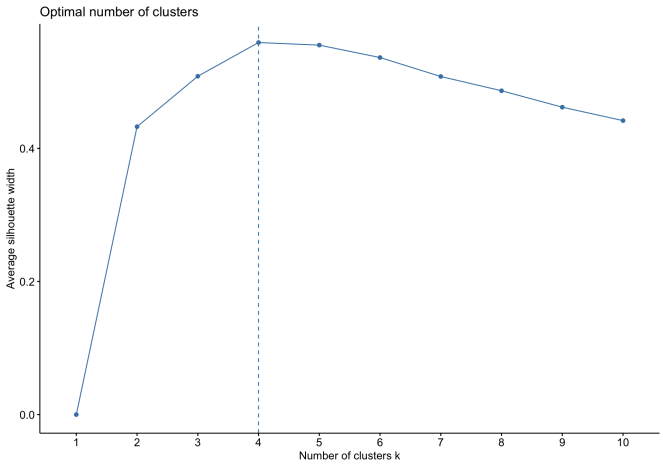


Figure 3: Average silhouette width to access optimal clustering solution

[17]:
```
# Assess different clustering solutions to specify the optimal number of clusters
fviz_nbclust(seq.OM, cluster::pam, method = "wss")
```

[17]:
```
Output in Figure 2
```

[18]:
```
# Assess different clustering solutions to specify the optimal number of clusters
fviz_nbclust(seq.OM, cluster::pam, method = "silhouette")
```

[18]:
```
Output in Figure 3
```

In this notebook I have used two fit statistics (see Figures 2 and 3) to assess various clustering solutions. The focus of this notebook is not to demonstrate the differences between fit statistics. Thus, I will not get into more detail on what every measure means. However, a useful tutorial can be found at https://rstudio-pubs-static.s3.amazonaws.-com/455393_f20bacf1329a49dab40eb393308b33eb.html. In short, they show how well separated each cluster is compared to other clusters (see Figure 2) but also how "compact" the observations are within each cluster (see Figure 3). The fit statistics here show that the optimal clustering solution is four clusters.

[19]:
```
# Run clustering algorithm with k = 4
pam.res <- pam(seq.OM, 4)
```

Having classified the sequences, it is then important to visualise the results to understand differences between the groups. Figure 4 and 5 show the four resulting transition patterns of young unemployment in NUTS 2 regions based on the quintiles they belonged from 2008 to 2018. In Figure 4 each line represents a region, each colour a quintile group and the x-axis represents each year. Figure 5 displays the year-specific distribution of each sequence group. Finally, the y-axis in Figure 4 represents the total number of sequences within each sequence group, while in Figure 5 it represents the distribution of sequences that belong to each sequence group at thus it ranges from 0 to 1.

[20]:
```
# Assign the cluster group into the tabular dataset
quant_data_wide$cluster <- pam.res$clustering
# Then rename clusters
quant_data_wide$cluster <- factor(quant_data_wide$cluster, levels=c(1, 2, 3, 4),
                                  labels=c("Stable Low youth unemployment",
                                           "Stable Moderate youth unemployment",
                                           "Increasingly High youth unemployment",
                                           "Stable High youth unemployment"))
```

For convenience and better communication of the results I assigned names to the four groups starting from the top left plot as:

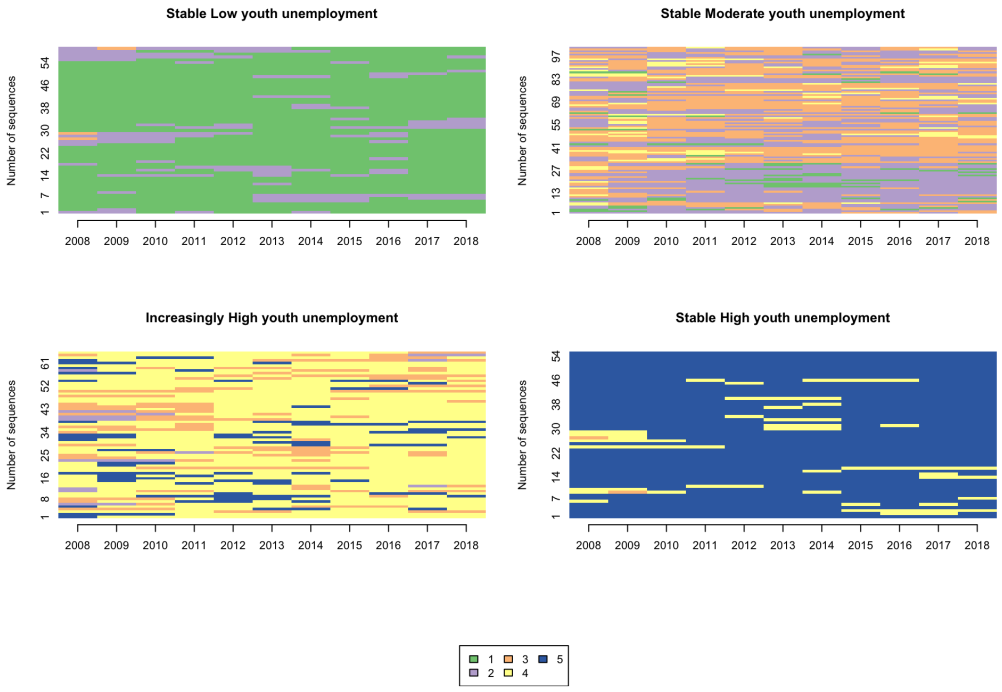


Figure 4: Individual sequences by sequence group

- Group 1 ... Stable Low youth unemployment
- Group 2 ... Stable Moderate youth unemployment
- Group 3 ... Increasingly High youth unemployment
- Group 4 ... Stable High youth unemployment

[21]:
```
# Plot of individual sequences split by sequence group
seqIplot(seq_obj, group = quant_data_wide$cluster, ylab = "Number of sequences")
```

[21]:
```
Output in Figure 4
```

[22]:
```
# Distribution plot by sequence group
seqdplot(seq_obj, group = quant_data_wide$cluster, border=NA,
ylab = "Distribution of sequences")
```

[22]:
```
Output in Figure 5
```

The results of this analysis mainly show patterns of stability in terms of youth unemployment. Group 1 contains regions that are in the lowest quintile, which means they have the lowest youth unemployment ratios over time. Group 4 is exactly the opposite of group 1, containing the regions that belong to the highest quintile over time (highest youth unemployment ratios). Group 2 consists of regions that classified either in the 2nd or 3rd quintile in the last 10 years. Finally, group 3 consists of regions that initially (i.e. 2008) belonged to 3rd, 4th, 5th and few on the 2nd quintile but gradually transformed to the 4th quintile, thus now having a higher percentage of young unemployed people.

5 Exploring spatio-temporal trends of youth unemployment in Europe

Youth unemployment as a socioeconomic phenomenon is of main concern in European policy. Thus, it is important to visualise the findings of this notebook, so that they can be

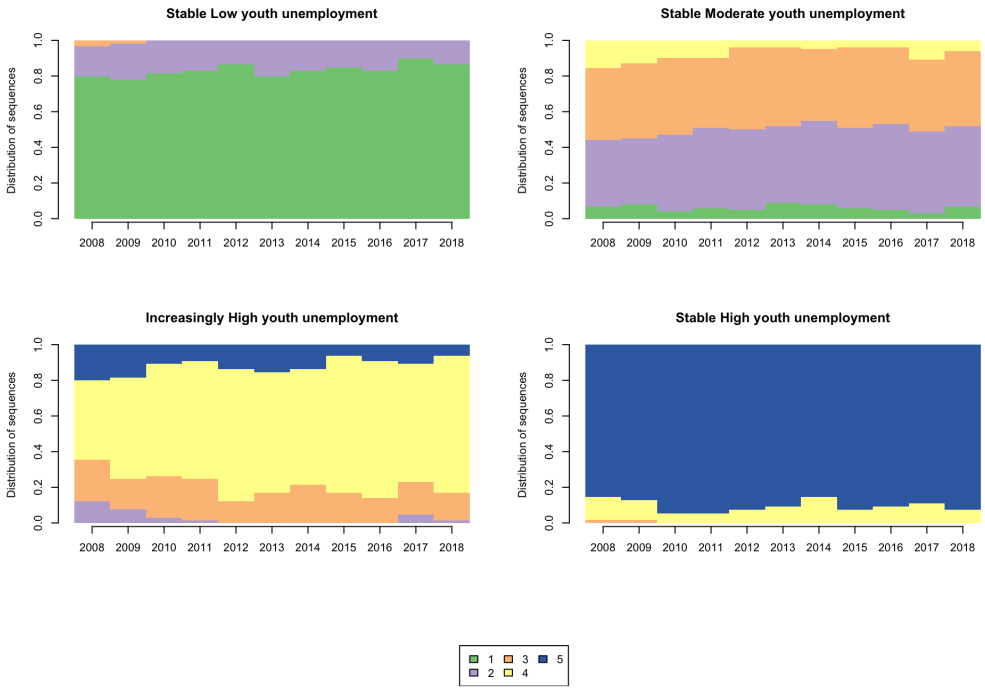


Figure 5: Distribution plot by sequence group

easily explored by the reader. To achieve this, I linked the results of the sequence analysis to the NUTS 2 region geographies so to create an interactive map. I then calculated the frequency of each trajectory group in every European country and created an interactive plot. In this way, the results are more accessible to everyone interested[1].

The map (see Figure 6) offers the opportunity to hover over the regions. Then by clicking on any region, information on the trajectory group, the region name and the country name is shown. The interactive plot (see Figure 7) offers an overview of the frequencies of trajectory groups across European countries. By hovering over the plot, one can observe the exact frequency of each group. It also offers the opportunity to zoom in on particular countries and to manually navigate through the graph (i.e. pan option on the toolbox on the right top of the plot). Finally, by clicking on the legend, particular group(s) can be selected to be shown.

[23]:
```
# Merge the spatial to the tabular dataset which includes the cluster names
map_data <- merge(geodata, quant_data_wide, by.x="FID", by.y="geo", all.x=TRUE)
```

Figures 6 and 7 show spatio-temporal variations of youth unemployment within and across European countries. As illustrated in the map, Mediterranean and Balkan countries (i.e. Greece, Italy, Spain, Turkey, Bulgaria and Romania) have stable high youth unemployment over time. On the other hand, northern countries and central European countries (i.e. Norway, Sweden, Netherlands, Germany, Austria and Switzerland) have stable low youth unemployment over time. Finally, the majority of central European countries and the United Kingdom have followed moderate levels of youth unemployment change. However, there are regional differences highlighting that socioeconomic inequalities are not only apparent between countries but also within their national boundaries. There is a clear split between the stable high youth unemployment in south Italy compared to lower but still increasingly high youth unemployment in the north. Spain has three tiers, split geographically, where the more northern the region, the lower the youth unemployment level. A similar pattern appeared in the United Kingdom, where northern regions have higher youth unemployment levels than southern regions over time.

[1]The interactive figures are included in the HTML-version of the paper or can be generated from the Rmd-file.

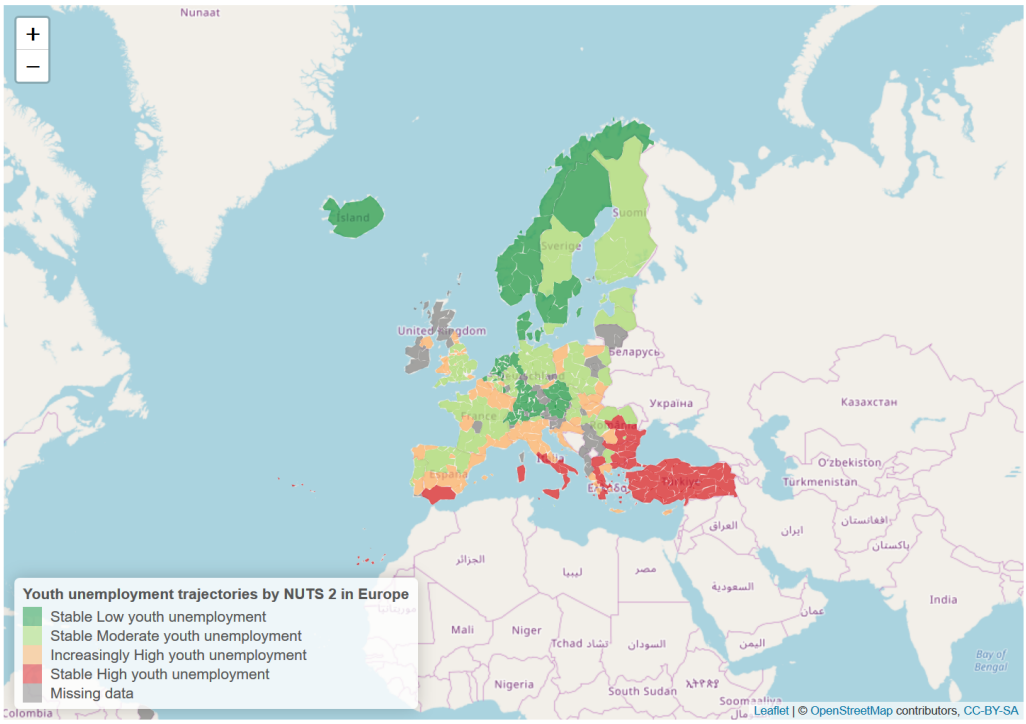


Figure 6: Interactive map of youth unemployment trajectories in NUTS 2 from 2008 to 2018

When looking in more detail at some major metropolitan regions, we can observe deviations from their neighbouring regions. Bucharest and Sofia seem to have lower unemployment levels compared to adjacent regions in Romania and Bulgaria respectively. On the other hand, while Belgium has moderate or low levels of youth unemployment on average, Brussels, its biggest city and capital, has stable higher youth unemployment. Austria follows similar pattern where the country has on average high concentration of 'stable low youth unemployment' regions but its biggest city (and capital) Vienna is classified as 'stable high youth unemployment'. This highlights that higher levels of socioeconomic inequalities and more disadvantaged groups are often aggregated in large metropolitan areas.

[24]:
```
# Create a map showing the distribution of sequence clusters
# Specify the colour palette
myColors <- rev(brewer.pal(4,"RdYlGn"))
pal <- colorFactor(myColors, domain = unique(map_data$cluster))
# Create the initial background map, zooming in Europe
colourmap <- leaflet() %>%
  addTiles() %>%
  setView(lat = 55, lng = 1, zoom = 3)
# Create the interactive map showing the sequence clusters
colourmap %>%
  addPolygons(data = map_data,
    fillColor = ~pal(cluster),
    weight = 0.2,
    opacity = 0.8,
    color = "white",
    dashArray = "3",
    fillOpacity = 0.7,
    popup = paste("Cluster: ", map_data$cluster, "<br>",
                  "NUTS 2 Name: ", map_data$NUTS_NAME, "<br>",
                  "Country Name: ", map_data$cntr_name, "<br>"),
    highlight = highlightOptions(
      weight = 5,
      color = "#666",
      dashArray = "",
```

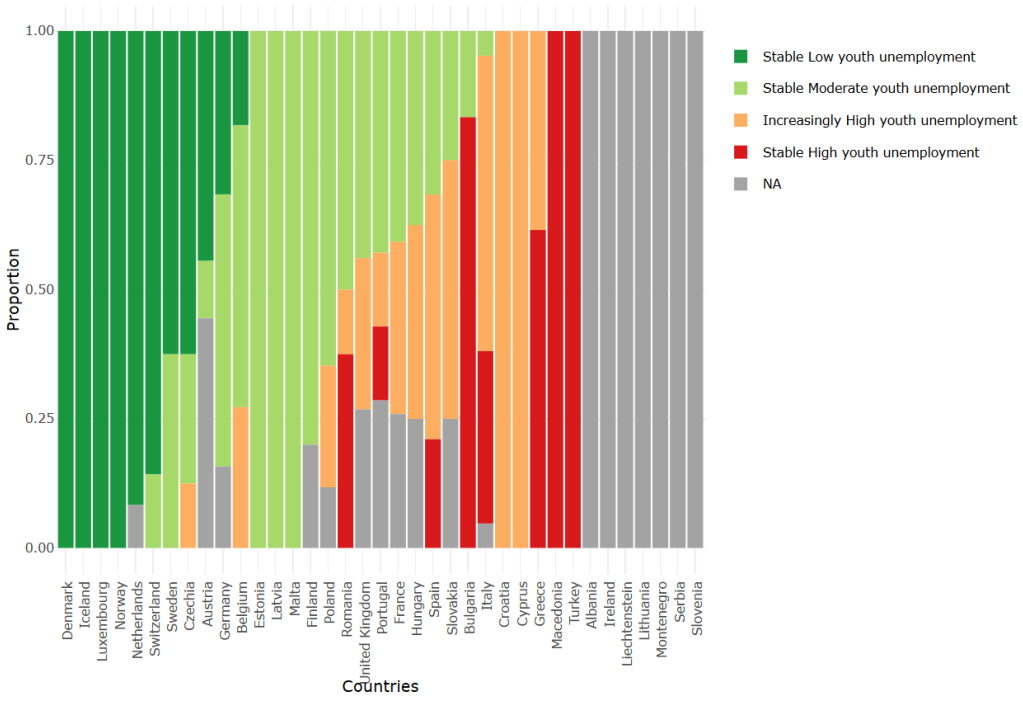


Figure 7: Distribution of youth unemployment trajectories across Europe from 2008 to 2018

```
      fillOpacity = 0.7,
      bringToFront = TRUE)) %>%
  addLegend(pal = pal,
            values  = map_data$cluster,
            na.label = "Missing data",
            position = "bottomleft",
            title = "Youth unemployment trajectories by NUTS 2 in Europe")
```

[24]:
```
Output in Figure 6
```

[25]:
```
# Calculate country summary statistics
freq_reg <- map_data@data %>%
  group_by(cntr_name, cluster) %>%
  summarise(n = n()) %>%
  mutate(freq = n / sum(n))
```

[26]:
```
# reformat the data to order them by clusters frequency
data_wide <- dcast(freq_reg, cntr_name ~ cluster, value.var="freq")
data_wide <- data_wide[order(-data_wide$`Stable Low youth unemployment`,
                             -data_wide$`Stable Moderate youth unemployment`,
                             -data_wide$`Increasingly High youth unemployment`,
                             -data_wide$`Stable High youth unemployment`),]
# Create a bar plot for country distribution of clusters
distribution_plot <- ggplot()
  geom_bar(aes(y=freq, x=cntr_name, fill=cluster), data=freq_reg, stat="identity")
  labs(title = "Distribution of youth unemployment trajectories across Europe",
       x = "Countries", y = "Proportion", fill = "")
  theme_minimal()
  theme(axis.text.x=element_text(angle = 90, hjust = 1))
  scale_x_discrete(limits=c(data_wide$cntr_name))
  scale_fill_brewer(palette="RdYlGn", na.value = "grey64", direction = -1)

# Set an interactive mode to the plot
ggplotly(distribution_plot)
```

[26]:
```
Output in Figure 7
```

6 Conclusion

Sequence analysis offers the opportunity to understand long-term socioeconomic trends over various levels of geographic regions. Clustering regions that follow similar socioeconomic trajectories can guide local, regional, national or European policy making by identifying and reducing the socioeconomic segregation of disadvantaged population groups. The first aim of this notebook was to highlight (NUTS 2) regions in Europe that maintain high, moderate or low youth unemployment levels, as well as regions that have transitioned from one level to another over the last decade. The findings of this notebook showed that northern Europe has high concentrations of regions with stable low youth unemployment, while southern Europe has high concentrations of regions with stable high youth unemployment. It is observed that southern countries struggled to adapt to the financial crisis of 2008. These findings can be used as a starting point to understand migration patterns that originated from these "disadvantaged" regions within or outside European boundaries. The second aim of this notebook was to provide a self-contained reproducible and transparent analytical workflow. This aim was achieved by providing detailed steps for successful data manipulation and make use of sequence analysis which is not a commonly used method in regional studies. Hence, I hope that data and regional scientists can benefit from the functionalities offered in the notebook and use it as a complementary guide when analysing their own data.

Acknowledgment

I would like to acknowledge the useful and constructive feedback received from my PhD supervisor Dr. Francisco Rowe throughout this research project. I would also like to thank my fellow PhD students at the University of Liverpool Krasen Samardzhiev and Patrick Ballantyne as well as the two anonymous reviewers for their useful comments on earlier versions of the notebook.

References

Bell DNF, Blanchflower DG (2011) Young people and the great recession. *Oxford Review of Economic Policy* 27[2]: 241–267. CrossRef.

Brzinsky-Fay C (2007) Lost in transition? Labour market entry sequences of school leavers in Europe. *European Sociological Review* 23[4]: 409–422. CrossRef.

Delmelle EC (2016) Mapping the DNA of urban neighborhoods: Clustering longitudinal sequences of neighborhood socioeconomic change. *Annals of the American Association of Geographers* 106[1]: 36–56. CrossRef.

Dietrich H (2012) Youth unemployment in Europe – Theoretical considerations and empirical findings. Friedrich ebert stiftung, bonn

Kaufman L, Rousseuw PJ (1991) Finding groups in data: An introduction to cluster analysis. Vol. 47. 2. CrossRef.

O'Reilly J, Eichhorst W, Gábos A, Hadjivassiliou K, Lain D, Leschke J, McGuinness S, Kureková LM, Nazio T, Ortlieb R, Russell H, Villa P (2015) Five characteristics of youth unemployment in Europe: Flexibility, education, migration, family legacies, and EU policy. *SAGE Open* 5[1]: 1–19. CrossRef.

Patias N, Rowe F, Cavazzi S (2020) A scalable analytical framework for spatio-temporal analysis of neighborhood change: A sequence analysis approach. In: Kyriakidis P, Hadjimitsis D, Skarlatos D, Mansourian A (eds), *Geospatial Technologies for Local and Regional Development*. Springer International Publishing, Cham, 223–241. CrossRef.

Peng RD (2011) Reproducible research in computational science. *Science* 334[6060]: 1226–1227. CrossRef.

Pop A, Kotzamanis B, Muller E, McGrath J, Walsh K, Peters M, Girejko R, Dietrich C (2019) YUTRENDS – Youth unemployment: Territorial trends and regional resilience. ESPON, Luxemburg

Rowe F, Casado-Díaz JM, Martínez-Bernabéu L (2017a) Functional labour market areas for Chile. *REGION* 4[3]: R7–R9. CrossRef.

Rowe F, Corcoran J, Bell M (2017b) The returns to migration and human capital accumulation pathways: Non-metropolitan youth in the school-to-work transition. *Annals of Regional Science* 59[3]: 819–845. CrossRef.

Rule A, Birmingham A, Zuniga C, Altintas I, Huang C, Knight R, Moshiri N, Nguyen MH (2019) Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. *PLoS Computational Biology* 15[7]. CrossRef.

Salmela-Aro K, Kiuru N, Nurmi J, Eerola M (2011) Mapping pathways to adulthood among finnish university students: Sequences, patterns, variations in family- and work-related roles. *Advances in Life Course Research* 16[1]: 25–41. CrossRef.

Sandve GK, Nekrutenko A, Taylor J, Hovig E (2013) Ten simple rules for reproducible computational research. *PLoS Computational Biology* 9[10]. CrossRef.

Sanger F, Nicklen S, Coulson AR (1977) DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences* 74[12]: 5463–5467. CrossRef.

Studer M, Ritschard G (2016) What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures. *Journal of the Royal Statistical Society. Series A: Statistics in Society* 179[2]: 481–511. CrossRef.