uc3m | Universidad Carlos III de Madrid

Master in Big Data Analytics
2017/2018

Master's Thesis

France Regional Electricity Consumption Clustering Using Generalized Cross Correlation

Pierre Mercatoris

Supervisors
Andrés M. Alonso
Daniel Peña
Madrid y fecha de presentación prevista

Keywords: Clustering, time series, electricity consumption, distance metric.
Summary: This work attempts to characterise the electricity consumption of the regions of France between 2013 and late 2017. It does this by applying the Generalised Cross Correlation allowing the clustering of time series by their linear dependency. Each cluster's trend and consumption patterns are then discussed.

Contents

List of Figures

1 Introduction

Clustering is a unsupervised machine learning task that tries to assign labels to data. It does this by minimising the within group similarities and maximising the between groups dissimilarities. It is unsupervised as we generally do not know those labels (Jain et al., 1999).
In this section, the main clustering techniques will be introduced followed by an explanation on how those are usually applied to time series. The importance and structure of this work will then be explained.

1.0.1 Static data clustering

As defined by Han et al. (2000), clustering can be classified into multiple kinds of methods: partitioning, hierarchical, density-based, grid-based and model-based. All those methods were developed and thought for static data type, or values fixed at a certain time, which are much easier to deal with without the time dimension.
Partitioning, can be crisp, as each data point is part of one and only one cluster, or fuzzy and can have various degrees of belonging to multiple clusters. Most common crisp partitioning algorithms are the k-means algorithm (MacQueen et al., 1967), and the k-medioids (Rousseeuw and Kaufman, 1990). Similarly for the fuzzy ones, there is the c-means (Bezdek, 1981) and the c-medioids (Krishnapuram et al., 2001).
Hierarchical clustering algorithms can either be agglomerative or divisive, depending on whether it splits or merge clusters at each of its iterations. The main advantage of these methods is that it doesn't require any knowledge about the number of clusters. However, once a merge (or division) of clusters occur, it cannot be redefined later. There exist some algorithms that try to solve this issue (Zhang et al., 1996; Guha et al., 1998; Karypis et al., 1999).
Density-based algorithms will grow the clusters until a certain density threshold is encountered in the neighbourhood. An example is the DBSCAN algorithm (Ester et al., 1996). Grid-based algorithms will split the space into a finite number of cells and perform the clustering operations of those cells Wang et al. (1997). Finally, model-based clustering is either based on statistical techniques like in Cheeseman et al. (1988) or on neural networks (see Carpenter and Grossberg (1987); Kohonen (1998)).

1.0.2 Dynamic Data and Time series clustering

Time series are usually large datasets that have inherited, over time, of a high dimensionality of dependent measurements. Interestingly, it is possible to look at a series either as a high dimensional vector or as a single data point. As shown by Liao (2005) and Aghabozorgi et al. (2015), the interest for time series clustering has been growing in a wide variety of fields. It is often an important step of exploratory analysis.
As defined by Liao (2005) in his review (see Fig. 1), there are three types of time series clustering. Raw-data based methods compare series directly and will usually compute a distance or similarity metric. Then, there are feature based and model based data that summarise each series into a set of parameters. The problem with those is that it often can oversimplify the series, and although faster to compute, requires many assumptions about each series to be made.

Furthermore, as explained in Alonso and Peña (2017), previous time series clustering methods have been using similarities among series based on their univariate models in a parametric framework, by its periodogram or their autocorrelation. Those methods are useful when the series are independent and can be classified using their univariate models. However, it is also possible use the linear dependencies present within those series, by using distances such as cross-correlation that will summarise the correlation at multiple lags. This can be important for series that do not demonstrate obvious classification patterns.
Aghabozorgi et al. (2015) has identified the four main aspects of time series clustering research: the **dimensionality reduction** necessary to remove noise and reduce the size of large series, **similarity/distance measures**, **clustering algorithms** and cluster **prototypes** (or how to characterise a group of data).

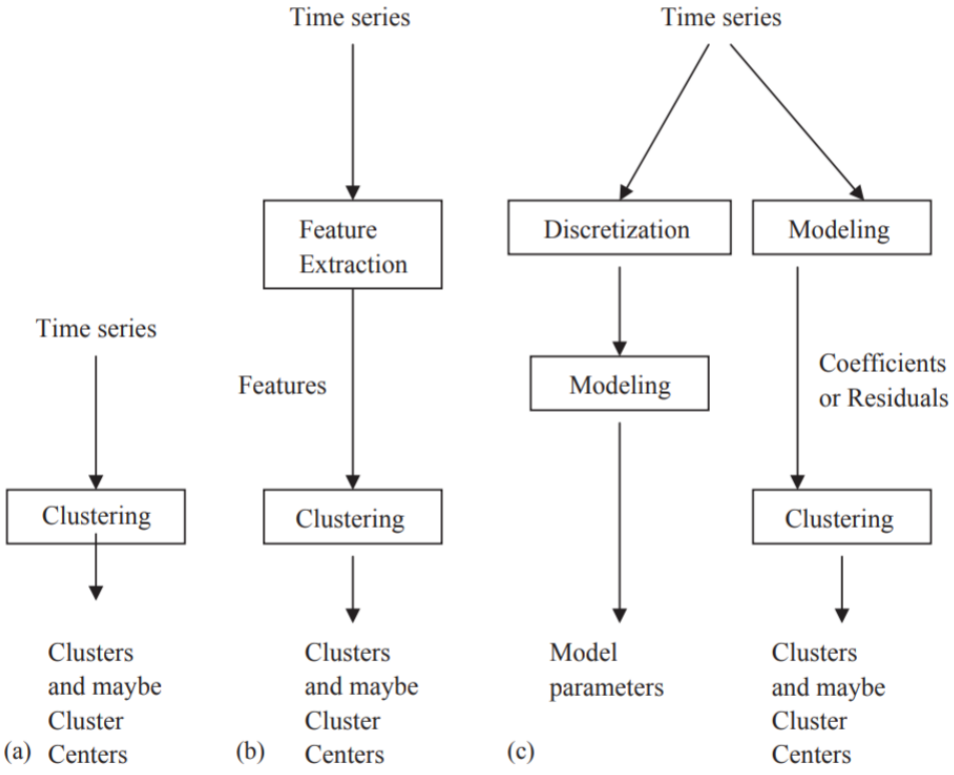

Figure 1: Three time series clustering approaches: (a) raw-data-based, (b) feature-based, (c) model-based. (Liao, 2005)

Essentially, once the distance matrix or extracted parameters of the time series extracted, most of the static data clustering methods can be used as is.

1.1 Cluster electricity consumption using GCC

The clustering of electricity consumption series is crucial for the detection of patterns and trends, and to predict the future consumption of a population. Van Wijk and Van Selow (1999) was one of the first to do this using the consumption in the Netherlands. Their approach was to use agglomerative hierarchical clustering on daily power consumption based on the root mean square distance.

The RMSD calculated between two series, x and y of length T, is calculated as such:

$$RMSD = \sqrt{\frac{\sum_{t=1}^{T}(x_t - y_t)^2}{T}} \tag{1}$$

and is basically just a Euclidean distance scaled to the length of those series. Those distances are considered univariate, as it does not look at the linear dependencies present within the series.

In this thesis, I would like to show an application of the Generalised Cross Correlation, defined by Alonso and Peña (2017). It is a similarity metric based on cross correlation that can cluster using the linear dependency between the series. As such, it is very general and non parametric. However, as demonstrated by Alonso and Peña (2017), clustering by cross dependency can produce significantly different results compared to those other methods, such as univariate distances like the Euclidean distance.

This study first describes the data and the required preparation be applying the GCC. The metric was implemented both in R and in Python (for comparison purpose). A simple agglomerative hierarchical clustering is used as it does not require any assumption on the number of clusters. The clusters will then be analysed to find further structure and identify their dynamic patterns. Finally, we will look at the differences with the Euclidean distance in finding clusters of electricity consumption in France.

2 Methodology

2.1 Data description

The electricity consumption was available at a 30 minutes frequency for each of the 12 regions of France from 2013 to 2017. Each year of each region can be downloaded from the French transmission operator (Rte) download portal[1].
Consumption from January 2013 to September of 2017 were downloaded for each of the 12 metropolitan mainland regions of France (excluding Corsica).
Those regions are still very young, as before 2016, those were 21 separate regions. In France, regions lack separate legislative power, but can manage a considerable part of their budget for main infrastructures such as education, public transport, universities and research, and help to businesses. It is therefore expected to find some interesting clusters, where we might see some reminiscence of the old regions.

2.2 Data preparation

2.2.1 Cleaning

The complete data set was spread across 60 different tables (years and regions) that were merged into one large table (Table 1).

Table 1: Original data structure.

Périmètre	Date	Heures	Consommation
Auvergne-Rhône-Alpes	2013-01-01	00:00	ND
Auvergne-Rhône-Alpes	2013-01-01	00:15	
Auvergne-Rhône-Alpes	2013-01-01	00:30	8173
Auvergne-Rhône-Alpes	2013-01-01	00:45	
Auvergne-Rhône-Alpes	2013-01-01	01:00	7944
.....			

As data rarely comes clean, there were some imperfections in the names of the regions. Some days the regions were named after the old ones e.g. Languedoc-Roussillon et Midi-Pyrénées instead of Occitanie, or Aquitaine, Limousin et Poitou-Charentes instead of Nouvelle-Aquitaine.
With the raw data cleaned from imperfections, each column was formatted to required data type. A pivot table was then used so as to move each region as a column, and each row as a consumption measurement. The date then needed to be set as UTC in order to avoid problems at the summer/winter time change. As the original frequency of the data is 15 minutes, with only data every 30 minutes available, the table was resampled by taking the sum for each 30 minutes, resulting in the table below (Table 2).
The region with the highest consumption are observed in the Iles-de-France and the lowest in the Centre-Val de Loire. We can also clearly see yearly seasonality with higher consumption during winter times (Figure 2), from October to March. The lowest mean and lowest variance consumption across France is at the end of July. We can also see a lower general lower consumption during the winter 2013-14, end December - early January. This is most

[1] `http://www.rte-france.com/en/eco2mix/eco2mix-telechargement-en`

Table 2: Regional series before splitting the series by time of the day.

Périmètre	Auvergne-Rhône-Alpes	Bourgogne-Franche-Comté	...
Datetime			
2013-01-01_00:00:00+00:00	NaN	NaN	...
2013-01-01_00:30:00+00:00	8173.0	2357.0	...
2013-01-01_01:00:00+00:00	7944.0	2289.0	...
2013-01-01_01:30:00+00:00	7896.0	2326.0	
2013-01-01_02:00:00+00:00	7882.0	2409.0	

likely due to the particularly mild winter as mentioned in the report of Meteo France[2], which reports it as the second warmest winter since 1989-90.

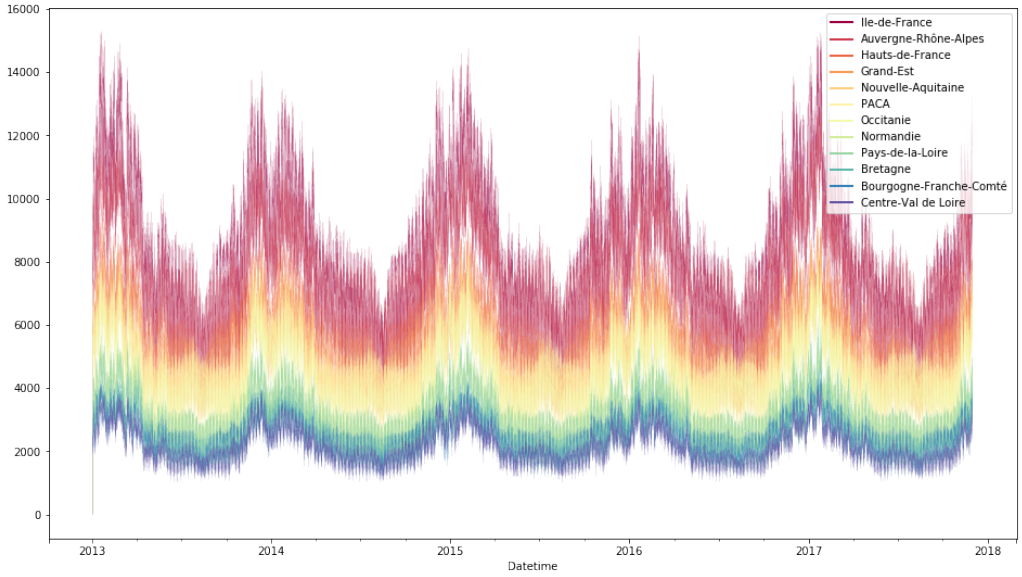

Figure 2: Electricity consumption of each of the French regions from 2013 to end 2017.

The pivot table was used again so that each time of the day is a column, and each row is a daily value for a certain time and region. Furthermore, the first day was removed as it does not have data for midnight. The resulting table has 576 columns (48 half hours x 12 regions) and 1794 rows (days).(Table 3).
In Figure 3, we can already see that consumption midday is much higher than at night, with more spread in the summer than in the winter. That larger spread might be due to the higher need to air conditioning in offices around midday.

2.2.2 Transformation

1. Stationarity

   The original series have a strong seasonality as show in Figure 4.

[2] `http://www.meteofrance.fr/climat-passe-et-futur/bilans-climatiques/bilan-2014/bilan-climatique-de-l-hiver-2013-2014`

Table 3: Final data format before export to csv.

Périmètre	Auvergne-Rhône-Alpes		
time	00:00:00	00:30:00	01:00:00
date			
2013-01-02	7847.0	7674.0	7427.0
2013-01-03	9028.0	8839.0	8544.0
2013-01-04	8982.0	8754.0	8476.0
2013-01-05	8625.0	8465.0	8165.0
2013-01-06	8314.0	8097.0	7814.0

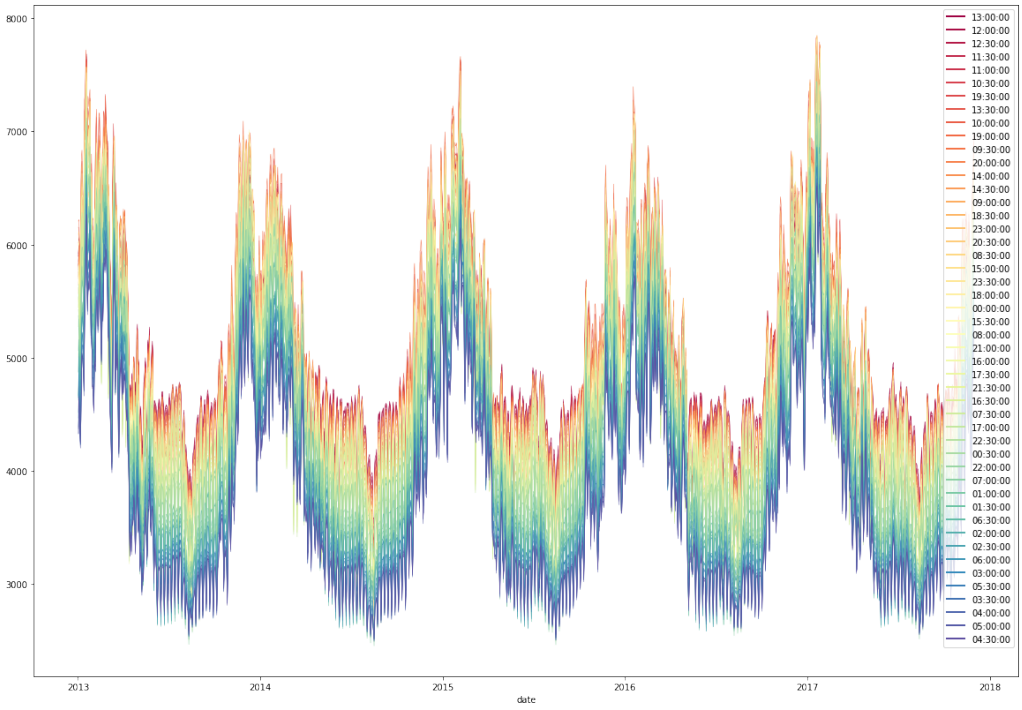

Figure 3: Regional mean electricity consumption at different times.

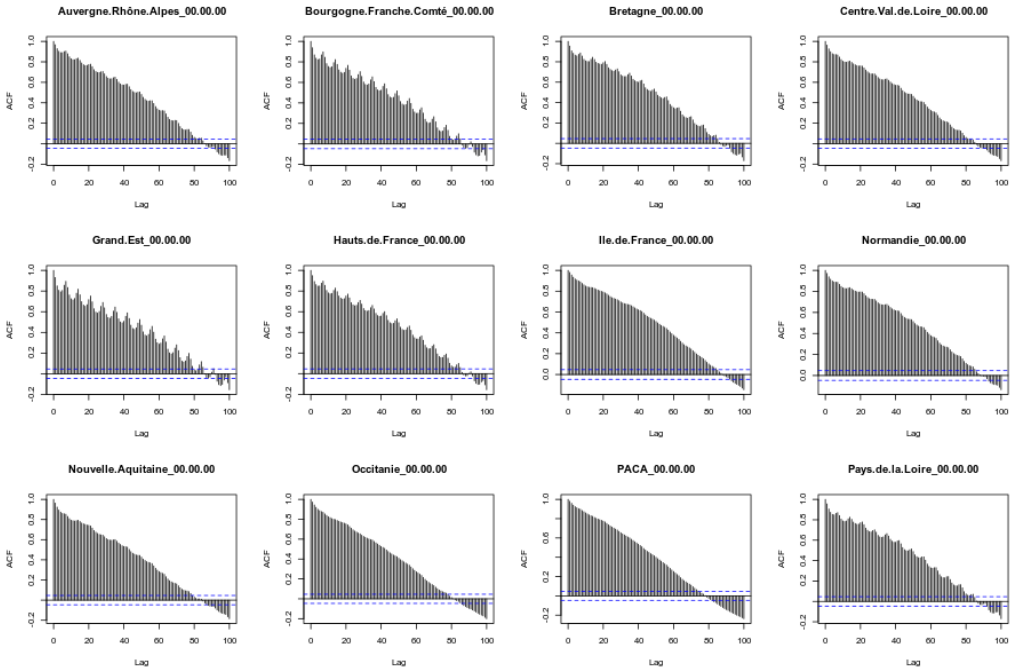

Figure 4: Autocorrelation function for the selected series of the original data.

To try and remove it, we first take a regular difference, since the series have strong dependence compatible to long memory processes. For sake of simplicity, a regular difference instead of fractional ones. (Fig. 5).

The weekly difference was then taken (difference between all the values separated by seven days). This was able to remove most of all seasonality, as most of the values stay within the confidence interval (fig. 6).

The Dickey-Fuller test was used on all the series and confirmed that all the series are now significantly stationary (all p-values lower than $10e^{-21}$).

2. Standardisation

In order to standardise the data so as to get a zero mean and standard deviation of 1, the z-score was applied to each individual series (2).

$$z_t = \frac{x_t - \hat{\mu}}{\hat{\sigma}}, \tag{2}$$

where $\hat{\mu} = \bar{X}_t$ and $\hat{\sigma}^2 = Var(X_t)$.

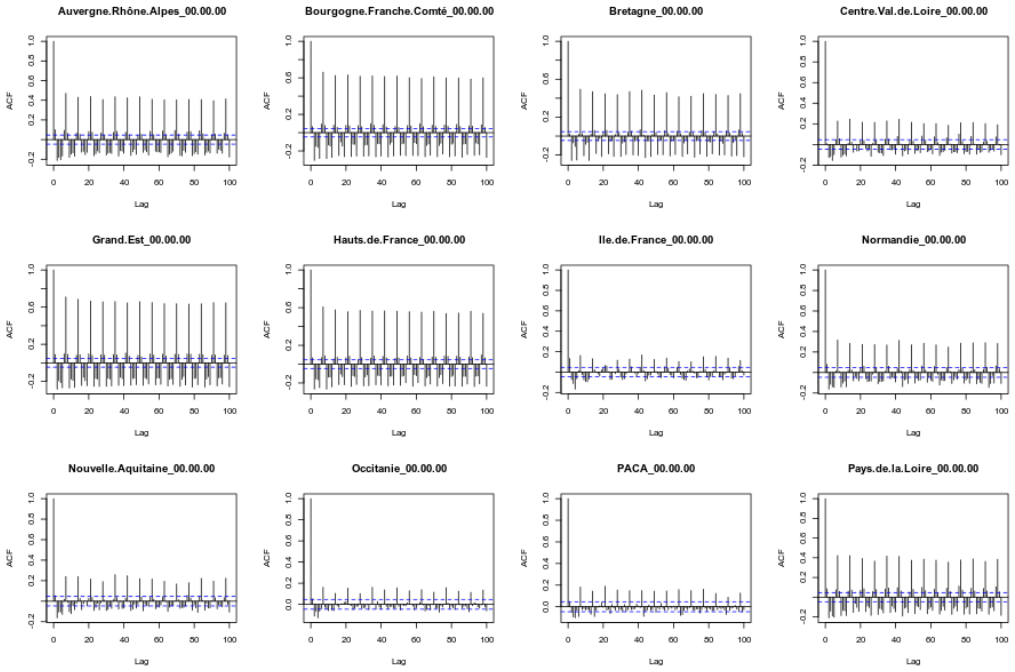

Figure 5: Autocorrelation function of the regularly differentiated series for selected series.

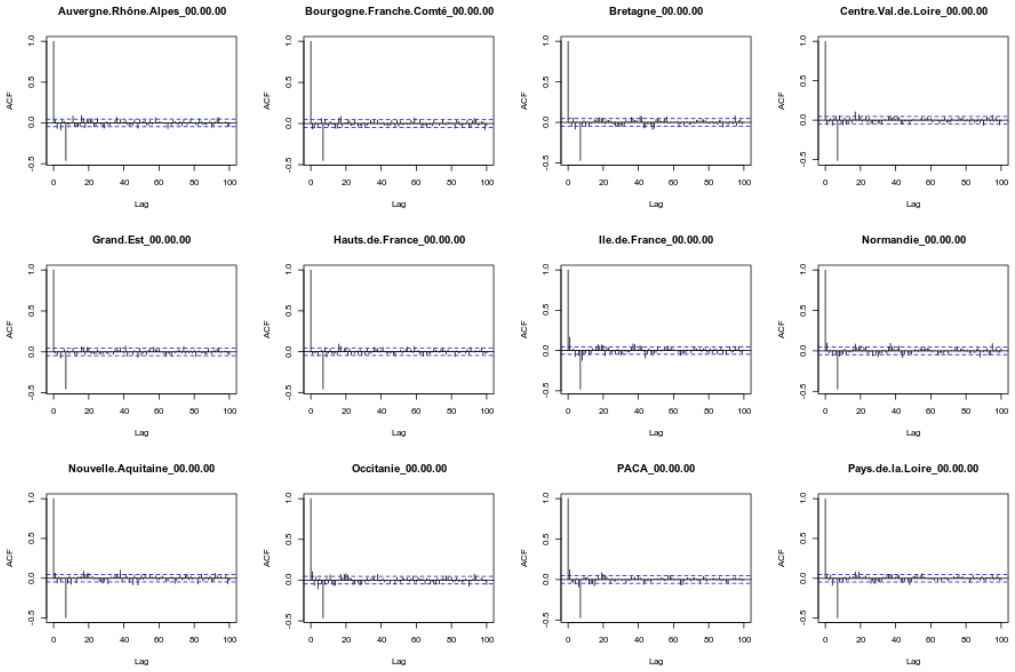

Figure 6: Autocorrelation function of the selected regularly and weekly differentiated series.

2.3 GCC description

As described before, the GCC is a general non parametric similarity metric (as it does not assume any parametric model for the series), that look at the dependencies between series using their cross correlation. The main idea is that it is possible to first cluster the series by the dependency among the series, without any assumption made on those. Then it is possible to break down those more homogeneous clusters looking at the internal dependency of those series.
The GCC computation is based on the determinant of the cross correlation matrices from lag zero to lag k. To do this, for a given k, it is necessary to construct the $X(i)$ and $X(j)$ matrices from the i-th and j-th series (of size T) as follow:

$$X(i) = \begin{pmatrix} X_{i,1} & X_{i,2} & \dots & X_{i,k+1} \\ X_{i,2} & X_{i,3} & \dots & X_{i,k+2} \\ \vdots & \vdots & \ddots & \vdots \\ X_{i,T-k} & X_{i,T-k+1} & \dots & X_{i,T} \end{pmatrix}. \tag{3}$$

With both $X(i)$ and $X(j)$ constructed, we can merge them to form

$$X(i,j) = (X(i), X(j)). \tag{4}$$

The GCC is can then be computed as:

$$\widehat{GCC}(X_i, X_i) = 1 - \frac{|\widehat{R}_{X(i,j)}|^{k+1}}{|\widehat{R}_{X(i)}|^{k+1}|\widehat{R}_{X(j)}|^{k+1}}, \tag{5}$$

where $\widehat{R}$ is the sample correlation matrix of each matrix. This gives a similarity value between 0 and 1 where 1 is the highest possible degree of similarity possible and 0 when there is absolutely no cross dependency between the series.
For clustering it is then necessary to build a distance matrix as such:

$$DM_{\widehat{GCC}} = \begin{pmatrix} 0 & 1 - \widehat{GCC}(X_1, X_2) & \dots & 1 - \widehat{GCC}(X_1, X_N) \\ 1 - \widehat{GCC}(X_2, X_1) & 0 & \dots & 1 - \widehat{GCC}(X_2, X_N) \\ \vdots & \vdots & \ddots & \vdots \\ 1 - \widehat{GCC}(X_N, X_1) & 1 - \widehat{GCC}(X_N, X_2) & \dots & 0 \end{pmatrix}. \tag{6}$$

It is necessary to do $1 - \widehat{GCC}(X_i, X_j)$ if the original measure was calculated as in equation (5), which is a similarity metric, and what is needed here is a distance where 0 corresponds to series close to each other and 1 to series that are far apart.
There are two ways for selecting the number of lag k. Either by taking the maximum order p of all series fitted an auto-regressive model with BIC as the model selection criterion, or using a Dynamic Factor Model which will give more information about the relevant number of lags for the cross correlations, as described in Alonso and Peña (2017).

2.4 GCC calculation

1. Selecting k

   In order to select k, the maximum lag was taken by fitting auto-regressive models to each of the series (using BIC). A maximum lag of 40 was used and was computed both in R and in Python (see Appendix). In both case, it found a maximum fitted lag of 37. This k was considered sufficiently large to capture the cross dependencies between the series and was therefore used.

   This lag seems appropriate when looking at the partial autocorrelation functions in Figure 7, as that is where the last significant value is observed.

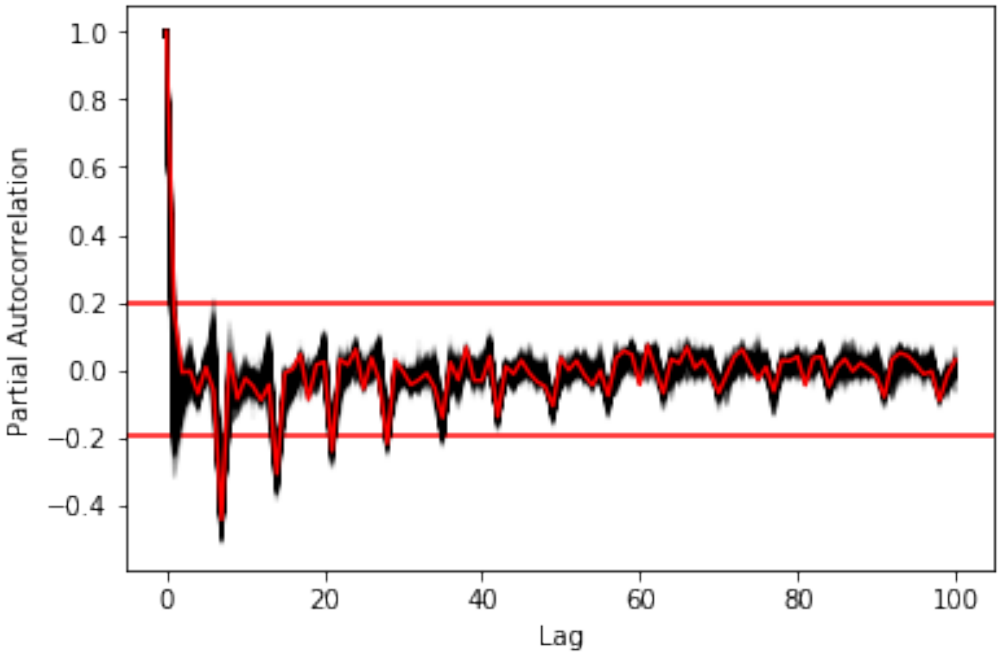

Figure 7: Partial autocorrelation functions of the stationary scaled data.

2. Distance matrix

   The GCC was computed in both R and in Python to validate the results (see Appendix). The maximum difference between the results of the computation in the two language was of $\pm 5.3e^{-15}$ and can therefore be considered equivalent.

3 Results

3.1 Clustering procedure

Hierarchical clustering was used, as it doesn't require a defined number of clusters to be set, and can directly be computed with a distance matrix.

3.1.1 Linkage

More specifically, agglomerative clustering was used, where each data points starts in its own cluster and iteratively gets merged with its closest cluster. There are different methods to compute that intra-cluster distance, referred to as linkage method. The most popular methods were compared using the cophenetic correlation, which is the correlation coefficient between the distances between each point using their cluster distances and the original distance. A value closer to 1 means that the defined clusters respect better the original distances.
As such, in both R and Python, the most conservative method was the average linkage and was therefore used to create the dendrogram (Table 4). Different results were obtained for the 'centroid' and 'median' method, but still didn't beat the 0.77 of cophenetic correlation of the 'average' linkage.

Table 4: Cophenetic correlation of linkage methods.

	Average	Centroid	Complete	Median	Single	Ward	Weighted
Python	0.77	0.73	0.69	0.70	0.69	0.66	0.74
R	0.77	0.55	0.69	0.29	0.69	0.66	0.74

In Figure 8 we can clearly see that there is a lot of structure. There are distances across the whole range of the GCC (0-1), making it easier to distinguish the groups. In fact, the regions appear the be the main influencing factor.

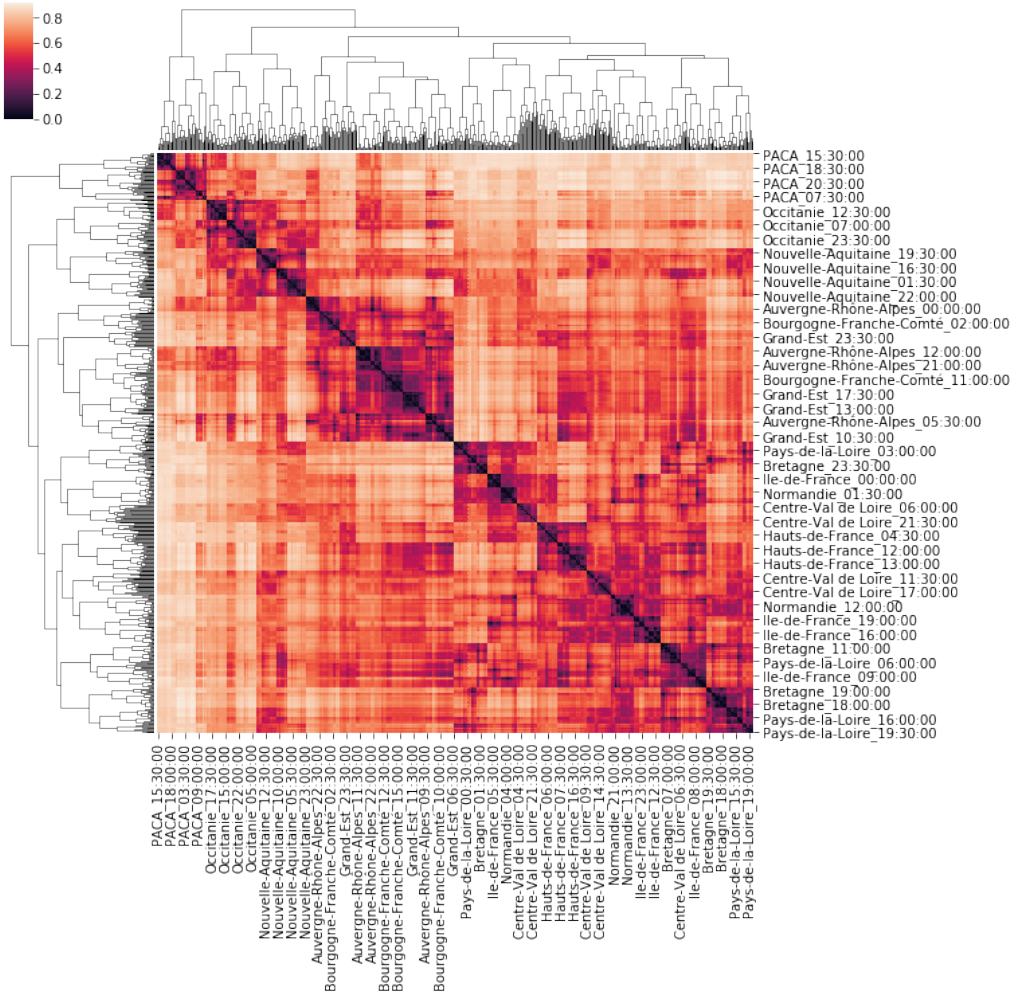

Figure 8: Heatmap of the distance matrix rearranged using the average linkage hierarchical clustering.

3.1.2 Number of clusters

Determining the number of cluster can be very challenging. The silhouette score were compared for different cluster number (Fig. 9). A higher value is observed at two clusters, then a slight peak at five clusters, until it increased rapidly to 44 clusters. That is a high number of cluster, but considering multiple time cluster within each of the 12 regions, we can see how that number can increase quickly. However, most of the analysis will be conducted on five clusters, as it allows to draw some more general conclusions.

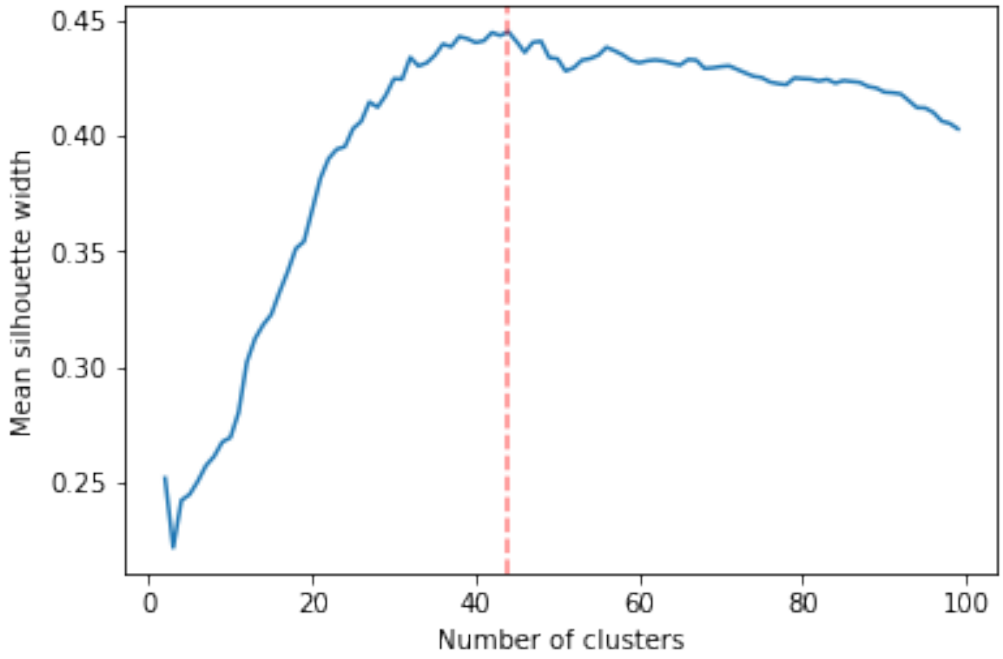

Figure 9: Mean silhouette width for increasing cluster number.

In Fig. 10, we can see that the regions are of high importance to the construction of the clusters. Although 42 clusters is suggested by the silhouette statistic, five cluster seem to be a clear cut. Another way to look at those clusters is by looking the first two principal coordinates of the distance matrix (Fig. 11), where the five clusters are again well defined. We can see that, although cluster 1 and 4 are contain points far away from each other, five clusters is a good generalisation of the data.
In Figure 12, we can see the silhouette width of each of the samples in their respective cluster. There seems to be some misclassification for some series in cluster 3, but overall each of the five cluster has notably high silhouette width.

3.2 Cluster description

In Fig. 13, we can see that cluster 1, 2 and 3 are not related to the time of the day as there are as many series for each of the half-hours. However, cluster 4 is a night cluster (21:30 to 6:00) and cluster 5 is a day cluster (5:30 to 00:00). In fact this is also reflected in Fig. 14, as clusters 4 and 5 share multiple regions. Surprisingly, all the series of the 'Hauts-de-France' were classified into the day cluster 5. Cluster 1 is the only cluster with only one region.

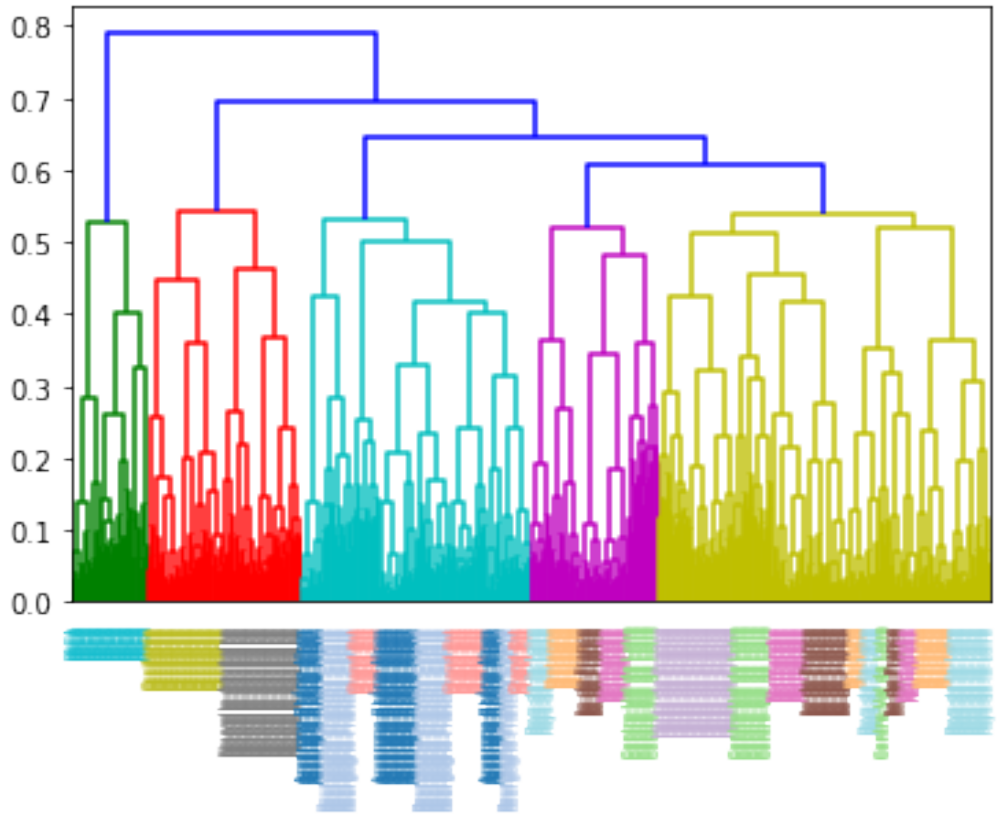

Figure 10: Dendrogram of the distance matrix using average linkage. The five clusters are shown in distinct colours. On the x axis the name of the regions are also shown in distinguishable colours.

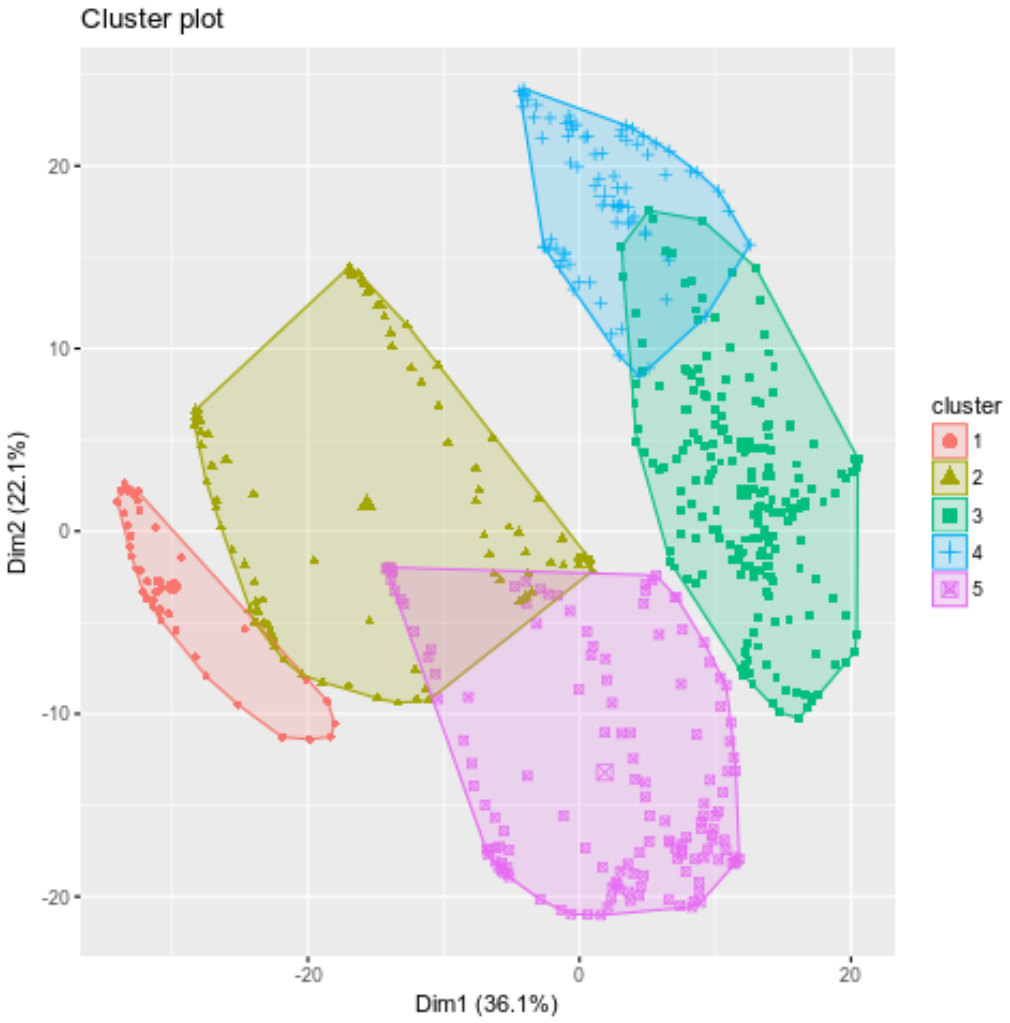

Figure 11: Five clusters over the two principal components of the distance matrix.

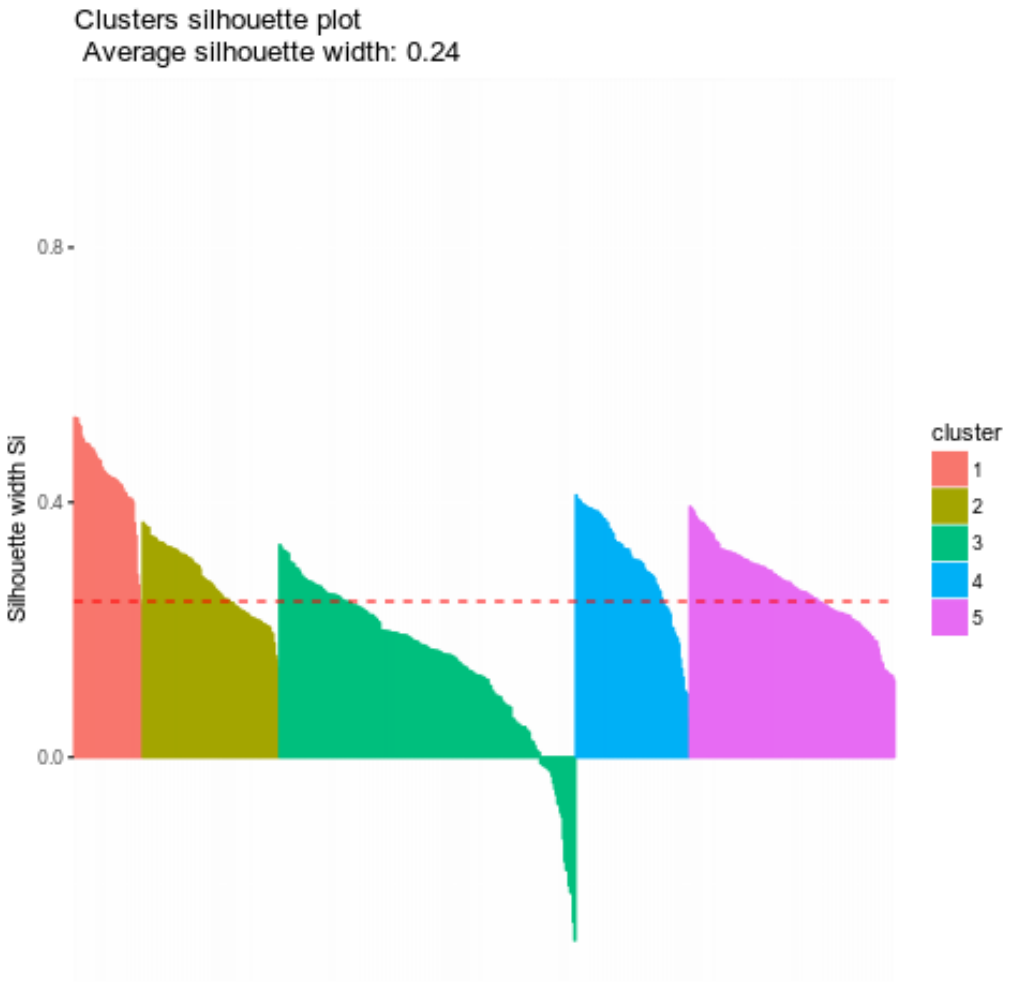

Figure 12: Silhouette width of the samples in each cluster.

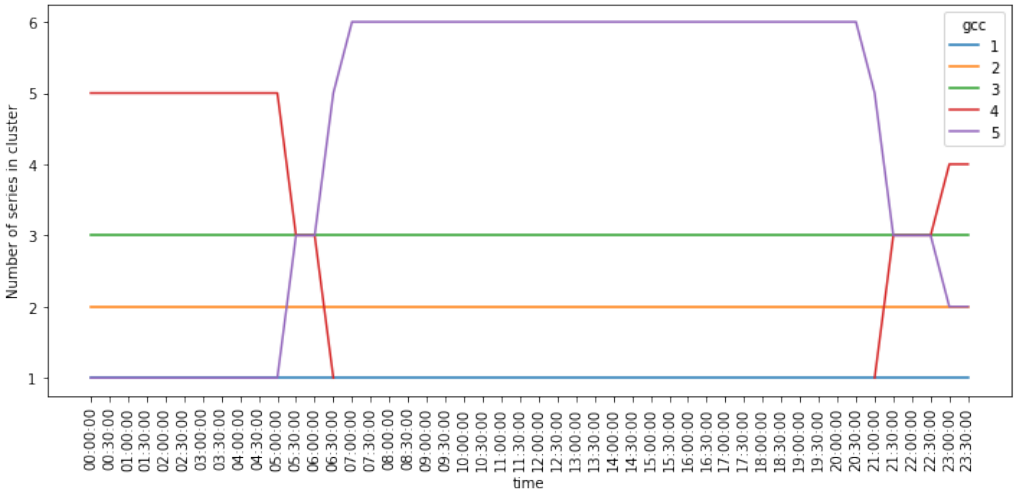

Figure 13: Composition of each cluster over the time of the day.

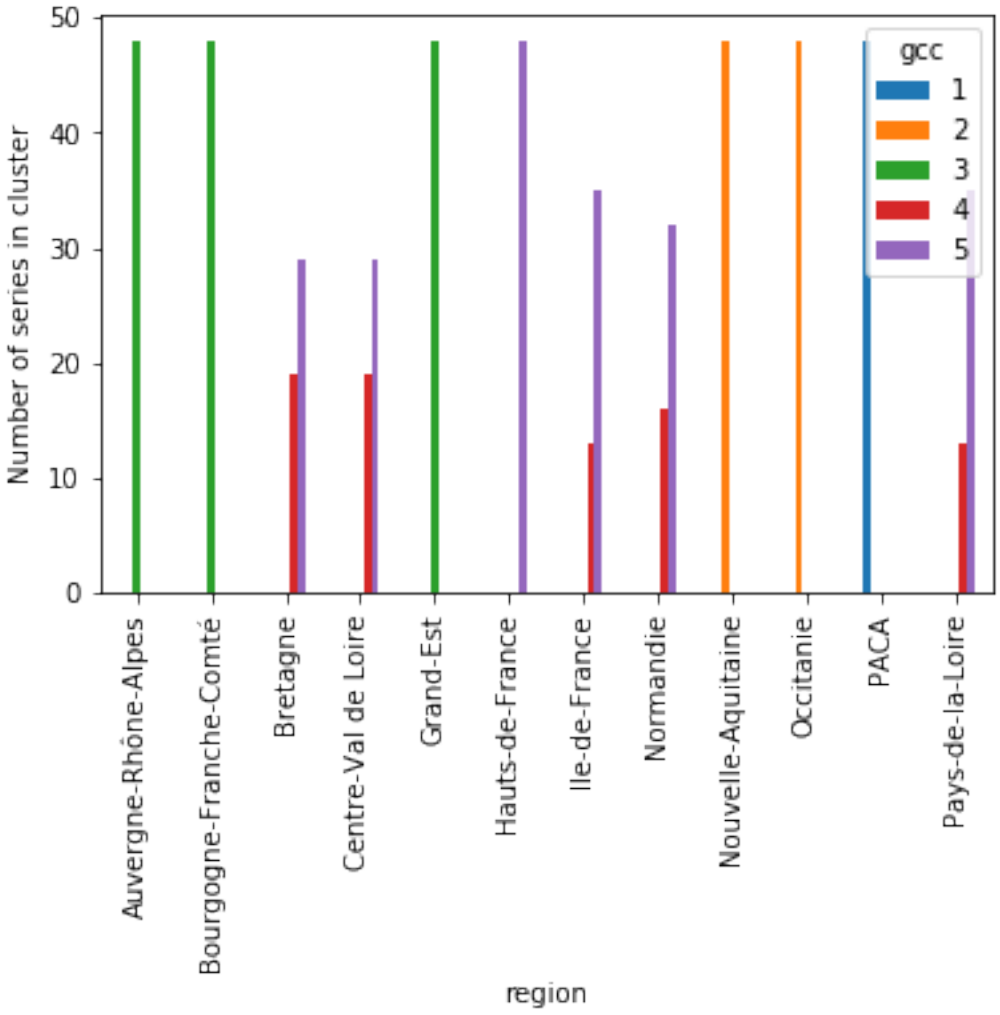

Figure 14: Regional composition of each cluster.

3.2.1 Mapping the clusters

If we were to only use two clusters, the PACA region is clearly the most distinct of all the regions (Fig. 15).

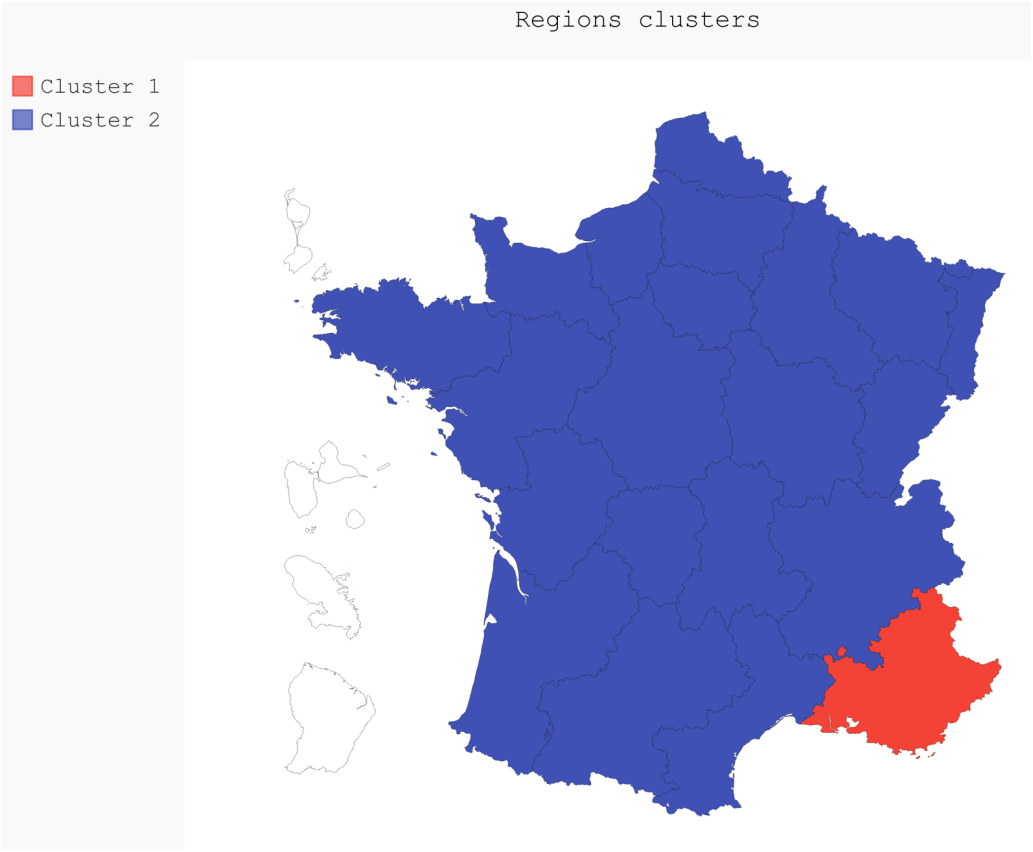

Figure 15: Map of the two clusters on the map of France. The regions shown are the old more numerous regions, but the boundaries of the 12 new regions are the same.

However, in order to have a deeper understanding of the composition of France, five clusters was the other clear delimitation. It is very clear here, that all the clusters have a strong geographical meaning. All regions are in different clusters apart from cluster 4 and 5 that are mixed geographically (Table 5 and Fig. 16), which are more defined by their consumption over time.

Table 5: Regions in each clusters.

1	2	3	4	5
PACA	N-A	A-R-A	Bretagne	Bretagne
	Occitanie	B-F-C	C-V-L	C-V-L
		G-E	I-F	I-F
			Normandie	Normandie
			P-L	P-L
				H-F

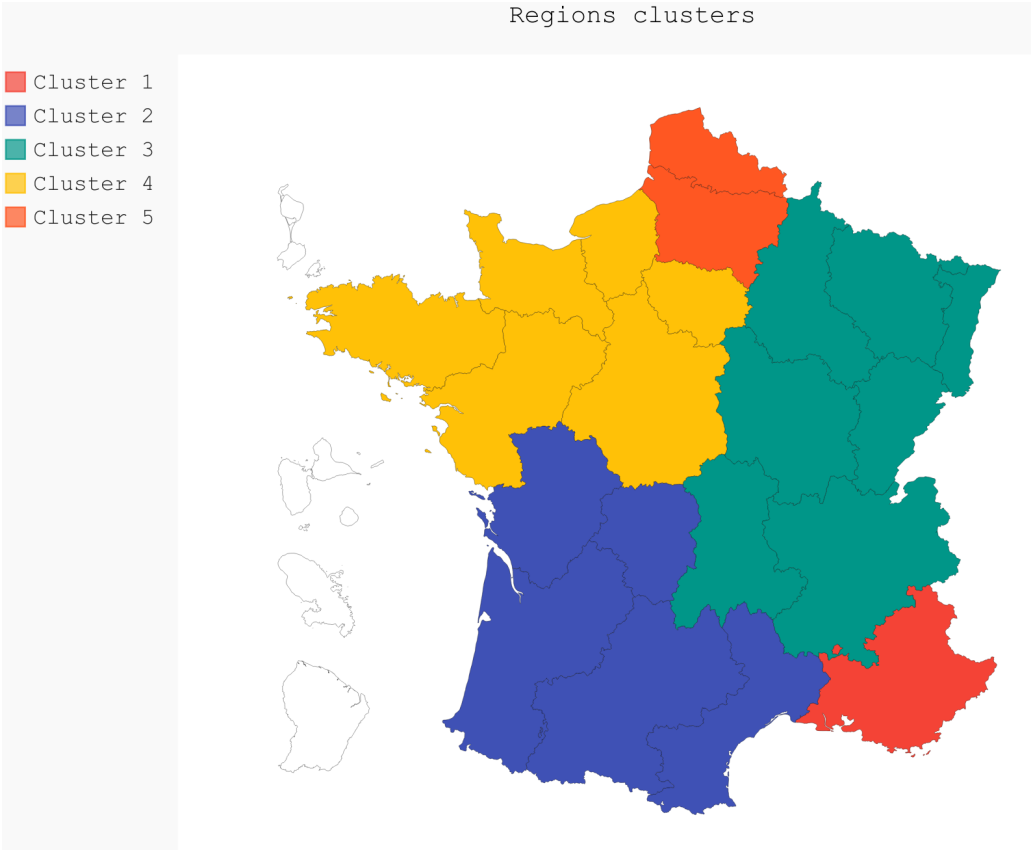

Figure 16: Map of the five clusters on the map of France. The regions shown are the old more numerous regions, but the boundaries of the 12 new reiongs are the same.

3.2.2 Within clusters structure

In this section, the goal was to find out which structure within each of the clusters is left unexplained. A dendrogram was plotted for each cluster and the label was coloured depending on the time of the day, where blue is late in the day and blue is early morning. The lighter colours are towards midday.
In Figure 17, Cluster 1 only contains the PACA region, so the only variable left to explore is the time of the day. As we can see that there are three main clusters, midday-afternoon from 11:30 to 20:00 in green, the night cluster from 20:00 to 6:00 in red and mornings from 6:30 to 11:00 in blue. Days (11:30 to 20:00) and nights (20:30 to 11:00) are however the most well defined.

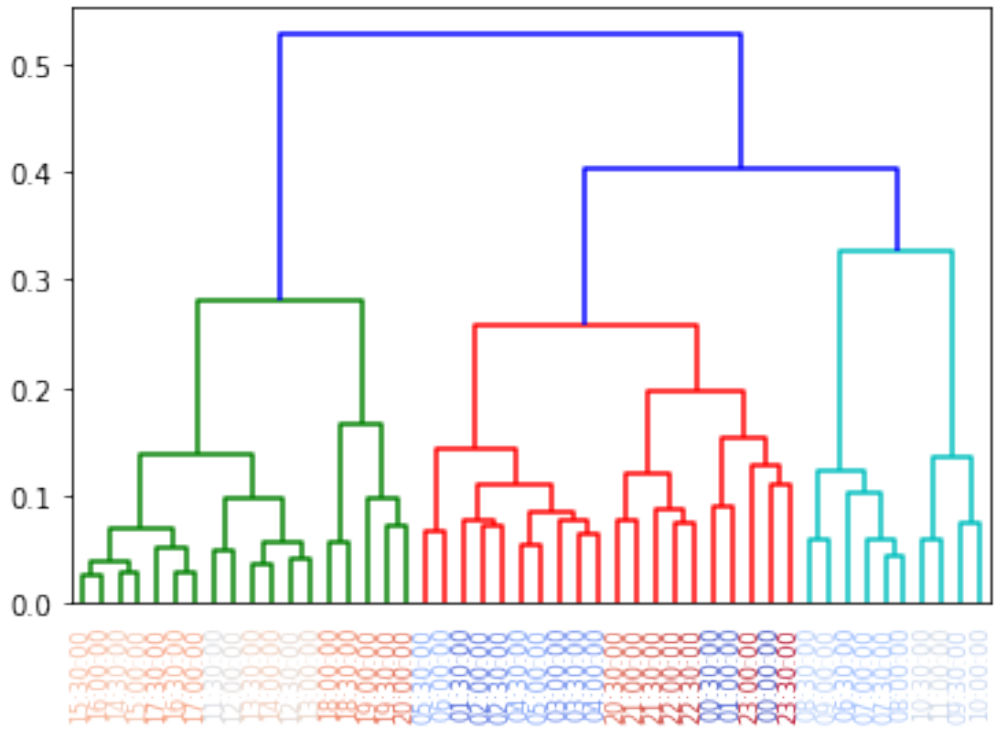

Figure 17: Dendrogram of cluster 1. Red is early morning (00:00 and later) and blue is late at night (until 23:30), lighter colours are towards midday.

Cluster two contains two regions (Nouvelle-Aquitaine and Occitanie). In Figure 18, in the top plot the label is coloured by the region and the bottom plot the label was coloured by the time of the day. We can see that the most important clustering is by region, but then similar clustering as in cluster 1, with mornings, afternoons and nights, as shown by the 6 coloured clusters.
In cluster 3 (Fig. 19), containing three regions (Auvergne-Rhône-Alpes, Bourgogne-Franche-Comté and Grand-Est), similar daily stratification as in cluster 1 and 2 is occurring (at a distance of 0.3). However, the time of the day is now more important than the regions. At first, 3 clusters are formed, mostly defined by time of the day (at a distance of 0.45), then those are split into regions. On the left, we can see the mornings of Bourgogne-Franche-Comté and Grand-Est show more similarity, as those are not split until a ~2.4

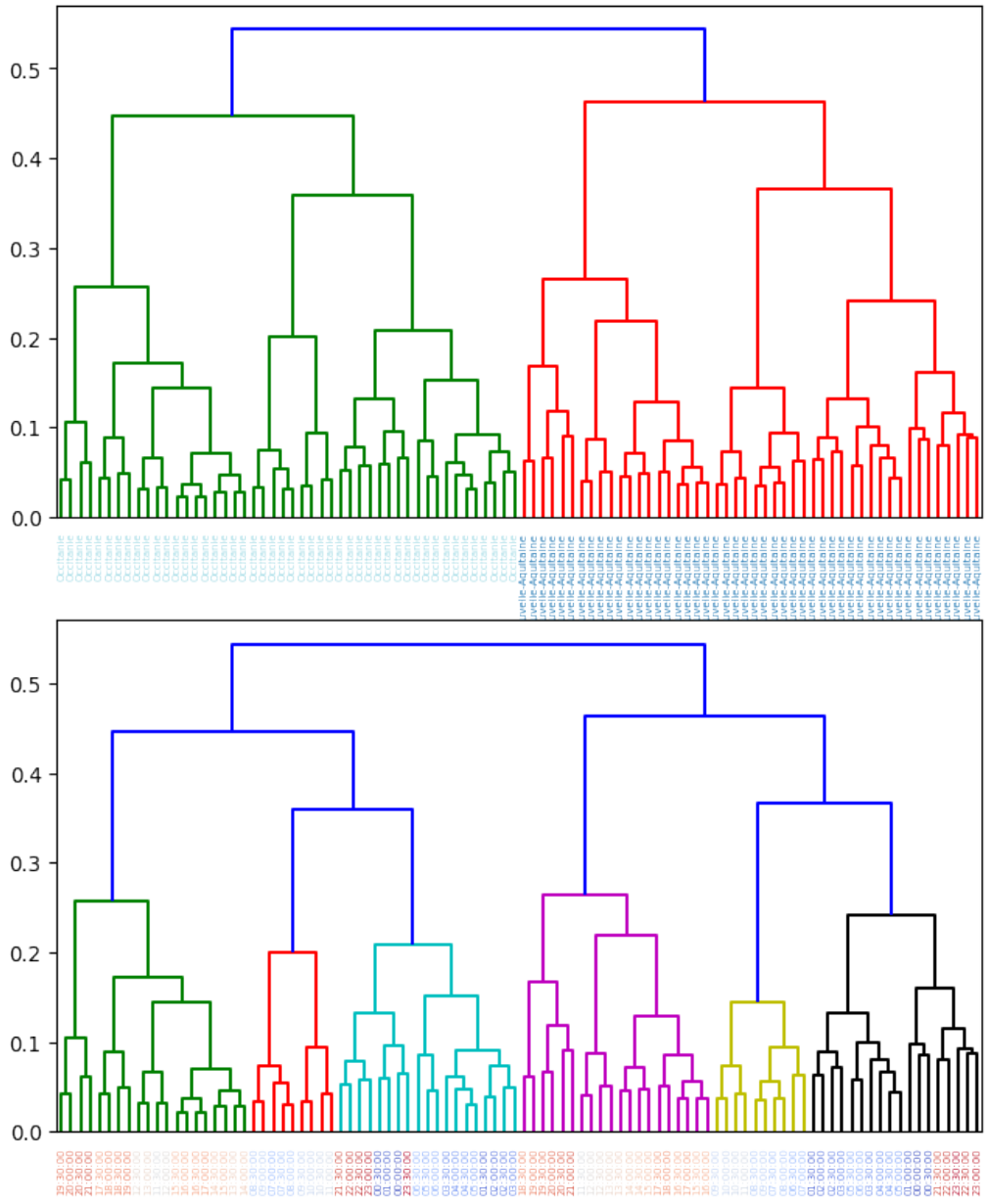

Figure 18: Dendrogram of cluster 2. The labels of the top plot are coloured by region and the bottom plot by time of the day. Red is early morning (00:00 and later) and blue is late at night (until 23:30), lighter colours are towards midday.

distance.
Cluster 4 (Fig. 20) is special, as it is a specifically a night cluster. For this reason, the only further grouping is done by region. On the left, Pays-de-la-Loire and Bretagne form a cluster, and in the middle, Ile-de-France and Normandie form another. The centre-Val de Loire is notably different from the other four regions.
In cluster 5 (Fig. 21), there are five regions, the same ones as in cluster 4 as well as Hauts-de-France. Similarly to cluster 4, as it is a day cluster, most of the delimitation occurs because of the regions. Then each regions is split between the earlier and later times. A clear example, is the Hauts-de-France that was dissociated from the other regions and feature the early and later times clustering.

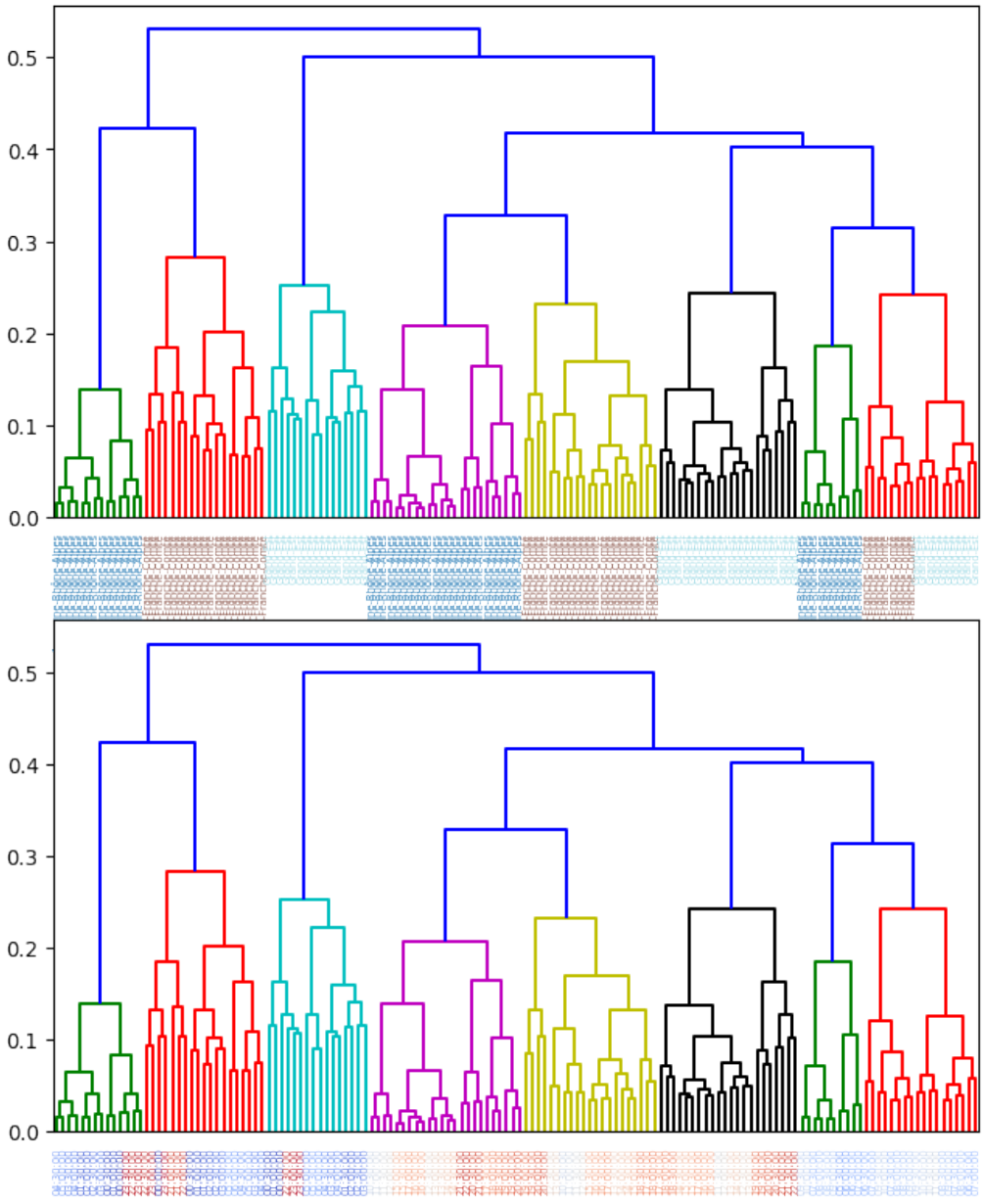

Figure 19: Dendrogram of cluster 3. The labels of the top plot are coloured by region and the bottom plot by time of the day. Red is early morning (00:00 and later) and blue is late at night (until 23:30), lighter colours are towards midday.

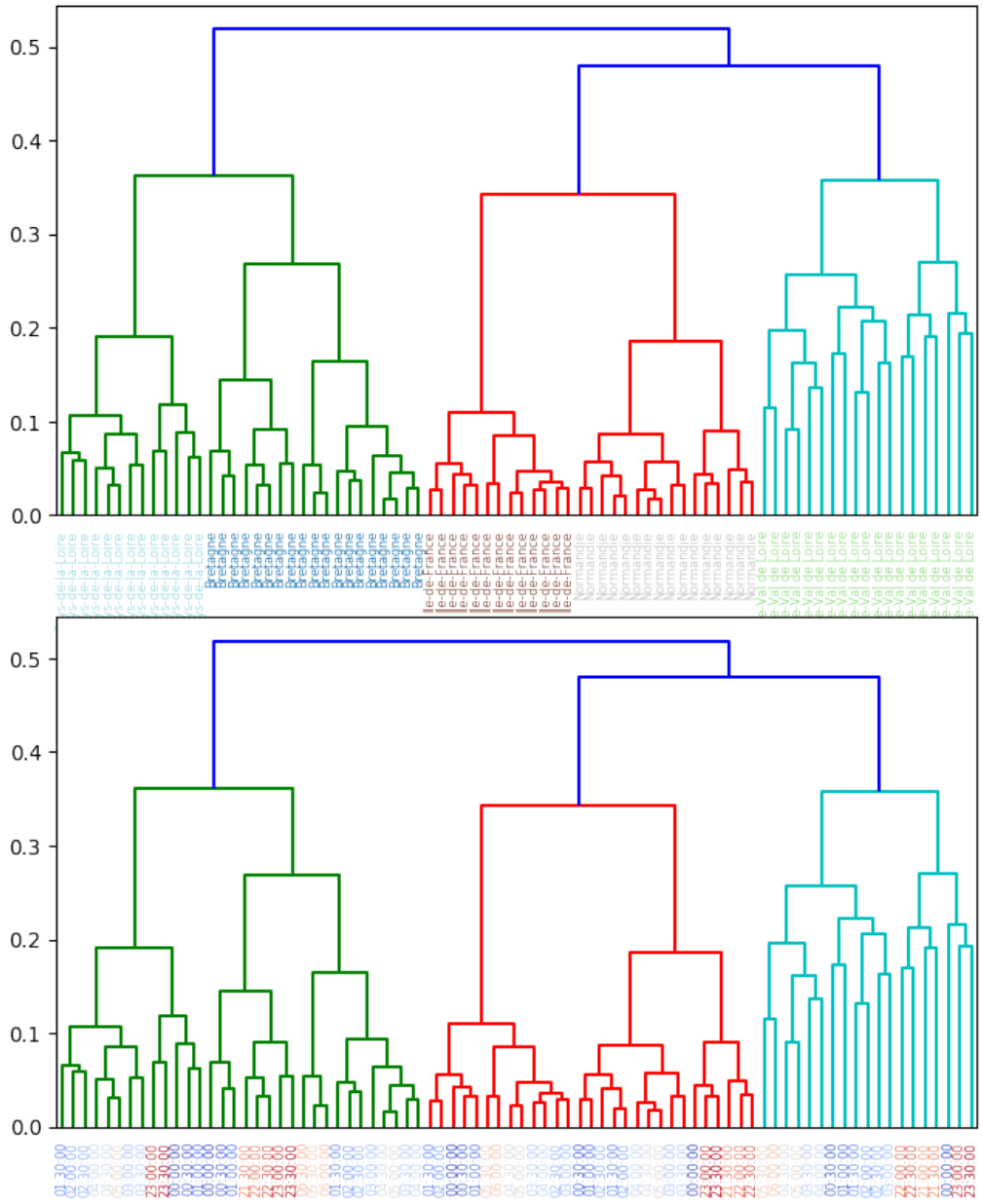

Figure 20: Dendrogram of cluster 4. The labels of the top plot are coloured by region and the bottom plot by time of the day. Red is early morning (00:00 and later) and blue is late at night (until 23:30), lighter colours are towards midday.

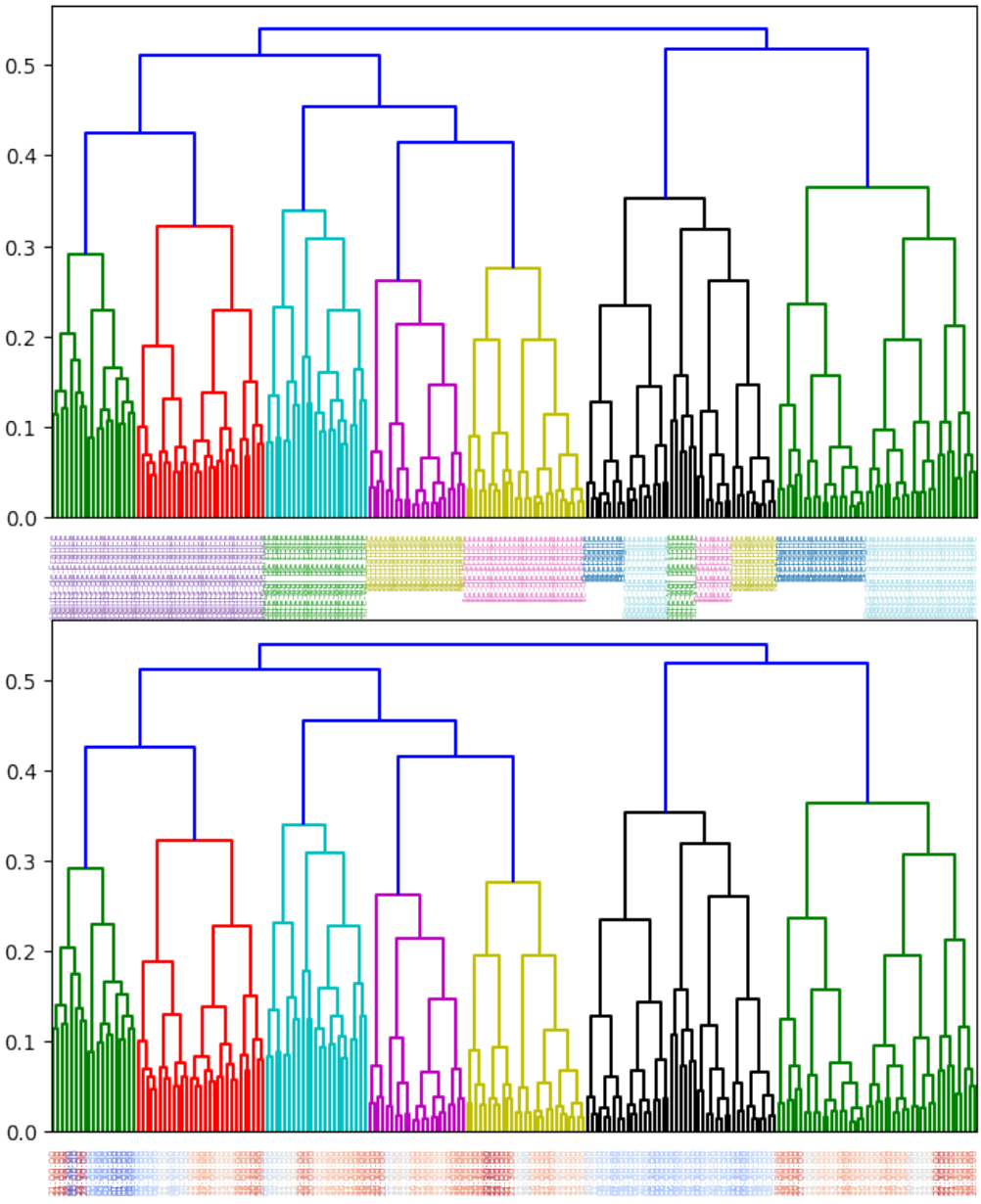

Figure 21: Dendrogram of cluster 5. The labels of the top plot are coloured by region and the bottom plot by time of the day. Red is early morning (00:00 and later) and blue is late at night (until 23:30), lighter colours are towards midday.

3.2.3 Clusters patterns

As no information about the size of the population in each region was used, the absolute consumptions were not compared between clusters. However, we can still compare relative changes over the years (Fig. 22), seasons (Fig. 23) and a typical day (Fig. 25).
On Fig. 22, The one year moving average trend estimate of each cluster seem to suggest that the regions that had lower consumptions in 2013-2014 have increased their consumptions in 2016-2017, and inversely for regions that had it higher in the 2013-2014 period. The PACA region (cluster 1) is also clearly differentiated from the other ones. Overall, it is difficult to draw conclusions over the longer term trends, as there are not enough data to analyse those. The general pattern observed during those 4.5 years are most likely due to weather conditions across each year.

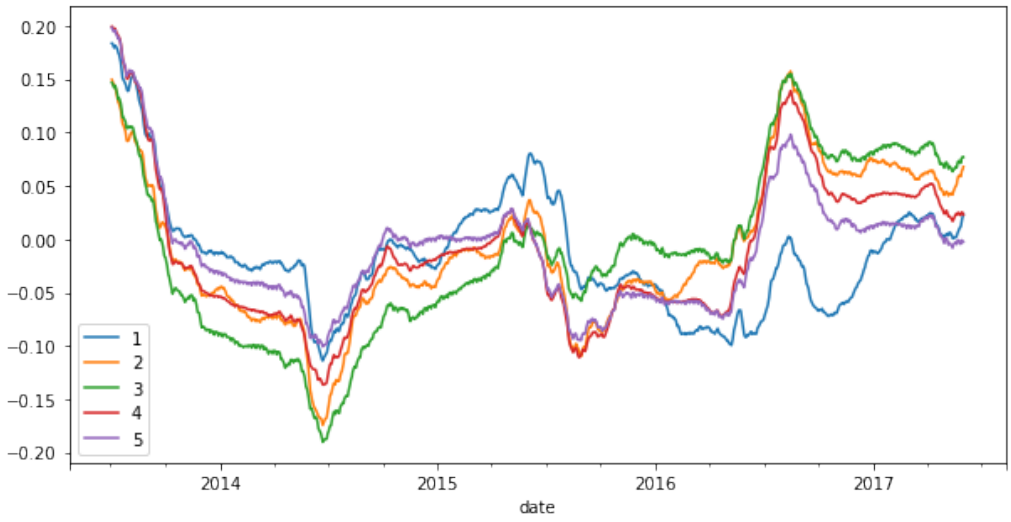

Figure 22: One year centered moving average trend of each cluster.

In the three months moving average trend (Fig. 23), we can see that cluster 1 and 2 have a higher energy consumption during the summer. This is most likely due to the use of air conditioning, as those two clusters are in the south of France, which is not really common (nor necessary) in the north. In general, the energy consumption is significantly higher during the winter.
Cluster 3 and 5 have the highest energy consumption during the week (24) and the lowest during the weekend. It is not surprising for cluster 5 is a day cluster. However, cluster 3 has a notably different consumption that the other two normal cluster (1 and 2). On the opposite, cluster 4, which is a night cluster, has higher consumption during the weekend than during the week.
Over the day (Fig. 25), cluster 1, and to a smaller extend cluster 2, tend to use electricity later than the other regions. Again, this is most likely due to the different life style between the north and south regions of France. As it is very warm during the days, people tend to go out more in the evenings, as shown by the higher consumption around 20:00.

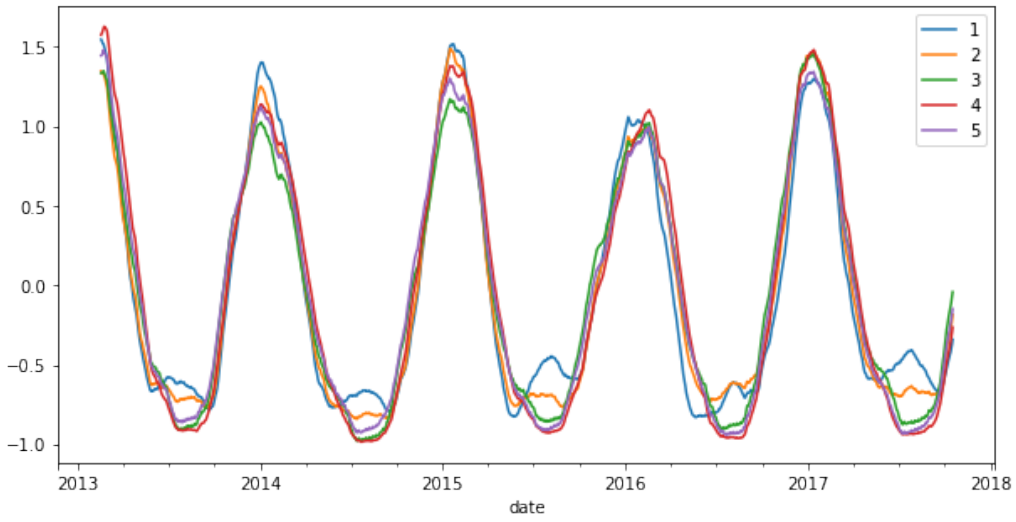

Figure 23: Three months centered moving average trend of each cluster.

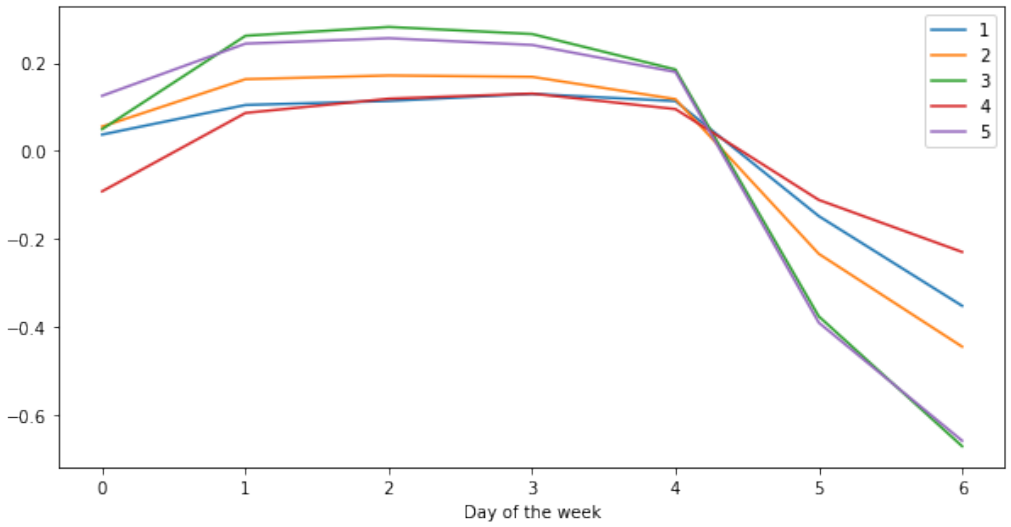

Figure 24: Mean energy consumption of each cluster across the days of the week.

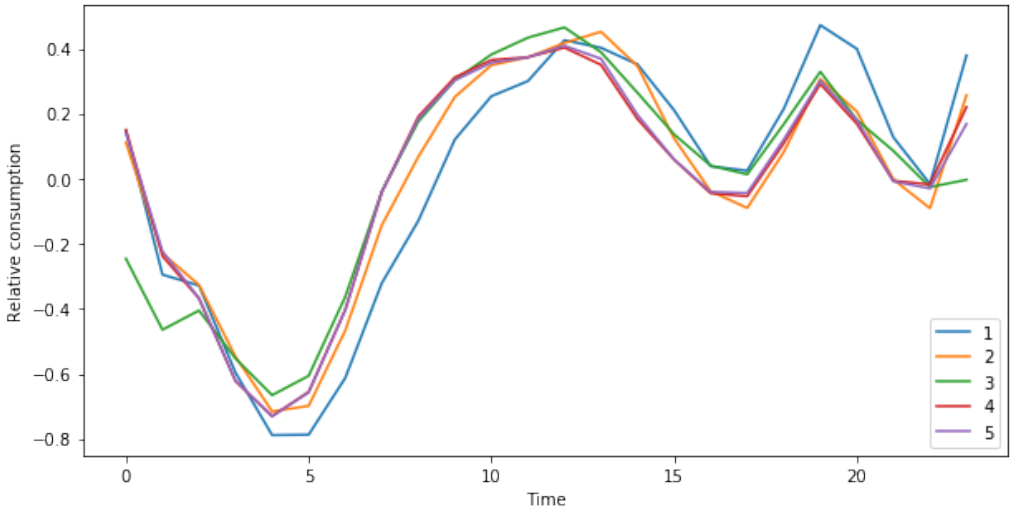

Figure 25: Hourly mean consumption of everyday for each cluster.

3.3 Comparison with Euclidean distance

The silhouette statistic seem to suggest a higher number of clusters when using the Euclidean distance (Fig. 26), with 59 clusters. As before, this study will only focus on the main clusters as shown in Fig. 27. There, the main observable cluster are of the nights, on the left, (00:00 to 9:30) and days, on the right (5:00 to 23:30). The time clustering is also shown in 28. It is only after this clustering that it is possible to look at regional clusters. Although that differentiation is possible, it occurs low in the tree.
In Fig. 29, we can see that the morning cluster consumes constantly more during the winter than during the day. This most likely due to keeping the household warm. And inversely for the summer, as the air conditioning is mostly used during the day.
In Fig. 30, the nights have a higher consumption during the weekend than during the week. This was also observed with the GCC distance (cluster 4 and 5 in Fig. 25).

4 Conclusion

The application of the GCC to the electricity consumption of the French regions was successful, as a high degree of meaningful clustering was detected. It was possible to group the 576 time series (12 regions $\times$ 48 half hours) into five clear clusters. But also detect further clustering possibilities within each of those. Some clusters were dominated by different consumption in the morning, afternoon and night. Some other had further geographical delimitation, or were a mix of both.
The amount of data was too small to detect any clear long term trends, but enough to detect periods of lower or higher consumption. On the other hand, the use of air conditioning in the summer was detected for the two clusters in the south of France. Furthermore, those two southern clusters also exhibited later lifestyle with a much higher consumption around 20:00. Most likely due to the weather being too hot during the days and people enjoying going out in the evenings. Furthermore, nights clusters tend to consume more electricity during the weekend than during the week.

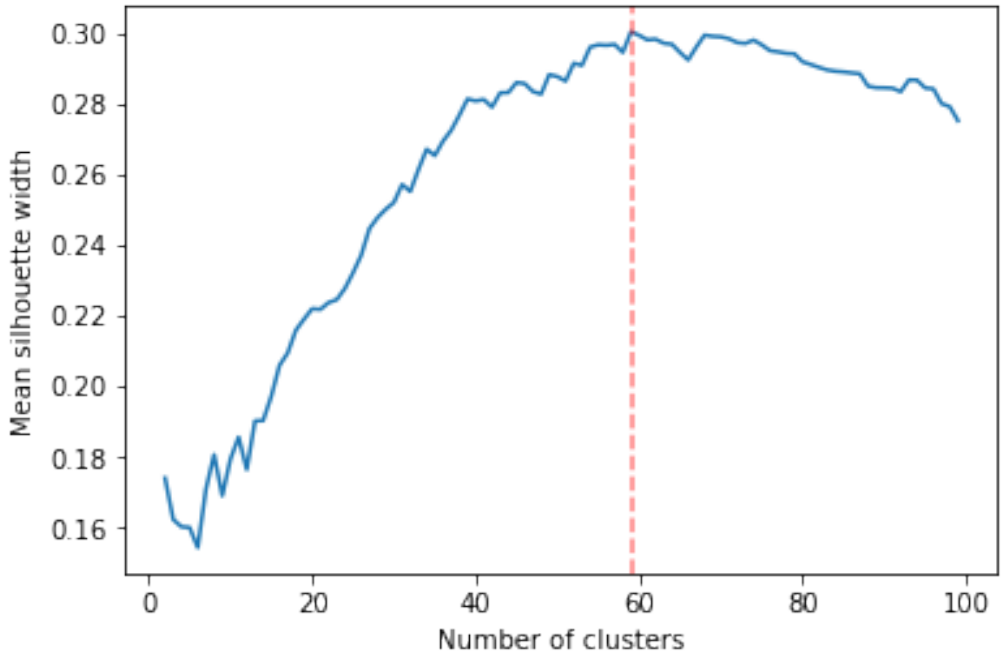

Figure 26: Mean silhouette width of each cluster number.

The GCC, as defined by Alonso and Peña (2017), was successfully able to detect cross dependencies between the series to a very fine detail. Although it can be computationally expensive and slow for large series datasets, it has allowed to detect geographical clusters that were more difficulty detected by the Euclidean distance. It is therefore a very competent metric that can greatly help for any exploratory analysis.
Both Python and R now provide some very mature environment for the manipulation of time series. The numpy (Walt et al., 2011), pandas (McKinney et al., 2010) and scipy (Jones et al., 2014) packages available in Python provide with a very easy, efficient and consistent set of tools, but can sometimes lack the automatic statistical reports provided by most functions in R Team et al. (2013). As such R has now many tried and tested packages that offer a very high level of details in the result with a very minimum amount of code. All in all, python seemed faster and easier for the manipulation of the data, but R provided with a much stronger level of details for statistical analysis and clustering tools (Kassambara and Mundt, 2016).

References

Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. (2015). Time-series clustering–a decade review. *Information Systems*, 53:16–38.

Alonso, A. M. and Peña, D. (2017). Clustering time series by dependency. Preprint.

Bezdek, J. C. (1981). Objective function clustering. In *Pattern recognition with fuzzy objective function algorithms*, pages 43–93. Springer.

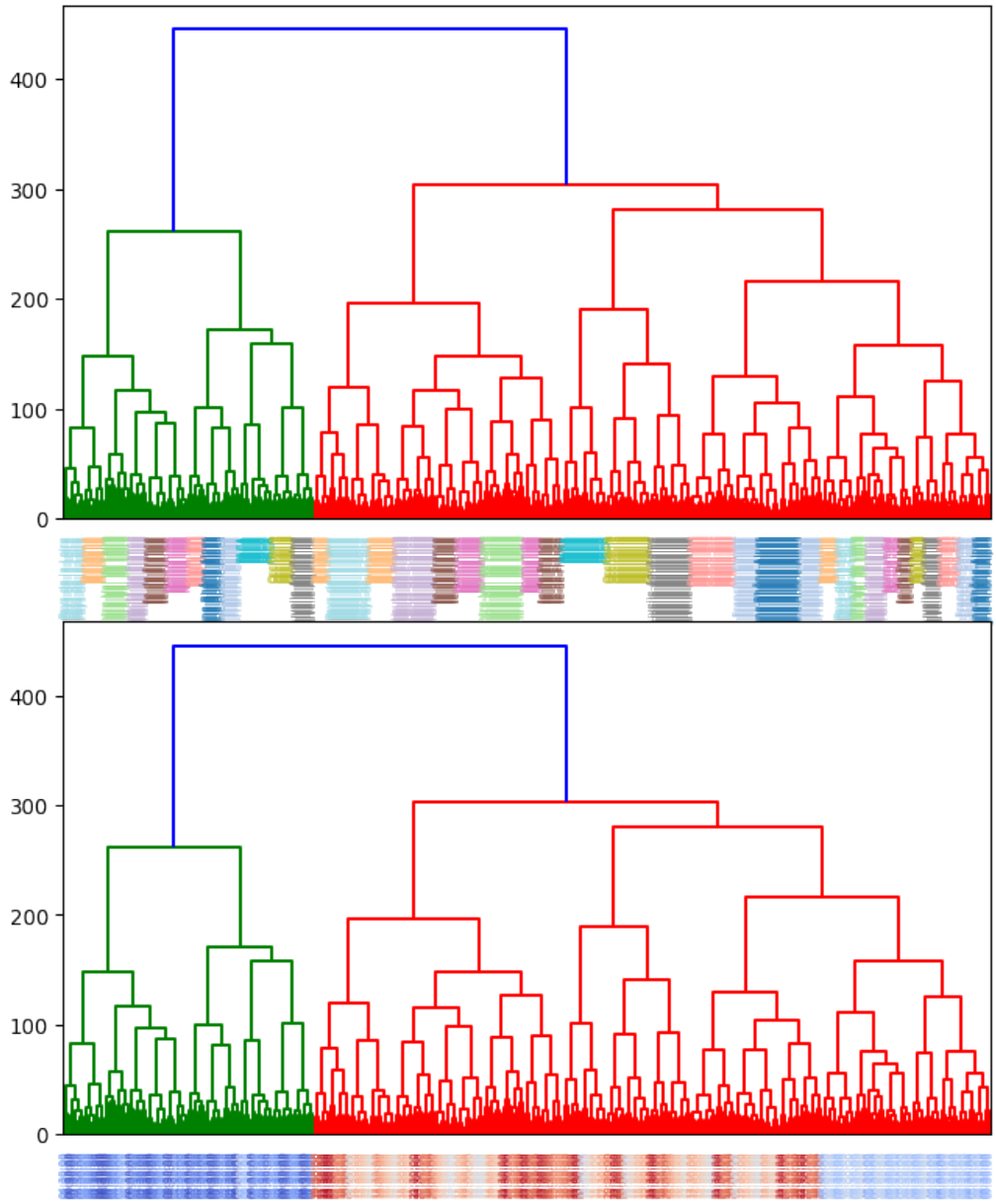

Figure 27: Dendrogram of clusters computed using the Euclidean distance. The labels of the top plot are coloured by region and the bottom plot by time of the day. Red is early morning (00:00 and later) and blue is late at night (until 23:30), lighter colours are towards midday.

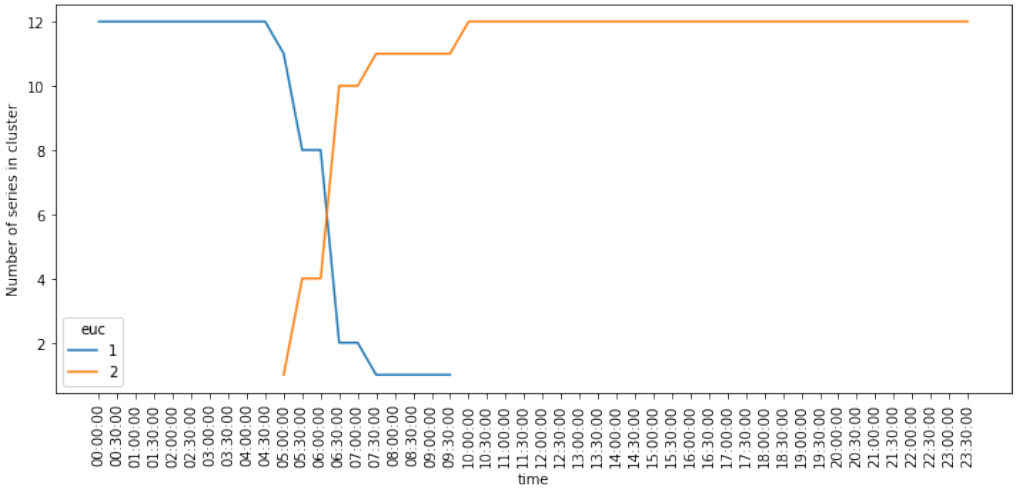

Figure 28: Cluster composition with time of the day

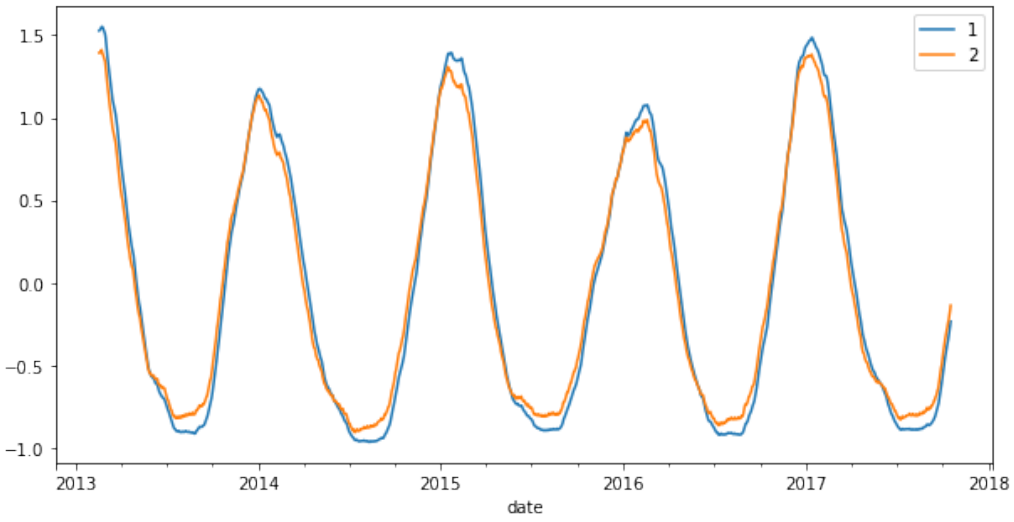

Figure 29: Three months centered moving average trend of each cluster.

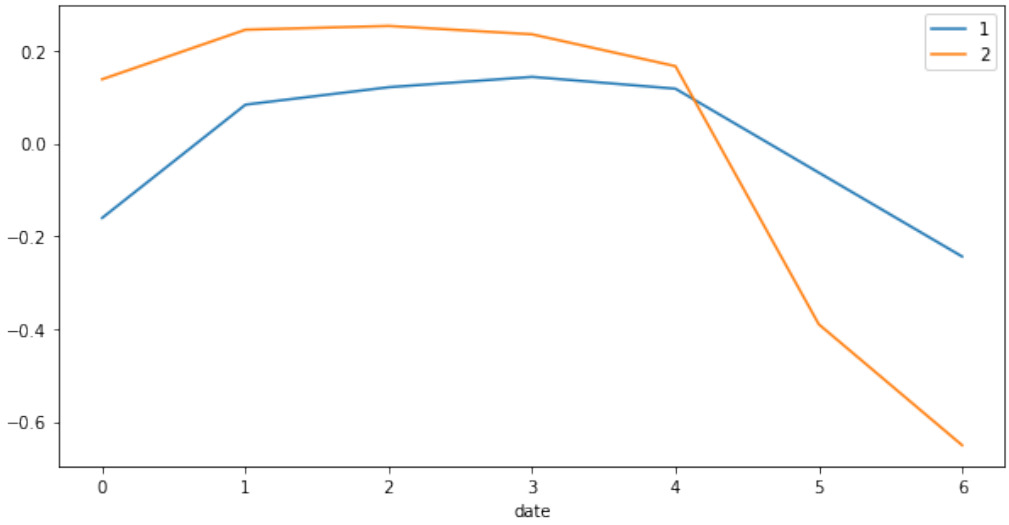

Figure 30: Mean energy consumption of each cluster across the days of the week.

Carpenter, G. A. and Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D. (1988). Autoclass: A Bayesian classification system. In *Machine Learning Proceedings 1988*, pages 54–64. Elsevier.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Guha, S., Rastogi, R., and Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. In *ACM Sigmod Record*, volume 27, pages 73–84. ACM.

Han, J., Kamber, M., and Pei, J. (2000). Data mining: concepts and techniques (the Morgan Kaufmann series in data management systems). *Morgan Kaufmann*.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31:264–323.

Jones, E., Oliphant, T., and Peterson, P. (2014). {SciPy}: open source scientific tools for {Python}.

Karypis, G., Han, E.-H., and Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32:68–75.

Kassambara, A. and Mundt, F. (2016). Factoextra: extract and visualize the results of multivariate data analyses. *R package version*, 1.

Kohonen, T. (1998). The self-organizing map. *Neurocomputing*, 21:1–6.

Krishnapuram, R., Joshi, A., Nasraoui, O., and Yi, L. (2001). Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE Transactions on Fuzzy Systems*, 9:595–607.

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern Recognition*, 38:1857–1874.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

McKinney, W. et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.

Rousseeuw, P. J. and Kaufman, L. (1990). *Finding groups in data*. Wiley Online Library Hoboken.

Team, R. C. et al. (2013). R: A language and environment for statistical computing.

Van Wijk, J. J. and Van Selow, E. R. (1999). Cluster and calendar based visualization of time series data. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 4–9. IEEE.

Walt, S. v. d., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13:22–30.

Wang, W., Yang, J., Muntz, R., et al. (1997). Sting: A statistical information grid approach to spatial data mining. In *VLDB*, volume 97, pages 186–195.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, volume 25, pages 103–114. ACM.

A Code for selection of k

- In R:

```r
library(FitAR)

getOrder <- function(ts, order.max=40) {
  SelectModel(ts, ARModel = 'AR', Criterion = 'BIC', lag.max = order.max)[1,1]
}

k <- max(apply(consommation, 2, getOrder))
```

- In Python:

```python
import statsmodels.api as sm

k = consommation.apply(
    lambda x: sm.tsa.arma_order_select_ic(
        x, ic='bic', trend='nc', max_ar=40, max_ma=1)['bic_min_order'][0]).max()
```

B Code for GCC computation

- In R:

```
kMatrix <- function(ts, k) {
  m <- ts[1 : (length(ts) - k)]
  for (i in seq(k)) {
    m <- cbind(m, ts[(i+1) : (length(ts) - k + i)])
  }
  m
}

GCC <- function(ts1, ts2, k) {
  Xi <-  kMatrix(ts1, k)
  Xj <-  kMatrix(ts2, k)

  Xij <- cbind(Xi, Xj)

  det(cor(Xij))^(1/(k+1)) /
    (det(cor(Xi))^(1/(k+1)) * det(cor(Xj))^(1/(k+1)))
}
k<-37
combinations <- combn(dim(consommation)[2], 2)
DM_GCC <- matrix(0, dim(consommation)[2], dim(consommation)[2])
for (d in seq(dim(combinations)[2])) {
  distance <- GCC(consommation[, combinations[,d][1]],
                  consommation[, combinations[,d][2]], k)
  DM_GCC[combinations[,d][1], combinations[,d][2]] <- distance
  DM_GCC[combinations[,d][2], combinations[,d][1]] <- distance
}
rownames(DM_GCC) <- colnames(consommation)
colnames(DM_GCC) <- colnames(consommation)
write.csv(DM_GCC, file="data/DM_GCC_37_R.csv")
```

- In Python:

```
import numpy as np
from scipy.spatial.distance import pdist
from scipy.spatial.distance import squareform
import pickle


def k_matrix(ts, k):
    T = ts.shape[0]
    return np.array(
        [ts[(shift):T - k + shift] for shift in np.arange(0, k + 1)])
```

```
def get_GCC(ts1, ts2):
    k = 37
    Xi = k_matrix(ts1, k)
    Xj = k_matrix(ts2, k)
    Xij = np.concatenate((Xi, Xj))
    GCC = np.linalg.det(np.corrcoef(Xij)) ** (1 / (k + 1)) / (
        np.linalg.det(np.corrcoef(Xi)) ** (1 / (k + 1)) \
        * np.linalg.det(np.corrcoef(Xj)) ** (1 / (k + 1)) )
    return GCC


pdist_gcc = pdist(consommation.values.T, get_GCC)
DM_GCC = squareform(pdist_gcc)
DM_GCC = pd.DataFrame(
    DM_GCC, index=consommation.columns, columns=consommation.columns)
DM_GCC.to_csv('data/DM_GCC_37.csv')
```