uc3m | Universidad Carlos III de Madrid

Master in Big Data Analytics
2017/2018

Master's Thesis

France Regional Electricity Consumption Clustering Using Generalised Cross Correlation.

Pierre Mercatoris

Supervisors
Andrés Modesto Alonso Fernández
Daniel Peña Sánchez de Rivera
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

1.1 Clustering time series

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.

1.1.1 Static data clustering

As defined by Han et al. (2000), clustering can be generalised 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)).

1.1.2 Dynamic Data or 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 an exploratory analysis.

As defined by Liao (2005) in his review (see fig. 1), there are 3 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.

Aghabozorgi et al. (2015) has identified the 4 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).

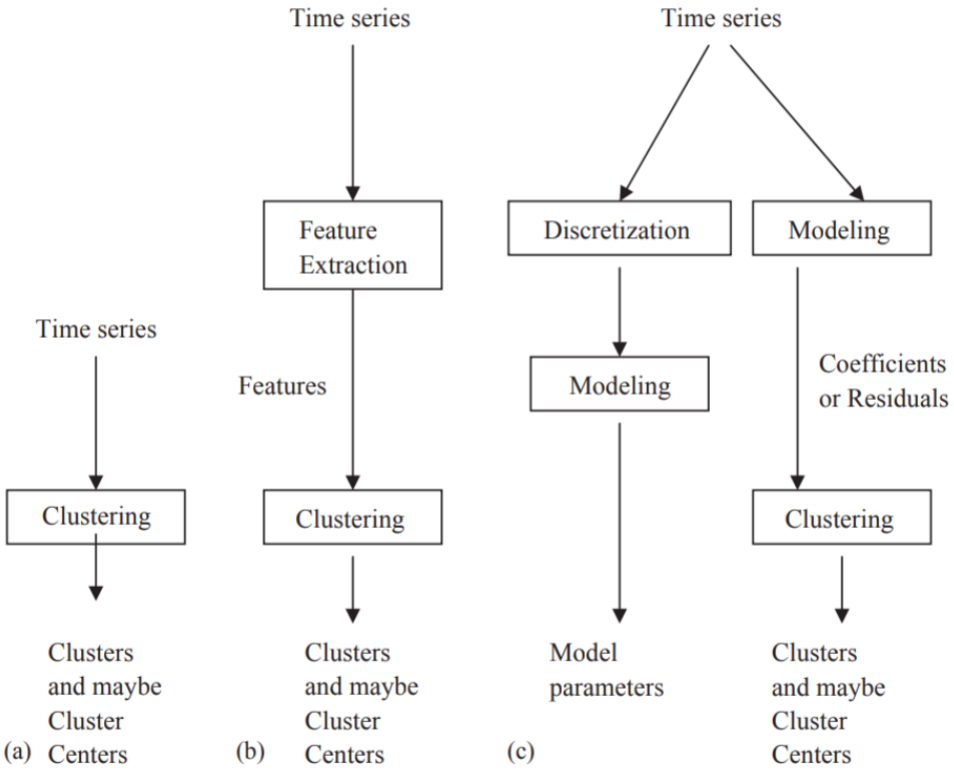

Figure 1: Three time series clustering approaches: (a) raw-data-based, (b) feature-based, (c) model-based. (Liao, 2005)

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

1.2 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. His approach was to use agglomerative hierarchical clustering on daily power consumption based on the root mean square distance.

In this thesis, I would like to show an application of the Generalised Cross Correlation defined by M. 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. 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. However, as demonstrated by M. Alonso and Peña (2017), clustering by their cross dependency can produce significantly different results compared to those other methods.

The goal of this study is to implement that metric in R and in Python (for comparison) and show its efficacy in finding clusters of electricity consumption in France that make sense. A simple agglomerative hierarchical clustering was 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 trends and patterns.

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
Grand-Est	2016-01-01	00:00	5130
Grand-Est	2016-01-01	00:15	
Grand-Est	2016-01-01	00:30	5130
Grand-Est	2016-01-01	00:45	
Grand-Est	2016-01-01	01:00	5014
.....			

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).

The pivot table was used again so that each time of the day is a columns, and each row is a daily value for a certain time and region, the resulting table has 576 columns (48 x 12 regions) and 1794 rows/days.(table 3).

[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	

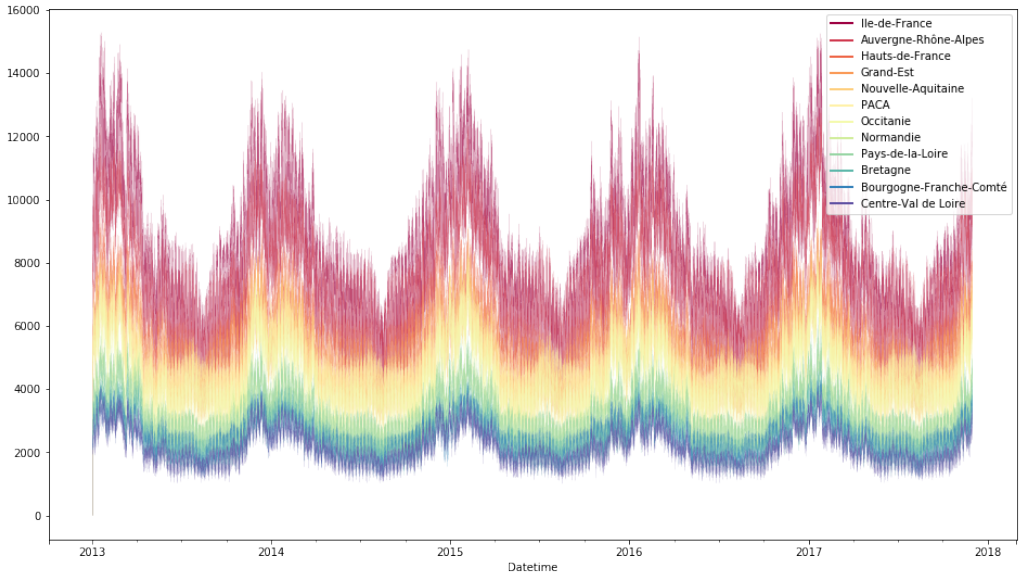

Figure 2: Mean electricity consumption of each of the french regions from 2013 to end 2017.

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

In figure 3, we can already see that consumption midday is much higher than at night, with more spread in the summmer than in the winter.

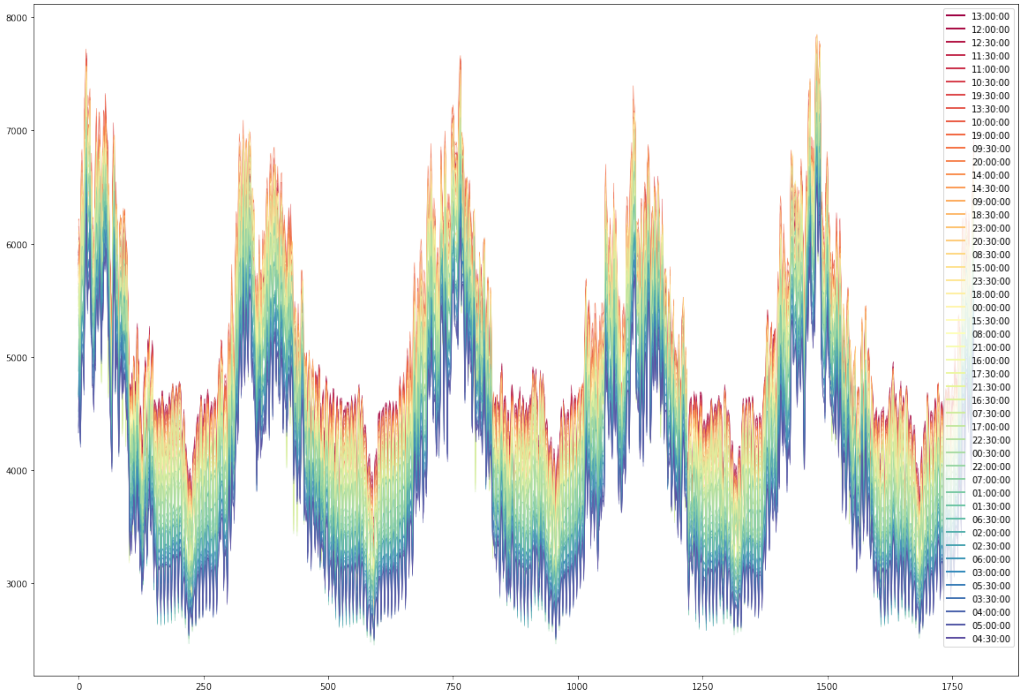

Figure 3: Mean electricity consumption for all the regions of France at different times.

2.2.2 Transformation

1. Stationarity

   The original series feature a strong seasonality as show in figure 4.

   To try and remove it, the weekly difference was taken (difference between all the values separated by 7 days). This was able to remove most of the seasonality (fig. 5).

   So as to get as close to stationarity as possible without loosing too much data, another difference was taken, but this time only 1 day difference. Now, 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 mean of 0 and standard deviation of 1, the z-score was applied to each individual series (1).

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

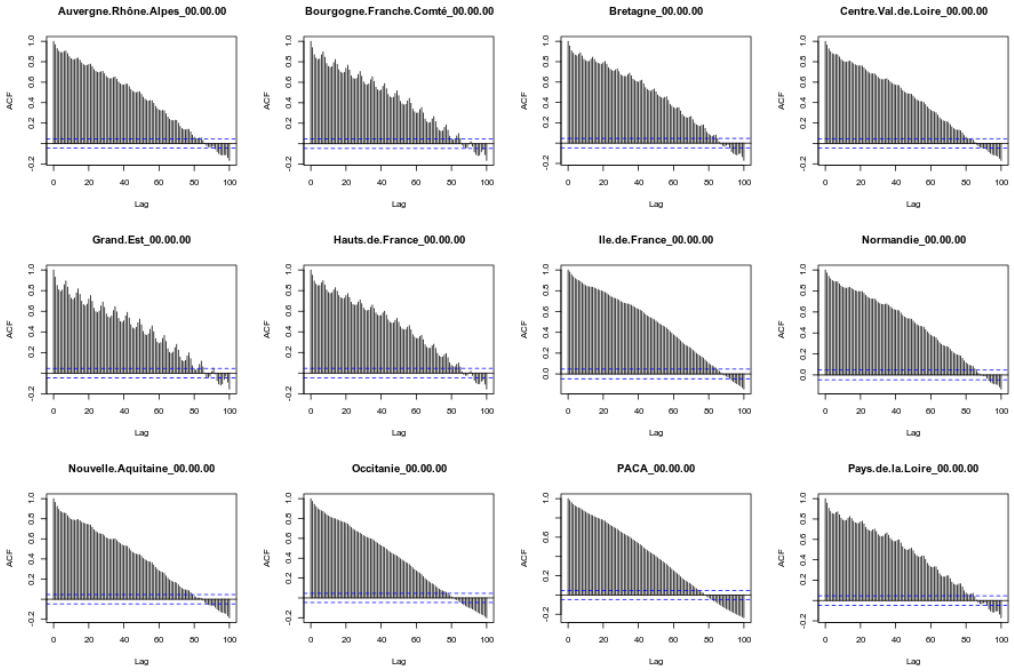

Figure 4: Autocorrelation function of the original data.

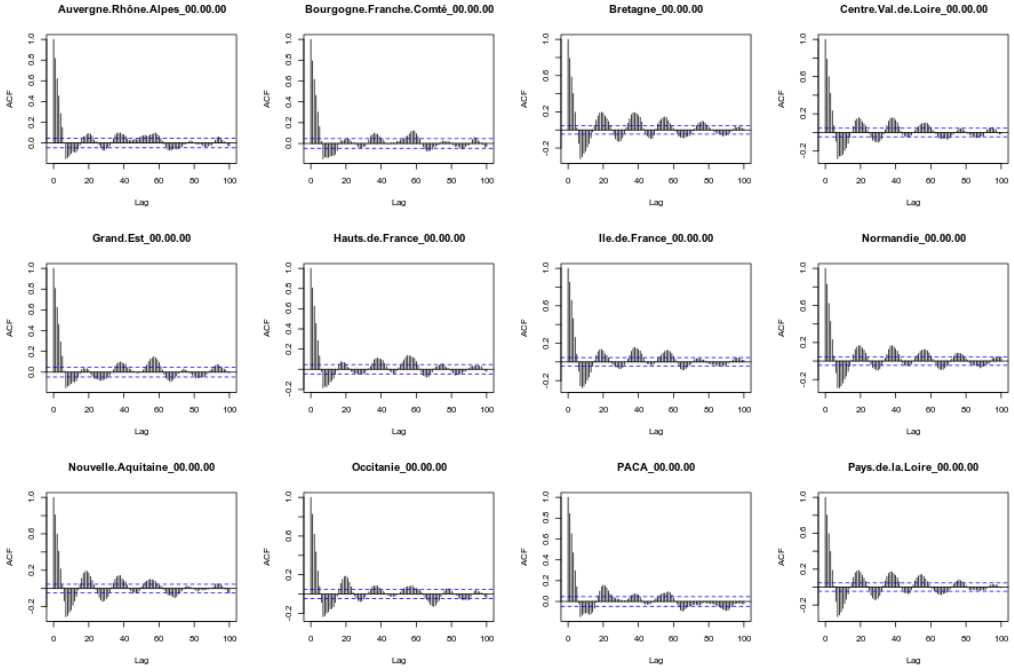

Figure 5: Autocorrelation function of the weekly differentiated series.

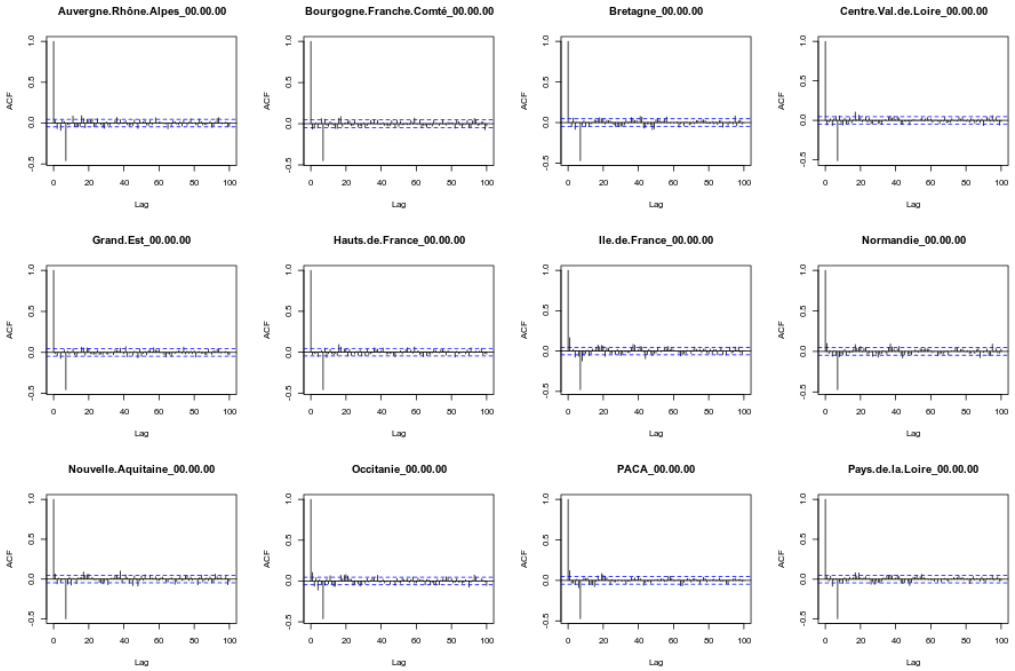

Figure 6: Autocorrelation function of the weekly differenciated series + another difference.

2.3 GCC description

As described before, the GCC is a very 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 each lag k, it is necessary to construct the $X(i)$ and $X(j)$ matrices from the i and j series of size N as follow:

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

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

$$X(i,j) = (X(i), X(j)) \tag{3}$$

The GCC is can then be computed as:

$$\widehat{GCC}(X_i, X_i) = 1 - \frac{|\widehat{R}_{X(i,j)}|^{k+1}}{|\widehat{R}_{X(i)}|^{k+1}|\widehat{R}_{X(j)}|^{k+1}} \tag{4}$$

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 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:

$$DM_{\widehat{GCC}} = \begin{pmatrix} 0 & 1-\widehat{GCC}(X_1,X_2) & \dots & 1-\widehat{GCC}(X_1,X_N) \\ 1-\widehat{GCC}(X_2,X_1) & 0 & \dots & 1-\widehat{GCC}(X_2,X_N) \\ \vdots & \vdots & \ddots & \vdots \\ 1-\widehat{GCC}(X_N,X_1) & 1-\widehat{GCC}(X_N,X_2) & \dots & 0 \end{pmatrix} \tag{5}$$

It is necessary to do $1-\widehat{GCC}(X_i,X_j)$ if the original measure was calculated as in equation 4, which is a similarity metric, and what is needed here is a distance where 0 means series close to each other and 1 means far apart.

There are 2 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 M. 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. 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.

   - In 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))
print(k)
```

```
[1] 37
```

   - In 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()
k
```

37

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

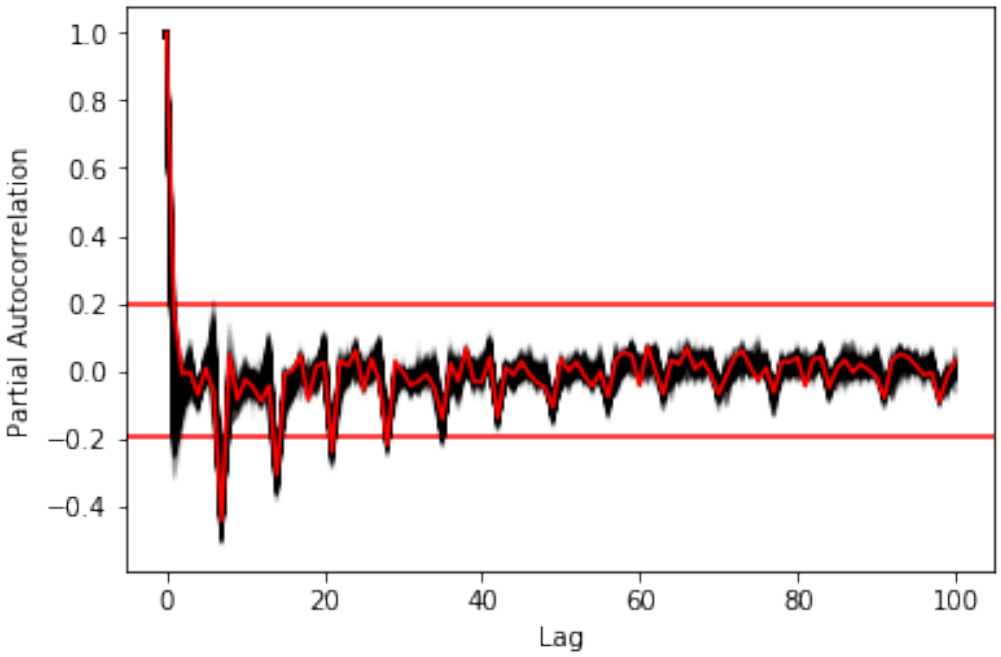

Figure 7: Partial autocorrelation of the stationary scaled data.

2. Distance matrix

   The GCC was computed in both R and in Python to validate the results.

   - 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')
```

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

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, refered to as linkage method. The most popular methods were compared using the cophonetic 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 cophonetic correlation of the 'average' linkage.

Table 4: Cophonetic 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, making it easier to distinguish the groups. In fact, the regions appear the be the main influencing factor.

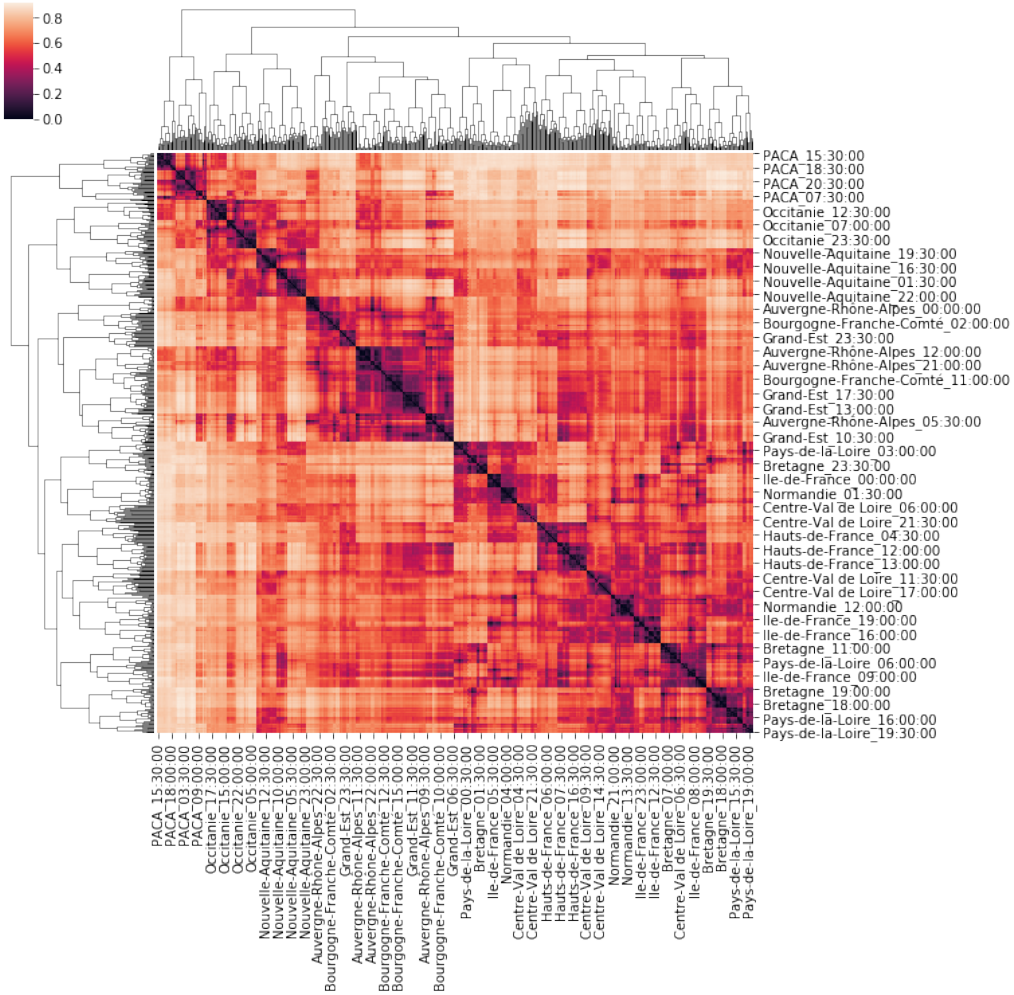

Figure 8: Heatmap of the distance matrix rearranged using the average linkage hierarchical clustering.

3.1.2 Cluster number

Determining the number of cluster can be very challenging. The *factoextra* package in R (Kassambara and Mundt, 2016) provides functions to intent finding that number automatically. However, as you can see in figure 9, it isn't always that obvious.

The larger silhouette width is observed at 2 clusters but there is a small peak at 5 clusters. We can also see that the more clusters the better the gap statistic. However, we can see a small peak at k=5. Looking at the sum of square distance, we can also notice a small "elbow" at k=5.

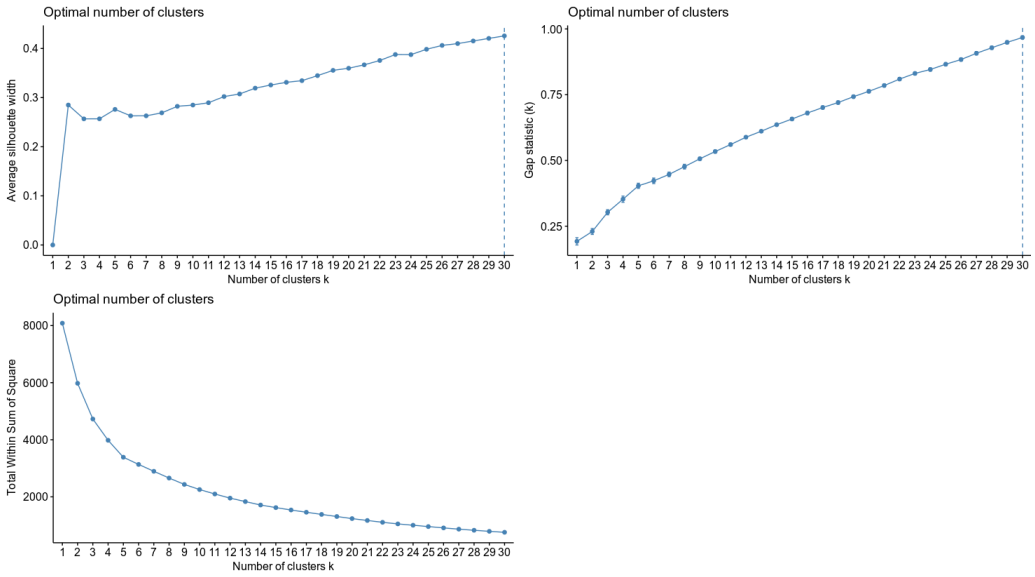

Figure 9: Mean silhouette width, gap statistic and total within cluster sum of square distance for each number of cluster.

This all suggest that there might be 5 clusters in our dataset, as shown on the dendrogram (fig. 10). Another way to look at those clusters is by looking the first 2 principal components of the distance matrix (fig. 11).

In fig. 12, we can see the silhouette width of each of the samples in their respective cluster. There seems to be some misclassification for some samples in cluster 3, but overall each cluster has significantly high silhouette width.

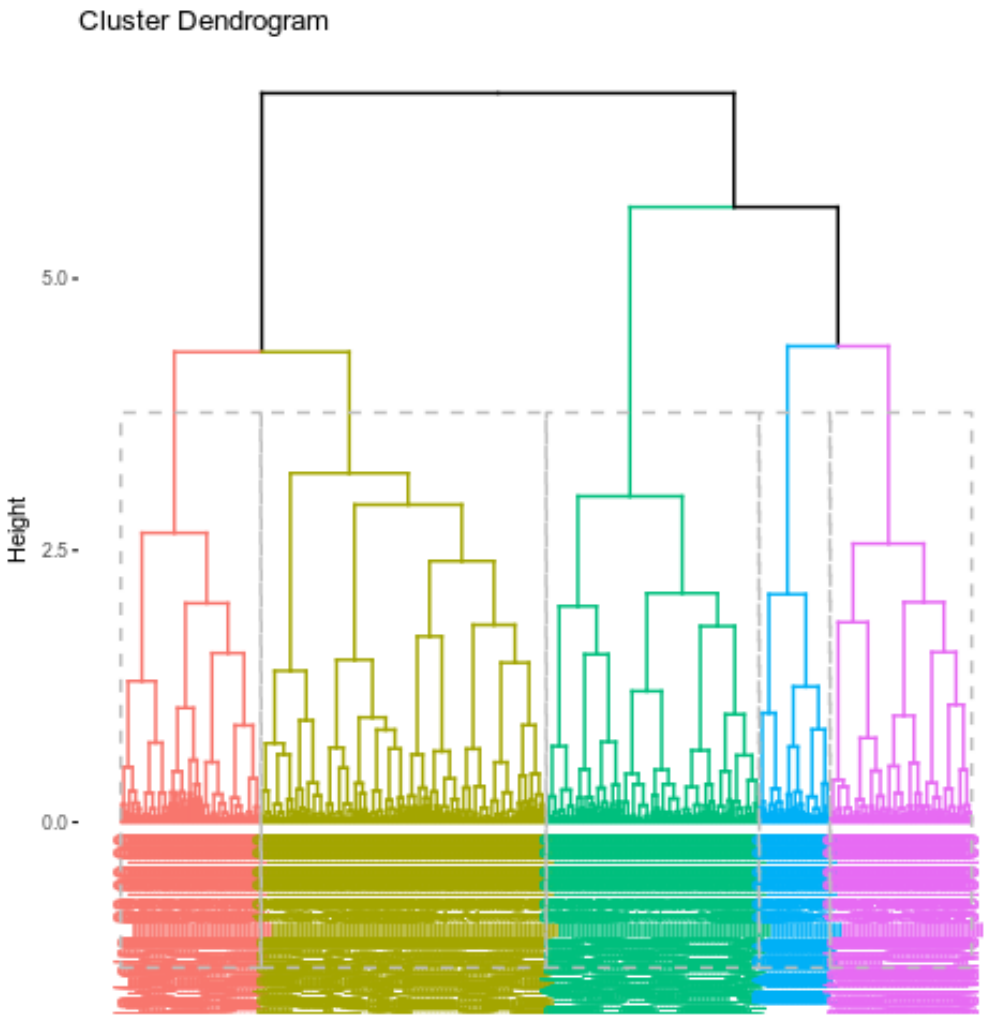

Figure 10: Dendrogram of the distance matrix using average linkage.

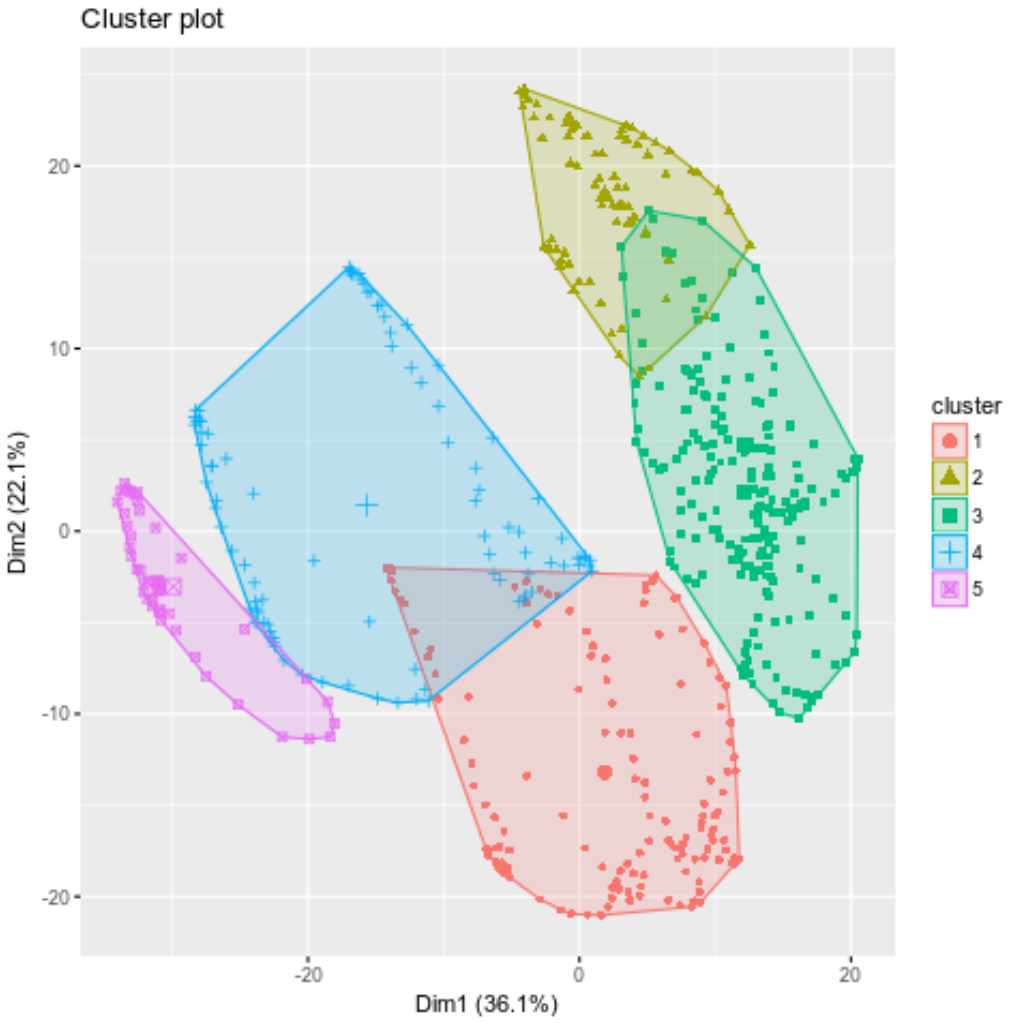

Figure 11: 5 clusters over the 2 principal components of the distance matrix.

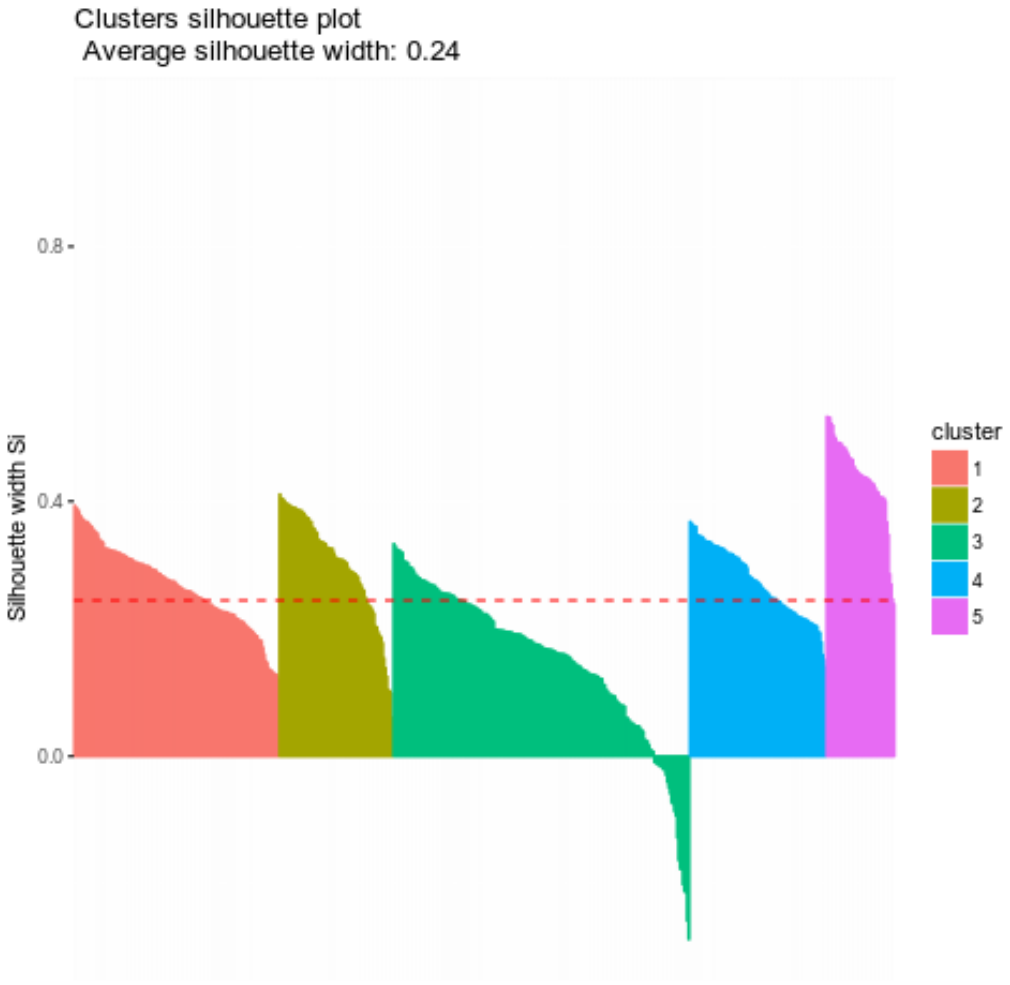

Figure 12: Silhouette width of the samples in each cluster.

3.2 Cluster analysis

3.2.1 Mapping the clusters

If we were to only use 2 clusters, the PACA region is clearly the most distinct of all the regions (fig. 13).

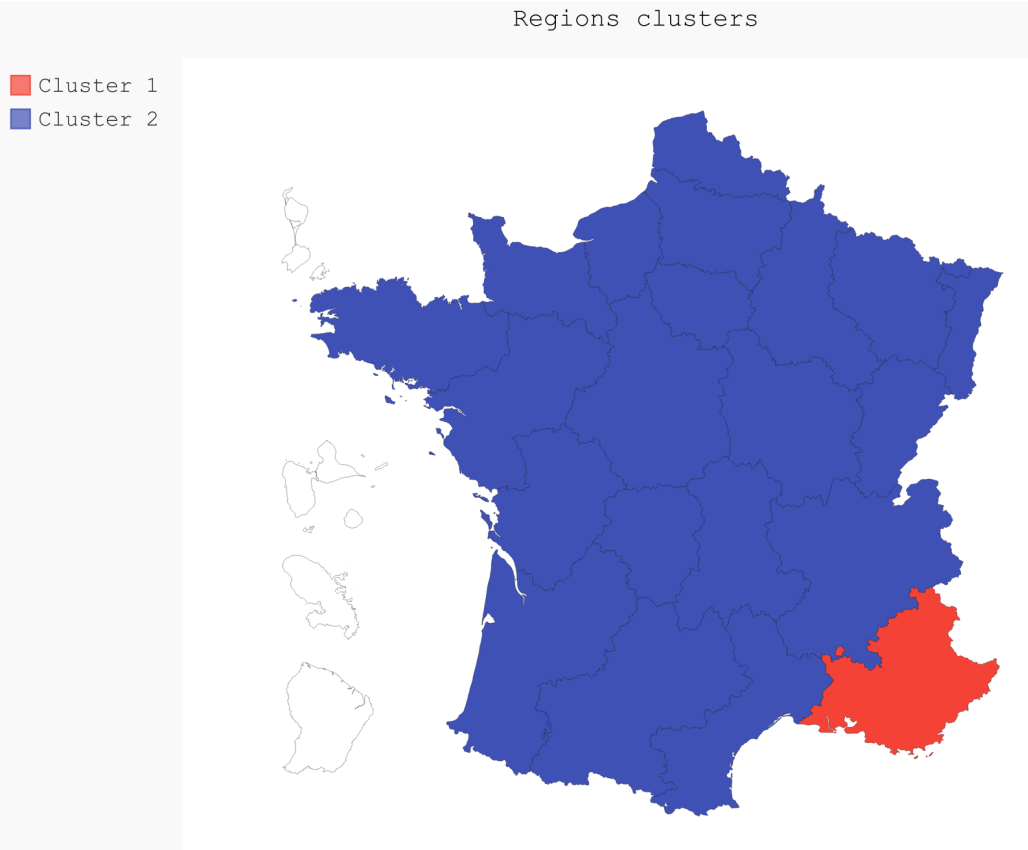

Figure 13: Map of the 2 clusters on the map of France. The regions shown are the old more numberous regions, but the boundaries of the 12 new reiongs are the same.

However, in order to have a deeper understanding of the composition of France, 5 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. 14), 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

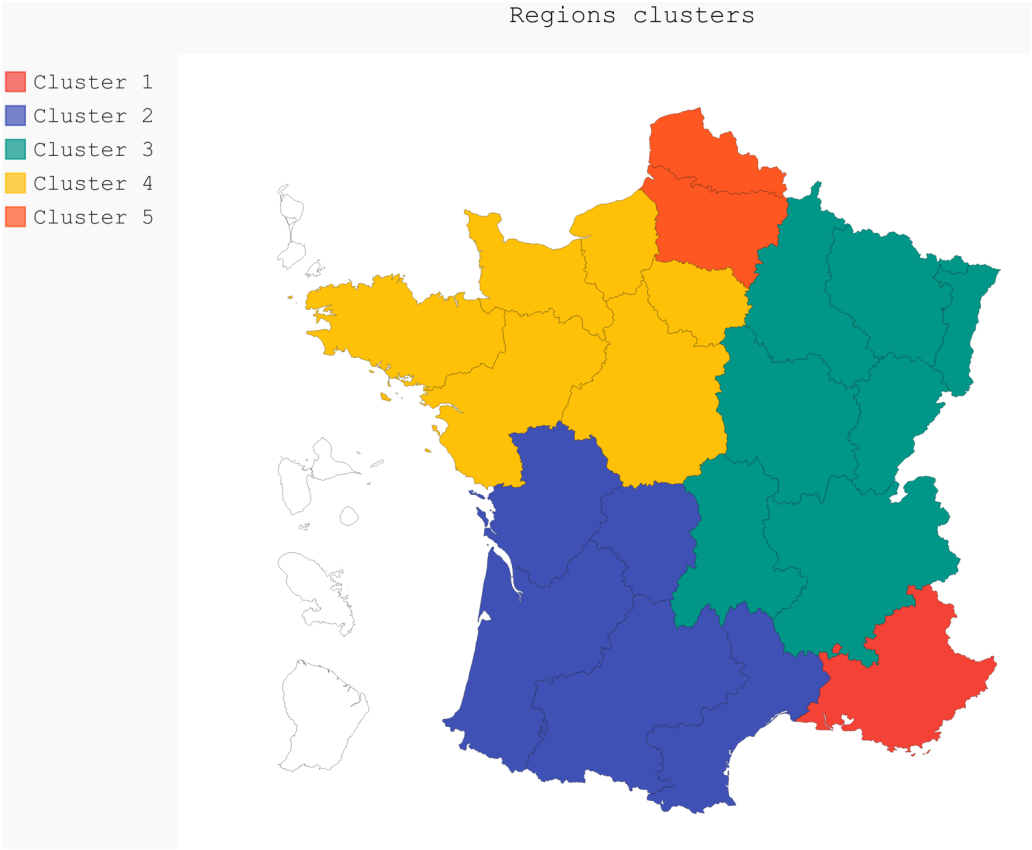

Figure 14: Map of the 5 clusters on the map of France. The regions shown are the old more numberous regions, but the boundaries of the 12 new reiongs are the same.

3.2.2 Within clusters structure

In this section, the goal was to find out if there was more structure within each of the clusters. A dendrogram was plotted for each cluster and the label was coloured depending on the time of the day, where black is late in the day and red is early morning. The lighter colours are towards midday.

Cluster 1 only contains the PACA region. In figure 15, we can see that there are 3 main clusters, mornings from 6:30 to 11:00, midday-afternoon from 11:30 to 20:00, and the night cluster from 20:00 to 6:00. Days (11:30 to 20:00) and nights (20:30 to 11:00) are however the most well defined.

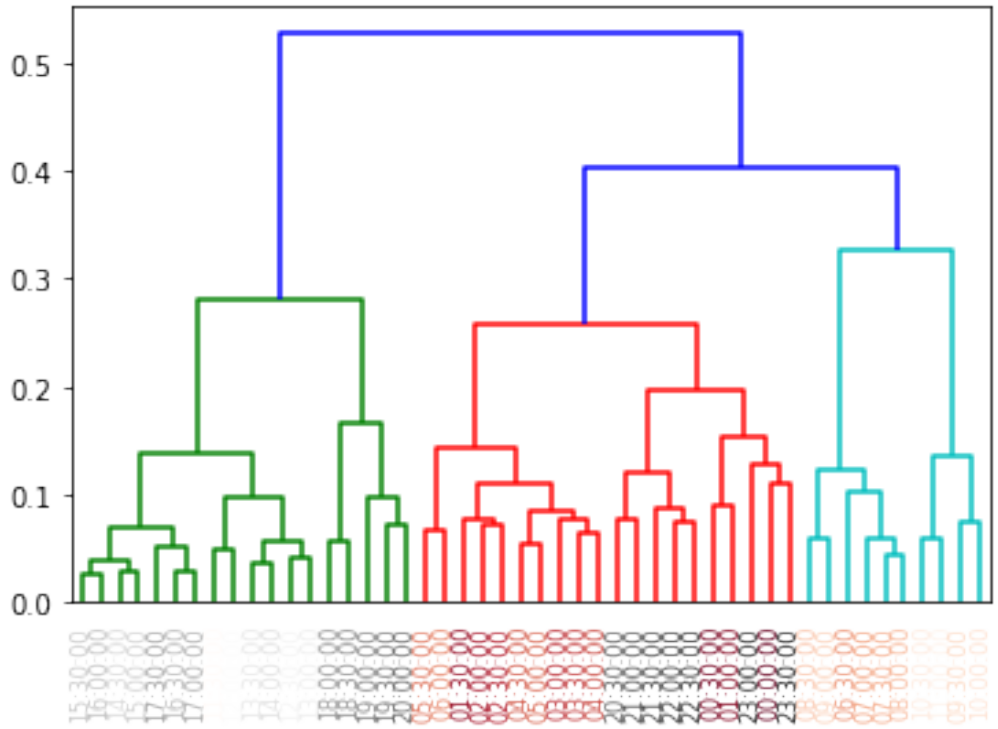

Figure 15: Dendrogram of cluster 1. Black is late in the day and red is early morning. The lighter colours are towards midday.

Cluster 2 contains 2 regions (Nouvelle-Aquitaine and Occitanie). In figure 16, in the top plot the label was 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, by time of the day, is observed.

In cluster 3, containing 3 regions (Auvergne-Rhône-Alpes, Bourgogne-Franche-Comté and Grand-Est) things are very different. The time of the day is the most important variable, as apart from Grand-Est, there are 2 main clusters, the late-night and early-morning cluster and the rest of the day (fig. 17).

Cluster 4 contains 4 regions (Bretagne, Centre-Val de Loire, Ile-de-France, Normandie and Pays-de-la-Loire), but only late night and early morning times . Here the regional clusters are very clear as all regions have been split with no clear time cluster (fig. 18).

In cluster 5, there are 5 regions, the same ones as in cluster 4 as well as Hauts-de-France.

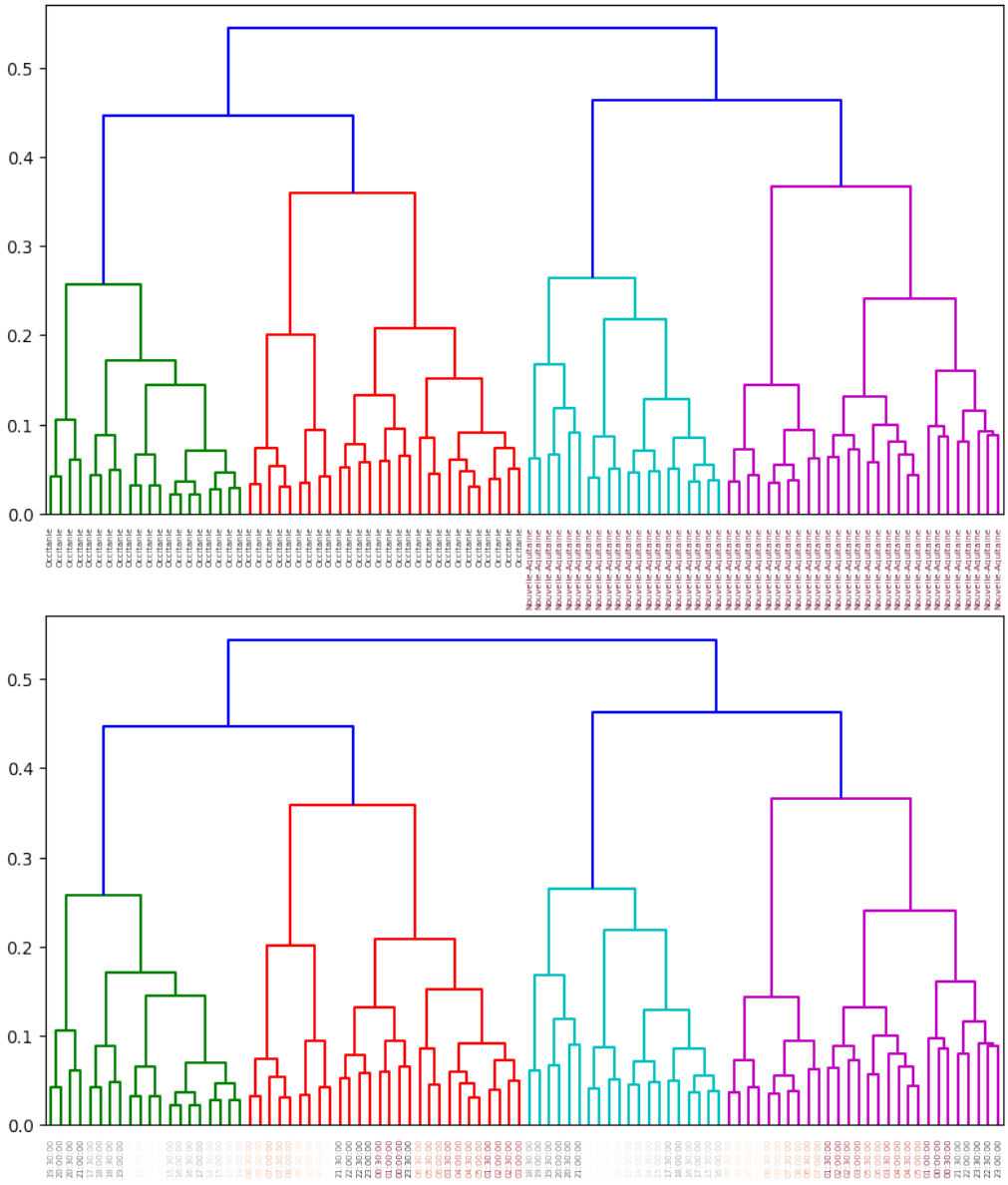

Figure 16: Dendrogram of cluster 2. Top: Black is Occitanie and red is Nouvelle-Aquitaine. Bottom: Black is late in the day and red is early morning. The lighter colours are towards midday.

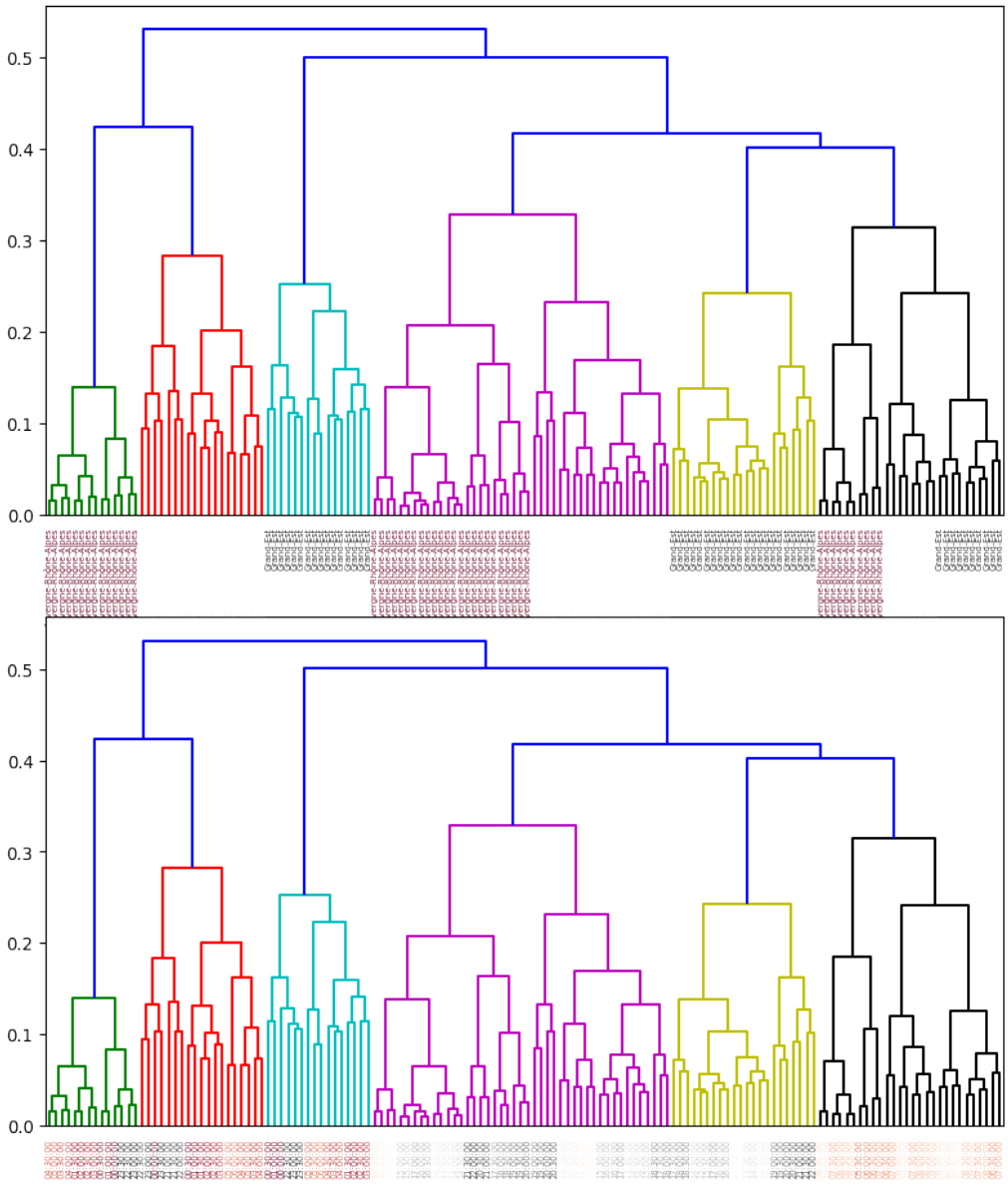

Figure 17: Dendrogram of cluster 3. Black is late in the day and red is early morning. The lighter colours are towards midday.

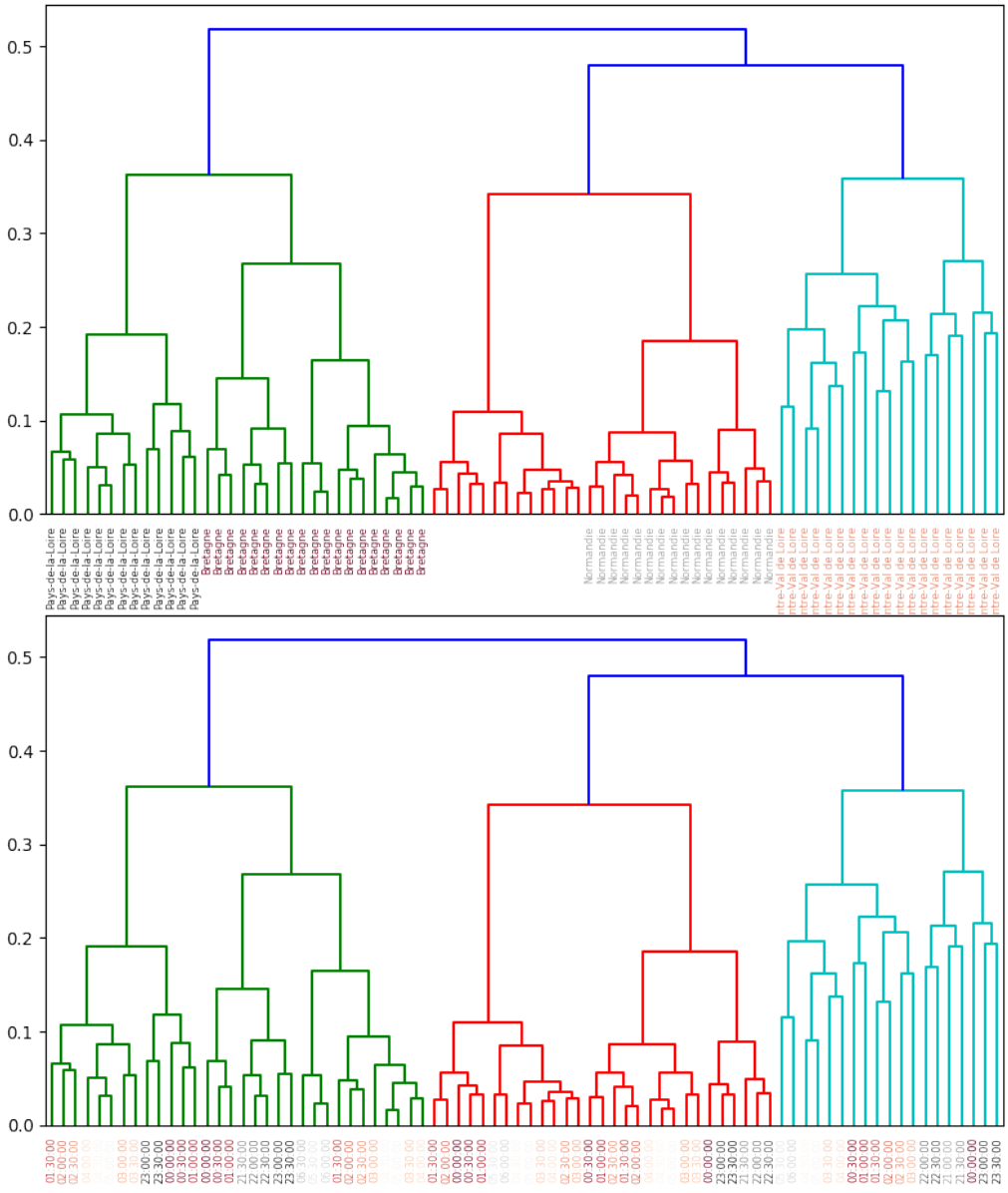

Figure 18: Dendrogram of cluster 4. Black is late in the day and red is early morning. The lighter colours are towards midday.

At all times, the Hauts-de-France was grouped with the evenings of the Centre-Val-de-Loire, Normandie and Iles-de-France, whereas the other regions are clustered into mornings and evenings.

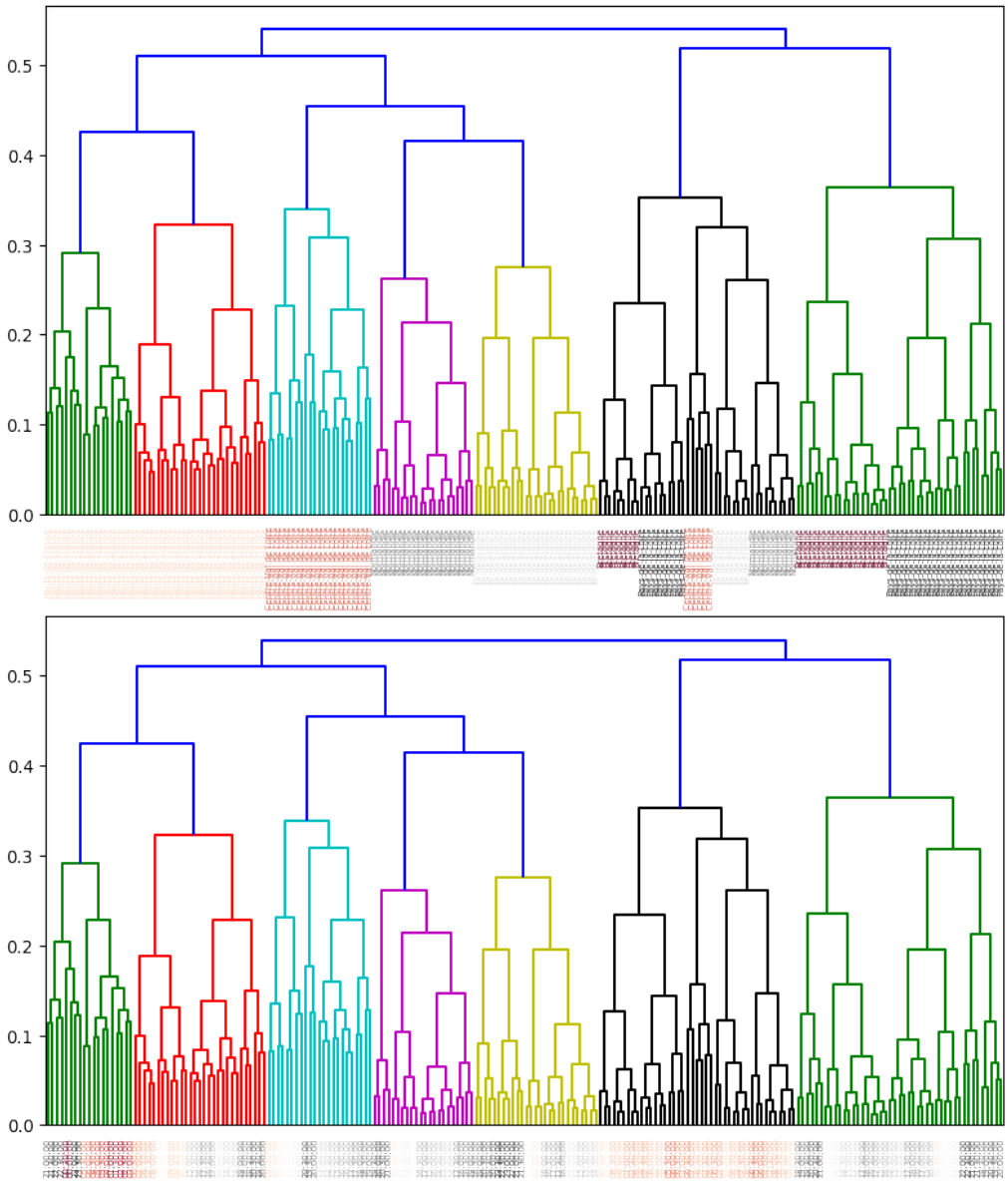

Figure 19: Dendrogram of cluster 5. Black is late in the day and red is early morning. The lighter colours are towards midday.

3.2.3 Clusters trends

As no information about the size of the population in each region was used, the absolute consumption were not compared between clusters. However, we can still compare relative changes over the years (fig. 20), seasons (fig. 21) and a typical day (fig. 22).

The 1 year trends 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 (fig. 20). The PACA region (cluster 1) is also clearly differenciated from the other ones. However, it is difficult to get clear conclusions as there are not enough data to analyse long term trends.

Figure 20: 1 year moving average trend of each cluster.

In the 3 months trend (fig. 21), 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 2 clusters are in the south of France, which is not really common (nor necessary) in the north.

Over the day (fig. 22), 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.

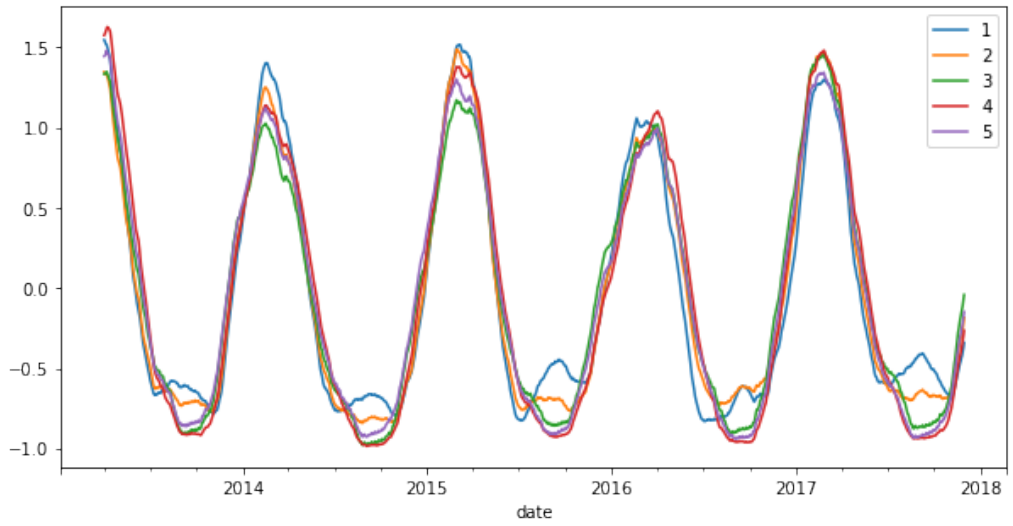

Figure 21: 3 months moving average trend of each cluster.

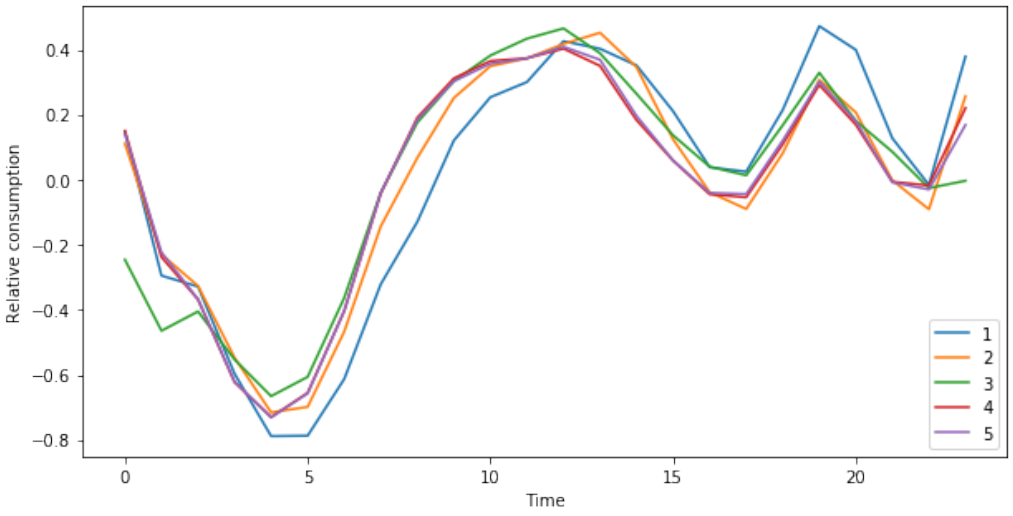

Figure 22: Hourly mean consumption of everyday for each cluster.

4 Conclusion

The clustering of the electricity consumption of the French regions was successful, as a very high degree of structure was observed. It was possible to group the 12×48 regions and time of day into 5 main geographical clusters. But also detect further clustering possibilities within each of those. Some clusters we dominated by different consumption in the morning, afternoon and night. Some other had further geographical delimitation, and others 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 2 clusters in the south of France. Furthermore, those 2 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.

The GCC, as defined by M. 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 their clustering without any pre-assumption about these. It is therefore a very competent metric 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.

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

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(1):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., Xu, X., et al. (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*.

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(8):68–75.

Kassambara, A. and Mundt, F. (2016). Factoextra: extract and visualize the results of multivariate data analyses. *R package version*, 1(3).

Kohonen, T. (1998). The self-organizing map. *Neurocomputing*, 21(1-3):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(4):595–607.

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874.

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

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(2):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.