

# Clustering time series based on probability distributions across temporal granularities

Sayani Gupta \*

Department of Econometrics and Business Statistics, Monash University  
and

Dianne Cook

Department of Econometrics and Business Statistics, Monash University  
and

Rob J Hyndman

Department of Econometrics and Business Statistics, Monash University

November 14, 2021

## Abstract

Clustering is a potential approach for organising large collections of times series into small homogeneous groups, but a difficult step is determining an appropriate metric to measure similarity between time series. The similarity metric needs to be capable of accommodating long, noisy, and asynchronous time series and also capture cyclical patterns. In this paper, two approaches for measuring distances between time series are presented, based on probability distributions over cyclic temporal granularities. Both are compatible with a variety of clustering algorithms. Cyclic granularities like hour-of-the-day, work-day/weekend, month-of-the-year, are useful for finding repeated patterns in the data. Measuring similarity based on probability distributions across cyclic granularities serves two purposes: (a) characterising the inherent temporal data structure of long, unequal-length time series in a manner robust to missing or noisy data; (b) small pockets of similar repeated behaviors can be captured. This approach is capable of producing useful clusters, as demonstrated on validation data designs and a sample of residential smart meter records.

*Keywords:* clustering, time granularities, probability distributions, Jensen-Shannon distances, periodic data, smart meter, electricity consumption behavior, R

---

\*Email: Sayani.Gupta@monash.edu

# 1 Introduction

Time-series clustering is the process of unsupervised partitioning of  $n$  time-series data into  $k$  ( $k < n$ ) meaningful groups such that homogeneous time-series are grouped together based on a certain similarity measure. The time-series features, length of time-series, representation technique, and, of course, the purpose of clustering time-series all influence the suitable similarity measure or distance metric to a meaningful level. The three primary methods to time series clustering (Liao (2005)) are algorithms that operate directly with distances or raw data points in the time or frequency domain (distance-based), with features derived from raw data (feature-based), or indirectly with models constructed from raw data (model-based). The efficacy of distance-based techniques is highly dependent on the distance measure utilized. Defining an appropriate distance measure for the raw time series may be a difficult task since it must take into account noise, variable lengths of time series, asynchronous time series, different scales, and missing data. Commonly used distance-based similarity measures as suggested by a decade review of time series clustering approaches (Aghabozorgi et al. (2015)) are Euclidean, Pearson’s correlation coefficient and related distances, Dynamic Time Warping (DTW), Autocorrelation, Short time series distance, Piecewise regularization, cross-correlation between time series, or a symmetric version of the Kullback–Liebler distances (Liao (2007)) but on a vector time series data. Among these alternatives, Euclidean distances have high performance but need the same length of data over the same period, resulting in information loss regardless of whether it is on raw data or a smaller collection of features. DTW works well with time series of different lengths (Corradini (2001)), but it is incapable of handling missing observations. Surprisingly, probability distributions, which may reflect the inherent temporal structure of a time series have not been considered in determining time series similarity.

This work is motivated by a need to cluster a large collection of residential smart meter data, so that households can be grouped into similar energy usage patterns. These can be considered to be univariate time series of continuous values which are available at fine temporal scales. These time series data are long (with more and more data collected at finer resolutions), are asynchronous, with varying time lengths for different houses and sporadic missing values. Using probability distributions is a natural way to analyze this types of

data because they are robust to uneven length, missing data, or noise. This paper proposes two approaches for obtaining pairwise similarities based on Jensen-Shannon distances between probability distributions across a selection of cyclic granularities. Cyclic temporal granularities (Gupta et al. 2021), which are temporal deconstructions of a time period into units such as hour-of-the-day, work-day/weekend, can measure repetitive patterns in large univariate time series data. The resulting clusters are expected to group customers that have similar repetitive behaviors across cyclic granularities. The benefits of this approach are as follows.

- When using probability distributions, data does not have to be the same length or observed during the exact same time period (unless there is a structural pattern).
- Jensen-Shannon distances evaluate the distance between two distributions rather than raw data, which is less sensitive to missing observations and outliers than other conventional distance methods.
- While most clustering algorithms produce clusters similar across just one temporal granularity, this technique takes a broader approach to the problem, attempting to group observations with similar distributions across all interesting cyclic granularities.
- It is reasonable to define a time series based on its degree of trend and seasonality, and to take these characteristics into account while clustering it. The modification of the data structure by taking into account probability distributions across cyclic granularities assures that there is no trend and that seasonal variations are handled independently. As a result, there is no need to de-trend or de-seasonalize the data before applying the clustering method. For similar reasons, there is no need to exclude holiday or weekend routines.

The primary application of this work is data from the Smart Grid, Smart City (SGSC) project (2010–2014) available through Department of the Environment and Energy. Half-hourly measurements of usage for more than 13,000 household electricity smart meters is provided from from October 2011 to March 2014. Households vary in size, location, and amenities such as solar panels, central heating, and air conditioning. The behavioral patterns differ amongst customers due to many temporal dependencies. Some households use

a dryer, while others dry their clothes on a line. Their weekly usage profile may reflect this. They may vary monthly, with some customers using more air conditioners or heaters than others, while having equivalent electrical equipment and weather circumstances. Some customers are night owls, while others are morning larks. Daily energy usage varies depending on whether customers stay home or work away from home. Age, lifestyle, family composition, building attributes, weather, availability of diverse electrical equipment, among other factors, make the task of properly segmenting customers into comparable energy behavior complex. The challenge is to be able to cluster consumers into these type of expected patterns, and other unexpected patterns, using only their energy usage history (Ushakova & Jankin Mikhaylov (2020)). There is a growing need to have methods that can examine the energy usage heterogeneity observed in smart meter data and what are some of the most common power consumption patterns.

There is a growing body of literature focused on time series clustering related to smart meter data. Tureczek & Nielsen (2017) conducted a systematic study of over 2100 peer-reviewed papers on smart meter data analytics. The most often used algorithm is  $k$ -means (Rhodes et al. 2014).  $k$ -means can be made to perform better by explicitly incorporating time series features such as correlation or cyclic patterns rather than performing it on raw data. To reduce dimensionality, several studies use principal component analysis (PCA) or factor analysis to pre-process smart-meter data before clustering (Ndiaye & Gabriel (2011)). PCA eliminates correlation patterns and decreases feature space, but loses interpretability. Other algorithms utilized in the literature include  $k$ -means variants, hierarchical clustering, and greedy  $k$ -medoids. Time series data, such as smart meter data, are not well-suited to any of the techniques mentioned in Tureczek & Nielsen (2017). Only one study (Ozawa et al. 2016) identified time series characteristics by first conducting a Fourier transformation, to convert data from time to frequency domain, followed by  $k$ -means to cluster by greatest frequency. Motlagh et al. (2019) suggests that the time feature extraction is limited by the type of noisy, patchy, and unequal time-series common in residential customers and addresses model-based clustering by transforming the series into other objects such as structure or set of parameters which can be more easily characterized and clustered. (Chicco & Akilimali 2010) addresses information theory-based clustering such as Shannon or Renyi

entropy and its variations. Melnykov (2013) discusses how outliers, noisy observations and scattered observations can complicate estimating mixture model parameters and hence the partitions. None of these methods focuses on exploring heterogeneity in repetitive patterns based on the dynamics of multiple temporal dependencies using probability distributions, which forms the basis of the methodology reported here.

This paper is organized as follows. Section~2 provides the clustering methodology. Section~3 shows data designs to validate our methods. Section~4 discusses the application of the method to a subset of the real data. Finally, we summarize our results and discuss possible future directions in Section~5.

## 2 Clustering methodology

The existing work on clustering probability distributions assumes we have an independent and identically distributed samples  $f_1(v), \dots, f_n(v)$ , where  $f_i(v)$  denotes the distribution from observation  $i$  over some random variable  $v = \{v_t : t = 0, 1, 2, \dots, T - 1\}$  observed across  $T$  time points. In our approach, instead of considering the probability distributions of the linear time series, we compare them across different categories of a cyclic granularity. We can consider categories of an individual cyclic granularity ( $A$ ) or combination of categories for two interacting granularities ( $A*B$ ) to have a distribution, where  $A, B$  are defined as  $A = \{a_j : j = 1, 2, \dots, J\}$  and  $B = \{b_k : k = 1, 2, \dots, K\}$ . For example, let us consider two cyclic granularities,  $A$  and  $B$ , representing hour-of-day and day-of-week, respectively. Then  $A = \{0, 1, 2, \dots, 23\}$  and  $B = \{Mon, Tue, Wed, \dots, Sun\}$ . In case individual granularities are considered, there are  $J = 24$  distributions of the form  $f_{i,j}(v)$  or  $K = 7$  distributions of the form  $f_{i,k}(v)$  for each customer  $i$ . In case of interaction,  $J * K = 168$  distributions of the form  $f_{i,j,k}(v)$  could be conceived for each customer  $i$ . Hence clustering these customers is equivalent to clustering these collections of conditional univariate probability distributions. Towards this goal, the next step is to decide how to measure distances between collections of univariate probability distributions. Here, we describe two approaches for finding distances between time series. Both of these approaches may be useful in a practical context, and produce very different but equally useful customer groupings. The distances can be supplied to any usual clustering algorithm, including  $k$ -means or hierarchical clustering, to



Figure 1: Flow chart illustrating the pipeline for our method for clustering time series.

group observations into a smaller more homogeneous collection. The flow of the procedures is illustrated in Figure 1.

## 2.1 Selecting granularities

(Gupta et al. 2021) provides a method for determining the significance of a cyclic granularity, and a ranking of multiple cyclic granularities. (This extends to harmonies, pairs of granularities that might interact with each other.) We define “significant” granularities as those with significant distributional differences across at least one category. The reason for subsetting granularities in this way, is that, clustering algorithms perform badly in the presence of nuisance variables. Granularities that do not have some difference between categories are likely to be nuisance variables. It should be noted that all of the time series in a collection may not have the same set of significant granularities. This is the approach for managing this generate a subset ( $S_c$ ) of significant granularities across a collection of time series:

- (a) Remove granularities from the comprehensive list that are not significant for any time series.
- (b) Select only the granularities that are significant for the majority of time series.

## 2.2 Data transformation

The shape and scale of the distribution of the measured variable (e.g. energy usage) affects distance calculations. Skewed distributions need to be symmetrised. Scales of individuals needs to be standardised, because clustering is to select similar patterns, not magnitude of usage. (Organising individuals based on magnitude can achieved simply by sorting on a statistic like the average value across time.) For the JS-based approaches, two data transformation techniques are recommended, normal-quantile transform (NQT) and robust scaling (RS). NQT is used in @Gupta et al. (2021) prior sorting granularities.

- RS: The normalized  $i^{th}$  observation is denoted by  $v_{norm} = \frac{v_t - q_{0.5}}{q_{0.75} - q_{0.25}}$ , where  $v_t$  is the actual value at the  $t^{th}$  time point and  $q_{0.25}$ ,  $q_{0.5}$  and  $q_{0.75}$  are the 25<sup>th</sup>, 50<sup>th</sup> and 75<sup>th</sup> percentile of the time series for the  $i^{th}$  observation. Note that,  $v_{norm}$  has zero mean and median, but otherwise the shape does not change.
- NQT: The raw data for all observations is individually transformed (Krzysztofowicz 1997), so that the transformed data follows a standard normal distribution. NQT will symmetrize skewed distributions. A drawback is that any multimodality will be concealed. This, this should be checked prior to applying NQT.

## 2.3 Data pre-preprocessing

Computationally in R, the data is assumed to be a “tsibble object” (Wang et al. (2020)) equipped with an index variable representing inherent ordering from past to present and a key variable that defines observational units over time. The measured variable for an observation is a time-indexed sequence of values. This sequence, however, could be shown in several ways. A shuffle of the raw sequence may represent hourly consumption throughout a day, a week, or a year. Cyclic granularities can be expressed in terms of the index set in the tsibble data structure.

The data object will change when cyclic granularities are computed, as multiple observations will be caetgorized into levels of the granularity, thus inducing multiple probability distributions. Directly computing Jensen-Shannon distances between the entire probability distributions can be computationally intensive. Thus it is recommended that quantiles are

used to characterize the probability distributions. In the final data object, each category of a cyclic granularity corresponds to a list of numbers which is composed of a few quantiles.

## 2.4 Distance metrics

The total (dis) similarity between each pair of customers is obtained by combining the distances between the collections of conditional distributions. This needs to be done in a way such that the resulting metric is a distance metric, and could be fed into the clustering algorithm. Two types of distance metric is considered:

### 2.4.1 JS-based distances

This distance metric considers two time series to be similar if the distributions of each category of an individual cyclic granularity or combination of categories for interacting cyclic granularities are similar. In this study, the distribution for each category is characterized using deciles (can potentially consider any list of quantiles), and the distances between distributions are calculated using the Jensen-Shannon distances (Menéndez et al. (1997)), which are symmetric and thus could be used as a distance measure.

The sum of the distances between two observations  $x$  and  $y$  in terms of cyclic granularity  $A$  is defined as

$$S_{x,y}^A = \sum_j D_{x,y}(A)$$

(sum of distances between each category  $j$  of cyclic granularity  $A$ ) or

$$S_{x,y}^{A*B} = \sum_j \sum_k D_{x,y}(A, B)$$

(sum of distances between each combination of categories  $(j, k)$  of the harmony  $(A, B)$ . After determining the distance between two series in terms of one granularity, we must combine them to produce a distance based on all significant granularities. When combining distances from individual  $L$  cyclic granularities  $C_l$  with  $n_l$  levels,

$$S_{x,y} = \sum_l S_{x,y}^{C_l} / n_l$$

is employed, which is also a distance metric since it is the sum of JS distances. This approach is expected to yield groups, such that the variation in observations within each



group is in magnitude rather than distributional pattern, while the variation between groups is only in distributional pattern across categories.

### 2.4.2 wpd-based distances

Compute weighted pairwise distances  $wpd$  (Gupta et al. (2021)) for all considered granularities for all observations.  $wpd$  is designed to capture the maximum variation in the measured variable explained by an individual cyclic granularity or their interaction and is estimated by the maximum pairwise distances between consecutive categories normalized by appropriate parameters. A higher value of  $wpd$  indicates that some interesting pattern is expected, whereas a lower value would indicate otherwise.

Once we have chosen  $wpd$  as a relevant feature for characterizing the distributions across one cyclic granularity, we have to decide how we combine differences between the multiple features (corresponding to multiple granularities) into a single number. The Euclidean distance between them is chosen, with the granularities acting as variables and  $wpd$  representing the value under each variable. With this approach, we should expect the observations with similar  $wpd$  values to be clustered together. Thus, this approach is useful for grouping observations that have a similar significance of patterns across different granularities. Similar significance does not imply a similar pattern, which is where this technique varies from JS-based distances, which detect differences in patterns across categories.

## 2.5 Clustering

### 2.5.1 Number of clusters

Different approaches to determining number of clusters are required for varied applications and research objectives. It is often beneficial to investigate numerous cluster statistics in order to arrive at a number of clusters, since clustering has a variety of objectives (e.g., between-cluster separation, within-cluster homogeneity, representation by centroids, etc.), which could be in conflict with one another. Some typical ways of determining the number of clusters in the literature (e.g., the gap statistic (Tibshirani et al. 2001), average silhouette width (Rousseeuw 1987), Dunn index (Dunn 1973)) are inspired by defining and balancing “within-cluster homogeneity” and “between-cluster separation”. The Cluster separation

index (*sindex*) proposed by Hennig (2014) is used as a guide to select the number of clusters in Section 3. *sindex* (Hennig 2020) is a compromise between minimum cluster separation and average cluster separation and is less sensitive to a single or few ambiguous points. *sindex* cannot be optimized over the number of clusters, since increasing the number of clusters reduces it. The number of clusters required is determined by the slope going from steep to shallow (an elbow).

### 2.5.2 Algorithm

With a way to obtain pairwise distances, any clustering algorithm can be employed that supports the given distance metric as input. A good comprehensive list of algorithms can be found in Xu & Tian (2015) based on traditional ways like partition, hierarchy, or more recent approaches like distribution, density, and others. We employ agglomerative hierarchical clustering in conjunction with Ward’s linkage. Hierarchical cluster techniques fuse neighboring points sequentially to form bigger clusters, beginning with a full pairwise distance matrix. The distance between clusters is described using a “linkage technique”. This agglomerative approach successively merges the pair of clusters with the shortest between-cluster distance using Ward’s linkage method.

### 2.5.3 Characterization of clusters

Cluster characterization is an important element of cluster analysis. Cook & Swayne (2007) provides several methods for characterizing clusters. *Parallel coordinate plots* (Wegman (1990)), *Scatterplot matrix*, *Displaying cluster statistics* (Dasu et al. (2005)), *MDS* (Borg & Groenen (2005)), *PCA*, *t-SNE* (Krijthe (2015)), *Tour* (Wickham et al. (2011)) are some of the graphical approaches used in this study. A Parallel Coordinates Plot features parallel axes for each variable and each axis is linked by lines. Changing the axes may reveal patterns or relationships between variables for categorical variables. However, for categories with cyclic temporal granularities, preserving the underlying ordering of time is more desirable. Displaying cluster statistics is useful when we have larger problems and it is difficult to read the parallel coordinate plots due to congestion. All of MDS, PCA and t-SNE use

a distance or dissimilarity matrix to construct a reduced-dimension space representation, their goals are diverse. Multidimensional scaling (Borg & Groenen (2005)) seeks to maintain the distances between pairs of data points, with an emphasis on pairings of distant points in the original space. The t-SNE embedding will compress data points that are close in high-dimensional space. Tour is a collection of interpolated linear projections of multivariate data into lower-dimensional space. The cluster characterization approach varies depending on the distance metric used. Parallel coordinate plots, scatter plot matrices, MDS or PCA are potentially useful ways to characterize clusters using wpd-based distances. For JS-based distances, plotting cluster statistics is beneficial for characterization and variable importance could be displayed through parallel coordinate plots.

### 3 Validation

To validate our clustering methods, we spiked many attributes in the data to create different data designs. Three circular granularities  $g1$ ,  $g2$  and  $g3$  are considered with categories denoted by  $\{g10, g11\}$ ,  $\{g20, g21, g22\}$  and  $\{g30, g31, g32, g33, g34\}$  and levels  $n_{g1} = 2$ ,  $n_{g2} = 3$  and  $n_{g3} = 5$ . These categories could be integers or some more meaningful labels. For example, the granularity “day-of-week” could be either represented by  $\{0, 1, 2, \dots, 6\}$  or  $\{Mon, Tue, \dots, Sun\}$ . Here categories of  $g1$ ,  $g2$  and  $g3$  are represented by  $\{0, 1\}$ ,  $\{0, 1, 2\}$  and  $\{0, 1, 2, 3, 4\}$  respectively. A continuous measured variable  $v$  of length  $T$  indexed by  $\{0, 1, \dots, T-1\}$  is simulated such that it follows the structure across  $g1$ ,  $g2$  and  $g3$ . We constructed independent replications of all data designs  $R = \{25, 250, 500\}$  to investigate if our proposed clustering method can discover distinct designs in small, medium, and big number of series. All designs employ  $T = \{300, 1000, 5000\}$  sample sizes to evaluate small, medium, and large-sized series. Variations in method performance may be due to different jumps between categories. So a mean difference of  $\mu = \{1, 2, 5\}$  between categories is considered. The performance of the approaches varies with the number of granularities which has interesting patterns across its categories. So three scenarios are considered to accommodate that.

### 3.1 Data generation

Each category or combination of categories from  $g1$ ,  $g2$  and  $g3$  are assumed to come from the same distribution, a subset of them from the same distribution, a subset of them from separate distributions, or all from different distributions, resulting in various data designs. As the methods ignore the linear progression of time, there is little value in adding time dependency to the data generating process. The data type is set to be “continuous,” and the setup is assumed to be Gaussian. When the distribution of a granularity is “fixed”, it means distributions across categories do not vary and are considered to be from  $N(0,1)$ .  $\mu$  alters in the “varying” designs, leading to varying distributions across categories.

### 3.2 Data designs

#### 3.2.1 Individual granularities

*Scenario (S1): All granularities significant*

Consider the instance where  $g1$ ,  $g2$ , and  $g3$  all contribute to design distinction. This means that each granularity will have significantly different patterns at least across one of the designs to be clustered. In Table 1 various distributions across categories are considered (top) which lead to different designs (bottom). Figure 2 shows the simulated variable’s linear (left) and cyclic (right) representations for each of these five designs. The structural difference in the time series variable is impossible to discern from the linear view, with all of them looking very similar. The shift in structure may be seen clearly in the distribution of cyclic granularities. The following scenarios use solely graphical displays across cyclic granularities to highlight distributional differences in categories.

*Scenario (S2): Few significant granularities*

This is the case where one granularity will remain the same across all designs. We consider the case where the distribution of  $v$  varies across  $g2$  levels for all designs, across  $g3$  levels for a few designs, and  $g1$  does not vary across designs. The proposed design is shown in Figure 3(b).

*Scenario (S3): Only one significant granularity*

Only one granularity is responsible for identifying the designs in this case. This is

Table 1: For S1, distributions of different categories when they vary (top). If distributions are fixed, they are set to  $N(0, 1)$ . 5 designs resulting from different distributions across categories (below)

granularity	Varying distributions
g1	$g_{10} \sim N(0, 1), g_{11} \sim N(2, 1)$
g2	$g_{21} \sim N(2, 1), g_{22} \sim N(1, 1), g_{23} \sim N(0, 1)$
g3	$g_{31} \sim N(0, 1), g_{32} \sim N(1, 1), g_{33} \sim N(2, 1), g_{34} \sim N(1, 1), g_{35} \sim N(0, 1)$

design	g1	g2	g3
design-1	fixed	fixed	fixed
design-2	vary	fixed	fixed
design-3	fixed	vary	fixed
design-4	fixed	fixed	vary
design-5	vary	vary	vary

depicted in Figure 3 (right) where only  $g3$  affects the designs significantly.

### 3.2.2 Interaction of granularities

The proposed methods could be extended when two granularities of interest interact and we want to group subjects based on the interaction of the two granularities. Consider a group that has a different weekday and weekend behavior in the summer but not in the winter. This type of combined behavior across granularities can be discovered by evaluating the distribution across combinations of categories for different interacting granularities (Weekend/Weekday and month-of-year in this example). As a result, in this scenario, we analyze a combination of categories generated from different distributions. Display of design and related results can be found in supplementary.

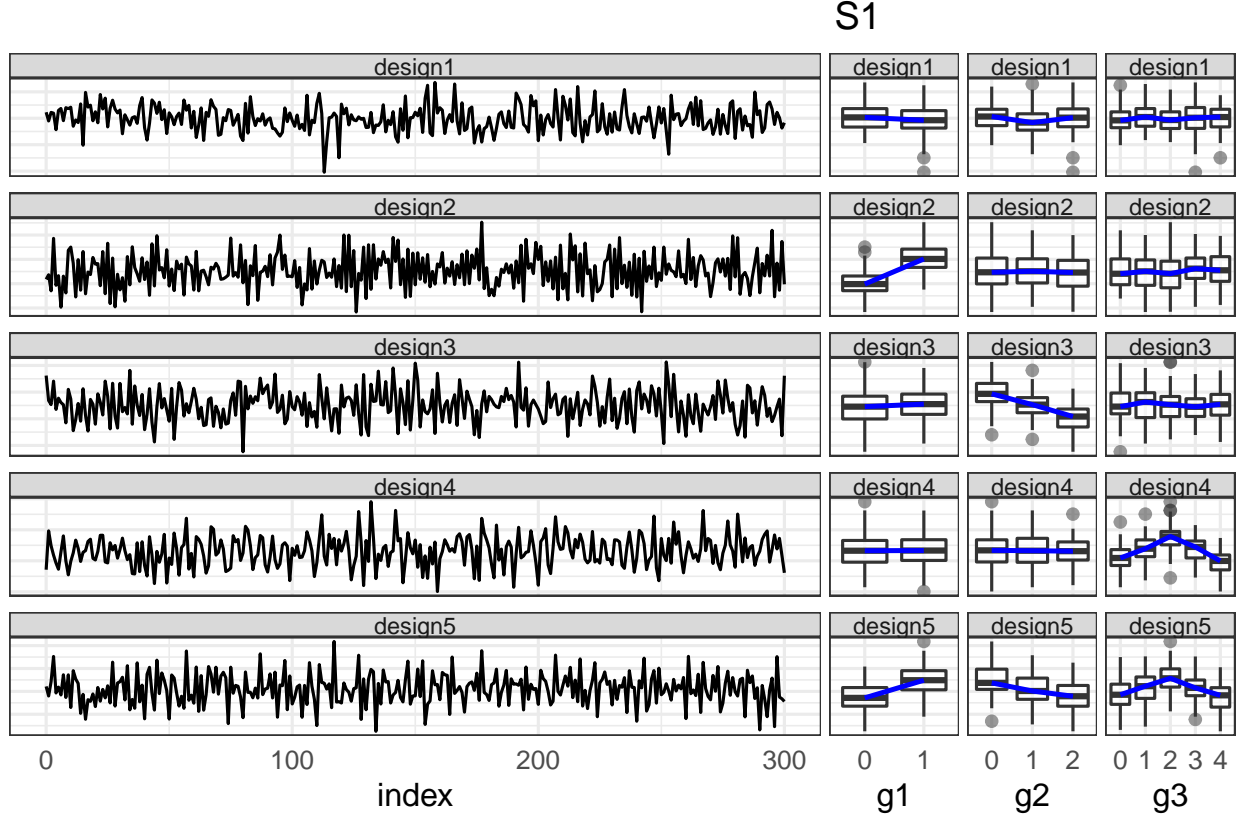


Figure 2: The linear (left) and cyclic (right) representation is shown under scenario S1 using line plots and boxplots respectively. Each row represents a design. Distributions of categories across  $g1$ ,  $g2$  and  $g3$  change across at least one design as can be observed in the cyclic representation. It is not possible to comprehend these structural differences in patterns just by looking at or considering the linear representation.

S2



S3



Figure 3: Boxplots showing distributions of categories across different designs (rows) and granularities (columns) for scenarios S2 and S3. In S2,  $g_2$ ,  $g_3$  change across atleast one design but  $g_1$  remains constant. Only  $g_3$  changes across different designs in S3.

### 3.3 Visual exploration of results

All of the approaches were fitted to each data design and for each combination of the considered parameters. The formed clusters have to match the design, be well separated, and have minimal intra-cluster variation. It is possible to study these desired clustering traits visually in a more comprehensive way than just looking at index values. So we use MDS and parallel coordinate graphs to demonstrate the findings:

- Figure 4 shows  $\text{sinde}_x$  plotted against the number of clusters ( $k$ ). This can be used to determine the number of clusters for each scenario. Each line in the plots represents a different simulation setting for the same scenario. When  $\text{sinde}_x$  for each scenario are examined, it appears that  $k = 5, 4, 4$  is justified for scenarios S1, S2, and S3, respectively, given the sharp decrease in  $\text{sinde}_x$  from that value of  $k$ . The number of clusters corresponds to the number of designs that were originally considered in each scenario.
- In Figure 6, we tried to see how separated our clusters are. We observe that in all scenarios and for different mean differences, clusters are separated. However, the separation increases with an increase in mean differences across scenarios. This is intuitive because, as the difference between categories increases, it gets easier for the methods to correctly distinguish the designs.
- Figure 5 depicts a parallel coordinate plot with the vertical bar showing total inter-cluster distances with regard to granularities  $g1$ ,  $g2$ , and  $g3$  for all simulation settings and scenarios. So one line in the figure shows the inter-cluster distances for one simulation setting and scenarios vary across facets. The lines are not colored by group since the purpose is to highlight the contribution of the factors to categorization rather than class separation. The first plot shows that no variable stands out in the clustering, but the following two designs show that  $\{g1\}$  and  $\{g1, g2\}$  have very low inter cluster distances, meaning that they did not contribute to the clustering. It is worth noting that these facts correspond to our original assumptions when developing the scenarios, which incorporate distributional differences over three (S1), two (S2), and one (S3) significant granularities. Hence, Figure 5 (S1), (S2), and (S3) validate



the construction of scenarios (S1), (S2), and (S3) respectively.

- The js-robust and wpd methods perform worse for  $nT = 300$ , then improve for higher  $nT$  evaluated in the study. Although, a complete year of data is the minimum requirement to capture distributional differences in winter and summer profiles, for example. Even if the data is only available for a month,  $nT$  with half-hourly data is expected to be at least 1000. As a result, as long as the performance is promising for higher  $nT = 300$ , this is not a challenge.
- In our study sample, the method js-nqt outperforms the method js-robust for smaller differences between categories. More testing, however, is required to corroborate this.

For more detailed results, please refer to the supplementary paper. The code for creating these data designs and running the methodologies is available at (<https://github.com/Sayani07/paper-gracsR/Validation>).

## 4 Application

The use of our methodology is illustrated on smart meter energy usage for a sample of customers from SGSC consumer trial data which was available through Department of the Environment and Energy and Data61 CSIRO. It contains half-hourly general supply in Kwh for 13,735 customers, resulting in 344,518,791 observations in total. The raw data for these consumers is of unequal length, with varying start and finish dates. Because our proposed methods evaluate probability distributions rather than raw data, neither of these data features would pose any threat to our methodology unless they contained any structure or systematic patterns. Additionally, there were missing values in the database but further investigation revealed that there is no structure in the missingness (see Supplementary paper for raw data features and missingness).

Clustering huge data sets presents a slew of difficulties. As a result, we dissect the larger problem and test our solutions on a small sample of prototype customers. To do this, data is first filtered to generate a clean sample, and then significant cyclic granularities (variables) for them are chosen. This is described in Section 4.1. The clean set is subsequently

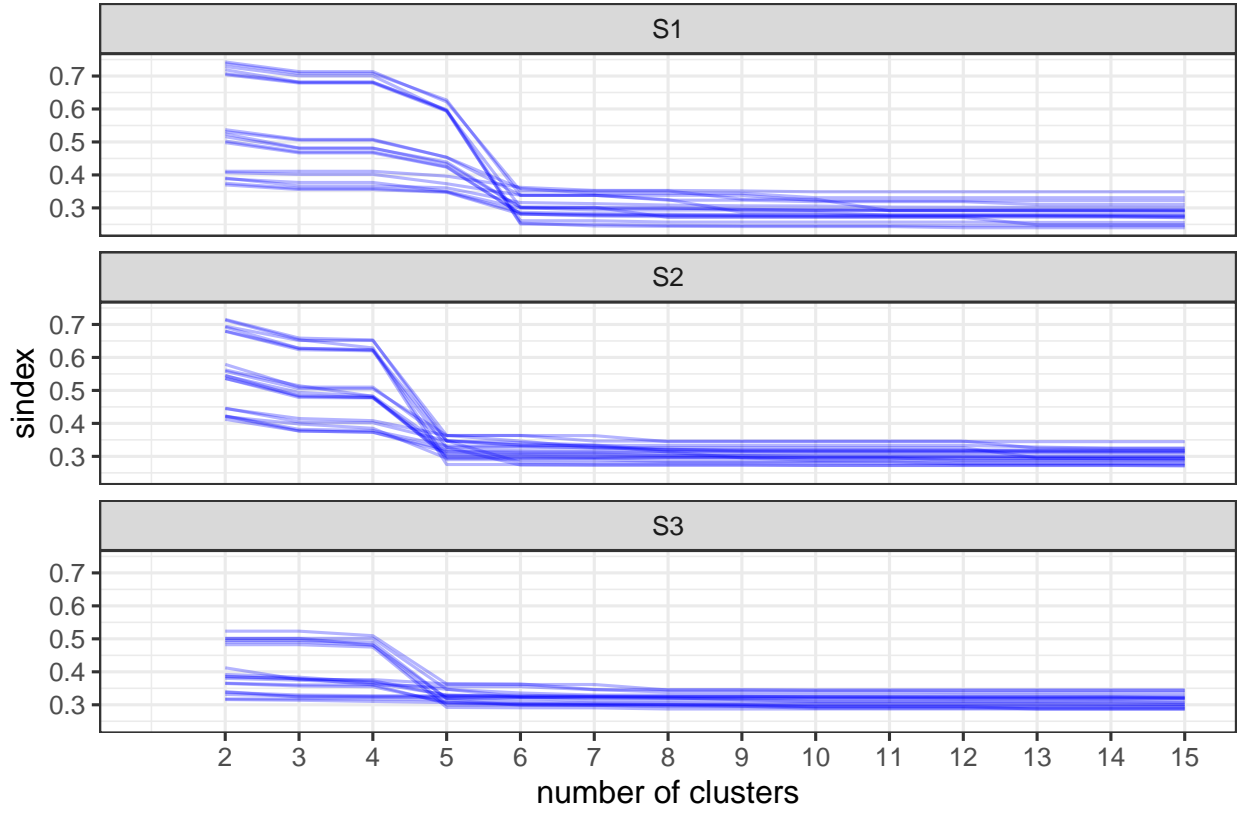


Figure 4: sindex plotted as a function of the number of clusters for S1, S2, and S3. S1 has a sharp decrease in sindex from 5 to 6, whereas S2 and S3 have a decrease from 4 to 5. As a result, the number of clusters considered for S1, S2, and S3 is 5, 4, 4, respectively. This corresponds to the number of designs taken into account in each scenario.

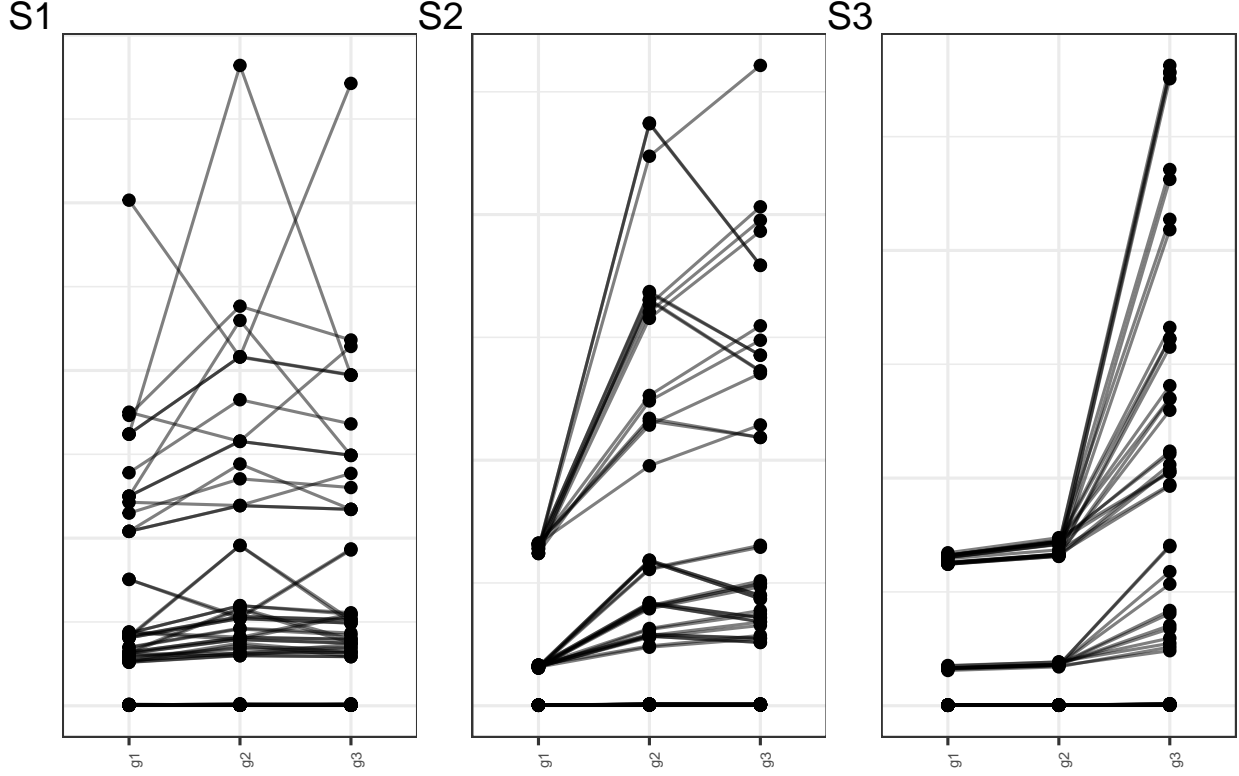


Figure 5: The parallel coordinate plot illustrates the total inter-cluster distances for granularities  $g_1$ ,  $g_2$ , and  $g_3$ . One line in the figure depicts the inter-cluster distances for a single simulation scenario. While the first plot indicates that no variable stands out during clustering, the next two designs demonstrate that  $g_1$  and  $(g_1, g_2)$  have relatively lower inter-cluster distances, indicating that they did not contribute to clustering. It is worth emphasising that these facts are consistent with our initial assumptions when designing the scenarios and S1, S2, S3 correspond to scenarios with three, two and one significant granularity respectively.

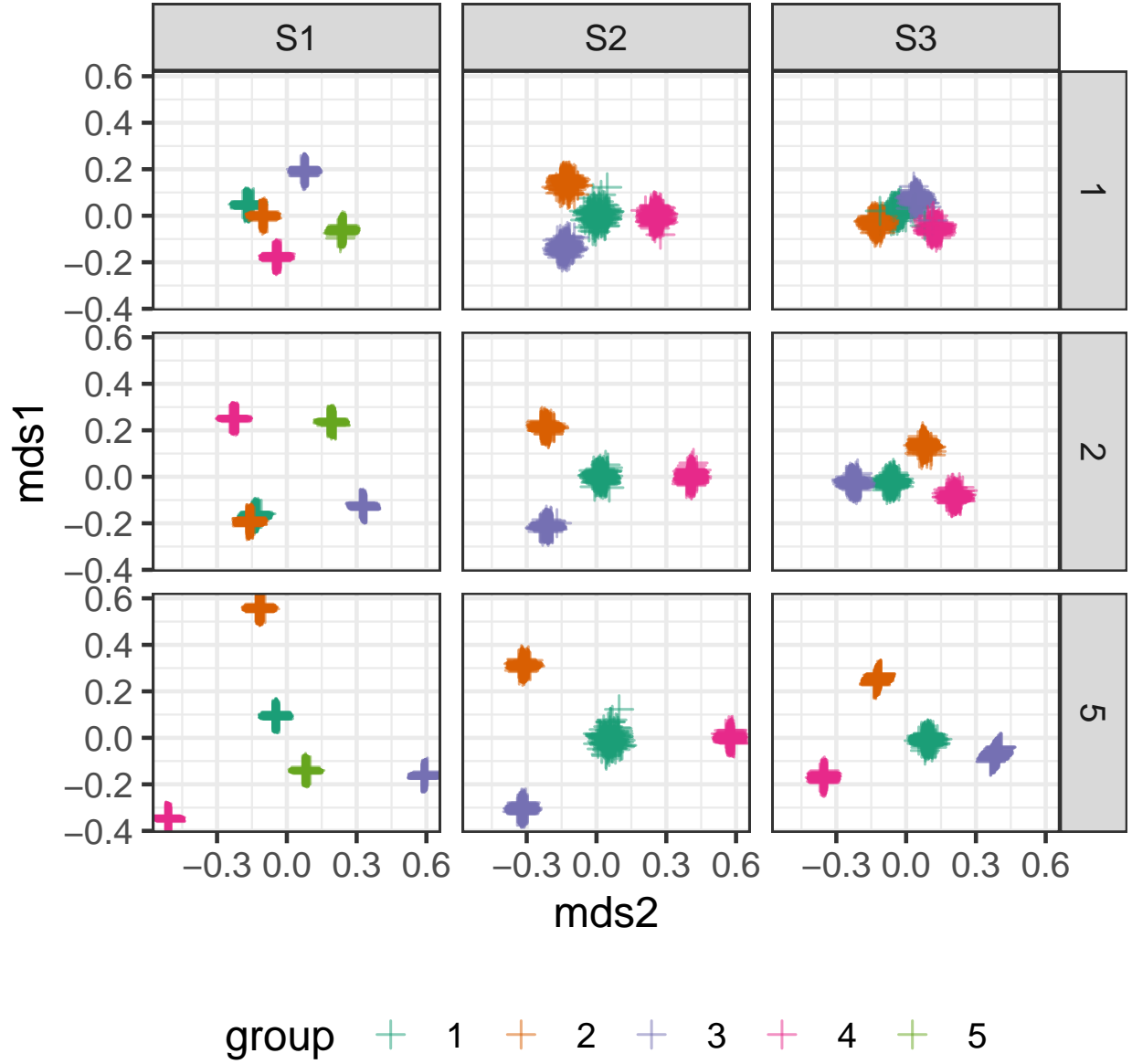


Figure 6: MDS summary plots to illustrate the cluster separation for the range of mean differences (rows) under the different scenarios (columns). It can be observed that clusters become more compact and separated for higher mean differences between categories across all scenarios. Between scenarios, separation is least prominent corresponding to Scenario (S3) where only one granularity is responsible for distinguishing the clusters.

subsampled along all dimensions of interest, to ensure that the sampled data reveals some patterns across at least one specified variable, which is described in Section 4.2. By grouping the prototypes using our methods in Section 4.3 and assessing their meaning, the study hopes to unravel some of the heterogeneities observed in energy usage data. Because our application does not employ additional customer data, we cannot explain why consumption varies, but rather try to identify how it varies.

## 4.1 Data filtering and variable selection

Things become far more complicated when we consider a larger data set with more uncertainty, as they do with any clustering problem.

- Choose a smaller subset of randomly selected 600 customers with no implicit missing values for 2013.
- Obtain *wpd* for all cyclic granularities considered for these customers. It was found that **hod** (hour-of-day), **moy** (month-of-year) and **wkndwday** (weekend/weekday) are coming out to be significant for most customers. We use these three granularities while clustering.
- Remove customers whose data for an entire category of **hod**, **moy** or **wnwd** is empty. For example, a customer who does not have data for an entire month is excluded because their monthly behavior cannot be analyzed.
- Remove customers whose energy consumption is 0 in all deciles. These are the clients whose consumption is likely to remain essentially flat and with no intriguing repeated patterns that we are interested in studying.

## 4.2 Prototype selection

Supervised learning uses a training set of known information to categorize new events through instance selection. Instance selection (Olvera-López et al. (2010)) is a method of rejecting instances that are not helpful for classification. This is analogous to subsampling the population along all dimensions of interest such that the sampled data represents the

primary features of the underlying distribution. Instance selection in unsupervised learning has received little attention in the literature, yet it could be a useful tool for evaluating model or method performance. There are several ways to approach prototype selection. Following Fan et al. (2021)’s idea of picking related examples (neighbors) for each instance (anchor), we can first use any dimensionality reduction techniques like MDS or PCA to project the data into a 2D space. Then pick a few “anchor” customers who are far apart in 2D space and pick a few neighbors for each. Unfortunately, this does not ensure that consumers with significant patterns across all variables are chosen. Tours can reveal variable separation that gets hidden in a single variable display better than static projections. Hence we perform a linked tour with a t-SNE layout using the R package `liminal` (Lee (2021)) to identify customers who are more likely to have distinct patterns across the variables studied. (Refer to Supplementary article for further details). Figure 8 shows the distribution across `hod` (a), `moy`(b) and `wnwd` (c) for the set of chosen 24 customers that were chosen. Few of these customers have similar distribution across `moy` and some are similar in their `hod` distribution.

### 4.3 Clustering

The 24 prototypes are clustered using the methodology described in Section 2 and results are reported below. In the following plots, the median is shown by a line, and the shaded region shows the area between the 25<sup>th</sup> and 75<sup>th</sup> percentiles. Groups by JS-based distances and wpd-based distances are colored differently as they represent different groupings. The plotting scales are not displayed since we want to emphasize comparable patterns rather than scales. The idea is that a customer in a cluster may have low total energy usage, but their behavior may be quite similar to a customer with high usage with respect to distributional pattern or significance across cyclic granularities.

#### 4.3.1 JS-based distances

For clustering based on JS-based distances, we chose the optimal number of clusters using (Hennig (2014)) as 5. The groupings are shown in Figure 8. Our methodology is useful for grouping similar distributions over `hod` and `moy` and they are placed closely for easy com-

parison. Few groups have mixed patterns across *hod* and *moy*, but few have all customers in the group having a similar profile. Figure ?? shows the summarized distributions across 5 groups and assists us in characterizing each cluster. It shows Groups 2 and 4 have *hod* pattern with a typical morning and evening peak, whereas groups 1, 3, and 5 show a *moy* pattern with higher usage in winter months. Differences in Weekend/Weekday between groups are not discernible, implying that it may not be a relevant variable in distinguishing various clusters unless maybe conditioned by *moy* or *hod*. It may be interesting to compare these two plots to verify if the summarized distributions across groups correctly characterised the groupings. If it has, then the majority of the group’s members should share a similar profile.

### 4.3.2 wpd-based distances

We chose the optimal number of clusters using (Hennig (2014)) as 3. A parallel coordinate plot with the three significant cyclic granularities is used to characterise the groups here. The variables are sorted according to their separation across classes (rather than their overall variation between classes). This means that *moy* is the most important variable in distinguishing the groups followed by *hod* and *wnwd*. There is only one customer who has significant *wpd* across *wnwd* and stands out from the rest of the customers. Group 3 has a higher *wpd* for *hod* than *moy* or *wkndwday*. Group 2 has the most distinct pattern across *moy*. Group 1 is a mixed group that has strong patterns on at least one of the three variables. The findings vary from *js*-based clustering, yet it is a helpful grouping.

Things become far more complicated when we consider a larger data set with more uncertainty, as they do with any clustering problem. Summarizing distributions across clusters with varied or outlying customers can result in a shape that does not represent the group. Furthermore, combining heterogeneous customers may result in similar-looking final clusters that are not effective for visually differentiating them. It is also worth noting that the Weekend/Weekday behavior in the given case does not characterize any cluster. This, however, will not be true for all of the customers in the data set. If more extensive prototype selection is used, resulting in more comprehensive prototypes in the data set, this method might be used to classify the entire data set into these prototype behaviors. However, the

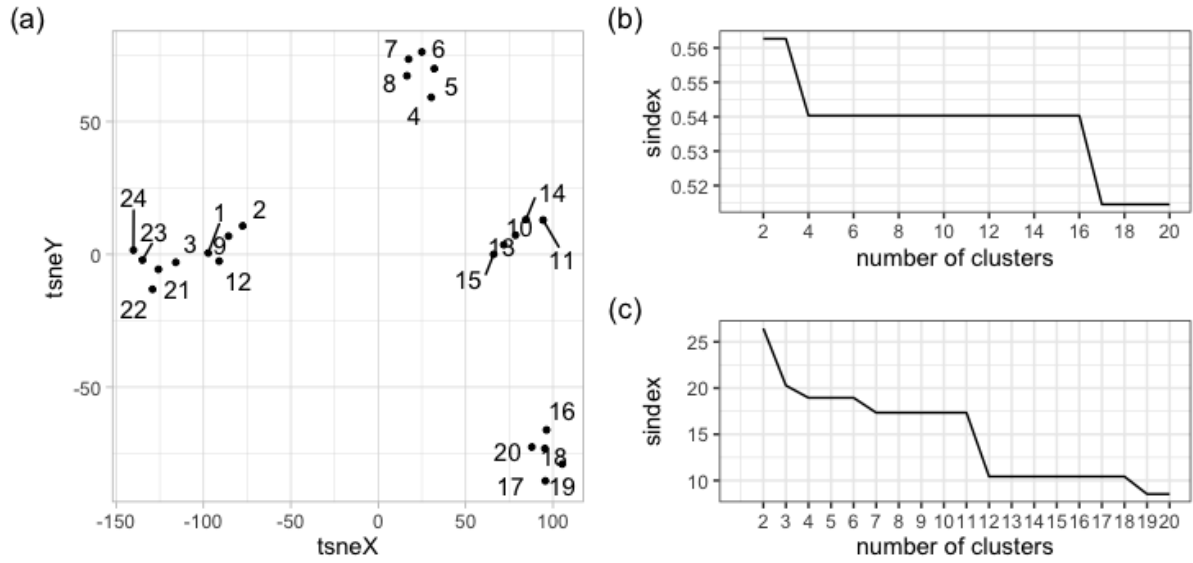


Figure 7: (a) shows t-SNE summary of the 24 selected customers chosen through prototype selection, (b) and (c) shows a plot of sindex for different cluster size for clustering based on JS-based distances and wpd-distances respectively. The plot suggests optimal number of clusters as 4 and 7 respectively.



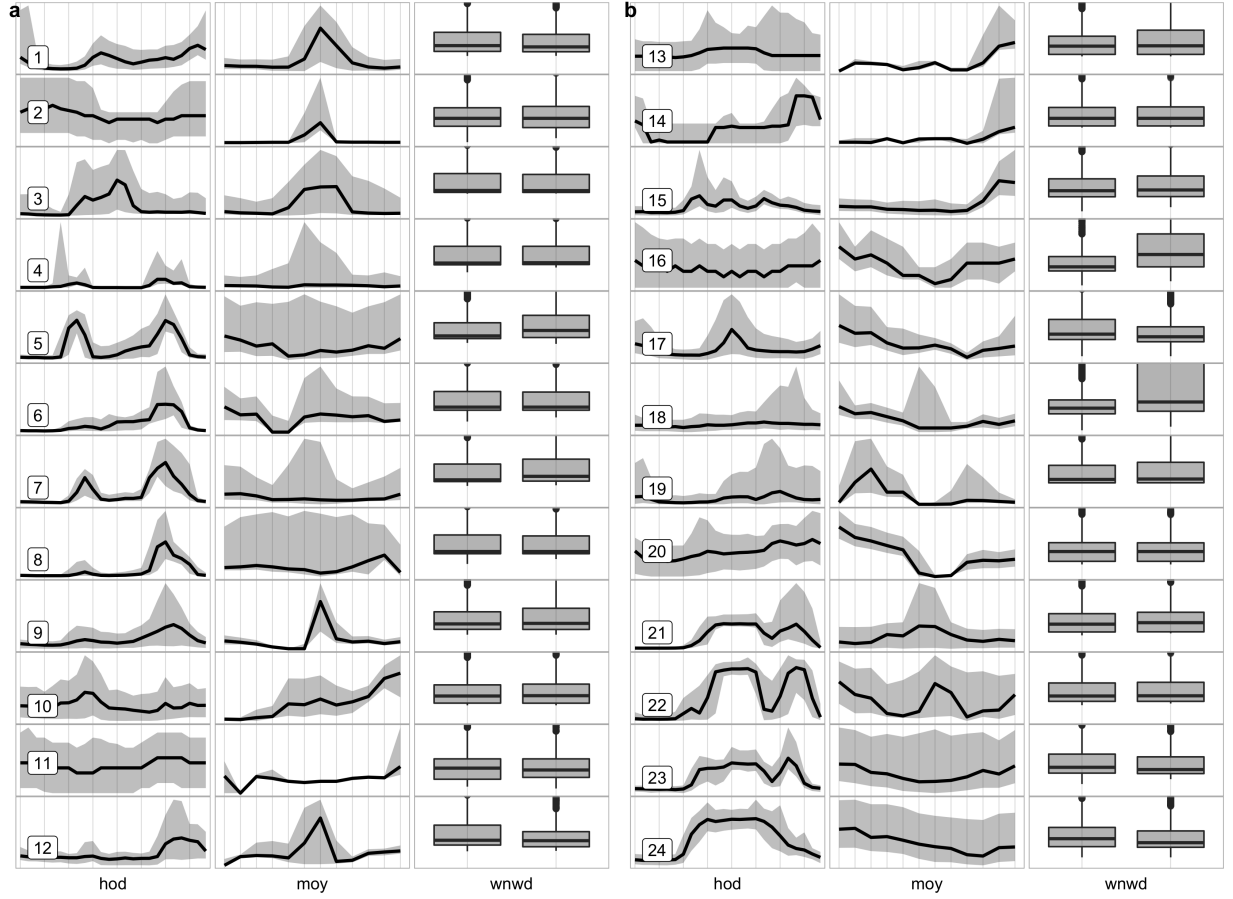


Figure 8: The distribution of electricity demand across individual customers over multiple granularities hod, moy, and wnwd are shown for the 24 selected customers using quantile and box plots. They are split into batches of 12 in (a) and (b), with each row in (a) or (b) representing a customer. Each customer is identified by their profile across all the granularities. A few of these customers have similar distributions across moy and some are similar in their hod distributions. Distributional differences across categories of wnwd are not discernable except for a couple of customers (id: 16 and 18).

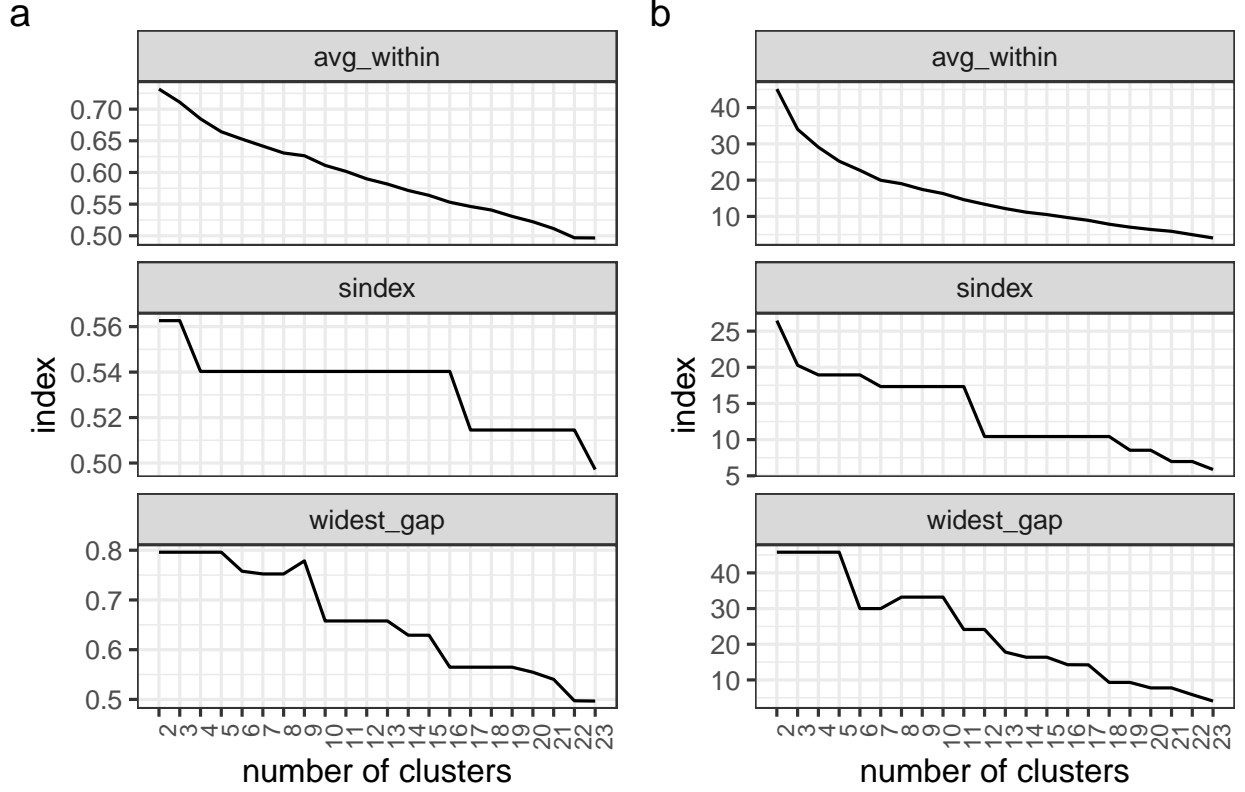


Figure 9: Different cluster statistics plotted as a function of number of clusters for JS-based (a) and wpd-based (b) approaches. avg within is not too useful for revealing any sharp change in average distance between clusters with increasing number of clusters. For (a), sindex sees a sharp fall at  $k = 4$  and then do not decrease until  $k = 16$  and the widest gap is high till  $k = 5$ . For (b), sindex sees a fall at  $k = 3$  and then at  $k = 7$  and widest gap has a sharp fall at  $k = 6$ .

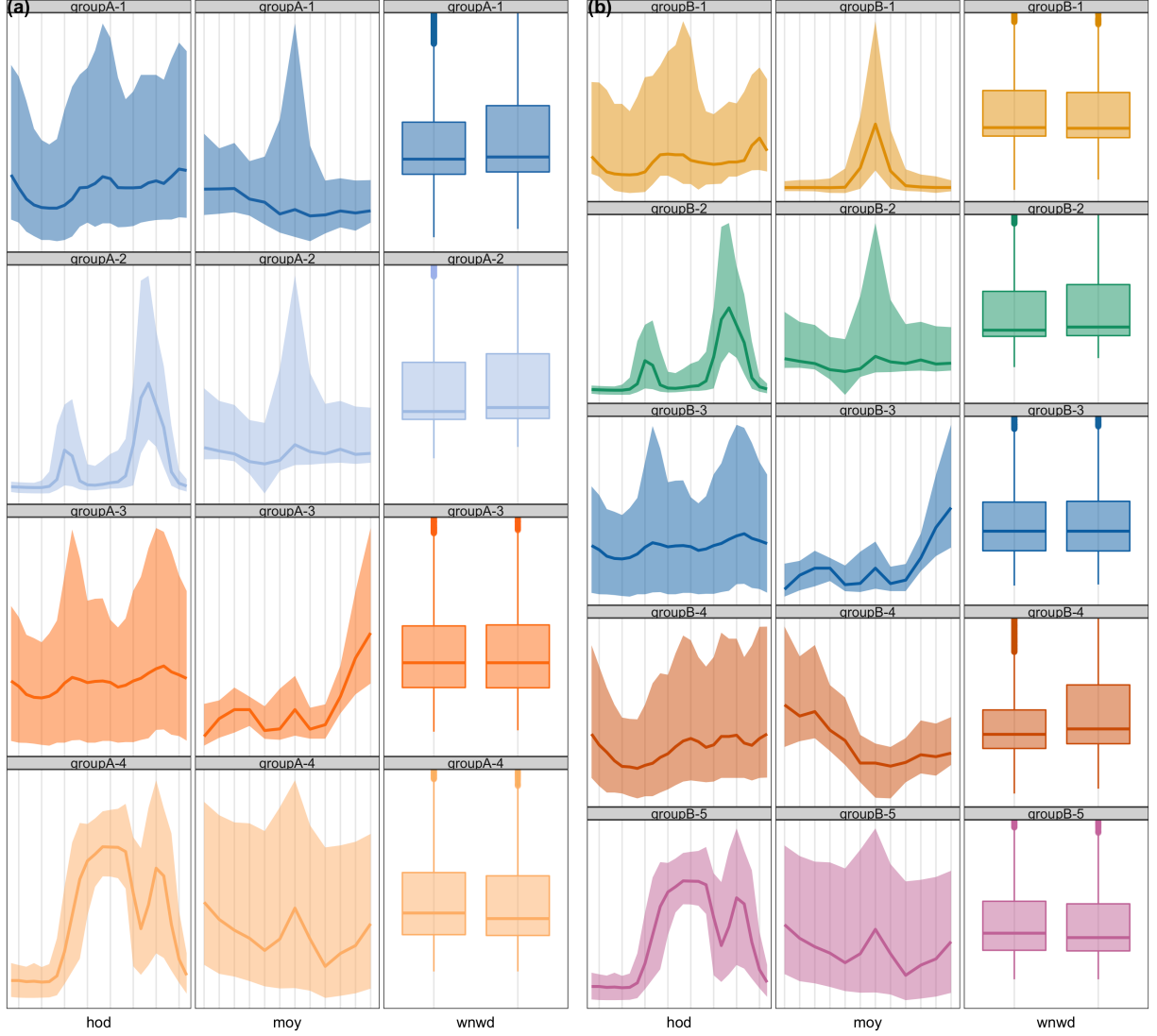


Figure 10: For  $k = 4$  (a) and  $k = 5$  (b), the distribution of electricity demand across groups over hod, moy, and wnwd is shown. groupA-2, groupA-3, and groupA-4 profiles correspond to groupB-2, groupB-3, and groupB-5, respectively. groupA-1 is split into groupB-1 and groupB-4, each of which has a distinct shape across moy and wnwd. groupB-4 (id: 16-20) is characterized by higher energy consumption in the first few months of the year and more variation in weekend usage. groupB-1 (id:1-3) is distinguished by higher consumption in the middle of the year (winter months) and similar weekday-weekend inter-quartile range. When  $k = 4$  is used, these two groups merge to form groupA-1, which has a moy profile of higher usage in both the beginning and middle of the year, which is not representative of the individuals in the group.

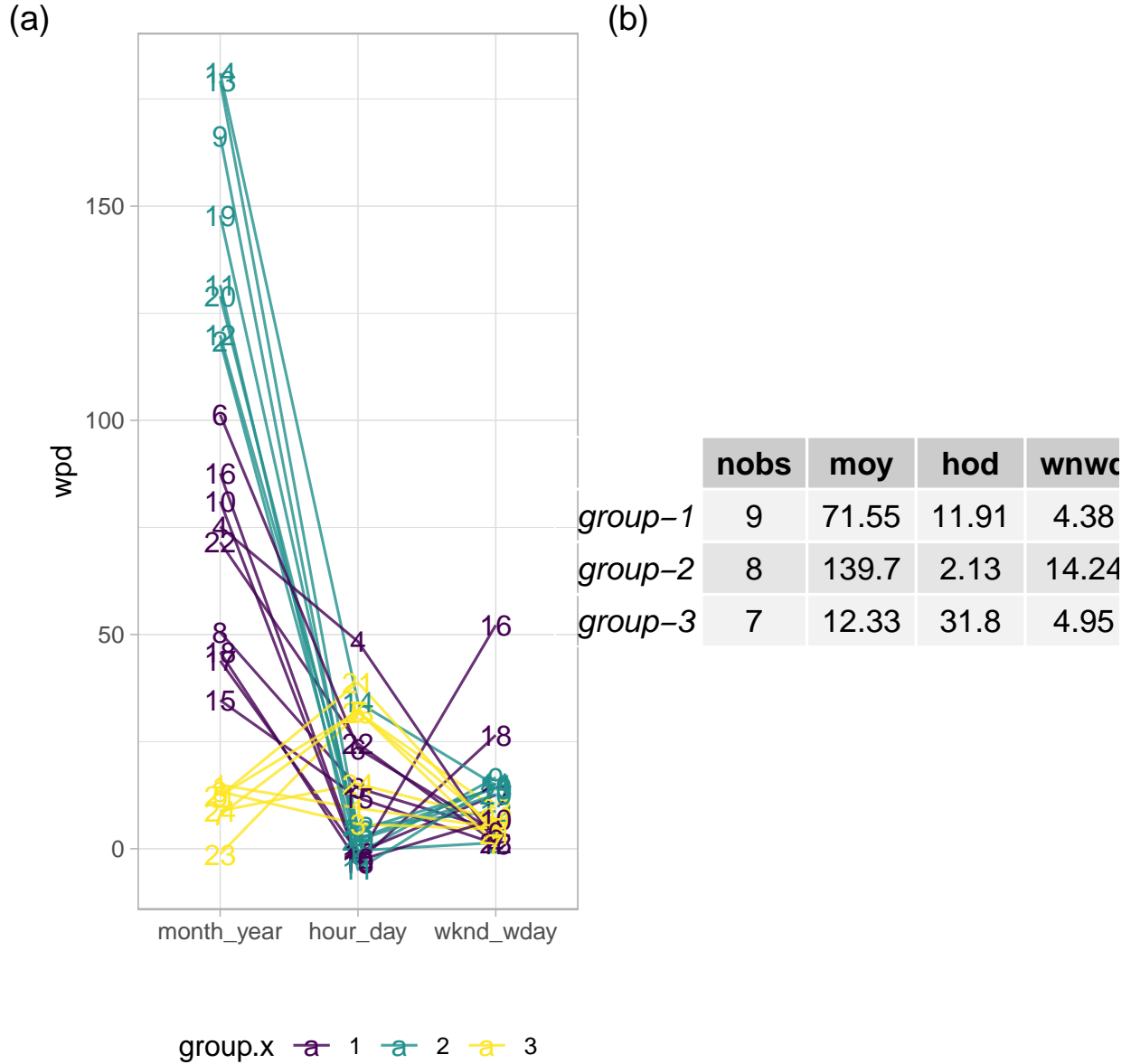


Figure 11: Each of the 24 customers is represented by a parallel coordinate plot (a) with three wpd-based groupings. Plot (b) which shows median wpd values for each group. The plot shows that moy is the most important variable in identifying clusters, whereas wnwd is the least significant and has the least fluctuation. Two customers (id: 16 and 18) with high wpd across wnwd stand out in this display. Group 3 has a higher wpd for hod than moy or wkndwday. Group 2 has most discernible pattern across moy. Group 1 is a mixed group with strong patterns on atleast one of the three variables. All of these could be observed from the plot or the table (b).

goal of this section was to have a few customers that have significant patterns over one or more cyclic granularities, apply our methodology to cluster them, and demonstrate that the method produces useful clusters.

## 5 Discussion

We offer two approaches for calculating pairwise distances between time series based on probability distributions over multiple cyclic granularities at once. Depending on the goal of the clustering, these distance metrics, when fed into a hierarchical clustering algorithm using Ward’s linkage, yield meaningful clusters. Probability distributions provide an intuitive method to characterise noisy, patchy, long, and unequal-length time series data. Distributions over cyclic granularities help to characterise the formed clusters in terms of their repeating behavior over these cyclic granularities. Furthermore, unlike earlier efforts that group customers based on behavior across only one cyclic granularity (such as hour-of-day), our method is more comprehensive in detecting clusters with repeated patterns at all relevant granularities.

There are few areas to extend this research. First, larger data sets with more uncertainty complicate matters, as is true for any clustering task. Characterizing clusters with varied or outlying customers can result in a shape that does not represent the group. Moreover, integrating heterogeneous consumers may result in visually identical end clusters, which are potentially not useful. Hence, a way of appropriately scaling it up to many customers such that anomalies are removed before clustering would be useful for bringing forth meaningful, compact and separated clusters. Secondly, we have assumed the time series to be stationary, and hence the distributions are assumed to remain constant for the observation period. In reality, however, it might change. For the smart meter example, the distribution for a customer moving to a different house or changing electrical equipment can change drastically. Our current approach can not detect these dynamic changes. Thirdly, it is possible that for a few customers, data for some categories from the list of considered significant granularities are missing. In our application, we have removed those customers and done the analysis but the metrics used should be able to incorporate those customers in the clustering by handling their missing categories. Finally, *wpd* is computationally heavy

even under parallel computation. Future work can make the computations more efficient so that they are easily scalable to a large number of customers. Moreover, experiments can also be run with non-hierarchy based clustering algorithms to verify if these distances work better with other algorithms.

## Acknowledgments

The authors thank the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) for supporting this research. Sayani Gupta was partially funded by Data61 CSIRO during her PhD. The Monash eResearch Centre and eSolutions-Study Support Services supported this research in part through the resource usage of the MonARCH HPC Cluster. The Github repository, [github.com/Sayani07/paper-gracsr](https://github.com/Sayani07/paper-gracsr), contains all materials required to reproduce this article and the code is also available online in the supplemental materials. This article was created with R (R Core Team 2021), `knitr` (Xie 2015, Xie (2020)) and `rmarkdown` (Xie et al. 2018, Allaire et al. (2020)). Graphics are produced with `ggplot2` (Wickham 2016) and `GGally` (Schloerke et al. 2021).

## 6 Supplementary Materials

**Data and scripts:** Data sets and R code to reproduce all figures in this article (main.R).

**Supplementary paper:** Additional tables, graphics and and R code to reproduce it (paper-supplementary.pdf, paper-supplementary.Rmd)

**R-package:** To implement the ideas provided in this research, the open-source R package ‘gracsr’ is available on Github (<https://github.com/Sayani07/gracsr>).

## 7 Bibliography

## References

Aghabozorgi, S., Seyed Shirkhorshidi, A. & Ying Wah, T. (2015), ‘Time-series clustering – a decade review’, *Inf. Syst.* **53**, 16–38.

- Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W. & Iannone, R. (2020), *rmarkdown: Dynamic Documents for R*. R package version 2.1.  
**URL:** <https://github.com/rstudio/rmarkdown>
- Borg, I. & Groenen, P. J. (2005), *Modern multidimensional scaling: Theory and applications*, Springer Science & Business Media.
- Chicco, G. & Akilimali, J. S. (2010), ‘Renyi entropy-based classification of daily electrical load patterns’, *IET generation, transmission & distribution* **4**(6), 736–745.
- Cook, D. & Swayne, D. F. (2007), *Interactive and Dynamic Graphics for Data Analysis: With R and Ggobi*, Springer, New York, NY.
- Corradini, A. (2001), Dynamic time warping for off-line recognition of a small gesture vocabulary, *in* ‘Proceedings IEEE ICCV workshop on recognition, analysis, and tracking of faces and gestures in real-time systems’, IEEE, pp. 82–89.
- Dasu, T., Swayne, D. F. & Poole, D. (2005), Grouping multivariate time series: A case study, *in* ‘Proceedings of the IEEE Workshop on Temporal Data Mining: Algorithms, Theory and Applications, in conjunction with the Conference on Data Mining, Houston’, Citeseer, pp. 25–32.
- Dunn, J. C. (1973), ‘A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters’.
- Fan, H., Liu, P., Xu, M. & Yang, Y. (2021), ‘Unsupervised visual representation learning via Dual-Level progressive similar instance selection’, *IEEE Trans Cybern* **PP**.
- Gupta, S., Hyndman, R. J. & Cook, D. (2021), ‘Detecting distributional differences between temporal granularities for exploratory time series analysis’, *unpublished*.
- Hennig, C. (2014), How many bee species? a case study in determining the number of clusters, *in* ‘Data Analysis, Machine Learning and Knowledge Discovery’, Springer International Publishing, pp. 41–49.

- Hennig, C. (2020), *fpc: Flexible Procedures for Clustering*. R package version 2.2-9.  
**URL:** <https://CRAN.R-project.org/package=fpc>
- Krijthe, J. H. (2015), *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation*. R package version 0.15.  
**URL:** <https://github.com/jkrijthe/Rtsne>
- Krzysztofowicz, R. (1997), ‘Transformation and normalization of variates with specified distributions’, *J. Hydrol.* **197**(1-4), 286–292.
- Lee, S. (2021), *liminal: Multivariate Data Visualization with Tours and Embeddings*. R package version 0.1.2.  
**URL:** <https://CRAN.R-project.org/package=liminal>
- Liao, T. W. (2005), ‘Clustering of time series data—a survey’, *Pattern recognition* **38**(11), 1857–1874.
- Liao, T. W. (2007), ‘A clustering procedure for exploratory mining of vector time series’, *Pattern Recognition* **40**(9), 2550–2562.
- Melnykov, V. (2013), ‘Challenges in model-based clustering’, *Wiley Interdiscip. Rev. Comput. Stat.* **5**(2), 135–148.
- Menéndez, M. L., Pardo, J. A., Pardo, L. & Pardo, M. C. (1997), ‘The Jensen-Shannon divergence’, *J. Franklin Inst.* **334**(2), 307–318.
- Motlagh, O., Berry, A. & O’Neil, L. (2019), ‘Clustering of residential electricity customers using load time series’, *Appl. Energy* **237**, 11–24.
- Ndiaye, D. & Gabriel, K. (2011), ‘Principal component analysis of the electricity consumption in residential dwellings’, *Energy Build.* **43**(2), 446–453.
- Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F. & Kittler, J. (2010), ‘A review of instance selection methods’, *Artificial Intelligence Review* **34**(2), 133–143.



- Ozawa, A., Furusato, R. & Yoshida, Y. (2016), ‘Determining the relationship between a household’s lifestyle and its electricity consumption in japan by analyzing measured electric load profiles’, *Energy and Buildings* **119**, 200–210.
- R Core Team (2021), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.  
**URL:** <https://www.R-project.org/>
- Rhodes, J. D., Cole, W. J., Upshaw, C. R., Edgar, T. F. & Webber, M. E. (2014), ‘Clustering analysis of residential electricity demand profiles’, *Appl. Energy* **135**, 461–471.
- Rousseeuw, P. J. (1987), ‘Silhouettes: a graphical aid to the interpretation and validation of cluster analysis’, *Journal of computational and applied mathematics* **20**, 53–65.
- Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., Elberg, A. & Crowley, J. (2021), *GGally: Extension to 'ggplot2'*. R package version 2.1.1.  
**URL:** <https://CRAN.R-project.org/package=GGally>
- Tibshirani, R., Walther, G. & Hastie, T. (2001), ‘Estimating the number of clusters in a data set via the gap statistic’, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(2), 411–423.
- Tureczek, A. M. & Nielsen, P. S. (2017), ‘Structured literature review of electricity consumption classification using smart meter data’, *Energies* **10**(5), 584.
- Ushakova, A. & Jankin Mikhaylov, S. (2020), ‘Big data to the rescue? challenges in analysing granular household electricity consumption in the united kingdom’, *Energy Research & Social Science* **64**, 101428.
- Wang, E., Cook, D. & Hyndman, R. J. (2020), ‘A new tidy data structure to support exploration and modeling of temporal data’, *Journal of Computational & Graphical Statistics* **29**(3), 466–478.
- Wegman, E. J. (1990), ‘Hyperdimensional data analysis using parallel coordinates’, *Journal of the American Statistical Association* **85**(411), 664–675.

Wickham, H. (2016), *ggplot2: Elegant Graphics for Data Analysis*, Springer-Verlag New York.

**URL:** <http://ggplot2.org>

Wickham, H., Cook, D., Hofmann, H., Buja, A. et al. (2011), ‘tourr: An r package for exploring multivariate data with projections’, *Journal of Statistical Software* **40**(2), 1–18.

Xie, Y. (2015), *Dynamic Documents with R and knitr*, 2nd edn, Chapman and Hall/CRC, Boca Raton, Florida.

**URL:** <https://yihui.name/knitr/>

Xie, Y. (2020), *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.28.

**URL:** <https://yihui.org/knitr/>

Xie, Y., Allaire, J. J. & Golemund, G. (2018), *R Markdown: The Definitive Guide*, Chapman and Hall/CRC, Boca Raton, Florida.

**URL:** <https://bookdown.org/yihui/rmarkdown>

Xu, D. & Tian, Y. (2015), ‘A comprehensive survey of clustering algorithms’, *Annals of Data Science* **2**(2), 165–193.