MONASH University

Visualization and analysis of probability distributions of large temporal data

Sayani Gupta

M.Stat, B.Sc (Stat Hons)

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2022

Department of Econometrics and Business Statistics

Contents

Copyright notice

Abstract

The research was motivated by the desire to understand some Australian smart meter data, which was collected half-hourly for two years at the household level. With temporal data available at ever finer scales, exploring periodicity can become overwhelming with so many possible temporal deconstructions to explore. Analysts are expected to comprehensively explore the many ways to view and consider temporal data. However, the plethora of choices and the lack of a systematic approach to do so quickly can make the task daunting.

This work investigates how time may be dissected, resulting in alternative data segmentation and, as a result, different visualizations that can aid in the identification of underlying patterns. The first contribution (Chapter 2) describes classes of time deconstructions using linear and cyclic time granularities. It provides tools to compute possible cyclic granularities from an ordered (usually temporal) index and also a framework to systematically explore the distribution of a univariate variable conditional on two cyclic time granularities by defining "harmony". A "harmony" denotes pairs of granularities that could be effectively analyzed together and reduces the search from all possible options. This approach is still overwhelming for human consumption due to the vast number of harmonies remaining. The second contribution (Chapter 3) refines the search for informative granularities by identifying those for which the differences between the displayed distributions are greatest and also rating them in order of importance of capturing maximum variation. The third contribution (Chapter 4) builds upon the first two to provide methods for exploring heterogeneities in repetitive behavior for many time series and over multiple granularities. It accomplishes this by providing a way to cluster time series based on probability distributions across informative cyclic granularities. Although we were motivated by the smart meter example, the problem and the solutions we propose are practically relevant to any temporal data observed more than once per year.

Declaration

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Student signature:

Student name: Sayani Gupta

Date: 2022-02-28

Acknowledgements

I am deeply grateful to my supervisors Rob J Hyndman and Dianne Cook, who are phenomenal leaders in their respective fields and lead by example. I am inspired by their creativity, wisdom, discipline, and dedication to contribute to the society through their research. Thank you for constantly pushing me to improve as a programmer and researcher and regularly sharing good practices for conducting research. Looking back, I am grateful for how my thoughts and work in statistical computing, graphics, and data analytics, in general, have evolved over the years. I still have a lot to learn, but I am a much more self-reliant and independent researcher than when I started. Di has been instrumental in exposing me to the potential benefits of effective data visualization. As a female researcher, I am encouraged by her willingness to pursue unconventional avenues and have frequently noticed her make conscious and proactive choices to question existing stereotypes and biases. Thank you, Di, for being a fantastic female role model. I would also like to express my gratitude to my supervisor Dr. Peter Toscas of Data61 CSIRO, for always being available to help and offering feedback on practical elements of analyzing smart meter data at our monthly meetings.

I want to thank the Department of Econometrics and Business Statistics, Monash Business School, ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS), Data61 CSIRO for their financial support. Thanks to the Department, I attended a few conferences in person (before COVID hit) and had a great time networking and being inspired by many people. A few highlights are UseR! 2018 (Brisbane), Young Statistician's Conference 2019 (Canberra), WOMBAT 2019 (Melbourne), and ROpenSci 2019 (Sydney). Thank you to Gael Martin, Farshid Wahid, Klaus Ackermann, and Didier Nibbering from my PhD committee for providing such a supportive environment.

Thank you, Puwasala Gamakumara, Stephanie Kobakian and Nicholas Tierney, for being so warm, kind and welcoming when I began my PhD. Being a part of such an inclusive environment meant a

lot to me as an international student. Thank you, Nicholas Spyrison and Mitchell O'Hara-Wild, for the numerous informative discussions, brainstorming sessions, as well as for generously sharing your expertise and experience in software development. A special thanks goes to Stephanie, Emi Tanaka, Nicholas Tierney, and Nicholas Spyrison for taking the time out of their busy schedules to proofread parts of my thesis. Thank you to each of you, NUMBATs, and the broader R community for directly and indirectly helping me countless times. You never fail to inspire me.

Doing a PhD in COVID has been hard, and I would not have been able to make it through this if not for the emotional support of my friends and family. Thank you Puwasala, Sium, and Ian, for being so kind and acting as a pillar of emotional support, especially during the last leg of my journey. Thank you Tushar, Samarpita and Nairita for always motivating me and believing in me when I did not believe in myself. Thank you to each one of you for being available, listening to me, and putting things in perspective when I lost sight. Thanks to my housemates Anjali, Surbhi, and Dulaji for the fun company, food and conversations that kept me sane in the COVID lockdowns.

A big thank you to my family for always being supportive of my choices. Thanks to my mum (Nupur Gupta) and dad (Arun Prasad Gupta), from whom I learnt that no matter where you start from, if you persevere and are sincere in your efforts, you can sail through any difficult situation. They are my constant cheerleaders for all little and big endeavors. Thanks to my brother (Avijit) for always having my back and inspiring me to dream bigger. Thank you, Juhi (sister-in-law), for reminding me of the value of organization in all aspects of life and the importance of prioritizing my physical health from time to time. Lastly, thank you to my one-year-old nephew (Rayan) and niece (Mehr) for being the ultimate stress relievers. Thank you all for being a part of this journey with me. I love you.

Throughout my PhD candidature, I have had the great privilege of working with and being inspired by my advisers and many other people, which resulted in a great lot of learning in both academics and life. Although the journey is coming to a close, I want to carry these life lessons forward, and I would like to conclude with one of my favorites from Rob. "Be kind; remember you are collaborating with your future self."

Preface

Chapter 2 has been published online at the *Journal of Computational and Graphical Statistics*. It has won the ACEMS Business Analytics Prize in 2020. The accompanying R package `gravitas` is on CRAN. Chapter 3 and Chapter 4 are yet to be submitted.

Open and reproducible research

This thesis is written in R Markdown (Xie, Allaire, and Grolemund, 2018) with `bookdown` (Xie, 2016), using `renv` (Ushey, 2019) to create reproducible environments. The online version of this thesis is hosted at `https://sayani07.github.io/thesis-SG`. All materials (including the data sets and source files) required to reproduce this document can be found at the Github repository `https://github.com/Sayani07/thesis-SG`.

License

Chapter 1

Introduction

The Smart Grid, Smart City (SGSC) project (2010–2014) available through the Department of the Environment and Energy provides half-hourly data of over 13,000 Australian electricity smart meter customers distributed unevenly from October 2011 to March 2014. The wide variety of customers means that there will be large variance in behavior, leading to greater uncertainty in the data. Behavioral patterns vary significantly due to differences in size, location, and amenities such as solar panels, central heating, and air conditioning. For example, some families use a dryer, while others hang their clothes to dry. This could be reflected in their weekly profile. They may vary on a monthly basis, with some customers using more air conditioners or heaters than others despite having comparable electrical equipment and weather conditions. Some customers are night owls, while others are morning larks, which may show up in their daily profile. Customers' day-off energy consumption varies depending on whether they stay at home or go outside.

With the availability of data at finer and finer time scales, exploration of time series data may be required to be carried out across both finer and coarser scales to draw useful inferences about the underlying process. To reduce the complexity of time, it is typical to divide it into years, months, weeks, days, and so on in a hierarchical manner (Aigner et al., 2011). These discrete abstractions of time are known as time granularities. Linear time granularities (Bettini et al., 1998), such as hours, days, weeks and months, respect the linear progression of time and are non-repeating. Cyclic temporal granularities representing cyclical repetitions in time (such as hour-of-the-day, work-day/weekend) are effective for analyzing repetitive patterns in time series data.

To acquire a comprehensive view of the repeated patterns, it is necessary to navigate through all of the conceivable cyclic granularities. This approach is consistent with the concept of exploratory data analysis EDA (Tukey, 1977), which stresses the utilization of multiple perspectives on data to assist with formulating hypotheses before proceeding to formal inferences or modeling. This, however, is a challenging process since it throws up a myriad of possible hypotheses. Furthermore, the transition from linear to cyclic granularities results in restructured data, with each level of the temporal deconstruction corresponding to multiple values of the observed variable. This motivates the research presented in this thesis, which aims to provide a platform for systematically exploring probability distributions induced by these multiple observations to support the discovery of regular patterns or anomalies, as well as the exploration of clusters of behaviors or the summarization of the behavior. While we were prompted by the example of the smart meters, the challenges and solutions presented in this thesis are essentially applicable to any temporal data collected more than once a year, such as data from traffic sensors, pedestrian movement data in metro stations or public buildings, cab or bike rides, or even more traditional measurements such as temperature or precipitation collected over time. In a larger sense, it may be appropriate for data observed over years, decades, or centuries, as in weather or astronomical data.

1.1 Visualizing probability distributions across bivariate cyclic temporal granularities

Chapter 2 describes classes of time deconstructions using linear and cyclic time granularities, which can be used to create data visualizations to explore periodicities, associations, and anomalies. It provides a formal characterization of cyclic granularities and facilitates manipulation of single- and multiple-order-up time granularities through cyclic calendar algebra, as well as providing a recommendation algorithm to check the feasibility of creating plots for any two cyclic granularities. Our proposed method is also applicable to non-temporal hierarchical granularities with an underlying ordered index. The methods are implemented in the open-source R package `gravitas` and are consistent with a tidy workflow (Wickham and Grolemund, 2016), with probability distributions examined using the range of graphics available in `ggplot2` (Wickham, 2016).

1.2 Detecting distributional differences between temporal granularities for exploratory time series analysis

Chapter 3 is a natural extension of Chapter 2. Many displays might be built using cyclic granularities. However, only a handful of them may reveal major patterns of interest. Identifying the displays which exhibit "significant" distributional differences and plotting only these would allow for more efficient exploration. Furthermore, a few of the displays in this collection will be more engaging than others. Chapter 3 provides a new distance metric for selecting and ranking the multiple granularities. The statistical significance of potential visual discoveries is aided by selecting a threshold for the proposed numerical distance measure. The distance measure is computed for a single or pairs of cyclic granularities, and it can be compared across different cyclic granularities as well as a collection of time series. This chapter also includes a case study using residential smart meter data from Melbourne to demonstrate how the suggested methodology may be utilized to automatically find temporal granularities with significant distributional differences. The methods are implemented in the open-source R package `hakear`.

1.3 Clustering time series based on probability distributions across temporal granularities

In Chapter 4, we look at the problem of using clustering to discover patterns in a large number of univariate time series across multiple temporal granularities. Time series clustering research is gaining traction as more data is collected at finer temporal resolution, over longer time periods, and for a larger number of individuals/entities. Many disciplines have noisy, patchy, uneven, and asynchronous time series that make it difficult to search for similarities. We propose a method for overcoming these constraints by calculating distances between time series based on probability distributions at various temporal granularities. Because they are based on probability distributions, these distances are resistant to missing or noisy data and aid in dimension reduction. When fed into a clustering algorithm, the distances can be used to divide large data sets into small pockets of similar repetitive behaviors. These subgroups can then be analyzed separately or used as distinct prototype behaviors in classification problems. The proposed method was tested on a group of residential electricity consumers from the Australian smart meter data set to show that it can

generate meaningful clusters. This chapter includes a brief review of the literature on traditional time series clustering and, more specifically, clustering residential smart meter data.

1.4 Thesis structure

The thesis is structured as follows. Chapter 2 provides details of the cyclic granularities, different classes, and computation, and also its usage in exploratory time series analysis through applications. This is implemented in the R package `gravitas` (GRAnularity VIsualization for Time-series AnalySis). Chapter 3 provides guidance on how to choose significant cyclic granularities, which are likely to have interesting patterns across its categories. This is available as the R package `hakear` (HArmonies KEeper And Rater). Chapter 4 provides methods to explore heterogeneity in repetitive behavior for multiple time series over multiple cyclic granularities. This is in the developing R package `gracsr` (Granularity CluStering in R). Chapter 5 summarizes the software tools developed for the work, and discusses future plans. Table 1.1 details the publications, including my and my fellow co-authors contributions.

Table 1.1: *Table outlining the main chapters' co-authorship arrangements.*

Thesis Chapter	**Publication Title**	**Status (published, in press, accepted or returned for revision)**	**Nature and % of student contribution**	**Co-author name(s) Nature and % of Co-author's contribution**	**Co-author(s), Monash student Y/N**
2	Visualizing probability distributions across bivariate cyclic temporal granularities	Published	70%. Concept, methodology development, writing first draft and software development	(1) Rob J Hyndman, Concept, methodology and software development, writing 12% (2) Dianne Cook, Concept, methodology and software development, writing 12% (3) Antony Unwin, Concept and software development 6%	N
3	Detecting distributional differences between temporal granularities for exploratory time series analysis	Working Paper	80%. Concept, methodology development, writing first draft and software development	(1) Rob J Hyndman, Concept, methodology development and writing 10% (2) Dianne Cook, Concept, methodology development and writing 10%	N
4	Clustering time series based on probability distributions across temporal granularities	Working Paper	80%. Concept, methodology development, writing first draft and software development	(1) Dianne Cook, Concept, methodology development and writing 15% (2) Rob J Hyndman, Concept, methodology development and writing 5%	N

Chapter 2

Visualizing probability distributions across bivariate cyclic temporal granularities

Deconstructing a time index into time granularities can assist in exploration and automated analysis of large temporal data sets. This paper describes classes of time deconstructions using linear and cyclic time granularities. Linear granularities respect the linear progression of time such as hours, days, weeks and months. Cyclic granularities can be circular such as hour-of-the-day, quasi-circular such as day-of-the-month, and aperiodic such as public holidays. The hierarchical structure of granularities creates a nested ordering: hour-of-the-day and second-of-the-minute are single-order-up. Hour-of-the-week is multiple-order-up, because it passes over day-of-the-week. Methods are provided for creating all possible granularities for a time index. A recommendation algorithm provides an indication whether a pair of granularities can be meaningfully examined together (a "harmony"), or when they cannot (a "clash").

Time granularities can be used to create data visualizations to explore for periodicities, associations and anomalies. The granularities form categorical variables (ordered or unordered) which induce groupings of the observations. Assuming a numeric response variable, the resulting graphics are then displays of distributions compared across combinations of categorical variables.

The methods implemented in the open source R package `gravitas` are consistent with a tidy workflow, with probability distributions examined using the range of graphics available in `ggplot2`.

2.1 Introduction

Temporal data are available at various resolutions depending on the context. Social and economic data are often collected and reported at coarse temporal scales such as monthly, quarterly or annually. With recent advancements in technology, more and more data are recorded at much finer temporal scales. Energy consumption may be collected every half an hour, energy supply may be collected every minute, and web search data might be recorded every second. As the frequency of data increases, the number of questions about the periodicity of the observed variable also increases. For example, data collected at an hourly scale can be analyzed using coarser temporal scales such as days, months or quarters. This approach requires deconstructing time in various possible ways called time granularities (Aigner et al., 2011).

It is important to be able to navigate through all of these time granularities to have multiple perspectives on the periodicity of the observed data. This aligns with the notion of exploratory data analysis (EDA) (Tukey, 1977) which emphasizes the use of multiple perspectives on data to help formulate hypotheses before proceeding to hypothesis testing. Visualizing probability distributions conditional on one or more granularities is an indispensable tool for exploration. Analysts are expected to comprehensively explore the many ways to view and consider temporal data. However, the plethora of choices and the lack of a systematic approach to do so quickly can make the task overwhelming.

Calendar-based graphics (Wang, Cook, and Hyndman, 2020b) are useful in visualizing patterns in the weekly and monthly structure and are helpful when checking for the effects of weekends or special days. Any temporal data at sub-daily resolution can also be displayed using this type of faceting (Wickham, 2016) with days of the week, month of the year, or another sub-daily deconstruction of time. But calendar effects are not restricted to conventional day-of-week or month-of-year deconstructions. There can be many different time deconstructions, based on the calendar or on categorizations of time granularities.

Linear time granularities (such as hours, days, weeks and months) respect the linear progression of time and are non-repeating. One of the first attempts to characterize these granularities is due to Bettini et al. (1998). However, the definitions and rules defined are inadequate for describing non-linear granularities. Hence, there is a need to define some new time granularities that can be useful in visualizations. Cyclic time granularities can be circular, quasi-circular or aperiodic. Examples of circular granularities are hour of the day and day of the week; an example of a quasi-circular granularity is day of the month; examples of aperiodic granularities are public holidays and school holidays.

Time deconstructions can also be based on the hierarchical structure of time. For example, hours are nested within days, days within weeks, weeks within months, and so on. Hence, it is possible to construct single-order-up granularities such as second of the minute, or multiple-order-up granularities such as second of the hour. The lubridate package (Grolemund and Wickham, 2011) provides tools to access and manipulate common date-time objects. However, most of its accessor functions are limited to single-order-up granularities.

The motivation for this work stems from the desire to provide methods to better understand large quantities of measurements on energy usage reported by smart meters in households across Australia, and indeed many parts of the world. Smart meters currently provide half-hourly use in kWh for each household, from the time they were installed, some as early as 2012. Households are distributed geographically and have different demographic properties as well as physical properties such as the existence of solar panels, central heating or air conditioning. The behavioral patterns in households vary substantially; for example, some families use a dryer for their clothes while others hang them on a line, and some households might consist of night owls, while others are morning larks. It is common to see aggregates (see Goodwin and Dykes, 2012) of usage across households, such as half-hourly total usage by state, because energy companies need to plan for maximum loads on the network. But studying overall energy use hides the distribution of usage at finer scales, and makes it more difficult to find solutions to improve energy efficiency. We propose that the analysis of smart meter data will benefit from systematically exploring energy consumption by visualizing the probability distributions across different deconstructions of time to find regular patterns and anomalies. Although we were motivated by the smart meter example, the problem and the solutions we propose are practically relevant to any temporal data observed more than once per year. In a

broader sense, it could be even suitable for data observed by years, decades, and centuries as might be the case in weather or astronomical data.

This work provides tools for systematically exploring bivariate granularities within the tidy workflow (Wickham and Grolemund, 2016). In particular, we

- provide a formal characterization of cyclic granularities;
- facilitate manipulation of single- and multiple-order-up time granularities through cyclic calendar algebra;
- develop an approach to check the feasibility of creating plots or drawing inferences for any two cyclic granularities.

The remainder of the paper is organized as follows: Section 2.2 provides some background material on linear granularities and calendar algebra for computing different linear granularities. Section 2.3 formally characterizes different cyclic time granularities by extending the framework of linear time granularities, and introducing cyclic calendar algebra for computing cyclic time granularities. The data structure for exploring the conditional distributions of the associated time series across pairs of cyclic time granularities is discussed in Section 2.4. Section 2.5 discusses the role of different factors in constructing an informative and trustworthy visualization. Section 2.6 examines how systematic exploration can be carried out for a temporal and non-temporal application. Finally, we summarize our results and discuss possible future directions in Section 2.7.

2.2 Linear time granularities

Discrete abstractions of time such as weeks, months or holidays can be thought of as "time granularities". Time granularities are **linear** if they respect the linear progression of time. There have been several attempts to provide a framework for formally characterizing time granularities, including Bettini et al. (1998) which forms the basis of the work described here.

2.2.1 Definitions

Definition 1. *A **time domain** is a pair $(T; \leq)$ where T is a non-empty set of time instants (equivalently, moments or points) and $\leq$ is a total order on T.*

The time domain is assumed to be *discrete*, and there is unique predecessor and successor for every element in the time domain except for the first and last.

Definition 2. *The **index set**, $Z = \{z : z \in \mathbb{Z}_{\geq 0}\}$, uniquely maps the time instants to the set of non-negative integers.*

Definition 3. *A **linear granularity** is a mapping G from the index set, Z, to subsets of the time domain such that: (1) if $i < j$ and $G(i)$ and $G(j)$ are non-empty, then each element of $G(i)$ is less than all elements of $G(j)$; and (2) if $i < k < j$ and $G(i)$ and $G(j)$ are non-empty, then $G(k)$ is non-empty. Each non-empty subset $G(i)$ is called a **granule**.*

This implies that the granules in a linear granularity are non-overlapping, continuous and ordered. The indexing for each granule can also be associated with a textual representation, called the label. A discrete time model often uses a fixed smallest linear granularity named by Bettini et al. (1998) **bottom granularity**. Figure 2.1 illustrates some common linear time granularities. Here, "hour" is the bottom granularity and "day", "week", "month" and "year" are linear granularities formed by mapping the index set to subsets of the hourly time domain. If we have "hour" running from $\{0,1,\ldots,t\}$, we will have "day" running from $\{0,1,\ldots,\lfloor t/24 \rfloor\}$. These linear granularities are uni-directional and non-repeating.

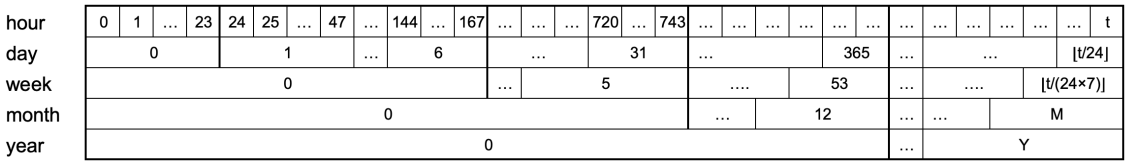

Figure 2.1: *Illustration of time domain, linear granularities and index set. Hour, day, week, month and year are linear granularities and can also be considered to be time domains. These are ordered with ordering guided by integers and hence are unidirectional and non-repeating. Hours could also be considered the index set, and a bottom granularity.*

2.2.2 Relativities

Properties of pairs of granularities fall into various categories.

Definition 4. *A linear granularity G is **finer than** a linear granularity H, denoted $G \preceq H$, if for each index i, there exists an index j such that $G(i) \subset H(j)$.*

Definition 5. *A linear granularity G **groups into** a linear granularity H, denoted $G \trianglelefteq H$, if for each index j there exists a (possibly infinite) subset S of the integers such that $H(j) = \bigcup_{i \in S} G(i)$.*

For example, both *day* $\trianglelefteq$ *week* and *day* $\preceq$ *week* hold, since every granule of *week* is the union of some set of granules of day and each day is a subset of a *week*. These definitions are not equivalent. Consider another example, where G_1 denotes "weekend" and H_1 denotes "week". Then, $G_1 \preceq H_1$, but $G_1 \not\trianglelefteq H_1$. Further, with G_2 denoting "days" and H_2 denoting "business-week", $G_2 \not\preceq H_2$, but $G_2 \trianglelefteq H_2$, since each business-week can be expressed as an union of some days, but Saturdays and Sundays are not subsets of any business-week. Moreover, with H_3 denoting "public holidays", $G_1 \not\preceq H_3$ and $G_1 \not\trianglelefteq H_3$.

Definition 6. *A granularity G is **periodic** with respect to a finite granularity H if: (1) $G \trianglelefteq H$; and (2) there exist R, $P \in \mathbb{Z}_+$, where R is less than the number of granules of H, such that for all $i \in \mathbb{Z}_{\geq 0}$, if $H(i) = \bigcup_{j \in S} G(j)$ and $H(i+R) \neq \phi$ then $H(i+R) = \bigcup_{j \in S} G(j+P)$.*

If G groups into H, it would imply that any granule $H(i)$ is the union of some granules of G; for example, $G(a_1), G(a_2), \ldots, G(a_k)$. Condition (2) in Definition 6 implies that if $H(i+R) \neq \emptyset$, then $H(i+R) = \bigcup(G(a_1+P), G(a_2+P), \ldots, G(a_k+P))$, resulting in a "periodic" pattern of the composition of H using granules of G. In this pattern, each granule of H is shifted by P granules of G. P is called the **period** (Bettini, Jajodia, and Wang, 2000).

For example, day is periodic with respect to week with $R = 1$ and $P = 7$, while (if we ignore leap years) day is periodic with respect to month with $R = 12$ and $P = 365$ as any month would consist of the same number of days across years. Since the idea of period involves a pair of granularities, we say that the pair $(day, week)$ has period 7, while the pair $(day, month)$ has a period 365 (ignoring leap years).

Granularities can also be periodic with respect to other granularities, *"except for a finite number of spans of time where they behave in an anomalous way"*; these are called **quasi-periodic** relationships (Bettini and De Sibi, 2000). In a Gregorian calendar with leap years, day groups quasi-periodically into month with the exceptions of the time domain corresponding to 29$^{\text{th}}$ February of any year.

Definition 7. *The **order** of a linear granularity is the level of coarseness associated with a linear granularity. A linear granularity G will have lower order than H if each granule of G is composed of lower number of granules of bottom granularity than each granule of H.*

With two linear granularities G and H, if G *groups into* or is *finer than* H then G is of lower order than H. For example, if the bottom granularity is hour, then granularity *day* will have lower order than *week* since each day consists of fewer hours than each week.

Granules in any granularity may be aggregated to form a coarser granularity. A system of multiple granularities in lattice structures is referred to as a **calendar** by Dyreson et al. (2000). Linear time granularities are computed through "calendar algebra" operations (Ning, Wang, and Jajodia, 2002) designed to generate new granularities recursively from the bottom granularity. For example, due to the constant length of day and week, we can derive them from hour using

$$D(j) = \lfloor H(i)/24 \rfloor, \qquad W(k) = \lfloor H(i)/(24*7) \rfloor,$$

where H, D and W denote hours, days and weeks respectively.

2.3 Cyclic time granularities

Cyclic granularities represent cyclical repetitions in time. They can be thought of as additional categorizations of time that are not linear. Cyclic granularities can be constructed from two linear granularities, that relate periodically; the resulting cycles can be either *regular* (**circular**), or *irregular* (**quasi-circular**).

2.3.1 Circular granularities

Definition 8. *A **circular granularity** $C_{B,G}$ relates linear granularity G to bottom granularity B if*

$$C_{B,G}(z) = z \bmod P(B,G) \quad \forall z \in \mathbb{Z}_{\geq 0} \tag{2.1}$$

where z denotes the index set, B groups periodically into G with regular mapping and period $P(B,G)$.

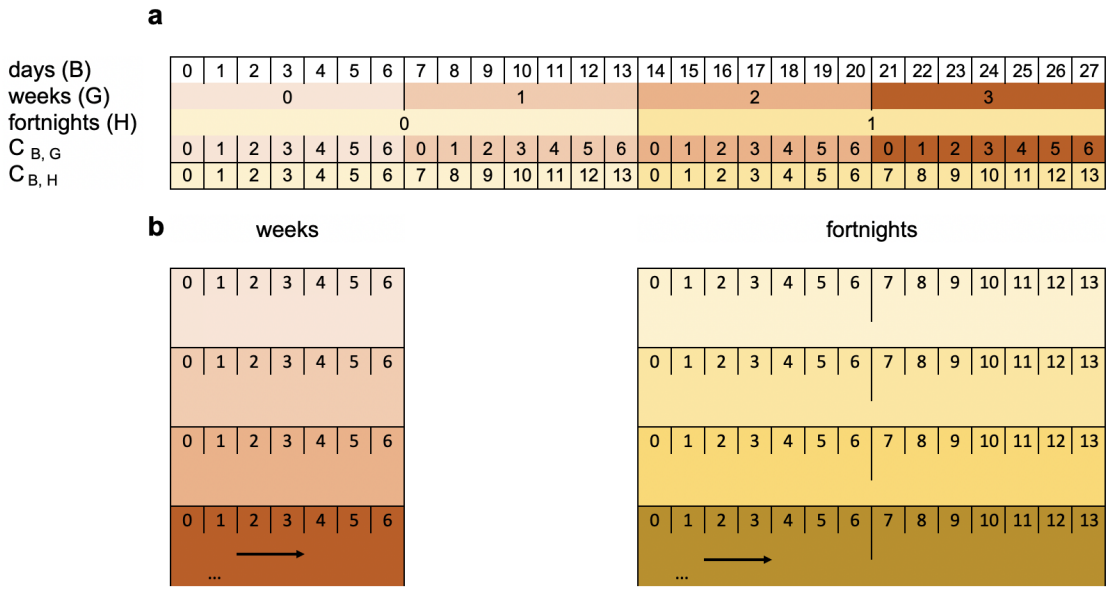

Figure 2.2: *Index sets for some linear and circular granularities (a). Circular granularities can be constructed by slicing the linear granularity into pieces and stacking them (b).*

Figure 2.2 illustrates some linear and cyclical granularities. Cyclical granularities are constructed by cutting the linear granularity into pieces, and stacking them to match the cycles (as shown in b). B, G, H (day, week, fortnight, respectively) are linear granularities. The circular granularity $C_{B,G}$ (day-of-week) is constructed from B and G, while circular granularity $C_{B,H}$ (day-of-fortnight) is constructed from B and H. These overlapping cyclical granularities share elements from the linear granularity. Each of $C_{B,G}$ and $C_{B,H}$ consist of repeated patterns $\{0,1,\ldots,6\}$ and $\{0,1,\ldots,13\}$ with $P = 7$ and $P = 14$ respectively.

Suppose L is a label mapping that defines a unique label for each index $\ell \in \{0,1,\ldots,(P-1)\}$. For example, the label mapping L for $C_{B,G}$ can be defined as

$$L : \{0,1,\ldots,6\} \longmapsto \{\text{Sunday}, \text{Monday}, \ldots, \text{Saturday}\}.$$

In general, any circular granularity relating two linear granularities can be expressed as

$$C_{G,H}(z) = \lfloor z/P(B,G) \rfloor \bmod P(G,H),$$

where H is periodic with respect to G with regular mapping and period $P(G,H)$. Table 2.1 shows several circular granularities constructed using minutes as the bottom granularity.

Table 2.1: *Examples of circular granularities with bottom granularity minutes. Circular granularity C_i relates two linear granularities, one of which groups periodically into the other with regular mapping and period P_i. Circular granularities can be expressed using modular arithmetic due to their regular mapping.*

Circular granularity	Expression	Period
minute-of-hour	$C_1 = z \bmod 60$	$P_1 = 60$
minute-of-day	$C_2 = z \bmod 60 * 24$	$P_2 = 1440$
hour-of-day	$C_3 = \lfloor z/60 \rfloor \bmod 24$	$P_3 = 24$
hour-of-week	$C_4 = \lfloor z/60 \rfloor \bmod 24 * 7$	$P_4 = 168$
day-of-week	$C_5 = \lfloor z/24 * 60 \rfloor \bmod 7$	$P_5 = 7$

2.3.2 Quasi-circular granularities

A **quasi-circular** granularity cannot be defined using modular arithmetic because of the irregular mapping. However, they are still formed with linear granularities, one of which groups periodically into the other. Table 2.2 shows some examples of quasi-circular granularities.

Table 2.2: *Examples of quasi-circular granularities relating two linear granularities with irregular mapping leading to several possible period lengths.*

Quasi-circular granularity	Possible period lengths
$Q_1 =$ day-of-month	$P_1 = 31, 30, 29, 28$
$Q_2 =$ hour-of-month	$P_2 = 24 \times 31, 24 \times 30, 24 \times 29, 24 \times 28$
$Q_3 =$ day-of-year	$P_3 = 366, 365$

Definition 9. *A **quasi-circular granularity** $Q_{B,G'}$ is formed when bottom granularity B groups periodically into linear granularity G' with irregular mapping such that the granularities are given by*

$$Q_{B,G'}(z) = z - \sum_{w=0}^{k-1} |T_{w \bmod R'}|, \quad for \quad z \in T_k, \tag{2.2}$$

where z denotes the index set, w denotes the index of G', R' is the number of granules of G' in each period of (B, G'), T_w are the sets of indices of B such that $G'(w) = \bigcup_{z \in T_w} B(z)$, and $|T_w|$ is the cardinality of set T_w.

For example, day-of-year is quasi-periodic with either 365 or 366 granules of B (days) within each granule of G' (years). The pattern repeats every 4 years (ignoring leap seconds). Hence $R' = 4$.

$Q_{B,G'}$ is a repetitive categorization of time, similar to circular granularities, except that the number of granules of B is not the same across different granules of G'.

2.3.3 Aperiodic granularities

Aperiodic linear granularities are those that cannot be specified as a periodic repetition of a pattern of granules as described in Definition 6. Aperiodic cyclic granularities capture repetitions of these aperiodic linear granularities. Examples include public holidays which repeat every year, but there is no reasonably small span of time within which their behavior remains constant. A classic example is Easter (in the Western tradition) whose dates repeat only after 5.7 million years (Reingold and Dershowitz, 2018). In Australia, if a standard public holiday falls on a weekend, a substitute public holiday will sometimes be observed on the first non-weekend day (usually Monday) after the weekend. Examples of aperiodic granularity may also include school holidays or a scheduled event. All of these are recurring events, but with non-periodic patterns. Consequently, P_i (as given in Table 2.2) are essentially infinite for aperiodic granularities.

Definition 10. *An **aperiodic cyclic granularity** is formed when bottom granularity B groups into an aperiodic linear granularity M such that the granularities are given by*

$$A_{B,M}(z) = \begin{cases} i, & for \quad z \in T_{i_j} \\ 0 & otherwise, \end{cases} \tag{2.3}$$

where z denotes the index set, T_{i_j} are the sets of indices of B describing aperiodic linear granularities M_i such that $M_i(j) = \bigcup_{z \in T_{i_j}} B(z)$, and $M = \bigcup_{i=1}^{n} M_i$, n being the number of aperiodic linear granularities in consideration.

For example, consider the school semester shown in Figure 2.3. Let the linear granularities M_1 and M_2 denote the teaching and non-teaching stages of the semester respectively. Both M_1, M_2 and $M = M_1 \bigcup M_2$ denoting the "stages" of the semester are aperiodic with respect to days (B) or weeks (G). Hence $A_{B,M}$ denoting day-of-the-stage would be an aperiodic cyclic granularity because the placement of the semester within a year would vary across years. Here, $Q_{B',M}$ denoting semester-day-of-the-stage would be a quasi-circular granularity since the distribution of semester

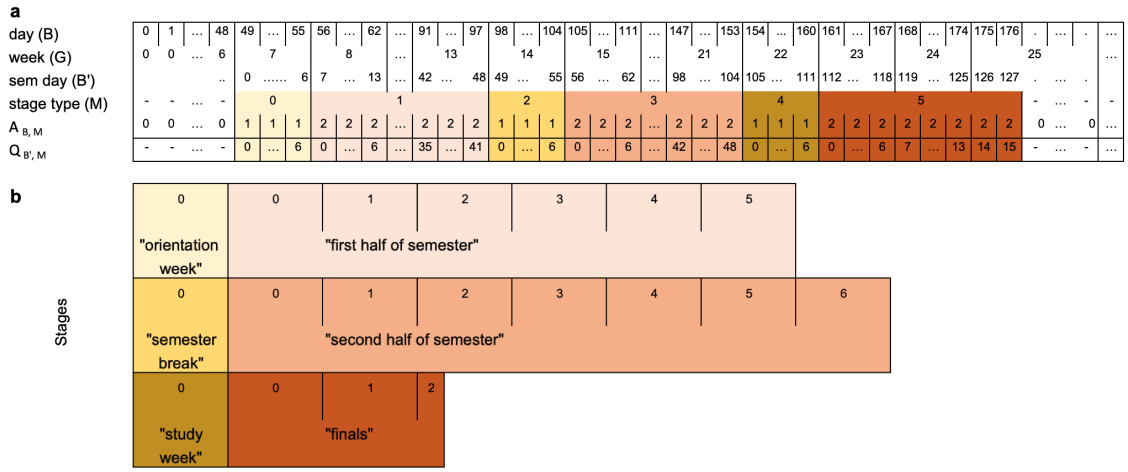

Figure 2.3: *Quasi-circular and aperiodic cyclic granularities illustrated by linear (a) and stacked-displays (b) progression of time. The linear display shows the granularities days (B), weeks (G), semester days (B'), and stages of a semester (M) indexed over a linear representation of time. The granules of B' are only defined for days when the semester is running. Here a semester spans 18 weeks and 2 days, and consists of 6 stages. It starts with a week of orientation, followed by an in-session period (6 weeks), a break (1 week), the second half of semester (7 weeks), a 1-week study break before final exams, which spans the next 16 days. This distribution of semester days remains relatively similar for every semester. $Q_{B',M}$ with $P = 128$ is a quasi-circular granularity with repeating patterns, while $A_{B,M}$ is an aperiodic cyclic granularity as the placement of the semester within a year varies from year to year with no fixed start and end dates.*

days within a semester is assumed to remain constant over years. Here semester-day is denoted by "sem day" (B') and its granules are only defined for the span of the semesters.

2.3.4 Relativities

The hierarchical structure of time creates a natural nested ordering which can be used in the computation of relative pairs of granularities.

Definition 11. *The nested ordering of linear granularities can be organized into a* ***hierarchy table****, denoted as $H_n : (G, C, K)$, which arranges them from lowest to highest in order. It shows how the n granularities relate through K, and how the cyclic granularities, C, can be defined relative to the linear granularities. Let G_ℓ and G_m represent the linear granularity of order ℓ and m respectively with $\ell < m$. Then $K \equiv P(\ell, m)$ represents the period length of the grouping (G_ℓ, G_m), if C_{G_ℓ,G_m} is a circular granularity and $K \equiv k(\ell, m)$ represents the operation to obtain G_m from G_ℓ, if C_{G_ℓ,G_m} is quasi-circular.*

Table 2.3: *Hierarchy table for Mayan calendar with circular single-order-up granularities.*

linear (G)	single-order-up cyclic (C)	period length/conversion operator (K)
kin	kin-of-uinal	20
uinal	uinal-of-tun	18
tun	tun-of-katun	20
katun	katun-of-baktun	20
baktun	1	1

For example, Table 2.3 shows the hierarchy table for the Mayan calendar. In the Mayan calendar, one day was referred to as a kin and the calendar was structured such that 1 kin = 1 day; 1 uinal = 20 kin; 1 tun = 18 uinal (about a year); 1 katun = 20 tun (20 years) and 1 baktun = 20 katun.

Like most calendars, the Mayan calendar used the day as the basic unit of time (Reingold and Dershowitz, 2018). The structuring of larger units, weeks, months, years and cycle of years, though, varies substantially between calendars. For example, the French revolutionary calendar divided each day into 10 "hours", each "hour" into 100 "minutes" and each "minute" into 100 "seconds", the duration of which is 0.864 common seconds. Nevertheless, for any calendar, a hierarchy table can be defined. Note that it is not always possible to organize an aperiodic linear granularity in a hierarchy table. Hence, we assume that the hierarchy table consists of periodic linear granularities only, and that the cyclic granularity $C_{G(\ell),G(m)}$ is either circular or quasi-circular.

Definition 12. *The hierarchy table contains* ***multiple-order-up*** *granularities which are cyclic granularities that are nested within multiple levels. A* ***single-order-up*** *is a cyclic granularity which is nested within a single level. It is a special case of multiple-order-up granularity.*

In the Mayan calendar (Table 2.3), kin-of-tun or kin-of-baktun are examples of multiple-order-up granularities and single-order-up granularities are kin-of-uinal, uinal-of-tun etc.

2.3.5 Computation

Following the calendar algebra of Ning, Wang, and Jajodia (2002) for linear granularities, we can define cyclic calendar algebra to compute cyclic granularities. Cyclic calendar algebra comprises two kinds of operations: (1) **single-to-multiple** (the calculation of *multiple-order-up* cyclic granularities from *single-order-up* cyclic granularities) and (2) **multiple-to-single** (the reverse).

Single-to-multiple order-up

Methods to obtain multiple-order-up granularity will depend on whether the hierarchy consists of all circular single-order-up granularities or a mix of circular and quasi-circular single-order-up granularities. Circular single-order-up granularities can be used recursively to obtain a multiple-order-up circular granularity using

$$C_{G_\ell,G_m}(z) = \sum_{i=0}^{m-\ell-1} P(\ell,\ell+i) C_{G_{\ell+i},G_{\ell+i+1}}(z), \tag{2.4}$$

where $\ell < m-1$ and $P(i,i) = 1$ for $i = 0,1,\ldots,m-\ell-1$, and $C_{B,G}(z) = z \bmod P(B,G)$ as per Equation (2.1).

For example, the multiple-order-up granularity $C_{\text{uinal,katun}}$ for the Mayan calendar could be obtained using

$$\begin{aligned} C_{\text{uinal,baktun}}(z) &= C_{\text{uinal,tun}}(z) + P(\text{uinal,tun}) C_{\text{tun,katun}}(z) + P(\text{uinal,katun}) C_{\text{katun,baktun}}(z) \\ &= C_{\text{uinal,tun}}(z) + 18 \times C_{\text{tun,katun}}(z) + 18 \times 20 \times C_{\text{katun,baktun}}(z) \end{aligned}$$

where z is the index of the bottom granularity *kin*.

Now consider the case where there is one quasi-circular single order-up granularity in the hierarchy table while computing a multiple-order-up quasi-circular granularity. Any multiple-order-up quasi-circular granularity $C_{\ell,m}(z)$ could then be obtained as a discrete combination of circular and quasi-circular granularities.

Depending on the order of the combination, two different approaches need to be employed leading to the following cases:

- $C_{G_\ell,G_{m'}}$ is circular and $C_{G_{m'},G_m}$ is quasi-circular

$$C_{G_\ell,G_m}(z) = C_{G_\ell,G_{m'}}(z) + P(\ell,m') C_{G_{m'},G_m}(z) \tag{2.5}$$

Table 2.4: *Hierarchy table for the Gregorian calendar with both circular and quasi-circular single-order-up granularities.*

linear (G)	single-order-up cyclic (C)	period length/conversion operator (K)
minute	minute-of-hour	60
hour	hour-of-day	24
day	day-of-month	k(day, month)
month	month-of-year	12
year	1	1

- $C_{G_\ell,G_{m'}}$ is quasi-circular and $C_{G_{m'},G_m}$ is circular

$$C_{G_\ell,G_m}(z) = C_{G_\ell,G_{m'}}(z) + \sum_{w=0}^{C_{G_{m'},G_m}(z)-1} (|T_w|) \tag{2.6}$$

where T_w is such that $G_{m'}(w) = \bigcup_{z\in T_w} G_\ell$ and $|T_w|$ is the cardinality of set T_w.

For example, the Gregorian calendar (Table 2.4) has day-of-month as a single-order-up quasi-circular granularity, with the other granularities being circular. Using Equations (2.5) and (2.6), we then have:

$$C_{hour,month}(z) = C_{hour,day}(z) + P(hour,day) * C_{day,month}(z)$$

$$C_{day,year}(z) = C_{day,month}(z) + \sum_{w=0}^{C_{month,year}(z)-1} (|T_w|),$$

where T_w is such that $month(w) = \bigcup_{z\in T_w} day(z)$.

Multiple-to-single order-up

Similar to single-to-multiple operations, multiple-to-single operations involve different approaches for all circular single-order-up granularities and a mix of circular and quasi-circular single-order-up granularities in the hierarchy. For a hierarchy table $H_n : (G,C,K)$ with only circular single-order-up granularities and $\ell_1,\ell_2,m_1,m_2 \in 1,2,\ldots,n$ and $\ell_2 < \ell_1$ and $m_2 > m_1$, multiple-order-up granularities can be obtained using Equation (2.7).

$$C_{G_{\ell_1},G_{m_1}}(z) = \lfloor C_{G_{\ell_2},G_{m_2}}(z)/P(\ell_2,\ell_1) \rfloor \bmod P(\ell_1,m_1) \tag{2.7}$$

For example, in the Mayan Calendar, it is possible to compute the single-order-up granularity tun-of-katun from uinal-of-baktun, since $C_{tun,katun}(z) = \lfloor C_{uinal,baktun}(z)/18 \rfloor \bmod 20$.

Table 2.5: *The data structure for exploring periodicities in data by including cyclic granularities in the tsibble structure with index, key and measured variables.*

index	key	measurements	C_1	C_2	$\cdots$	C_{N_C}

Multiple order-up quasi-circular granularities

Single-order-up quasi-circular granularity can be obtained from multiple-order-up quasi-circular granularity and single/multiple-order-up circular granularity using Equations (2.5) and (2.6).

2.4 Data structure

Effective exploration and visualization benefit from well-organized data structures. Wang, Cook, and Hyndman (2020a) introduced the tidy "tsibble" data structure to support exploration and modeling of temporal data. This forms the basis of the structure for cyclic granularities. A tsibble comprises an index, optional key(s), and measured variables. An index is a variable with inherent ordering from past to present and a key is a set of variables that define observational units over time. A linear granularity is a mapping of the index set to subsets of the time domain. For example, if the index of a tsibble is days, then a linear granularity might be weeks, months or years. A bottom granularity is represented by the index of the tsibble.

All cyclic granularities can be expressed in terms of the index set. Table 2.5 shows the tsibble structure (index, key, measurements) augmented by columns of cyclic granularities. The total number of cyclic granularities depends on the number of linear granularities considered in the hierarchy table and the presence of any aperiodic cyclic granularities. For example, if we have n periodic linear granularities in the hierarchy table, then $n(n-1)/2$ circular or quasi-circular cyclic granularities can be constructed. Let N_C be the total number of contextual circular, quasi-circular and aperiodic cyclic granularities that can originate from the underlying periodic and aperiodic linear granularities. Simultaneously encoding more than a few of these cyclic granularities when visualizing the data overwhelms human comprehension. Instead, we focus on visualizing the data split by pairs of cyclic granularities (C_i, C_j). Data sets of the form $<C_i, C_j, v>$ then allow exploration and analysis of the measured variable v.

2.4.1 Harmonies and clashes

The way granularities are related is important when we consider data visualizations. Consider two cyclic granularities C_i and C_j, such that C_i maps index set to a set $\{A_k \mid k = 1,\ldots,K\}$ and C_j maps index set to a set $\{B_\ell \mid \ell = 1,\ldots,L\}$. Here, A_k and B_ℓ are the levels/categories corresponding to C_i and C_j respectively. Let $S_{k\ell}$ be a subset of the index set such that for all $s \in S_{k\ell}$, $C_i(s) = A_k$ and $C_j(s) = B_\ell$. There are KL such data subsets, one for each combination of levels (A_k, B_ℓ). Some of these sets may be empty due to the structure of the calendar, or because of the duration and location of events in a calendar.

Definition 13. *A **clash** is a pair of cyclic granularities that contains empty combinations of categories.*

Definition 14. *A **harmony** is a pair of cyclic granularities that does not contain any empty combinations of its categories.*

Structurally empty combinations can arise due to the structure of the calendar or hierarchy. For example, let C_i be day-of-month with 31 levels and C_j be day-of-year with 365 levels. There will be $31 \times 365 = 11315$ sets $S_{k\ell}$ corresponding to possible combinations of C_i and C_j. Many of these are empty. For example, $S_{1,5}$ is empty because the first day of the month can never correspond to the fifth day of the year. Hence the pair (day-of-month, day-of-year) is a clash.

Event-driven empty combinations arise due to differences in event location or duration in a calendar. For example, let C_i be day-of-week with 7 levels and C_j be working-day/non-working-day with 2 levels. While potentially all of these 14 sets $S_{k\ell}$ can be non-empty (it is possible to have a public holiday on any day-of-week), in practice many of these will probably have very few observations. For example, there are few (if any) public holidays on Wednesdays or Thursdays in any given year in Melbourne, Australia.

An example of harmony is where C_i and C_j denote day-of-week and month-of-year respectively. So C_i will have 7 levels while C_j will have 12 levels, giving $12 \times 7 = 84$ sets $S_{k\ell}$. All of these are non-empty because every day-of-week can occur in every month. Hence, the pair (day-of-week, month-of-year) is a harmony.

2.4.2 Near-clashes

Suppose C_i denotes day-of-year and C_j denotes day-of-week. While any day of the week can occur on any day of the year, some combinations will be very rare. For example, the 366th day of the year will only coincide with a Wednesday approximately every 28 years on average. We refer to these as "near-clashes".

2.5 Visualization

The purpose is to visualize the distribution of the continuous variable (v) conditional on the values of two granularities, C_i and C_j. Since C_i and C_j are factors or categorical variables, data subsets corresponding to each combination of their levels form a subgroup and the visualization amounts to having displays of distributions for different subgroups. The response variable (v) is plotted on the y-axis and the levels of $C_i(C_j)$ on the x-axis, conditional on the levels of $C_j(C_i)$. This means, carrying out the same plot corresponding to each level of the conditioning variable. This is consistent with the widely used grammar of graphics which is a framework to construct statistical graphics by relating the data space to the graphic space (Wilkinson, 1999; Wickham, 2016).

2.5.1 Data summarization

There are several ways to summarize the distribution of a data set such as estimating the empirical distribution or density of the data, or computing a few quantiles or other statistics. This estimation or summarization could be potentially misleading if it is performed on rarely occurring categories (Section 2.4.2). Even when there are no rarely occurring events, the number of observations may vary greatly within or across each facet, due to missing observations or uneven locations of events in the time domain. In such cases, data summarization should be used with caution as sample sizes will directly affect the accuracy of the estimated quantities being displayed.

2.5.2 Display choices for univariate distributions

The basic plot choice for our data structure is one that can display distributions. For displaying the distribution of a continuous univariate variable, many options are available. Displays based on descriptive statistics include boxplots (Tukey, 1977) and its variants such as notched boxplots

(McGill, Tukey, and Larsen, 1978) or other variations as mentioned in Wickham and Stryjewski (2012). They also include line or area quantile plots which can display any quantiles and not only quartiles like in a boxplot. Plots based on kernel density estimates include violin plots (Hintze and Nelson, 1998), summary plots (Potter et al., 2010), ridge line plots (Wilke, 2020), and highest density region (HDR) plots (Hyndman, 1996). The less commonly used letter-value plots (Hofmann, Wickham, and Kafadar, 2017) is midway between boxplots and density plots. Letter values are order statistics with specific depths; for example, the median (M) is a letter value that divides the data set into halves. Each of the next letter values splits the remaining parts into two separate regions so that the fourths (F), eighths (E), sixteenths (D), etc. are obtained. They are useful for displaying the distributions beyond the quartiles especially for large data, where boxplots mislabel data points as outliers. One of the best approaches in exploratory data analysis is to draw a variety of plots to reveal information while keeping in mind the drawbacks and benefits of each of the plot choices. For example, boxplots obscure multimodality, and interpretation of density estimates and histograms may change depending on the bandwidth and binwidths respectively. In R package `gravitas` (Gupta et al., 2020), boxplots, violin, ridge, letter-value, line and area quantile plots are implemented, but it is potentially possible to use any plots which can display the distribution of the data.

2.5.3 Comparison across sub-groups induced by conditioning

Levels

The levels of cyclic granularities affect plotting choices since space and resolution may be problematic with too many levels. A potential approach could be to categorize the number of levels as low/medium/high/very high for each cyclic granularity and define some criteria based on human cognitive power, available display size and the aesthetic mappings. Default values for these categorizations could be chosen based on levels of common temporal granularities like days of the month, days of the fortnight, or days of the week.

Synergy of cyclic granularities

The synergy of the two cyclic granularities will affect plotting choices for exploratory analysis. Cyclic granularities that form clashes (Section 2.4.1) or near-clashes lead to potentially ineffective graphs. Harmonies tend to be more useful for exploring patterns. Figure 2.4a shows the distribution of half-hourly electricity consumption through letter value plots across months of the year conditional on quarters of the year. This plot does not work because quarter-of-year clashes with month-of-year, leading to empty subsets. For example, the first quarter never corresponds to December.

Conditioning variable

When C_i is mapped to the x position and C_j to facets, then the A_k levels are juxtaposed and each B_ℓ represents a group/facet. Gestalt theory (Wertheimer (1938)) suggests that when items are placed in close proximity, people assume that they are in the same group because they are close to one another and apart from other groups. Hence, in this case, the A_k's are compared against each other within each group. With the mapping of C_i and C_j reversed, the emphasis will shift to comparing B_ℓ levels rather than A_k levels. For example, Figure 2.4b shows the letter value plot across weekday/weekend partitioned by quarters of the year and Figure 2.4c shows the same two cyclic granularities with their mapping reversed. Figure 2.4b helps us to compare weekday and weekend within each quarter and Figure 2.4c helps to compare quarters within weekend and weekday.

2.6 Applications

2.6.1 Smart meter data of Australia

Smart meters provide large quantities of measurements on energy usage for households across Australia. One of the customer trials (Department of the Environment and Energy, 2018) conducted as part of the Smart Grid Smart City project in Newcastle and parts of Sydney provides customer level data on energy consumption for every half hour from February 2012 to March 2014. We can use this data set to visualize the distribution of energy consumption across different cyclic granularities in a systematic way to identify different behavioral patterns.

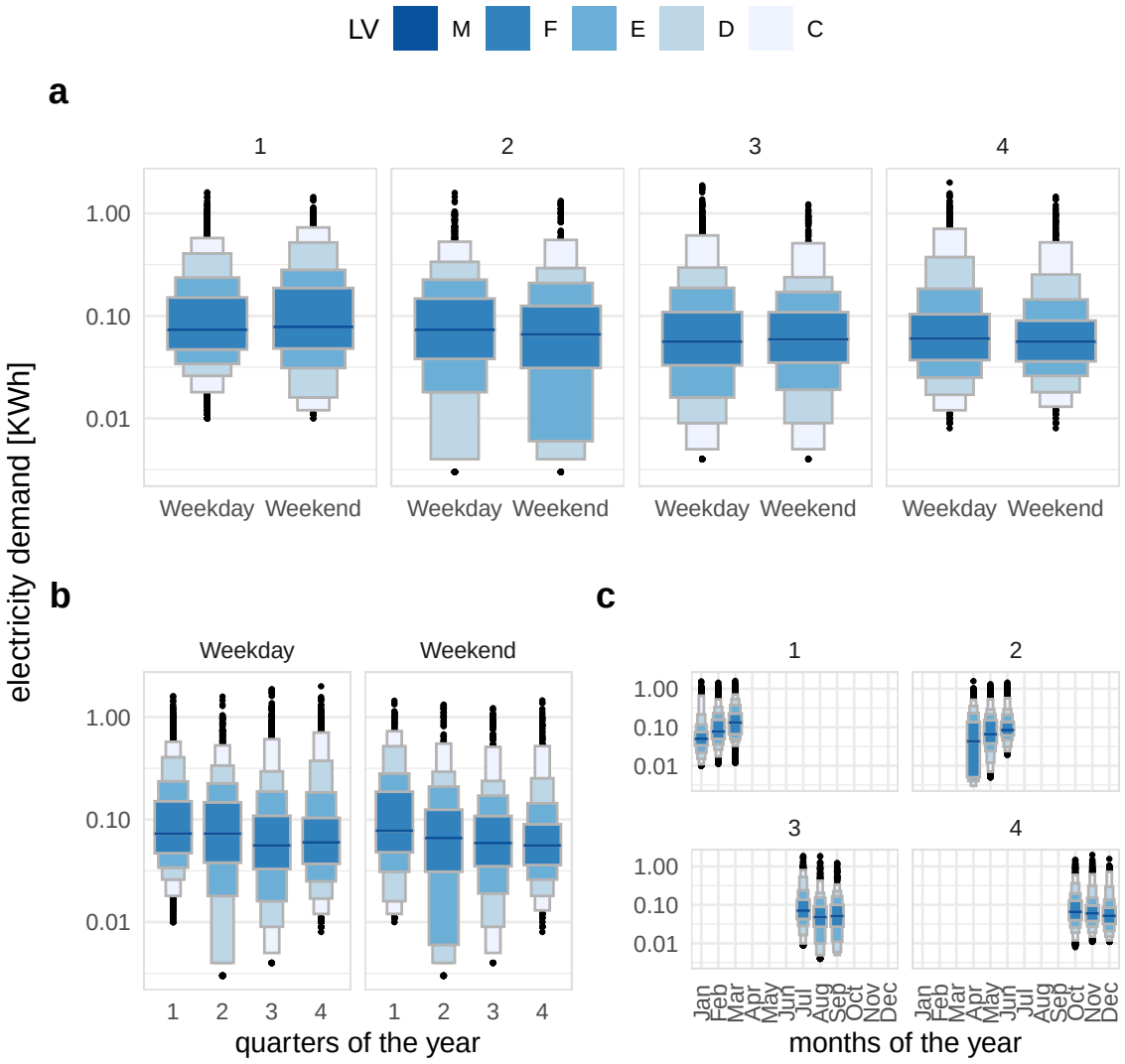

Figure 2.4: *Distribution of energy consumption displayed on a logarithmic scale as letter value plots, illustrating harmonies and clashes, and how mappings change emphasis:* ***a*** *weekday/weekend faceted by quarter-of-year produces a harmony,* ***b*** *quarter-of-year faceted by weekday/weekend produces a harmony,* ***c*** *month-of-year faceted by quarter-of-year produces a clash, as indicated by the empty sets and white space. Placement within a facet should be done for primary comparisons. For example, arrangement in* ***a*** *makes it easier to compare across weekday type (x-axis) within a quarter (facet). It can be seen that in quarter 2, more mass occupied the lower tail on the weekends (letter value E corresponding to tail area 1/8) relative to that of the weekdays (letter value D 1/16), which corresponds to more days with lower energy use in this period.*

Cyclic granularities search and computation

The tsibble object `smart_meter10` from R package `gravitas` (Gupta et al., 2020) includes the variables `reading_datetime`, `customer_id` and `general_supply_kwh` denoting the index, key and measured variable respectively. The interval of this tsibble is 30 minutes.

To identify the available cyclic time granularities, consider the conventional time deconstructions for a Gregorian calendar that can be formed from the 30-minute time index: half-hour, hour, day, week, month, quarter, half-year, year. In this example, we will consider the granularities hour, day, week and month giving six cyclic granularities "hour_day", "hour_week", "hour_month", "day_week", "day_month" and "week_month", read as "hour of the day", etc. To these, we add day-type ("wknd_wday") to capture weekend and weekday behavior. Now that we have a list of cyclic granularities to look at, we can compute them using the results in Section 2.3.4.

Screening and visualizing harmonies

Using these seven cyclic granularities, we want to explore patterns of energy behavior. Each of these seven cyclic granularities can either be mapped to the x-axis or to facets. Choosing 2 of the possible 7 granularities, gives ${}^7P_2 = 42$ candidates for visualization. Harmonies can be identified among those 42 possibilities to narrow the search. Table 2.6 shows 16 harmony pairs after removing clashes and any cyclic granularities with more than 31 levels, as effective exploration becomes difficult with many levels (Section 2.5.3).

A few harmony pairs are displayed in Figure 2.5 to illustrate the impact of different distribution plots and reverse mapping. For each of Figure 2.5b and c, C_i denotes day-type (weekday/weekend) and C_j is hour-of-day. The geometry used for displaying the distribution is chosen as area-quantiles and violins in Figure 2.5b and c respectively. Figure 2.5a shows the reverse mapping of C_i and C_j with C_i denoting hour-of-day and C_j denoting day-type with distribution geometrically displayed as boxplots.

In Figure 2.5b, the black line is the median, the purple (narrow) band covers the 25th to 75th percentile, the orange (middle) band covers the 10th to 90th percentile, and the green (broad) band covers the 1st to 99th percentile. The first facet represents the weekday behavior while the second facet displays the weekend behavior; energy consumption across each hour of the day is

Table 2.6: *Harmonies with pairs of cyclic granularities, one mapped to facets and the other to the x-axis. Only 16 of 42 possible combinations of cyclic granularities are harmony pairs.*

facet variable	x-axis variable	facet levels	x-axis levels
day_week	hour_day	7	24
day_month	hour_day	31	24
week_month	hour_day	5	24
wknd_wday	hour_day	2	24
hour_day	day_week	24	7
day_month	day_week	31	7
week_month	day_week	5	7
hour_day	day_month	24	31
day_week	day_month	7	31
wknd_wday	day_month	2	31
hour_day	week_month	24	5
day_week	week_month	7	5
wknd_wday	week_month	2	5
hour_day	wknd_wday	24	2
day_month	wknd_wday	31	2
week_month	wknd_wday	5	2

shown inside each facet. The energy consumption is extremely skewed with the 1st, 10th and 25th percentile lying relatively close whereas 75th, 90th and 99th lying further away from each other. This is common across both weekdays and weekends. For the first few hours on weekdays, median energy consumption starts and continues to be higher for longer compared to weekends.

The same data is shown using violin plots instead of quantile plots in Figure 2.5c. There is bimodality in the early hours of the day for weekdays and weekends. If we visualize the same data with reverse mapping of the cyclic granularities (Figure 2.5a), then the natural tendency would be to compare weekend and weekday behavior within each hour and not across hours. Then it can be seen that median energy consumption for the early morning hours is higher for weekdays than weekends. Also, outliers are more prominent in the later hours of the day. All of these indicate that looking at different distribution geometry or changing the mapping can shed light on different aspects of energy behavior for the same sample.

2.6.2 T20 cricket data of Indian Premier League

Our proposed approach can be generalized to other hierarchical granularities where there is an underlying ordered index. We illustrate this with data from the sport cricket. Cricket is played

Table 2.7: *Hierarchy table for cricket where overs are nested within an innings, innings nested within a match and matches within a season.*

linear (G)	single-order-up cyclic (C)	period length/conversion operator (K)
over	over-of-inning	20
inning	inning-of-match	2
match	match-of-season	k(match, season)
season	1	1

with two teams of 11 players each, with each team taking turns batting and fielding. This is similar to baseball, wherein the *batsman* and *bowler* in cricket are analogous to a batter and pitcher in baseball. A *wicket* is a structure with three sticks, stuck into the ground at the end of the cricket pitch behind the batsman. One player from the fielding team acts as the bowler, while another takes up the role of the *wicket-keeper* (similar to a catcher in baseball). The bowler tries to hit the wicket with a *ball*, and the batsman defends the wicket using a *bat*. At any one time, two of the batting team and all of the fielding team are on the field. The batting team aims to score as many *runs* as possible, while the fielding team aims to successively *dismiss* 10 players from the batting team. The team with the highest number of runs wins the match.

Cricket is played in various formats and Twenty20 cricket (T20) is a shortened format, where the two teams have a single *innings* each, which is restricted to a maximum of 20 *overs*. An over will consist of 6 balls (with some exceptions). A single *match* will consist of 2 innings and a *season* consists of several matches. Although there is no conventional time component in cricket, each ball can be thought to represent an ordering over the course of the game. Then, we can conceive a hierarchy where the ball is nested within overs, overs nested within innings, innings within matches, and matches within seasons. Cyclic granularities can be constructed using this hierarchy. Example granularities include ball of the over, over of the innings, and ball of the innings. The hierarchy table is given in Table 2.7. Although most of these cyclic granularities are circular by the design of the hierarchy, in practice some granularities are aperiodic. For example, most overs will consist of 6 balls, but there are exceptions due to wide balls, no-balls, or when an innings finishes before the over finishes. Thus, the cyclic granularity ball-of-over may be aperiodic.

The Indian Premier League (IPL) is a professional T20 cricket league in India contested by eight teams representing eight different cities in India. The IPL ball-by-ball data is provided in the

`cricket` data set in the `gravitas` package for a sample of 214 matches spanning 9 seasons (2008 to 2016).

Many interesting questions could be addressed with the `cricket` data set. For example, does the distribution of total runs depend on whether a team bats in the first or second innings? The Mumbai Indians (MI) and Chennai Super Kings (CSK) appeared in the final playoffs from 2010 to 2015. Using data from these two teams, it can be observed (Figure 2.6a) that for the team batting in the first innings there is an upward trend of runs per over, while there is no clear upward trend in the median and quartile deviation of runs for the team batting in the second innings after the first few overs. This suggests that players feel mounting pressure to score more runs as they approach the end of the first innings, while teams batting second have a set target in mind and are not subjected to such mounting pressure and therefore may adopt a more conservative run-scoring strategy.

Another question that can be addressed is if good fielding or bowling (defending) in the previous over affects the scoring rate in the subsequent over. To measure the defending quality, we use an indicator function on dismissals (1 if there was at least one wicket in the previous over, 0 otherwise). The scoring rate is measured by runs per over. Figure 2.6b shows that no dismissals in the previous over leads to a higher median and quartile spread of runs per over compared to the case when there has been at least one dismissal in the previous over. This seems to be unaffected by the over of the innings (the faceting variable). This might be because the new batsman needs to "play himself in" or the dismissals lead the (not-dismissed) batsman to adopt a more defensive play style. Run rates will also vary depending on which player is facing the next over and when the wicket falls in the previous over.

Here, wickets per over is an aperiodic cyclic granularity, so it does not appear in the hierarchy table. These are similar to holidays or special events in temporal data.

2.7 Discussion

Exploratory data analysis involves many iterations of finding and summarizing patterns. With temporal data available at ever finer scales, exploring periodicity can become overwhelming with so many possible granularities to explore. This work provides tools to classify and compute possible cyclic granularities from an ordered (usually temporal) index. We also provide a framework to

systematically explore the distribution of a univariate variable conditional on two cyclic time granularities using visualizations based on the synergy and levels of the cyclic granularities.

The `gravitas` package provides very general tools to compute and manipulate cyclic granularities, and to generate plots displaying distributions conditional on those granularities.

A missing piece in the package `gravitas` is the computation of cyclic aperiodic granularities which would require computing aperiodic linear granularities first. A few R packages including `almanac` (Vaughan, 2020) and `gs` (Laird-Smith, 2020) provide the tools to create recurring aperiodic events. These functions can be used with the `gravitas` package to accommodate aperiodic cyclic granularities.

We propose producing plots based on pairs of cyclic granularities that form harmonies rather than clashes or near-clashes. A future direction of work could be to further refine the selection of appropriate pairs of granularities by identifying those for which the differences between the displayed distributions is greatest and rating these selected harmony pairs in order of importance for exploration.

Acknowledgments

The Australian authors thank the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) for supporting this research. Thanks to Data61 CSIRO for partially funding Sayani's research and Dr. Peter Toscas for providing useful inputs on improving the analysis of the smart meter application. We would also like to thank Nicholas Spyrison for many useful discussions, sketching figures and feedback on the manuscript. The package `gravitas` was built during the Google Summer of Code, 2019. More details about the package can be found at sayani07.github.io/gravitas. The Github repository, github.com/Sayani07/paper-gravitas, contains all materials required to reproduce this article and the code is also available online in the supplemental materials. This article was created with `knitr` (Xie, 2015, 2020) and `rmarkdown` (Xie, Allaire, and Grolemund, 2018; Allaire et al., 2020).

Supplementary materials

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

R-package: The ideas presented in this article have been implemented in the open-source R (R Core Team, 2020) package `gravitas` (Gupta et al., 2020), available from CRAN. The R-package facilitates manipulation of single and multiple-order-up time granularities through cyclic calendar algebra, checks feasibility of creating plots or drawing inferences for any two cyclic granularities by providing list of harmonies, and recommends possible visual summaries through factors described in the article. Version 0.1.3 of the package was used for the results presented in the article and is available on Github (`https://github.com/Sayani07/gravitas`).

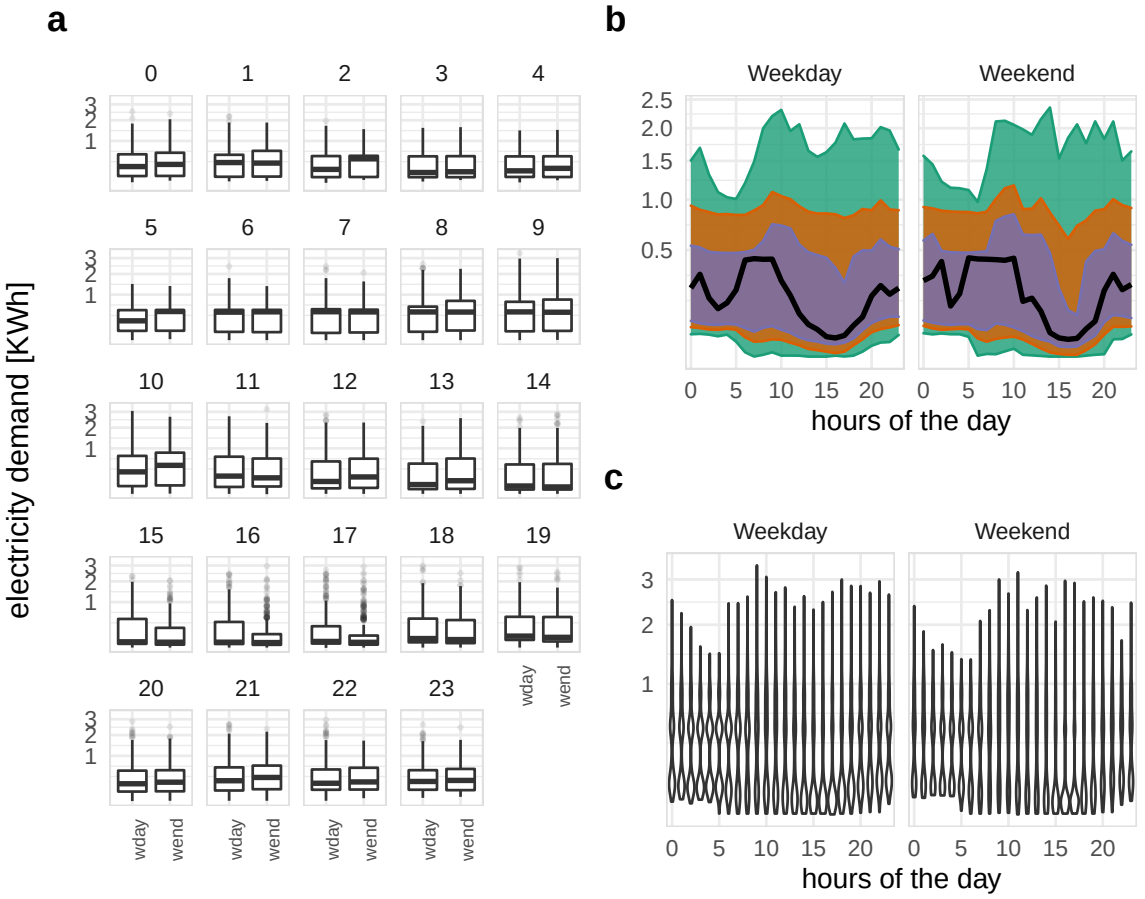

Figure 2.5: *Energy consumption of a single customer shown with different distribution displays, and granularity arrangements: hour of the day; and weekday/weekend.* ***a*** *The side-by-side boxplots make the comparison between day types easier, and suggest that there is generally lower energy use on the weekend. Interestingly, this is the opposite to what might be expected. Plots* ***b***, ***c*** *examine the temporal trend of consumption over the course of a day, separately for the type of day. The area quantile emphasizes time, and indicates that median consumption shows prolonged high usage in the morning on weekdays. The violin plot emphasizes subtler distributional differences across hours: morning use is bimodal.*

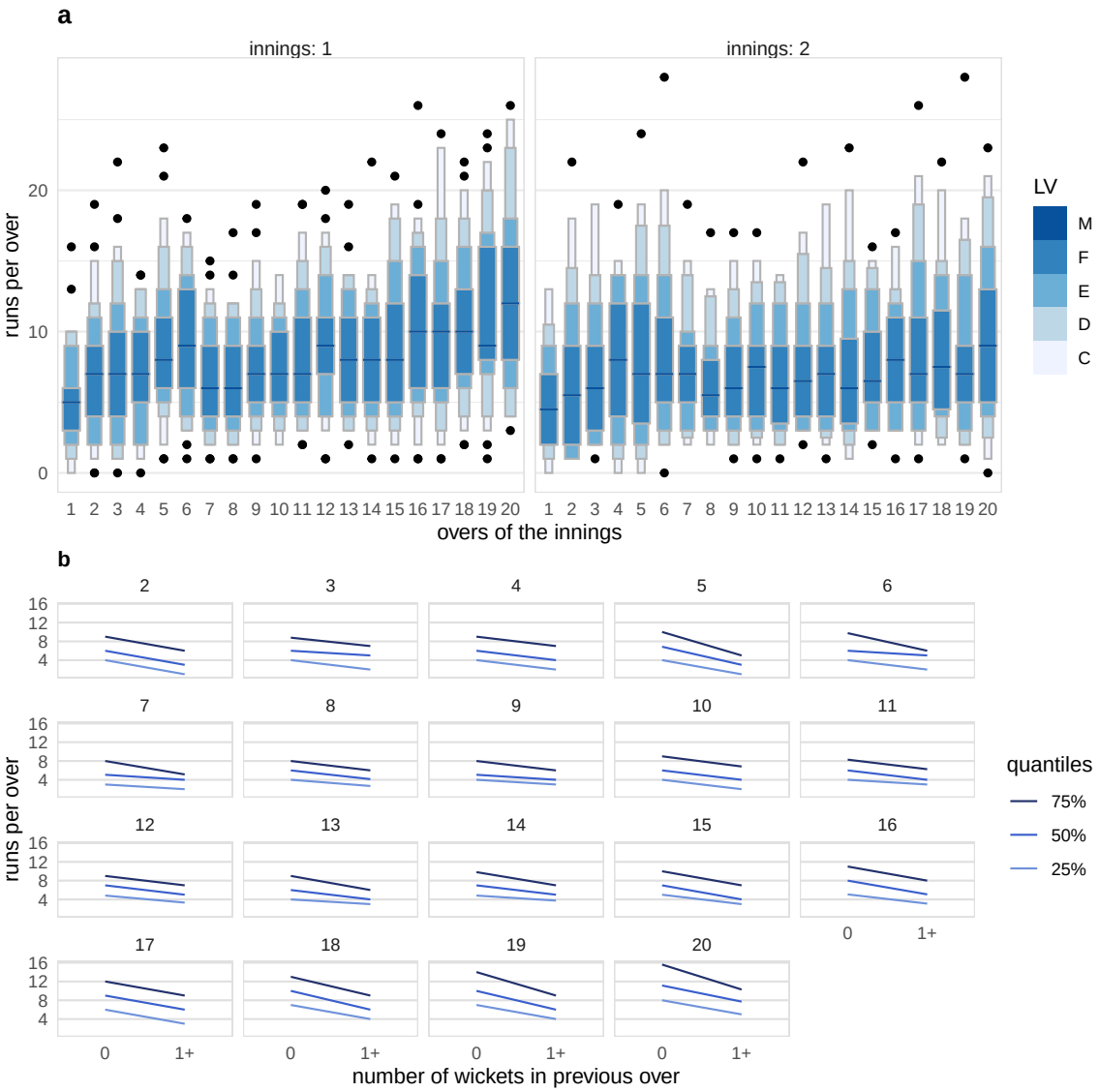

Figure 2.6: *Examining distribution of runs per innings, overs of the innings and number of wickets in previous innings. Plot **a** displays distribution using letter value plots. A gradual upward trend in runs per over can be seen in innings 1, which is not present in innings 2. Plot **b** shows quantile plots of runs per over across an indicator of wickets in the previous over, faceted by current over. When a wicket occurred in the previous over, the runs per over tends to be lower throughout the innings.*

Chapter 3

Detecting distributional differences between temporal granularities for exploratory time series analysis

Cyclic temporal granularities, which are temporal deconstructions of a time period into units such as hour-of-the-day, work-day/weekend, can be useful for measuring repetitive patterns in large univariate time series data. The granularities feed new approaches to exploring time series data. One use is to take pairs of granularities, and make plots of response values across the categories induced by the temporal deconstruction. However, when there are many granularities that can be constructed for a time period, there will also be too many possible displays to decide which might be the more interesting to display. This work proposes a new distance metric to screen and rank the possible granularities, and hence choose the most interesting ones to plot. The distance measure is computed for a single or pairs of cyclic granularities that can be compared across different cyclic granularities and also on a collection of time series. The methods are implemented in the open-source R package `hakear`.

3.1 Introduction

Cyclic temporal granularities (Bettini et al., 1998; Gupta et al., 2021) are temporal deconstructions that define cyclic repetitions in time, e.g. hour-of-day, day-of-month, or regularly scheduled public holidays. These granularities form ordered or unordered categorical variables. An example of an ordered granularity is day-of-week, where Tuesday is always followed by Wednesday, and so on. An unordered granularity example is week type in an academic semester: orientation, break, exam or regular classes. We can use granularities to explore patterns in univariate time series by examining the distribution of the measured variable across different categories of the cyclic granularities.

As a motivating example, consider Figure 3.1 which shows electricity smart meter data plotted against two granularities (hour-of-day, month-of-year). The data was collected on a single household in Melbourne, Australia, over a six month period, and was previously used in Wang, Cook,

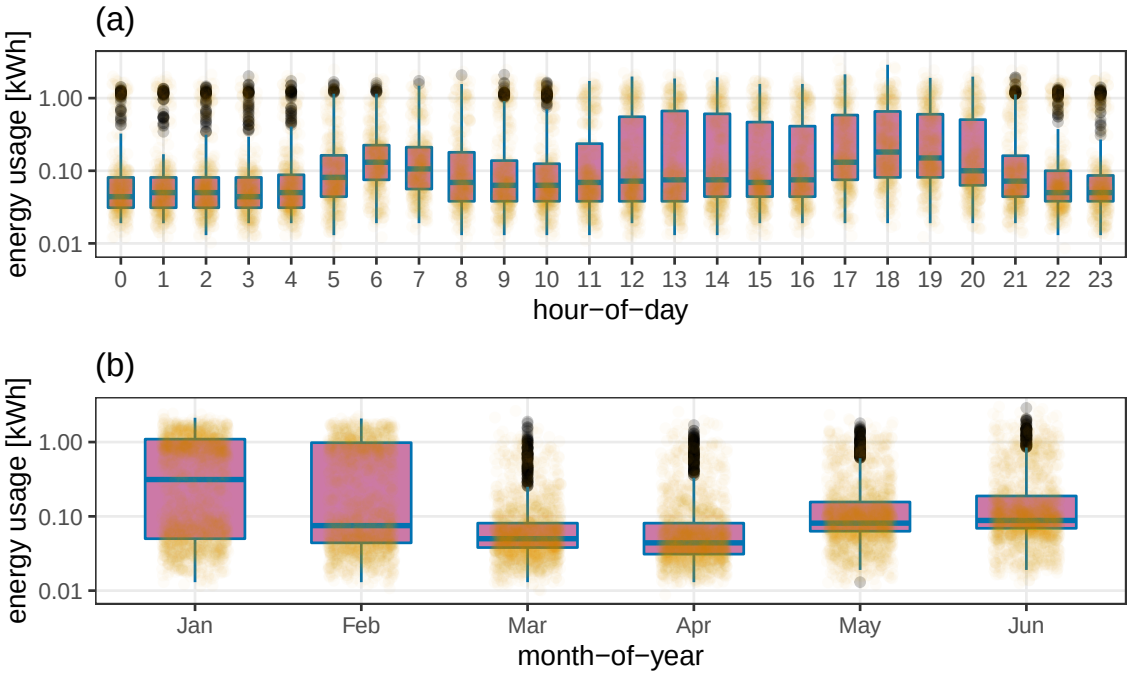

Figure 3.1: *A cyclic granularity can be considered to be a categorical variable, and used to break the data into subsets. Here, side-by-side boxplots overlaid on jittered dotplots explore the distribution of energy use (on a logarithmic scale) by a household for two different cyclic granularities: (a) hour-of-day and (b) month-of-year. Daily peaks occur in morning and evening hours, indicating a working household, where members leave for and return from work. More volatility of usage in summer months in Australia (Jan, Feb) is probably due to air conditioner use on just some days.*

and Hyndman (2020b). The categorical variable (granularity) is mapped to the x-axis, and the distribution of the response variable is displayed using both side-by-side jittered dotplots and boxplots. From panel (a) it can be seen that energy consumption is higher during the morning hours (5–8), when members in the household wake up, and again in the evening hours (17–20), possibly when members get back from work. In addition, the largest variation in energy use is in the afternoon hours (12–16), as seen in the length of the boxes. From panel (b), it is seen that the variability in energy usage is higher in Jan and Feb, probably due to the usage of air conditioners on some days. The median usage is highest in January, dips in February–April and rises again in May–June, although not to the height of January usage. This suggests that the household does not use as much electricity for heating as it does for air conditioning. A lot of households in Melbourne use gas heating and hence the heater use might not be reflected in the electricity data.

Many different displays could be constructed using different granularities including day-of-week, day-of-month, weekday/weekend, etc. However, only a few might be interesting and reveal important patterns in energy usage. Determining which displays have "significant" distributional differences between categories of the cyclic granularity, and plotting only these, would make for efficient exploration.

Exploring the distribution of the measured variable across two cyclic granularities provides more detailed information on its structure. For example, Figure 3.2(a) shows the usage distribution across hour-of-day conditional on month-of-year across two households. It shows the hourly usage over a day does not remain the same across months. Unlike other months, the 75th and 90th percentile for all hours of the day in January are high, similar, and are not characterized by a morning and evening peak. The household in Figure 3.2(b) has 90th percentile consumption higher in summer months relative to autumn or winter, but the 75th and 90th percentile are far apart in all months, implying that the second household resorts to air conditioning much less regularly than the first one. The differences seem to be more prominent across month-of-year (facets) than hour-of-day (x-axis) for this household, whereas they are prominent for both cyclic granularities for the first household.

Are all four displays in Figures 3.1 and 3.2 useful in understanding the distributional difference in energy usage? Which ones are more useful than others? If N_C is the total number of cyclic granularities of interest, the number of displays that could be potentially informative is N_C when considering displays of the form in Figure 3.1. The dimension of the problem, however, increases

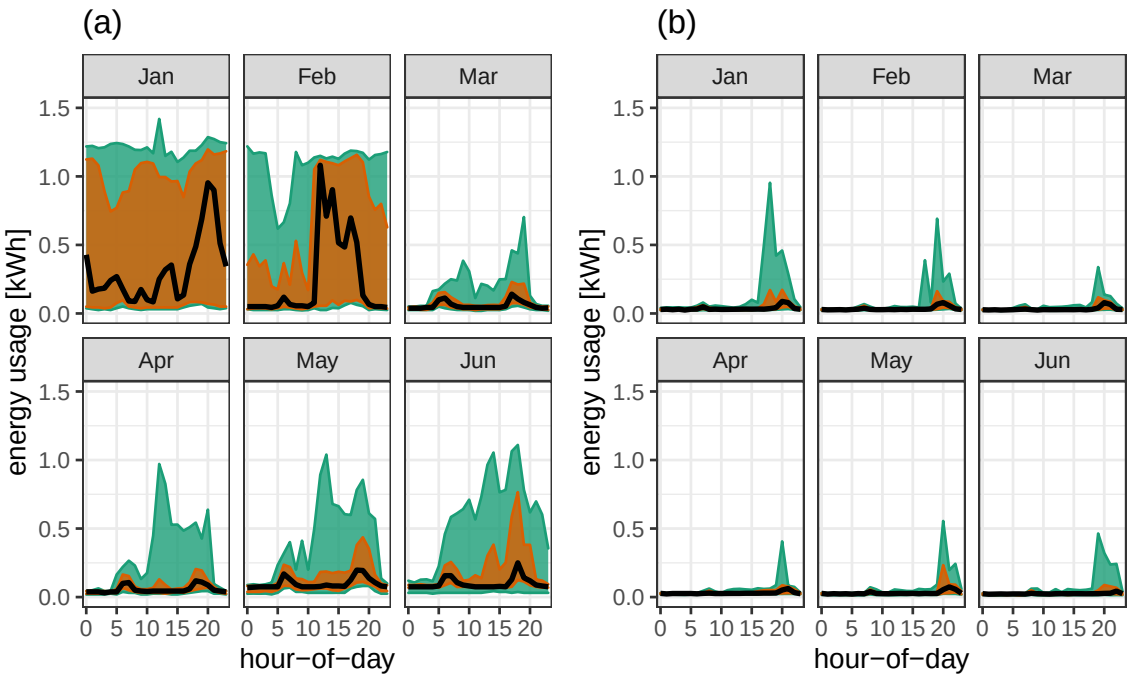

Figure 3.2: *Distribution of energy consumption displayed through area quantile plots across two cyclic granularities month-of-year and hour-of-day and two households. The black line is the median, whereas the orange band covers the* 25^{th} *(lower) to* 75^{th} *(upper) percentile and the green band covers the* 10^{th} *(lower) to* 90^{th} *(upper) percentile. Difference between the* 90^{th} *and* 75^{th} *percentile is less for (Jan, Feb) for the first household (a), suggesting that it is a more frequent user of air conditioners than the second household (b). Distribution of energy usage for (a) changes across both granularities, whereas for (b) daily pattern stays same irrespective of the months.*

when considering more than one cyclic granularity. When considering displays of the form in Figure 3.2, there are $N_C(N_C - 1)$ possible pairwise plots exhaustively, with one of the two cyclic granularities acting as the conditioning variable. This can be overwhelming for human consumption even for moderately large N_C. It is therefore useful to identify those displays that are informative across at least one cyclic granularity.

This problem is similar to Scagnostics (Scatterplot Diagnostics) by Tukey and Tukey (1988), which are used to identify meaningful patterns in large collections of scatterplots. Given a set of v variables, there are $p(p-1)/2$ pairs of variables, and thus the same number of possible pairwise scatterplots. Therefore, even for small v, the number of scatterplots can be large, and scatterplot matrices (SPLOMs) can easily run out of pixels when presenting high-dimensional data. Dang and Wilkinson (2014); Wilkinson, Anand, and Grossman (2005) provided potential solutions to this, where a few characterizations can be used to locate anomalies in density, shape, trend, and other features in the 2D point scatters.

In this paper, we provide a solution to narrowing down the search from $N_C(N_C-1)$ conditional distribution plots by introducing a new distance measure that can be used to detect significant distributional differences across cyclic granularities. This work is a natural extension of our previous work (Gupta et al., 2021) which narrows down the search from $N_C(N_C-1)$ plots by identifying pairs of granularities that can be meaningfully examined together (a "harmony"), or when they cannot (a "clash"). However, even after excluding clashes, the list of harmonies left may be too large for exhaustive exploration. Hence, there is a need to reduce the search even further by including only those harmonies that contain useful information.

Buja et al. (2009); Majumder, Hofmann, and Cook (2013) presented methods for statistical significance testing of visual findings using human cognition as the statistical tests. In this paper, the visual discovery of distributional differences is facilitated by choosing a threshold for the proposed numerical distance measure, eventually selecting only those cyclic granularities for which the distributional differences are sufficient to make it an interesting display. Two things need to be accomplished here: 1) to see if there are any statistically significant differences between independent groups, and 2) to quantify any differences that do exist. One way to address this problem is by using a one-way or two-way ANOVA (Fisher, 1992). Assume we want to examine whether there is a significant difference in electricity demand on various days of the week. In this case, each day of the week may be regarded as an independent group, and a one-way ANOVA can be used to assess how the means of the electricity demand varies across different days of the week. The approach proposed in this paper looks at the distributional differences in the quantitative variable instead of merely the mean or one measure of central tendency. Besides, it also takes into account that the levels in each group have an inherent order in a time series context.

The article is organized as follows. Section 3.2 introduces a distance measure for detecting distributional difference in temporal granularities for a continuous univariate dependent variable. This enables identification of patterns in the time series data; Section 3.3 devises a selection criterion by choosing a threshold, which results in detection of only significantly interesting patterns. Section 3.3.3 provides a simulation study on the proposed methodology. Section 3.4 presents an application to residential smart meter data in Melbourne to show how the proposed methodology can be used to automatically detect temporal granularities along which distributional differences are significant.

3.2 Proposed distance measure

We propose a measure called Weighted Pairwise Distances (*wpd*) to detect distributional differences in the measured variable across cyclic granularities.

3.2.1 Principle

The principle behind the construction of *wpd* is explained through a simple example in Figure 3.3. Each of these figures describes a panel with two x-axis categories and three facet levels, but with different designs. Figure 3.3a has all categories drawn from a standard normal distribution for each facet. It is not a particularly interesting display, as the distributions do not vary across x-axis or facet categories. Figure 3.3b has x categories drawn from the same distribution, but across facets the distribution means are three standard deviations apart. Figure 3.3c exhibits the opposite situation

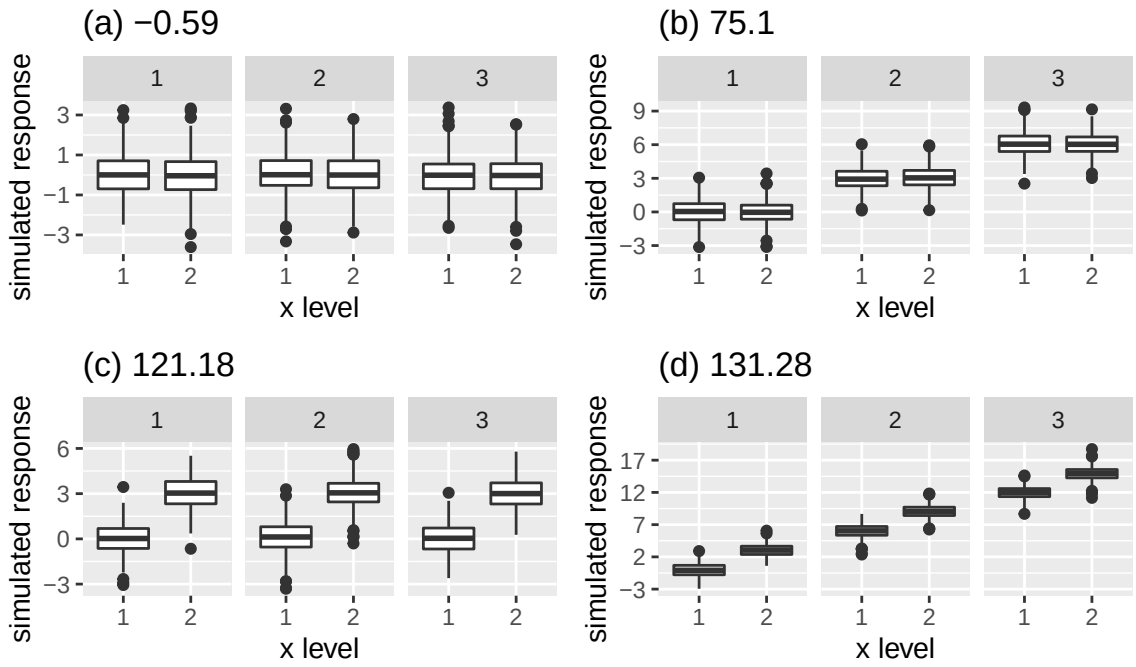

Figure 3.3: *An example illustrating the principle of the proposed distance measure, displaying the distribution of a normally distributed variable in four panels each with two x-axis categories and three facet levels, but with different designs. Panel (a) is not interesting as the distribution of the variable does not depend on x or facet categories. The data set in (a) is consistent with the null distribution. Panels (b) and (c) are more interesting than (a) since there is a change in distribution either across facets (b) or x-axis (c). Panel (d) is most interesting in terms of capturing structure in the variable as the distribution of the variable changes across both facet and x-axis variable. The value of our proposed distance measure is presented for each panel, the relative differences between which will be explained later in Section 3.3.2.*

where distribution between the x-axis categories are three standard deviations apart, but they do not change across facets. In Figure 3.3d, the distribution varies across both facet and x-axis categories by three standard deviations.

If the panels are to be ranked in order of capturing maximum variation in the measured variable from minimum to maximum, then an obvious choice would be (a) followed by (b), (c) and then (d). It might be argued that it is not clear if (b) should precede or succeed (c) in the ranking. Gestalt theory suggests items placed at close proximity can be compared more easily, because people assume that they are in the same group and apart from other groups. With this principle in mind, Panel (b) is considered less informative compared to Panel (c) in emphasizing the distributional differences.

For displays showing a single cyclic granularity rather than pairs of granularities, we have only two design choices corresponding to no difference and significant differences between categories of that cyclic granularity.

The proposed measure *wpd* is constructed in such a way that it can be used to rank panels of different designs as well as test if a design is interesting. This measure is aimed to be an estimate of the maximum variation in the measured variable explained by the panel. A higher value of *wpd* would indicate that the panel is interesting to look at, whereas a lower value would indicate otherwise.

3.2.2 Notation

Let the number of cyclic granularities considered in the display be m. The notations and methodology are described in detail for $m = 2$. But it can be easily extended to $m > 2$. Consider two cyclic granularities A and B, such that $A = \{a_j : j = 1,2,\ldots,n_x\}$ and $B = \{b_k : k = 1,2,\ldots,n_f\}$ with A placed across the x-axis and B across facets. Let $v = \{v_t : t = 0,1,2,\ldots,T-1\}$ be a continuous variable observed across T time points. This data structure with n_x x-axis levels and n_f facet levels is referred to as a (n_x, n_f) panel. For example, a $(2,3)$ panel will have cyclic granularities with two x-axis levels and three facet levels. Let the four elementary designs as described in Figure 3.3 be $D_\emptyset$ (referred to as "null distribution") where there is no difference in distribution of v for A or B. For example, the data set in Figure 3.3(a) is consistent with the null distribution. D_f denotes the set of designs where there is difference in distribution of v for B and not for A. Similarly, D_x denotes

Table 3.1: *Nomenclature table*

variable	description
N_C	number of cyclic granularities
H_{N_C}	set of harmonies
m	number of cyclic granularities to display together
n_x	number of x-axis categories
n_f	number of facet categories
λ	tuning parameter
ω	increment (mean or sd)
wpd	raw weighted pairwise distance
wpd_{norm}	normalized weighted pairwise distance
n_{perm}	number of permutations for threshold/normalization
n_{sim}	number of simulations
$wpd_{\text{threshold}}$	threshold for significance
$D_\emptyset$	null design with no distributional difference across categories
D_f	design with distributional difference only across facets categories
D_x	design with distributional difference only across x-axis categories
D_{fx}	design with distributional difference across both facet and x-axis
v	continuous univariate measured variable

the set of designs where difference is observed only across A. Finally, D_{fx} denotes those designs for which difference is observed across both A and B. We can consider a single granularity ($m = 1$) as a special case of two granularities with $n_f = 1$.

3.2.3 Computation

The computation of the distance measure *wpd* for a panel involves characterizing distributions, computing distances between distributions, choosing a tuning parameter to specify the weight for different groups of distances and summarizing those weighted distances appropriately to estimate maximum variation. Furthermore, the data needs to be appropriately transformed to ensure that the value of *wpd* emphasizes detection of distributional differences across categories and not across different data generating processes.

Data transformation

The intended aim of *wpd* is to capture differences in categories irrespective of the distribution from which the data is generated. Hence, as a pre-processing step, the raw data is normal-quantile

transformed (NQT) (Krzysztofowicz, 1997), so that the transformed data follows a standard normal distribution. The empirical NQT involves the following steps:

1. The observations of measured variable v are sorted from the smallest to the largest observation $v_{(1)},\ldots,v_{(n)}$.
2. The cumulative probabilities $p_{(1)},\ldots,p_{(n)}$ are estimated using $p_{(i)} = i/(n+1)$ (Hyndman and Fan, 1996) so that $p_{(i)} = \Pr(v \leq v_{(i)})$.
3. Each observation $v_{(i)}$ of v is transformed into $v^*(i) = \Phi^{-1}(p(i))$, with Φ denoting the standard normal distribution function.

Characterizing distributions

Multiple observations of v correspond to the subset $v_{jk} = \{s : A(s) = j, B(s) = k\}$. The number of observations might vary widely across subsets due to the structure of the calendar, missing observations or uneven locations of events in the time domain. In this paper, quantiles of $\{v_{jk}\}$ are chosen as a way to characterize distributions for the category $(a_j, b_k), \forall j \in \{1,2,\ldots,n_x\}, k \in \{1,2,\ldots,n_f\}$. We use percentiles with $p = 0.01, 0.02, \ldots, 0.99$ to reduce the computational burden in summarizing distributions. The assumption is that there is sufficient data for each level or combination of levels, and hence no adjustment is made for the varying number of observations across levels. However, if one or more levels has a small number of data points, consecutive categories should be collapsed prior to data transformation and quantile estimation.

Distance between distributions

A common way to measure divergence between distributions is the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951). The KL divergence denoted by $D(q_1||q_2)$ is a non-symmetric measure of the difference between two probability distributions q_1 and q_2 and is interpreted as the amount of information lost when q_2 is used to approximate q_1. The KL divergence is not symmetric and hence can not be considered as a "distance" measure. The Jensen-Shannon divergence (Menéndez et al., 1997) based on the Kullback-Leibler divergence is symmetric and has a finite value. Hence, in this paper, the pairwise distances between the distributions of the measured variable are obtained through the square root of the Jensen-Shannon divergence, called Jensen-Shannon

distance (JSD), and defined by

$$JSD(q_1||q_2) = \frac{1}{2}D(q_1||M) + \frac{1}{2}D(q_2||M),$$

where $M = \frac{q_1+q_2}{2}$ and $D(q_1||q_2) := \int_{-\infty}^{\infty} q_1(x) log(\frac{q_1(x)}{q_2(x)})dx$ is the KL divergence between distributions q_1 and q_2. There are other ways to obtain distance between distributions like Hellinger distance, total variation distance and Fisher information metric (all of which are special cases of f-divergence), but Jensen-Shannon distance was chosen for ease of computation.

Within-facet and between-facet distances

Pairwise distances could be within-facets or between-facets for $m = 2$. Figure 3.4 illustrates how they are defined. Pairwise distances are within-facets when $b_k = b_{k'}$, that is, between pairs of the form $(a_j b_k, a_{j'} b_k)$ as shown in panel (3) of Figure 3.4. If categories are ordered (like all temporal cyclic granularities), then only distances between pairs where $a_{j'} = (a_{j+1})$ are considered (panel (4)). Pairwise distances are between-facets when they are considered between pairs of the form $(a_j b_k, a_j b_{k'})$. There are a total of $\binom{n_f}{2} n_x$ between-facet distances, and $\binom{n_x}{2} n_f$ within-facet distances if they are unordered and $n_f(n_x - 1)$ within-facet distances if they are ordered.

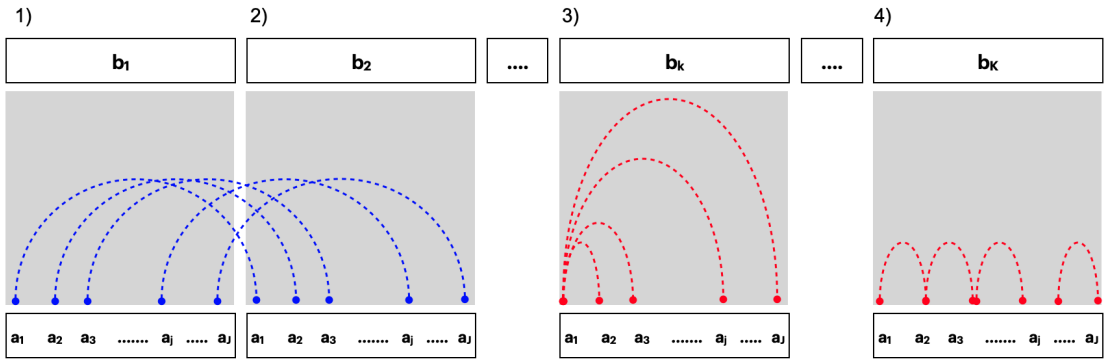

Figure 3.4: *Within and between-facet distances shown for two cyclic granularities A and B, where A is mapped to x-axis and B is mapped to facets. The dotted lines represent the distances between different categories. Panels 1) and 2) show the between-facet distances. Within-facet distances are illustrated in Panels 3) (when categories are un-ordered, shown only with respect to a1) and Panel 4) (when categories are ordered). When categories are ordered, distances should only be considered for consecutive x-axis categories. Between-facet distances are distances between different facet levels for the same x-axis category; for example, distances between (a_1,b_1) and (a_1, b_2) or (a_1,b_1) and (a_1, b_3).*

Tuning parameter

For displays with $m > 1$ granularities, we can use a tuning parameter to specify the relative weight given to each granularity. In general, the tuning parameters should be chosen such that $\sum_{i=1}^{m} \lambda_i = 1$.

Following the general principles of Gestalt theory, we wish to weight more heavily granularities that are plotted closer together. For $m = 2$ we choose $\lambda_x = \frac{2}{3}$ for the granularity on the x-axis and $\lambda_f = \frac{1}{3}$ for the granularity mapped to facets, giving a relative weight of $2:1$ for within-facet to between-facet distances. No human experiment has been conducted to justify this ratio. Specifying $\lambda_x > 0.5$ will weight within-facet distances more heavily, while $\lambda_x < 0.5$ would weight the between-facet distances more heavily. (See Section 2.1 supplements for more details.)

Raw distance measure

The raw distance measure, denoted by wpd_{raw}, is computed after combining all the weighted distance measures appropriately. First, NQT is performed on the measured variable v_t to obtain v_t^* (*data transformation*). Then, for a fixed harmony pair (A,B), percentiles of v_{jk}^* are computed and stored in q_{jk} (*distribution characterization*). This is repeated for all pairs of categories of the form $(a_j b_k, a_{j'} b_{k'}) : \{a_j : j = 1,2,\ldots,n_x\}, B = \{b_k : k = 1,2,\ldots,n_f\}$. The pairwise distances between pairs $(a_j b_k, a_{j'} b_{k'})$ denoted by $d_{(jk,j'k')} = JSD(q_{jk}, q_{j'k'})$ are computed (*distance between distributions*). The pairwise distances (*within-facet and between-facet*) are transformed using a suitable *tuning parameter* $(0 < \lambda < 1)$ depending on if they are within-facet(d_w) or between-facets(d_b) as follows:

$$d*_{(j,k),(j'k')} = \begin{cases} \lambda d_{(jk),(j'k')}, & \text{if } d = d_w; \\ (1-\lambda) d_{(jk),(j'k')}, & \text{if } d = d_b. \end{cases} \tag{3.1}$$

The wpd_{raw} is then computed as

$$wpd_{\text{raw}} = \max_{j,j',k,k'} \left(d*_{(jk),(j'k')}\right) \qquad \forall j, j' \in \{1,2,\ldots,n_x\}, \quad k,k' \in \{1,2,\ldots,n_f\}$$

The statistic "maximum" is chosen to combine the weighted pairwise distances since the distance measure is aimed at capturing the maximum variation of the measured variable within a panel. The

statistic "maximum" is, however, affected by the number of comparisons (resulting pairwise distances). For example, for a $(2,3)$ panel, there are 6 possible subsets of observations corresponding to the combinations $(a_1,b_1),(a_1,b_2),(a_1,b_3),(a_2,b_1),(a_2,b_2),(a_2,b_3)$, whereas for a $(2,2)$ panel, there are only 4 possible subsets $(a_1,b_1),(a_1,b_2),(a_2,b_1),(a_2,b_2)$. Consequently, the measure would have higher values for the panel $(2,3)$ as compared to $(2,2)$, since maximum is taken over higher number of pairwise distances.

3.2.4 Adjusting for the number of comparisons

Ideally, it is desired that the proposed distance measure takes a higher value only if there is a significant difference between distributions across categories, and not because the number of categories n_x or n_f is high. That is, under designs like $D_\emptyset$, their distribution should not differ for a different number of categories. Only then could the distance measure be compared across panels with different levels. This calls for an adjusted measure, which normalizes for the different number of comparisons.

Two approaches for adjusting the number of comparisons are discussed, both of which are substantiated using simulations. The first one defines an adjusted measure wpd_{perm} based on the permutation method to remove the effect of different comparisons. The second approach fits a model to represent the relationship between wpd_{raw} and the number of comparisons and defines the adjusted measure (wpd_{glm}) as the residual from the model.

Permutation approach

This method is somewhat similar in spirit to bootstrap or permutation tests, where the goal is to test the hypothesis that the groups under study have identical distributions. This method accomplishes a different goal of finding the null distribution for different groups (panels in our case) and standardizing the raw values using that distribution. The values of wpd_{raw} are computed on many (n_{perm}) permuted data sets and stored in $wpd_{\text{perm-data}}$. Then wpd_{perm} is computed as follows:

$$wpd_{\text{perm}} = \frac{wpd_{\text{raw}} - \text{mean}(wpd_{\text{perm-data}})}{\text{sd}(wpd_{\text{perm-data}})}$$

where $\text{mean}(wpd_{\text{perm-data}})$ and $\text{sd}(wpd_{\text{perm-data}})$ are the mean and standard deviation of $wpd_{\text{perm-data}}$ respectively. Standardizing *wpd* in the permutation approach ensures that the distribution of wpd_{perm}

under $D_\emptyset$ has zero mean and unit variance across all comparisons. While this works successfully to make the location and scale similar across different n_x and n_f, it is computationally heavy and time consuming, and hence less user-friendly. Hence, another approach to adjustment, with potentially less computational time, is proposed.

Modeling approach

In this approach, a Gamma generalized linear model (GLM) for wpd_{raw} is fitted with the number of comparisons as the explanatory variable. Since, wpd_{raw} is a Jensen-Shannon distance, it follows a Chi-square distribution (Menéndez et al., 1997), which is a special case of a Gamma distribution. Furthermore, the mean response is bounded, since any JSD is bounded by 1 if a base 2 logarithm is used (Lin, 1991). Hence, by Faraway (2016), an inverse link is used for the model, which is of the form $y = a + b \times \log(z) + e$, where $y = wpd_{\text{raw}}$, $z = (n_x \times n_f)$ is the number of groups and e are idiosyncratic errors. Let $\mathrm{E}(y) = \mu$ and $a + b \times \log(z) = g(\mu)$ where $g(\mu) = 1/\mu$ and $\hat{\mu} = 1/(\hat{a} + \hat{b}\log(z))$. The residuals from this model $(y - \hat{y}) = (y - 1/(\hat{a} + \hat{b}\log(z)))$ would be expected to have no dependency on z. Thus, wpd_{glm} is defined as the residuals from this model given by

$$wpd_{\text{glm}} = wpd_{\text{raw}} - 1/(\hat{a} + \hat{b} \times \log(n_x \times n_f))$$

The distribution of wpd_{glm} under $D_\emptyset$ will have approximately zero mean and a constant variance (not necessarily 1).

Combination approach

The simulation results (in Section 3.3.3) show that the distribution of wpd_{glm} under the null design is similar for high n_x and n_f (levels higher than 5) but less so for lower values of n_x and n_f. An empirical explanation for this is that with smaller n_x and n_f, the variance in the values of wpd_{raw} is greater Figure 13 of supplements. As a result, our modeling approach that fits wpd_{raw} to the number of comparisons performs poorly. In order to address this, a combination approach is proposed where we use a permutation approach for categories with small numbers of levels, and a modeling approach for categories with higher numbers of levels. This ensures that the computational load of the permutation approach is alleviated while maintaining a similar null distribution across different categories. This approach, however, requires that the adjusted variables from the two approaches

are brought to the same scale. We define $wpd_{\text{glm-scaled}} = wpd_{\text{glm}} \times \sigma^2_{\text{perm}} / \sigma^2_{\text{glm}}$ as the transformed wpd_{glm} with a similar scale as wpd_{perm}. The adjusted measure from the combination approach, denoted by *wpd* is then defined as follows:

$$wpd = \begin{cases} wpd_{\text{perm}}, & \text{if } n_x, n_f \leq 5; \\ wpd_{\text{glm-scaled}} & \text{otherwise.} \end{cases} \tag{3.2}$$

3.3 Ranking and selection of cyclic granularities

A cyclic granularity is referred to as "significant" if there is a significant distributional difference of the measured variable between different categories of the harmony. In this section, a selection criterion to choose significant harmonies is provided, thereby eliminating all harmonies that exhibit non-significant differences in the measured variable. The distance measure *wpd* is used as a test statistic to test the null hypothesis that no harmony/cyclic granularity is significant. We select only those harmonies/cyclic granularities for which the test fails. They are then ranked based on how well they capture variation in the measured variable.

3.3.1 Selection

A threshold (and consequently a selection criterion) is chosen using the notion of randomization tests (Edgington and Onghena, 2007). The data is permuted several times and *wpd* is computed for each of the permuted data sets to obtain the sampling distribution of *wpd* under the null hypothesis. If the null hypothesis is true, then *wpd* obtained from the original data set would be a likely value in the sampling distribution. But in case the null hypothesis is not true, then it is less probable that *wpd* obtained for the original data will be from the same distribution. This idea is utilized to come up with a threshold for selection, denoted by $wpd_{\text{threshold}}$, defined as the 99^{th} percentile of the sampling distribution. A harmony is selected if the value of *wpd* for that harmony is greater than the chosen threshold. The detailed algorithm for choosing a threshold and selection procedure (for two cyclic granularities) is listed as follows:

- **Input:** All harmonies of the form $\{(A,B),\ A = \{a_j : j = 1,2,\ldots,n_x\},\ B = \{b_k : k = 1,2,\ldots,n_f\}\}, \forall (A,B) \in H_{N_C}$.

- **Output:** Harmony pairs (A,B) for which *wpd* is significant.

1. For each harmony pair $(A,B) \in H_{N_C}$, the following steps are taken.
   a. Given the measured variable; $\{v_t : t = 0,1,2,\ldots,T-1\}$, *wpd* is computed and is represented by $wpd_{obs}^{A,B}$.
   b. For $i = 1,\ldots,M$, randomly permute the original time series: $\{v_t^i : t = 0,1,2,\ldots,T-1\}$ and compute $wpd_i^{A,B}$ from $\{v_t^i\}$.
   c. Define $wpd_{\text{sample}} = \{wpd_1^{A,B},\ldots,wpd_M^{A,B}\}$.
2. Stack the wpd_{sample} vectors as $wpd_{\text{sample}}^{\text{all}}$ and compute its $p = 100(1-\alpha)$ percentiles as $wpd_{\text{threshold}p}$.
3. If $wpd_{obs}^{A,B} > wpd_{\text{threshold}p}$, harmony pair (A,B) is selected at the $1 - p/100$ level, otherwise rejected.
4. Harmonies selected using the 99^{th}, 95^{th} and 90^{th} thresholds are tagged as ***, **, * respectively.

3.3.2 Ranking

The distribution of *wpd* is expected to be similar for all harmonies under the null hypothesis, since they have been adjusted for different number of categories for the harmonies or underlying distribution of the measured variable. Hence, the values of *wpd* for different harmonies are comparable and can be used to rank the significant harmonies. A higher value of *wpd* for a harmony indicates that higher maximum variation in the measured variable is captured through that harmony.

Figure 3.3 also presents the results of *wpd* from the illustrative designs in Section 3.2. The value of *wpd* under null design (a) is the least, followed by (b), (c) and (d). This aligns with the principle of *wpd*, which is expected to have lowest value for null designs and highest for designs of the form D_{fx} (d). Moreover, note the relative differences in *wpd* values between (b) and (c). The value of the tuning parameter λ is set to 2/3, which gives greater emphasis to differences in x-axis categories than facets.

Again consider Figures 3.1(a) and 3.1(b) with a *wpd* value of 20.5 and 145 respectively. This is because there is a more gradual increase across hours of the day than across months of the year. If

the order of categories is ignored, the resulting *wpd* values are 97.8 and 161 respectively, because differences between any hours of the day tend to be larger than differences only between consecutive hours. Similarly, Figures 3.2(a) and (b) have *wpd* values of 110.79 and 125.82 respectively. The ranking implies that the distributional differences are more prominent for the second household, as is also seen from the bigger fluctuations in the 90^{th} percentile than for the first household.

3.3.3 Simulations

Simulations were carried out to explore how the behavior of *wpd* as n_x and n_f were varied, in order to compare and evaluate different normalization approaches for both $m = 1$ and $m = 2$. Here the simulation design and results corresponding to $m = 2$ are presented. Similar design and results for $m = 1$, although important, are not included in the paper but in the supplements along with other more detailed simulation results.

Simulation design

Observations were generated from a N(0,1) distribution for each combination of n_x and n_f from $\{2,3,5,7,14,20,31,50\}$, with $n_{\text{times}} = 500$ observations drawn for each of the 64 combinations. This design corresponds to $D_\emptyset$. For each of the categories, there were $n_{\text{sim}} = 200$ replications, so that the distribution of *wpd* under $D_\emptyset$ could be observed.

Results

Figure 3.5 shows that both the location and scale of the distributions change across panels. This is not desirable under $D_\emptyset$ as it would mean comparisons of *wpd* values are not appropriate across different n_x and n_f values. Table 3.2 gives the summary of a Gamma generalized linear model to capture the relationship between wpd_{raw} and the number of comparisons. The intercepts and slopes are similar, independent of the underlying distributions (see Table 3 supplementary paper for details) and hence the coefficients are shown for the case when observations are drawn from a N(0,1) distribution. Figure 3.6 shows the distribution of wpd_{perm} and $wpd_{\text{glm-scaled}}$ on the same scale to show that a combination approach could be used for higher values of n_x and n_f to alleviate the computational time of the permutation approach.

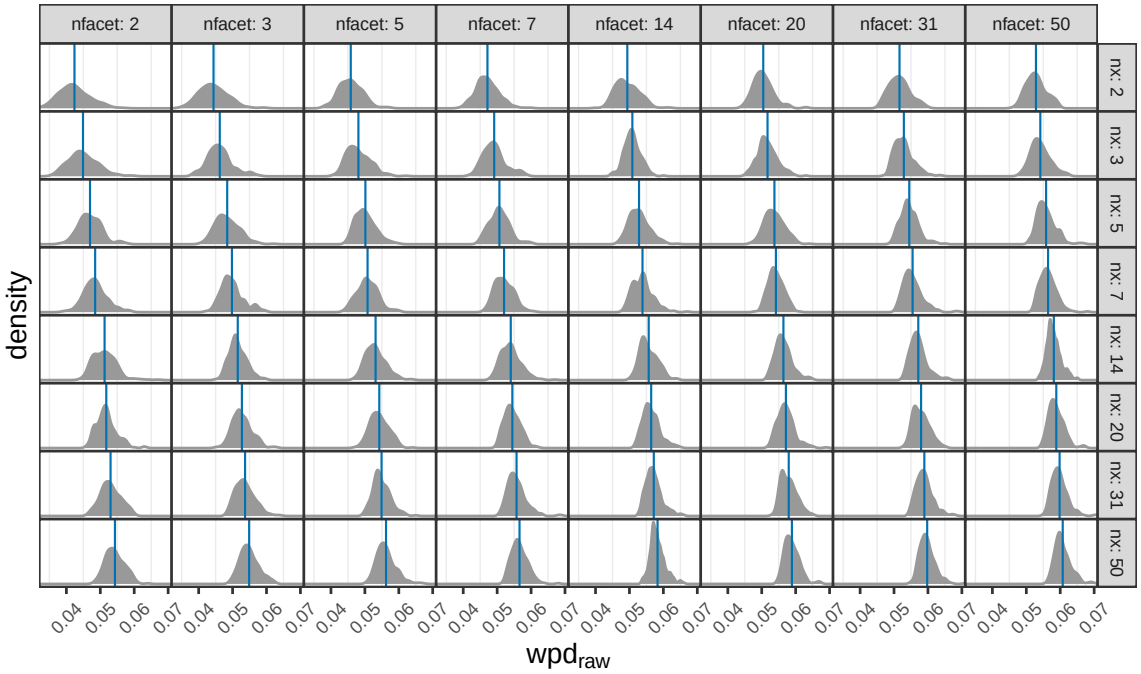

Figure 3.5: *Distribution of wpd_{raw} is plotted across different n_x and n_f categories under $D_\emptyset$ through density and rug plots for $m = 2$. Both location (blue line) and scale of the distribution shifts for different panels. This is not desirable since under the null design, the distribution is not expected to capture any differences.*

Table 3.2: *Results of generalised linear model to capture the relationship between wpd_{raw} and the number of comparisons.*

term	estimate	std.error	statistic	p.value
Intercept	23.40	0.22	104.14	<.001
$\log(n_x \times n_f)$	-0.96	0.04	-21.75	<.001

These results justify our use of the permutation approach when $n_x \leq 5$ and $n_f \leq 5$, and the use of the GLM otherwise.

3.4 Application to residential smart meter dataset

The smart meter data set for eight households in Melbourne was procured by downloading individual level data from energy suppliers/retailers. To reflect various energy behaviours, data was obtained from friends and colleagues with different professions, lifestyles, and household sizes. This level of data is private, and we thank our friends and colleagues for agreeing to have their data used for this analysis. The data has been cleaned to form a `tsibble` (Wang, Cook, and Hyndman, 2020a) containing half-hourly electricity consumption from July to December 2019 for each of the

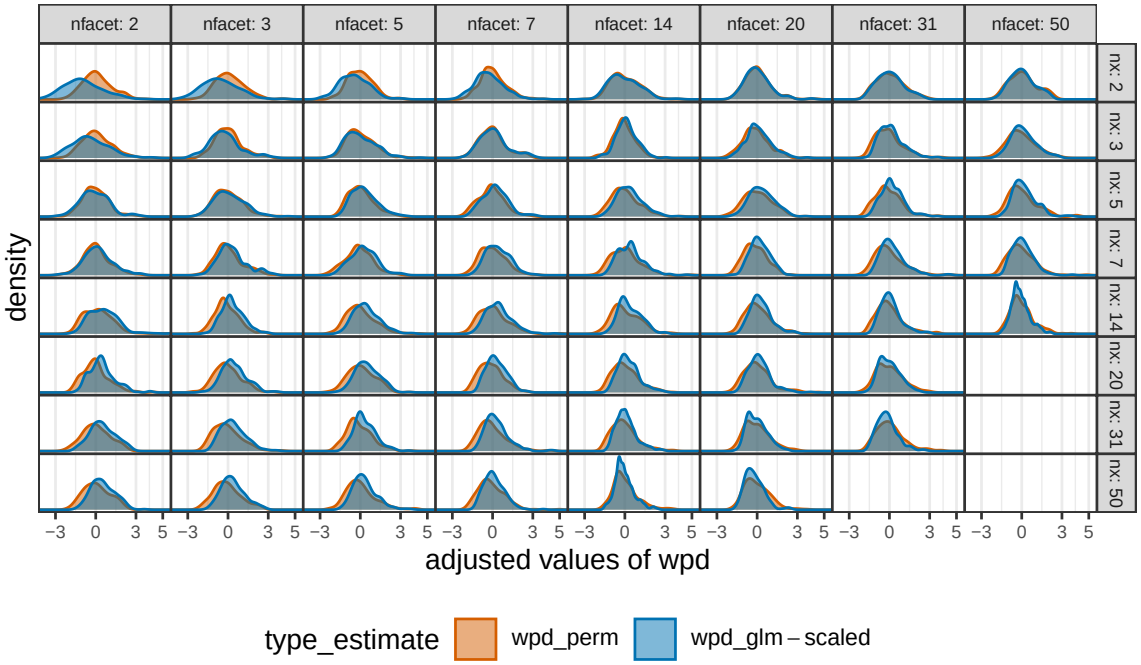

Figure 3.6: *The distributions of wpd_{perm} and $wpd_{glm\text{-}scaled}$ are overlaid to compare the location and scale across different n_x and n_f for $m = 2$. wpd_{norm} takes the value of wpd_{perm} for lower levels, and $wpd_{glm\text{-}scaled}$ for higher levels to to alleviate the problem of computational time in permutation approaches. This is possible as the distribution of the adjusted measure looks similar for both approaches for higher levels.*

households. No behavioral pattern is likely to be discerned from the time plot of energy usage over the entire period, since the plot will have too many observations squeezed in a linear representation. When we zoom into the September 2019 data in Figure 3.7(b), some patterns are visible in terms of peaks and troughs, but we do not know if they are regular or what is their period.

Electricity demand, in general, has daily, weekly and annual seasonal patterns. However, it is not apparent from this view if all households have those patterns, or how strong they are in each case. It is also not clear from this view if any other periodic patterns are present in any household. We start the analysis by choosing a few harmonies, and ranking them for each of these households. The ranking and selection of significant harmonies is validated by analyzing the distribution of energy usage across significant harmonies.

Choosing cyclic granularities of interest and removing clashes

Let $v_{i,t}$ denote the electricity demand for the i^{th} household in time period t. The series $\{v_{i,1}, \ldots, v_{i,T}\}$ is the linear granularity corresponding to half-hour since the interval of the `tsibble` is 30 minutes. We consider coarser linear granularities like hour, day, week and month from the commonly used

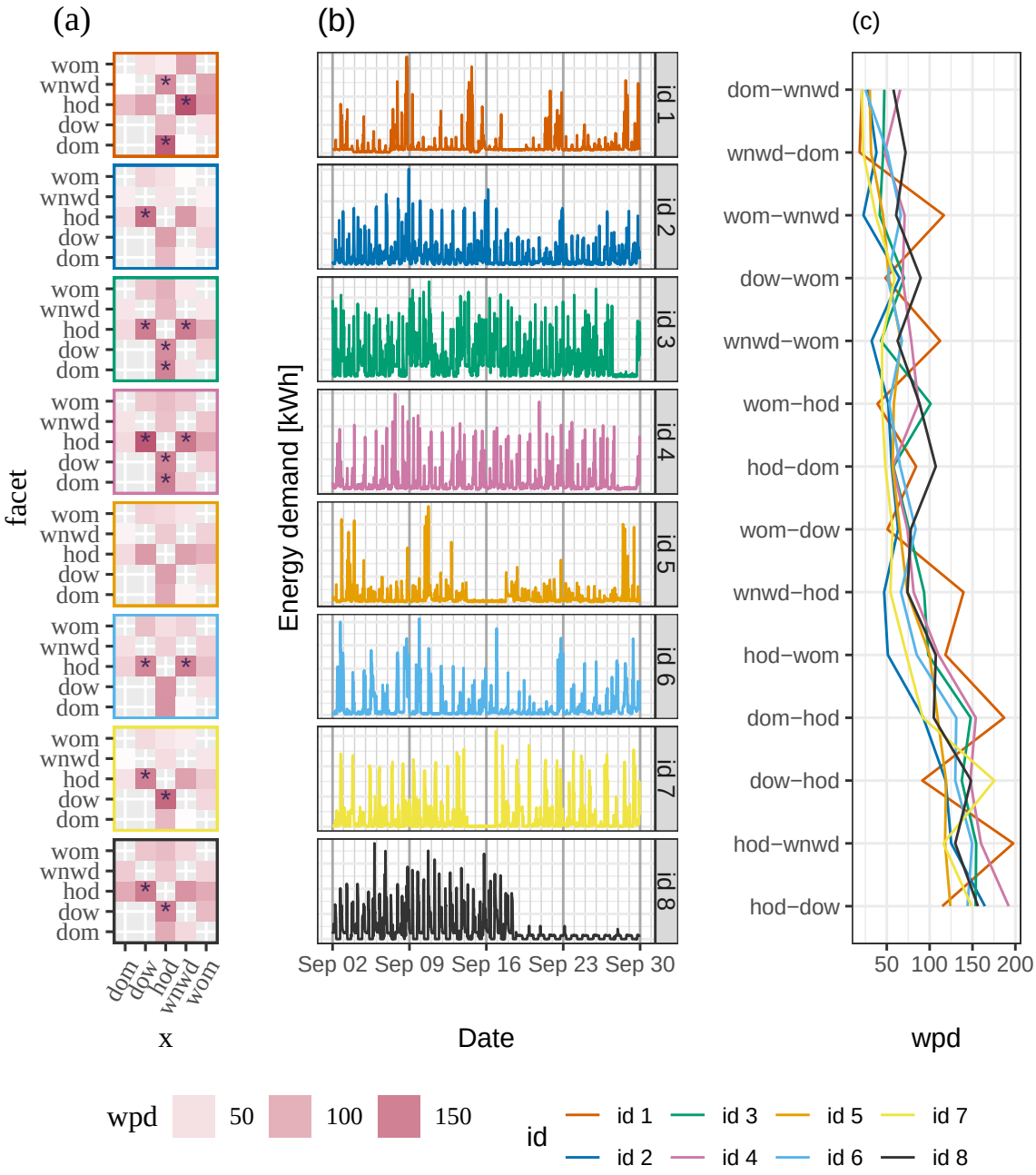

Figure 3.7: *An ensemble plot with a heatmap (a), line plot (b), parallel coordinate plot (c) to demonstrate energy behavior of the households in different ways. (a) and (b) display the household ids in the same order as indicated by the same colour of the line plot in (b) and box border in (a). (b) shows the raw demand series for September to highlight the repetitive patterns of energy demand. (a) shows wpd values across harmonies where a darker shade of red indicates a higher ranking harmony and gray represents a clash. A significant harmony is shown with an asterisk. For example, ids 7 and 8 have significant patterns across the pair (*hod, dow*) since the tile with* hod *as facet and* dow *as x variable has an asterisk. (c) is useful for comparing households across harmonies. For example, for the harmony (*dow-hod*), ids 1 and 7 have the least and highest wpd respectively.*

Gregorian calendar. From the four linear granularities of hour, day, week, and month, we obtain $N_C = 4 \times 3/2 = 6$ cyclic granularities: "hour_day", "hour_week", "hour_month", "day_ week", "day_month" and "week_month" abbreviated as *hod*, *how*, *hom*, *dow*, *dom* and *wom* respectively. Further, we add cyclic granularity day-type ("wknd wday") to capture weekend and weekday behavior (abbreviated as *wnwd*). Thus, seven cyclic granularities are considered to be of interest. The pairs of cyclic granularities (C_{N_C}) will have $7 \times 6 = 42$ elements. The set of possible harmonies H_{N_C} from C_{N_C} are chosen by removing clashes using procedures described in Gupta et al. (2021). Table 3.3 shows 14 harmony pairs that belong to H_{N_C}.

Table 3.3: *Ranking of harmonies for the eight households with significance or different thresholds. ***, ** and * corresponds to the 99^{th}, 95^{th} and 90^{th} percentile threshold. Rankings are different and at most three harmonies are significant for any household. The number of pair of cyclic granularities to explore is reduced from 42 to 3.*

facet variable	x variable	id 1	id 2	id 3	id 4	id 5	id 6	id 7	id 8
hod	wnwd	1^{***}	2^{*}	1^{**}	2^{**}	3	1^{**}	3	3^{*}
dom	hod	2^{***}	4	3^{**}	3^{**}	4	3^{*}	4	6
wnwd	hod	3^{**}	10	7	7	6	8	8	10
hod	wom	4	9	6	5	5	5	5	5
wom	wnwd	5	14	14	10	12	9	12	13
hod	dow	6	1^{***}	2^{**}	1^{***}	1^{*}	2^{**}	2^{**}	1^{**}
wnwd	wom	7	12	13	8	7	7	10	12
dow	hod	8	3	4^{**}	4^{**}	2	4^{*}	1^{***}	2^{**}
hod	dom	9	7	10	13	10	10	9	4
wom	dow	10	6	8	9	8	6	7	9
dow	wom	11	5	9	11	11	12	6	7
wom	hod	12	8	5	6	9	11	11	8
dom	wnwd	13	13	11	12	14	14	14	14
wnwd	dom	14	11	12	14	13	13	13	11

Selecting and ranking harmonies for all households

wpd_i is computed on $v_{i,t}$ for all harmony pairs $\in H_{N_C}$ and for each household $i \in \{1,2,\ldots,8\}$. The harmony pairs are then arranged in descending order and highlighted with $***$, $**$ and $*$ corresponding to the 99^{th}, 95^{th} and 90^{th} percentile threshold. Table 3.3 shows the rank of the harmonies for different households and the harmonies are sorted according to id 1. The rankings are different for different households, which is a reflection of their varied behaviors. Most importantly, there are at most three harmonies that are significant for any household. This is a huge reduction in the number of potential harmonies to explore.

Detecting patterns not apparent from linear display

Figure 3.7 helps to compare households through the heatmap (a) across harmony pairs (lexicographic order). Each household is represented by 25 tiles, each tile representing a pair of cyclic granularities with x-variables on the horizontal axis and facet-variables on the vertical axis. The colors (in shades of red) represent the value of *wpd* for each of the harmony pairs (in Table 3.3) and the gray tiles correspond to clashes. A darker shade of red corresponds to higher values of *wpd*. Those with $*$ correspond to *wpd* values above $wpd_{\text{threshold95}}$.

We can now see some patterns in 3.7(a) that were not discernible in 3.7(b), including:

1. id 7 and 8 have the same significant harmonies despite having very different total energy usage.
2. id 6 and 7 differ in the sense that for id 6, the difference in patterns is only during weekday/weekends, whereas for id 7 all or few other days of the week are also important. This might be due to their flexible work routines or different day-off.
3. There are no significant periodic patterns for id 5 when we fix the threshold to $wpd_{\text{threshold95}}$.

Note that the *wpd* values are computed over the entire range, but the linear display in (b) is only for September, with the major and minor x-axis corresponding to weeks and days respectively.

Comparing households and validating rank of harmonies

Figure 3.7(c) shows a parallel coordinate plot across different harmonies with harmonies arranged from highest to lowest *wpd* values averaged over all households. This display is useful for

comparing households across harmonies. For example, for the harmony pair (*dow-hod*), household id 7 has the greatest value of *wpd*, while id 1 has the least. From Table 3.3 it can be seen that the harmony pair (*dow*, *hod*) is important for id 7; however, it has been labeled as an insignificant pair for id 1. The distribution of energy demand for both of these households, with *dow* as the facet and *hod* on the x-axis, may help explain the choice. Figure 3.8 demonstrates that for id 7, the median (black) and quartile deviation (orange) of energy consumption fluctuates for most hours of the day and days of the week, while for id 1, daily patterns are more consistent within weekdays and weekends. As a result, for id 1, it is more appropriate to examine the distributional difference solely across (*dow*, *wnwd*), which has been rated higher in Table 3.3.

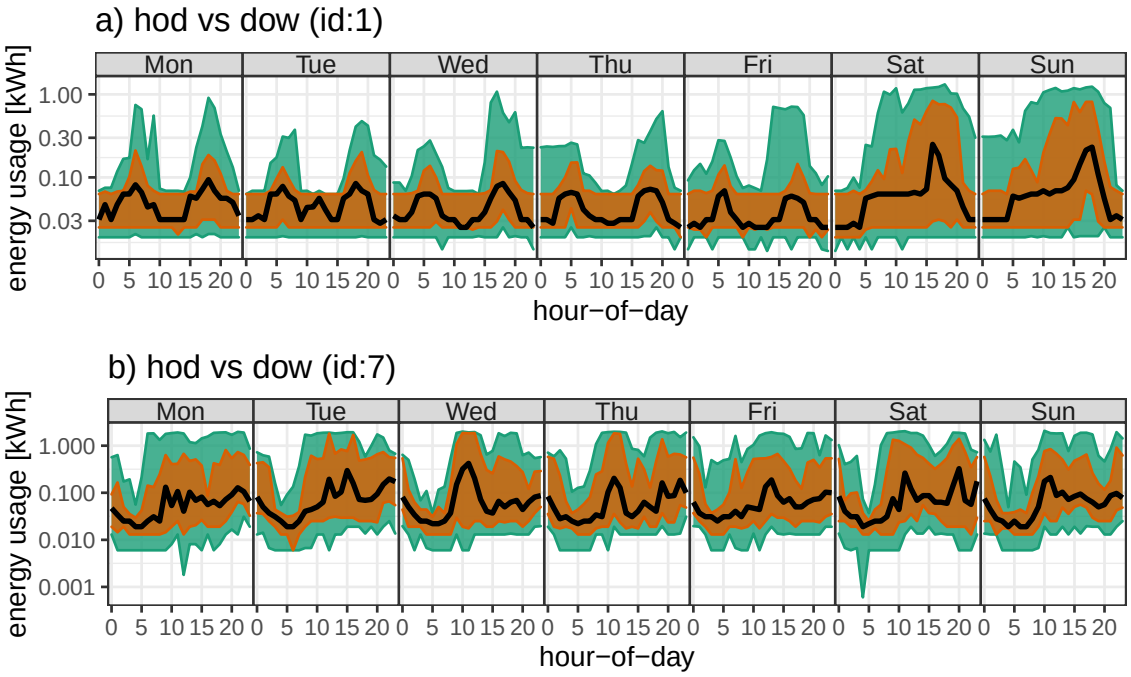

Figure 3.8: *Comparing distribution of energy demand shown for household id 1 (a) and 7 (b) on logarithmic scale across* hod *in x-axis and* dow *in facets through quantile area plots. The value of wpd in Table 3 suggests that the harmony pair (*dow, hod*) is significant for household id 7, but not for id 1. This implies that distributional differences are captured more by this harmony for id 7, which is apparent from this display with more fluctuations across median and 75th percentile for different hours of the day and day of week. For id 1, patterns look similar within different days of weekdays and weekends. Here, the median is represented by the black line, the orange area corresponds to quartile deviation and the green area corresponds to area between* 10^{th} *and* 90^{th} *quantile. The display with* hod *vs* wnwd *would have shown higher values of wpd for id:1 (as can be verified from Table 3).*

3.5 Discussion

Exploratory data analysis involves many iterations of finding and summarizing patterns. With temporal data available at finer scales, exploring time series has become overwhelming with so many possible granularities to explore. A common solution is to aggregate and look at the patterns across the usual granularities such as hour-of-day or day-of-week, but there is no way to know the "interesting" granularities a priori. A huge number of displays need to be analyzed or we might end up missing informative granularities. This work refines the search for informative granularities by identifying those for which the differences between the displayed distributions are greatest and rating them in order of importance of capturing maximum variation.

The significant granularities across different datasets (individuals/subjects) do not imply similar patterns across different datasets. They simply mean that maximum distributional differences are being captured across those granularities. A future direction of work is to be able to explore and compare many individuals/subjects together for similar patterns across significant granularities.

Computational time

The computation time of *wpd* for a cyclic granularity or harmony is determined by its levels. For example, in a harmony with $6 \leq n_x, n_f \leq 50$, the computation time grows with the number of levels and ranges from 30 to 300 seconds. *wpd* becomes computationally more expensive for cyclic granularities with lower levels $n_x, n_f \leq 5$. It is around 40 to 50 minutes for $n_x, n_f = \{2,3\}$. To perform these computations, R version 4.0.1 (2020-06-06) with the platform: x86 64-apple-darwin17.0 (64-bit) running under: macOS Mojave 10.14.6 and MonaRCH is utilized. Even with parallel computation, computing *wpd* for many individuals and cyclic granularity/harmonies takes a long time. One solution could be to obtain the significant granularities for a smaller number of individuals in order to get a sense of the significant granularities for the context and then computing *wpd* only for those granularities for other individuals. However, information may be lost for individuals with different sets of significant granularities. In any case, additional studies may be conducted to speed up the processing of cyclic granularities with low levels.

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 Github repository, github.com/Sayani07/paper-hakear, 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, 2020), `knitr` (Xie, 2015, 2020) and `rmarkdown` (Xie, Allaire, and Grolemund, 2018; Allaire et al., 2020). Graphics are produced with `ggplot2` (Wickham, 2016).

Supplementary materials

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

Supplementary paper: All simulation tables, graphics and and R codes to reproduce the supplementary paper (paper-supplementary.Rmd, paper-supplementary.pdf).

R-package: The open-source R package `hakear` is available on Github (`https://github.com/Sayani07/hakear`) to implement ideas presented in this paper.

Chapter 4

Clustering time series based on probability distributions across temporal granularities

Clustering is a potential approach for organizing large collections of time 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, and 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) characterizing 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.

4.1 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 of 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 review of time series clustering approaches (Aghabozorgi, Shirkhorshidi, and Wah, 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 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 customers 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 these 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, Hyndman, and Cook, 2021), which are temporal deconstructions of a time period into units such as hour-of-the-day or 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 fair to describe a time series based on its degree of trend and seasonality, and to cluster it based on these features. The addition of probability distributions across cyclic granularities to the data structure ensures, there is no need to de-trend or de-seasonalize the data prior to using the clustering algorithm. 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 (2018). Half-hourly measurements of usage for more than 13,000 electricity smart meter customers are provided from October 2011 to March 2014. Customers 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 customers 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. When there is no further customer data available, such as property type, location, or family size, the problem is to cluster customers into these sorts of predicted patterns, as well as other unexpected patterns, using just their energy consumption history (Ushakova and 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 an extensive body of literature focused on time series clustering related to smart meter data. Tureczek and 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 and 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. Many techniques mentioned in Tureczek and Nielsen (2017) fail to recognize smart meter readings as a data type with a temporal component (Tureczek, Nielsen, and Madsen, 2018). Only one study (Ozawa, Furusato, and Yoshida, 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, Berry, and O'Neil (2019) suggested 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 and Akilimali (2010) addressed information theory-based clustering such as Shannon or Renyi entropy and its variations. Melnykov (2013) discussed 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 4.2 provides the clustering methodology. Section 4.3 shows data designs to validate our methods. Section 4.4 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 4.5.

4.2 Clustering methodology

The existing work on clustering probability distributions assumes we have independent and identically distributed samples $f_1(v),\ldots,f_n(v)$, where $f_i(v)$ denotes the distribution from observation i over some random variable $v=\{v_t : t=0,1,2,\ldots,T-1\}$ 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 and B are defined as $A=\{a_j : j=1,2,\ldots J\}$ and $B=\{b_k : k=1,2,\ldots K\}$. For example, let us consider two cyclic granularities, A and B, representing hour-of-day and day-of-week, respectively. Then $A=\{0,1,2,\ldots,23\}$ and $B=\{Mon,Tue,Wed,\ldots,Sun\}$. 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, $J\times K=168$ 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 group observations into a smaller more homogeneous collection. The flow of the procedures is illustrated in Figure 4.1 and is further described in the following subsections.

4.2.1 Selecting granularities

Gupta, Hyndman, and Cook (2021) provide a distance measure (*wpd*) for determining the significance of a cyclic granularity, and a ranking of multiple cyclic granularities. (This extends

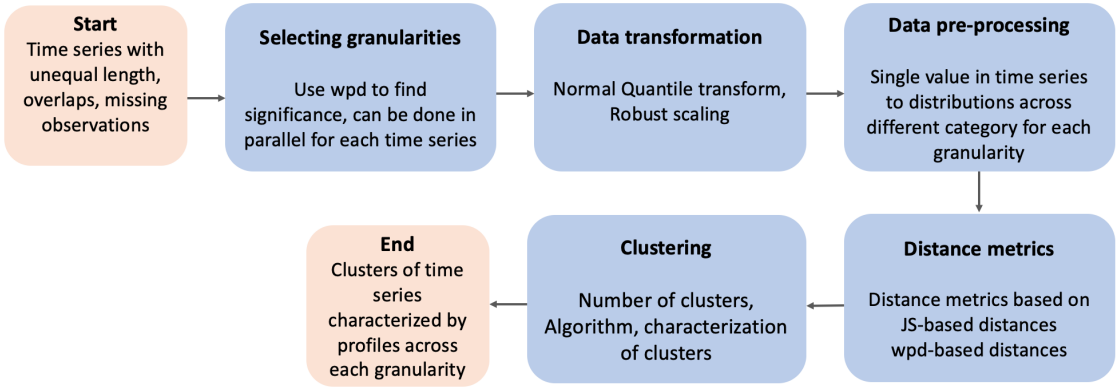

Figure 4.1: *Flow chart illustrating the pipeline for our method for clustering time series.*

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 generating 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.

4.2.2 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 symmetrized. Scales of individuals need to be standardized, because clustering is to select similar patterns, not magnitude of usage. (Organizing individuals based on magnitude can be 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) referred to as JS-NQT and JS-RS respectively. The approach based on *wpd* is referred to as WPD. While Gupta, Hyndman, and Cook (2021) already use NQT when computing *wpd*, it could be useful to standardize it for the selected set of significant granularities prior to computing the distances. In both the JS-NQT

and JS-RS approaches, each observation is scaled individually, but in the WPD approach, each observation may or may not be scaled, resulting in usable groups in either scenario.

- 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^{th}, 50^{th} and 75^{th} percentiles 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 should be checked prior to applying NQT.

4.2.3 Data pre-preprocessing

The initial data in R is supposed to be a "tsibble object" (Wang, Cook, and Hyndman, 2020a) with an index variable representing inherent ordering from past to present, a key variable that specifies observational units through time, and measured variables. As a result, the measured variable for a key is a sequence of values that is time-indexed. However, this sequence may be shown in a variety of ways. A shuffling of the raw sequence may reflect hourly consumption over the course of a day, a week, or a year.

The data object will change when cyclic granularities are computed, as multiple observations will be categorized into levels of the granularity, thus inducing multiple probability distributions. Directly calculating Jensen-Shannon distances between all probability distributions can be time-consuming. As a result, it is suggested that quantiles be employed to characterize probability distributions. In the final data object, each category of cyclic granularity corresponds to a list of numbers, which is composed of a few quantiles.

4.2.4 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 metrics are considered:

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 a cyclic granularity A is defined as

$$S^{A}_{x,y} = \sum_{j \in A} D(x_j, y_j)$$

where D is the Jensen-Shannon distances, x_j is the set of quantiles over the values filtered by j^{th} level of granularity A for observation x (similar for y).

The sum of the distances between two observations x and y in terms of a pair of cyclic granularities (A,B) is defined as

$$S^{A,B}_{x,y} = \sum_{(j,k) \in (A,B)} D(x_{jk}, y_{jk})$$

x_{jk} is the set of quantiles over the values filtered by the combination of j^{th} level of granularity A and k^{th} level of granularity B for observation x (similar for y).

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,

$$S_{x,y} = \sum_{l \in L} S^{C_l}_{x,y} / n_l$$

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.

wpd-based distances

We compute weighted pairwise distances *wpd* (Gupta, Hyndman, and Cook, 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. It is estimated by the maximum pairwise distances between distributions across consecutive categories normalized by appropriate parameters. A higher value of *wpd* indicates that some interesting patterns are 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.

4.2.5 Clustering

Number of clusters

Determining the number of clusters is typically a difficult task. Many metrics have been defined for choosing clusters. Most metrics for choosing the optimal number of clusters are based on comparing distances between observations within a class to those distances between observations between classes, which makes the assumption that there are some separated clusters. Some common procedures include the gap statistic (Tibshirani, Walther, and Hastie, 2001), average silhouette width (Rousseeuw, 1987), Dunn index (Dunn, 1973) and the separation index (*sindex*) (Hennig, 2019, 2014). These are constructed by balancing within-cluster homogeneity and between-cluster separation.

All of the common approaches can give contradictory suggestions for the optimal number of clusters, particularly when the data does not naturally break into groups, or in the presence of nuisance variables (no contribution to clustering) or nuisance observations (inlying and outlying observations falling between clusters). There is no one best metric, which is perhaps a reason why so many metrics exist.

In this work, we have chosen to use *sindex*. It is a very simple but effective metric. This is computed by averaging the smallest 10% of inter-cluster distances. It is relatively robust to nuisance observations. The value of *sindex* always decreases, and sharp drops in value indicate candidates for the optimal number of clusters. The number of clusters corresponding to the value **before the drop** is the recommendation.

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 and 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.

Characterization of clusters

Cluster characterization is an important final stage of a cluster analysis. The primary purpose is to compare the homogeneity within a cluster to the heterogeneity of clusters. This can be done numerically, by tabulating cluster means and standard deviations (Dasu, Swayne, and Poole, 2005), and visually using methods for graphics multivariate data. Cook and Swayne (2007) provide visual examples using both tours (Asimov, 1985) and parallel coordinate plots (Wegman, 1990). Dimension reduction techniques like principal component analysis (Jolliffe and Cadima, 2016), multidimensional scaling (MDS) (Borg and Groenen, 2005), t-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten and Hinton, 2008) and linear discriminant analysis (LDA) (Fisher, 1936) are also useful.

Table 4.1: *The range of parameters used for the validation study, for the three different scenarios, number of simulations (R) for each design, differences between means (μ) across granularities and series lengths (T).*

scenario	designs	R	μ	T
S1	5	25, 250, 500	1, 2, 5	300, 1000, 5000
S2	4			
S3	4			

4.3 Validation

To validate our clustering methods, we have created several different data designs containing different granularity features. There are three circular granularities $g1$, $g2$ and $g3$ with categories denoted by $\{g1_0, g1_1\}$, $\{g2_0, g2_1, g2_2\}$ and $\{g3_0, g3_1, g3_2, g3_3, g3_4\}$ and levels $n_{g_1} = 2$, $n_{g_2} = 3$ and $n_{g_3} = 5$. These categories could be integers or some more meaningful labels. For example, the granularity "day-of-week" could be either represented by $\{0,1,2,\ldots,6\}$ or $\{Mon, Tue, \ldots, Sun\}$. 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 $\{0,1,\ldots T-1\}$ 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 numbers 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. Table 4.1 shows the range of parameters considered for each scenario.

4.3.1 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). μ alters in the "varying" designs, leading to varying distributions across categories.

4.3.2 Data designs

Individual granularities

Scenario 1 (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 4.2 various distributions across categories are considered (top) which lead to different designs (bottom). Figure 4.2 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 2 (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 4.3(right).

Scenario 3 (S3) - Only one significant granularity: Only one granularity is responsible for identifying the designs in this case. This is depicted in Figure 4.3 (right) where only $g3$ affects the designs significantly.

Table 4.2: *Summary of the data generating process for S1. The various distributions across levels of different granularities result in five designs. The granularities g1, g2 and g3 have 2,3 and 5 levels, each of which follows a normal distribution. The means of the distribution are presented for each level, and the standard deviation is set to 1. For example, g1 in design-2 is (0,2) implies* $g1_1 \sim N(0,1)$ *and* $g1_2 \sim N(2,1)$.

design	g1	g2	g3
design-1	(0,0)	(0,0,0)	(0,0,0,0,0)
design-2	(0,2)	(0,0,0)	(0,0,0,0,0)
design-3	(0,0)	(2,1,0)	(0,0,0,0,0)
design-4	(0,0)	(0,0,0)	(0,1,2,1,0)
design-5	(0,2)	(2,1,0)	(0,1,2,1,0)

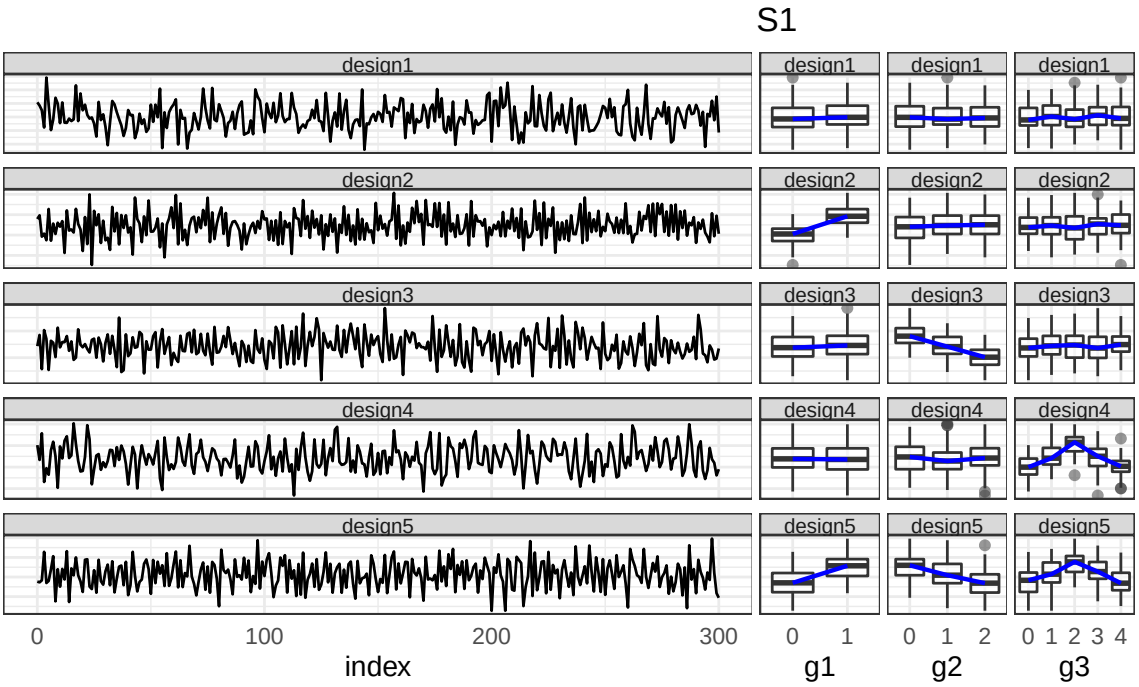

Figure 4.2: *The linear (left) and cyclic (right) representation is shown under scenario S1 using line plots and boxplots respectively. Each row represents a design. Distributions of categories across g1, g2 and g3 change across at least one design as can be observed in the cyclic representation. It is not possible to comprehend these structural differences in patterns just by looking at or considering the linear representation.*

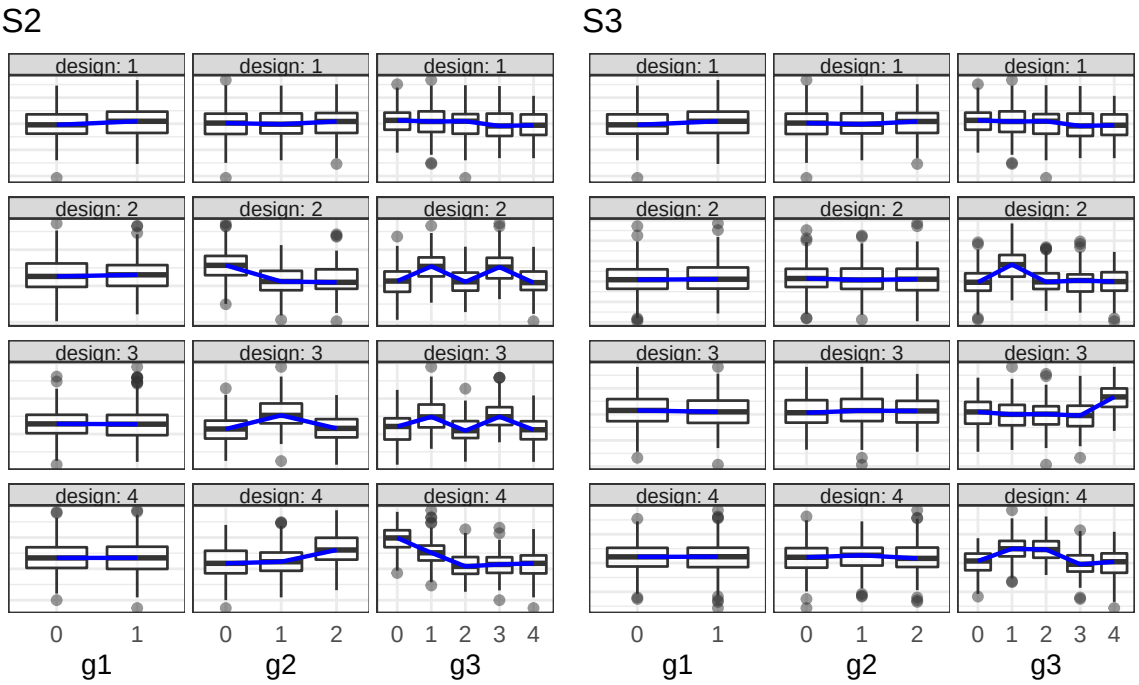

Figure 4.3: *Boxplots showing distributions of categories across different designs (rows) and granularities (columns) for scenarios S2 and S3. In S2, g2, g3 change across at least one design but g1 remains constant. Only g3 changes across different designs in S3.*

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 material.

4.3.3 Visual exploration of results

All of the approaches were fitted to each data design and to each combination of the considered parameters considered in Table 4.1. The formed clusters have to match the design, be well separated, and have minimal intra-cluster variation. MDS and parallel coordinate graphs are used to demonstrate the findings, as well as an index value plot to provide direction on the number of clusters. In the following plots, results for JS-NQT are reported, and results with JS-RS or wpd-based distances are in the supplementary material.

Figure 4.4 shows *sindex* plotted against the number of clusters (k) for the range of mean differences (rows) under the different scenarios (columns). This can be used to determine the number of clusters for each scenario. When *sindex* 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 *sindex* from that value of k. Thus, the number of clusters corresponds to the number of designs that were originally considered in each scenario.

Figure 4.5 shows separation of our clusters. It can be observed 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 4.6 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. Panel S1 shows that no variable stands out in the clustering, but the following two panels 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 4.6 (S1), (S2), and (S3) validate the construction of scenarios (S1), (S2), and (S3) respectively.

The JS-RS and wpd-based methods perform worse for $nT = 300$, then improve for higher nT evaluated in the study. However, 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, this is not a challenge.

In our study sample, the method JS-NQT outperforms the method JS-RS for smaller differences between categories. More testing, however, would be needed to to be confident in this conclusion.

4.4 Application

Clustering with the new distances is illustrated on the smart meter energy usage for a sample of customers from Department of the Environment and Energy (2018). The full data 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 starting and end dates. Additionally, there were missing values in many series. (The supplementary material contains details from checking for systematic missingness.) Because our proposed methods evaluate probability distributions rather than raw data, these data issues are not problematic, unless there is any systematic structure related to granularities.

Huge data sets present more complications for clustering. Clustering algorithms work well when there are well-separated clusters, with no nuisance variables or nuisance observations. When converting a series to granularities, many variables (each level of a granularity) are generated, possibly creating a slew of nuisance variables. Some customers may have a mix of energy use

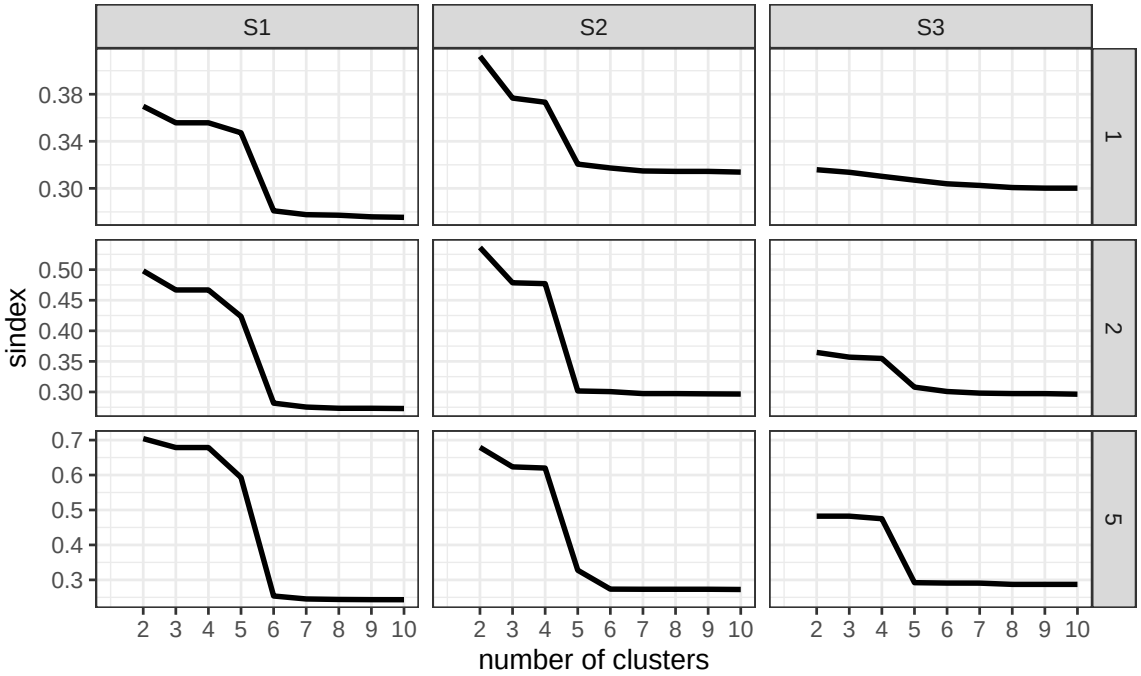

Figure 4.4: *Choosing optimal cluster number across the range of scenarios and mean differences used in the validation study, using the cluster separation index (sindex) for the JS-NQT. S1 has a sharp decrease in sindex from 5 to 6, whereas S2 and S3 have a decrease from 4 to 5, especially when mean difference is large, providing the recommended number of clusters to be 5, 4, 4, respectively. This precisely reflects the structure in designs that we would hope the clustering could recover.*

patterns, which could be considered nuisance observations located between major clusters. For this reason, we have chosen to select a small group of customers with relatively distinct and different patterns in order to illustrate the clustering more simply. Figure 4.7 shows the distribution across *hod*, *moy* and *wnwd* for the set of 24 customers used to illustrate clustering. The customers are displayed in two columns of 12 for space reasons. Each row, of each column, represents the profile of a single customer across different variables. Each customer is associated with an identifier of the form [*a*-*b*], where $a \in \{1,2,\ldots,24\}$ represents the customer-prototype id and $b \in \{1,2,\ldots,5\}$ indicates the label of the prototype in which a customer was placed. This is often a good approach to tackling a big analysis task, to start with a simpler task. The approach, however, is applicable to all customers.

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 small sample, and then significant cyclic granularities (variables) for them are chosen (as described in Section 4.4.1). The sample set is subsequently examined along all dimensions of interest, to ensure that they reveal some patterns

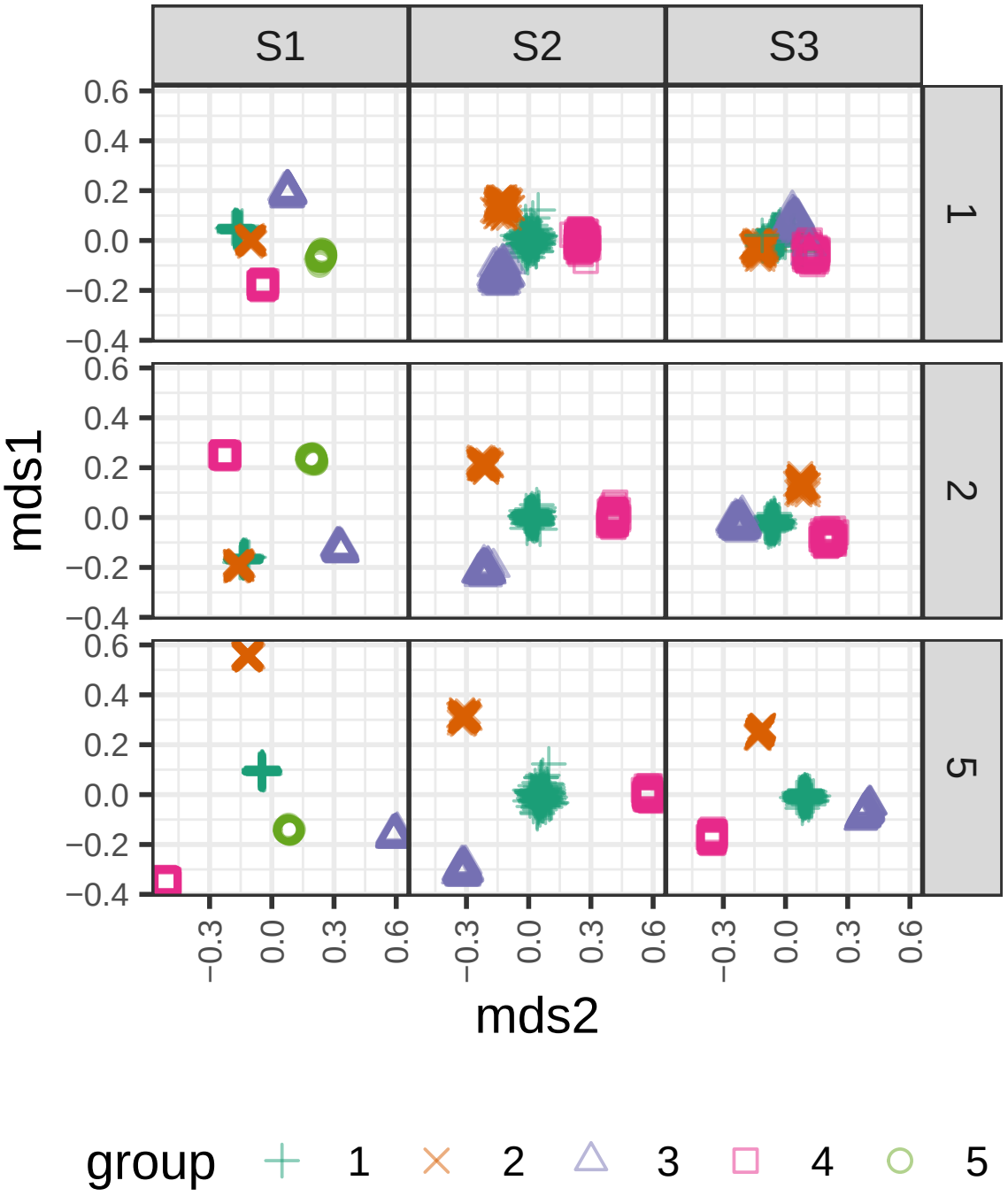

Figure 4.5: *MDS summary plots of clusters obtained from employing JS-NQT on the validation data sets. It illustrates the cluster separation for the range of mean differences (rows) under the different scenarios (columns). It can be observed that clusters become more compact and separated for higher 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.*

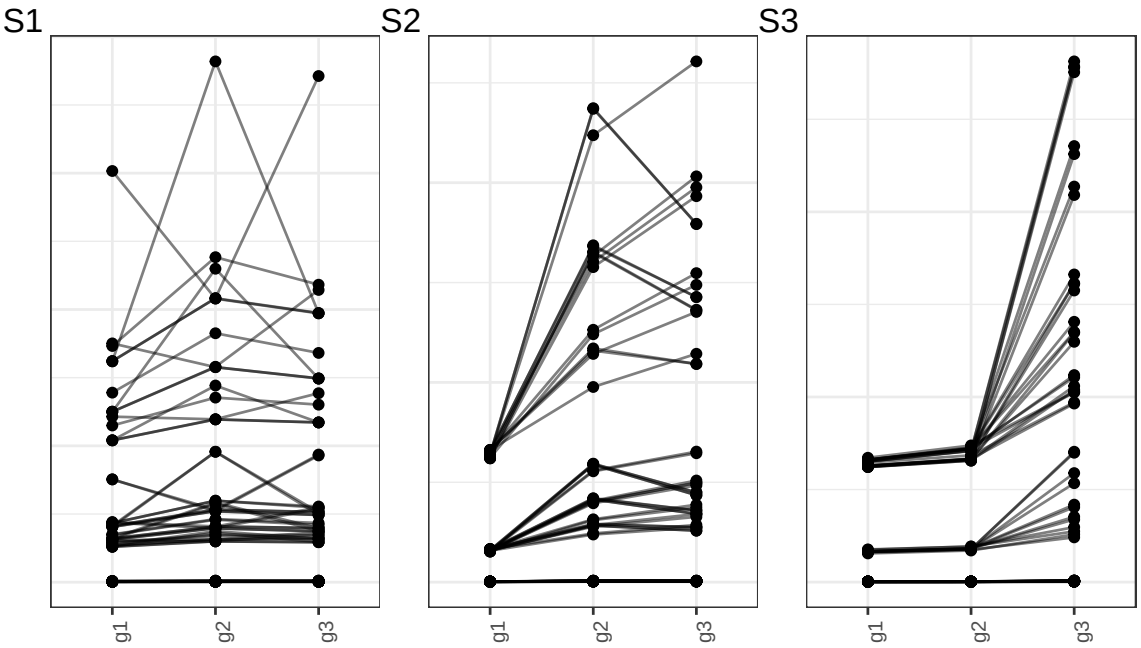

Figure 4.6: *Exploring the contribution of granularities in the clustering for scenarios S1, S2, S3, using parallel coordinate plots. Inter-cluster distances are displayed vertically. All three granularities g*1*, g*2*, and g*3 *have high inter-cluster distances for S1, suggesting all are important. In S2 g*1 *and in S3 both g*1 *and g*2 *have smaller inter-cluster distances, indicating that they did not contribute to clustering.*

across at least one specified variable (as described in Section 4.4.2). Because the data does not contain additional customer characteristics, we cannot explain why consumption varies, but can only identify how it varies.

4.4.1 Data filtering and variable selection

The steps for customer filtering and variable selection were:

1. Choose a smaller subset of randomly selected 600 customers with no implicit missing values for 2013.
2. Obtain *wpd* for all cyclic granularities considered for these customers. It was found that *hod* (hour-of-day), *moy* (month-of-year) and *wnwd* (weekend/weekday) are significant for most customers. We use these three granularities while clustering.
3. 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.

4. 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.

4.4.2 Selecting prototypes

It is common to filter data prior to fitting a supervised classification model using instance selection (Olvera-López et al., 2010) which removes observations that might impede the model building. For clustering, this is analogous to identifying and removing nuisance observations. Prototype selection is more severe than instance selection, because only a handful of cases is selected. Cutler and Breiman (1994) proposed a method called archetypal analysis which has inspired this approach but the procedure we have used follows Fan et al. (2021). First, dimension reduction such as t-SNE, MDS or PCA is used to project the data into a 2D space. Second, a few "anchor" customers far apart in 2D space are selected. Additional close neighbors to the anchors are selected. To check the selections relative to the full set of variables, we used a tour linked to a t-SNE layout using the R package `liminal` (Lee, 2021). This ensured that the final sample of clustered customers were also far apart in the high-dimensional space. (See the supplementary materials for further details.)

4.4.3 Clustering results

Clustering of the 24 prototypes was conducted with all three distances, JS-NQT, JS-RS and WPD, and is summarized in Figures 4.8, 4.9 and 4.10. The t-SNE visualization suggests that there are four well-separated clusters. It is possible that because the representation is only 2D, the fifth group from the original prototype selection is distinctly different in high dimensions. The *sindex* plots for the three methods indicate some disagreement: JS-NQT suggests 3, JS-RS suggests 2 or 5 and WPD suggests 3 or 5. JS-RS would appear to match the original prototypes with the five cluster solution, but it actually differs. Even though the *sindex* for JS-NQT suggests three clusters, the five cluster solution more closely matches the original prototypes. The WPD clustering provides a different grouping of the customers, and even though it disagrees with the original prototypes it is a useful grouping.

Figure 4.9 displays the summarized distributions across 4 and 5 clusters from JS-NQT clustering in (a) and (b) respectively, and helps to characterize each cluster. In the quantile plots the line

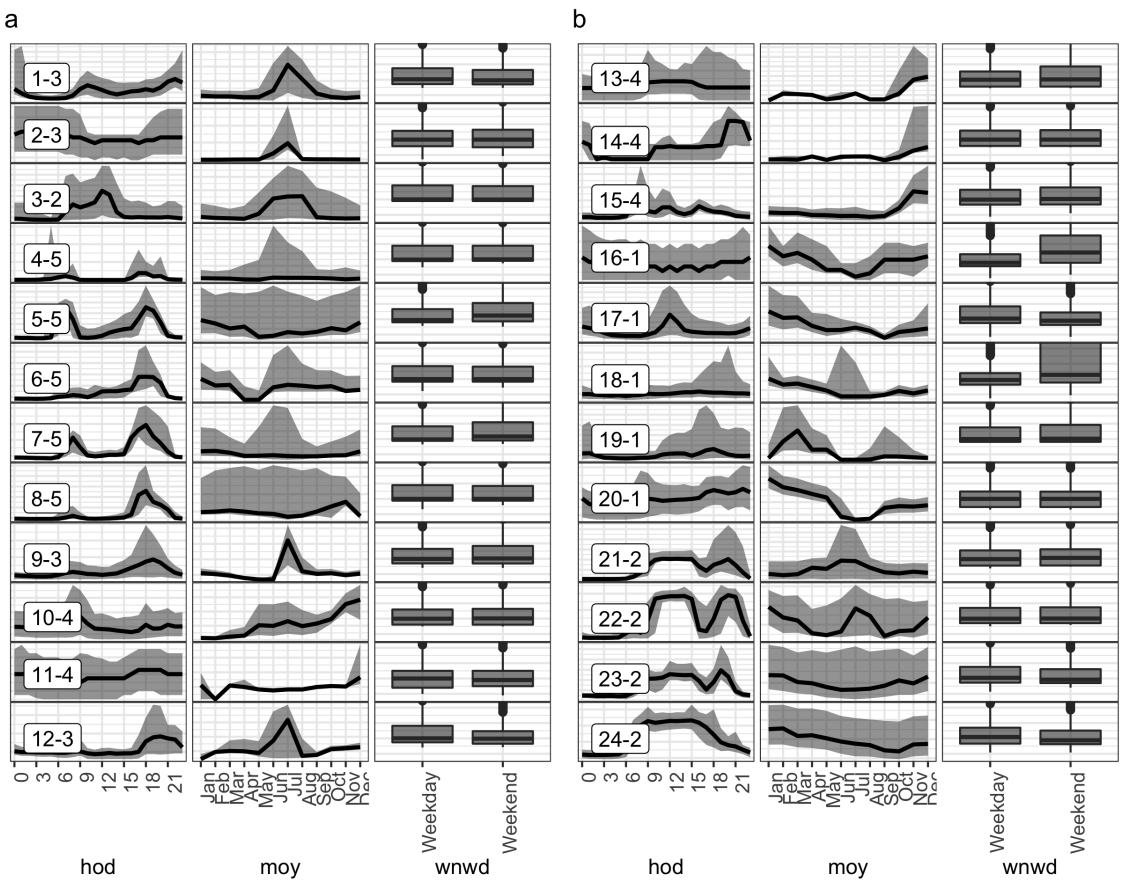

Figure 4.7: *The distribution of electricity demand across individual customers over three granularities* hod*,* moy*, and* wnwd *are shown for the* 24 *selected customers using quantile and box plots. They are split into batches of* 12 *in (a) and (b), with each row in (a) or (b) representing a customer. The number indicates a unique customer id and a prototype id. In each of the plots, the line represents the median, and the gray region shows the area between the 25th and 75th percentiles.*

represents the median, and the region shows the area between the 25^{th} and 75^{th} percentiles. The only difference between the four and five cluster solution is that A-4 divides further into B-4 and B-5. This additional division makes a clearer clustering, because it resolves the heterogeneity in *moy* creating a group (B-5, customers 1-3) which has a winter peak in usage, and a group (B-4, customers 16-20) which has a start of the year peak in usage. B-2 (customers 4-9) and B-1 (customers 21-24) have distinctive *hod* patterns but are both heterogeneous in *moy* and *wnwd*. B-3 (customers 10-15) has peak usage at the end of the year, but is heterogeneous on *hod* and *wnwd*. This clustering almost agrees with the clusters visible in the t-SNE plot. This display also serves as a visual summary of why three clusters are insufficient using JS-NQT approach, however it is recommended by *sindex* summary plots in Figure 4.8.

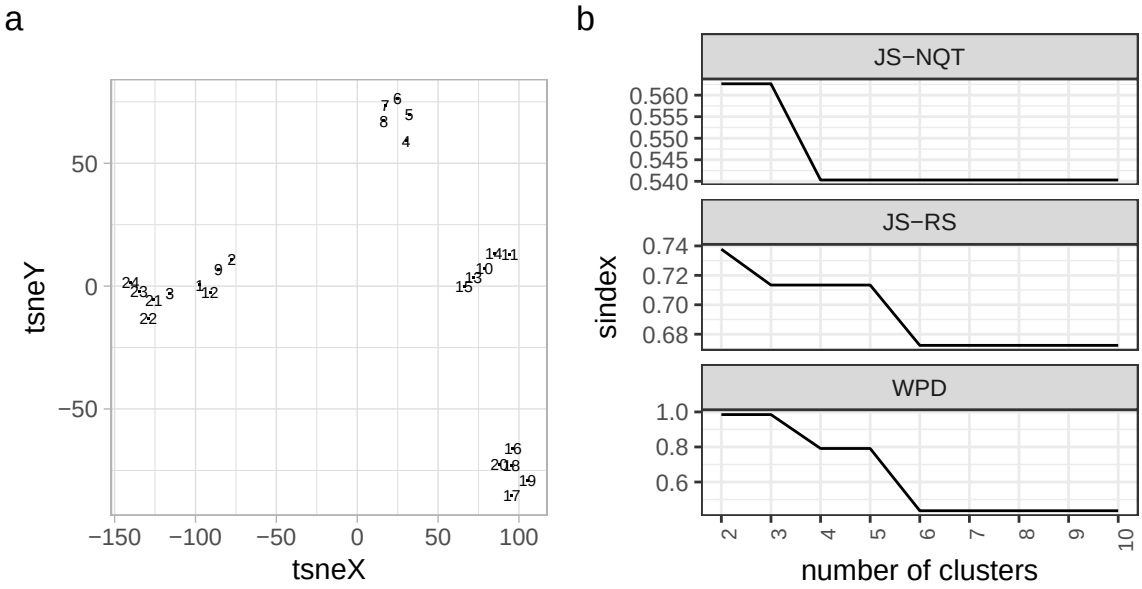

Figure 4.8: *Clustering summaries: (a) t-SNE computed on the* 24 *selected customers, and (b) separation index (*sindex*) for* 2-10 *clusters using JS-NQT, JS-RS and WPD. Various choices in number of clusters would be recommended. Four clusters are visible in t-SNE, although it might hide a fifth cluster because dimension reduction to 2D may be insufficient to see the difference. JS-NQT suggests* 3*, JS-RS suggests* 2 *or* 5 *and WPD suggests* 3 *or* 5*.*

Table 4.3: *Summary table from WPD clusters showing median wpd values (moy, hod, wnwd), cluster size (nobs) and the list of the customer-prototype id for each cluster with* 3 *and* 5 *number of clusters (k). It is to be noted that* $P-1$*,* $Q-1$ *and* $P-2$*,* $Q-2$ *are identical and* $P-3$ *gets split into* $Q-3$*,* $Q-4$ *and* $Q-5$*.*

k	group	nobs	moy	hod	wnwd	customer-prototype id
3	P-1	2	66.7	-2.7	39.4	18, 16
	P-2	9	129.0	-0.4	12.7	12, 9, 17, 2, 19, 13, 20, 10, 11
	P-3	13	14.9	24.5	4.4	8, 22, 23, 24, 14, 15, 3, 1, 4, 21, 5, 6, 7
5	Q-1	2	66.7	-2.7	39.4	18, 16
	Q-2	9	129.0	-0.4	12.7	12, 9, 17, 2, 19, 13, 20, 10, 11
	Q-3	4	88.2	29.4	2.6	22, 14, 4, 6
	Q-4	4	10.1	32.1	4.2	23, 21, 5, 7
	Q-5	5	14.9	11.9	4.6	8, 24, 15, 3, 1

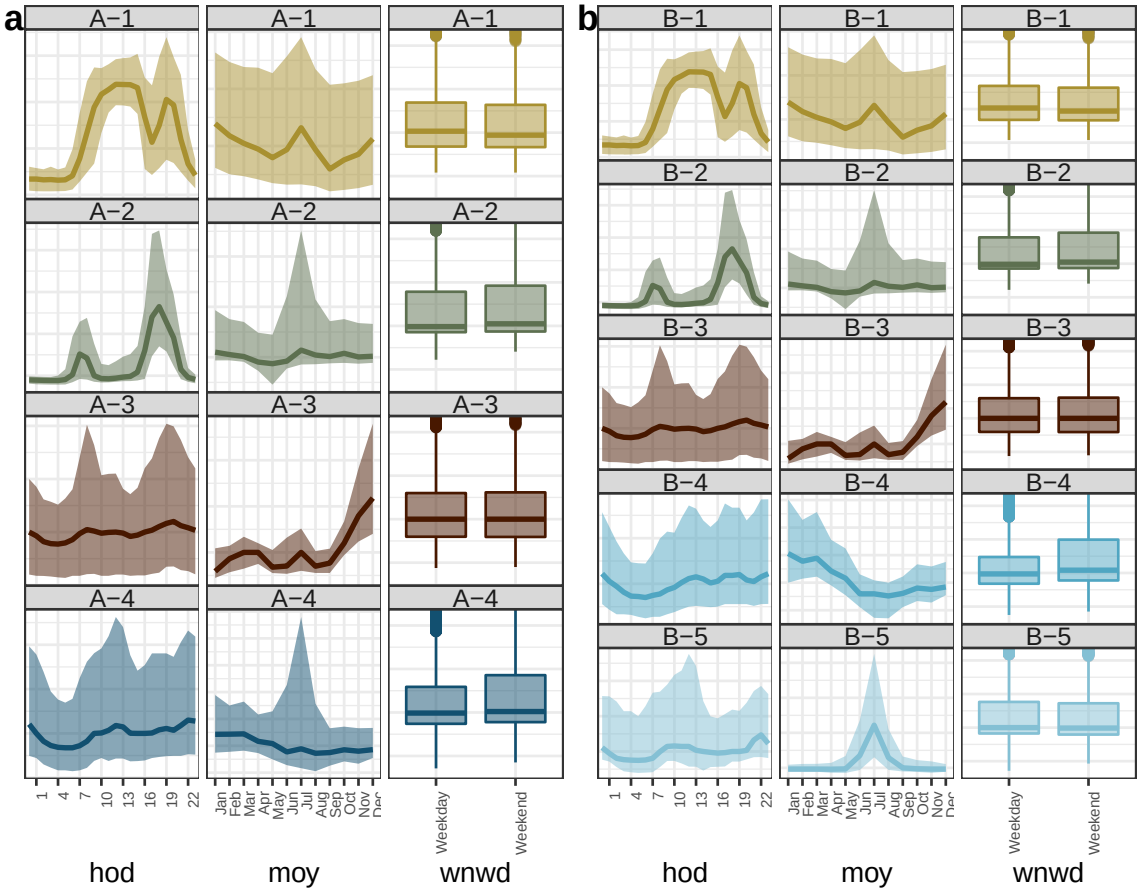

Figure 4.9: *Visual summary of why three clusters are insufficient for JS-NQT approach, however it is recommended by* sindex*. The plot shows four and five clusters from JS-NQT showing the distribution of electricity demand combined for all members over* hod*,* moy*, and* wnwd*. Groups A-1 (customers 21-24), A-2 (customers 4-9), and A-3 (customers 10-15) profiles correspond to Groups B-1, B-2, and B-3, respectively. Cluster A-4 splits into B-4 (customers 16-20) and B-5 (customers 1-3) to produce the five clusters, which better resolves the* moy *distribution.*

Figure 4.10 shows the *wpd* values of the 24 customers over *hod*, *moy*, and *wnwd* , colored by 3 (a) and 5 (b) clusters from WPD clustering using a parallel coordinate plot. The variables (*wpd* for different granularities) are standardized prior to clustering using WPD. In the display, the variables are sorted according to their separation across groups. This means that *wnwd* is the most important variable in distinguishing the groups, followed by *hod* and *moy* for both (a) and (b). Groups P-1 and P-2 correspond to Q-1 and Q-2 respectively. Cluster P-3 splits into Q-3, Q-4 and Q-5. Customers 16 and 18 are characterized by unusual high values of *wpd* on *wnwd* compared to the rest of the customers and hence form a group. This could again be verified from Figure 4.7, where these were the only two customers with a difference in their *wnwd* behavior. They are represented by P-1. P-2 has lower *wpd* for *hod* than *moy* and *wnwd*. P-3 behaves opposite to P-2 with higher *wpd* for *hod*

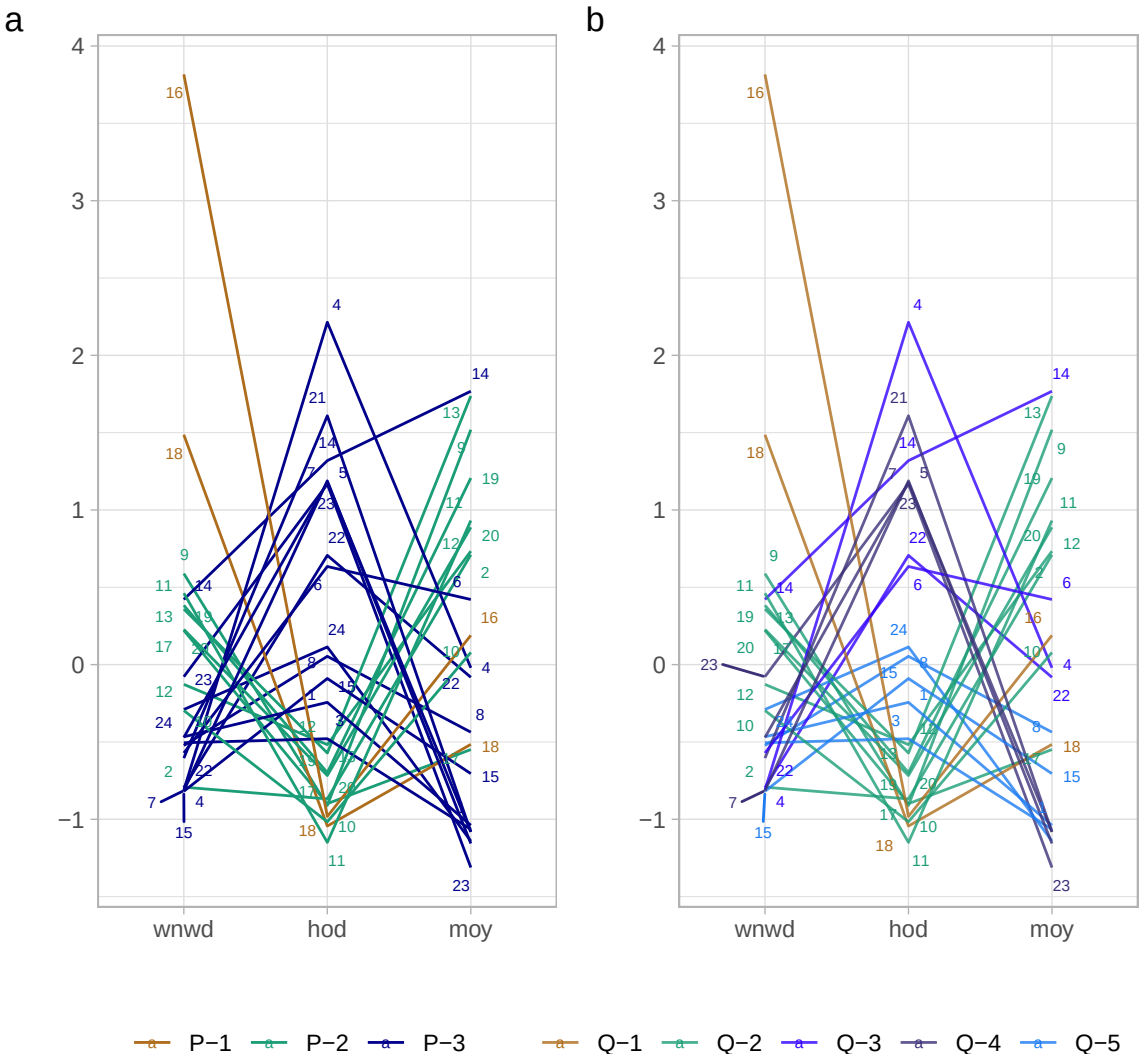

Figure 4.10: *Summary plots for three (a) and five (b) clusters from WPD approach showing the wpd values of each customers across* hod, moy, *and* wnwd *through a parallel coordinate plot. P-1 and P-2 groups correspond to Q-1 and Q-2, respectively. Cluster P-3 is subdivided into Q-3, Q-4, and Q-5. P-1 (customers 16, 18) is distinguished by high wpd on* wnwd *values. P-2 has lower wpd than* moy *and* wnwd *for* hod*. P-3 operates in the opposite way as P-2, with larger wpd for* hod *in comparison to* moy *and* wnwd*. For the* 5 *cluster solution, this group is divided into Q-3, Q-4, Q-5, which are distinct due to their different relative significance of* moy *and* wnwd*.*

compared to *moy* and *wnwd*. These are the customers who have some significant pattern across *hod* and this can again be validated by looking at Figure 4.7. For a 5 cluster solution, this group gets split into Q-3, Q-4 and Q-5 characterizing different relative significance of *moy* and *wnwd*. For example, Q-4 and Q-5 have almost no pattern across *moy*, but Q-3 has a *moy* pattern and thus it is reasonable to split them. The patterns could be different, but they are significant. Q-4 and Q-5 are separated because of their different significance of *hod*. All of these can also be verified from Table 4.3 which shows the cluster summaries with members and median values of *wpd* for the three variables.

In summary, none of the methods captured the five original prototypes exactly. JS-NQT was almost identical, but WPD produced quite a different grouping. This is quite a reasonable result and illustrates both the difficulties of clustering to obtain a particularly expected grouping and the ability to learn unexpected patterns in the data. It is possible that the JS-based distances were distracted by the presence of nuisance variables, levels of the granularities that do not contribute to clustering. This would also be supported by the results of the validation study, where clustering was less effective in S2 and S3, where only some granularities had differences between levels. Clustering using WPD is expected to produce quite different results because it will group only by overall value of a granularity, not a particular pattern. A cluster summary like Figure 4.9 is not possible because there may be different but equally interesting patterns (e.g. high evening *hod* and high daytime *hod*) in the same cluster. Simply it provides information that across a collection of customers this specific cluster has interesting patterns in a granularity (e.g. *hod*). One would need to post-process the cluster to separate specific patterns.

4.5 Discussion

4.5.1 Conclusions

Clustering methods perform effectively when clusters are well-separated and there are no nuisance variables or observations. In practice, most problems will contain nuisance observations or variables, and well-separated clusters are uncommon. As a result, picking a small group of observations with somewhat diverse and distinctive patterns (prototypes) rather than a brute force method is usually a good way to learn about the clustering and the data set, as demonstrated in this chapter. 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 for the prototypes. Probability distributions provide an intuitive method to characterize noisy, patchy, long, and unequal-length time series data. Distributions over cyclic granularities help to characterize 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.

4.5.2 Limitations and future research

There are a few areas to extend this research. Firstly, 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, the conditional 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 cannot 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 with such structured missingness. Moreover, experiments can also be run with non-hierarchy based clustering algorithms to verify if these distances work better with other algorithms.

For larger data sets, most clustering tasks become increasingly computationally intensive. The computational time for determining pairwise distances presented in this paper is determined by the number of cyclic granularities evaluated. For JS-NQT, the time required to compute distances between two observations based on 1, 2, and 3 cyclic granularities is 4.5s, 5.8s, and 8.2s, respectively.

For JS-RS, the time is comparable. The time required for WPD will be longer because the statistic *wpd* must be computed for each observation and cyclic granularity. Even with parallel computing, *wpd* is computationally costly for cyclic granularities with fewer levels. Further research may improve the efficiency of the computations to tackle large-scale clustering challenges.

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, contains all materials required to reproduce this article and the code is also available online in the supplementary materials. This article was created with R (R Core Team, 2020), `knitr` (Xie, 2015, 2020) and `rmarkdown` (Xie, Allaire, and Grolemund, 2018; Allaire et al., 2020). Graphics are produced with `ggplot2` (Wickham, 2016) and `GGally` (Schloerke et al., 2021).

Supplementary materials

Data and scripts: Data sets and R codes to reproduce all figures in this article (validation.R, application.R)

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

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`).

Chapter 5

Conclusion

This thesis presents methods for visualizing and analyzing distributions of large temporal data by deconstructing time into temporal granularities. This chapter summarizes the thesis content, outlines the original contributions, software development, and possible directions for future research.

5.1 Original contributions

Exploratory time series analysis entails numerous iterations of identifying and summarizing temporal dependencies. It is common practice to divide time into years, months, weeks, days, and so on in order to make inferences at both finer and coarser scales. In the literature, the formalization of these temporal deconstructions (granularities) is limited to linear time granularities such as hours, days, weeks, and months that respect the linear progression of time and are non-repeating. Cyclic granularities that are repeating in nature are useful for finding patterns in temporal data. They can be circular, quasi-circular, or aperiodic in nature. Hour-of-the-day and day-of-the-week are examples of circular granularities; the day-of-the-month is an example of a quasi-circular granularity; and public holidays and school holidays are examples of aperiodic granularities. Additionally, time deconstructions can be based on a time hierarchy. Thus, single-order-up granularities such as second of minute or multiple-order-up granularities such as second of hour can be envisioned. The definitions and rules defined in the literature for linear granularities are insufficient for describing various types of cyclic granularities.

Chapter 2 provides a formal characterization of cyclic granularities as well as tools for classifying and computing potential cyclic granularities from an ordered temporal index. It also allows for the manipulation of single- and multiple-order-up time granularities via cyclic calendar algebra. The approach is generalizable to non-temporal hierarchical granularities with an ordered index. Visualizing probability distributions conditional on one or more cyclic granularities is a powerful exploration tool. However, there may be too many cyclic granularities to look at manually for comprehensive exploration, and not all pairs of granularities can be effectively explored together. Chapter 2 also provides a recommendation on whether a pair of granularities can be meaningfully plotted or analyzed together (a "harmony") or when they cannot (a "clash" or "near-clash").

Cyclic granularities could be used to create a wide range of displays. And, when there are numerous granularities to choose from, deciding which one to display can be difficult. Moreover, only a few of them may be useful in revealing major patterns. In Chapter 3, the search for informative granularities is facilitated by selecting "significant" granularities. A cyclic granularity is referred to as "significant" if there is a significant distributional difference of the measured variable between different categories. Chapter 3 defines a distance measure to quantify these distributional differences. A higher value of the distance measure for a cyclic granularity or harmony implies that they could be interesting for further investigation, whereas a low value indicates that nothing noteworthy is unfolding. A threshold and, consequently, a selection criterion are chosen using a permutation test such that cyclic granularities with significant values of the distance measure are selected. In addition, the distance metric has been appropriately adjusted, allowing it to be compared not only across cyclic granularities with different numbers of categories but also across a set of time series. As a result, it can also be used to rank the displays according to their ability to capture the greatest amount of variation across one or multiple time series.

The ideas in Chapters 2 and 3 can be used for studying patterns in individual time series or comparing a few time series together. This is extended in Chapter 4 to allow for the exploration of distributions for multiple time series at the same time using unsupervised clustering. In the time series clustering literature, probability distributions across cyclic granularities have not been considered in determining similarity. However, such a similarity measure can be useful for characterizing the inherent temporal data structure of long, unequal-length time series in a way that is resistant to missing or noisy data while allowing for the detection of similar repeated

patterns. Chapter 4 proposes two approaches for calculating distances between time series based on probability distributions across cyclic granularities. The first approach considers two time series to be similar if the distributions of each category of one or more cyclic granularities are similar. The second approach considers two time series to be similar if they have a similar significance of patterns across different granularities. A similar significance does not imply a similar pattern, which is where this technique varies from the former. When the distances from these approaches are fed into a hierarchical clustering algorithm, they yield small groups of time series with similar distributions or significance over multiple granularities. Our method is capable of producing useful clusters for both approaches, as demonstrated by testing on a range of validation data designs and a sample of residential smart meter consumers.

5.2 Software development

This thesis focuses on translating research approaches into open source R packages for reproducibility and ease of use in other applications. So a significant amount of work has been devoted to the development of R packages `gravitas`, `hakear`, and `gracsr`, each of which corresponds to a chapter presented in this thesis.

5.2.1 gravitas

The `gravitas` package provides very general tools to compute and manipulate cyclic granularities and generate plots displaying distributions conditional on those granularities. The functions `search_gran()`, `create_gran()`, `harmony()`, `gran_advice()` and `prob_plot()` provides the entire workflow for an analyst to systematically explore large quantities of temporal data across different harmonies (pairs of granularities that can be analyzed together). This package was developed as part of my internship at Google Summer of Code, 2019. It has been on CRAN since January 2020. The website (https://sayani07.github.io/gravitas) includes full documentation and two vignettes about the package usage. There has been a total of 12K downloads from the RStudio mirror dating from 2020-11-01 to 2021-11-01. This package supplements the paper corresponding to Chapter 2, which has won the ACEMS Business Analytics Award 2021. The package can be generalized to non-temporal applications for which a hierarchical structure can be construed similar to time.

5.2.2 hakear

The R package `hakear` (https://github.com/Sayani07/hakear) provides tools for selecting and sorting significant cyclic granularities. The function `wpd()` computes the weighted pairwise distances (*wpd*) for each cyclic granularity or pair of granularities, and the function `select_harmonies()` chooses those with significant patterns and ranks them from highest to lowest *wpd*. This package is reliant on parallel processing using multiple multi-core computers for faster computation of *wpd*. The selected harmonies can be plotted using package `gravitas` for potentially interesting displays. Currently, `hakear` implements ideas presented in Chapter 3, but it will be integrated with `gravitas` in the future to explore distributions of a smaller number of time series.

5.2.3 gracsr

The R package `gracsr` (https://github.com/Sayani07/gracsr) has functions for exploring a large number of time series using the clustering methodology described in Chapter 4. The workflow begins with the function `scale_gran()`, which may be used to scale individual series using NQT/RS. The distances for the JS and WPD approaches are computed in the second phase of the workflow using `dist_gran()`/`dist_ wpd()`. The distances can then be used to do clustering with `clust_gran()`. The package has received a grant (AUD 3000) as part of the ACEMS Business Analytics Prize towards polishing the functions and preparing it for CRAN.

5.2.4 Computational resources

Simulation studies were carried out to study the behavior of *wpd*, build the normalization method as well as compare and evaluate different normalization approaches in Chapter 3. In Chapter 4, our methods were tested on several data designs with different parameters to evaluate their performance. *wpd* is computationally heavy for cyclic granularities with smaller levels. JS and WPD approaches are also computationally intensive when run on large number of customers. Hence, most of the scripts used to run these studies use parallel processing for better computational speed. R version 4.0.1 (2020-06-06) is utilized on the platform x86 64-apple-darwin17.0 (64-bit) operating on macOS Mojave 10.14.6, as well as the High Performance Computing (HPC) resources provided by MonARCH.

5.3 Limitations and future ideas

We address several limitations of the current framework that might serve as natural next steps for this work and some potential short- and long-term aims in future ideas.

5.3.1 Limitations

The time series are observed over a short period of time (1–3 years) in the motivating example of this research, and they are assumed to be stationary. But it is possible that the distributions change over time, even over a short period. For the smart meter example, the distribution for a customer moving to a different house or changing electrical equipment may change drastically. To detect these dynamic changes, non-stationarity in time series has to be incorporated while visualizing distributions and also computing distances for two non-stationary time series or one stationary and another non-stationary time series.

Additionally, it is possible that data for a whole category of cyclic granularity is unavailable or that there are insufficient observations to compute distributions. For example, a customer may not have data for a particular day of the week or month throughout their observation period. While visualizing probability distributions across categories in Chapters 2 and 3, this can be indicated by displaying dot plots instead of summarizing distributions. But the distances in Chapter 4 can not handle missing observations if they are structured like this. We would like to be able to think about designing a distance metric that can incorporate customers with structural missingness and also comprehend its implications while visually characterizing them. Another related direction is how to manage the swarm of nuisance variables produced by transforming the time series to cyclic granularities. Because the design of our existing framework treats each level of granularity as a variable, it is important to identify levels of granularity that do not contribute to clustering and remove them from the distance metric in order to improve the performance of the clustering algorithm.

5.3.2 Future ideas

The standard methodologies provide contradicting recommendations for the optimal number of clusters when there are nuisance variables (no contribution to clustering) or nuisance observations

(inlying and outlying observations falling between clusters). As a result, a more realistic short-term goal should be to test the persistence of clustering solutions in the presence of nuisance variables for the chosen number of clusters. This can be done by introducing slightly different samples into the study and observing how the clustering methods handle increased heterogeneity.

Another possible direction is to reduce computational time so that the proposed methods are easily scalable to many customers. The distance measure, *wpd* is computationally heavy even under parallel computation. Moreover, while computing distances between time series, the proposed methods compute all possible pairwise distances, which acts as a computational barrier. Faster nearest-neighbor search algorithms can be employed here to decrease the computational load.

A longer-term goal would be to create a similar framework for visualizing and analyzing multivariate time series data. With multiple time series available for each observation, the complexity of efficient exploration and visualization grows exponentially. In this case, conditional distributions include not only temporal dependency but also variables and their dependencies. This adds to the already high-dimensional data structures that result from studying distributions. This big problem can be tackled by first incorporating time's inherent characteristics while visualizing one or a few multivariate time series data. Unsupervised clustering can then be used to group multiple time series across multiple time granularities and variables. This is a method similar to the one used in this thesis for dealing with univariate time series.

Another long-term goal would be to develop a comparable framework with a spatial component for analysis. Electricity data is commonly found in a spatio-temporal format with some type of area code, such as a postcode or administrative unit. Assuming that households in certain residential neighborhoods have similar energy usage patterns, future research could look into clustering and visualization methods for analyzing probability distributions of spatio-temporal data, perhaps through some variation of the glyph maps.

Bibliography

Aghabozorgi, S, AS Shirkhorshidi, and TY Wah (2015). Time-series clustering–a decade review. *Information systems* **53**, 16–38.

Aigner, W, S Miksch, H Schumann, and C Tominski (2011). *Visualization of time-oriented data*. Springer Science & Business Media.

Allaire, J, Y Xie, J McPherson, J Luraschi, K Ushey, A Atkins, H Wickham, J Cheng, W Chang, and R Iannone (2020). *Rmarkdown: dynamic documents for r*. R package version 2.1. https://github.com/rstudio/rmarkdown.

Asimov, D (1985). The grand tour: a tool for viewing multidimensional data. *Siam journal on scientific and statistical computing* **6**(1), 128–143.

Bettini, C and R De Sibi (2000). Symbolic representation of user-defined time granularities. *Annals of mathematics and artificial intelligence* **30**(1), 53–92.

Bettini, C, CE Dyreson, WS Evans, RT Snodgrass, and XS Wang (1998). "A glossary of time granularity concepts". In: *Temporal databases: research and practice*. Ed. by O Etzion, S Jajodia, and S Sripada. Berlin, Heidelberg: Springer, pp.406–413.

Bettini, C, S Jajodia, and S Wang (2000). *Time granularities in databases, data mining, and temporal reasoning*. Springer Science & Business Media.

Borg, I and PJ Groenen (2005). *Modern multidimensional scaling: theory and applications*. Springer Science & Business Media.

Buja, A, D Cook, H Hofmann, M Lawrence, EK Lee, DF Swayne, and H Wickham (2009). Statistical inference for exploratory data analysis and model diagnostics. *Royal society philosophical transactions a* **367**(1906), 4361–4383.

Chicco, G and JS Akilimali (2010). Renyi entropy-based classification of daily electrical load patterns. *Iet generation, transmission & distribution* **4**(6), 736–745.

Cook, D and DF Swayne (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.

Cutler, A and L Breiman (1994). Archetypal analysis. *Technometrics* **36**(4), 338–347.

Dang, TN and L Wilkinson (2014). ScagExplorer: exploring scatterplots by their scagnostics. In: *2014 IEEE pacific visualization symposium*, pp.73–80.

Dasu, T, DF Swayne, and D Poole (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.

Department of the Environment and Energy (2018). *Smart-grid smart-city customer trial data*. Australian Government, Department of the Environment and Energy. https://data.gov.au/dataset/4e21dea3-9b87-4610-94c7-15a8a77907ef.

Dunn, JC (1973). A fuzzy relative of the ISODATA process and its use in detecting compact Well-Separated clusters. *Journal of cybernetics* **3**(3), 32–57.

Dyreson, C, W Evans, H Lin, and R Snodgrass (2000). Efficiently supporting temporal granularities. *IEEE transactions on knowledge and data engineering* **12**(4), 568–587.

Edgington, E and P Onghena (2007). *Randomization tests*. CRC press.

Fan, H, P Liu, M Xu, and Y Yang (2021). Unsupervised visual representation learning via dual-level progressive similar instance selection. *Ieee transactions on cybernetics*, 1–11.

Faraway, JJ (2016). *Extending the linear model with R : generalized linear, mixed effects and nonparametric regression models, second edition*. 2nd Edition. Chapman and Hall/CRC.

Fisher, RA (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics* **7**(2), 179–188.

Fisher, RA (1992). "Statistical methods for research workers". In: *Breakthroughs in statistics*. Springer, pp.66–70.

Goodwin, S and J Dykes (2012). Visualising variations in household energy consumption. In: *2012 IEEE conference on visual analytics science and technology (VAST)*. Seattle, WA: IEEE, pp.217–218.

Grolemund, G and H Wickham (2011). Dates and times made easy with lubridate. *Journal of statistical software* **40**(3), 1–25.

Gupta, S, R Hyndman, D Cook, and A Unwin (2020). *Gravitas: explore probability distributions for bivariate temporal granularities*. R package version 0.1.3. https://github.com/Sayani07/gravitas/.

Gupta, S, RJ Hyndman, and D Cook (2021). *Detecting distributional differences between temporal granularities for exploratory time series analysis*. Working Paper 20/21. Department of Econometrics & Business Statistics, Monash University. https://robjhyndman.com/publications/hakear/.

Gupta, S, RJ Hyndman, D Cook, and A Unwin (2021). Visualizing probability distributions across bivariate cyclic temporal granularities. *Journal of computational & graphical statistics*. to appear.

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 (2019). "Cluster validation by measurement of clustering characteristics relevant to the user". In: *Data analysis and applications 1*. Hoboken, NJ, USA: John Wiley & Sons, Inc., pp.1–24.

Hintze, JL and RD Nelson (1998). Violin plots: a box plot-density trace synergism. *American statistician* **52**(2), 181–184.

Hofmann, H, H Wickham, and K Kafadar (2017). Letter-value plots: boxplots for large data. *Journal of computational & graphical statistics* **26**(3), 469–477.

Hyndman, RJ (1996). Computing and graphing highest density regions. *American statistician* **50**(2), 120–126.

Hyndman, RJ and Y Fan (1996). Sample quantiles in statistical packages. *American statistician* **50**(4), 361–365.

Jolliffe, IT and J Cadima (2016). Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society a: mathematical, physical and engineering sciences* **374**(2065), 20150202.

Krzysztofowicz, R (1997). Transformation and normalization of variates with specified distributions. *J. hydrol.* **197**(1-4), 286–292.

Kullback, S and RA Leibler (1951). On information and sufficiency. *The annals of mathematical statistics* **22**(1), 79–86.

Laird-Smith, J (2020). *Gs: a grammar of recurring calendar events*. R package version 0.0.0.9000. https://github.com/jameslairdsmith/gs.

Lee, S (2021). *Liminal: multivariate data visualization with tours and embeddings*. R package version 0.1.2. https://CRAN.R-project.org/package=liminal.

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

Liao, TW (2007). A clustering procedure for exploratory mining of vector time series. *Pattern recognition* **40**(9), 2550–2562.

Lin, J (1991). Divergence measures based on the shannon entropy. *Ieee transactions on information theory* **37**(1), 145–151.

Majumder, M, H Hofmann, and D Cook (2013). Validation of visual statistical inference, applied to linear models. *Journal of the american statistical association* **108**(503), 942–956.

McGill, R, JW Tukey, and WA Larsen (1978). Variations of box plots. *American statistician* **32**(1), 12–16.

Melnykov, V (2013). Challenges in model-based clustering. *Wiley interdisciplinary reviews: computational statistics* **5**(2), 135–148.

Menéndez, ML, JA Pardo, L Pardo, and MC Pardo (1997). The Jensen-Shannon divergence. *Journal of the franklin institute* **334**(2), 307–318.

Motlagh, O, A Berry, and L O'Neil (2019). Clustering of residential electricity customers using load time series. *Applied energy* **237**, 11–24.

Ndiaye, D and K Gabriel (2011). Principal component analysis of the electricity consumption in residential dwellings. *Energy build.* **43**(2), 446–453.

Ning, P, XS Wang, and S Jajodia (2002). An algebraic representation of calendars. *Annals of mathematics and artificial intelligence* **36**(1), 5–38.

Olvera-López, JA, JA Carrasco-Ochoa, JF Martínez-Trinidad, and J Kittler (2010). A review of instance selection methods. *Artificial intelligence review* **34**(2), 133–143.

Ozawa, A, R Furusato, and Y Yoshida (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.

Potter, K, J Kniss, R Riesenfeld, and CR Johnson (2010). Visualizing summary statistics and uncertainty. **29**(3), 823–832.

R Core Team (2020). *R: a language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/.

Reingold, EM and N Dershowitz (2018). *Calendrical calculations*. en. 4th. Cambridge University Press.

Rhodes, JD, WJ Cole, CR Upshaw, TF Edgar, and ME Webber (2014). Clustering analysis of residential electricity demand profiles. *Applied energy* **135**, 461–471.

Rousseeuw, PJ (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational & applied mathematics* **20**, 53–65.

Schloerke, B, D Cook, J Larmarange, F Briatte, M Marbach, E Thoen, A Elberg, and J Crowley (2021). *Ggally: extension to 'ggplot2'*. R package version 2.1.1. https://CRAN.R-project.org/package=GGally.

Tibshirani, R, G Walther, and T Hastie (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.

Tukey, JW (1977). *Exploratory data analysis*. Reading, Mass.: Addison-Wesley.

Tukey, JW and PA Tukey (1988). Computer graphics and exploratory data analysis: an introduction. *The collected works of john w. tukey: graphics: 1965-1985* **5**, 419.

Tureczek, A, PS Nielsen, and H Madsen (2018). Electricity consumption clustering using smart meter data. en. *Energies* **11**(4), 859.

Tureczek, AM and PS Nielsen (2017). Structured literature review of electricity consumption classification using smart meter data. *Energies* **10**(5), 584.

Ushakova, A and S Jankin Mikhaylov (2020). Big data to the rescue? challenges in analysing granular household electricity consumption in the united kingdom. *Energy research & social science* **64**, 101428.

Ushey, K (2019). *Renv: project environments*. R package version 0.7.0-54. https://rstudio.github.io/renv.

Van der Maaten, L and G Hinton (2008). Visualizing data using t-sne. *Journal of machine learning research* **9**(11).

Vaughan, D (2020). *Almanac: tools for working with recurrence rules*. R package version 0.1.1. https://CRAN.R-project.org/package=almanac.

Wang, E, D Cook, and RJ Hyndman (2020a). A new tidy data structure to support exploration and modeling of temporal data. *Journal of computational and graphical statistics* **29**(3), 466–478.

Wang, E, D Cook, and RJ Hyndman (2020b). Calendar-based graphics for visualizing people's daily schedules. *Journal of computational and graphical statistics* **29**(3), 490–502.

Wegman, EJ (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the american statistical association* **85**(411), 664–675.

Wertheimer, M (1938). "Gestalt theory". In: *A source book of gestalt psychology*. Kegan Paul, Trench, Trubner & Company, pp.1–11.

Wickham, H (2016). *Ggplot2: elegant graphics for data analysis*. Springer-Verlag New York. http://ggplot2.org.

Wickham, H and G Grolemund (2016). *R for data science: import, tidy, transform, visualize, and model data*. Sebastopol, California: O'Reilly Media.

Wickham, H and L Stryjewski (2012). *40 years of boxplots*. Tech. rep. had.co.nz. https://vita.had.co.nz/papers/boxplots.html.

Wilke, CO (2020). *Ggridges: ridgeline plots in 'ggplot2'*. R package version 0.5.2. https://CRAN.R-project.org/package=ggridges.

Wilkinson, L (1999). *The grammar of graphics*. New York: Springer.

Wilkinson, L, A Anand, and R Grossman (2005). Graph-theoretic scagnostics. In: *Ieee symposium on information visualization, 2005. infovis 2005.* IEEE, pp.157–164.

Xie, Y (2015). *Dynamic documents with R and knitr*. 2nd. Boca Raton, Florida: Chapman and Hall/CRC. https://yihui.name/knitr/.

Xie, Y (2016). *Bookdown: authoring books and technical documents with R markdown*. Boca Raton, Florida: Chapman and Hall/CRC. https://github.com/rstudio/bookdown.

Xie, Y (2020). *Knitr: a general-purpose package for dynamic report generation in r*. R package version 1.28. https://yihui.org/knitr/.

Xie, Y, JJ Allaire, and G Grolemund (2018). *R markdown: the definitive guide*. Boca Raton, Florida: Chapman and Hall/CRC. https://bookdown.org/yihui/rmarkdown.

Xu, D and Y Tian (2015). A comprehensive survey of clustering algorithms. *Annals of data science* **2**(2), 165–193.