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 2021

Department of Econometrics and Business Statistics

# Contents

```
## Warning: package 'knitr' was built under R version 4.0.2
```

# Copyright notice

# Abstract

The research was motivated by the desire to understand the Australian smart meters 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 overwhelming.

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 **??**) 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". "harmony" denotes pairs of granularities that could be analyzed together and reduces the search from all possible options. This approach is still overwhelming for human consumption because there would still be a huge number of harmonies. Hence, the second contribution (Chapter **??**) refines the search of 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 third contribution (Chapter **??**) builds upon the first two to provide methods for exploring heterogeneities in repetitive behavior for many households 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

I have not renumbered sections of submitted or published papers in order to generate a consistent presentation within the thesis.

**Student name:** Sayani Gupta

**Student signature:**

**Date:** 2021-11-05

# Acknowledgements


## 0.1 supervisors

Rob and Di are phenomenal leaders in their respective fields, who lead by example. I am inspired by their creativity, wisdom, discipline, and dedication to really contribute to the society through their research. Thank you for constantly pushing me to improve as a programmer and researcher, and for regularly sharing best 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 I was 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 a conscious decision to question existing stereotypes and biases. Thank you Di for being a fantastic female role model.

## 0.2 department and PhD colleagues

## 0.3 R community

## 0.4 friends

PhD in Covid has been harder 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 for being an epitome of kindness and acting as a pillar of support. Thank you Tushar, Samarpita and Nairita for always motivating me and believing in me when I didn't believe in myself. Thank you Ian for always putting things in perspective when I lost sight. Thank you each one of you for being available,

listening to me, and not giving up on me at my lowest. Thanks to my housemates Anjali, Surbhi and Dulaji for the fun company and conversations that kept me sane in the covid lockdowns.

## 0.5 family

Finally, a big thanks 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 endeavours. Thanks to my brother (Avijit) for always having my back and inspiring me to dream bigger. Thank you, Juhi (sister-in-law) for instilling in me the value of organisation in all aspects of life and the importance of prioritizing my physical health from time to time. Thank you all for your part in supporting me in this journey. I love you.

# Preface

Chapter **??** has been accepted by 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 **??** and Chapter **??** are yet to be submitted.

### Open and reproducible research

This thesis is written in R Markdown (**rmarkdown**) with **bookdown** (**bookdown**), using **renv** (**renv**) to create reproducible environments. The online version of this thesis is hosted at `https://sayani.netlify.app/`, powered by Netlify. 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`.

### License

# Chapter 1

# Introduction

With the availability of data at more and more 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. For example, data collected at an hourly scale could be analyzed using coarser temporal scales such as days, months or quarters. This approach requires deconstructing time in various possible ways. Moreover, often it might be interesting to capture calendar or periodic effects like month-of-year, day-of-week or hour-of-day. They help us in answering questions like if certain levels of those time deconstructions are characterized by unusual/routine values of the observed variable. For example, certain days of the week or months of the year are likely to be characterized by higher values. It is important to be able to navigate through all the temporal deconstructions that accommodate for periodicities to have multiple perspectives of the observed data. This idea aligns with the notion of EDA (Tukey 1977) which emphasizes the use of multiple perspectives on data to help formulate hypotheses before proceeding to hypothesis testing.

This chapter will provide an introduction to the study by first discussing the background and context, followed by the research aims, objectives and questions.

The motivation for this work comes from the desire to provide methods to better understand large quantities of measurements on energy usage reported by smart meters in household across Australia, and indeed many parts of the world. Smart meters currently provide half-hourly use in kwh for each household, from the time that they were installed, some as early as 2012. Households are distributed geographically, and have different demographic 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 of usage across households, total kwh used each half hour by state, for example, because energy companies need to understand maximum loads that they will have to plan ahead to accommodate. But studying overall energy use hides the distributions of usage at finer scales, and making it more difficult to find solutions to improve energy efficiency.

However, restructuring time in this manner leads to restructured data where each level of the time deconstructions correspond to multiple values of the observed variable. It is common to see aggregation or summarization of these multiple observations with a unique value to study calendar effects. For example, using aggregates of usage across each hour/half-hour has been common in the literature because energy companies need to plan for maximum loads on the network. But studying overall energy use hides the distributions of usage at finer scales, and making it more difficult to find solutions to improve energy efficiency. Summarizing the probability distribution of these multiple observations to capture both the shape and uncertainty could be a potential way to understand the underlying distribution of these observations. Studying probability distributions is likely to focus on features of the data which are not transparent through raw data or a unique summary statistic.

Hence, the overarching research goal is to study the periodic behavior of temporal data in a structured way by studying the probability distributions by best exploiting the characteristics of time. Slicing and dicing the data in all possible temporal scales as suggested by EDA can be a daunting task as it leads to a myriad of possibilities. This inspires the research presented in this thesis, which aims to provide a platform to systematically explore periodicities in temporal data and support finding regular patterns or anomalies, explore clusters of behaviors or summarize the behavior. The first part of the work discusses computation of all possible combinations of cyclic time granularities and a graphical mapping such that distributions of a numeric response variable is displayed across combinations of two cyclic granularities. Even analyzing the distribution of the measured variable across two cyclic granularities at once could amount to displaying many plots in search of potential patterns. Thus, the first part of the research (**Gupta2020-vo**) also introduces "harmony" to denote pairs of granularities that could be analyzed together and reduces the search

from all possible options. But this approach is still overwhelming for human consumption because there would still be huge number of harmonies. Hence, the second part of the research extends this work and narrows the search further by finding pair of cyclic granularities which are informative enough and rank them according to their importance. However, to explore periodic patterns of many households, we have to resort to clustering which has been addressed in the third part of the research. Although the motivation came through the smart meter example, this is a problem that is relevant to any temporal data observed more than once per year.

## 1.1 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`.

## 1.2 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 can 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`.

## 1.3 Clustering time series based on probability distributions across temporal granularities

With more and more time series data being collected at much finer temporal resolution, for a longer length of time, and for a larger number of individuals/entities, time series clustering research is getting a lot of traction. The sort of noisy, patchy, uneven, and asynchronous time series that is typical in many disciplines limits similarity searches among these lengthy time series. In this work, we suggest a method for overcoming these constraints by grouping time series based on probability distributions over cyclic temporal granularities. Cyclic granularities are temporal deconstructions of a time period into units such as hour-of-the-day, work-day/weekend, and so on, and can be helpful for detecting repeating patterns. Looking at probability distributions across cyclic granularities results in an approach that is robust to missing or noisy data, aids in dimension reduction, and ensures small pockets of similar repeated behaviours. The proposed method was tested using a collection of residential electricity customers. The simulated and empirical evidence demonstrates that our method is capable of producing meaningful clusters.

## 1.4 Summary

The thesis is structured as follows. Chapter **??** 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**. Chapter **??** 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**. The chapter **ref** (ch: gracsr) provides similarity measures for comparing multiple time series.This is in the developing R package **gracsr**. Chapter **??** summarizes the software tools developed for the work, and discusses some future plans.

# 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 (**aigner2011visualization**).

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) (**Tukey1977-jx**) 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 (**wang2020calendar**) 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 (**Wickham2009pk**) 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 **Bettini1998-ed**. 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 (**Grolemund2011-vm**) 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 (**Goodwin_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 in weather or astronomical data.

This work provides tools for systematically exploring bivariate granularities within the tidy workflow (**wickham2016r**). 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 **??** provides some background material on linear granularities and calendar algebra for computing different linear granularities. Section **??** 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 **??**. Section **??** discusses the role of different factors in constructing an informative and trustworthy visualization. Section **??** 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 **??**.

## 2.2 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 **Bettini1998-ed** 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 **Bettini1998-ed bottom granularity**. **??** 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.

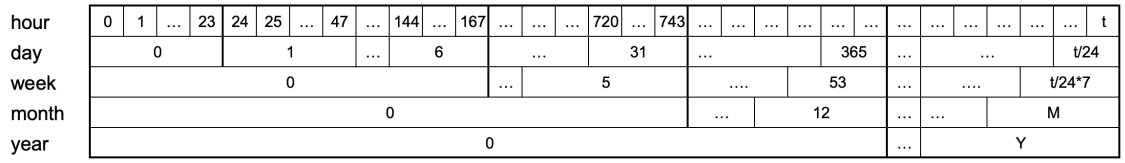



**Figure 2.1:** *Illustration of time domain, linear granularities and index set. Hour, day, week, month and year are linear granularities and can also be considered to be time domains. These are ordered with ordering guided by integers and hence is unidirectional and non-repeating. Hours could also 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 subset 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 **??** 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** (**Bettini2000-qk**).

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 (**Bettini2000-vy**). 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 *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 consist 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_2000**. Linear time granularities are computed through "calendar algebra" operations (**Ning_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 $P(B,G)$.*

**??** 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

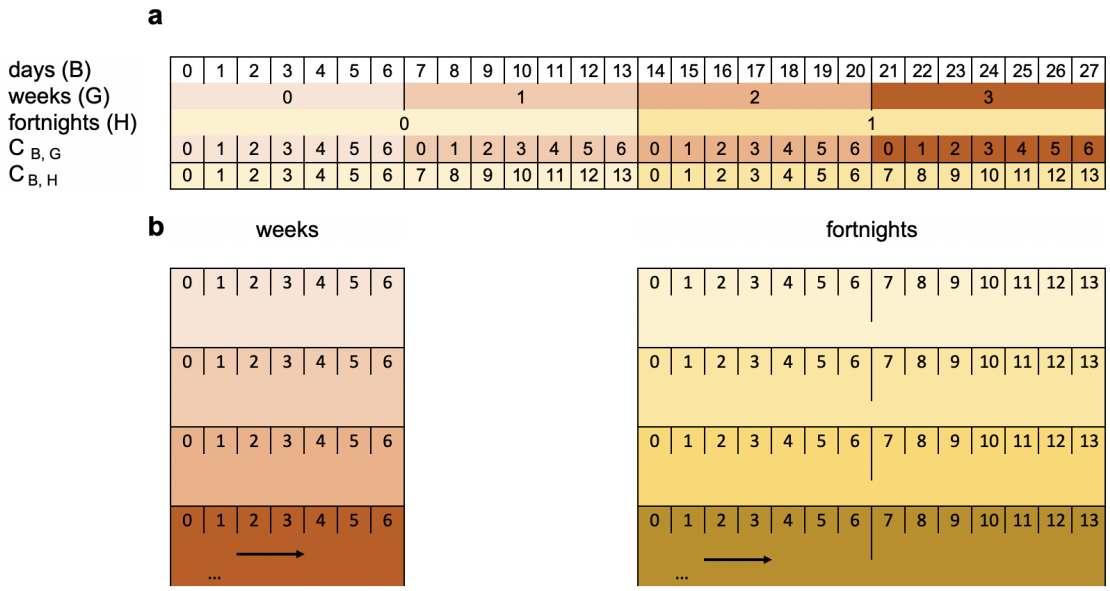



**Figure 2.2:** *Index sets for some linear and circular granularities (a). Circular granularities can be constructed by slicing the linear granularity into pieces and stacking them (b).*

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 **??** shows several circular granularities constructed using minutes as the bottom granularity.

| 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 | $C_5 = \lfloor z/24 * 60 \rfloor \bmod 7$ | $P_5 = 7$ |

**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.*

### 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. **??** shows some examples of quasi-circular granularities.

| 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 | $P_3 = 366, 365$ |

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

**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 **??**. 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 (**Reingold2001-kf**). 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 **??**) 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 **??**. 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 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.

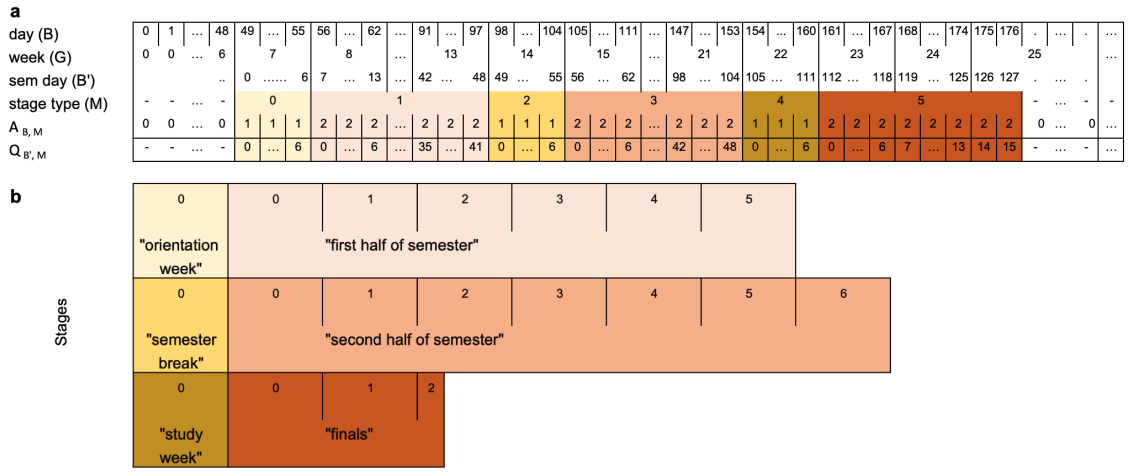



**Figure 2.3:** *Quasi-circular and aperiodic cyclic granularities illustrated by linear (a) and stacked-displays (b) progression of time. The linear display shows the granularities days (B), weeks (G), semester days ($B'$), and stages of a semester (M) indexed over a linear representation of time. The granules of $B'$ is 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.*

### 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 $G_\ell$ and $G_m$ represent the linear granularity of order $\ell$ and m respectively with $\ell < m$. Then $K \equiv P(\ell, m)$ represents the period length of the grouping $(G_\ell, G_m)$, if $C_{G_\ell, G_m}$ is a circular granularity and $K \equiv k(\ell, m)$ represents the operation to obtain $G_m$ from $G_\ell$, if $C_{G_\ell, G_m}$ is quasi-circular.*

For example, **??** 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.

**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 |

Like most calendars, the Mayan calendar used the day as the basic unit of time (**Reingold2001-kf**). 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 **??**), 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_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 (**??**).

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

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

- $C_{G_\ell,G_{m'}}$ 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 $T_w$.

**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 |

For example, the Gregorian calendar (**??**) has day-of-month as a single-order-up quasi-circular granularity, with the other granularities being circular. Using Equations (**??**) and (**??**), 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 (**??**).

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

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

### 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 (**??**) and (**??**).

**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** | $C_1$ | $C_2$ | $\cdots$ | $C_{N_C}$ |
|---|---|---|---|---|---|---|
| | | | | | | |

## 2.4 Data structure

Effective exploration and visualization benefit from well-organized data structures. **wang2020tsibble** 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. **??** 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 $B_\ell$ 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 ($A_k$, $B_\ell$). 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 (**Wilkinson1999-nk**).

### 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 **??**). 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 box plots (**Tukey1977-jx**) and its variants such as notched box plots (**McGill1978-hg**) or other variations as mentioned in **boxplots**. 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 (**Hintze1998-zi**), summary plot (**Potter2010-qc**), ridge line plots (**R-ggridges**), and highest density region (HDR) plots (**Hyndman1996-ft**). The less commonly used Letter-Value plots (**Hofmann2017-sg**) 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` (**R-gravitas**), 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 **??**) or near-clashes lead to potentially ineffective graphs. Harmonies tend to be more useful for exploring patterns. **??**a 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 $B_\ell$ represents a group/facet. Gestalt theory 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 $B_\ell$ levels rather than $A_k$ levels. For example, **??**b shows the letter value plot across weekday/weekend partitioned by quarters of the year and **??**c shows the same two cyclic granularities with their mapping reversed. **??**b helps us to compare weekday and weekend within each quarter and **??**c 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 (**smart-meter**) 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.

#### Cyclic granularities search and computation

The tsibble object `smart_meter10` from R package `gravitas` (**R-gravitas**) 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 **??**.

### 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. **??** 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 **??**).

**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 |

A few harmony pairs are displayed in **??** to illustrate the impact of different distribution plots and reverse mapping. For each of **??**b 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 **??**b and c respectively. **??**a 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 **??**b, the black line is the median, whereas 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 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 **??**c. 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 (**??**a), 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 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

**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 |

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 **??**. 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 (**??**a) 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. **??**b 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 playstyle. 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`(**R-almanac**) and `gs`(**R-gs**) 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` (**knitr**) and `rmarkdown` (**rmarkdown**).

## 2.8 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-language**) package `gravitas` (**R-gravitas**), 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`).

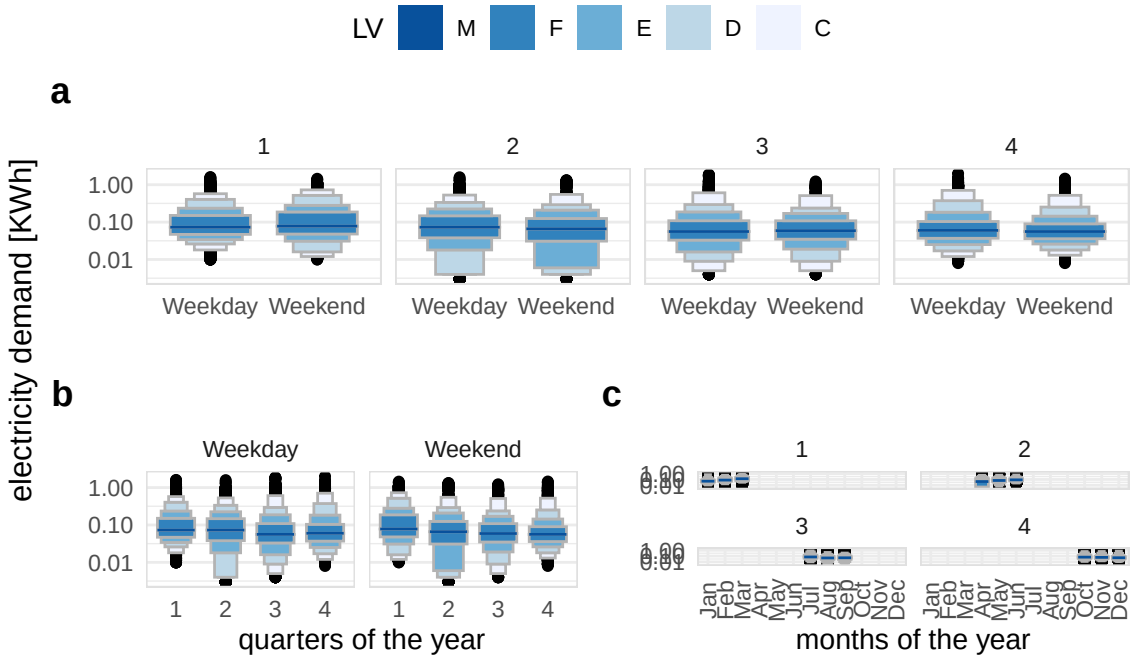



**Figure 2.4:** *Distribution of energy consumption displayed as letter value plots, illustrating harmonies and clashes, and how mappings change emphasis:* ***a*** *weekday/weekend faceted by quarter-of-year produces a harmony,* ***b*** *quarter-of-year faceted by weekday/weekend produces a harmony,* ***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, there is 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.*

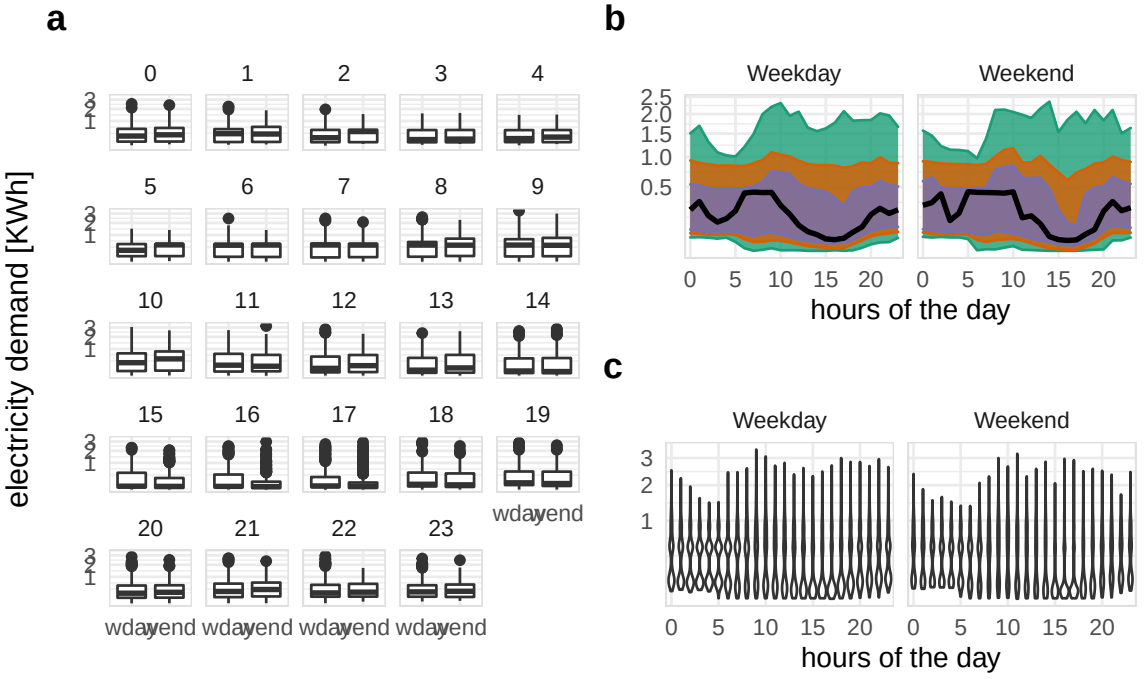



**Figure 2.5:** *Energy consumption of a single customer shown with different distribution displays, and granularity arrangements: hour of the day; and weekday/weekend.* ***a*** *The side-by-side boxplots make the comparison between day types easier, and suggest that there is generally lower energy use 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.*

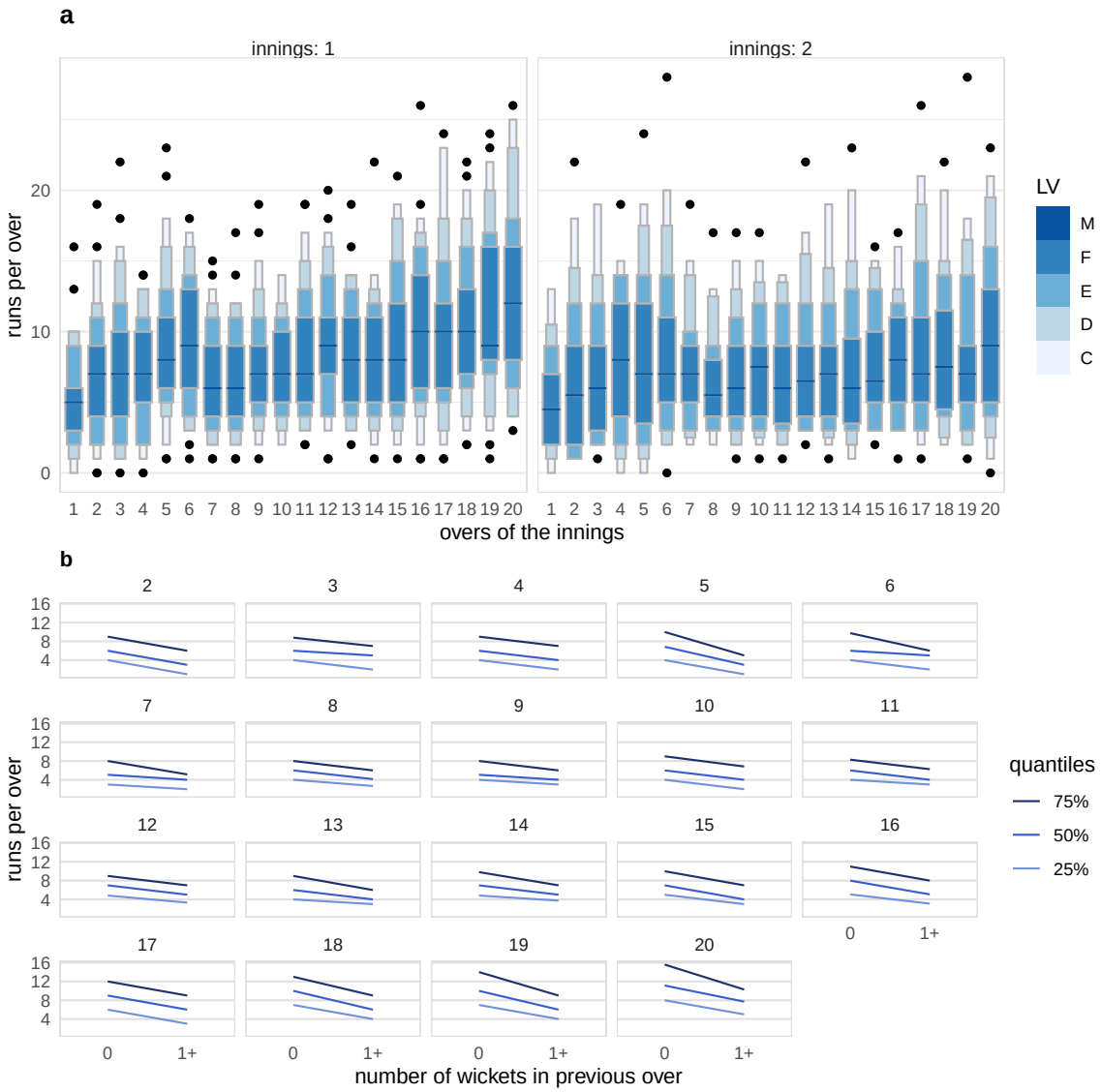



**Figure 2.6:** *Examining distribution of runs per by innings, over of the innings and number of wickets in previous innings. Plot **a** displays distribution using letter value plots. A gradual upward trend in runs per over can be seen in innings 1, which is not present in innings 2. Plot **b** shows 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 can 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 (**Bettini1998-ed**) 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 different week types in a semester, orientation, break, exam or regular classes in an academic calendar. Using granularities to explore patterns in univariate time series can be considered to be examining the distribution of the measured variable across different categories of the cyclic granularities.

Figure **??** 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, as was used in **wang2020calendar**. The categorical variable (granularity) is mapped to the x-axis, and the distribution of response variable is displayed using both side-by-side jittered dotplots and boxplots. From plot (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 perceived from sizes of the boxes. From plot (b) the variability in energy usage is higher in Jan and Feb, possibly 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 might imply that this household does not use as much energy for heating as it does for air conditioning. A lot of households in Victoria use gas heating and hence the heater use might not be reflected in the electricity data. Many, many different displays could be constructed using different granularities, day-of-week, day-of-month, weekday/weekend, etc. However, only a few might be interesting, that is, reveal important patterns in energy usage. Determining which displays which 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 tends to provide more detailed information on its structure. For example, Figure **??**(a) slice down further by showing

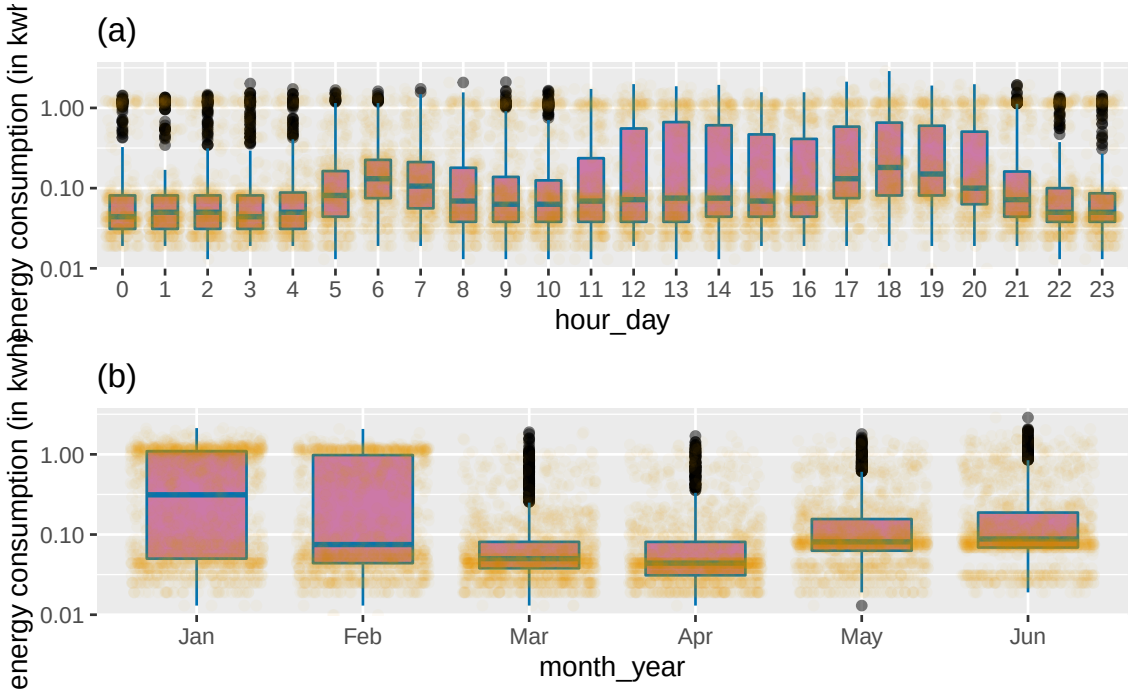



**Figure 3.1:** *A cyclic granularity can be considered to be a categorical variable, and used to break the data into subsets. Here, side-by-side boxplots overlaid on jittered dotplots explore the distribution of of energy use by a household for two different cyclic granularities: (a) hour-of-day and (b) and 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 (Jan, Feb) is probably due air conditioner use on just some days .*

the usage distribution across hour-of-day conditional on month-of-year across two households (id 2 and 4). 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, pretty close, and are not characterized by a morning and evening peak. The household in Figure **??**(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 of these four displays in Figures **??** and **??** 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 **??**. The dimension of the problem, however,

increases when considering more than one cyclic granularity. When considering displays of the form in Figure **??**, there are ${}^{N_C}P_2$ possible pairwise plots exhaustively, with one of the two cyclic granularities acting as the conditioning variable. This is huge and overwhelming for human consumption even for moderately large $N_C$. It could be immensely useful to make the transition from all potential displays to the ones that are informative across atleast one cyclic granularity.

This problem is similar to Scagnostics (Scatterplot Diagnostics) by **tukey1988computer**, which is used to identify meaningful patterns in large collections of scatterplots. Given a set of $v$ variables, there are $v(v-1)/2$ 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) could easily run out of pixels when presenting high-dimensional data. **Dang2014-tw** and **wilkinson2005graph** provide potential solutions to this, where 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 ${}^{N_C}P_2$ 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 (**Gupta2021-hd**) that narrows down the search from ${}^{N_C}P_2$ 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 could be enormous for exhaustive exploration. Hence, there is a need to reduce the search even further by including only those harmonies which are informative enough. **inference** and **Majumder2013-hb** present 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.

The article is organized as follows. Section **??** introduces a distance measure for detecting distributional difference in temporal granularities, which enables identification of patterns in the time series data; Section **??** devises a selection criterion by choosing a threshold, which results in detection of only significantly interesting patterns. Section **??** provides some simulation study on the proposed methodology. Section **??** presents an application to residential smart meter data in Melbourne to

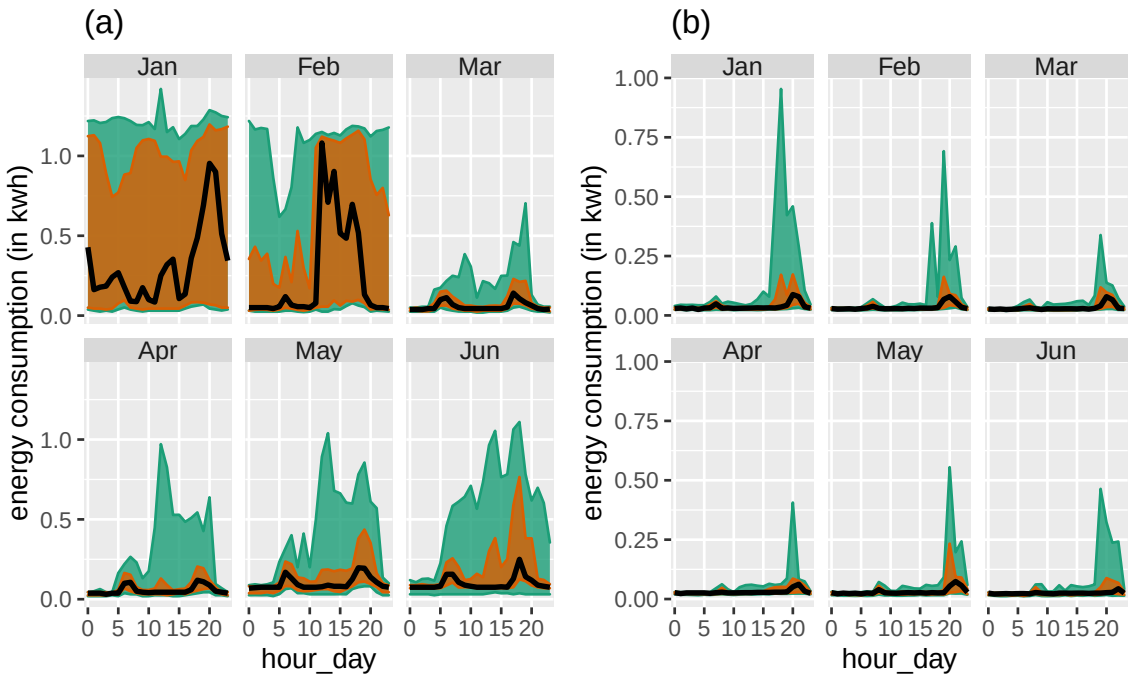



**Figure 3.2:** *Distribution of energy consumption displayed through area quantile plots across two cyclic granularities month-of-year and hour-of-day and two households. The black line is the median, whereas the orange band covers the 25th to 75th percentile and the green band covers the 10th to 90th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

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 explained in Figure **??**. Each of these figures describes a panel with 2 x-axis categories and 3 facet levels, but with different designs. Figure **??**a has all categories drawn from N(0, 1) distribution for each facet. It is not an interesting display particularly, as distributions do not vary across x-axis or facet

categories. Figure **??**b has x categories drawn from the same distribution, but across facets the distributions are 3 standard deviations apart. Figure **??**c exhibits an exact opposite situation where distribution between the x-axis categories are 3 standard deviations apart, but they do not change across facets. Figure **??**d takes a step further by varying the distribution across both facet and x-axis categories by 3 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 placing (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, display (b) is considered less informative as compared to display (c) in emphasizing the distributional differences. Considering one cyclic granularity, we would have only two design choices similar to (a) and (c), corresponding to no difference and significant differences between categories of that cyclic granularity only. The proposed measure *wpd* is constructed in a way so that it could 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 Notations

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 $A = \{a_j : j = 1,2,\ldots,J\}$ and $B = \{b_k : k = 1,2,\ldots,K\}$ with $A$ placed across 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 $J$ x-axis levels and $K$ facet levels is referred to as a $(J,K)$ panel. For example, a $(2,3)$ panel will have cyclic granularities with 2 x-axis levels and 3 facet levels. Let the four elementary designs as described in Figure **??** be $D_{null}$ (referred to as "null distribution") where there is no difference in distribution of $v$ for $A$ or $B$, $D_{var_f}$ denotes the set of designs where there is difference in distribution of $v$ for $B$ and not for $A$. Similarly, $D_{var_x}$ denotes the set of designs where difference is observed only across A. Finally, $D_{var_{all}}$ denotes those designs for which difference is observed across both $A$ and $B$. $m = 1$ is a special case of $m = 2$ with $J = 1$.

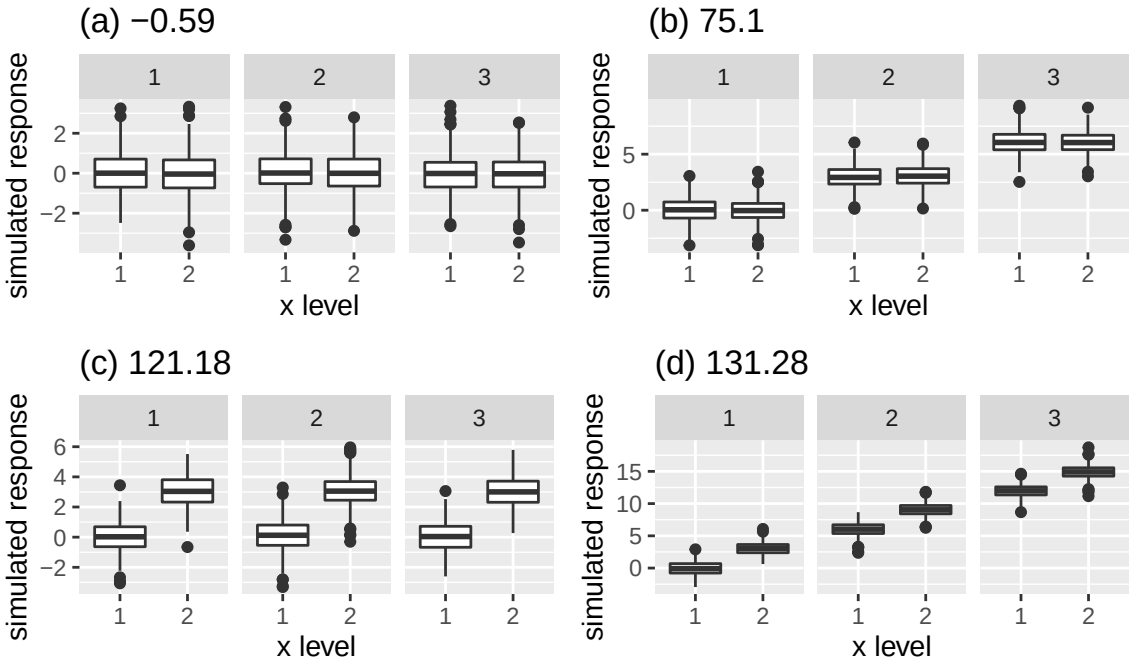



**Figure 3.3:** *An example illustrating the principle of the proposed distance measure, displaying the distribution of a normally distributed variable in four panels each with* 2 *x-axis categories and* 3 *facet levels, but with different designs. Display (a) is not interesting as the distribution of the variable does not depend on x or facet categories. Display (b) and (c) are more interesting than (a) since there is a change in distribution either across facets (b) or x-axis (c). Display (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 3.2.*

**Table 3.1:** *Nomenclature table*

| variable | description |
|---|---|
| $N_C$ | number of cyclic granularities |
| $H_{N_C}$ | set of harmonies |
| nx | number of x-axis categories |
| nfacet | number of facet categories |
| $\lambda$ | tuning parameter |
| $\omega$ | increment (mean or sd) |
| $wpd$ | raw weighted pairwise distance |
| $wpd_{norm}$ | normalized weighted pairwise distance |
| $nperm$ | number of permutations for threshold/normalization |

| variable | description |
|---|---|
| $nsim$ | number of simulations |
| $wpd_{threshold}$ | threshold for significance |
| $D_{null}$ | null design with no distributional difference across categories |
| $D_{var_f}$ | design with distributional difference only across facets categories |
| $D_{var_x}$ | design with distributional difference only across x-axis categories |
| $D_{var_{all}}$ | design with distributional difference across both facet and x-axis |

### 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 weightage for different group 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) (**Krzysztofowicz1997-bv**), so that the quantiles of the transformed data follows a standard normal distribution. This sort of transformation is common in the fields of geo-statistics to make most asymmetrical distributed real world measured variables more treatable and normal-like (**Bogner2012-az**).The empirical NQT involves the following steps:

1. The sample of measured variable $v$ is sorted from the smallest to the largest observation $v_{(1)},\ldots,v_{(i)},..,v_{(n)}$.

2. The cumulative probabilities $p_{(1)},\ldots,p_{(i)},..,p_{(n)}$ are estimated using a plotting position like $i/(n+1)$ such that $p_{(i)} = P(v \leq v_{(i)})$.

3. Each observation $v_{(i)}$ of $v$ is transformed into observation $v^*(i) = Q^{-1}(p(i))$ of the standard normal variate $v^*$, with $Q$ denoting the standard normal distribution and $Q^{-1}$ its inverse.

### Characterising 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 $v_{jk}$'s are chosen as a way to characterize distributions for the category $(a_j, b_k), \forall j \in \{1,2,\ldots,J\}, k \in \{1,2,\ldots,K\}\}$. The quantile of a distribution with probability $p$ is defined as $Q(p) = F^{-1}(p) = inf\{x : F(x) > p\}$, $0 < p < 1$ where $F(x)$ is the distribution function. There are two broad approaches to quantile estimation, viz, parametric and non-parametric. Sample quantiles (**Hyndman1996-ty**) are used for estimating population quantiles in a non-parametric setup, which is desirable because of less rigid assumptions made about the nature of the underlying distribution of the data. The default quantile chosen in this paper is percentiles computed for $p = 0.01, 0.02, \ldots, 0.99$, where for example, the $99^{th}$ percentile would be the value corresponding to $p = 0.99$ and hence 99% of the observations would lie below that.

### Distance between distributions

One of the most common ways to measure divergence between distributions is the Kullback-Leibler (KL) divergence (**Kullback1951-jy**). 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, however, is not symmetric and hence can not be considered as a "distance" measure. The Jensen-Shannon divergence (**Menendez1997-in**) 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 is 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 $D(q_1||q_2) := \int_{-\infty}^{\infty} q_1(x) f(\frac{q_1(x)}{q_2(x)})$ is the KL divergence between distributions $q_1$ and $q_2$. Other common measures of distance between distributions are Hellinger distance, total variation distance and Fisher information metric.

**Within-facet and between-facet distances**

Pairwise distances could be within-facets or between-facets for $m \geq 2$. Figure **??** 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 **??**. 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 $(a_j b_k, a_j b_{k'})$. Number of between-facet distances would be ${}^{K}C_2 * J$ and number of within-facet distances are $K * (J-1)$ (ordered) and ${}^{J}C_2 * K$ (un-ordered).

**Tuning parameter**

A tuning parameter specifying the weightage given to the within-facet or between-facet categories can help to balance weightage between designs like **??**(b) and (c). The tuning parameters should be chosen such that $\sum_{i=1}^{m} \lambda_i = 1$. When $m = 2$, following the principle of Gestalt theory, $\lambda = \frac{2}{3} = 0.67$ is chosen to put a relative weightage of $2:1$ for within-facet and between-facet distances. No human experiment is conducted to justify this ratio, however, typically a tuning parameter $\lambda > 0.5$ would tend to upweigh the within-facet distances and that with $< 0.5$ would upweigh the between-facet distances (refer to the Supplementary paper for more details). For $m = 1$, there are no conditioning variables or groups, and hence $\lambda = 1$.

**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 $(a_j b_k, a_{j'} b_{k'}) : \{a_j : j = 1, 2, \ldots, J\}, B = \{b_k : k = 1, 2, \ldots, K\}$. The pairwise distances between pairs $(a_j b_k, a_{j'} b_{k'})$ denoted by $d_{(jk,j'k')} = JSD(q_{jk}, q_{j'k'})$ is 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

$$wpd = max_{j,j',k,k'}(d*_{(jk),(j'k')}) \forall j, j' \in \{1,2,\ldots,J\}, k, k' \in \{1,2,\ldots,K\}$$

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 obervations 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.

**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 $J$ or $K$ is high. That is, under designs like $D_{null}$, their distribution should not differ for a different number of categories. Only then the distance measure could be compared across panels with different levels. This calls for an adjusted measure, which normalizes for the different number of comparisons. We denote it by $wpd$. 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 $wpd_{raw}$ is computed on many ($nperm$) permuted data sets and stored in $wpd_{perm-data}$. Then $wpd_{perm}$ is computed as follows:

$$wpd_{perm} = \frac{(wpd_{raw} - \bar{wpd}_{perm-data})}{sd(wpd_{perm-data})}$$

where $\bar{wpd}_{perm-data}$ and $sd(wpd_{perm-data})$ are the mean and standard deviation of $wpd_{perm-data}$ respectively. Standardizing $wpd$ in the permutation approach ensures that the distribution of $wpd_{perm}$ under $D_{null}$ has the same $mean = 0$ and $\sigma^2_{perm} = 1$ across all comparisons. While this works successfully to make the location and scale similar across different $nx$ and $nfacet$, it is computationally heavy and time consuming, and hence less user friendly when being actually used in practice. 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 number of comparisons as the explanatory variable. Since, $wpd_{raw}$ is a Jensen-Shannon distance, it follows a Chi-square distribution (**Menendez1997-in**), which is a special case of Gamma distribution. Furthermore, the mean response is bounded, since any JSD is bounded by 1 given that base 2 logarithm is used (**Lin1991-pt**). Hence, by **Faraway2016-uk**, an inverse link is used for the model, which is of the form $y = a + b * log(z) + e$, where $y = wpd_{raw}$, $z = (nx * nfacet)$ is the number of groups and $e$ are idiosyncratic errors. Let $E(y) = \mu$ and $a + b * log(z) = g(\mu)$ where $g$ is the link function. Then $g(\mu) = 1/\mu$ 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 chosen as the residuals from this model and is defined as:

$$wpd_{glm} = wpd_{raw} - 1/(\hat{a} + \hat{b} * log(nx * nfacet))$$

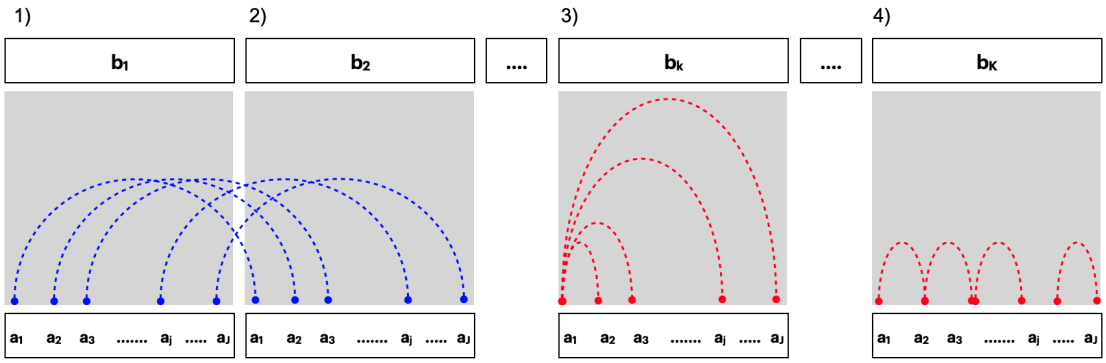



**Figure 3.4:** *Within and between-facet distances shown for two cyclic granularities A and B, where A is mapped to x-axis and B is mapped to facets. The dotted lines represent the distances between different categories. Panel 1) and 2) show the between-facet distances. Panel 3) and 4) are used to illustrate within-facet distances when categories are un-ordered or ordered respectively. 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* $(a_1, b_3)$.

The distribution of $wpd_{glm}$ under $D_{null}$ will have $mean = 0$, since it is the residuals from the model, and a constant variance $\sigma^2_{glm}$, which might not equal 1.

*Combination approach*

The simulation results (in Section **??**) show that the distribution of $wpd_{glm}$ under null is similar for high $nx$ and $nfacet$ (levels higher than 5) and not so much for lower $nx$ and $nfacet$. Hence, a combination approach is proposed which chooses permutation approach for categories with smaller levels and modeling approach for categories with higher levels. This ensures that the computational load of the permutation approach is alleviated while maintaining 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 $wpd_{glm-scaled} = wpd_{glm} * \sigma^2_{perm} / \sigma^2_{glm}$ 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:

$$wpd = \begin{cases} wpd_{perm}, & \text{if } J,K <= 5 \\ wpd_{glm-scaled} & \text{otherwise} \end{cases} \tag{3.2}$$

## 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 complete randomness 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 basis 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 (**edgington2007randomization**). 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 $wpd_{threshold}$, 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 $\{(A,B), A = \{a_j : j = 1,2,\ldots,J\}, B = \{b_k : k = 1,2,\ldots,K\}\}, \forall (A,B) \in H_{N_C}$.
- **Output:** Harmony pairs $(A,B)$ for which *wpd* is significant.

1. Fix harmony pair $(A,B)$.
2. Given the measured variable; $\{v_t : t = 0,1,2,\ldots,T-1\}$, *wpd* is computed and is represented by $wpd_{obs}^{A,B}$.
3. From the original sequence a random permutation is obtained: $\{v_t^1 : t = 0,1,2,\ldots,T-1\}$.

4. $wpd$ is computed for the permuted sequence of the data and is represented by $wpd_1^{A,B}$.

5. Steps (3) and (4) are repeated a large number of times $M$ (e.g. $M = 200$).

6. For each permutation, one $wpd_i^{A,B}$ is obtained. Define $wpd_{sample} = \{wpd_1^{A,B} wpd_2^{A,B}, \ldots, wpd_M^{A,B}\}$.

7. Repeat Steps (1-6) for all harmony pairs $(A,B) \in H_{N_C}$ and stored $wpd_{sample}^{all}$.

8. $99^{th}$ percentiles of $wpd_{sample}^{all}$ is computed and stored in $wpd_{threshold99}$

9. If $wpd_{obs}^{A,B} > wpd_{threshold99}$, harmony pair $(A,B)$ is selected, otherwise rejected.

Similarly, a harmony pair $(A,B)$ could be selected if $wpd_{obs}^{A,B} > wpd_{threshold95}$ and $wpd_{obs}^{A,B} > wpd_{threshold90}$, where $wpd_{threshold95}$ and $wpd_{threshold90}$ denote the $95^{th}$ and $90^{th}$ percentile of $wpd_{sample}^{all}$ respectively. A harmony selected using $99^{th}$, $95^{th}$ and $90^{th}$ threshold 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 **??** presents the results of $wpd$ from the illustrative designs in Section **??**. 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 $D_{var_{all}}$ (d). Moreover, note the relative differences in $wpd$ values between (b) and (c). The value of the tuning parameter $\lambda$ is set to 0.67, which has resulted in giving more emphasis to differences in x-axis categories. Again consider **??**(a) and **??**(b) with a $wpd$ value of 20.5 and 145 respectively. This is because there is more gradual increase across hours of the day than months of the year. If order of the categories is not considered, they result in a $wpd$ value of 97.8 and 161 respectively, which follows from the fact that if we consider difference between any hours of the day, the magnitude will be much larger than if we consider difference between consecutive categories.

Similarly, **??**(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 the first household.

## 3.4 Simulations

### 3.4.1 Behavior of raw and adjusted distance measures

*Simulation design*

$m = 1$

Observations are generated from a N(0,1) distribution for a wide range of levels from very low to moderately high. $nx = \{2,3,5,7,9,14,17,20,24,31,42,50\}$ is considered. $ntimes = 500$ observations are drawn for each combination of the categories, that is, for a panel with $nx = 3$, 500 observations are simulated for each of the categories. This design corresponds to $D_{null}$ as each combination of categories in a panel are drawn from the same distribution. Furthermore, the data is simulated for each of the categories $nsim = 200$ times, so that the distribution of $wpd$ under $D_{null}$ could be observed. The values of $wpd$ is obtained for each of the panels. $wpd_{l,s}$ denotes the value of $wpd$ obtained for the $l^{th}$ panel and $s^{th}$ simulation.

$m = 2$

Similarly, observations are generated from a N(0,1) distribution for each combination of $nx$ and $nfacet$ from the following sets: $nx = nfacet = \{2,3,5,7,14,20,31,50\}$. That is, data is being generated for each of the panels $(2,2),(2,3),(2,5)\ldots,(50,31),(50,50)$. For each of the 64 panels, $ntimes = 500$ observations are drawn for each combination of the categories. That is, if we consider a $(2,2)$ panel, 500 observations are generated for each of the possible subsets, namely, $\{(1,1),(1,2),(2,1),(2,2)\}$.

*Results*

Figure **??** shows that both the location and scale of the distributions change across panels. This is not desirable under $D_{null}$ as it would mean comparisons of $wpd$ values is not appropriate across different $nx$ and $nfacet$. Table **??** gives the summary of the generalized linear model to capture the relationship between $wpd_{raw}$ and number of comparisons. For example, the model

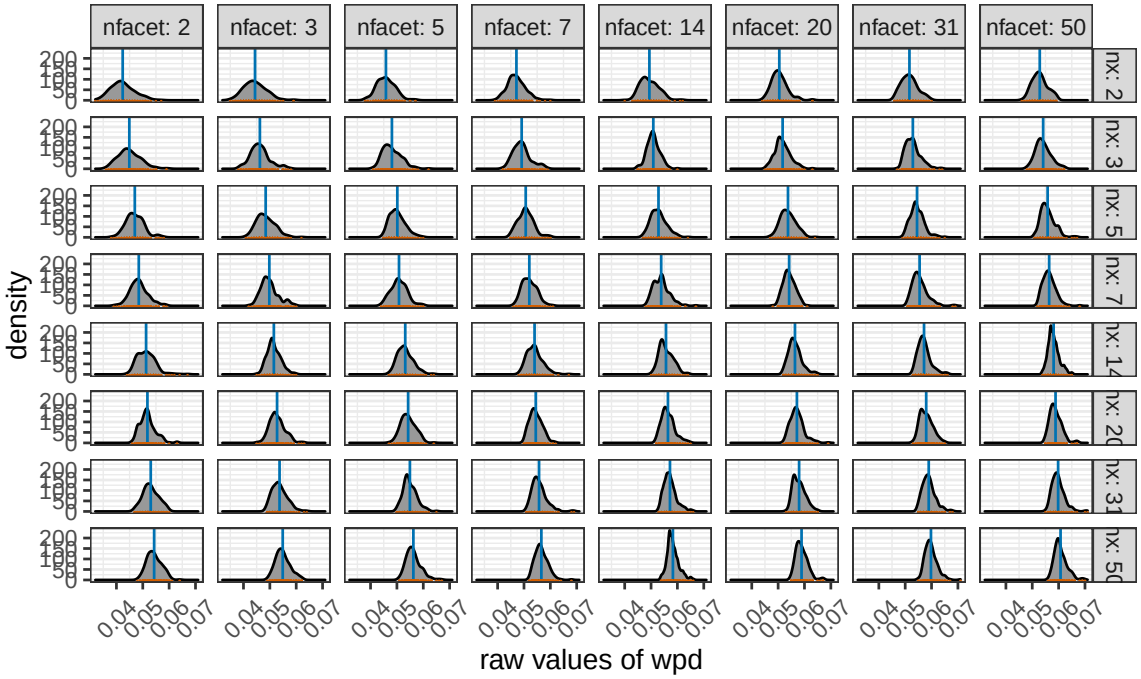



**Figure 3.5:** *Distribution of $wpd_{raw}$ is plotted across different nx and nfacet categories under $D_{null}$ through density and rug plots. Both location (blue line) and scale (orange marks) of the distribution shifts for different panels. This is not desirable since under null design, the distribution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

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

| m | term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|---|
| 1 | (Intercept) | 26.09 | 0.54 | 48.33 | 0 |
| 1 | log(‘nx * nfacet‘) | -1.87 | 0.19 | -9.89 | 0 |
| 2 | (Intercept) | 23.40 | 0.22 | 104.14 | 0 |
| 2 | log(‘nx * nfacet‘) | -0.96 | 0.04 | -21.75 | 0 |

considered for $m = 2$ is $wpd_{l,s} = 23.4 - 0.96 * log(nx * nfacet)) + e$. The intercepts and slopes are similar independent of the starting distributions (see supplementary paper for details) and hence the coefficients are shown for the case when observations are drawn from a N(0, 1) distribution. Figure **??** shows the distribution of $wpd_{perm}$ and $wpd_{glm-scaled}$ in the same scale to show that a combination approach could be used for higher values of levels to alleviate the computational time of permutation approach.

### 3.4.2 Choosing threshold

*Simulation design*

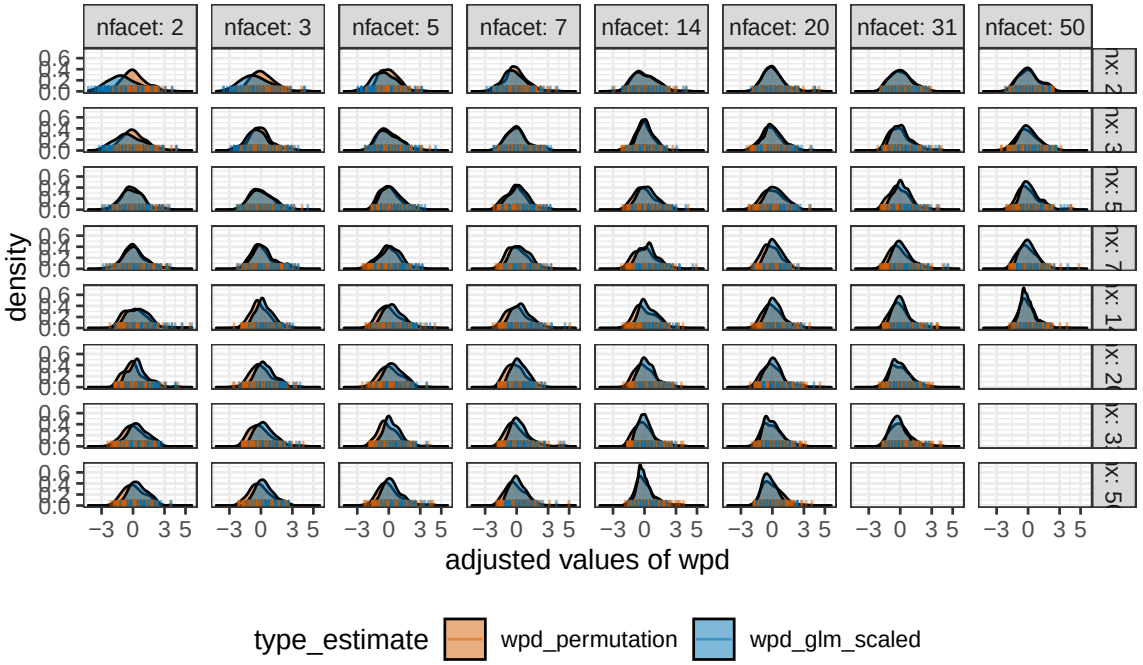



**Figure 3.6:** *The distribution of $wpd_{perm}$ and $wpd_{glm-scaled}$ are overlaid to compare the location and scale across different nx and $nfacet$. $wpd_{norm}$ takes the value of $wpd_{perm}$ for lower levels, and $wpd_{glm-scaled}$ for higher levels to to alleviate the problem of computational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

Observations are generated from a N(0,1) distribution for each combination of $nx$ and $nfacet$ from the following sets: $nx = \{3,7,14\}$ and $nfacet = \{2,9,10\}$. This would result in 9 panels, viz, $(3,2),(3,9),(3,10),\ldots,(14,9),(14,10)$. Few experiments were conducted. In the first scenario, data for all panels are simulated using the null design $D_{null}$. In other scenarios, data simulated from the panel $(14,2)$ and $(3,10)$ are considered under $D_{vary_{all}}$. Moreover, $\omega = \{0.5,2,5\}$ are considered to examine if the proposed test is able to capture subtle differences and non-subtle differences when we shift from the null design.

*Results*

For the first scenario, size of the test is obtained as 0.1 with $wpd_{threshold99}$ as the threshold. This implies that the proportion of times a panel is rejected when it is under $D_{null}$ is 0.1. The level of significance for each test is 1% (as a result of choosing $99^{th}$ percentile as the threshold) and we have 9 multiple tests. Hence, it is reasonable that the level of significance of the composite tests would be larger than the individual tests. We also compute the proportion of times a panel is rejected when it actually belongs to a non-null design. These proportions constitute to be the estimated size

and power of the test. Power can depend on many things like sample size, number of designs that deviate from the null and extent to which from the null. It is found that as we increase from low to high changes from the null distribution, the power increased. The results and graphics are included in details in the Supplementary paper.

### 3.4.3 Environment

Simulation studies were carried out to study the behavior of *wpd*, build the normalization method as well as compare and evaluate different normalization approaches. R version 4.0.1 (2020-06-06) is used with the platform: x86_64-apple-darwin17.0 (64-bit) running under: macOS Mojave 10.14.6 and MonaRCH, which is a next-generation High Power Computing (HPC) Cluster, addressing the needs of the Monash HPC community.

## 3.5 Application to residential smart meter dataset

The smart meter data set for eight households in Melbourne has been utilized to see the use of *wpd* proposed in the paper. The data has been cleaned to be a `tsibble` (**wang2020tsibble**) containing half-hourly electricity consumption from Jul-2019 to Dec-2019 for each of the households, which is procured by them by downloading their data from the energy supplier/retailer. No behavioral pattern is likely to be discerned from the line graph of energy usage over the entire period, since the plot will have too many measurements squeezed in a linear representation. When we zoom into the linear representation of this series in Figure **??** (b) for September, 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 a daily and weekly periodic pattern. However, it is not apparent from this view if all of these households have those patterns and in case they have if they are significant enough. Also, it is not clear if any other periodic patterns are present in any household which might have been hidden with this view. We start the analysis by choosing few harmonies, ranking them for each of these households, compare households to get more insights into what these rankings imply. Furthermore, 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 $i^{th}$ household for time period $t$. The series $v_{i,t}$ 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 Gregorian calendar. Considering 4 linear granularities hour, day, week, month, the number of cyclic granularities is $N_C = (4 * 3/2) = 6$. We obtain cyclic granularities namely "hour_day", "hour_week", "hour_month", "day_ week", "day_month" and "week_month", read as "hour of the day", etc. Further, we add cyclic granularity day-type ("wknd wday") to capture weekend and weekday behavior. Thus, 7 cyclic granularities are considered to be of interest. The set consisting of pairs of cyclic granularities ($C_{N_C}$) will have $7_{P_2} = 42$ elements which could be analyzed for detecting possible periodicities. The set of possible harmonies $H_{N_C}$ from $C_{N_C}$ are chosen by removing clashes using procedures described in (**Gupta2021-hd**). Table **??** shows 14 harmony pairs that belong to $H_{N_C}$.

*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 households $i \in \{1,2,\ldots,8\}$. 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 **??** shows the rank of the harmonies for different households. The rankings are different for different households, which is a reflection of their varied behaviors. Most importantly, there are at most 3 harmonies that are significant for any household. This is a huge reduction in the number of potential harmonies to explore closely, starting from 42.

*Detecting patterns not apparent from linear display*

Figure **??** helps to compare households through the heatmap (a) across harmony pairs with the cyclic granularity mapped to x-axis and facet being plotted on the x-axis and y-axis of the heatmap. Here *dom*, *dow*, *wdwnd* are abbreviations for day-of-month, day-of-week, weekday/weekend and so on. The colors represent the value of $wpd$. Darker cells correspond to more significant harmony pairs. Also, the ones with $*$ corresponds to the ones above $wpd_{threshold95}$. Few observations that emphasizes patterns not discernible through (b) includes - (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 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 $wpd_{threshold95}$. Note that the *wpd* values are computed over the entire range, but the linear display in (b) is zoomed only for September, with the major and minor x-axis corresponding to weeks and days respectively.

**Table 3.3:** *Ranking of harmonies for the eight households with significance marked for different thresholds. Rankings are different and at most three harmonies are significant for any household. The number of harmonies to explore are 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 | wdwnd | 1 *** | 2 * | 1 ** | 2 ** | 3 | 1 ** | 3 | 3 * |
| dom | hod | 2 *** | 4 | 3 ** | 3 ** | 4 | 3 * | 4 | 6 |
| wdwnd | hod | 3 ** | 10 | 7 | 7 | 6 | 8 | 8 | 10 |
| hod | wom | 4 | 9 | 6 | 5 | 5 | 5 | 5 | 5 |
| wom | wdwnd | 5 | 14 | 14 | 10 | 12 | 9 | 12 | 13 |
| hod | dow | 6 | 1 *** | 2 ** | 1 *** | 1 * | 2 ** | 2 ** | 1 ** |
| wdwnd | 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 | wdwnd | 13 | 13 | 11 | 12 | 14 | 14 | 14 | 14 |
| wdwnd | dom | 14 | 11 | 12 | 14 | 13 | 13 | 13 | 11 |

*Comparing households and validating rank of harmonies*

According to Figure **??**(c), for the harmony pair (dow-hod), household id 7 has the greatest value of *wpd*, while id 1 has the least. Also, from table **??** it could be seen that the harmony pair $(dow, hod)$ is important for id 7, however it has been labeled as an inconsequential 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 **??** demonstrates that for id 7, the median (black) and quartile

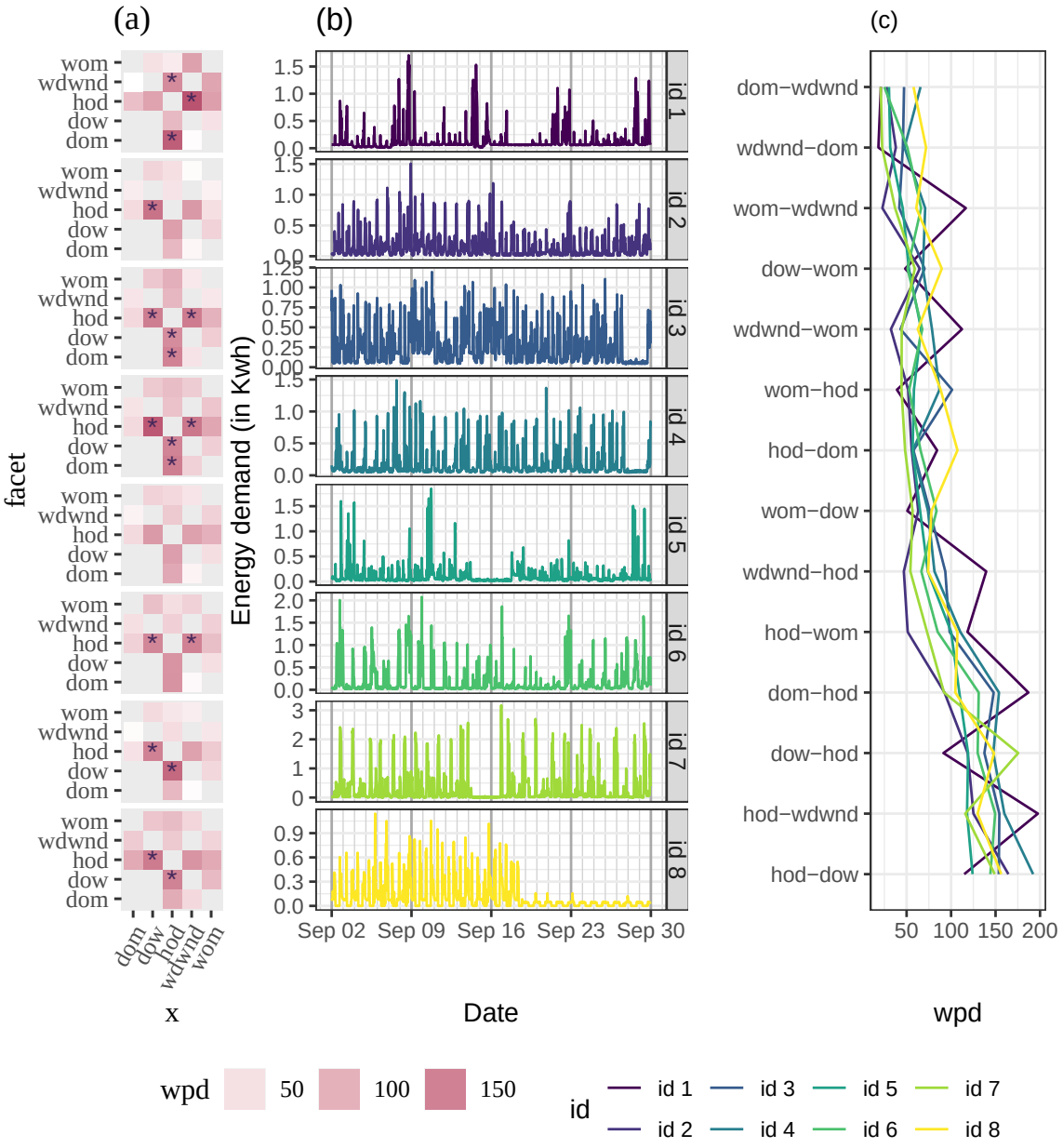



**Figure 3.7:** *An ensemble plot with a heatmap (a), line plot (b), parallel coordinate plot (c) to demonstrate energy behavior of the households in different ways. (b) is the zoomed-in raw demand series for September to highlight the repetitive patterns of energy demand. (a) shows wpd values across harmonies with the x variable of harmony placed across x-axis and facet variable placed across y-axis. The darker the colour in (a), the higher the harmony is ranked. Visualizing harmonies through (a) allows to view the significant cyclic granularities along the x-axis, facet or both for each household. For eg, ids 7 and 8 have significant patterns across (hod, dow) and (dow, hod), which was not apparent through (a). (c) is useful for comparing households across harmonies, for eg, for the harmony (dow-hod), ids 1 and 7 have the least and highest wpd respectively.*

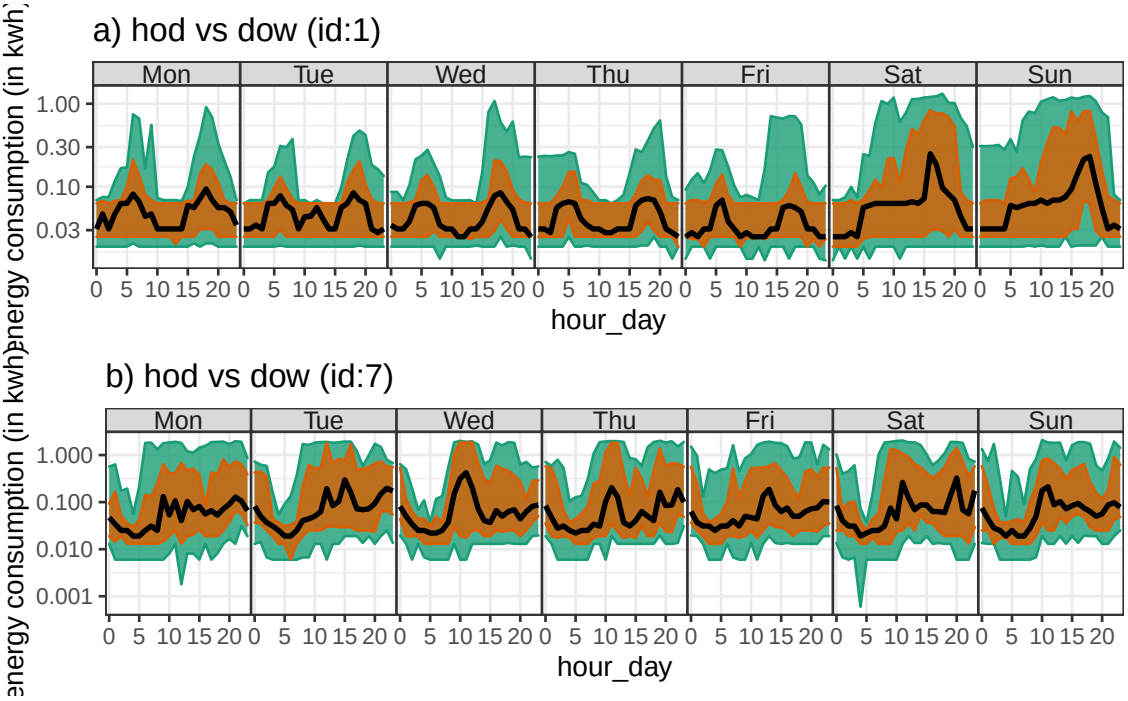



**Figure 3.8:** *Comparing distribution of energy demand shown for household id 1 (a) and 7 (b) across hod in x-axis and dow in facets through quantile area plots. The value of wpd in Table 3 suggests that the harmony pair (dow, hod) is significant for household id 7, but not for 1. This implies that 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 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.*

deviation (orange) of energy consumption fluctuates throughout for most hours of the day and day of the week, while for id 1, daily patterns are consistent within weekdays and weekends. As a result, for id 1, it is more appropriate to examine the distributional difference solely across (dow, wdwnd), which has been rated higher in Table **??**.

## 3.6 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 usual granularities like 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 of 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.

## Acknowledgments

The Australian 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-language**), `knitr` (**knitr2015**) and `rmarkdown` (**rmarkdown2018**). Graphics are produced with **Wickham2009pk**.

## 3.7 Supplementary Materials

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

**Simulation results and scripts:** All simulation table, graphics and and R code to reproduce it (paper-supplementary.pdf, paper-supplementary.Rmd)

**R-package:** The open-source R (**R-language**) 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

With more and more time series data being collected at much finer temporal resolution, for a longer length of time, and for a larger number of individuals/entities, time series clustering research is getting a lot of traction. Long, noisy, patchy, uneven, and asynchronous time series are common in many fields, limiting similarity searches or lowering method efficiency when clustering is based on a distance metric. In this work, we suggest two approaches for obtaining similarity between time series based on probability distributions over cyclic temporal granularities for distance-based clustering approaches. Cyclic granularities like hour-of-the-day, work-day/weekend, month-of-the-year and so on, are useful for finding repeated patterns in the data. Looking at probability distributions across cyclic granularities serves two purposes: (a) "Probability distributions" characterise the inherent temporal data structure of these large unequal-length time series and are robust to missing or noisy data. (b) Using probability distributions over "cyclic granularities" ensures small pockets of similar "repeated" behaviors. Our method is capable of producing useful clusters, as demonstrated by testing on validation data designs and a sample of residential smart meter consumers.

## 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 to time series clustering (**liao2005clustering**) are algorithms that operate directly with distances or raw data points in the time or frequency domain (distance-based), with features derived from raw data (feature-based), or indirectly with models constructed from raw data (model-based). The efficacy of distance-based techniques is highly dependent on the distance measure utilized. Defining an appropriate distance measure for the raw time series may be a difficult task since it must take into account noise, variable lengths of time series, asynchronous time series, different scales, and missing data. Commonly used Distance-based similarity measures as suggested by a decade review of time series clustering approaches (**Aghabozorgi2015-ct**) are Euclidean, Pearson's correlation coefficient and related distances, Dynamic Time Warping, Autocorrelation, Short time series distance, Piecewise regularization, cross-correlation between time series, or a symmetric version of the Kullback–Liebler distances (**liao2007clustering**) but on a vector time series data. Among these alternatives, Euclidean distances have high performance but need the same length of data over the same period, resulting in information loss regardless of whether it is on raw data or a smaller collection of features. DTW works well with time series of different lengths (**corradini2001dynamic**), 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.

We consider the problem of clustering a large number of univariate time series of continuous values which are available at fine temporal scales, being motivated by the residential smart meter data. 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 missing observations and characterized by noisy and patchy behavior that can quickly become overwhelming and hard to interpret, requiring summarizing the large number of customers into pockets of similar energy behavior. Choosing probability distributions seem to be a natural way to analyze these types

of data sets since 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 significant cyclic granularities. Cyclic temporal granularities (**Gupta2021-hakear**), 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 resulting clusters are expected to group customers that have similar repetitive behaviors across each of the interesting cyclic granularities. Below are some of the benefits of our method:

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

*Background and motivation*

Large spatio-temporal data sets, both from open and administrative sources, offer up a world of possibilities for research. One such data sets for Australia is the Smart Grid, Smart City (SGSC) project (2010–2014) available through Department of the Environment and Energy The project provides half-hourly data of over 13,000 household electricity smart meters distributed

unevenly from October 2011 to March 2014. Larger data sets include greater uncertainty about customer behavior due to growing variety of customers. Households vary in size, location, and amenities such as solar panels, central heating, and air conditioning. The behavioral patterns differ amongst customers due to many temporal dependencies. Some households use a dryer, while others dry their clothes on a line. Their weekly 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. Day-off energy use varies depending on whether customers stay home or go outside. 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 a fascinating one. This challenge is worsened when all we know about our consumers is their energy use history (**Ushakova2020-rl**). To safeguard the customers' privacy, it is probable that such information is not accessible. Also, energy suppliers may not always update client information, such as property features, in a timely manner. Thus, there is a growing need to have research that examines how much energy usage heterogeneity can be found in smart meter data and what are some of the most common power consumption patterns, rather than explaining why consumption differs.

*Related work*

A multitude of papers have emerged around smart meter time series clustering for deepening our knowledge of consumption patterns. **Tureczek2017-pb** conducted a systematic study of over 2100 peer-reviewed papers on smart meter data analytics. None of the 34 articles chosen for their emphasis use Australian smart meter data. The most often used algorithm is K-Means. Using K-Means without considering time series structure or correlation results in inefficient clusters. Principal Component Analysis (PCA) or Self-Organizing Maps (SOM) eliminate correlation patterns and decrease feature space, but lose interpretability. To reduce dimensionality, several studies use principal component analysis or factor analysis to pre-process smart-meter data before clustering (**Ndiaye2011-pf**). Other algorithms utilized in the literature include k-means variants, hierarchical approaches, and greedy k-medoids. Time series data, such as smart meter data, are not well-suited to any of the techniques mentioned in **Tureczek2017-pb**. Only one study (**ozawa2016determining**) identified time series characteristics using Fourier transformation, which

converts data from time to frequency and then uses K-Means to cluster by greatest frequency. **Motlagh2019-yj** suggests that the time feature extraction is limited by the type of noisy, patchy, and unequal time-series common in residential and addresses model-based clustering by transforming the series into other other objects such as structure or set of parameters which can be more easily characterized and clustered. (**chicco2010renyi**) addresses information theory-based clustering such as Shannon or Renyi entropy and its variations. **Melnykov2013-sp** discusses how outliers, noisy observations and scattered observations can complicate estimating mixture model parameters and hence the partitions. To our knowledge, none of the methods focus on exploring heterogeneity in repetitive behaviors based on the dynamics of multiple temporal dependencies using probability distributions.

The remainder of the paper is organized as follows: Section **??** provides the clustering methodology. Section **??** shows data designs to validate our methods. Section **??** 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 **??**.

## 4.2 Clustering methodology

The foundation of our method is unsupervised clustering algorithms based exclusively on the time-series data. The proposed methodology aims to leverage the intrinsic data structure hidden within cyclic temporal granularities. The existing work on clustering probability distributions assumes we have an iid sample $f_1(v),\ldots,f_n(v)$, 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 this work, instead of considering the probability distributions of the linear time series, we assume it across different categories of any cyclic granularity. We can consider categories of an individual cyclic granularity ($A$) or combination of categories for two interacting granularities ($A*B$) to have a distribution, where $A,B$ are defined as $A = \{a_j : j = 1,2,\ldots J\}$ 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. 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, $J*K = 168$ distributions of the form $f_{i,j,k}(v)$ could be conceived for each customer $i$. Hence clustering these customers is equivalent to clustering

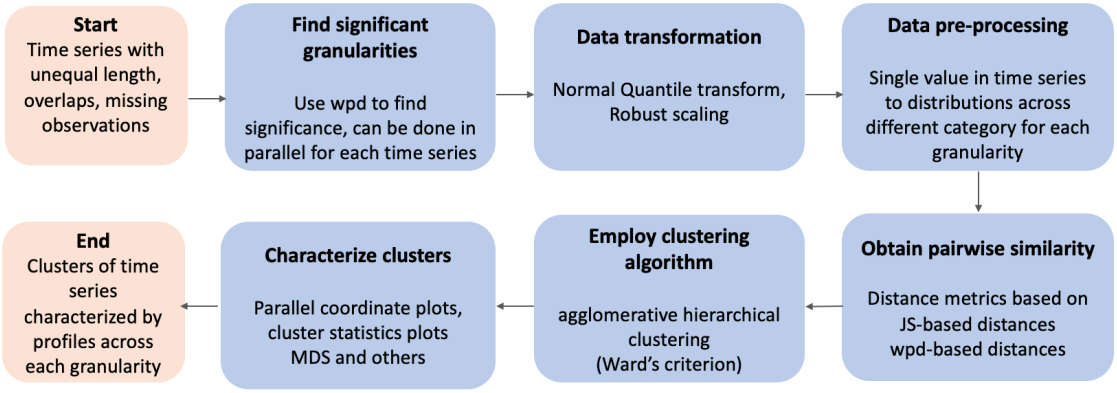



**Figure 4.1:** *Flow chart illustrating the pipeline for methodology*

these collections of conditional univariate probability distributions. Towards this goal, we need to decide how to measure similarities between collections of univariate probability distributions. There are multiple ways to measure similarities depending on the aim of the analysis. This paper considers a two approaches for finding distances between time series. Both of these approaches may be useful in a practical context and, depending on the data set, may or may not propose the same customer classification. The obtained distances could be fed into a clustering algorithm to break large data sets into subgroups that can then be analyzed separately. The methodology is explained in the Figure **??** and each element of the pipeline is discussed.

- *Find significant granularities or harmonies*

(**Gupta2021-hakear**) proposes a method for choosing significant cyclic granularities and harmonies (interacting granularities that can be studied together) (**Gupta2021-gravitas**), which is used in this work. We define "significant" granularities as those with significant distributional differences across categories. It is preferable to consider those since it is more probable that there will be some intriguing repeated behavior worth investigating. It should be noted that all of the observations in the study may not have the same set of important granularities. The following is a method for generating a list ($S_c$) of significant granularities for all observations:

(a) Remove granularities from the comprehensive list that are insignificant for all observations.

(b) select only the granularities that are significant for the majority of observations.

There will be observations in both cases where one or a few selected granularities are boring. Even in that case, having this group of observations with no interesting patterns at a granularity that regularly discovers patterns may be valuable. If, on the other hand, the granularities under examination are truly important for a group of data, unique patterns may be identified while clustering them.

- *Data transformation*

Observations often have a somewhat skewed time series distribution and their ranges might vary greatly. Statistical transformations are employed to bring all of them to the same range or normalize each observation. This is important because we are not interested in trivial clusters that vary in magnitudes, but rather in uncovering comparable patterns of distributional differences between categories. For the JS-based approaches, two data transformation techniques are utilized viz, Normal-Quantile Transform (NQT) and Robust scaling. NQT is a built-in transformation for computing $wpd$, which is the foundation of wpd-based distances.

*Robust scaling* 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 $75^{th}$ percentile of the time series for the $i^{th}$ observation. $v_{norm}$ has zero mean and median, as well as a standard deviation of one, with the outliers having same relative connections to other values.

*Normal-Quantile transform* The raw data for all observations is individually normal-quantile transformed (NQT) (**Krzysztofowicz1997-bv**), so that the transformed data follows a standard normal distribution. NQT will make the skewed distributions bell-shaped. As a result, determining which raw distribution was used is difficult using the modified distribution. Multimodality is also concealed or reversed. As a result, while displaying transformed data using NQT, one must exercise caution. But NQT is a useful transformation that often improves clustering performance.

- *Data pre-preprocessing*

We start with a "tsibble" (**wang2020tsibble**) data structure with an index variable representing inherent ordering from past to present and a key variable that defines observational units over time. The measured variable for an observation is a time-indexed sequence of values. This sequence,

however, could be shown in several ways. A shuffle of the raw sequence may represent hourly consumption throughout a day, a week, or a year. Cyclic granularities can be expressed in terms of the index set in the "tsibble" data structure. But the data structure changes while transporting from linear to cyclic scale of time as multiple observations now correspond to each category and induce a probability distribution. Directly computing Jensen-Shannon distances between the entire probability distributions can be very costly. Hence, in this paper, quantiles are chosen to characterize the probability distributions. So, in the final data structure, each category of a cyclic granularity corresponds to a list of numbers which is essentially a few chosen quantiles.

- *Distance metrics*

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

**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 (**Menendez1997-in**), which are symmetric and thus could be used as a distance measure.

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

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

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

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

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

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

is employed, which is also a distance metric since it is the sum of JS distances. This approach is expected to yield groups, such that the variation in observations within each group is in magnitude rather than distributional pattern, while the variation between groups is only in distributional pattern across categories.

**wpd-based distances**

Compute weighted pairwise distances *wpd* (**Gupta2021-hakear**) for all considered granularities for all observations. *wpd* is designed to capture the maximum variation in the measured variable explained by an individual cyclic granularity or their interaction and is estimated by the maximum pairwise distances between consecutive categories normalized by appropriate parameters. A higher value of *wpd* indicates that some interesting pattern is expected, whereas a lower value would indicate otherwise.

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

- *Clustering 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 **Xu2015-ja** 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 element of cluster analysis. **Cook2007-qe** provides several methods for characterizing clusters. *Parallel coordinate plots* (**wegman1990hyperdimensional**), *Scatterplot matrix*, *Displaying cluster statistics* (**dasu2005grouping**), *MDS* (**borg2005modern**), PCA, *t-SNE*(**R-tsne**), *Tour* (**wickham2011tourr**) are some of the graphical approaches used in this study. A Parallel Coordinates Plot features parallel axes for each variable and each axis is linked by lines. Changing the axes may reveal patterns or relationships between variables for categorical variables. However, for categories with cyclic temporal granularities, preserving the underlying ordering of time is more desirable. Displaying cluster statistics is useful when we have larger problems and it is difficult to read the parallel coordinate plots due to congestion. All of MDS, PCA and t-SNE use a distance or dissimilarity matrix to construct a reduced-dimension space representation, their goals are diverse. Multidimensional scaling (**borg2005modern**) seeks to maintain the distances between pairs of data points, with an emphasis on pairings of distant points in the original space. The t-SNE embedding will compress data points that are close in high-dimensional space. Tour is a collection of interpolated linear projections of multivariate data into lower-dimensional space. The cluster characterization approach varies depending on the distance metric used. Parallel coordinate plots, scatter plot matrices, MDS or PCA are potentially useful ways to characterize clusters using wpd-based distances. For JS-based distances, plotting cluster statistics is beneficial for characterization and variable importance could be displayed through parallel coordinate plots.

## 4.3 Validation

To validate our clustering methods, we spiked many attributes in the data to create different data designs. Three circular granularities $g1$, $g2$ and $g3$ are considered with categories denoted by $\{g10, g11\}$, $\{g20, g21, g22\}$ and $\{g30, g31, g32, g33, g34\}$ and levels $n_{g_1} = 2$, $n_{g_2} = 3$ 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 number of series. All designs employ $T = \{300, 1000, 5000\}$ sample sizes to evaluate small, medium, and large-sized series. Variations in method performance may be due to different jumps between categories. So a mean difference of $\mu = \{1, 2, 5\}$ between categories is considered. The performance of the approaches varies with the number of granularities which has interesting patterns across its categories. So three scenarios are considered to accommodate that.

### 4.3.1 Data generating processes

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

### 4.3.2 Data designs

#### Individual granularities

*Scenario (a): All significant granularities*

**Table 4.1:** *For Scenario (a), distributions of different categories when they vary (top). If distributions are fixed, they are set to N(0, 1). 5 designs resulting from different distributions across categories (below)*

| granularity | Varying distributions |
|---|---|
| g1 | g10 ~ N(0, 1), g11 ~ N(2, 1) |
| g2 | g21 ~ N(2, 1), g22 ~ N(1, 1), g23 ~ N(0, 1) |
| g3 | g31 ~ N(0, 1), g32 ~ N(1, 1), g33 ~ N(2, 1), g34 ~ N(1, 1), g35 ~ N(0, 1) |

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

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 **??** various distributions across categories are considered (top) which lead to different designs (bottom). Figure **??** 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 (b): 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 **??**(b).

*Scenario (c): One significant granularity*

Only one granularity is responsible for identifying the designs in this case. This is depicted in Figure **??** (right) where only $g3$ affects the designs significantly.

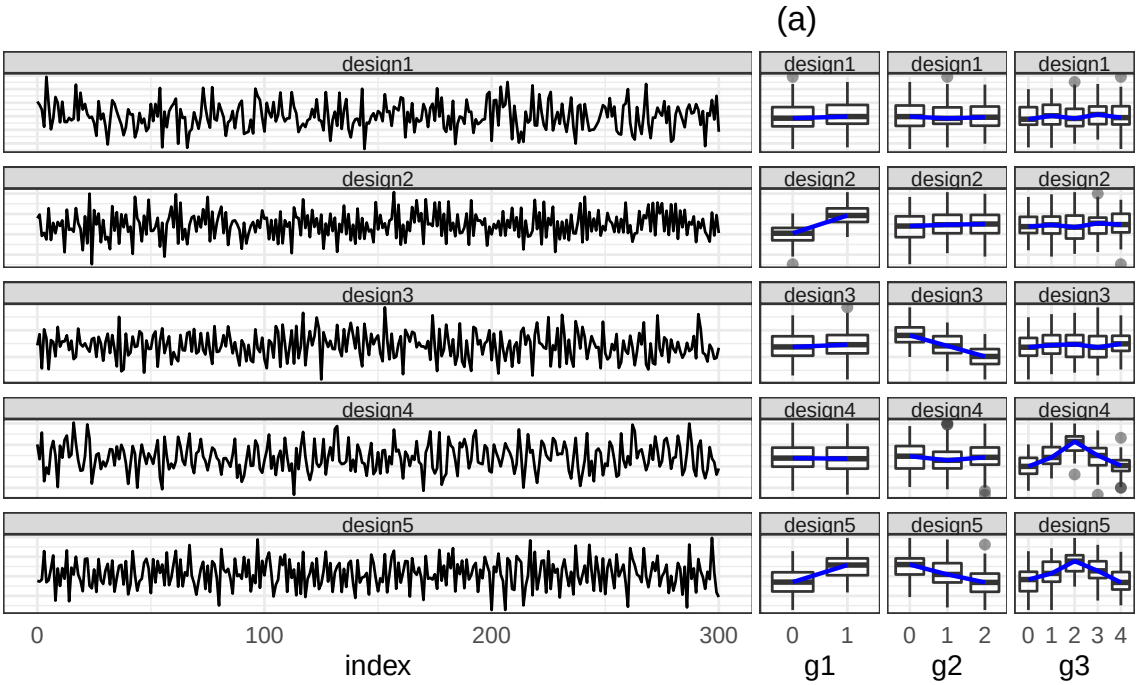



**Figure 4.2:** *The linear (left) and cyclic (right) representation of the simulated variable is shown. Each row represents a design in Scenario (a). In this scenario, all of g1, g2 and g3 changes across at least one design. Also, it is not possible to comprehend these differences in patterns just by looking at or considering the linear representation.*

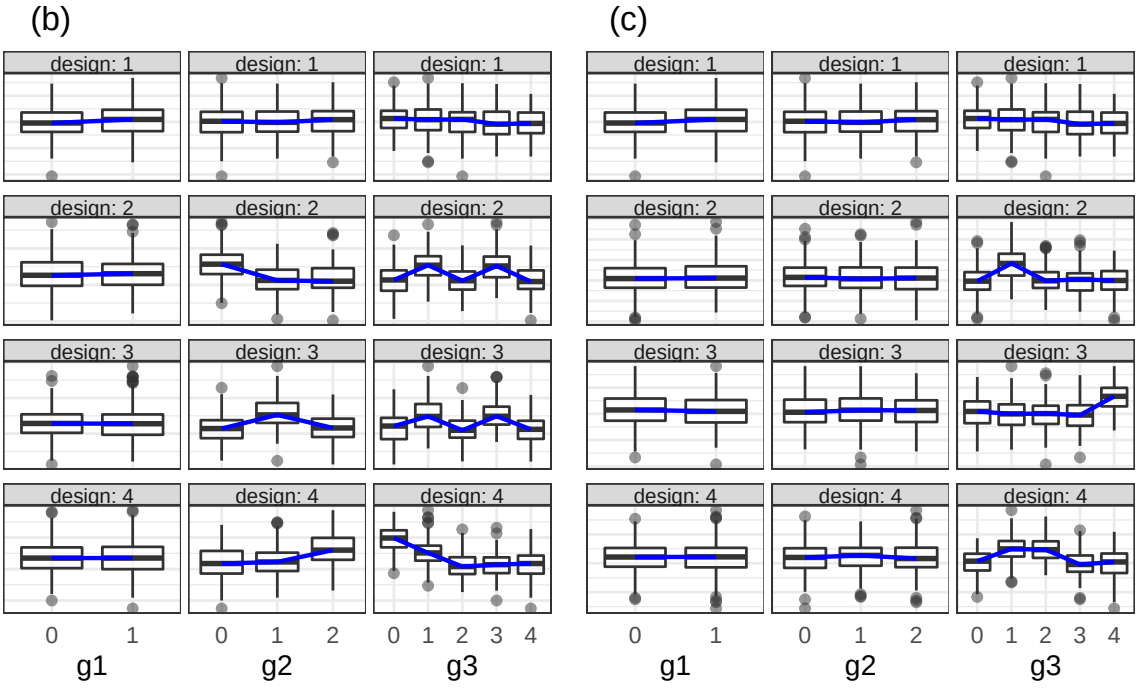



**Figure 4.3:** *Plots (b) and (c) correspond to Design (b) and (c) respectively. In (b) g2, g3 changes across atleast one design but g1 remains constant. Only g3 changes across different designs in (c).*

**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. Consider a case in which there are only two interacting granularities of interest, $g1$ and $g2$. In contrast to the previous situation, when we could study distributions across $n_{g_1} + n_{g_2} = 5$ separate categories, with interaction, we must evaluate the distribution of the $n_{g_1} * n_{g_2} = 6$ combination of categories. Consider the 4 designs in Figure **??**, where various distributions are assumed for different combinations of categories, resulting in different designs. Design $D1$ exhibits no change in distributions across $g1$ or $g2$, whereas Designs $D2$ and $D3$ alter across only $g1$ and $g2$, respectively. D4 varies across both $g1$ and $g2$ categories. D3 and D4 appear similar based on their relative differences across consecutive categories, but $D4$ also changes across facets, unlike $D3$, which has all facets look the same.

### 4.3.3 Visual exploration of findings

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

- In Figure **??**, we tried to see how separated our clusters are. We observe that in all scenarios and for different mean differences, clusters are separated. However, the separation increases with an increase in mean differences across scenarios. This is intuitive because, as the difference between categories increases, it gets easier for the methods to correctly distinguish the designs.

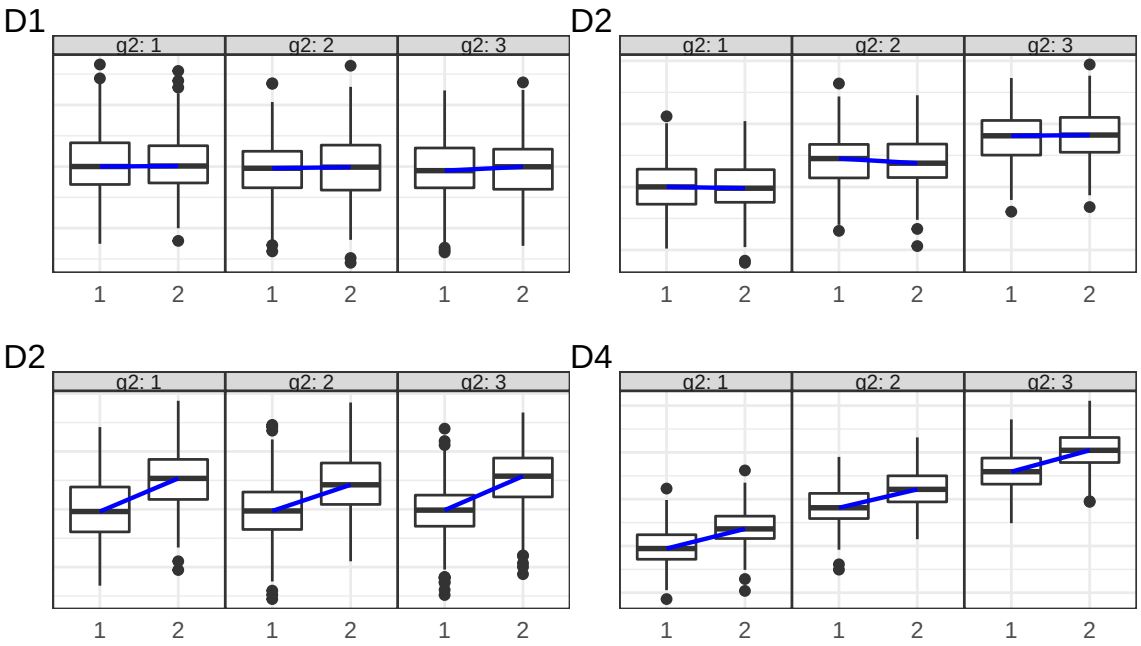



**Figure 4.4:** *Distribution of the simulated variable across $g1$ conditional on $g2$ is shown through boxplots for 4 designs. D1 has no change in distributions across different categories of $g1$ or $g2$, while D2 and D3 change across only $g1$ and $g2$ respectively. D4 changes across categories of both $g1$ and $g2$.*

- Figure **??** depicts a parallel coordinate plot with the vertical bar showing total inter-cluster distances with regard to granularities $g1$, $g2$, and $g3$ for all simulation settings and scenarios. So one line in the figure shows the inter-cluster distances for one simulation setting and scenarios vary across facets. The lines are not colored by group since the purpose is to highlight the contribution of the factors to categorization rather than class separation. The first plot shows that no variable stands out in the clustering, but the following two designs show that {g1} and {g1, g2} have very low inter cluster distances, meaning that they did not contribute to the clustering. It is worth noting that these facts correspond to our original assumptions when developing the scenarios, which incorporate distributional differences over three (a), two (b), and one (c) significant granularities. Hence, Figure **??** (a), (b), and (c) validate the construction of scenarios (a), (b), and (c) respectively.

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

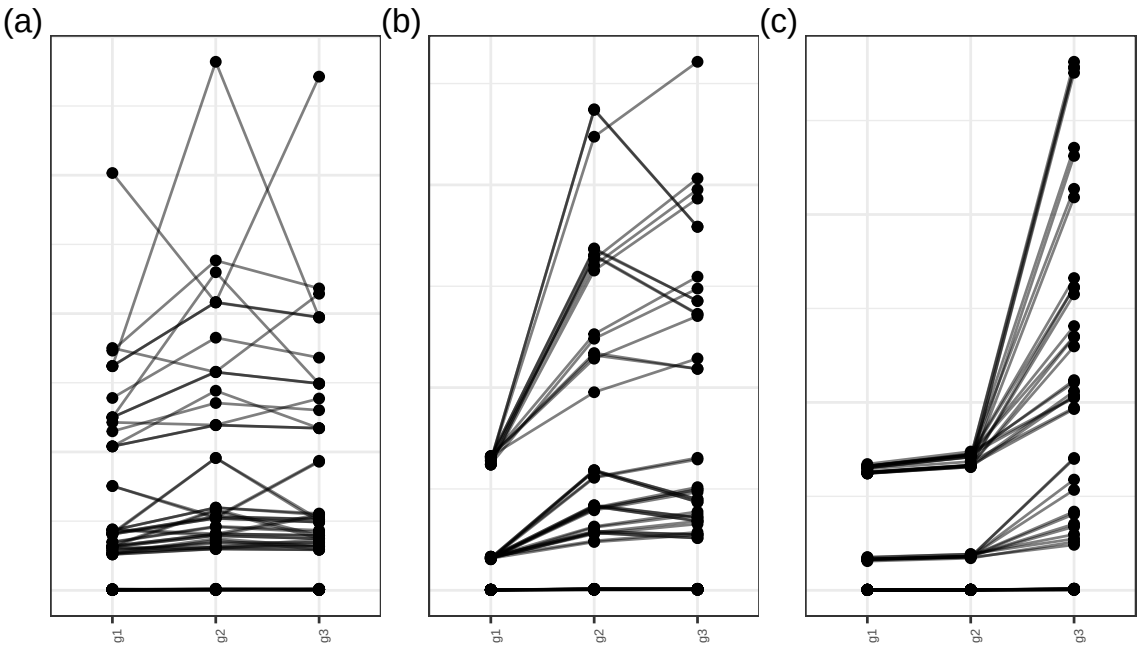



**Figure 4.5:** *The parallel coordinate plot illustrates the total inter-cluster distances for granularities g1, g2, and g3. One line in the figure depicts the inter-cluster distances for a single simulation scenario. While the first plot indicates that no variable stands out during clustering, the next two designs demonstrate that g1 and g1, g2 have extremely low inter-cluster distances, indicating that they did not contribute to clustering. It is worth emphasising that these facts are consistent with our initial assumptions when designing the scenarios and (a), (b), and (c) correspond to Scenario (a), (b) and (c) respectively.*

- In our study sample, the method js-nqt outperforms the method js-robust for smaller differences between categories. More testing, however, is required to corroborate this.

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

## 4.4 Case study

The use of our methodology is illustrated on smart meter energy usage for a sample of customers from SGSC consumer trial data which was available through Department of the Environment and Energy and Data61 CSIRO. It contains half-hourly general supply in KwH for 13,735 customers, resulting in 344,518,791 observations in total. In most cases, electricity data is expected to have multiple seasonal patterns like daily, weekly or annual. We do not learn about these repetitive behaviors from the linear view because too many measurements all squeezed in that representation.

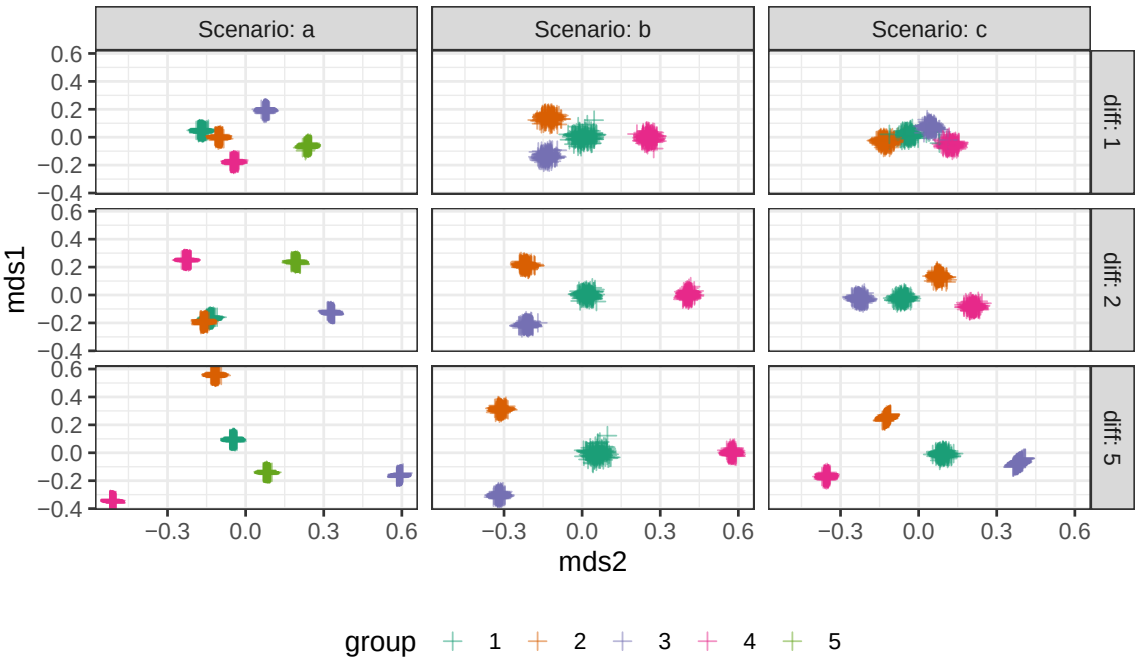



**Figure 4.6:** *Relative positions of clusters corresponding to different scenarios (columns) for different values of mean differences between categories (rows) are shown using the first two dimensions of MDS. It can be observed that clusters become more compact and separated for higher mean differences between categories across all designs. Between designs, separation is least prominent corresponding to scenario (c) where only granularity is responsible for the clusters.*

Hence we transition into looking at cyclic granularities, that can potentially provide more insight on their repetitive behavior. The raw data for these consumers is of unequal length, with varying start and finish dates. Because our proposed methods evaluate probability distributions rather than raw data, neither of these data features would pose any threat to our methodology unless they contained any structure or systematic patterns. Additionally, there were missing values in the database but further investigation revealed that there is no structure in the missingness (see Supplementary paper for raw data features and missingness). The study begins by subsetting a data set along all dimensions of interest using data filtering and prototyping. By grouping the prototypes using our methods and assessing their meaning, the study hopes to unravel some of the heterogeneities observed in energy usage data. Because our application does not employ additional customer data, we cannot explain why consumption varies, but rather try to identify how it varies.

*Data filtering and variable selection*

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

*Prototype selection*

Supervised learning uses a training set of known information to categorize new events through instance selection. Instance selection (**olvera2010review**) is a method of rejecting instances that are not helpful for classification. This is analogous to subsampling the population along all dimensions of interest such that the sampled data represents the primary features of the underlying distribution. Instance selection in unsupervised learning has received little attention in the literature, yet it could be a useful tool for evaluating model or method performance. There are several ways to approach prototype selection. Following **Fan2021-bq**'s idea of picking related examples (neighbors) for each instance (anchor), we can first use any dimensionality reduction techniques like MDS or PCA to project the data into a 2D space. Then pick a few "anchor" customers who are far apart in 2D space and pick a few neighbors for each. Unfortunately, this does not ensure that consumers with significant patterns across all variables are chosen. Tours can reveal variable separation that was hidden in a single variable display better than static projections. Hence we perform a linked tour with a t-SNE layout using the R package `liminal` (**R-liminal**) to identify customers who are more likely to have distinct patterns across the variables studied. (Refer to Supplementary article for further details). Figure **??** shows the distribution across `hod` (a), `moy`(b) and `wnwd` (c) for the set of chosen 24 customers that were chosen. Few of these customers have similar distribution across `moy` and some are similar in their `hod` distribution.

### 4.4.1 Clustering

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

**JS-based distances**

For clustering based on JS-based distances, we chose the optimal number of clusters using (**Hennig2014-ah**) as 5. The groupings are shown in Figure **??**. Customers with the same color represent the same clustered groups. Our methodology is useful for grouping similar distributions over `hod` and `moy` and they are placed closely for easy comparison. Few groups have mixed patterns across `hod` and `moy`, but few have all customers in the group having a similar profile. Figure **??** shows the summarized distributions across 5 groups and assists us in characterizing each cluster. It shows Groups 2 and 4 have `hod` pattern with a typical morning and evening peak, whereas groups 1, 3, and 5 show a `moy` pattern with higher usage in winter months. Differences in Weekend/Weekday between groups are not discernible, implying that it may not be a relevant variable in distinguishing various clusters unless maybe conditioned by `moy` or `hod`. It may be interesting to compare these two plots to verify if the summarized distributions across groups correctly characterised the groupings. If it has, then the majority of the group's members should share a similar profile.

**wpd-based distances**

We chose the optimal number of clusters using (**Hennig2014-ah**) as 3. A parallel coordinate plot with the three significant cyclic granularities is used to characterise the groups here. The variables are sorted according to their separation across classes (rather than their overall variation between classes). This means that *moy* is the most important variable in distinguishing the groups followed

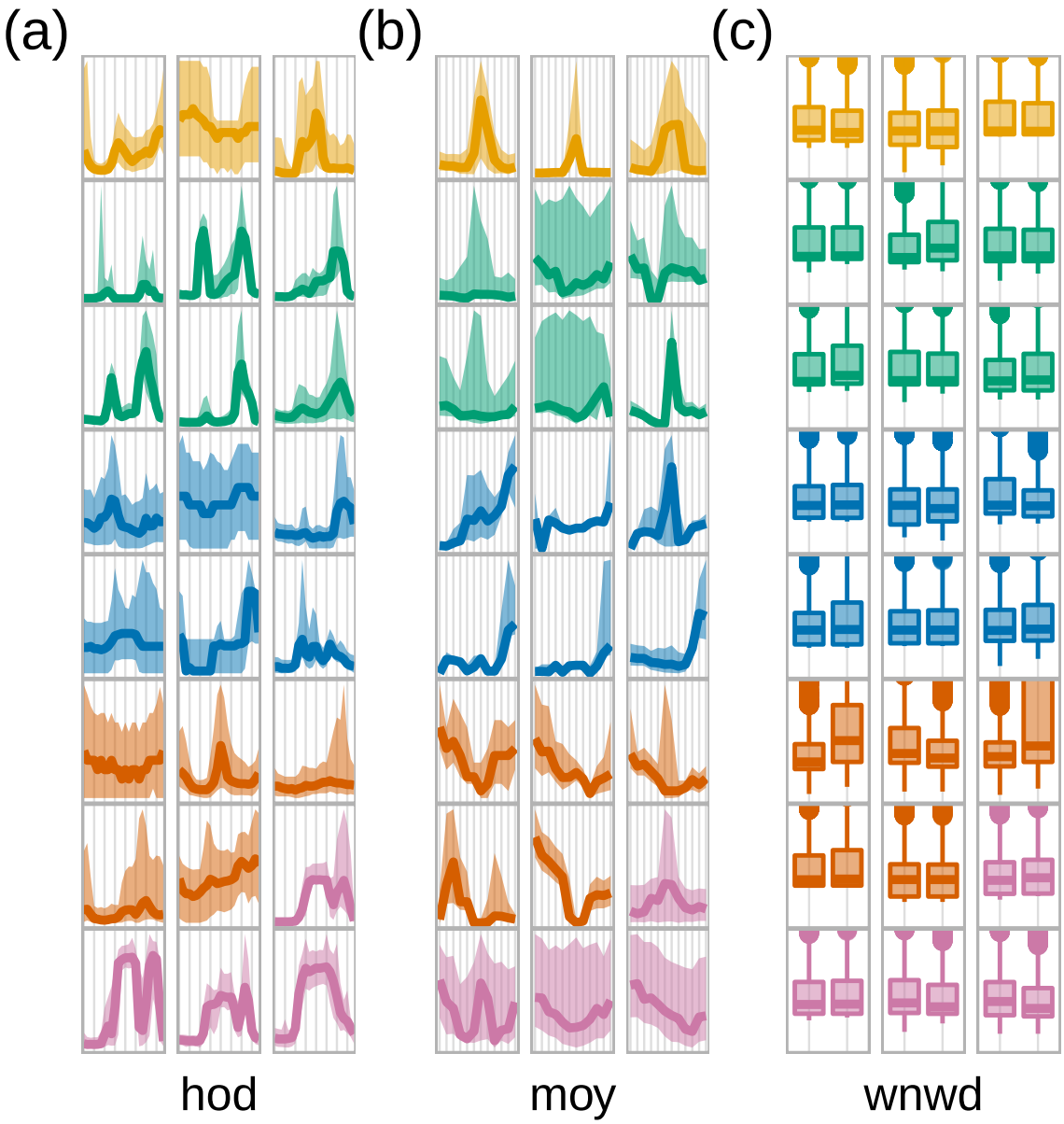



**Figure 4.7:** *The distribution of selected customers over hod (a), moy (b), and wnwd (c) for the* 24 *selected customers is shown. In each case, the same colour denotes the same group using js-based clustering and are placed together to facilitate comparison. Individuals with similar distributional differences across categories of 'hod' or 'moy' are grouped together, with some exceptions.*

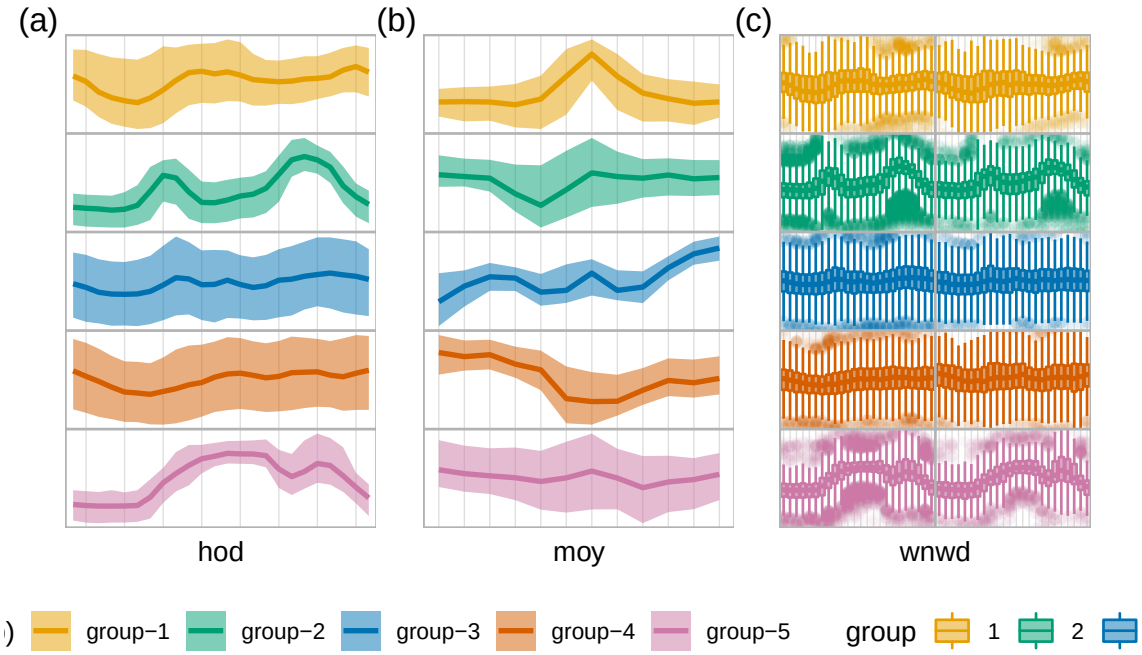
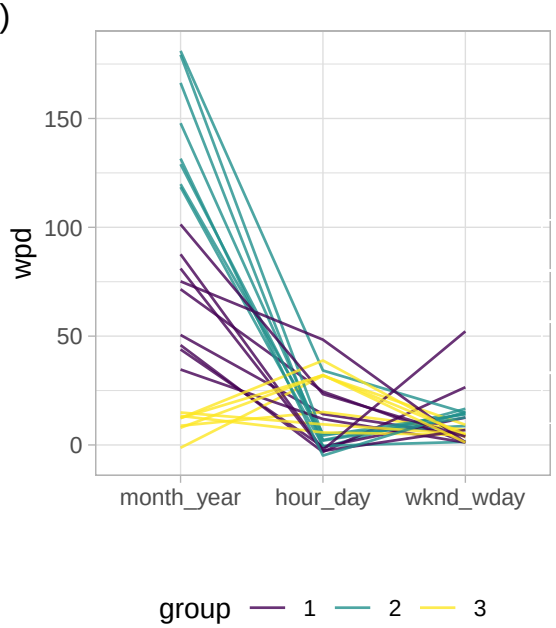



**Figure 4.8:** *The distribution of electricity demand over hod (a), moy (b), and wnwd*hod (c). Groups 2 and 5 appear to have a hod pattern among its members, whereas groups 1, 3, and 5 appear to have a moy pattern. Weekend/weekday differences across groups are not discernible, showing that it is not a critical variable for clustering on its own. It is probable that the distribution over Weekend/Weekday will differ conditional on the categories of hod or moy. (c) demonstrates that the distribution is not different, except for the tails. It is useful to compare the summarised distributions of groups to those of individuals in order to establish that the majority of individuals in the group share the same characteristics.*

by *hod* and *wnwd*. There is only one customer who has significant *wpd* across *wnwd* and stands out from the rest of the customers. Group 3 has a higher wpd for hod than moy or wkndwday. Group 2 has the most distinct pattern across moy. Group 1 is a mixed group that has strong patterns on at least one of the three variables. The findings vary from js-based clustering, yet it is a helpful grouping.

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

(a)




(b)

| | nobs | moy | hod | wnwd |
|---|---|---|---|---|
| *group−1* | 9 | 71.55 | 11.91 | 4.38 |
| *group−2* | 8 | 139.7 | 2.13 | 14.24 |
| *group−3* | 7 | 12.33 | 31.8 | 4.95 |




**Figure 4.9:** *Each of the 24 customers is represented by a parallel coordinate plot (a) with three wpd-based groupings. The plot shows that moy is the most important variable in identifying clusters, whereas wkdn-wday is the least significant and has the least fluctuation. One particular customer with high wpd across wknwday stands out in this display. Group 3 has a higher wpd for hod than moy or wkndwday. Group 2 has most discernible pattern across moy. Group 1 is a mixed group with strong patterns on atleast one of the three variables. All of these could be observed from the plot or the table (b) which shows median wpd values for each group.*

prototypes in the data set, this method might be used to classify the entire data set into these prototype behaviors. However, the goal of this section was to have a few customers that have significant patterns over one or more cyclic granularities, apply our methodology to cluster them, and demonstrate that the method produces useful clusters.

## 4.5 Conclusion and future work

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

granularity (such as hour-of-day), our method is more comprehensive in detecting clusters with repeated patterns at all relevant granularities.

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

## Acknowledgments

The authors thank the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) for supporting this research. Sayani Gupta was partially funded by Data61 CSIRO during her PhD. The Monash eResearch Centre and eSolutions-Study Support Services supported this research in part through the resource usage of the MonARCH HPC Cluster. The Github repository, github.com/Sayani07/paper-gracsr, 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-language**), `knitr` (**knitr**) and `rmarkdown` (**rmarkdown**). Graphics are produced with `ggplot2` (**Wickham2009pk**) and GGally (**R-GGally**).

## 4.6 Supplementary Materials

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

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

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

# Chapter 5

# Conclusion and future plans

In this thesis, I present a systematic approach for visualizing and analyzing large temporal data distributions. The building blocks of this framework are presented in this thesis. Chapter **??** includes tools for computing all cyclic granularities as well as a recommendation system for selecting pairs of granularities that may be effectively investigated together. These temporal granularities may be used to generate data visualizations to search for patterns, associations, and anomalies. 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. Chapter **??** presents a framework for selecting displays that are interesting, with the greatest differences between the displayed distributions, and ranking them in order of priority for capturing the most variation. Both of these are used for studying patterns in individual time series or comparing a few time series together. In Chapter **??**, it is extended to allow for the exploration of distributions for many time series at the same time by clustering them based on probability distributions across informative cyclic granularities. Through the use of probability distribution, this technique is more comprehensive in recognizing clusters with recurring patterns over many important granularities and more resilient to noisy, patchy, and uneven length time series.

## 5.1 Software development

This thesis focuses on integrating research approaches into open source R packages such as **gravitas**, **hakear**, and **gracsr**.

### 5.1.1 gravitas

The **gravitas** package provides very general tools to compute and manipulate cyclic granularities, and to generate plots displaying distributions conditional on those granularities. It can be utilized in non-temporal cases for which a hierarchical structure can be construed similar to time. It is on CRAN. The website (https://sayani07.github.io/gravitas) includes full documentation and two vignettes about the package usage. There has been a grand total of 12K downloads from the RStudio mirror dating from 2020-11-01 to 2021-11-01.

### 5.1.2 hakear

The open-source R package `hakear` is available on Github (https://github.com/Sayani07/hakear) to implement ideas presented in Chapter **??**. Given a `tsibble` and context granularities of interest, the function `wpd()` provides support for computing the wpd for each cyclic granularities or pair of granularities and `select_harmonies()` chooses the ones with significant patterns and ranks them from highest to lowest wpd.

### 5.1.3 gracsr

The open-source R package `gracsr` is available on Github (https://github.com/Sayani07/gracsr) to implement ideas presented in Chapter **??**. The package provides functions to carry out the entire clustering methodology discussed in the paper. It is still a work in progress and has won the ACEMS Business Analytics Prize 2021 with a prize money of AUD 3000, which would be utilized in polishing this package and preparing it for CRAN.

## 5.2 Future work

### 5.2.1 Putting all functionalities on CRAN

I plan to integrate `hakear` with `gravitas` as one R package to systematically exploring few time series, whereas `gracsr` would provide the clustering framework for exploring many time series together. I plan to run it through https://ropensci.org/software-review/ to develop them further for more visibility and efficient usage.

### 5.2.2 Scaling up the clustering method to incorporate large uncertainty and improved computational efficiency that comes with large data

With the volume of data projected to grow in the future, potentially leading to increased variability in patterns, research is needed to understand the issues that may arise with the existing methodology and how to adapt our current algorithm. Computational efficiency is also critical when scaling up for analysis of huge data sets, let alone when adding features to existing highly dimensional data structures.

### 5.2.3 Comparing our clustering method with other benchmark methods

Testing needs to be carried out with non-hierarchy based clustering methods to see whether these distances perform better with other algorithms. Also, to evaluate how much information is lost when aggregating individual level demand versus distributions.

### 5.2.4 Check generalizability of our methods

We provide solutions that are realistically applicable to any temporal data observed more than once per year. We could just verify its value in the electricity smart meter context. With numerous open-source benchmark data sets accessible, it is necessary to test how well the approaches operate in various disciplines.

## 5.3 Final words

# Appendix A

# Data dictionary

**Temporary page!**

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.