



**MONASH** University

# **Visualization and analysis of probability distributions of large temporal data**

Yiru (Earo) Wang

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

<b>Copyright notice</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Declaration</b>	<b>ix</b>
<b>Acknowledgements</b>	<b>xi</b>
<b>Preface</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Visualizing probability distributions across bivariate cyclic temporal granularities . . . . .	3
1.2 Detecting distributional differences between temporal granularities for exploratory time series analysis . . . . .	4
1.3 Clustering time series based on probability distributions across temporal granularities . . . . .	4
1.4 Summary . . . . .	5
<b>2 Visualizing probability distributions across bivariate cyclic temporal granularities</b>	<b>7</b>
2.1 Introduction . . . . .	8
2.2 Linear time granularities . . . . .	10
2.3 Cyclic time granularities . . . . .	13
2.4 Data structure . . . . .	21
2.5 Visualization . . . . .	23
2.6 Applications . . . . .	25
2.7 Discussion . . . . .	29
Acknowledgments . . . . .	30
2.8 Supplementary Materials . . . . .	30
<b>3 Detecting distributional differences between temporal granularities for exploratory time series analysis</b>	<b>35</b>
3.1 Introduction . . . . .	36
3.2 Proposed distance measure . . . . .	39
3.3 Ranking and selection of cyclic granularities . . . . .	48

3.4 Simulations . . . . .	50
3.5 Application to residential smart meter dataset . . . . .	53
3.6 Discussion . . . . .	57
Acknowledgments . . . . .	58
3.7 Supplementary Materials . . . . .	58
<b>4 Data representation, visual and analytical techniques for demystifying temporal missing data</b>	<b>59</b>
<b>5 Conclusion and future plans</b>	<b>61</b>
5.1 Software development . . . . .	61
5.2 Future work . . . . .	63
5.3 Final words . . . . .	65
<b>A Data dictionary</b>	<b>67</b>
<b>Bibliography</b>	<b>69</b>

# Copyright notice

© Sayani Gupta (2021).

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.



# Abstract

This work is driven by the need for a conceptual framework to tackle temporal analyses in a data-centric workflow. Most research on temporal data focuses on modeling. Corresponding software requires very stringently formatted data, but does little in providing guidelines or tools for wrangling raw data into the required format. This has led to ad-hoc, and once-off solutions, which this research repairs.

There are three original contributions for the temporal data analysis in this research. They are grounded in the spirit of exploratory data analysis for time-indexed data. The first contribution (Chapter 2) is a new technique for visualizing data using a calendar layout. It is most useful when the data relates to daily human activity, and patterns can be explored in relation to people’s schedules. The second contribution (Chapter 3) is a new data abstraction which streamlines transformation, visualization, and modeling in temporal contexts. This “tsibble” object is a data infrastructure forming the foundation of temporal data pipelines. The third contribution (Chapter ??) is to equip analysts with exploratory and explanatory tools for discovering missing patterns in time, and thus facilitating decisions on appropriate imputation methods for further downstream analysis.





# Declaration

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

This thesis includes 3 publications, two of which have been revised and resubmitted and one not yet submitted. The core theme of the thesis is “tidy tools for temporal data”. The ideas, development and writing up of all the papers in the thesis were the principal responsibility of myself, the student, working within the Department of Econometrics and Business Statistics under the supervision of Professor Dianne Cook and Professor Rob J Hyndman.

The inclusion of co-authors reflects the fact that the work came from active collaboration between researchers and acknowledges input into team-based research.

In the case of Chapter 2, 3 and 4 my contribution to the work involved the following:

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

**Student name:** Yiru (Earo) Wang

**Student signature:**

**Date:** 2021-10-25

<b>Thesis Chapter</b>	<b>Publication Title</b>	<b>Status (published, in press, accepted or re-turned for revision)</b>	<b>Nature and % of student contribution</b>	<b>Co-author name(s) Nature and % of Co-author's contribution</b>	<b>Co-author(s), Monash student Y/N</b>
2	Calendar-based graphics for visualizing people's daily schedules	Returned for revision	70%. Concept and developing software and writing first draft	(1) Dianne Cook, Concept and input into manuscript 20% (2) Rob J Hyndman, input into manuscript 10%	N
3	A new tidy data structure to support exploration and modeling of temporal data	Returned for revision	80%. Concept and developing software and writing first draft	(1) Dianne Cook, input into manuscript 15% (2) Rob J Hyndman, input into manuscript 5%	N
4	Data representation, visual and analytical techniques for demystifying temporal missing data	To be submitted	80%. Concept and developing software and writing first draft	(1) Dianne Cook, Concept and input into manuscript 15% (2) Rob J Hyndman, input into manuscript 5%	N

# Acknowledgements

This thesis would not be possible without many people.

First and foremost, I would like to express my gratitude to my supervisors, Dianne Cook and Rob J Hyndman. They are outstanding supervisors, providing excellent insights on statistical computing and graphics during our weekly supervisory meetings. I'm deeply impressed by their dynamics, enthusiasm, and visions about open source software and data science, all of which have formed the support and encouragement for me to continue with this non-traditional research path.

I thank Heike Hofmann for hosting me at Iowa State University during the summer of 2018. I enjoyed many discussions with Heike, where I received brilliant questions and insightful advice on my research. I thank Stuart Lee and Mitchell O'Hara-Wild for countless conversations on software development, that helped to shape this thesis. I thank David Frazier for being the best pingpong partner, truly boosting the creative thinking process.

I thank Emi Tanaka for sending me PRs for proofreading my thesis on Github, unlocking the power of an open-sourced research compendium. I thank Wenjing Wang and Carson Sievert for being so helpful in the early days of my PhD. I thank the R community, the tidyverse team, and all NUMBATs for ongoing inspiration.

Last but not least, I'd like to thank my mother for her unceasing love.



# Preface

Chapter 2 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 3 and Chapter ?? are yet to be submitted.

## Open and reproducible research

This thesis is written in R Markdown (Xie, Allaire, and Golemund, 2018) with **bookdown** (Xie, 2016), using **renv** (Ushey, 2019) 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

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#).

The code used in this document is available under the [MIT license](#).



# Chapter 1

## Introduction

To tame the complexity of time, breaking it into years, months, weeks, days and so on in a hierarchical manner is a common way to relate data to time. Such discrete human made abstractions of time can be thought of as time granularities.(Aigner et al., 2008).

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.

Temporal data analysis has assumed that the entry point to data analysis is at a model-ready data format, which provides little organization or conceptual oversight on how one should get the wild data into a tamed state. This mind-set is related to a long-held belief that exploratory data analysis is a highly ad hoc statistical area, impossible to teach or formalize. However, the **tidyverse** framework implemented in the statistical software R (R Core Team, [2018](#)), as originating in Wickham ([2014](#)), fundamentally overturns this thinking. Data plots and data wrangling, which the “tidy data” conceptualization supports, can be formally described using an abstract grammar. The grammar of graphics and data manipulation, as implemented in the **ggplot2** (Wickham et al., [2019a](#)) and **dplyr** (Wickham et al., [2019b](#)) R packages respectively, form the core of the **tidyverse** suite of tools. My contributions extend the **tidyverse** way of thinking to the temporal domain, by providing tidy tools, built as R packages, for supporting fluent workflow in temporal data analysis.

## 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 and 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 calendar plot, algorithm and applications. This is implemented in the R package **sugrrants**. Chapter ?? explains the new data abstraction—**tsibble**—and illustrates how it can be used to for the basis of exploratory methods, visualization and modeling of temporal data. This is available as the R package **tsibble**. Chapter 4 describes new procedures for exploring temporal missing patterns, and assisting in handling missing values prior to modeling. This is in the developing R package **mists**.

Chapter 5 summarizes the software tools developed for the work and their impact, 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 [2.2](#) provides some background material on linear granularities and calendar algebra for computing different linear granularities. Section [2.3](#) formally characterizes different cyclic time granularities by extending the framework of linear time granularities, and introducing cyclic calendar algebra for computing cyclic time granularities. The data structure for exploring the conditional distributions of the associated time series across pairs of cyclic time granularities is discussed in Section [2.4](#). Section [2.5](#) discusses the role of different factors in constructing an informative and trustworthy visualization. Section [2.6](#) examines how systematic exploration can be carried out for a temporal and non-temporal application. Finally, we summarize our results and discuss possible future directions in Section [2.7](#).

## 2.2 Linear time granularities

Discrete abstractions of time such as weeks, months or holidays can be thought of as “time granularities”. Time granularities are **linear** if they respect the linear progression of time. There have been several attempts to provide a framework for formally characterizing time granularities, including [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  $T$ .



**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**.

hour	0	1	...	23	24	25	...	47	...	144	...	167	...	...	...	720	...	743	...	...	...	...	...	...	...	...	...	t							
day	0				1				...	6				...				31				...				365				...	...				t/24
week	0												...	5				...				53				...	...				t/24*7				
month	0																...	12				...	...	M											
year	0																				...	Y													

### 2.2.2 Relativities

**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  $H(i+R) \neq \emptyset$  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,  $G(a_1), G(a_2), \dots, G(a_k)$ . Condition (2) in Definition 6 implies that if  $H(i+R) \neq \emptyset$ , then  $H(i+R) = \bigcup (G(a_1+P), G(a_2+P), \dots, G(a_k+P))$ , 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<sup>th</sup> 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

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

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**).

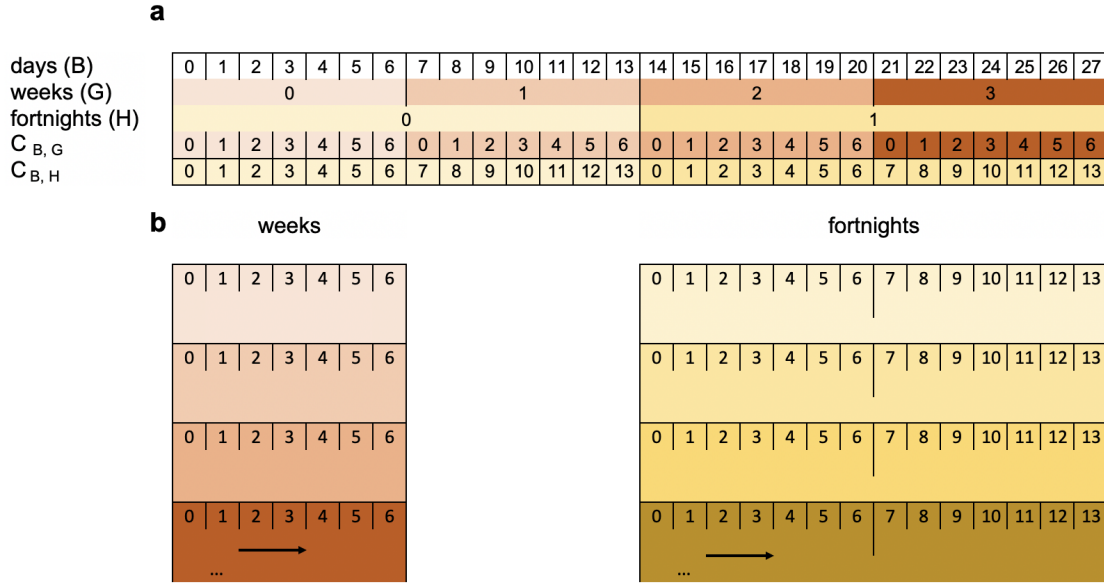
### 2.3.1 Circular granularities

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

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

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

**Figure 2.2** illustrates some linear and cyclical granularities. Cyclical granularities are constructed by cutting the linear granularity into pieces, and stacking them to match the cycles (as shown in b).  $B, G, H$  (day, week, fortnight, respectively) are linear granularities. The circular granularity  $C_{B,G}$  (day-of-week) is constructed from  $B$  and  $G$ , while circular granularity  $C_{B,H}$  (day-of-fortnight) is



**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  $\{0, 1, \dots, 6\}$  and  $\{0, 1, \dots, 13\}$  with  $P = 7$  and  $P = 14$  respectively.

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

$$L : \{0, 1, \dots, 6\} \mapsto \{\text{Sunday, Monday, } \dots, \text{Saturday}\}.$$

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

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

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

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. Table 2.2 shows some examples of quasi-circular granularities.

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

**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

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

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

For example, day-of-year is quasi-periodic with either 365 or 366 granules of  $B$  (days) within each granule of  $G'$  (years). The pattern repeats every 4 years (ignoring leap seconds). Hence  $R' = 4$ .  $Q_{B,G'}$  is a repetitive categorization of time, similar to circular granularities, except that the number of granules of  $B$  is not the same across different granules of  $G'$ .

### 2.3.3 Aperiodic granularities

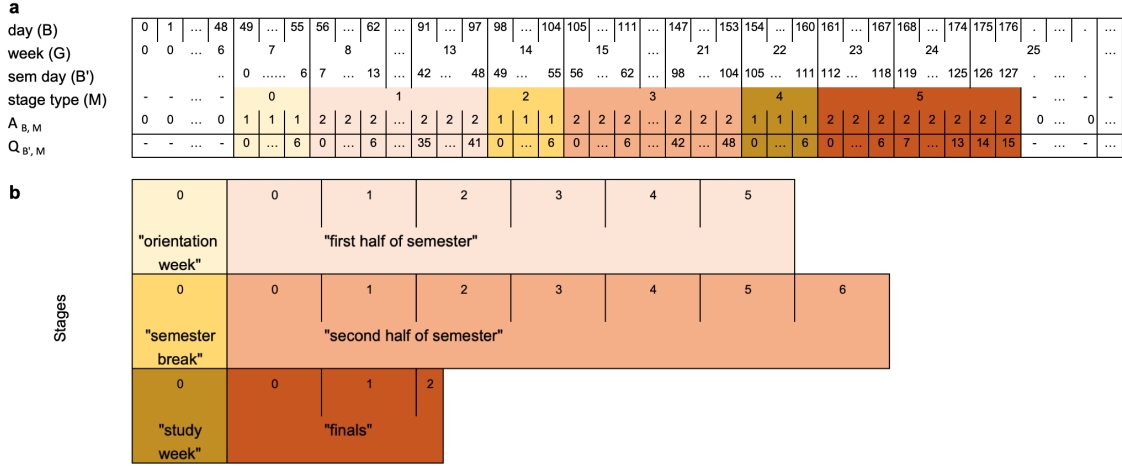
Aperiodic linear granularities are those that cannot be specified as a periodic repetition of a pattern of granules as described in Definition 6. Aperiodic cyclic granularities capture repetitions of these aperiodic linear granularities. Examples include public holidays which repeat every year, but there is no reasonably small span of time within which their behavior remains constant. A classic example is Easter (in the western tradition) whose dates repeat only after 5.7 million years (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 Table 2.2) are essentially infinite for aperiodic granularities.

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

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

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

For example, consider the school semester shown in Figure 2.3. Let the linear granularities  $M_1$  and  $M_2$  denote the teaching and non-teaching stages of the semester respectively. Both  $M_1$ ,  $M_2$  and  $M = M_1 \cup M_2$  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, Table 2.3 shows the hierarchy table for the Mayan calendar. In the Mayan calendar, one day was referred to as a kin and the calendar was structured such that 1 kin = 1 day; 1 uinal = 20 kin; 1 tun = 18 uinal (about a year); 1 katun = 20 tun (20 years) and 1 baktun = 20 katun.

**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 2.3), kin-of-tun or kin-of-baktun are examples of multiple-order-up granularities and single-order-up granularities are kin-of-uinal, uinal-of-tun etc.

### 2.3.5 Computation

Following the calendar algebra of **Ning\_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

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

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

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

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

, 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

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

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

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

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

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

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

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

### Multiple-to-single order-up

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

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

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 (2.5) and (2.6).

**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	$C_1$	$C_2$	$\dots$	$C_{N_C}$

## 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. [Table 2.5](#) shows the tsibble structure (index, key, measurements) augmented by columns of cyclic granularities. The total number of cyclic granularities depends on the number of linear granularities considered in the hierarchy table and the presence of any aperiodic cyclic granularities. For example, if we have  $n$  periodic linear granularities in the hierarchy table, then  $n(n-1)/2$  circular or quasi-circular cyclic granularities can be constructed. Let  $N_C$  be the total number of contextual circular, quasi-circular and aperiodic cyclic granularities that can originate from the underlying periodic and aperiodic linear granularities. Simultaneously encoding more than a few of these cyclic granularities when visualizing the data overwhelms human comprehension. Instead, we focus on visualizing the data split by pairs of cyclic granularities  $(C_i, C_j)$ . Data sets of the form  $\langle C_i, C_j, v \rangle$  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  $\{A_k \mid k = 1, \dots, K\}$  and  $C_j$  maps index set to a set  $\{B_\ell \mid \ell = 1, \dots, L\}$ . 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 2.4.2). Even when there are no rarely occurring events, the number of observations may vary greatly within or across each facet, due to missing observations or uneven locations of events in the time domain. In such cases, data summarization should be used with caution as sample sizes will directly affect the accuracy of the estimated quantities being displayed.

### 2.5.2 Display choices for univariate distributions

The basic plot choice for our data structure is one that can display distributions. For displaying the distribution of a continuous univariate variable, many options are available. Displays based on descriptive statistics include 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-ggribes**), 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 2.4.1) or near-clashes lead to potentially ineffective graphs. Harmonies tend to be more useful for exploring patterns. Figure 2.4a shows the distribution of half-hourly electricity consumption through letter value plots across months of the year conditional on quarters of the year. This plot does not work because quarter-of-year clashes with month-of-year, leading to empty subsets. For example, the first quarter never corresponds to December.

### Conditioning variable

When  $C_i$  is mapped to the  $x$  position and  $C_j$  to facets, then the  $A_k$  levels are juxtaposed and each  $B_\ell$  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, [Figure 2.4b](#) shows the letter value plot across weekday/weekend partitioned by quarters of the year and [Figure 2.4c](#) shows the same two cyclic granularities with their mapping reversed. [Figure 2.4b](#) helps us to compare weekday and weekend within each quarter and [Figure 2.4c](#) helps to compare quarters within weekend and weekday.

## 2.6 Applications

### 2.6.1 Smart meter data of Australia

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

#### 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 2.3.4.

### Screening and visualizing harmonies

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

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



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

### 2.6.2 T20 cricket data of Indian Premier League

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

**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

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

The Indian Premier League (IPL) is a professional T20 cricket league in India contested by eight teams representing eight different cities in India. The IPL ball-by-ball data is provided in the cricket data set in the `gravitas` package for a sample of 214 matches spanning 9 seasons (2008 to 2016).

Many interesting questions could be addressed with the `cricket` data set. For example, does the distribution of total runs depend on whether a team bats in the first or second innings? The Mumbai Indians (MI) and Chennai Super Kings (CSK) appeared in the final playoffs from 2010 to 2015. Using data from these two teams, it can be observed (Figure 2.6a) that for the team batting in the first innings there is an upward trend of runs per over, while there is no clear upward trend in the median and quartile deviation of runs for the team batting in the second innings after the first few overs. This suggests that players feel mounting pressure to score more runs as they approach the

end of the first innings, while teams batting second have a set target in mind and are not subjected to such mounting pressure and therefore may adopt a more conservative run-scoring strategy.

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

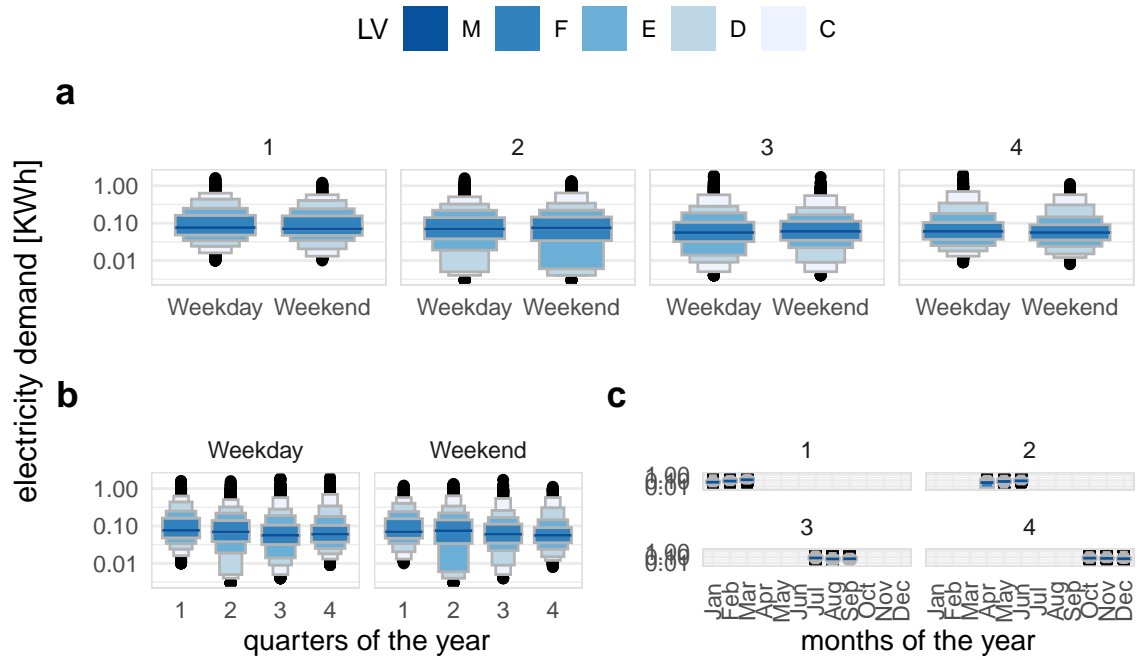
## Acknowledgments

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 Spyrisson 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` ([knitr2015](#)) and `rmarkdown` ([rmarkdown2018](#)).

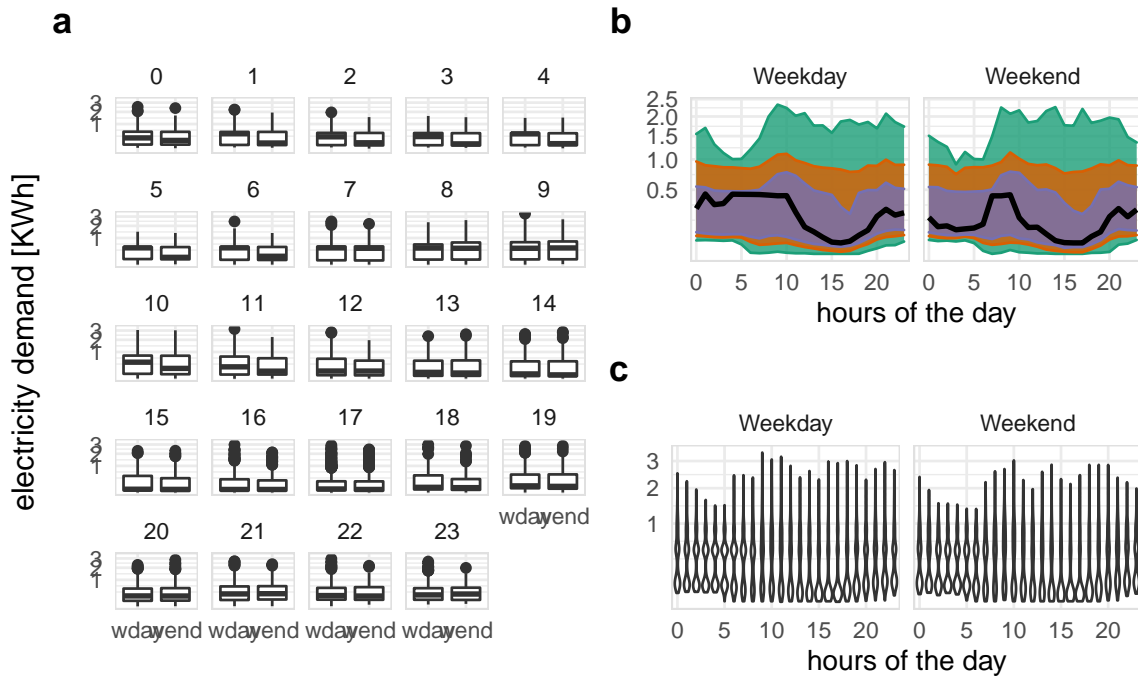
## 2.8 Supplementary Materials

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

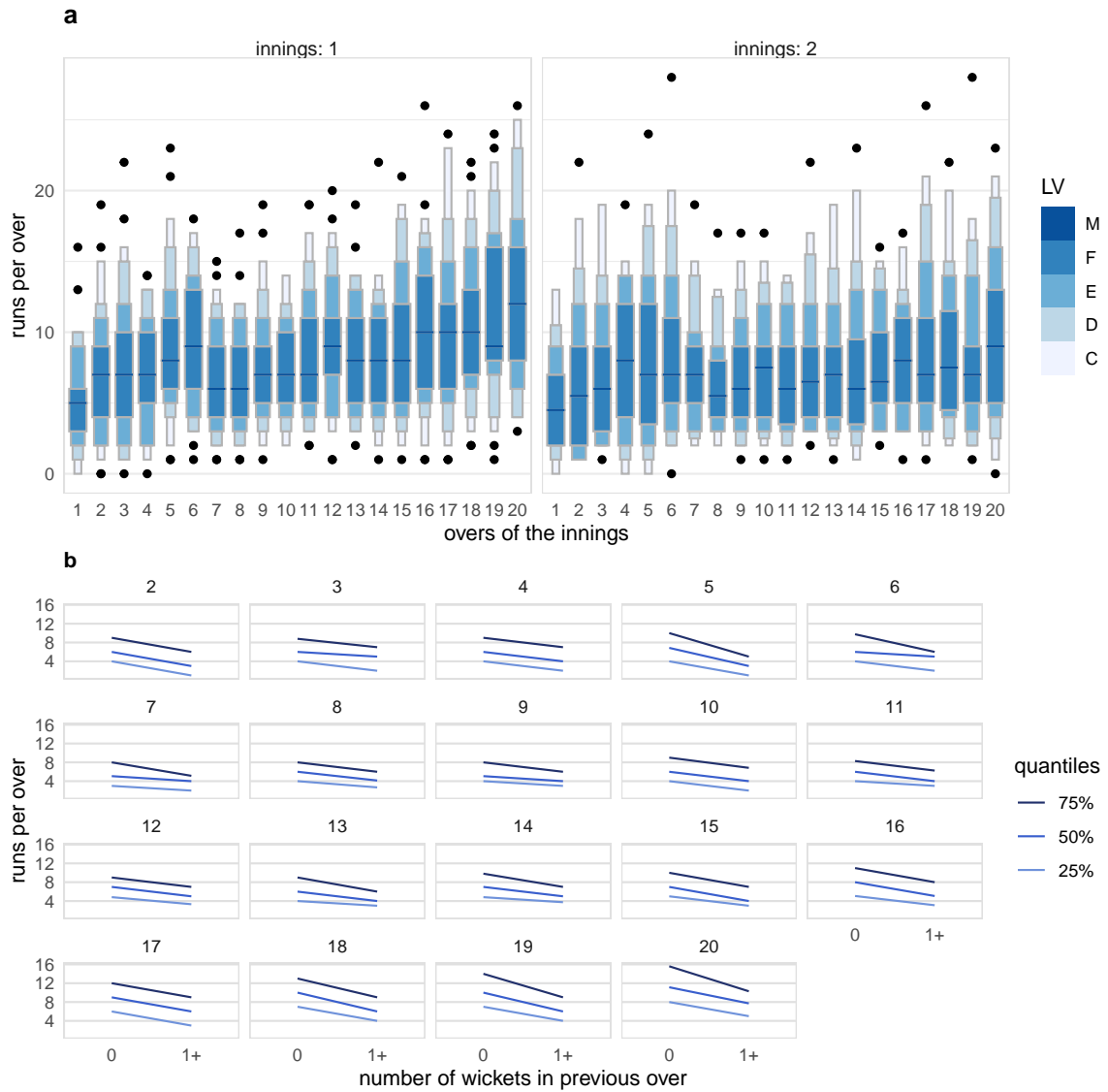
**R-package:** The ideas presented in this article have been implemented in the open-source R (**R-language**) package `gravitas` (**R-gravitas**), available from CRAN. The R-package facilitates manipulation of single and multiple-order-up time granularities through cyclic calendar algebra, checks feasibility 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

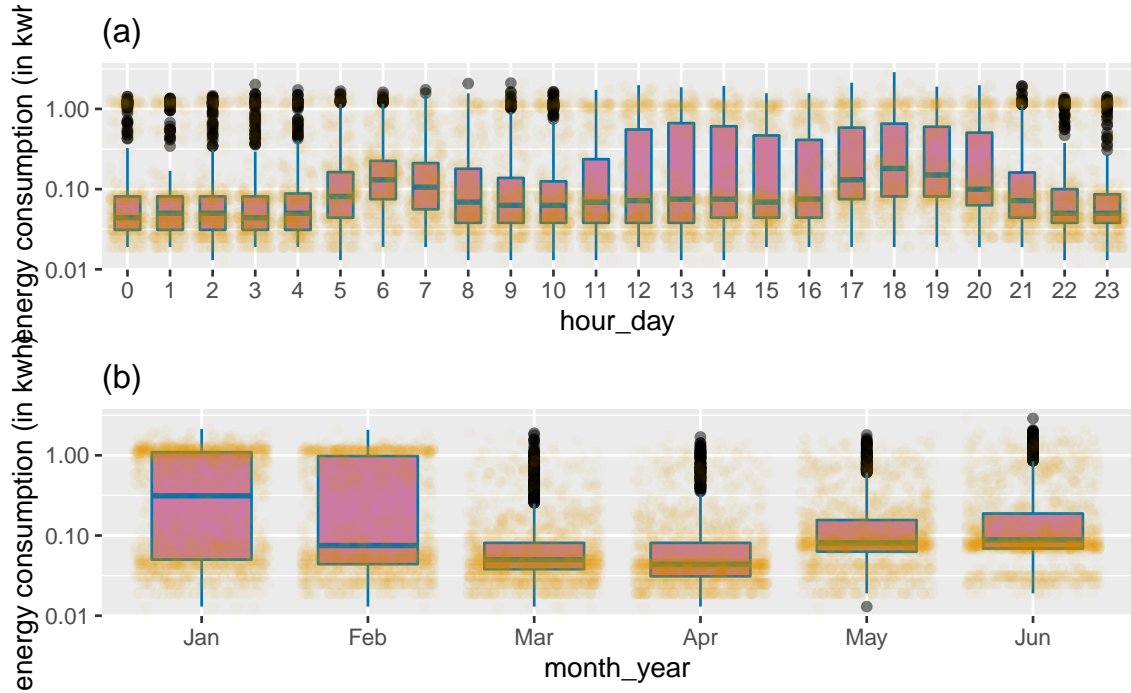
Cyclic temporal granularities, which are temporal deconstructions of a time period into units such as hour-of-the-day, work-day/weekend, can be useful for measuring repetitive patterns in large univariate time series data. The granularities feed new approaches to exploring time series data. One use is 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 and 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 3.1 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 3.2(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 energy use by a household for two different cyclic granularities: (a) hour-of-day and (b) month-of-year. Daily peaks occur in morning and evening hours, indicating a working household, where members leave for and return from work. More volatility of usage in summer months (Jan, Feb) is probably due air conditioner use on just some days.

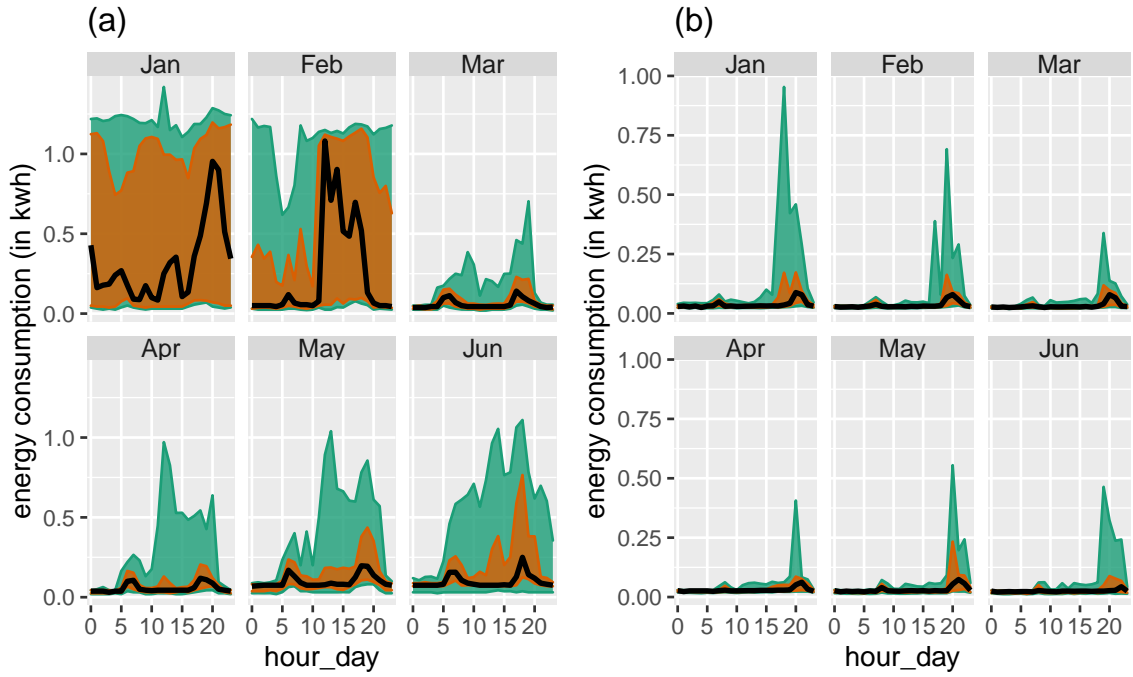
the usage distribution across hour-of-day conditional on month-of-year across two households (id 2 and 4). It shows the hourly usage over a day does not remain the same across months. Unlike other months, the 75th and 90th percentile for all hours of the day in January are high, pretty close, and are not characterized by a morning and evening peak. The household in Figure 3.2(b) has 90th percentile consumption higher in summer months relative to autumn or winter, but the 75th and 90th percentile are far apart in all months, implying that the second household resorts to air conditioning much less regularly than the first one. The differences seem to be more prominent across month-of-year (facets) than hour-of-day (x-axis) for this household, whereas they are prominent for both cyclic granularities for the first household.

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

increases when considering more than one cyclic granularity. When considering displays of the form in Figure 3.2, 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 3.2 introduces a distance measure for detecting distributional difference in temporal granularities, which enables identification of patterns in the time series data; Section 3.3 devises a selection criterion by choosing a threshold, which results in detection of only significantly interesting patterns. Section 3.4 provides some simulation study on the proposed methodology. Section 3.5 presents an application to residential smart meter data in



**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.*

Melbourne to show how the proposed methodology can be used to automatically detect temporal granularities along which distributional differences are significant.

## 3.2 Proposed distance measure

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

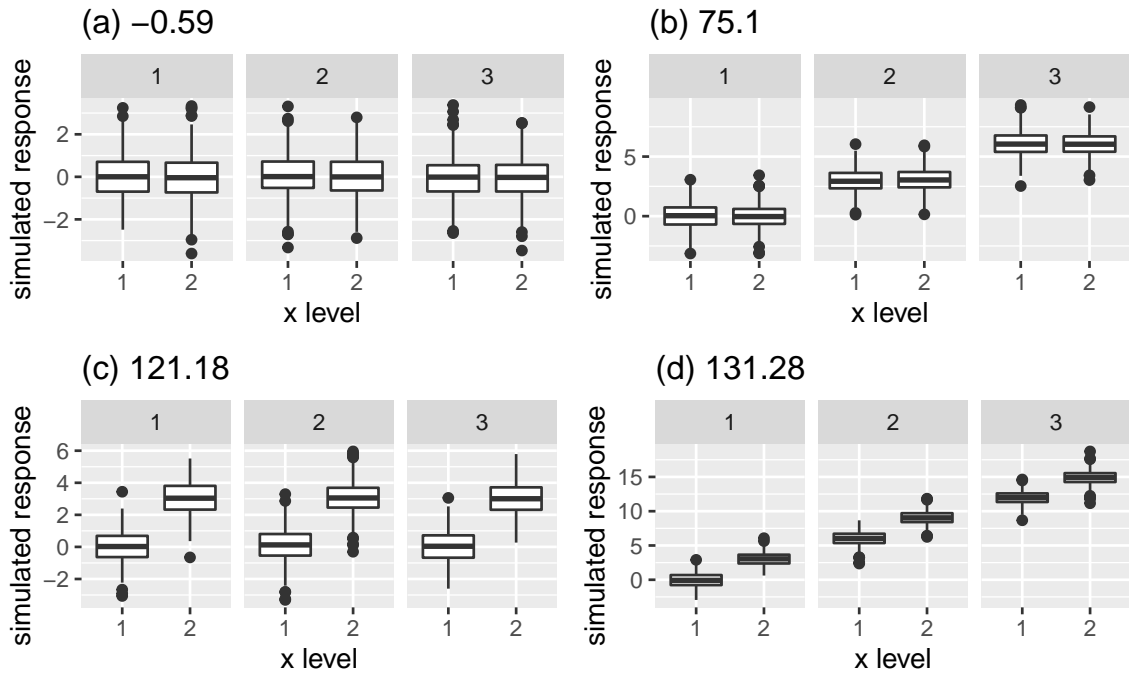
### 3.2.1 Principle

The principle behind the construction of *wpd* is explained through a simple example explained in Figure 3.3. Each of these figures describes a panel with 2 x-axis categories and 3 facet levels, but with different designs. Figure 3.3a 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 3.3b has  $x$  categories drawn from the same distribution, but across facets the distributions are 3 standard deviations apart. Figure 3.3c 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 3.3d 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  $A = \{a_j : j = 1, 2, \dots, J\}$  and  $B = \{b_k : k = 1, 2, \dots, K\}$  with  $A$  placed across  $x$ -axis and  $B$  across facets. Let  $v = \{v_t : t = 0, 1, 2, \dots, T - 1\}$  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 3.3 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.

variable	description
$N_C$	number of cyclic granularities
$H_{N_C}$	set of harmonies
$n_x$	number of x-axis categories
$n_{\text{facet}}$	number of facet categories
$\lambda$	tuning parameter
$\omega$	increment (mean or sd)
$wpd$	raw weighted pairwise distance
$wpd_{\text{norm}}$	normalized weighted pairwise distance
$n_{\text{perm}}$	number of permutations for threshold/normalization
$n_{\text{sim}}$	number of simulations
$wpd_{\text{threshold}}$	threshold for significance

variable	description
$D_{null}$	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  $v_{(1)}, \dots, v_{(i)}, \dots, v_{(n)}$ .
2. The cumulative probabilities  $p_{(1)}, \dots, p_{(i)}, \dots, p_{(n)}$  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  $(a_j, b_k)$ ,  $\forall j \in \{1, 2, \dots, J\}, k \in \{1, 2, \dots, K\}$ . 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  $p = 0.01, 0.02, \dots, 0.99$ , where for example, the 99<sup>th</sup> 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  $D(q_1 || q_2) := \int_{-\infty}^{\infty} q_1(x) f\left(\frac{q_1(x)}{q_2(x)}\right)$  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 3.4 illustrates how they are defined. Pairwise distances are within-facets when  $b_k = b_{k'}$ , that is, between pairs of the form  $(a_j b_k, a_{j'} b_k)$  as shown in panel (3) of Figure 3.4. If categories are ordered (like all temporal cyclic granularities), then only distances between pairs where  $a_{j'} = (a_{j+1})$  are considered (panel (4)). Pairwise distances are between-facets when they are considered between pairs of the form  $(a_j b_k, a_{j'} b_{k'})$ . 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 3.3(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  $(a_j b_k, a_{j'} b_{k'}) : \{a_j : j = 1, 2, \dots, J\}, B = \{b_k : k = 1, 2, \dots, K\}$ . 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:

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

The  $wpd_{raw}$  is then computed as

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

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

### 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:

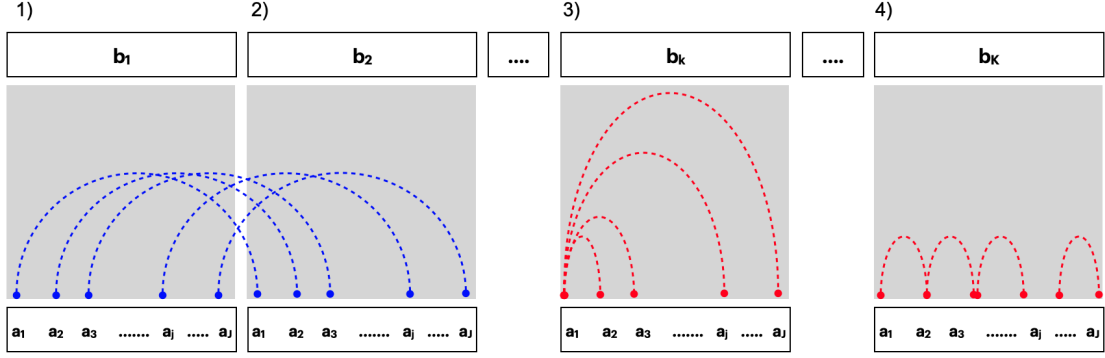
$$wpd_{perm} = \frac{(wpd_{raw} - \overline{wpd_{perm-data}})}{sd(wp_{perm-data})}$$

where  $\overline{wpd_{perm-data}}$  and  $sd(wp_{perm-data})$  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  $\sigma_{perm}^2 = 1$  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  $\sigma_{glm}^2$ , which might not equal 1.

#### Combination approach

The simulation results (in Section 3.4) 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  $wpd_{glm-scaled} = wpd_{glm} * \sigma_{perm}^2 / \sigma_{glm}^2$  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:

$$wpd = \begin{cases} wpd_{perm}, & \text{if } J, K \leq 5 \\ wpd_{glm-scaled} & \text{otherwise} \end{cases} \quad (3.2)$$

### 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<sup>th</sup> 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  $\{(A,B), A = \{a_j : j = 1, 2, \dots, J\}, B = \{b_k : k = 1, 2, \dots, K\}\}$ ,  $\forall (A,B) \in H_{N_C}$ .
  - **Output:** Harmony pairs  $(A,B)$  for which  $wpd$  is significant.
1. Fix harmony pair  $(A,B)$ .
  2. Given the measured variable;  $\{v_t : t = 0, 1, 2, \dots, T-1\}$ ,  $wpd$  is computed and is represented by  $wpd_{obs}^{A,B}$ .
  3. From the original sequence a random permutation is obtained:  $\{v_t^1 : t = 0, 1, 2, \dots, T-1\}$ .

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  $wpd_{sample} = \{wpd_1^{A,B}, wpd_2^{A,B}, \dots, wpd_M^{A,B}\}$ .
7. Repeat Steps (1-6) for all harmony pairs  $(A, B) \in H_{N_C}$  and stored  $wpd_{sample}^{all}$ .
8. 99<sup>th</sup> 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<sup>th</sup> and 90<sup>th</sup> percentile of  $wpd_{sample}^{all}$  respectively. A harmony selected using 99<sup>th</sup>, 95<sup>th</sup> and 90<sup>th</sup> 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 3.3 presents the results of  $wpd$  from the illustrative designs in Section 3.2. The value of  $wpd$  under null design (a) is the least, followed by (b), (c) and (d). This aligns with the principle of  $wpd$ , which is expected to have lowest value for null designs and highest for designs of the form  $D_{var_{all}}$  (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 3.1(a) and 3.1(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, 3.1(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<sup>th</sup> 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.

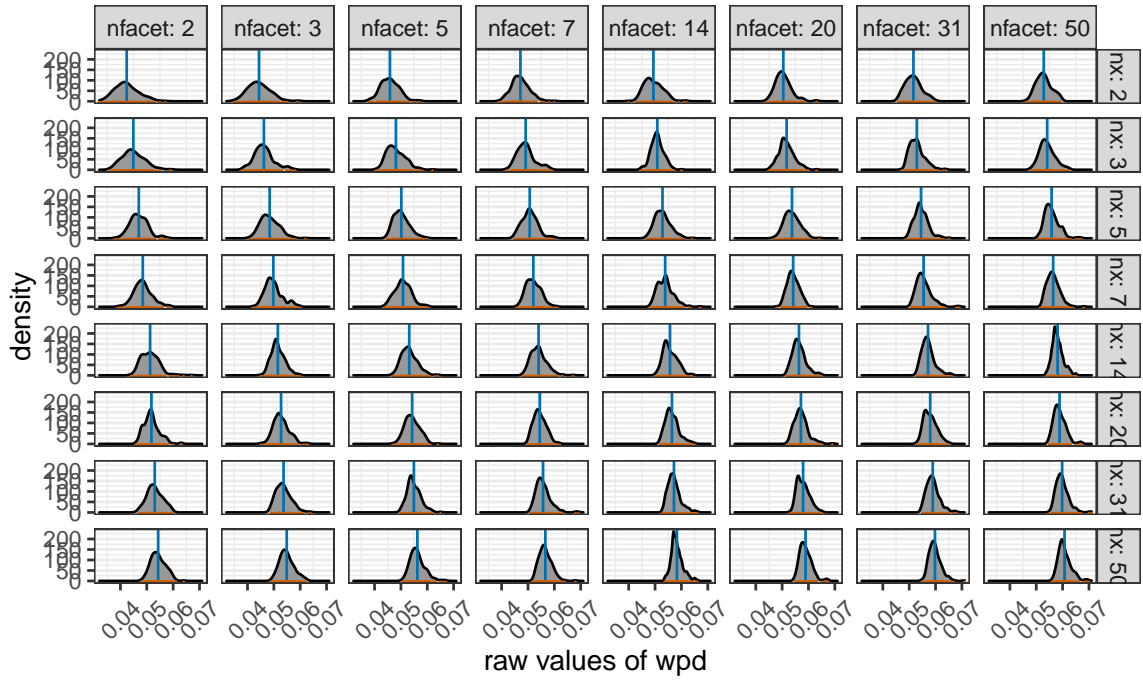
$m = 2$

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  $(2,2), (2,3), (2,5) \dots, (50,31), (50,50)$ . 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 3.5 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 3.2 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.

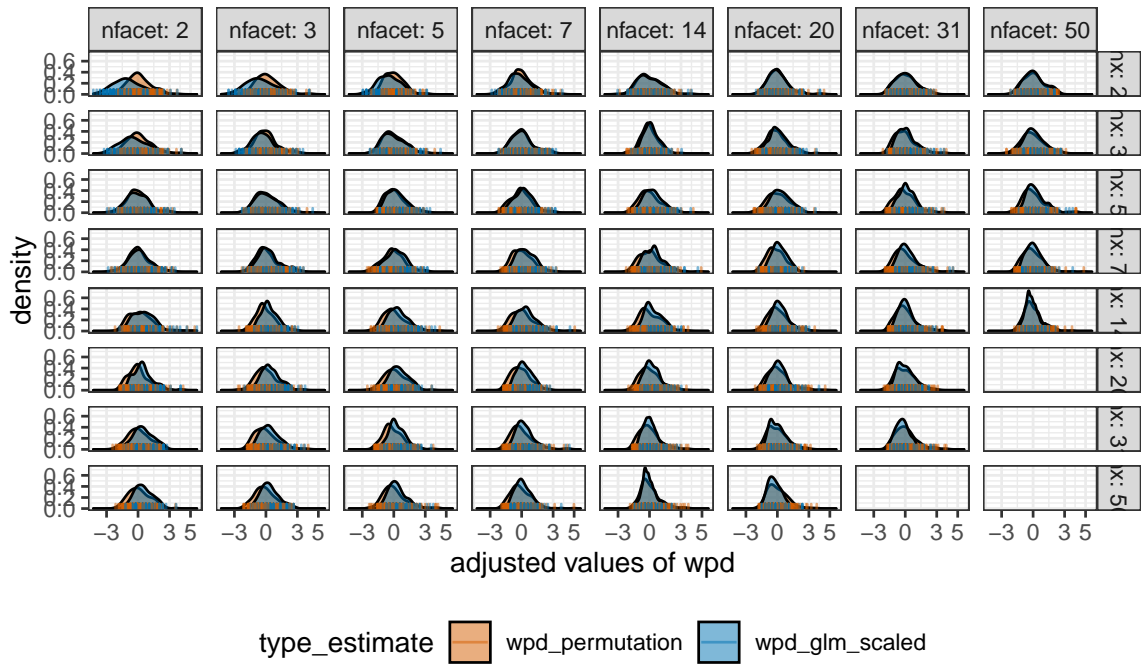
**Table 3.2:** Results of generalised linear model to capture the relationship between  $wpd_{raw}$  and number of comparisons.

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  $wpd_{ls} = 23.4 - 0.96 * \log(nx * nfacet) + e$ . 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 3.6 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 alleviate the problem of computational time in permutation approaches. This is possible as the distribution of the adjusted measure looks similar for both approaches for higher levels.

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,  $(3, 2), (3, 9), (3, 10), \dots, (14, 9), (14, 10)$ . 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  $D_{varyall}$ . 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<sup>th</sup> 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 3.7 (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  $7P_2 = 42$  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  $i \in \{1, 2, \dots, 8\}$ . The harmony pairs are then arranged in descending order and highlighted with \*\*\*, \*\* and \* corresponding to the 99<sup>th</sup>, 95<sup>th</sup> and 90<sup>th</sup> 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*

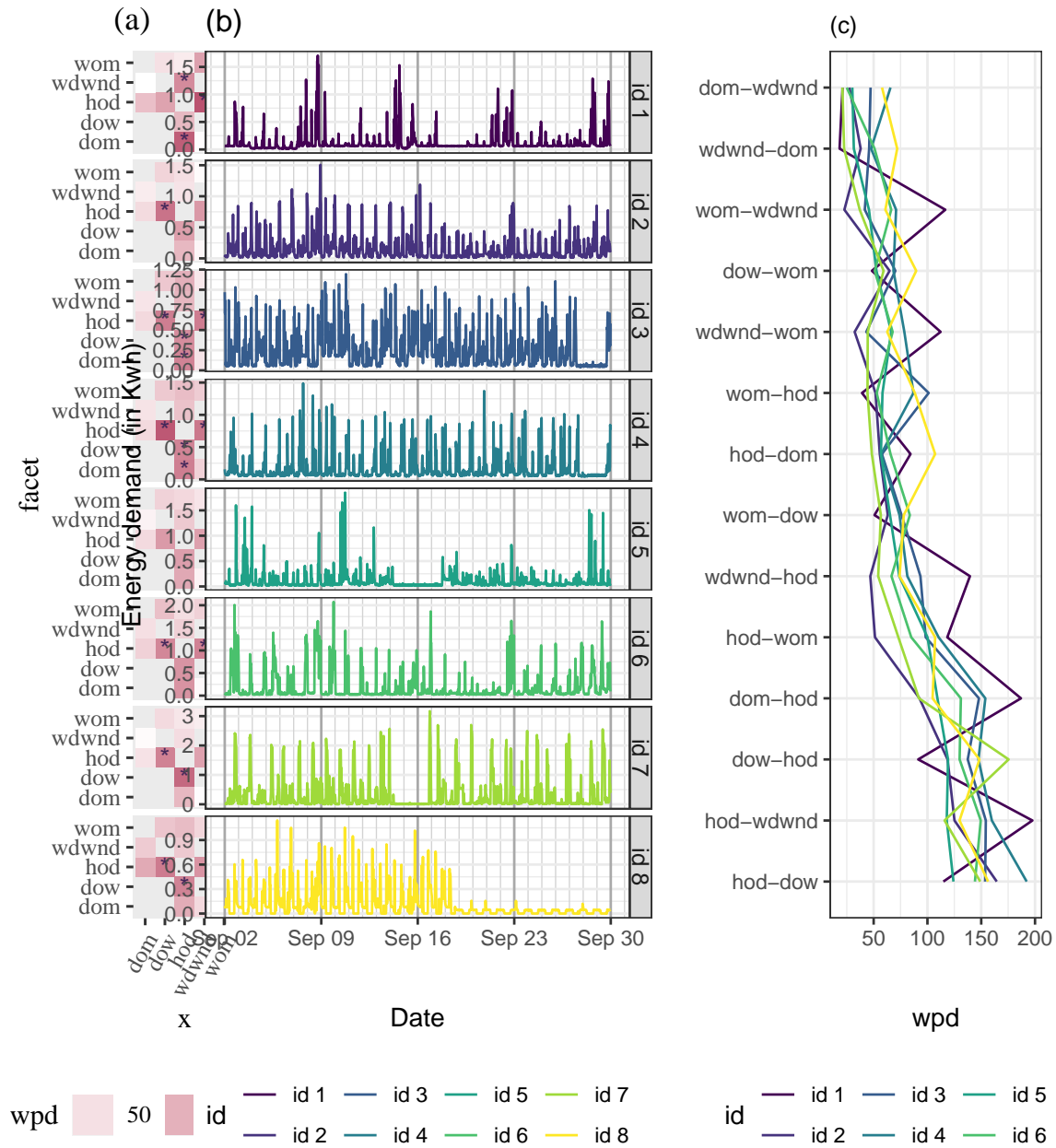
Figure 3.7 helps to compare households through the heatmap (a) across harmony pairs with the cyclic granularity mapped to x-axis and facet being plotted on the x-axis and y-axis of the heatmap. Here *dom*, *dow*, *wdwnd* are abbreviations for day-of-month, day-of-week, weekday/weekend and so on. The 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.

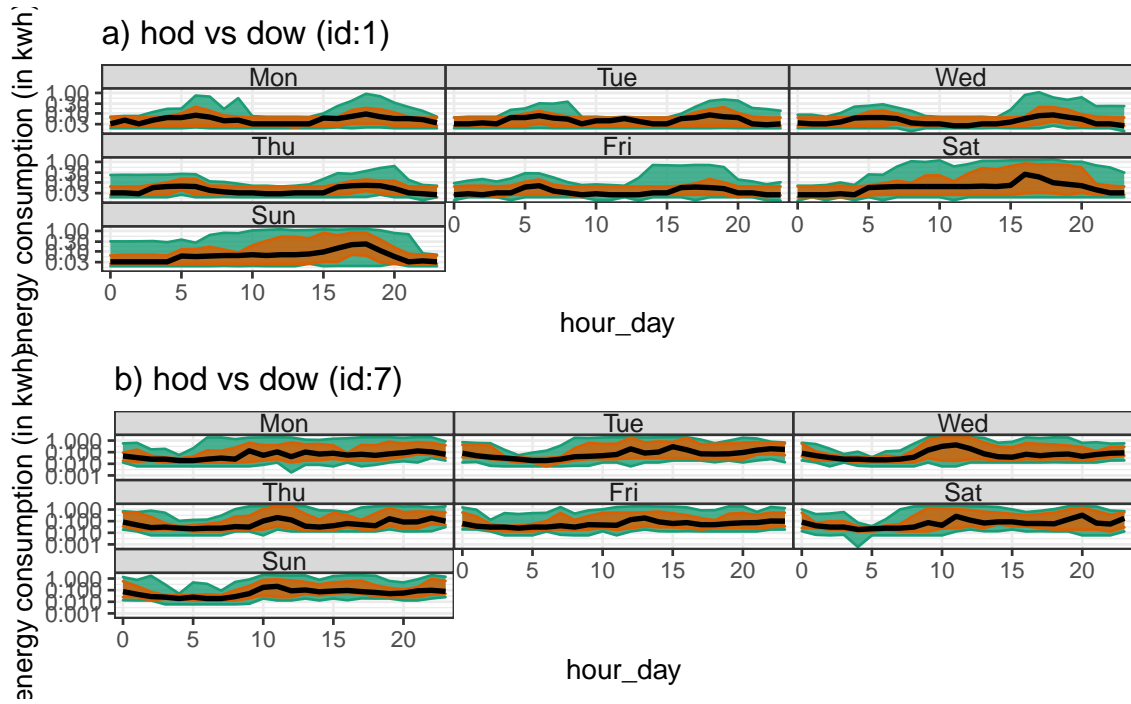
facet variable	x variable	id 1	id 2	id 3	id 4	id 5	id 6	id 7	id 8
hod	wdwnd	1 ***	2 *	1 **	2 **	3	1 **	3	3 *
dom	hod	2 ***	4	3 **	3 **	4	3 *	4	6
wdwnd	hod	3 **	10	7	7	6	8	8	10
hod	wom	4	9	6	5	5	5	5	5
wom	wdwnd	5	14	14	10	12	9	12	13
hod	dow	6	1 ***	2 **	1 ***	1 *	2 **	2 **	1 **
wdwnd	wom	7	12	13	8	7	7	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 3.7(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 3.8 demonstrates that for id 7, the median (black) and quartile 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



**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<sup>th</sup> and 90<sup>th</sup> quantile.

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](https://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**

# **Data representation, visual and analytical techniques for demystifying temporal missing data**



## Chapter 5

# Conclusion and future plans

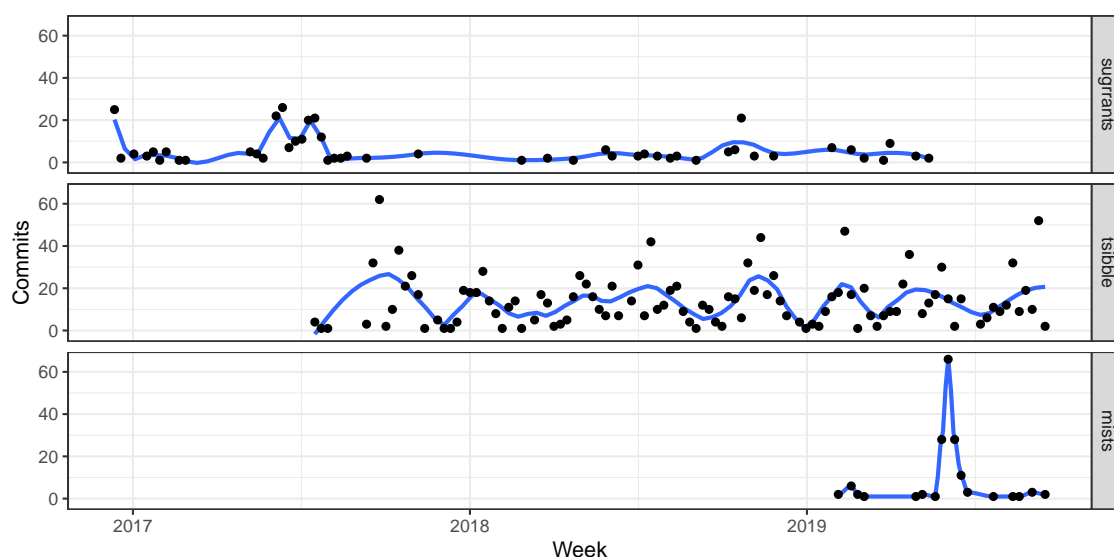
The three papers assembled in this thesis share a common theme of exploratory analysis for temporal data using tidy tools. Chapter ??, “Calendar-based graphics for visualizing people’s daily schedules”, described a new calendar-based display. Chapter ??, “A new tidy data structure to support exploration and modeling of temporal data”, proposed a new temporal data abstraction. Chapter 4, “Data representation, visual and analytical techniques for demystifying temporal missing data”, explored missing data in time. These papers are bundled with software. In this conclusion, I will briefly summarize each package and their impact, and discuss the future directions of my research.

## 5.1 Software development

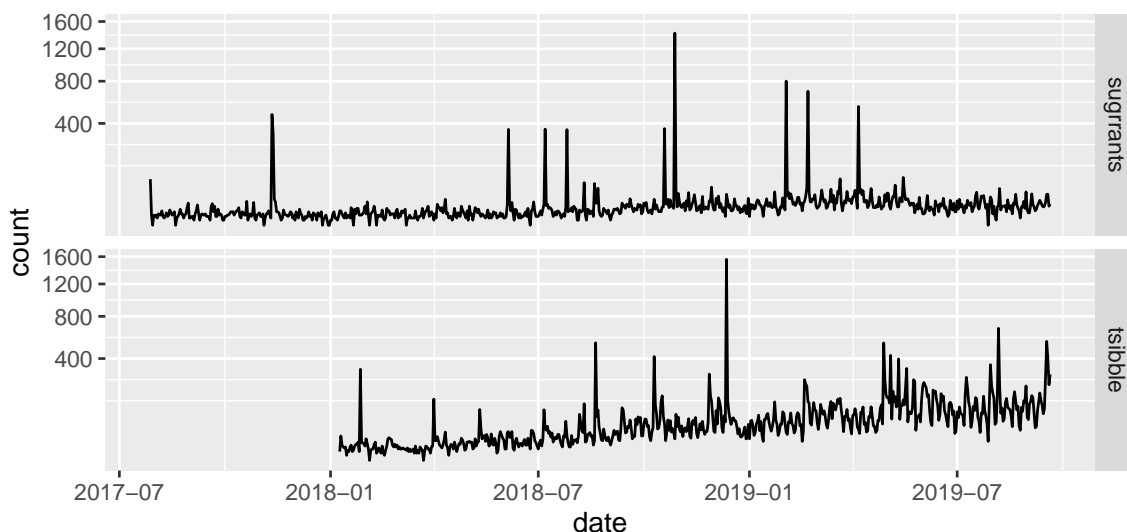
A particular emphasis of this thesis is on translating research methodologies in the form of open source R packages: **sugrrants**, **tsibble**, and **mists**. Figure 5.1 gives an overview of my Git commits to these repositories, and Figure 5.2 shows the daily downloads of the packages from the RStudio mirror (one of 90 CRAN mirrors) since they were available on CRAN.

### 5.1.1 sugrrants

The **sugrrants** package implements the idea of displaying data in the familiar calendar style using `frame_calendar()` and `facet_calendar()`. The research article, a shorter version of Chapter ??, has been awarded the best student paper prize from ASA Sections on Statistical Computing



**Figure 5.1:** Patterns of my package development effort during my PhD years based on Git commits to three repositories, *sugrrants*, *tsibble*, and *mists*. Scatter plots of weekly totals are overlaid with a loess smoother. The *sugrrants* package was the first project with much initial energy, followed by small constant maintenance. The *tsibble* package has been a major project with ongoing constant development and bursts of effort in response to users' feedback. The *mists* package has been a recent intense project.



**Figure 5.2:** Impact of these works (*sugrrants* and *tsibble*) as measured by daily downloads (on square root scale) from the RStudio mirror since they landed on CRAN. The *tsibble* package has an increasing trend, suggesting the steady adoption of the new data structure.

and Statistical Graphics and ACEMS Business Analytics in 2018. There has been a grand total of 15,148 downloads from the RStudio mirror dating from 2017-07-28 to 2019-09-20; and it has been starred 48 times on Github so far. The homepage at <https://pkg.earo.me/sugrrants> contains detailed documentation and a vignette on `frame_calendar()`.

### 5.1.2 tsibble

The **tsibble** package provides a data infrastructure and a domain specific language in R for representing and manipulating tidy temporal data. This package provides the fundamental architecture that other temporal tools will be built upon. For example, a new suite of time series analysis packages, titled “**tidyverts**”, have been developed for the new “tsibble” object. The **tsibble** package has won the 2019 John Chambers Statistical Software Award from the ASA Sections on Statistical Computing and Statistical Graphics. It has been downloaded 41,290 times from the RStudio mirror since it landed on CRAN; and it has received 241 stars on Github. These metrics are the indicators of my research impact, the recognition by professionals, and the uptakes by users. The website (<https://tsibble.tidyverts.org>) includes full documentation and three vignettes about the package usage.

### 5.1.3 mists

The **mists** package aims at exploring missing values for temporal data analytically and graphically. It implements a compact abstraction for efficiently indexing missing data in time, along with numerical and visual methods. It also provides new missing data polishing techniques. The Github repository has received 22 stars, but the package is not on CRAN yet. The documentation site is available at <https://pkg.earo.me/mists>.

## 5.2 Future work

### 5.2.1 Process for generating missing data in time

Missing values in multivariate data are typically characterized by the overall, row-wise, and column-wise numbers of missings. However, none of these captures the dynamics in temporal data. A

well-defined characteristic is need to characterize temporal missingness, and this could possibly shed light on the processes for generating and imputing missing data in time.

Generating temporal missingness can be decomposed into two steps: (1) injecting missings at time points to reflect the functional form of time, and (2) generating the corresponding run lengths to reflect the temporal dependency. I plan to expand on Chapter 4 to generalize missing data generating processes in temporal contexts. Because of the evolving nature of time, the underpinning mechanisms of missing data may change from one period to another. Applying the new characteristic to the data, on a rolling window basis, could indicate the missing data status and thus lead to appropriate missing data remedies.

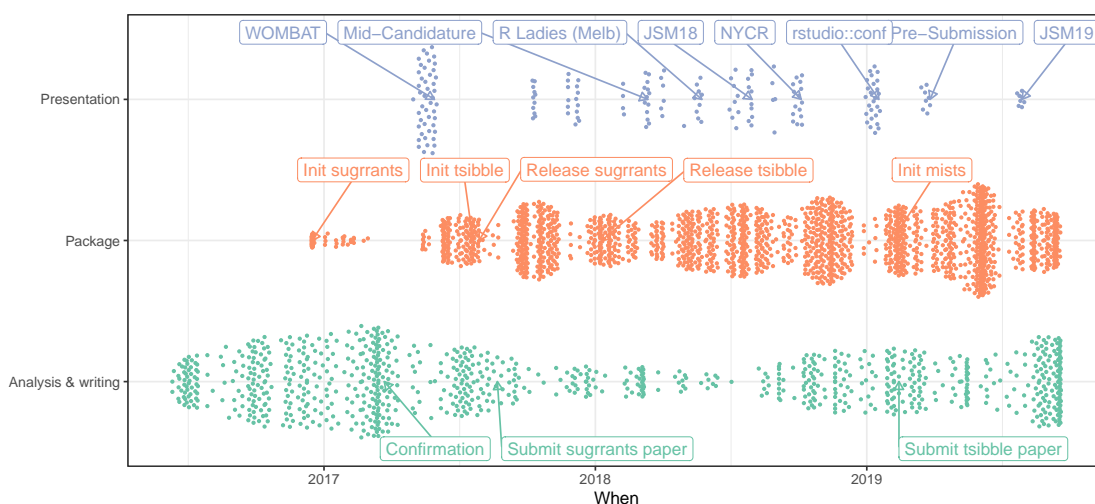
### **5.2.2 Visual methods for temporal data of nesting and crossing interactions**

A collection of time series are often structured in a way that allows nesting and crossing interactions (Hyndman and Athanasopoulos, 2017). For example, a manufacturing company can add up every store's sales by region, by state and by country, which gives a strictly hierarchical time series; alternatively, they can gather the sales based on common attributes such as store, brand, price range and so forth, which leads to a crossed configuration. Nesting is a special case of crossing, with parent-children relations involved. Temporal information such as date-times is often also intrinsically hierarchical, seconds nested within minutes, hours, and etc. The new tsibble structure has the neat capability of supporting these structural embeddings.

Numerous nesting and crossing combinations can yield unwieldy plots, in many of which an abundance of information is possibly buried. Focus-plus-context visualization with interactivity comes to the rescue. Dual contexts, structurally informative subjects, and time provide the source and visual clues for elegant navigation. Interactions on contextual plots control what is to be visualized in the main plots. Many kinds of visual displays can be generated to progressively build a richer data picture through guided or self explorations.

### 5.3 Final words

Presentations, package development, and writing are the three primary types of activities that shape this thesis. I have developed a habit of using Git and Github to track and synchronize my academic work since I started the PhD program. All commits are grouped by the activity types, with annotations of important milestones, shown in Figure 5.3. It has been a fruitful program.



**Figure 5.3:** *Beeswarm plots of my Git commits split by the activity types during my PhD years, labeled with some milestones.*





## **Appendix A**

### **Data dictionary**



# Bibliography

- Department of the Environment and Energy (2018). *Smart-Grid Smart-City Customer Trial Data*. Australian Government, Department of the Environment and Energy. <https://data.gov.au/dataset/4e21dea3-9b87-4610-94c7-15a8a77907ef> (visited on 11/19/2018).
- Hyndman, RJ and G Athanasopoulos (2017). *Forecasting: Principles and Practice*. Melbourne, Australia: OTexts. [OTexts.org/fpp2](https://otexts.org/fpp2).
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. <https://www.R-project.org/>.
- Ushey, K (2019). *renv: Project Environments*. R package version 0.7.0-54. <https://rstudio.github.io/renv>.
- Wickham, H (2014). Tidy Data. *Journal of Statistical Software* **59**(10), 1–23.
- Wickham, H, W Chang, L Henry, TL Pedersen, K Takahashi, C Wilke, and K Woo (2019a). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 3.1.1. <https://CRAN.R-project.org/package=ggplot2>.
- Wickham, H, R François, L Henry, and K Müller (2019b). *dplyr: A Grammar of Data Manipulation*. <http://dplyr.tidyverse.org>, <https://github.com/tidyverse/dplyr>.
- Xie, Y (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. ISBN 978-1138700109. Boca Raton, Florida: Chapman and Hall/CRC. <https://github.com/rstudio/bookdown>.
- Xie, Y, J Allaire, and G Golemund (2018). *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman and Hall/CRC. <https://bookdown.org/yihui/rmarkdown>.