Visualizing probability distributions across bivariate cyclic temporal granularities

Sayani Gupta [*]

Department of Econometrics and Business Statistics, Monash University, Australia

and

Rob J Hyndman

Department of Econometrics and Business Statistics, Monash University, Australia

and

Dianne Cook

Department of Econometrics and Business Statistics, Monash University, Australia

and

Antony Unwin

University of Augsburg, Germany

April 22, 2021

Abstract

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.

---

[*]Email: Sayani.Gupta@monash.edu

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`.

Keywords: data visualization, statistical distributions, time granularities, calendar algebra, periodicities, grammar of graphics, R

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

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

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

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

3

of a quasi-circular granularity is day of the month; examples of aperiodic granularities are public holidays and school holidays.

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

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

This work provides tools for systematically exploring bivariate granularities within the tidy workflow (Grolemund & Wickham (2017)). In particular, we

- provide a formal characterization of cyclic granularities;
- facilitate manipulation of single- and multiple-order-up time granularities through cyclic

4

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 provides some background material on linear granularities and calendar algebra for computing different linear granularities. Section 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 4. Section 5 discusses the role of different factors in constructing an informative and trustworthy visualization. Section 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 7.

2   Linear time granularities

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

2.1   Definitions

Definition 2.1. A **time domain** is a pair $(T; \leq)$ where T is a non-empty set of time instants and $\leq$ is a total order on T.

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

Definition 2.2. The **index set**, $Z = \{z : z \in \mathbb{Z}_{\geq 0}\}$, uniquely maps the time instants to the set of non-negative integers.

Definition 2.3. 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

5

than all elements of $G(j)$; and (2) if $i < k < j$ and $G(i)$ and $G(j)$ are non-empty, then $G(k)$ is non-empty. Each non-empty subset $G(i)$ is called a **granule**.

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

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												0				...		12		...	...			M
year											0								...			Y

Figure 1: *Illustration of time domain, linear granularities and index set. Hour, day, week, month and year are linear granularities and can also be considered to be time domains. These are ordered with ordering guided by integers and hence is unidirectional and non-repeating. Hours could also be considered the index set, and a bottom granularity.*

2.2 Relativities

Properties of pairs of granularities fall into various categories.

Definition 2.4. 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 2.5. 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*.

Definition 2.6. A granularity G is **periodical** 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 $i \in \mathbb{Z}_{\geq \not{\mathrm{K}}}$, if $H(i) = \bigcup_{j \in S} G(j)$ and $H(i+R) \neq \phi$ then $H(i+R) = \bigcup_{j \in S} G(j+P)$.

If G groups into H, it would imply that any granule $H(i)$ is the union of some granules of G, for example, $G(a_1), G(a_2), \ldots, G(a_k)$. Condition (2) in Definition 2.6 implies that if $H(i+R) \neq \emptyset$, then $H(i+R) = \bigcup(G(a_1+P), G(a_2+P), \ldots, G(a_k+P))$, resulting in a "periodic" pattern of the composition of H using granules of G. In this pattern, each granule of H is shifted by P granules of G. P is called the **Period** (Bettini et al. (2000)).

For example, day is periodical with respect to week with $R = 1$ and $P = 7$, while (if we ignore leap years) day is periodical with respect to month with $R = 12$ and $P = 365$ as any month would consist of 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 periodical with respect to other granularities, except for a finite number of periods where they behave in an anomalous way; these are called **quasi-periodic** relationships (Bettini & De Sibi 2000). In a Gregorian calendar with leap years, day groups quasi-periodically into month with the exceptions of the time domain corresponding to 29th February of any year.

Definition 2.7. 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 et al. (2000). Linear time granularities are computed through "calendar algebra" operations (Ning et al. 2002) designed to generate new granularities recursively from the bottom granularity. For example, due to the constant length of day and week, we can derive them from hour using

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

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

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**).

3.1 Circular granularities

Definition 3.1. A **circular granularity** $C_{B,G}$ relates linear granularity G to bottom granularity B if

$$C_{B,G}(z) = z \bmod P(B,G) \quad \forall z \in \mathbb{Z}_{\geq 0} \tag{1}$$

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

Figure 2 illustrates some linear and cyclical granularities. Cyclical granularities are constructed be cutting the linear granularity into pieces, and stacking them to match the cycles (as shown in b). B, G, H (day, week, fortnight, respectively) are linear granularities. The circular granularity $C_{B,G}$ (day-of-week) is constructed from B and G, while circular granularity $C_{B,H}$ (day-of-fortnight) is constructed from B and H. These overlapping cyclical granularities share elements from the linear granularity. Each of $C_{B,G}$ and $C_{B,H}$ consist of repeated patterns $\{0,1,\ldots,6\}$ and $\{0,1,\ldots,13\}$ with $P = 7$ and $P = 14$ respectively.

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

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

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

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

Figure 2: *a. Index sets for some linear and circular granularities. b. Circular granularities can be constructed by slicing the linear granularity into pieces and stacking them.*

where H is periodic with respect to G with regular mapping and period $P(G,H)$. Table 1 shows several circular granularities constructed using minutes as the bottom granularity.

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 shows some examples of quasi-circular granularities.

Definition 3.2. 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

$$Q_{B,G'}(z) = z - \sum_{w=0}^{k-1} |T_{w \bmod R'}|, \quad \text{for} \quad z \in T_k, \tag{2}$$

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.

9

Circular granularity	Expression	Period
minute-of-hour	$C_1 = z \bmod 60$	$P_1 = 60$
minute-of-day	$C_j = z \bmod 60 * 24$	$P_2 = 1440$
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 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.*

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

Table 2: *Examples of quasi-circular granularities relating two linear granularities with irregular mapping leading to several possible period lengths.*

For example, day-of-year is quasi-periodic with either 365 or 366 granules of B (days) within each period 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'.

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 2.6. Aperiodic cyclic granularities capture repetitions of these aperiodic linear granularities. Examples include public holidays which repeat every year, but there is no reasonably small period within which their behavior remains constant. A clas-

sic example is Easter (in the western tradition) whose dates repeat only after 5.7 million years (Reingold & Dershowitz 2018). In Australia, if a standard public holiday falls on a weekend, a substitute public holiday will sometimes be observed on the first non-weekend day (usually Monday) after the weekend. Examples of aperiodic granularity may also include school holidays or a scheduled event. All of these are recurring events, but with non-periodic patterns. Consequently, P_i (as given in Table 2) are essentially infinite for aperiodic granularities.

Definition 3.3. An **aperiodic cyclic granularity** is formed when bottom granularity B groups into an aperiodic linear granularity M such that the granularities are given by

$$A_{B,M}(z) = \begin{cases} i, & \text{for} \quad z \in T_{i_j} \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

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 3. Let the linear granularities M_1 and M_2 denote the in-session semester period and semester break period respectively. Both M_1, M_2 and $M = M_1 \bigcup M_2$ denoting the "semester week type" are aperiodic with respect to days (B) or weeks (G). Hence $A_{B,M}$ denoting day-of-the-"semester week type" would be an aperiodic cyclic granularity, because the placement of the semester within an year would vary across years. Here, $Q_{H,M}$ denoting week-of-the-"semester week type" would be a quasi-circular granularity since the distribution of semester weeks within a semester is assumed to remain constant over years.

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 3.4. The nested ordering of linear granularities can be organized into a **hierarchy table**, denoted as $H_n : (G, C, K)$, which arranges them from lowest to highest in order. It shows how the n granularities relate through K, and how the cyclic granularities, C, can be defined relative to the linear granularities. Let G_ℓ and G_m represent the linear granularity of order ℓ

11

a

days (B)	0	1	...	48	49	...	55	56	...	62	...	91	...	97	98	...	104	105	...	111	...	147	...	153	154	...	160	161	...	167	168	...	174	175	176	.	...	.
weeks (G)	0	0	...	6	7			8			...		13		14			15			...		21		22			23			24				25				
sem wk (H)	-	-	...	-	0			1			...		6		7			8			...		14		15			16			17				18		-	...	-
sem wk type (M)	-	-	...	-	0					1					2				3					4					5							-	...	-
$Q_{H,M}$	-	-	...	-	0		0		...	5			0		0		...	6		0			0		1			2									-	...	-
$A_{B,M}$	0	0	...	0	1	1	1	2	2	2	...	2	2	2	1	1	1	2	2	2	...	2	2	2	1	1	1	2	2	2	2	2	2	2	2	0	...	0

b

semester weeks

| 0 "orientation week" |
| 0 | 1 | 2 | 3 | 4 | 5 | "first half of semester"
| 0 "semester break" |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | "second half of semester"
| 0 "study week" |
| 0 | 1 | 2 | "finals"

Figure 3: *Quasi-circular and aperiodic cyclic granularities illustrated through (a) linear and (b) stacked displays of time. The linear display shows granularities days, weeks, semester weeks, semester week type distributed over linear time. Here a semester lasts for 18 weeks and 2 days, each starting with one week of orientation followed by an in-session period of 6 weeks, a semester-break of 1 week, an in-session period of 7 weeks, a 1-week study break, before final exams which continue for 16 days. This pattern remains same for all semester and hence $Q_{H,M}$ with P = 128 days will be a quasi-circular granularity with repeating patterns. $A_{B,M}$ will be an aperiodic cyclic granularity since the placement of the semester within an year varies across years.*

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

For example, Table 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

Table 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

katun.

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

Definition 3.5. 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 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.

3.5   Computation

Following the calendar algebra of Ning et al. (2002) for linear granularities, we can define cyclic calendar algebra to compute cyclic granularities. Cyclic calendar algebra comprises two kinds of

operations: (1) **single-to-multiple** (the calculation of *multiple-order-up* cyclic granularities from *single-order-up* cyclic granularities) and (2) **multiple-to-single** (the reverse).

Single-to-multiple order-up

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

$$C_{G_\ell,G_m}(z) = \sum_{i=0}^{m-\ell-1} P(\ell,\ell+i)C_{G_{\ell+i},G_{\ell+i+1}}(z), \tag{4}$$

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

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

$$C_{\text{uinal,baktun}}(z) = C_{\text{uinal,tun}}(z) + P(\text{uinal},\text{tun})C_{\text{tun,katun}}(z) + P(\text{uinal},\text{katun})C_{\text{katun,baktun}}(z)$$

$$= C_{\text{uinal,tun}}(z) + 18 \times C_{\text{tun,katun}}(z) + 18 \times 20 \times C_{\text{katun,baktun}}(z)$$

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

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

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

$$C_{G_\ell,G_m}(z) = C_{G_\ell,G_{m'}}(z) + P(\ell,m')C_{G_{m'},G_m}(z) \tag{5}$$

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

$$C_{G_\ell,G_m}(z) = C_{G_\ell,G_{m'}}(z) + \sum_{w=0}^{C_{G_{m'},G_m}(z)-1} (|T_w|) \tag{6}$$

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.

14

Table 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 4) has day-of-month as a single-order-up quasi-circular granularity, with the other granularities being circular. Using Equations (5) and (6), we then have:

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

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

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

Multiple-to-single order-up

Similar to single-to-multiple operations, multiple-to-single operations involve different approaches for all circular single-order-up granularities and a mix of circular and quasi-circular single-order-up granularities in the hierarchy. For a hierarchy table $H_n : (G,C,K)$ with only circular single-order-up granularities and $\ell_1, \ell_2, m_1, m_2 \in 1, 2, \ldots, n$ and $\ell_2 < \ell_1$ and $m_2 > m_1$, multiple-order-up granularities can be obtained using (7).

$$C_{G_{\ell_1}, G_{m_1}}(z) = \lfloor C_{G_{\ell_2}, G_{m_2}}(z)/P(\ell_2, \ell_1) \rfloor \bmod P(\ell_1, m_1) \tag{7}$$

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

Table 5: *The data structure for exploring periodicities in data by including cyclic granularities in the tsibble structure with index, key and measured variables.*

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

Multiple order-up quasi-circular granularities

Single-order-up quasi-circular granularity can be obtained from multiple-order-up quasi-circular granularity and single/multiple-order-up circular granularity using Equations (5) and (6).

4 Data structure

Effective exploration and visualization benefits from well-organized data structures. Wang et al. (2020*b*) 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 5 shows the tsibble structure (index, key, measurements) augmented by columns of cyclic granularities. The total number of cyclic granularities depends on the number of linear granularities considered in the hierarchy table and the presence of any aperiodic cyclic granularities. For example, if we have n periodic linear granularities in the hierarchy table, then $n(n-1)/2$ circular or quasi-circular cyclic granularities can be constructed. Let N_C be the total number of contextual circular, quasi-circular and aperiodic cyclic granularities that can originate from the underlying periodic and aperiodic linear granularities. Simultaneously encoding more than a few of these cyclic granularities when visualizing the data overwhelms human comprehension. Instead, we focus on visualizing the data split by pairs of cyclic granularities (C_i, C_j). Data sets of the form $<C_i, C_j, v>$ then allow

exploration and analysis of the measured variable v.

4.1 Harmonies and clashes

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

Definition 4.1. A **clash** is a pair of cyclic granularities that contains empty combinations of categories.

Definition 4.2. 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 a 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.

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".

5 Visualization

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

5.1 Data summarization

There are several ways to summarize a distribution like estimating empirical distribution or density of the variable's distribution, computing few number summaries or quantiles. This data estimation or summarization could be potentially misleading if they are performed on rarely occurring categories (Section 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.

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 continuous univariate variable, many options are available. Displays based on descriptive statistics include box plots (Tukey 1977) and its variation such as notched box plots (McGill et al. 1978) or other variations as mentioned in (**?**). They also include line or area quantile plots which can display any quantiles and not only quartiles like in a boxplot. Plots based on kernel density estimates include violin plots (Hintze & Nelson 1998), summary plot (Potter et al. 2010), ridge line plots (Wilke 2020), and highest density region (HDR) plots (Hyndman 1996). The less commonly used Letter-Value plots (Hofmann et al. 2017) is a midway between boxplots and density plots. Letter values are order statistics with specific depths. The median (M) 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 specially 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 bin widths respectively. In R package `gravitas` (Gupta et al. 2020), boxplots, violin, ridge, letter-value, line and area quantile plots are implemented, but it is potentially possible to use any plots which can display the distribution of the data.

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 4.1) or near-clashes lead to potentially ineffective graphs. Harmonies tend to be more useful for exploring patterns. Figure 4a shows the distribution of half-hourly electricity consumption through letter value plots across months of the year conditional on quarters of the year. This plot does not work because quarter-of-year clashes with month-of-year, leading to empty subsets. For example, the first quarter never corresponds to December.

Conditioning variable

When C_i is mapped to the x position and C_j to facets, then the A_k levels are juxtaposed and each B_ℓ represent 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_ks are compared against each other within each group. With the mapping of C_i and C_j reversed, the emphasis will shift to comparing B_ℓ levels rather than A_k levels. For example, Figure 4b shows the letter value plot across weekday/weekend partitioned by quarters of the year and Figure 4c shows the same two cyclic granularities with their mapping reversed. Figure 4b helps us to compare weekday and weekend within each quarter and Figure 4c helps to compare quarters within weekend and weekday.

6 Applications

6.1 Smart meter data of Australia

Smart meters provide large quantities of measurements on energy usage for households across Australia. One of the customer trials (Department of the Environment and Energy 2018) conducted as part of the Smart Grid Smart City project in Newcastle and parts of Sydney provides customer level data on energy consumption for every half hour from February 2012 to March 2014. We can use this data set to visualize the distribution of energy consumption across different cyclic granularities in a systematic way to identify different behavioral patterns.
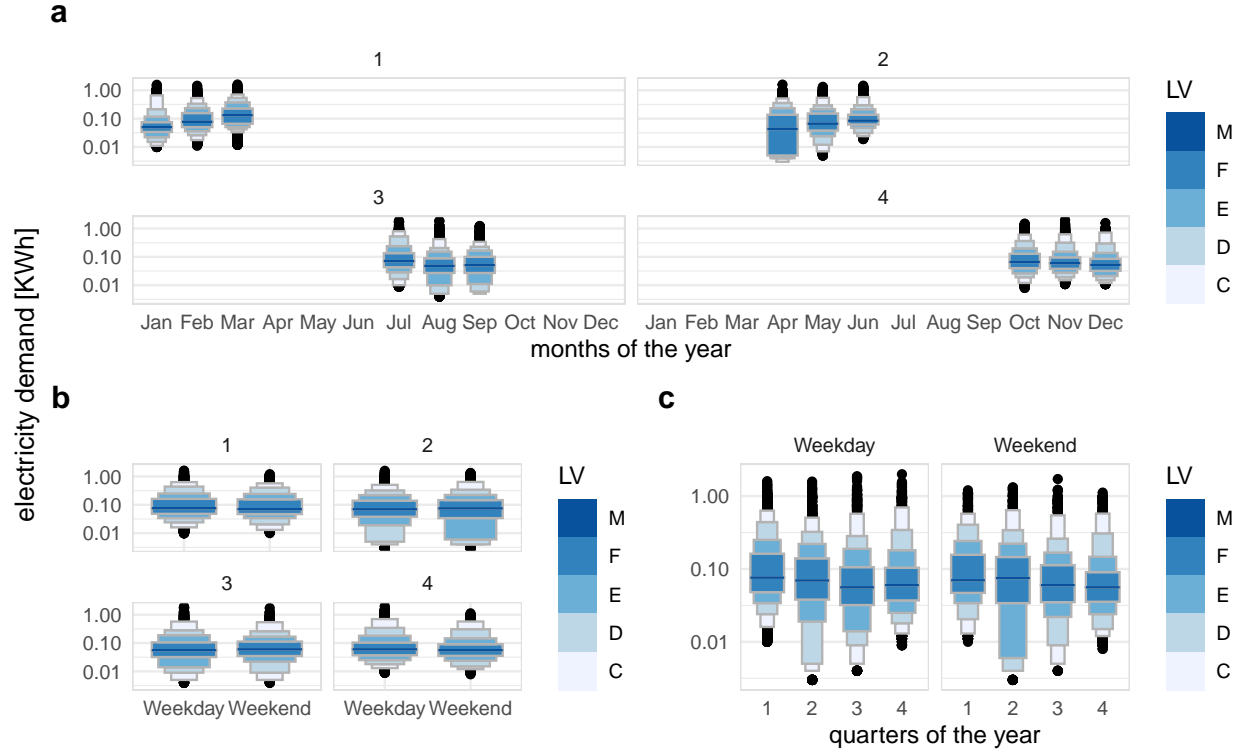
Figure 4: *Distribution of energy consumption displayed through letter value plots: plot a across month-of-year faceted by quarter-of-year; plot b across weekday/weekend faceted by quarter-of-year; plot c across quarter-of-year faceted by weekday/weekend. Plot a shows a clash since there are empty combinations. Plots b and c show harmonies since each quarter includes both weekdays and weekends. It can be seen in plot b that for every quarter (mapped to facet), weekend and weekday consumption are fairly similar except for the second quarter where the letter values below D and E behave differently. This is probably because of lower temperatures in the second quarter compared to the first quarter (summer in Australia). Plot c switches the granularities mapped to the x-axis and the facets and helps to compare quarters within weekdays and weekends.*

Cyclic granularities search and computation

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

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

A few harmony pairs are displayed in Figure 5 to illustrate the impact of different distribution plots and reverse mapping. For each of Figure 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 5b and c respectively. Figure 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 5b, the black line is the median, whereas the purple band covers the 25th to 75th percentile, the orange band covers the 10th to 90th percentile, and the green 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 5c. There is

Table 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

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 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 latter 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.
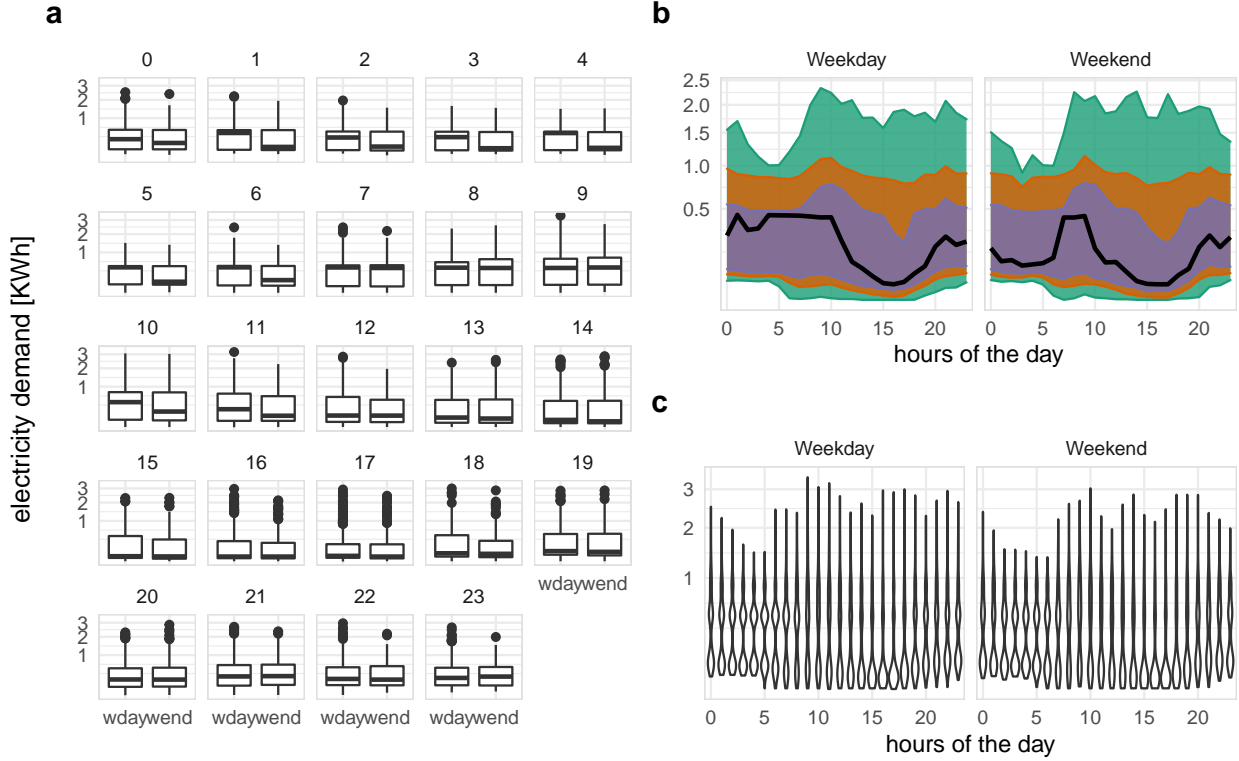
Figure 5: *Energy consumption of a single customer shown with different distribution displays, and granularity arrangements. Two granularities are used: 1. hour of the day; and 2. weekday/weekend. Plot a shows granularity 2 faceted by granularity 1; plots b and c shows the reverse mapping. Plot a allows comparison of usage by workday within each hour of the day using side-by-side boxplots, showing that on work days there is more consumption early in the day. Plots b and c examine the temporal trend of consumption over the course of a day, separately for the type of day. Plot b uses an area quantile to emphasize the time component; e.g., median consumption shows prolonged high usage in the morning on weekdays. Plot c uses a violin plot emphasize subtler distributional differences across hours, showing that morning use on weekdays is bimodal, and on some work days there is low usage possibly indicating the person is working from home or having a late start.*

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

24

with two teams of 11 players each, with each team taking turns batting and playing the field. 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, placed at the end of the cricket pitch behind the batsman. A batsman tries to prevent the bowler from hitting the wicket by hitting the *ball* using a *bat*. Two batsmen of a team are on the pitch at the same time. There are nine *fielders* on the opposing team, with one player out of the 11 acting as the bowler, while another takes up the role of the *wicket-keeper*. To score a *run*, a batsman can run after hitting the ball and changing places on the pitch or also by hitting the ball out of the field on a fly (6 runs) or on a bounce (4 runs). A batsman is *dismissed* when the period of batting is brought to an end either due to good bowling of the bowler or good fielding of the rest of the opposing team.

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 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) such that each over has 6 balls, each innings has 20 overs and each match has 2 innings.

Many interesting questions could be addressed with the `cricket` data set. For example, does

Table 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

the distribution of total runs vary depending on if 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 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 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.

7 Discussion

Exploratory data analysis involve many iterations of finding and summarizing patterns. With temporal data available at ever finer scales, exploring periodicity can become overwhelming with so many possible granularities to explore. This work provides tools to classify and compute possible cyclic granularities from an ordered (usually temporal) index. We also provide a framework to systematically explore the distribution of a univariate variable conditional on two cyclic time granularities using visualizations based on the synergy and levels of the cyclic granularities.

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

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

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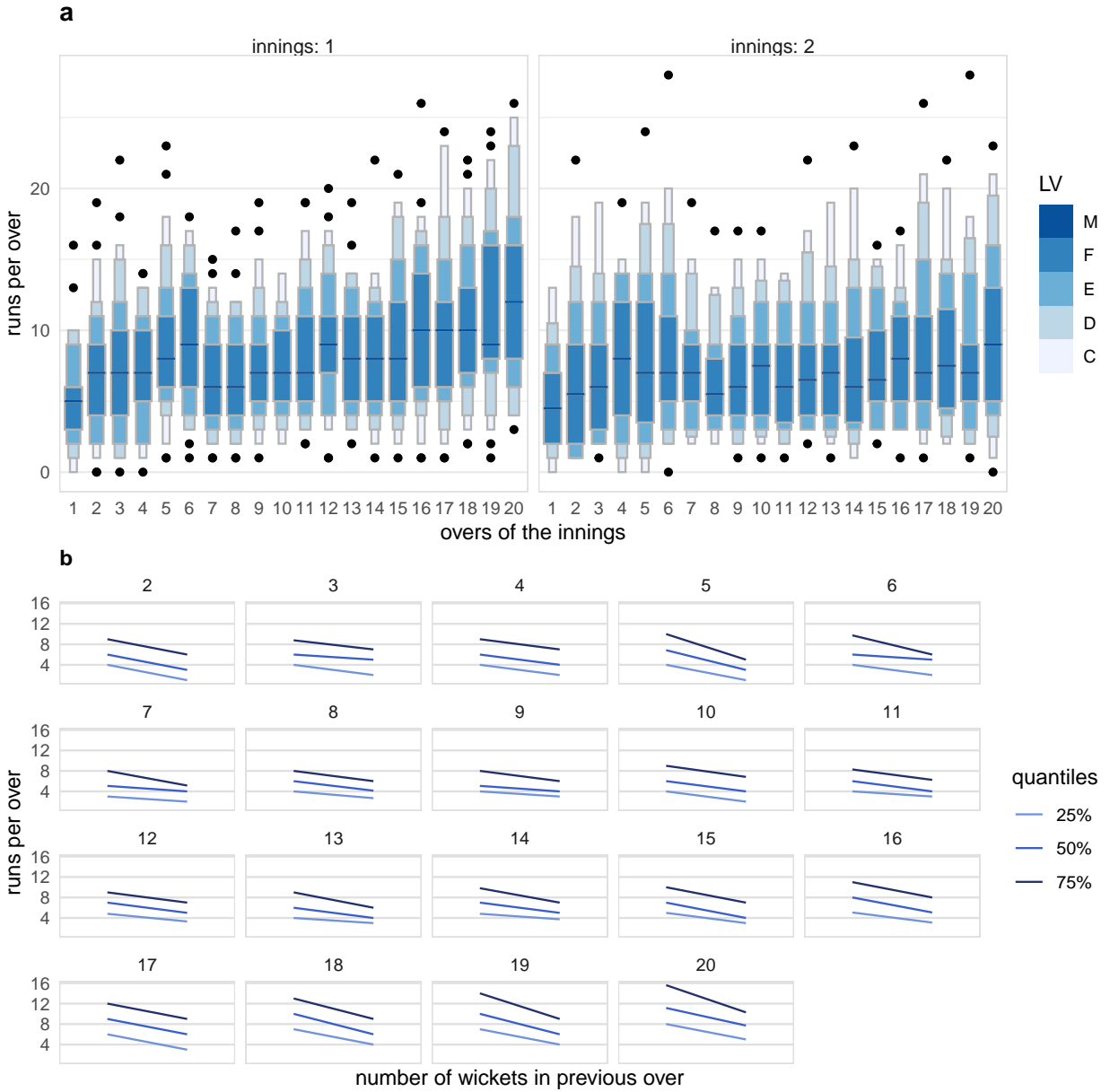
Acknowledgments

Figure 6: *Runs per over shown with different distribution displays, and granularities. Plot a shows letter value plots across overs faceted by innings. For the team batting in the first innings, there is an upward trend of runs per over, while there is no such pattern of runs for the teams batting in the second innings. Plot b shows quantile plots of runs per over across an indicator of wickets in the previous over faceted by current over. This indicates that at least one wicket in the previous over leads to a lower median run rate and quartile spread in the subsequent over, regardless of the over of the innings.*

at sayani07.github.io/gravitas. The Github repository, github.com/Sayani07/paper-gravitas, contains all materials required to reproduce this article and the code is also available online in the supplemental materials. This article was created with `knitr` (Xie 2015, Xie (2020)) and `rmarkdown` (Xie et al. 2018, Allaire et al. (2020)).

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

References

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

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W. & Iannone, R. (2020), *rmarkdown: Dynamic Documents for R*. R package version 2.1.
    URL: *https://github.com/rstudio/rmarkdown*

Bettini, C. & De Sibi, R. (2000), 'Symbolic representation of user-defined time granularities', *Ann. Math. Artif. Intell.* **30**(1), 53–92.

Bettini, C., Dyreson, C. E., Evans, W. S., Snodgrass, R. T. & Wang, X. S. (1998), A glossary of time granularity concepts, *in* O. Etzion, S. Jajodia & S. Sripada, eds, 'Temporal Databases: Research and Practice', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 406–413.

Bettini, C., Jajodia, S. & Wang, S. (2000), *Time Granularities in Databases, Data Mining, and Temporal Reasoning*, Springer Science & Business Media.

Department of the Environment and Energy (2018), *Smart-Grid Smart-City Customer Trial Data*, Australian Government, Department of the Environment and Energy.
URL: *https://data.gov.au/dataset/4e21dea3-9b87-4610-94c7-15a8a77907ef*

Dyreson, C., Evans, W., Lin, H. & Snodgrass, R. (2000), 'Efficiently supporting temporal granularities', *IEEE Transactions on Knowledge and Data Engineering* **12**(4), 568–587.

Goodwin, S. & Dykes, J. (2012), Visualising variations in household energy consumption, *in* '2012 IEEE Conference on Visual Analytics Science and Technology (VAST)', IEEE, Seattle, WA, pp. 217–218.

Grolemund, G. & Wickham, H. (2011), 'Dates and times made easy with lubridate', *Journal of Statistical Software* **40**(3), 1–25.
URL: *http://www.jstatsoft.org/v40/i03/*

Grolemund, G. & Wickham, H. (2017), *R for data science*, O'Reilly Media.

Gupta, S., Hyndman, R., Cook, D. & Unwin, A. (2020), *gravitas: Explore Probability Distributions for Bivariate Temporal Granularities*. R package version 0.1.3.
URL: *https://github.com/Sayani07/gravitas/*

Hintze, J. L. & Nelson, R. D. (1998), 'Violin plots: A box plot-density trace synergism', *American Statistician* **52**(2), 181–184.

Hofmann, H., Wickham, H. & Kafadar, K. (2017), 'Letter-value plots: boxplots for large data', *J. Comput. Graph. Stat.* **26**(3), 469–477.

Hyndman, R. J. (1996), 'Computing and graphing highest density regions', *American Statistician* **50**(2), 120–126.

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

McGill, R., Tukey, J. W. & Larsen, W. A. (1978), 'Variations of box plots', *American Statistician* **32**(1), 12–16.

Ning, P., Wang, X. S. & Jajodia, S. (2002), 'An algebraic representation of calendars', *Ann. Math. Artif. Intell.* **36**(1), 5–38.

Potter, K., Kniss, J., Riesenfeld, R. & Johnson, C. R. (2010), 'Visualizing summary statistics and uncertainty', *Comput. Graph. Forum* **29**(3), 823–832.

R Core Team (2020), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Reingold, E. M. & Dershowitz, N. (2018), *Calendrical Calculations*, 4th edn, Cambridge University Press.

Tukey, J. W. (1977), *Exploratory data analysis*, Addison-Wesley, Reading, Mass.

Vaughan, D. (2020), *almanac: Tools for Working with Recurrence Rules*. R package version 0.1.1.
URL: *https://CRAN.R-project.org/package=almanac*

Wang, E., Cook, D. & Hyndman, R. J. (2020*a*), 'Calendar-based graphics for visualizing people's daily schedules', *Journal of Computational and Graphical Statistics* . to appear.

Wang, E., Cook, D. & Hyndman, R. J. (2020*b*), 'A new tidy data structure to support exploration and modeling of temporal data', *Journal of Computational and Graphical Statistics* . to appear.

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

Wilke, C. O. (2020), *ggridges: Ridgeline Plots in 'ggplot2'*. R package version 0.5.2.
URL: *https://CRAN.R-project.org/package=ggridges*

Wilkinson, L. (1999), *The Grammar of Graphics*, Springer, New York.

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

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

Xie, Y., Allaire, J. & Grolemund, G. (2018), *R Markdown: The Definitive Guide*, Chapman and Hall/CRC, Boca Raton, Florida.
URL: *https://bookdown.org/yihui/rmarkdown*