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

August 28, 2020

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 time granularities respect the linear progression of time such as hours, days, weeks and months. Cyclic time 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. A recommendation of appropriate distribution displays is provided.

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

The methods are implemented in the open source R package `gravitas`, providing functions for creating granularities and exploring the associated time series which are consistent with a tidy workflow (Grolemund & Wickham (2017)), and the probability distributions can be examined using the range of graphics available in `ggplot2` (Wickham 2016).

Keywords: data visualization, statistical distributions, time granularities, calendar algebra, periodicties, 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 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 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 relevant to any temporal data observed more than once per year.

This work provides tools for systematically exploring bivariate granularities within the tidy workflow. 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;

- recommend prospective probability distributions for exploring distributions of a univariate dependent variable across pairs of 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 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. *The **index set**, $Z = \{z : z \in \mathbb{Z}_{\geq 0}\}$, uniquely maps the time instants to the set of non-negative integers.*

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

This implies that the granules in a linear granularity are non-overlapping, continuous and ordered. The indexing for each granule can also be associated with a textual representation, called the label. A discrete time model often uses a fixed smallest linear granularity named by Bettini et al. (1998) **bottom granularity**. Figure 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 $\{0,1,\dots,t\}$, we will have “day” running from $\{0,1,\dots,\lfloor t/24 \rfloor\}$. These linear granularities are uni-directional and non-repeating.

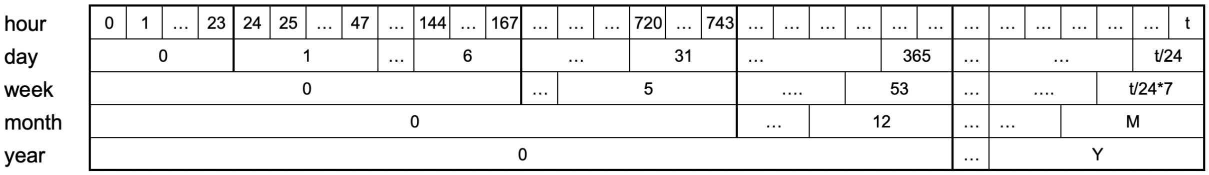

hour
0 | 1 | ... | 23 | 24 | 25 | ... | 47 | ... | 144 | ... | 167 | ... | ... | ... | 720 | ... | 743 | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | t
day
0 | 1 | ... | 6 | ... | 31 | ... | 365 | ... | ... | t/24
week
0 | ... | 5 | .... | 53 | ... | .... | t/24*7
month
0 | ... | 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 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*. The relationship has period 7.

The relationship $day \trianglelefteq month$ has a more complicated period. If leap years are ignored, each month is a grouping of the same number of days over years, hence the period of the grouping $(day, month)$ is one year. With the inclusion of leap years, the grouping period is 400 years.

Definition 6. *A granularity G is* **periodical** *with respect to a 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 $i \in \mathbb{Z}$, if $H(i) = \bigcup_{j \in S} G(j)$ and $H(i+R) \neq \phi$ then $H(i+R) = \bigcup_{j \in S} G(j+P)$.*

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 $P = 365$.

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 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 day, then granularity week will have lower order than month since each week consist of fewer days than each month.

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

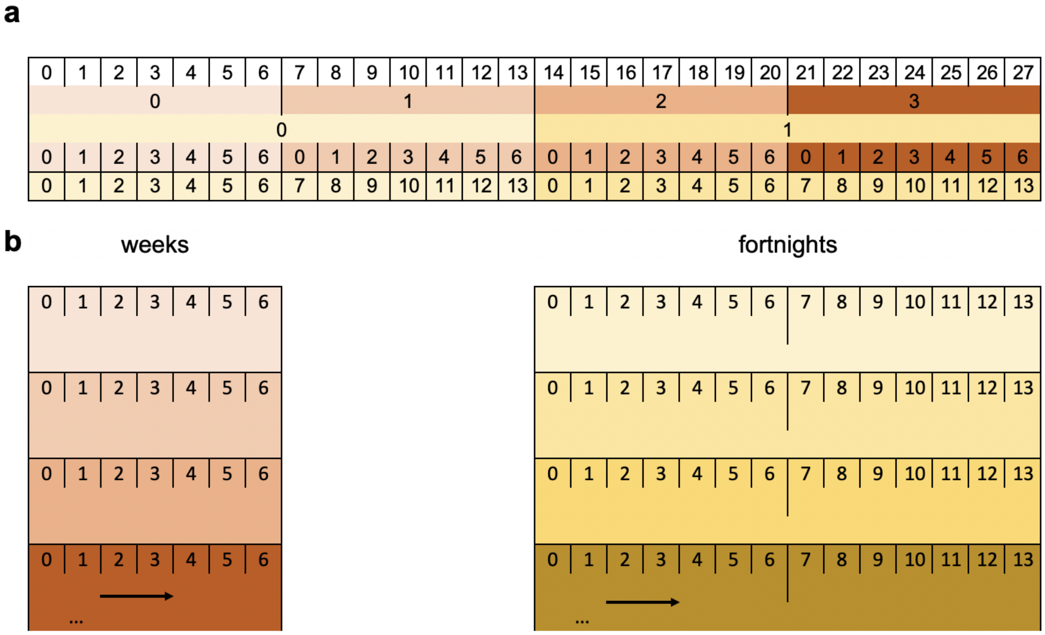
Cyclic granularities represent cyclical repetitions in time. They can be thought of as additional categorizations of time that are not linear. Cyclic granularities can be constructed from two linear granularities, that relate periodically; the resulting cycles can be either *regular* (**circular**),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

3.1 Circular

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

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

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



days (B)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
weeks (G)	0							1							2							3						
fortnights (H)	0														1													
$C_{B,G}$	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
$C_{B,H}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	0	1	2	3	4	5	6	7	8	9	10	11	12	13

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.*

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 $\{0,1,\dots,6\}$ 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

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

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 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	$C_j = z \bmod 60 * 24$	$P_2 = 1440$
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.*

3.2 Quasi-circular

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.

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	$P_3 = 366, 365$
$Q_4 =$ week-of-month	$P_4 = 5, 4$

Table 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*** $Q_{B,G'}$ *relates linear granularity* G' *and bottom granularity* B *if*

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

where z *denotes the index set,* B *groups periodically into* G' *with irregular mapping,* R' *is the number of granules of* G' *in each repetition of the grouping,* 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*.*

This is not a condition. What is Q used for?

For example, suppose consecutive granules of G' are alternately made up of 7 and 5 granules of B within each period, so that $R' = 2$ Then $Q_{B,G'}$ 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'. Here, $T_0 = \{0, 1, \dots, 6\}$ and $T_1 = \{7, \dots 11\}$. Hence using Definition 9 we will have:

$$Q_{B,H'}(10) = 10 - \sum_{w=0}^{1-1} |T_{w \bmod 2}| = 10 - 7 = 3$$

since $10 \in T_1$.

If linear granularity G' is periodical with respect to B with irregular mapping, then there exist R', $P' \in \mathbb{Z}_+$ such that if $G'(w) = \bigcup_{z \in T_w} B(z)$ then

$$G'(w) = \bigcup_{z \in T_{w \bmod R'}} B(P' * \lfloor w/R' \rfloor + z)$$

(from Definition 6). Here w mod R' represents the index that must be shifted to obtain $G'(w)$. The idea here is if we know the composition of each of the granules of G' in terms of granules of B for one period, we can find the composition of any granule of G' beyond a period since the "pattern" repeats itself along the time domain due to the periodic property. The periodic property also ensures that $|T_w| = |T_{w \bmod R'}|$ since every w^{th} and $(w+R')^{\text{th}}$ granule of G' will have the same number of granules of B. The term $\sum_{w=0}^{k-1} |T_w|$ in Definition 9 denotes the number of granules of B till the $(k-1)^{\text{th}}$ granule of G'. Since $|T_w| = |T_{w \bmod R'}|$, the number of granules of B till the $(k-1)^{\text{th}}$ granule of G' becomes $\sum_{w=0}^{k-1} |T_{w \bmod R'}|$ in Definition 9.

3.3 Aperiodic

Aperiodic time granularities are those that can not be specified as a periodic repetition of a pattern of granules. Most public holidays repeat every year, but there is no reasonably small period within which their behavior remains constant. A classic example can be that of Easter, whose dates repeat only after 5,700,000 years. 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. As such, plausible P_i from Table 2 could be possibly infinite for aperiodic granularities.

Aperiodic cyclic granularities are defined using aperiodic linear granularities. Consider n aperiodic linear granularities M_i, $i \in \{1, 2, \dots, n\}$, such that Definition 6 does not hold true with respect to B. However, $B \trianglelefteq M_i$. Then according to Definition 5, for each index j there exists a (possibly infinite) subset $T_{\{i_j\}}$ of the integers such that $M_i(j) = \bigcup_{z \in T_{i_j}} B(z)$. Suppose $M = \bigcup_{i=1}^{n} M_i$ is formed by collecting the granules of $\{M_1, M_2, \dots, M_n\}$. Here, index $\{i_j\}$ stands for the j^{th} granule of the i^{th} linear aperiodic granularity.

Definition 10. *An* ***aperiodic cyclic granularity*** $A_{B,M}$ *relates a linear granularity* M *to the bottom granularity* B*, if*

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

where, $z \in \mathbb{Z}_{\geq 0}$ *denotes the index set,* B *denotes a full-integer labelled bottom granularity which groups into linear granularity* M *but not periodically,* 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* $M = \bigcup_{i=1}^{n} M_i$.

This is not a condition. What is A used for?

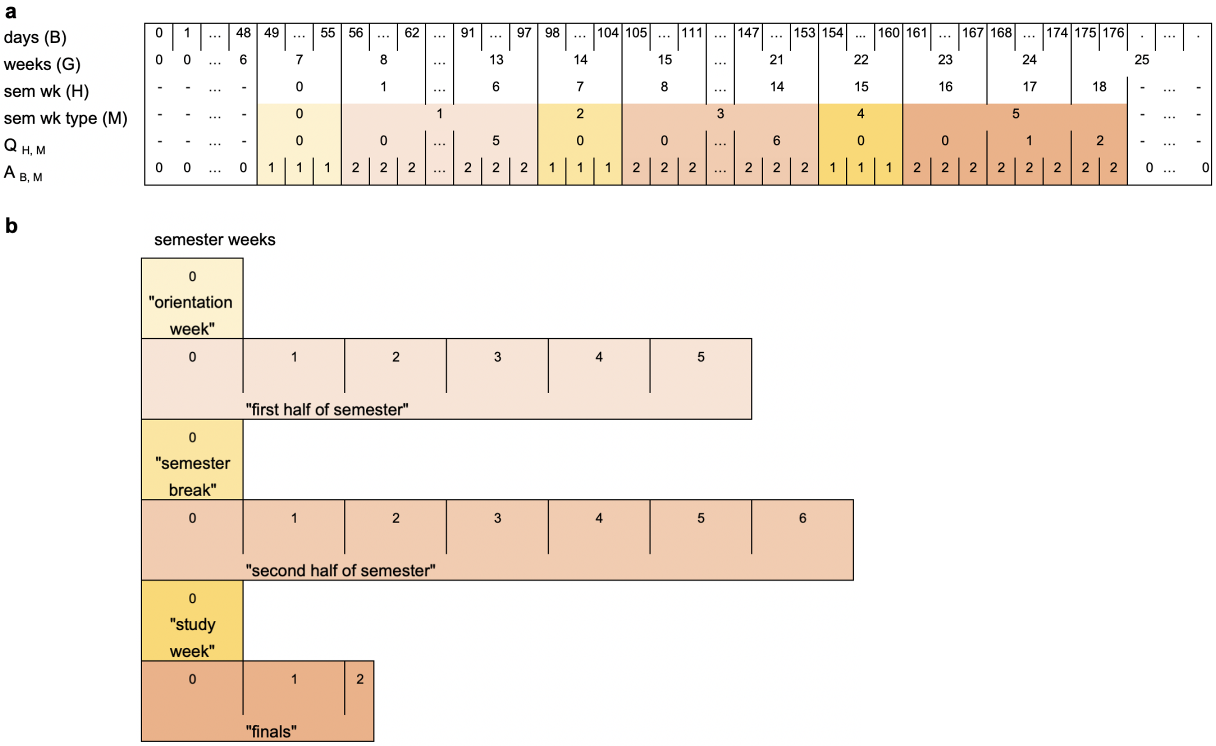


Figure 3: *Quasi-circular and aperiodic cyclic granularities illustrated through linear (a) and stack (b) display of time. The linear display shows linear granularities days, weeks, semester weeks, semester week type distributed linearly from past to future. Here a semester lasts for 18.3 weeks, each starting with one week of orientation followed by an in-session period of 6 weeks, a semester-break of 1 week and again an in-session period of 7 weeks and 1 week study break before final exams which continue for 1.3 weeks. This pattern remains same for all semester and hence $Q_{H,M}$ with P = 18.3 weeks 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.*

Consider a school semester which always lasts for 18 weeks and 2 days, starting with one week of orientation followed by an in-session period of 6 weeks, a semester-break of 1 week and again an in-session period of 7 weeks and 1 week study break before final exams which continue for 16 days. Let the linear granularities M_1 and M_2 denote in-session semester period and semester break periods 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. This is because the placement of the semester within an year would vary across years. Figure 3 b shows the stack display of time with granules representing same categories (semester break/in-session) stacked on top of one another. Figure 3 (a) representing the linear display of time is useful in manifesting how the categories would be determined with respect to the bottom granularities using Definition 10. Here, $Q_{H,M}$ denoting semester 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. This is illustrated using two examples, a Mayan calendar, which is simple, in addition to the more complex Gregorian calendar.

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

For example, Table 3 shows the hierarchy table for the Mayan calendar. In the Mayan calendar, one day is referred to as a kin and the calendar was structured such that 1 kin = 1 day; 1 uinal = 20 kin; 1 tun = 18 uinal (about a year); 1 katun = 20 tun (20 years) and 1 baktun = 20 katun.

Like most calendars, the Mayan calender used the day as the basic unit of time (Reingold & Dershowitz 2001). 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

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

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 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).

3.5.1 Single-to-multiple

Methods to obtain multiple-order-up granularity will be different based on if the hierarchy consists of all circular single-order-up granularities or a mix of circular and quasi-circular single-order-up granularities. The methods require the use of integer arithmetic, and hence it is important that the label mapping of the individual single-order-up granularity is an identity function, that is, $L(x) = x \quad \forall x$. The label mapping of the resultant multiple-order-up granularity can be chosen depending on the context. Circular single-order-up granularities can be used recursively to obtain a multiple-order-up circular granularity using Equation 4 for $\ell < m-1$ and $P(i,j) = 1, \quad \forall i = j \in \{0,1,\dots,m-\ell-1\}$.

$$\begin{aligned}
C_{G_\ell,G_m}(z) &= C_{G_\ell,G_{\ell+1}}(z) + P(\ell,\ell+1)C_{G_{\ell+1},G_m}(z) \\
&= C_{G_\ell,G_{\ell+1}}(z) + P(\ell,\ell+1)(C_{G_{\ell+1},G_{\ell+2}}(z) + P(\ell+1,\ell+2)C_{G_{\ell+2},G_m}(z)) \\
&= C_{G_\ell,G_{\ell+1}}(z) + P(\ell,\ell+1)C_{G_{\ell+1},G_{\ell+2}}(z) + P(\ell,\ell+1)P(\ell+1,\ell+2)C_{G_{\ell+2},G_m} \\
&= C_{G_\ell,G_{\ell+1}}(z) + P(\ell,\ell+1)C_{G_{\ell+1},G_{\ell+2}}(z) + P(\ell,\ell+2)C_{G_{\ell+2},G_{\ell+m}}(z) \\
&\vdots \\
&= \sum_{i=0}^{m-\ell-1} P(\ell,\ell+i)C_{G_{\ell+i},G_{\ell+i+1}}(z)
\end{aligned} \tag{4}$$

For example, the multiple-order-up granularity $C_{uinal,katun}$ for Mayan calendar could be obtained using Equation 5.

$$\begin{aligned}
C_{uinal,baktun}(z) =& C_{uinal,tun}(z) + P(uinal,tun)C_{tun,katun}(z) + P(uinal,katun)C_{katun,baktun}(z) \\
=& \lfloor z/20 \rfloor \bmod 18 + 18 * \lfloor z/20*18 \rfloor \bmod 20 \\
&+ 18*20\lfloor z/20*18*20 \rfloor \bmod 20
\end{aligned} \tag{5}$$

We consider the case of only 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 dicrete combination of circular and quasi-circular granularities.

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

- $C_{\ell,m'}(z)$ is circular and $C_{m',m}(z)$ is quasi-circular

$$C_{G_\ell,G_m}(z) = C_{G_\ell,G_{m'}}(z) + P(\ell,m')C_{G_{m'},G_m}(z) \tag{6}$$

- $C_{\ell,m'}(z)$ is quasi-circular and $C_{m',m}(z)$ is circular

$$C_{G_\ell,G_m}(z) = C_{G_\ell,G_{m'}}(z) + \sum_{w=0}^{C_{m',m}(z)-1} (|T_w|) \tag{7}$$

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 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

Consider a hierarchy using linear granularties from a Gregorian calendar in Table 4. This is an example of a hierarchy structure which has both circular and quasi-circular single-order-up granularities with day-of-month as the only single-order-up quasi-circular granularity.

Using Equations 6 and 7, 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 $month(w) = \bigcup_{z \in T_w} day(z)$.

3.5.2 Multiple-to-single

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 in the hierarchy 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 could be obtained using 8.

$$C_{G_{\ell_1}, G_{m_1}}(z) = \lfloor C_{G_{\ell_2}, G_{m_2}}(z) / P(\ell_2, \ell_1) \rfloor \bmod P(\ell_1, m_1) \tag{8}$$

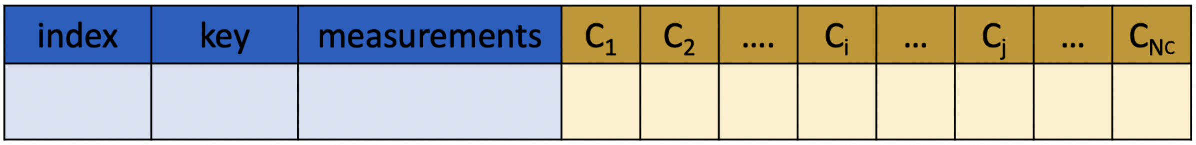
For example, in Mayan Calendar, it is possible to compute the single-order-up granularity tun-of-katun from uinal-of-baktun, since $C_{tun,katun}(z) = \lfloor C_{uinal,baktun}(z)/18 \rfloor$ mod 20.

Multiple order-up quasi-circular granularities Single-order-up quasi-circular granularity could be obtained from multiple-order-up quasi-circular granularity and single/multiple-order-up circular granularity using Equations 6 and 7.

4 Data structure

Effective exploration and visualization benefits from well-organized data structures. Wang et al. (2020*b*) introduced a tidy data structure, tsibble, to support exploration and modeling of temporal data. This forms the basis of the structure for cyclic granularities. A tsibble consists of an index, key 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, and hence, we introduce the data structure in Figure 4. This basic structure of a tsibble (index, key, measurements) is augmented by columns of cyclic granularities. The total number of cyclic granularities would be based on the number of linear granularities considered in the hierarchy table and presence of any aperiodic cyclic granularities, for example, if we have n periodic linear granularities in the hierarchy table, $n(n-1)/2$ circular or quasi-circular cyclic granularities could be constructed. Let N_C be the total number of contextual circular, quasi-circular and aperiodic cyclic granularities that could originate from N_L periodic and aperiodic linear granularities. Any attempt to encode all or many of these cyclic granularities at the same time to develop insights on periodicity might fail or otherwise become too numerous for comprehensive human consumption. Instead, this big problem is broken down by focusing on pair of cyclic granularities (C_i, C_j) at a time for $i, j \in \{1, 2, \ldots, N_C\}$. Data sets of the form <C_i, C_j, v> then forms the basis for exploration and analysis of the measured variable v.



index	key	measurements	C_1	C_2	....	C_i	...	C_j	...	C_{Nc}

Figure 4: *The data structure for exploring periodicities in data by including cyclic granularities in the tsibble structure with index, key and measured variables.*

4.1 Harmonies and clashes

The way cyclic granularities relate become important when we consider the data structure in Figure 4. Let us consider two cyclic granularities C_i and C_j, such that C_i maps index set to a set $\{A_k \mid k \in \mathbb{N}, k \leq K\}$ and C_j maps index set to a set $\{B_\ell \mid \ell \in \mathbb{N}, \ell \leq L\}$. Here, A_k's or B_ℓ's 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 $C_j(s) = B_\ell$. Data subsets for each combination of levels (A_k, B_ℓ) like <A_k, B_ℓ, $v(s)$> can be obtained for all $k \in 1,2,\ldots,K$ and $\ell \in 1,2,\ldots,L$ which will lead to KL data subsets. Now, some situations can lead to few or many of these sets being empty. We will discuss few cases, where one or more of these KL sets will be empty either due to the structure of the calendar, duration and location of events in a calendar or just by the construction of the cyclic granularities.

Definition 13. *A **clash** is a pair of cyclic granularities which contains structurally, event-driven or build-based empty combinations of its categories.*

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

First, empty combinations can arise due to the structure of the calendar or hierarchy. These are called "structurally" empty combinations. Let us take a specific example, where C_i be day-of-month with 31 levels and C_j be week-of-month with 5 levels. There will be $31 \times 5 = 155$ sets $S_{k\ell}$ corresponding to possible combinations of C_i and C_j. Many of these like $S_{1,5}$, $S_{21,2}$ are empty. This is also intuitive since the first day of the month can never correspond to fifth week of the month. Hence the pair (day-of-month, week-of-month) is a clash.

Second, empty combinations can turn up due to differences in event location or duration in a calendar. These are called "event-driven" empty combinations. Let us consider C_i be day-of-week with 7 levels and C_j be WorkingDay/NonWorkingDay 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.

Third, empty combinations can be a result of how granularities are constructed. These are called "build-based" empty combinations. Let C_i be Business-days, which are days from Monday to Friday except holidays and C_j be day-of-month. Then the days denoting weekends in a month would not correspond to any Business days. This is different from structurally empty combinations because structure of the calendar does not lead to these missing combinations, but the construction of the granularities does.

An example when there will be no empty combinations could be where C_i and C_j maps index set to day-of-week and month-of-year respectively. Here C_i can have 7 levels while C_j can have 12 levels. So there are $12 \times 7 = 84$ 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 Summarizing the measured variable

Restructuring time from linear to cyclic time granularities leads to re-organisation of the data structure, where each level of a cyclic granularity corresponds to multiple values of the measured variable. It is common to see summarization of these multiple values through aggregation or an unique summary statistic like mean or median to have each level of cyclic granularity correspond to an unique value of the measured variable. However, this approach hides the distributions of the measured variable induced by the re-organised data structure. Summarizing the distribution of the measured variable using these multiple observations could be a potential way to explore and bring forward different features of the data.

However, we need to consider the effect of number of observations on the summarization even for harmonies. Consider a data set with $(T+1)$ observations, and two cyclic granularities C_i and C_j with K and L categories respectively. Let *nobs* be the number of observations for a combination of levels from (C_i, C_j). Now, *nobs* might be equal for all combinations on an average so that $nobs = (T+1)/(KL)$ as $T \to \infty$ or ideally take any value between $\{0, 1, \dots, T\}$.

The statistical transformation used for summarizing the measured variable should be chosen in line with *nobs*. For example, it might be useful to compute deciles for summarization only when $nobs \geq 10$. Rarely occurring categories such as the 366th day of the year, or the 31st day of the month are likely to suffer from this problem.

5 Visualization

The grammar of graphics introduced a framework to construct statistical graphics by relating data space to the graphic space (Wilkinson 1999). The layered grammar of graphics proposed by Wickham (2016), which is an alternate and modified parametrization of the grammar suggests that graphics are made up of distinct layers of grammatical elements. Drawing from the grammar of graphics, if $<C_i, C_j, v>$ serves as the basis of visualizing the distribution of the measured variable, the following layers can be specified:

- Data: $<C_i, C_j, v>$
- Aesthetic mapping (mapping of variables to elements of the plot): C_i mapped to x position and v to y position
- Statistical transformation (data summarization): any descriptive or smoothing statistics that summarizes distribution of v
- Geometric objects (physical representation of the data): any geometry displaying distribution, for example, boxplot, letter value, violin, ridge or highest density region plots
- Facet (split plots): C_j

5.1 Choice of statistical transformations and geometric objects

Choice of plots are dictated by the statistical transformations and geometric objects used for the visualization. The basic plot choice for our data structure is the one that can display distributions. We will discuss few conventional and recent ways to plot distributions using both Kernel density estimates and descriptive statistics. Descriptive statistics based displays include box plots (Tukey 1977) or different variations of it like notched box plots (Mcgill et al. 1978). More recent ways are the letter-value box plot (Hofmann et al. 2017) or quantile plots which display quantiles instead of quartiles in a traditional boxplot. Kernel density based plots for displaying a distribution include violin plots (Hintze & Nelson 1998), summary plot (Potter et al. 2010), ridge line plots, highest density regions (HDR) box (Hyndman 1996). Each type of density display has different parameters, that need to be estimated given the data. Each is equipped with some benefits and challenges which should be borne in mind while using them for exploration.

5.2 Facet and aesthetic variables

5.2.1 Levels

The levels of cyclic granularities has a role to play on the choice of plots since space and resolution might become a problem with too many levels. A potential approach could be to categorize the levels as very high/high/medium/low for each cyclic granularity and define some criteria based on usual cognitive power, display size available 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.

5.2.2 Synergy of cyclic granularities

For data sets of the form $(C_i, C_j, v)\ \forall i, j \in N_C$, if there are levels of C_i (or C_j) not spanned by levels of C_j (or C_i), empty sets are formed leading to potential ineffective graphs. The conjecture is that the synergy of these cyclic granularities are thus playing a role while deciding if the resulting plot would be a good candidate for exploratory analysis. Harmonies are pairs of granularities that do not contain empty combinations and could possibly be useful for exploring patterns. However, plotting clashes should be avoided since they might contain combinations which are empty or have too few observations. For illustration, Figure 5 (a) shows the distribution of half-hourly electricity consumption through letter value plot across months of the year faceted by quarters of the year. This plot does not work because every quarter does not correspond to all months of the year, for example, the first quarter of the year would never correspond to the month December of an year.

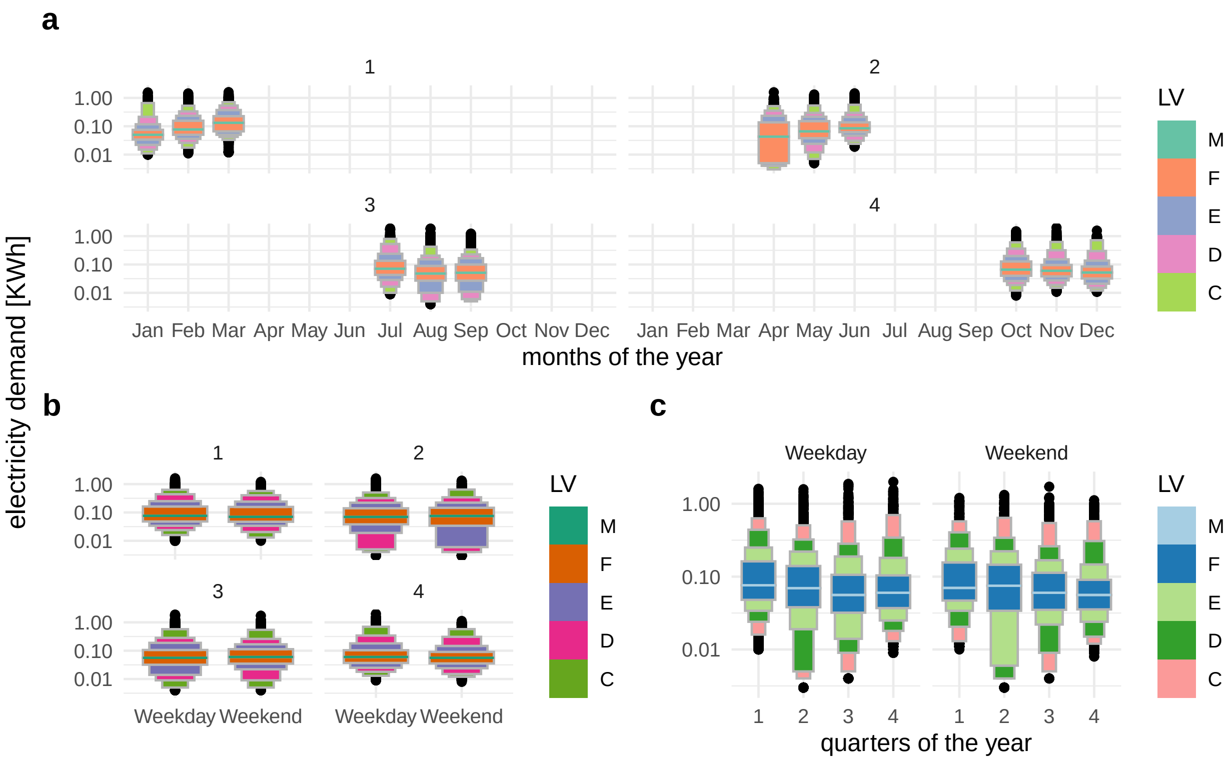


Figure 5: *Distribution of energy consumption displayed through letter value plots. Plot (a) displays consumption across month-of-year faceted by quarters of the year, (b) across weekday/weekend faceted by quarters of the year and (c) across quarters of the year faceted by weekday/weekend. Plots (b) and (c) both show harmonies since each quarter consists of both weekdays and weekends. Conversely, weekend and weekday can occur in every quarter. Analysts should avoid plotting such clashes. Plot (b) helps to compare weekends and weekdays for each quarter. It can be seen that for every quarter weekend and weekday consumption are fairly similar except for the second quarter where the letter values below D and E behave differently, whereas, Plot (c) helps to compare quarters within weekdays and weekends. For example, the quartile spread of consumption shrinks/lowers from first to fourth quarter for weekdays, whereas this pattern is not true for weekends. Plot (a) shows a clash since all quarters do not correspond to all months of the year.*

5.2.3 Interchangeability of mappings

We will discuss the effect of the mapping of cyclic granularities in this section. When we consider data sets of the form $<C_i, C_j, v>$ with C_i mapped to x position and C_j to facets, then A_k's are placed in close proximity 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 A_k's are compared against each other within each group. With the mapping of C_i and C_j reversed, emphasis will shift to different behavior of the variables. Figure 5 (b) shows the letter value plot across weekday/weekend faceted by quarters of the year and Figure 5 (c) shows the same two cyclic granularities with their mapping reversed. Figure 5 (b) helps us to compare weekday and weekend within each quarter and Figure 5 (c) helps to compare quarters within weekend and weekday.

5.3 Number of observations and statistical transformations

Visualizing distributions can be misleading if statistical transformations are performed on rarely occurring categories (Section 4.2). Even when there are no rarely occurring events, number of observations might vary hugely within or across each facet. This might happen due to missing observations in the data or uneven locations of events in time domain. In such cases, the statistical transformations should be used with caution as sample sizes would directly affect both the variance and consequently the confidence interval of the estimators.

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, New South Wales and some parts of Sydney provides customer wise data on energy consumption for every half hour from February 2012 to March 2014. The idea here is to show how to visualize the distribution of the energy consumption across different cyclic granularities in a systematic way to identify different behavioral patterns.

6.1.1 Cyclic granularities search and computation:

The tsibble object `smart_meter10` from R package `gravitas` (Gupta et al. 2020) consisting of `reading_datetime`, `customer_id` and `general_supply_kwh` denoting the index, key and measured variable of the tsibble is used to facilitate the systematic exploration. While trying to explore the energy behavior of these customers systematically across cyclic time granularities, the first thing to consider is which cyclic time granularities we can look at exhaustively. Let us consider conventional time deconstructions for a Gregorian calendar (second, minute, half-hour, hour, day, week, month, year). Since the interval of this tsibble is 30 minutes, the temporal granularities may range from half-hour to year. Considering 6 linear granularities half-hour, hour, day, week, month and year in the hierarchy table, $N_C = (6*5/2) = 15$. If N_C seem too large, the smallest and largest linear granularities could be considered to be removed for analysis. Granularities half-year and year are removed to have $N_C = (4*3/2) = 6$ and 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. Now that we have a list of cyclic granularities to look at, we should be able to compute them using Section 3.4.

6.1.2 Screening and visualizing harmonies

From the search list, $N_C = 7$ cyclic granularities are chosen for which we would like to derive insights of energy behavior. Recalling the data structure <C_i, C_j, `general_supply_kwh`> for exploration $\forall i, j \in \{1, 2, \ldots, 7\}$, each of these 7 cyclic granularities can either be mapped to x-axis or to facet. Choosing 2 of the possible 7 granularities, which is equivalent to having $^7P_2 = 42$ candidates for visualization. Harmonies can be identified among those 42 possibilities to narrow the search. Table 5 shows 16 harmony pairs after removing clashes and any cyclic granularities with levels more than 31, as effective exploration becomes difficult with many levels (Section 5.2.1).

Few harmony pairs are displayed in Figure 6 to illustrate the impact of different distribution

Table 5: *Harmonies with a pair of cyclic granularity one placed on facet and the other on x-axis. Out of 42 possible combinations of cyclic granularities, only 16 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

plots and reverse mapping. For each of Figure 6 (b) and (c), C_i is the circular granularity day-type (weekday/weekend) and C_j is hour of the day. The geometry used for displaying the distribution is chosen as area-quantiles and violins in Figure 6 (b and c respectively). Figure 6 (a) displays reverse mapping of C_i and C_j with C_i denoting hour of the day and C_j denoting day-type with distribution geometrically displayed as boxplots. In Figure 6 (b), the black line is the median, whereas the purple band covers 25th to 75th percentile, the orange band covers 10th to 90th percentile and the green band covers 1st to 99th percentile. The first facet represents the weekday behavior while the second one displays the weekend behavior and energy consumption across each hours of the day is shown inside each facet. The energy consumption is extremely (positive-

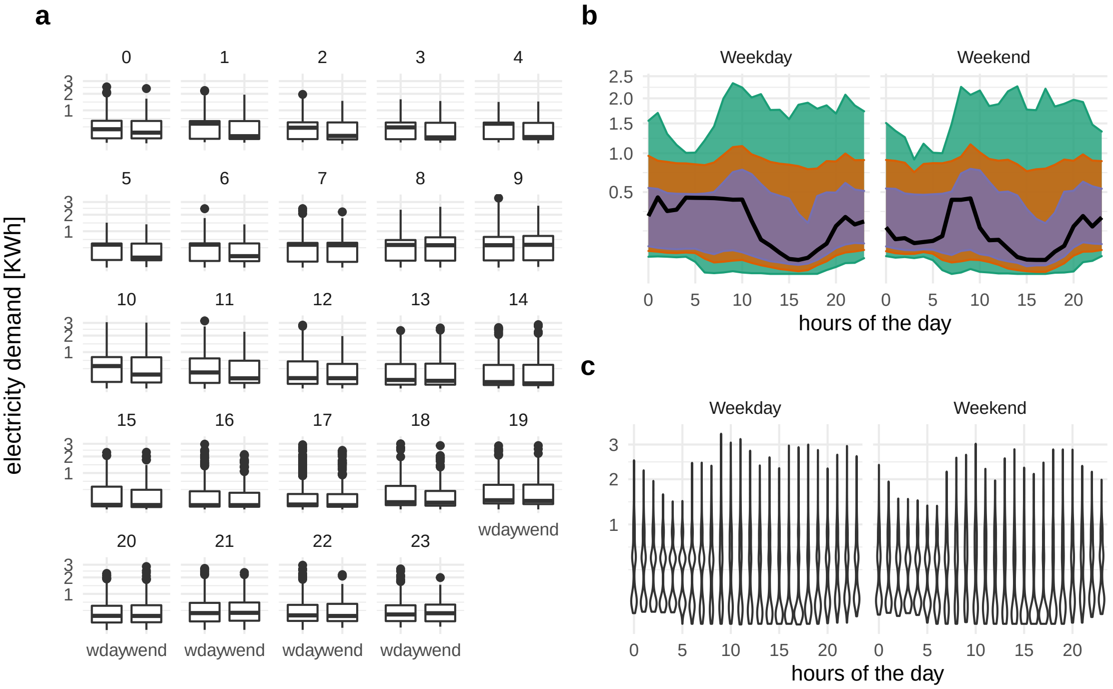


Figure 6: *Energy consumption of a single customer shown with different distribution displays, and granularity arrangements. Two granularities are used: hour of the day (I) and weekday/weekend (II). Plot (a) shows granularity I faceted by granularity II, and plots (b), (c) shows the converse mapping. Plot (a) makes a comparison of usage by workday within each hour of the day using side-by-side boxplots. Generally, on a work day 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 put the emphasis on the time series, for example, the median consumption over time shows prolonged usage in the morning on weekdays. Plot (c) uses a violin plot to place emphasis on distributional differences across hours. It can be seen that the morning use on weekdays is bimodal, some work days there is low usage, which might indicate the person is working from home and also having a late start.*

or right-) 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 as compared to weekends.

Consider looking at violin plots instead of quantile plots to look at the same data in Figure 6(c) for some more insights. There is bimodality in the early hours of the day, implying both low and high energy consumption is probable in the early hours of the day both for weekdays and weekends. If we visualize the same data with reverse mapping of the cyclic granularities, then the natural tendency would be to compare weekend and weekday behavior within each hour and not across hours. For example in Figure 6(a), 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 might shed lights on different aspect of the energy behavior for the same sample population.

If the data for all keys are visualized together, it might lead to Simpson's paradox, which occurs when one observation shows a particular behavior, but this behavior paradoxically becomes obscured by aggregation. For example in a particular neighborhood one household may have the least daily power consumption for a full week, yet still not be the household with the minimum weekly power consumption. This is an intuitive possibility, because heterogeneous `customer_id`'s with very different occupation or demographics will tend to have very different energy behavior and combining them together will somehow weaken any typical or extreme behavior. A strategy for analyzing periodicities of multiple keys together could be to group the keys basis their energy behavior across different cyclic granularities. This is beyond the scope of the current work.

6.2 T20 cricket data of Indian Premiere League

The method is not only restricted to temporal data, and can be generalized to many hierarchical granularities (with continuous and uni-directional nature). We illustrate this with an application to the sport cricket. Although there is no conventional time component in cricket, each ball can be thought to represent an ordering from past to future with the game progressing forward with each ball. In the Twenty20 format, an over will consist of 6 balls (with some exceptions), an innings is restricted to a maximum of 20 overs, a match will consist of 2 innings and a season consists of several matches. Thus, similar to time, there is a hierarchy where ball is nested within overs, overs nested within innings and innings within matches. The idea of cyclic granularities

Table 6: *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

can be likewise mapped to this hierarchy. Example granularites then include ball of the over, over of the innings and ball of the innings. Although most of these cyclic granularities are circular in design of the hierarchy, in application of the rules some granularities are aperiodic. For example, in most cases an over will consist of 6 balls with some exceptions like wide balls or when an innings finishes before the over finishes. Thus, the cyclic granularity ball-of-over will be circular in most cases and aperiodic in others.

The Indian Premier League (IPL) is a professional Twenty20 cricket league in India contested by eight teams representing eight different cities in India. The ball by ball data for IPL season 2008 to 2016 is fetched from Kaggle. The `cricket` data set in the `gravitas` package summarizes the ball-by-ball data across overs and contains information 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. This could be useful in a periodic world when we wish to compute any circular/quasi-circular granularity based on a hierarchy table which look like Table 6. However, even if the situation is not periodic, it can be interesting to visualize the distribution of a measured variable to shed light on the aperiodic behavior of a non-temporal data set similar to aperiodic events like formal meetings, workshops, conferences, school semesters in a temporal set up.

There are many interesting questions that could possibly be answered with the `cricket` data set. For example, it would be interesting to see if 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 final playoffs from 2010 to 2015. We take their example in order to dive deeper into this question. From Figure 7(a), it can be observed 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 median and quartile deviation of runs for the teams batting in the second inning. This seem to indicate that players feel mounting pressure to score more runs as they approach towards the end of the first inning. Whereas teams batting in the second innings have a set target in mind and are not subjected to such mounting pressure and may adopt a more conservative strategy, to score runs. Thus winning teams like CSK and MI seem to employ different inning strategies when it comes to their batting order.

Another fascinating exploration would be to examine if runs per over decrease in the subsequent over if fielding (defending) was good in the previous over? For establishing the fielding quality, we apply an indicator function on dismissals (1 if there was at least one wicket in the previous over due to run out or catch, 0 otherwise). Runs in the current over is then the observation variable. Dismissals in the previous over can lead to a batsman adopting a more defensive play style. Figure 7(b) shows that no dismissals in the previous over leads to a higher median and quartile spread of runs per over as compared to the case when there has been at least one dismissal in the previous over. Wickets per over are considered as an aperiodic cyclic granularity with wickets as an aperiodic linear granularity. These granularities do not appear in the hierarchy table since it is difficult to position them in a hierarchy. These are similar to holidays or special events in temporal data.

7 Discussion

Exploratory data analysis and data analysis in general 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 a framework to systematically explore distribution of an univariate measured variable across two cyclic time granularities by creating any cyclic granularity, ranking a list of harmonies and thereby identifying possible distribution plots for effective visualization based on relationship and levels of the cyclic 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 like `almanac` and `gs` provide functionality to create recurring events that are not periodic. These

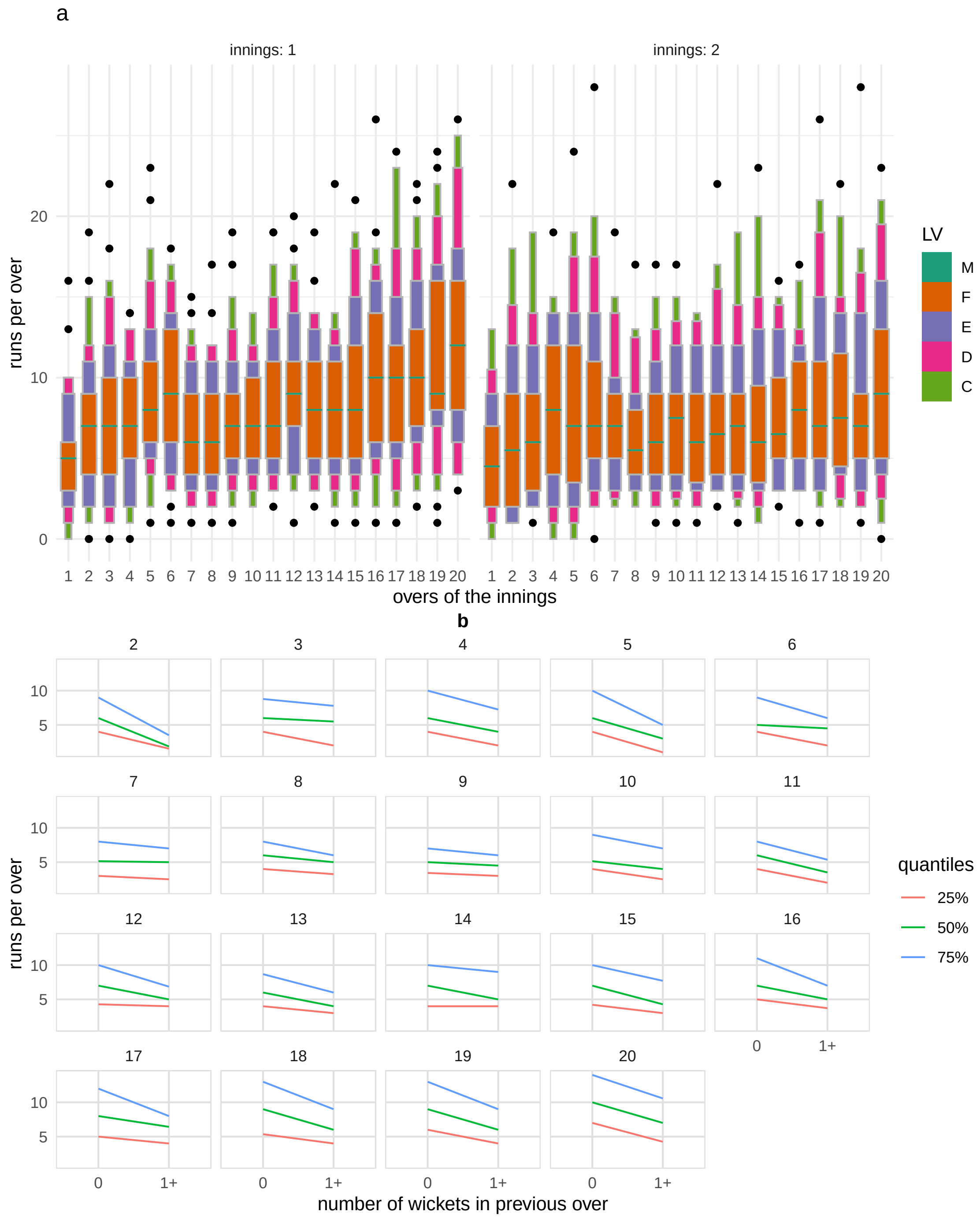


Figure 7: *Runs per over shown with different distribution displays, and granularities. Plot (a) shows letter value plot across overs faceted by innings. For the team batting in the first innings there is an upward trend of runs per over, while there is no such pattern of runs for the teams batting in the second innings. Plot (b) shows quantile plot of runs per over across an indicator of wickets in previous over faceted by current over. This indicates that at least one wicket in the previous over leads to lower median run rate and quartile spread in the subsequent over.*

functions can be imported in the gravitas package to accommodate for aperiodic cyclic granularities. Another future direction of work could be to further refine the search of harmonies by only selecting pairs of cyclic granularities for which variation in measured variable is significant and rating these selected harmony pairs in order of importance for exploration.

Acknowledgements

The authors would like to thank the cohort NUMBATS, Monash University for sharing their wisdom and experience in developing R packages, and Dr Peter Toscas from Data61 CSIRO for providing useful inputs on improving the analysis of the smart meter application. The package gravitas was built during the Google Summer of Code, 2019. We would also like to thank Nicholas Spyrison for many useful discussions, sketching figures and feedback on the manuscript. More details about the package can be found at sayani07.github.io/gravitas. This article was created with knitr (Xie 2015, Xie (2020)) and rmarkdown (Xie et al. 2018, Allaire et al. (2020)). 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.

8 Supplementary Materials

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

R-package: The ideas presented in this article have been implemented in the open-source R (R Core Team 2019) 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, check feasibility of creating plots or drawing inferences for any two cyclic granularities by providing list of harmonies and recommend prospective probability distributions through factors described in the article. Version 0.1.2 of the package was used for the results presented in the article and is available on Github (`https://github.com/Sayani07/gravitas`).

R-packages: Each of the R packages used in this article are available online with URLs provided in the bibliography.

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.

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.

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', *Am. Stat.* **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', *Am. Stat.* **50**(2), 120–126.

Mcgill, R., Tukey, J. W. & Larsen, W. A. (1978), 'Variations of box plots', *Am. Stat.* **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 (2019), *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. (2001), *Calendrical Calculations*, millennium edition edn, Cambridge University Press.

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

Wang, E., Cook, D. & Hyndman, R. J. (2020*a*), 'Calendar-based graphics for visualizing people's daily schedules'.

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* **0**(0), 1–13.

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

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. ISBN 978-1498716963.
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. ISBN 9781138359338.
URL: *https://bookdown.org/yihui/rmarkdown*