Visualizing probability distributions across bivariate cyclic temporal granularities

Sayani Gupta *
Department of Econometrics and Business Statistics, Monash University
and
Rob J Hyndman
Department of Econometrics and Business Statistics, Monash University
and
Dianne Cook
Department of Econometrics and Business Statistics, Monash University
and
Antony Unwin
University of Augsburg

March 19, 2020

Abstract

Recent advances in technology greatly facilitates recording and storing data at much finer temporal scales than was previously possible. As the frequency of time-oriented data increases, the number of questions about the observed variable that can be addressed by visual representation also increases. We propose some new tools to explore this type of data, which deconstruct time in many different ways. There are several classes of time deconstructions including 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, and day of the week, quasi-circular such as day of the month, and aperiodic such as public or school holidays.

The hierarchical structure of linear granularities creates a natural nested ordering resulting in single-order-up and multiple-order-up granularities. For example, hour of the week and second of the hour are both multiple-order-up, while hour of the day and second of the minute are single-order-up.

Visualizing data across granularities which are either single-order-up or multiple-order-up or periodic/aperiodic helps us to understand periodicities, pattern and

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

anomalies in the data. Because of the large volume of data available, using displays of probability distributions conditional on one or more granularities is a potentially useful approach. This work provides tools for creating granularities and exploring the associated time series within a tidy workflow, so that 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 like GDP is 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 idea aligns with the notion of EDA (Tukey 1977) which emphasizes the use of multiple perspectives on data to help formulate hypotheses before proceeding to hypothesis testing. Visualizing probability distributions conditional on one or more granularities is a potentially useful approach for exploration. Analysts are expected to iteratively explore many possible choices of time granularities for comprehending possible periodicities in the data. But too many choices and a lack of a systematic approach to do so might become overwhelming.

Calendar-based graphics (Wang et al. 2018) 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 respect the linear progression of time such as hours, days, weeks and months, and 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 cyclic or repeating granularities. Hence, there is a need to define some new cyclic 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 examples 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 (G Grolemund 2011) provides tools to access and manipulate common date-time objects. But 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 that they were installed, some as early as 2012. Households are distributed geographically and have different demographic properties such as the existence of solar panels, central heating or air conditioning. The behavioral patterns in households vary substantially, for example, some families use a dryer for their clothes while others hang them on a line, and some households might consist of night owls, while others are morning larks. It is common to see aggregates (see Goodwin, S And Dykes, 2012) of usage across households, such as half-hourly total usage by state, because energy companies need to plan for maximum loads on the network. But studying overall energy use hides the distributions 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/anomalies. Although, the motivation came through the smart meter example, this is a problem that is relevant to any temporal data observed more than once per year.

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 pair of granularities.

The remainder of the paper is organized as follows: Section 2 provide some background material on linear granularities and introduces calendar algebra for computing different linear granularities. Section 3 formally characterizes different cyclic time granularities by extending the framework of linear time granularities. Section 4 introduces calendar algebra for computing cyclic time granularities. Section 5 discusses the data structure for exploring the conditional distributions of the associated time series across pairs of cyclic time granularities. Section 6 discusses the role of different factors in constructing an informative and trustworthy visualization. Section 7 examines how systematic exploration can be carried out for a temporal and non-temporal application. Section 8 summarizes this paper and discusses possible future direction.

2 Linear time granularities

Discrete abstraction 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.

2.1 Definitions and relationships

There have been several attempts to provide a framework for formally characterizing time granularities. One of the first attempts was provided by Bettini et al. (1998) who proposed following definitions:

Definition 1 (Time domain) *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*.*

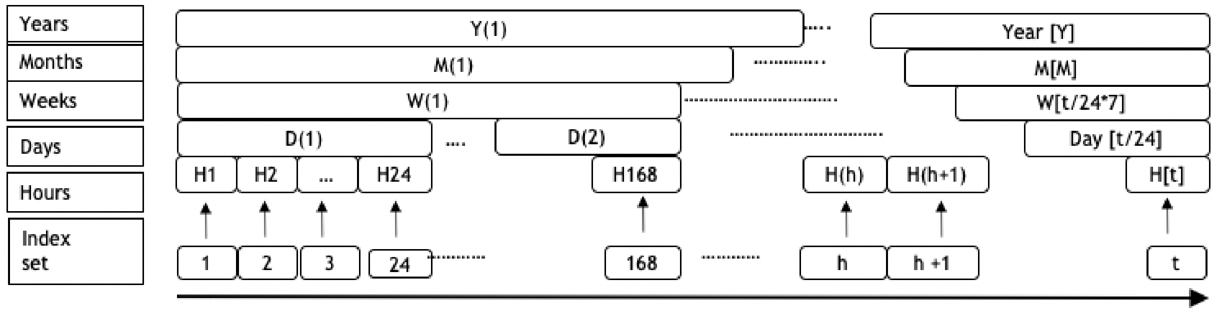

Figure 1: The time domain distributed as linear granularities hours, days, weeks and months. These are ordered with ordering guided by integers and hence is unidirectional and non-repeating.

We will assume the time domain is *discrete*; i.e., there is unique predecessor and successor for every element in the time domain except for the first and last.

Definition 2 (Linear granularity) *A linear granularity is a mapping G from the integers (the index set) 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.

Definition 3 (Granule) *Each non-empty subset $G(i)$ is called a granule, where i is one of the indexes and G is a linear granularity.*

The first condition in Definition 2 implies that the granules in a linear granularity are non-overlapping and their index order is same as time order. Figure 1 illustrates the linear time granularites hours, days, weeks, months and years, each of which are non-repeating in nature and arranged from past to future. If we consider T hours, we will have $\lfloor T/24 \rfloor$ days, $\lfloor T/(24*7) \rfloor$ weeks and so on.

In addition to the integer index, each granule could be associated with a textual representation which is called the label. When the label set is exactly the integers, the granularities are called "full-integer" labeled granularity, such as in Definition 2.

Bettini et al. (1998) discusses the relationships between linear time granularities, periodicities in time, and the structure of a calendar.

Definition 4 (Groups into) *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, day $\trianglelefteq$ week since every granule of "week" is the union of some set of granules of "day". Consider another example where day $\trianglelefteq$ month. This relationship however is incomplete without its association to periodicity. Each month is a grouping of the same number of days over years, hence the period of the grouping (day, month) is one year, if leap years are ignored. The period becomes 400 years with the inclusion of leap years and all their exceptions.

Definition 5 (Groups periodically) *A granularity H is periodical with respect to a granularity G 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 $H(i+R) = \bigcup_{j \in S} G(j+P)$.*

If $S_0, \ldots, S_{R-1}$ are the sets of indexes of G describing $H(0), \ldots, H(R-1)$, respectively, then the description of an arbitrary granule $H(j)$ is given by $\bigcup_{i \in S_{j \bmod R}} G(P * \lfloor j/R \rfloor + i)$. Also, granularities can be periodical with respect to other granularities, except for a finite number of spans of time where they behave in an anomalous way (Bettini & De Sibi 2000). These are called quasi-periodic relationships.

For example, in a Gregorian calendar without leap years we could say days group periodically into months with the period $P = 365$ and the number of granules of "months" in each period given by $R = 12$. In a Gregorian calendar with leap years, days group quasi-periodically into months with the exceptions of the time domain corresponding to 29^{th} February of any year.

A discrete time model often uses a fixed smallest linear granularity also known as *bottom granularity* (Bettini et al. (1998)) Granules in bottom granularity or any finer granularity may be aggregated in some manner to form larger granules belonging to a coarser granularity. A system of multiple granularities in lattice structures is referred to as a *calendar* (see Dyreson et al. (2000)).

2.2 Computation through calendar algebra

Linear time granularities are computed through an algebraic representation for time granularities, which is referred to as calendar algebra (Ning et al. 2002). It is assumed that there exists a *bottom granularity* and calendar algebra operations are designed to generate new granularities recursively from the bottom one. Some relevant calendar algebra operations are discussed below; these will be used in Section 3 for illustrations in circular and aperiodic cyclic granularities.

The grouping operation: Let G_1 be a full-integer labeled granularity, and m a positive integer. The grouping operation $\text{Group}_m(G)$ generates a new granularity G_2, by partitioning the granules of G_1 into m-granule groups and making each group a granule of the resulting granularity. More precisely, $G_2 = \text{Group}_m(G_1)$ is the full-integer labeled granularity such that for each integer i,

$$G_2(i) = \bigcup_{j=(i-1)m+1}^{im} G_1(j).$$

Example: minute = Group_{60}(second); hour = Group_{60}(minute)

The altering-tick operation: Let G_1, G_2 be full-integer labeled granularities, and l, k, m integers, where G_2 partitions G_1, and $1 \leq l \leq m$. The altering-tick operation $\text{Alter}^m_{l,k}(G_2, G_1)$ operation modifies the granules of G_1 so that the l^{th} granule of each period has k additional granules of G_2.

For example, due to the uneven length of months, we need to use the altering-tick operation to compute months from days. We can consider a granularity "pseudomonth" which computes months only for non-leap years by grouping 31 days, and then shrink April (4), June (6), September (9) and November (11) by 1 day, and shrink February (2) by 3 days (Ning et al. 2002). For more variations of calendar algebra operations, see Ning et al. (2002).

3 Cyclic time granularities

We propose a formalism of cyclic time granularities through the tsibble (Wang et al. 2019) data structure. 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 time domain, as defined by Bettini et al. (1998), is essentially a mapping of the index set to the time index of a tsibble. A linear granularity is a mapping of the index set to subsets of the time domain. For example, if the time 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.

Cyclic granularities can be constructed by relating two linear granularities. The mappings between these linear granularities could be *regular* with every granule of coarser linear granularity being made up of same number of granules of finer linear granularity or *irregular*. Cyclic time granularities are referred to as **circular** if mappings are regular and **quasi-circular** if the mappings are irregular. Examples of common circular granularities include hour of the day, and day of the week, whereas examples for quasi-circular granularities can be day of the month or week of the month.

3.1 Circular

Definition 6 (Circular granularity) *A circular granularity $C_{B,G}$ relates a linear granularity G to the bottom granularity B, if*

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

where z denotes the index set, B denotes a full-integer labelled bottom granularity which groups periodically into linear granularity G with regular mapping, $P \equiv P(B,G)$ is the number of granules of B in each granule of G, and L is a label mapping that defines an unique label for each index $l \in \{0, 1, \ldots, (P-1)\}$.

Figure 2 depicts the circular granularity "Day-of-Week", constructed from linear granularities "day" (the bottom granularity) and "week". The granules that overlap share elements from the underlying time domain. In this example, $P = 7$, and the circular granularity $C_{\text{day,week}}$ representing day-of-week consists of the repeated pattern $\{L(0), L(1), L(2), L(3), L(4), L(5), L(6)\}$.

Each circular granularity can use descriptive label mappings, or be simply labelled using

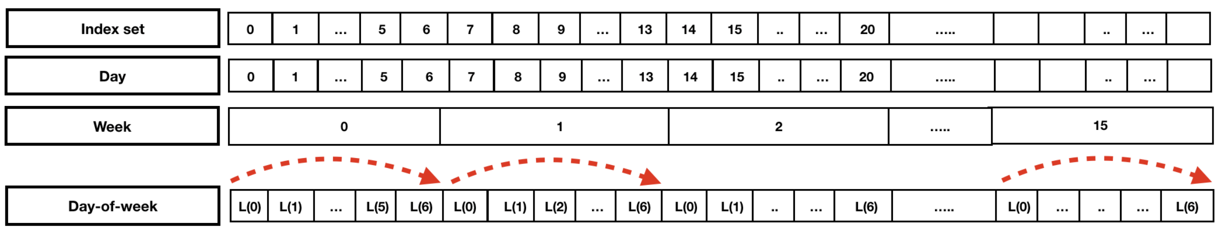

Figure 2: The time domain distributed as linear granularities Day and Week and circular granularity Day-of-Week with $P = 7$ is constructed using the two linear granularities. Day-of-Week has a repeating pattern unlike Day and Week and has an ordering only within a period represented by hashed arrows.

the index. The label mapping L in Figure 2 can be defined as

$$L : \{0, 1, 2, \ldots, 6\} \longmapsto \{\text{Sun}, \text{Mon}, \ldots, \text{Sat}\}$$

or

$$L : \{0, 1, 2, \ldots, 6\} \longmapsto \{\text{Sunday}, \text{Monday}, \ldots, \text{Saturday}\}$$

for example.

In general, any circular granularity relating two linear granularity can be expressed as $C_{(G,H)}(z) = L(\lfloor z/P(B, G) \rfloor \bmod P(G, H))$, where linear granularity H is periodic with respect to linear granularity G with regular mapping such that the number of granules of G in each granule of H is $P(G, H)$. Table 1 shows representation of circular granularities C_i relating two linear granularities with P_i being the number of granules of the finer granularity in each granule of the coarser granularity and L_i is the associated label mapping. The bottom granularity is assumed to be minutes in this case.

3.2 Quasi-circular

A **quasi-circular** granularity can not be defined using modular arithmetic since they are formed using two linear granularities with irregular mapping. However, they are still formed with linear granularities, one of which "groups periodically into" the other. Table 2 shows some example of quasi-circular granularities (Q_i) with (P_i) denoting the plausible choices of number of granules of the finer granularity inside each granule of the coarser one.

minute-of-hour:	$C_1 = L_1(z \bmod 60)$	$P_1 = 60$
minute-of-day:	$C_2 = L_2(z \bmod 60 * 24)$	$P_2 = 1440$
hour-of-day:	$C_3 = L_3(\lfloor z/60 \rfloor \bmod 24)$	$P_3 = 24$
hour-of-week:	$C_4 = L_4(\lfloor z/60 \rfloor \bmod 24 * 7)$	$P_4 = 168$
day-of-week:	$C_5 = L_5(\lfloor z/24 * 60 \rfloor \bmod 7)$	$P_5 = 7$

Table 1: Illustrative circular granularities with bottom granularity "minutes". These circular granularities relate two linear granularities one of which groups periodically into the other with regular mapping.

day-of-month:	Q_1	$P_1 = 31, 30, 29, 28$
hour-of-month:	Q_2	$P_2 = 24 * 31, 24 * 30, 24 * 29, 24 * 28$
day-of-year:	Q_3	$P_3 = 366, 365$
week-of-month:	Q_4	$P_4 = 5, 4$

Table 2: Illustrative quasi-circular granularities with potential period lengths. These quasi-circular granularities relate two linear granularities one of which groups periodically into the other with irregular mapping. Illustrative quasi-circular granularities

Definition 7 *(quasi-circular granularity): A quasi-circular granularity $Q_{B,G'}$ that relates linear granularities G' and bottom granularity B, if*

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

where $z \in \mathbb{Z}_{\geq 0}$ denotes the index set, B denotes a full-integer labelled bottom granularity which groups periodically into linear granularity G' with irregular mapping, P' and R' denote the period of the grouping (B, G') and the number of granules of G' in each of these periods, L is a label mapping that defines an unique label for each index $l \in \{0, 1, \ldots, (\lceil P'/R' \rceil - 1)\}$, T_w are the sets of indices of B describing $G'(w)$ such that $G'(w) = \bigcup_{z \in T_w} B(z)$ and $|T_w|$ is the cardinality of set T_w.

Example: Example showing quasi-circular granularities relating two linear granularities each with bottom granularities are visually depicted in a series of slots in Figure 3. Each granule is represented by a box. Two linear granularities $G' = Alter^2_{(1,-1)}(BG, Group_3(B))$ and $H' = Alter^2_{(1,-2)}(BG, Group_7(B))$ are considered. $Q_{B,G'}$ and $Q_{B,H'}$ are repetitive categorization of time, similar to circular granularities, except that the number of granules of B is not necessarily the same across different granules of G' or H'. For G', $T_0 = \{0, 1\}$ and $T_1 = \{2, 3, 4\}$. For H', $T_0 = \{0, 1, 2, 3, 4, 5, 6\}$ and $T_1 = \{7, 8, 9, 10, 11\}$, then we will have Equation 3 and Equation 1.

$$\begin{aligned} Q_{B,G'}(8) &= L(8 - \sum_{w=0}^{3-1} |T_{w \bmod 2}|), \quad since \quad 8 \in T_3 \\ &= L(1) \end{aligned} \tag{3}$$

$$\begin{aligned} Q_{B,H'}(10) &= L(10 - \sum_{w=0}^{1-1} |T_{w \bmod 2}|), \quad since \quad 10 \in T_1 \\ &= L(3) \end{aligned} \tag{4}$$

Discussion: 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 5) . Here $w \bmod R'$ 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 $\sum_{w=0}^{k-1} |T_w|$ 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 7.

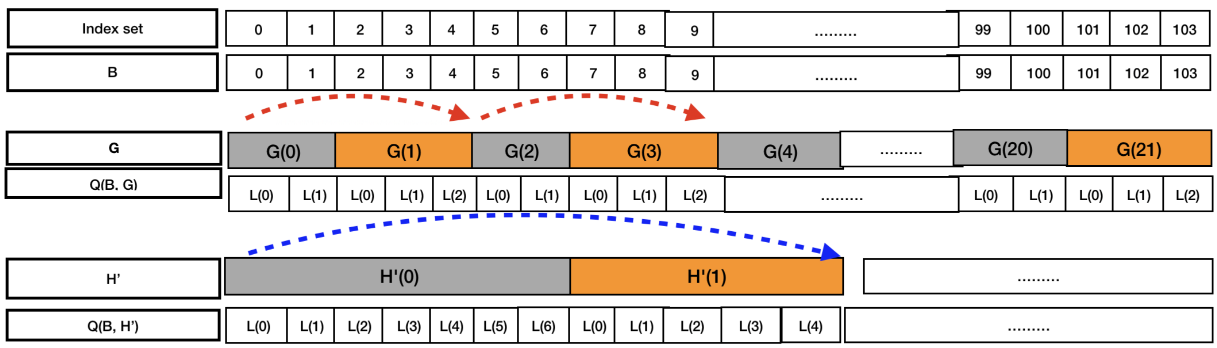

Figure 3: The time domain distributed as linear granularities B, G', H' and quasi-circular granularities $Q(B, G')$ and $Q(B, H')$. Both $Q(B, G')$ and $Q(B, H')$ have repeating patterns unlike G' or H' and the patterns are guided by the distribution of B within G' and H' within a period shown by the hashed arrow.

3.3 Aperiodic

Aperiodic time granularities are the ones which can not be specified as a periodic repetition of a pattern of granules. Most public holidays repeat every year, but there is no finite (or 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 scheduling 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 $\forall\{i \in 1, 2, \dots, n\}$ such that 5 does not hold true for them with respect to B (the bottom granularity). However, $B \trianglelefteq M_i$ $\forall\{i \in 1, 2, \dots, n\}$. Then according to Definition 4, for each index j there exists a (possibly infinite) subset $T_{\{i_j\}}$ of the integers such that $M_i(j) = \bigcup_{z \in T_{i_j}} B(z)$ and 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 8 (Aperiodic cyclic granularity) *An aperiodic cyclic granularity $A_{B,M}$ re-*

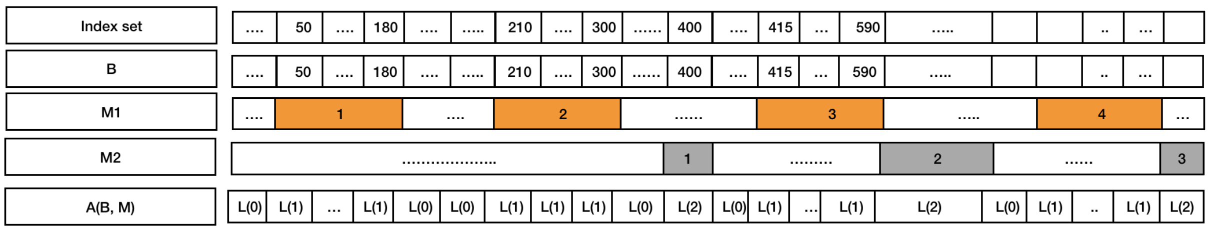

Figure 4: The time domain distributed as linear aperiodic granularities B, $M1$, $M2$, M and aperiodic cyclic granularity $A(B, M)$. $A(B, M)$ have repeating patterns with three categories, two of which correspond to the aperiodic linear granularities $M1$ and $M2$ and other one correponding to no aperiodic granularity.

lates aperiodic linear granularity M and bottom granularity B, if

$$\begin{aligned} A_{B,M}(z) &= L(i), \quad z \in T_{i_j} \quad \textit{and} \quad \forall i \in 1, 2, \ldots, n \\ &= L(0), \quad \textit{otherwise} \end{aligned} \tag{5}$$

where, $z \in \mathbb{Z}_{\geq 0}$ denotes the index set, T_{i_j} are the sets of indices of B describing aperiodic linear granularities M_i such that $M_i(j) = \bigcup_{z \in T_{i_j}} B(z)$, $M = \bigcup_{i=1}^{n} M_i$.

Example: Figure 4 shows two aperiodic linear granularity M_1 and M_2. The first slot in the diagram shows the index set and $A_{B,M}$ consists of three categories, $L(1)$ corresponding to the first aperiodic event M_1, $L(2)$ corresponding to aperiodic event M_2 and the $L(0)$ corresponding to no event/aperiodic linear granularity.

Discussion: Here, M_1 can be an aperiodic linear granularity denoting block leaves during Christmas, M_2 can be the one denoting Easter, M_n can be Labor day and M can be conceived as the aperiodic linear granularity denoting any holidays.

4 Cyclic calendar algebra

In Section 3, we discussed how we can define cyclic granularities by extending the framework of linear granularities defined in Bettini et al. (1998). In this section, we will see how to obtain cyclic granularities through an algebraic representation of other cyclic granularities through cyclic calendar algebra operations.

The cyclic calendar algebra consists of broadly two kinds of operations: (1) **single-to-multiple:** and (2) **multiple-to-single:** which entails the representation of multiple-order-up cyclic granularities from single order-up cyclic granularities and vice versa. The hierarchical structure of time creates a natural nested ordering which can produce *single-order-up* or *multiple-order-up* granularities. We shall use the notion of a hierarchy table and order to define them.

Order of a linear granularity can be comprehended as the level of coarseness associated with a linear granularity. For two linear granularities G and H, if G groups into H or H is composed of higher number of granules of the bottom granularity than G, then G is of lower order than H. In any hierarchy table, linear granularities are arranged from lowest to highest order.

Hierarchy table Let $H_n : (G, C, K)$ be a hierarchy table containing n linear granularities. G_l and G_m represent the linear granularity of order l and m respectively with $l < m$. $K \equiv P(l, m)$ represents the period length of G_l with respect to G_m for circular granularity $C_{G(l),G(m)}$ and $K \equiv k(l, m)$ represents the operation to obtain G_m from G_l for quasi-circular granularity $C_{G(l),G(m)}$, $\forall l, m \in 1, 2, \ldots, n$. We assume that the cyclic granularity $C_{G(l),G(m)}$ is either circular or quasi-circular. It is difficult to position an aperiodic linear granularity in a hierarchy and hence the possibility of an aperiodic cyclic granularity is not considered.

We refer to granularities which are nested within multiple levels of the hierarchy table as *multiple-order-up* and those concerning a single level as *single-order-up* granularities.

Example: So far we have used example of cyclic granularities from Gregorian calendar as it is the most widely used calendar. But it is far from being the only one. The day is the basic unit of time underlying most calendars (Reingold & Dershowitz 2001). Various calendars, however, use different conventions to structure days into larger units: weeks, months, years and cycle of years. The French revolutionary calendar divided each day into 10 “hours”, each “hour” into 100 “minutes” and each “minute” into 100 “seconds”. Nevertheless, for any calendar a hierarchy table can be defined. For example, in Mayan calendar, one day was referred to as 1 kin and the calendar was structured such that 1 kin = 1 day; 1 uinal = 20 kin; 1 tun = 18 uinal; 1 katun = 20 tun and 1 baktun = 20 katun. Table 3 represents the the hierarchy table for the Mayan calendar.

Table 3: Hierarchy table for Mayan calendar. G and C represent the linear and single order-up cyclic granularities respectively. All cyclic granularities are circular with K as the period length.

G	C	K
kin	kin-of-uinal	20
uinal	uinal-of-tun	18
tun	tun-of-katun	20
katun	katun-of-baktun	20
baktun	1	1

Examples of multiple-order-up granularities can be kin-of-tun or kin-of-baktun whereas examples of single-order-up granularities may include kin-of-uinal, uinal-of-tun etc.

4.1 Single-to-multiple

4.1.1 All circular single order-up granularities

Circular single-order-up granularities can be used recursively to obtain multiple order up circular granularity. Since, the operation requires the use of modular arithmetic, it is important that the label mapping of the individual circular 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 however be chosen depending on the context.

$$
\begin{aligned}
C_{(G_l,G_m)}(z) &= L(C_{G_l,G_{l+1}}(z) + P(l,l+1)(C_{(G_{l+1},G_m)}(z))) \\
&= L(C_{(G_l,G_{l+1}}(z) + P(l,l+1)[C_{G_{l+1},G_{l+2}}(z) + P(l+1,l+2)(C_{G_{l+2},G_m}(z))) \\
&= L(C_{G_l,G_{l+1}}(z) + P(l,l+1)(C_{G_{l+1},G_{l+2}}(z)) + P(l,l+1)P(l+1,l+2)(C_{G_{l+2},G_m})) \\
&= L(C_{G_l,G_{l+1}}(z) + P(l,l+1)(C_{G_{l+1},G_{l+2}}(z)) + P(l,l+2)(C_{G_{l+2},G_{l+m}}(z))) \\
&\vdots \\
&= L(\sum_{i=0}^{m-l-1} P(l,l+i)(C_{G_{l+i},G_{l+i+1}}(z)))
\end{aligned}
\tag{6}
$$

, for $l < m+1$ and $P(i,j) = 1, \quad \forall i = j \in \{0, 1, \ldots, m-l-1\}$

Example: Let us use Equation 6 to compute the multiple-order-up granularity uinal_katun for Mayan calendar.

$$
\begin{aligned}
C_{uinal,baktun}(z) =& L(C_{uinal,tun}(z) + P(uinal,tun)C_{tun,katun}(z) + P(uinal,katun)C_{katun,baktun}(z)) \\
=& L(\lfloor z/20 \rfloor \bmod 18 + 18 * \lfloor z/20 * 18 \rfloor \bmod 20 \\
& \quad + 18 * 20 \lfloor z/20 * 18 * 20 \rfloor \bmod 20)
\end{aligned}
\tag{7}
$$

4.1.2 Circular or quasi-circular single order-up granularities

We consider the case of only one quasi-circular single order-up granularity in the hierarchy table. Any multiple order-up quasi-circular granularity $C_{l,m}(z)$ can then be obtained as a function of a single (or multiple) order-up circular granularity $C_{l,m'}(z)$ and single order-up quasi-circular granularity $C_{m',m}(z)$ or vice versa. Two different approaches need to be employed for the following cases:

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

$$C_{G_l,G_{l+2}}(z) = L(C_{G_l,G_{m'}}(z) + P(l,m')(C_{G_{m'},G_m}(z))) \tag{8}$$

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

$$C_{G_l,G_{l+2}}(z) = L(C_{G_l,G_{m'}}(z) + \sum_{w=0}^{C_{m',m}(z)-1} (|T_w|)) \tag{9}$$

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

Example: Suppose we have a hierarchy using linear granularties from Gregorian calendar. Since months consists of unequal number of days, any linear granularity with higher order than months will also have unequal number of days. This is an example of a hierarchy structure which has both circular and quasi-circular single-order-up granularities. Only one single-order-up granularity day-of-month is quasi-circular.

G	C	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

Using Equations 8 and 9, we have:

- $C_{hour,month}(z) = L(C_{hour,day}(z) + P(hour, day) * C_{day,month}(z))$
- $C_{day,year}(z) = L(C_{day,month}(z) + \sum_{w=0}^{C_{month,year}(z)-1}(|T_w|))$, where, T_w is such that $month(w) = \bigcup_{z \in T_w} day(z)$.

4.2 Multiple-to-single

4.2.1 Muliple order-up circular granularities

For a hierarchy table $H_n : (G, C, K)$ with $l_1, l_2, m_1, m_2 \in 1, 2, \ldots, n$ and $l_2 < l_1$ and $m_2 > m_1$, we have

$$C_{G_{l_1}, G_{m_1}}(z) = C_{G_{l_2}, G_{m_2}}(\lfloor z/k(l_2, l_1) \rfloor \bmod P(m_1, m_2)) \tag{10}$$

Example: Considering the same example of Mayan Calendar, it is possible to compute the single-order-up granularity tun-of-katun given the multiple-order-up granularity uinal-baktun using equation 10.

$$C_{tun,katun}(z) = L(\lfloor C_{uinal,baktun}(z)/18 \rfloor \bmod 20) \tag{11}$$

4.2.2 Muliple order-up quasi-circular granularities

The representation of single order-up quasi-circular granularities using multiple order-up quasi-circular granularities is not discussed in this paper and is left for future work.

5 Data structure

Effective exploration or good visualization require good data structures. It is recognized that there is a crucial influence of linear vs cyclic time characteristics on the expressiveness of visualization and analysis. Moreover, one can use calendars based on application domain that define contextual system of linear and cyclic granularities. A recent tidy data structure to support exploration and modeling of temporal data is tsibble (Wang et al. 2019), where data is structured in a semantic manner with reference to observations and variables, with the time index stated explicitly. Since all cyclic granularities can be expressed in terms of the index set, we consider the data structure in (Figure 5 for exploration of temporal data of this kind. This is a two-dimensional array extending the columns of tsibble by including the cyclic granularities. Let us consider two cyclic granularities C_1 and C_2, such that C_1 maps index set to a set $\{A_1, A_2, A_3, \dots, A_n\}$, and C_2 maps index set to a set $\{B_1, B_2, B_3, \dots, B_m\}$ and v denotes the measurement variable. Data sets of the form $<C_1$, C_2, $v>$ can then form the basis for exploration and analysis of the measured variable across bivariate cyclic granularities.

index	cyclic granularity 1	cyclic granularity 2	key	measurements

Figure 5: The data structure for exploring periodicities in data by including two cyclic granularities in the tsibble structure

5.0.1 Harmonies and Clashes

The way cyclic granularities relate become important when we consider the data structure in Figure 5. While considering data sets of the form $< C_1, C_2, v >$, let S_{ij} be the set of index set such that for all $s \in S_{ij}$, $C_1(s) = A_i$ and $C_2(s) = B_j$. Data subsets for each combination of levels (A_i, B_j) like $<A_i$, B_j, $v(s)>$ can be obtained for all $i \in 1, 2, \dots, n$ and $j \in 1, 2, \dots, m$ which will lead to nm 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 nm 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 9 (Clash) *A pair of cyclic granularities which contains structurally, event-driven or build-based empty combinations of its categories.*

Definition 10 (Harmony) *A pair of cyclic granularities that do not contain any missing combinations of its categories.*

Firstly, 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_1 be day-of-month with 31 levels and C_2 be week-of-month with 5 levels. There will be $31 \times 5 = 155$ sets S_{ij} corresponding to possible combinations of C_1 and C_2. 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.

Secondly, 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_1 be day-of-week with 7 levels and C_2 be WorkingDay/NonWorkingDay with 2 levels. While potentially all of these 14 sets S_{ij} 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.

Thirdly, empty combinations can be a result of how granularities are constructed. These are called "build-based" empty combinations. Let C_1 be Business-days, which are days from Monday to Friday except holidays and C_2 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 granularity does.

An example when there will be no empty combinations could be where C_1 and C_2 maps index set to day-of-week and month-of-year respectively. Here C_1 can have 7 levels while C_2 can have 12 levels. So there are $12 \times 7 = 84$ sets S_{ij}. 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.

5.0.2 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 correspond to multiple values of the measured variable. This is unlike the case of linear granularities where each time point corresponds to an unique value 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. But that approach hides the distributions of the measured variable induced by the re-organised data structure. Summarizing the distribution of these multiple observations across cyclic granularities using some summary statistics, each of which can highlight different features of the data or through probability densities could be a potential way to compare and contrast periocities in the data.

However, we need to consider the effect of number of observations on the summarization even for harmonies. Suppose we have T observations, and two cyclic granularities C_1 with n categories and C_2 with m categories. Each element of C_1 occurs approximately T/n times while each element of C_2 occurs approximately T/m times. There are no empty combinations, and each combination will occur on average an equal number of times as $T \to \infty$, so the average number of observations per combination is $T/(mn)$. If we require at least k observations per combination to create a meaningful summarization, then only $T \geq mnk$ will be acceptable. The value of k will depend on the statistical transformation we are producing. For computing deciles, $k = 10$ may be acceptable, but for density estimates, $k \geq 30$ is suggested. Rarely occurring categories such as the 366th day of the year, or the 31st day of the month can suffer from such problem.

6 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 $<C_1, C_2, v>$ serves as the basis of visualizing the distribution of the measured variable, the following layers can be specified:

- Data: $<C_1, C_2, v>$
- Aesthetic mapping (mapping of variables to elements of the plot): C_1 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_2

6.1 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.

6.2 Facet and aesthetic variables

6.2.1 Levels

We will discuss the effect of the levels of cyclic granularities on the choice of plots since space and resolution might become a problem if the number of levels are too high. 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 can be chosen based on levels of common temporal granularities like days-of-month, days-of-fortnight and days-of-week.

6.2.2 Synergy of cyclic granularities

If there are levels of cyclic granularities plotted across x-axis which are not spanned by levels of cyclic granularities plotted across facets or vice versa, we will have empty sets leading to potential ineffective graphs. We hypothesize 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 thus aid exploratory data analysis, and clashes are pairs that do not aid the process of exploratory analysis because they contain empty combinations. As a result, we should avoid plotting clashes.

For illustration, distribution of half-hourly electricity consumption of Victoria is plotted across different time granularities in each of the panel in Figure 6. Figure 6 (a) shows the letter value plot across days of the month faceted by months like January, March and December. This plot works because every day-of-week can occur in every month-of-year.

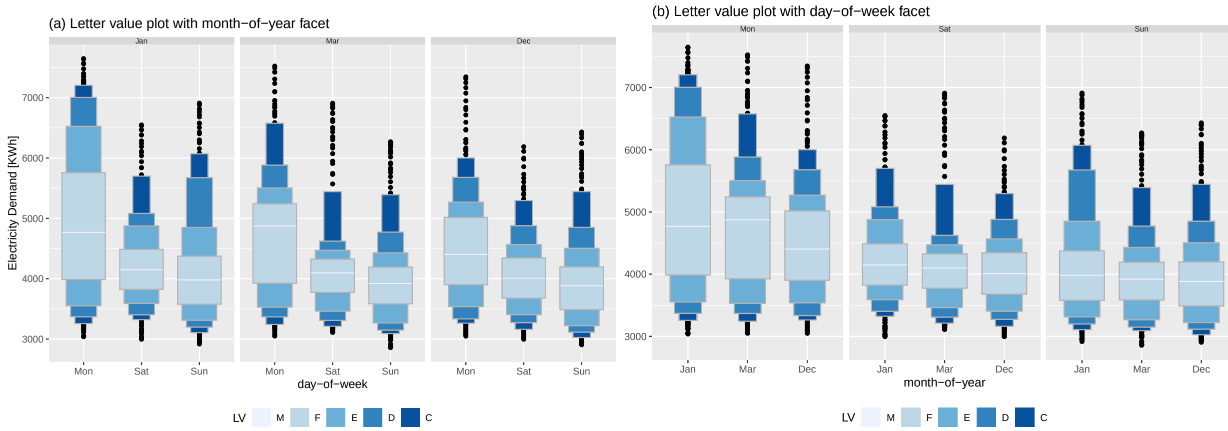

Figure 6: Letter value plot of electricity consumption data of Victoria from 2012 to 2014 (a) across day-of-week faceted by month-of-year and (b) across month-of-year faceted by day-of-week. Plots (a) and (b) both show harmonies since each month consists of all days of the week and every days of the week can occur in every month. However, Plot (a) helps comparison of different days of the week within each month and Plot (b) facilitates comparing months of the year for different days of the week.

6.2.3 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 $<C_1, C_2, v>$ with C_1 mapped to x position and C_2 to facets A_i's are placed in close proximity and each B_j 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_i's are compared against each other within each group.. With the mapping of $C1$ and C_2 reversed, emphasis will shift to different behavior of the variables. Figure 6 (a) shows the letter value plot across days of the week faceted by months like January, March and December and Figure 6 (b) shows letter value of across months faceted by days. The first one helps us to compare different days of the weeks within a month and the second one helps to compare different months for each day of the week.

6.3 Number of observations and statistical transformations

Even with harmonies, visualizing distributions can be misleading if statistical transformations are performed on rarely occuring categories (5.0.2) or unevenly distributed events.

6.3.1 Unevenly distributed events

Even when there are no rarely occuring 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 based on density should be used with caution as sample sizes would directly affect both the variance and consequently the confidence interval of the estimators.

7 Applications

7.1 Smart meter data of Australia

Smart meters provide large quantities of measurements on energy usage for households across Australia. One of the customer trial (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 look at the energy consumption across different cyclic granularities in a systematic way to identify different behavioral patterns.

7.1.1 Data structure for visualization:

The data structure for a sample of 50 households from that trial is depicted in where the variable `reading_datetime`, `customer_id` and `general_supply_kwh` denote the index, key and measured variable of the tsibble.

customer_id	reading_datetime	general_supply_kwh
10006414	2012-02-10 08:00:00	0.141
10006414	2012-02-10 08:30:00	0.088
10006414	2012-02-10 09:00:00	0.078
10006414	2012-02-10 09:30:00	0.151
10006414	2012-02-10 10:00:00	0.146

7.1.2 Cyclic granularities search and computation:

R package **gravitas** (Gupta et al. 2019) is used to facilitate the systematic exploration here. While trying to explore the energy behavior of these customers systematically across cyclic time granularities, the first thing we should have at our disposal is to know which all cyclic time granularities we can look at exhaustively. These cyclic granularities are a function of the index variable and are computed based on if they are circular , pseudo-circular or aperiodic . If we consider conventional time deconstructions for a Gregorian calendar (second, minute, half-hour, hour, day, week, month, year), the following cyclic time granularities can be considered for this analysis. The interval of this tsibble is 30 minutes, and hence the temporal granularities may range from half-hour to year. The cyclic granularities which could be considered are "hhour_hour", "hhour_day", "hhour_week", "hhour_month", "hhour_year", "hour_day", "hour_week", "hour_month", "hour_year", "day_week", "day_month" , "day_year", "week_month", "week_year" and "month_year". These options are too many and hence the most coarse temporal unit can be set to a "month" and some intermediate temporal units which might not be pertinent to the analysis like hhour could be dropped. Then we will be left with six cyclic granularites namely "hour_day", "hour_week", "hour_month", "day_week", "day_month" and "week_month". Now that we have a list of cyclic granularities to look at, we should be able to compute them from the data using Sections 3.1, 3.2 and 3.3.

7.1.3 Screening and visualizing harmonies:

From the search list, we found six cyclic granularities for which we would like to derive insights of energy behavior. Given the data structure $<C1\ C2$, `general_supply_kwh`$>$, each of those six cyclic granularities can either be mapped to x-axis or to facet. Thus the problem is equivalent to taking 2 granularities at a time from 6, which essentially is equivalent to having 30 data subsets for visualization. However, harmony/clash pairs can be identified among those 30 possibilities to determine feasibility of plotting any pairs together. We are left with 13 harmonies pair, each of which can be plotted together to look at the energy behavior from different perspectives.

facet_variable	x_variable	facet_levels	x_levels
day_week	hour_day	7	24
day_month	hour_day	31	24
week_month	hour_day	5	24
day_month	hour_week	31	168
week_month	hour_week	5	168
day_week	hour_month	7	744
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
hour_day	week_month	24	5
day_week	week_month	7	5

In Figure 7, the distribution of energy consumption is plotted across the harmony pair (wknd_wday, hour_day) through an area quantile plot. The black line is the median, whereas the pink 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 skewed with 1st, 10th and 25th percentile lying very 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 **??**. There is additional information that we can derive looking at the distribution. 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. Also hours from 7 to 13 look most volatile. 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 **??**, it can be seen that median energy consumption for the early morning hours is extremely high for weekdays compared to weekends. Also, outliers are more prominent in the latter part of the day. All of these indicates that looking at different distribution plots or changing the mapping might shed lights on different aspect of the energy behavior for the same customer.

If the data for all keys are visualized together, it might lead to Simpson's paradox, which occurs when one `customer_id` shows a particular behavior, but this behavior is reversed when all keys are combined together. This is also intuitive 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 multiple keys together could be to first group them basis time series or demographic features and then look at their energy behavior. This is beyond the scope of the current work.

This case study shows systematic exploration of energy behavior for a random household to gain some insights on periodic behavior of the households. First, it helps us to find the list of cyclic granularities to look at, then shrinks the number of possible visualizations by identifying harmonies, visualize a harmony pair and shows the effect of different distribution plots or reverse mapping.

7.2 T20 cricket data of Indian Premiere League

The method is not only restricted to temporal data. We provide an example of cricket to illustrate how this can be generalized to other applications. Although there is no conventional time component in cricket, each ball can be thought to represent an ordering from

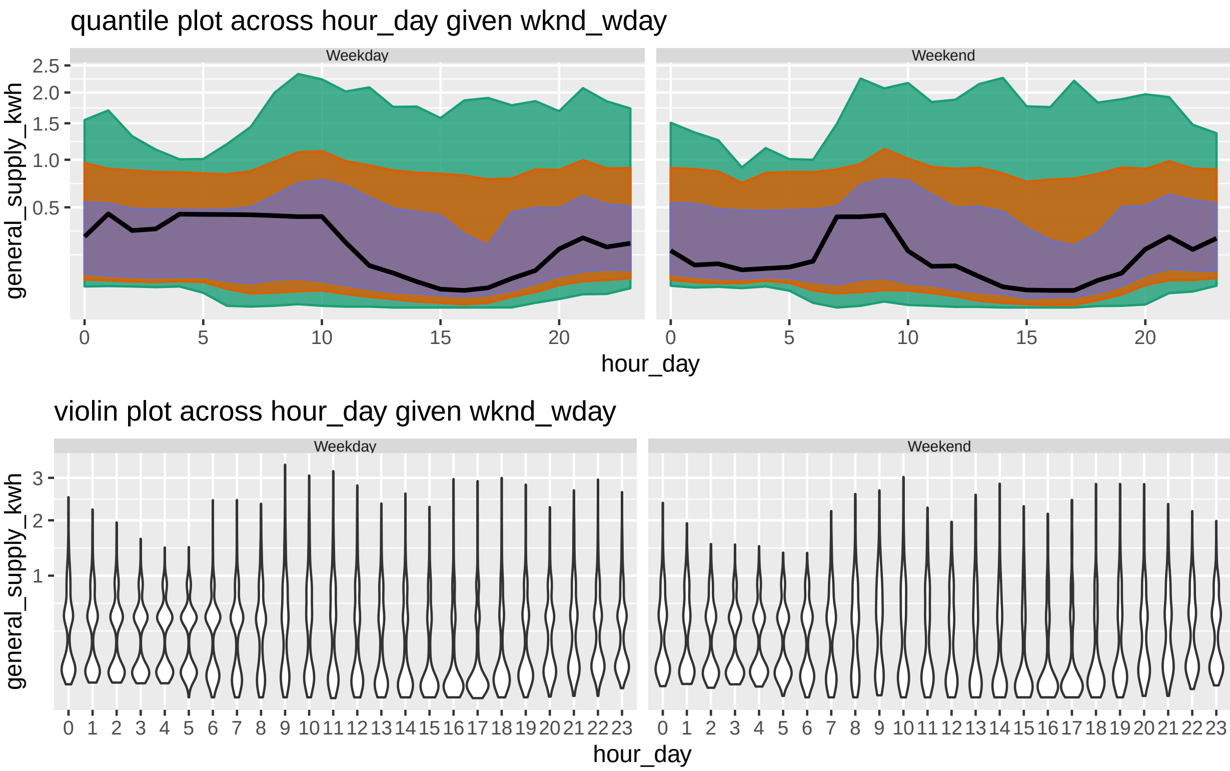

Figure 7: (a) Area quantile and (b) violin plot of energy consumption across hours of the day faceted by weekday/weekend for customer id: 10017936. Plot (a) shows that the customer typically consumes more in the morning hours following which their consumption is low till late afternoon hours. Median consumption for the early morning hours starts higher and continue to remain high for longer during weekdays compared to weekdend. Plot (b) shows early morning hours are bimodal implying both high and low consumptions are probable. Bimodality can be caused by difference in behavior across different seasons. Volatility is highest from 7 to 13 hours as evident from the narrow violins.

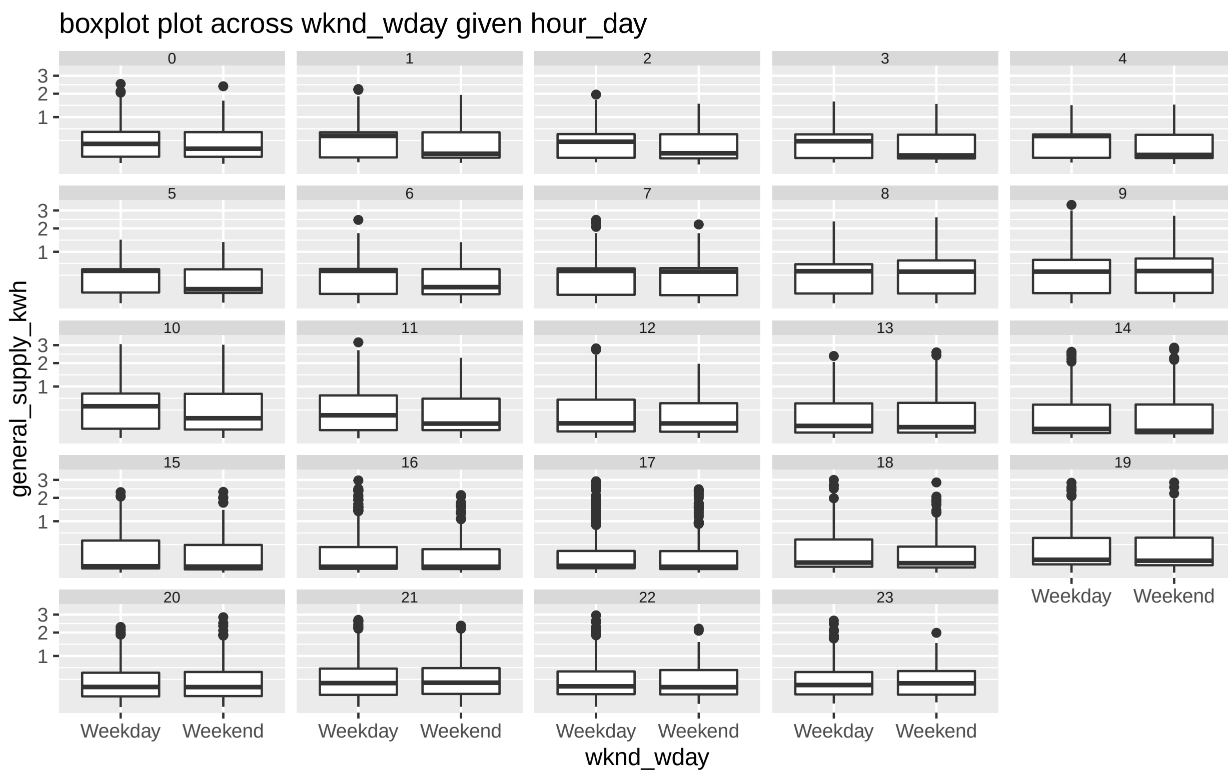

Figure 8: Boxplot of energy consumption across hours of the day faceted by weekday/weekend for customer id: 10017936. The change in mapping of the cyclic granularities leads to easier comparison within weekdays and weekends for each hour. Median consumption till 6am is higher for weekends compared to weekdays although the quartiles look similar. Outliers appear more during the end of the day implying more uncertain behavior in latter hours of the day compared to early morning hours.

past to future with the game progressing forward with each ball. In a Twenty20 game, 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. An idea of cyclic granularities can be conceived with this hierarchy in mind. Examples may include ball of the over, over of the innings or ball of the innings. Although most of these cyclic granularities are circular in theory by design of the hierarchy, in reality these may be aperiodic. For example, in most cases an over will consist of 6 balls with some exceptions like wide balls or when an inning finishes before the over finishes. Thus, the cyclic granularity ball-of-the-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 inning 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 the following:

G	C	P
over	over-of-inning	20
inning	inning-of-match	2
match	match-of-season	aperiodic
season	1	1

However, even if the situation is not periodic and a similar hierarchy can not be formed, it can be interesting to visualize the distribution of a measured variable across these cyclic granularities to throw light on the periodic behavior of a non-temporal dataset similar to any temporal dataset. There are many interesting questions that can possibly be answered with such a data set irrespective of the type of cyclic granularities.

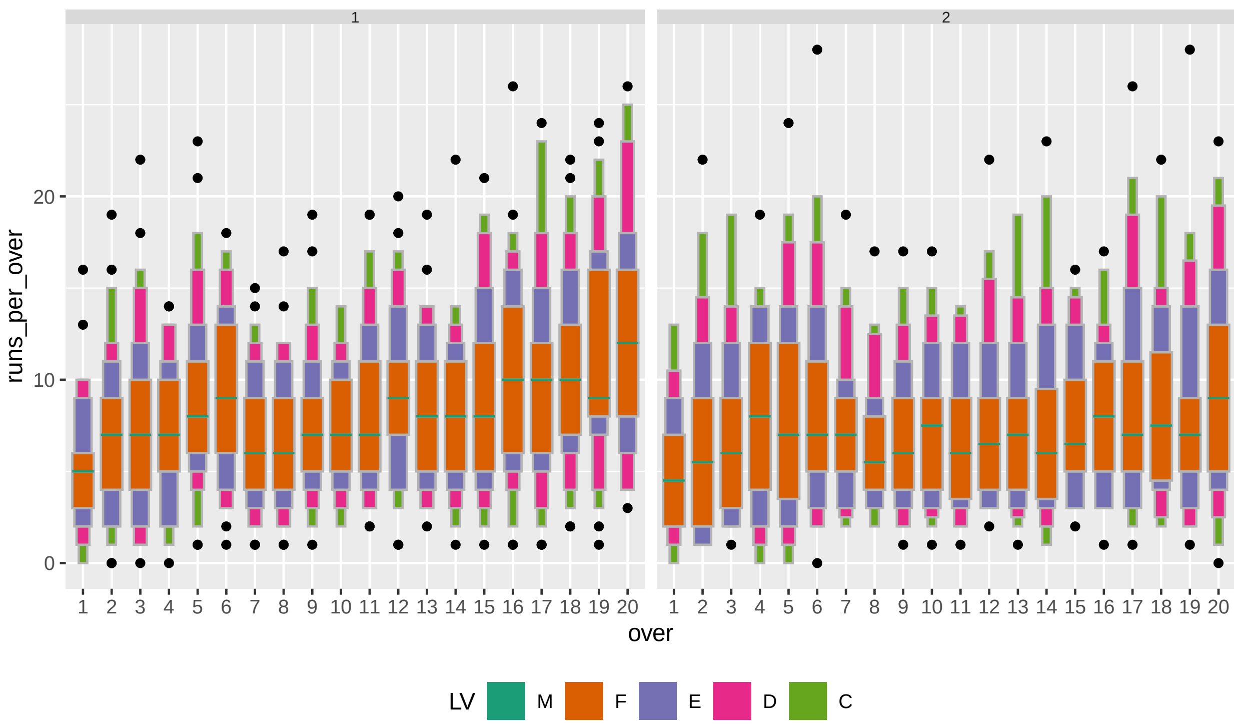

Figure 9: Letter value plot of runs per over across overs of the inning faceted by innings of the match. No upward shift in runs in the second innings like that in the first implying teams are more vulnerable to score more in the first innings as they approach the end of the inning.

Q1: How total runs vary depending on if a team bats first or second?

Mumbai Indians (MI) and Chennai Super kings (CSK) appeared in final playoffs from 2010 to 2015. It would be interesting to take their example in order to dive deeper into this question. From Figure 9, it can be observed that there is no clear upward shift in median and quartile deviation of runs in the second innings as compared to the first innings. This might indicate that players are vulnerable to score through each over and make more runs as they approach towards the end of the first innings, whereas if they bat in the second innings they have a target in mind and are not as vulnerable. To answer Q1, winning teams like CSK and MI have indeed different strategies when it comes to batting in the first or second innings.

Q2: Is runs per over set to reduce in subsequent over if fielding is good in the previous over?

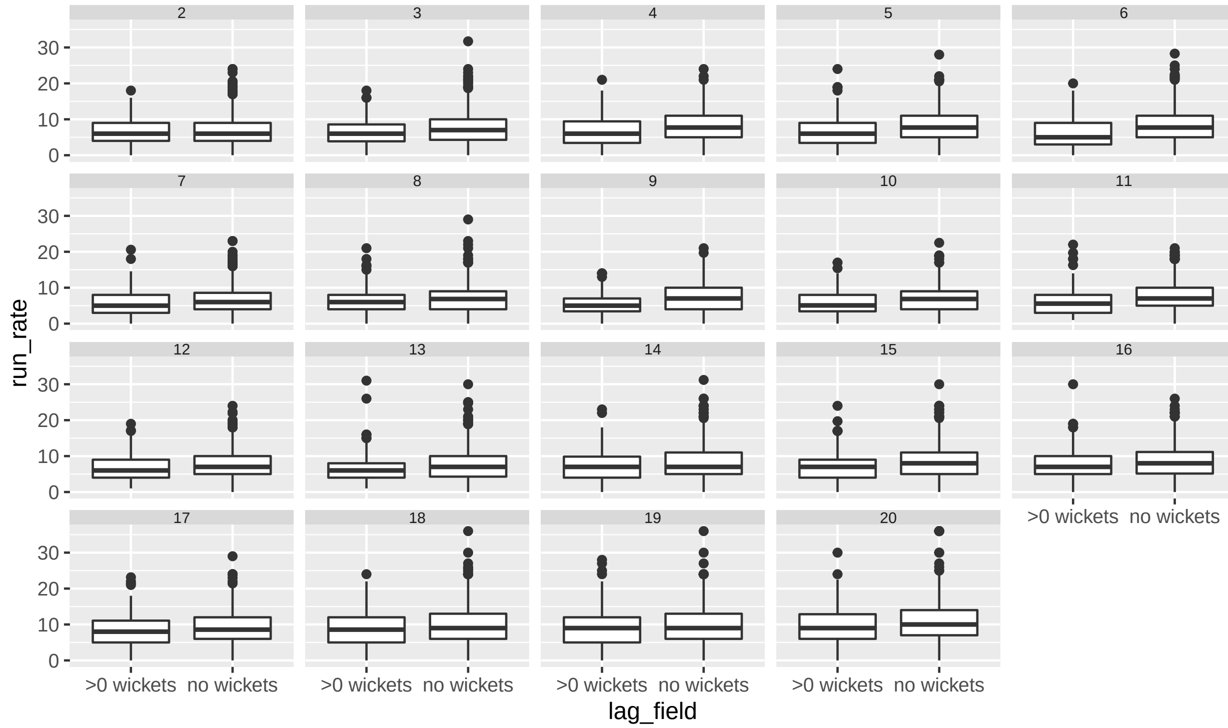

Figure 10: Box plots showing distribution of run rate across overs of the innings and (lagged) wickets dismissed due to fielding. This shows at least one wicket in the previous over leads to downward shift of median run rate and quartile spread in the subsequent over for most overs.

For establishing that the fielder fielded well in a particular over, we can see how many catches and run outs were made in that particular over. Dismissals in previous over can lead to a batsman being either defensive or play more vulnerable. Figure 10 shows that no dismissals in the previous over will lead to a higher median and quartile spread of run rates compared to the case when there has been at least one dismissals.

In Q2, wickets per over are considered as aperiodic cyclic granularities with wickets as aperiodic linear granularities. 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.

8 Discussion

Exploratory data analysis is iteratively finding and summarizing patterns. With temporal data available at more finer scales, exploring periodicity can become overwhelming with so many possible cyclic granularities that could be generated for a context. This work provides a framework to systematically explore distribution of an observed variable across two cyclic time granularities by creating any cyclic granularity, choosing a list of harmonies and thereby recommending possible distribution plots for effective visualization based on relationship and levels of the cyclic granularities.

A missing piece is to enable user-defined temporal calendars and compute multiple order-up quasi-circular granularities from the hierarchy table. Also, computation of cyclic aperiodic granularities 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 functionalities can be imported in the gravitas package to accommodate for aperiodic cyclic granularities.

9 Software implementation

The ideas presented in this article have been implemented in the open-source R (R Core Team 2019) package gravitas (Gupta et al. 2019), available from CRAN.

Acknowledgements

The authors would like to thank the cohort at NUMBATS, Monash University for sharing their wisdom and experience of developing R packages, Dr. Peter Toscas from Data61 CSIRO for providing useful inputs on improving the analysis of smart meter application. The package `gravitas` was built under Google Summer of Code, 2019. More details about the package can be found on the package website `https://sayani07.github.io/gravitas/`. This article was created with knitr (Xie 2015, Xie (2020)) and R Markdown (Xie et al. 2018, Allaire et al. (2020)). The project's Github repository `https://github.com/Sayani07/paper-gravitas` contains all materials required to reproduce

this article and the code is also available online in the supplemental materials.

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. E., Evans, W. S., Lin, H. & Snodgrass, R. T. (2000), 'Efficiently supporting temporal granularities', *IEEE Trans. Knowl. Data Eng.* **12**(4), 568–587.

G Grolemund, H. W. (2011), 'Dates and times made easy with lubridate', *Journal of Statistical Software* .

Goodwin, S And Dykes, (2012), 'Visualising variations in household energy consumption', *IEEE Conference on Visual Analytics Science and Technology (VAST)* .

Gupta, S., Hyndman, R., Cook, D. & Unwin, A. (2019), *gravitas: Explore Probability Distributions for Bivariate Temporal Granularities*. R package version 0.1.0.
URL: *https://CRAN.R-project.org/package=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*, Cambridge University Press.

Tukey, J. W. (1977), *Exploratory data analysis*, Vol. 2, Reading, Mass.

Wang, E., Cook, D. & Hyndman, R. J. (2018), 'Calendar-based graphics for visualizing people's daily schedules'.

Wang, E., Cook, D. & Hyndman, R. J. (2019), 'A new tidy data structure to support exploration and modeling of temporal data', *Journal of Computational and Graphical Statistics* .

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

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

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