# 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

February 10, 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 need to 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 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 the 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 like monthly, quarterly or annually. With recent advancement in technology, more and more data are recorded at much finer temporal scales. Energy consumption is collected every half an hour, while energy supply is 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 like 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 temporal 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 all 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 well and are capable of checking the weekends or special days. Any sub-daily resolution temporal data can also be displayed using this type of faceting (Wickham 2016) with days of the week, month of the year and another sub-daily deconstruction of time. But calendar effects are not restricted to conventional day-of-week, month-of-year ways of deconstructing time. There can be several classes of time deconstructions, viz. based on the arrangement (linear vs. cyclic) or hierarchical order of the calendar. Linear time granularities respect the linear progression of time such as hours, days, weeks and months. One of the first attempts of characterizing these granularities occur in Bettini et al. (1998). The definitions and rules defined are inadequate to reflect periodicities in time. Hence, there is a need to define cyclic time granularities in a different approach, which can be useful in visualizing periodic behavior. Cyclic time granularities

can be circular, quasi-circular or aperiodic. Examples of circular can be hour of the day, day of the week, that of quasi-circular can be day of the month and examples of aperiodic granularities can be public or 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 like second of the minute or multiple-order-up granularities like second of the hour. lubridate (G Grolemund 2011) creates easy access and manipulation of common date-time objects. But most accessor functions are limited to single-order-up granularities.

The motivation for this work comes from the desire to provide methods to better understand large quantities of measurements on energy usage reported by smart meters in 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, total kWh used each half hour by state, for example, because energy companies need to understand maximum loads that they will have to plan ahead to accommodate. But studying overall energy use hides the distributions of usage at finer scales, and making it more difficult to find solutions to improve energy efficiency. We propose that the analysis of smart meter data can be benefited 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 which relates to any data that needs to be analyzed for different periodicities.

This work provides tools for systematically exploring bivariate granularities within the tidy workflow through the following:

- Formal characterization of cyclic granularities
- Facilitate manipulation of single and multiple order-up time granularities through

cyclic calendar algebra

- Checking feasibility of creating plots or drawing inferences for any two cyclic granularities
- Recommend prospective probability distributions for exploring distributions of univariate dependent variable across pair of granularities

The remainder of the paper is organized as follows: Section 2 details the background of linear granularities in depth and briefly explain 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 cyclic calendar algebra for computing cyclic time granularities. Section 5 discusses the data structure for exploring distribution of the associated time series across pair 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

Often we partition time into months, weeks or days to relate it to data. Such discrete abstractions of time can be thought of as time granularities (Aigner et al. 2011). Time granularities are **linear** if they respect the linear progression of time. Examples include hours, days, weeks and months.

## 2.1 Definitions and relationships

There has been several attempts to provide the framework for formally characterizing time-granularities. One of the first attempts occur in Bettini et al. (1998) with the help of the 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* $T$*.*

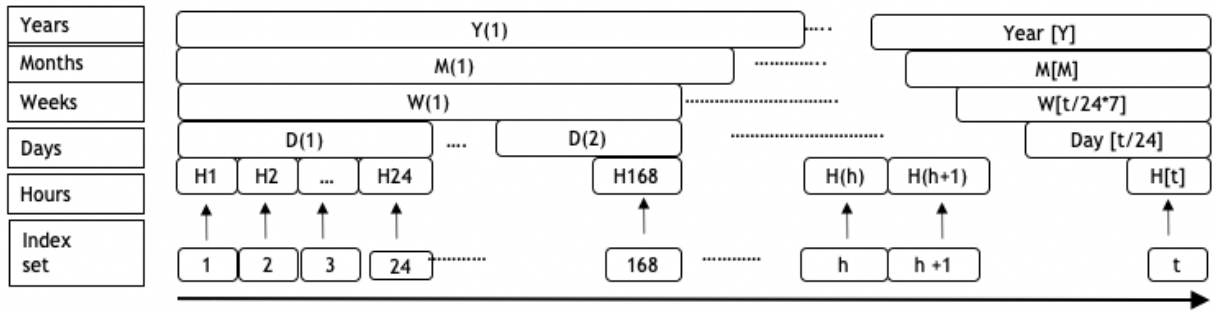


Figure 1: The time domain distributed as linear granularities

A time domain can be **discrete** (if there is unique predecessor and successor for every element except for the first and last one in the time domain), or it can be **dense** (if it is an infinite set). A time domain is assumed to be discrete for the purpose of our discussion.

**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.*

**Discussion:** 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 shows the implication of this condition. It shows the linear granularities hours, days, weeks, months and years, each of which are unidirectional in nature and arranged from past to future. If we consider the chronon (see Aigner et al. (2011)) as hourly, the time domain with T hours will have $\lfloor T/24 \rfloor$ days, $\lfloor T/(24 * 7) \rfloor$ weeks and so on. Each granule of the linear granularities shown are represented through a box.

(Bettini et al. 1998) talks about the relationships of linear time granularities and structure of a calendar and also relates them to the notion of periodicity in time.

**Definition 4** *(finer than): 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 $G(i) \subset H(j)$.*

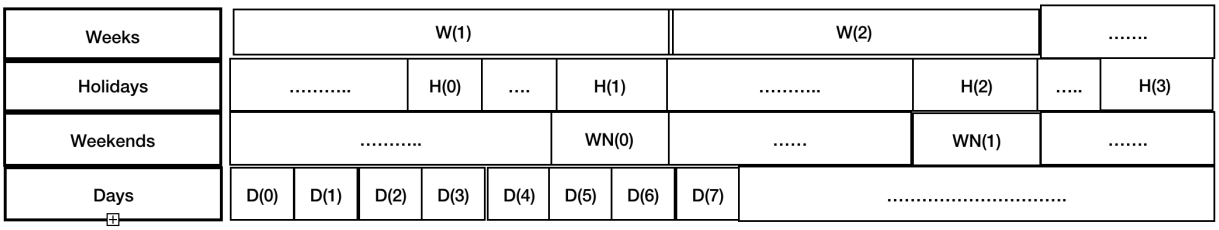


Figure 2: Illustrations of groups into and finer than relationships. Weekends are finer than weeks but do not group into weeks. And weekends neither groups into or finer than public holidays

.

**Definition 5** *(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*

$$H(j) = \bigcup_{i \in S} G(i) \tag{1}$$

**Discussion:** $G \trianglelefteq H$ if each granule H(j) is the union of some granules of G. $G \preceq H$ if every granule in G is a subset of some granule in H. In Figure 2, both $day \trianglelefteq week$ and $day \preceq week$ hold true. However, weekends are finer than weeks but do not group into weeks. And weekends neither groups into or finer than public holidays. Consider another example where $day \trianglelefteq month$. This relationship is not fully described until we associate it with 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 6** *(groups periodically): A granularity H is periodical with respect to a granularity G if (1) $G \trianglelefteq H$, and (2) there exist $R, P \epsilon Z+$, where R is less than the number of granules of H, such that for all $i \epsilon Z$, if $H(i) = \bigcup_{j \in S} G(j)$ and $H(i+R) \neq \phi$ then $H(i+R) = \bigcup_{j \in S} G(j+P)$.*

**Discussion:** A granularity H which is periodical with respect to G is specified by: (i) the R sets of indexes of G, $S_0, ..., S_{R-1}$ describing the granules of H within one period; (ii)

the value of P; (iii) the indexes of first and last granules in H, if their value is not infinite. Then, if $S_0, ..., S_{R-1}$ are the sets of indexes of G describing $H(0), ..., H(R-1)$, respectively, then the description of an arbitrary granule H(j) is given by: $\bigcup_{i \in S_{j \mod R}} G(P * \lfloor j/R \rfloor + i)$. 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).

**Definition 7** *(groups quasi-periodically): A granularity H is quasi-periodical with respect to a granularity G if (1) $G \trianglelefteq H$, and (2) there exist a set of intervals E1,...,Ez (the granularity exceptions) and positive integers R,P, where R is less than the minimum of the number of granules of H between any 2 exceptions, such that for all $i \epsilon Z$, if $H(i) = \bigcup_{k \in [0,k]} G(j_r)$ and $H(i+R) \neq \phi$ and $i + R < min(E)$, where E is the closest existing exception after $H(i)^2$, then $H(i+R) = \bigcup_{k \in [0,k]} G(j_r + P)$.*

**Discussion:** Intuitively, the definition requires that all granules of H within the span of time between two exceptions have the same periodical behavior, characterized by $R$ and $P$.

**Definition 8** *(bottom granularity): Given a granularity order relationship g-rel and a set of granularities having the same time domain, a granularity B in the set is a bottom granularity with respect to g-rel, if B g-rel H for each granularity H in the set.*

**Discussion:** Given the set of all granularities defined over the time domain $(Z; <)$, and the granularity relationship $\trianglelefteq$ (groups into), the granularity mapping each index into the corresponding instant (same integer number as the index) is a bottom granularity with respect to $\trianglelefteq$.

## 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 from the bottom one or recursively, from those already generated. The calendar algebra consists of two kinds of operations: grouping-oriented and granule-oriented

operations. The grouping-oriented operations combine certain granules of a granularity together to form the granules of the new granularity. Example can be to consider a calendar with only two linear granularities "minute" and "hour" and "hour" is generated by grouping every 60 "minutes". The granule-oriented operations do not change the granules of a granularity, but rather make choices of which granules should remain in the new granularity. For example, one can choose to look at the granularity "Monday" and hence select only "Monday" from linear granularity "day".

Some relevant grouping oriented operations are discussed, which will be used in Section 3 to define circular and aperiodic granularities.

- **The grouping operation** : Let $G1$ be a full-integer labeled granularity, and $m$ a positive integer. The grouping operation $Group_m(G)$ generates a new granularity $G2$, by partitioning the granules of $G$ into m-granule groups and making each group a granule of the resulting granularity. More precisely, $G2 = Group_m(G1)$ is the full-integer labeled granularity such that for each integer $i$, $G2(i) = \bigcup_{j=(i-1)m+1}^{im} G1(j)$.

**Examples** :

- $minute = Group_{60}(second)$
- $hour = Group_{60}(minute)$,
- $day = Group_{24}(hour)$,
- $week = Group_7(day)$,

where, second(1) would start a minute and likewise.

- **The altering-tick operation** : Let G1, G2 be full-integer labeled granularities, and $l$, $k$, $m$ integers, where G2 partitions G1, and $1 \leq l \leq m$. The altering-tick operation $Alter^m_{l,k}(G2, G1)$ generates a new full-integer labeled granularity by periodically expanding or shrinking granules of G1 in terms of granules of G2. The altering-tick operation modifies the granules of G1 so that the $l$th granule of each group has $|k|$ additional (or fewer when k $< 0$) granules of G2.

**Example** :

$$pseudomonth = Alter^{12}_{11,-1}(day, Alter^{12}_{9,-1}(day,$$
$$Alter^{12}_{6,-1}(day, Alter^{12}_{4,-1)}(day,$$
$$Alter^{12}_{(2,-3)}(day, Group_{31}(day))))))$$

, where the granularity pseudomonth is generated 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 grouping operations and granule oriented operations, the readers are recommended to see Ning et al. (2002).

# 3 Cyclic time granularities

We propose a formalism of cyclic time granularities through the tsibble (Wang et al. 2019) framework of organizing temporal data. A tsibble consists of an index, key and measured variables. An index is a variable with inherent ordering from past to present and key is a set of variables that define observational units over time. In a tsibble each observation should be uniquely identified by index and key.

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. Lining up with the the assumption in Bettini & De Sibi (2000), that all linear granularities can be generated from the bottom granularity by calendar algebraic operations, we assume that a bottom granularity exists. In a cyclic organization of time, the domain is composed of a set of recurring time values, where any time value can be preceded and succeeded by another time value (for e.g Monday comes before Wednesday but Monday also succeeds Wednesday). Hence intuitively, cyclic granularities are additional abstractions of time which are not linear and hence the index order of linear granularities needs to be manipulated so that it leads to repetitive categorization of time.

Cyclic time granularities which accommodate for periodicities can be constructed by

relating two linear granularities. The mappings between these linear granularities can be regular or irregular. For example, a regular mapping exists between minutes and hours, where 60 minutes always add up to 1 hour. On contrary, an irregular mapping exists between days and months, since one might have days ranging from 28 to 31. To elaborate more, mappings will be regular if, for example, linear granularities are formed by grouping operation. Grouping operations combine fixed granules of a linear granularity to form a new granule of the resulting linear granularity. However, if granularities are formed by the altering-tick operation, for example, granules of the resulting granularity is composed of different number of granules of the root granularity. Cyclic time granularities are referred to as **circular** if mappings between linear time granularities are regular and **quasi-circular** if the same is 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. The formalism would differ for circular and quasi-circular granularities although both accommodate for periodicities in data.

## 3.1 Circular

**Definition 9** *(circular granularity): A circular granularity $C_{B,G}$ relates a linear granularity $G$ to the bottom granularity $B$, if*

$$C_{B,G}(z) = L(z \mod P(B,G)) \quad \forall z \in \mathbb{Z}_{\geq 0} \tag{2}$$

*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 $l \in \{0, 1, \ldots, (P-1)\}$.*

**Example**: Example showing circular granularity relating two linear granularities "day" and "week" is visually depicted in a series of slots in the Figure 3. Each granule is represented by a box. The diagram also illustrates that the granules that overlap share elements from the underlying time domain. The first slot in the diagram shows the index set. The bottom granularity, represented by the index of the tsibble considered is "day". "day"

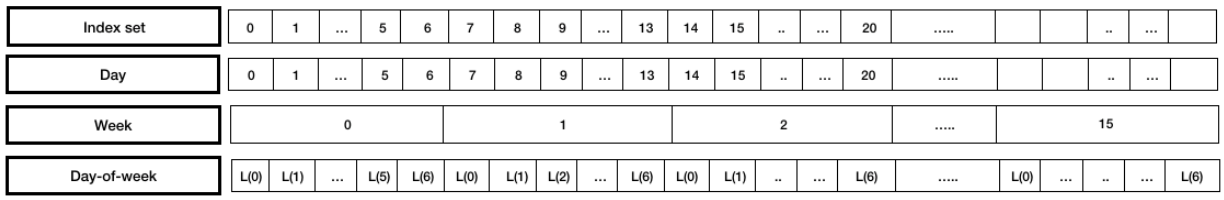


Figure 3: Circular granularity day-of-week

and "week" are two linear granularities with $P = 7$. The circular granularity $C_{day,week}$ representing day-of-week will thus consist of pattern {L(0), L(1), L(2), L(3), L(4), L(5), L(6)} which repeats itself after each period length.

**Discussion**: Note that each circular granularity can use different label mappings. In general, the set of labels will be a set of strings that is more descriptive than the index and used to identify a categorization of the circular granularity. However, the labels can coincide with indexes in which case integers are directly used to refer to categorizations of the circular granularity. Hence, the label mapping $L$ in Figure 3 can be defined as $L : (0, 1, 2, \ldots, 6) \longmapsto (Sun, Mon, \ldots, Sat)$ or $L : \{0, 1, 2, \ldots, 6\} \longmapsto \{Sunday, Monday, \ldots, Saturday\}$ or $L : \{0, 1, 2, \ldots, 6\} \longmapsto \{0, 1, \ldots, 6\}$ depending on the context.

In general, any circular granularity relating two linear granularity, none of which are bottom granularity can be expressed as $C_{(G,H)}(z) = L(\lfloor z/P(B,G) \rfloor \mod P(G,H))$, 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. It is possible that none of the two linear granularities are bottom granularities. But the representation of the resultant circular granularity will be a function of the index set. The bottom granularity is assumed to be minutes.

## 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

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

Table 1: Illustrative circular granularities with time index in minutes

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.

| | | |
|---|---|---|
| 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

**Definition 10** *(quasi-circular granularity): A quasi-circular granularity $Q_{B,G'}$ that relates linear granularities $G'$ and bottom granularity $B$, if*

$$Q_{B,G'}(z) = L(z - \sum_{w=0}^{k-1} |T_w \mod R'|), \quad z \in T_k \tag{3}$$

*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 4. 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. This implies that $G'$ is made up by shrinking every 1st granule of $Group_3(B)$ by 1 granule and $H'$ is made up of shrinking every 1st granule of $Group_3(B)$ by 2 granules. Number of granules of $G'$ and $H'$ in each period of $B$ is 2 but the number of granules of $B$ in each of those granules are different. $Q_{B,G'}$ 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 4 and Equation 2.

$$
\begin{aligned}
Q_{B,G'}(8) &= L(8 - \sum_{w=0}^{3-1} |T_{w \mod 2}|), \quad since \quad 8 \in T_3 \\
&= L(8 - \sum_{w=0}^{2} |T_{w \mod 2}|) \\
&= L(8 - |T_{0 \mod 2}| - |T_{1 \mod 2}| - |T_{2 \mod 2}|) \\
&= L(8 - 2 * |T_0| - |T_1|) \\
&= L(8 - 2 * 2 - 3) \\
&= L(1)
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
Q_{B,H'}(10) &= L(10 - \sum_{w=0}^{1-1} |T_{w \mod 2}|), \quad since \quad 10 \in T_1 \\
&= L(10 - \sum_{w=0}^{0} |T_{w \mod 2}|) \\
&= L(10 - |T_{0 \mod 2}|) \\
&= L(10 - |T_0|) \\
&= L(10 - 7) \\
&= L(3)
\end{aligned}
\tag{5}
$$

**Discussion**: If linear granularity $G'$ is periodical with respect to $B$ with irregular mapping, then the following holds true:

- $B \trianglelefteq G'$

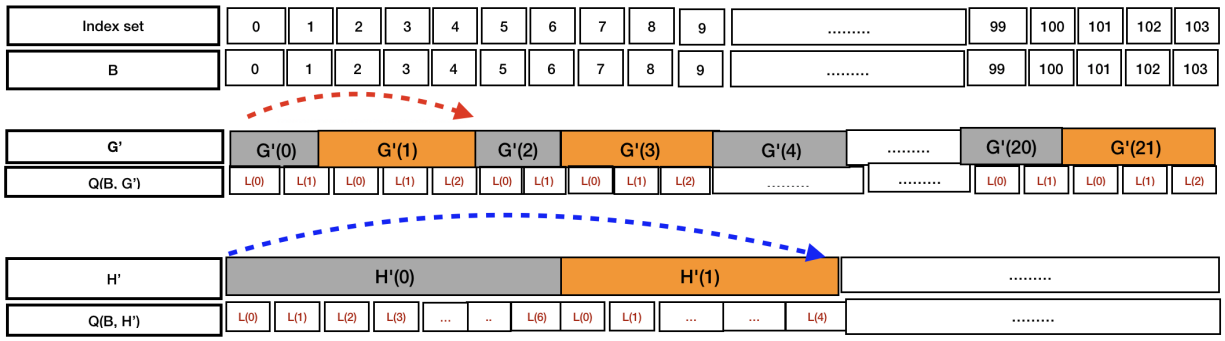


Figure 4: Quasi-circular granularities $Q(B, G')$ and $Q(B, H')$ relating bottom granularity $B$ and linear granularities $G'$ and $H'$ respectively.

- there exist $R', P' \epsilon Z_+$ such that if $G'(w) = \bigcup_{z \in T_w} B(z)$ then

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

from definition 6

and, number of granules of $B$ in each granule of $G'$ is not a constant (irregular mapping)

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 $|T_w| = |T_{w \mod R'}|$ since every $w^{th}$ and $(w + R')^{th}$ 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 $(k-1)^{th}$ granule of $G'$. Since $|T_w| = |T_{w \mod R'}|$, the number of granules of $B$ till the $(k-1)^{th}$ granule of $G'$ becomes $\sum_{w=0}^{k-1} |T_{w \mod R'}|$ in Definition 10.

It can be noted here that if a linear granularity $G'$ is quasi-periodic (Definition 7) with respect to $B$, then Equation 3 can be modified as follows to account for exceptions $E = [e_{begin}, e_{end}]$ (by Definition 7).

$$A_{B,G'}(z) = L(z - \sum_{w=0}^{k-1} |T_{w \mod R'}| - \sum_{u=0}^{e_k} |E_u|) \tag{6}$$

for $z \in T_k$ and $e_k$ is the number of exceptions in $\bigcup_{w=0}^{k} T_w$ and $E_u = [e_{begin}(u), e_{end}(u)]$.

## 3.3 Aperiodic

Periodic and Quasi-periodic behavior can be defined by a repeating pattern, except for a finite number of granules that can be seen as discontinuity points in the granularity in case of quasi-periodic behavior. 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. U.S labour day is the first Monday in September and U.S. Memorial Day is the last day in May. 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 that might cover the first and third Monday of the month between June and October, except for state holidays. 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 $M_i \; \forall\{i \in 1, 2, \ldots, n\}$, that is, none of them can be expressed as periodic or quasi-periodic with respect to the bottom granularity. Further assume that $B \trianglelefteq M_i \; \forall\{i \in 1, 2, \ldots, n\}$. Then according to 5, for each index $j$ there exists a (possibly infinite) subset $T_{\{i_j\}}$ of the integers such that

$$M_i(j) = \bigcup_{z \in T_{i_j}} B(z)$$

and suppose $M = \bigcup_{i=1}^{n} M_i$ is formed by collecting the granules of $\{M_1, M_2, \ldots, M_n\}$. Here, index $\{i_j\}$ stands for the $j^{th}$ granule of the $i^{th}$ linear aperiodic granularity.

**Definition 11** *(aperiodic cyclic granularity): An aperiodic cyclic granularity $A_{B,M}$ relates aperiodic linear granularity $M$ and bottom granularity $B$, if*

$$\begin{aligned} A_{B,M}(z) &= L(i), \quad z \in T_{i_j} \quad \textit{and} \quad \forall i \in 1, 2, \ldots, n \\ &= L(0), \quad \textit{otherwise} \end{aligned} \tag{7}$$

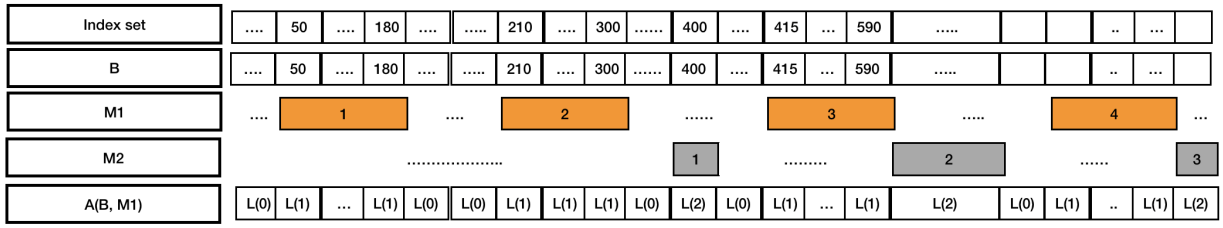


Figure 5: Aperiodic cyclic granularity $A(B, M1)$ relating aperiodic linear granularity $M1$ and bottom granularity $B$

*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* $M_i(j)$ *such that* $M_i(j) = \bigcup_{z \in T_{i_j}} B(z)$, $M = \bigcup_{i=1}^{n} M_i$.

**Example**: Figure 5 shows two aperiodic linear granularity $M_1$ and $M_2$. Each granule is represented by a box. The diagram also illustrates that the granules that overlap share elements from the underlying time domain. 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 Labour day and M can be conceived as the aperiodic linear granularity denoting any holidays. The number of categories obtained coincides with the number of aperiodic linear granularities considered and an additional category representing no event.

# 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. This is similar to calendar algebra in Ning et al. (2002), where linear granularities are generated from the bottom linear granularity or from those that are already generated. Since, our method caters to computation of cyclic granularities, we shall refer to it as

"cyclic calendar algebra".

The cyclic calendar algebra consists of broadly two kinds of operations:
(1) **single-to-multiple:** representation of multiple-order-up cyclic granularities from single order-up cyclic granularities
(2) **multiple-to-single:** representation of single-order-up cyclic granularities from multiple order-up cyclic granularities

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.

**Hierarchy table** Let $H_n : (G, C, P)$ be a hierarchy model containing n linear granularities. $G_l$ represents the linear granularity of order l and $P(l, m)$ represents the period length of $G_l$ with respect to $G_m$ and $C_{G(l),G(m)}$ represents the cyclic granularity that relates linear granularity of order l and m, $\forall l, m \in 1, 2, \ldots, n$ and $l < m$.

**Order** of a linear granularity can be comprehended as the level of graininess associated with a linear granularity. For example, if we consider two linear granularities $G$ and $H$, such that $G$ is finer than or groups into $H$, then $H$ is of higher order than $G$. In any hierarchy table, linear granularities are arranged from lowest to highest order of linear granularities.

We refer to granularities which are nested within multiple levels of the hierarchy table as multiple-order-up granularities 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. All calendars fall under three types - solar, lunar or lunisolar/solilunar but the day is the basic unit of time underlying all 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 can be defined. For example, in Mayan calendar, one day was referred to as 1 kin and the calendar was structured as follows:

- 1 kin = 1 day

- 1 uinal = 20 kin
- 1 tun = 18 uinal
- 1 katun = 20 tun
- 1 baktun = 20 katun

Thus, the hierarchy table for the Mayan calendar would look like the following:

| G | C | P |
|---|---|---|
| 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 arbitrarily, depending on the context.

$$
\begin{aligned}
C_{(G_l,G_m)}(z) &= L(C_{G_l,G_{l+1}}(z) + P(l,l+1)(C_{(G_{l+1},G_m)}(z))) \\
&= L(C_{(G_l,G_{l+1}}(z) + P(l,l+1)[C_{G_{l+1},G_{l+2}}(z) + P(l+1,l+2)(C_{G_{l+2},G_m}(z))) \\
&= L(C_{G_l,G_{l+1}}(z) + P(l,l+1)(C_{G_{l+1},G_{l+2}}(z)) + P(l,l+1)P(l+1,l+2)(C_{G_{l+2},G_m})) \\
&= L(C_{G_l,G_{l+1}}(z) + P(l,l+1)(C_{G_{l+1},G_{l+2}}(z)) + P(l,l+2)(C_{G_{l+2},G_{l+m}}(z))) \\
&\vdots \\
&= L(\sum_{i=0}^{m-l-1} P(l,l+i)(C_{G_{l+i},G_{l+i+1}}(z)))
\end{aligned}
\tag{8}
$$

*Example:* Let us use Equation 8 to compute the multiple-order-up granularity uinal_katun for Mayan calendar.

$$
\begin{aligned}
C_{uinal,baktun}(z) &= L(C_{uinal,tun}(z) + P(uinal,tun)C_{tun,katun}(z) + C_{uinal,katun}C_{katun,baktun}(z)) \\
&= L(\lfloor z/20 \rfloor \mod 18 + 20 * \lfloor z/20 * 18 \rfloor \mod 20 + 20 * 18 * 20 \lfloor z/20 * 18 * 20 \rfloor \mod 20)
\end{aligned}
\tag{9}
$$

### 4.1.2 Circular or quasi-circular single order-up granularities

Let us revisit Gregorian calendar for addressing this case. Suppose we have a hierarchy table using some 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. The single-order-up granularity day-of-month is quasi-circular. Any single-order-up granularities which are formed by linear granularities below days are circular. Similarly, all single-order-up granularities which are formed using linear granularities with orders higher than months are also circular.

| G | C | P |
|---|---|---|
| minute | minute-of-hour | 60 |
| hour | hour-of-day | 24 |

| G | C | P |
|---|---|---|
| day | day-of-month | quasi-circular |
| month | month-of-year | 12 |
| year | 1 | 1 |

There can be three scenarios for obtaining multiple-order-granularities here:
- cyclic granularities relating two linear granularities whose orders are less than day
- cyclic granularities relating two linear granularities whose orders are more than month
- granularities relating two linear granularities with order at most day and another with order at least month

The multiple order-up cyclic granularites resulting from the first two cases are circular and has been handled in subsubsection 4.1.1. Any cyclic granularity resulting from the last case are quasi-circular. Examples might include hour-of-month or day-of-year. Let us consider the computation of $C_{hour,month}(z)$ for a $z$ such that $C_{hour,day}(z) = 12$, $C_{day,month}(z) = 20$ and $C_{month,year}(z) = 9$.

$$\begin{aligned} C_{hour,month}(z) &= L(C_{hour,day}(z) + P(hour, day) * C_{day,month}(z)) \\ &= L(12 + 24 * 20) \end{aligned} \tag{10}$$

$$C_{day,year}(z) = L(C_{day,month}(z) + \sum_{w=0}^{C_{month,year}(z)-1} (|T_w|)) \tag{11}$$

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

Clearly, the computation of multiple order-up granularities become more involved with quasi-circular granularities. $C_{hour,month}(z)$ can be represented as function of circular granularity $C_{hour,day}(z)$ and quasi-circular granularity $C_{day,month}(z)$, both of which are single order-up. However, $C_{day,year}(z)$ can not be computed only with single order-up granularities $C_{day,month}(z)$ and $C_{month,year}(z)$. It also requires the knowledge of the composition of the linear granularity *days* within *months* in an *year* for computation of $|T_w|$.

Generally speaking, to keep it simple, we consider the case of only one quasi-circular single order-up granularity in the hierarchy. Any multiple order-up quasi-circular granular-

ity can then be considered to be a function of some circular (single or multiple order-up) and a quasi-circular single order-up granularity.

For example, with $C_{l,m}(z) = f(C_{l,m'}(z), C_{m',m}(z))$, two different approaches need to be employed for computing multiple order-up granularities from single order-up ones.

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

$$C_{G_l,G_{l+2}}(z) = L(C_{G_l,G_{m'}}(z) + P(l,m')(C_{G_{m'},G_m}(z))) \tag{12}$$

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

$$C_{G_l,G_{l+2}}(z) = L(C_{G_l,G_{m'}}(z) + \sum_{w=0}^{C_{m',m}(z)-1} (|T_w|)) \tag{13}$$

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

If $C_{G_l,G_{m'}}(z)$ or $C_{G_l,G_{m'}}(z)$ are multiple order-up circular granularities, they could be computed using similar approach in Section 4.1.1.

## 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

$$C_{G_{l1},G_{m1}}(z) = C_{G_{l2},G_{m2}}(\lfloor z/k(l_2,l_1) \rfloor \mod k(m_1,m_2)) \tag{14}$$

**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 14

$$C_{tun,katun}(z) = L(\lfloor C_{uinal,baktun}(z)/18 \rfloor \mod 20) \tag{15}$$

### 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. Commonly, simple sequences of time value pairs ($< t_0, v_0 > \ldots, < t_n, v_n >$) are the basis of analysis and visualization. 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 granularities. Data can then be consolidated for different levels of granularity enabling statistical summaries of values along granularities. Since, we are interested in detection of unknown periodic behavior of data, it makes sense to support the detection of patterns by obtaining statistical summaries across cyclic time granularities. Any attempt to encode all or many cyclic granularities at the same time to develop insights on periodicity might fail or become clumsy. Instead, the big problem is broken down into smaller pieces by focusing on two cyclic granularities at a time.

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 6 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 $\{A_1, A_2, A_3, \ldots, A_n\}$, and $C_2$ maps index set to a set $\{B_1, B_2, B_3, \ldots, B_m\}$ 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 6: The data structure for exploring periodicities in data by including two cyclic granularities in the tsibble structure

## 5.1 Synergy of the cyclic granularities

The way cyclic granularities relate become important when we consider the data structure in Figure 6. 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 like $<A_i, B_j, v(s)>$ can be obtained for all $i \in 1, 2, \ldots, n$ and $j \in 1, 2, \ldots, m$ which will lead to $nm$ data subsets. We will discuss few cases, where one or more of these $nm$ sets will be empty, which will essentially mean that the measured variable can not be explored for all combinations of the categories/levels of the cyclic granularities. To consider a general framework, we will define pairs of cyclic granularities as harmonies or clashes.

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 $C_1$ maps index set to day-of-month and $C_2$ maps index set to week-of-month. Here $C_1$ can have 31 levels/categories while $C_2$ can have 5 categories. There will be $31 \times 5 = 155$ sets $S_{ij}$ corresponding to the possible combinations of day-of-month and week-of-month. Many of these are empty. For example $S_{1,5}$, $S_{21,2}$, etc.This is also intuitive since the first day of the month can never correspond to fifth week of the month. These are structurally empty sets in that it is impossible for them to have any observations.

Secondly, empty combinations can turn up due to differences in event location or duration in a calendar. These are called "event-driven" empty combinations. Again, let us consider a specific example to illustrate this. Let $C_1$ be day-of-week and $C_2$ be Working-Day/NonWorkingDay. Here $C_1$ can have 7 levels while $C_2$ can have 2 levels. So there are 14 sets $S_{ij}$ corresponding to the possible combinations of day-of-week and Working-Day/NonWorkingDay. While potentially all of these can be non-empty (it is possible to have a public holiday on any day-of-week), in practice many of these combinations 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. Let $C_1$ maps index set to Business-days, which are days from Monday to Friday except holidays and $C_2$ is day-of-month. Then the days denoting weekends in a month would not correspond to any Business days and would have missing observations due to the way the granularities are constructed. 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. Hence, they are referred to as "build-based" empty combinations.

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

A pair of cyclic granularities which lead to structurally, event-driven or build-based empty-combinations are referred to as to as **clashes**. And the ones that do not lead to any missing combinations are referred as **harmonies**.

## 5.2 Measured variable

Summarizing properties of the distribution of the measured variable can give valuable indication of features like central tendency, skewness, multimodality, tail behavior or variation. With huge amount of data being available, mean, median or any one summary statistic might not be enough to understand the different properties of the measured variable. Instead the distribution could be summarized using some summary statistics together, each of which can highlight different properties of the data. Tukey's five number summary, letter values or quantiles can serve as examples of distributional summaries. Another way of summarizing could be estimating the probability density function (continuous) or probability mass function (dicrete). The approach can be parametric or non-parametric. A non-parametric approach makes less rigid assumptions about the distribution of the observed data making the data speak more for itself. These resonate more with the notion of

EDA which advocates exploring data for patterns and relationships without requiring prior hypotheses. Several forms of non-parametric density estimators and quantile estimators exist in the literature, the most commonly used form ( Silverman (1986), Jones (1992)) are as follows:

If we assume that $X_1, X_2, \ldots, X_n \sim F$ are independent and identically distributed observations from an univariate distribution with probability density/mass function $f(.)$ and $X_{(1)}, \ldots X_{(n)}$ denote the order statistics of $X_1, X_2, \ldots, X_n$, the Kernel density estimator $\hat{f(x)}$ with Kernel $K$ is defined by:

$$\hat{f(x)} = (1/nh) \sum_{i=1}^{n} K(x - X_i)/h \tag{16}$$

where $h$ is the smoothing parameter or the bandwidth. The sample quantile estimator $\hat{q(u)}$ with Kernel $K$ is given by:

$$\hat{q(u)} = \sum_{i=2}^{n} (X_{(i)} - X_{(i-1)}) K(u - (i-1)/n)) - X_{(n)} K(u-1) + X_{(1)} K(u) \tag{17}$$

where $Q(u) = F^{-1}(u) = inf(x : F(x) > u)$, $0 < u < 1$ and $q \equiv Q'$.

The MSE of the estimators are given as follows:

$$\begin{aligned} MSE(\hat{q(u)}) &= (1/4)h^4 (q''(u))^2 \sigma_k^4 + (1/nh) q^2(u) \kappa \\ MSE(\hat{f(x)}) &= (1/4)h^4 (f''(x))^2 \sigma_k^4 + (1/nh) f(x) \kappa \end{aligned} \tag{18}$$

where $\kappa = \int k^2(y) dy$.

# 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. There are seven grammatical elements in total, out of which data, aesthetics and geometries are essential elements. Data are observations made on a number of variables, typically

thought of as a sample from a population distribution. Aesthetic mapping describes how variables are mapped to elements of the plot, such as horizontal axis or color. Geometries determine the physical representation of the data in the plot. The remaining grammatical elements of a graphic namely facets, statistics, coordinates and themes are optional and contains detail of the plot. Facets dictates how to split up our plot and is a powerful tool for exploratory data analysis as we can rapidly compare patterns in different parts of the data and can see whether they are the same or different.

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: $C_1$ mapped to $x$ position and $v$ to $y$ position
- Statistical transformation: Any descriptive or smoothing statistics that summarizes distribution of $v$
- Geometric objects: Any geometry displaying distribution, for example, boxplot, letter value, violin, ridge or highest density region plots
- Facet: $C_2$

## 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 densities. In Section 5.2, it is discussed that these densities could be estimated using Kernel density estimates or some descriptive statistics. Kernel density estimates provide more detailed information about the distribution but depend on smoothing parameters and hence can vary for the same data set. The descriptive statistics based methods do not allow us to see the shape, skewness, nature of the tail or multi-modality with so much clarity as the former. However, they avoid clutter and help us to focus on some specific properties of the data. Also, the summaries based on descriptive statistics remain free of any tuning parameters or Kernels, which is sometimes more desirable for exploration.

We will discuss few conventional and recent ways to plot distributions using both of these methods.

### 6.1.1 Descriptive statistics based displays

Most commonly used displays include the box plots (Tukey 1977) which convey statistical features such as the median, quartile boundaries, hinges, whiskers and extreme outliers. These displays are based on descriptive statistics, for example, $([n/2]+1)/2^{th}$ and $(n+1-([n/2]+1)/2)^{th}$ order statistics are taken as estimates of the quartiles. Due to wide spread popularity and simplicity in implementation, a number of variations are proposed to the original one which provides alternate definitions of quantiles, whiskers, fences and outliers. Notched box plots (Mcgill et al. 1978) has box widths proportional to the number of points in the group and display confidence interval around medians aims to overcome some drawbacks of box plots. The standard box plot and all of it's variations are helpful to get an idea of the distribution at a glance. However, for data less than 1000 observations, detailed estimates of tail behavior beyond the quartiles are not trustworthy. Also, the number of outliers is large for larger data set since number of outliers is proportional to the number of observations.

The letter-value box plot (Hofmann et al. 2017) was designed to adjust for number of outliers proportional to the data size and display more reliable estimates of tail. These are particularly useful for large data sets. "outliers" in letter value plots are those observations beyond the most extreme letter value. The letter values are shown till the depths where the letter values are reliable estimates of their corresponding quantiles and hence might lead to a lot of letter values being shown, leading to overload of information in one plot.

Quantile plots display quantiles instead of quartiles in a traditional boxplot. These plots give more information than box plots and yet avoid clutter by enabling us to focus just on specific probabilities. While in a boxplot, an outlier is defined as a data point that is located outside the fences ("whiskers") of the boxplot (e.g. outside 1.5 times the interquartile range above the upper quartile and below the lower quartile), in quantile plots outliers are open to interpretation and not shown.

### 6.1.2 Kernel density based displays

Traditional ways to visualize kernel densities include violin plots (Hintze & Nelson 1998). The shape of the violin represents the density estimate of the variable. The more data points

in a specific range, the larger the violin is for that range. Adding two density plots gives a symmetric plot which makes it easier to see the magnitude of the density and compare across categories, enabling easier detection of clusters or bumps within a distribution.

The summary plot (Potter et al. 2010) combines a minimal box plot with glyphs representing the first five moments (mean, standard deviation, skewness, kurtosis and tailings), and a sectioned density plot crossed with a violin plot and an overlay of a reference distribution. This suffers from the same problem as boxplots or violin plot, as it is combination of those two.

A Ridge line plot (sometimes called Joy plot) shows the distribution for several groups. Distribution can be represented using density plots, all aligned to the same horizontal scale and presented with a slight overlap. Like other density based displays, these plots allow us to see multimodality in the distribution. However, these plots can be obscuring when there is overlap of distribution for two or more categories of the y-axis. Also, with lot of categories, it is difficult to compare the height of the densities across categories.

The highest density region (HDR) box plot proposed by (Hyndman 1996) displays a probability density region that contains points of relatively highest density. The probabilities for which the summarization is required can be chosen based on the requirement. These regions do not need to be contiguous and help identify multi-modality.

Each type of density display has different parameters, that need to be estimated given the data. Each is equipped with some benefits and challenges. For example, if we have too many categories in the variables mapped to facet or x-axis, the quantile plots might be useful for comparing patterns, whereas, other more involved methods of plotting are useful for studying anomalies, outlier or multimodal behavior.

## 6.2 Facet and aesthetic variables

### 6.2.1 Levels

We will discuss the effect of the levels of cyclic granularities in this section. Recall that cyclic granularities $C_1$ and $C_2$ are such that $C_1$ maps index set to a set $\{A_1, A_2, A_3, \ldots, A_n\}$, and $C_2$ maps index set to a set $\{B_1, B_2, B_3, \ldots, B_m\}$. $A_i$'s and $B_j$'s are referred to as the levels of $C_1$ and $C_2$ for all $i \in 1, 2, \ldots, n$ and $j \in 1, 2, \ldots, m$ respectively. The levels have

an impact on the choice of plots since space and resolution might become a problem if the number of levels are too high. A potential approach can be to categorize the levels as very high/high/medium/low for each cyclic granularities 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. For example, any levels above 31 can be considered as very high, any levels between 14 to 31 can be taken as high and that between 7 to 14 can be taken as medium and below 7 as low.

The following principles are useful while choosing distribution plots given two temporal granularities:

- If levels of both granularity plotted are low/medium, then any distribution plots might be chosen depending on which feature of the distribution needs focus.
- If level of the granularity plotted across x-axis is more than medium, ridge plots should be avoided to escape overlap of categories.
- If level of the granularity plotted across x-axis is more than or equal to high, quantile plots are preferred.
- If levels of any granularity plotted are more than medium, quantile based methods of visualizing distribution is preferred as they use less space by design than most density based methods.

### 6.2.2 Interaction

We will discuss the effect of the interaction of cyclic granularities in this section. In Section 5.1, we discussed how pairs of granularities can have empty combinations either due to structure of calendar, event location or duration or due to the way they are built. In this section, we will see how these empty combinations affect the visualization when a dependent variable is plotted against these granularities.

For illustration, distribution of half-hourly electricity consumption of Victoria is plotted across different time granularities in each of the panel in Figure 7. Figure 7 (a) shows the letter value plot across days of the month faceted by months like January, March and

December. Figure 7 (c) shows box plot across days of the year by the 1st, 15th, 29th and 31st days of the month. Figure 7 (d) showing violin plot across days of the month faceted by week of the month. Figure 7 (e), variations across week of the year conditional on week of the month can be observed through a ridge plot and Figure 7 (f) shows decile plots across day of the year and month of the year.

Clearly, in Figure 7, we observe that some choices of cyclic time granularities work and others do not. In Figure 7 (c), there will be no observations for some combinations day-of-month and day-of-year. In particular, the 1st day of the month can never correspond to 2nd, 3rd or 4th day of the year. On the contrary, for Figure 7 (a), we will not have any combinations with zero combinations because every day-of-week can occur in any month-of-year. Thus the graphs that don't work are those where many of the combination sets are empty. In other words, 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.

We redefine harmony and clashes as follows: harmonies are pairs of granularities that aid exploratory data analysis, and clashes are pairs that are incompatible with each other for exploratory analysis. As a result, we should avoid plotting clashes as they hinder the exploratory process by having missing combinations of time granularities in each panel.

### 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. When $C_2$ is mapped to facets and $C_1$ is mapped to $x$ axis, comparisons would ideally happen across all $B_j$ within each $A_i$. Although the variables used in the layers are the same, the difference in mapping

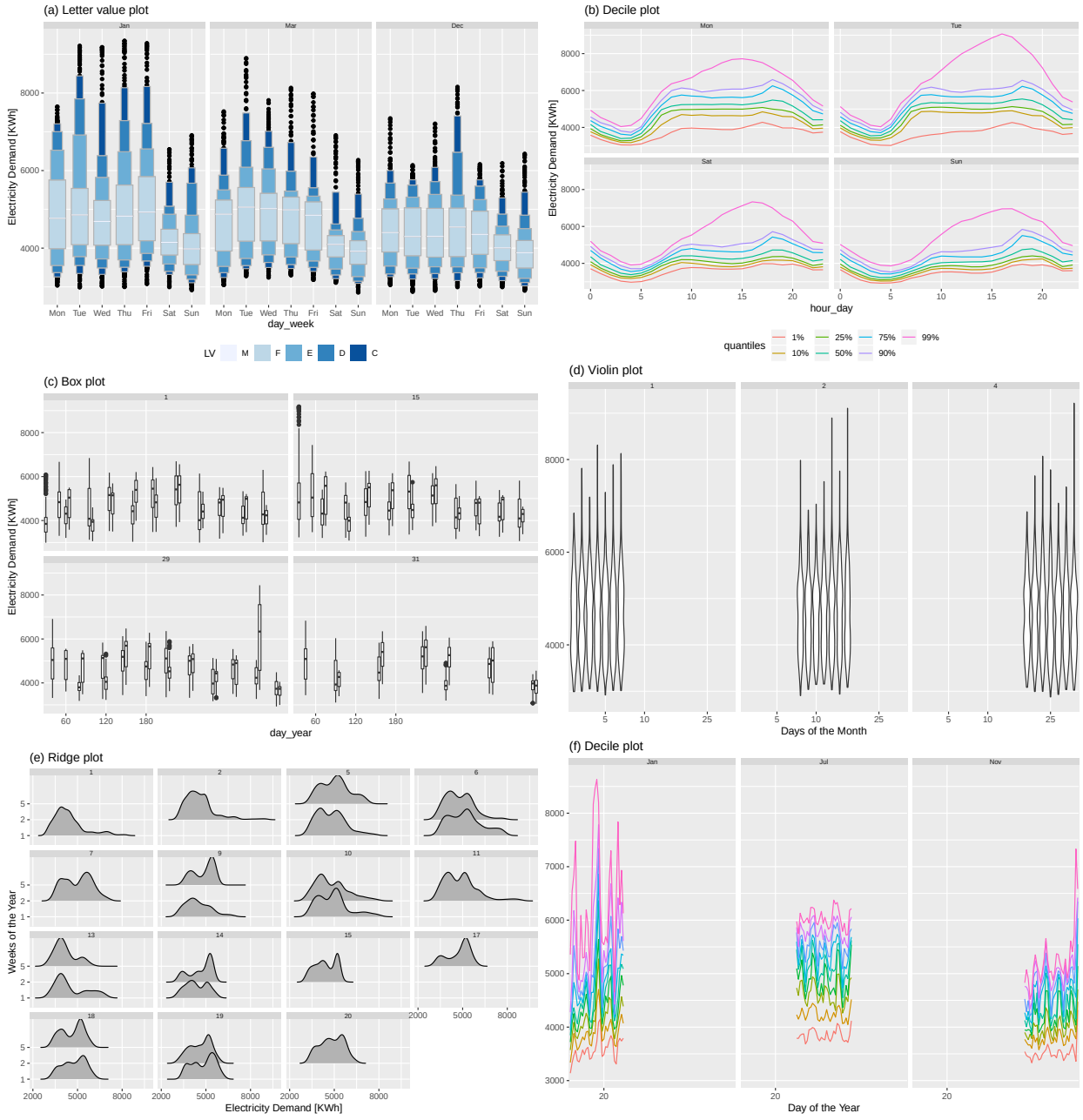


Figure 7: Various probability distribution plots of electricity consumption data of Victoria from 2012 to 2014. (a) Letter value plot by day-of-month and month-of-year (b) Decile plot by hour-of-day and day-of-week (c) Box plot by day-of-year and day-of-month, (d) Violin plot of day-of-month and week-of-month, (e) Ridge plot by week-of-month and week-of-year, (f) Decile plot by day-of-year and month-of-year. Only plots (a) and (b) show harmonised time variables.

leads to emphasis of different behavior of the variables.

## 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 or unevenly distributed events.

### 6.3.1 Rarely occuring events

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 to create a meaningful panel, then provided $T \geq mnk$, the visualization 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, we would need $k \geq 30$. Rarely occurring categories such as the 366th day of the year, or the 31st day of the month can suffer from such problem.

### 6.3.2 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 or quantile estimates should be used with caution as sample size would directly affect both the variance and consequently the confidence interval of the estimators. Equation 18 shows that the variance of both $f(\hat{x})$ and $q(\hat{u})$ is a function of the sample size $n$.

# 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 (SGSC) project (2010-2014) 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. It would be interesting to explore the energy consumption distribution for these customers and gain insights on their energy behavior which are lost either due to aggregation or looking only at coarser temporal units. The idea here is to show how looking at the energy consumption across different cyclic granularities together can help identify different behavioral patterns.

### 7.1.1 Data structure for visualization:

The data structure for a sample of 50 households from that trial is as follows:

```
#> # A tsibble: 1,450,232 x 3 [30m] <UTC>
#> # Key:       customer_id [50]
#>    customer_id reading_datetime    general_supply_kwh
#>    <chr>       <dttm>                           <dbl>
#>  1 10006414    2012-02-10 08:00:00              0.141
#>  2 10006414    2012-02-10 08:30:00              0.088
#>  3 10006414    2012-02-10 09:00:00              0.078
#>  4 10006414    2012-02-10 09:30:00              0.151
#>  5 10006414    2012-02-10 10:00:00              0.146
#>  6 10006414    2012-02-10 10:30:00              0.077
#>  7 10006414    2012-02-10 11:00:00              0.052
#>  8 10006414    2012-02-10 11:30:00              0.055
#>  9 10006414    2012-02-10 12:00:00              0.055
#> 10 10006414    2012-02-10 12:30:00              0.252
#> # ... with 1,450,222 more rows
```

The variable `reading_datetime` is the index of the tsibble and represents the time variable. The variable `customer_id` denoting different households represent the key of the tsibble and `general_supply_kwh` is the time series variable which needs to be analyzed. Data sets of the form $<C1$, $C2$, `general_supply_kwh`$>$ will form the basis of exploration for each key in the data set, where the cyclic granularity $C1$ would be mapped to $x$ axis and `general_supply_kwh` to $y$ axis with the cyclic granularity $C2$ mapped to facet. Any geometry displaying the probability distribution of `general_supply_kwh` like boxplot, letter value, violin, ridge or highest density region plots can then be used to explore the data set.

### 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 (Section 3.1), pseudo-circular (Section 3.2) or aperiodic (Section 3.3). If we consider conventional time deconstructions for a Gregorian calendar (second, minute, half-hour, hour, day, week, fortnight, month, quarter, semester, 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.

```
library(gravitas)
smart_meter %>% search_gran()
```

```
#>  [1] "hhour_hour"        "hhour_day"         "hhour_week"
#>  [4] "hhour_fortnight"   "hhour_month"       "hhour_quarter"
#>  [7] "hhour_semester"    "hhour_year"        "hour_day"
#> [10] "hour_week"         "hour_fortnight"    "hour_month"
#> [13] "hour_quarter"      "hour_semester"     "hour_year"
#> [16] "day_week"          "day_fortnight"     "day_month"
#> [19] "day_quarter"       "day_semester"      "day_year"
```

```
#> [22] "week_fortnight"     "week_month"          "week_quarter"
#> [25] "week_semester"      "week_year"           "fortnight_month"
#> [28] "fortnight_quarter"  "fortnight_semester" "fortnight_year"
#> [31] "month_quarter"      "month_semester"      "month_year"
#> [34] "quarter_semester"   "quarter_year"        "semester_year"
```

If these options are considered too many, the most coarse temporal unit can be set to a "month".

```
smart_meter10 %>%
  search_gran(highest_unit = "month")
```

```
#>  [1] "hhour_hour"      "hhour_day"       "hhour_week"      "hhour_fortnight"
#>  [5] "hhour_month"     "hour_day"        "hour_week"       "hour_fortnight"
#>  [9] "hour_month"      "day_week"        "day_fortnight"   "day_month"
#> [13] "week_fortnight"  "week_month"      "fortnight_month"
```

Also, some intermediate temporal units that might not be pertinent to the analysis can be removed from the list of cyclic granularities we want to look at.

```
smart_meter10 %>% search_gran(highest_unit = "month",
                              filter_out = c("hhour", "fortnight")
                              )
```

```
#> [1] "hour_day"   "hour_week"  "hour_month" "day_week"   "day_month"
#> [6] "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 visualisation. 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.

```
smart_meter10 %>% harmony(
  ugran = "month",
  filter_out = c("hhour", "fortnight")
)
```

```
#> # A tibble: 13 x 4
#>    facet_variable x_variable facet_levels x_levels
#>    <chr>          <chr>              <int>    <int>
#>  1 day_week       hour_day              7       24
#>  2 day_month      hour_day             31       24
#>  3 week_month     hour_day              5       24
#>  4 day_month      hour_week            31      168
#>  5 week_month     hour_week             5      168
#>  6 day_week       hour_month            7      744
#>  7 hour_day       day_week             24        7
#>  8 day_month      day_week             31        7
#>  9 week_month     day_week              5        7
#> 10 hour_day       day_month            24       31
#> 11 day_week       day_month             7       31
#> 12 hour_day       week_month           24        5
#> 13 day_week       week_month            7        5
```

In Figure 8, 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 letter value plots instead of quantile plots to look at the same data in Figure 9. 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 Figure 9, 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.

```
smart_meter10 %>%
  filter(customer_id %in% c(10017936))%>%
  gran_advice("wknd_wday",
                              "hour_day")
```

```
#> The chosen granularities are harmonies
#>
#> Recommended plots are: violin lv quantile boxplot
#>
#> Number of observations are homogenous across facets
#>
#> Number of observations are homogenous within facets
```

```
#>
#> Cross tabulation of granularities :
#>
#> # A tibble: 24 x 3
#>    hour_day Weekday Weekend
#>    <fct>      <dbl>   <dbl>
#>  1 0            910     366
#>  2 1            908     366
#>  3 2            909     366
#>  4 3            910     366
#>  5 4            910     366
#>  6 5            910     366
#>  7 6            909     366
#>  8 7            908     366
#>  9 8            908     366
#> 10 9            908     366
#> # ... with 14 more rows
```

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 `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 screen some 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.

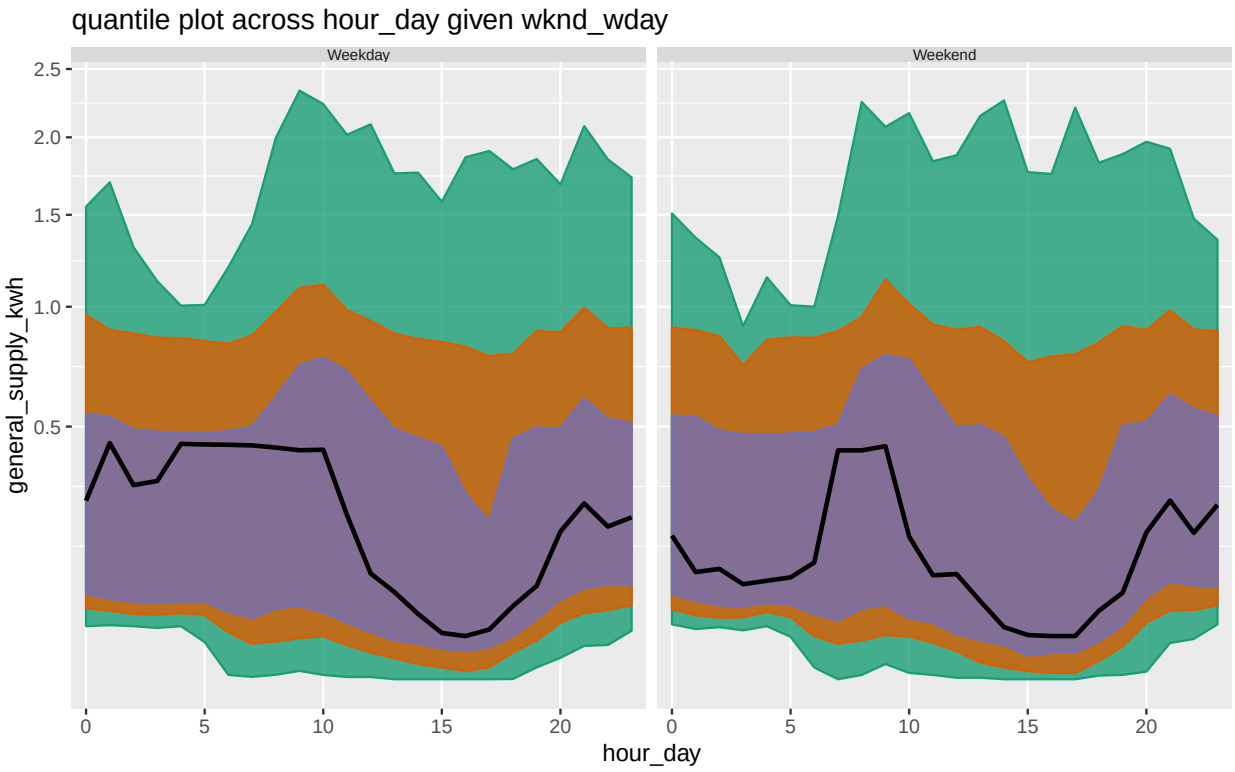


Figure 8: Area quantile plot of energy consumption across hours of the day faceted by weekday/weekend for customer id: 10017936 displays the smoothed energy consumption curve for the entire day. The customer typically consumes more in the morning hours following which their consumption is low till late afternoon hours. Median consumption for the early morning hours starts higher and continue to remain high for longer during weekdays compared to weekdend.

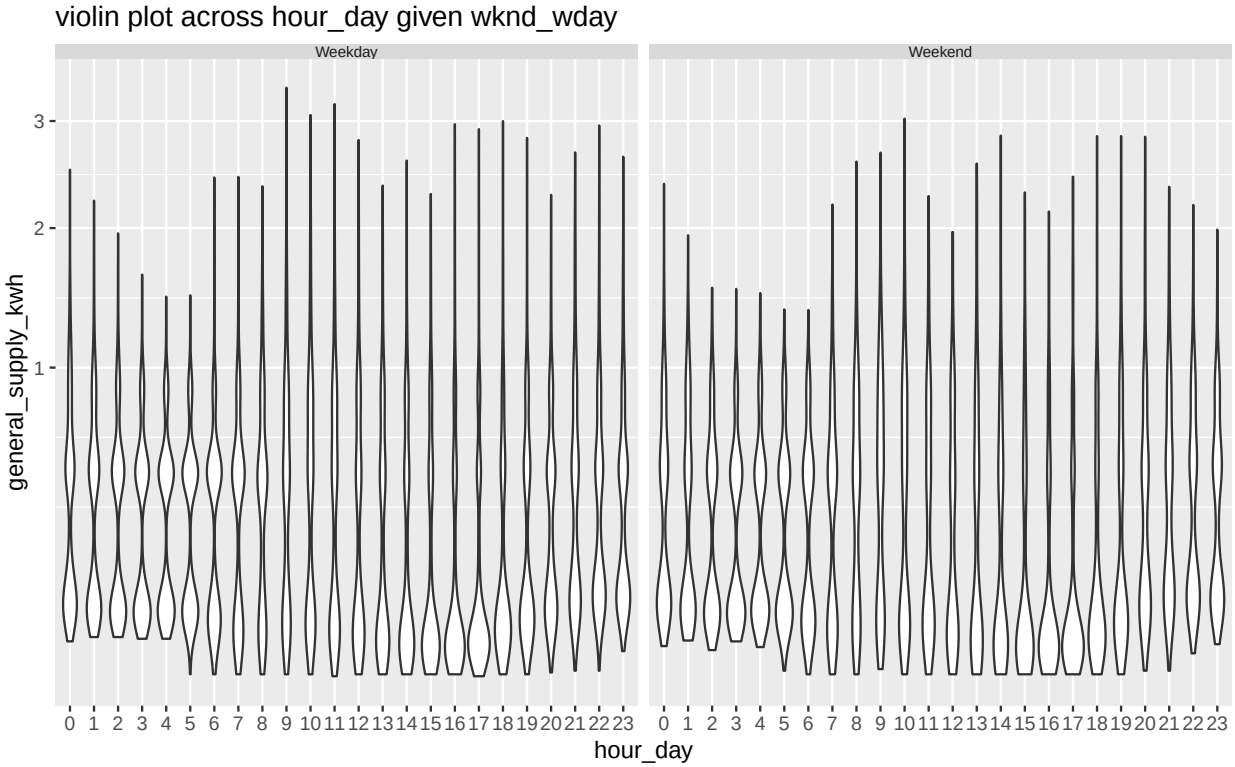


Figure 9: Violin plots of energy consumption across hours of the day faceted by weekday/weekend for customer id: 10017936. Early morning hours are bimodal implying both high and low consumptions are probable. Bimodality can be caused by difference in behavior across different seasons. Volatility is highest from 7 to 13 hours as evident from the narrow violins.

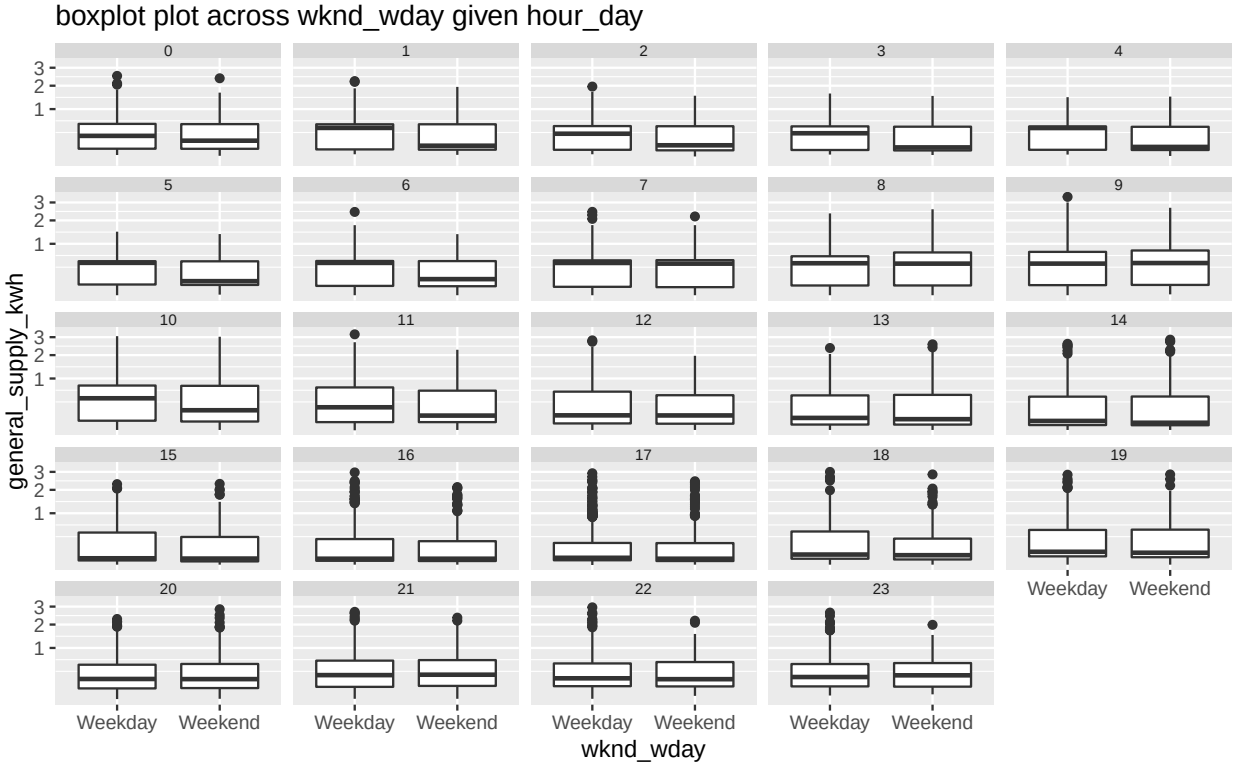


Figure 10: Boxplot of energy consumption across hours of the day faceted by weekday/weekend for customer id: 10017936. The change in mapping of the cyclic granularities leads to easier comparison within weekdays and weekends for each hour. Median consumption till 6am is higher for weekends compared to 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.

## 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 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. We will explore a few and understand how the proposed approach in the paper can help answer some of the questions.

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

Mumbai Indians (MI) and Chennai Super kings (CSK) are considered one of the best teams in IPL with multiple winning titles and always appearing in final playoffs from 2010 to 2015. It would be interesting to take their example in order to dive deeper into this question. Circular granularities "over-of-inning" and "inning-of-match" can be computed using 3.1 with over as index of the tsibble. From Figure 11, 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 in 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?

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. If a batsman is bowled out, it does not necessarily signify good fielding. So we only include number of catches and run out in an over as a measure of good fielding. Difference in runs across overs are likely to be negative if good fielding has an impact on the runs scored in the subsequent overs. Figure 12 shows that at least one dismissal in an over leads to a lower median and less volatility in the distribution of difference in run rates.

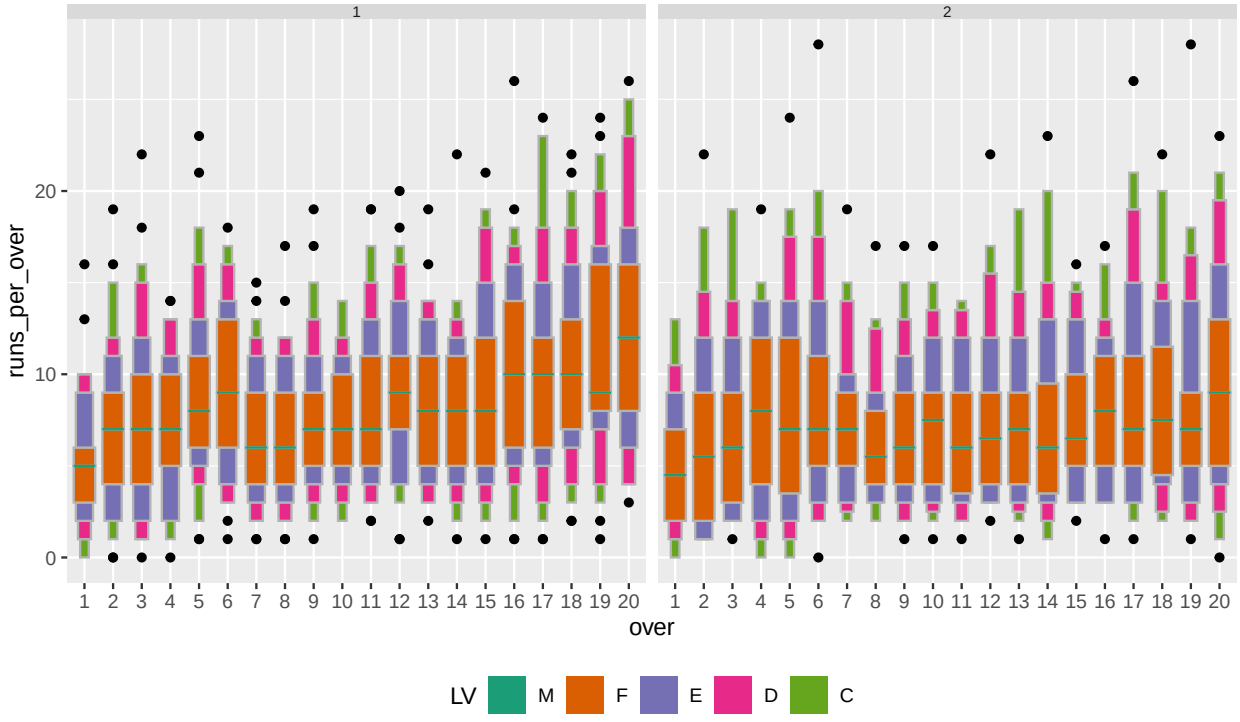


Figure 11: Letter value plot of runs per over across overs of the inning faceted by innings of the match. No upward shift in runs in the second innings like that in the first implying teams are more vulnerable to score more in the first innings as they approach the end of the inning.

```
cricket_data %>% gran_obs("field", "over")
```

```
#> # A tibble: 20 x 3
#>    over  `>0 wickets` `no wickets`
#>    <fct>        <dbl>        <dbl>
#>  1 1              125         1039
#>  2 2              155          997
#>  3 3              179          972
#>  4 4              173          978
#>  5 5              189          962
#>  6 6              191          957
#>  7 7              144         1002
#>  8 8              134         1012
#>  9 9              148          993
#> 10 10             157          984
#> 11 11             173          962
#> 12 12             177          951
#> 13 13             169          954
#> 14 14             205          913
#> 15 15             208          892
#> 16 16             219          865
#> 17 17             231          833
#> 18 18             291          743
#> 19 19             287          695
#> 20 20             306          572
```

```
cricket_data %>% gran_obs("dot", "over")
```

```
#> # A tibble: 20 x 3
#>    over  `>0 dot balls` `no dot balls`
#>    <fct>          <dbl>          <dbl>
```

```
#>  1 1              1146              18
#>  2 2              1140              12
#>  3 3              1111              40
#>  4 4              1112              39
#>  5 5              1099              52
#>  6 6              1099              49
#>  7 7              1057              89
#>  8 8              1026             120
#>  9 9              1027             114
#> 10 10             1010             131
#> 11 11              992             143
#> 12 12              971             157
#> 13 13              970             153
#> 14 14              957             161
#> 15 15              935             165
#> 16 16              904             180
#> 17 17              881             183
#> 18 18              852             182
#> 19 19              788             194
#> 20 20              683             195
```

Q3: Are runs set to reduce in the next over for dot balls in the previous over?

A dot ball is a delivery bowled without any runs scored off it. The number of dot balls is reflective of the quality of bowling in the game. Run rate of an over should ideally decrease if the number of dot balls increase. However, what is the effect of dot balls on runs scored in the subsequent over. Will players batsman likely to go for big shots because they couldn't score good runs in the previous over? Or they should play consistently and avoid scoring high? Figure 13 shows the quantile plot of runs across overs for at least one dot ball per over (facet 1) or no dot balls per over (facet 2). With at least one dot balls per over, the distribution of difference in run rates as observed through the quantiles plotted have shifted downwards. This implies that run rates are likely to decrease in the subsequent

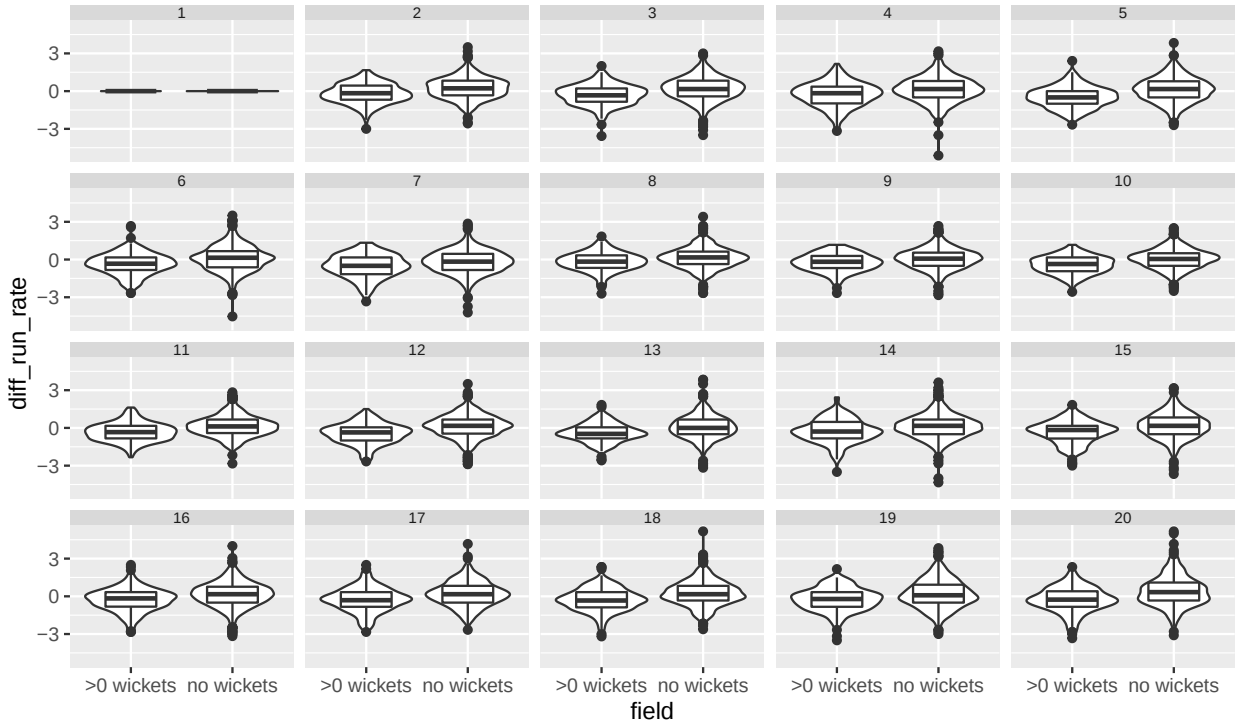


Figure 12: Violin plots with an overlaid boxplot showing distribution of lagged run rate across overs of the innings and wickets dismissed due to fielding. This shows good fielding in previous over leads to lower run rate in the subsequent over. Also run rates are much less volatile in case there are at least one wicket in the previous over.

over as a result of dot balls in the previous over.

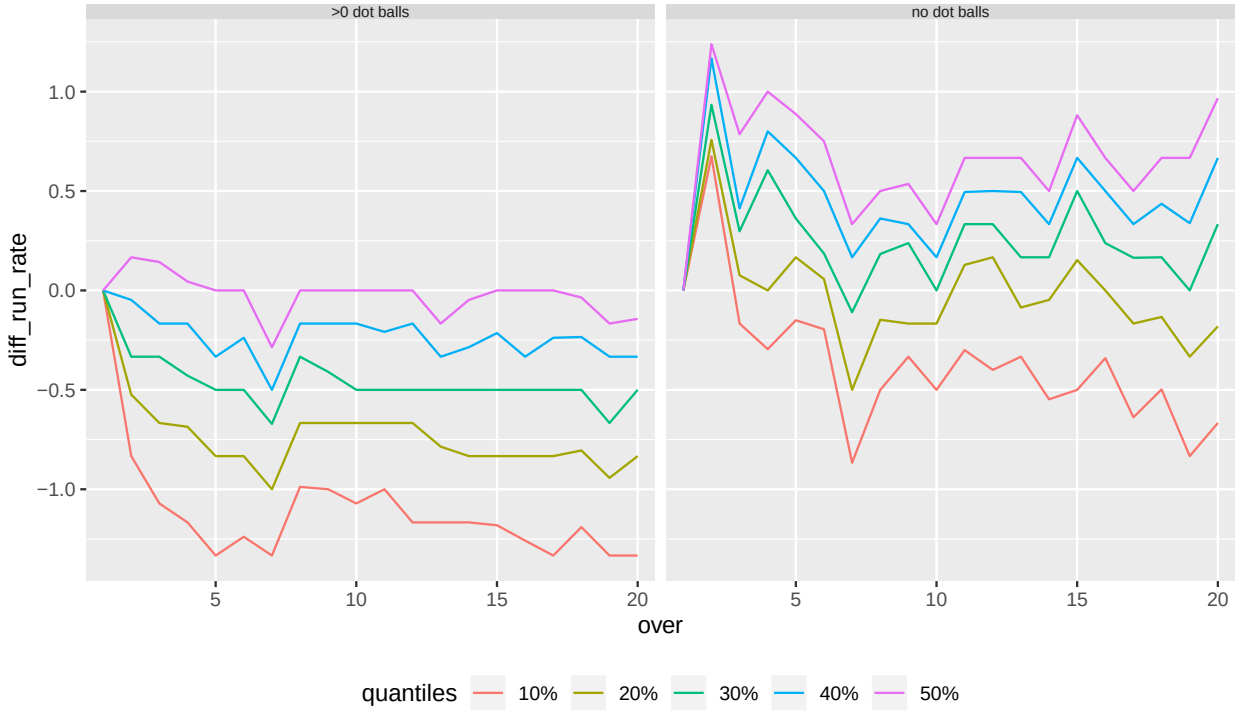


Figure 13: 10th, 20th, 30th, 40th and 50th quantiles of runs per over are drawn across overs of the innings with no (facet 1), more than zero (facet 2) dot balls per over. For all quantiles, difference in runs between subsequent overs decrease with at least one dot ball in the previous over.

In Q2 and Q3, dots balls per over or catches per over are considered as aperiodic cyclic granularities with dot balls or catches as aperiodic linear granularities. These aperiodic linear 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. While any special event that corresponds to a time domain can be treated as an aperiodic linear granularity in a temporal case, dot balls or wickets that corresponds to certain balls (index) could be treated as aperiodic events in cricket.

# 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.

The infrastructure of systematic exploration of observed variable across pair of cyclic time granularities has been implemented in the R package `gravitas`. A missing piece of the gravitas package 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.

# 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. This article was created with knitr (Xie 2015) and R Markdown (Xie et al. 2018). The project's Github repository `https://github.com/Sayani07/paper-gravitas` contains all materials required to reproduce this article.

# References


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

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*

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.

Jones, M. C. (1992), 'Estimating densities, quantiles, quantile densities and density quantiles', *Ann. Inst. Stat. Math.* **44**(4), 721–727.

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.

Reingold, E. M. & Dershowitz, N. (2001), *Calendrical Calculations Millennium Edition*, Cambridge University Press.

Silverman, B. W. (1986), *Density estimation for statistics and data analysis*, Monographs on statistics and applied probability ; [26], Chapman and Hall, London ; New York.

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., Allaire, J. & Grolemund, G. (2018), *R Markdown: The Definitive Guide*, Chapman and Hall/CRC, Boca Raton, Florida.

**URL:** *https://bookdown.org/yihui/rmarkdown*