Exploring probability distributions for bivariate temporal granularities

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 time granularities, circular time granularities and aperiodic calendar categorizations. Linear time granularities respect the linear progression of time such as hours, days, weeks and months. Circular time granularities accommodate periodicities in time such as hour of the day, and day of the week. Aperiodic calendar categorizations are neither linear nor circular, such as day of the month or public 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 (Wickham 2016).

Contents

1 Introduction

Temporal data are available at various resolution depending on the context. Social and economic data like GDP are often collected and reported at coarser temporal scales like monthly, quarterly or annually. With recent advancement in technology, more and more data are recorded and stored 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 collected at every second. As the frequency of data increases, the number of questions about 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. Calendar-based graphics[] can unpack the temporal variable, at different resolutions, to digest multiple seasonalities, and special events. They mainly pick out 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 [reference trellis plot] 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. A temporal granularity which results from such a deconstruction may be intuitively described as a sequence of time granules, each one consisting of a set of time instants[]. There can be several classes of time deconstructions including linear time granularities, circular time granularities and aperiodic calendar categorizations. Linear time granularities respect the linear progression of time such as hours, days, weeks and months. Circular time granularities accommodate periodicities in time such as hour of the day, and day of the week. Aperiodic calendar categorizations are neither linear nor circular, such as day of the month or public holidays. Also, the hierarchical structure of time creates a natural nested ordering. For example, hours are nested within days, days within weeks, weeks within months, and so on. Hence, we can construct single-order-up granularities like second of the minute or multiple-order-up granularities like second of the hour.

It is important to be able to navigate through all of these temporal granularities to have multiple perspectives on 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.

The motivation for this work comes from the desire to provide methods to better understand large quantities of measurements on energy usage reported by smart meters in household across Australia, and indeed many parts of the world. Smart meters currently provide half-hourly use in kWh for each household, from the time that they were installed, some as early as 2012. Households are distributed geographically, and have different demographic properties such as the existence of solar panels, central heating or air conditioning. The behavioral patterns in households vary substantially, for example, some families use a dryer for their clothes while others hang them on a line, and some households might consist of night owls, while others are morning larks.

It is common to see aggregates of usage across households, total kWh used each half hour by state, for example, because energy companies need to understand maximum loads that they will have to plan ahead to accommodate. But studying overall energy use hides the distributions of usage at finer scales, and making it more difficult to find solutions to improve energy efficiency.

We propose that the analysis of probability distributions of smart meter data at finer or coarser scales can be benefited from the approach of Exploratory Data Analysis (EDA). EDA calls for utilizing visualization and transformation to explore data systematically. It is a process of generating hypothesis, testing them and consequently refining them through investigations.

This paper utilizes the nestedness of time granularities to obtain multiple-order-up granularities from single-order-up ones.

Finally, visualizing data across single/multiple order-up granularities help 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. However, this approach can lead to a myriad of choices all of which are not useful. Analysts are expected to iteratively visualize these choices for exploring possible patterns in the data. But too many choices might leave him bewildered.

This work provides tools for systematically exploring bivariate granularities within the tidy workflow through proper study of what can be considered a prospective graphic for exploration. Pairs of granularities are categorized as either a *harmony* or *clash*, where harmonies are pairs of granularities that aid exploratory data analysis, and clashes are pairs that are incompatible with each other for exploratory analysis. Probability distributions can be examined using the range of graphics available in the ggplot2 package.

In particular, this work provides the following tools.

- Functions to create multiple-order-up time granularities. This is an extension to the lubridate package, which allows for the creation of some calendar categorizations, usually single-order-up.
- Checks on the feasibility of creating plots or drawing inferences from two granularities together. Pairs of granularities can be categorized as either a *harmony* or *clash*, where harmonies are pairs of granularities that aid exploratory data analysis, and clashes are pairs that are incompatible with each other for exploratory analysis.

2 Definitions of 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). Examples of time abstractions may also include day-of-week, time-of-day, week-of-year, day-of-month, month-of-year, working day/non-working day, etc which are useful to represent different periodicities in the data.

2.1 Arrangement: Linear vs. Circular vs. aperiodic

The arrangement of the time domain can result in deconstructing time in linear, circular or nearly circular ways. Time granularities are **linear** if they respect the linear progression of time. Examples include hours, days, weeks and months. However, periodicity is very common in all kinds of data. Time granularities, which are constructed by using two linear granularities, can be utilized to explain such periodicities. The mappings between 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 can have days ranging from 28 to 31. Hence, time granularities can also be **circular** or **aperiodic** depending on if the mapping of the linear granularities is regular or irregular. Examples of circular time granularities include hour of the day, and day of the week, whereas examples for aperiodic time granularities can be day of the month or public holidays.

Providing a formalism to some of these abstractions is important to model a time series across differently grained temporal domains.

2.1.1 Linear

There has been several attempts to provide the framework for formally characterizing time-granularities and identifying their structural properties, relationships and symbolic representations. One of the first attempts occur in (Bettini et al. 1998) with the help of the following definitions:

Definition: 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.

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: A linear **granularity** is a mapping G from the integers (the index set) to subsets of the time domain such that:

(C1) 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
(C2) if $i < k < j$ and $G(i)$ and $G(j)$ are non-empty, then $G(k)$ is non-empty.

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

The first condition 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. If we consider the bottom linear granularity (Aigner et al. 2011) as hourly and the entire horizon has T hours then it will have $\lfloor T/24 \rfloor$ days, $\lfloor T/(24 * 7) \rfloor$ weeks and so on.

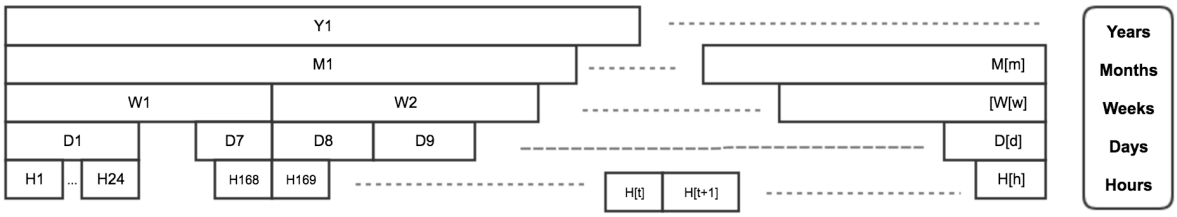


Figure 1: The time domain distributed as linear granularities

The definitions and rules for linear granularities are inadequate to reflect periodicities in time, like weekly, monthly or yearly seasonality. Hence, there is a need to define circular time granularities in a different approach.

2.1.1.1 Relationships between two linear granularities and association to Periodicity

(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: 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 $G(i) \subset H(j)$.

Definition: 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

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

According to this definition, $G \trianglelefteq H$, if each granule H (i) is the union of some granules of G. For example, $Days \trianglelefteq Months$,. This relationship, however, is not sufficient to fully describe a granularity in terms of another one. For example, it is clear that $Days \trianglelefteq Months$,, since each month is a grouping of a number of days, however it is also a periodical grouping. If leap year is ignored, the periodicity of the grouping is 1 year, since each month would be defined as the grouping of the same number of days (31, 28, or 30, depending on the month) every year. Considering leap years and all their exceptions, the period becomes 400 years, but the grouping is always periodic. In order to define a granularity in terms of another granularity including the notion of periodicity, a particular case of the groups into relationship is considered.

Definition: Groups periodically into 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 $H(i+R) = \bigcup_{j \in S} G(j+P)$.

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: $\bigcup_{i \in S_j} G(P * \lfloor j/R \rfloor + i)$.

This formula can be explained observing that H (j) is either one of H (0), . . . , H (R - 1) or, for the periodicity of H, is one of these granules "shifted" ahead or behind on the time line of a finite number of periods. If $H(j) \epsilon H(0), ..., H(R-1)$, then the formula 'j mod R' defines the index (among those in {0,. . . ,R -

1}) of the granule that must be shifted to obtain H (j).The number of periods each granule of G composing H (j mod R) should be shifted is given by $\lfloor j/R \rfloor$.

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 and De Sibi 2000).

Definition: Quasi Periodical 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 $H(i+R) = \bigcup_{k \in [0,k]} G(j_r + P)$.

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: Bottom granularity Given a granularity order relationship g-rel and a set of granularities having the same time domain, a granularity G in the set is a bottom granularity with respect to g-rel, if G g-rel H for each granularity H in the set.

Example: 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 $\trianglelefteq$.

2.1.1.2 Computation of linear time granularities

Linear time granularities are computed through an algebraic representation for time granularities, which is referred to as calendar algebra(Ning, Wang, and Jajodia 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. Thus, the relationship between the operand(s) and the resulting granularities are encoded in the operations.

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 Mondays while looking at the linear granularity "day".

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

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

Example :

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

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 lth 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, Wang, and Jajodia 2002)

2.1.2 Cyclic

In a cyclic organization of time, the domain is composed of a set of recurring time values. Hence, any time value can be preceeded and succeeded by another time value (for e.g Monday comes before Wednesday but Monday also suceeds Wednesday). 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.

We propose a formalism of cyclic time granularities through the tsibble (Wang, Cook, and Hyndman 2019) framework of organizing temporal data. A time domain, as defined by Bettini, 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 is days, then a linear granularity might be weeks, months or years. Lining up with the the assumption in (Bettini and 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 and is represented by the index of the tsibble.

The mapping of two linear granularities forming a cyclic granularity can be regular or irregular. It will be regular if, for example, they are formed by grouping operation. Grouping operation would ensure that each granule of the resulting granularity consists of same number of granules of bottom granularity. However, if they are formed by the altering-tick operation, granules of the resulting granularity is composed of different number of granules of bottom granularity. However, the relationship can still be periodic and hence they are also useful in addressing periodicities. The formalisms would however differ for circular and aperiodic granularities.

2.1.2.1 Circular

Definition: Circular granularity A circular granularity C(BG, G) relates a linear granularity G to the bottom granularity, if

$$C_{BG,G}(z) = BG(z \mod k(BG, G)) \forall z \in \mathbb{Z}^{+}, BG(z) \neq \phi \tag{2}$$

where,
z denotes the index set.
BG denotes the index of the tsibble (bottom granularity).
G be a linear granularity, periodic with respect to BG.
k(BG,G) be the period length of the bottom granularity with respect to linear granularity G.

Discussion: Example showing circular granularities relating two linear granularities each with bottom granularities are visually depicted in a series of slots in the diagram Figure 2. Each granule is represented by a box. The diagram also illustrate that the granules that overlap share elements from the underlying time domain. The first slot in the diagram shows the index set and the bottom granularity. The index of the tsibble considered is days. G and H are two linear granularities denoting days and weeks respectively. The period length of days with respect to weeks is 7, hence, $C_{G,BG}(z)$ represented by C(BG, G) is given by $BG(z \mod 7)$. The circular granularity C(BG, G) consists of repetitive pattern {BG(0), BG(1), BG(2), BG(3), BG(4), BG(5), BG(6)} which repeats itself after each period length. Similarly, C(BG, H) is given by $BG(z \mod 14)$ since the period length of days with respect to fortnight is 14. C(BG, H) also repeats the

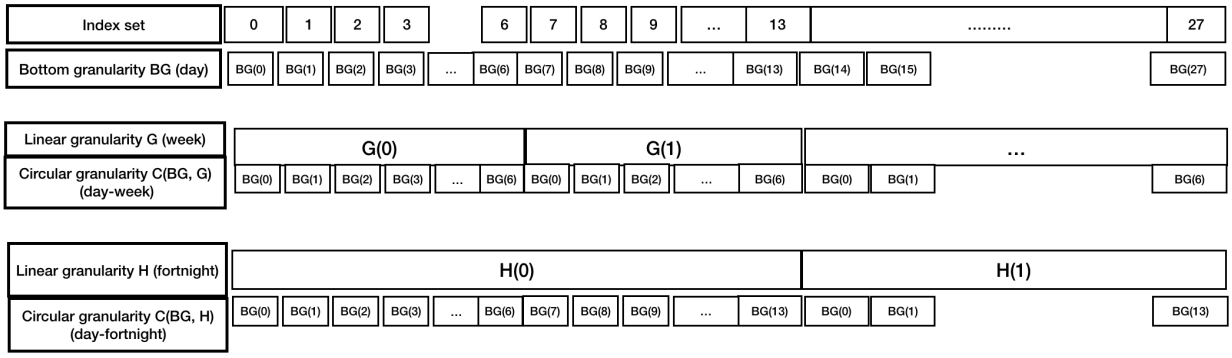


Figure 2: Relating circular granularities and bottom linear granularity

pattern $BG(0), BG(1), \ldots, BG(13)$ in every period. Hence, circular granularities are represented as repetitive categorization of time in the diagram.

All circular granularities can be expressed in terms of the bottom linear granularity with equation 2.

In general, any circular granularity relating two linear granularity, none of which are bottom granularity can be expressed as $C_{(G,H)}(z) = BG(\lfloor z/k(BG,G) \rfloor \mod k(G,H))$, where linear granularity H is periodic with respect to linear granularity G and k(G, H) represents the period length of G within H.

Table **??** shows representation of circular granularities relating two linear granularities. It is possible that none of these two linear granularities are bottom granularities. But the representation of the resultant circular granularity will be a function of the bottom granularity. For example, suppose the bottom granularity is minutes, and let k_i is the period length of C_i.

Minute-of-Hour:	$C_1(z) = z \mod 60$	$k_1 = 60$
Minute-of-Day:	$C_2(z) = z \mod 60 * 24$	$k_2 = 1440$
Hour-of-Day:	$C_4(z) = \lfloor z/60 \rfloor \mod 24$	$k_3 = 24$
Hour-of-Week:	$C_6(z) = \lfloor z/60 \rfloor \mod 24 * 7$	$k_4 = 168$
Day-of-Week:	$C_7(z) = \lfloor z/24 * 60 \rfloor \mod 7$	$k_5 = 7$

Table 1: Illustrative circular granularities with time index in minutes

2.1.3 Aperiodic

Definition: An **Aperiodic circular granularity** can not be defined using modular arithmetic in a similar fashion. The modulus for these type of calendar categorizations are not constant due to unequal length of some linear granularities. For example, please refer to the table below:

HOM:	$C_3(s) = s \mod 720$ (approximately)	$n_3 = 744$
HOY:	$C_4(s) = s \mod 8760$ (except for leap years)	$n_4 = 8784$
DOM:	$C_6(s) = \lfloor s/24 \rfloor \mod 30$ (approximately)	$n_6 = 31$
DOY:	$C_7(s) = \lfloor s/24 \rfloor \mod 365$ (except for leap years)	$n_7 = 366$
WOM:	$C_8(s) = \lfloor s/168 \rfloor \mod 4$ (approximately)	$n_8 = 5$
WOY:	$C_9(s) = \lfloor s/168 \rfloor \mod 52$ (approximately)	$n_9 = 53$
MOY:	$C_{10}(s) = \lfloor s/720 \rfloor \mod 12$ (approximately)	$n_{10} = 12$

Table 2: Illustrative aperiodic circular granularities with time index in hours

C1 and C2 are two cyclic granularities in the hierarchy table with $order(C1) < order(C2)$. Let $f(C_1, C_2)$ denotes the accessor function for representing the finer unit C1 in terms of a coarser unit C2 and $const(C1, C2)$

is a constant which relates C1 and C2.

Then $f(x, y)$ can be computed using modular arithmetic as follows:

$$f(x, y) = \lfloor z/c(z, x) \rfloor \mod c(x, y)$$

The constant c(x,y) is for granularities for groups into periodically, for granualrities which group into each other in a quasi-periodic way, no such constants exist and no such close form solution can be obtained.

To consider the exhaustive set of temporal regularities that might exist in the data, we can also define temporal granularities based on the hierarchical structure of a calendar.

2.2 Order: Single vs. Multiple

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.

Definition: Order of a linear granularity can be comprehended as the level graininess asscociated with a linear granularity. If we consider two linear granularities G and H, such that $G \preceq H$, then H is of higher order than G.

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

In a hierarchy table, linear granularities are arranged from highest to lowest order of linear granularities.

We refer to granularities which are nested within multiple levels as **multiple-order-up** granularities and those concerning a single level as **single-order-up** granularities. Let us look at few calendars to see examples of single and multiple order-up granularities.

2.2.1 Computation of multiple-order from single-order

$$\begin{aligned}
C_{(G_l,G_m)}(z) &= C_{(G_l,G_{(l+1)})}(z) + k(l, l+1)(C_{(G_{(l+1)},G_m)}(z) - 1) \\
&= C_{(G_l,G_{(l+1)})}(z) + k(l, l+1)[C_{G_{(l+1)},G_{(l+2)}}(z) + k(l+1, l+2)(C_{G_{(l+2)},G_m)}(z) - 1) - 1] \\
&= C_{(G_l,G_{(l+1)})}(z) + k(l, l+1)(C_{G_{(l+1)},G_{(l+2)}}(z) - 1) + k(l, l+1)k(l+1, l+2)(C(l+2, m) - 1) \\
&= C_{(G_l,G_{(l+1)})}(z) + k(l, l+1)(C_{G_{(l+1)},G_{(l+2)}}(z) - 1) + k(l, l+2)(C_{G_{(l+2)},G_{(l+m)}}(z) - 1) \\
&\vdots \\
&= \sum_{i=0}^{order(m)-order(l)-1} k(l, l+i)(C_{G_{(l+i)},G_{(l+i+1)}}(z) - 1)
\end{aligned} \tag{3}$$

Example: So far we have used the 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. 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	k
kin	kin-of-uinal	20
uinal	uinal-of-tun	18
tun	tun-of-katun	20
katun	katun-of-baktun	20
baktun	1	1

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

Let us use the equation 3 to compute the multiple-order-up granularity uinal_katun for Mayan calendar, which is periodic.

$$\begin{aligned} C(uinal, baktun) &= C(uinal, tun) + kuinal, tun)C(tun, katun) + C(uinal, katun)C(katun, baktun) \\ &= \lfloor z/20 \rfloor \mod 18 + 20 * \lfloor z/20 * 18 \rfloor \mod 20 + 20 * 18 * 20 \lfloor z/20 * 18 * 20 \rfloor \mod 20 \end{aligned} \tag{4}$$

2.2.2 Computation of single-order from multiple-order

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

$$C_{G_{l1}, G_{m1}}(z) = C_{G_{l2}, G_{m2}}(\lfloor z/k(l_2, l_1) \rfloor \mod k(m_1, m_2)) \tag{5}$$

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 5

$$C(tun, katun) = \lfloor C(uinal, baktun)(z)/18 \rfloor \mod 20 \tag{6}$$

–>

3 Synergy of cyclic time granularities

Before exploring the behavior of a "dependent variable" across time granularities, it is important to know how these granularities interact with each other. If two time granulaties of interest interact, the relationship between each of the interacting variables and the dependent variable will depend on the value of the other interacting variable. To define a general framework, we define harmony and clashes.

3.1 Harmony and Clashes

Suppose we have two circular or aperiodic granularities C_1 and C_2, such that C_1 maps row numbers to a set $\{A_1, A_2, A_3, \ldots, A_n\}$, and C_2 maps index set to a set $\{B_1, B_2, B_3, \ldots, B_m\}$. That is, let S_{ij} be the set of index set such that for all $s \in S_{ij}$, $C_1(s) = i$ and $C_2(s) = j$. Since, C_1 has n levels and C_2 has m levels there will be nm such sets S_{ij}.Now, many situations can lead to any of these sets being empty. Let us discuss the following cases, where one or more of these sets can be empty.

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 row numbers to Day-of-Month and C_2 maps row numbers to Week-of-Month. Here C_1 can take 31 values while C_2 can take 5 values. There will be $31 \times 5 = 155$ sets S_{ij} corresponding to the possible combinations of WOM and DOM. 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 DOW and C_2 be WorkingDay/NonWorkingDay. Here C_1 can take 7 values while C_2 can take 2 values. So there are 14 sets S_{ij} corresponding to the possible combinations of DOW and WD/NWD. While potentially all of these can be non-empty (it is possible to have a public holiday on any DOW), 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.

Thirdly, empty combinations can be a result of how granularities are constructed. Let C_1 maps row numbers to "Business days", which are days from Monday to Friday except holidays and C_2 is Day-of-Month. Then the weekends in Days-of-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 the 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 row numbers to Day-of-Week and C_2 maps row numbers to Month-of-Year. Here C_1 can take 7 values while C_2 can take 12 values. So there are $12 \times 7 = 84$ sets S_{ij} corresponding to the possible combinations of DOW and MOY. All of these are non-empty because every DOW can occur in every month.

Therefore, pair of circular/aperiodic 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 called **harmonies**.

4 Visualization

The grammar of graphics introduced a framework to incorporate underlying features for statistical graphics to construct them by relating data space to the graphic space (Wilkinson 1999). The layered grammar of graphics proposed by (Wickham 2016), which is a modification over the former, allows us to produce graphics using the same structured thinking that is used to design analysis. Briefly, any graphics are made up of **Layers**, **Facet**,**Scale** and a **coordinate system**. **Layers** are made up of **data**, **mapping**, **statistical transformation**, **geometric objects** and **position adjustments**.

Creating a visualisation requires a number of nuanced judgements. However, we exploit some features of layers and faceting to come up with a recommendation system while visualizing the distribution of the "dependent" variable across bivariate temporal granularities. The general framework of visualizing the distribution invloves a faceting approach with one temporal granularity plotted along the x-axis, the other one across facet and the dependent time series variable on the y-axis. Different distribution plots might be appropriate depending on which features of the distrbution we are interested to look at, the levels of time granularities being plotted, how granularities interact and number of observations available. We will look at each of these aspects separately and analyze the effect of each on visualization.

While we accomplish that, we need to bear in mind that any customized species of visualization might be constructed by varying the set of mappings between data properties and visual attributes such as position, shape and color. We are trying to address problems where even with good aesthetic qualities and good data, the graph might be confusing or misleading because of how people process the information.

4.1 Choice of Plots

There are several ways to plot statistical distributions. The displayed probabilities in these plots are either computed using kernel density estimation methods or empirical statistical methods. The latter 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 much clutter, where some specific probabilities representing typical and extreme behavior are on focus. The probabilities displayed are based on actual data, which is aligned to principles that governed Tukey's original boxplot. Density plots which uses a kernel density estimate to show the probability density function of the variable can show the entire distribution, unlike discretizing the distribution and showing only parts of it. They can be useful in spotting multimodality, however, they might become obscuring with too many categories and information on the entire distribution can add to cognitive load. Also, the probabilities in density plot are estimated through kernel density estimation and thus makes assumptions when selecting kernel or bandwidth. As a result, the plot shows smooth summaries based on the assumptions and not only on actual observations. The density based visualizations are subject to the same sample size restrictions and challenges that apply to any density estimation. In practice, the densities are estimated reasonably with at least 30 observations. Even with sample sizes of several hundred, however, choosing too large a value for bandwidth can cause the estimate to oversmooth the data.

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

4.1.1 Empirical methods

Most commonly used techniques to display a distribution of data include the histogram (Karl Pearson), which shows the prevalence of values grouped into bins and the box-and-whisker plots (Tukey 1977) which convey statistical features such as the median, quartile boundaries, hinges, whiskers and extreme outliers. The box plot is a compact distributional summary, displaying less detail than a histogram. 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, Tukey, and Larsen 1978, 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 these variations are helpful to get an idea of the distribution at a glance. Moreover, 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, Wickham, and Kafadar 2017, 2006) was designed to adjust for number of outliers proportional to the data size and display more reliable estimates of tail. "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 and leading to overload of information in one plot.

Quantile plots visually portray the quantiles of the distribution of sample data. Much like the quartiles divide the data set equally into four equal parts, extensions include dividing the data set even further. These plots are referred to as "quantile" plots. For example, number of quantiles would be 9 for a decile plot and 99 for percentile plot. It avoids much clutter and just enable us to focus on specific probabilities, typically representing typical and extreme behaviors. A large data set is required for the extreme percentiles to be estimated with any accuracy. These plots can display any quantiles instead of quartiles in a traditional boxplot. When reviewing 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). However, outliers are open to interpretation and not shown in an quantile plot.

4.1.2 Kernel density estimation methods

Traditional ways to visualize densities include violin plots (Hintze and Nelson 1998, 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, 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 (both color and width are mapped to estimated density), 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 of a numeric value for several groups. Distribution can be represented using histograms or density plots, all aligned to the same horizontal scale and presented with a slight overlap. A clear advantage over boxplots is that 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, if there are 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.

Given a context, it is good to be conversant with the benefits and challenges while choosing a distribution plot. As a general rule, if we have too many categories, the quantile plots are useful for comparing patterns, whereas, other more involved methods of plotting are useful for studying anomalies, outlier or multimodal behavior.

4.1.3 Effect of levels of temporal granularities

The levels of the two granularities plotted have an impact on the choice of plots since space and resolution might become a problem if the number of levels are too high. The criteria for different levels could be based on usual cognitive power while comparing across facets and display size available to us. Plot choices will also vary depending on which granularity is placed on the x-axis and which one across facets.

Levels are categorized as very high/high/medium/low each for the facet variable and the x-axis variable. Default values for these levels are chosen based on levels of common temporal granularities like day of the month, day of a fortnight or day of a week. 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. 31, 14 and 7 are the levels of days-of-month, days-of-fortnight and days-of week respectively.

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, empirical based methods of distribution visualizing are preferred as they use less space by design than most density based methods.

4.2 Effect of synergy of time granularities

In Section **??**, 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 build. 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 3. Figure 3 (a) shows the letter value plot across days of the month faceted by months like January, March and December. Figure 3 (c) shows box plot across days of the

year by the 1st, 15th, 29th and 31st days of the month. Figure 3 (d) showing violin plot across days of the month faceted by week of the month. Figure 3 (e), variations across week of the year conditional on week of the month can be observed through a ridge plot and Figure 3 (f) shows decile plots across day of the year and month of the year.

Clearly, in Figure 3, we observe that some choices of time granularities work and others do not. In Figure 3 (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 3 (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 time granularities plotted across x-axis which are not spanned by levels of the time granularities plotted across facets or vice versa, we will have empty sets leading to potential ineffective graphs. We hypothesize that the synergy of these granularities are in play 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.

4.3 Effect of number of observations

Even with harmonies, visualizing probability distributions can be misleading either due to rarely occuring categories or unevenly distributed events.

4.3.1 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 structurally 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 what type of visualization we are producing. For a decile plot, even $k = 10$ 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.

4.3.2 Unevenly distributed events (TODO)

Even when there are no rarely occuring events, number of observations might vary hugely across within or across each panel. This might happen due to missing observations in the data or uneven locations of events in time domain. We use Gini's measure to compute if differences in number of observations across or within facets are significantly different.

5 Applications

5.1 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 half-hourly energy usage and detailed information on appliance use, climate, retail and distributor product offers, and other related factors. It would be interesting to explore the energy consumption distribution for these customers and gain more insights on their energy behavior which are otherwise lost either due to aggregation or looking only at coarser temporal units. The idea here is to show how looking at the time across different granularities together can help identify different behavioral patterns and identify the extreme and regular households amongst these 50 households.

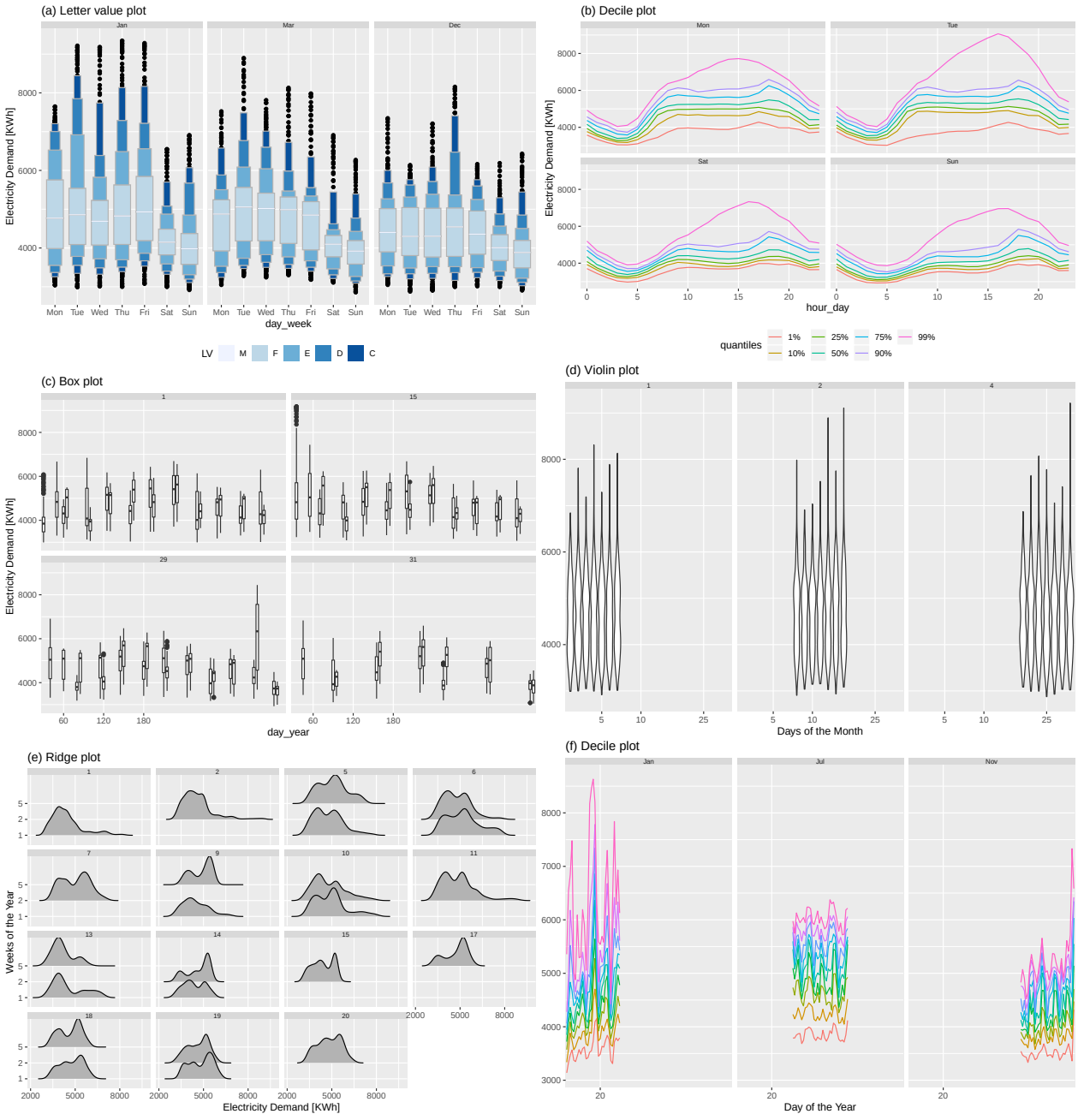
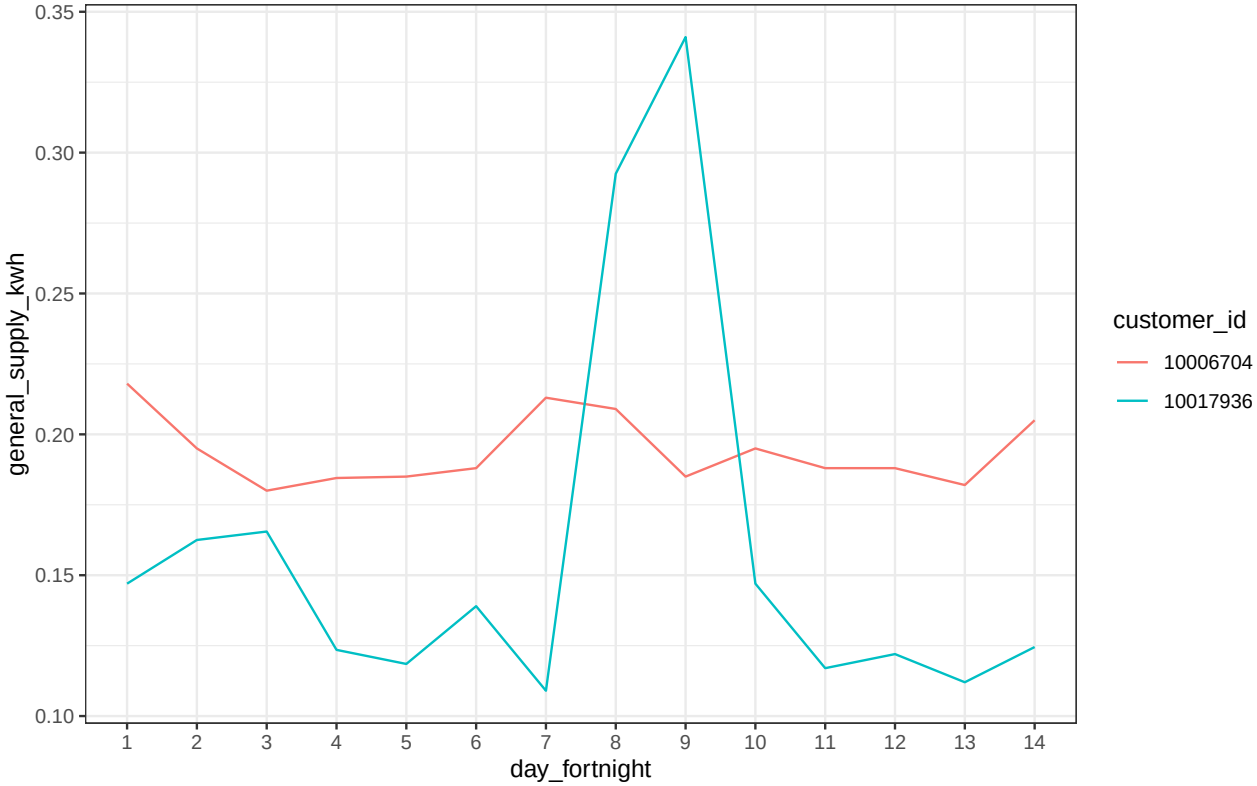
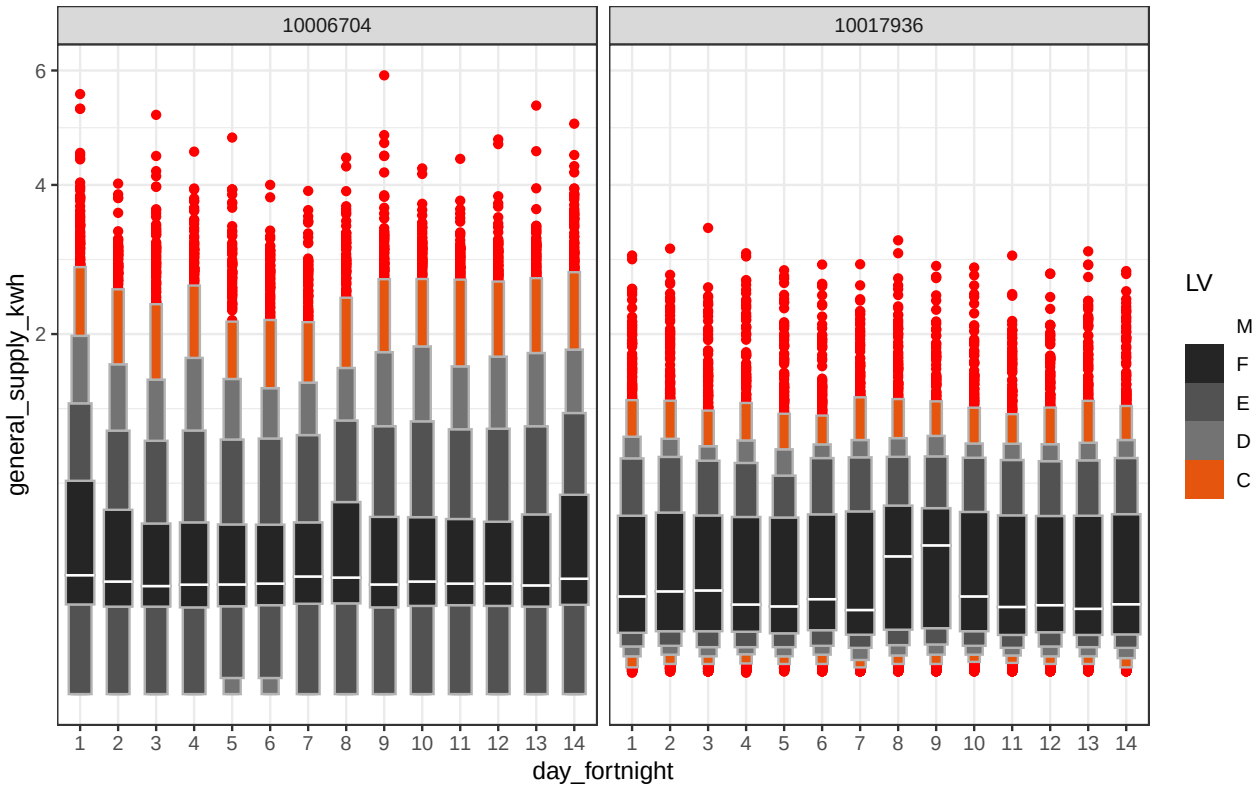


Figure 3: Various probability distribution plots of electricity consumption data of Victoria from 2012 to 2014. (a) Letter value plot by DoM and MoY, (b) Decile plot by HoD and DoW (c) Box plot by DoY and DoM, (d) Violin plot of DoM and WoM, (e) Ridge plot by WoM and WoY, (f) Decile plot by DoY and MoY. Only plots (a) and (b) show harmonised time variables.

In Figure **??**, the smart meter data is filtered for two customers to illustrate what kinds of insights can be drawn for the energy behavior of these two customers. it can be seen that for most days of the fortnight, the second household has much less consumption than the first one. However, there is additional information which we can derive looking at the distribution. If we consider letter value F as a regular behavior and letter values beyond F as not-so-regular behavior, we can conclude that the regular behavior of the first household is more stable than the second household. However, the distribution of tail of the first household is more variable, observed through distinct letter values, implying that their not-so-regular behavior is quite extreme. This shows, how looking at the distribution of the dependent variable can throw more light on the energy behavior of the customers, which are lost using aggregate or summary statistics.





While trying to explore the energy behavior of these customers systematically across time granularities, the first thing we should have at our disposal is to know the number of time granularities we can look at exhaustively. If we consider conventional time deconstructions for a Gregorian calendar (second, minute, half-hour, hour, day, week, fortnight, month, quarter, semester, year), the following time granularities can be considered for this analysis.

```
#> 30m
```

```
#>  [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"
```

The interval of this tsibble is 30 minutes, and hence the default in this case, provides temporal granularities ranging from half-hour to year. If these options are considered too many, the default options can be modified to limit the possibilities. For example, the most coarse temporal unit can be set to be a "month".

```
#>  [1] "hhour_hour"       "hhour_day"        "hhour_week"
#>  [4] "hhour_fortnight"  "hhour_month"      "hour_day"
#>  [7] "hour_week"        "hour_fortnight"   "hour_month"
#> [10] "day_week"         "day_fortnight"    "day_month"
#> [13] "week_fortnight"   "week_month"       "fortnight_month"
```

This looks better. However, some intermediate temporal units might not be pertinent to the analysis and we might want to remove them from the list of granularities.

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

Now that we have the list of granularities to look at, we can visualize the distribution. From the search list, we found that we can look at six granularities, that amounts to analyzing six graphics. However, what happens if we want to see the distribution of energy across two granularities at a time? This is equivalent to looking at the distribution of energy consumption across one granularity conditional on another one. One way can be to plot one of the granularities on the x-axis and another on the facet. Different perspectives of the data can be derived depending on where the granularities are placed.

So, what is the number of pairs of granularities we can look at? It is equivalent to taking 2 granularities from 6, which essentially means we need to examine 30 plots. The good news is, not all time granularities can be plotted together and we do not have to analyze so many plots!

Harmony/clash can be identified to considerably reduce the number of visualizations that can aid exploratory analysis.

```
smart_meter10 %>%
  is_harmony(gran1 = "hour_day",
             gran2 = "day_week")
```

```
#> [1] "TRUE"
```

```
smart_meter10 %>%
  is_harmony(gran1 = "hour_day",
             gran2 = "day_week",
             facet_h = 14)
```

```
#> [1] "FALSE"
```

```
smart_meter10 %>%
  is_harmony(gran1 = "day_month",
             gran2 = "week_month")
```

```
#> [1] "FALSE"
```

Let us now look at all the harmonies that we can examine. Fortunately, we are left with only 13 out of 30 visualizations.

```
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
```

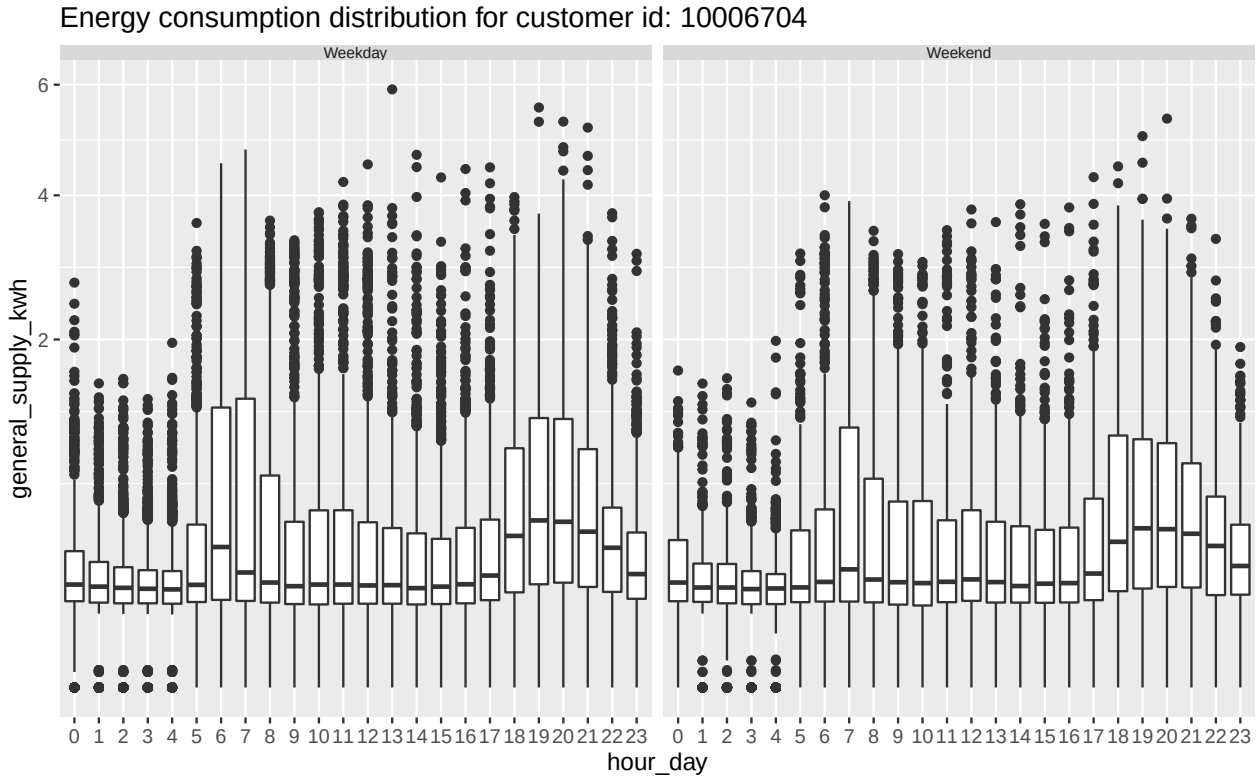
```
#> 10 hour_day       day_month            24       31
#> 11 day_week       day_month             7       31
#> 12 hour_day       week_month           24        5
#> 13 day_week       week_month            7        5
```

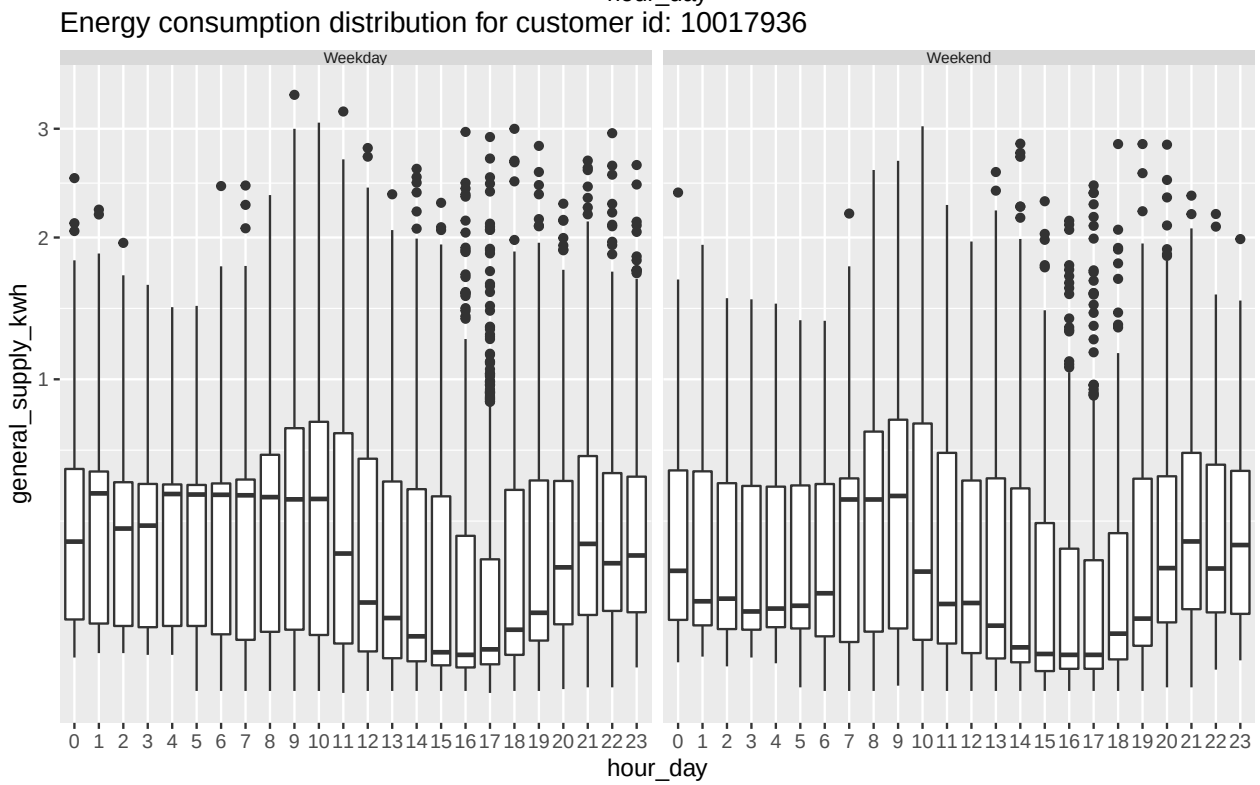
```
smart_meter10 %>% gran_advice("wknd_wday", "hour_day")
```

```
#> 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           7705    3097
#>  2 1           7698    3100
#>  3 2           7698    3101
#>  4 3           7698    3102
#>  5 4           7699    3099
#>  6 5           7701    3098
#>  7 6           7700    3099
#>  8 7           7700    3098
#>  9 8           7695    3098
#> 10 9           7696    3098
#> # ... with 14 more rows
```

In **??** We visualize the harmony pair (wknd_wday, hour_day) through a box plot. Boxplot of energy consumption is shown across wknd_wday (facet) and hour-day (x-axis) for the same two households. For the second household, outliers are less prominent implying their regular behavior is more stable. For the first household, energy behavior is not significantly different between weekdays and weekends. For the second household, median energy consumption for the early morning hours is extremely high for weekends compared to weekdays.

For the second household, outliers are less prominent implying their regular behavior is more stable. For the first household, energy behavior is not significantly different between weekdays and weekends. For the second household, median energy consumption for the early morning hours is extremely high for weekends compared to weekdays.



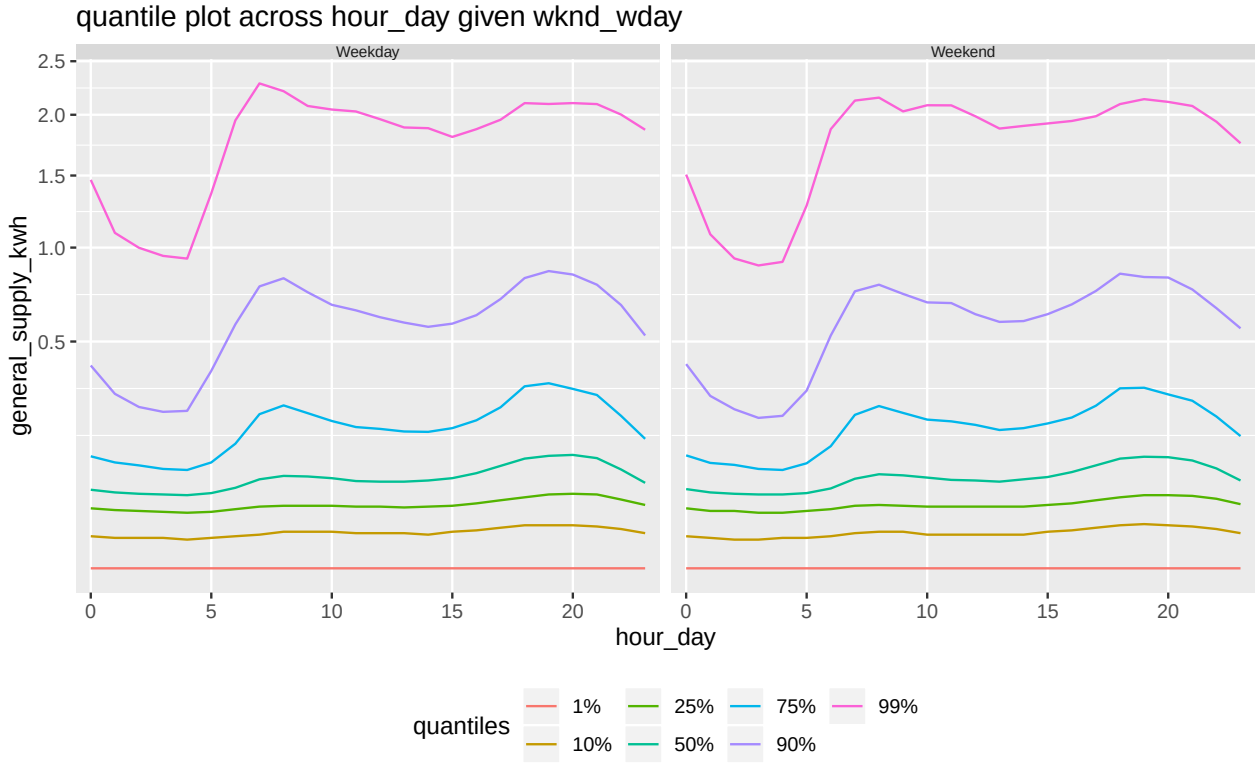


Now, we would like to see how these customers' behavior relate to the rest of the 50 households across these two measures -

- if energy distribution is skewed towards extreme?
- if the weekend and weekday behavior are different for most of these households?

Figure **??** shows the quantile plot of energy consumption of 50 households. Similar quantiles for weekend and weekdays indicate that behaviors of most of these customers do not alter between weekdays and weekends.

Figure **??** shows the area quantile plots of energy consumption across hours of the day faceted by months of the year. 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. It can be observed that the median is very close to the lower boundaries of all the other bands implying energy consumption for these households are left skewed. Moreover, only the pink band changes significantly in winter months (May - August). The median consumption or width of other bands doesn't vary much across seasons, implying extreme behavior or extreme customers does not vary across seasons much. Most of the behavioral changes occur in the quartiles, that too in peak hours of the day. It is to be noted here that off course the level of quantiles increased too in winter months, the interesting part is to notice that the relationship between bands other than quartile stayed same across seasons.

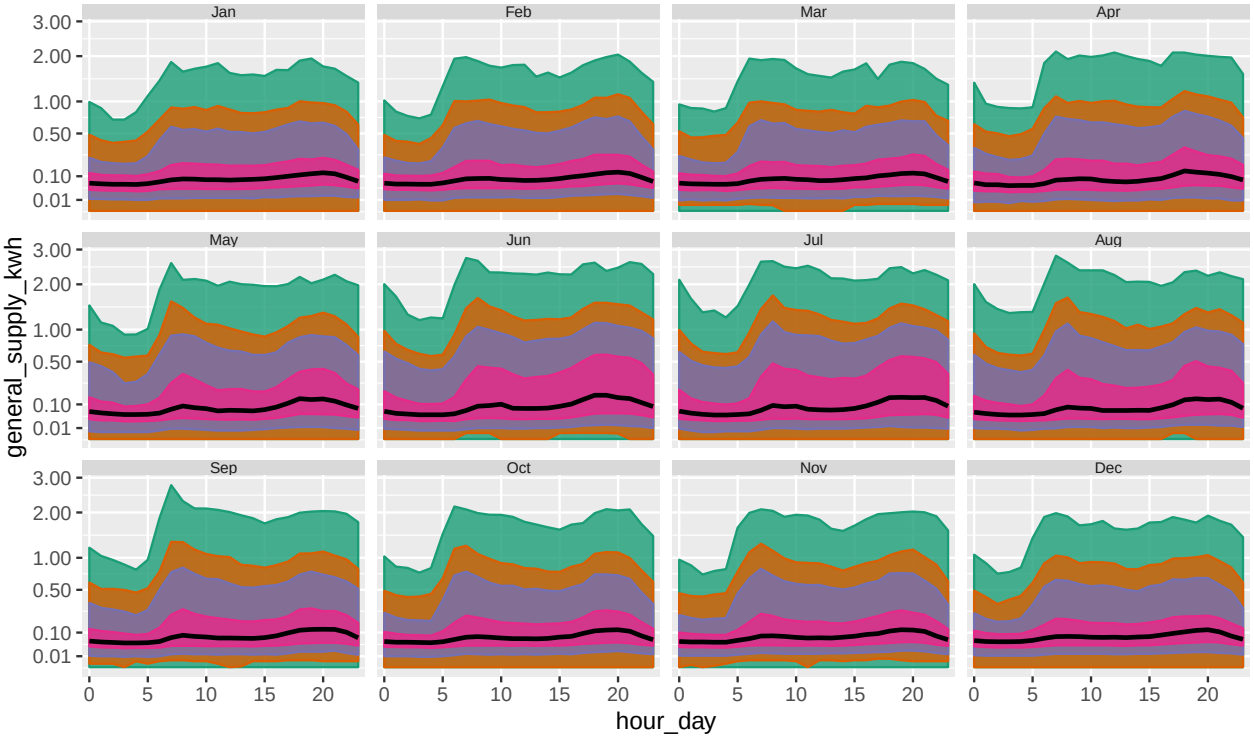



```
library(gravitas)
library(tsibble)

library(ggplot2)
smart_meter %>%
  prob_plot("month_year", "hour_day",
    response = "general_supply_kwh",
    plot_type = "quantile",
    quantile_prob = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99),
    symmetric = TRUE) +
  ggtitle("") + scale_y_sqrt(breaks = c(0.01, 0.1, 0.5, 1:3))
```

Table 4: A hierarchy table for T20 cricket

units	convert_fct
ball	6
over	20
inning	2
match	1



5.2 T20 cricket data of Indian Premiere League

The application is not only restricted to temporal data. We provide an example of cricket to illustrate how this can be generalized in other applications. The Indian Premier League (IPL) is a professional Twenty20 cricket league in India contested by eight teams representing eight different cities in India. With eight teams, each team plays each other twice in a home-and-away round-robin format in the league phase. In a Twenty20 game the two teams have a single innings each, which is restricted to a maximum of 20 overs. Hence, in this format of cricket, a match will consist of 2 innings, an innings will consist of 20 overs, an over will consist of 6 balls. A hierarchy like table 4 can be construed for this game format.

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 cross overs and contains information for a sample of 214 matches spanning 9 seasons (2008 to 2016).

```
#> Observations: 8,560
#> Variables: 10
#> $ season        <dbl> 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2008, ...
#> $ match_id      <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ...
#> $ batting_team  <chr> "Chennai Super Kings", "Chennai Super Kings", "C...
#> $ bowling_team  <chr> "Kings XI Punjab", "Kings XI Punjab", "Kings XI ...
#> $ inning        <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
#> $ over          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
```

```
#> $ wicket        <dbl> 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, ...
#> $ dot_balls     <dbl> 4, 2, 4, 3, 3, 3, 1, 3, 1, 2, 1, 0, 2, 1, 0, 3, ...
#> $ runs_per_over <dbl> 5, 14, 2, 6, 9, 11, 9, 8, 13, 5, 19, 20, 4, 10, ...
#> $ run_rate      <dbl> 1, 2, 0, 1, 2, 2, 2, 1, 2, 1, 3, 3, 1, 2, 3, 1, ...
```

Although there is no conventional time granularity in cricket, we can still represent the data set `cricket` through a `tsibble`, where each over, which represents an ordering from past to future, can form the index of the tsibble. The hierarchy table would look like the following:

units	convert_fct
index	1
over	20
inning	2
match	1

There are many interesting questions that can possibly be answered with such a data set, however, we will explore a few and understand how the proposed approach in the paper can help answer some of the questions.

First, we look at the distribution of runs per over across over of the innings and seasons in 4. The distribution of runs per over has not significantly changed from 2008 to 2016. There is no clear pattern/trend that runs per over is increasing or decreasing across seasons. Hence, we work with subsets of seasons and try to see how the strategies of the winning teams differ across seasons or how in a particular season the strategy was different for the winning team and the ones who did not qualify for the playoffs.

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 4 from 2010 to 2015. It would be interesting to observe their strategies throughout all matches in the two seasons. The following two questions might help us partially understand their strategies.

Q1: How run rates vary depending on if a team bats first or second?

Q2: How number of wickets and dot balls in the previous over affect the runs in the subsequent over across different overs of the innings?

From figure **??**, it can be observed that there is no clear upward shift in runs in the second innings as compared to the first innings. The variability of runs also increases as the teams approach towards the end of the innings, as observed through the longer and more distinct letter values.

–>

Q3: Is run rate set to reduce in subsequent over if fielding/bowling is good in the previous over? Between fielding and bowling which penalizes runs in the subsequent over more?

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 catches and run out as a measure of fielding. Difference in run rates should be negative if fielding is good. Let us see if this fact is true.Figure 6 shows the difference between run rate between two subsequent overs are negative when good fielding leads to one or two dismissals in an over.

```
#> # A tibble: 4 x 21
#>   fielding_wckts   `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `9`
#>   <fct>          <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 0               1039   997   972   978   962   957  1002  1012   993
#> 2 1                120   145   171   165   180   181   138   124   140
#> 3 2                  5    10     8     8     9    10     6    10     8
#> 4 3                  0     0     0     0     0     0     0     0     0
#> # ... with 11 more variables: `10` <dbl>, `11` <dbl>, `12` <dbl>,
```

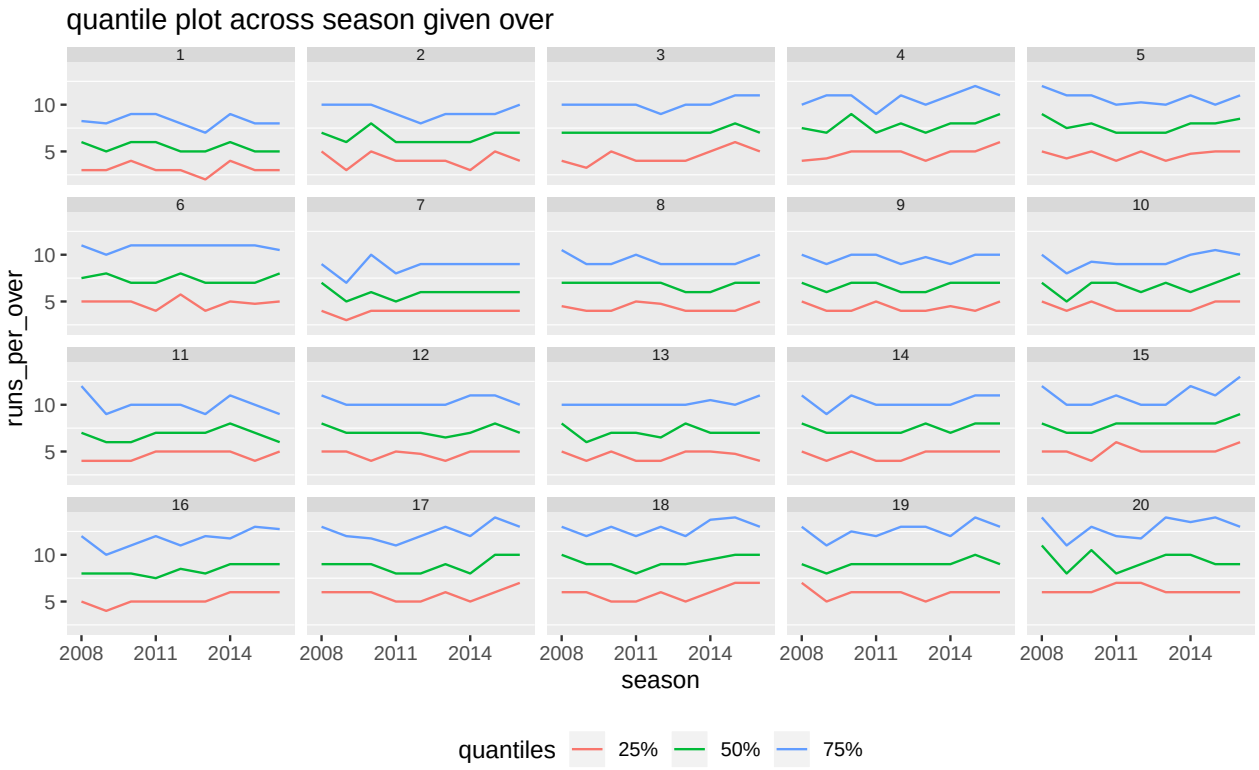


Figure 4: Quantile plot of runs per over across overs of different seasons.

```
#> #   `13` <dbl>, `14` <dbl>, `15` <dbl>, `16` <dbl>, `17` <dbl>,
#> #   `18` <dbl>, `19` <dbl>, `20` <dbl>
```

Q4: Is run rate set to reduce in overs where number of dot balls are more/number of wickets are more?

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 in previous over on run rate 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 **??** shows that for any over, increase in dot balls lead to decrease in run rate.

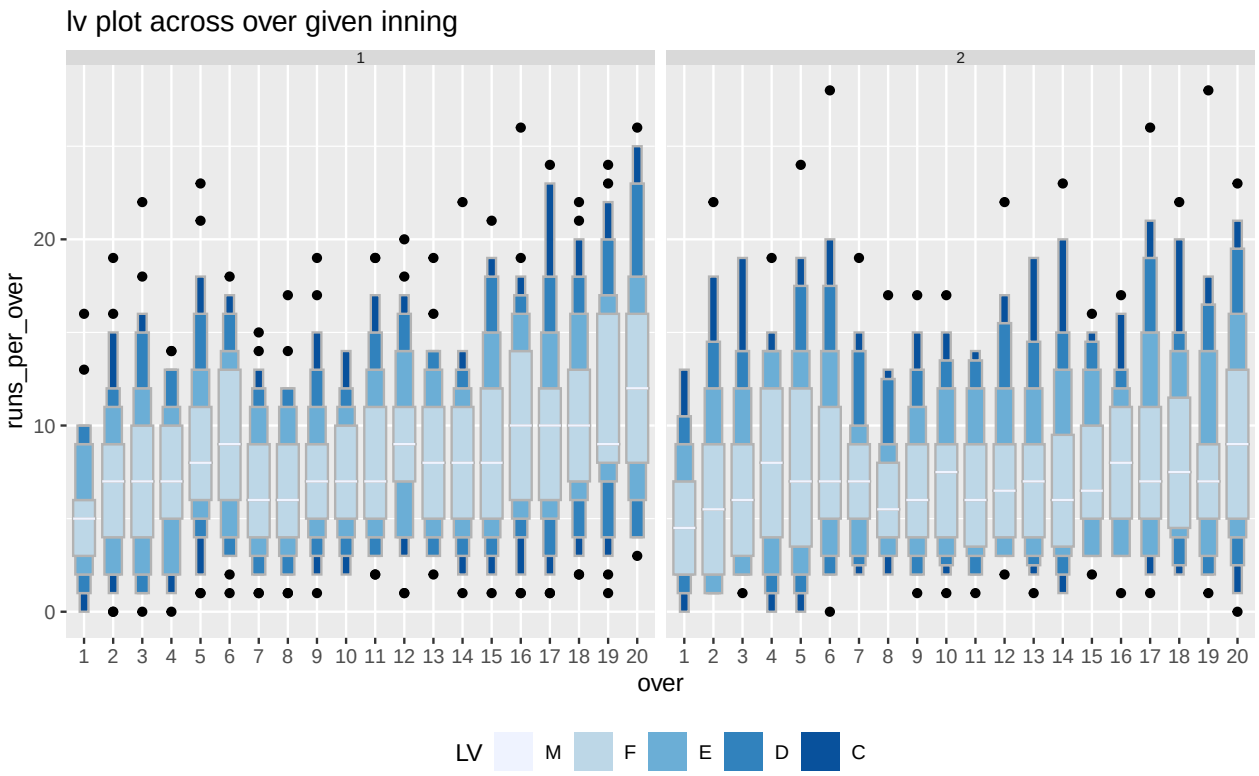


Figure 5: Letter value plot of runs per over across overs of the inning faceted by innings of the match. No upward shift in runs in the second innings like that in the first implying teams are more vulnerable to score more in the first innings as they approach the end of the inning.

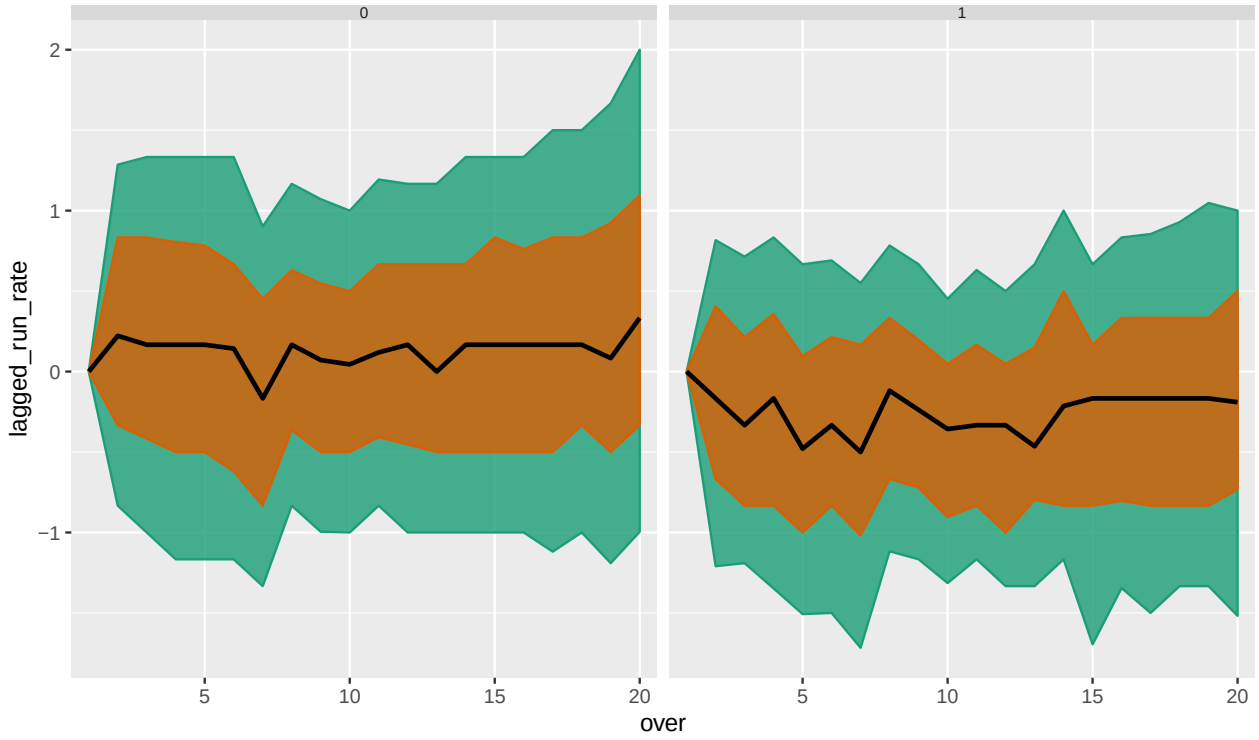
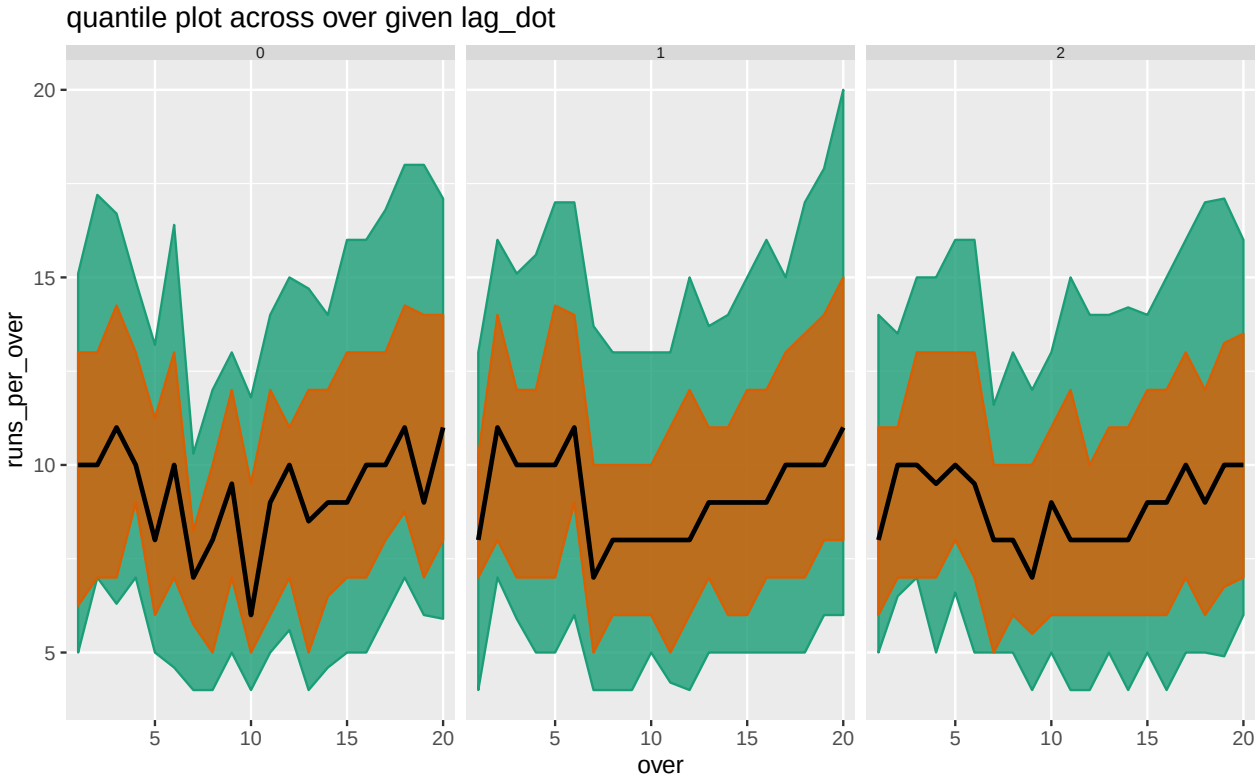
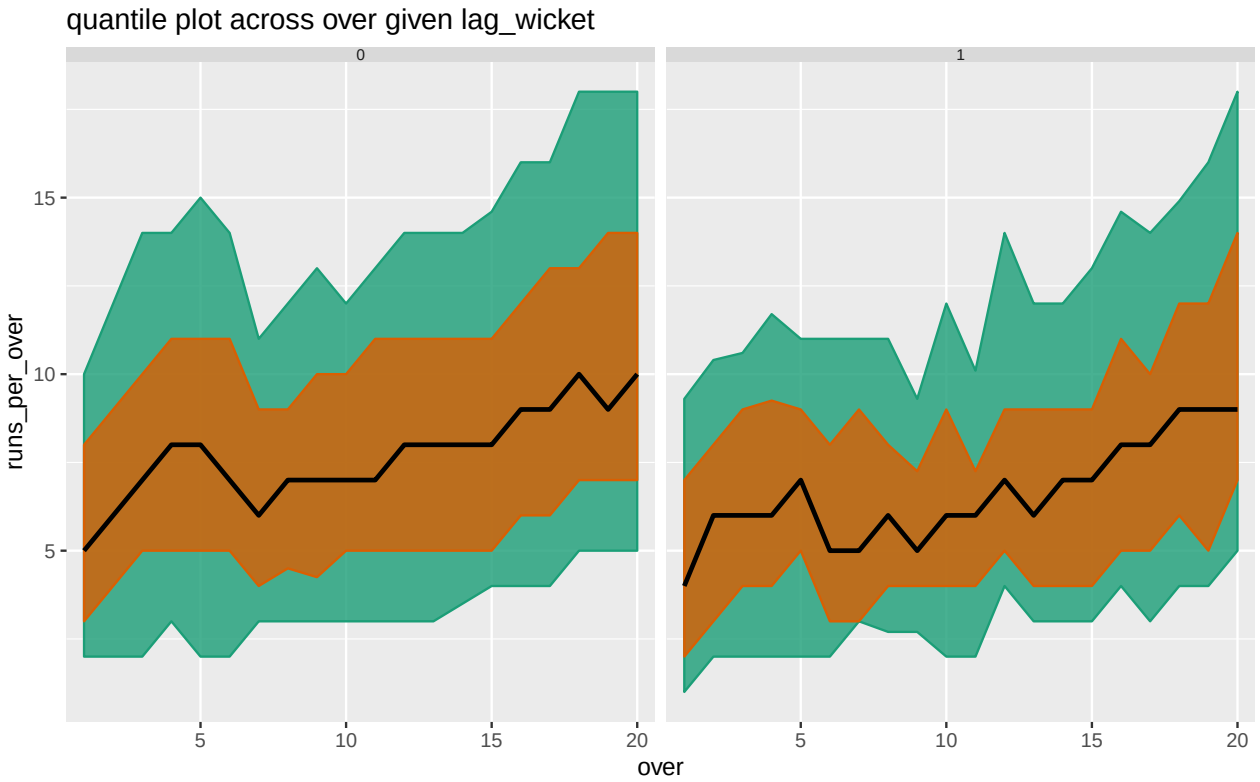


Figure 6: Distribution of lagged run rate across overs of the innings and dismissals by catch and catch and bowled.





Aigner, Wolfgang, Silvia Miksch, Heidrun Schumann, and Christian Tominski. 2011. *Visualization of Time-Oriented Data*. Springer Science & Business Media.

Bettini, Claudio, and Roberto De Sibi. 2000. "Symbolic Representation of User-Defined Time Granularities." *Ann. Math. Artif. Intell.* 30 (1): 53–92.

Bettini, Claudio, Curtis E Dyreson, William S Evans, Richard T Snodgrass, and X Sean Wang. 1998. "A Glossary of Time Granularity Concepts." In *Temporal Databases: Research and Practice*, edited by Opher Etzion, Sushil Jajodia, and Suryanarayana Sripada, 406–13. Berlin, Heidelberg: Springer Berlin Heidelberg.

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

Hintze, Jerry L, and Ray D Nelson. 1998. "Violin Plots: A Box Plot-Density Trace Synergism." *Am. Stat.* 52 (2). [American Statistical Association, Taylor & Francis, Ltd.]: 181–84.

Hofmann, Heike, Hadley Wickham, and Karen Kafadar. 2017. "Letter-Value Plots: Boxplots for Large Data." *J. Comput. Graph. Stat.* 26 (3). Taylor & Francis: 469–77.

Hyndman, Rob J. 1996. "Computing and Graphing Highest Density Regions." *Am. Stat.* 50 (2). [American Statistical Association, Taylor & Francis, Ltd.]: 120–26.

Mcgill, Robert, John W Tukey, and Wayne A Larsen. 1978. "Variations of Box Plots." *Am. Stat.* 32 (1). Taylor & Francis: 12–16.

Ning, Peng, Xiaoyang Sean Wang, and Sushil Jajodia. 2002. "An Algebraic Representation of Calendars." *Ann. Math. Artif. Intell.* 36 (1): 5–38.

Potter, K, J Kniss, R Riesenfeld, and C R Johnson. 2010. "Visualizing Summary Statistics and Uncertainty." *Comput. Graph. Forum* 29 (3): 823–32.

Tukey, John W. 1977. *Exploratory Data Analysis*. Vol. 2. Reading, Mass.

Wang, Earo, Dianne Cook, and Rob J Hyndman. 2019. "A New Tidy Data Structure to Support Exploration and Modeling of Temporal Data," January.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. http://ggplot2.org.

Wilkinson, Leland. 1999. *The Grammar of Graphics*.