Exploring probability distributions for bivariate temporal granularities

Abstract

Smart meters measure energy usage at fine temporal scales, and are now installed in many households around the world. 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 many granularities creates a natural nested ordering. For example, hours are nested within days, days within weeks, weeks within months, and so on. We refer to granularities which are nested within multiple levels as "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 various granularities 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 within the tidy workflow, so that probability distributions can be examined using the range of graphics available in the ggplot2 package.

Contents

1 Introduction

Temporal data can be available at various resolution depending on the context. Social and economic data are often collected and reported at coarses temporal scales like monthly, quarterly or annually. But with recent advancement in technology, more and more data are recorded and stored at much finer temporal scales than that was previously possible. For example, it might be sufficient to observe energy consumption every half an hour, but energy supply needs to be monitored every minute and number of web searches requires

optimisation every second. As the frequency of data increases, the number of questions about the observed variable that need to be addressed 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.

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 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.

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 (**???**) 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.

The hierarchical structure of many granularities creates a natural nested ordering. For example, hours are nested within days, days within weeks, weeks within months, and so on. We refer to granularities which are nested within multiple levels as "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.

lubridate#acad paper in R allows creation of granularities that are mostly single-order-up like hour of the day, second of the minute. This paper utilises 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 visualise 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.

The work is inspired by Wickham et al. (2012), which uses modulus arithmetic to display spatio-temporal data as glyphs on maps. It is also related to recent work by Hafen (2019) which provides methods in the **geofacet** R package to arrange data plots into a grid, while preserving the geographical position. Both of these show data in a spatial context.

In contrast, calendar-based graphics unpack the temporal variable, at different resolutions, to digest multiple seasonalities and special events. There are some existing works in this area. For example, Van Wijk and Van Selow (1999) developed a calendar view of the heatmap to represent the number of employees in the work place over a year, where colors indicate different clusters derived from the days. It contrasts week days and weekends, highlights public holidays, and presents other known seasonal variation such as school vacations, all of which have influence over the turn-outs in the office. Jones (2016), Wong (2013), Kothari and Ather (2016), and Jacobs (2017) implemented some variants of calendar-based heatmaps as in R packages: **TimeProjection**, **ggTimeSeries**, and **ggcal** respectively. However, these techniques are limited to color-encoding graphics and are unable to use time scales smaller than a day. Time of day, which serves as one of the most important aspects in explaining substantial variations arising from the pedestrian sensor data, will be neglected through daily aggregation. Color-encoding is also low on the hierarchy of optimal variable mapping (Cleveland and McGill 1984; Lam, Munzner, and Kincaid 2007).

The proposed algorithm goes beyond the calendar-based heatmap. The approach is developed with three conditions in mind: (1) to display time-of-day variation in addition to longer temporal components such as day-of-week and day-of-year; (2) to incorporate line graphs and other types of glyphs into the graphical toolkit for the calendar layout; (3) to enable patterns related to special events more easily pop-up to viewers. The proposed algorithm has been implemented in the `frame_calendar()` and `facet_calendar()` functions in the **sugrrants** package using R.

The remainder of the paper is organized as follows. Section 2 details the construction of the calendar layout in depth. It describes the algorithms of data transformation (Section **??**), the available options (Section **??**), and variations of its usage (Section **??**). Section **??** explains the full faceting extension, that is equipped with formal labels and axes. An analysis of half-hourly household energy consumption, using the calendar display, is illustrated in a case study in Section 3. Section 8 discusses the limitations of calendar displays and possible new directions.

2 Computing multiple order-up granularities recursively from single order-up granularities

2.1 Time granularities

Time can be represented at varied levels of abstraction depending on the accuracy required for the context. Time granularity can be defined as the resolution power of the temporal qualification of a statement. Providing a formalism with the concept of time granularity makes it possible to model time information with respect to differently grained temporal domains. ### Linear time granularities{#linear-gran-def}

Linear granularities are defined in (**???**), as follows.

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.

The algorithm of transforming data for constructing a calendar plot uses linear algebra, similar to that used in the glyph map displays for spatio-temporal data (Wickham et al. 2012). To make a year long calendar requires cells for days, embedded in blocks corresponding to months, organized into a grid layout for a year. Each month conforms to a layout of 5 rows and 7 columns, where the top left is Monday of week 1, and the bottom right is Sunday of week 5 by default. These cells provide a micro canvas on which to plot the data. The first day of the month could be any of Monday–Sunday, which is deterministic given the year of the calendar. Months are of different lengths, ranging from 28 to 31 days. Some months could extend over six weeks, but for these months the last few days are wrapped up to the top row of the block for compactness, and because it is convention. The notation for creating these cells is as follows:

- $k = 1, \ldots, 7$ is the day of the week, that is the first day of the month.
- $d = 28, 29, 30$ or 31 representing the number of days in any month.
- (i, j) is the grid position where $1 \leq i \leq 5$, is week within the month, $1 \leq j \leq 7$, is day of the week.
- $g = k, \ldots, (k + d)$ indexes the day in the month, inside the 35 possible cells.

The grid position for any day in the month is given by

$$\begin{aligned} i &= \lceil (g \bmod 35)/7 \rceil, \\ j &= g \bmod 7. \end{aligned} \tag{1}$$

Figure 1 illustrates this (i, j) layout for a month where $k = 5$.

2.1.1 Formal conceptualisation of circular time granularities

Suppose we have a tsibble (**???**) with a time index in one column and keys and variables in other columns. A time domain, as defined by Bettini, is essentially a mapping of row numbers (the index set) to the time index. A linear granularity is a mapping of row numbers to subsets of the time domain. For example, if the time index is days, then a linear granularity might be weeks, months or years.

For circular granularities, we need a symbolic representation of time to represent periodicity, which we call 'calendar categorization'. Anything that maps a time index to a categorical variable can be considered a calendar categorization. The number of categories is essentially the periodicity of a circular time granularity.

We want to use modular arithmetic on the domain of the circular granularity to define the calendar categorization. Hence, we start with the definition of equivalence classes and then move on to define a circular granularity.

Definition: Equivalence class Let $m \in N \backslash 0$. For any $a \in Z$ (set of integers), $[a] = \{b \in Z | a \equiv (b \bmod m)\}$ where $[a]$ is defined as the equivalence class to which a belongs.

The set of all equivalence classes of the integers for a modulus m is called the ring of integers modulo m, denoted by Z_m. Thus $Z_m = \{[0], [1], ..., [m-1]\}$. However, we often write $Z_m = \{0, 1, ..., (m-1)\}$, which is the set of integers modulo m.

				k=5, g=5 *i=1, j=5*	*g=k+1* *i=1, j=6*	*g=k+2* *i=1, j=7*
g=k+3 *i=2, j=1*	*g=k+4* *i=2, j=2*	*g=k+5* *i=2, j=3*	*g=k+6* *i=2, j=4*	*g=k+7* *i=2, j=5*	*g=k+8* *i=2, j=6*	*g=k+9* *i=2, j=7*
g=k+10 *i=3, j=1*	*g=k+11* *i=3, j=2*	*g=k+12* *i=3, j=3*	*g=k+13* *i=3, j=4*	*g=k+14* *i=3, j=5*	*g=k+15* *i=3, j=6*	*g=k+16* *i=3, j=7*
g=k+17 *i=4, j=1*	*g=k+18* *i=4, j=2*	*g=k+19* *i=4, j=3*	*g=k+20* *i=4, j=4*	*g=k+21* *i=4, j=5*	*g=k+22* *i=4, j=6*	*g=k+23* *i=4, j=7*
g=k+24 *i=5, j=1*	*g=k+25* *i=5, j=2*	*g=k+26* *i=5, j=3*	*g=k+27* *i=5, j=4*	...	...	*g=k+d* *i=5, j=7*

Figure 1: (ref:month-diagram-cap)

Definition: A **circular granularity** C with a modular period m is defined to be a mapping from the integers Z (Index Set) to Z_m, such that $C(s) = (s \mod m)$ for $s \in Z$.

For example, suppose C is a circular granularity denoting Hour-of-Day and we have hourly data for 100 hours. The modular period $m = 24$, since each day consists of 24 hours and C is a mapping from $1, 2, \ldots, 100$ to $0, 1, 2, \ldots, 23$ such that $C(s) = s \mod 24$ for $s \in 1, 2, \ldots, 100$.

Definition: A **cycle** is defined as the progression of each circular granularity with modular period m through $\{1, 2, \ldots, (m-1), 0\}$ once.

Definition: A **circular granule** represents an equivalence class inside each cycle.

A few categorizations are listed below using modular arithmetic with appropriate period to illustrate further. We assume that the time index is in hours, and n_i is the number of categories created by C_i. Then the following categorizations can be computed.

HOD:	$C_1(s) = s \mod 24$	$n_1 = 24$
HOW:	$C_2(s) = s \mod 168$	$n_2 = 168$
HOM:	$C_3(s) = s \mod 720$ (approximately)	$n_3 = 744$
HOY:	$C_4(s) = s \mod 8760$ (except for leap years)	$n_4 = 8784$
DOW:	$C_5(s) = \lfloor s/24 \rfloor \mod 7$	$n_5 = 7$
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 1: Illustrative calendar categorizations with hourly time index

Note that most of the formulas are approximations only due to unequal month and year lengths, and due to the fact that there are not an integer number of weeks per month or weeks per year.

2.2 Reasons to use calendar-based graphics

The purpose of the calendar display is to facilitate quick discoveries of unusual patterns in people's activities, which is consistent with why analysts should and do use data visualization. It complements the traditional graphical toolbox used to understand general trends, and better profiles vivid and detailed data stories about the way we live. Comparing the conventional displays (Figure **??** and **??**) with the new display (Figure **??**), it can be seen that the calendar display is more informatively compelling: when special events happened, and on what day of the week, and whether they were day or night events. For example, Figure **??** informs the reader that many events were held in Birrarung Marr on weekend days, while September's events took place on Friday evenings, which is difficult to discern from the conventional displays.

3 Case study

The use of the calendar display is illustrated on smart meter energy usage from four households in Melbourne, Australia. Individuals can download their own data from the energy supplier, and the data used in this section is sourced from four colleagues of the authors. The calendar display is useful to help people understand their energy use. The data contains half-hourly electricity consumption in the first half of 2018. The analysis begins by looking at the distribution over days of week, then time of day split by work days and non-work days, followed by the calendar display to inspect the daily schedules.

Figure **??** shows the energy use across days of week using boxplots. Inspecting the medians across households tells us that household 3, a family size of one couple and two children, uses more energy over the week days than other households. The relatively larger boxes for household 2 indicate greater variability in daily energy consumption with noticeable variations on Thursdays, and much higher usage over the weekends. The other two households (1 and 4) tend to consume more energy with more variation on the weekends relative to the week days, reflecting work and leisure patterns.

Figure **??** shows energy consumption against time of day, separately by week day and weekend. Household 1 is an early riser, starting their day before 6am and going back home around 6pm on week days. They switch air conditioning on when they get home from work and keep it operating until midnight, evident from the small horizontal cluster of points around 0.8 kWh. On the other hand, the stripes above 1 kWh for household 2 indicates that air conditioning may run continuously for some periods, consuming twice the energy as household 1. A third peak occurs around 3pm for household 3 only, likely coinciding when the children arrive home from school. They also have a consistent energy pattern between week days and weekends. As for household 4, their home routine starts after 6pm on week days. Figures **??** and **??**, part of a traditional graphical toolkit, are useful for summarizing overall deviations across days and households.

Figure **??** displays the global scaling of each household's data in a calendar layout, unfolding their day-to-day life via electricity usage. Glancing over household 1, their overall energy use is relatively low. Their week day energy use is distinguishable from their weekends, indicating a working household. The air conditioner appears to be used in the summer months (January and February) for a couple of hours in the evening and weekends. In contrast, household 2 keeps a cooling system functioning for much longer hours, which becomes more evident from late Wednesday through Thursday to early Friday in mid-January. These observations help to explain the stripes and clusters of household 2 in Figure **??**. It is difficult to give a succinct description of household 3 since everyday energy pattern is variable, but May and June see more structure than the previous months. Individual data can be idiosyncratic, hence aggregated plots like Figure **??** and **??** are essential for assembling pieces to form a picture. However, the calendar plots speak the stories that are untold by previous plots, for example, their vacation time. Household 1 is on vacation over three weeks of mid-June, and household 2 also took some days off in the second week of June. Further, household 3 takes one short trip in January and the another one starting in the fourth week of June. Household 4 is away over two or three weeks in early April and late June. They all tend to take breaks during June probably due to the fact that the University winter break starts in June.

4 Harmony and clashes based on conditioning - issue of data structure

5 Visualisation

6 Case study: Analysis on cricket

7 Case study: Analysis on smart meter data

8 Discussion

Acknowledgements

We would like to thank Stuart Lee and Heike Hofmann for their feedback on earlier versions of this work. We thank Thomas Lin Pedersen for pointing out some critical **ggplot2** internals, which makes the `facet_calendar()` implementation possible. We are very grateful to anonymous reviewers for helpful comments that have led to many improvements of the paper. The **sugrrants** R package is available from CRAN https://CRAN.R-project.org/package=sugrrants and the development version is available on Github https://github.com/earowang/sugrrants. All materials required to reproduce this article and a history of the changes can be found at the project's Github repository https://github.com/earowang/paper-calendar-vis.

9 Bibliography

Cleveland, William S, and Robert McGill. 1984. "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods." *Journal of the American Statistical Association* 79 (387). Taylor & Francis: 531–54.

Hafen, Ryan. 2019. *Geofacet: 'Ggplot2' Faceting Utilities for Geographical Data*. https://CRAN.R-project.org/package=geofacet.

Jacobs, Jay. 2017. *Ggcal: Calendar Plot Using "Ggplot2"*. https://github.com/jayjacobs/ggcal.

Jones, Holly. 2016. *Calendar Heatmap*. https://rpubs.com/haj3/calheatmap.

Kothari, Aditya, and Ather. 2016. *GgTimeSeries: Nicer Time Series Visualisations with Ggplot Syntax*. https://github.com/Ather-Energy/ggTimeSeries.

Lam, Heidi, Tamara Munzner, and Robert Kincaid. 2007. "Overview Use in Multiple Visual Information Resolution Interfaces." *IEEE Transactions on Visualization and Computer Graphics* 13 (6). IEEE: 1278–85.

Van Wijk, Jarke J., and Edward R. Van Selow. 1999. "Cluster and Calendar Based Visualization of Time Series Data." In *Information Visualization, 1999. INFOVIS 1999 Proceedings. IEEE Symposium on*, 4–9. IEEE.

Wickham, Hadley, Heike Hofmann, Charlotte Wickham, and Dianne Cook. 2012. "Glyph-Maps for Visually Exploring Temporal Patterns in Climate Data and Models." *Environmetrics* 23 (5): 382–93.

Wong, Jeffrey. 2013. *TimeProjection: Time Projections*. https://CRAN.R-project.org/package=TimeProjection.