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 (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.

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.

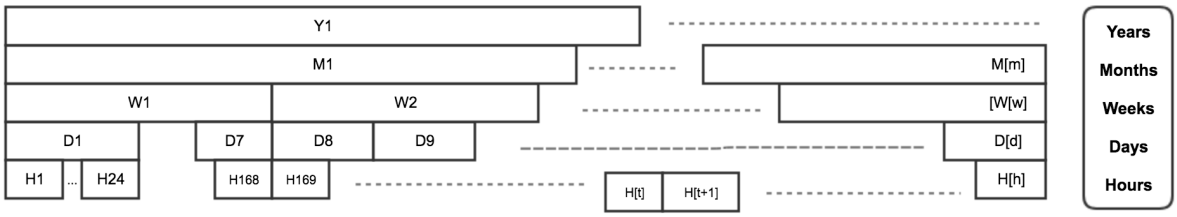

Figure 1: The time domain distributed as linear granularities

- 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 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 is important to model time information across differently grained temporal domains (Euzenat and Montanari 2005).

2.1 Linear time granularities

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, $\lfloor s/(24 * 7) \rfloor$ weeks and so on.

These definitions and rules for linear granularities are inadequate to reflect periodicities in time, like weekly, monthly or yearly seasonality.

2.2 Formal conceptualisation of calendar categorisations

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.

What we need to add to this are additional categorizations of time that are not linear granularities and are useful to represent periodicity. Examples include day-of-week, time-of-day, week-of-year, day-of-month, month-of-year, workingday/non-workingday, etc. Many of these are circular, such as day-of-week, time-of-day. Some are nearly circular such as day-of-month. Some are irregular such as workingday/non-working day. Let's call all of these "calendar categorizations". Anything that maps a time index to a categorical variable can be considered a **calendar categorization**.

We specify the circular categorizations using modular arithmetic and call these **circular granularities**. The number of categories is essentially the periodicity of a circular time granularity. For example, suppose the time index is in minutes, and let n_i be the number of categories created by the circular granularity C_i. Then the following categorizations can be computed.

MOH:	$C_1(s) = s \mod 60$	$n_1 = 60$
MOD:	$C_2(s) = s \mod 1440$	$n_2 = 1440$
HOD:	$C_4(s) = \lfloor s/60 \rfloor \mod 24$	$n_4 = 24$
HOW:	$C_6(s) = \lfloor s/60 \rfloor \mod 24 * 7$	$n_5 = 168$
DOW:	$C_7(s) = \lfloor s/24 * 60 \rfloor \mod 7$	$n_6 = 7$

Table 1: Illustrative circular granularities with time index in minutes

It is easy to note here that due to unequal length of some linear granularities like months or years, these formulae can not be used for computing week-of-month or day-of-year. Thus, we start with the definition of circular granularity and then move on to computing aperiodic granularities.

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

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.

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.

Defintion: An **Aperiodic circular granularity** can not be defined using modular arithmetic. The modulus for these type of calendar categorisations are not constant due to unequal length of some linear granularities.

2.3 Computation of multiple order-up granularities from single order-up granularities

The hierarchical structure of time creates a natural nested ordering, where 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**.

The index of a tsibble can be any date-time object which comes in different formats. (Grolemund, Wickham, and Others 2011) goes a long way in facilitating analysis of date-time data and extracting each date-time element with its own accessor function. However, table shows the accessor functions mostly allows for the creation of single-order-up granularities.

Categorisation	Type	Accessor
Second-of-minute	Periodic	second()
Minute-of-hour	Periodic	minute()
Hour-of-day	Periodic	hour()
Day-of-week	Periodic	wday()
Day-of-month	Aperiodic	mday()
Week-of-year	Aperiodic	week()
Day-of-year	Aperiodic	yday()
Month-of-year	Periodic	month()

The hierarchical nature of time helps to create a framework where the single-order-up granularities can be used recursively to create multiple-order-up granularities.The idea is to incorporate and access different calendar categorisations to account for varied levels of periodicities in the data. The computation of multiple-order-up granularities from single-order-up granularities can differ basis the single-order-up granularities are circular or aperiodic.

2.3.1 All single-order-up granularities are periodic

Definition: A **hierarchy table** has three columns:
- The first column represents the linear granularities in ascending order of temporal hierarchy.
- The second column represents the constant which relates subsequent linear granularities.
- The third column provides the accessor function for single-order up circular granularities.

Definition: **Order** is defined as the position of the linear granularities in the hierarchical table.

Suppose, z is the index set of the tsibble, x,y are two linear granularities with $order(x) < order(y)$. Also, $f_{(x,y)}$ denotes the accessor function for computing circular granularity "x_y" and $c(x, y)$ is a constant which relates x and y. It is easy to see that for $order(x + 1) = order(y)$, the function is same as the single-order-up granularities.

Then, the accessor function f can be used recursively to obtain any multiple-order-up granularities as follows:

$$\begin{aligned} f_{(x,y)}(z) &= f_{(x,x+1)}(z) + c(x, x+1)(f_{(x+1,y)}(z) - 1) \\ &= f(x, x+1) + c(x, x+1)[f(x+1, x+2) + c(x+1, x+2)(f(x+2, y) - 1) - 1] \\ &= f(x, x+1) + c(x, x+1)(f(x+1, x+2) - 1) + c(x, x+1)c(x+1, x+2)(f(x+2, y) - 1) \\ &= f(x, x+1) + c(x, x+1)(f(x+1, x+2) - 1) + c(x, x+2)(f(x+2, y) - 1) \\ &\vdots \\ &= \sum_{i=0}^{order(y)-order(x)-1} c(x, x+i)(f(x+i, x+i+1) - 1) \end{aligned} \tag{1}$$

To elaborate more, let us consider the following hierarchy table.

linear granularities	Conversion factor	Single-order-up accessors
second	60	second_minute
minute	30	minute_hhour
hhour	2	hhour_hour
hour	24	hour_day
day	7	day_week
week	1	1

From Equation(1), we have

$$\begin{aligned} f_(hhour, week)(z) = f_(hhour, hour)(z) + c(hhour, hour) f_(hour, day) + c(hhour, day) f(day, week) \\ = hhour_hour + 2hour_day + 2 * 24day_week \end{aligned} \tag{2}$$

2.3.2 Single-order-up granularities are mixed - that is circular or periodic:

This is valid in a scenario when the relationship between x and (x+1) is periodic (Define periodic in Definition section). However, when x and x+1 is not periodic, then some modification is required in Equation(1).

Consider, a hierarchy table of the form below:

units	conversion_fac
minute	60
hour	24
day	NA
month	3
quarter	2
semester	2
year	1

The table suggests that one hour is composed of 60 minutes, 1 day is composed of 24 hours, 1 quarter is composed of 3 months, 1 semester is consists of 2 quarters and 1 year is made up of 2 semesters. These partitions are periodic, since the number of finer units within each of these coarses units are same always.

Obtaining multiple-order-up granularities from just granularity which relates the lowest temporal unit to the highest one in the hierarchical table.

Suppose, we are not given the intermediate single-order-up granularities but still want all possible multiple order-up granularties. In that case, it is necessary to atleast have the have the functional form that relates the smallest temporal unit to the highest temporal unit. In that case, we are able to form single-order-up from this functional relationship and still use the recursive relation to get the multiple order-up granularities.

In the above equations, it is shown how to obtain f(x,z) given f(x,y) and the hierarchy table, provided $order(z) > order(y) \neq order(x)$. In general,

$f(x, z) = \sum_{i=0}^{order(z)-order(y)+1} c(x, x+i) f(x+1, x+i)$

We could also derive f(w, y) given f(x, y) and the hierarchy table, for $order(y)>=order(w) > order(x).

$f(w, y) = f(x, y) \mod c(x, w)$, for $f(x, y) \mod c(x, w)! = 0 = c(x, w))$, otherwise.

Let us investigate if that is true:

f(ball, over) = 1 f(over, quarter) = 4 f(quarter, semester) = 2 f(semester, match) = ‘1

then, f(ball_quarter) = 1 + 64 = *25, f(ball, semester) = 1 + 64 + 561* = 55 f(ball, semester) = 1 + 64 + *561 + 5621* = 115

3 Harmony and Clashes

We investigate some combinations of circular time granularities which facilitate or hinder exploratory analysis. The combinations of circular granularities which promote the exploratory analysis through visualization are referred to as **harmonies** and the ones which impede the analysis are referred to as **clashes**.

Let's 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. In fact, most of these 155 sets will be empty, making the combination of C_1 and C_2 in a graph unhelpful. These are structurally empty sets in that it is impossible for them to have any observations.

Another example 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. So graphics involving C_1 and C_2 aare potentially useful.

4 Visualisation

Analysts often want to fit their data to statistical models, either to test hypotheses or predict future values. However, improper choice of models can lead to wrong predictions. One important use of visualization is exploratory data analysis, which is gaining insight into how data is distributed to inform data transformation and modeling decisions.

But with huge amount of data being available, sometimes mean, median or any one summary statistic is not enough to understand a dataset. Soon enough following questions become more interesting:

- Are values clustered around mean/median or mostly around tails? In other words, what is the combined weight of tails relative to the rest of the distribution?
- Does values rise very quickly between 25th percentile and median but not as quickly between median and 75th percentile? More generally, how the variation is the dataset changes across different percentiles/deciles?
- Is the tail on the left hand side longer than that on the right side? Or are they equally balanced around mean/median?

This is when displaying a probability distribution becomes a potentially useful approach.

The entire distribution can be visualized or some contextual summary statistics can be visualised to emphasize certain properties of the distribution. These properties can throw light on central tendency, skewness, kurtosis, variation of the distribution and can also be useful in detecting extreme behavior or anolmalies in the dataset.

4.1 Statistical distribution plots

Most commonly used techniques to display 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 vase plot (Benjamini 1988, 1988) was a major revision from the concept of box plots where the width of box at each point is proportional to estimated density. Violin plots (Hintze and Nelson 1998, 1998) display the density for all data points and not only the box. 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. 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. 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. Because this display just adds extra letter values, it suffers from the same problems as the original box plot, and multimodality is almost impossible to spot(Wickham and Stryjewski, n.d.).

Moreover, much like the quartiles divide the dataset equally into four equal parts, extensions might include dividing the dataset even further. The deciles plots consist of 9 values that split the dataset into ten parts and the percentile plot consists of 99 values that split the dataset into hundred parts. A large dataset is required before the extreme percentiles can be estimated with any accuracy.

Finally, a density plot which uses a kernel density estimate to show the probability density function of the variable can show the entire distribution. Also, 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.

4.2 Distributions conditional on bivariate granularities

The time series variable can be plotted against many time granularities to get more understanding of the underlying periodicty, however, we will restrict ourselves to see the distribution of the time series across bivariate temporal granularities. That neccessiates plotting one temporal granularity along the x-axis and the other one across facets.

Now, due to the hierarchical arrangement of the granularities, there are certain granularities which when plotted together do not give us the layout to do exploration, for example, structurally empty combinations (clashes) are not recommended to plot together. The harmonies when plotted together can help exploration. But still the question remains that which distribution plot should be chosen to bring out the best of exploratory data analysis. This is a function of which features of the distribution we are interested to look at, how much display space is available to us and also if the number of observations are enough for that distribution plot.

4.3 Advice algorithm for exploring conditional probability distributions

Recommendations for distribution plots depend on the levels(very high/high/medium/low) of the two granularities plotted. They will vary depending on which granularity is placed on the x-axis and which one across facets. Assumptions are made to ensure display is not too cluttered by the space occupied by various kinds of distribution plots. Moreover, the recommendation system ensures that there are just enough observations before choosing a distribution plot.

5 Case study: Analysis on smart meter data

6 Case study: Analysis on cricket

7 Discussion

Acknowledgements

8 Bibliography

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

Benjamini, Yoav. 1988. “Opening the Box of a Boxplot.” *Am. Stat.* 42 (4). Taylor & Francis: 257–62.

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.

Euzenat, Jerome, and Angelo Montanari. 2005. “Time Granularity.”

Grolemund, Garrett, Hadley Wickham, and Others. 2011. “Dates and Times Made Easy with Lubridate.” *J. Stat. Softw.* 40 (3): 1–25.

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.

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.

Wickham, Hadley, and Lisa Stryjewski. n.d. “40 Years of Boxplots.”