gravitas: exploring probability distributions for bivariate temporal granularities

Student information

Name: Sayani Gupta

University: Monash University, Australia

Email: gupta.sayani@gmail.com

Student Email : Sayani.gupta@monash.edu

Github: https://github.com/Sayani07

Twitter: https://twitter.com/SayaniGupta07

Timezone: AEST (UTC + 11:00)

Abstract

Project gravitas aims to provide methods to operate on time in an automated way, to deconstruct it in many different ways. Deconstructions of time that respect the linear progression of time like days, weeks and months are defined as linear time granularities and those that accommodate for periodicities in time like hour of the day or day of the month are defined as circular granularities or calendar categorizations. Often visualizing data across these circular granularities are a way to go when we want to explore periodicities, pattern or anomalies in the data. Also, because of the large volume of data in recent days, using probability distributions for display is a potentially useful approach.The project will provide these techniques into the tidy workflow, so that probability distributions can be examined in the range of graphics available in the ggplot2 package.

Motivation

Considerable data is accumulated by sensors today. An example is data measuring energy usage on a fine scale using smart meters. Smart meters are installed on many households in many countries now. Providing tools to explore this type of data is an important activity. Probability distributions are induced by various aggregations of the data, by temporal components, by spatial region or by type of household. Visualizing the probability distribution of different households across different circular granularities will help energy retailers find similar households or understand consumer behavior better and consequently increase efficiency in planning ahead.

Project gravitas

Project gravitas will consist of four parts:

1. **R Package**: Develop an R package consisting of modules **BUILD** and **COMPATIBLE**
2. **Shiny UI**: Develop an user interface using RShiny to enable user to walk through different modules of the package
3. **Application**: Provide examples of probability visualization of smart meter data collected on Australian households
4. **Vignette**: Document the R package functionality in a vignette

Module BUILD

The module **BUILD** will provide the methods to exhaustively construct any granularities. Some of the functions that will be created as part of this module are as follows:

Function name	Description	Arg 1	Arg 2
ghour (.data, .grantype)	define granularities in combination with hour, e.g. hour of the week	a tsibble object	week
gqhour (.data, .grantype)	define granularities in combination with quarter hour, e.g. quarter hour of the day	a tsibble object	day
gweek (.data, .grantype)	define granularities in combination with week, e.g. week of the month	a tsibble object	month

The data fed is a tsibble so that time indices are preserved as an essential data component. The function will create granularities in combination with any time index that are one or multiple levels higher in resolution than the time index in the given tsibble.

The idea in this module is to have exhaustive set of granularities at our disposal so that we can use them in later module to figure out periodicities, patterns or anomalies in the data.

Module COMPATIBLE

The module **COMPATIBLE** will provide checks on the feasibility of plotting or drawing inference from two granularities together. The function will categorize pairs of granularities as either a harmony or clash, where harmonies are pairs of circular granularities that aid exploratory data analysis. Clashes are pairs that are incompatible with each other for exploratory analysis. This module will provide appropriate data structures to visualize with the grammar of graphics for harmonies.

Function name: compatible

Description: provide checks on the feasibility of plotting or drawing inference from bivariate temporal granularities

Usage: compatible(.data, build1, build2)

Arguments:

.data - A tsibble

build1, build2 - granularities computed in module BUILD through functions like gqhour, ghhour, ghour, gweek, gmonth, gquarter, gsemester, gyear. gqhour and ghour imply granularities that are obtained in combination with quarter of an hour and hour respectively. Can be numeric vector or functions.

Value: a data structure specifying number of observations per combination of the two builds, variation in the number of observations across combinations and declare them as “harmony” or “clash”. These suggestions will serve as a guide to users who are looking to explore distributions of variables across different harmonies.

Related work

- lubridate is an R package that makes it easier to work with time and also has functions for creating calendar categorizations like hour of the day, day of the week, minutes of the hour. But it mostly creates calendar categorizations that are one step up. The proposed R package will allow creating calendar categorizations that are more than one step ahead, for example, hour of the week or one step up that are not present in lubridate package like week of the month.
- Calendar based graphics in the package sugrrants help explore data across linear time granularities in a calendar format, whereas this package would help explore circular time granularities.

- ggplot2 facilitates the process of mapping different variables to a 2D frame through grammar of graphics. But it does not tell us which all variables to plot together to promote exploration of data. The proposed package would provide the list of harmonies given a time variable.
- This will use as inputs tsibble objects which complement the tibble and extend the tidyverse concept to temporal data.

Brief Timeline

- Phase 0 | Pre-GSoC Period | 11 Apr - 6 May: Literature reading, compiling application and test data
- Phase 1 | Community Bonding Period | 7 May: 26 May: Development related setup and brainstorming ideas
- Phase 2 | Coding Period 1 | 27 May - 23 June: Complete R package with CRAN tests
- Phase 3 | Phase 1 Evaluations | 24 June - 28 June: Communicate with social media community for inputs and code review
- Phase 4 | Coding Period 2 | 29 June -22 July: Incorporate suggestions from community and build Shiny UI for the package
- Phase 5 | Phase 2 Evaluations | 23 July - 26 July: Communicate with social media community for inputs and features review and create user documentation of shiny UI
- Phase 6 | Coding Period 3 | 27 July - 18 August: R package vignette, application on Australian smart meter data and improving features based on suggestions from community
- Phase 7 | Final week of submitting finished product | 19 August - 26 August: Wrap-up and finish working on remaining issues
- Phase 8 | Final evaluation | 26 August - 2 September: Blog on analysis of Australian smart meters using R package developed and upload progess report

Detailed Project Timeline

Phase		Time frame	Task Description
0	Pre-GSoC Period (11 Apr - 6 May)	weeks 1-3	Literature reading, and outline broad range of functions to be developed for each modules. Compiling range of applications and data, additional to smart meters, and creation of small testing sets.
1	Community Bonding Period (7 May - 26 May)	weeks 1-2	Lay out the road map for the project with the guidance of the mentors and establish modes of communication. Create and share github site for the project, and set up issues, code structure, invite mentors as collaborators. Add travis CI to site to continuously check code as it is developed. Brainstorm the ideas with mentors, range of applications and test data sets compiled, and code structure to be developed.

Phase		Time frame	Task Description
2	Coding Period 1 (27 May - 23 June)	week 1	Code simple granularity functions (BUILD). Create unit tests.
		week 2	Code remaining granularity functions (BUILD), and unit tests. Document with roxygen. Provide example code on test data sets.
		week 3	Code advice functions (COMPATIBILTY), and unit tests. Document with roxygen. Provide example code on test data sets.
		week 4	Documentation in the form of a vignette on broader range of data applications. CRAN tests completed.
3	Phase 1 Evaluations (24 June - 28 June)		Announce package base to social media community, and request input, and code review. Report on current progress at project's wiki page after feedback from mentors. Deliverables: 1. **R package with granularity and advice functions** 2. **documentation with roxygen** 3. **unit tests** 4. **example code on test data** 5. **CRAN tests**6. **progess report**
4	Coding Period 2 (29 June - 22 July)	week 1	Address issues and improvements from community on functions (BUILD and COMPATIBILITY) and make the package ready for submission to CRAN.
		weeks 2-3	Develop shiny UI to guide an user to navigate through the modules of the R package.
5	Phase 2 Evaluations (23 July - 26 July)		Create user documentation for the shiny app. Announce shiny app to social media community, and request input, code and features review. Report on current progress at project's wiki page after feedback from mentors. Deliverables: 1. **address issues and improvements on R package** 2. **Shiny UI** 3. **documentation for shiny app** 4. **progess report**
6	Coding Period 3 (27 July - 18 August)	week 1	Work on improving the shiny UI features based on reviews and issues.
		week 2	Document the R package functionality in a vignette
		week 3	Provide examples of probability visualization of smart meter data collected on Australian households

Phase	Time frame	Task Description
7	Final Week of submitting product (19 August - 26 August)	Use as buffer to wrap up and address remaining issues. Document ideas that resulted from discussion with mentors and community but could not be implemented due to time limitations. These can then form the basis of future work.
8	Final evaluation (27 August - 2 September)	Announce vignette to social media community. Write a blog on analysis of Australian smart meter data exemplifying the usefulness of functions developed in this project . Report on all work done at project's wiki page after feedback from mentors. Deliverables: 1. **improve Shiny UI features** 2. **R package vignette** 3. **application on Australian smart meters** 4. **progess report**

Additional information regarding timeline

- The above timeline is tentative and gives a rough idea of my planned project work. I've no major commitments during summer (winter in Australia) and hence, will be able to dedicate 40 hours to 50 hours a week. During the last month of the project, my university will reopen and I'll be able to dedicate around 30 hours a week. I plan to complete major chunk of the work before university reopens.
- I'll publish blogs at the end of each coding phase that will include highlights of the development process, hurdles that I came across and how I overcame them. Also, I would document some good practices that I learnt while looking at codes from developers of R community who are working in similar fields.
- I'll be setting up weekly meetings with my mentors where I update them on where I am on the project and discussing ideas on improving functions and features.

Mentors

- Dianne Cook dicook@monash.edu
- Antony Unwin au50au@me.com

Brief bio

I'm a PhD student in the Department of Econometrics and Business Statistics at Monash University, Australia. My research interests are in data visualization, exploratory analysis, time series analysis and computational statistics. I had completed M.Stat (Masters in Statistics) from the Indian Statistical Institute with Quantitative Economics specialization and Bachelors (Honors) in Statistics from St.Xavier's College, University of Calcutta, following which I had worked with KPMG and PAYBACK in the Analytics/ Modelling team. Although I have had a training in Pure Statistics, I have always been fascinated by the role of statistics in real world applications. I love working with data and create useful tools that can be deployed by users in different application. One of the interesting projects that I had implemented using R can be found here.

Thus, I have used R as a tool to implement statistical techniques earlier. However, looking at it from the perspective of open source technology has given it a totally different meaning. It made me realize that working on an open source project amounts to improving the quality of the final product through transparent collaboration with others. Moreover, I also learnt the importance of documented and easy to read codes. Some of my open source contributions can be found here.