tsfeaturex: An R Package for Automating Time Series Feature Extraction

Dr. Nelson A. Roque¹ and Dr. Nilam Ram²

¹ T32 Postdoctoral Fellow, Pennsylvania State University, Center for Healthy Aging, University Park, PA, USA
² Professor, Pennsylvania State University, Human Development & Family Studies, University Park, PA, USA

Statement of Need

In today’s digital world, data collection and storage costs are quite low. Humans are collectively outputting 2.5 quintillion bytes of data every day; by 2020, each person will generate ~ 1.7 MB every second (IBM Marketing Cloud, 2017). At this scale, intensive longitudinal data about humans’ behavior facilitates new discovery about the patterning of thought and action and potentially better prediction and optimization of health and well-being. In raw form, the 2.5 quintillion bytes of data generated daily are noisy time-series and difficult to interpret. Extraction of features from the time-series, however, allows:

1. Researchers to reduce the dimensionality of their time-series data (e.g., reducing millions of time-stamped observations to, for example, summary feature vector of length 100);
2. Summary characterizations of time-series data that may be used as predictors, correlates, or outcomes in study of between-person differences; and
3. Improved and detailed description of human behavior streams (e.g., characterizing a behavioral time series in terms of its features; the mean is ‘X’, the range is ‘Y’, the peaks are at ‘T12’ and ‘T30’).

Short data streams are easily summarized using basic features (e.g., mean, standard deviation, IQR). However, as the time-series get longer, numerous other features may be needed and/or can be accessed. Study of intraindividual variability has outlined the wide variety of time-series features that can be used to characterize between-person differences and within-person change – with features such as probability of acute change (PAC) or mean square of successive differences (MSSD) providing useful information about individuals’ cognitive, emotional, and behavioral dynamics (for more info on intraindividual variability metrics, see: (Jahng, Wood, & Trull, 2008)).

Summary

Functionality

tsfresh is an R package for automating time series feature extraction, inspired and modeled after the Python package tsfresh (blue-yonder, 2016a; Christ, Braun, Neuffer,
& Kempa-Liehr, 2018). The R language (R Core Team, 2019) allows for an easy to use interface, with the underlying processing speed advantage of C languages (and flexibility to run on the web, with the help of the shiny package in R (Chang, Cheng, Allaire, Xie, & McPherson, 2019)). The API for tsfeaturex is a wrapper for the highly-trafficked dplyr (Wickham, François, Henry, & Müller, 2019), mainly to lend on the flexibility of the grammar of data manipulation and shortcuts for non-standard evaluation. The API for tsfeaturex was designed to facilitate the extraction of features for any dataset in long format, including grouping of summaries by other factor. For example, if every person in your dataset has one observation each day for eight days, and they do this in two bursts, once every six months, you can calculate features of the overall series (i.e., 16 observations from both bursts, or separately for each burst). Some features are integrated from other packages, such as e1071 (Meyer, Dimitriadou, Hornik, Weingessel, & Leisch, 2019), Hmisc (Harrell Jr, Charles Dupont, & others., 2019), forecast (Hyndman & Khandakar, 2008), zoo (Zeileis & Grothendieck, 2005), viridis (Garnier, 2018), psych (Revelle, 2018), entropy (Hauser & Strimmer, 2014), and Langevin (Friedrich, Peinke, Sahimi, & Reza Rahimi Tabar, 2011; Rinn, Lind, Wächter, & Peinke, 2016).

By design, tsfeaturex is able to cope with missing data (in R, of class NA), a key deviation from tsfresh (blue-yonder, 2016b). In addition to feature extraction, this package also calculates feature correlations amongst outputted features. tsfeaturex is capable of outputting both long and wide data structures – both of use for different purposes (e.g., long format preferred for plotting in ggplot2) and analyses (e.g., wide format preferred for repeated measures ANOVA in most statistical software).

Purpose & Audience

tsfeaturex is intended for use by researchers with time-series data, and will be of most interest to those developing their statistical and coding skills – allowing them to extract many features from their time-series data with easy to use code and without need for high-level mathematics background. The desire for feature extraction tools is widespread across all domains of data science, including, but not limited to, applications in: biological systems, finance, and psychology.

Feature Roadmap

The current expectation is that over time, tsfeaturex, will allow for two-levels of feature extraction from almost any data form (e.g., text, audio, images): (1) extracting time-series descriptive features from numerical data (already implemented); (2) extracting numerical features from non-numerical data (e.g., number of exclamation points in Twitter data; coming soon).

Figures

Figure 1 depicts example wide(top) and long(bottom) data structures for a dataset containing two (2) measurements from two (2) individuals. Notice that there is one row for each individual in the wide format, and two (2) rows for each individual in the long format, one for each column.

Figure 2 depicts sample time series data from two participants, both with mean value of 5. You will notice, although they have identical means, the shape of the time series, and locations of peaks is different. tsfeaturex calculates features to better characterize differences such as these.
Wide format

<table>
<thead>
<tr>
<th>ID</th>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>500</td>
<td>550</td>
</tr>
<tr>
<td>R2</td>
<td>600</td>
<td>650</td>
</tr>
</tbody>
</table>

Long format

<table>
<thead>
<tr>
<th>ID</th>
<th>Column #</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>1</td>
<td>500</td>
</tr>
<tr>
<td>R1</td>
<td>2</td>
<td>550</td>
</tr>
<tr>
<td>R2</td>
<td>1</td>
<td>600</td>
</tr>
<tr>
<td>R2</td>
<td>2</td>
<td>650</td>
</tr>
</tbody>
</table>

Figure 1: Figure 1. Flexible data structure output – request long or wide format

Figure 2: Figure 2. The mean doesn’t tell the whole story

Mentions of Ongoing Projects

tsfeaturex is currently being used in analysis of experience sampling and multi-trial performance data in a variety of projects at the interface of data science and psychological science, including:

- Intraindividual Study of Affect, Health, and Interpersonal Behavior (iSAHIB)
  - Learn More About This Project
- Midlife in the United States (MIDUS), National Study of Daily Experiences
  - Learn More About This Project
- Einstein Aging Study (EAS)
  - Learn More About This Project
- Effects of Stress on Cognitive Aging, Physiology, and Emotion (ESCAPE)
  - Learn More About This Project

Acknowledgements

Nelson A. Roque was supported by National Institute on Aging Grant T32 AG049676 to The Pennsylvania State University.

We thank Github user @blue-yonder, and other contributors, for creating tsfresh (https://github.com/blue-yonder/tsfresh) and inspiring tsfeaturex. We would like to acknowledge and thank Github user @stas-g, for code on finding peaks (stas-g (2017)), and Dr. Nilam Ram for code on calculating probability of acute change (PAC).

We gratefully acknowledge contributions from Dr. Nilam Ram, Dr. Anthony Ong, Dr. Martin Sliwinski, and the Sliwinski lab throughout the early development process.

References


blue-yonder. (2016b). Allow NaN or None values to be passed in, and silently ignored. GitHub repository. https://github.com/blue-yonder/tsfresh/issues/90; GitHub.


