timesquirrel

Release 0.7rc

December 07, 2014

Contents

timesquirrel is a python module that provides simple functionality for storing and recalling pandas timeseries data in sqlite3 databases.

Contents

CHAPTER 1

What is "timesquirrel" for?

The purpose behind **timesquirrel** is to simplify record-keeping using python. pandas itself provides two commonly used approaches for I/O with dataframes:

- *pandas.HDFStore:* Stores the dataframe in an HDF5 file using the pytables layout. This layout is not easily read by other languages, and is known to exhibit extreme file size bloat if a dataset is continually appended to (as is common for timeseries data). This is known as the repack problem, and the author has personally seen H5 files occupying >100GB space for 100MB of data (once repacked with h5repack).
- *pandas.io.sql:* Supports creating and writing tables in a sqlite database, but does not have many options, particularly regarding table creation and dynamics of the INSERT operation. The API appears to be rapidly evolving at the time of writing.

The goal is to provide functionality similar to HDFStore with the speed and support of the sqlite database engine.

1.1 What is "timesquirrel" not?

timesquirrel is not intended to be a panacea, and in particular it is not intended to be a general SQL-python interface. It only handles pandas dataframes with a TimeSeriesIndex, or an index whose entries entirely consist of datetime-like objects (as determined by pandas). However, the entries in the datatable can be any datatype that can be stringified!

CHAPTER 2

How does sqlite handle datetime entries?

SQLite stores entries as strings, and datetime objects as ISO8601-like timestamps. In particular, it follows the Javascript-convention of storing datetimes without the "T" as

YYYY-MM-DD HH:MM:SS

This means that sorting by datetime is equivalent to a string sort! Note that it does not recognise milliseconds or timezones - these must be converted first and/or stored elsewhere.

sqlite3 does provide a number of datetime functions for converting between different acceptable representations (including timezones) but entries in the database *must* conform to this definition. In particular, the *datetime* function will accept a variety of input specifications (including "now") and produce an acceptable timestamp for the database.

CHAPTER 3

Indices and tables

- *genindex*
- *modindex*
- *search*