

# Automated test generation for `ctsa`

@hgfernan

January 30, 2025

## Abstract

A routine for the automated test generation for the statistical library `ctsa` is outlined. It involves the generation of simple tasks of model fitting and prediction using `ctsa`, compared with equivalent code in the Python libraries `pmdarima` and `statsmodels`, `simple`, and in the R library `forecast`.

## 1 Motivation

## 2 Introduction

*Overview*

## 3 Tables

Since the automated test generation is based on a database, here follows a description of the database to be used.

It has a mixed relational and document architecture: the main tables follow a conventional relational structure, but the parameters and test results are stored as JSON values. That's for pragmatic reasons: the parameters and test results are varied and have different structures. That could be easily mapped to a relational database structure, but it would be too laborious and cumbersome.

As such, test parameters and results will be stored as JSON objects encoded as strings, and dealt with by classes specialized in their content. That is: an *ARIMA* model will have three basic parameters, while a *SARIMA* model will have 7 basic parameters. There will be a class for each one of the models, that will be able to unpack and allow the use those JSON values.

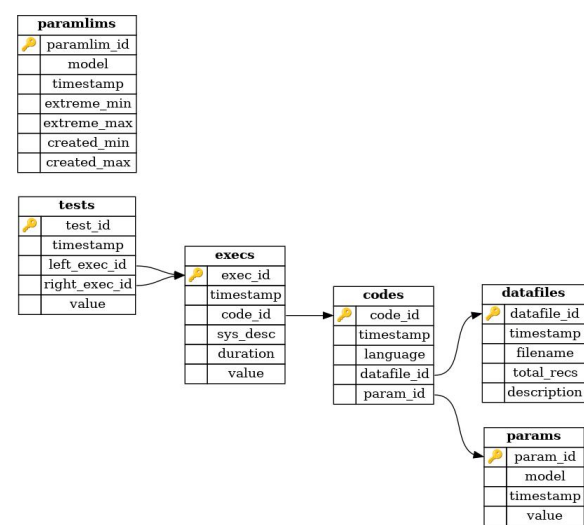


Figure 1: Database used for automated test generation

In a few words, the algorithm to generate tests follow this way:

1. The table `paramlims` (on the upper left corner of Figure 1) is used to select a sequence of parameters for each model, according to the limits present in the fields `extreme_min` and `extreme_max`. They are stored in the table `params` to be detailed below.

Not all parameters are generated, and the range of the parameters created are saved in the fields `created_min` and `created_max`.

The field `timestamp` contains the last alteration of any or all values in the range `created_min` and `created_max` for its corresponding model;

2. The list of data files available (stored in the folder `data/`) are listed and each name is contained in the field `filename` of the table `datafiles`. In this table the date of each file inclusion is recorded in the field `timestamp`. The field `total_recs` of the same table contain the number of records of each file. The field `description` can be used to introduce details of the file: its origin, the transformations used to generate it, etc.
3. There are simple templates for each model, and they're used to generate a single file for each of the elements of the cartesian product between the allowed range of `params` for each model, and the data files in `datafiles`.

This process is replicated for each of the languages in use (C, Python, and R) and their corresponding libraries `ctsa`, `pmdarima` and `statsmodels`, and `forecast`. Those results are stored in the table `codes`;

- 4.