

**Аналоги роботи:** Розробка Доменно-Специфікованої мови програмування для опису часових рядів

**Автор:** Глушко Богдан Сергійович, ІП-71

**Науковий Керівник:** ас. Очеретяний Олександр Костянтинівич

### Опис:

Найвідомішим продуктом, що забезпечує функціонал аналізу часових рядів, є розробка компанії Empirical Solutions, однойменна предметно-орієнтована мова Empirical. За своєю структурою, розробка є класичною декларативною мовою програмування, проте предметно-орієнтована складова у ній забезпечена окремим типом даних - так званими дата-фреймами, підоб'єкт яких містить функції для аналізу часових рядів.

Сам приклад часового ряду реалізований у вигляді табличного синтаксису, з можливістю ідентифікації певних полів:

```
>>> from trades select where symbol == "AAPL" and size > 1000
symbol          timestamp      price size
AAPL 2019-05-01 09:37:45.647850 205.0600 1010
AAPL 2019-05-01 09:38:24.754932 204.9200 2010
AAPL 2019-05-01 09:42:57.450065 203.7332 1130
```

Основною відмінністю мови від класичних аналогів є статична типізація - таким чином, існування та призначення полів відоме компілятору ще до ініціалізації користувацького коду:

```
>>> sort quotes by (asks - bid) / bid
Error: symbol asks was not found
```

Саме ця особливість відрізняє мову від математично-спрямованих мов, що пропонуються для задач аналізу часових рядів: Julia, R, q/kdb+. Такі мови, попри можливість використання для аналізу, однак, належать до різновиду мов загального спрямування, і для взаємодії з структурами часових рядів використовують додаткові бібліотеки. Ось короткий перелік деяких з них:

- NumPy for Python - одне з популярних математичних розширень для мови Python забезпечує користувача не лише базовим пакетом функціональних компонентів, проте також пропонує використання

системи часових штампів `datetime64`, що за структурою є векторизованим типом з підтримкою стандартних часових штампів мови. Після ініціалізації точки входу для такого масиву, користувач може застосовувати векторизовані операції, як-от на прикладі нижче:

```
In [4]: import numpy as np
        date = np.array('2015-07-04', dtype=np.datetime64)
        date

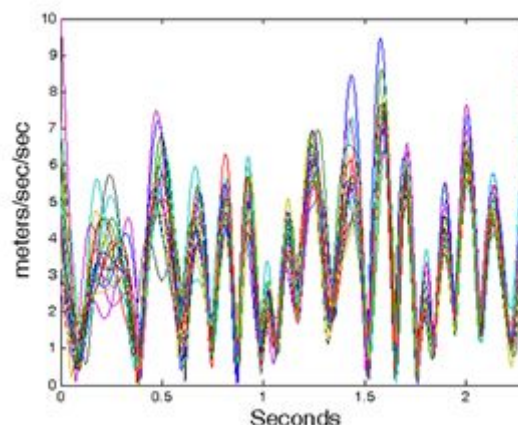
Out[4]: array(datetime.date(2015, 7, 4), dtype='datetime64[D]')
```

Once we have this date formatted, however, we can quickly do vectorized operations on it:

```
In [5]: date + np.arange(12)

Out[5]: array(['2015-07-04', '2015-07-05', '2015-07-06', '2015-07-07',
              '2015-07-08', '2015-07-09', '2015-07-10', '2015-07-11',
              '2015-07-12', '2015-07-13', '2015-07-14', '2015-07-15'], dtype=
```

В контексті мови R, існує бібліотека FDA (скорочення від Functional Data Analysis), що дозволяє подавати на вхід векторизовані типи даних, надалі застосовуючи елементи трансформації даних. Подальша обробка дає змогу виконувати класичні підходи до аналітики даних, а також порівнювати графіки різних часових рядів в рамках єдиної осі:



Основним недоліком кожного з цих інструментів є широка орієнтованість та відсутність предметності: окрім поставленої задачі аналізу часових рядів, кожна з мов надає широкий інструментарій

для загальної розробки. Це призводить до відсутності підтримки продукту як такого, вузької направленості для виконання конкретних задач та малого обсягу функцій, доступних для комплексного аналізу обраної сфери задач.

Це доводить актуальність обраної теми дипломної роботи, демонструючи актуальність продукту в контексті комплексних предметних досліджень, пов'язаних із роботою над часовими рядами.