

Санкт-Петербургский государственный университет
Прикладная математика и информатика

Отчет по производственной практике (педагогической) (семестр 3)

НАПИСАНИЕ ФАЙЛА ПОМОЩИ ДЛЯ SSA НА PYTHON

Выполнил:

Козак Михаил Валерьевич

группа 21.M03-мм



Научный руководитель:

к. ф.-м. н., доцент

Голяндина Нина Эдуардовна

Кафедра Статистического Моделирования

Оглавление

Глава 1. Введение	3
Глава 2. Документация для pyrssa	5
2.1. Создание заготовок для веб-страниц	5
2.2. Написание документации	9
2.3. Публикация сайта	12
Глава 3. Заключение	13
Список литературы	14

Глава 1

Введение

Singular spectrum analysis (SSA) [1] – метод для анализа временных рядов, основанный на преобразовании одномерного временного ряда в многомерный, с последующим применением к нему метода главных компонент. Данный метод находит применение в различных сферах: климатология, океанология, геофизика, техника, обработка изображений, медицина, эконометрика и многие другие.

Данный метод активно применяется в обучении студентов, в связи с чем требуется своевременная адаптация инструментов, позволяющих применять его на практике. С этой целью для использования метода SSA в популярном языке программирования Python ведется разработка пакета `pyrssa` [2], который является адаптацией существующей для языка R библиотеки `Rssa` [3, 4], использующей алгоритмы, реализованные на языке C с целью повышения скорости работы.

Пакет `Rssa` содержит исчерпывающую документацию, в которой есть как информация по самим функциям библиотеки и примерам их использования, так и ссылки на литературу с описанием алгоритмов, которые легли в основу пакета. Вся эта информация позволяет обучающимся достаточно быстро получать справку и о методах, и об инструментах для их применения. Необходимость переноса существующей документации вызвана в первую очередь тем, что адаптированная для Python версия пакета в силу отличий от языка R содержит изменения как в структуре модулей, так и в доступных и используемых типах данных. В силу отличий синтаксиса языков возникла необходимость произвести адаптацию примеров использования пакета для документации, а также для удобства обучающихся: вся необходимая информация должна быть доступна пользователям сразу и в наглядном формате.

Таким образом, в рамках данной работы стояла задача создания и оформления файлов помощи для пакета `pyrssa`, основываясь на уже существующей документации для `Rssa`.

Помимо непосредственно адаптации также стояла задача максимально упростить для пользователей взаимодействие с документацией – предоставить различные способы обращения к ней, а также создать удобную структуру и возможность навигации

по файлам помощи. Наиболее универсальным способом на текущий момент являются веб-страницы, в связи с тем, что они адаптированы для различных устройств, будь то стационарные компьютеры или мобильные устройства, а также для различных операционных систем. В связи с этим главной задачей являлось написание документации в формате веб-сайта и его публикация в интернете для общего доступа.

Глава 2

Документация для pyrssa

Для оформления документации был выбран пакет sphinx [5], обладающий обширной функциональностью и, что самое главное, достаточно высоким уровнем автоматизации работы: определение структуры модуля, к которому пишется документация, создание удобных заготовок веб-страниц с навигацией, простое, но многофункциональное форматирование текста документации, связывание ссылок на определенные в модуле классы – лишь часть доступных возможностей. Во многом благодаря этому данный пакет стал стандартом для оформления документации библиотек, написанных на языке Python. Далее будут описаны предпринятые для создания файлов помощи действия.

2.1. Создание заготовок для веб-страниц

В первую очередь, после установки самого пакета, требуется выполнить набор простых команд в терминале для создания заготовки файла конфигурации, который содержит основную информацию о библиотеке, такую как название и автор, а также позволяет выбрать используемые расширения, позволяющие ускорить работу с документацией, указать директории, в которых будет производиться поиск при создании веб-страниц, и исключить те, которые не должны приниматься в рассмотрение. Ниже приведено примерное содержание данного файла за исключением комментариев, которые создаются для навигации в случае возникновения вопросов при оформлении.

Листинг 2.1. Содержимое конфигурационного файла.

```
from sphinx.ext import autodoc
import os
import sys

sys.path.insert(0, os.path.abspath('..'))

project = 'pyrssa'
copyright = '2023, Fleyderer'
author = 'Fleyderer'
```

```

release = '1.0.6'

extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.viewcode',
    'sphinx.ext.napoleon'
]

templates_path = ['_templates']
exclude_patterns = ['_build', 'Thumbs.db', '.DS_Store']

html_theme = 'sphinx_rtd_theme'
html_static_path = ['_static']
add_module_names = False

class MockedClassDocumenter(autodoc.ClassDocumenter):
    def add_line(self,
                 line: str,
                 source: str, *lineno: int) -> None:
        if line == "    Bases: :py:class:'object'":
            return
        super().add_line(line, source, *lineno)

autodoc.ClassDocumenter = MockedClassDocumenter

```

В указанном выше фрагменте кода импортируются необходимые модули, указывается основная директория для поиска при сборке HTML-файлов, а также информация о пакете, выбираются необходимые расширения, которые проводят автоматический парсинг написанной документации в соответствии с различными стилями форматирования текста. Помимо этого указываются исключаемые файлы, устанавливается визуальный стиль оформления веб-страниц, а также создается дополнительная обертка для того, чтобы исключать из генерируемых файлов ненужную информацию, которая не пона-

добится пользователю. Наполнение этой обертки остается на усмотрение пользователя.

После того, как данный файл успешно оформлен, необходимо сгенерировать файлы с расширением `.rst` – это расширение файлов типа `reStructuredText` [6], являющегося облегчённым языком разметки. С помощью таких файлов оформление навигации становится проще, чем при использовании стандартного HTML. Для этого терминал открывается в основной директории модуля и при выполнении команды для создания файлов выбирается дочерняя директория, внутри которой эти файлы должны находиться. После этого, если при выполнении предыдущих действий не были допущены ошибки, создаются файлы разметки. Во фрагменте кода, приведенном ниже, указано примерное содержание заглавной страницы документации, написанной с помощью `reStructureText`.

Листинг 2.2. Содержимое файла заглавной страницы документации.

```
Welcome to pyrssa 's documentation!
```

```
=====
```

```
Methods and tools for Singular Spectrum Analysis
in Python...
```

```
.. toctree::
: maxdepth: 2
: caption: Contents:
```

```
pyrssa
```

```
Indices and tables
```

```
=====
```

```
* :ref: 'genindex '
* :ref: 'modindex '
* :ref: 'search '
```

После внесения необходимых правок в относящиеся к модулю .rst-файлы, будь то изменения в структуре документации или же наполнении отдельных страниц, с помощью простой команды создаются файлы HTML, которые доступны для просмотра на различных устройствах. Примерный внешний вид заглавной веб-страницы приведен на иллюстрациях ниже.

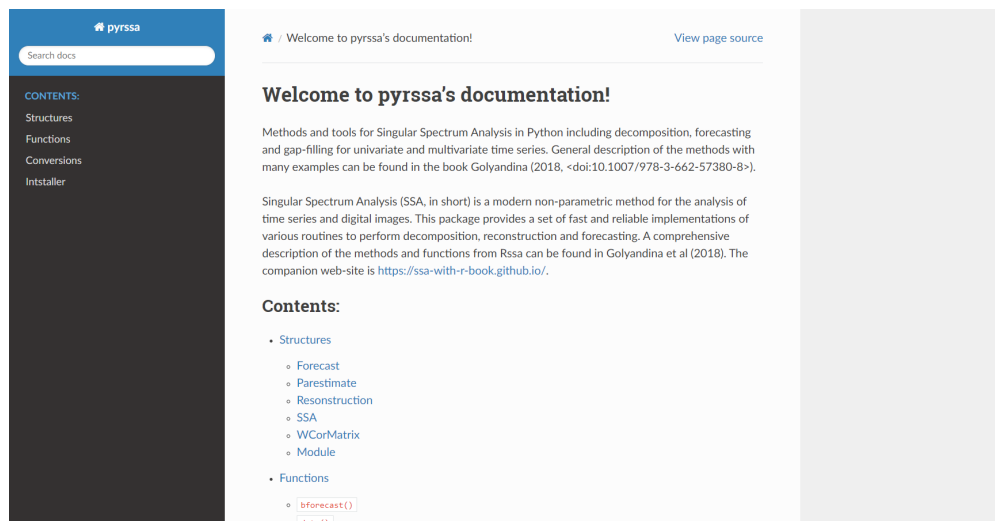


Рис. 2.1. Внешний вид страницы для стационарных устройств.

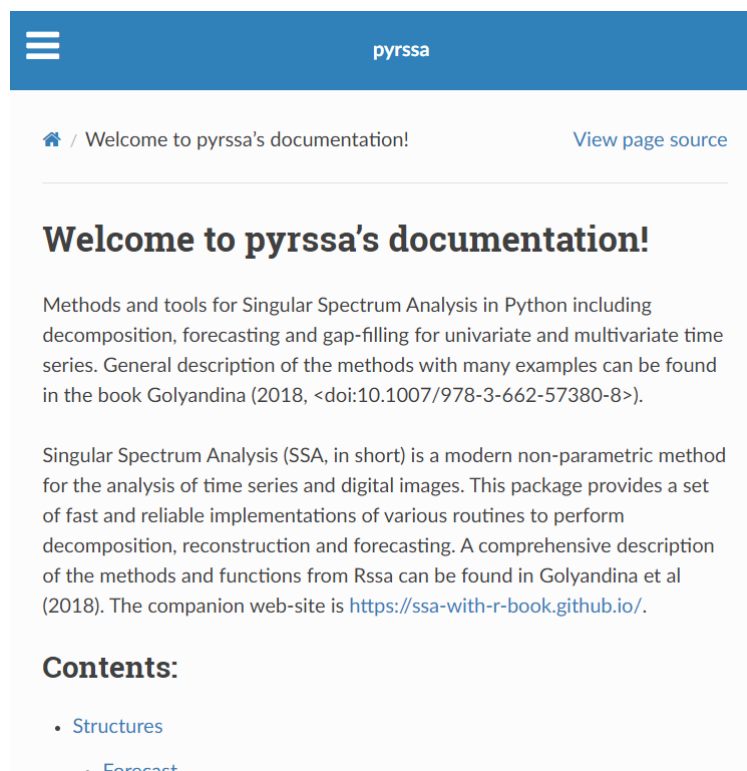


Рис. 2.2. Внешний вид страницы для мобильных устройств.

2.2. Написание документации

После описания структуры документации и создания заглавной страницы необходимо наполнить документацию. Для описания параметров функций, подробной информации, примеров использования и замечаний в Python создаются так называемые docstrings [7] – строковые переменные, которые указываются сразу после объявления объектов в языке – таким образом и происходит описание функций в языке Python. Структура возможной строки будет приведена в листинге ниже.

Листинг 2.3. Сокращенное содержание docstring для функции reconstruct.

```
:param x: SSA object
:type x: SSA
...
```

Description

Reconstruct the data given the SSA decomposition and the desired grouping of the elementary components.

Details

Reconstruction is performed in a common form for different types of input objects

...

Note

By default (argument `drop.attributes`) the routine tries to preserve all the attributes of the input object

...

References

Golyandina N., Korobeynikov A., Zhigljavsky A. (2018):
 Singular Spectrum Analysis with R. Use R!. Springer,
 Berlin, Heidelberg
 ...

Examples

```
.. code-block:: python
...
```

Когда строка информации создана, пользователь может получить информацию о работе функций либо с помощью встроенной в Python функции `help()`, либо, при использовании продвинутых сред программирования, наведя курсор на функцию, получить информацию в отформатированном виде.

При повторной генерации HTML-файлов веб-страниц, строки документации также учитываются и сопоставляются с функциями модуля внутри файлов помощи, подхватывая форматирование на языке `reStructuredText`. Результат для функции `reconstruct` приведен на иллюстрациях ниже.

`reconstruct(x, groups, drop_attributes=False, cache=True)` [\[source\]](#)

- Parameters:**
- **x** (*SSA*) – SSA object
 - **groups** (*list or dict*) – list of numeric vectors, indices of elementary components used for reconstruction, the entries of the list can be named, see 'Value' for more information
 - **drop_attributes** (*bool*) – if *True* then the attributes of the input objects are not copied to the reconstructed ones.
 - **cache** (*bool*) – if *True* then intermediate results will be cached in the SSA object.
- Returns:** List of reconstructed objects. Elements of the list have the same names as elements of groups. If the group is unnamed, then corresponding component will obtain name 'Fn', where 'n' is its index in groups list.
- Return type:** [Reconstruction](#)

Description

Reconstruct the data given the SSA decomposition and the desired grouping of the elementary components.

Details

Reconstruction is performed in a common form for different types of input objects. See Section 1.1.2.6 in Golyandina et al (2018) for the explanation. Formal algorithms are described in this book in Algorithm 2.2 for 1D-SSA, Algorithm 4.3 for MSSA, Algorithm 5.2 for 2D-SSA and Algorithm 5.6 for Shaped 2D-SSA.

Рис. 2.3. Внешний вид документации для параметров и описания функции `reconstruct`.

Note

By default (argument `drop.attributes`) the routine tries to preserve all the attributes of the input object. This way, for example, the reconstruction result of 'ts' object is the 'ts' object with the same time scale.

References

Golyandina N., Korobeynikov A., Zhigljavsky A. (2018): Singular Spectrum Analysis with R. Use R!. Springer, Berlin, Heidelberg.

Korobeynikov, A. (2010): Computation- and space-efficient implementation of SSA. Statistics and Its Interface, Vol. 3, No. 3, Pp. 257-268

Golyandina, N., Korobeynikov, A., Shlemov, A. and Usevich, K. (2015): Multivariate and 2D Extensions of Singular Spectrum Analysis with the Rssa Package. Journal of Statistical Software, Vol. 67, Issue 2. doi:10.18637/jss.v067.i02

Examples

```
import pyrssa as prs
import pandas as pd
import numpy as np

AustralianWine = prs.data("AustralianWine")
fort = AustralianWine["Fortified"][:174]
fort.index = pd.date_range(start="1980/01/01", freq="M", periods=len(fort))
s_fort = prs.ssa(fort, L=84, kind="1d-ssa")
r_fort = prs.reconstruct(s_fort, groups={"Trend": 1, "Seasonality": range(2, 13)})

r_fort_2 = prs.reconstruct(s_fort, groups=[1, range(2, 13)])

co2 = data("co2")
s2 = prs.ssa(co2.value, column_projector="centering", row_projector="centering")
rec = prs.reconstruct(s2, groups={"Linear_trend": range(1, s2.nspecial() + 1)})
```

Рис. 2.4. Внешний вид для замечаний и примеров функции `reconstruct`.

2.3. Публикация сайта

Для публикации сайта был выбран хостинг Github Pages [8], который позволяет опубликовать документацию для общего доступа на основе репозитория, в котором хранится сам пакет `rutssa`. Для решения данной задачи необходимо лишь провести предварительную настройку источника загрузки для файлов, относящихся к сайту, и немного изменить структуру их хранения, чтобы Github мог их распознать. Ключевым фактором является наличие в корневой папке файла `index.html`, который и будет выдавать хостинг при переходе по ссылке. По умолчанию при создании HTML-файлов они находятся во внутренних папках, поэтому сперва проводится корректировка файлов генерации с указанием нужных директорий, а также в корневой папке документации создается небольшой файл `index.html`, который переадресует на действительную заглавную страницу сайта.

Листинг 2.4. Содержимое переадресующего файла `index.html`.

```
<!DOCTYPE html>
<html lang="en">
<meta http-equiv="refresh" content="0;
  url=./html/index.html" />
</html>
```

После проведения указанных действий Github при каждом обновлении репозитория, а если быть точнее, ветки, из которой должна подгружаться документация, автоматически их обновляет и спустя пару минут обновленное наполнение сайта становится доступным пользователям.

Глава 3

Заключение

В рамках данной работы были успешно адаптированы файлы помощи пакета Rssa для основных реализованных в `pyrssa` функций и структур данных, состоящие из их описания, подробной информации и примеров использования в языке Python.

Для достижения этой цели были адаптированы тексты документации Rssa с учетом особенностей реализации на языке Python, к которым относятся типы данных и структуры объектов. Также для большей наглядности были адаптированы и приведены в документации примеры использования функций пакета `pyrssa`.

С учетом задачи повышения удобства взаимодействия обучающихся с документацией, помимо адаптации существующих текстов и примеров выполнена работа по написанию документации в формате веб-сайта, содержащего как саму документацию, так и навигацию по страницам.

Данный веб-сайт опубликован в общем доступе [9], в связи с чем обучающиеся и пользователи могут быстро получить необходимую справку не только внутри используемой среды программирования, но также и при поиске информации в интернете.

Список литературы

- [1] Голяндина Н.Э. *Метод «Гусеница»-SSA: анализ временных рядов: Учеб. пособие.* СПб: Изд-во СПбГУ, 2004.
- [2] *pyrssa package on PyPi.* [Электронный ресурс]. 2023. URL: <https://pypi.org/project/pyrssa/> (дата обр. 05.03.2023).
- [3] Nina Golyandina, Anton Korobeynikov и Anatoly Zhigljavsky. *Singular Spectrum Analysis with R.* Springer, 2018.
- [4] *Rssa: A collection of methods for singular spectrum analysis.* [Электронный ресурс]. 2023. URL: <http://CRAN.R-project.org/package=Rssa> (дата обр. 05.03.2023).
- [5] *Sphinx documentation.* [Электронный ресурс]. 2023. URL: <https://www.sphinx-doc.org/en/master/> (дата обр. 07.03.2023).
- [6] *reStructuredText Markup.* [Электронный ресурс]. 2023. URL: <https://devguide.python.org/documentation/markup/> (дата обр. 06.03.2023).
- [7] *Python Docstring Conventions.* [Электронный ресурс]. 2023. URL: <https://peps.python.org/pep-0257/> (дата обр. 07.03.2023).
- [8] *Github Pages.* [Электронный ресурс]. 2023. URL: <https://pages.github.com/> (дата обр. 07.03.2023).
- [9] *pyrssa package documentation.* [Электронный ресурс]. 2023. URL: <https://fleyderer.github.io/pyrssa/html/index.html> (дата обр. 07.03.2023).