**Скользящая средняя в Pandas (Python)**

[ПРОГРАММИРОВАНИЕ И РАЗРАБОТКА](https://bestprogrammer.ru/programmirovanie-i-razrabotka)

Библиотека Python «Pandas» предоставляет множество встроенных функций? В том числе для реализации методов фильтрации скользящей средней. Это используется для анализа информации о временных рядах и их наглядного отображения получения средних значений отдельных подмножеств всего набора данных. Обычно это используется для исключения кратковременных изменений в данных, оставляя длительные тенденции изменения временного тренда.

**Различные алгоритмы «скользящей средней»**

Pandas реализует три различных метода «скользящего среднего». Эти методы:

* Простая скользящая средняя.
* Экспоненциальная скользящая средняя.
* Кумулятивная скользящая средняя.

**Пример 1**

Для применения метода фильтрации «скользящим средним» нужно создать DataFrame. Для этого сначала генерируем словарь, который называется «Group\_data», и вставляем «Group\_leader», в который вводим значения «Олив, Роуэн, Джулиан, Говард, Нова, Джульетта, Элла и Ава».

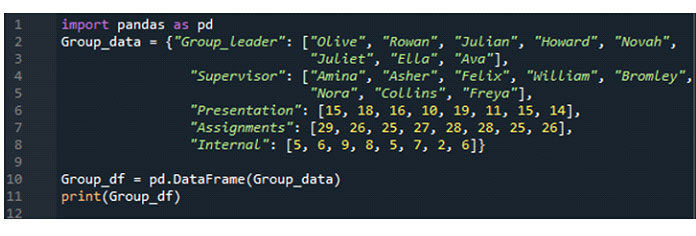
В «Supervisor» добавляют значения «Амина, Ашер, Феликс, Уильям, Бромли, Нора, Коллинз и Фрейя».

«Presentation» содержит отметки: «15, 18, 16, 10, 19, 11, 15 и 14».

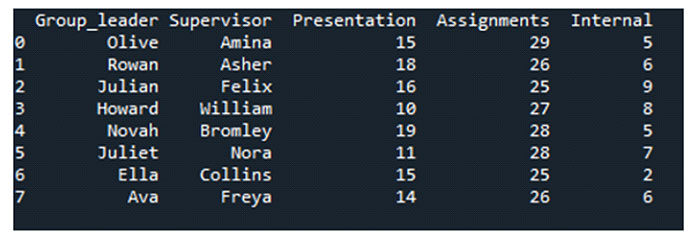
В «Assignments» добавили оценки заданий: «29, 26, 25, 27, 28, 28, 25 и 26».

«Internal»: «5, 6, 9, 8, 5, 7, 2 и 6».

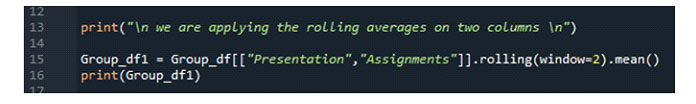
Мы должны преобразовать этот словарь «Group\_data» в DataFrame «Group\_df». Для этого мы использовали метод Pandas «pd.DataFrame()», который использует имя словаря в качестве параметра. После этого мы выводим «Group\_df», используя метод «print()».



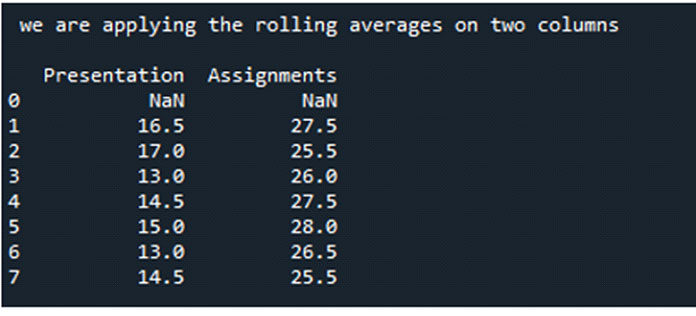
Применим метод скользящего среднего к столбцам этого DataFrame.



Сейчас используется «Простая скользящая средняя» применительно к двум столбцам DataFrame. Сохраним значения в переменной «Group\_df1», которую получаем после применения скользящего среднего по двум столбцам. Скользящее окно равно «2». Сам метод реализуется функцией «mean()». Это скользящее окно используется для получения средних значений по заранее определенному количеству временных интервалов.

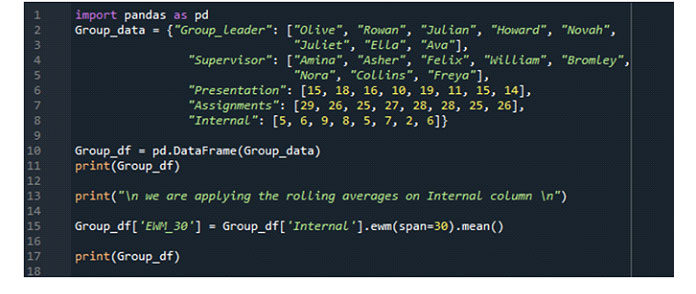


Значения, которые мы получаем, показаны ниже. .

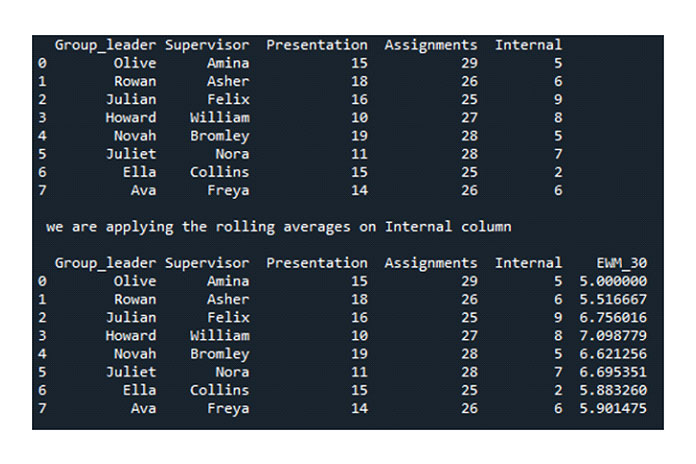


**Пример 2**

К данным «Group\_df» применим «Экспоненциальное скользящее среднее». Для этого используем метод «ewm()».



Ниже показан DataFrame до применения метода «ewm()» и после применения метода «ewm()».



**Пример 3**

Импортируются библиотеки «pandas» и «numpy», так как используются методы обеих библиотек. Созданный DataFrame содержит четыре столбца.

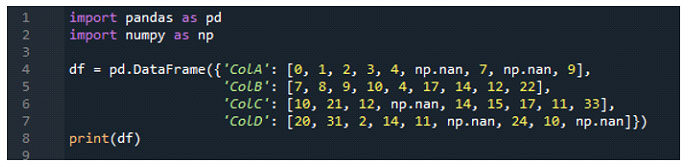
Первый столбец «ColA» содержит «0, 1, 2, 3, 4, np.nan, 7, np.nan и 9».

В «ColB» добавили «7, 8, 9, 10, 4, 17, 14, 12 и 22».

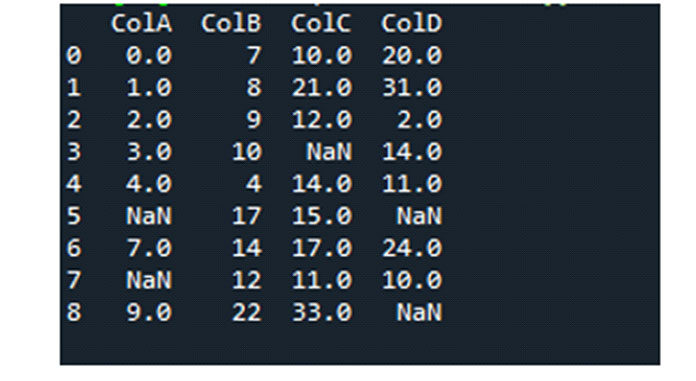
В «ColC» добавили «10, 21, 12, np.nan, 14, 15, 17, 11 и 33».

В «ColD»: «20, 31, 2, 14, 11, np.nan, 24, 10 и np.nan».

Затем мы используем «print (df)», который напечатает этот DataFrame.

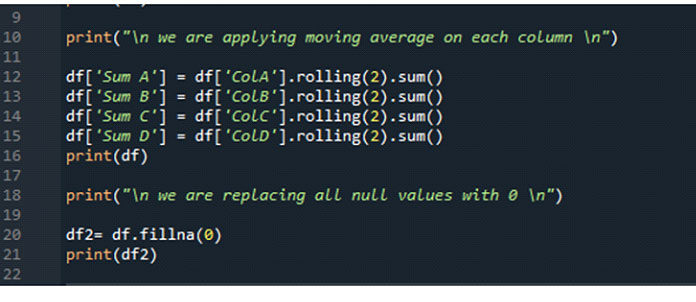


DataFrame перед применением любого метода скользящего среднего показан ниже. Видно, что здесь отображаются все столбцы. Применим скользящее среднее ко всем этим столбцам и отобразим результат.

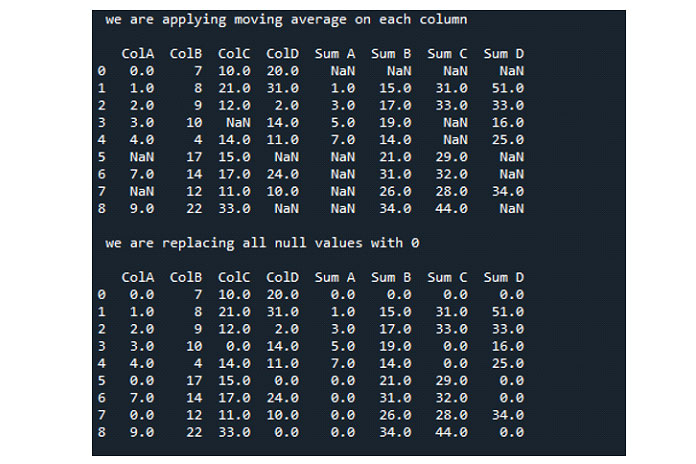


Здесь мы применяем простое скользящее. В этом методе скользящего окна необходимо передать значение скользящего окна (оно равно 2»). Затем применяем метод «sum()» с этим методом скользящего среднего. Сохраним значения, которые мы получили после применения метода «rolling()» к столбцам; значения «ColA» хранятся в «Sum A», значения «ColB» в «Sum B» и значения «ColC» в «Sum C», а также значения «ColD» в переменной «Sum D».

Применили метод «fillna ()» и передали «0» в качестве параметра, который изменит все неопределенные значения, которые мы получаем после применения метода, на «0» и сохраним новый DataFrame в «df2».



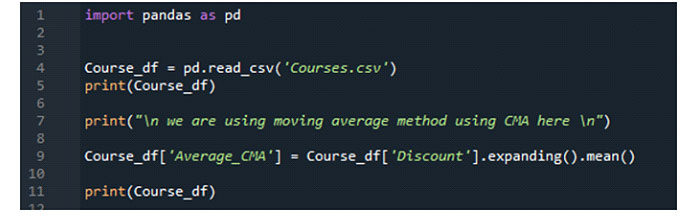
Ниже показан результат выполнения этого кода, в котором отображаются значения, которые получены после использования метода «rolling ()» для всех столбцов и отображения значений скользящего среднего в новом столбце. После этого заменяются все нулевые значения на «0», а также отображает новый кадр данных ниже, который содержит «0» вместо «NaN», которые являются нулевыми значениями.



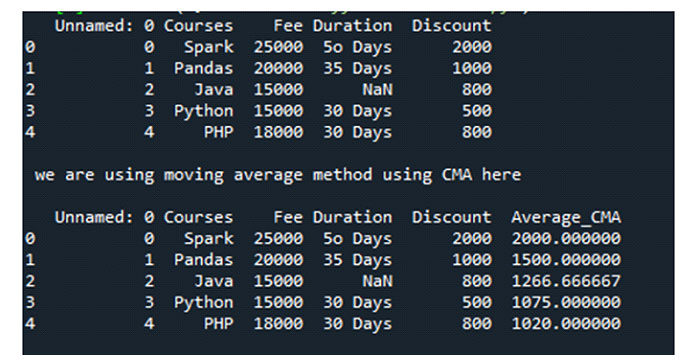
**Пример 4**

Мы также можем рассчитать скользящие средние CSV-файла после чтения данных в файл. Здесь мы читаем файл «Courses.csv», помещая метод «pd.read\_csv()», а затем сохраняя данные как DataFrame в «Courses\_df».

Применим метод «кумулятивной скользящей средней». В этом методе нам не нужно добавлять фиксированный размер окна, потому что размер окна меняется со временем. Чтобы использовать это, мы должны испльзовать метод «expanding()», а после этого мы поместим «mean()». Мы применили этот метод к столбцу «Скидка», который присутствует в CSV-файле, и сохранили результат в столбце «Среднее\_CMA».



DataFrame, который мы получили после чтения файла «Courses.csv», отображается первым. Затем здесь применяется метод кумулятивного скользящего среднего, и результат, который мы получаем после применения этого метода, также отображается ниже в столбце «Average\_CMA».



**Задание на практическую работу**

1. Сгенерировать зашумленные сигналы из практической работы 7. Отобразить на графике
2. Применить к ним фильтрацию методом скользящей средней (использовать все три метода). Отобразить графики.
3. Загрузить данные из файла CSV (по вариантам, см.табл.). Вывести их на экран в виде графика.

|  |  |  |  |
| --- | --- | --- | --- |
| **Вариант** | **Датасет** | **Столбец 1** | **Столбец 2** |
| 1 | oscar\_age\_female.csv | Year | Age |
| 2 | Experiment.csv | Device1 | Device2 |
| 3 | Experiment.csv | device4 | device5 |
| 4 | Experiment.csv | Device2 | Device4 |
| 5 | tips.csv | total\_bil | tip,sex |
| 6 | Experiment.csv | Device3 | Device5 |
| 7 | mlb\_players.csv | Height(inches) | Weight(lbs) |
| 8 | Experiment.csv | Device1 | Device2 |
| 9 | oscar\_age\_male.csv | Year | Age |
| 10 | Experiment.csv | device1 | device2 |

1. Применить в этим данных фильтрацию скользящего среднего (три метода). Вывести на экран в виде графика одновременно с исходными значениями.