Основы работы с Pandas

САМОСТОЯТЕЛЬНАЯ РАБОТА

http://www.kimrt.ru
Онлайн-курс
Суперкомпьютерные технологии в задачах моделирования

Pandas — программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами [1].

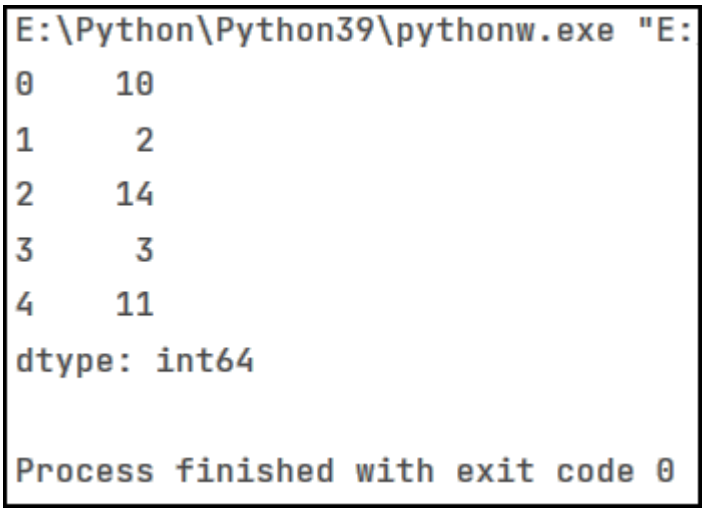
Установка. Для установки pandas выполните pip install pandas [2].

Работа с pandas. Основные структуры для хранения данных *Series* [3] и *DataFrame* [4]. *Series* – это проиндексированный одномерный массив значений. Он похож на простой словарь типа dict, где имя элемента будет соответствовать индексу, а значение – значению записи. *DataFrame* — это проиндексированный многомерный массив значений, соответственно каждый столбец *DataFrame*, является структурой *Series* [5]. Рассмотрим некоторые примеры обработки и запустим код из листинга 1 [6].

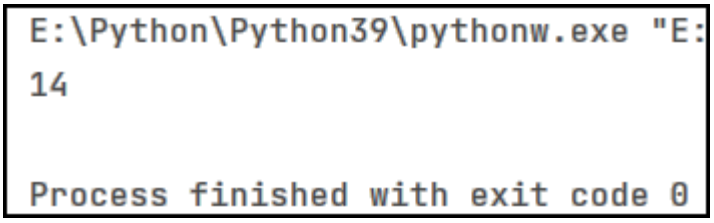
Листинг 1

```
# импорт библиотеки pandas
import pandas
# создание структуры series
series_1 = pandas.Series([10, 2, 14, 3, 11])
# вывод структуры
print(series_1)
```

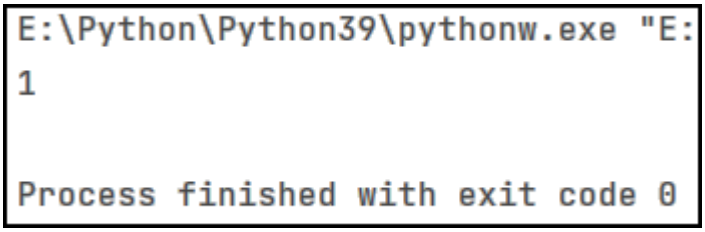
Вывод при запуске кода.



Слева расположены индексы, а справа сами элементы. Также в конце таблицы указан тип хранимых данных. Можно получить элемент по индексу, например, *print(series_1[2])*.



С помощью *argmin* узнаем индекс элемента в структуре с наименьшим значением *print(series_1.argmin())*.



С помощью *between* можно найти элементы, расположенные на определенном промежутке *print(series_1.between(10, 13))*. Вошедшие элементы имеют значение *True* в левом столбце.

```
E:\Python\Python39\pythonw.exe "E:
0     True
1    False
2    False
3    False
4     True
dtype: bool

Process finished with exit code 0
```

Функция имеет параметр включения границ, по умолчанию левая и правая границы входят в промежуток. Добавим параметр, исключающий обе границы *print(series_1.between(10, 13, inclusive="neither"))*.

```
E:\Python\Python39\pythonw.exe "E:
0    False
1    False
2    False
3    False
4     True
dtype: bool

Process finished with exit code 0
```

Рассмотрим код из листинга 2.

Листинг 2

```
# импорт библиотеки
```

```
import pandas
# создание массива с датами
week = pandas.date_range("2022-04-20", periods=7)
# создание структуры DataFrame
df1 = pandas.DataFrame({"day": week,
                        "max temp(C)": ["+24",
"+23", "+19", "+20", "+19", "+19", "+22"],
                        "min temp(C)": ["+13",
"+13", "+14", "+12", "+9", "+11", "+11"],
                        "max wind speed(m/s)":
["15", "14", "8", "9", "9", "7", "5"]
                        }) # вывод
print(df1)
```

При запуске кода получаем следующий вывод.

```
         day max temp(C) min temp(C) max wind speed(m/s)
0 2022-04-20        +24        +13                   15
1 2022-04-21        +23        +13                   14
2 2022-04-22        +19        +14                    8
3 2022-04-23        +20        +12                    9
4 2022-04-24        +19         +9                    9
5 2022-04-25        +19        +11                    7
6 2022-04-26        +22        +11                    5

Process finished with exit code 0
```

Эта таблица содержит прогноз погоды на неделю. В первом столбце указаны даты, во втором – максимальная температура, в третьем – минимальная температура, в четвертом – максимальная скорость ветра. Для создания массива с датами используется функция

date_range, параметры которой начальная дата и количество периодов. Параметр, отвечающий за периодичность по умолчанию, имеет значение одни сутки, поэтому количество периодов – это количество суток. В данном случае *DataFrame* построен из словаря, где ключами являются названия столбцов, а значениями – массивы с данными. С помощью *iloc* получим данные из первой строки *print(df1.iloc[0])*.

```
day                      2022-04-20 00:00:00
max temp(C)                              +24
min temp(C)                              +13
max wind speed(m/s)                       15
Name: 0, dtype: object

Process finished with exit code 0
```

Также можно получить данные за определенную дату с помощью следующей конструкции *print(df1.loc[df1['day'] == '2022-04-24'])* [7].

```
         day max temp(C) min temp(C) max wind speed(m/s)
4 2022-04-24         +19          +9                   9

Process finished with exit code 0
```

Чтение и запись. Записать данные можно с помощью функции *to_csv*. Запишем файл *df1.to_csv('df1.csv')* и посмотрим результат.

df1.csv

```
,day,max temp(C),min temp(C),max wind speed(m/s)
0,2022-04-20,+24,+13,15
1,2022-04-21,+23,+13,14
2,2022-04-22,+19,+14,8
3,2022-04-23,+20,+12,9
4,2022-04-24,+19,+9,9
5,2022-04-25,+19,+11,7
6,2022-04-26,+22,+11,5
```

Теперь считаем этот файл *df2 = pandas.read_csv('df1.csv', sep=',')* и выведем результат *print(df2)*.

```
   Unnamed: 0         day  max temp(C)  min temp(C)  max wind speed(m/s)
0           0  2022-04-20           24           13                   15
1           1  2022-04-21           23           13                   14
2           2  2022-04-22           19           14                    8
3           3  2022-04-23           20           12                    9
4           4  2022-04-24           19            9                    9
5           5  2022-04-25           19           11                    7
6           6  2022-04-26           22           11                    5

Process finished with exit code 0
```

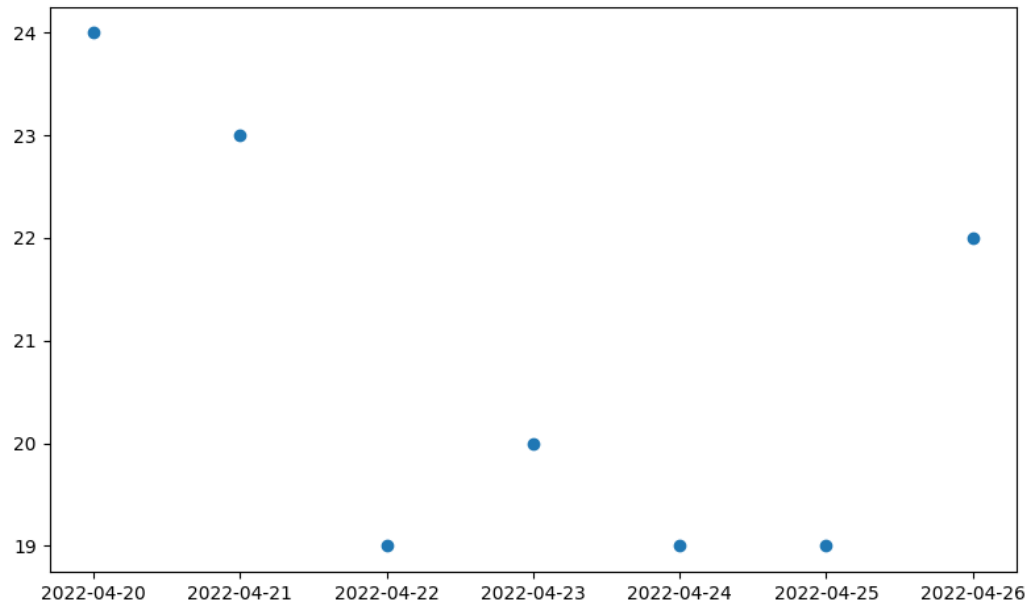
Визуализация. Рассмотрим листинг 3 [8].

Листинг 3

```
# импорт библиотек
import pandas
import matplotlib.pyplot as plt
# создание массива с датами
week = pandas.date_range("2022-04-20", periods=7)
# создание структуры DataFrame
df1 = pandas.DataFrame({"day": week,
```

```
                        "max temp(C)": ["+24",
"+23", "+19", "+20", "+19", "+19", "+22"],
                        "min temp(C)": ["+13",
"+13", "+14", "+12", "+9", "+11", "+11"],
                        "max wind speed(m/s)":
["15", "14", "8", "9", "9", "7", "5"]
                        })
# Визуализация
temp = df1["max temp(C)"].astype(int)
days = df1["day"]
plt.plot(days, temp, ls='none', marker='o')
plt.show()
```

При запуске кода получим следующий график.



На этом графике отображена максимальная температура для каждой даты. В *days* хранятся данные для оси *x*, а в *temp* – для *y*. Значения получаем из *DataFrame*, но для температуры производит

преобразования в тип данных *int* с помощью *astype*. Далее с помощью *matplotlib* строим график.

http://www.kimrt.ru/index/course_stm/0-24

Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Источники

1. Pandas. URL: https://ru.wikipedia.org/wiki/Pandas
2. Pandas. Getting started. URL: https://pandas.pydata.org/docs/getting_started/index.html
3. Series. URL: https://pandas.pydata.org/docs/reference/api/pandas.Series.html#pandas.Series
4. DataFrame. URL: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html#pandas.DataFrame
5. Введение в анализ данных с помощью Pandas. URL: https://habr.com/ru/post/196980/
6. 10 minutes to pandas. URL: https://pandas.pydata.org/docs/user_guide/10min.html#min
7. Select rows from DataFrame. URL: https://datatofish.com/select-rows-pandas-dataframe/
8. Введение в Pandas. URL: https://khashtamov.com/ru/pandas-introduction/