

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт цифрового развития
Кафедра инфокоммуникаций

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
по дисциплине «Анализ данных»**

«Введение в Pandas. Создание объекта Series в Pandas»

Вариант 18

Выполнил:

Харченко Богдан Романович
студент 3 курса, группы ИВТ-б-о-21-1
направление подготовки Информатика и
вычислительная техника, очная форма
обучения

(подпись)

Проверил:

Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г

Цель: исследование назначения и способов установки пакета Pandas, исследование способов создания основного объекта Series пакета Pandas.

Порядок выполнения:

1. Создал общедоступный репозиторий на github.
2. Выполнил клонирование репозитория.
3. Дополнил файл gitignore.
4. Организовал репозиторий в соответствии с моделью ветвления git-flow.

```
C:\Users\super\OneDrive\Рабочий стол\DI\ВУЗь\Воронкин х3\AD_2>git branch develop
C:\Users\super\OneDrive\Рабочий стол\DI\ВУЗь\Воронкин х3\AD_2>git checkout develop
Switched to branch 'develop'

C:\Users\super\OneDrive\Рабочий стол\DI\ВУЗь\Воронкин х3\AD_2>git flow init

Which branch should be used for bringing forth production releases?
- develop
- main
Branch name for production releases: [main]

Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] fe
Bugfix branches? [bugfix/] bu
Release branches? [release/] re
Hotfix branches? [hotfix/] ho
Support branches? [support/] su
Version tag prefix? [] ve
Hooks and filters directory? [C:/Users/super/OneDrive/Рабочий стол/DI/ВУЗь/Воронкин х3/AD_2/.git/hooks] hoo
```

Рисунок 1. Организация репозитория в соответствии с git flow

5. Решил следующие задачи:

Задание 1. Создать объект Series из списка чисел [1, 2, 3] с метками [a, b, c] и вывести его на печать.

Задание 1. Создать объект Series из списка чисел [1, 2, 3] с метками [a, b, c] и вывести его на печать.

```
] s = pd.Series([1,2,3], index=['a', 'b', 'c'])

:] print(s)

a    1
b    2
c    3
dtype: int64
```

Рисунок 2. Выполнение задания 1

Задание 2. Создать объект Series из словаря {'a': 1, 'b': 2, 'c': 3} и вывести его на печать.

Задание 2. Создать объект Series из словаря {'a': 1, 'b': 2, 'c': 3} и вывести его на печать.

```
my_list = {'a': 1, 'b': 2, 'c': 3}
s = pd.Series(my_list)
```

```
print(s)
```

```
a    1
b    2
c    3
dtype: int64
```

Рисунок 3. Выполнение задания 2

Задание 3. Создать объект массива NumPy arr = [1, 2, 3, 4, 5] , затем из него создать объект Series и вывести его на печать.

Задание 3. Создать объект массива NumPy arr = [1, 2, 3, 4, 5] , затем из него создать объект Series и вывести его на печать.

```
my_array = np.array([1, 2, 3, 4, 5])
s = pd.Series(my_array)
```

```
print(s)
```

```
0    1
1    2
2    3
3    4
4    5
dtype: int32
```

Рисунок 4. Выполнение задания 3

Задание 4. Создать объект Series с заполнением по умолчанию значением 0 и метками [a, b, c] и вывести его на печать.

Задание 4. Создать объект Series с заполнением по умолчанию значением 0 и метками [a, b, c] и вывести его на печать.

```
s = pd.Series(0, index=['a', 'b', 'c'])
```

```
print(s)
```

```
a    0
b    0
c    0
dtype: int64
```

Рисунок 5. Выполнение задания 4

Задание 5. В первой строке ввода с клавиатуры поступает строка с символами разлепленными пробелом, во второй строка с целыми числами. На их основе создать список кортежей и из него объект Series, вывести его на печать.

Задание 5. В первой строке ввода с клавиатуры поступает строка с символами разлепленными пробелом, во второй строка с целыми числами. На их основе создать список кортежей и из него объект Series, вывести его на печать.

```
str1 = input("Введите символы, разделенные пробелом ")
str2 = input("Введите целые числа ")
s1 = tuple(str(item) for item in str1.split(' '))
s2 = tuple(int(item) for item in str2.split(' '))
my_list = [s1, s2]
s = pd.Series(my_list)
```

Введите символы, разделенные пробелом a s d f
Введите целые числа 1 2 3 4

Рисунок 6. Выполнение задания 5

Задание 6. На вход поступает строка слов, разделенных пробелом. Создать на ее основе объект Series с метками 0, 1, 2,... Вывести его на печать.

Задание 6. На вход поступает строка слов, разделенных пробелом. Создать на ее основе объект Series с метками 0, 1, 2,... Вывести его на печать.

```
str1 = input("Введите слова, разделенные пробелом ")
s1 = tuple(str(item) for item in str1.split(' '))
s = pd.Series(s1)
```

Введите слова, разделенные пробелом aa ss dd ff

```
print(s)
```

```
0    aa
1    ss
2    dd
3    ff
dtype: object
```

Рисунок 7. Выполнение задания 6

Задание 7. На вход поступает последовательность из чисел 0 и 1, разделенных пробелом. Создать на ее основе объект Series, содержащий булевы значения True (1) , False (0) с метками [a, b, c, d,...]

Задание 7. На вход поступает последовательность из чисел 0 и 1, разделенных пробелом. Создать на ее основе объект Series, содержащий булевы значения True (1) , False (0) с метками [a, b, c, d,...]

```
s = input()
s = s.replace("1", "True")
s = s.replace("0", "False")
st = s.split(' ')

st = pd.Series(st)
st = st.rename(lambda x: chr(97 + x))
```

```
1 0 1 1 0
```

```
print(st)
```

```
a    True
b    False
c     True
d     True
e    False
dtype: object
```

Рисунок 8. Выполнение задания 7

Задание 8. Создать объект Series из списка дат ['2022-01-01', '2022-01-02', '2022-01-03'] , вывести его на печать. Даты в объекте Series должны быть

представлены объектами `datetime` стандартной библиотеки языка программирования Python.

```
s = pd.Series(  
    data = pd.date_range('2022-01-01', periods=3))  
  
print(s)  
  
0    2022-01-01  
1    2022-01-02  
2    2022-01-03  
dtype: datetime64[ns]
```

Рисунок 9. Выполнение задания 8

Задание 9. На вход подаются два целых числа n и m через пробел. На их основе создать объект `Series` с числами от n до m (включительно) с шагом 10 и метками $[1, 3, 5, 7, \dots]$. Вывести его на печать.

```
a = int(input("Начальное число: "))  
b = int(input("Конечное число: "))  
s = []  
  
sq = 0  
for i in range(10):  
    s.append(a + sq)  
    sq += 10  
    if s[i] > b:  
        break  
  
st = pd.Series(s)  
st = st.rename(lambda x: int(1 + 2*x))  
  
Начальное число: 3  
Конечное число: 33  
  
print(st)  
  
1      3  
3     13  
5     23  
7     33  
9     43  
dtype: int64
```

Рисунок 10. Выполнение задания 9

6. Самостоятельно придумал и решил задачу, где использовался бы объект Series библиотеки Pandas.

У нас есть список студентов с их оценками, и мы хотим рассчитать средний балл и определить, сколько студентов получили балл выше среднего. Для этой задачи мы можем использовать объект Series из библиотеки Pandas.

```
import pandas as pd

# Создаем словарь с данными о студентах и их оценках
student_data = {
    'Имя': ['Анна', 'Петр', 'Мария', 'Иван', 'Екатерина'],
    'Оценка': [85, 92, 78, 95, 88]
}

# Создаем объект Series на основе словаря
student_series = pd.Series(student_data['Оценка'], index=student_data['Имя'])

# Рассчитываем средний балл
average_score = student_series.mean()

# Определяем, сколько студентов получили балл выше среднего
above_average = student_series[student_series > average_score]

# Выводим результаты
print("Средний балл студентов:", average_score)
print("Количество студентов с баллом выше среднего:", len(above_average))
print("Студенты с баллом выше среднего:\n", above_average)
```

Средний балл студентов: 87.6
Количество студентов с баллом выше среднего: 3
Студенты с баллом выше среднего:
Петр 92
Иван 95
Екатерина 88

Рисунок 11. Индивидуальное задание

Контрольные вопросы:

1. Какое назначение библиотеки Pandas?

Pandas - это библиотека на языке Python для работы с данными, которая предоставляет широкий набор инструментов для анализа и манипуляции табличных данных. Pandas использует основной объект - DataFrame - для хранения и работы с данными, а также предоставляет многочисленные функции для их обработки.

2. Каким образом можно загрузить данные в Pandas DataFrame?

3. Какие типы данных могут содержать столбцы в Pandas DataFrame?

Каждый столбец может иметь разные типы данных, например, числа, строки или даты.

4. Какие недостатки могут быть у библиотеки Pandas?

- Потребление памяти. Pandas может быть не очень эффективной в использовании памяти для больших наборов данных. Это может привести к тому, что операции на больших данных могут занимать много времени и памяти.

- Сложность. Pandas имеет множество функций и методов, что делает ее мощной, но возможно, сложной в использовании, особенно для новичков в анализе данных.

- Производительность. Несмотря на то, что Pandas является достаточно быстрой библиотекой, некоторые операции, особенно на больших наборах данных, могут быть довольно медленными.

- Ограниченная поддержка параллелизма. Pandas не имеет полной поддержки параллельных вычислений, что может замедлить операции на больших наборах данных.

- Поддержка типов данных. Pandas может быть не очень гибкой в работе с некоторыми типами данных, например, с данными, которые имеют смешанные типы или с данными, которые требуют высокой точности.

5. Какая команда используется для установки библиотеки Pandas в Python?

```
pip install pandas.
```

6. Как проверить, установлена ли библиотека Pandas на компьютере? Чтобы убедиться, что Pandas установлен правильно, можно запустить интерпретатор Python и импортировать Pandas:

```
import pandas as pd
```

7. Какие операционные системы поддерживаются библиотекой Pandas?

Mac OS Linux, Windows

8. Как создать объект Series из списка значений?

```
# создание объекта Series из списка
my_list = [1, 2, 3, 4, 5]
s = pd.Series(my_list)
print(s)
#0    1
#1    2
#2    3
#3    4
#4    5
#dtype: int64
```

9. Как создать объект Series с явным заданием индексов?


```
import pandas as pd

# создание объекта Series из скалярного значения
s = pd.Series(5, index=['a', 'b', 'c'])

print(s)
```

10. Как создать объект Series из словаря?

```
# создание объекта Series из словаря Python
my_dict = {'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}
s = pd.Series(my_dict)
print(s)
#a    1
#b    2
#c    3
#d    4
#e    5
#dtype: int64
```

11. Как создать объект Series с заполнением по умолчанию?

```
s = pd.Series(0, index=['a', 'b', 'c'])
```

12. Как создать объект Series из массива NumPy?

```
my_array = np.array([1, 2, 3, 4, 5])
s = pd.Series(my_array)
```

13. Как создать объект Series из списка кортежей?

```
str1 = input("Введите символы, разделенные пробелом ")
str2 = input("Введите челье числа ")
s1 = tuple(str(item) for item in str1.split(' '))
s2 = tuple(int(item) for item in str2.split(' '))
my_list = [s1, s2]
s = pd.Series(my_list)
```

Вывод: в результате выполнения работы были исследованы назначения и способов установки пакета Pandas, исследованы способы создания основного объекта Series пакета Pandas.