

Wizualizacja danych

Dr Anna Muranova
Semestr letni 2024, UWM w Olsztynie

9. Zajęcie (19.04.2024)

Upewnij się, że w projekcie masz podpięta biblioteki Pandas oraz Numpy.

Ściągnij: https://github.com/pandas-dev/pandas/blob/master/doc/cheatsheet/Pandas_Cheat_Sheet.pdf

<https://www.amazeddeveloper.pl/blog/pandas-basics-1>

Ćwiczenie 1 (program91.py). Stwórz program i wykonaj w nim poniższe czynności:

- (a)
- stwórz jednowymiarową serię danych przechowującą liczby całkowite
 - stwórz jednowymiarową serię danych przechowującą stringi
 - stwórz listę a następnie przekształć ją na serię
 - przekształć jedną z serii stworzonych wcześniej na listę
 - stwórz tablicę jednowymiarową (z biblioteki Numpy) i przekształć ją na serię
 - przekształć jedną z serii stworzonych wcześniej na tablicę (z biblioteki Numpy)
 - wykonaj dodawanie, odejmowanie, mnożenie i dzielenie na dwóch dowolnych seriach danych z indeksami (nazwami). Jak to działa?
 - stwórz serię danych przechowującą 10 liczb losowych z przedziału $[-10, 10]$ z krokiem 0.1 (jak to zrobić?), a następnie stwórz serię zawierającą liczby ujemne z wcześniej stworzonej serii w tym podpunkcie.

<https://pandas.pydata.org/docs/reference/api/pandas.Series.sample.html>

- (b) Przekształć listę, słownik, tablicę Numpy oraz serię danych na ramkę danych.

```
my_list = [1, 32, -37, 91, 12, 11, -5]
my_dict = {'a' : [1, 3, 2], 'b' : [2, 5, 7], 'c' : [3, 4, 8], 'd' : [4, 10, 12]}
my_array = np.array([[1, 2, 5], [-2, 3, 3], [5, 2, 6]])
my_series = pd.Series ([1, 32, -37, 91, 12, 11, -5],
                       index = ['a', 'b', 'c', 'd', 'e', 'f', 'g'])
```

dodaj nazwy wierszy i kolumny w ramkach danych (jeżeli jeszcze nie ma).

Przekształć odwrotnie (w taki sam list, słownik, tablicę, serię!).

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_dict.html

<https://www.geeksforgeeks.org/how-to-convert-pandas-dataframe-into-a-list/>

- (c) Stwórz ramką danych i poćwicz na nią operacje związane z wyciągnięciem elementów, sortowaniem w kolumnie, zmianą kształtu.

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	
<i>l1</i>	1	2	4	5
<i>l2</i>	-3	8	0.5	10
<i>l3</i>	2	-5	7	3

Ćwiczenie 2 (Praca z danymi program92.py). Dla podanych tablic stwórz ramki danych (numery jako indeksy, nazwy Name, Age, ... jako nazwy kolumn)

ID	Name	Age
2312	Anna	21
2336	Zofia	40
2942	Sylwia	13
9840	Katarzyna	31
2794	Teresa	34
2933	Zenon	28

ID	Name	W	H	Glasses
2942	Sylwia	64	151	F
9840	Katarzyna	69	177	T
2794	Teresa	74	170	F
8891	Tomasz	61	157	T
8111	Cezary	66	151	F
2933	Zenon	61	153	T

```
df1 = pd.DataFrame([[2942,9840,2794,8891,8111,2933,8301,9125],
                    ['Sylwia', 'Katarzyna', 'Teresa', 'Tomasz',
                     'Cezary', 'Zenon', 'Filip', 'Adrian'],
                    [13, 31, 34, 14, 13, 28, 20, 15]]).T
df1.columns = ['ID', 'Name', 'Age']
weight = [65, 80, 64, 69, 74, 61, 66, 61]
height = [179, 179, 151, 177, 170, 157, 151, 153]
glasses = [False, True, False, True, False, True, False, True]
df2 = pd.DataFrame([[2312,2336,2942,9840,2794,8891,8111,2933],
                    ['Anna', 'Zofia', 'Sylwia', 'Katarzyna', 'Teresa',
                     'Tomasz', 'Cezary', 'Zenon'],weight,height,glasses]).T
df2.columns = ['ID', 'Name', 'W', 'H', 'G1']
```

- Połącz tabele metodę `inner` (zrób nowa tabele `df0`), porównaj z metodą `outer`.
- posortuj imiona alfabetycznie
- stwórz tablice przechowująca imiona osób noszących okulary (kolejność w tej tablicy musi odpowiednio zachować kolejność z wyjściowej tablicy)
- stwórz tablice zawierająca imiona osób w wieku z przedziału lat [20, 30]
- dodaj kolumną z *bmi* dla wszystkich osób i wynik zapisz w tablicy ($bmi = \frac{waga}{wzrost^2}$).
- policz średni wiek i wyświetl na konsoli.
- policz osobna średni *bmi* osób noszących i nienoszących okulary i wyświetl na konsoli.
- policz osobna średni wiek osób noszących i nienoszących okulary i wyświetl na konsoli.

<https://stackoverflow.com/questions/51153010/slicing-with-a-logical-boolean-express>

Ćwiczenie 3 (Praca z plikami `program93.py`). Za pomocą eksplorera plików i notatnika stwórz plik tekstowy `tekst1.txt` w tym samym folderze co projekt, wpisz do pliku dowolny tekst. Następnie za pomocą instrukcji `open` i różnych parametrów sprawdź różne tryby otwarcie plików. Przykładowy kod:

```
f = open('tekst1.txt', 'r+')
s= f.read()
print(s)
print(type(s))
```

Następnie ustaw break pointa na trzeciej linijce w powyższym przykładzie i spróbuj zmodyfikować i zapisać zmieniona zawartosc pliku tekstowego przez notatnik, WordPad, inny aplikacje.

Napraw sytuacje wywołując metodę `close`.

Ew. zmień kod na taki z użyciem `with`.

Spróbuj dodać polskie znaki do pliku tekstowego i dodaj parametr `encoding` do funkcji `open` (poćwicz różne wersje kodowania).

Na koniec spróbuj zapisać coś nowego do pliku. Sprawdź w jakim trybie otwarcia plików jest to możliwe.

Ćwiczenie 4 (Praca domowa). Na kolejne ćwiczenia przygotuj 5 plików w formacie `csv` dostępnych na darmowych repozytoriach danych. Napisz do nich plik `README` i umieść informacje o źródle danych w postaci:

Nazwa pliku: `sample1.csv`

Opis: [krótki opis czego dotyczy dane zjawisko]

Zródło: [adres strony www, nazwa serwisu]

Licencja: [rodzaj licencji, o ile dotyczy]