



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

Wstęp do programowania

Laboratorium nr 6

Ewelina Cupak, Nr albumu – 73132

Grupa laboratoryjna – L6

Informatyka, rok I, sem. I

Prowadzący:

Mgr Inż. Przemysław Skubel

Zadanie 1

Stwórz tablicę 5x5 (wartości od 0 do 24) oraz wyciągnij z niej obramowanie (pierwszy/ostatni wiersz i kolumna).

```
import numpy as np
a = np.arange(0,25).reshape(5,5)
print(a)

print(a[0,:])
print(a[1:,-1])
print(a[-1,:-1][::-1])
print(a[1:-1,0][::-1])
```

Wynik programu:

```
C:\Users\eweli\PycharmProjects\PythonProject4\.venv\Scripts\python.exe C:\Users\eweli\PycharmProjects\PythonProject4\Zad1.py
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]
 [15 16 17 18 19]
 [20 21 22 23 24]]
[0 1 2 3 4]
[ 9 14 19 24]
[23 22 21 20]
[15 10  5]

Process finished with exit code 0
```

Zadanie 2

Dla macierzy losowej 6x4 ($N(0,1)$) znajdź: średnią każdej kolumny, indeksy min/max w każdej kolumnie.

```
import numpy as np

u = np.random.default_rng(42)

mac = u.normal(0,1,(6,4))
print(mac)

print("-----")

print(np.mean(mac, axis=0))

print("-----")

min_ind = np.argmin(mac, axis=0)
print(min_ind)

print("-----")

max_ind = np.argmax(mac, axis=0)
print(max_ind)

print("-----")
```

Wynik programu:

```
C:\Users\eweli\PycharmProjects\PythonProject4\.venv\Scripts\python.exe C:\Users\eweli\PycharmProjects\PythonProject4\Zad2.py
[[ 0.30471708 -1.03998411  0.7504512   0.94056472]
 [-1.95103519 -1.30217951  0.1278404  -0.31624259]
 [-0.01680116 -0.85304393  0.87939797  0.77779194]
 [ 0.0660307   1.12724121  0.46750934 -0.85929246]
 [ 0.36875078 -0.9588826   0.8784503  -0.04992591]
 [-0.18486236 -0.68092954  1.22254134 -0.15452948]]
-----
[-0.23553336 -0.61796308  0.72103176  0.05639437]
-----
[1 1 1 3]
-----
[4 3 5 0]
-----
```

Zadanie 3

Utwórz tablice $a = \text{np.array}([1, 2, 3, 4])$ $b = \text{np.array}([10, 20, 30, 40])$. Pomnóż je element po elemencie oraz wyznacz iloczyn skalarny (dot product). (Wskazówka: $a*b$, $a @ b$ lub $\text{np.dot}(a,b)$)

```
import numpy as np

a = np.array([1, 2, 3, 4])

b = np.array([10, 20, 30, 40])

print(a)
print(b)

print("-----")

print(a*b)

print("-----")

print(np.dot(a,b))
```

Wynik programu:

```
C:\Users\eweli\PycharmProjects\PythonProject4\.venv\Scripts\python.exe C:\Users\eweli\PycharmProjects\PythonProject4\Zad3.py
[1 2 3 4]
[10 20 30 40]
-----
[ 10  40  90 160]
-----
300
Process finished with exit code 0
```

Zadanie 4

Utwórz dwie tablice rozmiaru $A = (2,3)$ oraz $B = (3,2)$. Oblicz iloczyn skalarny tych tablic ($A@B$ oraz $B@A$). Następnie zmień rozmiar tablicy B na $(3,3)$. Czy mnożenia się udały? Jeżeli nie, to dlaczego?

```
import numpy as np

A = np.arange(1, 7).reshape(2, 3)
B = np.arange(7, 13).reshape(3, 2)
```

```

print(A)
print(B)

print("-----")

print(A@B)
print("-----")
print(B@A)

print("-----")

BB = np.arange(1,10).reshape(3,3)
print(BB)

print("-----")

print(A@BB)
print("-----")
print(BB@A)
print("-----")

```

Wynik programu:

```

C:\Users\eweli\PycharmProjects\PythonProject4\.venv\Scripts\python.exe C:\Users\eweli\PycharmProjects\PythonProject4\Zad4.py
[[1 2 3]
 [4 5 6]]
[[ 7  8]
 [ 9 10]
 [11 12]]
-----
[[ 58  64]
 [139 154]]
-----
[[ 39  54  69]
 [ 49  68  87]
 [ 59  82 105]]
-----

```

```

-----
[[1 2 3]
 [4 5 6]
 [7 8 9]]
-----
[[30 36 42]
 [66 81 96]]
-----
Traceback (most recent call last):
  File "C:\Users\eweli\PycharmProjects\PythonProject4\Zad4.py", line 24, in <module>
    print(BB@A)
ValueError: matmul: Input operand 1 has a mismatch in its core dimension 0, with gufunc signature (n?,k),(k,m?)->(n?,m?) (size 2 is different from 3)

Process finished with exit code 1

```

Odpowiedź na pytanie:

Mnożenia się nie udały, ponieważ nie jest spełniony warunek zgodności wymiarów macierzy. Liczba kolumn macierzy B (3) nie jest równa liczbie wierszy macierzy A (2), co uniemożliwia wykonanie iloczynu skalarnego.

Zadanie 1

Wczytaj plik CSV. Dodaj kolumnę marża = (sprzedaż - koszt) / sprzedaż. Posortuj po rok i po kraj rosnąco.

```
import pandas as pd

df = pd.read_csv("dane.csv", sep = ";")
print(df)

df["Marża"] = (df["Sprzedaż"] - df["Koszt"]) / df["Sprzedaż"]

print(df)

df = df.sort_values(by = ["Rok", "Kraj"],
                    ascending = [True, True])

print(df)
```

Wynik programu:

```
C:\Users\eweli\PycharmProjects\PythonProject4\.venv\Scripts\python.exe C:\Users\eweli\PycharmProjects\PythonProject4\Zad1P.py
```

	Kraj	Rok	Sprzedaż	Koszt
0	PL	2023	100	60
1	PL	2024	120	70
2	DE	2023	90	55
3	DE	2024	95	60
4	US	2024	200	110

	Kraj	Rok	Sprzedaż	Koszt	Marża
0	PL	2023	100	60	0.400000
1	PL	2024	120	70	0.416667
2	DE	2023	90	55	0.388889
3	DE	2024	95	60	0.368421
4	US	2024	200	110	0.450000

	Kraj	Rok	Sprzedaż	Koszt	Marża
2	DE	2023	90	55	0.388889
0	PL	2023	100	60	0.400000
3	DE	2024	95	60	0.368421
1	PL	2024	120	70	0.416667
4	US	2024	200	110	0.450000

Zadanie 2

Zwróć rekordy dla DE w latach ≥ 2023 i tylko kolumny sprzedaż i marża.

```
import pandas as pd

df = pd.read_csv("dane.csv", sep = ";")
print(df)

df["Marża"] = (df["Sprzedaż"] - df["Koszt"]) / df["Sprzedaż"]
print(df)

kol = df.loc[(df["Kraj"] == "DE") & (df["Rok"] >= 2023), ["Sprzedaż",
"Marża"]]

print(kol)
```

Wynik programu:

```
C:\Users\eweli\PycharmProjects\PythonProject4\.venv\Scripts\python.exe C:\Users\eweli\PycharmProjects\PythonProject4\Zad2P.py
Kraj Rok Sprzedaż Koszt
0 PL 2023 100 60
1 PL 2024 120 70
2 DE 2023 90 55
3 DE 2024 95 60
4 US 2024 200 110
Kraj Rok Sprzedaż Koszt Marża
0 PL 2023 100 60 0.400000
1 PL 2024 120 70 0.416667
2 DE 2023 90 55 0.388889
3 DE 2024 95 60 0.368421
4 US 2024 200 110 0.450000
Sprzedaż Marża
2 90 0.388889
3 95 0.368421
Process finished with exit code 0
```

Zadanie 3

Wykorzystaj poniższy kod:

- Stwórz nową kolumnę Marża (wzór wcześniej w instrukcji). Co gdyby sprzedaż pewnego produktu w danym roku była równa 0?
- Zwróć wiersze, gdzie Kraj = PL i Produkt = A
- Policz sumę Sprzedaż wg Kraj (malejąco)
- Zwróć Top20 dni o najwyższej marży w każdym z kraju (w jednej tabeli!) (Wskazówka: `groupby('Kraj').apply(lambda g: g.nlargest(3, 'Marża'))` lub `rank`)

```
import pandas as pd
import numpy as np

rng = np.random.default_rng(42)
dates = pd.date_range("2024-01-01", periods=120, freq="D")
kraj = rng.choice(["PL", "DE", "US"], size=len(dates))
sprzedaz = rng.integers(50, 250, size=len(dates))
koszt = sprzedaz * rng.uniform(0.4, 0.8, size=len(dates))
produkt = rng.choice(["A", "B", "C"], size=len(dates))
df = pd.DataFrame({
    "Data": dates,
    "Kraj": kraj,
    "Produkt": produkt,
    "Sprzedaż": sprzedaz,
    "Koszt": np.round(koszt, 2),
    "Waluta": "PLN"
})
df.head()
print(df)

#a)
df["Marża"] = (df["Sprzedaż"] - df["Koszt"]) / df["Sprzedaż"]

print(df)

#b)
df_filtr = df.loc[(df["Kraj"] == "PL") & (df["Produkt"] == "A"), : ]
```

```

print(df_filtr)

#c)
sum = df.groupby("Kraj") ["Sprzedaż"].sum()

sum = sum.sort_values(ascending=False)
print(sum)

#d)
Tp = df.groupby("Kraj").apply(lambda g:g.nlargest(20, "Marża"),
include_groups=False)

print(Tp)

```

Wynik programu:

```

C:\Users\ewel1\PycharmProjects\PythonProject4\.venv\Scripts\python.exe C:\Users\ewel1\PycharmProjects\PythonProject4\Zad3P.py

```

	Data	Kraj	Produkt	Sprzedaż	Koszt	Waluta
0	2024-01-01	PL	A	137	86.81	PLN
1	2024-01-02	US	C	183	120.77	PLN
2	2024-01-03	DE	A	180	78.08	PLN
3	2024-01-04	DE	C	144	81.55	PLN
4	2024-01-05	DE	B	221	92.08	PLN
..
115	2024-04-25	DE	C	193	139.30	PLN
116	2024-04-26	PL	B	182	106.83	PLN
117	2024-04-27	PL	C	136	97.09	PLN
118	2024-04-28	PL	B	111	45.19	PLN
119	2024-04-29	PL	B	175	77.64	PLN

[120 rows x 6 columns]

	Data	Kraj	Produkt	Sprzedaż	Koszt	Waluta	Marża
0	2024-01-01	PL	A	137	86.81	PLN	0.366350
1	2024-01-02	US	C	183	120.77	PLN	0.340055
2	2024-01-03	DE	A	180	78.08	PLN	0.566222
3	2024-01-04	DE	C	144	81.55	PLN	0.433681
4	2024-01-05	DE	B	221	92.08	PLN	0.583348
..
115	2024-04-25	DE	C	193	139.30	PLN	0.278238
116	2024-04-26	PL	B	182	106.83	PLN	0.413022
117	2024-04-27	PL	C	136	97.09	PLN	0.286103
118	2024-04-28	PL	B	111	45.19	PLN	0.592883
119	2024-04-29	PL	B	175	77.64	PLN	0.556343

```

[120 rows x 7 columns]

```

	Data	Kraj	Produkt	Sprzedaż	Koszt	Waluta	Marża
0	2024-01-01	PL	A	137	86.81	PLN	0.366350
35	2024-02-05	PL	A	182	121.74	PLN	0.331099
44	2024-02-14	PL	A	133	82.72	PLN	0.378045
51	2024-02-21	PL	A	68	41.21	PLN	0.393971
55	2024-02-25	PL	A	142	59.35	PLN	0.582042
57	2024-02-27	PL	A	82	65.35	PLN	0.203049
65	2024-03-06	PL	A	139	92.39	PLN	0.335324
73	2024-03-14	PL	A	122	89.36	PLN	0.267541
86	2024-03-27	PL	A	202	91.45	PLN	0.547277
87	2024-03-28	PL	A	243	163.07	PLN	0.328930
99	2024-04-09	PL	A	230	128.58	PLN	0.440957
103	2024-04-13	PL	A	90	70.40	PLN	0.217778

Kraj

US	7183
DE	6024
PL	4744

Name: Sprzedaż, dtype: int64

	Data	Produkt	Sprzedaż	Koszt	Waluta	Marża	
DE	84	2024-03-25	C	141	57.19	PLN	0.594397
	29	2024-01-30	A	61	24.93	PLN	0.591311
	43	2024-02-13	C	131	54.36	PLN	0.585038
	4	2024-01-05	B	221	92.08	PLN	0.583348
	19	2024-01-20	B	137	59.33	PLN	0.566934
	2	2024-01-03	A	180	78.08	PLN	0.566222

2	2024-01-03	A	180	78.08	PLN	0.566222
67	2024-03-08	B	126	55.16	PLN	0.562222
72	2024-03-13	C	171	76.80	PLN	0.550877
54	2024-02-24	B	189	88.04	PLN	0.534180
10	2024-01-11	C	163	76.32	PLN	0.531779
75	2024-03-16	B	67	31.60	PLN	0.528358
78	2024-03-19	C	117	55.99	PLN	0.521453
52	2024-02-22	A	204	102.49	PLN	0.497598
69	2024-03-10	C	110	55.55	PLN	0.495000
25	2024-01-26	B	220	111.49	PLN	0.493227
26	2024-01-27	B	56	29.83	PLN	0.467321
3	2024-01-04	C	144	81.55	PLN	0.433681
56	2024-02-26	B	193	110.79	PLN	0.425959
28	2024-01-29	A	214	123.17	PLN	0.424439
20	2024-01-21	A	246	146.29	PLN	0.405325
118	2024-04-28	B	111	45.19	PLN	0.592883
55	2024-02-25	A	142	59.35	PLN	0.582042
8	2024-01-09	C	164	72.38	PLN	0.558659
40	2024-02-10	B	249	110.33	PLN	0.556908
119	2024-04-29	B	175	77.64	PLN	0.556343
96	2024-04-06	C	65	29.41	PLN	0.547538
86	2024-03-27	A	202	91.45	PLN	0.547277
32	2024-02-02	C	233	106.27	PLN	0.543906
110	2024-04-20	C	142	64.77	PLN	0.543873
71	2024-03-12	B	176	87.36	PLN	0.503636
6	2024-01-07	B	65	34.58	PLN	0.468000
99	2024-04-09	A	230	128.58	PLN	0.440957
116	2024-04-26	B	182	106.83	PLN	0.413022
17	2024-01-18	B	56	33.20	PLN	0.407143

17	2024-01-18	B	56	33.20	PLN	0.407143
101	2024-04-11	C	141	83.96	PLN	0.404539
82	2024-03-23	B	149	89.44	PLN	0.399732
51	2024-02-21	A	68	41.21	PLN	0.393971
44	2024-02-14	A	133	82.72	PLN	0.378045
0	2024-01-01	A	137	86.81	PLN	0.366350
9	2024-01-10	B	176	111.77	PLN	0.364943
US 15	2024-01-16	B	110	45.00	PLN	0.590909
106	2024-04-16	C	211	86.50	PLN	0.590047
109	2024-04-19	C	85	37.60	PLN	0.557647
34	2024-02-04	C	136	60.31	PLN	0.556544
88	2024-03-29	B	102	45.77	PLN	0.551275
97	2024-04-07	B	69	31.02	PLN	0.550435
83	2024-03-24	A	189	86.48	PLN	0.542434
7	2024-01-08	B	202	92.48	PLN	0.542178
74	2024-03-15	C	241	111.18	PLN	0.538672
92	2024-04-02	A	207	99.34	PLN	0.520097
85	2024-03-26	A	103	50.66	PLN	0.508155
42	2024-02-12	A	131	64.46	PLN	0.507939
36	2024-02-06	C	75	38.44	PLN	0.487467
104	2024-04-14	B	194	99.83	PLN	0.485412
50	2024-02-20	B	71	36.66	PLN	0.483662
100	2024-04-10	A	75	39.03	PLN	0.479600
79	2024-03-20	C	242	126.84	PLN	0.475868
63	2024-03-04	A	189	99.48	PLN	0.473651
48	2024-02-18	C	116	61.12	PLN	0.473103
13	2024-01-14	B	161	86.74	PLN	0.461242

Odpowiedź na pytanie:

Jeżeli wartość Sprzedaż wyniosłaby 0, wówczas wystąpiłoby dzielenie przez zero, co skutkowałoby: wartością inf lub NaN, potencjalnym błędem logicznym w analizie danych.

Aby temu zapobiec, należałoby: zastąpić 0 wartością NaN, albo użyć warunku, np. `np.where(Sprzedaż == 0, 0, ...)`.

W obecnym zbiorze danych problem nie występuje, ponieważ sprzedaż jest losowana z przedziału 50–250.

Wnioski ogólne: Podczas laboratorium zapoznałam się z podstawowymi mechanizmami pracy z bibliotekami NumPy i Pandas, które są fundamentem analizy danych w Pythonie. W części poświęconej NumPy nauczyłam się tworzyć tablice wielowymiarowe, operować na ich kształcie, wykonywać indeksowanie, maskowanie oraz obliczenia statystyczne. W części dotyczącej Pandas przećwiczyłam pracę na obiektach typu DataFrame, w tym wczytywanie danych, filtrowanie, sortowanie oraz tworzenie nowych kolumn obliczeniowych. Ważnym elementem było także czyszczenie danych, obsługa braków i błędnych typów. Zastosowanie grupowania (`groupby`) i agregacji pozwoliło na analizę danych na wyższym poziomie, a operacje takie jak ranking czy wybór rekordów o najwyższych wartościach pokazały praktyczne możliwości Pandas w analizie biznesowej.