



Лабораторна робота №1

Роботу виконував: Іванов Віктор Віталійович, ФФ-03

Завдання по Polars DataFrame

Завдання 1 Операції з числовими даними

Завантажте таблицю із числовими даними, проведіть нормалізацію та стандартизацію колонки із числовими даними, додайте нові колонки до таблиці, виведіть статистику по колонкам розширеної таблиці

First.py

```
import polars as pl
import logging

# Налаштування логування
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

# Функція для нормалізації колонки 'population' на діапазон [0, 1]
def normalize_population(data):
    min_population = data['population'].min()
    max_population = data['population'].max()
    normalized_data = data.with_columns((data['population'] -
min_population) / (max_population - min_population))
    return normalized_data

# Функція для стандартизації колонки 'population'
def standardize_population(data):
    mean_population = data['population'].mean()
    std_population = data['population'].std()
    standardized_data = data.with_columns((data['population'] -
mean_population) / std_population)
    return standardized_data

# Функція для нормалізації колонки 'population' на [0, 1]
def normalize_population_01(data):
    normalized_data = data.with_columns(data['population'] /
data['population'].max())
    return normalized_data

if __name__ == '__main__':
    # Завдання по Polars DataFrame
    '''
        Завдання 1 Операції з числовими даними
        Завантажте таблицю із числовими даними, проведіть
        нормалізацію та стандартизацію колонки із числовими даними,
```

```
        додайте нові колонки до таблиці, виведіть статистику по
колонкам розширеної таблиці
'''
# Завантаження даних з файлу
data = pl.read_csv('UATown.csv')

# Виведення початкової таблиці
logger.info("Початкова таблиця:")
logger.info(data)

# Нормалізація колонки 'population' на [0, 1]
normalized_data_01 = normalize_population_01(data)
logger.info("Результат нормалізації колонки 'population' на
[0, 1]:")
logger.info(normalized_data_01)

normalized_data = normalize_population(data)
logger.info("Результат нормалізації колонки 'population':")
logger.info(normalized_data)

# Стандартизація колонки 'population'
standardized_data = standardize_population(data)
logger.info("Результат стандартизації колонки 'population':")
logger.info(standardized_data)

# Додавання нових колонок до таблиці
data = data.with_columns(
    data['lat'] * data['lng'],
    data['population'] + data['population_proper']
)

# Логування розширеної таблиці
logger.info("Розширена таблиця:")
logger.info(data)

# Логування статистики по колонкам розширеної таблиці
logger.info("Статистика по розширеній таблиці:")
logger.info(data.describe())
```

Результат:

```
INFO:__main__:Початкова таблиця:
```

```
INFO:__main__:shape: (216, 9)
```

city	lat	lng	country	...	admin_name	capital	population	population_
---	---	---	---		---	---	---	proper
str	f64	f64	str		str	str	f64	---
								f64
Kyiv	50.45	30.5233	Ukraine	...	Kyiv, Misto	primary	2.963199e6	2.96218e6
Kharkiv	49.9925	36.2311	Ukraine	...	Kharkivska	admin	1.446107e6	1.433886e6
					Oblast			
Odesa	46.4775	30.7326	Ukraine	...	Odeska	admin	1.017699e6	1.015826e6
					Oblast			
Dnipro	48.4675	35.04	Ukraine	...	Dnipropetro	admin	980948.0	980948.0
					vska Oblast			
...
Krasnokamia	44.5625	34.2931	Ukraine	...	Krym,	null	1074.0	1074.0
nka					Avtonomna			
					Respublika			
Bairachky	48.3883	38.5992	Ukraine	...	Luhanska	null	1065.0	1065.0
					Oblast			
Marianivka	48.3118	33.5294	Ukraine	...	Dnipropetro	null	1016.0	1016.0
					vska Oblast			
Volodymyr-V	50.8525	24.3236	Ukraine	...	Volynska	minor	null	null
olynskyi					Oblast			

INFO:__main__:Результат нормалізації колонки 'population' на [0, 1]:

INFO:__main__:shape: (216, 9)

city	lat	lng	country	...	admin_name	capital	population	population_
---	---	---	---		---	---	---	proper
str	f64	f64	str		str	str	f64	---
								f64
Kyiv	50.45	30.5233	Ukraine	...	Kyiv, Misto	primary	1.0	2.96218e6
Kharkiv	49.9925	36.2311	Ukraine	...	Kharkivska	admin	0.488022	1.433886e6
					Oblast			
Odesa	46.4775	30.7326	Ukraine	...	Odeska	admin	0.343446	1.015826e6
					Oblast			
Dnipro	48.4675	35.04	Ukraine	...	Dnipropetro	admin	0.331044	980948.0
					vska Oblast			
...
Krasnokamia	44.5625	34.2931	Ukraine	...	Krym,	null	0.000362	1074.0
nka					Avtonomna			
					Respublika			
Bairachky	48.3883	38.5992	Ukraine	...	Luhanska	null	0.000359	1065.0
					Oblast			
Marianivka	48.3118	33.5294	Ukraine	...	Dnipropetro	null	0.000343	1016.0
					vska Oblast			
Volodymyr-V	50.8525	24.3236	Ukraine	...	Volynska	minor	null	null
olynskyi					Oblast			

```
INFO:__main__:Результат нормалізації колонки 'population':  
INFO:__main__:shape: (216, 9)
```

city	lat	lng	country	...	admin_name	capital	population	population_
---	---	---	---		---	---	---	proper
str	f64	f64	str		str	str	f64	---
								f64
Kyiv	50.45	30.5233	Ukraine	...	Kyiv, Misto	primary	1.0	2.96218e6
Kharkiv	49.9925	36.2311	Ukraine	...	Kharkivska	admin	0.487847	1.433886e6
					Oblast			
Odesa	46.4775	30.7326	Ukraine	...	Odeska	admin	0.343221	1.015826e6
					Oblast			
Dnipro	48.4675	35.04	Ukraine	...	Dnipropetro	admin	0.330814	980948.0
					vska Oblast			
...
Krasnokamia	44.5625	34.2931	Ukraine	...	Krym,	null	0.00002	1074.0
nka					Avtonomna			
					Respublika			
Bairachky	48.3883	38.5992	Ukraine	...	Luhanska	null	0.000017	1065.0
					Oblast			
Marianivka	48.3118	33.5294	Ukraine	...	Dnipropetro	null	0.0	1016.0
					vska Oblast			
Volodymyr-V	50.8525	24.3236	Ukraine	...	Volynska	minor	null	null
olynskyyi					Oblast			

```
INFO:__main__:Результат стандартизації колонки 'population':  
INFO:__main__:shape: (216, 9)
```

city	lat	lng	country	...	admin_name	capital	population	population_ proper
---	---	---	---		---	---	---	---
str	f64	f64	str		str	str	f64	f64
Kyiv	50.45	30.5233	Ukraine	...	Kyiv, Misto	primary	10.569011	2.96218e6
Kharkiv	49.9925	36.2311	Ukraine	...	Kharkivska	admin	4.958262	1.433886e6
					Oblast			
Odesa	46.4775	30.7326	Ukraine	...	Odeska	admin	3.373856	1.015826e6
					Oblast			
Dnipro	48.4675	35.04	Ukraine	...	Dnipropetro	admin	3.237938	980948.0
					vska Oblast			
...
Krasnokamia	44.5625	34.2931	Ukraine	...	Krym,	null	-0.385987	1074.0
nka					Avtonomna			
					Respublika			
Bairachky	48.3883	38.5992	Ukraine	...	Luhanska	null	-0.38602	1065.0
					Oblast			
Marianivka	48.3118	33.5294	Ukraine	...	Dnipropetro	null	-0.386202	1016.0
					vska Oblast			
Volodymyr-V	50.8525	24.3236	Ukraine	...	Volynska	minor	null	null
olynskyyi					Oblast			

```
INFO:__main__:Розширена таблиця:
INFO:__main__:shape: (216, 9)
```

city	lat	lng	country	...	admin_name	capital	population	population
---	---	---	---		---	---	---	_proper
str	f64	f64	str		str	str	f64	---
								f64
Kyiv	1539.90048	30.5233	Ukraine	...	Kyiv,	primary	5.925379e6	2.96218e6
	5				Misto			
Kharkiv	1811.28326	36.2311	Ukraine	...	Kharkivska	admin	2.879993e6	1.433886e6
	7				Oblast			
Odesa	1428.37441	30.7326	Ukraine	...	Odeska	admin	2.033525e6	1.015826e6
	7				Oblast			
Dnipro	1698.3012	35.04	Ukraine	...	Dnipropetr	admin	1.961896e6	980948.0
					ovska			
					Oblast			
...
Krasnokami	1528.18626	34.2931	Ukraine	...	Krym,	null	2148.0	1074.0
anka	9				Avtonomna			
					Respublika			
Bairachky	1867.74966	38.5992	Ukraine	...	Luhanska	null	2130.0	1065.0
	9				Oblast			
Marianivka	1619.86566	33.5294	Ukraine	...	Dnipropetr	null	2032.0	1016.0
	7				ovska			
					Oblast			
Volodymyr-	1236.91586	24.3236	Ukraine	...	Volynska	minor	null	null
Volynskyi	9				Oblast			

```
INFO:__main__:Статистика по розширеній таблиці:  
INFO:__main__:shape: (9, 10)
```

describe	city	lat	lng	...	admin_nam	capital	populatio	populatio
---	---	---	---		e	---	n	n_proper
str	str	f64	f64		---	str	---	---
					str		f64	f64
count	216	216.0	216.0	...	216	216	216.0	216.0
null_count	0	0.0	0.0	...	0	130	1.0	1.0
mean	null	1591.2965	32.819833	...	null	null	209309.39	103868.23
		78					0698	2558
std	null	238.76683	5.077098	...	null	null	539467.00	269113.70
		5					3957	9477
min	Aeroflots	1073.4109	22.1639	...	Cherkaska	admin	2032.0	1016.0
	kyi	75			Oblast			
25%	null	1473.0641	30.1564	...	null	null	9474.0	4737.0
		25						
50%	null	1597.4308	34.0044	...	null	null	54338.0	27088.0
		22						
75%	null	1822.1041	37.5278	...	null	null	158730.0	79272.0
		32						
max	Zorynsk	1920.3024	39.7833	...	Zhytomyrs	primary	5.925379e	2.96218e6
		45			ka Oblast		6	

Завдання 2 із завантаження статистичних даних

Курс гривні

перейти на сторінку <https://bank.gov.ua/ua/markets/exchangerate-chart?cn%5B%5D=EUR&startDate=01.01.1999&endDate=> , завантажити собі на локальний комп'ютер в файл Official hrivnya exchange rates.csv, перенести файл в Google Colab і відкрити звідти.

Дані інфляції

прямо завантажити з: <https://data.gov.ua/dataset/12f4fe34-0759-4271-b1f6-780995f0ec4a/resource/072ca996-993f-4796-a6f5-a7c1b2001b17/download/136-indeksi-spozhyvchikh-tsin.csv>

завантажити з excel

файла: https://www.ukrstat.gov.ua/operativ/open_data/2022/136.xlsx із 4-ого листа, вказати параметр sheet_id = 4

Second.py

```
import pandas as pd
from matplotlib import pyplot as plt
import polars as pl
import mplcyberpunk

plt.style.use("cyberpunk")

def fnc():
    try:
        # Читання CSV файлу
        hrv = pl.read_csv('Official hrivnya exchange rates.csv')
        print(hrv.head())

        # Перетворення типу даних для колонки "Date"
        T_hrv = hrv.with_columns(hrv["Date"].str.strptime(pl.Date,
"%d.%m.%Y", strict=False))
        hrv_rate = hrv["Official hrivnya exchange rates, UAH"] /
hrv["Unit"]

        # Відображення графіку курсу євро
        plt.plot(T_hrv["Date"].to_list(), hrv_rate.to_list(), 'r',
label='euro')
        plt.legend()
        plt.title('Курс Євро')
        mplcyberpunk.add_glow_effects()
        plt.show()

    except FileNotFoundError:
        print("Файл не знайдено. Переконайтеся, що шлях до файлу
правильний.")
    except Exception as e:
        print(f"Виникла помилка: {e}")

def infla_df():
    """
```

```

        Дані інфляції прямо завантажити з 136-indeksi-
        spozhivchikh-tsin.csv
        """
        file_path =
r'C:\Users\prime\PycharmProjects\KPI_tasks\HW_1\136-indeksi-
        spozhivchikh-tsin.csv'
        # Завантажте дані в DataFrame
        df = pd.read_csv(file_path)
        print(df.head())

def func_idx():
    """
        Завантажити з excel файла:
        https://www.ukrstat.gov.ua/operativ/open_data/2022/136.xlsx із 4-
        ого листа,
        вказати параметр sheet_id = 4
        """
        # Шлях до файлу на вашому комп'ютері
        file_path =
r'C:\Users\prime\PycharmProjects\KPI_tasks\HW_1\136.xlsx'

        # Індекс листа, який вас цікавить (4-ий лист)
        sheet_id = 3 # Зауважте, що листи нумеруються з нуля

        # Завантаження даних з XLSX-файлу
        try:
            df = pd.read_excel(file_path, sheet_name=sheet_id)
            # Операції з DataFrame
            # Наприклад, можна вивести перші 5 рядків
            print(df.head())
        except FileNotFoundError:
            print("Файл не знайдено. Переконайтеся, що шлях до файлу
            правильний.")
        except Exception as e:
            print(f"Виникла помилка при завантаженні та обробці даних:
            {e}")

if __name__ == "__main__":
    print("Table is NBU:")
    fnc()
    print("Table is 136 sheet Consumer price indices:")
    func_idx()
    print("Дані інфляції")
    infla_df()

```



Table is NBU:

shape: (5, 7)

Date	Time	Digital code	Letter code	Unit	Currency name	Official hryvnia
---	---	---	---	---	---	exchange rates,...
str	f64	i64	str	i64	str	---
						f64
01.01.1999	0.0	978	EUR	100	null	399.8452
02.01.1999	0.0	978	EUR	100	null	399.8452
03.01.1999	0.0	978	EUR	100	null	399.8452
04.01.1999	0.0	978	EUR	100	null	399.8452
05.01.1999	0.0	978	EUR	100	null	404.009

Table is 136 sheet Consumer price indices:

	code	...		index5
0	code	...	consumer price indices (to December 2010)	
1	00	...		135.0
2	00	...		142.2
3	00	...		157.5
4	00	...		179.6

[5 rows x 8 columns]

Дані інфляції

	_id	...		index5
0	1	...	consumer price indices (to December 2010)	
1	2	...		135.0
2	3	...		142.2
3	4	...		157.5
4	5	...		179.6

[5 rows x 9 columns]

Завдання 3 із завантаження даних

Завантажте дані по вказаному посиланню, відкрийте як csv файл, в параметрах функції вкажіть що немає заголовка в першому рядку через додатковий параметр `has_header = False`, виведіть кілька рядків таблиці.

<https://raw.githubusercontent.com/karlrupp/microprocessor-trend-data/master/50yrs/transistors.dat>

Third.py

```
import polars as pl
# Візуалізувати дані
import matplotlib.pyplot as plt
import numpy as np
import mplcyberpunk

plt.style.use("cyberpunk")

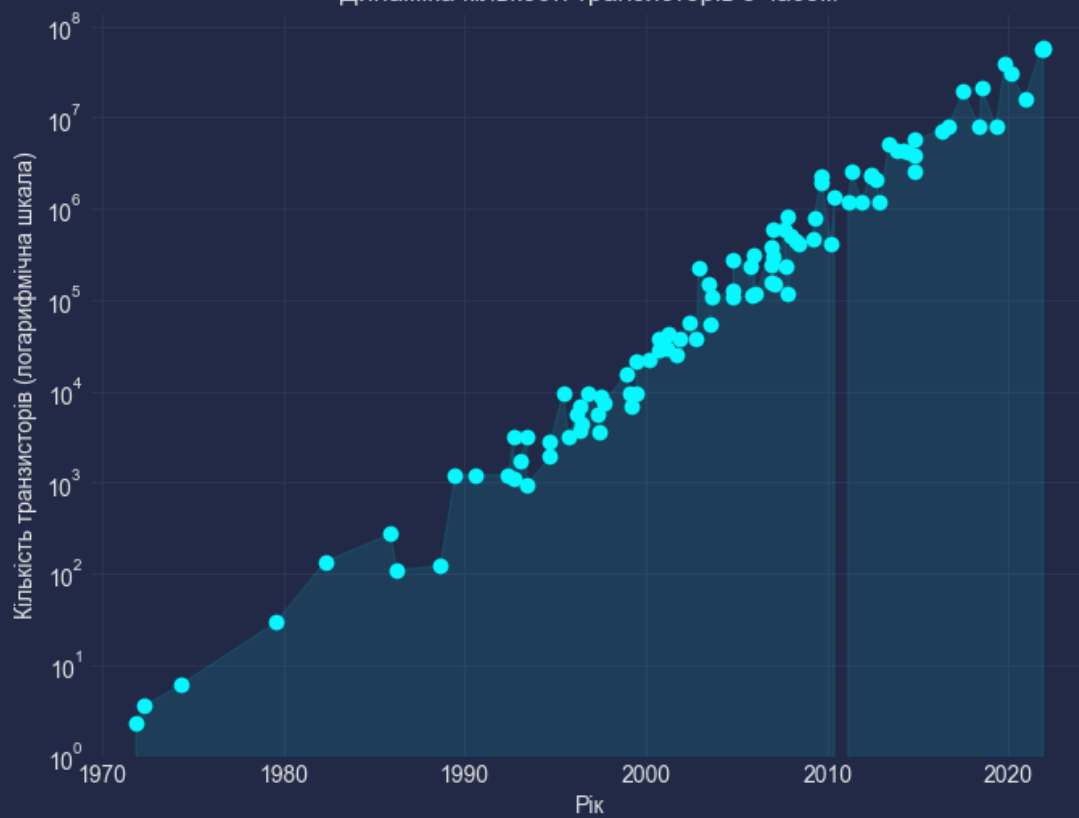
"""
    Завдання 3 із завантаження даних
    Завантажте дані по вказаному посиланню, відкрийте як csv
    файл, в параметрах функції
    вкажіть що немає заголовка в першому рядку через додатковий
    параметр has_header = False, виведіть кілька рядків таблиці.
    https://raw.githubusercontent.com/karlrupp/microprocessor-
    trend-data/master/50yrs/transistors.dat
"""

# Завантажити дані CSV без заголовка
data = pl.read_csv(
    'https://raw.githubusercontent.com/karlrupp/microprocessor-
    trend-data/master/50yrs/transistors.dat',
    has_header=False
)

# Перейменувати стовпці та перетворити їх до потрібних типів
df = data.select(pl.col('column_1').str.splitn(' ',
2).struct.rename_fields(['date', 'value'])).unnest("column_1")
df = df.select(
    pl.col('date').cast(pl.Float64, strict=False),
    pl.col('value')
)

plt.figure(figsize=(8, 6))
plt.plot(df['date'], np.asarray(df['value'].to_list(), float),
'o')
plt.yscale("log")
plt.xlabel('Рік')
plt.ylabel('Кількість транзисторів (логарифмічна шкала)')
plt.title('Динаміка кількості транзисторів з часом')
plt.grid(True)
mplcyberpunk.add_glow_effects()
plt.show()
```

Динаміка кількості транзисторів з часом



Завдання 4 Знайти мінімуми і максимуми на запису потенціалу нейрона

Завантажити файл NeuronRecords.*, узяти окрему колонку із записом, вивести перелік точок в таблицях, а також застосувати код для виводу графіка ISI на нових даних

Fourth.py

```
import logging
import pandas as pd
import matplotlib.pyplot as plt
import mplcyberpunk

plt.style.use("cyberpunk")

"""
    Завдання 4 Знайти мінімуми і максимуми на запису потенціалу
    нейрона
    Завантажити файл NeuronRecords.*, узяти окрему колонку із
    записом, вивести перелік точок в таблицях,
    а також застосувати код для виводу графіка ISI на нових даних
"""

# Ініціалізація системи логування з режимом запису 'w' (перезапис)
logging.basicConfig(filename='neuron_analysis.log',
                    level=logging.INFO, encoding='utf-8', filemode='w') # Додайте
filemode='w'

# Завантаження даних з файлу з вказанням кодування
data = pd.read_csv('NeuronRecords.csv', encoding='utf-8') #
Замініть 'NeuronRecords.csv' на відповідний шлях до вашого файлу

# Перегляд першого стовпця даних (часу)
time = data['time']

# Перегляд і аналіз кожного стовпця з потенціалом нейрона
for col in data.columns[1:]: # Проходимо по всіх стовпцях, крім
першого (часу)
    neuron_potential = data[col]

    # Знаходимо мінімум та максимум для потенціалу нейрона
    min_potential = neuron_potential.min()
    max_potential = neuron_potential.max()

    # Логуємо результати
    logging.info(f"Мінімум для {col}: {min_potential}")
    logging.info(f"Максимум для {col}: {max_potential}")

    # Побудова графіка ISI (міжспайковий інтервал)
    plt.figure(figsize=(10, 5))
    plt.plot(time, neuron_potential)
    plt.xlabel('Час')
    plt.ylabel('Потенціал нейрона')
    plt.title(f'Графік потенціалу нейрона для {col}')
    mplcyberpunk.add_glow_effects()
    plt.show()
```

```
INFO:root:Мінімум для 20: -83.57959
INFO:root:Максимум для 20: 39.64673
INFO:root:Мінімум для 30: -83.22241
INFO:root:Максимум для 30: 40.00391
INFO:root:Мінімум для 40: -82.86523
INFO:root:Максимум для 40: 41.43262
INFO:root:Мінімум для 50: -83.22241
INFO:root:Максимум для 50: 37.86084
INFO:root:Мінімум для 60: -81.43652
INFO:root:Максимум для 60: 40.36108
INFO:root:Мінімум для 70: -80.00781
INFO:root:Максимум для 70: 40.71826
INFO:root:Мінімум для 80: -82.50806
INFO:root:Максимум для 80: 42.50415
INFO:root:Мінімум для 90: -82.50806
INFO:root:Максимум для 90: 42.50415
INFO:root:Мінімум для 100: -81.7937
INFO:root:Максимум для 100: 41.78979
INFO:root:Мінімум для 110: -82.86523
INFO:root:Максимум для 110: 43.57568
```



