Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrarea de laborator nr. 6

# La disciplina „Programarea Declarativa”

Tema: "Manipularea datelor folosind limbajul Python”

A efectuat: st. gr. TI-214 Buza Cătălin

A verificat: M. Rusu

Chișinău – 2023

**Mersul lucrarii:**

**Exerciții:**

1. Scrieți secvența de cod care regrupeză datele din cele 3 fișiere csv (obținute la laboratul

precedent) după o caracteristică de bază (de ex: cursul valutar de la 3 bănci pentru aceeași

zi).

import pandas as pd

print("\nSarcina 1")

micb\_rates\_df = pd.read\_csv(r'C:\\Users\\catal\\Desktop\\PD\\Lab5\\micb\_exchange.csv', index\_col=0)

bcr\_rates\_df = pd.read\_csv(r'C:\\Users\\catal\\Desktop\\PD\\Lab5\\bcr\_exchange.csv', index\_col=0)

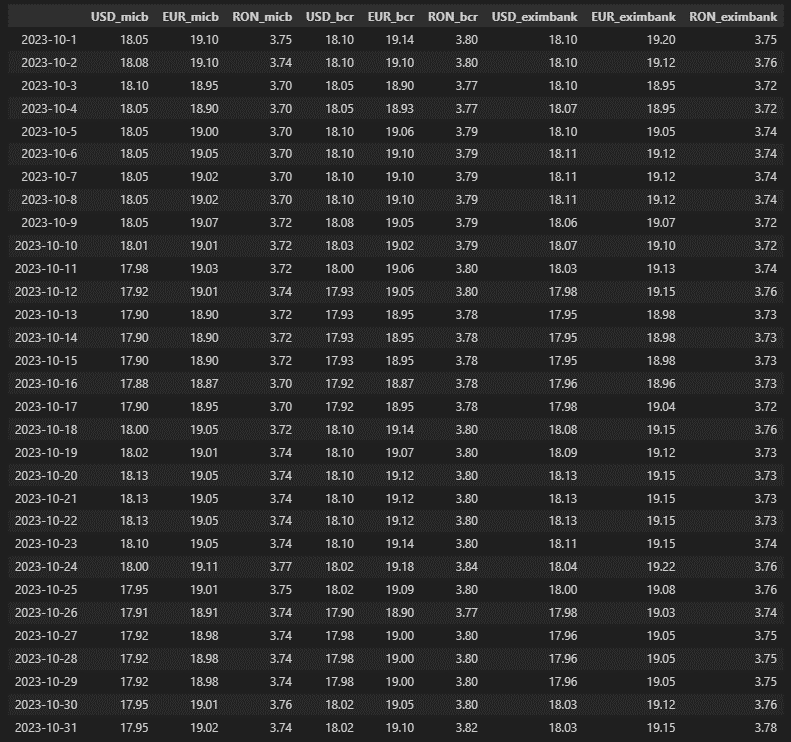
eximbank\_rates\_df = pd.read\_csv(r'C:\\Users\\catal\\Desktop\\PD\\Lab5\\eximbank\_exchange.csv', index\_col=0)

df\_regrupat = pd.merge(micb\_rates\_df, bcr\_rates\_df, left\_index=True, right\_index=True, suffixes=('\_micb','\_bcr'))

df\_regrupat = pd.merge(df\_regrupat, eximbank\_rates\_df.add\_suffix('\_eximbank'), left\_index=True, right\_index=True)

df\_regrupat.to\_csv("Df\_grupat.csv",index=True)

Rezultat :



2. Determinați min, max, media pentru valorile regrupate pentru fiecare zi aparte și min, max

media pentru toate datele stocate.

print("\nSarcina 2")

min\_pe\_zi = df\_regrupat.min(axis=1)

max\_pe\_zi = df\_regrupat.max(axis=1)

media\_pe\_zi = df\_regrupat.mean(axis=1)

# Calculul pentru toate datele stocate

min\_total = df\_regrupat.min().min()

max\_total = df\_regrupat.max().max()

media\_total = df\_regrupat.mean().mean()

# Afisează rezultatele

print("Minim pe zi:")

print(min\_pe\_zi)

print("\nMaxim pe zi:")

print(max\_pe\_zi)

print("\nMedie pe zi:")

print(media\_pe\_zi)

print("\nMinim total:")

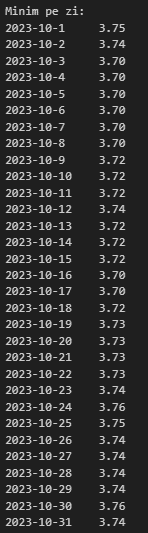
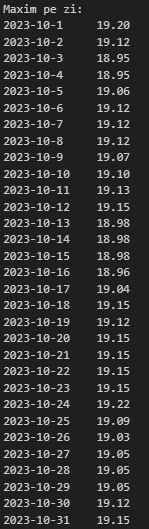
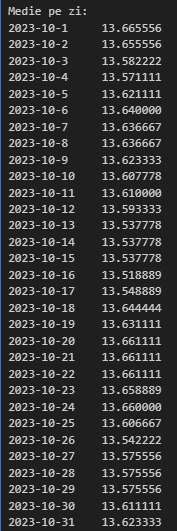
print(min\_total)

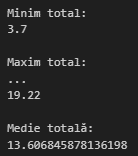
print("\nMaxim total:")

print(max\_total)

print("\nMedie totală:")

print(media\_total)

Rezultat:



3. Scrieți secvența de cod care va împărți fișierul csv obținut după concatinare în N fișiere,

unde fiecare fișier individual va corespunde unei săptămâini (7 rânduri). Fișierele sunt

denumite după prima și ultima dată pe care le conțin (de ex 20231103\_20231109.csv).

df\_regrupat.index = pd.to\_datetime(df\_regrupat.index)

df\_regrupat = df\_regrupat.sort\_index()

lines\_per\_week = 7

for i in range(0, len(df\_regrupat), lines\_per\_week):

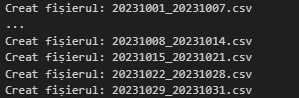
    week = df\_regrupat.iloc[i:i+lines\_per\_week]

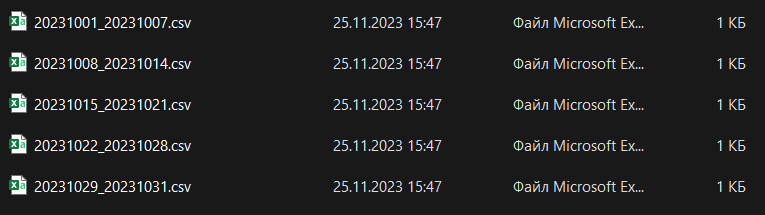
    filename = f"{week.index[0].strftime('%Y%m%d')}\_{week.index[-1].strftime('%Y%m%d')}.csv"

    week.to\_csv('./exchange/' + filename, index=True)

    print(f"Creat fișierul: {filename}")

Rezultat:





4. Scrieți funcția ce are o dată de intrare tip datetime și returnează liniile din DataFrame (din

fișierul cu date concatinate) pentru această dată sau None dacă nu există date pentru această

dată.

def get\_data\_for\_date(data\_frame, target\_date):

    if target\_date in data\_frame.index:

        return data\_frame.loc[target\_date]

    else:

        print(f"Nu există date pentru data: {target\_date}")

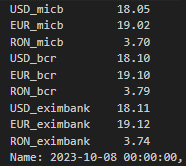
        return None

target\_date = pd.to\_datetime("2023-10-08")

result = get\_data\_for\_date(df\_regrupat, target\_date)

if result is not None:

    print(result)

Rezultat:   


5. Scrieți secvența de cod care permite afișarea datelor pe secvențe de timp. De ex. pentru

fiecare săptămănă separat (funcția are o dată de intrare tip datetime) și returnează liniile

din DataFrame pentru această perioadă sau None dacă nu există date pentru zilele indicate.

def get\_data\_for\_week(data\_frame, target\_date):

    if not isinstance(data\_frame.index, pd.DatetimeIndex):

        data\_frame.index = pd.to\_datetime(data\_frame.index)

    start\_of\_week = target\_date - pd.Timedelta(days=target\_date.dayofweek)

    end\_of\_week = start\_of\_week + pd.Timedelta(days=6)

    if start\_of\_week in data\_frame.index and end\_of\_week in data\_frame.index:

        return data\_frame.loc[start\_of\_week:end\_of\_week]

    else:

        print(f"Nu există date pentru săptămâna care conține data: {target\_date}")

        return None

target\_date = pd.to\_datetime("2023-10-08")

result = get\_data\_for\_week(df\_regrupat, target\_date)

if result is not None:

    print(result)

Rezultat:   


**Concluzie:**

În concluzie, dupa efectuarea acestui laborator am primit o experiență practică în manipularea și analiza datelor utilizând Python și biblioteca pandas. Prin regruparea datelor din mai multe fișiere CSV, calcularea statisticilor pentru valorile regrupate, împărțirea datelor în fișiere separate pe săptămâni și dezvoltarea de funcții pentru manipularea datelor pe baza unor date specifice, am dobândit cunoștințe și abilități esențiale în prelucrarea eficientă a seturilor de date. Aceste competențe sunt cruciale în contextul analizei și interpretării datelor din diverse domenii și sunt utile pentru elaborarea de rapoarte și luarea deciziilor bazate pe informații solide extrase din date.

Top of Form

Bottom of Form