**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Programul de studii: Tehnologia informației**

RAPORT

# LUCRARE DE LABORATOR NR. 6

# la Programarea Declarativă

**Tema: Manipularea datelor folosind limbajul Python**

A efectuat:

st. gr. TI-211 Popa Cătălin

A verificat: lect.dr. Mariana Rusu

UTM, Chișinău 2023

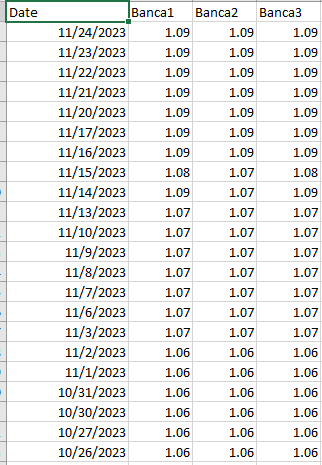
**Tema**

Manipularea datelor folosind limbajul python.

**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  
  
df1 = pd.read\_csv('curs1.csv')  
df2 = pd.read\_csv('curs2.csv')  
df3 = pd.read\_csv('curs3.csv')  
  
merged\_df = pd.merge(df1, df2, on='Date', how='inner')  
merged\_df = pd.merge(merged\_df, df3, on='Date', how='inner')  
merged\_df.columns = ['Date', 'Banca1', 'Banca2' , 'Banca3']  
  
merged\_df.to\_csv('rezultat.csv', index=False)

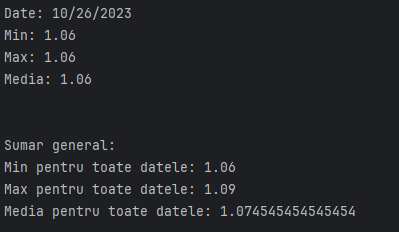


**Figura 1- Afisare csv.**

**2.**

Determinați min, max, media pentru valorile regrupate pentru fiecare zi aparte și min, max media pentru toate datele stocate.

import pandas as pd  
  
df = pd.read\_csv('rezultat.csv',dayfirst=True)  
# Calcularea valorilor pentru fiecare zi  
for date in df['Date'].unique():  
 df\_date = df[df['Date'] == date]  
 print(f"\nDate: {date}")  
 print(f"Min: {df\_date.min(numeric\_only=True).min()}")  
 print(f"Max: {df\_date.max(numeric\_only=True).max()}")  
 print(f"Media: {df\_date.mean(numeric\_only=True).mean()}")  
  
# Calcularea valorilor pentru toate datele  
print("\n\nSumar general:")  
print(f"Min pentru toate datele: {df.min(numeric\_only=True).min()}")  
print(f"Max pentru toate datele: {df.max(numeric\_only=True).max()}")  
print(f"Media pentru toate datele: {df.mean(numeric\_only=True).mean()}")

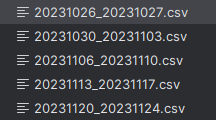


**Figura 2- Afisare min, max si media.**

**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).

import pandas as pd  
  
df = pd.read\_csv('rezultat.csv')  
# Convertirea coloanei 'Date' la tipul de date datetime  
df['Date'] = pd.to\_datetime(df['Date'], format='%m/%d/%Y')  
  
# Calcularea săptămânii pentru fiecare dată  
df['Week'] = (df['Date'] - pd.to\_timedelta(df['Date'].dt.dayofweek, unit='D')).dt.strftime('%Y%m%d')  
grouped = df.groupby('Week')  
  
for name, group in grouped:  
 start\_date = group['Date'].min().strftime('%Y%m%d')  
 end\_date = group['Date'].max().strftime('%Y%m%d')  
 filename = f'{start\_date}\_{end\_date}.csv'  
 group.to\_csv(filename, index=False)  
 print(f'Creat fișierul {filename}')

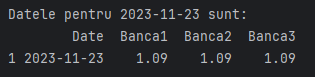


**Figura 3- Repartizarea pe saptamani.**

**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.

import pandas as pd  
def get\_data(file\_path, data\_necesara):  
 date = pd.read\_csv(file\_path)  
 date['Date'] = pd.to\_datetime(date['Date'], format='%m/%d/%Y')  
 result\_date = date[date['Date'] == data\_necesara]  
 if result\_date.empty:  
 return None  
 else:  
 return result\_date  
  
data\_necesara = pd.to\_datetime('2023.11.23')  
fisier = 'rezultat.csv'  
result = get\_data(fisier, data\_necesara)  
if result is not None:  
 print(f'Datele pentru {data\_necesara.date()} sunt:\n{result}')  
else:  
 print(f'Nu există date pentru {data\_necesara.date()} în fișier.')

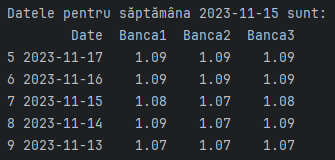


**Figura 4 – Afișare date pentru o zi anumită.**

**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.

import pandas as pd  
def get\_data(file\_path, target\_date):  
 df = pd.read\_csv(file\_path)  
 df['Date'] = pd.to\_datetime(df['Date'], format='%m/%d/%Y')  
  
 start\_date = target\_date - pd.to\_timedelta(target\_date.dayofweek, unit='D')  
 end\_date = start\_date + pd.DateOffset(days=6)  
 result\_date = df[(df['Date'] >= start\_date) & (df['Date'] <= end\_date)]  
  
 if result\_date.empty:  
 return None  
 else:  
 return result\_date  
  
data\_necesara = pd.to\_datetime('2023-11-15')  
fisier = 'rezultat.csv'  
result = get\_data(fisier, data\_necesara)  
if result is not None:  
 print(f'Datele pentru săptămâna {data\_necesara.strftime("%Y-%m-%d")} sunt:\n{result}')  
else:  
 print(f'Nu există date pentru săptămâna {data\_necesara.strftime("%Y-%m-%d")} în fișier.')



**Figura 5 – Afișare date pentru o săptămână anumită.**

**Concluzie**

În cadrul acestui laborator, am efectuat o serie de operațiuni de manipulare și analiză a datelor utilizând biblioteca Pandas în Python. Am concatenat fișierele CSV, am regrupat datele în funcție de caracteristici cheie, am calculat statistici relevante precum minimul, maximul și media, și am implementat funcții utile pentru accesarea și afișarea datelor în funcție de date specifice și intervale de timp. Aceste tehnici sunt esențiale în explorarea și înțelegerea seturilor de date, furnizând instrumente eficiente pentru analiza și interpretarea informațiilor.