**Seminar 2 Python**

|  |
| --- |
| *Citirea fisierelor*  *Prelucrare date in pandas*  *Selectia coloane, inregistrari*  *Selectii conditionate cu metode si operatori*  *Modificarea datelor*  *Adaugare coloane*  *Stergere coloane*  *Redenumirea coloanelor*  *Statistici descriptive*  *Functii de grup*  *Prelucrare fisier json* |

* Prelucrare date din fișiere: *.txt*, *.csv*, *.json*

2.1 Citire fișier Data1.txt

Funcția open() returnează un obiect de tip fișier. Metoda read() citește conținutul fișierului.

Funcția open()are doi parametri: numele fișierului și modul în care poate fi deschis:

"r" - Read - implicit. Deschide fișierul pentru citire, apare eroare dacă fișierul nu există.

"a" - Append - Deschide fișierul pentru adăugare, creează fișierul dacă nu există.

"w" - Write - Deschide fișierul pentru scriere, creează fișierul dacă nu există.

"x" - Create - creează fișierul, apare eroare dacă fișierul există.

* Metoda readline()citește o singură linie
* Poate fi specificat numărul de caractere afișate: read(100)

|  |  |
| --- | --- |
| Exemplu 1 | Exemplu 2 |
| f = open(**"Data1.txt"**, **"r"**) print(f.read())  f = open(**"Data1.txt"**, **"r"**) print(f.read(100))  f = open(**"Data1.txt"**, **"r"**) print(f.readline())  f = open(**"Data1.txt"**, **"r"**) **for** x **in** f:  print(x) | numara\_inregistrari = sum(1 **for** line **in** open(**'Data1.txt'**)) print(numara\_inregistrari)  '''numara\_inregistrari = len(open(**'Data1.txt'**).read().splitlines()) print(numara\_inregistrari) '''  403374 |

Suplimentar, se pot specifica:

"t" - Text - Implicit. Modul text

"b" - Binary - Modul binar (de ex. imagine)

f = open("Data1.txt") și f = open("Data1.txt", "rt") sunt echivalente.

|  |
| --- |
| Exemplu 3 |
| f1 = open(**"Data1.txt"**, **"a"**) f2 = open(**"Data2.txt"**, **"r"**) **for** r **in** f2:  f1.write(r) num\_inreg = sum(1 **for** line **in** open(**'Data1.txt'**)) print(num\_inreg) |

2.2 Citire fișier *.csv*

|  |
| --- |
| Exemplu 4. Citirea unui fișier *.csv*, prin intermediul bibliotecii **csv** |
| **import** csv **with** open(**'clienti\_leasing.csv'**, **'r'**) **as** f:  reader = csv.reader(f)  **for** row **in** reader:  print (row) |

|  |
| --- |
| Exemplu 5. Citirea primei coloane dintr-un fișier *.csv* |
| **import** csv **with** open(**'clienti\_leasing.csv'**, **'r'**) **as** f:  reader = csv.reader(f)  **for** row **in** reader:  print (row[0]) |

|  |
| --- |
| Exemplu 6. Citirea primelor două coloane dintr-un fișier *.csv* |
| **import** csv **with** open(**'clienti\_leasing.csv'**, **'r'**) **as** f:  reader = csv.reader(f)  **for** row **in** reader:  print (row[0],row[1]) # print (row[0:2]) |

|  |
| --- |
| Exemplu 7. Citirea primelor zece înregistrări dintr-un fișier *.csv* |
| **import** csv **with** open(**'clienti\_leasing.csv'**, **'r'**) **as** f:  reader = csv.reader(f)  **for** i, row **in** enumerate(reader):  print(row)  **if**(i >= 10):  **break** |

|  |
| --- |
| Exemplu 8. Citirea primelor zece înregistrări dintr-un fișier *.csv* cu islice |
| **import** csv **from** itertools **import** islice **with** open(**'clienti\_leasing.csv'**) **as** f:  reader = csv.reader(f)  **for** row **in** islice(reader, 10):   print(row) |

|  |
| --- |
| Exemplu 9. Citirea unui fișier și încărcarea unor coloane în liste (vectori) |
| **import** csv id\_client = [] nume\_client = [] sex = [] **with** open(**'clienti\_leasing.csv'**) **as** f:  reader = csv.reader(f)  **for** row **in** reader:  id\_client.append(row[0])  nume\_client.append(row[1])  sex.append(row[3]) print(id\_client) print(nume\_client) print(sex) |

|  |
| --- |
| Exemplu 10. Pentru mai multe fișiere identice, poate fi realizată o funcție pentru încărcarea datelor în liste |
| **import** csv **def** read\_f(filename):  col1 = []  col2 = []   **with** open(filename) **as** f:  reader = csv.reader(f)  **for** row **in** reader:  col1.append(row[1])  col2.append(row[2])   **return** col1, col2  col1, col2 = read\_f(**'clienti\_leasing.csv'**)  print(col1) print(col2)  print(len(col1)) |

|  |
| --- |
| Exemplu 11. Citește un fișier text cu csv.reader și prelucrează înregistrările |
| **import** csv **with** open(**'angajati.txt'**) **as** f:  reader = csv.reader(f, delimiter=**','**)  line\_count = 0  **for** row **in** reader:  **if** line\_count == 0:  print(**f'Numele coloanelor sunt: {", ".join(row)}'**)  line\_count += 1  **else**:  print(**f'\t{row[0]} lucreaza in departmentul {row[1]}, nascut in {row[2]}.'**)  line\_count += 1  print(**f'Au fost procesate {line\_count} linii.'**) |

|  |
| --- |
| Exemplu 12. Citește un fișier text sub formă de dicționar |
| **import** csv **with** open(**'angajati.txt'**) **as** f:  reader = csv.DictReader(f, delimiter=**','**)  **for** row **in** reader:  print(row) |
| Exemplu 13. Citește un fișier text sub formă de dicționar. Specificarea delimitatorului |
| **import** csv **with** open(**'angajati1.txt'**) **as** f:  reader = csv.DictReader(f, delimiter =**';'**)  **for** row **in** reader:  print(row)  #vs.  **import** csv **with** open(**'angajati1.txt'**) **as** f:  reader = csv.reader(f, delimiter =**';'**)  **for** row **in** reader:  print(row) |

|  |
| --- |
| Exemplu 14. Scrierea unui fișier *.csv* din liste |
| **import** csv **with** open(**'angajati.csv'**, mode=**'w'**, newline=**''**) **as** f:  writer = csv.writer(f)  writer.writerow([**'Cosmin Antonescu'**, **'Marketing'**, **'Noiembrie'**])  writer.writerow([**'Eugenia Marin'**, **'Vanzari'**, **'Iulie'**]) **with** open(**'angajati.csv'**) **as** f:  reader = csv.reader(f)  **for** row **in** reader:  print(row) |

|  |
| --- |
| Exemplu 15. Scrierea unui fișier text |
| f = open(**'angajati2.txt'**,**'w'**) f.write(**'Cosmin Antonescu, Marketing, Noiembrie \n'**) f.write(**'Eugenia Marin, Vanzari, Iulie'**) f = open(**'angajati2.txt'**) **for** row **in** f:  print(row) |

|  |
| --- |
| Exemplu 16. Scrierea unui fișier *.csv* din dicționare |
| **import** csv **with** open(**'angajati1.csv'**, mode=**'w'**) **as** f:  fieldnames = [**'Nume'**, **'Dep'**, **'Luna'**]  writer = csv.DictWriter(f, fieldnames=fieldnames)  writer.writeheader()  writer.writerow({**'Nume'**: **'Constantinescu Andrei'**, **'Dep'**: **'Contabilitate'**, **'Luna'**: **'Aprilie'**})  writer.writerow({**'Nume'**: **'Iliescu Emil'**, **'Dep'**: **'IT'**, **'Luna'**: **'Mai'**}) |

* Pachetul ***pandas***

Obiectele din *pandas* sunt cunoscute sub denumirea *DataFrame*, o structură de date tabelară, orientată pe coloane, cu etichete pentru rând și coloană, și *Series*, un obiect matricial unidimensional.

*Pandas* deține capacități de manipulare a datelor din foi de calcul și baze de date relaționale, oferă indexare pentru a ușura remodelarea datelor și efectueză agregări complexe.

|  |
| --- |
| Exemplu 17. Citirea unui fișier *.csv*, prin intermediul bibliotecii **pandas** |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df) |

|  |
| --- |
| Exemplu 18. Afișarea prin localizare cu iloc a unor coloane din setul inițial |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.iloc[:,2:4]) |

|  |
| --- |
| Exemplu 19. Afișarea cu iloc a unor coloane și înregistrări din setul inițial |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.iloc[8:10,2:4]) |

|  |
| --- |
| Exemplu 20. Afișarea cu loc a unor coloane și a unei înregistrări din setul inițial |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[0, **'NAME\_CLIENT'**:**'SEX'**]) |

|  |
| --- |
| Exemplu 21. Afișarea cu loc a unor coloane și înregistrări din setul inițial |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[[0,2], **'NAME\_CLIENT'**:**'SEX'**])  **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[0, [**'NAME\_CLIENT'**,**'SEX'**]])  **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[[0,3], [**'NAME\_CLIENT'**,**'SEX'**]]) |

|  |
| --- |
| Exemplu 22. Afișarea cu iloc și loc a unor coloane și înregistrări din *dataframe* inițial |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.iloc[0:3, [1,6,5]]) print(df.loc[0:3, **'NAME\_CLIENT'**:**'SEX'**]) |

|  |
| --- |
| Exemplu 23. Modificarea venitului pentru o anumită înregistrare |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[0, **'VENIT\_PER\_YEAR'**]) df.loc[0,**'VENIT\_PER\_YEAR'**]=30000 print(df.loc[0, **'VENIT\_PER\_YEAR'**]) |

|  |
| --- |
| Exemplu 24. Modificarea venitului dacă acesta este mai mic decât 5000 și vârsta este mai mare decât 30. Crearea cu copy() a unui *dataframe* df1 și calcularea salariului mediu |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**)  df1=df.copy()  df1.loc[(df1[**'VENIT\_PER\_YEAR'**]<5000)&(df1[**'AGE'**]>30),**'VENIT\_PER\_YEAR'**]=10000  print(df1[[**'NAME\_CLIENT'**,**'VENIT\_PER\_YEAR'**, **'AGE'**]]) print(df1.loc[(df1[**'VENIT\_PER\_YEAR'**]==10000)&(df1[**'AGE'**]>30), [**'NAME\_CLIENT'**,**'VENIT\_PER\_YEAR'**, **'AGE'**]])  print(**'Salariul mediu '**, df1[**'VENIT\_PER\_YEAR'**].mean())  df1.to\_csv(**'clienti\_leasing20mod.csv'**) df1 = pandas.read\_csv(**'clienti\_leasing20mod.csv'**,  header=0,  names=[**'ID'**, **'Nume'**, **'Functie'**, **'Sex'**,**'Moneda'**, **'Venit'**, **'Data'**, **'Varsta'**]) df1.to\_csv(**'clienti\_leasing20mod.csv'**) |

|  |
| --- |
| Exemplu 25. Afișarea clienților cu vârsta = 35 sau sex masculin |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[(df[**'AGE'**]==35)| (df[**'SEX'**]==**'m'**),[**'NAME\_CLIENT'**,**'JOB'**,**'SEX'**,**'AGE'**]]) |

|  |
| --- |
| Exemplu 26. Afișarea clienților care nu dețin funcția inginer |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[df[**'JOB'**] != **'Inginer'**, **'NAME\_CLIENT'**:**'VENIT\_PER\_YEAR'**]) |

|  |
| --- |
| Exemplu 27. Afișarea clienților al căror nume se termină în *a* |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[df[**'NAME\_CLIENT'**].str.endswith(**"a"**),[**'NAME\_CLIENT'**,**'SEX'**]]) |

|  |
| --- |
| Exemplu 28. Afișarea clienților al căror nume începe cu *Ha* |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[df[**'NAME\_CLIENT'**].str.startswith(**"Ha"**),[**'NAME\_CLIENT'**,**'SEX'**]])  /  **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) pandas.options.display.max\_columns = 10 print(df.loc[df[**'NAME\_CLIENT'**].str.startswith(**"Ha"**)]) |

|  |
| --- |
| Exemplu 29. Afișarea clienților care dețin funcțiile inginer și profesor |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.loc[df[**'JOB'**].isin([**'Inginer'**, **'Profesor'**]),[**'NAME\_CLIENT'**,**'SEX'**, **'JOB'**]]) |

|  |
| --- |
| Exemplu 30. Afișarea primelor 5 înregistrări (implicit) |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.head()) |

|  |
| --- |
| Exemplu 31. Afișarea ultimelor 7 înregistrări |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**) print(df.tail(7)) |

|  |
| --- |
| Exemplu 32. Preluarea unor coloane usecols și eliminarea unor înregistrări skiprows/skipfooter din setul inițial |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**, skiprows = [6,7,9], usecols = [**'NAME\_CLIENT'**,**'JOB'**]) print(df) |

|  |
| --- |
| Exemplu 33. Identificarea și modificarea tipului de dată |
| **import** pandas **as** pd df = pd.read\_csv(**'clienti\_leasing20.csv'**) print(df.dtypes) df = pd.read\_csv(**'clienti\_leasing20.csv'**, parse\_dates = [**'DATA'**]) df.AGE = df.AGE.astype(float) print(df.dtypes) |

|  |
| --- |
| Exemplu 34. Însumarea a două valori din aceeași coloană |
| **import** pandas df = pandas.read\_csv(**'clienti\_leasing20.csv'**, nrows=6, usecols = [**'NAME\_CLIENT'**,**'VENIT\_PER\_YEAR'**]) print(df) print(df[**'VENIT\_PER\_YEAR'**][0] + df[**'VENIT\_PER\_YEAR'**][5]) |

|  |
| --- |
| Exemplu 35. Funcția describe() |
| **import** pandas **as** pd df = pd.read\_csv(**'clienti\_leasing20.csv'**,usecols=[**'NAME\_CLIENT'**,**'JOB'**,**'SEX'**,**'CURRENCY'**,**'VENIT\_PER\_YEAR'**,**'DATA'**,**'AGE'**]) print(df.describe()) print(**'\n\*\*\*\*\*Functia describe() pe coloana JOB\*\*\*\*\*'**) print(df[**'JOB'**].describe()) print(**'\n\*\*\*\*\*Functia describe() pe coloana CURRENCY\*\*\*\*\*'**) print(df[**'CURRENCY'**].describe()) |

|  |
| --- |
| Exemplu 36. Selectare coloane în *dataframe* |
| **import** pandas **as** pd df = pd.read\_csv(**'clienti\_leasing20.csv'**,usecols=[**'NAME\_CLIENT'**,**'JOB'**,**'SEX'**,**'CURRENCY'**,**'VENIT\_PER\_YEAR'**,**'DATA'**,**'AGE'**]) *#Selectare coloana in dataframe* print(df.JOB.head()) print(df[**'JOB'**].head()) print(df.loc[:,**'JOB'**].head()) print(df.iloc[:,1].head()) |

|  |
| --- |
| Exemplu 37. Funcții sum(), mean(), median(), nunique(), max(), min() |
| **import** pandas **as** pd df = pd.read\_csv(**'clienti\_leasing20.csv'**,usecols=[**'NAME\_CLIENT'**,**'JOB'**,**'SEX'**,**'CURRENCY'**,**'VENIT\_PER\_YEAR'**,**'DATA'**,**'AGE'**]) print(**'Suma veniturilor '**, df[**'VENIT\_PER\_YEAR'**].sum(), **'\nMedia veniturilor'**, df[**'VENIT\_PER\_YEAR'**].mean()) print(**'Valoarea mediana '**, df[**'VENIT\_PER\_YEAR'**].median()) print(**'Valori unice '**, df[**'VENIT\_PER\_YEAR'**].nunique()) print(**'Venitul maxim '**, df[**'VENIT\_PER\_YEAR'**].max()) print(**'Venitul minim '**, df[**'VENIT\_PER\_YEAR'**].min()) |

|  |
| --- |
| Exemplu 38. Înlocuiește valoarea lipsă fillna() |
| **import** pandas **as** pd df = pd.read\_csv(**'clienti\_leasing20.csv'**,usecols=[**'NAME\_CLIENT'**,**'JOB'**,**'SEX'**,**'CURRENCY'**,**'VENIT\_PER\_YEAR'**,**'DATA'**,**'AGE'**]) print(df[**'AGE'**]) print(df[**'AGE'**].fillna(**'valoare lipsa'**))  **import** pandas **as** pd df = pd.read\_csv(**'clienti\_leasing20.csv'**,usecols=[**'NAME\_CLIENT'**,**'JOB'**,**'SEX'**,**'CURRENCY'**,**'VENIT\_PER\_YEAR'**,**'DATA'**,**'AGE'**]) values = df[**'AGE'**].mean() df[**'AGE'**] = df[**'AGE'**].fillna(value = values) print(df) |

|  |
| --- |
| Exemplu 39. Șterge coloane |
| **import** pandas **as** pd  pd.set\_option(**"display.max\_columns"**,10) df = pd.read\_csv(**'clienti\_leasing20.csv'**) *#Sterge o coloana cu axis = 1* df = df.drop(**"ID\_CLIENT"**, axis=1) print(df) *#Sterge o coloana cu parametrul columns* df = df.drop(columns=**"AGE"**) print(df) *#Sterge mai multe coloane* df = df.drop([**"JOB"**, **"SEX"**], axis=1) print(df) df.to\_csv(**'clienti\_df.csv'**, index = **False**) *#Sterge o coloana, daca inplace = True, returneaza None* df.drop(**"VENIT\_PER\_YEAR"**, axis=1, inplace = **True**) print(df) |

|  |
| --- |
| Exemplu 40. Șterge înregistrări |
| **import** pandas **as** pd pd.set\_option(**"display.max\_columns"**,10) df = pd.read\_csv(**'clienti\_leasing20.csv'**) *# Sterge liniile 3, 5, 8* df = df.drop([3,5,8], axis=0) print(df) *# Sterge clientii care detin functia Inginer. Pentru a sterge utilizand denumirea coloanei se seteaza indexul pe coloana respectiva* df = df.set\_index(**"JOB"**) df = df.drop(**"Inginer"**, axis=0) print(df) *# Afisez primele cinci linii cu iloc, dupa stergerea inregistrarilor* df = df.set\_index(**"ID\_CLIENT"**) df = df.iloc[0:5, ] print(df) |

|  |
| --- |
| Exemplu 41. Redenumire coloane |
| **import** pandas **as** pd pd.set\_option(**"display.max\_columns"**,10) df = pd.read\_csv(**'clienti\_leasing20.csv'**)  df = df.rename(columns={**"ID\_CLIENT"**: **"ID"**}) df = df.rename(columns={**"JOB"**: **"Functia"**}); print(df)  df=df.rename(  columns={  **"AGE"**: **"Varsta"**,  **"VENIT\_PER\_YEAR"**: **"Venit"** } ) print(df)  df=df.rename(columns=str.lower) print(df) |

2.3 Citire fișier *.json*

|  |
| --- |
| Exemplu 42. Import fișier *json* |
| **import** json **from** pprint **import** pprint **with** open(**'clienti\_daune.json'**) **as** f:  data = json.load(f) pprint(data) |

|  |
| --- |
| Exemplu 43. Prelucrare *json.* Numără frecvența de apariție a cuvintelor din Dauna din fișierul *json* |
| **import** json *#deschidem fisierul clienti\_daune.json* **with** open (**'clienti\_daune.json'**) **as** f:  data=json.load(f) *#obtinem lista tuturor cuvintelor din descrierea daunelor* lista\_cuvinte=[] **for** dauna **in** data:  lista\_cuvinte=lista\_cuvinte + str(str(dauna[**'Dauna'**]).lower()).split() print(lista\_cuvinte) *# numaram cuvintele si le adaugam intr-un dictionar* dictionar = {} **for** cuvant **in** lista\_cuvinte:  **if** cuvant **not in** dictionar:  dictionar[cuvant] = 1  **else**:  dictionar[cuvant] += 1 *#formam lista cuvintelor si a frecventei de aparitie sortata descrescator* aparitie = [] **for** key, value **in** dictionar.items():  aparitie.append((value, key)) aparitie.sort(reverse=**True**) print(aparitie) |

**Referințe**

<https://realpython.com/python-csv/>

<https://www.shanelynn.ie/python-pandas-read_csv-load-data-from-csv-files/>

<https://www.shanelynn.ie/select-pandas-dataframe-rows-and-columns-using-iloc-loc-and-ix/>

<https://pandas.pydata.org/pandas-docs/version/0.21/api.html#id2>

**Exerciții**

1) Să se creeze un nou set de date din clienti\_leasing.csv având coloanele name\_client, deposit\_amount și prescoring, dacă val\_credits\_ron = 0 și deposit\_amount > 150.000 și să se modifice coloana prescoring = 6 dacă deposit\_amount > 500.000.

2) La execițiul 43, se vor afișa cuvintele a căror frecvență de apariție > 1000, eliminându-se cuvintele *the, and, to, a*.