**ACADEMIA DE STUDII ECONOMICE DIN BUCUREȘTI, FACULTATEA DE CIBERNETICĂ, STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ**



*Proiect pachete software*

**Profesor coordonator: Student:**

asistent.univ.dr. Dobrita Gabriela Ariton Alexandru

grupa 1086, seria C

**București, 2024**

**DESPRE DATE**

Datele prelucrate se refera la o colectie de clienti care au cumparat 4 categorii de produse. Datele prezinta caracteristici precum pretul unui produs, id-ul clientului, data tranzactiei, metoda de plata etc.

Datele au fost preluate de pe Kaggle. - <https://www.kaggle.com/datasets/fahadrehman07/retail-transaction-dataset>

**Coloane**

CustomerID: Identificator unic pentru fiecare client.

ProductID: Identificator unic pentru fiecare produs.

Quantity: Numărul de unități achiziționate pentru un anumit produs.

Price: Prețul unitar al produsului.

StoreLocation: Locația unde a avut loc tranzacția.

ProductCategory: Categoria din care face parte produsul.

DiscountApplied: Procentul de reducere aplicat produsului.

TotalAmount: Suma totală plătită pentru tranzacție.

**PYTHON**

1. Importare date din fisier csv folosind pandas

Am importat fișierul CSV cu ajutorul pachetului Pandas și al funcției pd.read\_csv() și apoi am afișat datele conținute de acesta cu funcția print().

Exercitiul 1

*df=pd.read\_csv('./DATE/Retail\_Transaction\_Dataset.csv', index\_col=0)  
# cu index\_col am setat ca 'customer\_id' sa fie indexul data frame-ului  
pd.set\_option('display.max\_rows', 50)  
pd.set\_option('display.max\_columns', 6)  
print(df)*

*A screenshot of a computer

Description automatically generated*

1. Utilizare dictionare, liste

Exercitiul 2

*# Sa se creeze o lista cu toate locatiile magazinelor si sa se ordoneze alfabetic  
locatii=list(df['StoreLocation'])  
locatii.sort()  
print(locatii)*

Output:

A screen shot of a computer screen

Description automatically generated

Se folosește metoda list() pentru a transpune înregistrările din coloana “StoreLocation” într-o listă de stringuri.

La final aplicam functia .sort() pe lista nostra, care va modifica lista initiala si va sorta in alfabetic locatiile.

Avand in vedere ca sunt multe inregistrari, print-screen-ul arata doar cateva locatii.

Exercitiul 3

*# Pe baza listei de produse sa se afle cantitatea totala vanduta din fiecare produs  
cantitate\_vanduta={}  
for l in produse:  
 cantitate\_vanduta[l]=produse.count(l)*

Output

{'C': 2531, 'A': 2502, 'D': 2497, 'B': 2469}

Pentru a numara de cate ori apare un produs in lista am folosit functia .count(numeProdus)

Exercitiul 4

Sa se determine determine numarul total de produse vandute folosindu-ne de dictionarul creat mai sus.

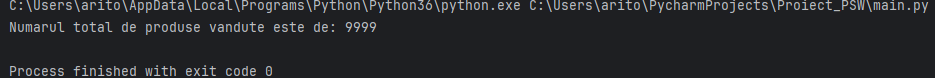
numar\_total\_produse=0

for key in cantitate\_vanduta:

numar\_total\_produse +=cantitate\_vanduta[key]

print("Numarul total de produse vandute este de: "+str(numar\_total\_produse))

Output:

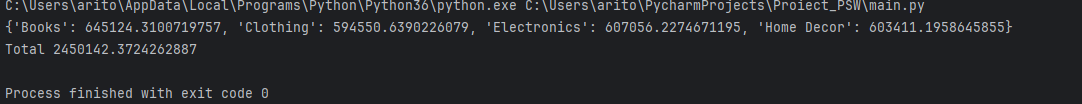


Cu ajutorul unui for am parcurs dictionarul si in variabila numar\_total\_produse am adaugat numarul total de produse vandute, rezultatul final fiind 9999

Exercitiul 5

*Sa se creeze un dictionar care are ca si cheie categoria produsului iar ca si valoare* *totalul amount-ul platit de client pe tranzactii*

*tranzactii\_categorie=df.groupby("ProductCategory").sum()["TotalAmount"].to\_dict()  
print(tranzactii\_categorie)  
total=0  
for key in tranzactii\_categorie:  
 total+=tranzactii\_categorie[key]  
print("Total "+str(total))*



Am grupat setul dupa ”ProductCategory” si am realizat suma in functie de coloana „TotalAmount”.Pe urma am apelat metoda .to\_dict() pentru a transpune rezultatul obtinut intr-un dictionar care are ca si cheie categoria produselor iar ca valoare totalul amount-ul platit de client pe tranzactii.

Putem observa, de exemplu, ca pentru produsele de tipul Books s-a inregistrat o valoare totala a tranzactiilor facute de catre clienti de 645124.31 euro.

1. Definirea si apelarea de functii

Exercitiul 6

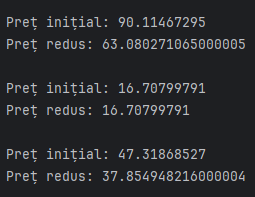
Sa se aplice o reducere produselor:

- disscount-ul care ii este deja atribuit daca pretul este pana in 40 euro  
- 20% daca pretul este intre 40-50 euro  
- 30% daca pretul este peste 50 euro

*def reducere( pret, disscount):  
 if pret<=40:  
 if pret-disscount\*pret < 0:  
 return pret  
 else:  
 return pret-disscount\*pret  
 elif pret>40 and pret<=50:  
 return pret-0.20\*pret  
 elif pret>50:  
 return pret-0.30\*pret  
 else:  
 return pret*

Aplicare functie:

*for i in df.itertuples():  
 print('Preț inițial:', i.Price)  
 print('Preț redus:', reducere(i.Price, i.DiscountApplied), '\n')*

**

Am definit o functie care primeste ca si parametru pretul prodului, iar cu ajutorul unei structuri conditionale-IF am aplicat reducerea.Vom parcurge mai apoi randurile cu metoda .itertuples() si vom aplica functia.

1. Utilizarea structuri conditionale

Exercitiul 7

Se va afisa DA daca produsul are pretul >35 pentru a primi reducere in caz contrar NU

Am creat doua liste pentru a putea accesa mai usor valorile din respectivele coloane

lista\_coloane=list(df)

*produsele=df[lista\_coloane[0]]*

prices=df[lista\_coloane[21]]

*for index, row in df.iterrows():  
 produs = row[lista\_coloane[0]]  
 price = row[lista\_coloane[2]]  
 if price is not None and float(price) > 35:  
 print(f"Produsul: {produs} cu pretul: {price} reducere: DA")  
 elif price is not None:  
 print(f"Produsul: {produs} cu pretul: {price} reducere: NU")  
 else:  
 print(f"Pretul pentru produsul: {produs} nu este disponibil.")*

*A screenshot of a computer

Description automatically generated*

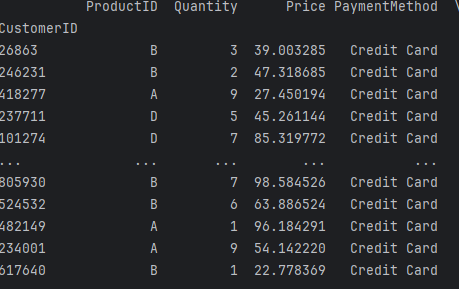
Pentru a putea rezolva mai usor exercitiile, am retinut in doua variabile o lista a produselor si una a preturilor. Cu ajutorul unui FOR am iterat prin lista de produse, verificandu-le pretul cu ajutorul lui IF, iar pe urma am afisat conform pretului.

1. Accesarea datelor folosind loc si iloc

Exercitiul 8

Sa se afiseze doar randurile care au ca si modalitate de plata 'Credit Card' si au discount-ul < 15

*print(df.loc[(df['PaymentMethod']=='Credit Card')&(df['DiscountApplied']<15)])*

**

*A screenshot of a computer screen

Description automatically generated*

Pentru acest exercitiu ne-am folosit de .loc() in interiorul caruia am verificat si niste conditii, daca modalitatea de plata este ‚Credit Card’ si daca discount-ul este mai mic de 15%.

.loc()-selecteaza linii si coloane dupa denumirea liniilor/colonelor

Exercitul 9

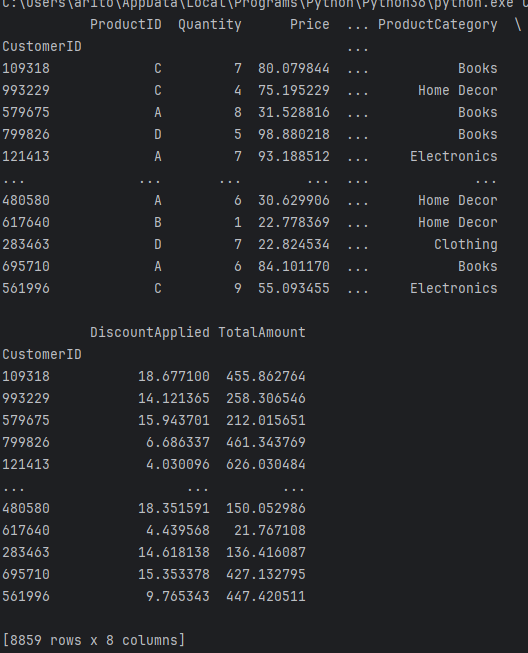
Sa se stearga clientii care au cumparat produsele care au pretul<20

Am retinut in variabila rows\_to\_delete index-urile randurilor care au pretul <20 iar cu functia .drop() am sters randurile precizand axis=0 care indica ca vom sterge randuri

*rows\_to\_delete=df.loc[df['Price'] < 20].index.values*

df=df.drop(rows\_to\_delete,axis=0)

print(df)



Daca intial erau  acum sunt 

1. Utilizare de functii de grup

Exercitiul 10

Sa se calculeze pretul mediu pentru fiecare produs

print('Prețul mediu de vânzare pentru fiecare produs: \n')

*df\_pret\_mediu=df.groupby('ProductID')['Price'].mean()*

print(df\_pret\_mediu)

*A screenshot of a computer

Description automatically generated*

Cu ajutorul functiei .groupby() am grupat datele dupa coloana ‚ProductID’ si am apelat functia .mean() pe coloana ‚Price’, obtinand media preturilor aferente fiecarui produs.

Exercitiul 11

Creare raport:cantitatea de produse vandute din fiecare tip de la fiecare categorie

*print("Numarul de produse vandute din fiecare tip organizate in functie de categorie")*

*raport=df.groupby(['ProductCategory','ProductID'])['ProductID'].count()*

print(raport)

*A screenshot of a computer

Description automatically generated*

Pentru acest exercitiu am grupat datele dupa coloana ‚ProductID’ cu ajutorul functiei .groupby(), pe urma am aplicat functia count() pe aceeasi coloana si am afisat corespunzator, numele categoriei produselor urmat de numele produselor si cantitatile aferente.

1. Prelucrari statistice, agregarea si gruparea datelor

Exercitiul 12

Sa se afiseze pretul minim si maxim pentru fiecare produs

*prices\_max\_min=df.groupby(['ProductID']).agg({'Price':[min,max]})*

print("Pretul minim si maxim pentru fiecare produs:")

print(prices\_max\_min)

A screenshot of a computer

Description automatically generated

Am grupa datele dupa coloana ‚ProductID’ si am apelat functia .agg(), dandu-i ca si parametru coloana dupa care facem agregare ‚Price’ si lista [min.max] pentru a afisa pretul minim si maxim pentru fiecare produs.

1. Grafice

Exercitiul 13

Se va face un grafic de tip bar pentru produsele ale caror pret>30 euro.

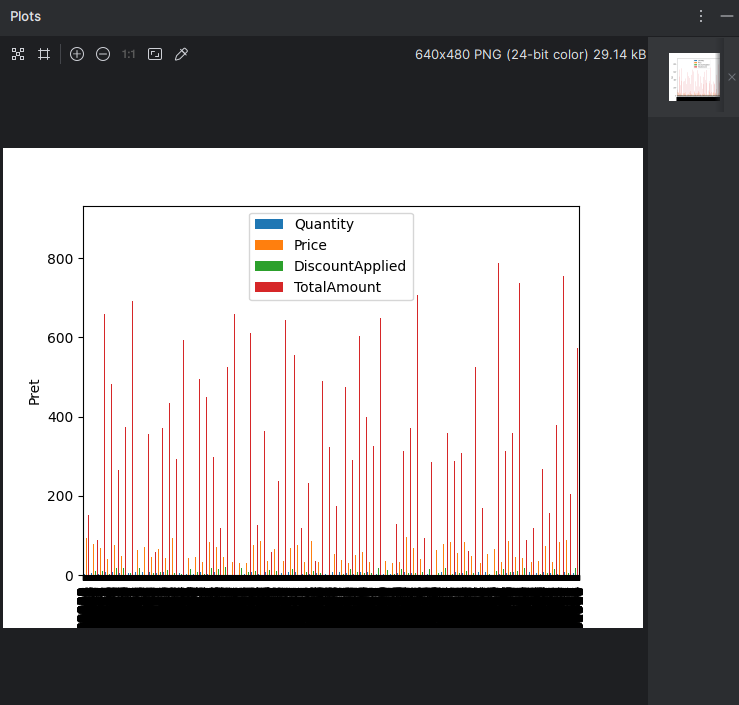
import matplotlib.pyplot as plt

df[df['Price']>30].plot(kind='bar')

plt.xlabel('Produs')

plt.ylabel('Pret')

plt.show()

**

Cu ajutorul biblioteci matplotlib am creat un grafic de tip ‚bar’ pentru acele randuri care au pretul produselor mai mare decat 30. Am setat labelul pentru axa OX si OY si pe urma am afisat graficul cu ajutorul functiei .show(). Se poate vedea, asadar, distributia preturilor aferente produselor.

**SAS**

1. Creare set de date dintr-un fisier extern.

S-au importat datele dintr-un fisier excel si s-a creat un set de date persistent.

Exercitiul 1

libname mydata '/home/u63865503/Proiect\_PSW';

proc import datafile='/home/u63865503/Proiect\_PSW/Retail\_Transaction\_Dataset.xlsx'

out=mydata.retail\_data

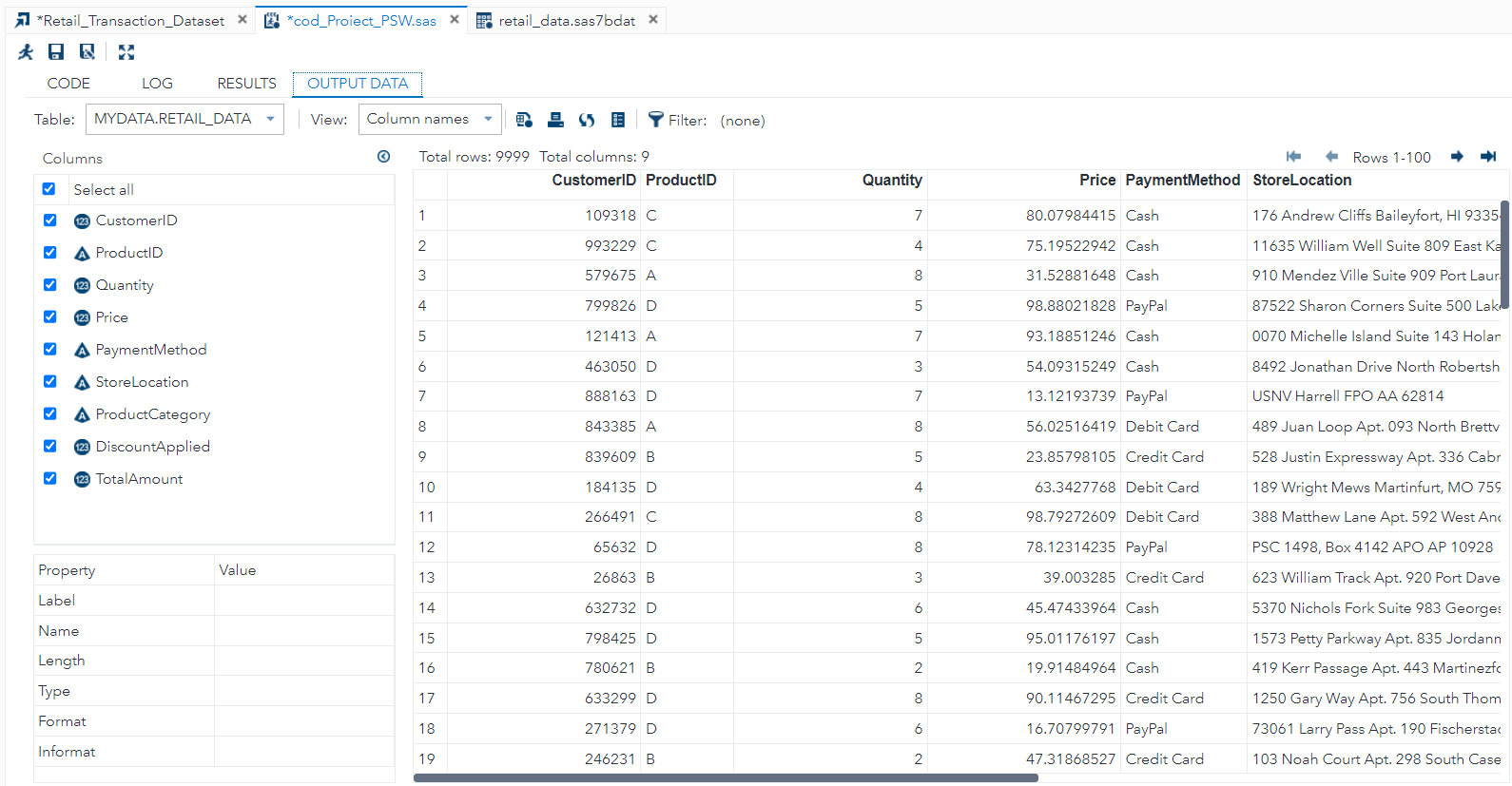
dbms=xlsx

replace;

run;

Pentru acest exercitiu ne-am folosit de instructiunea LIBNAME pentru a crea libraria.

Pe urma ne-am folosit de procedura PROC IMPORT in care am specificat calea setului de date,tipul de fisier cu DBMS, unde sa fie stocate datele cu OUT.



1. Creare si utilizare formate create de utilizator.

Se va crea un format care schimba denumirea categoriilor produselor din engleza in romana

Exercitiul 2

proc format;

value $prodcatf

'Books' = 'Carti'

'Electronics' = 'Electronice'

'Home Decor' = 'Decoratiuni Casa';

run;

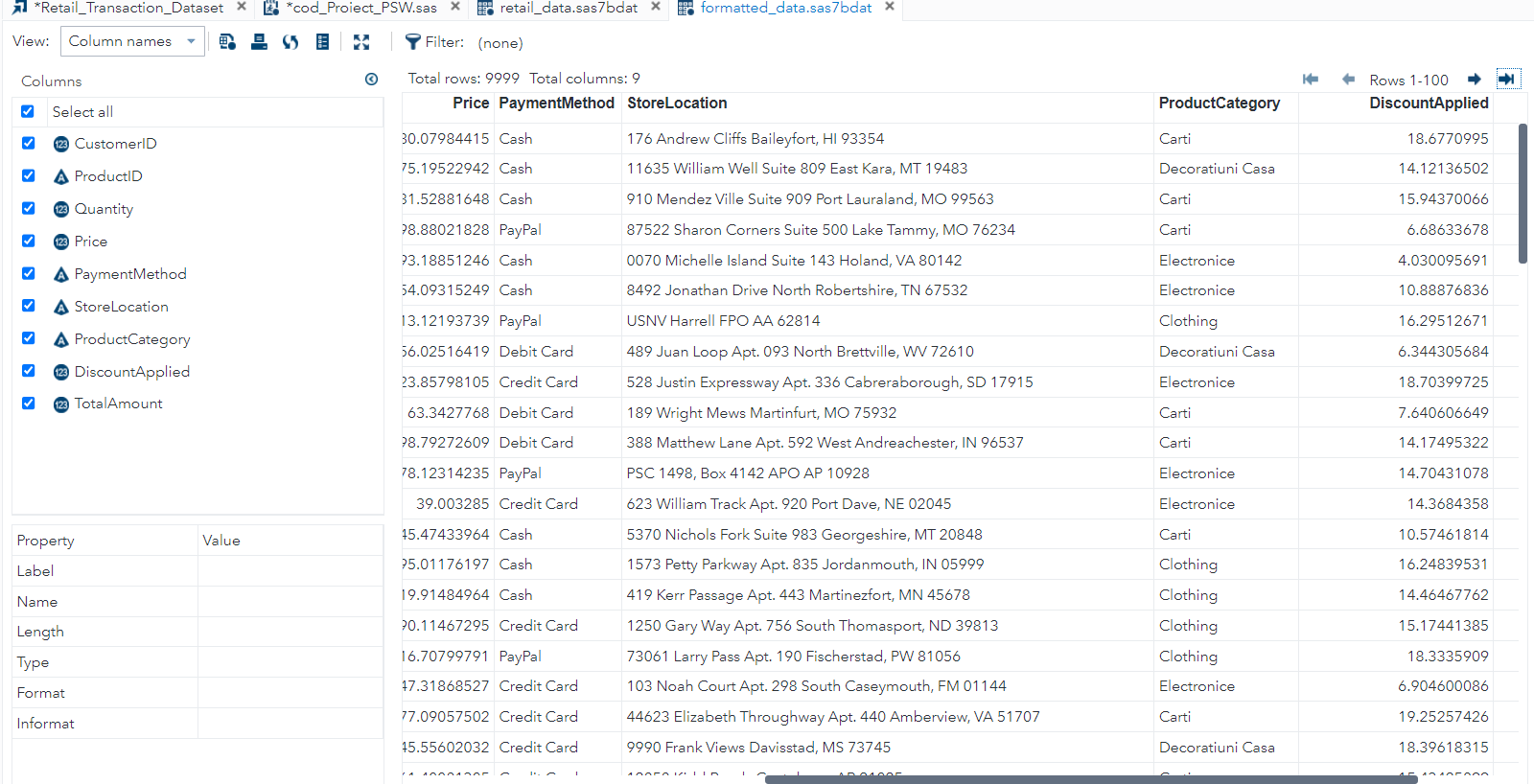
data mydata.formatted\_data;

set mydata.retail\_data;

format ProductCategory $prodcatf.;

run;

Pentru acest exercitiu ne-am folosit de value, urmat de numele formatului care aici este $prodcatf care va reprezenta valoarea aferenta inregistrarii din tabel. Aplicarea formatului se face prin mentionarea denumirii urmata de semnul punct.



Interpretare:

Se poate observa ca valorile din coloana **ProductCategory** au fost inlocuite acum,in urma aplicarii formatului.

Frecventa cu care apare fiecare produs

Exercitiul 3

title 'Frecventa de aparitie a fiecarui produs';

proc FREQ data= mydata.retail\_data;

TABLES ProductID/nocum nopercent;

run;

Cu ajutorul lui FREQ vom afla frecventele de aparitie a fiecarui produs. Nocum si nopercent ne ajuta sa nu mai afisam procentul de aparitie in tabel.

*A screenshot of a computer

Description automatically generated*

Frecventa de aparitie a fiecarei modalitati de plata

Exercitiul 4

title 'Frecventa de aparitie a fiecarei modalitati de plata';

proc FREQ data= mydata.retail\_data;

TABLES PaymentMethod/nocum nopercent;

run;

Asemanatorul cu exercitiul 3.

A screenshot of a computer screen

Description automatically generated

Conform rezultatelor obtinute, putem observa ca plata Cash a fost cea mai folosita.

1. Procesarea iterativa si conditionata a datelor

Sa se afiseze produsele care au pretul cuprins intre 50-80 euro.

Exercitiul 5

title 'Produsele cu pretul intre 50-80 euro';

proc print data=mydata.retail\_data;

where Price between 50 and 80;

var ProductID Price;

run;

Pentru rezolvarea acestui exercitiu am avut nevoie de instructiunea WHERE pentru a crea o procesare conditionala a datelor.Pentru a indica intervalul in care vrem sa se afle pretul produsului am folosit operatorul BETWEEN.

Iar cu ajutorul proceduri PRINT am afisat produsele, selectand doar coloanele ProductID si Price folosind VAR.

Astfel, se iterează printre înregistrările din setul de date importat și se afișează acele produse care îndeplinesc cerințele specificate.

A screenshot of a computer

Description automatically generated

Asadar au fost afisate doar produsele care au pretul intre 50 si 80 euro.

Sa se acorde o reducere produselor care au pretul mai mare de 66 euro si sa se creeze un set nou de date.

Pentru acest exercitiu vom folosi instructiunea conditionala IF pentru a verifica pretul produselor.

In plus vom preciza lungimea coloanei DiscountApplied ca fiind de maxim 2 caractere, valori posibile fiind DA(se acorda reducere) NU(nu se acorda reducere);

Exercitiul 6

data acordaReducere replace;

set mydata.retail\_data;

length Reducere $2;

if Price>66 then

do;

Reducere="DA";

end;

else

do;

Reducere="NU";

end;

run;

title 'Produse care beneficiaza sau nu de reducere';

proc print data=acordaReducere;

var ProductID Price Reducere;

run;

A screenshot of a computer

Description automatically generated

Doar produsele care au pretul peste 66 vor fi eligibile pentru a primi reducerea.

1. Creare de subseturi de date, utilizarea de functii SAS

Am creat un top al produselor vandute pe care mai apoi le-am sortat descrescator in functie de cantitatea lor.

title 'Frecventa de aparitie a fiecarui produs';

proc FREQ data=mydata.retail\_data;

TABLES ProductID/nocum nopercent out=mydata.top\_produse;

run;

proc sort data=mydata.top\_produse out=mydata.top\_produse\_sortate;

by descending COUNT;

run;

Pentru a sorta datele am folosit SORT si opertaroul BY specificand ordinea -DESCENDING si coloana dupa care ordonez, in cazul meu fiind COUNT.

Am creat doua subseturi de date, unul pentru o categorie – Books si unul pentru alta categorie –Electronics.

Pentru primul subset am ales doar acele produse care au avut totalul tranzactiei mai mare de 600 euro.

Exercitiul 7

DATA mydata.Books; SET mydata.retail\_data;

WHERE ProductCategory eq 'Books' AND TotalAmount gt 600;

RUN;

*A screenshot of a computer

Description automatically generated*

Pentru cel de-al doilea subset am ales acele produse care au totalul tranzactiei mai mare decat 700 euro.

Exercitiul 8

DATA mydata.Electronics; SET mydata.retail\_data;

WHERE ProductCategory eq 'Electronics' AND TotalAmount gt 700;

RUN; *A screenshot of a computer

Description automatically generated*

Pentru acest exercitiu am folosit instructiunea WHERE pentru a filtra datele in functie de denumirea si total amount-ul produsului.

Subseturile create au fost adaugate in libraria MYDATA

A screenshot of a computer

Description automatically generated

1. Utilizarea de proceduri de raportare

Sa se creeze un raport privind totalul de produse vandute in functie de categorie si sa se insumeze totalul la final.

Exercitiul 9

PROC SORT DATA=mydata.retail\_data; BY ProductCategory;

PROC PRINT DATA=mydata.retail\_data sumlabel='Total vanzari #byval(ProductCategory)'

grandtotal\_label='Total';

BY ProductCategory;

SUM Price;

TITLE 'Numarul de produse cumparate dintr-o anumita categorie'; RUN;

Am folosit SORT pentru a sorta datele dupa(BY) numele categorie. Pentru afisare am folosit PRINT, definind o eticheta aferenta totalului cu ajutorul SUMLABEL iar la final afisand totalul insumat pentru toate categoriile setand labelul ‚Total’ cu GRANDTOTAL\_LABEL.

Aceasta insumare se face dupa(BY) numele categoriei si se face suma(SUM) pe coloane Price;

A screenshot of a computer

Description automatically generated

Se poate observa ca totalul pentru categorie BOOKS este de 140624.04449euro.

Sa se creeze un raport privind cantitatea vanduta din fiecare produs care este oferit de categoria ‚Books’ si totalul vanzarilor aferente fiecarui produs.

Exercitiul 10

PROC SORT data=mydata.retail\_data; WHERE ProductCategory eq 'Books';

by ProductID;

PROC PRINT data=mydata.retail\_data

sumlabel="Total #byval(ProductID)" grandtotal\_label="Total";

by ProductID;

sum Price;

title "Total cantitati vandute";

RUN;

Ne-am folosit de SORT pemtru a sorta inregistrarile dupa numele produselor, doar pentru categoria BOOKS.

Cu PRINT am afisat datele atribuind cate un label personalizat sumei fiecarui produs folosind notatia #byval(nume\_coloana). Datele au fost insumate cu ajutorul instructiuni SUM urmata de numele coloanei.

A screenshot of a computer

Description automatically generated

1. Proceduri statistice

Sa se identifice cele mai mari si cele mai mici 5 preturi distincte.

Exercitiul 11

PROC UNIVARIATE DATA=mydata.retail\_data

NEXTRVAL=5 NEXTROBS=0;

VAR Price;

Title "Indicatori statistici cu valori limita distincte pentru pretul produselor";

RUN;

De ajutor pentru acest exercitiu ne-au fost instructiunele UNIVARIATE pentru prelucrarea statistica, si instructiunile NEXTRVAL pentru care am atribuit valoarea 5, ne ajuta sa specificam numarul de variabile pe care vrem sa le includem in analiza, iar NEXTROBS specifica cate observatii sa omitem inainte de a incepe analiza.

*A screenshot of a computer

Description automatically generated*

1. Grafice

Bar chart pentru primele 10 produse.

Se va afisa numele produsului si totalul vanzarilor.

Exercitiul 12

data mydata.top\_10\_produse;

set mydata.top\_produse\_sortate(obs=10);

TITLE 'TOP PRODUSE VANDUTE';

proc sgplot data = mydata.top\_10\_produse;

vbar ProductID / response=Count;

xaxis display=(nolabel);

yaxis grid;

run;

Cu ajutorul PROC SGPLOT am creat un grafic de tip bar, pe verticala am setat frecventa de aparitie cu ajutorul lui VBAR, iar pe axa orizontala am afisat denumirile produselor.

*A graph with text on it

Description automatically generated*

Grafic- date statistice

TITLE 'GRAFIC';

proc univariate DATA=mydata.retail\_data PLOT;

VAR TotalAmount;

RUN;

*A screenshot of a graph

Description automatically generated*

A graph of a number of bars

Description automatically generated with medium confidence

A graph with a line

Description automatically generated

**SAS** **Enterprise Guide**

1. importul unui fisier non-SAS

A screenshot of a computer

Description automatically generated

1. grafice

A screenshot of a computer

Description automatically generated

1. prelucrări statistice

A screenshot of a computer

Description automatically generated