**ACADEMIA DE STUDII ECONOMICE BUCUREŞTI**

**FACULTATEA DE CIBERNETICĂ, STATISTICĂ ŞI INFORMATICĂ ECONOMICĂ**

**DAMEN SHIPYARDS GRONINGEN**

**Coordonator științific**

Conf. Univ. Dr. Gabriela ENE

Student:

Batog Florin

**Bucureşti**

**2021**

Cuprins

[EXCEL 2](#_Toc72959467)

[SAS Studio 9](#_Toc72959468)

[SAS ENTERPRISE GUIDE 20](#_Toc72959469)

[PYTHON 33](#_Toc72959470)

# EXCEL

1. Firma Damen Shipyards Groningen dorește să afle care segment de activitate contribuie mai mult în realizarea venitului său. Să se creeze o nouă coloană în care se va afișa denumirea segmentului care contribuie cel mai mult la veniturile companiei în trimestrul respectiv.

Insert->Formula->Logical->IF. În primul input introducem expresia după care se face comparația. În următoarele două input-uri, introducem textul „Produse” dacă expresia este adevărată. În caz contrar, se va trece la textul „Servicii”.

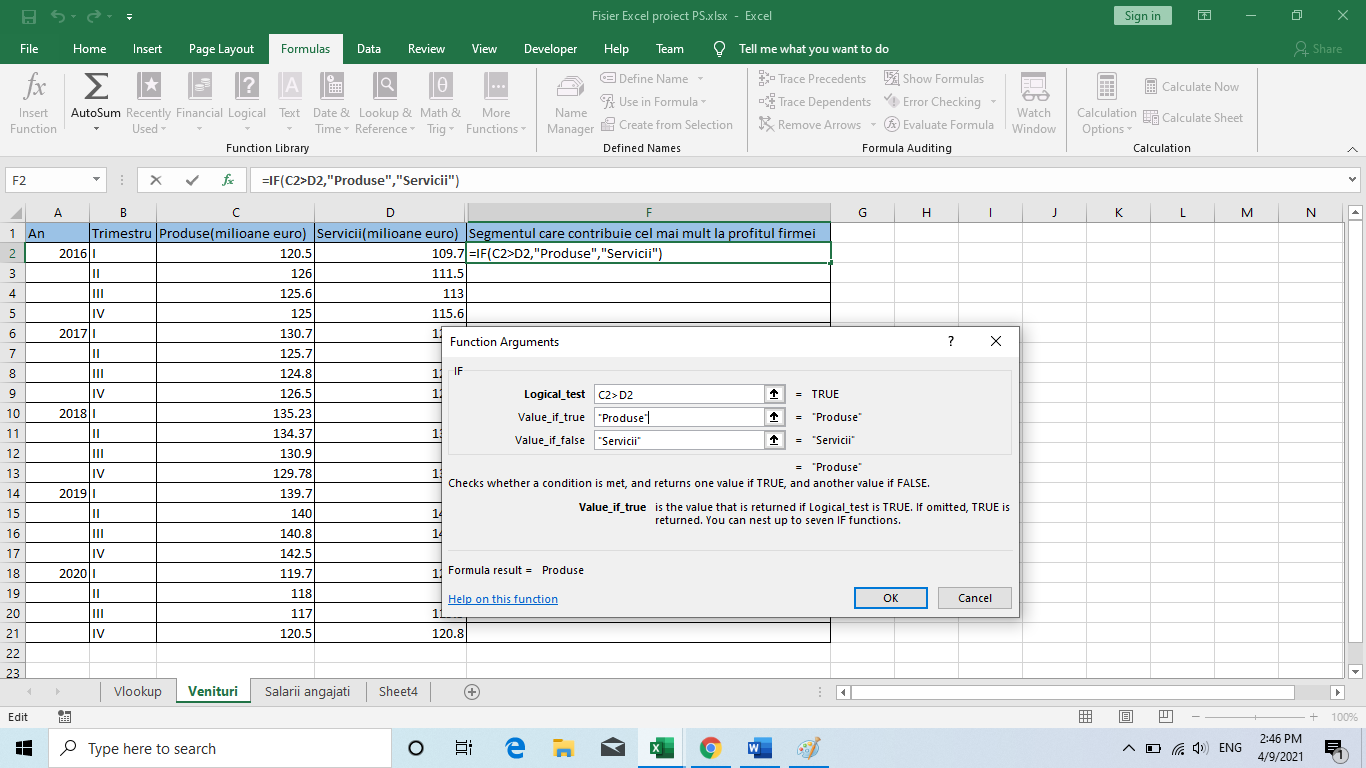


Figura 1. Aplicarea funcției logice IF

1. Să se creeze două coloane: una în care vor fi afișate veniturile totale obținute pe fiecare trimestru și o coloană pentru a calcula venitul mediu obținut în fiecare an.

Se face suma dintre celulele C2 și D2 , rezultatul fiind stocat in celula E2. Pentru cea de-a doua coloană se folosește funcția AVERAGE care se găsește în Formulas->More Functions->Statistical->AVERAGE. Parametrii acestei funcții sunt valorile pentru care se face media, adică veniturile înregistrate în trimestrele I,II,III și IV.

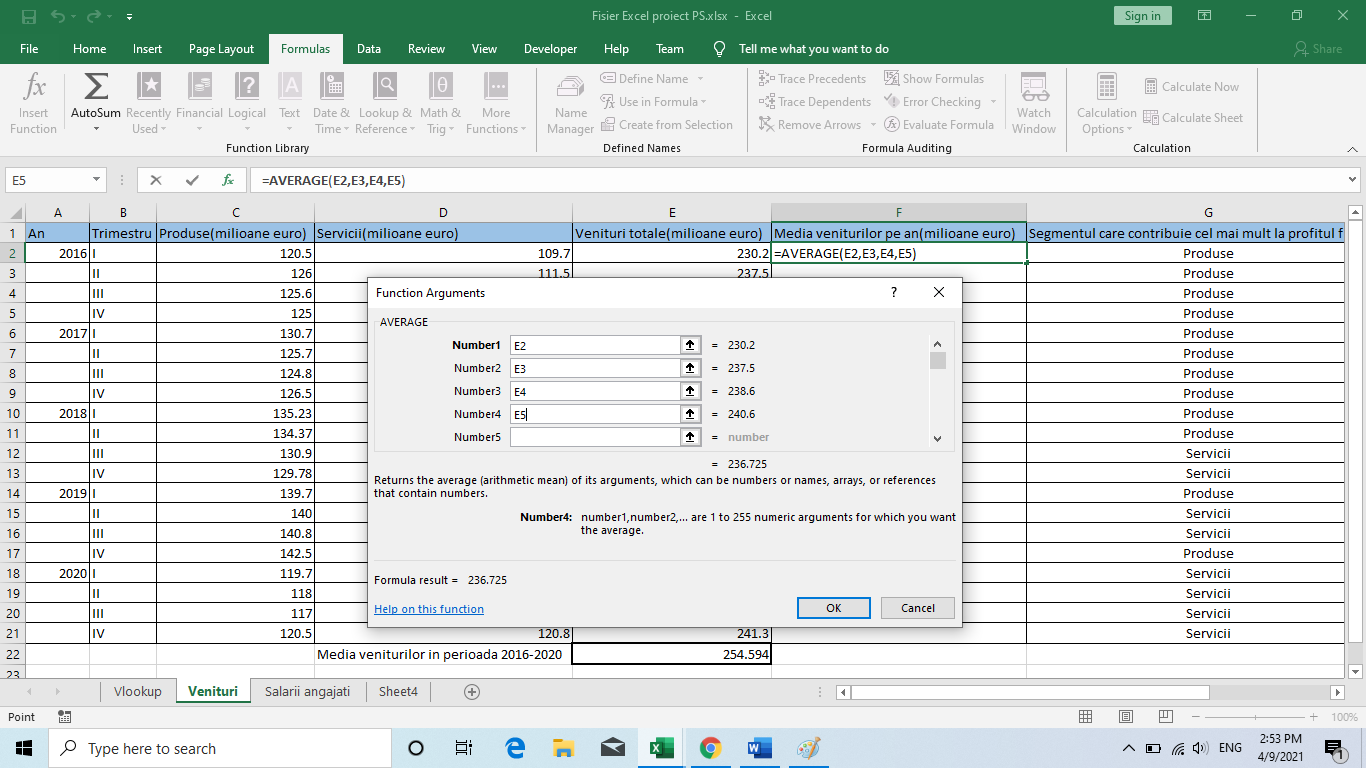


Figura 2. Aplicarea funcției statistice AVERAGE

1. Să se realizeze un grafic tip linie în care să se evidențieze evoluția venitului firmei Damen pe segmente de activitate. Graficul va fi vizibil într-o nouă foaie de calcul.

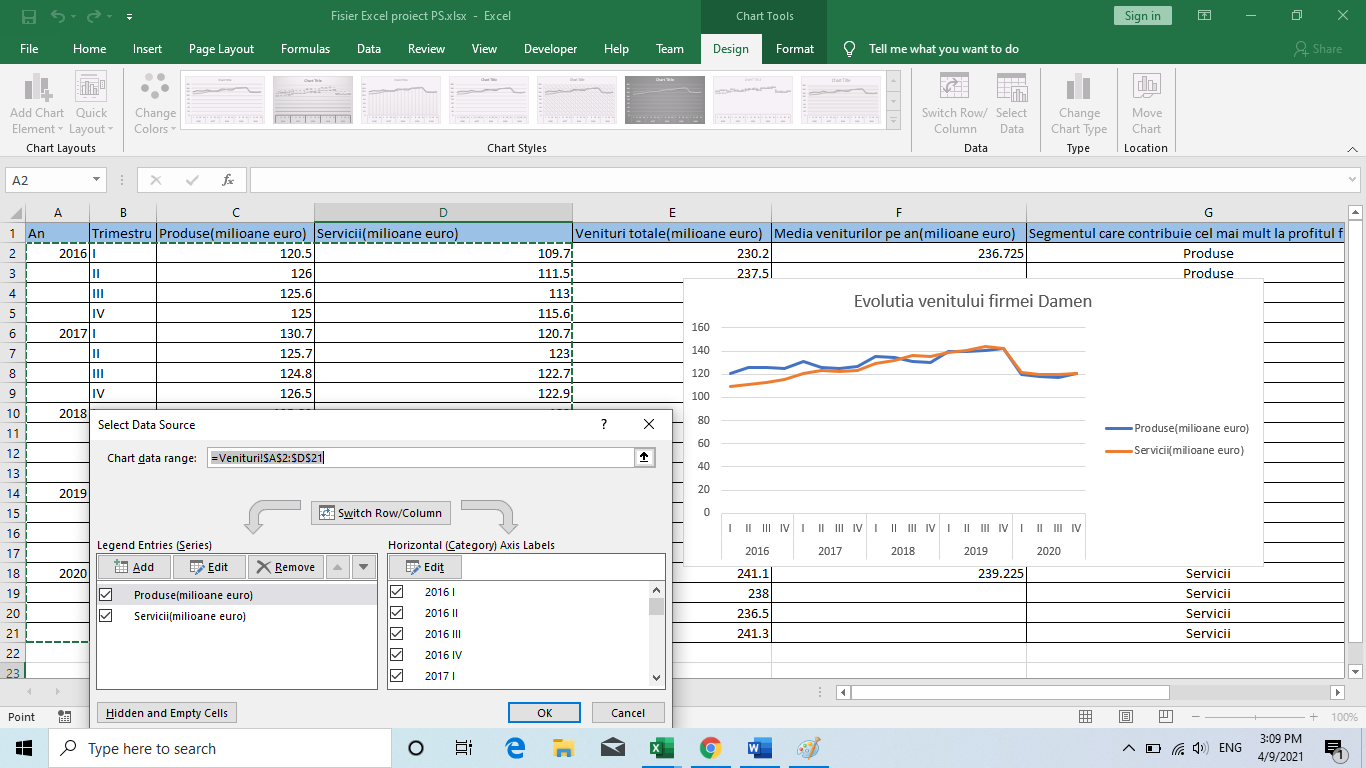


Figura 3. Introducerea datelor pe segmente de activitate pentru reprezentarea grafică

Pentru a realiza graficul tip linie mergem în Insert->Charts de unde alegem graficul tip linie. Pentru a introduce datele apăsăm pe Select Data. Odată apăsat, Excel-ul vă va deschide o fereastră unde va trebui să specificați porțiunea din tabel de unde trebuie preluate datele.

Pentru mutarea graficului într-o nouă foaie de calcul se va apăsa pe Move Chart. La deschiderea ferestrei, se bifează opțiunea New Sheet.

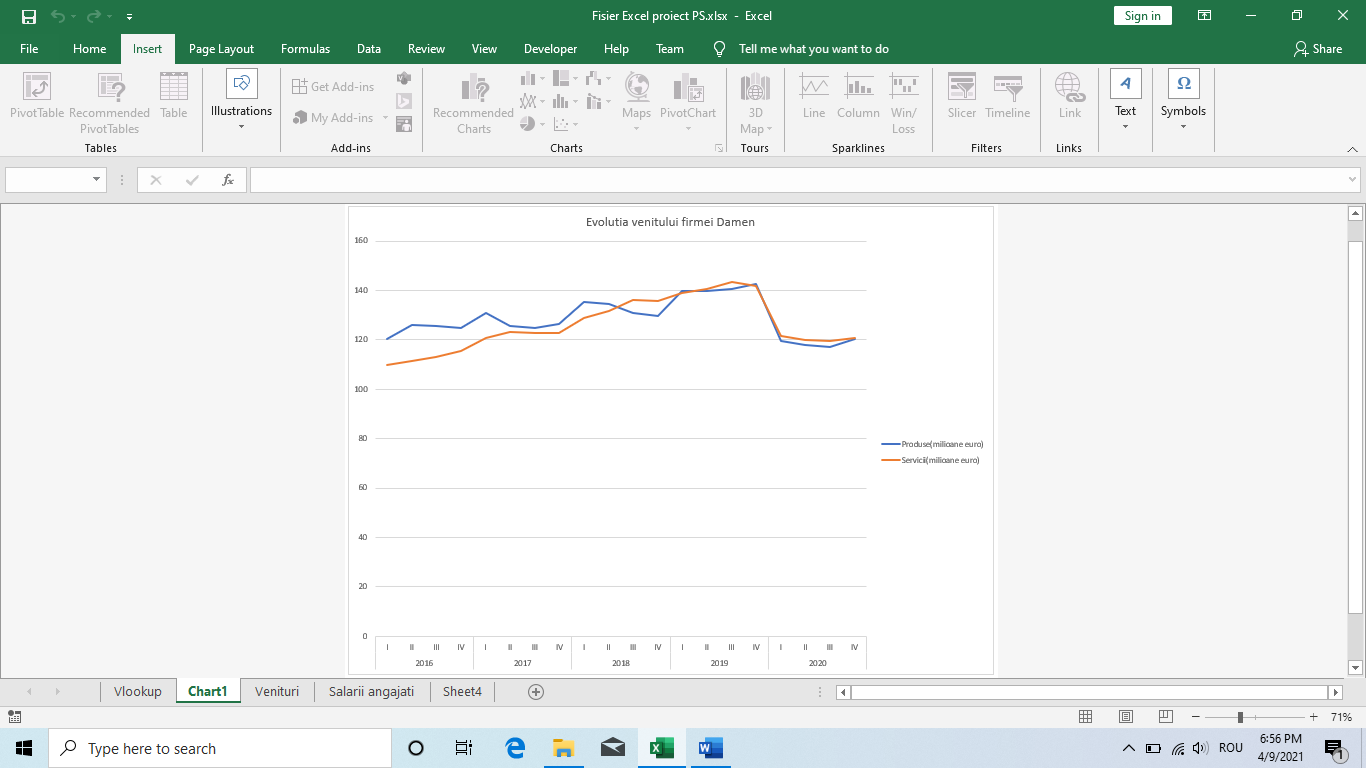


Figura 3. Rezultatul mutării graficului într-o foaie de calcul nouă

1. În luna ianuarie a anului 2020, firma Damen a primit multe comenzi din partea clienților săi, astfel că angajații săi au fost nevoiți să efectueze ore suplimentare. O oră este plătită cu 60 lei. Să se găsească o soluție astfel încât angajații de la contabilitate să poată obține salariul după plata orelor suplimentare, introducând doar numărul de ore lucrate.

Pentru a ușura sarcina departamentului de contabili­­­­tate putem realiza o listă derulantă cu toți angajații. Pentru acest lucru mergem în Data-> Data Tools->Data Validaton. Se va selecta opțiunea List și se va alege porțiunea din tabela cu angajați din sheet-ul Salarii angajați, apoi se va apăsa pe Ok.

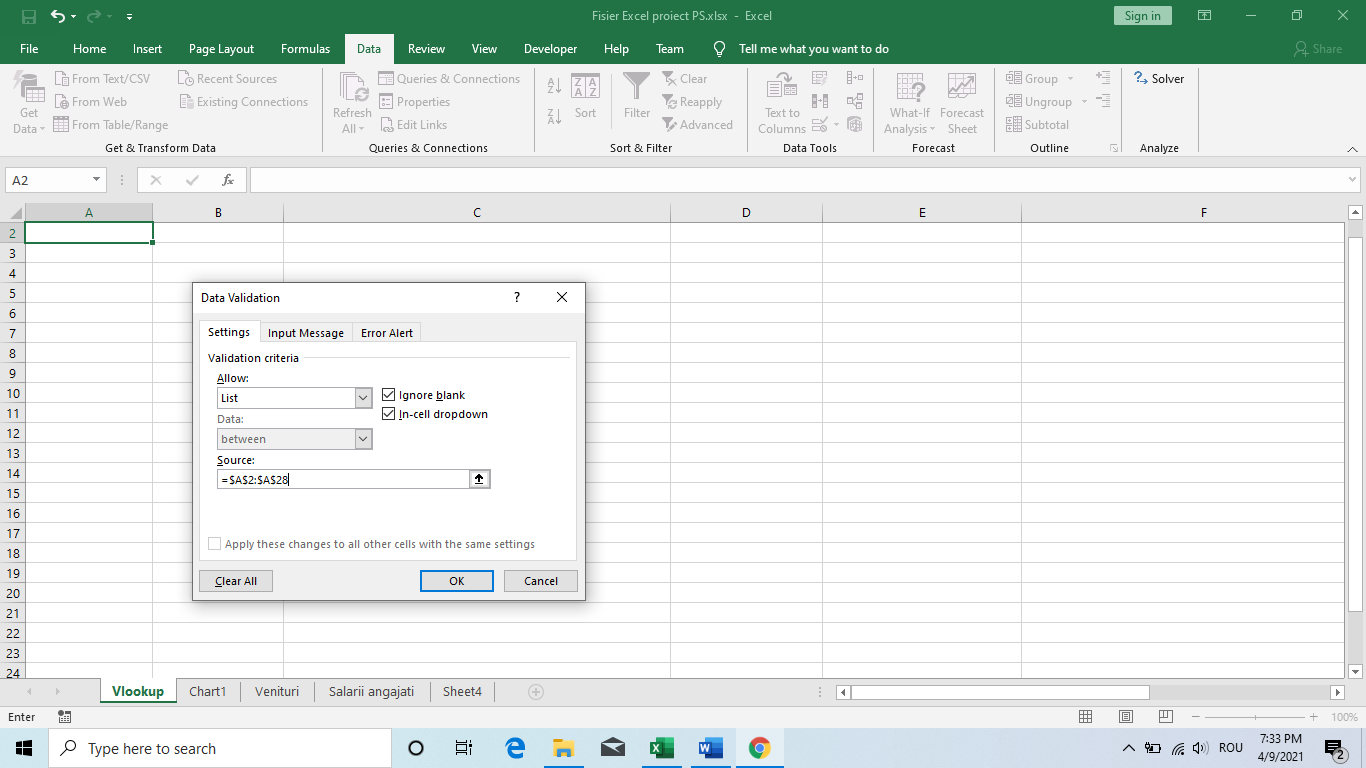


Figura 4.

Pentru a obține salariile angajaților din sheet-ul Salarii angajati vom folosi funcția VLOOKUP. Scriem = urmat de VLOOKUP. Primul parametru reprezintă celula din sheet-ul Vlookup unde va fi afișată lista cu angajați, adică celula A2. Parametrul 2 reprezintă tabelul din care se caută salariul, adică tabelul din sheet-ul Salarii angajati. Următorul parametru reprezintă indexul coloanei din tabel. În cazul nostru indexul va fi 2 pentru că coloana a doua conține salariile angajațiilor. Ultimul parametru este opțional. Îl vom seta cu 0.

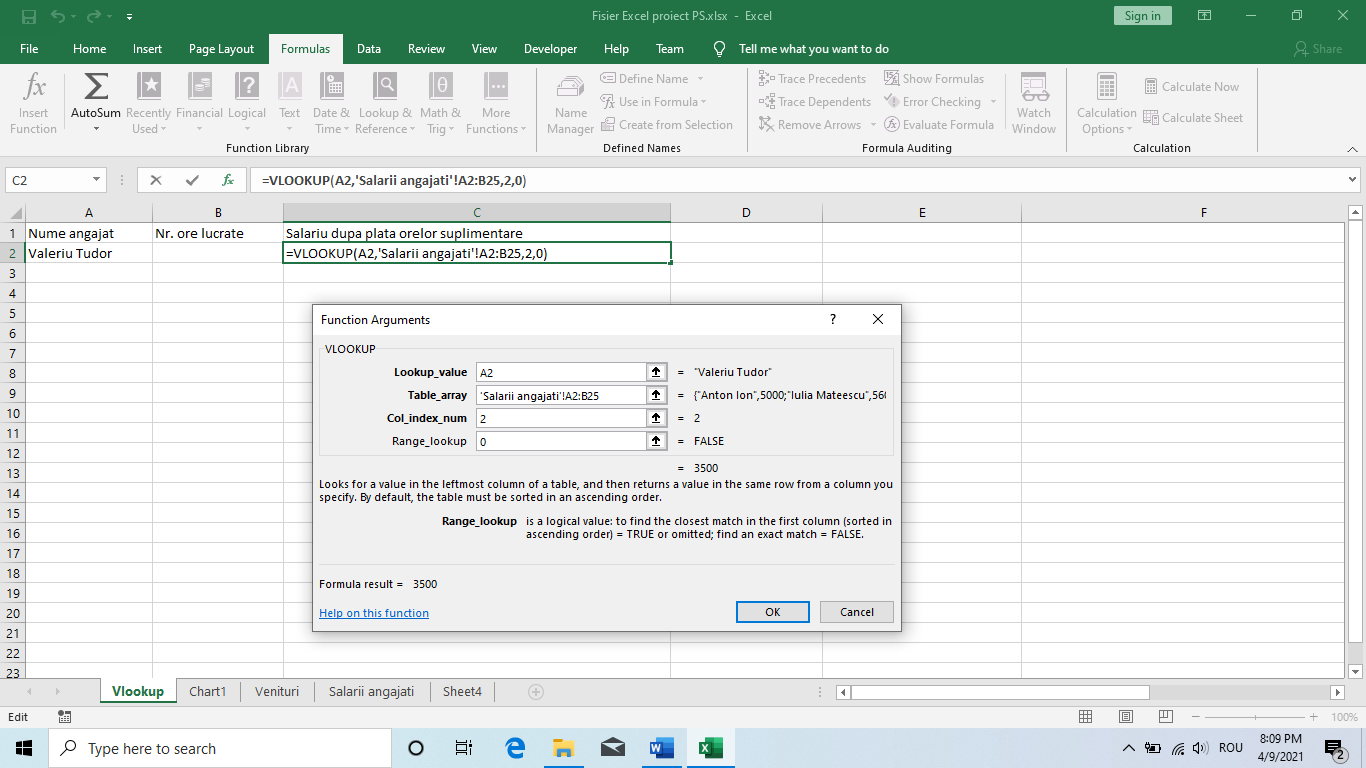


Figura 5. Aplicarea funcției de VLOOKUP

Pentru a aplica plata orelor suplimentare vom înmulți valoarea din celula B2 cu 60, această valoare adăugându-se la salariul angajatului.

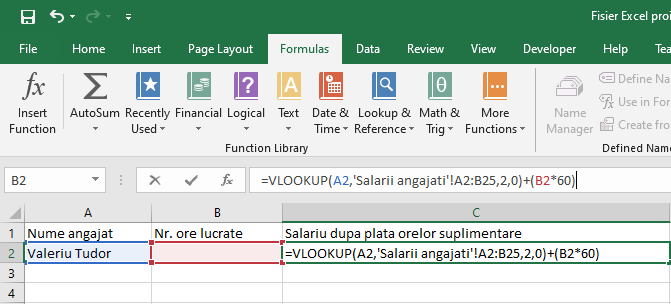


Figura 6. Calculul noului salariu după plata orelor suplimentare

1. Având în vedere contextual pandemic, firma Damen este nevoită să micșoreze suma plătită pentru o oră suplimentară, de la 60 de lei la 40 de lei. Să se realizeze un scenariu în ceea ce privește modificarea salariilor angajaților.

Vom realiza o nouă coloană unde vom introduce suma plătită de firma Damen pentru o oră suplimentară. Pentru a realiza un scenariu, mergem în Data-> Forecast-> What-If-Analysis->Scenario.

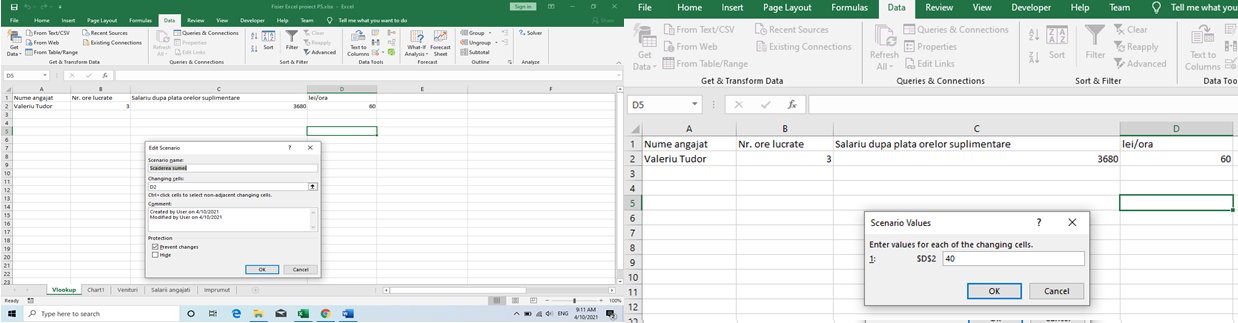


Figura 7. Implementarea scenariului

La Changing cells vom introduce celula în care se află suma plătită pentru o oră suplimentară. După ce apăsăm pe Ok vom trece valoare cu care se modifică celula D2, adică 40. După apăsarea butonului Ok, vom apăsa pe Summary. Nu schimbăm nimic la următoarea fereastră, apăsăm pe Ok. Rezultatul va fi cel din figura 8.

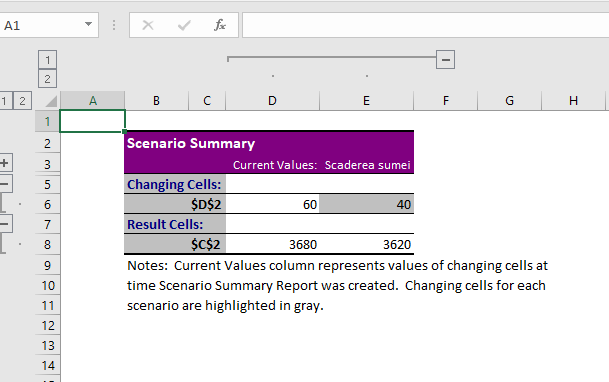


Figura 8. Output-ul final al scenariului

1. Firma Damen cumpără un utilaj de producţie, care se estimează că va aduce în următorii 5 ani un venit anual de 20.000€. Pentru finanţarea utilajului, firma are nevoie de un credit. Banca selectată acordă credite cu dobânda de 5% pe an. Care ar trebui să fie preţul maxim al utilajului, dacă ar trebui să fie plătit numai din veniturile pe care le aduce? Cheltuielile de întreţinere nu sunt luate în considerare. Plata către bancă se face în fiecare lună, la începutul lunii.

Pentru a afla prețul utilzajului vom folosi funcția PV(Present Value) pe care o găsim în Formulas-> Financial-> PV.

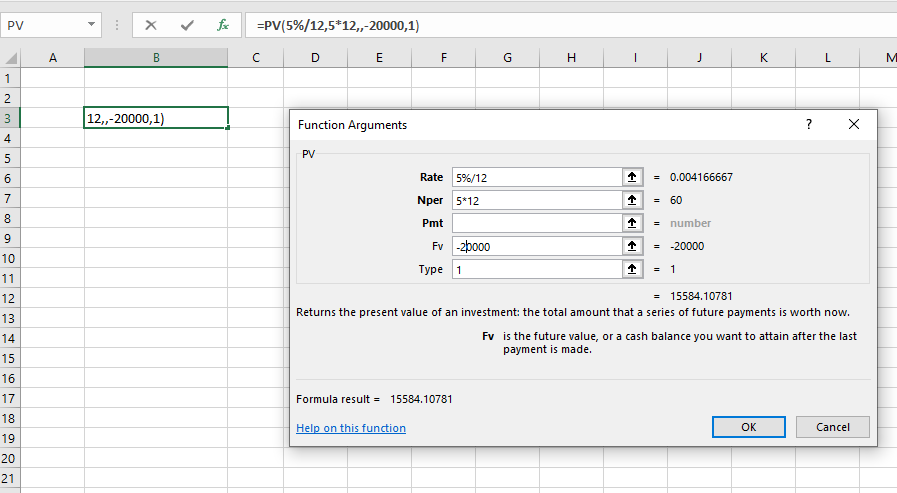


Figura 9. Aplicarea funcției PV

Parametrii au următoarele semnificții:

* Rate este rata dobânzii. În cazul noustru rata anuală de 5% va fi împărțită la cele 12 luni din an
* Nper reprezintă numărul de perioade. Creditul va fi dat pe 5 ani iar plata se face lunar, deci vom înmulți numărul de ani cu numărul de luni.
* Pmt reprezintă suma care va fi plătită lunar. În cazul nostru vom furniza Excel-ului suma pe care dorim să o obținem în viitor, deci nu vom mai trece Pmt-ul.
* Fv reprezintă suma pe care vrem să o obținem în viitor de pe urma investiției, adică 20000€.
* Type indică la ce moment al lunii se va face plata. În cazul nostru plata se face la începutul lunii, deci valoarea introdusă va fi 1.

# SAS Studio

1. Firma Damen dorește să aibă acces la fișierul cu comenzile pe care le are in perioada martie- aprilie 2021. Pentru a realiza acest lucru se apelează la SAS Studio. Să se creeze un set de date în SAS Studio folosindu-se fișierul extern comenzi.txt al firmei Damen. Să se analizeze distribuția datelor pentru variabila Preț.

Pentru rezolvarea cerinței vom folosi următorul script:

proc format;

value dateLipsaNumeric . ="Valoare lipsa";

run;

data comenziDamen;

infile "/home/u58189391/Damen/comenzi.txt" dlm=',';

length Client $32;

length NumarBucati $2;

length Tip $31;

input Client $ NumarBucati Tip $ DataLivrarii ddmmyy10. Pret ;

proc print data=comenziDamen;

format DataLivrarii ddmmyyp10.;

format Pret dateLipsaNumeric.;

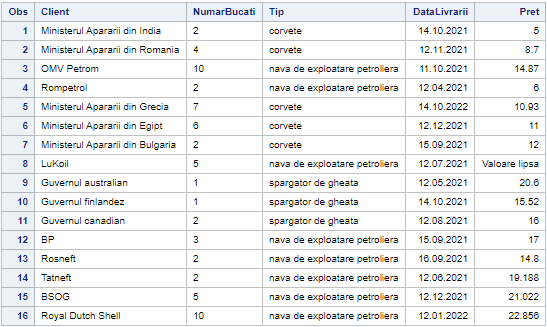
run;

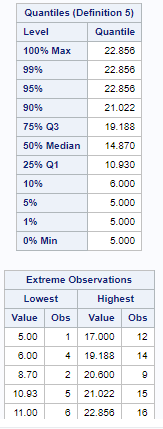
proc univariate data=comenzidamen;

var Pret;

histogram Pret ;

run;





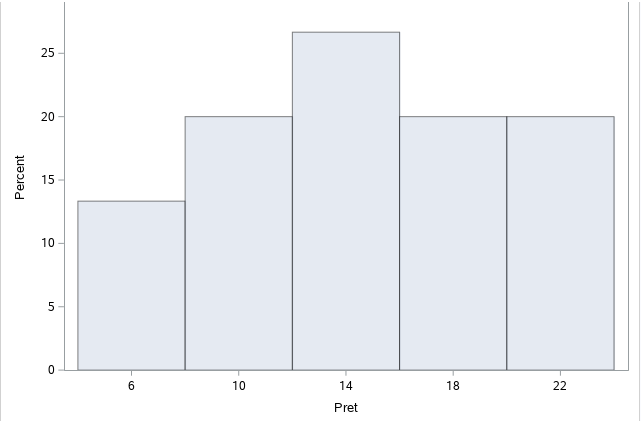


Figura 10. Distribuția datelor pentru variabila Preț după rularea scriptului

În script am preluat datele din fișierul comenzi.txt pe care l-am încărcat pe server-ul de la SAS. Preluarea s-a realizat folosind comanda INFILE. Mai departe, am setat lungimea șirului de caractere pentru coloanele de tip caracter, respectiv cât din valoarea coloanelor tip numeric va apărea în raport. Comanda INPUT va afișa capul de tabel. Pentru coloanele DataLivrarii și Pret aplicam formatele ddmmyyp10. și dateLipsaNumeric.. Ulterior folosim procedura UNIVARIATE pentru a realiza calcule statistice pentru variabila Pret. Aceste calcule pot fi vizibile în figura 10. Observăm că, în medie, prețul unei nave este de 14.37 milioane euro. Valorile din setul de date se abat de la această valoare cu 5.4 milioane euro. În ceea ce privește forma distribuției, avem o formă asimetrică negativă( Skewness este negativ, având valoarea de -0.2066278), iar indicele de boltire este platicurtic (Kurtosis este negativ, având valoarea de -0.7750923). În tabela Quantile ne sunt prezentate următoarele aspecte: prețul minim al unei nave este de 5 milioane euro, 25% din prețurile din setul de date sunt mai mici de 10.930 milioane euro, 50% din prețurile din setul de date sunt mai mici de 14.870 milioane euro, 75% din prețurile din setul de date sunt mai mici de 19.188 milioane euro, iar prețul maxim al unei nave este de 22.856 milioane de euro. Tabela Extreme Observations arată observațiile care au valorile cele mai mici, respectiv observațiile care au valorile cele mai mari.

Dacă analizăm graficul distribuției prețurilor observăm că peste 25% dintre prețuri au valori mai mari de 14 milioane de euro, 20% au valori de până la 10 milioane de euro, 20% au valori de până la 18 milioane de euro, 20% au valori de până la 22 milioane de euro, iar sub 15% au valori de până la 6 milioane euro.

1. Să se realizeze un grafic circular în care sa se arate frecvența tipurilor de nave din fișierul comenzi.txt.

proc gchart data=comenzidamen;

donut Tip;

run;

quit;

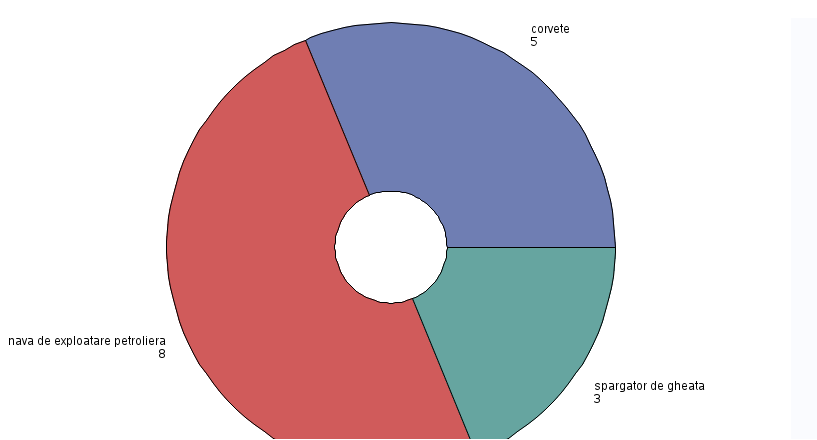


Figura 11. Frecvența tipurilor de nave livrate

În urma analizării graficului din figura 11 putem observa faptul că firma Damen a livrat în anul 2020 8 nave de exploatare petrolieră, 5 corvete și 3 spărgătoare de gheață.

1. Firma Damen dorește să afle ce proiecte pot fi amânate. Să se realizeze o tabelă în SAS Studio în care să se afișeze aceste proiecte și zilele rămase la dispoziție.

data comenziUrgente;

mn=today();

infile "/home/u58189391/Damen/comenzi.txt" dlm=',';

length Client $35;

length NumarBucati $2;

length Tip $31;

input Client $ NumarBucati Tip $ DataLivrarii ddmmyy10. Pret;

if DataLivrarii gt mn;

proc sql;

alter table comenziUrgente add NumarZileRamase numeric;

update comenziUrgente set NumarZileRamase=DataLivrarii-mn;

quit;

proc print data=comenziUrgente;

format DataLivrarii ddmmyyp10.;

format Pret dateLipsaNumeric.;

run;

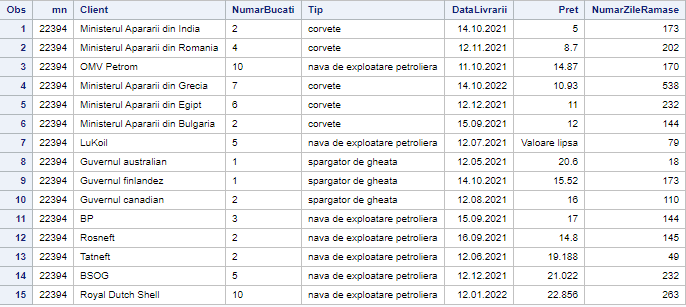


Figura 12. Proiecte care pot fi amânate de firma Damen

Ne vom folosi de același fișier ca la exercițiul 1. Diferența o va face variabila mn care va fi egală cu numărul de zile scurse de la data de 1 ianuarie 1960 până în prezent. De asemenea, comanda INPUT va fi însoțită și de clauza IF pentru a afișa valorile mai mari decât mn. Vom folosi procedura SQL pentru a putea adăuga coloana NumarZileRamase unde va fi afișată numărul de zile rămase până la data scadentă.

1. Având în vedere faptul că prețul aluminiului a crescut ca urmare a pandemiei de Coronavirus, să se previzioneze prețul navelor peste un an, știind că în anul 2020, prețul aluminiului la finalul anului a fost de 2520 de euro, iar creșterea anuală este de 7%.

data previziuneCostProductieNave;

infile "/home/u58189391/Damen/previziuneCostProductieNaveVechi.txt" dlm=',';

length DenumireNava $27;

input DenumireNava $ PretActual CantitateAluminiuTone AlteCosturi CostAluminiu 10 PretPreconizat 10;

do luni=1 to 11;

crestereLunara=2520\*(1.07/12);

PretAluminiuPeTona=2520+crestereLunara;

CostAluminiu=CantitateAluminiuTone\*PretAluminiuPeTona;

PretPreconizat=PretActual+CostAluminiu+AlteCosturi;

end;

proc print data=previziuneCostProductieNave;

format crestereLunara 6.2;

run;



Figura 13. Output-ul analizei prețului navelor peste 1 an în contextul creșterii prețului aluminiului

Pentru acest exercițiu vom folosi fișierul text previziuneCostProductieNaveVechi.txt unde avem detalii despre compunerea prețului navelor. Coloanele luni, crestereLunara, CostAluminiu, PretPreconizat au fost generate în urma utilizării instrucțiunii DO. Coloana luni arată de câte ori a fost parcursă această instrucțiune. Variabila crestereLunara arată cu cât crește prețul aluminiului. Această valoare va fi adăugată la valoarea de 2520 euro din coloana PretAluminiuPeTona. Coloana CostAluminiu va avea valori ce vor fi egale cu produsul dintre valoarea din coloana CantitateAluminiuTone și valoarea din coloana PretAluminiuPeTona. Prețul preconizat pentru o navă va fi suma dintre valorile din coloanele PretActual, CostAluminiu și AlteCosturi.

1. Pentru a încuraja angajații să realizeze ore suplimentare, firma Damen dorește să le ofere angajaților care au realizat ore suplimentare o primă de 150 RON. Să se aplice această primă pentru angajații care au efectuat mai mult de 240 de ore în luna martie.

data situatieAngajati;

infile "/home/u58189391/Damen/situatieAngajati.txt" dlm=",";

length NumeAngajat $21;

input NumeAngajat $ NumarOreLucrate Salariu;

if NumarOreLucrate gt 240 then

do;

Salariu+150;

end;

proc print data=situatieAngajati;

run;





Figura 14. Salariile angajaților după aplicarea primei

Vom folosi fișierul situatieAngajati.txt. Folosind Instrucțiunea IF înainte de instrucțiunea DO spunem SAS-ului să aplice calculul din instrucțiunea DO doar dacă numărul de ore lucrate de angajatul respectiv este strict mai mare de 240, adică să fi realizat ore suplimentare.

1. Firma Damen dorește să afle detalii privind angajații săi. Aceste detalii se referă la distanța parcursă de la domiciliu la locul de muncă, vârsta și anii petrecuți pe piața forței de muncă. Să se realizeze o tabelă în care să se arate explicit dacă trebuie să se deconteze transportul sau nu, dacă angajatul respectiv se apropie de pensionare și trebuie căutat un înlocuitor și dacă trebuie să se aplice spor de vechime la salariu.

proc format;

value naveta low-20="Nu i se deconteaza transportul"

20-high="I se deconteaza transportul";

run;

proc format;

value varsta low-60="Nu se apropie de pensionare"

60-high="Se apropie de pensionare. Trebuie cautat un inlocuitor";

run;

proc format;

value vechime low-15="Nu i se aplica spor de vechime la salariul actual"

15-high="Se aplica spor de vechime la salariul actual";

run;

data detaliiAngajati;

infile "/home/u58189391/Damen/detaliiAngajati.txt" dlm=",";

length NumeAngajat $21;

input NumeAngajat $ DataNasterii ddmmyy10. DistantaDomiciliuLocDeMunca DataAngajarii ddmmyy10.;

do;

Varsta=YRDIF(DataNasterii,TODAY(),"ACT/ACT");

Vechime=YRDIF(DataAngajarii,TODAY(),"ACT/ACT");

end;

proc print data=detaliiangajati;

format DataAngajarii ddmmyyp10.;

format DataNasterii ddmmyyp10.;

format DistantaDomiciliuLocDeMunca naveta.;

format Varsta varsta.;

format Vechime vechime.;

run;

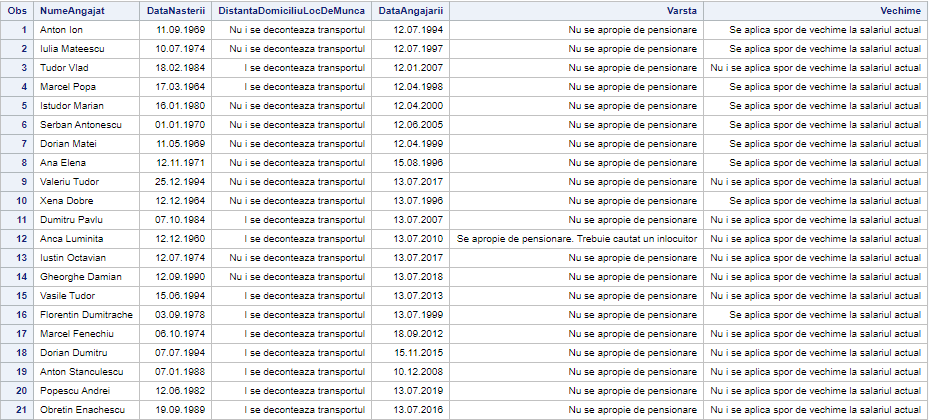


Figura 15. Tabelă situație angajați

Pentru a arăta explicit detaliile referitoare la angajați vom folosi formate predefinite. Acestea trebuie rulate împreună cu setul de date când intrăm pentru prima dată în SAS Studio deoarece durata de viață a unui format prestabilit este egală cu o sesiune de SAS Studio. Dacă distanța de la domiciliu de la destinație este mai mică de 20 km nu i se decontează transportul. În caz contrar se decontează transportul. Daca vârsta unui angajat este mai mare de 60 de ani atunci trebuie găsit un înlocuitor. Dacă vechimea pe piața forței de muncă este mai mare de 15 ani atunci i se va acorda un spor de vechime.

Am folosit instrucțiunea DO pentru a crea coloanele Varsta și Vechime. Valorile din aceste coloane se calculează cu ajutorul funcției YRDIF(). Parametri reprezintă valorile cu care se va face diferența.

1. Să se realizeze joncțiunea dintre seturile de date situatieAngajati și detaliiAngajati, atât prin procedura SAS cât și prin procedura SQL. Afișarea se va realiza în funcție de mărimea salariului.

proc sort data=situatieAngajati;

by NumeAngajat;

proc sort data=detaliiAngajati;

by NumeAngajat;

data angajati;

merge situatieAngajati detaliiAngajati;

by NumeAngajat;

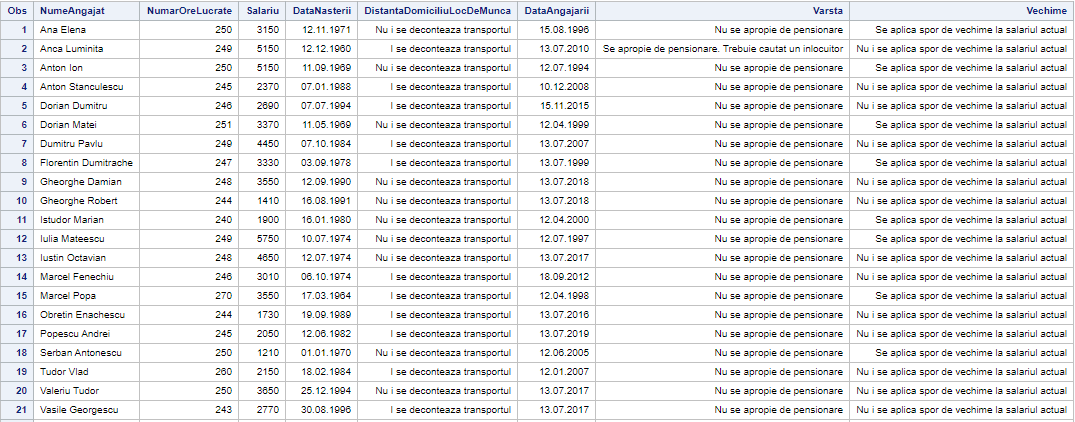
run;

proc sql;

create table angajatiSQL as select \* from situatieAngajati inner join detaliiAngajati on

situatieAngajati.NumeAngajat=detaliiAngajati.NumeAngajat order by situatieAngajati.Salariu;

quit;



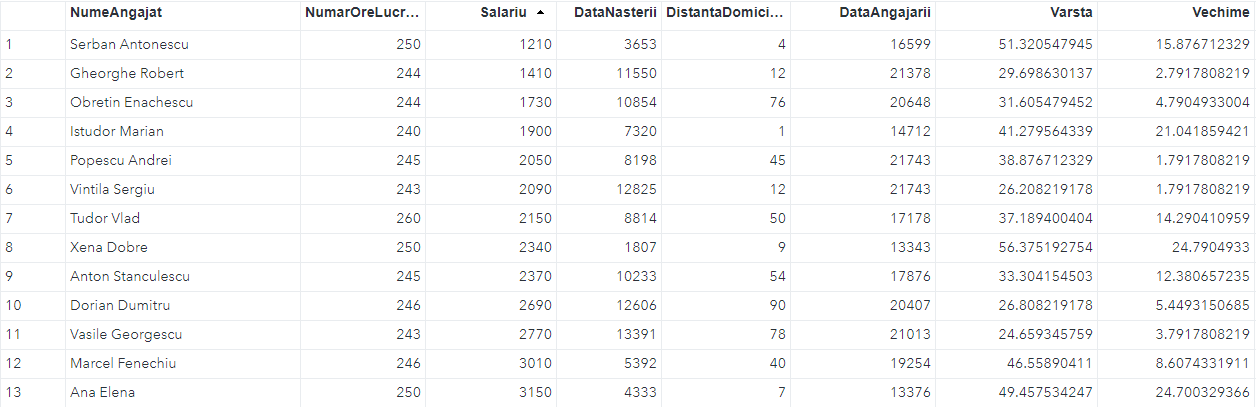


Figura 16. Situațiile și detaliile angajaților folosind SAS și SQL

Rolul joncțiunii este de a uni două tabele după o coloană care se regăsește în ambele tabele. Pentru a realiza acest lucru în SAS mai întâi trebuie să sortăm datele din ambele tabele după coloana comună. Instrucțiunea MERGE este echivalentul instrucțiunii INNER JOIN din SQL. Pentru a folosi joncțiunea în SQL vom folosi procedura cu același nume.

1. Firma Damen dorește să analizeze dacă forța de muncă influențează în mod semnificativ cifra de afaceri. Realizați această analiză.

data evolutieCifraDeAfaceri;

infile "/home/u58189391/Damen/evolutieCifraDeAfaceri.txt" dlm=",";

input An CifraDeAfaceri NumarAngajati;

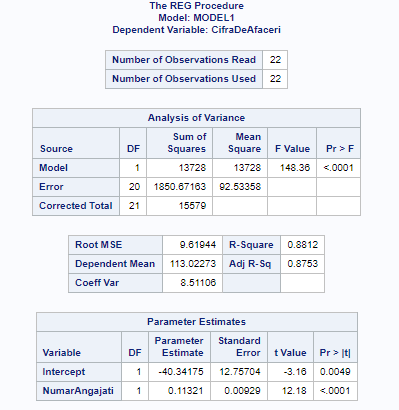
run;

proc reg data=evolutiecifradeafaceri;

model CifraDeAfaceri=NumarAngajati;

run;

quit;



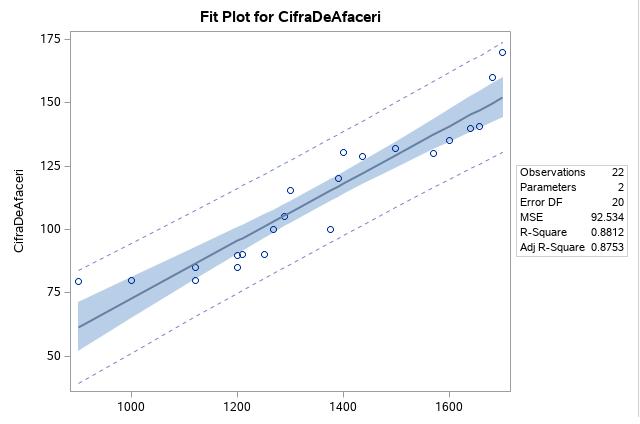


Figura 17. Output-ul pentru modelul relației cauză- efect dintre cifra de afacei și forța de muncă

În urma vizualizării tabelului și graficului din figura x3 constatăm faptul că modelul este unul semnificativ din punct de vedere statistic, întrucât probabilitatea testului Fischer este mai mică decât nivelul de semnificație α=0.05.

În urma analizării tabelului testului t-Student valoarea critică este de 1.722. Având în vedere faptul că t calculat este de 12.18, putem spune că variabila independentă NumarAngajati este semnificativă din punct de vedere statistic.

R-Square indică cât influențează variabila independent variabila dependent. Având în vedere faptul că R-Square= 0.8812 putem spune că variabila NumarAngajati cauzează în proporție de 88.12% modificarea cifrei de afaceri, restul de până la 100% fiind cauzate de alți factori care nu au fost luați în calcul.

# SAS ENTERPRISE GUIDE

1. Să se importe fișierul cliențiDamen.txt.

Deschidem SAS Enterprise Guide. În stânga sus găsim opțiunea File->New->Project. După crearea proiectului mergem pe File->Import Data și selectăm fișierul cliențiDamen.txt. Ne va apărea o nouă fereastră. La un moment dat, va apărea Text format. Alegem Delimited fields unde vom selecta semnul “,”.

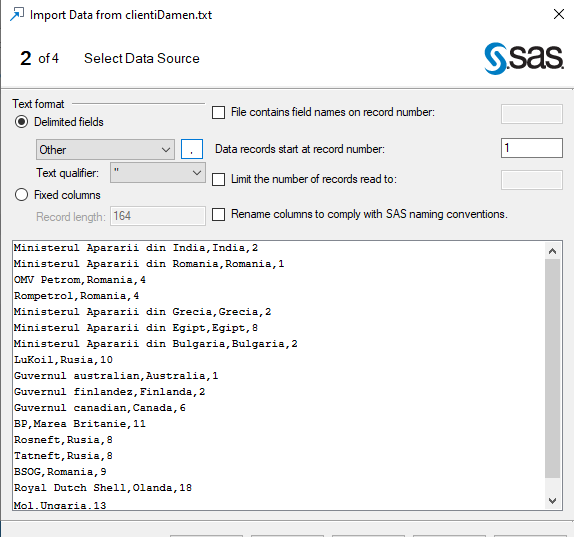


Figura 18.

Pasul următor îl constituie redenumirea coloanelor din tabel. Pentru asta selectăm coloana și apăsăm pe Modify. După ce am efectuat modificările apăsăm pe Ok. Facem acesastă operațiune pentru fiecare coloană în parte. Tabela SAS generată este vizibilă în tab-ul Process Flow.

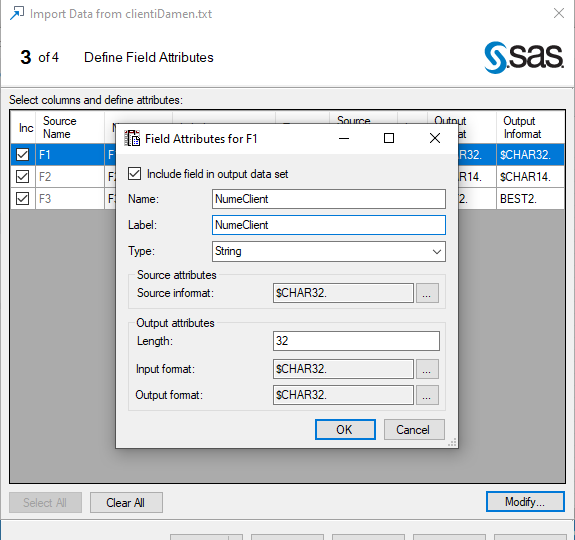


Figura 19.

1. Firma Damen dorește să aibă afișată situația clienților sai după fidelitatea acestora. Un client este considerat fidel dacă a făcut cel puțin 5 comenzi în ultimul an.

Selectăm tabela creată la exercițiul 1, mergem la tab-ul Taks->Browse->Data->Create format. Ne va apărea o fereastră ca cea din figura de mai jos. Îi vom specifica numele formatului și tipul numeric.

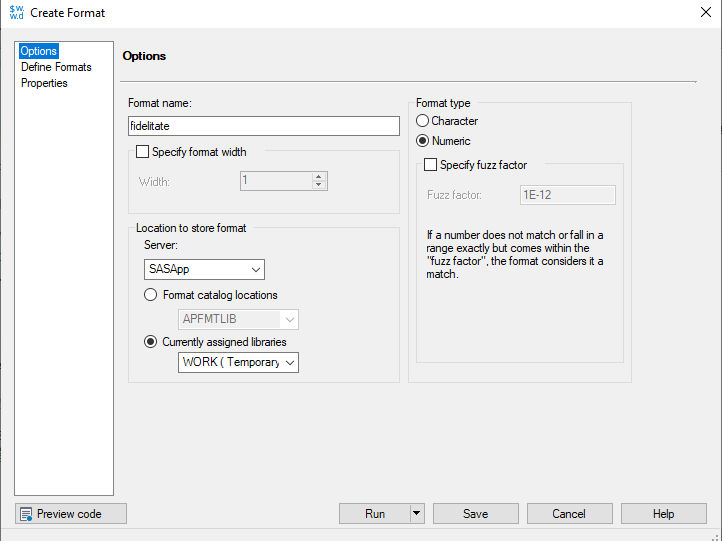


Figura 20.

După aceste specificări vom apăsa pe Define Formats care se găsește în partea stângă a ferestrei. Aici vom spune SAS-ului ce etichete să afișeze în funcție de criteriul specificat în cerință. În prima căsuță din mijlocul ferestrei vom specifica numele etichetelor, iar în cea de-a doua căsuță criteriile după care se va face afișarea etichetelor din prima căsuță.

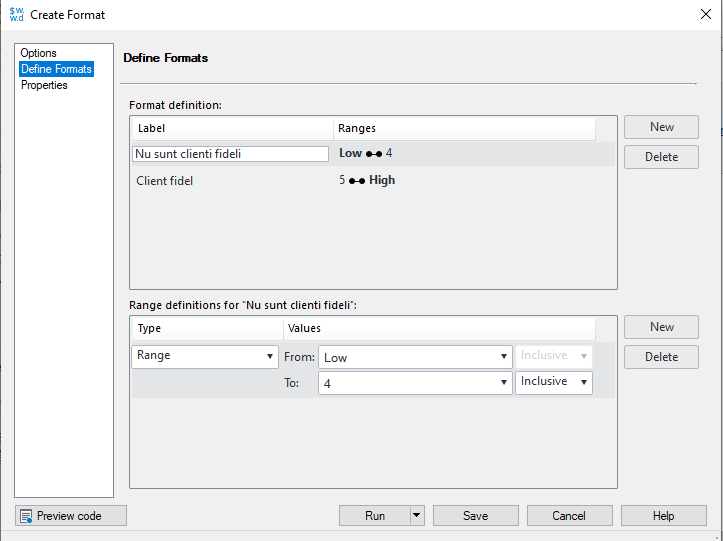


Figura 21.

1. Să se afișeze clienții care au sediul în România.

Click dreapta pe tabela SAS creată la exercițiul 1 și apăsăm pe opțiunea Query Builder. Ni se va deschide o fereastră ca cea din figura de mai jos.

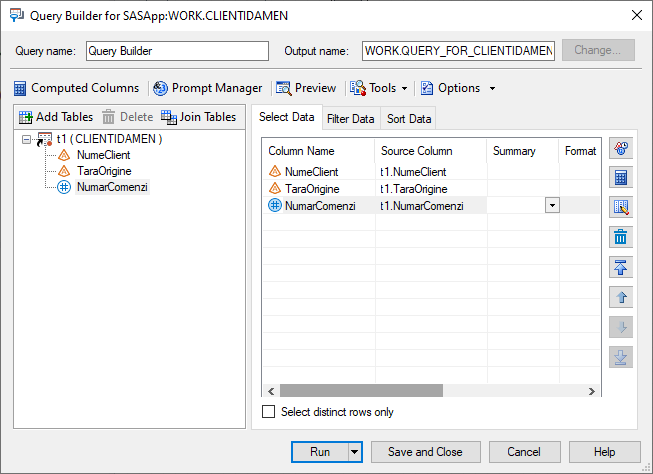


Figura 22.

Luăm din stânga cu drag&drop câmpurile care vrem să fie afișate în urma realizării query-ului. Pe tab-ul Filter Data vom lua cu drag&drop câmpul după care facem query-ul. Ni se va deschide o fereastră precum cea din figura de mai jos. Această fereastră este echivalentul unui WHERE dintr-o instrucțiune SQL.

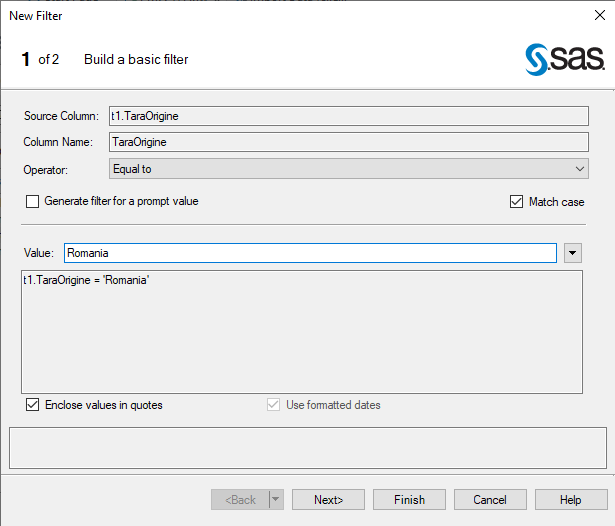


Figura 23. Fereastră setare parametru

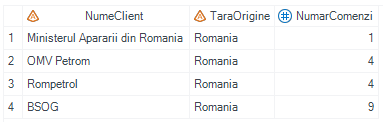


Figura 24. Output-ul query-ului

1. Să se creeze o tabelă care să cuprindă informațiile din fișierele clientiDamen.txt și din creanteClienți.txt.

Începem prin a crea o tabelă bazată pe informațiile din fișierul creanteClienți.txt la fel cum am făcut pentru fișierul clientiDamen.txt la exercițiul 1.

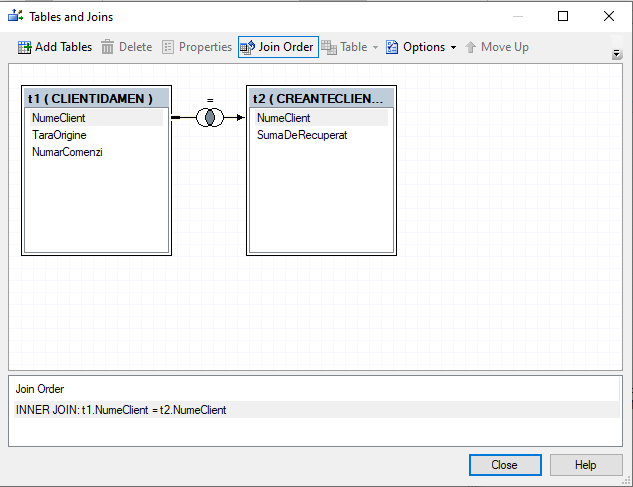


Figura 25. Joncțiunea dintre tabelele create anterior

Apăsăm pe butonul Close. Nu uităm să luăm cu drag&drop câmpurile pe care vrem să le afișeze SAS-ul. Apăsăm pe Run.

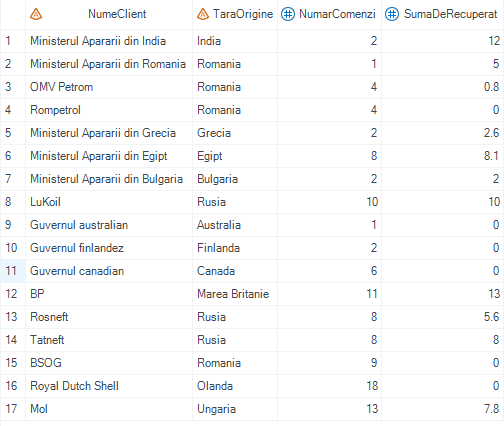
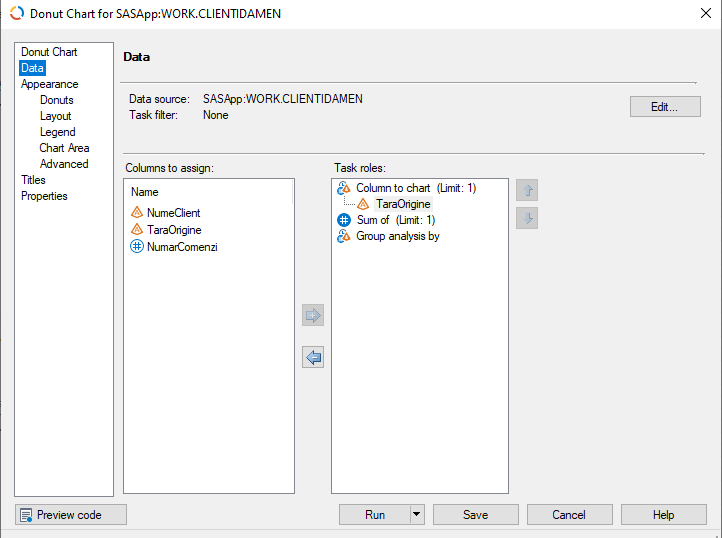


Figura 26. Output joncțiune

1. Să se reprezinte grafic repartizarea procentuală a comenzilor pe țări.

Selectăm tabela cu clienții firmei Damen, apăsăm pe Tasks->Graph->Donut Chart. Va apărea o fereastră precum cea din figura de mai jos. La Appearenced vom selecta percentage. La Data vom lua cu drag&drop câmpul TaraDeOrigine.



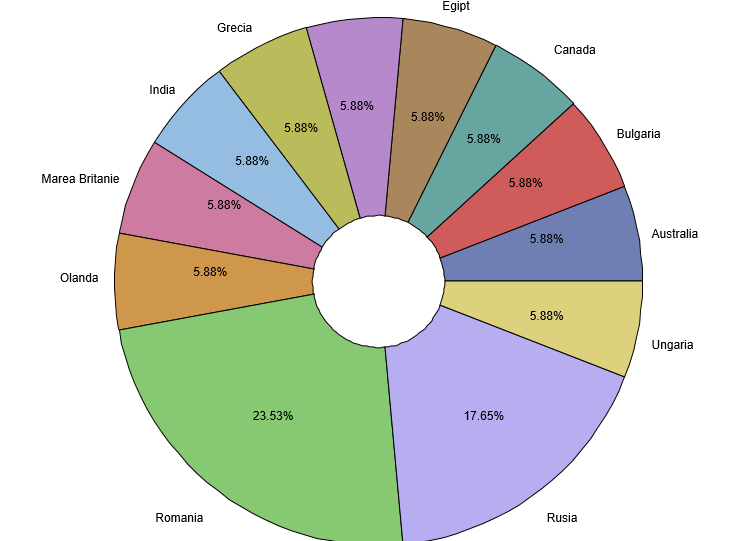


Figura 27. Repartizarea comenzilor pe țări

Observăm că cele mai multe comenzi sunt din partea firmelor din România(23,53%), urmate de cele din Rusia(17.65%). Restul comenzilor au o repartizare uniformă, de 5,88% din totalul comenzilor.

1. Firma Damen dorește să-și stabilească o strategie de extindere în anumite țări, respectiv de restrângere sau chiar părăsire a pieței unei animite țări. Să se realizeze o statistică pentru a ajuta firma Damen să ia o decizie în acest sens.

Creăm o nouă tabelă cu date din fișierul situatieSucursale.txt. Odată creată tabela o selectăm, mergem în Tasks->Describe-> Summary Statistics. Variabila analizată va fi variabila Profit, iar variabila de clasificare va fi tare de origine a sucursalelor. Apăsăm pe Run.

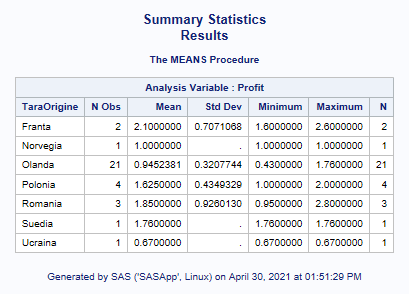


Figura 28.

Analizând figura de mai sus, observăm că sucursalele din Franța și Suedia, deși sunt puține la număr, aduc un profit considerabil, ceea ce va determina firma Damen să se dezvolte pe aceste piețe.

1. Firma Damen dorește să aibă raportul privind situația sucursalelor în format pdf. De asemenea, se dorește realizarea unui raport compus ce va conține raportul despre clienți și raportul despre sucursalele firmei Damen.

Selectăm Summary Statistics, după care click dreapta pe Properties. Ne va apărea fereastra din figura 29.

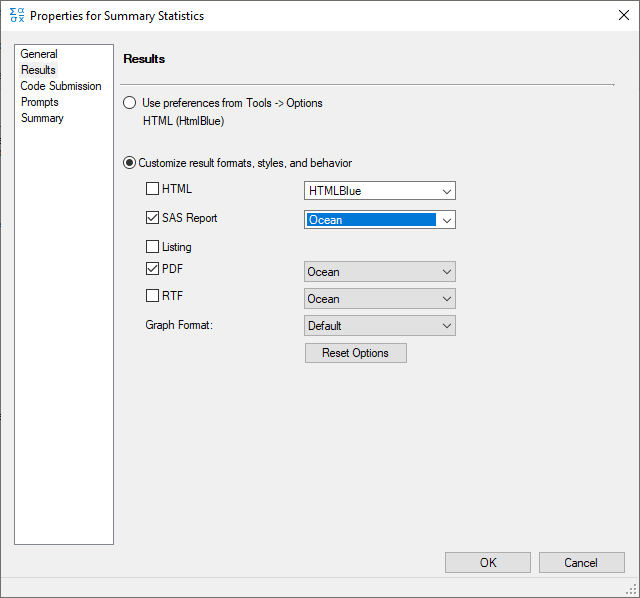


Figura 29

Selectăm opțiunea Results din partea stângă a ferestrei. Selectăm la fel ca în figura 29.

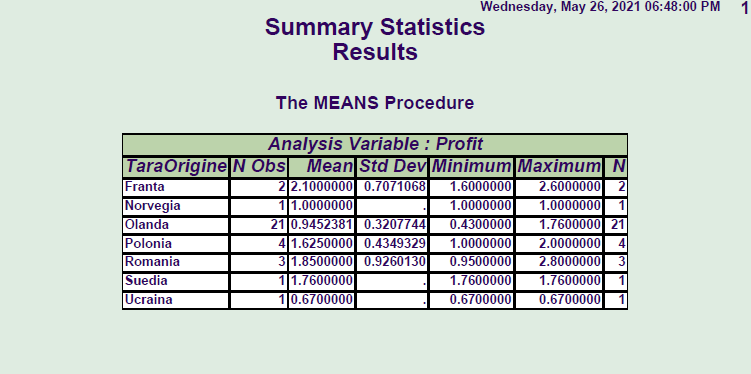


Figura 30. Raportul în format PDF

Pentru a realiza un raport compus mergem pa File->New->Report. Ne va apărea fereastra din figura 31. Nu uităm să aplicăm pentru Lista de clienți ce am aplicat pentru Summary Statistics.

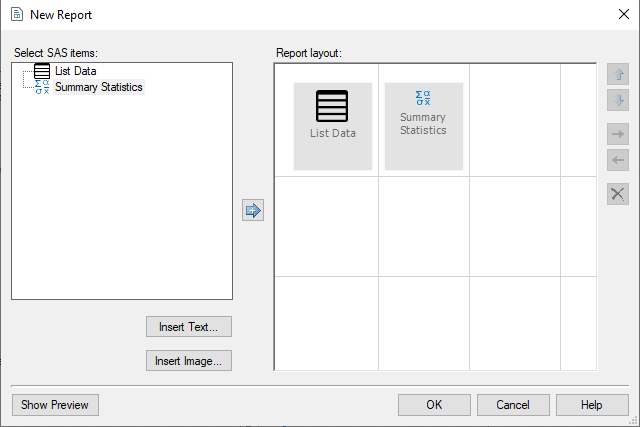


Figura 31

Apăsăm pe Ok. Raportul compus va arăta la fel ca cel din figura 32.

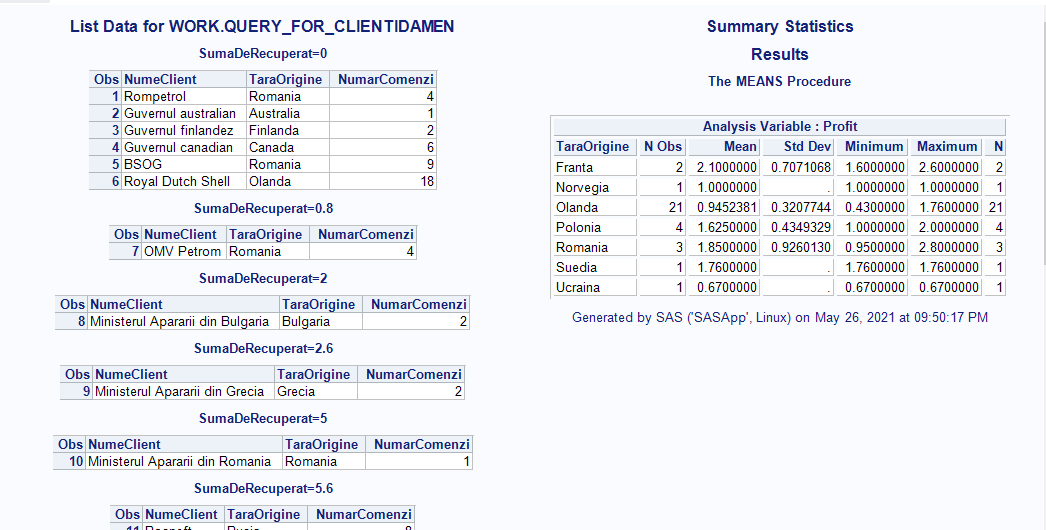


Figura 32.Raportul compus din raportul despre clienți și raportul despre sucursalele firmei Damen

# PYTHON

1. Să se citească din fișierul furnizori.csv. Datele din fișier vor fi transformate din string în date. Pentru valorile lipsă din coloana Nr\_clienti\_multumiti\_de\_oferta se va afișa mesajul "Nu se cunoaste numarul de clienti"

**import** pandas **as** pd

**import** numpy **as** np

fisier=pd.read\_csv(**"furnizori.csv"**, delimiter=**","**)  
fisier[**"Data\_afisare\_oferta"**]=pd.to\_datetime(fisier[**"Data\_afisare\_oferta"**])  
fisier[**"Nr\_clienti\_multumiti\_de\_oferta"**]=fisier[**"Nr\_clienti\_multumiti\_de\_oferta"**].replace(np.nan,**"Nu se cunoaste numarul de clienti"**)  
print(fisier)

În primul rând vom importa bibliotecile pandas și numpy. Citirea fisierului o vom face prin metoda read\_csv() care va avea ca parametrii numele fișierului, respectiv simbolul care delimiteaza datele din fisier.

Pentru conversia din string în date a elementelor din coloana Data\_afisare\_oferta folosim metoda to\_datetime() care va primi ca parametru coloana pe care vrem sa aplicăm această conversie.

Pentru a inlocui valoarea NaN cu un string folosim metoda replace() care primeste ca parametrii valoarea ce trebuie înlocuită și stringul cu care înlocuim. Pentru a prelua valoarea NaN vom folosi din biblioteca numpy proprietatea nan. Acum putem afișa fișierul.

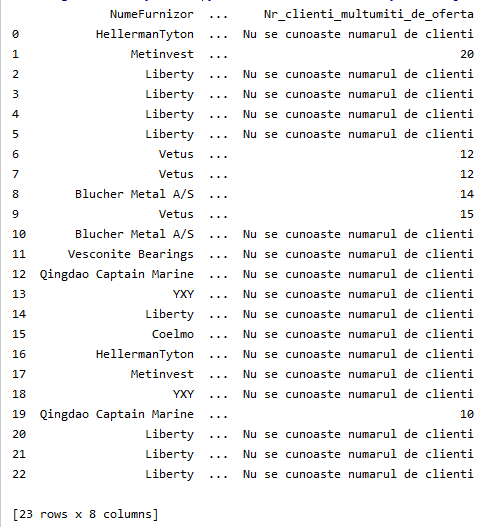


Figura 33. Fișierul furnizori.csv după executarea codului

1. Să se șteargă din DataFrame-ul fisier ofertele a căror dată nu mai este valabilă.

today=pd.to\_datetime(**"today"**)  
**for** row **in** fisier.index:  
 **if** fisier[**"Data\_afisare\_oferta"**][row]>today:  
 fisier = fisier.drop(row,axis=0)  
print(fisier)

Pentru a afla data curentă vom stoca în variabila denumită today rezultatul oferit de metoda din pandas to\_datetime() care va primi ca parametru cuvântul cheie today.

Pentru a verifica dacă data unei oferte este mai mare decât data curentă vom itera prin DataFrame-ul fisier folosind proprietatea index. Dacă fisier[eticheta\_coloana][index]> data curentă atunci aplicăm metoda drop(). Îi vom da ca parametrii indexul la care ne aflăm și modul de ștergere, adică pe linie sau pe coloană. Acest mod de parcurgere este dat de axis. Dacă este 0 parcurgem pe linie. Dacă este 1 parcurgem pe coloană.

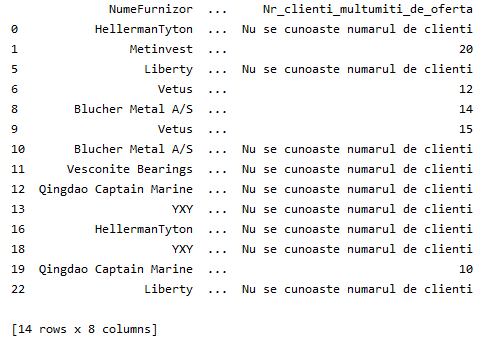


Figura 34. Ofertele cu data valabilă

1. Având în vedere că multe oferte nu au numărul de clienți mulțumiți, se cere ștergerea coloanei Nr\_clienti\_multumiti\_de\_oferta.

fisier=fisier.drop([**"Nr\_clienti\_multumiti\_de\_oferta"**],axis=1)  
print(fisier)

Din moment ce vom șterge datele dintr-o coloană, valoarea lui axis trebuie să fie 1.

1. Sucursalele firmei Damen din România au nevoie de oțel. Însă sucursalele doresc ca furnizorii să aibă puncte de lucru în România, iar prețul să nu fie mai mare de 4000 de euro. Să se afișeze furnizorii care îndeplinesc aceste criterii.

conditieTara=fisier[**'TaraOrigine'**]==**'Romania'**conditiePret=fisier[**'Pret'**]<4000  
conditieTip\_prod=fisier[**"Tip\_prod"**]==**"otel"**furnizoriOtelRomania=fisier.loc[conditieTara & conditiePret & conditieTip\_prod]  
print(furnizoriOtelRomania)

Declarăm un nou DataFrame numit furnizoriOtelRomania care va conține toate ofertele care au la TaraOrigine stringul Romania, al căror valoare din coloana Pret e mai mică de 4000 de euro, iar tipul de produs va fi otel. Pentru a realiza acest lucru vom folosi loc. Folosim loc și nu iloc întrucât ne folosim de etichetele coloanelor pentru a selecta ofertele corespunzătoare cerinței. Pentru o vizibilitate mai bună a criteriilor de selecție a ofertelor, am ales să stochez în variabilele conditieTara și conditiePret criteriile de selecție a ofertelor.

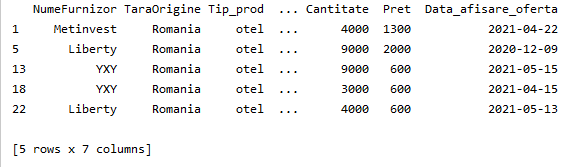


Figura 35. DataFrame-ul fisierOtelRomania

1. Pentru alegerea celei mai bune opțiuni, sucursalele firmei Damen din România vor să cunoască prețul oțelului pe metrul pătrat. Să se introducă o nouă coloană care să arate acest lucru.

fisier[**'Pret\_pe\_unitate\_de\_masura'**]=fisier[**'Pret'**]/fisier[**'Cantitate'**]  
fisierFurnizoriOtelRomania=fisier.loc[conditieTara & conditiePret & conditieTip\_prod]  
print(fisierFurnizoriOtelRomania)

Scriem numele DataFrame-ului, iar ca index punem eticheta noii coloane. Aceasta va fi egală cu raportul valorilor din coloanele Pret și Cantitate. Folosim loc la fel ca la ultimul exercițiu.

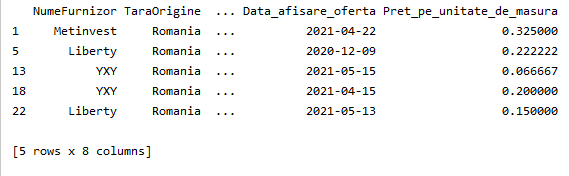


Figura 36. DataFrame-ul fisierOtelRomania actualizat

1. Pentru a alege furnizorul potrivit firma dorește să afle prețul minim, prețul maxim și prețul mediu practicat de fiecare furnizor din fișierul fisierFurnizoriOtelRomania.

fisierAnalizaOferte=fisierFurnizoriOtelRomania.groupby([**"NumeFurnizor"**]).agg({  
 **"Pret\_pe\_unitate\_de\_masura"**: [min,max,**"mean"**]  
})  
print(fisierAnalizaOferte)

Pentru afișarea detaliilor despre preț pe unitate de măsură pentru fiecare element folosim metoda groupby care va primi ca parametru eticheta coloanei după care să se grupeze, urmat de metoda agg() care va conține un dicționar unde cheia va fi eticheta coloanei pe care se vor aplica calculele min, max, respectiv mean.

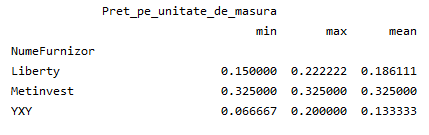


Figura 37.

1. Să se compare grafic prețul minim, maxim și mediu practicat de fiecare furnizor de oțel din Romania.

**import** matplotlib.pyplot **as** plt

fisierAnalizaOferte.plot(kind=**"bar"**)  
plt.show()

Pentru a realiza graficul trebuie să importăm biblioteca matplotlib.pyplot. Asupra DataFrame-ului vom aplica metoda plot() care are ca parametru tipul graficului.

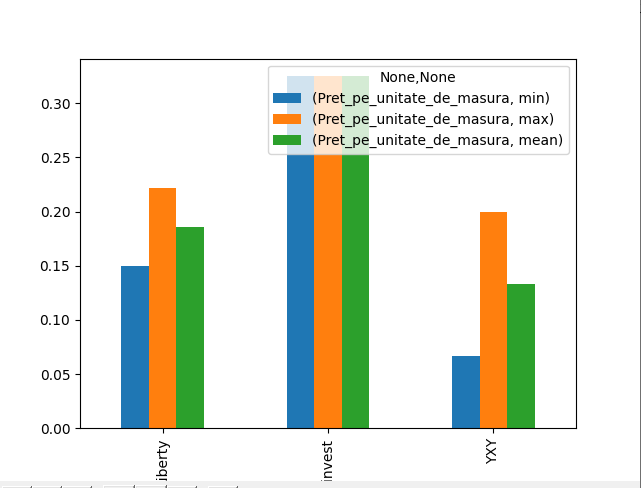


Figura 38. Grafic de comparare a ofertelor furnizorilor de oțel din România

Observăm că cel mai bun preț se poate obține din ofertele firmei YXY, în timp ce ofertele firmei MetInvest sunt cel mai puțin atrăgătoare.